CAIET DE SARCINI

SISTEME INFORMATICE PENTRU GESTIONAREA CONCEDIILOR MEDICALE ŞI A BILETELOR DE TRIMITERE ELECTRONICE, CONECTAREA LA DES A FURNIZORILOR DE SERVICII DE RECUPERARE ŞI ÎNGRIJIRI LA DOMICILIU

Intocmit:

Anca ALEXANDRESCU .....

Dan URSULEANU .....

`

Lucian MEDVES .....

Carmen ALDEA .....

Costel ANCA .....

Prescurtări şi definiţii

Termen	Explicaţie
BI	Business Intelligence
CASMB	Casa de Asigurări de Sănătate a Municipiului Bucureşti
CASAOPSNAJ	Casa Asigurărilor de Sănătate a Apărării, Ordinii Publice, Siguranţei Naţionale şi Autorităţii Judecătoreşti
CDA	Clinical Document Architecture
CEAS	(Sistemul Informatic pentru) Cardul Electronic de Asigurări de Sănătate
CID	Cod unic de identificare a asiguratului în sistemele din platforma PIAS
CAS/CJAS	Casa Judeţeană de Asigurări de Sănătate
CNAS	Casa Naţională de Asigurări de Sănătate
CNP	Codul Numeric Personal
DES	Sistemul Informatic pentru Dosarul Electronic de Sănătate
DMR	Date Medicale Relevante
ETL	Extract Transform Load
FNUASS	Fondul Naţional Unic de Asigurări Sociale de Sănătate
HIS	Hospital Information System
HL7	Health Level Seven
HL7 V3	Health Level Seven Version 3
HTTP	Hyper Text Transfer Protocol
HTTPS	Secure Hyper Text Transfer Protocol
ISO	International Organization for Standardization
ITIL	Information Technology Infrastructure Library
LDAP	Lightweight Directory Access Protocol
LOINC	Logical Observation Identifiers Names and Codes
MF	Medic de Familie
MS	Ministerul Sănătăţii
OCSP	Online Certificate Status Protocol
PIAS	Platforma Informatică a Asigurărilor de Sănătate
SAN	Storage Area Network
SI	Sistem Informatic
SIPE	Sistemul Informatic pentru Prescripţia Electronică
Sistemul	Denumire generică folosită pentru referirea sistemului

Termen	Explicaţie
	informatic pentru serviciile solicitate prin Caietul de sarcini
SIUI	Sistemul Informatic Unic Integrat al CNAS
SNOMED	Systematized Nomenclature of Human Medicine
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SSO	Single Sign On
TI	Tehnologia Informaţiei
TIC	Tehnologia Informaţiei şi a Comunicaţiei
UE	Uniunea Europeană
UI	User Interface
UPS	Uninterruptible Power Supply
UTF-8	Unicode Transformation Format 8
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WS	Web Services
WSDL	Web Service Definition Language
XDS	Cross Enterprise Clinical Document Sharing

1. Informaţii Generale

Autoritatea contractantă

1.1. Descrierea instituţiei – domeniu de activitate şi atribuţii

Casa Naţională de Asigurări de Sănătate (CNAS) este instituţie publică, autonomă, de interes naţional, cu personalitate juridică, al cărei principal obiect de activitate îl reprezintă asigurarea funcţionării unitare şi coordonate a sistemului asigurărilor sociale de sănătate din România.

Activitatea Casei Naționale de Asigurări de Sănătate presupune îndeplinirea anumitor funcții. Acestea presupun administrarea Fondului Național Unic al Asigurărilor Sociale de Sănătate (FNUASS) precum și finanțarea prin intermediul acestui fond a serviciilor medicale necesare asiguraților.

Furnizarea serviciilor medicale se face în funcție de cerere si ofertă, fapt ce asigură cadrul necesar pentru eliminarea risipei și la raționalizarea cheltuielilor. CNAS are rolul de a valorifica acest cadru, prin verificarea furnizării serviciilor medicale și farmaceutice în conformitate cu prevederile cadrului legislativ și normativ în vigoare.

Relatiile dintre medici și casele de asigurări se desfașoară în baza unui Contract-cadru în care sunt specificate criteriile cantitative și calitative de evaluare a activității medicale, în funcție de care se realizează plata medicilor pentru serviciile furnizate. CNAS are rolul de a urmări respectarea Contractului-cadru și aplicarea lui într-un mod unitar, la nivelul întregii țări.

CNAS funcţionează pe baza Statutului propriu şi are următoarele obligaţii:

- să asigure logistica funcţionării unitare şi coordonate a sistemului asigurărilor sociale de sănătate;
- să urmărească folosirea cu eficienţă a FNUASS;
- să folosească mijloace adecvate de mediatizare pentru reprezentarea, informarea şi susţinerea intereselor asiguraţilor pe care îi reprezintă;
- să acopere nevoile de servicii de sănătate ale persoanelor, în limita fondurilor disponibile

CNAS are în subordine casele judeţene de asigurări de sănătate (CJAS), Casa de Asigurări de Sănătate a Municipiului Bucureşti (CASMB) și Casa Asigurărilor de Sănătate a

Apărării, Ordinii Publice, Siguranţei Naţionale şi Autorităţii Judecătoreşti (CASAOPSNAJ).

Principalele atribuţii ale CNAS:

- asigură, supraveghează şi controlează funcţionarea sistemului de asigurări sociale de sănătate;
- gestionează FNUASS, prin intermediul preşedintelui CNAS, în concordanţă cu legile în vigoare şi având în subordine casele judeţene de asigurări de sănătate, inclusiv CASMB şi CASAOPSNAJ
- propune, cu avizul Ministerului Sănătății (MS), proiecte de acte normative pentru asigurarea funcţionării sistemului de asigurări de sănătate şi acordă avize conform proiectelor de acte normative care au incidenţă asupra FNUASS
- elaborează, implementează şi gestionează procedurile şi formularele unitare, avizate de către MS, pentru administrarea sistemului de asigurări de sănătate
- elaborează şi publică rapoarte anuale
- asigură organizarea sistemului informatic şi informaţional unic integrat pentru înregistrarea asiguraţilor şi pentru gestionarea FNUASS. Indicatorii folosiţi în raportarea datelor în sistemul de asigurări de sănătate sunt unitari şi se stabilesc de către MS la propunerea CNAS, Colegiul Medicilor din România şi Colegiul Medicilor Stomatologi din România
- aprobă anual bugetele de venituri şi cheltuieli ale caselor de asigurări de sănătate în condiţiile legii şi, după caz, cu avizul ministerelor şi al instituţiilor centrale cu reţele sanitare proprii, corespunzătoare unui plan de activităţi, precum şi obiectivele de investiţii, la propunerea acestora
- negociază criteriile privind calitatea asistenţei medicale acordate asiguraţilor din sistemul de asigurări sociale de sănătate, elaborate de Colegiul Medicilor din România
- participă la licitaţiile naţionale organizate de Ministerul Sănătăţii în vederea achiziţiei de medicamente şi materiale sanitare specifice pentru realizarea programelor de sănătate şi încheie şi derulează contracte de achiziţii publice pentru medicamente şi materiale sanitare specifice pentru realizarea acestor programe
- asigură şi controlează respectarea dreptului asiguraţilor la servicii medicale, medicamente şi materiale sanitare în mod nediscriminatoriu, în condiţiile legii
- asigură, monitorizează şi controlează modalitatea de eliberare a medicamentelor

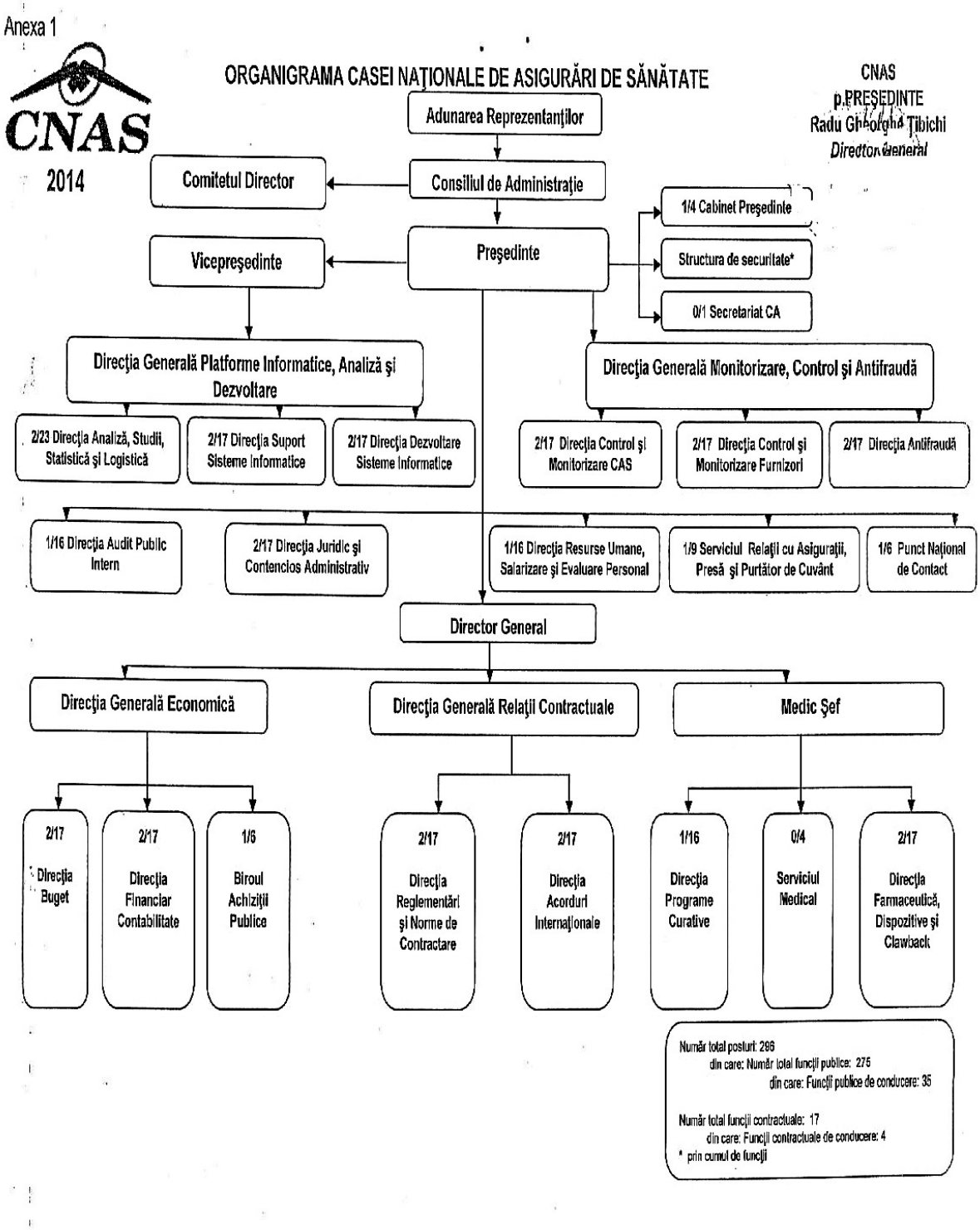
- participă la acreditarea furnizorilor de servicii medicale, de dispozitive medicale şi de medicamente, care pot fi admişi să lucreze în sistemul de asigurări sociale de sănătate
- acordă gratuit informaţii, consultanţă şi asistenţă în domeniul asigurărilor sociale de sănătate persoanelor asigurate, angajatorilor şi furnizorilor de servicii medicale.

1.2. Structura CNAS

Structura organizatorică CNAS evidenţiază următoarele compartimente funcţionale principale prin prisma proiectului, conform Organigramei de la pagina următoare:

- Preşedinte
- Vicepreşedinte
- Director General
- Direcţia Monitorizare, Control și Antifraudă
    - Direcția Control și Monitorizare CAS
    - Direcția Control și Monitorizare Furnizori
    - Direcția Antifraudă
- Direcţia Generală Platforme Informatice, Analiză și Dezvoltare
    - Direcția Analiză, Studii, Statistică şi Logistică
    - Direcţia Suport Sisteme Informatice
    - Direcția Dezvoltare Sisteme Informatice
- Direcția Audit Public Intern
- Direcţia Juridic și Contencios Administrativ
- Direcţia Resurse Umane, Salarizare și Evaluare Personal
- Direcția Relații cu Asigurații, Presă și Purtător de Cuvânt
- Medic Șef
- Direcţia Generală Relații Contractuale
    - Direcția Reglementări și Norme de Contractare
- Direcţia Generală Economică
    - Direcția Buget
    - Direcția Financiar Contabilitate
    - Direcția Achiziții Publice

Organigrama Casei Naţionale de Asigurări de Sănătate:



1.3. Justificarea necesităţii proiectului

1.3.1. Contextul legislativ şi organizaţional

Până la apariţia Legii Asigurărilor Sociale de Sănătate nr. 145/1997, sistemul de ocrotire a sănătăţii a fost coordonat în mod centralizat de către Ministerul Sănătăţii (MS), direct sau prin intermediul direcţiilor sanitare judeţene şi direcţia sanitară a municipiului Bucureşti.

Reforma sanitară din România a presupus reorganizarea serviciilor de sănătate şi a sistemului de finanţare a serviciilor de sănătate. În iulie 1997 a fost Legea Asigurărilor Sociale de Sănătate – Legea nr. 145/1997. Din ianuarie 1999, şi-au început funcţionarea, conform legii, casele judeţene de asigurări de sănătate (CJAS), ca instituţii publice autonome, conduse de reprezentanţii asiguraţilor şi patronatului prin consiliile de administraţie. A fost înființată de asemenea Casa Naţională de Asigurări de Sănătate (CNAS).

Reformele au continuat cu Legea 95/2006, intitulată „Reforma sănătăţii" care a pus baze noi cadrului instituţional din domeniul sănătăţii, stabilind printre altele: principiile sistemului de asigurări de sănătate, funcţiile şi atribuţiile CNAS, organele de conducere ale CNAS/CJAS, precum şi drepturile şi obligaţiile persoanelor asigurate. Această lege introduce şi necesitatea existenţei unui sistem informaţional şi informatic integrat pentru managementul sănătăţii publice.

În ceea ce privește obiectivele prezentului Caiet de Sarcini, legislația din Romania prevede o serie de reglementări privind modalitatea de acces la servicii medicale a pacienților asigurați, prin intermediul biletelor de trimitere. Există de asemenea un ansamblu de reglementări privitoare la acordarea de concedii medicale pentru asigurați. Ambele tipuri de documente se supun unor proceduri stricte de generare și validare conform prevederilor legislative, astfel încât să fie posibilă decontarea din FNUASS a serviciilor medicale asociate. CNAS dorește informatizarea acestor proceduri.

1.3.2. Sistemele Informatice Naționale actuale

Sistemul informatic al sănătății din România, existent în prezent la scară națională, cuprinde următoarele componente:

- Sistemul Informatic Unic Integrat (SIUI) care este sistemul informatic de bază al CNAS pentru gestionarea în condițiile legii a întregii activități de gestionare și control al utilizării FNUASS la nivelul tuturor furnizorilor de servicii medicale și farmaceutice

- Sistemul Informatic pentru Prescripţia Electronică (SIPE), care asigură gestionarea integrală a activităților privind utilizarea în condițiile legii a rețetelor medicale acoperite parțial sau integral din FNUASS. Acest sistem a fost pus în funcțiune în luna iulie 2012 și din ianuarie 2013 este folosit exclusiv pentru operarea rețetelor medicale acoperite din FNUASS
- Sistemul Informatic pentru Cardul Electronic de Asigurări de Sănătate (CEAS), care asigură gestionarea utilizării cardului electronic de sănătate, ca mijloc de identificare în sistemul medical a persoanelor asigurate și ca purtător al unor categorii de informații medicale ale asiguratului. Acest sistem este pus în funcțiune în luna decembrie 2012 și este utilizat treptat, pe măsura distribuirii către asigurați a cardurilor CEAS
- Sistemul Informatic pentru Dosarul Electronic de Sănătate (DES) - finalizat in aprilie 2014, este un instrument de gestionare a informațiilor medicale relevante pentru fiecare pacient beneficiar al serviciilor medicale suportate din FNUASS

Aceste sisteme utilizează resurse informatice comune, fiind integrate în ceea ce se numește Platforma Informatică a Asigurărilor de Sănătate (PIAS), care asigură la nivel național cadrul unitar informaţional şi strategic de gestionare a Fondului Naţional Unic de Asigurări Sociale de Sănătate (FNUASS).

Proiectul unui Sistem Informatic pentru Bilete de Trimitere Electronice şi Certificate Medicale Electronice, prin care beneficiarii şi furnizorii serviciilor medicale pot interacţiona online cu restul sistemelor naționale de sănătate, reducând documentele birocratice, reprezintă o continuare firească a implementării la nivel naţional a PIAS de către CNAS. De asemenea, o parte din cerințele acestui Caiet de Sarcini sunt referitoare la implementarea unei noi etape privind sistemul informatic DES, prin cuplarea la acest sistem a serviciilor medicale de recuperare şi de îngrijiri la domiciliu.

1.3.3. Interacţiunea cu furnizorii de servicii

Activitatea Casei Naţionale de Asigurări de Sănătate presupune îndeplinirea unor funcţii specifice domeniului asigurărilor sociale de sănătate. Acestea înseamnă, printre altele, administrarea FNUASS în scopul finanţării serviciilor medicale necesare asiguraţilor. Principial, furnizarea serviciilor medicale se face în România în funcţie de cerere şi ofertă, fapt ce conduce la eliminarea risipei şi la raţionalizarea cheltuielilor.

Realizarea serviciilor medicale catre asigurati este facuta de catre entitati externe CNAS, furnizorii de servicii medicale, farmaceutice şi dispozitive medicale. Relaţiile dintre acestia şi CJAS-uri se desfăşoară în baza unui Contract Cadru (CoCa) în care sunt specificate criteriile cantitative şi calitative de evaluare a activităţii medicale, în funcţie de care se realizează plăţile pentru serviciile furnizate (decontarea serviciilor medicale).

Casa Naţională de Asigurări de Sănătate are rolul de a urmări respectarea cadrului legal şi aplicarea lui într-un mod unitar, la nivelul întregii ţări. Această activitate este desfăşurată prin intermediul Sistemului Informatic Unic Integrat (SIUI) şi a celorlalte sisteme informatice care constituie Platforma Informatică a Asigurărilor de Sănătate (PIAS).

De asemenea, în baza principiului descentralizării, Casele de Asigurări de Sănătate județene, CASMB şi CASAOPSNAJ se bucură de autonomie în rezolvarea şi controlul aspectelor specifice ce se regăsesc la nivel local. În acest sens, au loc întâlniri frecvente între membrii Casei Naţionale de Asigurări de Sănătate şi reprezentanţii locali pentru integrarea tuturor aspectelor locale într-un cadru general, standardizat şi unitar.

În relaţia furnizorilor de servicii medicale cu Casele de Asigurări de Sănătate intervin deseori probleme legate de raportări datorate invalidării serviciilor medicale deja acordate. Aceste probleme conduc la îngreunarea procesului decontării şi indirect au repercusiuni asupra calităţii actului medical. Suspiciunile şi eventualele incoerenţe în datele operate pot fi eliminate prin accesul online al furnizorilor de servicii medicale la validările sistemului informatic.

Un obiectiv major al proiectului din acest Caiet de Sarcini (denumit *Sistemul*) este reprezentat de creşterea eficacităţii şi calităţii procesului de inter-relaţionare dintre Casele de Asigurări de Sănătate şi furnizorii de servicii medicale, datorită faptului că între aceşti actori ai sistemului de sănătate din România se derulează un spectru larg de operaţiuni specifice: informări, contractări, evaluări şi audit, raportări, decontări, raporturi administrative şi juridice. Introducerea modului de lucru online conduce la îmbunătăţirea colaborării dintre aceştia, reducând timpul de răspuns din partea angajaţilor CJAS prin reducerea numărului de cazuri semnalate datorită asistării medicilor în redactarea documentelor medicale şi validarea imediată a acestora.

Necesitatea utilizării de mijloace electronice în procesul de inter-relaţionare este impusă de creşterea numărului de informaţii vehiculate între departamentele şi angajaţii Casei Naţionale de Asigurări de Sănătate direct implicaţi în relaţia cu furnizorii, precum şi de interesul celor

din urmă de a avea acces rapid, facil şi continuu la informaţii actualizate privind legislaţia în domeniu, modalităţile de contractare şi decontare a serviciilor medicale, informaţii legate de activitatea medicală – proceduri şi protocoale terapeutice, modalităţi de evaluare, noutăţi şi evenimente, modalitatea de raportare a indicatorilor şi actualizări ale sistemului informatic al asigurărilor de sănătate.

Casa Naţională de Asigurări de Sănătate, prin departamentele implicate în relaţia cu furnizorii de servicii şi angajaţii acestora, doreşte să îndeplinească obiectivul major al *Sistemului* prin intermediul sistemelor electronice, urmărind utilizarea în cât mai mare măsură a aplicațiilor informatice dedicate relaţiei cu furnizorii de servicii medicale.

Soluţia noului *Sistem* va permite, prin intermediul secţiunilor sale dedicate, evitarea incoerenţelor şi eliminarea problemelor la decontare pentru furnizorul de servicii medicale, care vor avea acces direct la validările sistemului informatic în momentul efectuării actului medical şi nu la sfârşitul perioadei de raportare, când, se poate constata imposibilitatea de decontare a unor servicii, asociat cu pierderi pentru furnizorul acestora.

Simultan, *Sistemul* va continua direcţia deschisă de sistemul Dosarului Electronic de Sănătate de a construi suportul informatic pentru îmbunătăţirea actului medical, prin consolidarea la nivel naţional a datelor medicale ale pacienţilor, provenite de la serviciile de recuperare/reabilitare şi îngrijiri la domiciliu. Sistemul informatic propus va contribui astfel la completarea cu noi surse a datelor medicale relevante pentru dosarul medical al pacientului.

În acelaşi timp sistemul informatic propus va contribui prin utilizarea documentelor electronice la creşterea calităţii actului medical din partea furnizorilor de servicii medicale prin eliminarea erorilor determinate de lizibilitatea redusă a scrisului de mână, iar prin utilizarea semnăturii electronice şi a validării online va reduce gradul de nesiguranţă cu privire la valabilitatea documentelor medicale şi va elimina posibilitatea unor pierderi materiale din partea furnizorilor cum se întâmpla în cazul nedecontării unor servicii medicale datorată nevalidării acestora post-factum.

2. Date generale despre proiect

2.1. Obiectivul general al proiectului

Proiectul propus spre realizare (*Sistemul*) de către Casa Naţională de Asigurări de Sănătate (CNAS) are ca obiectiv principal creşterea calităţii şi eficienţei serviciilor furnizate de către CNAS, prin intermediul sistemelor informatice online, în beneficiul persoanelor asigurate şi furnizorilor de servicii medicale. Aceste sisteme informatice implică de asemenea implementarea instrumentelor de monitorizare şi de control, cu rezultate favorabile asupra scăderii costurilor actului medical, deziderat de actualitate în Romania, în special în perioada de criză a sistemului medical, perioadă caracterizată de insuficienţa fondurilor alocate pentru sănătate.

Concret, *Sistemul* va asigura implementarea de instrumente de monitorizare și control în trei tipuri de activități de mare impact atât pentru sistemul medical românesc, cât și pentru mai buna gestionare a FNUASS:

- generarea și controlul utilizării biletelor de trimitere pentru servicii medicale
- generarea și controlul concediilor medicale
- utilizarea serviciilor Dosarului Electronic de Sănătate (DES) de către furnizorii de servicii de recuperare/remediere și de îngrijiri la domiciliu.

Cu referire la domeniile de mai sus, *Sistemul* va asigura optimizarea fluxului de date şi monitorizarea electronică a calităţii serviciilor medicale decontate de către CNAS, şi va contribui în acelaşi timp la realizarea unui management intern performant. De asemenea, va fi pus la dispoziţie un pachet complet de servicii de consultare şi validare, pentru domeniile menţionate, care va facilita pentru furnizorii de servicii raportarea datelor privind serviciile medicale efectuate, în contextul legislativ al asigurărilor medicale din România.

Unul din beneficiile majore aduse de crearea şi funcţionarea serviciilor publice online este reducerea timpului de reacţie şi a costurilor aferente, ceea ce se traduce prin creşterea productivităţii la nivelul instituţiei CNAS.

Programul Operaţional Sectorial „Creşterea Competitivităţii Economice” (POS CCE) reprezintă principalul instrument pentru realizarea primei priorităţi tematice a Planului Naţional de Dezvoltare 2007-2013 – Creşterea competitivităţii economice şi dezvoltarea unei economii bazate pe cunoaştere. Obiectivul general al POS CCE este creşterea productivităţii întreprinderilor româneşti, cu asigurarea principiilor dezvoltării durabile şi reducerea

decalajelor faţă de productivitatea medie la nivelul UE. Ţinta este o creştere medie anuală a productivităţii de circa 5,5% până în anul 2015. Aceasta va permite României să atingă un nivel de aproximativ 55% din media UE.

2.2. Obiective specifice

Sistemul Informatic pentru Bilete de Trimitere Electronice şi Certificate Medicale Electronice este un instrument deosebit de important pentru entităţile implicate în sistemul de asigurări de sănătate, asigurând următoarele obiective specifice:

- *Eliminarea incertitudinilor și clarificarea* anumitor aspecte legate de funcţionarea sistemului medical și a sistemului asigurărilor de sănătate. Astfel, acordarea și mai ales utilizarea biletelor de trimitere, de asemenea acordarea concediilor medicale, sunt reglementate printr-o serie de condiții și restricții de aplicare, aceste documente (bilete de trimitere, concedii medicale) avand astfel un regim special. Documentele pentru aceste prescrieri au o circulație strict controlată, fiind documente înseriate, deoarece produc cheltuieli din FNUASS, care sunt strict reglementate. Medicii care eliberează aceste documente au nevoie de un ansamblu de informații care de multe ori sunt neclare, sau lipsesc. Sistemul va asigura astfel gestionarea biletelor de trimitere si concediilor medicale, estimate la un numar mediu lunar total de aproximativ 2.2 milioane.

- De asemenea, în evaluarea stării de sănătate a unui pacient, prin intermediul DES, medicii trebuie să aibă la dispoziție datele relevante din toate episoadele medicale ale pacientului. DES asigură în prezent doar parțial aceste date, deoarece nu permite conectarea tuturor categoriilor de furnizori de servicii medicale. De aici apare incertitudinea unei decizii medicale. Proiectul permite conectarea la DES, în scopul eliminării incertitudinilor menționate, a două noi categorii de furnizori de servicii medicale.

  Sistemele propuse pun la dispoziție toate informațile înregistrate în platforma PIAS pentru a elimina incertitudinile și a aduce clarificările necesare pentru personalul medical care se conectează la sistemele de sănătate.

- *Reducerea posibilităților de eroare medicală*, prin eliminarea scrisului de mână și prin susținerea deciziilor medicale cu date concrete din istoricul medical al pacientului

- *Sprijinirea medicilor în actul medical*, prin suport informatic şi procedural în editarea documentelor medicale şi mai ales prin susţinerea deciziilor medicale cu date concrete din istoricul medical al pacientului
- *Validarea online a documentelor medicale* este de asemenea o formă de sprijinire a medicilor în actul medical, deosebit de importantă pentru eliminarea erorilor care ar putea conduce la neplăceri atât pentru pacient (pierdere de timp în cazul unor documente completate greşit de medic), cât şi pentru furnizorul de servicii medicale (evitarea refuzului la plată de către CJAS a unor servicii medicale deja efectuate)
- *Creşterea transparenţei* privind decontarea serviciilor de sănătate din FNUASS
- *Eliminarea posibilităţilor de fraudare* a FNUASS, prin asigurarea mijloacelor de control pro-activ asupra furnizorilor de servicii medicale, oferind astfel posibilitatea de a consulta şi verifica valabilitatea serviciilor medicale înainte de efectuare şi validarea documentelor medico-legale în format electronic încă din momentul derulării procesului de emitere a lor

Prin obiectivele specifice menţionate, *Sistemul* este în conformitate cu obiectivul specific POS CCE, Axa prioritară 3, operaţiunea 3.2.4 şi anume de a pune la dispoziţie serviciile administrative prin mijloace electronice, atât pentru utilizatori (cetăţeni şi mediul de afaceri), cât şi pentru administraţia publică:

- **Axa 3** „Tehnologia Informaţiei şi Comunicaţiilor (TIC) pentru sectoarele privat şi public”
- **Domeniul Major de Intervenţie 2** „Dezvoltarea şi creşterea eficienţei serviciilor publice electronice”
- **Operaţiunea 4 „Susţinerea implementării de soluţii de e-sănătate şi asigurarea conexiunii la broadband, acolo unde este necesar”** – proiecte la nivel central.

Obiectivul principal al operaţiunii 3.2.4 este dezvoltarea şi/sau extinderea unor platforme şi aplicaţii informatice pentru sănătate (e-sănătate), disponibile online pentru cetăţeni/ mediu de afaceri/ administraţie publică, la un nivel de sofisticare de minimum 3, cu scopul prioritar de a oferi simultan următoarele beneficii:

- Îmbunătăţirea calităţii serviciilor publice prin introducerea serviciilor online furnizate utilizatorilor (asiguraţi şi furnizori de servicii medicale, farmaceutice şi de dispozitive

medicale) şi asigurarea transparenţei prin expunerea datelor legate de asigurările medicale.

Acest obiectiv este conform obiectivului principal al operaţiunii şi anume:

- furnizarea de servicii publice online către cetăţeni/mediul de afaceri/administraţie publica

- Creşterea eficienţei activităţilor interne ale CNAS prin implementarea de mijloace informatice ce vor asigura transferul bidirecţional de date între asiguraţi şi furnizorii de servicii şi echipamente medicale de o parte şi instituţia publică pe de alta parte

  Acest obiectiv este conform obiectivului principal al operaţiunii şi anume:

  - eficientizarea activităţilor interne ale instituţiei publice care contribuie la furnizarea respectivului serviciu, utilizând mijloace specifice TIC

2.3. Indicatori

INDICATORI	Valoare la începutul perioadei de implementare	Valoare la sfârşitul perioadei de implementare	Valoare la sfârşitul perioadei obligatorii de menținere a investiției (60 luni de la data finalizării proiectului)
Realizare			
Numărul instituțiilor sanitare în care s-a implementat proiectul	0	100	12.000
Număr de persoane instruite pentru folosirea aplicației informatice	0	50	50
Număr de servicii electronice disponibile prin proiect	0	4*)	4*)
Rezultat			
Numărul de utilizatori înregistrați ai proiectului, angajați ai solicitantului	0	30	150
Numărul de utilizatori ai proiectului, angajați în domeniul sanitar	0	100	12.000

[Indicatorii de realizare se monitorizează/reportează la sfârşitul perioadei de implementare]

*) Indicatorul "Număr de servicii electronice disponibile în proiect" se referă la implementarea serviciilor: funcționalități specifice biletelor de trimitere electronice; funcționalități specifice certificatelor medicale electronice; funcționalități pentru conectarea

la DES a furnizorilor de recuperare-reabilitare; funcționalități pentru conectarea la DES a furnizorilor de îngrijiri medicale la domiciliu

2.4. Beneficiile proiectului

Sistemul informatic propus îndeplineşte aceste obiective prin implementarea şi punerea la dispoziţie, prin acces securizat la internet, către diferitele grupuri ţintă a unor servicii online care vor permite:

- Pentru Casele de Asigurări de Sănătate
    - reducerea riscului de apariţie a fraudelor privind decontarea de servicii medicale de specialitate fără trimitere
    - eficientizarea validării raportărilor de servicii medicale
    - eficientizarea decontării concediilor medicale
- Pentru medicii prescriptori
    - reducerea timpului alocat activităţilor administrative în avantajul actului medical;
    - reducerea erorilor cauzate de implicarea factorului uman.
- Pentru furnizorii de servicii medicale
    - reducerea timpului alocat activităţilor administrative;
    - accesul la datele medicale relevante ale pacientului;
    - reducerea erorilor cauzate de implicarea factorului uman.
- Beneficiarii direcţi ai serviciilor de sănătate (persoanele asigurate)
    - scade numărul erorilor cauzate de interpretarea greşită a scrisului de mână
    - servicii medicale îmbunătăţite, printr-o mai bună utilizare a fondului de sănătate
    - servicii medicale îmbunătăţite, prin informarea mai bună a medicului curant referitor la istoricul medical al pacientului.

3. Descrierea tehnică a proiectului

Cerinţele prezentate în acest caiet de sarcini sunt considerate minime şi obligatorii.Oriunde în caietul de sarcini se întâlnesc specificaţii tehnice care indică o anumită origine, sursă, producţie, un procedeu special, un standard, o marcă de fabrică sau de comerţ, o licenţă de fabricaţie, acestea sunt menţionate doar pentru identificarea cu uşurinţă a tipului de produs şi nu au ca efect favorizarea sau eliminarea anumitor operatori economici sau a anumitor produse. Aceste specificaţii vor fi considerate ca având menţiunea "sau echivalent".

3.1. Cerinţele funcţionale ale sistemului

În această secţiune sunt prezentate cerinţele funcţionale minime obligatorii pe care soluţia tehnică pentru realizarea sistemului informatic trebuie să le îndeplinească, pentru ca acesta să atingă obiectivele asumate.

Sunt prezentate atât funcţionalităţile care vor fi implementate de sistemul informatic central adresate personalului de la nivelul CJAS/CNAS, pentru îndeplinirea sarcinilor de control şi validare specifice, cât şi cele care trebuie implementate de aplicaţiile de raportare ale medicilor, pentru a permite interacţiunea cu sistemul central în scopul colectării şi validării datelor.

3.1.1. Funcţionalităţi specifice biletelor de trimitere electronice

Conform reglementărilor, anumite servicii medicale pot fi decontate din FNUASS numai dacă sunt recomandate de medicii de familie sau medicii specialişti. În această categorie intră consultaţiile la medicii specialişti, diferite tipuri de analize medicale, servicii de reculerare. Recomandarea are forma tipizatului numit Bilet de Trimitere, care este un document cu statut special și circulație controlată, cu serie și număr unice. Biletul de trimitere este eliberat de medic dintr-un carnet achiziționat sub semnătură de la CJAS. Biletul este înmânat pacientului care se poate prezenta pentru efectuarea serviciilor recomandate la medicii de specialitate sau la laboratoare. Biletul de trimitere este preluat de furnizorul care execută serviciul medical și care își poate deconta din FNUASS serviciul respectiv numai în condițiile în care SIUI recunoaște biletul de trimitere respectiv. Această procedură este aplicată pentru evitarea exceselor din partea pacienților și pentru protejarea consumului fraudulos al FNUASS.

Eliberarea biletului de trimitere are două aspecte importante:

- Din punct de vedere clinic, este necesar pentru medicii care consultă istoricul unui pacient să poată evalua recomandările medicilor curanți de efectuare a unor consulturi

sau investigații medicale, acestea conturând suspiciuni medicale din istoricul pacientului. De aceea, trimiterea medicală este necesar să fie înscrisă în DES, fără însă să fie necesară înregistrarea elementelor trimiterii care țin de aspectul legal al acesteia

- Din punct de vedere al asigurărilor de sănătate, pentru validarea posibilității de plată a serviciului medical, conform reglamentărilor în vigoare. Aceste elemente sunt analizate atât la emiterea, cât și la recepționarea biletului de trimitere. Avand în vedere consumul important de fonduri din FNUASS, validat prin intermediul biletelor de trimitere, precum și necesitatea evitării cheltuielilor frauduloase, CNAS a solicitat dezvoltarea sistemului specializat, conținut în acest Caiet de Sarcini, pentru controlul și validarea biletelor de trimitere. Aspectele vizate de acest sistem sunt separate de aspectul clinic menționat la punctul anterior, care se rezolvă în sistemul DES prin reținerea datelor medicale specifice, relevante, din biletul de trimitere.

În continuarea acestei secțiuni sunt grupate şi detaliate cerinţele funcţionale specifice pentru biletele de trimitere electronice. Aceste cerințe sunt referitoare la sistemul central, care va fi integrat cu restul sistemelor din platforma PIAS, cât și la aplicațiile informatice locale, folosite de medicii care vor accesa sistemul central.

Notă: Referitor la aplicațiile utilizate local de furnizorii de servicii medicale se va face apel la paragraful 3.1.5 din prezentul document.

3.1.1.1. Generarea de plaje de serii de bilete de trimitere electronice

Seriile şi numerele biletelor de trimitere electronice sunt gestionate de sistemul informatic central. La cerere sistemul central va permite generarea de plaje de serii şi numere de bilete de trimitere unice pentru fiecare furnizor de servicii medicale în parte, funcționalitatea fiind la dispoziția unui rol special.

Pentru fiecare medic care eliberează bilete de trimitere va fi acordată automat de sistem o plajă de numere şi serii, pe baza contractului sau a convenţiei medicale încheiate între unitatea la care îşi desfăşoară activitatea şi CJAS.

Furnizorii se conectează online la sistemul informatic şi pot genera şi descărca o plajă de serii şi numere, folosită ulterior pentru a emite bilete de trimitere. Gestionarea centralizată a seriilor şi numerelor biletelor de trimitere asigură unicitatea acestora în cadrul sistemului.

La fiecare emitere de bilet de trimitere aplicaţia furnizorului generează următorul număr de bilet din plaja alocată, unicitatea fiind verificată apoi online de sistemul informatic central la transmiterea spre validare a biletului de trimitere.

3.1.1.2. Emiterea de bilete de trimitere electronice online

Medicul de familie sau medicul specialist poate emite pacientului, în cadrul consultaţiei, un bilet de trimitere electronic către o specialitate clinică sau către investigaţii paraclinice de laborator, prin intermediul aplicaţiei de raportare proprii, conform normelor CNAS.

Pacientul căruia i se eliberează biletul de trimitere electronic va fi identificat în cabinetul medicului curant prin cardul CEAS dacă acesta este disponibil. În caz contrar, sistemul trebuie să asigure identificarea pacienţilor după numărul unic de asigurat, obţinut din CNP-ul de pe actul de identitate prezentat conform procedurii pusă la dispoziţie de CNAS pentru toate sistemele constituente ale PIAS.

Această funcţionalitate va fi implementată prin aplicaţiile dedicate ale furnizorilor, folosite în prezent de către aceştia pentru conectarea la sistemele din platforma PIAS. Sistemul central trebuie să expună un set de servicii-web pentru interacţiunea cu aceste aplicaţii. Accesul la aceste servicii web permite verificarea pro-activă a calităţii de asigurat a pacienţilor, avertizând medicul asupra situațiilor în care biletul de trimitere nu îndeplinește condițiile de valabilitate.

Biletul de trimitere/recomandarea va conţine un set minim de informaţii:

- Serie, număr şi dată de emitere;
- Parafa medicului emitent;
- Casa de asigurări de sănătate cu care medicul emitent are contract;
- Codul unic de identificare al furnizorului de servicii medicale;
- Numărul unic de asigurat (CID) al pacientului;
- Codul de diagnostic medical;
- Specialitatea medicului către care este trimis pacientul spre consultaţie, pentru trimiterile către specialişti;
- Dispozitivul recomandat
- Lista serviciilor de ingrijiri la domiciliu recomandata

- Lista de investigaţii paraclinice pe care pacientul trebuie să le efectueze, pentru trimiterile către laborator;
- Semnătura electronică extinsă a medicului emitent, utilizând un certificat digital calificat;
- Semnătura electronică realizată utilizând cardul CEAS al pacientului, dacă pacientului i s-a emis un card.

Aceste informatii vor fi individualizate pentru fiecare tip de serviciu medical solicitat, conform legislatiei in vigoare care reglementeaza tipul de formular de bilet de trimitere/recomandare.

3.1.1.3. Validarea biletelor de trimitere electronice online

Odată cu încărcarea în sistemul central are loc validarea automată a biletelor de trimitere/recomandarilor online. Biletele de trimire/recomandarile vor putea fi tipărite numai după ce sistemul central a validat datele transmise şi a generat un număr unic de înregistrare al cererii de validare, care va fi şi el tipărit pe biletul de trimitere/recomandare. Pentru a se valida in sistemul informatic, biletul de trimitere/recomandare electronic trebuie să îndeplinească minim următoarele condiţii:

- Parafa medicului este valabilă la data emiterii biletului;
- Furnizorul are contract sau convenţie cu casa de asigurări valabile, şi contractul / convenţia îi permite să emită bilet de trimitere;
- Seria şi numărul biletului de trimitere/recomandare trebuie să fie unică şi să corespundă unei plaje alocate pentru furnizor;
- Pacientul trebuie să existe în baza de date a persoanelor din PIAS;
- Codul de diagnostic trebuie să fie valid;

Dacă biletul de trimitere este pentru specialităţi clinice, acesta trebuie să respecte regula:

- Specialitatea către care se trimite pacientul trebuie să fie validă.

Dacă biletul de trimitere este pentru investigaţii paraclinice, acesta va trebui să conţină cel puţin o investigaţie recomandată, iar investigaţiile trebuie să respecte următoarele reguli:

- Codul de investigaţie trebuie să fie valid;
- Se verifică competenţa medicului emitent de a recomanda un anumit cod de investigaţie.

Daca este o recomandare pentru recuperare, aceasta trebuie sa contina cel putin un serviciu, iar procedurile recomandate trebuie sa respecte urmatoarele reguli:

- Codul de procedurii trebuie să fie valid;
- Se verifică competenţa medicului emitent de a recomanda un anumit set de proceduri

Daca este o recomandare pentru dispozitive medicale, aceasta trebuie sa contina cel putin un dispozitiv, iar dispozitivul recomandat trebuie sa respecte urmatoarele reguli:

- Codul de dispozitivului trebuie să fie valid si vor fi respectare regulile aferente pentru recomandarea si procurarea acestuia;
- Se verifică competenţa medicului emitent de a recomanda un anumit dispozitiv;

Aceste reguli de validare se verifică prin intermediul serviciilor web expuse atât de noul sistem informatic cât şi prin reutilizarea unor servicii existente ale PIAS, pentru accesul, spre exemplu, la nomenclatoarele de diagnostice, specialităţi, precum şi la registrele de medici şi de asiguraţi.

Accesul la aceste servicii web vor permite validarea online pro-activă a biletelor de trimitere, eliminând necesitatea de raportare periodică şi notificând din timp medicii asupra regulilor de validare care nu au fost satisfăcute, anterior emiterii propriu-zise a biletului.

De asemenea, este necesara securizarea operatiunilor de semnare digitala, marcare temporala si verificare a semnaturii digitale asociate, astfel incat sa fie asigurate următoarele cerințe funcționale:

- Semnarea electronică folosind certificate digitale X.509v3 și validarea documentelor semnate;
- Apelarea funcțiilor de semnare de către aplicații sau utilizatori diferiți;
- Utilizarea unor chei de semnare individuale pentru fiecare aplicație sau utilizator care apelează serviciul;
- Va asigura utilizarea de algoritm RSA, de cel puțin 2048 bit;
- Va asigura utilizarea a diverși algoritmi de hashing, respectiv cel puțin, SHA-1, SHA-2;

- Va oferi funcționalități interne integrate de marcare temporala, în conformitate cu RFC 3161,
- Va oferi funcționalități de validare a valabilității certificatelor digitale cu care au fost realizate semnăturile, atât prin metode de tip catalog (CRL), cat și de tip tranzacțional (OCSP conform RFC 2560);
- Va asigura semnarea și marcarea temporala a oricărui format de documente, inclusiv documente PDF și XML;
- Va asigura semnarea de documente si fișiere generice prin mecanisme standardizate conform Cryptographic Message Syntax (CMS) - RFC 5652 asigurând crearea de semnături multiple pe același nivel și de contrasemnături;
- Va asigura crearea de semnături atașate și semnături detașate conform RFC 5652;
- Va asigura crearea de semnături integrate în formatul documentelor PDF definit conform standardului ISO 32000 1:2008;
- Va asigura semnarea de hash-uri precalculate;
- Va include interfețe standardizate pentru interacțiune controlata programatic prin Web Services, dar și în linie comanda;
- Va asigura procesare batch a cererilor de semnare;
- Va permite arhivarea opționala a documentelor semnate;
- Va asigura jurnalizarea și auditul operațiunilor adminsitrative și a celor specifice, de semnare / validare;
- Va putea fi integrat în fluxuri de solicitare și aprobare de documente certificate, respectiv prin semnare și marcare temporala, automata sau cu interventia operatorului uman;
- Va putea fi instalata în configurații de tip cluster, pentru asigurarea unui nivel înalt de disponibilitate operațională.

3.1.1.4. Utilizarea semnăturii electronice pentru biletele de trimitere electronice

Prin adăugarea semnăturii electronice şi implicit validarea online a biletului de trimitere electronic se verifică automat identitatea medicului care emite biletul de trimitere. Odată cu confirmarea de către medic a operației de editare a biletului de trimitere, biletul electronic

este trimis în sistem, declanșându-se operațiile de verificare și validare (paragraful 3.1.1.3). Medicul este eventual notificat imediat, prin intermediul aplicaţiei sale, asupra eventualelor nereguli.

Folosind pentru autentificare un certificat digital calificat (acelaşi pe care îl foloseşte pentru autentificarea în sistemele PIAS), medicul trimite în sistemul informatic biletul de trimitere electronic utilizând semnătura electronică creată pe baza aceluiaşi certificat digital.

Dacă pacientul are cardul electronic de asigurări de sănătate (CEAS) activat, acesta va fi utilizat în aplicaţia furnizorului pentru a realiza o semnătura electronică simplă utilizând certificatul digital stocat pe CIP-ul cardului.

3.1.1.5. Emiterea de bilete de trimitere electronice offline

Această funcţionalitate este necesară pentru a asigura funcţionarea sistemului în situaţii excepţionale, cum ar fi imposibilitatea accesării sistemului central din motive întemeiate. În acest caz se va permite eliberarea şi tipărirea biletului de trimitere în mod offline, fără ca acesta să fie transmis şi validat în sistemul central. Accesul la această funcţionalitate va fi limitat, prin măsuri care se vor stabili împreună cu CNAS în cursul fazei de analiză, pentru a descuraja abuzul şi tentativele de a ocoli anumite reguli de validare.

Datorită faptului că biletul de trimitere electronic offline nu comunică cu sistemul central nu pot fi făcute validările referitoare la corectitudinea elementelor componente (coduri de diagnostice şi de investigaţii, parafă şi contract medic.)

De asemenea, în lipsa unei conexiuni online, nu se poate aplica şi verifica semnătura electronică a medicului sau a pacientului, caz în care se pot aplica de către CNAS restricţii normative pentru utilizarea acestor bilete de trimitere. Totodată se va introduce obligativitatea din partea medicilor de a completa şi transmite online datele de pe aceste bilete de trimitere emise offline, în momentul în care situaţia excepţională încetează.

3.1.1.6. Generarea codurilor de bare bi-dimensionale și tipărirea biletului de trimitere

După ce biletul de trimitere a fost creat (online sau offline) medicul tipăreşte un formular al biletului şi îl înmânează pacientului. Biletul de trimitere tipărit conţine un cod de bare bi-dimensional, în care se codifică informaţiile esenţiale de pe bilet: parafa medicului emitent, specialitatea către care se trimite, codul de diagnostic precum şi lista de coduri de investigaţii paraclinice.

Pacientul se prezintă cu biletul de trimitere la medicul specialist sau la laboratorul de investigaţii paraclinice, după caz.

Medicul specialist sau cel de laborator poate identifica în sistem biletul de trimitere în mai multe moduri:

- Scanarea codului de bare bi-dimensional: daca codul de bare este valid se pot afla informaţiile esenţiale de pe bilet: parafa medicului emitent, specialitatea către care se trimite, codul de diagnostic precum şi lista de coduri de investigaţii paraclinice, după caz
- Interogarea sistemului prin intermediul codului tipărit al biletului, sau prin CID-ul pacientului. Această facilitate este posibilă dacă biletul a fost încărcat în sistem de către medicul emitent, caz în care se pot consulta toate informaţiile aferente.

Datele incluse în codul de bare pot fi utilizate de aplicaţiile medicilor pentru a completa automat informaţiile necesare de pe consultaţia sau serviciul pe care medicul îl efectuează.

Observatie: Tipărirea biletului de trimitere nu este principial necesară, în situaţia în care formatul electronic devine obligatoriu. Se poate elimina astfel necesitatea unui suport de hârtie și a consumurilor asociate. Sistemul va fi proiectat astfel încât să permită setarea opțiunilor de funcționare fără a mai fi folosit suportul de hârtie.

3.1.1.7. Verificarea biletului de trimitere la furnizorul de servicii destinatar

Această funcţionalitate permite furnizorilor de servicii medicale să verifice online existentenţa şi valabilitatea unui bilet de trimitere, anterior prestării serviciului medical. Accesul la varianta electronică a biletului de trimitere permite dezvoltarea unor funcţionalităţi de precompletare a datelor de către medicul prescriptor la nivelul aplicaţiei medicale a medicului care prestează serviciul medical. În acelaşi timp verificarea online a biletului de trimire garantează autenticitatea şi valabilitatea acestuia.

Această funcţionalitate presupune următoarele:

- Asigurarea în sistemul central a modalităţilor şi funcţiilor de identificare a biletelor de trimitere iniţiate de la nivelul furnizorilor de servicii
- actualizarea aplicaţiilor distribuite de CNAS, utilizate de furnizorii de servicii pentru raportarea în SIUI

- efectuarea modificărilor necesitate de sistemul propus în celelalte sisteme ale platformei PIAS (SIUI, Reţeta Electronică, cardul CEAS, DES).

Accesul la această funcţionalitate permite verificarea pro-activă online a existenţei şi valabilităţii unui bilet de trimitere, avertizând furnizorul de servicii medicale destinatar (medic specialist, laborator de analize, etc) asupra riscului potenţial al nedecontării unui serviciu, în cazul în care nu pot fi confirmate regulile de validare.

3.1.1.8. Transmiterea biletelor de trimitere electronice către DES

Datele din biletele de trimitere electronice sunt necesare şi pentru completarea datelor medicale gestionate în sistemul pentru Dosarul Electronic de Sănătate. Biletele de trimitere reprezintă o componentă din procesele de bază ale comunicării dintre medici, pentru facilitarea şi îmbunătățirea actelor medicale complexe.

Astfel, odată cu transmiterea datelor din biletul de trimitere către noul sistem propus, datele medicale relevante pentru aspectul clinic se vor transmite şi către sistemul pentru Dosarul Electronic de Sănătate utilizând interfeţele oferite de acest sistem şi formatul documentului clinic de tip trimitere (clinică, paraclinică) specificat în cadrul standardului de integrare DES, mai precis în ghidului de implementare HL7 CDA al sistemului DES .

În cadrul proiectului nu este necesară crearea documentului clinic CDA de tipurile ”bilet de trimitere pentru investigații clinice”, ”bilet de trimitere pentru investigații paraclinice”, ”recomandări îngrijire la domiciliu” și ”recomandări dispozitive medicale” ci se vor utiliza cele disponibile în sistemul DES.

Se vor transmite către sistemul Dosarul Electronic de Sănătate toate tipurile de bilete de trimitere, incluzând biletele de trimitere pentru analize medicale de specialitate, investigaţii imagistice şi explorări funcţionale, servicii de recuperare şi tratament şi altele.

Setul exact de date medicale ce sunt transmise către DES (numite Date Medicale Relevante - DMR) este stabilit de o comisie medicală de specialitate, cu sprijinul și coordonarea CNAS.

3.1.1.9. Consultarea programului de lucru al medicilor specialişti/laboratoarelor în contract cu CNAS

Această funcţionalitate va permite medicului care emite un bilet de trimitere să obţină din sistemul SIUI o listă cu medicii specialişti respectiv laboratoarele, în funcţie de tipul de bilet de trimitere, care pot presta serviciile recomandate, aflaţi în relaţii contractuale cu CNAS.

Această lista va fi înmânată pacientului la cerere şi va conţine inclusiv programul de lucru delaliat aşa cum a fost prezentat de furnizor la contractare.

Această listă va fi pre-filtrată după diverse criterii, cum ar fi: cabinetul medical/laboratorul au contract cu aceeaşi Casă de Asigurări cu a medicului emitent, precum şi alte criterii opţionale care vor putea fi selectate de medic, cum ar fi: localitatea în care cabinetul medical/laboratorul funcţionează.

Această funcţionalitate presupune următoarele:

- Asigurarea în sistemul central a modalităţilor şi funcţiilor de identificare a furnizorilor de servicii (cabinete medicale/laboratorare) care au contractat anumite tipuri de servicii cu CNAS, precum şi a funcţiilor de acces la programul de lucru contractat;
- actualizarea aplicaţiilor distribuite gratuit de catre CNAS, utilizate de furnizorii de servicii pentru raportarea în SIUI;
- actualizarea platformei PIAS (SIUI, Reţeta Electronică, cardul CEAS, DES) pentru integrarea cu sistemul propus, prin asigurarea accesului la datele primare de contractare din SIUI.

Accesul la această funcţionalitate permite accesul online la informaţiile existente în sistemele informatice ale CNAS, asigurând trasparenţa informaţiei în beneficiul asiguraţilor.

3.1.2. Funcţionalităţi specifice certificatelor medicale electronice

Acordarea Concediilor Medicale se poate face în anumite condiții și de către anumite categorii de medici. Certificatul Medical (CM) este un document contabil, cu circulație, condiții de eliberare și chiar de vizualizare reglementate strict. În prezent CM este completat manual pe un formular înseriat într-un carnet cu circulație specială. Datele trebuie să fie corecte, să nu existe ștersături, să fie citeț înscrise, iar certificatul trebuie semnat, ștampilat și apoi înmânat și circulat cu restricții, fiind purtător de informație medicală confidențială.

Numeroase restricții care trebuie verificate la acordarea unui Concediu Medical nu țin de aspectul medical, care este cunoscut de medic, ci de aspecte ale regulilor impuse de normativele casei de asigurări. Unele astfel de aspecte pot fi validate într-un cadru care depășește cabinetul medicului. Aceasta se face prin sistemul SIUI, dar numai în condițiile în care medicul este conectat on-line. Apar uneori situații când SIUI refuză raportarea unui Concediu Medical datorită erorii „*Certificatul a mai fost raportat*”, el fiind de fapt raportat eronat de către un alt medic sau angajator din acelaşi judeţ sau din ţară, care eventual nu

fusese conectat la SIUI. În prezent rezolvarea unei astfel de situaţii este destul de costisitoare: CJAS notifică medicul/angajatorul care a raportat greşit seria şi numărul certificatului medical sau notifică alt CJAS dacă cel care a raportat greşit este din alt judeţ.

Precauțiile la acordarea Concediilor Medicale și vehicularea Certificatelor respective se pretează la informatizare în cadrul unui sistem dedicat, care ar putea elimina multe din problemele de pe tot fluxul de viață al certificatului, de la emitere și până la arhivare.

Implementarea unui astfel de sistem va fi însoțită de acțiuni cu caracter de reglementare, de exemplu privind legiferarea unui alt tip de document pentru certificatul medical, acțiuni care vor fi în responsabilitatea CNAS.

În această secţiune sunt grupate şi detaliate cerinţele funcţionale specifice pentru sistemul de generare, control și validare a certificatelor medicale electronice. Sistemul va fi constituit dintr-o componentă centrală, care va fi integrată cu restul sistemelor din platforma PIAS, și din funcționalități la nivelul aplicațiilor informatice locale, folosite în prezent de medicii care accesează sistemele din platforma PIAS.

Notă: Referitor la aplicațiile utilizate local de furnizorii de servicii se va face apel la paragraful 3.1.6 din prezentul document.

3.1.2.1. Generarea de plaje de serii pentru certificate medicale electronice

Seriile şi numerele certificatelor medicale electronice sunt gestionate de sistemul informatic central. La cerere sistemul central generează plaje de serii şi numere de certificate medicale unice pentru fiecare furnizor de servicii medicale în parte.

Pentru fiecare medic care eliberează certificate medicale va fi acordată o plaja de numere şi serii pe baza convenţiei medicale încheiate între unitatea la care îşi desfăşoară activitatea şi CJAS.

Furnizorii se conectează online la sistemul informatic şi pot genera şi descărca o plajă de serii şi numere folosită ulterior pentru a emite certificate medicale. Gestionarea centralizată a seriilor şi numerelor pentru certificate medicale asigură unicitatea acestora în cadrul sistemului.

La fiecare emitere de certificat medical aplicaţia furnizorului generează următorul număr de certificat din plaja alocată, unicitatea fiind validată apoi online de sistemul informatic central.

3.1.2.2. Emiterea certificatelor medicale electronice online

Medicii care au în îngrijire pacienţi şi trebuie să le acorde certificate de concediu medical, vor folosi suportul funcțional al noului sistem utilizând aplicaţiile dedicate raportării către CNAS, aplicaţii care vor fi completate pentru a include şi funcţionalităţile specifice eliberării de certificate medicale electronice.

Pacientul căruia i se eliberează un certificat medical electronic va fi identificat în cabinetul medicului curant prin cardul CEAS dacă acesta va fi disponibil. În caz contrar, sistemul trebuie să asigure identificarea pacienţilor după numărul unic de asigurat, obţinut din CNP-ul de pe actul de identitate prezentat conform procedurii pusă la dispoziţie de CNAS pentru toate sistemele constituente ale PIAS. Identificarea este de altfel parte a procesului solicitat de celelalte sisteme ale platformei PIAS, odată cu raportarea on-line a consultației (conexiune cu SIUI) sau pentru prescrierea unei rețete electronice (conexiune cu SIPE). Cu alte cuvinte, noul sistem, care răspunde acestui Caiet de Sarcini, nu va solicita o altă procedură de identificare a medicului sau a pacientului, dacă ea a fost deja efectuată pentru serviciul medical curent.

Certificatul medical electronic va conţine următoarele informaţii minimale, conform prevederilor Ordinului nr.233/2006 pentru aprobarea modelului unic al certificatului de concediu medical şi a instrucţiunilor privind utilizarea şi modul de completare a certificatelor de concediu medical pe baza cărora se acordă indemnizaţii asiguraţilor din sistemul asigurărilor sociale de sănătate:

- data acordării concediului medical electronic, precum şi luna şi anul de valabilitate;
- date identificare medic curant: unitatea sanitară emitentă, numărul convenţiei şi casa de asigurări de sănătate cu care este încheiată această convenţie;
- date identificare pacient: nume, prenume, cod numeric personal, adresa de bază şi casa de asigurări la care este luat în evidenţă asiguratul;
- semnătura electronică extinsă a medicului emitent, utilizând un certificat digital calificat;
- Semnătura electronică realizată utilizând cardul national de asigurari de sanatate al pacientului, dacă pacientului i s-a emis un card.

Pe cât posibil, aceste elemente vor fi pre-completate de sistem, sau vor fi selectabile din liste cu valori valabile în context, simplificând foarte mult activitatea de editare a certificatului.

Accesul la această funcţionalitate permite verificarea pro-activă a calităţii de asigurat a pacienţilor, avertizând medicul asupra riscurilor potenţiale, acolo unde este cazul.

Emiterea unui certificat de concediu medical in continuare online

In cazul emiterii unui certificat de concediu medical în continuare, medicul prescriptor va avea posibilitatea să completeze datele în certificatul medical iniţial, în conformitate cu prevederile OUG nr.158/2005 privind concediile şi indemnizaţiile de asigurări sociale de sănătate şi ale Ordinului nr.60/2006 pentru aprobarea Normelor de aplicare a prevederilor Ordonanţei de urgenţă a Guvernului nr. 158/2005 privind concediile şi indemnizaţiile de asigurări sociale de sănătate, cu modificările şi completările ulterioare.

3.1.2.3. Validarea certificatelor medicale electronice online

Odată cu confirmarea editării formularului de certificat, în sistemul central are loc validarea automată online a concediului medical. Astfel, pentru autentificare şi autorizare se vor utiliza mecanismele existente în PIAS. Abia după ce medicul curant a fost identificat şi a primit autorizarea prin certificatul digital calificat, el va avea acces la componenta de eliberare certificate medicale electronice.

Astfel certificatele medicale vor putea fi tipărite numai după ce sistemul central a validat datele transmise şi a generat un număr unic de înregistrare al cererii de validare, care va fi şi el tipărit pe certificatul medical.

Pentru a se valida în sistemul informatic central, certificatul medical electronic trebuie să îndeplinească minim următoarele condiţii:

- Parafa medicului este valabilă la data emiterii certificatului;
- Furnizorul are convenţie valabilă cu CAS pentru certificate medicale;
- Seria şi numărul certificatului medical trebuie să fie unică şi să corespundă unei plaje alocate pentru furnizor;
- Pacientul trebuie să existe în baza de date a persoanelor din PIAS;
- Codul de indemnizaţie trebuie să fie valid;
- Parafele medicilor care avizează certificatul medical, dacă e cazul, trebuie să fie şi ele valabile;
- Data acordării certificatului medical, număr raport medical unde se impune.

Mai mult, pentru validarea efectivă a unui certificat medical electronic, sistemul relaţionează permanent cu celelalte componente ale PIAS pentru a se determina dacă pacientul este eligibil pentru a primi concediu medical, dacă nu depăşeşte perioada de timp pentru care se poate acorda certificat de concediu medical conform Ordonanţei de urgenţă a Guvernului nr. 158/2005 şi Ordinului nr.60/2006, cu modificările şi completările ulterioare.

Aceste informaţii sunt gestionate şi în alte sisteme componentele ale PIAS, care vor răspunde în timp real solicitărilor venite din exterior sau celor datorate eliberării unui certificat medical electronic.

Mai mult, pentru validarea efectivă a unui certificat medical electronic, sistemul relaţionează permanent cu celelalte componente ale PIAS (respectiv SIUI) pentru a se determina dacă pacientul este eligibil pentru a primi concediu medical, dacă nu depăşeşte perioada de timp pentru care se poate acorda certificat de concediu medical conform Ordonanţei de urgenţă a Guvernului nr. 158/2005 şi conform normelor de aplicare a acesteia.

Aceste informaţii sunt gestionate şi în alte sisteme componentele ale PIAS, care vor răspunde în timp real solicitărilor venite din exterior sau celor datorate eliberării unui certificat medical electronic.

Accesul la aceste funcţionalităţi permite validarea online pro-activă a certificatelor medicale, eliminând necesitatea de raportare periodică şi notificând medicii asupra regulilor de validare care nu au fost satisfăcute din timp, anterior emiterii propriu-zise a certificatului.

3.1.2.4. Utilizarea semnăturii electronice pentru certificatele medicale electronice

Prin adăugarea semnăturii electronice şi implicit validarea online a certificatului medical electronic se verifică automat identitatea medicului care emite certificatul, acesta fiind notificat, prin intermediul aplicaţiei de raportare asupra eventualelor nereguli.

Folosind pentru autentificare un certificat digital calificat (acelaşi pe care îl foloseşte pentru autentificarea în PIAS, în care trimite consultaţiile, serviciile şi reţetele), medicul trimite în sistemul informatic certificatul medical electronic utilizând semnătura electronică creată pe baza aceluiaşi certificat digital.

Dacă pacientul are cardul electronic de asigurări de sănătate (CEAS) acesta va fi utilizat în aplicaţia furnizorului pentru a realiza o semnătura electronică simplă utilizând certificatul digital stocat pe CIP-ul cardului.

3.1.2.5. Emiterea certificatelor medicale electronice offline

Această funcţionalitate este necesară pentru a asigura funcţionarea sistemului în situaţii excepţionale, cum ar fi imposibilitatea accesării sistemului central din motive întemeiate. În acest caz se va permite eliberarea şi tipărirea certificatului medical în mod offline, fără ca acesta să fie transmis şi validat în sistemul central. Accesul la această funcţionalitate va fi limitat, prin reglementări ale CNAS, pentru a nu încuraja abuzul şi tentativele de a ocoli anumite reguli de validare.

Datorită faptului că certificatul medical electronic offline nu este transmis în sistemul central, nu pot fi făcute validările referitoare la corectitudinea elementelor componente (coduri de indemnizaţie, parafă şi contract medic.)

De asemenea, în lipsa unei conexiuni online, nu se poate aplica şi verifica semnătura electronică a medicului sau a pacientului. Pentru astfel de cazuri, în care este permisă acordarea offline de concedii medicale, CNAS va prevedea restricţii normative de utilizare a acestor certificate medicale. Totodată se va introduce obligativitatea din partea medicilor de a completa şi transmite online datele de pe aceste certificate medicale emise offline, în momentul în care situaţia excepţională încetează.

Certificatele medicale emise offline sau online vor fi imprimate pe hârtie pentru pacient, pe un tip de formular agreat în conjuncție cu condițiile oferite de noul sistem informatic.

Observație: Tipărirea certificatului de CM nu este principial necesară, în situația în care formatul electronic devine obligatoriu. Se poate elimina astfel necesitatea unui suport de hârtie și a consumurilor asociate. Sistemul va fi proiectat astfel încât să permită setarea opțiunilor de funcționare fără a mai fi folosit suportul de hârtie.

3.1.2.6. Generarea codurilor de bare bi-dimensionale

Pe certificatul tipărit va fi imprimat un cod de bare bi-dimensional prin care se vor codifica cel puţin numărul unic de identificare al asiguratului, codul de indemnizaţie, codul de diagnostic, luna, anul şi data acordării, data de început şi data de sfârşit a concediului, parafa medicului curant. Codul de bare bi-dimensional este imprimat de către aplicaţiile folosite de medicii care eliberează concedii medicale electronice, iar formularul imprimat va fi înmânat pacientului pentru a fi utilizat la angajator.

Angajatorul are posibilitatea de a identifica în sistemul informatic propriu certificatul medical prin scanarea codului de bare bi-dimensional. Astfel dacă codul de bare este valid se pot afla

informaţiile esenţiale de pe certificat: parafa medicului emitent, numărul unic de asigurat, codul de indemnizaţie, data acordării, data de început şi de sfârşit a concediului medical acordat.

3.1.2.7. Generarea raportării lunare de certificate medicale electronice

Raportarea lunară a certificatelor de concedii medicale ale medicului curant se va realiza prin preluarea din sistemul informatic propus a tuturor certificatelor medicale electronice eliberate de către medic în respectiva lună. Aceste informaţii vor fi transmise, la cererea furnizorului, în Sistemul Informatic Unic Integrat (SIUI) unde vor intra în fluxul informaţional actual, reutilizând astfel investiţia iniţială a CNAS prin evitarea duplicării unor funcţionalităţi în cadrul sistemelor care constituie PIAS.

Mecanismul prezentat va fi implementat pentru a funcţiona în modul de lucru online. În plus, dacă medicul nu are posibilitatea să lucreze online, din motive întemeiate, trebuie prevăzută şi posibilitatea raportării către sistemul central a certificatelor emise în mod offline, pentru a constitui evidenţa completă a acestor documente la nivel central.

Raportarea angajatorului

Angajatorul se autentifică în sistem în baza semnăturii electronice, identifică certificatul de concediu medical emis pentru angajatul său şi completează electronic datele minimale din secţiunea plătitor, conform prevederilor Anexei 1 din Ordinul nr.233/2006.

Datele minimale corespunzătoare angajatorului sunt următoarele:

- Date identificare angajator (denumire, CUI);
- Date identificare tip asigurat;
- Procent de plată în funcţie de care se va calcula indemnizaţia de asigurări sociale de sănătate, indemnizaţia suportată de către angajator, indemnizaţia suportată din bugetul FNUASS şi indemnizaţia suportată din bugetul fondului de asigurare pentru accidente de muncă şi boli profesionale.
- semnătura electronică extinsă a angajatorului, utilizând un certificat digital calificat.

In baza datelor completate de către angajator, certificatul de concediu medical va fi prevalidat în sistem, după care va fi încărcat în sistem.

In termen de 90 zile de la depunerea Declaraţiei D112 pentru luna în care a fost eliberat certificatul de concediu medical, angajatorul va raporta certificatele medicale la casa de

asigurări de sănătate Aceste informaţii vor fi transmise, la cererea angajatorului , în Sistemul Informatic Unic Integrat (SIUI) unde vor completa fluxul informaţional actual .

Viza expertului

In conformitate cu prevederile art.13 din OUG nr.158/2005 şi ale Ordinului nr.60/2006, cu modificările şi completările ulterioare, viza expertului/medicului de medicina muncii pentru risc maternal, va fi acordată pentru incapacitate temporară de muncă, excepţie făcând situaţiile referitoare la bolile speciale, accidente de muncă şi boli profesionale confirmate şi pacienţii trataţi în străinătate (se vor detalia în faza de analiză distinct).

Experţii vor fi incluşi în sistem şi vor avea acces la certificatele de concediu medical emise pentru care este necesară viza, în baza semnăturii electronice.

Datele minimale obligatorii pentru medicul expert sunt următoarele:

- Date de identificare ale medicului expert: nume, prenume, cod parafă;
- semnătura electronică extinsă a medicului expert, utilizând un certificat digital calificat.

Pacienţi trataţi în străinătate

In cazul pacienţilor trataţi în străinătate, există următoarele situaţii:

- în baza actelor doveditoare traduse şi autentificate la o dată ulterioară, dar nu mai târziu de 15 zile de la data revenirii în ţară, certificatul medical se eliberează de medicul curant cu avizul DSP;
- în baza actelor doveditoare traduse şi autentificate în condiţiile şi pe duratele prevăzute de lege, certificatul de concediu medical se eliberează retroactiv de medicul curant, cu avizul DSP, în termen de maximum 5 zile de la obţinerea avizului DSP;
- în baza prevederilor Ordinului nr.592/2008 pentru aprobarea Normelor metodologice privind utilizarea în cadrul sistemului de asigurări sociale de sănătate din România a formularelor emise în aplicarea Regulamentului (CEE) nr. 1.408/71 al Consiliului privind aplicarea regimurilor de securitate socială în raport cu lucrătorii salariaţi, cu lucrătorii independenţi şi cu familiile acestora care se deplasează în cadrul Comunităţii, precum şi a Regulamentului (CEE) nr. 574/72 de stabilire a normelor de aplicare a Regulamentului (CEE) nr. 1.408/71, cu modificările şi completările ulterioare, casele de asigurări de sănătate pot elibera certificate de incapacitate de

muncă la o dată ulterioară, dar nu mai târziu de 30 zile de la data la care a fost primit raportul medical întocmit corect.

Accidente de muncă şi boli profesionale

Pentru cazurile raportate în sistem de către angajator şi confirmate ulterior ca fiind accidente de muncă sau boli profesionale, acestea vor fi extrase din sistem şi vor fi raportate la casa teritorială de pensii.

3.1.3. Cerinţe generale privind conectarea la DES a noi tipuri de furnizori de servicii medicale

Conectarea furnizorilor de servicii medicale la DES presupune, la un nivel minim, introducerea următoarelor funcţionalităţi pentru fiecare nouă categorie de servicii medicale:

- crearea mecanismelor pentru transmiterea în DES a serviciilor medicale efectuate, sub formă de documente medicale în formatul acceptat de DES;
- extinderea mecanismelor existente în DES pentru a asigura consultarea datelor medicale specifice furnizorilor de servicii medicale introduşi în sistem.

Pentru a asigura accesul liber al acestor furnizori de servicii medicale la noul sistem este necesar să fie prevăzute atât situaţiile în care aceştia dispun deja de o aplicaţie informatică medicală pe care o utilizează pentru activităţile zilnice, cât şi situaţia în care utilizează aplicaţiile puse la dispoziţie gratuit de CNAS pentru raportarea activităţii și conectarea la DES şi la celelalte sisteme componente ale PIAS.

Conectarea furnizorilor la sistemul Dosarul Electronic de Sănătate se va realiza prin servicii web care vor utiliza standardul deschis în domeniul sănătăţii HL7 v3. Prin conectarea la sistemul DES furnizorii de servicii medicale vor putea utiliza aplicațiile conectate atât pentru a transmite documente clinice către sistemul DES cât și pentru consultarea dosarului de sănătate al unui pacient – conform politicilor de integrare și de securitate ale DES. Documentele medicale vor respecta structura CDA (Clinical Document Architecture) - Release 2. Pentru a facilita utilizarea acestei interfeţe programatice, este necesară elaborarea unui Ghid de Implementare ce va fi actualizat pentru a descrie schimbările necesare pentru conectarea acestor noi tipuri de furnizori de servicii medicale la sistemul DES.

Structurarea datelor în documente clinice se va realiza utilizând un format structurat pe mai multe niveluri ierarhice, care va utiliza un set comun de vocabulare medicale pentru

codificarea informaţiei. Acest lucru va presupune detalierea în cadrul documentului clinic a informațiilor medicale care au relevanță cum ar fi date de identificare a pacientului, medicului și furnizrului de servicii medicale, diagnostice, investigații, proceduri, tratament acordat sau recomandat, medicație acordată sau recomandată. Nomenclatoarele utilizate pentru entităţile implicate vor fi oferite de către sistemul central şi vor fi în general conforme nomenclatoarelor internaţionale LOINC şi SNOMED, sau echivalent, pentru a permite interoperabilitatea cu alte sisteme similare la nivel internaţional.

Pentru situaţia în care furnizorii de servicii medicale nu dispun de un soft medical, furnizorul soluției informatice va construi o interfaţă de tip portal web integrata in portalul oferit furnizorilor de servicii medicale prin intermediul sistemului DES, componentă a PIAS. Portalul este disponibil la adresa www.des-cnas.ro si utilizeaza IBM WebSphere Portal Server, IBM Mobile Portal Accelerator. Informatiile vehiculate de/in portal sunt extrase din DES utilizand servicii web.

Aceste interfete furnizate in cadrul proiectului vor oferi următoarele funcţionalităţi:

- consultarea dosarului de sănătate (inclusiv a datelor medicale relevante specifice noilor furnizori de servicii medicale);
- introducerea documentelor medicale pe care noile tipuri de furnizori de servicii medicale le completează în timpul sau după furnizarea serviciilor medicale specifice.

Portalul se va adresa şi pacienţilor, permiţându-le acestora atât consultarea dosarului propriu de sănătate sau al persoanelor pentru care are calitatea de aparţinător, cât şi transmiterea de date (acolo unde este cazul ca pacientul sa raporteze antecedente).

Notă: Referitor la aplicațiile utilizate local de furnizorii de servicii se va face apel la paragraful 3.1.6 din prezentul document.

Pentru actualizarea aplicațiilor folosite de furnizorii de servicii medicale de recuperare/ reabilitare și respectiv de îngrijiri la domiciliu, se vor implementa mecanisme pentru:

- auditare şi jurnalizare: pentru a asigura trasabilitatea acţiunilor;
- autentificare, autorizare: pentru a asigura un acces sigur in cadrul unor scenarii bine definite;
- co-autentificare pacient: pentru a asigura consultarea pacientului doar in prezenta sa sau cu acceptul prealabil al acestuia;

- semnare digitala a datelor noi sau actualizate: pentru a asigura identificarea expeditorului si non-repudierea.

Pentru a asigura succesul conectării la DES, este necesară asigurarea unui nivel de compatibilitate crescut între aplicaţiile informatice utilizate de furnizorii de servicii medicale şi funcţionalităţile noi ale sistemului introdus în acest proiect. Acest nivel poate fi asigurat doar prin verificarea funcţionării corecte a acestui sistem informatic înainte de a le permite utilizarea în regim operaţional.

Se impune astfel ca pentru fiecare tip de furnizor să se construiască funcţionalităţi specifice de certificare a producătorilor de software medical. Acest mecanism va permite unui producător de software să execute anumite rutine stabilite în procedura de certificare sub incidenţa unui plan de certificare. La finalizarea testelor de certificare, sistemul va extrage un raport de certificare ce va conţine şi rezultatul final privind atribuirea certificării.

Implementarea funcţionalităţilor de conectare la DES a noilor tipuri de furnizori, descrise în continuare, se va realiza prin interconectare şi interoperabilitate cu sistemul informatic pentru Dosarul Electronic de Sănătate (DES), ceea ce va asigura consolidarea informaţiilor medicale într-un depozit de date central, dar şi securizarea uniformă a informaţiilor cu caracter medical ale pacienţilor.

3.1.4. Funcţionalităţi pentru conectarea la DES a furnizorilor de recuperare și reabilitare în ambulatoriu de specialitate

În această secţiune sunt grupate şi detaliate cerinţele funcţionale specifice furnizorilor de recuperare și reabilitare, în scopul conectării acestora la sistemul DES.

Serviciile de recuperare şi reabilitare medicală au ca obiectiv prevenirea sau reducerea la minim a consecinţelor fizice, funcţionale, psihice, socioprofesionale, rezultate din apariţia unei boli. Aceste servicii se acordă în anumite circumstanţe, respectând reglementările stabilite de lege.

Pachetul de servicii medicale acordate în ambulatoriul de specialitate pentru specialitatea clinică reabilitare medicală cuprinde:

- Consultaţia medicală de specialitate
- Servicii medicale de tip proceduri (cură) de tratament acordate în cabinetele sau bazele de tratament

Prin intermediul solicitărilor din prezentul Caiet de Sarcini, se vizează două obiective principale:

- Procesarea de către furnizorii de servicii de recuperare și reabilitare a biletelor de trimitere
- Accesul la informațiile medicale relevante din istoricul pacientului, înregistrat în DES, și furnizarea în DES a datelor medicale relevante rezultate din serviciul medical efectuat pentru pacient.

În urma serviciului medical efectuat pentru pacient (consultație, cură) medicul completează un document medical. Datele medicale din acest document medical vor fi consolidate în cadrul Dosarul Electronic al Pacientului și vor putea fi astfel consultate de toate categoriile de furnizori de servicii medicale și respectiv de pacienți, prin integrarea cu portalul public din DES precum și prin intermediul aplicațiilor medicale care utilizează serviciile web puse la dispoziție de sistemul DES.

Astfel, în urma prestarii unor servicii de recuperare și reabilitare (cum ar fi kinetoterapie, fizioterapie, masaj, asistenţă psihologică, consiliere socială, terapie ocupațională/ergoterapie), se vor transmite în DES documente medicale de tip consultație sau alte evidențe medicale ale serviciilor prestate pentru fiecare pacient.

Fluxul de lucru, care este inițiat odată cu prescrierea trimiterii, este următorul:

- Medicul prescriptor, care recomandă pacientului investigații de specialitate de recuperare-reabilitare, completează biletul de trimitere, folosind pentru aceasta facilitățile sistemului informatic pentru biletele electronice, descris în secțiunea 3.1.1.
- După ce biletul de trimitere a fost încărcat de prescriptor şi validat în sistemul pentru gestiunea biletelor de trimitere electronice (a se vedea cerințele de la secțiunea 3.1.1), medicul prescriptor tipăreşte un exemplar şi îl înmânează pacientului. Biletul de trimitere tipărit poate conţine un cod de bare bi-dimensional, care poate conţine informaţiile esenţiale de pe bilet: parafa medicului emitent, specialitatea către care se trimite, codul de diagnostic.
- Pe baza acestui bilet de trimitere, pacientul face o programare la furnizorul de recuperare-reabilitare, pentru a putea beneficia de serviciile recomandate prin trimiterea medicală.

- Odată cu prezentarea pacientului în cabinetul său, furnizorul de servicii de recuperare-reabilitare poate identifica în sistem biletul de trimitere în mai multe moduri:
    - Scanare cu ajutorul unui scaner a codului de bare bi-dimensional de pe biletul prezentat de pacient. Dacă codul de bare este valid, se pot afla din sistemul electronic informaţiile esenţiale care sunt și pe bilet: parafa medicului emitent, specialitatea către care se trimite, codul de diagnostic;
    - Interogarea sistemului, folosind alt identificator al pacientului decât biletul de trimitere tipărit: dacă biletul a fost încărcat în sistemul pentru bilete de trimitere electronice de către medicul emitent, toate informațiile sunt recuperate din acel sistem.
- Furnizorul de servicii de recuperare-reabilitare execută serviciul medical solicitat. Medicul poate consulta informațiile medicale din dosarul pacientului, înregistrate în DES. Pentru aceasta, aplicația sa va fi completată în cadrul proiectului cu funcționalitățile necesare pentru conectarea la DES.
- Furnizorii de servicii de recuperare-reabilitare completează pentru fiecare serviciu efectuat pacientului un document medical de consultaţie şi tratament. Serviciul efectuat generează două categorii de informații:
    - Date necesare pentru raportarea pe care furnizorul o efectuează către CJAS, în scopul decontării din FNUASS a serviciului respectiv. Aceste date sunt operate în prezent prin intermediul aplicației utilizate de orice furnizor de servicii de recuperare-reabilitare care are contract cu CJAS. Operațiile legate de acest tip de date nu fac obiectul prezentului Caiet de sarcini
    - Date medicale despre pacient și despre serviciul medical efectuat. Aceste date sunt necesare pentru includerea lor în istoricul medical al pacientului. Din punctul de vedere al prezentului Caiet de Sarcini, aceasta este categoria de date care trebuie transmisă în DES, prin completarea cu funcționalitățile necesare a aplicației furnizorului.

Sistemul proiectat poate prelua documente medicale specifice de la furnizorii de servicii de recuperare-reabilitare și va asigura transmiterea în DES a Datelor Medicale Relevante (DMR) si atunci când serviciile respective nu s-au prestat in baza unei trimiteri.

Serviciile de recuperare și reabilitare sunt accesibile pentru pacienții asigurați în baza biletelor de trimitere. Furnizorii de servicii din această categorie pot cere decontarea din FNUASS a acestor servicii numai dacă ele sunt executate în baza unor bilete de trimitere validate de SIUI.

Pentru conectarea la DES a furnizorilor de servicii de recuperare și reabilitare, setul de Date Medicale Relevante va fi extins pentru a include datele corespunzătoare acestei categorii de servicii. Pentru aceasta, în cadrul fazei de analiză a proiectului, se vor identifica eventualele extinderi ale setului DMR din DES, prin implicarea Comisiei desemnate special de CNAS pentru identificarea setului DMR pentru proiectul DES.

Setul de date medicale din DES, extins prin conectarea furnizorilor de servicii de recuperare-reabilitare, va putea fi consultat astfel:

- de toate categoriile de furnizori de servicii medicale prin intermediul serviciilor web
- de medici prin intermediul portalului
- de pacienți prin intermediul portalului
- de toate categoriile de furnizori de servicii medicale prin extinderea aplicațiilor utilizate în prezent de aceștia pentru conectarea la sistemele din platforma PIAS.

Notă: Referitor la aplicațiile utilizate local de furnizorii de servicii de recuperare și reabilitare se va face apel la paragraful 3.1.6 din prezentul document.

Pentru a utiliza sistemul propus, furnizorii vor avea la dispoziţie toate metodele descrise în capitolul de cerinţe generale privind conectarea furnizorilor in DES: portal online, servicii web şi aplicaţiile extinse de conectare la PIAS a furnizorilor de servicii de recuperare și reabilitare, distribuite gratuit de CNAS, extinse conform prevederilor acestui Caiet de Sarcini.

Consultarea datelor colectate se va face conform regulilor de acces impuse de sistemul DES atât din punct de vedere tehnic, cât și procedural (permisiuni de acces).

Datele medicale transmise vor fi organizate într-un sistem de tip depozit de date (data-warehouse) ce va permite rapoarte si analize inteligente, inclusiv analize integrate utilizând datele din Dosarul Electronic de Sănătate.

Funcţionalităţi pentru conectarea la DES a furnizorilor de recuperare reabilitare unităţi sanitare cu paturi (sanatorii şi preventorii)

Pachetul de servicii

Servicii de internare, adună datele din fişele de spitalizare
Documentele sunt următoarele:

- Biletele de trimitere/internare;
- Foaia de observaţie;
- Bilet de externare;
- Scrisoare medicală.

3.1.4.1 Funcţionalităţi pentru conectarea la DES a furnizorilor de îngrijiri medicale la domiciliu şi îngrijiri paliative la domiciliu cat si a dispozitivelor medicale destinate recuperării unor deficienţe organice sau funcţionale în ambulatoriu

În această secţiune sunt grupate şi detaliate cerinţele funcţionale specifice furnizorilor de îngrijiri medicale la domiciliu, în scopul conectării acestora la sistemul DES.

Serviciile de îngrijire la domiciliu și îngrijiri paliative la domiciliu sunt necesare pentru o parte considerabilă a populaţiei cu afecţiuni de diferite tipuri şi contribuie la sporirea gradului de confort al pacienţilor. Îngrijirea se acordă în anumite circumstanţe, respectând reglementările stabilite de lege.

Sistemul îşi propune să faciliteze accesul rapid al asiguratului la serviciile de îngrijire la domiciliu/ îngrijirile paliative la domiciliu care i-au fost recomandate acestuia de către medic.

Sistemul propus va fi proiectat şi implementat să se integreze cu sistemele componente ale platformei PIAS operate de CNAS, astfel încât să fie asigurată consistenţa din punct de vedere logic a informațiilor care ajung în diferitele sisteme ale platformei PIAS, provenind din sursa serviciilor medicale de îngrijiri la domiciliu.

În prezent, o serie de informații despre serviciile de îngrijiri la domiciliu ajung în sistemul SIUI (parte a PIAS) în scopul de a asigura decontarea serviciilor respective din FNUASS. De asemenea, conform reglementărilor legate de identificarea asiguraților prin intermediul cardului CEAS, furnizorii de servicii de îngrijiri la domiciliu se vor conecta la sistemul CEAS pentru a confirma întânirea cu pacientul, ca o condiție a validării serviciului pentru decontarea din FNUASS. Aceste cerințe sunt îndeplinite prin intermediul funcționalităților dedicate implementate deja în aplicațiile locale utilizate de furnizorii de servicii de îngrijire la domiciliu.

Noile cerințe impuse prin prezentul Caiet de Sarcini pentru această categorie de furnizori de servicii medicale se referă la operarea cu sistemul de gestiune a biletelor de trimitere (conform secțiunii 3.1.1) și cu sistemul informatic pentru Dosarul Electronic de Sănătate (DES).

Operarea furnizorilor de servicii de îngrijiri la domiciliu cu sistemul DES presupune două categorii de funcționalități:

- conectarea la sistemul DES pentru ca furnizorul de servicii medicale să poată consulta, la nevoie, istoricul medical al pacientului
- furnizarea în sistemul DES a datelor medicale importante pentru istoricul medical al pacientului și care provin din serviciile de îngrijiri la domiciliu.

Furnizorii de servicii de îngrijiri la domiciliu completează pentru fiecare serviciu efectuat pacientului un document medical specific. Serviciul efectuat generează două categorii de informații:

- Date necesare pentru raportarea pe care furnizorul o efectuează către CJAS, în scopul decontării din FNUASS a serviciului respectiv. Aceste date sunt operate în prezent prin intermediul aplicației utilizate de orice furnizor de servicii de îngrijiri la domiciliu care are contract cu CJAS. Operațiile legate de acest tip de date nu fac obiectul prezentului Caiet de sarcini
- Date medicale despre pacient și despre serviciul medical efectuat. Aceste date sunt necesare pentru includerea lor în istoricul medical al pacientului, în situația în care fac parte din setul de Date Medicale Relevante (DMR) definit de Comisia de Specialitate desemnată de CNAS pentru stabilirea DMR. Din punctul de vedere al prezentului Caiet de Sarcini, aceasta este categoria de date care trebuie transmisă în DES, prin completarea cu funcționalitățile necesare a aplicației furnizorului.

Sistemul proiectat va prelua documentele medicale specifice de la furnizorul de servicii de îngrijire la domiciliu și va asigura transmiterea în DES a Datelor Medicale Relevante (DMR).

Serviciile de îngrijiri la domiciliu sunt accesibile pentru pacienții asigurați în baza biletelor de trimitere. Furnizorii de servicii din această categorie pot cere decontarea din FNUASS a acestor servicii numai dacă ele sunt executate în baza unor bilete de trimitere validate de SIUI.

Informaţiile care vor fi considerate relevante din punct de vedere medical pentru pacient vor putea fi adăugate la Dosarul Electronic de Sănătate (din sistemul DES) al pacientului.

Astfel, în urma prestarii unor servicii de îngrijire la domiciliu, se vor transmite în DES documente medicale de tip consultație la domiciliu și planurile de îngrijire pentru fiecare pacient.

Datele medicale din aceste documente medicale vor fi consolidate în Dosarul Electronic al Pacientului si vor putea fi astfel consultate de toate categoriile de furnizori de servicii medicale, dar și de pacienți, prin integrarea cu portalul public din DES precum și prin intermediul aplicațiilor medicale care utilizează serviciile web puse la dispoziție de sistemul DES

Prin conectarea la DES a furnizorilor de servicii de îngrijire la domiciliu şi îngrijiri paliative la domiciliu, setul de Date Medicale Relevante va fi extins pentru a include datele corespunzătoare acestei categorii de servicii. Pentru aceasta, în cadrul fazei de analiză a proiectului, se vor identifica eventualele extinderi ale setului DMR din DES, prin implicarea Comisiei desemnate special de CNAS pentru identificarea setului DMR pentru proiectul DES.

Setul de date medicale extins prin conectarea furnizorilor de îngrijiri medicale la domiciliu va putea fi consultat astfel:

- de toate categoriile de furnizori de servicii medicale prin intermediul serviciilor web
- de medici prin intermediul portalului
- de pacienți prin intermediul portalului
- de toate categoriile de furnizori de servicii medicale prin extinderea aplicațiilor utilizate în prezent de aceștia pentru conectarea la sistemele din platforma PIAS.

Pentru a utiliza sistemul propus, furnizorii vor avea la dispoziţie toate metodele descrise în capitolul de cerinţe generale privind conectarea furnizorilor in DES: portal online, servicii web şi aplicaţiile extinse de conectare la PIAS a furnizorilor de servicii de îngrijire la domiciliu, distribuite gratuit de CNAS, extinse conform prevederilor acestui Caiet de Sarcini.

Consultarea datelor colectate se va face conform regulilor de acces impuse de sistemul DES atât din punct de vedere tehnic, cât și procedural (permisiuni de acces).

Datele medicale transmise vor fi organizate într-un sistem de tip depozit de date (data-warehouse) ce va permite rapoarte si analize inteligente, inclusiv analize integrate utilizând datele din Dosarul Electronic de Sănătate.

Pachetul de bază pentru dispozitivele medicale destinate recuperării unor deficienţe organice sau funcţionale în ambulatoriu se regăseşte în Anexa nr. 38 la Ordinul nr. 619/360/2014 lit. A

Dispozitivele medicale destinate recuperării unor deficienţe organice sau funcţionale în ambulatoriu se acordă pentru o perioadă determinată ori nedeterminată, **în baza unei prescripţii medicale sub forma unei recomandări medicale tipizate** conform modelului prevăzut în anexa nr. 39 D la Ordinul nr. 619/360/2014. **P**entru dispozitivele de protezare stomii în cazul pacienţilor cu stome permanente şi pentru dispozitive pentru incontinenţă/retentie urinară, în cazul pacienţilor cu incontinenţă urinară permanentă, medicul va menţiona pe prescripţia medicală "stomă permanentă" respectiv "incontinenţă urinară permanentă".

FLUX DE LUCRU:

- Prescripţia medicală **se încarcă în sistem; medicul tipăreşte două exemplare pe care le înmânează pacientului;**
- Un exemplar al prescripţiei se depune la casa de asigurări de sănătate şi un exemplar rămâne la asigurat (pe care îl predă furnizorului de dispozitive medicale împreună cu decizia de aprobare a dispozitivului medical eliberată de casa de asigurări de sănătate).
- **Asiguratul depune la CAS un exemplar al prescripţiei, cererea şi celelalte documente prevăzute de reglementările legale în vigoare (ex.** *copie act de identitate**,** document justificativ care atestă calitatea de asigurat, declaraţia pe propria răspundere din care să rezulte că deficienţa organică sau funcţională nu a apărut în urma unei boli profesionale, a unui accident de muncă sau sportiv, certificatul de naştere pentru copii cu vârsta până la 14 ani*.)

  Pentru unele dispozitive medicale se depun la CAS documente suplimentare care însoţesc prescripţia medicală - ex.:

  - pentru protezare auditivă prescripţia este însoţită de audiograma tonală liminară şi audiograma vocală

  - pentru protezarea vizuală - implant cu lentile intraoculare – prescripţia este însoţită de biometrie.
- Cererile sunt supuse aprobării CAS care în termen de cel mult 3 zile lucrătoare de la data înregistrării cererii, este obligată să ia o hotărâre privind acceptarea sau

respingerea cererii. CAS are trei posibilităţi: respinge cererea / aprobă cererea şi emite decizie sau asiguratul este trecut pe o listă de prioritate.

- În cazul aprobării cererii, CAS emite o decizie de aprobare pentru procurarea/închirierea dispozitivului medical în limita fondului aprobat cu această destinaţie.
- Decizia se ridică de la casa de asigurări de sănătate de către beneficiar/ aparţinător/împuternicit legal în acest sens sau se expediază prin poştă în maximum 2 zile de la emitere în cazul în care asiguratul solicită prin cerere.
- Pentru dispozitivele de protezare stomii şi incontinenţă urinară decizia este însoţită de o anexă cu 3 taloane lunare;

  Decizia de aprobare pentru închirierea fotoliilor rulante, a echipamentelor pentru oxigenoterapie şi ventilaţie noninvazivă precum şi a dispozitivelor pentru administrarea medicaţiei şi alimentaţiei artificiale este însoţită de o anexă cu 3 taloane lunare;

- Pentru procurarea dispozitivului medical, asiguratul, se adresează, în perioada de valabilitate a deciziei, unuia dintre furnizorii din lista furnizorilor de dispozitive medicale aflaţi în relaţie contractuală cu CAS, **cu următoarele documente:** decizia emisă de casa de asigurări de sănătate şi prescripţia medicală
- document de validare întocmit de către un medic de specialitate otorinolaringologie, pe baza raportului probei de protezare pentru protezarea auditivă;

3.1.5. Aplicațiile utilizate de furnizorii de servicii medicale

În prezent, furnizorii de servicii medicale în contract cu Casele Județene de Asigurări de Sănătate se conectează la sistemele din platforma PIAS (SIUI, SIPE, CEAS, DES) folosind aplicații locale în scopul de a accesa FNUASS. Conectarea la oricare din sistemele PIAS se face folosind aceeași aplicație și respectând procedurile impuse de accesul la această platformă, prin intermediul certificatului digital calificat cu care sunt înregistrați în SIUI.

Aplicațiile locale folosite de furnizorii de servicii medicale și farmaceutice pot fi de două categorii:

- Furnizate gratuit de CNAS, prin descărcarea acestora de pe site-ul CNAS dedicat. Aceste aplicații asigură îndeplinirea obligațiilor și procedurilor de raportare și accesare a informațiilor impuse de reglementările CNAS în legătură cu sistemele din platforma PIAS
- Achiziționate de pe piața de produse IT, care în plus față de facilitățile categoriei de mai sus, oferă funcţionalităţi avansate pentru activitatea de cabinet a medicului sau a furnizorului de servicii medicale sau farmaceutice (ex. gestiune stocuri, salarizare.).

Ofertantul va asigura implementarea funcționalităților de la nivelul aplicaţiilor locale, solicitate prin prezentul Caiet de Sarcini, prin actualizarea tuturor aplicațiilor actuale, din prima categorie de mai sus, cu obligația ca aplicațiile rezultate să fie distribuite gratuit prin descărcare de pe site-ul CNAS, acestea fiind utilizate de medici la nivel local pentru conectarea la sistemele PIAS (SIUI, SIPE, CEAS, DES).

Referitor la cea de-a doua categorie de aplicații, ofertantul va asigura în cadrul proiectului publicarea specificațiilor funcționale complete pentru producătorii de aplicații de pe piața IT, pentru a asigura acestora accesul la cerințele tehnice de implementare în aplicațiile lor a funcționalităților impuse de utilizarea noului sistem. Aceste specificații funcționale vor fi publicate conform aceloraşi proceduri utilizate pentru toate actualizările aplicațiilor folosite de furnizorii de servicii medicale şi farmaceutice care se conectează la sistemele din platforma PIAS.

În cadrul proiectului producătorii de aplicații IT vor avea acces la mediul de testare, pentru verificarea funcționării aplicațiilor lor în concordanţă cu noul sistem. Mediul de testare se va accesa în mod securizat, conform unor proceduri similare cu cele utilizate în cadrul celorlalte sisteme din platforma PIAS (SIUI, SIPE, CEAS şi DES).

În legătură cu aplicațiile locale de la nivelul furnizorilor de servicii medicale, se evidențiază următoarele cerințe, care se vor realiza prin completarea aplicaţiilor folosite actual, conform precizărilor de mai sus:

- Accesarea de către furnizorii de servicii a noului sistem informatic se va face prin intermediul aceloraşi aplicații prin care utilizatorul se conectează la sistemele actuale din platforma PIAS, evitând operarea aceloraşi informaţii în mai multe aplicaţii din partea utilizatorilor. Cu alte cuvinte, în cadrul proiectului nu se vor proiecta aplicații noi, în schimb, se vor extinde aplicațiile existente cu funcționalitățile noi, determinate de necesitatea accesării noului sistem integrat în PIAS
- Proiectarea noilor funcționalități se va realiza prin integrarea în aplicațiile furnizorilor a următoarelor elemente:
    - Conectarea la sistem folosind procedurile de acces şi autentificare utilizate de sistemele din platforma PIAS, evitând astfel dubla înregistrare a certificatelor calificate atât la nivelul aplicaţiei locale, cât şi la nivelul sistemului central
    - Funcţionalităţi noi, în funcție de specificul furnizorului de servicii, pentru:

- Acces la sistemul de bilete de trimitere electronice
- Acces la sistemul de Certificate Medicale electronice
- Acces la DES pentru vizualizarea dosarului electronic de sănătate al pacientului
- Transmiterea în DES a documentelor medicale rezultate din serviciul medical curent, operație care se va preciza în cursul fazei de analiză, urmărindu-se pe cât posibil automatizarea preluării datelor DMR din raportarea curentă a serviciului efectuat de furnizor

- Formulare de colectare a datelor pentru funcționalitățile noi. Aceste formulare vor asigura suport informațional în procesul de editare, prin utilizarea extensivă a datelor existente deja în sistemele PIAS. Astfel, vor fi completate automat o serie de câmpuri cu informații a căror determinare este unică (de exemplu, datele demografice ale pacientului, cabinetul medical, medicul, contractul cu CJAS). Pentru informațiile cu valori limitate se vor oferi liste de selecție
- Pre-validarea datelor completate în formulare. După confirmarea dată de medic pentru datele completate în formularul de colectare a datelor, aplicația transmite datele spre validare către sistem. Validarea se realizează prin apelarea funcționalităților sistemului central sau a serviciilor Web puse la dispoziție de sistemele din PIAS, în funcție de caz. Ca urmare, sistemul returnează mesaje de confirmare sau semnalează eventuale erori sau inadvertențe
- Tipărirea documentului medical: bilet de trimitere, certificat de concediu medical, condiționată de validarea respectivului document.

3.1.6. Arhitectura funcţională a sistemului

Prin arhitectura sistemului informatic înţelegem structurile, mecanismele şi interfeţele utilizate, precum şi comunicarea între părţile componente. Arhitectura de sistem descrie viziunea fizică şi logică a sistemului propus, relevă modul în care sistemul va fi construit, defineşte modul în care vor fi utilizate diferite concepte, cât şi aspecte vizând posibilitatea dezvoltării viitoare a sistemului.

Soluţia tehnică pentru sistemul informatic propus va include, pe lângă sistemul de producţie, un subsistem de dezvoltare/testare şi instruire necesar exploatării în conformitate cu bunele practici internaţionale şi cu metodele actuale în domeniul formării profesionale continue a personalului. Astfel sistemul va include un mediu de producţie şi un mediu auxiliar cu rol de a acoperi nevoile de dezvoltare, testare şi de integrare.

Mediul de producţie este mediul principal folosit de utilizatorii soluţiei, implementând toate cerinţele funcţionale şi non-funcţionale. Toate componentele mediului de productie vor fi prevazute si instalate astfel incat sa se asigure o inalta disponibilitate si redundanta a sistemului.

- **Mediul de dezvoltare** contine componentele cheie necesare dezvoltarii unor noi versiuni aplicative specifice, fiind utilizat pentru:
    - dezvoltarea de modificări ce vor fi aduse mediului principal de producţie
    - validarea modificărilor la nivelul aplicatiei specifice înaintea promovării acestora pe mediul de producţie
    - validarea integrării cu sisteme externe
    - instruirea utilizatorilor prin simularea de scenarii de utilizare reale folosind date de test, pentru formarea deprinderilor de utilizare fără a afecta mediul de producţie şi datele reale; Componentele administrative ale sistemului (administrare si monitorizare, colectare si analiza a evenimentelor de securitate, mascare dinamica a datelor, backup) vor putea fi folosite atat mediul de productie cat si cel de testare/dezvoltare, nefiind necesara prezenta lor in cadrul ambelor medii.

Din punct de vedere al arhitecturii fizice şi logice sistemul informatic propus trebuie să adere la următoarele principii arhitectonice:

- **Modularitate** – sistemul este descompus în subsisteme cu roluri şi caracteristici/ proprietăţi bine definite fără a avea o suprapunere de funcţionalităţi între două subsisteme
- **Deschidere** – pot fi adăugate componente, proprietăţi noi. Modulele (componentele), soluţiei au caracteristici generale, adaptabile în funcţie de cerinţele clientului
- **Stratificare** – arhitectură separată pe mai multe straturi, fiecare strat având roluri şi responsabilităţi bine delimitate

- **Flexibilitate** – arhitectura propusă se bazează pe standarde şi tehnologii deschise, sistemul fiind conceput modular astfel încât să poată integra uşor atât modificări ale funcţionalităţilor existente cat şi introducerea de noi funcţionalităţi
- **Scalabilitate** – utilizând tehnologii ce au suport pentru lucrul în arhitectura cluster suportă o scalare atât pe orizontală cat şi pe verticală pentru a acomoda uşor creşterea numărului de utilizatori sau a volumului de operaţii efectuate de aceştia
- **Eficienţă** – prin optimizarea utilizării resurselor hardware şi de comunicaţii şi prin eficientizarea activităţii operatorilor prin automatizarea activităţilor repetitive
- **Ergonomie** – prin utilizarea conceptelor moderne şi bunelor practici ale interacţiunii om-calculator, a standardelor legate de proiectarea interfeţelor utilizator, precum şi a principiilor de uzabilitate moderne şi a tendinţelor contemporane din domeniul aplicaţiilor Web
- **Interoperabilitate** – prin integrarea programatică cu alte sisteme informatice bazate pe tehnologii eterogene prin intermediul uneia sau mai multor interfeţe programatice. Nefuncţionarea unuia din sistemele externe nu trebuie să afecteze disponibilitatea şi funcţionalitatea soluţiei.

Din punct de vedere funcţional sistemul informatic propus trebuie să îndeplinească următoarele obiective:

- Sistemul informatic online permite accesul sigur şi de încredere la informaţiile referitoare la asiguraţi şi la furnizorii de servicii medicale prin expunerea unor servicii de validare online pentru medici
- Sistemul informatic aplică măsuri uniforme pentru protecţia şi siguranţa datelor, indiferent de locul sau timpul accesării acestora
- Autentificarea medicilor se va realiza prin reutilizarea modalităţilor de acces actuale implementate de celelalte sisteme componente ale PIAS
- Autentificarea asiguraţilor se va realiza, în mod optim, prin folosirea cardului electronic de asigurat de sănătate, pentru asiguraţii posesori ai acestui card. Pentru asiguraţii care nu dispun încă de cardul CEAS, identificarea asiguraţilor se va face prin utilizarea CID (codului unic de identificare a asiguratului, care se poate obţine din CNP prin funcţionalităţi ale sistemului)

- Sistemul informatic verifică autenticitatea informaţiilor transmise de utilizatori prin utilizarea semnăturii electronice şi menţine date referitoare la timp şi sursă pentru datele transmise
- Sistemul informatic face posibilă o colectare eficientă a datelor prin intermediul importului din fişiere formatate, conform unei specificaţii standard.

Pentru a putea fi utilizat cardului electronic de asigurat de sănătate de catre asigurati pentru accesarea sistemului este necesar ca acesta sa poata fi integrat in cadrul sistemului de operare al echipamentului utilizatorului. In acest sens este necesara dezvoltarea unor interfete standardizate de comunicare cu cardul criptografic. In acest moment, standardul in vigoare pentru comunicarea cu asemenea dispozitive este definit de PKCS#11. Pentru sistemele de tip Microsoft, de asemenea este necesara existenta unui Cryptographic Service Provider (CSP). Prin intermediul acestor doua interfete standardizate asiguratul va putea utiliza componenta criptografica a cardului electronic de asigurat de sănătate pentru conectarea la sistem si accesarea serviciilor la care acesta are dreptul.

Dezvoltarea acestor functionalitati presupune urmatoarele activiati:

analiza functionalitatilor criptografice de baza ale componentei smartcard a cardului electronic de asigurat de sănătate;

elaborarea planului de dezvoltare pentru interfata de tip PKCS#11 si validarea functionalitatilor expuse de aceasta

elaborarea planului de dezvoltare pentru interfata de tip CSP

dezvoltarea interfetelor PKCS#11 si CSP

testarea functionala a interfetelor dezvoltate;

efectuarea de teste de integrare utilizand dispozitivele criptografice, certificatele stocate pe cardul electronic de asigurat de sănătate si sistemul informatic;

dezvoltarea unui pachet de instalare dupa acceptanta finala a interfetelor

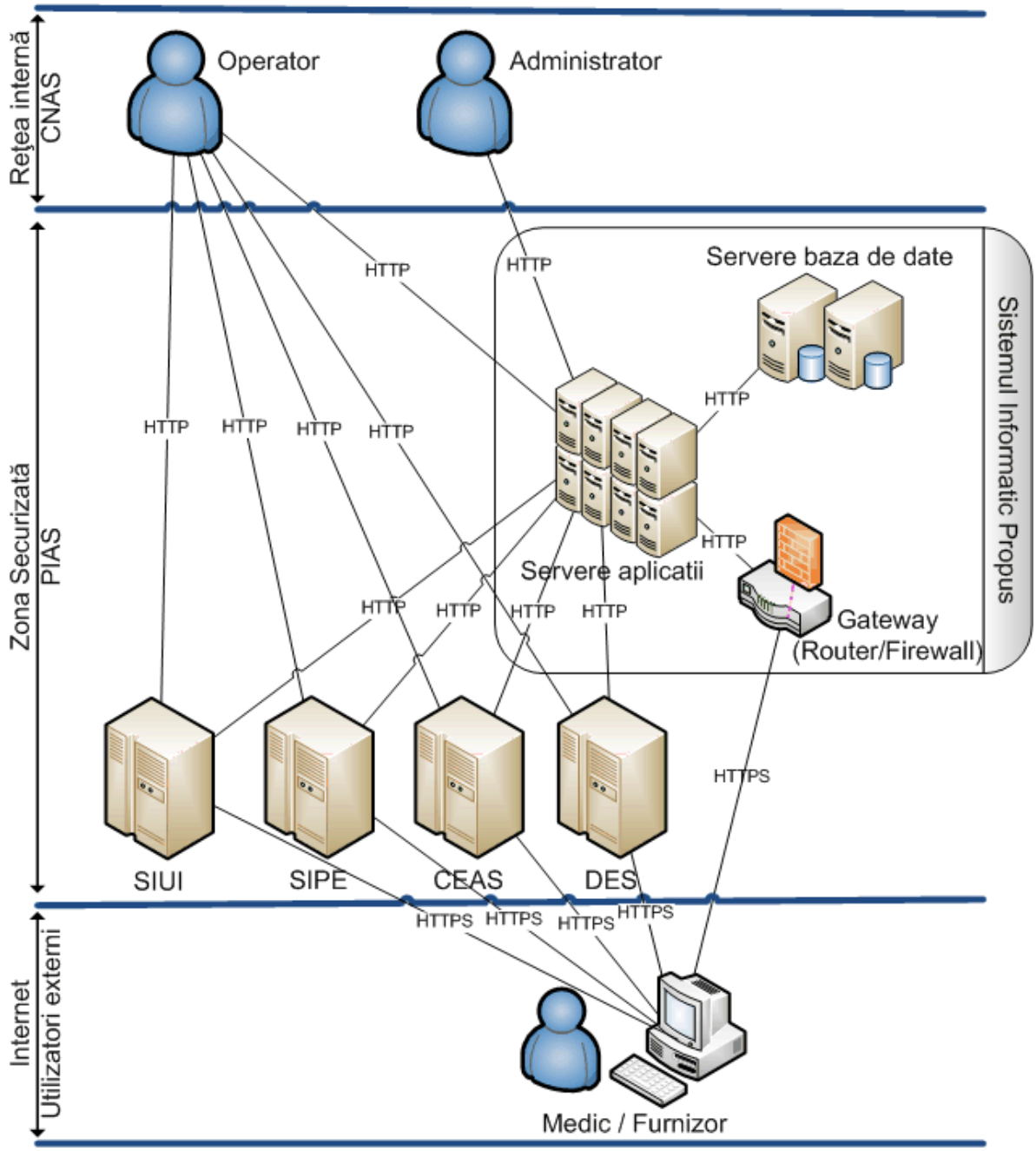
In cadrul activitatilor de integrare a noilor interfete dezvoltate se vor avea in vedere

autentificarea utilizatorilor prin intermediul certificatului digital stocat pe cardul electronic de asigurat de sănătate

validarea starii certificatului prezentat pentru autentificare de catre utilizator

obligativitatea introducerii codului PIN pentru accesul la informatiile criptografice stocate pe cardul electronic de asigurat de sănătate

Diagrama următoare prezintă interacţiunile dintre principalii actori care for realiza activităţi prin intermediul sistemului informatic propus, precum şi interacţiunile dintre noul sistem şi sistemele informatice existente ale PIAS:



Actorii

- **Asigurat** - cetăţean înregistrat în sistemul de asigurări de sănătate, beneficiar de servicii medicale sau de concedii medicale suportate din FNUASS

- **Furnizor** - persoană juridică care are contract cu CNAS pentru prestarea de servicii medicale pentru asiguraţi, sau cu convenţie de eliberare bilete de trimitere sau certificate medicale
- **Administrator** - personal specializat în administrarea sistemului informatic, responsabil pentru administrarea utilizatorilor şi pentru gestiunea şi întreţinerea conţinutului publicat
- **Operator** - personal care verifică datele din sistem, creează situaţii statistice, sau efectuează alte operaţii permise prin funcționalități ale sistemului.

Sistemul informatic central va fi proiectat astfel încât să funcţioneze în regim de înaltă performanţă şi disponibilitate şi va fi separat pe trei niveluri, conform celor mai bune în domeniu: stocarea datelor, prelucrare şi prezentare.

Proiectarea sistemului se va face astfel incat sa respecte recomandarile de performanta si securitate in domeniu. Astfel, sistemul va fi de tip “no single point of failure” cu componente redundante atat la nivel aplicativ cat si la nivel hardware.

Sistemul astfel proiectat va prezenta 3 zone distince:

- **Nivel de prezentare**: este compus din servere web de prezentare si acces la serviciile sistemului.
- **Nivel de aplicatie**: este compus din servere de aplicatii ce gestioneaza modulele software ce vor fi instalate
- **Nivel baze de date**: trebuie sa fie constituit din servere de baza de date configurate in cluster mod activ activ care sa asigure balansarea incarcarii, precum si scalabilitate si disponibilitate maxima, pe care ruleaza sistemul de gestiune a bazei de date.

In cazul in care se utilizeaza mecanisme de virtualizare a resurselor hardware trebuie avut in vedere ca alocarile de resurse hardware pe cele 3 nivelel sa fie facute intr-un mod eficient si care sa asigure parametrii de performanta necesari functionarii in conditii optime. Astfel se va tine cont ca atat la nivel de aplicatie cat si la nivel de baze de date, structura in cadrul careia ruleaza componentele sistemului (masina fizica, masina virtuala, partitie, etc) sa aiba alocata un minim de 4 core-uri fizice de procesare si sa dispuna de 8GB de memorie RAM alocata per core.

Zonele pot fi delimitate la nivel logic si separate prin intermediul functionalitatilor de tip firewall. Accesul utilizatorilor se va permite doar la nivelul de prezentare, iar cererile catre

nivelul aplicativ vor gestionate de acesta. Utilizatorii nu trebuie sa aiba acces direct sub nici o forma catre nivelul bazelor de date.

3.1.7. Componente hardware

Echipamentele vor fi găzduite de beneficiar la sediul acestuia sau într-o locaţie desemnată de beneficiar, ce va asigura condiţiile de alimentare cu surse neîntreruptibile şi climatizare a echipamentelor de calcul. Furnizorul soluţiei va preciza în cadrul ofertei cerinţele de mediu (număr rack-uri, necesar putere, necesar climatizare, necesar circuite electrice) pentru toată infrastructura de calcul necesar a fi găzduită la nivel central. Echipamentele indicate în tabelul de mai jos vor fi plasate in rack-uri 19'' pe care furnizorul le va include în cadrul ofertei.

Sistemul informatic proiectat va fi compus din următoarele echipamente hardware:

Nr.	Descriere	Cantitate
1	Sasiu servere de productie	1
2	Sasiu servere blade de productie	2
3	Sasiu servere blade de test/dezvoltare	1
4	Server de baza de date de productie	1
5	Server de baza de date de test/dezvoltare	2
6	Server de aplicatie de productie	12
7	Server de aplicatie de test/dezvoltare	2
8	Masiv stocare date	1
9	Switch SAN	2
10	Server administrare infrastructura IT	2
11	Server backup	1
12	Echipament backup pe banda	1
13	Server solutie monitorizare si gestiune incidente	3
14	Switch Ethernet agregare comunicatii interne	2
15	Switch Ethernet comunicatii externe	2
16	Switch Ethernet management infrastructura IT	2
17	Echipament modular servicii de securitate	2
18	Echipament echilibrare incarcare si degrevare SSL	2

În continuare sunt prezentate caracteristicile componentelor hardware prevăzute:

Articol 1	
Sasiu servere de productie	
Format	rack-universal sau dedicat
Module inteconectare	Conexiune SMP inter-procesoare
Sistem de ventilatie	N+1 Redundant, hot-swap
Sistem de alimentare	N+N Redundant, hot-swap
Procesoare	Sistemul sa poata fi configurat cu minim 32 de procesoare octo-core

Articol 1	
	RISC/EPIC 64 biti cu minim 8 nuclee, minim 32 MB cache L3
RAM	Sistemul sa accepte minim 8TB in configuratie maxima.
Conectivitate Ethernet	2 * switch-uri 10 Geth cu 24 porturi
Conectivitate SAN	2 * switch-uri FC 24 porturi de 8 GB
Sistem de stocare local (intern sau extern)	incinte multiple cu cate 24 HDD SFF SAS 6G DP 300GB 15KRPM, RAID 0, 1, 3, 5, 6, 10, 50
Management sistem	2* procesoare dedicate in fail-over, remote management
Porturi	Sistemul sa ofere minim 90 porturi PCIe v2 in configuratie maxima
Garantie	3 ani, 24*7 24h acoperire, 6 ore timp de reparatie HW

Articol 2	
Sasiu servere blade de productie	
Format	Maxim 10U
Sistem de ventilatie	Minim 8 ventilatoare capabile sa asigure racirea componentelor intregului sasiu, N+1 Redundant, hot-swap
Sistem de alimentare	Minim 4 surse de alimentare cu energie electrica, N+N redundanta, hot-swap
Tip blade-uri	RISC/EPIC, x86 si storage; Sistemul sa poata fi configurat cu minim 32 de procesoare octo-core RISC/EPIC 64 biti cu minim 8 nuclee, minim 32 MB cache L3 Sistemul sa accepte minim 8TB in configuratie maxima. Sistem de stocare local (intern sau extern): incinte multiple cu cate 24 HDD SFF SAS 6G DP 300GB 15KRPM, RAID 0, 1, 3, 5, 6, 10, 50
Tip module interconectare disponibile	Ethernet, Fibre Channel, InfiniBand, SAS
Module interconectare	posibilitatea de a monta minim 8 module retea Ethernet, SAN, SAS sau Infiniband.
Servere blade instalabile	Minim 16
Conectivitate Ethernet	2 x switch-uri 10 Geth cu minim 8 porturi uplink Suport pentru capabilitati de virtualizare Capabilitati de stackare sau tehnologii echivalente a minim 8 switch-uri din cadrul aceleasi familii Capabilitati de partajare a latimii de banda per port
Conectivitate SAN	2 x switch-uri FC 8 porturi uplink de 8 GB
Integrare ETH/FC	FCoE la nivel blade si switch in sasiu ptr. unificarea retelei interne ETH/FC prin protocoale tip FCoE. Optional.
Management sistem	2 module de administrare redundante . - asigura set-up-ul si controlul intregului sasiu; - inventariaza si toate dispozitivele din sasiu ; - ofera informatii legate de temperatura si consum in timp real pentru

	fiecare server in parte, precum si pentru intregul sasiu; - dispune de un display frontal cu taste pentru a facilita operarea din DataCenter Software de management dedicat care asigura atat managementul serverelor discrete cat si al serverelor virtuale, intr-un mod unitar. Acesta va asigura vizibilitatea starii de buna functionare pentru intreg ansamblul pe o pagina unica, permitand personalului de administrare sa investigheze diferitele alarme prin accesare detalii in mod ierarhic efectuate prin selectari simple cu mouse-ul, fara a fi necesar sa lucreze in aplicatii multiple. Prin software-ul de management se vor asigura functionalitati de limitare a consumului ansamblului si masurarea puterii consumate pentru fiecare server in parte. Este asigurat controlul de la distanta al oricarui tip de activitate pentru serverele instalate in enclosure: rebootare, instalari OS, preluare ecran la distanta, activitati de depanare. Functionalitatile minime asigurate vor include: - Vizualizare event-uri, monitorizarea starii de buna functionare, colectare date de inventar, descoperire si identificare optiuni HW instalate, raportare software; - Capabilitati de management de la distanta pentru administrare curenta si pentru asistenta tehnica in caz de incidente; - Capabilitati de console replay pentru review si documentare incidente tehnice sau activitati de administrare; - Capabilitati de a transmite de la distanta comenzi de power-on si power-off; - Posibilitatea de a virtualiza de la distanta unitati de CD-ROM/DVD pentru efectuarea instalarilor de software de catre administrator; - Accelerarea instalarilor de imagini software si sistem de operare; - Simplificarea inventarierii si instalarii patch-urilor corectoare; - Masurarea conditiilor termice de operare. Alte porturi: - Minim 1 x DB9, 1 x RJ45 , 1 x USB pentru fiecare modul de administrare.
Garantie	3ani, 24x7 24h acoperire, 6 ore timp de reparatie HW

Articol 3	
Sasiu servere blade de test/dezvoltare	
Format	Maxim 10U
Sistem de ventilatie	Minim 8 ventilatoare capabile sa asigure racirea componentelor intregului sasiu, N+1 Redundant, hot-swap
Sistem de alimentare	Minim 4 surse de alimentare cu energie electrica, N+N redundant, hot-swap
Tip blade-uri	RISC/EPIC, x86 si storage Sistemul sa poata fi configurat cu minim 32 de procesoare octo-core RISC/EPIC 64 biti cu minim 8 nuclee, minim 32 MB cache L3

	Sistemul sa accepte minim 8TB in configuratie maxima. Sistem de stocare local (intern sau extern): incinte multiple cu cate 24 HDD SFF SAS 6G DP 300GB 15KRPM, RAID 0, 1, 3, 5, 6, 10, 50
Tip module interconectare disponibile	Ethernet, Fibre Channel, InfiniBand, SAS
Module interconectare	posibilitatea de a monta minim 8 module retea Ethernet, SAN sau infiniband.
Servere blade instalabile	Min 16
Conectivitate Ethernet	2 x switch-uri 10 Geth cu minim 8 porturi uplink Suport pentru capabilitati de virtualizare Capabilitati de stackare sau tehnologii echivalente a minim 8 switch-uri din cadrul aceleasi familii Capabilitati de partajare a latimii de banda per port
Conectivitate SAN	2 x switch-uri FC 8 porturi uplink de 8 GB
Integrare ETH/FC	Posibilitatea de acomoda FCoE la nivel blade si switch in sasiu ptr. unificarea retelei interne ETH/FC prin protocoale tip FCoE.
Management sistem	2 module de administrare redundante care: - asigura set-up-ul si controlul intregului sasiu; - inventariaza si toate dispozitivele din sasiu ; - ofera informatii legate de temperatura si consum in timp real pentru fiecare server in parte, precum si pentru intregul sasiu; - dispune de un display frontal cu taste pentru a facilita operarea din DataCenter Software de management dedicat care asigura atat managementul serverelor discrete cat si al serverelor virtuale, intr-un mod unitar. Acesta va asigura vizibilitatea starii de buna functionare pentru intreg ansamblul pe o pagina unica, permitand personalului de administrare sa investigheze diferitele alarme prin accesare detalii in mod ierarhic efectuate prin selectari simple cu mouse-ul, fara a fi necesar sa lucreze in aplicatii multiple. Prin software-ul de management se vor asigura functionalitati de limitare a consumului ansamblului si masurarea puterii consumate pentru fiecare server in parte. Este asigurat controlul de la distanta al oricarui tip de activitate pentru serverele instalate in enclosure: rebootare, instalari OS, preluare ecran la distanta, activitati de depanare. Functionalitatile minime asigurate vor include: - Vizualizare event-uri, monitorizarea starii de buna functionare, colectare date de inventar, descoperire si identificare optiuni HW instalate, raportare software; - Capabilitati de management de la distanta pentru administrare curenta si pentru asistenta tehnica in caz de incidente; - Capabilitati de console replay pentru review si documentare incidente tehnice sau activitati de administrare;

	- Capabilitati de a transmite de la distanta comenzi de power-on si power-off; - Posibilitatea de a virtualiza de la distanta unitati de CD-ROM/DVD pentru efectuarea instalarilor de software de catre administrator; - Accelerarea instalarilor de imagini software si sistem de operare; - Simplificarea inventarierii si instalarii patch-urilor corectoare; - Masurarea conditiilor termice de operare. Alte porturi: - Minim 1 x DB9, 1 x RJ45 , 1 x USB pentru fiecare modul de administrare.
Garantie	3 ani, 24x7 24h acoperire, 6 ore timp de reparatie HW

Articol 4	
Server de baza de date de productie	
Nr Procesoare Instalate	16 procesoare instalate, upgradabile la 32 procesoare. Din cele 16 procesoare instalate, 8 sunt active, iar 8 procesoare sunt activate la cerere.
Tip Procesoare	Procesoare RISC/EPIC octo-core, minim 2.5GHz, minim 32 MB cache L3.
Memorie	2 TB GB DDR3 ECC instalata, upgradabila la 8TB. 1 TB RAM activ si 1 TB activat la cerere odata cu procesoarele.
Numar partitii instalate	2 partitii, upgradabile la minim 16.
Tip partitii instalate	Partitii cu separare electrica sau partitii logice cu resurse hw (core-uri, RAM, porturi IO) unic alocate si nevirtualizate pe partitie.
Resurse alocate initial per partitie (partitii cu resurse identice).	8 procesoare instalate de tip RISC/EPIC octo-core, minim 2.5GHz, 32 MB Cache L3, 1TB RAM, 24 x sloturi PCIe, 16 x porturi 8 Gb/s FC, 8 x porturi Ethernet 1 Gb/s, 16 x porturi Ethernet 10Gb/s. Din resursele instalate per partitie, 4 procesoare si 512 GB RAM sint resurse active per partitie, iar 4 procesoare si 512 GB RAM pot fi activate la cerere ca resurse in cadrul aceleiasi partitii. Activarea se va putea face fie definitiv fie pe baza unor cuante de timp de utilizare.
Subsistem stocare	Conectat pe FC 8 GB, conectate 2 incinte cu cate 24 HDD SFF SAS 6G DP 300GB 15KRPM, RAID 0, 1, 3, 5, 6, 10, 50
Controlor retea	32 x porturi 10 GeEh si 16 x porturi Geth
Controlor fiber channel	32 x porturi 8Gb/s FC
Sloturi I/O instalate	48 x PCIe upgradabile la 90 sloturi
Facilitati partitionare	Partitii fizice, Partitii logice/virtuale, Masini Virtuale
Management	Controlor on-board dual in fail-over, remote management
Form factor	Compatibil Sasiu servere de productie
Sisteme de operare suportate	tip UNIX „Mission-Critical” 64 biti, cu licente incluse pentru a se realiza un cluster intre partitiile fizice de tip Cluster File System
Garantie	3 ani, 24 x 7 24h acoperire, 6 ore timp de reparatie HW

Articol 5	
Server de baza de date de test/dezvoltare	
Procesor	4 x procesor RISC/EPIC octo-core, min. 2.5 GHz, minim 32 MB cache L3
Memorie	Minim 768 GB instalata
Protectie memorie	Memorie DDR3 de tip ECC
Controlor HDD	SAS, suport RAID 0 si 1, minim 512MB cache
Unitati de disk interne	4 x 300 GB SAS 15k
Video	Controller video integrat
Adaptor convergent	8 porturi convergente 10 GEth
Controlor fiber channel	2 x 8 Gb/s dual port HBA
Sloturi I/O	Minim 6 x PCIe
Management	- consola virtuala la distanta; - posibilitatea efectuării managementului de la distanţă, in mod securizat, pe nivele de utilizatori; - funcţie de virtual media pentru flexibilitate în instalare, administrare şi mentenanţă.
Form factor	Compatibil Sasiu servere blade de test/dezvoltare
Sistem de operare	tip UNIX „Mission-Critical” 64 biti, cu licente incluse pentru a se realiza un cluster de tip Cluster File System
Garantie	3 ani, 24x7 24h acoperire, 6 ore timp de reparatie HW

Articol 6	
Server de aplicatie de productie	
Procesor	4 x procesor min. 2.7 GHz, 8 core, 20 MB cache sau echivalent
Memorie interna	512 GB instalata, expandabila la minim 1 TB
Protectie memorie	Suport pentru: Advanced ECC, Failed DIMM Isolation, Lockstep, Rank Sparing, SDDC
Controlor HDD	SAS, suport RAID 0 si 1, minim 512MB cache
Unitati de disk interne	4 x 300 GB SSD
Video	Controller video integrat
Controlor retea	4 x 10Gb Ethernet
Controlor fiber channel	2 x 8 Gb/s dual port HBA
Sloturi I/O	Minim 3 x PCIe
Management	- consola virtuala la distanta; - posibilitatea efectuării managementului de la distanţă, in mod securizat, pe nivele de utilizatori; - funcţie de virtual media pentru flexibilitate în instalare, administrare şi mentenanţă.
Securitate:	- suport pentru modul criptografic de tip TPM 1.2;

	- port USB intern pentru autentificare.
Form factor	Compatibil Sasiu servere blade de productie
Sistem de operare	LINUX sau echivalent
Garantie	3 ani, 24x7 24h acoperire, 6 ore timp de reparatie HW

Articol 7	
Server de aplicatie de test/dezvoltare	
Procesor	4 x procesor min. 2.7 GHz, 8 core, 20 MB cache sau echivalent
Memorie	512 GB instalata, expandabila la minim 1 TB
Protectie memorie	Suport pentru:Advanced ECC, Failed DIMM Isolation, Lockstep, Rank Sparing, SDDC
Controlor HDD	SAS, suport RAID 0 si 1, minim 512MB cache
Unitati de disk interne	2 x 300 GB SAS 15k
Video	Controller video integrat
Controlor retea	4 x 10 Gb Ethernet
Controlor fiber channel	1 x 8 Gb/s dual port HBA
Sloturi I/O	Minim 3 x PCIe
Management	- consola virtuala la distanta; - posibilitatea efectuării managementului de la distanţă, in mod securizat, pe nivele de utilizatori; - funcţie de virtual media pentru flexibilitate în instalare, administrare şi mentenanţă.
Securitate:	- suport pentru modul criptografic de tip TPM 1.2; - port USB intern pentru autentificare.
Form factor	Compatibil Sasiu servere blade de test/dezvoltare
Sistem de operare	LINUX sau echivalent
Garantie	3 ani, 24x7 24h acoperire, 6 ore timp de reparatie HW

Articol 8	
Masiv stocare date	
Arhitectura	Activa-Activa astfel incat un LUN sa fie accesibil prin toate controloarele simultan, cu virtualizare atat pentru spatiul de date cat si pentru spatiul de rezerva. Virtualizarea se va efectua prin mecanisme interne unitatii de stocare.
Tip HDD suportate	SAS, S-ATA, SSD in incinte SFF sau/si LFF Suport pentru minim 450 de discuri distribuite in minim 2 nivele de perfomanta pe discuri SAS(10k/15k) si S-ATA(7.2k) Suport pentru minim 180 de discuri de tip SSD SLC
Capacitate instalata	Capacitatea bruta instalata va fi distributita pe mai multe tier-uri de performanta astfel: Minim 98 TB brut instalati din care min 54TB vor fi in diskuri de SAS 6Gbps de viteza mare (300GB, 15.000 rot/min), min 40TB discuri de 1TB, si minim 20 * 200GB SSD
Performanta	Pentru familia de produse din care face parte masivul de stocare ofertat trebuie sa existe cel putin un test de performanta publicat de catre o entitate independenta (Storage Performance Council sau echivalent) care sa ateste o performanta de cel putin 230,000 IOPS.
RAID	RAID 0, 1, 5 si 6
Nr Controloare	4 instalate
Cache	64 GB instalat nativ in unitatea de stocare
Conectivitate	Fibre Channel, iSCSI
Conectivitate SAN	Minim 24 x 8GB FC instalat
Management sistem	Prin unitate de management dedicata
Form factor	Rackabil – compatibil cu Rack echipamente
Asigurarea disponibilitatii	Controloare duale, capabile sa sustina functionalitatile intregului ansamblu in caz de defectare de tip activ-activ astfel incat o singura unitate logica sa poate fi accesata de ambele controloare in acelasi timp. Sistemul ofertat va fi configurat pentru asigurarea unei inalte disponibilitati cu “No Single Point of Failure” pentru controloare, memorie cache, ventilatoare si surse de alimentare. Mecanisme de protectie a datelor existente in memoria cache prin scriere pe o memorie de tip non-volatila, in cazul unor caderi de tensiune electrica. Suport de upgrade firmware on-line pentru controloare si discuri. Surse de alimentare redundante, cu baterii integrate protectia cache-ului. Suport nativ fara echipamente suplimentare pentru replicare locala si/sau la distanta prin intermediul protocolului iSCSI . Posibilitatea de upgrade de firmware fara scoaterea sistemului din functiune.
Functionalitatile echipamentului licentiat	Aplicatie care asigura provizionarea sistemului de stocare in functie de cerintele aplicatiilor cu posibilitate de definire a unor rutine de provizionare pe diverse grupuri de aplicatii, precum si raportarea statistica in functie de utilizarea discurilor per aplicatie. Capabilitati de tip „thin provisioning” licentiate pentru intreaga capacitate a echipamentului, in cazul unor upgrade-uri ulterioare nefiind necesara achizitionarea unor noi licente. Posibilitati de management prin sesiuni de tip telnet, SSH si SMI-S. Suport pentru realizarea de snapshot-uri consistente pentru componentele aplicative si de baze de date folosite in cadrul proiectului. Sisteme de operare suportate: Microsoft Windows, VMWare, IBM AIX, HP-UX, Sun Solaris, Linux.

Articol 9	
Switch SAN	
Tip Sasiu	Rackabil, 1U, kit inclus pentru montarea in rack
Nr porturi instalate	48 x 16 Gb FC
Administrare si monitorizare	administrare prin intermediul retelei Ethernet protocol http, posibilitatea de realizare de zone in reteaua SAN
Facilitati incluse	ISL Trunking, Extended Fabric, Adaptive Networking sau echivalent
Compatibilitate	Switch-ul trebuie sa fie compatibil cu platforma de stocare si serverele de mai sus
Garantie	3 ani, 24x7 24h acoperire, 6 ore timp de reparatie HW

Articol 10	
Server administrare infrastructura IT	
Procesor	2 x procesor min. 2.9 GHz, 6 core, 15 MB cache sau echivalent
Memorie	128 GB instalata, expandabila la minim 512 GB
Protectie memorie	Suport pentru: Advanced ECC, Memory Online Spare Mode, Lockstep Mode
Controlor HDD	SAS, suport RAID 0 si 1, minim 512MB cache
Unitati de disk interne	2 x 300 GB SAS 15K
Video	Controller video integrat
Controlor retea	2 x 10Gb Ethernet
Controlor fiber channel	1 x 8 Gb/s dual port HBA
Sloturi I/O	Minim 2 x PCIe
Management	- consola virtuala la distanta; - posibilitatea efectuării managementului de la distanţă, in mod securizat, pe nivele de utilizatori; - funcţie de virtual media pentru flexibilitate în instalare, administrare şi mentenanţă.
Securitate:	- suport pentru modul criptografic de tip TPM 1.2; - port USB intern pentru autentificare.
Form factor	Compatibil Sasiu servere blade de productie
Sistem de operare	WINDOWS STORAGE SERVER sau echivalent
Garantie	3 ani, 24x7 24h acoperire , 6 ore timp de reparatie HW

Articol 11	
Server backup	
Procesor	4 x procesoare RISC/EPIC octo-core, min. 2.5 GHz, minim 32 MB cache L3 sau echivalent
Memorie	Minim 768 GB instalata
Protectie memorie	Memorie DDR3 de tip ECC
Controlor HDD	SAS, suport RAID 0 si 1, minim 512MB cache
Unitati de disk interne	4 x 300 GB SAS 15k
Video	Controller video integrat

Adaptor convergent	8 porturi convergente 10 GEth
Controlor fiber channel	2 x 8 Gb/s dual port HBA
Sloturi I/O	Minim 6 x PCIe
Management	- consola virtuala la distanta; - posibilitatea efectuării managementului de la distanţă, in mod securizat, pe nivele de utilizatori; - funcţie de virtual media pentru flexibilitate în instalare, administrare şi mentenanţă.
Form factor	Compatibil Sasiu servere blade de productie
Sistem de operare	Tip UNIX „Mission-Critical” 64 biti, cu licente incluse pentru a se realiza un cluster de tip Cluster File System
Garantie	3 ani, 24x7 24h acoperire , 6 ore timp de reparatie HW

Articol 12	
Echipament backup pe banda	
Nr drive-uri	Minim 4 instalate, suport pana la 40
Tip drive-uri	LTO6
Nr sloturi	Minim 80 instalate, suport pana la minim 500
Benzi livrate	Minim 60 LTO6
Conectivitate	FC 8Gb
Partitionare	DA
Criptare	Posibila
Compresie	DA
Suport WORM	DA
Throughput	Minim 60 TB/ora
Capacitate maxima de stocare a librariei	Minim 3.4 PB
Form factor	Rackabil 6U
Sistem de alimentare	Redundante
Software	Licente software pentru backup si management corespunzatoare cu echiparea hardware. Functionalitati software: Backup automat, cu management centralizat, scalabil pina la 2500 de clienti Functionalitati software optionale: Deduplicare, backup cu zero-downtime, criptare, backup online al bazelor de date si al fisierelor deschise
Garantie	3 ani, 24 x 7 24h acoperire, 6 ore timp de reparatie HW

Articol 13	
Server monitorizare si gestiune incidente	
Procesor	2 x procesor min. 2.9 GHz, 6 core, 15 MB cache sau echivalent
Memorie	128 GB instalata, expandabila la minim 512 GB
Protectie memorie	Suport pentru :Advanced ECC, Memory Online Spare Mode, Lockstep Mode

Controlor HDD	SAS, suport RAID 0 si 1, minim 512MB cache
Unitati de disk interne	2 x 300 GB SAS 15k
Controlor retea	2 x 10Gb Ethernet
Controlor fiber channel	1 x 8 Gb/s dual port HBA
Sloturi I/O	Minim 2 x PCIe
Management	- consola virtuala la distanta; - posibilitatea efectuării managementului de la distanţă, in mod securizat, pe nivele de utilizatori; - funcţie de virtual media pentru flexibilitate în instalare, administrare şi mentenanţă.
Form factor	Compatibil Sasiu servere blade de productie
Sistem de operare	LINUX sau echivalent
Garantie	3 ani, 24x7 24h acoperire, 6 ore timp de reparatie HW

Articol 14	
Switch Ethernet agregare comunicatii interne	
Format	Rackabil, 19” Modular, 6 sloturi, maxim 12RU Arhitectura scalabila, performanta, bazata pe VoQ, in care componentele switch fabric sa fie pozitionate separat (nu pe cardul procesor/supervizor) iar modulele de retea I/O si modulele procesor/supervizor sunt conectate intre ele prin componentele switch fabric (multistage crossbar switching). Echipamentul sa fie recomandat in zona de Core a unui Data Center
Modul Management	Se asigura redundanta 1+1 la modulul de supervizor (management) Ofera conectivitate fizica (RJ45) pentru management Out-of-Band Ofera conectivitate fizica seriala pentru management local (consola) Este corespunzator tuturor facilitatilor cerute
Performanta Sasiu	Minim 3200Gbps capacitate totala switching Suport pentru minim 128 porturi de 10G
Echipare	Se asigura un numar de 96 porturi de 10G, de tip SFP+, populate cu 96 transceivere de tip 10Gbase-SR Modulul sau fiecare din modulele ce asigura, in total, necesarul de porturi de mai sus trebuie sa indeplineasca urmatoarele cerinte: - 256,000 intrari FIB IPv4

Articol 14	
	- 128,000 intrari FIB IPv6 - 64,000 ACL - 128,000 intrari in tabela MAC - MPLS, MPLS VPN L3 si MPLS VPN L2/VPLS - Routing/Forwarding L3 distribuit la nivel de modul (suporta si interfete Layer 3) - Facilitati de virtualizare Hardware prin care Switch-ul fizic poate fi virtualizat in mai multe Switch-uri logice (minim 4) la nivel de control plane, data plane si capabilitati de forwarding cu alocari de resurse hardware (exemplu procesor, memorie dinamica, stocare) - Facilitati de interconectare Data Centere la Layer 2 Ethernet peste o retea Layer 3 de tip IP (nu MPLS) si minim 6 locatii Data Center. Minim 250 VLAN-uri si minim 10 instante de retea trebuiesc suportate cu aceasta tehnologie. - 256MB packet buffer la 2 port de 10Gbps
Data Center	- Facilitati de virtualizare Hardware prin care Switch-ul fizic poate fi virtualizat in mai multe Switch-uri logice (minim 4) la nivel de control plane, data plane si capabilitati de forwarding cu alocari de resurse hardware (exemplu procesor, memorie dinamica, stocare) - Facilitati de virtualizare de tip N:1, prin care, minim 2 (echipamente interconectate prin interfete standard (de 10Gbps) sa se comporte ca un singur echipament L2, L3 si management (un singur fisier de configurare) prin unificarea zonelor de control (control plane) si sa ofere facilitati de routing distribuit si link aggregation distribuit. - Facilitati de interconectare Data Centere la Layer 2 Ethernet peste o retea Layer 3 de tip IP (nu MPLS) si minim 6 locatii Data Center. Minim 250 VLAN-uri si minim 10 instante de retea trebuiesc suportate cu aceasta tehnologie. - ISSU (In Service Software Upgrade), la nivel de Sasiu, prin care sistemul de operare sa permita upgrade-uri si repornirea de module software fara a afecta alte module software ce ruleaza in acelasi timp - Facilitati avansate de queuing cu VoQ pentru prevenirea blocarii de tip Head-Of-Line la nivel de port - Facilitati de link aggregation distribuit a.i. un switch ToR sa poata termina un link agregat pe 2 sau mai multe switchuri de core.
Management	SNMP v2c, v3 LLDP

Articol 14	
	SSH Acces rapid la loguri, debug si trap-uri (centralizate) cu nivelul de securitate aferent sFLOW
Securitate	ACL-uri wirespeed implementate in Hardware (IPV4 si IPV6) Protectia zonei de control impotriva atacurilor de tip DoS
Layer 2	4k Vlan-uri QinQ 802.1ag PBB Port mirroring local si la distanta
Layer 3	Ipv4 si Ipv6, static si dinamic (RIP, OSPF, BGP, IS-IS) ECMP PIM-SM, PIM-DM, PIM-SSM MSDP MPLS, VPLS Policy Routing
Disponibilitate	ISSU – In Service Software Upgrade la nivel de sasiu Non Stop Routing Graceful Restart Sa suporte topologie de tip inel, bazata pe standard, cu o convergenta mai rapida decat a protocolului RSTP Redundanta modul management de tip 1+1 Redundanta surse de alimentare de tip 4+2 (4 incluse, 2 redundante) Procesoare separate pentru sarcini separate: control plane, pentru protocoalele de convergenta rapida (exemplu: BFD) si managementul sasiului (exemplu: temperatura) Redundanta la nivel de switch fabric de tip N+1 Protectie control plane impotriva atacurilor DoS
Alimentare si Racire	Alimentare 200-240AC, compatibila Romania, cabluri de alimentare incluse Surse de alimentare suficiente pentru alimentarea si functionalitatea completa a componentelor cerute precum si nivelul de redundanta cerut cat si un surplus de 2000W in configuratia ceruta mai sus
Garantie	Hardware: 3 ani, 9 x 5 acces suport prin telefon si email,

Articol 14	
	inlocuire in urmatoarea zi lucratoare Software: 3 ani, 9 x 5 acces suport prin telefon si email, acces la upgrade-uri si update-uri de firmware. Accesul la upgrade-uri si update-uri de software se poate face si dupa expirarea garantiei incluse.

Articol 15	
Switch Ethernet comunicatii externe	
Format	Rackabil, 19” maxim 1RU Arhitectura moderna, flexibila, performanta destinata zonei de data center (Top of Rack). Sistem de operare modular care sa permita upgrade-ul de module software si permite monitorizarea si restartarea unor procese fara a afecta alte procese active
Performanta	Minim 480Gbps Switching Fabric Suport pentru minim 24 porturi de 10G sau 24 porturi de 1G 3.4GB buffer pachete 360mpps throughput
Echipare	Se asigura un numar de 24 porturi de 10G, de tip SFP+, populate cu 20 transceivere de tip 10Gbase-SR si 4 transceivere de 1Gbps, de tip 1000Base-T, conector RJ45 1 port ethernet RJ45 pentru management out-of-band 1 port consola
Data Center	- ISSU (In Service Software Upgrade), la nivel de Sasiu, prin care sistemul de operare sa permita upgrade-uri si repornirea de module software fara a afecta alte module software ce ruleaza in acelasi timp - Facilitati avansate de queuing cu VoQ pentru prevenirea blocarii de tip Head-Of-Line la nivel de port - Facilitati de virtualizare de tip N:1, prin care, 4 sau mai multe echipamente interconectate prin interfete standard (de 10Gbps) sa se comporte ca un singur echipament L2, L3 si management (un singur fisier de configurare) prin unificarea zonelor de control (control plane) si sa ofere facilitati de switching distribuit, routing distribuit si link agregation distribuit. - TRILL - EVB - Racire fata-spate - Suport FcoE, FIP Snooping, FSPF, Virtualizare N-port, servicii

Articol 15	
	fabric: name server, registered state change notification; FC traceroute, ping - DCB: 802.1Qbb (PFC), DCBX, 802.1Qaz (ETS)
Management	SNMP v1, v2c, v3 LLDP SNTP SSH Acces la distanta cu multiple niveluri de acces, bazate pe rol Configurare automata prin DHCP sFLOW SYSLOG Facilitati de limitare/filtrare loguri Autorizarea comenzilor prin Radius Mirroring trafic selectiv (cu ACL), pe port sau VLAN, local sau la distanta Permite aplicare de patch-uri software fara repornirea echipamentului 802.1ag 802.3ah DHCP Server Adaugare de utilizatori cu privielgii diferite si restrictionarea la comenzi critice (multiple privilegii sunt suportate) Acces rapid la loguri, debug si trap-uri (centralizate) cu nivelul de securitate aferent
Layer 2	4k VLAN-uri Suport pentru VLAN bazat pe port, protocol, MAC Guest VLAN VLAN Mapping (manipulare VLAN) FlowControl 120 000 dimensiunea tabelei MAC 120 grupuri pentru link aggregation cu pana la 12 porturi / grup LACP, facilitati de convergenta rapida pentru LACP Reverse ARP, APR proxy
Layer 3	Rutare statica si dinamica Ipv4 si Ipv6 pentru RIP, OSPF, BGP, IS-IS ECMP VRRP, VRRP extins Policy Routing

Articol 15	
	802.1ag 16000 dimensiunea tabelei de rutare Tunelare Ipv6: ISATAP si 6to4 Multicast Routing
Securitate	TACACS+ ACL Layer 3
QoS	Crearea de clase de trafic cu clasificatori bazati pe ACL SP, WFQ, WRED, ECN, WDRR
Disponibilitate	ISSU – In Service Software Upgrade la nivel de sasiu BFD aplicabil protocoalelor de rutare, VRRP si solutiei de virtualizare switchuri oferind convergenta de sub 50msec DLDP Graceful Restart Redundanta surse de alimentare de tip 1+1, interne Redundanta la bateria de ventilatoare de tip 1+1
Alimentare si Racire	Racire de tip fata-spate Alimentare 200-240AC, compatibila Romania Consum maxim 380W Surse de alimentare suficiente pentru alimentarea si functionalitatea completa a componentelor cerute precum si nivelul de redundanta cerut
Garantie	Hardware: 3 ani, 9 x 5 acces suport prin telefon si email, inlocuire in urmatoarea zi lucratoare Software: 3 ani, 9 x 5 acces suport prin telefon si email, acces la upgrade-uri si update-uri de firmware. Accesul la upgrade-uri si update-uri de software se poate face si dupa expirarea garantiei incluse.

Articol 16	
Switch Ethernet management infrastructura IT	
Sasiu	Rackabil, 19’’ Arhitectura modulara
Porturi	Minim 24 porturi 10/100/1000Mbps; 4 x SFP+ ce pot acomoda atat interfete de 1Gbps cat si 10Gbps si pot functiona simultan cu cele de tip RJ45 existente. Se populeaza cu 2 x 10Gbps de tip 10Gbase-SR Suporta module cu 4 x 10Gbps (SFP+), cu 14 x 10/1000/1000 sau modul cu 14 x SFP, fiecare modul adaugat poate functiona simultan cu

Articol 16	
	cele 24 x 1Gbps si cele 4 x SFP+ 1 x RJ-45 serial console port 1 x USB
Management	CLI WEB (HTTPS) SSH Telnet Facilitati operare USB pentru a copia fisiere de pe si pe o memorie de tip FLASH USB Optiuni DHCP: DNS Relay si SMTP Redirection, DHCP Server, DHCP Client, DHCP Optiunea 82 sFLOW: bazat pe ASIC SNMP v1, v2c, v3 RMON Sa suporte mai multe fisere de configurare stocate VLAN de Voce ce permite asignarea automata a VLAN-ului si prioritatii telefoanelor IP LLDP-MED IGMP v1, v2, v3, ASM (Any-Source Multicast), SSM (Source-Specific Multicast) pentru Multicast IPv4 PIM pentru aplicatii multicast de tip IPv4, IPV6; PIM-SM, PIM-DM, PIM-SSM MLD Snopping Acces la upgrade-uri de firmware nelimitat
QoS	Bazate pe clasificatori cu ACL, Ipv4 si Ipv6 802.1p DSCP Prioritizare, marcare, remarcare WRR, SP, WFQ, WDRR, SP+WDRR
Layer 2	4094 VLAN-uri simultane IEEE 802.1ad
Layer 3	Suporta interfete setate in mod rutat (nu necesita configurare vlan pentru routing) ci adresele ip se pot atasa interfetelor fizice Suport nativ IPv4 Suport nativ IPv6 RIP, RIPv2, RIPng OSPF, OSPF v3 BGP 4, BGP 4+

Articol 16	
	IS-IS, IS-ISv6 BFD VRRP Policy Based Routing PIM-SSM, PIM-DM si PIM-SM (atat pentru IPv4 Cat si pentru IPv6) ECMP MPLS, MPLS VPN, MPLS-TE, VPLS
Securitate	URPF 802.1X cu posibilitate asignare dinamica a facilitatilor de QoS, ACL si VLAN VLAN de tip Guest IP Source Guard MCE-MVRF PKI ACL – Access Control Lists de tip wirespeed, in Hardware (bazate pe TCAM)
Disponibilitate si fiabilitate	Sa suporte topologie de tip inel, bazat pe standard, cu o convergenta mai rapida decat a protocolului RSTP Facilitati de virtualizare de tip N:1, prin care, mai multe echipamente interconectate (minim 8) prin interfete standard de 10Gbps (SFP+) sa se comporte ca un singur echipament L2, L3 si management (un singur fisier de configurare) prin unificarea zonelor de control (control plane) si sa ofere facilitati de switching distribuit, routing distribuit si link agregation distribuit.
Monitorizare si troubleshooting	Port Mirroring OAM – IEEE 802.3ah CFD – IEEE 802.1ag
Surse de alimentare	1 sursa de alimentare AC inclusa, 220V, compatibila Romania Posibilitate Sursa de alimentare redundanta externa
Memorie	512 MB compact flash, 512 MB DDR SDRAM
Throughput	150 milioane pps
Capacitate de Routing/Switching	204 Gbps
Marime tabela de routare	16000 intrari
Marime tabela	32000 intrari

Articol 16	
adrese MAC	
Mediul de operare suportat	Interval temperatura de functionare: 0ºC – 45ºC Interval umiditate relativa in operare: 15% - 90%
Garantie	Hardware: 3 ani, 9 x 5 acces suport prin telefon si email, inlocuire in urmatoarea zi lucratoare Software: 3 ani, 9 x 5 acces suport prin telefon si email, acces la upgrade-uri si update-uri de firmware. Accesul la upgrade-uri si update-uri de software se poate face si dupa expirarea garantiei incluse.

Articol 17	
Echipament modular servicii de securitate	
Sasiu/Arhitectura	Rackabil, 19’’ Arhitectura modulara, distribuita L2 si L3, cu 5 sloturi: 2 pentru Management/Switch Fabric si 3 pentru module de retea, cu o densitate de porturi de pana la 26 porturi de 10Gbps sau 140 porturi de 1Gbps sau 12 x 40G si module de servicii cum ar fi: Firewall si IPSec VPN, Load Balancing. Backplane de tip pasiv
Echipare	2 x Module Management/Switch Fabric ce functioneaza simultan, balansat si redundant si ofera o performanta de pana la 380Gbps fiecare. Fiecare modul ofera : 1 x RJ45 de tip serial pentru consola, 1 x RJ45 de tip Ethernet 10/100/1000Mbps pentru management out of band 2 x 10Gbps echipate cu 10Gbase-SR, FO MM, 850nm, 300m Modul cu 8 x 10Gbps, de tip 10Gbase-SR, cu suport pentru : - 120 000 adrese MAC - 120 000 rute Ipv4 - OAM 802.1ag - MPLS, MPLS VPN L2/VPLS si L3 - Multi-VRF (2000 instante) Modul Firewall/VPN cu urmatoarele facilitati: - 2 x 1Gbase-T si 2 x 1Gbps combo (1Gbase-T sau SFP) - 1 x Port USB - 1 x Port Serial Consola

Articol 17	
	- 6 Gbps throughput Firewall - 1.8 M sesiuni concurente - 50K Sesiuni pe secunda - 18,000 politici de securitate - 2Gbps 3DES throughput - 8000 tunele L2TP - 5000 tunele IPSec - 4096 tunele GRE - 4000 VLAN-uri - Packet Filtering - ACL extinse/avansate, bazate pe timp, MAC - 250 Firewall-uri Virtuale - 250 zone de securitate - Dynamic Packet Filtering - Support firewall pentru : DNS, FTP, ILS, NBT, MSN, PPTP, SIP, HTTP, SMTP, RTSP, H.323, MGCP, Blocare Java/Active X, Pachete fragmentate - Urmarirea starii protocoalelor si verificarea conformitatii acestuia cu standardul - Protectii la : Land, Smurf, Fraggle, WinNuke, Ping of Death, Tear Drop, IP spoofing, SYN flood, ICMP flood, UDP flood, DNS Query flood si ARP spoofing defense, ARP active reverse lookup, TCP flag packet attack defense, TCP Intercept, Large ICMP packet attack defense, Address/port scanning defense, DoS/DDoS, Filtru protectie SQL Injection - Mod de functionare: Firewall Transparent, Routing, Mixt - Protectie aplicatii P2P (BT, eDonkey) cu limitare largime de banda NAT ALG pentru: DNS, FTP, MSN, H323, RTSP, SQLNET, SIP, PPTP, ICMP, TFTP, HTTP, - Filtrare e-mail: SMTP – adresa e-mail (sursa si destinatie), subiect e-mail, continut e-mail, atasament e-mail (nume si continut), dimensiune mail; POP3 - mail address (sursa si destinatie), Mail subject, Mail attachment (nume si continut) - 4k subinterfete vlan Filtrare web: URL Hostname, URL IP, HTTP Header, HTTP Body, ActiveX, Java Applet - Disponibilitate: Active/Active, Active/Pasive cu sincronizare: configurare politici firewall, sesiuni firewall, SA-uri VPN IPSec
Management	CLI WEB (HTTPS)

Articol 17	
	SSH Telnet FTP, TFTP, SFTP pentru transfer fisiere Posibilitatea inspectarii calitatii conexiunilor si a serviciilor cu verificare pachete pierdute, delay, jitter Creearea utilizatorilor de sistem de tip ierarhic pentru access la comenzile sistemului SNMP v1, v2c, v3 RMON NTP LLDP Ofera posibilitatea stocarii a doua fisiere de firmware pentru backup in timpul upgrade-ului Syslog Debug
Multicast	IGMP v1/v2/v3 cu utilizare Any-Source Multicast (ASM) si Source-Specific Multicast (SSM) IGMP Snooping MSDP LLDP-MED PIM-DM, PIM-SM, PIM-SSM MBGP Multicast VLAN pentru IPv4 si IPv6 Voice VLAN prin asignare automata a VLAN-urilor si prioritatii pentru telefoane IP
Layer 2	4096 VLAN-uri VLAN bazat pe port, protocol, Subnet IP, MAC Facilitati manipulare VLAN (VLAN Mapping) Private VLAN Guest VLAN pentru 802.1x Tunelare BPDU GVRP Port Mirroring: 4 Grupuri de mirror, fara limita de porturi / grup de mirror, mirror intre module MLD Snooping RFC 3069 IEEE 802.1ad QinQ Frame-uri Jumbo

Articol 17	
	IEEE 802.1ag Storm Protection: broadcast, multicast, unicast necunoscut
Layer 3	uRPF Graceful Restart for OSPF, BGP, IS-IS UDP Helper VRF Stiva duala IPv4 si IPv6 Tunelare IPv6 DHCPv6 RIP v1/v2, RIPng OSPFv2/OSPFv3 BGP4, MBGP, BGP pentru adresare IPv6 IS-IS, IS-ISv6 ECMP Policy Routing ICMPv6 Tunelare IPv6: IPv6 peste IPv4, ISATAP MPLS, MPLS L3 VPN, MPLS L2 VPN, VPLS
QoS	Clase de trafic bazate pe ACL standard si extinse pe baza adresa MAC, adrese IP sursa si destinatie (IPv4 si IPv6), protocol, port Remarcarea pachetelor cu 802.1p, Precedenta IP, si DSCP 8 cozi de prioritizare per port Precedenta 802.1p Precedenta DSCP,ToS Scheduling: SP, WRR, SP+WRR, WFQ Evitarea congestiei: WRED, Tail-Drop Traffic Shapping Traffic Metering Traffic Accounting Traffic Policing: Commited Access Rate (intrare , iesire): granularitate 64kbps
Securitate	ACL IPv4 si IPv6 ACL standard si extinse ACL bazate pe VLAN ACL de tip ingress/egress 802.1X cu posibilitate asignare dinamica a facilitatilor de QoS, ACL si VLAN precum si integrarea intr-o solutie de Network Access

Articol 17	
	Control (NAC) uRPF 802.1X Server IP Source Guard ACL bazate pe perioada de timp Sa suporte module de Firewall si VPN/IPSec, VPN SSL
Disponibilitate si fiabilitate	Sa suporte topologie de tip inel cu o convergenta de sub 100ms bazata pe standard Facilitati de virtualizare de tip N:1, prin care, minim 2 echipamente interconectate prin interfete standard (de 10Gbps, minim 6) sa se comporte ca un singur echipament L2, L3 si management (un singur fisier de configurare) prin unificarea zonelor de control (control plane) si sa ofere facilitati de switching distribuit, routing distribuit si link agregation distribuit. Redundanta 1+1 la nivel de switch fabric (cu load sharing) Redundanta 1+1 la nivel de sursa de alimentare Backplane pasiv Toate modulele sunt de tip HotSwap BFD VRRP Cai diferite pentru Control si Servicii a.i. un anumit serviciu configurat sa nu influenteze controlul asupra echipamentului LACP: pana la 120 grupuri de trunk
Monitorizare si troubleshooting	Port Mirroring, Remote Port Mirroring, Mirroring intre module Flow Mirroring OAM – IEEE 802.3ah IEEE 802.1ag
Surse de alimentare Ventilatoare Racire	2 surse de alimentare AC incluse, 220V, maxim 1500W fiecare, redundanta 1+1 O baterie de ventilatoare
Memorie	64 MB flash, 512 MB SDRAM
Throughput	240 mpps
Marime tabela de routare	250000 intrari ipv4, ipv6
Marime tabela adrese MAC	Pana la 500000 intrari
Interconectare	Pentru indeplinirea cerintelor de securitate si disponibilitate ale proiectului, va include interconectarea pentru detectia si prevenirea intruziunilor care sa respecte urmatoarele cerinte:

Articol 17	
	▪ Să permita securizarea utilizatorului cu ajutorul unui mecanism in-line. ▪ Să protejeze zonele interne de atacuri. ▪ Să ofere opțiuni de înaltă disponibilitate, astfel încât o simplă pană de curent să nu producă pagube majore; sistemul va permite configurarea ulterioara in mod high availbility de tip activ-activ, sau activ-pasiv ▪ Să poată fi completată cu alte soluții de prevenție ce oferă protecție pentru cazurile cu lățime mare de bandă, cum ar fi centrele de date. ▪ Să ofere posibilitatea unui număr minim de 100.000 de conexiuni pe secundă și 6.000.000 de sesiuni concurente. ▪ Se va avea in vedere posibilitatea conectarii a cel putin 5 segmente de tip curpu (minim 10 porturi cupru), si minim 5 segmente fibra (minim 10 porturi 1G fibra) ▪ Traficul inspectat trebuie sa fie de minim 1,5Gb/s, la capacitate maxima, cu cel putin jumatate din filtre activate. ▪ Detecția atacului să se poată face fie bazându-se pe semnătură, fie pe protocoale ce identifică anomaliile. ▪ Inspecția pachetului să se poată face la nivel de header și payload. ▪ Să fie posibilă detecția de tip multi-tier pentru inspecție paralelă a unui flow în hardware. ▪ Să fie disponibil un motor de securitate cu inspecție bazată pe flow-uri de date. ▪ Să se poată face inspecția atât a traficului IPv4 , cât și IPv6. ▪ Protecția să se facă atât la nivel de aplicație(sisteme de operare, aplicații), la nivel de infrastructură(a lărgimii de bandă și a echipamentelor de rețea: routere, firewall), cât și prin blocarea sau limitarea ratei (rate limit) a aplicațiilor de tip Instant Messaging, Peer-2-Peer, Streaming Media. ▪ Să fie oferite facilități de management de trafic cu reguli de permitere, blocare, respective limitare a ratei traficului (rate limit). ▪ Să fie permisă captura selectivă a traficului ce traversează IPS-ul. ▪ Protecția să se facă la nivel de vulnerabilitate (nu doar exploit).

Articol 17	
	▪ Sa includă protecție pentru următoarele atacuri: Worm, Virus,Trojan, P2P, VoIP, Phishing, Suspicious, Reconnaissance, Walk-in-Worm, Backdoor, Spyware, DDoS, Bandwidth Hijacking, Blended Threat, DoS: SYN Floods, Established Connection Floods, Connections Per Second Floods, Vulnerability Protection, Attack Tool Protection, Threshold Filters ▪ Sistemul trebuie sa fie certificate de organisme independente de testare a dispozitivelor de protectie la nivel de retea; minim ICSA Labs, NSS si TOlly Group. ▪ Sa suporte inspectia traficului de tip VLAN 802.1Q, VLAN QinQ 802.1ad, GRE, MPLS ▪ In cazul in care apare o eroare de software, sau congestie, sistemul trebuie sa treaca in mod transparent, sa nu blocheze traficul, si sa functioneze in mod L2 fallback ▪ Sistemul trebuie sa ofere suport pentru inspectie in cazul traficului asimetric ▪ Filtrele configurate trebuie sa poata fi configurate indepemendent, iar in cazul in care un filtru incarca prea mult sistemul, sa notifice administratorul si sa se dezactiveze, pentru a preveni pierderea de trafic ▪ Sistemul trebuie sa suporte efectuarea de actualizari in timp ce dispozitivul de securitate ruleaza, fara a afecta traficul inspectat ▪ Sistemul de securitate trebuie sa prezinte semnaturi pentru exploituri, vulnerabilitati, incercari de furt al identitatii, spyware, virusi, activitati de explorare a infrastructurii, detectie trafic de instant messaging, trafic P2P, cat si detectia de aplicatii care controleaza fluxul de date multimedia ▪ Actualizarile de filtre trebuie sa se faca minim odata pe saptamana ▪ Baza de date de reputatii pentru IPv4, IPv6, si nume DNS trebuie sa permita sortarea dupa tara de origine, adresa IP, relevanta si descrierea potentialului pericol detectat ▪ Sa ofere intern, sau prin intermediul unei aplicatii, posibilitatea de creare de filtre customizate, care sa fie aplicate ulterior pe traficul inspectat ▪ Actiunile trebuie sa fie minim: blocare – drop pachet, tcp reset, permitere, notificare, captura de pachet, limitare bandwidth, cat si carantinare

Articol 17	
	▪ Solutia trebuie sa ofere suport pentru cel putin 256 de VLAN translated groups ▪ Solutia de trebuie sa detecteze si blocheze atacuri bazate pe adrese sau nume DNS cunoscute ca fiind generatoare de spyware, phishing sau botnet. Traficul trebuie sa poata fi identificat pe baza de locatie, si sa permita detectarea rapida a tarii din care provine traficul ▪ Va detecta si bloca atacuri asupra infrastructurii de retea, incluzand switchuri, routere, firewalluri, si switchuri wireless. De asemenea, trebuie sa ofere protectie pentru protocoale folosite in telefonia IP ▪ Solutia trebuie sa ofere suport pentru trapuri SNMP si transmitere mesaje prin intermediul syslog • Solutia trebuie sa permita managementul local, prin intermediul unei interfete web, la distanta prin intermediul unei interfete SSL, cat si sa permita extinderea catre managementul prin intermediul unui server specializat.
Mediul de operare suportat	Interval temperatura de functionare: 0ºC – 45ºC Interval umiditate relativa in operare: 10% - 95%
Garantie	Hardware: 3 ani, 9 x 5 acces suport prin telefon si email, inlocuire in urmatoarea zi lucratoare Software: 3 ani, 9 x 5 acces suport prin telefon si email, acces la upgrade-uri si update-uri de firmware. Accesul la upgrade-uri si update-uri de software se poate face si dupa expirarea garantiei incluse.

Articol 18	
Echipament echilibrare incarcare si degrevare SSL	
Caracteristici	- Rackabil in rack 19’’ - Trebuie sa aiba minim un procesor quad-core si min. 32GB memorie RAM - Trebuie sa aiba cel putin doua hard diskuri interne de 1TB configurabile RAID1 - Trebuie sa aiba surse de alimentare redundante - Trebuie sa aiba cel putin 8 interfete 10GbE, din care două echipate cu transceivere SR - Trebuie sa suporte failover de tip stateful cu pastrarea conexiunilor

	dinspre clienti spre servere si invers fara a impacta traficul - Trebuie sa suporte cel putin 1.600.000 conexiuni L7 pe secunda - Trebuie sa suporte algoritmii recunnoscuti de balansare a conexiunilor - Trebuie sa suporte multiplexarea conexiunilor HTTP pe partea de server - Trebuie sa suporte conexiuni multiple intre browser si servere pentru a imbunatati transferul HTTP - Trebuie sa suporte compresie trafic HTTP, minim 18Gbps - Trebuie sa ofere functii de sanitizare a header-elor HTTP - Trebuie sa suporte optimizarea traficului SSL si TCP in functie de aplicatie - Trebuie sa suporte module de monitorizare a aplicatiilor de pe servere - Utilizeaza un set de tehnologii de comutare de nivel 4 pentru distribuirea incarcarii si de nivel 7 pentru comutare bazata pe continut - Posibilitatea de a aplica translatii de adresa IP sursa sau destinatie pentru ferme de servere - Capacitatea de a utiliza tehnici de persistenta pe baza adresei IP sursa/desinatie, cookie, header HTTP sau identificatorul sesiunii SSL. - Trebuie sa asigure tehnologii de accelerare SSL, licentiate pentru cel putin 25000 TPS utilizand chei de 2048 biti
Garantie	3 ani, 24 x 7 24h acoperire

3.1.8. Cerinţe pentru dotările tehnice necesare amenajării locaţiei

Pentru instalarea sistemului informatic proiectat se vor avea vedere următoarele cerinţe referitoare la dotările tehnice necesare amenajării locaţiei unde vor fi instalate echipamentele descrise în secțiunea 3.2.1 pe care va rula sistemul:

Articol 1	
Instalaţie de electroalimentare	
Caracteristici	Rack-urile de echipamente vor fi protejate la căderi de tensiune de un sistem UPS (Uninterruptible Power Supply). Sistemul UPS trebuie să respecte specificațiile tehnice de mai jos, care sunt considerate minimale: • Să utilizeze o arhitectură modulară şi scalabilă, atât pentru putere cât şi

	pentru timpul de susținere pe baterii, instalată în rack-uri cu dimensiunile fizice ale rack-urilor standard de servere, pentru a putea fi integrat în rândurile de rack-uri de echipamente. • Să conțină bypass de mentenață și sistem de distribuție electrică pentru sarcinile IT și comunicații integrate în sistemul UPS. • Să dispună de accesorii opționale care să facă posibilă integrarea într-o soluție de răcire pentru densități mari de putere împreună cu rack-urile de echipamente protejate. • Sa fie în tehnologie on-line • Va fi dimensionat cu o capacitate inițială de minimum 125 KW, în redundanța N+1 și va permite upgrade pentru o capacitate de minimum 250 KW • Sa utilizeze baterii VRLA (Valve Regulated Lead Acid), fără mentenanță • Sa conțină un sistem de protecție a bateriilor cu circuite de monitorizare și control care să limiteze nivelul de descărcare a bateriilor • Sistemul UPS să fie controlat de două module de control, redundante, care să poată fi înlocuite, în caz de defectare, fără a fi necesară oprirea sistemului UPS. Fiecare modul de control trebuie să aibă căi de comunicații separate, izolate optic, la modulele de putere și comutare statică și vor fi alimentate din surse de alimentare redundante. • Distorsiunea tensiuni la ieșire: < 2%. • Sa aiba o eficienta mai mare de 95% • Tensiune nominala de intrare: 400V 3PH, 480V 3PH (40-70Hz) • Tensiune nominala de iesire: 400V 3PH, 480V 3PH (50Hz) • Distorsiuni armonice ala tensiunii la ieșire: <2%THD pentru incarcari liniare si <3%THD pentru incarcari neliniare • Operare în condiții de suprasarcina: 10 minute la 125% încărcare, 60 secunde la 150% incarnare Sistemul UPS va fi racordat la tabloul de distributie general al spatiului tehnic. Sistemul UPS va fi alimenta tot ansamblul de echipamnete rack propopuse in proiect pe doua cai redundante. In acest sens, ofertantii vor prevedea toate elementele si materialele necesare realizarii acestor lucrari, inclusiv un tablou de distributie electrica aferent subansamblului de echipamente rack propusa astfel incat sa poata fi posibila intreruperea alimentarii cu energie electrica pe fiecare rack si pe fiecare ramura.

Articol 2	
Instalație de climatizare	
Caracteristici	Sistemul de racire va fi dimensionat astfel incat sa fie asigurata o incarcare de 160kW. Sistemul va fi compus din unitate de producere apa racita, unitate de distributie a agentului si unitati interne de racire a echipamentelor. Sistemul trebuie sa permita functionare in regim de tip free-cooling si sa foloseasca agent de racire de tip apa-glycol. Echipamentele de răcire pentru pentru rack-urile de echipamente vor fi instalate în rândurile de rack-uri. Aceasta amplasare a echipamentelor de răcire profesionale pentru centre de date a fost luată în considerare din următoarele motive: • Spațiul salii de calculatoare nu permite instalarea de echipamente profesionale de răcire profesionale perimetrale, cu livrarea aerului rece prin grile în pardoseala supraînălțată situate în fata rack-urilor, fără a avea limitări asupra debitelor fluxurilor de aer și asigurării condițiilor de răcire pentru densități mari de putere în rack-uri; • Echipamentele de răcire în rând minimizează lungimea fluxurilor de aer, prin furnizarea aerului rece în față și recuperarea aerului cald din spatele rack-urilor de echipamente, în acest fel puterea necesară pentru ventilatoare pentru circulația volumului de aer fiind diminuată crescând eficiența energetică a sistemului de răcire; • Permite mai ușor adoptarea de soluții pentru izolarea fluxurilor de aer cald sau rece în interiorul sălii de calculatoare, maximizând eficiența energetică globală a sistemului de răcire. Echipamentele de racire trebuie sa respecte urmatoarele caracteristici: • Alimentare electrică 200-240V, 50/60 Hz. • Capacitate de răcire minimum 18 KW • Refularea aerului rece să se facă frontal, orizontal, pe toată înălțimea echipamentului • Colectarea aerului cald trebuie să se facă în partea din spate a echipamentului de răcire, pe toată înălțimea echipamentului. • Echipamentele de răcire vor avea interfața de rețea Ethernet care să permită administrarea, monitorizarea și notificarea evenimentelor într-o rețea prin TCP/IP. Ofertantii vor dimensiona numarul de echipamente de racire functie de

	solutia propusa. Astfel, se va prezenta justificarea solutiei propuse functie de puterea instalata.

Articol 3	
Echipamente pentru stingerea incendiilor	
Caracteristici	Pentru securizarea la incendiul a infrastructurii instalate in centrul de date, se are in vedere extinderea sistemului de stingere a incendiilor cu gaz inert deja existent.

Pentru instalarea dotărilor tehnice (mijloacelor fixe) aferente amenajării centrului de date ce va găzdui echipamentele de calcul din cadrul proiectului, vor trebui desfăşurate următoarele activităţi:

- Instalarea şi configurarea echipamentelor de tip sursă neîntreruptibilă
- Instalarea tabloului electric de distribuţie către echipamentele de centru de date şi echipamentele de calcul
- Instalarea şi configurarea echipamentelor de climatizare
- Instalarea şi testarea sub presiune a circuitului de agent refrigerant
- Instalarea şi configurarea sistemului de semnalare şi stingere a incendiilor
- Testarea sistemelor de alimentare, climatizare şi stingere a incendiilor

3.1.9. Cerinţe pentru reţeaua de comunicaţii

Arhitectura reţelei de comunicaţii este o componentă a arhitecturii sistemului informatic. Reţeaua de comunicaţii va fi construită astfel încât să răspundă cerinţelor sistemului informatic şi să asigure conectarea componentelor harware descrise anterior. Reţeaua de comunicaţii va fi găzduită de beneficiar la sediul acestuia sau într-o locaţie desemnată de beneficiar, ce va asigura condiţiile de alimentare şi climatizare a echipamentelor de calcul, precum şi accesul la infrastructura WAN broadband.

Sistemul de comunicatii de tip LAN trebuie furnizat şi trebuie să permită:

- interconectarea echipamentelor descrise mai sus;
- rețeaua de date trebuie să suporte stiva de protocoale TCP / IP V4 si V6;
- furnizorul trebuie să livreze elementele hardware și software de rețea, care sunt necesare pentru functionarea corectă și sigură a soluției;

- viteza de interconectare între switch-uri de minim 10Gbps;
- conexiuni redundante de la serverele cu interfețe de minim 1Gbps.

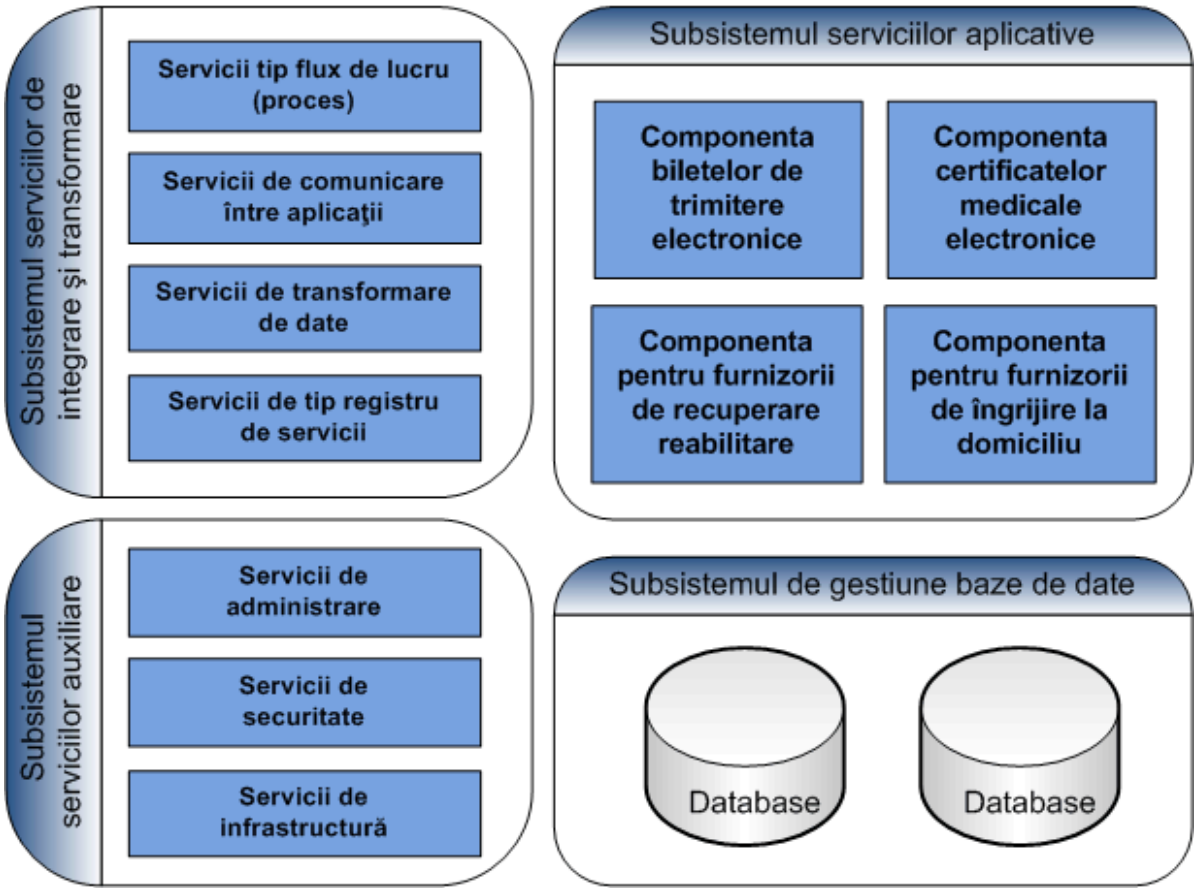
Furnizorul trebuie să instaleze şi să certifice după caz, reţeaua de cablare pentru toate componentele arhitecturale ale sistemului.

Securitatea reţelei de comunicaţii:

- Conexiunile externe trebuie protejate de cel putin un firewall. Acest lucru se aplică locaţiei unde este obligatoriu ca firewall-ul sa fie redundant cel puțin în modul activ-standby
- Arhitectura sistemului trebuie să fie echipată cu câte un dispozitiv inline IPS / IDS, senzori care vor fi plasati pe (cel puțin) segmentul de rețea conectat la interfața internă(e) a firewall-ului ce deservește conexiunile externe ale sistemlui. Prin conexiuni externe înțelegem legăturile cu alte rețele, de genul internet sau rețea WAN
- Fișierele de log ale IPS / IDS trebuie să fie arhivate și trebuie să fie disponibile pentru consultare.

3.1.10. Subsisteme funcţionale

Diagrama următoare prezintă arhitectura funcţională a sistemului informatic propus:



În continuare sunt detaliate subsistemele logice şi componentele constituente ale acestora.

3.1.10.1. Subsistemul serviciilor aplicative

Serviciile de tip aplicativ sunt serviciile principale pe care trebuie să le ofere sistemul din punct de vedere funcţional. Acest subsistem trebuie să asigure implementarea tuturor cerinţelor expuse în acest document.

Serviciile aplicative trebuie să respecte principiile „cuplajului slab”, pentru a permite extinderea funcţionalităţilor într-o manieră scalabilă, folosind tehnologii de tip servicii web. Această cerinţă este cu atât mai importantă cu cât se preconizează o dezvoltare iterativă, cu adăugarea de noi funcţionalităţi specifice în cazul adăugării de noi funcţionalităţi pentru sistemele informatice constituente ale PIAS.

Subsistemul serviciilor aplicative conţine componentele principale ale noului sistem:

- Componenta Biletelor de trimitere electronice

- Componenta Certificatelor de concedii medicale electronice
- Componenta pentru Furnizorii de recuperare-realbilitare
- Componenta pentru Furnizorii de ingrijire la domiciliu

Subsistemul serviciilor aplicative va conține pentru Componenta biletelor de trimitere electronice și pentru Componenta certificatelor medicale electronice următoarele module:

- Modulul Central – conține funcționalitățile de control și monitorizare a componentei, precum și o serie de componente specializate, de tip Business Intelligence, Raportare in format fix, precum și componentele pentru serverele de aplicatie, capabile să ruleze în mod disponibilitate inaltă activ-activ
- Modulul Prescriere – conține funcționalitățile de suport pentru generarea documentelor medicale de tip bilet de trimitere electronic, respectiv certificat de concediu medical electronic
- Modulul Validare – conține funcționalitățile pentru verificarea corectitudinii documentelor medicale prescrise (bilet de trimitere, respectiv certificat de concediu medical)
- Modulul Accesare – conține funcționalitățile pentru punerea la dispoziția utilizatorilor a documentelor medicale în format electronic: bilete de trimitere, respectiv certificate de concediu medical. Utilizatorii sunt personalul CJAS/CNAS, precum și furnizorii de servicii cărora li se adresează biletele de trimitere, respectiv reprezentanții autorităților și instituțiilor care necesită accesarea certificatelor de concedii medicale.

Subsistemul serviciilor aplicative va conține următoarele module pentru Componenta pentru furnizorii de recuperare și reabilitare, respectiv pentru Componenta pentru furnizorii de îngrijiri la domiciliu:

- Modulul Central – conține funcționalitățile de control și monitorizare a componentei, precum și o serie de componente specializate, precum și componentele pentru serverele de aplicație, capabile să ruleze în mod disponibilitate înaltă activ-activ
- Modulul de acces la istoricul de DMR din sistemul DES – conține funcționalitățile de accesare a DES și de punere la dispoziția furnizorului de servicii medicale datele din istoricul pacientului, conform procedurilor implementate în sistemul DES

- Modulul de filtrare a datelor medicale raportate de furnizorul de servicii și transmitere în sistemul DES a DMR, conform procedurilor și mijloacelor tehnice disponibilizate de sistemul DES

Cerinţe pentru componenta de Business Intelligence din Modulul Central

Componenta de Business Intelligence trebuie sa ofere posibilitatea de a rula pe diverse platforme hardware precum si pe sistemele de operare majore existente pe piata (Windows, Linux, Unix).

Componenta de business intelligence va oferi o interfata web prin care utilizatorii sa poata interactiona cu toate componentele sistemului, atat pentru accesul la rapoarele si tablorile de bord (dashboard) existente, cat si pentru a crea noi analize ad-hoc. Componenta de BI va oferi posibilitatea de a prezenta informatia in formate multiple cum ar fi grafice, tabele, tabele pivotante, iar atunci cand un raport include o reprezentare multipla (ex text si grafic) a aceleiasi informatii, componenta de business intelligence va trebui permită afișarea informației fără a repeta execuția interogării. Pentru o mai bună vizualizare și înțelegere a informațiilor afișate, este necesar ca aplicația de raportare să poată afișa valori identificate ca și critice, să semnaleze depășirea unor praguri ale acestor valori, să semnaleze apariția unor evenimente.

Componenta de BI trebuie să ofere posibilitatea modelarii datelor si prezentarii informatiei intr-un format familiar utilizatorilor finali, astfel incat acestia sa poata accesa informatiile disponibile fara a cunoaste structura si particularitatile fiecarei baze de date accesate. Acest nivel de metadate expus utilizatorilor trebuie sa fie comun la nivelul tuturor modulelor sistemului de raportare si analiza.

În cadrul acestei componente, rapoartele vor putea fi construite pe diverse surse de date, de pe platforme diferite (Oracle, SQL Server, DB2, SQL Anywhere), in mod transparent pentru utilizatorul final. Rapoartele analitice vor putea fi construite pe un numar variabil de interogari analitice, fara ca instrumentul de business intelligence sa limiteze numarul de astfel de interogari.

Pentru intelegerea mai buna a informatiilor prezentate, componenta de BI va oferi posibilitatea de drill down (afisarea datelor aggregate si detalierea acestora pe baza ierarhiilor implementare) sau drill through (posibilitatea de a naviga catre un alt raport/dashboard preluand contextul raportului din care s-a declansat actiunea), atat pentru rapoarte cat si pentru grafice.

Componenta de business intelligence va permite accesarea informatiilor prin portalul de business intelligence, iar prin folosirea functionalitatilor instrumentului de raportare, va permite salvarea rapoartelor in diferite formate cum ar fi Excel, PDF, Word, HTML, Powerpoint. Componenta de business intelligence va permite modificarea tablourilor de bord sau a rapoartelor (fara costuri de licentiere suplimentare) si va oferi posibilitatea includerii rapoartelor/graficelor in tablouri de bord pentru toti utilizatorii finali, fara costuri de licentiere suplimentare.

Componenta de BI nu trebuie sa limiteze accesul utilizatorior la o singura sursa de date, permitand combinarea rezultatelor obtinute de pe platforme diferite la momentul interogarii, astfel incat setul de date rezultat sa fie unitar. Pentru functionare optima, componenta de business intelligence trebuie sa foloseasca capabilitatile bazelor de date sursa, fara a necesita replicarea datelor pe un server suplimentar si sa implementeze un mechanism de generare a interogarilor care sa tina seama de tehnologia de baze de date accesata.

Utilizatorii sistemului vor putea fi preluati din sisteme LDAP, insa componenta de raportare va trebuie sa ofere capabilitati proprii de definire a rolurilor pentru restrictionarea in detaliu a accesului la rapoarte.

Componenta de analiza si raportare manageriala trebuie fie scalabila si sa dispuna de mecanisme de clustering a componentelor (de prezentare sau la nivel de server de acces la date), astfel incat sa poata fi folosite resurse hardware suplimentare.

Instrumentul de raportare trebuie sa ofere un mecanism de programare a executiei rapoartelor sau a preincarcarii in serverul de BI a unui set de date astfel incat sa minimizeze timpii de executie ai interogarilor analitice, in functie de sursele de date.

Componenta de BI va trebui sa permită vizualizarea trasabilităţii unei componente dintr-un raport, si dacă aceasta reprezintă o valoare calculată să permită vizualizarea formulei de calcul utilizate, integrand informatiile de calcul din instrumentul de business intelligence cu regulile de transformare aplicate in instrumentul de ETL. Componenta de BI va rula toate procesele si operatiile pe datele din sistemul de tip depozit de date (*data-warehouse*) in care vor fi stocate datele structurate. Acest depozit va fi prezent la nivelul componentei de baze de date si va avea resurse de procesare dedicate. Informatiile vor fi prelucrate la nivelul depozitului de date prin intermediul instrumentului de tip ETL. Instrumentul de ETL trebuie sa fie compatibil si suportat de producatorul bazei de date ofertate. Platforma de integrare de date trebuie sa fie disponibila comercial (COTS – Commercial off the Shelf) si sa ofere

posibilitatea de a rula pe diverse platforme hardware precum si pe sistemele de operare majore existente pe piata (Windows, Linux si Unix). Pentru implementarea regulilor de transformare a datelor, platforma de integrare de date va trebui sa ofere posibilitatea crearii de interfete de date (activitati individuale care preiau datele din tabelele sursa, aplica regulile de transformare si scriu rezultatele in tabelele rezultat) si secvente de procesare (o inlantuire logica de interfete de date care se pot executa conditionat sau neconditionat, in paralel sau secvential).

Din punct de vedere al frecventei de executie a proceselor ETL, platforma de integrare de date trebuie sa ofere urmatoarele posibilitati:

- Executia proceselor de intregrare de date in batch, cu o frecventa variabila de tipul executie zilnica, saptamanala, lunara.
- Executia proceselor in micro-batch-uri, astfel incat sa permita divizarea informatiei care va fi procesata pe parcursul unei perioade in mai multe executii succesive
- Executia in timp aproape real, caz in care se va realiza o latenta scazuta intre momentul de timp la care informatiile au fost introduse/modificate in sistemul sursa si momentul propagarii informatiei in baza de date destinatie.

Componenta de integrare de date trebuie sa indeplineasca minimal urmatoarele cerinte:

- Trebuie sa permita acoperirea cel putin a operatiilor uzuale de prelucrare a datelor cum ar fi incarcarea datelor din diferite baze de date sau din fisiere text si XML, scrierea informatiilor in baze date sau fisiere text si XML, implementarea dimensiunilor cu schimbare lenta (Slowly Changing Dimensions), extragerea informatiilor din serviciile web expuse de alte aplicatii si publicarea datelor odata procesate sub forma de servicii web care sa poata fi accesate de aplicatiile externe;

- Avand in vedere volumul de date existent in sistemele sursa precum si ritmul de crestere al acestora, componenta de integrare de date trebuie sa ofere cel putin urmatoarele functionalitati:

    - încărcare masivă de date (Bulk Load) – sa ofere mecanisme specifice prin care sa permita extragerea, transformarea si incarcarea volumelor mari de date ;

    - încărcare incrementală - sa ofere mecanisme specifice prin care sa permita actualizarea incrementala a bazei de date destinatie, adaugand informatiile noi si modificand informatiile existente, acolo unde este cazul;

    - captarea datelor modificate in sistemele sursa – sa ofere mecanisme pentru detectarea informatiilor modificate in sistemele sursa si extragerea doar a acestor informatii, minimizand astfel volumul de date procesat si incarcarea suplimentara a sistemelor accesate.

- Componenta de integrare de date trebuie sa permita extinderea sistemului prin adaugarea de noi surse de date, suplimentar fata de tehnologiile mentionate anterior, prin folosirea unor drivere generice de tip ODBC/JDBC sau echivalent;
- Platforma de integrare de date trebuie sa permita extinderea functionalitatilor de baza prin definirea de componente (template-uri de operatii) reutilizabile specializate in executarea operatiilor specifice de ETL, pornind de la cele existente deja sau prin crearea de componente noi;
- Incarcarea datelor trebuie sa se faca conform unor programari stabilite anterior si sa poata sa fi declansta automat sau manual;
- Pentru a permite intretinerea facila a platformei de integrare de date si a interfetelor prin care sunt prelucrate datele, platforma va trebui sa permita definirea proceselor ETL in forma grafica, intuitiva, prin selectarea şi maparea vizuală a tabelelor implicate şi menţionarea componentelor de transfer necesare;
- Platforma de integrare de date va genera automat secventele de cod pentru procesarea datelor in mod optimizat pentru fiecare baza de date accesata, atat pentru citire cat si pentru scriere, folosind functiile native ale bazelor de date;
- Să permită vizualizarea procesului sau codului generat astfel încât să se detecteze eventualele erori;
- In definirea prceselor ETL, să se permită cel putin urmatoarele operatii:
    - definirea de filtre şi de restricţii asupra câmpurilor implicate;
    - agregarea datelor prin folosirea functiilor comune de agregare (sum, count, average);
    - divizarea, in cadrul unei interfete individuale, a fluxului de date de intrare in mai multe fluxuri de date paralele, pe fiecare flux de procesare putand avea reguli de transformare diferite, cu posibilitatea de a scrie informatia finala in mai multe tabele destinatie in cadrul aceleiasi interfete ETL;
    - definirea de operatii de lookup prin care fluxul de date principal sa poate fi extins cu informatii preluate din tabele de lookup sau sa permita derivarea datelor de intrare prin aplicarea regulilor de transformare definite in tabelele de lookup;
    - alocarea si rezolutia cheilor surogat folosite pentru asigurerea unicitatii inregistrarilor in baza de date;
    - suport pentru implementarea si executia regulilor de validare a datelor procesate;

- Platforma de integrare de date va trebui sa perita executia conditionala proceselor de intregare de date, conditiile fiind evaluate dinamic la momentul executiei;
- Să poată accesa şi integra date din baze de date diferite şi să ofere suport pentru accesarea datelor aflate în fişiere (.txt, .csv, .xml, etc);
- Platforma de integrare de date va trebui sa perita executia executia in loop a unui set de interfete de integrare de date, numarul de bucle efectuate putand fi definit static in momentul crearii secventei de procesare de date sau dinamic, calculat in momentul executiei;
- In cadrul maparilor de date, sa permita definirea de preocese multi-sursa, prin care datele sa fie citite din mai multe tabele, si multi-target, caz in care in aceeasi mapare informatiile vor fi scrise concomitent in mai multe tabele destinatie;
- Platforma de integrare de date trebuie sa ofere o interfata grafica pentru definirea proceselor de integrare de date disponibila cel putin pentru sistemele de operare Windows si Linux;
- Sa permita, in cadrul unei mapari, ramificarea fluxului de date in functie de un criteriu definit, astfel incat sa poata implementa reguli de transformare diferite pentru ramurile desprinse;
- Sa permita definirea de mapari reutilizabile astfel incat sa regulile de procesare inglobate intr-o astfel de mapare sa poate fi reutilizate, ori de cate ori este nevoie, in cadrul altor mapari;
- Sa ofere posibilitatea de a depana o interfata prin executia pas cu pas a transformarilor, prin care sa se poata vizualiza inclusiv operatiile SQL generate inainte de executia acestora, permitand astfel evaluarea impactului si verificarea corectitudinii rezultatelor;
- Platforma de integrare de integrare de date va trebui sa permita definirea mediilor de lucru utilizate (productie, dezvoltare, testare) si executia proceselor ETL pe unul din mediile definite prin specificarea contextului de lucru;
- Platforma de integrare de date va expune un set de servicii web prin care alte aplicatii vor putea interactiona cu platforma de integrare de date in scopul interogarii starii unui proces ETL, pornirea sau oprirea unui proces ETL;
- Platforma de integrare de date va trebui sa ofere functionalitati de gestiunea calitatii datelor, furnizand cel putin mecanisme prin se pot defini reguli de validare si managementul automat al erorilor (crearea tabelelor de erori, directionarea datelor in tabelelor de erori cu specificarea motivelor de eroare, includerea datelor in setul de date de intrare la o noua executie a interfetei ETL);

- Sa dispuna de o consola web de administrare astfel incat functiile de operare si mentenanta sa poata fi accesate printr-o pagina web, fara a solicita instalarea de programe sau aplicatii pe statiile client;
- Sa permita notificarea utilizatorilor la aparitia erorilor de procesare si, in functie de nivelul de severitate sa poata opri complet sau partial secventa de procesare de date;
- Sa permita executia paralela a interfetelor ETL in cadrul aceleiasi secvente de transformare de date;
- Sa permita sincronizarea intre diferitele secvente de procesare de date intre care exista o interdependenta logica, cel putin prin fisiere de sincronizare si sa permita definirea unui time-out pentru fisierul de sincronizare asteptat, la expirarea timpului putandu-se opta pentru executia activitatilor urmatoare sau pentru oprirea completa a secventei ETL;
- In cazul in care o secventa de transformare de date se termina cu eroare, platforma de integrare de date va trebui sa permita reluarea doar a pasilor/interfetelor ETL care au generat eroarea sau nu au fost inca executati, evitandu-se astfel prelucarea multipla a datelor prin executia repetata a interfetelor ETL;
- Pentru o utilizare eficienta si o mentenanta facila, mediul de lucru să nu necesite cunoştinţe avansate de programare, definirea regulilor de transformare facandu-se in mod grafic;
- Pentru a permite o implementare incrementala a sistemului si extinderea ulterioata a acestuia, platforma de integrare de date va trebui să permită păstrarea istoricului diverselor versiuni ale mapărilor de date, facand posibila revenirea la o versiune anterioara sau comparatia intre doua versuni ale aceleaisi interfete cu evidentierea modificarilor;
- În cadrul interfetelor trebuie să permită utilizarea unor funcţii native ale bazei de date accesate;
- Pentru funcţionare, modulul de integrare al datelor nu trebuie să necesite un server adiţional faţă de serverele sursă de unde se extrag datele sau serverele destinaţie unde vor fi încărcate datele, regulile de tranformarea datelor executandu-se in interiorul bazelor de date accesate;

Componenta de BI va oferi functionalitati de caching care sa permita stocarea rezultatelor interogarilor intr-o zona de cache urmand ca interogarile ulterioare ale utilizatorilor care refera acelasi set de date sau un subset al acestuia sa fie servite din zona de cache, fara a mai accesa baza de date sursa. Pentru gestionarea memoriei cache, sistemul va trebui sa ofere posibilitatea de a reactualiza, manual sau automat memoria cache astfel incat sa reflecte modificarile survenite in bazele de date accesate. Componenta de raportare trebuie sa ofere

suport pentru functionarea si in cazul in care unul din servere este nefunctional (de exemplu mentenanta sau defect), fara ca utilizatorul sa fie deconectat de la sistem (continuarea activității trebuie sa fie transparentă pentru utilizatori). Componenta de Business Inteligence va fi dimensionata pentru un numar de 45 de utilizatori simultan iar componenta va rula in arhitectura activ-pasiv

Cerinţe pentru componenta de raportare cu format fix din Modulul Central

Această componentă trebuie sa asigure posibilitatea de generare de rapoarte cu continut fix pentru utilizatorii care nu au, prin natura activităţii pe care o desfăşoară, necesități de modificare de rapoarte sau creare de rapoarte noi sau în cazul rapoartelor de natura oficială, cu formă fixă, destinate disponibilizării în masă sau imprimării şi distribuirii în formă fizică.

Componenta de analiza si raportare cu format fix trebuie sa fie un instrument de business intelligence care sa ofere posibilitatea de a rula pe diverse platforme hardware precum si pe sistemele de operare majore existente pe piata (Windows, Linux, Unix).

Componenta de raportare cu format fix va permite reprezentarea informatiei atat in format tabular cat si in format grafic. Selectia datelor ce urmeaza a fi afisate in aceste rapoarte se va face dinamic, utilizatorul putand preciza la momentul de executiei criteriile de filtrare a informatiei.

Din punct de vedere al surselor de date, componenta de raportare cu format fix trebuie sa permita accesarea informatiei din surse de date de tehnologii diferite cum ar fi:

- baze de date relationale: SQL Server, Oracle, DB2.
- baze de date multidimensionale: SQL Analysis Services, Oracle OLAP.
- fisiere: XML, CSV/Tab, Excel, MDB.
- Servicii Web.

Cerinţe pentru componenta platforma pentru rularea aplicatiilor – server de aplicatie

Componenta software a serverelor de aplicatii oferă suport pentru asigurarea infrastructurii software necesara executiei aplicatiilor moderne bazate pe standarde deschise. Serverul de aplicatii trebuie sa asigure un set de servicii standard pe care toate aplicatiile dezvoltate si instalate sa il poata accesa si utiliza:

- servicii de clusterizare pentru o scalabilitate si disponibilitate ridicata;
- servicii de balansare si dirijare a incarcarii;
- servicii de securitate pentru protejarea resurselor gazduite;
- servicii de definire si context de executie pentru resursele de aplicatie: conexiuni catre baze de date relationale, cozi de mesaje;
- servicii de manipulare a datelor in format XML;
- servicii de management al tranzactiilor la nivelul aplicatiilor.

Server-ul de aplicaţie trebuie să fie o platformă robustă şi agilă ce oferă suport pentru rularea de aplicaţii J2EE, simplificarea dezvoltării, performanţă ridicată şi administrare inteligentă. De asemenea, trebuie să ofere suport pentru tehnologiile standard recente şi cadre de dezvoltare menite să simplifice modelul de programare. Solutia propusa trebuie să respecte standardul Java EE 5, să ofere suport pentru EJB 3.0, JPA (Java Persistence API) şi JDK 6.0, si sa permită simplificare interoperabilităţii prin suport pentru servicii web incluzând JAX-WS, SOAP 1.2, MTOM, XOP, WS-ReliableMessaging, WS-Trust, WS-SecureConversation, WS-Policy, şi Kerberos Token Profile.

Server-ul de aplicaţie trebuie să ofere suport WEB 2.0, să ofere suport pentru Session Initiation Protocol (SIP), trebuie să permită comunicare prin mesaje asincrone utilizând un provider integrat de Java Message Service (JMS), care respectă standardul JMS 1.1, trebuie să ofere suport pentru Java EE Connector Architecture (JCA) 1.5 pentru conectivitatea intre serverele de aplicaţii şi diferite Enterprise Information Systems (EIS).

De asemenea, server -ul de aplicaţie trebuie să ofere suport pentru Java Transaction API (JTA) 1.1 pentru gestionarea tranzacţiilor, trebuie să ofere acces autentifict şi autorizat pentru a securiza funcţii administrative şi aplicaţii, trebuie să ofere suport pentru LDAP registry, Custom registry, file-based registry, sau federate registry.

Platforma trebuie ofere suport pentru Java Authorization Contract for Containers (JACC) 1.1, care să ofere securizarea resurselor gestionate de către server-ul de aplicaţie, sa suporte Java Standards Layering si EJB Bundles cat si publicarea serviciilor OSGi, trebuie să ofere suport pentru dezvoltare bazată pe modele cum ar fi Spring, să permită instalarea într-un singur pas, să ofere integrare nativă cu o platforma de dezvoltare compatibilă. Din punct de vedere functional, server-ul de aplicaţie trebuie să permită expunerea securizată a serviciilor intranet către utilizatorii de internet, să permită utilizare securizată a serviciilor web externe de către

sistemele localizate în intranet, să permită structuri de cluster, să poate funcţiona ca un server web, care să direcţioneze cereri din browser către aplicaţiile ce rulează pe server-ul de aplicaţii, să permită distribuţie inteligentă a sarcinii de lucru în cluster, trebuie să poată realiza balansarea traficului ce intră in sistem si să ofere disponiblitate ridicată (high availability) şi rezervă (backup) în interiorul clusterului. Solutia propusa trebuie să ofere disponibilitate ridicată a tranzacţiilor şi backup pentru tranzactii prin replicarea informatiilor referitoare la tranzactiile in lucru pe toate nodurile active si să ofere mecanisme de reglare a performanţei de la nivelul serverelor pana la nivelul cel mai detaliat al aplicaţiilor şi al componentelor utilizate de acestea.

Acces universal de date şi persistenţă:

- Server-ul de aplicaţie trebuie să suporte standardul Java DataBase Connectivity (JDBC) API 4.0 pentru conectare la orice tip de bază de date
- Server-ul de aplicaţie trebuie să suporte Java Persistence API (JPA) pentru a asigura persistenţa şi reutilizarea obiectelor
- Server-ul de aplicatie trebuie sa suporte standardul SCA
- Server-ul de aplicaţie trebuie să suporte standardul Service Data Objects (SDO), pentru a accesa şi manipula in mod uniform date din sisteme heterogene, sub forma de obiecte de colecţii de structuri de tip arbore sau graph-uri,
- Server-ul de aplicaţie trebuie să suporte Portlets conform standardului Java Specification Requests (JSR) 286 (Portlet 2.0)
- Server-ul de aplicaţie trebuie să suporte aplicaţii Session Initiation Protocol (SIP) conform standardului JSR 116
- Server-ul de aplicaţie trebuie să suporte standardele Java Servlet 2.5 (JSR 154) şi JavaServer™ Pages (JSR 245)

Securitatea aplicaţiilor:

- Server-ul de aplicaţie trebuie să ofere o granularitate ridicata
- Server-ul de aplicaţie trebuie să ofere flexibilitate de definire şi control al utilizatorilor
- Server-ul de aplicaţie trebuie să ofere capabilităţi de auditare pentru a asigura respectarea regulilor şi standardele existente în anumite domenii de activităţi
- Server-ul de aplicaţie trebuie să permită implementări de tip secure proxy pentru a putea configura mai uşor serverul de aplicaţii când acesta este utilizat în DMZ
- Server-ul de aplicaţie trebuie să ofere funcţionalităţi Single Sign-On pentru interoperabilitate imbunătăţită intre diferite aplicaţii şi medii

- Server-ul de aplicaţie trebuie să permită auditarea informaţiilor de securitate a acţiunilor administrative, cum ar fi modificări de configurări de securitate, gestionare de chei şi certificate , modificări de politici de control al accesului.
- Server-ul de aplicaţie trebuie să ofere adminsitrare a securităţii imbunătăţită la nivelul consolei de adminsitrare, cu acces pe bază de drepturile in funcţie de roluri la nivelul cell, node, cluster sau aplicaţie

Administrare inteligentă:

- Server-ul de aplicaţie trebuie să permită anticiparea şi ajustarea parametrilor critici pentru platformă.
- Server-ul de aplicaţie trebuie să ofere infrastructură simplificată şi flexibilă pentru controlul eficienţei mediilor de rulare.
- Server-ul de aplicaţie trebuie să ofere administrarea de la distanţă a mai multor servere de aplicaţii
- Server-ul de aplicaţie trebuie să ofere controlul mediilor complexe cu topologii complexe
- Server-ul de aplicaţie trebuie să ofere activităţi administrative simplificate pentru aplicaţiile de tip “multi-component”
- Server-ul de aplicaţie trebuie să ofere o consolă de administrare web-based pentru gestionarea centralizată a tuturor componentelor din topologii ce includ mai multe servere de aplicaţii şi/sau web
- Server-ul de aplicaţie trebuie să permită gestionare centralizată a transferului de informaţii intre medii cum ar fi: deploy, start, stop aplicaţii şi distribuirea fişierelor in topologii ce includ mai multe servere de aplicaţii şi/sau web
- Server-ul de aplicaţie trebuie să ofere capabilitatea de a realiza instalări centralizate către diferite medii remote.
- Server-ul de aplicaţie trebuie să ofere posibilitatea de a grupa şi gestiona mai multe artefacte Java EE sub o singură definiţie de aplicaţie
- Server-ul de aplicaţie trebuie să ofere capabilităţi de a configure cu uşurinţă securitatea şi conectivitatea la baza de date şi un client stand-alone pentru administrarea eficientă a mediului de deployment.
- Server-ul de aplicaţie trebuie să ofere scripturi avansate de administrare, care accelerază automatizarea implementărilor

- Server-ul de aplicaţie trebuie să ofere capabilităţi consolidate de administrare pentru dispozitive externe care pot fi integrate şi gestionate in mediul de servere de aplicaţii
- Server-ul de aplicaţie trebuie să ofere suport avansat de gestionare a resurselor ce constă in a alege doar funcţionalităţile necesare pentru memorie şi spaţiu, in mod dinamic, in funcţie de necesităţile aplicaţiilor care rulează, pornind doar componenetele necesare aplicaţiilor ce rulează la un moment dat pe server şi astfel reducând timpul de pornire şi spaţiul alocat server-ului.

  Server-ul de aplicaţie trebuie să ofere suport pentru noile standarde de Servicii Web cum ar fi:
  - Web Services Interoperability Organization (WS-I) Basic Profile 1.2 şi 2.0
  - WS-I Reliable Secure Profile
  - Java API for XML Web Services (JAX-WS)
  - SOAP 1.2
  - SOAP Message Transmission Optimization Mechanism (MTOM)
  - XML-binary Optimized Packaging (XOP)
  - WS-ReliableMessaging
  - WS-Trust
  - WS-SecureConversation
  - WS-Policy

Platforma de aplicatii propusa trebuie sa respecte urmatoarele functionalitati:

- Producatorul să ofere tool-uri de migrare aplicaţii de la alte versiuni ale soluţiei oferite sau de la alte servere de aplicaţii cum ar fi Oracle WebLogic, Jboss sau Tomcat.
- Upgradarea aplicatiilor sa fie realizabila fara intreruperea lucrul utilizatorilor
- Sa existe posibilitatea rularii mai multor versiuni ale unei aplicatii – rutarea automata a utilizatorilor catre o anumita aplicatie
- Sa existe posibilitatea activarii pentru o perioada extinsa de timp a multiple editii
- Serverul de aplicatii sa ofere support pentru politici de rollout pentru comutarea intre o editie a unei aplicatii si alta editie fara pierderi
- Sa existe posibilitatea upgrade-ului sistemlui de operare sau a serverului de aplicatie fara a exista down time
- Administrarea editiilor sa fie controlata din consola de administrare
- Serverul de aplicatii sa ofere suport pentru scripturi

- Serverul de aplicatii sa ofere posibilitatea detectarii si tratarii anumitor probleme ale unei aplicatii fara a fi nevoide de interventie umana
- Serverul de aplicatii sa ofere posibilitatea tratarii problemelor unei aplicatii intr-un mod inteligent permitant continuarea utilizarii aplicatiei – fiecare politica de tratare a erorilor sa constea in conditii, una sau mai multe actiuni si un set de procese
- Serverul de aplicatii sa include politici de tratare a exceptiilor pentru un set comun de probleme
- Serverul de aplicatii sa include conditii si actiuni customizabile pentru tratarea problemelor unei aplicatii
- Serverul de aplicatii sa ofere posibilitatea de definire a politicilor de functionare optima pentru mentinerea in parametrii a serverului – actiunile de corectare sa fie executate automat sau sa necesite aprobare:
    - Sa notifice administratorul
    - Sa captureze informatii pentru diagnosticarea problemei – heap dump sau java core
    - Sa restarteze serverul intr-un mod in care sa evite intreruperile
- Serverul de aplicatii sa permita administratorilor sa specifice importanta relative a aplicatiilor si optional, sa stabileasca un timp de raspuns. Serverul de aplicatii sa gestioneze aplicatiile in conformitate cu aceasta politica:
    - Politicile associate serviciilor sa fie folosite pentru definirea SLA-urilor aferente aplicatiilor
    - Sa permita fluxurilor de lucru sa fie clasifilate, prioritizate si rutate in mod inteligent
    - Sa permita monitorizarea performantelor aplicatiilor
    - Alocarea dinamica a resurselor pt a staisface SLA-urile stabilite
- Serverul de aplicatii sa permita crearea clusterelor dinamice – definirea parametrilor nodurilor unui cluster sa fie realizate dynamic sau sterse pe baza politicii de apartenenta la nod (serverele sunt create/sterse daca un nod este adaugat/eliminate dintr-un grup de noduri)
- Serverul de aplicatii sa permita pornirea sau oprirea serverelor dintr-un cluster pe baza politicilor aplicatiei curente
- Serverul de aplicatii sa ofere posibilitatea de prioritizare a cererilor si a le ruta in functie de reguli predefinite de catre administratorul serverului de aplicatie.

- Serverul de aplicatii sa ofere posibilitatea de rutare random sau round robin
- Serverul de aplicatii sa detecteze in mod dinamic cand informatiile de rutare sunt schimbate
- Serverul de aplicatii sa ofere posibilitatea de creare a clusterilor dinamice pentru scalarea incarcarii bazandu-se pe diferite politici
- Serverul de aplicatii sa ofere o interfata de monitorizare a performantei serverului
- Serverul de aplicatii sa ofere unelte de analize pentru log-uri
- Serverul de aplicatii sa ofere unelte de analiza a dump-urilor
- Serverul de aplicatii sa ofere unelte de vizualizare a garbage collector si a memoriei

Se vor licenta minim de 32 de core-uri pentru server de aplicatie. Componenta va rula in arhitectura activ-activ.

Componenta de rulare a aplicatiilor (server de aplicatii) va beneficia de o resursa de tip baze de date pentru mecanismele proprii de functionare pentru a nu avea un impact de performamta asupra subsistemul central de gestiune a bazelor de date. Aceasta subcomponenta va respecta urmatoarele caracteristici:

Cerinţe generale

- Baza de date trebuie sa fie un sistem de gestiune a bazelor de date de tip relational.
- Baza de date trebuie sa poată rula pe sisteme de operare incluzând cel puţin Windows, Linux, AIX.
- Baza de date trebuie sa suporte comunicarea cu aplicaţiile client folosind protocolul de transport pe reţea TCP/IP.
- Baza de date trebuie sa fie compatibila cu standardul ANSI SQL.
- Baza de date trebuie sa aiba suport UTF-8 sau echivalent.
- Baza de date trebuie sa ofere suport pentru proceduri stocate si triggeri.
- Baza de date trebuie sa suporte mecanismul de connection pooling.
- Baza de date trebuie să suporte tranzactii autonome.
- Baza de date trebuie sa suporte aplicatii scrise in limbaje procedurale SQL caracteristice mai multor sisteme de gestiune a bazelor de date.
- Baza de date trebuie sa suporte nativ stocarea si gestiunea datelor de tip XML.
- Baza de date trebuie sa suporte aplicatii bazate pe .NET, OLE DB, ODBC, JDBC, SQLJ.
- Baza de date trebuie sa permita gestiunea datelor de tip spatial.
- Baza de date trebuie sa permita cautarea in date de tip text folosind instructiuni SQL.

- Baza de date trebuie sa includa un mecanism nativ care istorizeaza modificarile pe datele unei tabele fara a necesita dezvoltarea de triggeri sau rutine definite de utilizator, salvare periodica sau utilizarea functiei de audit.
- Baza de date trebuie sa ofere suport pentru NoSQL.
- Baza de date trebuie sa ofere suport pentru stocarea si interogarea documentelor JSON.
- Baza de date trebuie sa fie capabila a stoca mai mult de 10 terabytes.
- Baza de date trebuie sa fie capabila sa gestioneze mai mult de 64 gigabytes de memorie.

Securitatea si Auditul Datelor

- Baza de date trebuie sa permita restrictionarea accesului la obiectele bazei de date.
- Baza de date trebuie sa ofere o lista cu operatiile pe care un utilizator le poate executa.
- Baza de date trebuie sa ofere o facilitate care înregistrează informaţii în legătura cu operaţiunile de modificare, inserare, ştergere şi selectare a obiectelor interne bazei de date.
- Baza de date trebuie sa permita criptarea datelor stocate in baza de date.
- Baza de date trebuie sa permita protejarea tabelelor la nivel de rand si coloana pe baza de etichete de securitate care definesc criteriile de acces la date.
- Baza de date trebuie sa permita mascarea la nivel de coloana a datelor returnate utilizatorilor sau aplicatiilor, daca acestia nu sunt autorizati sa vizualizeze datele respective.
- Baza de date trebuie sa permita restrictionarea si filtrarea la nivel de rand a datelor returnate utilizatorilor, in functie de cine le acceseaza.
- Baza de date trebuie sa permita implementarea de politici de securitate prin care aceeasi interogare sa returneze rezultate diferite, in functie de rolul utilizatorului care lanseaza interogarea.

Salvarea şi recuperarea datelor

- Baza de date trebuie sa ofere o facilitate pentru salvarea-restaurarea întregii baze de date.
- Baza de date trebuie sa ofere o facilitate pentru înregistrarea tuturor modificărilor bazei de date in vederea recuperarii (înregistrarea tranzacţiilor).
- Baza de date trebuie sa ofere o facilitate pentru recuperarea întregii baze de date corespunzător unui moment de timp specificat de administrator.
- Baza de date trebuie sa ofere posibilitatea de a realiza salvări pentru unul sau mai multe spatii alocate tabelelor aşa cum este specificat de către administratorul bazei de date.
- Baza de date trebuie sa permita salvarea automata a bazei de date in regim online.

- Baza de date trebuie sa ofere posibilitatea de a recupera una sau mai multe tabele corespunzător unui moment de timp specificat de administrator.
- Baza de date trebuie sa ofere o facilitate care afiseaza informatii despre salvarile bazei de date.
- Baza de date trebuie sa salveze jurnalele de tranzactii separat de salvarea intregii baze de date.
- Baza de date trebuie sa permita compresia salvarilor efectuate pentru baza de date in vederea reducerii spatiului ocupat de acestea.

Integritatea datelor

- Baza de date trebuie să identifice şi să rezolve situaţiile de blocaj la acces concurent ("dead-locks").
- Baza de date trebuie sa permita impunerea constrangerilor de tip cheie primara.
- Baza de date trebuie sa permita definirea de coloane care nu accepta valori nule.
- Baza de date trebuie sa permita impunerea de constrangeri care interzic valori duplicate in coloane care nu participa la o cheie primara.
- Baza de date trebuie sa permita impunerea de constrangeri asupra tipurilor si valorilor datelor.
- Baza de date trebuie sa permita impunerea de constrangeri de integritate referentiala intre doua tabele.
- Baza de date trebuie sa permita blocarea tabelelor la nivel de rand.

Baza de date trebuie sa suporte un mecanism care asigura ca tranzactiile distribuite intre mai multe noduri sunt fie comise, fie anulate pe toate nodurile in cauza.

Performanta şi scalabilitatea bazei de date

- Baza de date trebuie sa suporte crearea de mai multe baze de date care sa aparţină unei singure instanţe.
- Baza de date trebuie sa suporte instanţe multiple, izolate şi complet funcţionale ale bazei de date pe un singur nod SMP.
- Baza de date trebuie sa permită instalarea unei singure baze de date pe mai multe noduri in mod activ – activ (arhitectura de tip cluster) pentru a asigura toleranta la defecte hardware sau nefunctionare planificata.
- Baza de date trebuie sa ofere disponibilitate continua în cazul aparitiei unei defectiuni la unul din serverele cluster-ului de baza de date.

- Baza de date trebuie sa permită stoparea temporara a unui nod din clusterul de baza de date pentru mentenanta, suport sau upgrade, sistemul ramanand disponibil în tot acest timp.
- Baza de date trebuie sa permită reconectarea automata la nodul sau nodurile ramase disponibile dupa aparitia unei defectiuni.
- Baza de date trebuie sa ofere software de cluster integrat astfel incat să nu necesite achizitionare de soft de clustering aditional de la producatorul sistemului de operare.
- Baza de date trebuie să dispuna de facilitati de replicare a datelor între doua baze de date prin diverse metode incluzand SQL si cozi de mesaje.
- Baza de date trebuie sa permita accesul federat la datele din sisteme relationale de baze de date heterogene.
- Baza de date trebuie sa permita mentinerea a doua baze standby pentru aceeasi baza de date.
- Baza de date trebuie sa permita captura modificarilor de date din logurile bazei de date si replicarea acestora in timp real in alta baza de date din alt sistem de gestiune a bazelor de date.
- Baza de date trebuie sa ofere abilitatea sa partiţioneze tabele si indecsi.
- Baza de date trebuie sa permita reorganizarea, mutarea si redefinirea de tabele fara blocarea activitatii.
- Baza de date trebuie sa permita compresia datelor din tabele.
- Baza de date trebuie sa permita interogarea tabelelor cu compresie fara a decompresa in prealabil datele in memorie.
- Baza de date trebuie sa suporte tabele organizate pe disc in ordinea data de o coloana, mai multe coloane sau grupuri de coloane.
- Baza de date trebuie sa permita optiuni de stocare pe rand si pe coloana in aceeasi baza de date.
- Baza de date trebuie sa permita stocarea in mod automat a datelor pe discuri a caror performanta e corelata cu frecventa de acces la date.
- Baza de date trebuie să-si poata ajusta automat memoria, atat in plus cat si in minus, in functie de cerintele de executie a sarcinilor curente din baza de date.
- Baza de date trebuie să se ajusteze automat in cazul suplimentarii spaţiului de stocare.
- Baza de date trebuie sa permita executia paralelă a operatiilor SQL.

- Baza de date trebuie sa permită prin parametrizare folosirea unui sistem care sa trimită datele de pe disc în memorie înainte ca o cerere de tip SQL sa aibă nevoie de acele date.
- Pentru optimizarea performantei lucrului cu anumite tabele, baza de date trebuie sa permita definirea de zone in memorie pentru citirea si manipularea datelor din tabele special indicate de administrator.
- Baza de date trebuie sa aibă un optimizator bazat pe cost pentru a optimiza interogările.
- Baza de date trebuie sa permita controlul automat al incarcarii bazei de date prin prioritizare si limitare de resurse de executie in functie de tipul de incarcare.

Administrarea/dezvoltarea/ raportarea bazei de date

- Baza de date trebuie sa includa un instrument grafic de administrare si dezvoltare a bazei de date.
- Baza de date trebuie sa includa un instrument grafic de analiza si optimizare a performantelor din baza de date.
- Baza de date trebuie sa includa un instrument de urmarire a modificarilor de configurare bazei de date cu generarea de alerte atunci cand apar deviatii de la configurarea recomandata de cele mai bune practici.
- Baza de date trebuie sa dispuna de un instrument de tip ELT (extract – load – transform).
- Baza de date trebuie sa dispuna de un instrument pentru modelarea datelor.
- Baza de date trebuie sa includa si capabilitati pentru a efectua data mining.Baza de date trebuie sa includa un instrument de business intelligence cu ajutorul caruia sa se poata construi rapoarte asupra datelor stocate in baza de date in formate variate, cum ar fi tabele, grafice de tip hărţi, pie, chart sau combinaţii ale acestora.

  Subcomponenta de gestiune a datelor aferente platformei aplicative va fi dimensionata pentru functionare in inalta disponibilitate si va fi dimensionata pentru a asigura functionare si performantele corespunzator cu solutia propusa de catre ofertant.

In vederea optimizarii resurselor, distributiei uniforme a acestora, dar si pentru o inalta disponibilitate, fiabilitate si scalabilitate, sistemul poate fi prevazut cu o platforma de virtualizare a resurselor hardware, a carei licentiere sa fie in concordanta cu solutia propusa, cu cerintele de performanta ale sistemului si cu recomandarile in domeniu. Platforma de virtualizare trebuie sa indeplineasca urmatoarele caracteristici minimale:

- asigurarea unei componente informatice care să permită rularea unor aplicaţii de gestiune a datelor din sistem într-un mediu virtual, care să ofere disponibilitate, securitate, scalabilitate şi portabilitate.

- solutia de virtualizare va permite interconectarea maşinilor virtuale şi aplicaţiilor găzduite/dezvoltate pe acestea, cu echipamentele hardware şi aplicaţiile software livrate in cadrul proiectului.
- asigurarea unui înalt nivel de performanţă, disponibilitate şi securitate pentru toate aplicaţiile şi informaţiile gestionate în mediul virtual, astfel încât să fie îndeplinite condiţiile necesare continuării acţiunilor pe linia prelucrării, actualizării şi schimbului de date.
- Să permită gestionarea (administrarea, furnizarea de informaţii.) optimă a informaţiilor aferente sistemului;
- Să asigure disponibilitate permanentă a serviciilor şi datelor;
- Să asigure o eficienţă ridicată de utilizare a resurselor şi să asigure un management performant al utilizării acestor resurse.
- Sa dispuna de mecanisme native de tip high availability, migrare online a masinilor virtuale si management centralizat;

Cerinţe pentru componenta de balansare a cererilor către serverele de aplicaţie

Serverele web trebuie sa permita prezentarea continutului sistemului catre utilizatori si transferul de date dinspre client spre sistem (prin intermediul browserelor web). In acelasi timp, serverele web trebuie sa asigure primul nivel de securitate software din punct de vedere al accesului – configurare in mod reverse proxy, suport pentru SSL si autentificare de baza (in conjunctie cu serverele de control acces ale solutiei).

Comunicatiile cu exteriorul retelei trebuie sa se realizeze atat criptat cat si in clar, in functie de tipul informatiei; din punct de vedere al suport SSL v3. Serverele web trebuie sa permita integrarea cu solutii de accelerare hardware a criptarii/decriptarii si sa dispuna de functionalitati de rescriere a adreselor URL.

Din motive de securitate si usurinta in utilizare, serverul web trebuie sa permita implementarea unui mecanism de tip SSO in conjunctie cu solutia de control acces si in acelasi timp sa ofere suport pentru autentificare cu certificate digitale prin integrare cu solutia de control acces si cu o infrastructura PKI.

In plus, serverele web trebuie sa ofere suport pentru IPv4 si IPv6 astfel incat sa permita utilizarea in contextul noilor scheme de adresare Internet.

Serverele web trebuie sa poata rula pe toate distributiile majore de sisteme de operare prezente pe piață (Windows, Linux, Unix). Se solicita cel putin 24 buc licente pentru aceasta componenta.

3.1.10.2. Subsistemul serviciilor de integrare şi transformare

Acest subsistem asigură orchestrarea serviciilor aplicative din cadrul sistemului, precum şi interoperabilitatea cu alte sisteme şi este compus din următoarele categorii se servicii:

- **Servicii tip flux de lucru (*proces*)** - asigură capacităţile de control necesare organizării fluxului de lucru şi a interacţiunii între aplicaţii. Fluxurile de lucru sunt reprezentate ca procese de business sau sub forma unor maşini de stare.
- **Servicii de comunicare între aplicaţii** - permit schimbul de informaţii între module aplicative interne ale sistemului şi module externe ale sistemelor din PIAS.
- **Servicii de transformare de date** - permit transferul de date externe, în diferite formate, astfel:
    - Trebuie transformate şi armonizate datele de la diferite surse, tipice pentru sursele curente de date din sistemul medical pentru a putea fi compatibile cu modelul de date intern;
    - Se va implementa un mecanism care să permită extinderea în viitor a formatului datelor externe, conform standardelor internaţionale;
    - Se vor defini şi aplica reguli de calitate a datelor;
    - Se va defini şi implementa un mecanism de tratare a erorilor.
- **Servicii de tip registru de servicii** - permite publicarea şi accesarea automată a celorlalte servicii operaţionale oferite pe magistrala de servicii, dacă este cazul.

Subsistemul va include o componenta specializata de tip integrare si transformare, capabila sa ruleze in mod disponibilitate inalta activ-activ.

Cerinţe pentru componenta de tip integrare si transformare

Aceasta componenta va avea suport pentru interoperabilitatea diverselor module ale solutiei conform cu principiile si conceptele arhitecturilor "Service Oriented Architecture" si "Event Driven Architecture": WS-I Basic Profile, WSDL, WS-*, XML, SOAP. Componenta va pune la dispozitie posibilitatea definirii de fluxuri de proces bazate pe standardul OASIS

BPEL iar aceste servicii vor fi expuse printr-un modul de virtualizare servicii de tip "Service Bus".

Componenta de integrare si transformare mesaje trebuie să indeplinească următoarele specificatii:

- Să ofere suport complet pentru dezvoltarea, testarea, execuţia, monitorizarea, optimizarea şi administrarea proceselor de afaceri
- Să ofere suport pentru soluţii moderne şi deschise de integrare conform principiilor şi conceptelor arhitecturilor Service Oriented Architecture (SOA) şi Event Driven Architecture (EDA)
- Să fie bazată pe standardele deschide de interoperabilitate a aplicaţiilor WS-I Basic Profile, WSDL, WS-*, XML, SOAP, UDDI
- Să permită modelarea declarativă a proceselor de afaceri utilizând standardul OASIS BPEL cu ajutorul mediului de dezvoltare integrat al sistemului
- Platforma de integrare şi mediul său de dezvoltare să asigure instalarea proceselor modelate pe serverele de execuţie şi să susţină scrierea de suite de teste asociate fluxurilor de afaceri
- Permite comunicaţii sincrone şi asincrone inter-aplicaţii
- Oferă mecanisme transparente de persistenţă a stării proceselor şi informaţiilor de audit într-o bază de date relaţională
- Permite folosirea canalelor de notificare moderne (email, voce, SMS, fax) pentru informarea utilizatorilor despre evenimentele semnificative apărute în aplicaţii
- Din punct de vedere al interacţiunii umane cu procesele de integrare, să ofere şabloane predefinite de dirijare a activităţilor către utilizatorii cu roluri specifice la nivelul aplicaţiilor precum şi interfeţe grafice de lucru cu fluxurile automatizate
- Se integrează cu soluţii de tip Service Registry bazate pe standardul deschis Universal Description Discovery and Integration (UDDI) v3
- Permite auditarea şi înregistrarea proceselor şi serviciilor executate sau în curs de execuţie precum şi a datelor transportate
- În scopul detectării problemelor de performanţă a platformei de fluxuri, infrastructura de execuţie va permite colectarea permanentă de statistici de execuţie – timpi de execuţie, frecvenţă de apariţie evenimente, stări fluxuri – pentru procesele instalate
- Sistemul va include un modul dedicat de stocare şi evaluare a regulilor de business, externe proceselor modelate, pe care personalul non-tehnic le va putea accesa şi modifica on-line prin intermediul unei console web

- Suportă transformări şi manipulări de date complexe pentru implementarea logicii proceselor de business
- Tipurile de mesajele transportate suportate de soluţia de tip Service Bus vor fi: XML, text, ne-tipat, binar, attachment
- Sistemul va include capabilităţi extinse de transformare a mesajelor XML utilizând standarde deschise W3C Extensible Stylesheet Language (XSL) , xQuery, xPath;
- Oferă soluţii de conectare predefinite la principalele tipuri de tehnologii: baze de date relaţionale, cozi de mesaje (Java JMS, Oracle Advanced Queuing (AQ), IBM MQ, MS MQ), sisteme de fişiere, servere FTP
- Suportă soluţii de conectare la principalele sisteme de aplicaţii
- Oferă un cadru de dezvoltare pentru noi soluţii de conectare la sisteme externe bazat pe standarde deschise
- Oferă servicii de transport cu suport pentru persistenţa datelor
- Oferă servicii de transport cu suport pentru garantarea livrării datelor

- Trebuie să ofere nativ următoarele capabilităţi de dirijare mesaje în interiorul

  infrastructurii:

  - conţinut mesaj;

  - table dinamice de dirijare;
  - prioritate;

  - performanţa serviciilor;

  - versiune serviciu;
  - service level agreement;

- conţinut header SOAP;

- originea mesajului;

- user ID şi rol;

- retransmitere în caz de failover;

- Soluţia va implementa următoarele modele de servicii: sincron cerere/răspuns, asincron one-to-one, asincron one-to-many, asincron cerere/răspuns (synchronous-to-asynchronous bridging)
- Service Bus-ul va oferi posibilitatea definirii, la momentul execuţiei, a adreselor de destinaţie a mesajelor (Dynamic Routing), eventual prin interogarea unei soluţii de tip Service Registry
- Soluţia va implementa mecanisme de control şi garantare a livrării mesajelor conform următoarelor tipuri de politici: Exactly-Once, At-Least-Once, Best-Effort
- Să includă o soluţie de tipul monitorizare a activităţii de afaceri - Business Activity Monitoring (BAM)
- Să permită monitorizarea în timp real a indicatorilor de tipul Key Performance Indicators (KPI)
- Să permită monitorizarea în timp real a SLA-urilor de business sau operaţionale
- Să colecteze date utilizând diferite canale de date precum: conexiuni la baze de date relaţionale, cozi de mesaje, publicare directă în cache-ul de monitorizare în timp real
- Definirea şi prezentarea tablourilor de bord se va putea face de către personal non-tehnic utilizând un simplu browser web
- Soluţia va permite definirea de alerte şi planuri de acţiuni asociate nerespectării KPI-urilor şi SLA-urilor impuse sau altor evenimente ale sistemului
- Să ofere un modul centralizat de gestiune şi aplicare a politicilor de securitate peste portofoliul de fluxuri electronice instalat
- Să ofere servicii de securitate atât la nivel transport cât şi la nivel de aplicaţie

- Pentru asigurarea securităţii la nivel transport, soluţia va permite utilizarea protocolul Secure Socket Layer (SSL) şi a certificatelor compatibile X.509
- Să ofere servicii de securitate specifice lucrului cu serviciile web standard:
    - autentificarea accesului la servicii;
    - autorizarea accesului la servicii
- Soluţia să fie bazată pe standardele deschise de securitate a serviciilor web, precum WS-Security, Security Assertation Markup Language (SAML)
- Să ofere suport pentru standardele deschise de securitate privind mesajele în format XML:
    - XML Encryption pentru criptarea / decriptarea mesajelor XML în vederea asigurării confidenţialităţii mesajelor transportate;
    - XML Signature pentru semnarea / verificarea digitală a mesajelor XML în vederea asigurării integrităţii şi non-repudierii mesajelor transportate
- Soluţia va putea securiza totalitatea apelurilor către serviciile existente printr-un mod de funcţionare de tip poartă de acces (gateway) fără să fie necesară modificarea proceselor instalate
- Să ofere agenţi ce vor putea fi instalaţi pe staţiile client în vederea împachetării (adăugare informaţii securitate) apelului către serviciile web dorite
- Mecanisme de grupare a serverelor în clustere pentru toate componentele platformei de integrare atât în topologii de tip activ-activ cât şi activ-pasiv
- Stoparea temporară a unui nod din cluster pentru mentenanţă şi suport, sistemul în acest timp fiind disponibil pentru activităţi normale
- Mecanisme de balansare dinamică a încărcarii sistemului între resursele administrate în cadrul aceluiaşi cluster
- Mecanisme de scalare a sistemului pe orizontală (Scale Out) şi verticală (Scale Up)
- Să permită rularea platformei de integrare pe toate distribuţiile majore de sisteme de operare prezente pe piaţă: Windows, Linux şi Unix
- Suport pentru includerea sub-proceselor apelate dintr-un proces principal în tranzacţia fluxului iniţiator
- Implementarea unui mecanism de export al informaţiilor – variabile proces, activităţi, excepţii – din flux direct în baze de date relaţionale sau cozi de mesaje
- Facilităţi de activare a auditării numai în cazul terminării cu eroare a unui flux

- Specificarea şi modificarea fluxurilor de mesaje să se poată face atât utilizând mediul de dezvoltare integrat al sistemului cât şi un simplu browser web
- Managementul încarcării livrării mesajelor către serviciile destinaţie înregistrate la nivelul magistralei de interconectare folosind cozi de mesaje tampon care permit:

  - definirea concurenţei maxime admise de serviciul destinaţie;

  - definirea unei perioade de expirare pentru mesajele trimise;

  - definirea de priorităţi asociate mesajelor
- Actualizarea informaţiilor în panourile de bord se va face automat, în timp real, fără a fi necesar un "Refresh" manual din partea utilizatorului
- Se va permite apelul de fluxuri de business ca pas al unui plan de acţiuni
- Platforma de monitorizare va fi direct integrată cu infrastructura de fluxuri de procesare oferind posibilitatea publicării directe de informaţii relevante la nivel de proces către cache-ul sistemului de monitorizare
- Definirea politicilor de securitate se va face declarativ, utilizând un instrument de configurare web-based, pe baza unor primitive de activităţi de securitate pre-definite – exemplu: verificarea credenţialelor utilizator versus un director LDAP, auditarea accesului la un serviciu.

- Să ofere nativ următoarele capabilităţi de logging:

  - Stare endpoint serviciu;
  - erori;
  - apel serviciu;
  - timp de răspuns;

  - depaşirea SLA;

3.1.10.3. Subsistemul de gestiune a bazei de date

Subsistemul de gestiune a bazei de date trebuie să asigure necesarul de persistenţă operaţională pentru componentele aplicative ale sistemului, prin satisfacerea categorii de cerinţe descrise în continuare.

Sistemul de gestiune al bazelor de date relationale trebuie sa fie un sistem de gestiune a bazelor de date de tip relational si sa ofere posibilitatea de a rula pe diverse platforme hardware precum si pe sistemele de operare majore existente pe piata (Windows, Linux, Unix), oferind urmatoarele capabilitati:

- posibilitatea de a suspenda temporar operatii consumatoare de resurse (de exemplu incarcari masive de date), cu reluarea ulterioara a acestora in momentul cand sistemul permite, precum si posibilitatea de a implementa scheme de prioritate in modul de accesare a bazei de date in functie de tipul de utilizator inclusiv limitarea numarului de procesoare folosite de baza de date fara a fi necesara folosirea unei solutii de virtualizare;
- interogarea direct din baza de date a fisierelor text externe, fara a necesita in prealabil o operatiune de incarcare intr-o tabela din baza de date inclusiv posibilitatea de a rula anumite scripturi la momentul interogarii acestor fisiere din interiorul bazei de date;
- parametrii de memorie sa poata fi ajustati dinamic si automat de catre baza de date astfel incat zonele de memorie sa fie dimensionate in concordanta cu tipul de operatii ce se desfasoara la un moment dat iar pentru a face fata unui numar foarte mare de utilizatori, baza de date trebuie sa ofere un mecanism de connection pooling care sa optimizeze folosirea resurselor server-ului la operatiile de tip login/logout

Din punct de vedere al managementului datelor, baza de date trebuie sa ofere mecanisme avansate de ILM (Information Lifecycle Managment) care sa permita organizarea acestora in mod dinamic si transparent fata de aplicatii prin asigurarea compresiei datelor stocate in tabele si partitionarea tabelelor cat si a indecsilor dupa diverse criterii ajutand astfel la paralelizarea operatiilor de interogare a datelor.

Baza de date trebuie sa permita functionarea intr-o arhitectura de disponibilitate inalta de tip cluster activ-activ, o singura baza de date putand fi instalata pe mai multe noduri asigurandu-se toleranta la defecte hardware sau nefunctionare planificata, scalabilitatea si disponibilitatea crescuta a sistemului. Din punct de vedere al performantei bazei de date, fiecare nod al

cluster-ului bazei de date trebuie sa poata fi capabil sa acceseze date din memoria cache a celorlalte noduri reducand astfel la minim necesitatea de a citi blocurile de date direct de pe disc. Totodata arhitectura de tip cluster trebuie sa asigure si o balansare a incarcarii intre noduri la nivelul cererilor si executiilor pe baza de date aflata in cluster oferind o incarcare uniforma a acesteia iar din punctul de vedere al utilizatorilor trebuie sa ofere o disponibilitate de tip 24x7 in cazul aparitiei unei defectiuni hardware la unul din serverele cluster-ului de baza de date. Securitatea tranzactionala in cazul aparitiei unor erori hardware sau software in clusterul de baza de date trebuie sa fie tratata de mecanismele interne ale bazei de date iar in cazul unei defectiuni hardware si/sau software sa permita reconectarea automata la nodul sau nodurile ramase disponibile.

Pentru asigurarea performantei maxime si asigurarea unui raspuns proactiv la eventualele degradari ale acesteia, baza de date trebuie sa ofere :

- un repository in care se vor stoca permanent informatiile legate de modul de functionare al bazei de date, pe baza caruia administratorii sa poata realiza oricand o analiza curenta sau istorica a performantelor bazei de date prin rularea de rapoarte predefinite care sa contina indicatorii cheie de performanta (CPU Load, Sistem IO, Wait-uri, top sql, top sesiuni, consum resurse, interogari neoptimizate) ;
- mecanisme interne de monitorizare si diagnosticare continua, automatizand colectarea parametrilor de functionare ai bazei de date, precum si stocarea acestora pentru a putea furniza o imagine pe termen lung a modului de functionare al bazei de date.
- automatizarea procesului de optimizare al instrucţiunilor SQL, explorand in mod cuprinzator toate modalitatile de optimizare ale unei instructiuni SQL (inclusiv recomandarea de a fi creati indecsi sau partitii aditionale), oferind sugestii de implementatare pentru administratori si posibilitatea de implementare fara a fi necesara modificarea codului aplicatiei ;
- identificarea automata a interogarilor neoptimizata ce sunt executate in baza de date, analizarea acestora in vederea identificarii posibilitatilor de optimizare iar in functie de modalitatea de optimizare gasita fie sa implementeze automat optimizarile fie sa notifice administratorul cu privire la problemele identificate.

Solutia de baza de date trebuie sa ofere un utilitar grafic pentru modelarea relaţională şi dimensională a datelor precum si o unealta cu interfata grafica accesibila web pentru administrarea bazei de date, care sa includa urmatoarele facilitati:

- construirea si executare scripturi SQL
- gestionarea obiectelor bazei de date
- efectuarea de functii de backup si restaurare;
- administrare a utilizatorilor
- monitorizarea bazei de date si vizualizarea fisierelor de tip log
- vizualizarea in timp real a incarcarii bazei de date, a activitatii utilizatorilor, a operatiilor mari consumatoare resurse (I/O si CPU) precum si rapoartarea acestor evenimente catre administratori

Din punctul de vedere al operatiunilor de backup, baza de date trebuie sa permita operatiuni de backup si restaurare a datelor in regim de lucru online, salvarea totala si/sau partiala a bazei de date atat pe disc cat si direct pe banda iar toate aceste operatiuni sa fie facute intr-o forma unitara, centralizata si usor de administrat. De asemenea pentru optimizarea timpului alocat acestor operatiuni baza de date trebuie sa permita compresia si efectuarea de backup numai pentru fisierele care au suferit schimbari de la ultimul backup si pentru fisierele nou create (backup incremental) si sa permita citirea si scrierea paralela (simultan din/in mai multe fisiere) in timpul operatiilor de backup si restore. In functie de nevoie baza de date trebuie sa permita, pe baza datelor de backup restaurarea partiala asigurand o imagine consistenta a acesteia de la un moment de timp specificat de cel ce realizeaza operatia de restaurare.

Ca mecanisme de securitate oferite, baza de date trebuie sa permita aplicarea simultana a mai multor politici de securitate pe un acelasi obiect al bazei de date precum si posibilitatea de a restrictiona, prin mecanisme native, accesul utilizatorilor la nivel de inregistrare si coloana intr-o tabela. Mecanismele de securitate ale bazei de date trebuie sa asigure filtrarea continutului in functie de utilizator, astfel:

- implementarea conceptului de securitate multi-nivel prin clasificarea in mod diferit a randurilor dintr-o tabela in functie de nivelul de acces pe care trebuie sa-l indeplineasca un utilizator pentru a putea accesa acea informatie ;
- implementarea nativa si in mod transparent fata de aplicatii, a unui mecanism de etichetarea si clasificare a datelor la nivel de inregistrare, permitand in acelasi timp si aplicarea mai multe politici de clasificare asupra aceleiasi tabele astfel incat utilizatorii sa acceseze doar datele pentru care sunt autorizati ;

- sa permita automat, clasificarea datelor în cadrul bazei de date, prin adaugarea de etichete de securitate (public, secret de serviciu, confidential, etc) astfel încât un utilizator să aibă acces doar la datele care corespund nivelului său de clasificare.

Din prisma activitatilor de audit baza de date va oferi o lista cu operatiile pe care un grup sau o clasa de utilizatori le poate executa si va avea abilitatea de a se ajusta la gradul de detalii, capturate de catre facilitatea de audit prin introducerea de politici de audit care sa determine cand un utilizator este sau nu auditat (spre exemplu situatia cand utilizatorul acceseaza doar anumite informatii dintr-o tabela sau cand conectarea nu se face printr-o anumita aplicatie).

Din punct de vedere functional informatia si datele prelucrate de solutia de Business Inteligence trebuie sa fie disponible si sincronizate in timp real cu informatia bruta existenta in sistem.

Se vor licenta minim de 32 de core-uri. Componenta va rula in arhitectura activ-activ.
Pentru schimbul de date cu sisteme informatice existente la nivel de baza de date, se va folosi un mecanism de replicare a datelor cu urmatoarele functionalitati:

- transportul datelor intre baza sursa si destinatie sa se realizeze arhivat si criptat asfel incat se se minimizeze traficul pe retea si sa se asigure securitatea datelor;
- sa existe mecanisme de replicare a tabelelor (inclusiv tabele IOT), precum si a tuturor comenzilor DDL;
- sa nu existe dependenta de versiune si platfoma a bazei destinatie (in cadrul PIAS existand mai multe versiuni instalate pe platfome tehnologice multiple);
- sa permita replicarea intregii baze de date, sau a unui set de tabele, sau chiar a unei singure tabele. De asemenea sa permita doar replicarea unui subset din coloanele, sau randurile tabelei;
- sa permita transformarea comenzilor de la sursa (de exemplu un DELETE la sursa sa poata fi transformat intr-un INSERT la destinatie);
- procesul de replicare trebuie sa fie unul extern bazei de date, iar sistemul sursa nu trebuie sa astepte un nici un fel dupa procesul de replicare, iar consumul de resurse sa fie minim;

- trebuie sa permita pastrarea bazelelor de date destinatie deschise astfel incat poata fi utilizate pentru operatiunile curente specifice aplicatiilor care ruleaza pe aceste baze de date;
- sa aiba integrate in interiorul procesului de replicare functionalitati de comparare si reparare ale datelor, astfel incat sa fie posibila introducerea de noi tabele in replicare, cat si reparare sincronizarea a tabelelor existente in replicare;
- trebuie sa dispuna de un mecanism de captura si replicare a modificarilor in momentul in care acestea au loc, fara sa astepte finalizarea tranzactiei, astfel incat sa fie reduse semnificativ intarzierea la transferul datelor, necesarul de banda utilizat si memoria consumata de procesul de replicare;
- sa permita transmportul datelor prin retele LAN si WAN in care sunt instalate mecanisme de securitate specifice echipamentelor firewall

3.1.10.4. Subsistemul serviciilor auxiliare

Serviciile auxiliare sunt în principal servicii de natură tehnică, care asigură funcţionalităţi transversale de bază pentru celelalte subsisteme componente ale sistemului. Serviciile de securitate si cele de infrastructura sunt detaliate în secţiunile relevante din acest document. Prezentăm în continuare serviciile administrative.

3.1.10.4.1. Cerinţe specifice componentei de administrare şi monitorizare

Această componentă va fi însărcinată cu funcţionalităţile necesare administrării şi monitorizării sistemului în vederea îndeplinirii obiectivelor de funcţionalitate şi disponibilitate. Este formată din mai multe subcomponente:

- Componenta Tablou de Bord
- Componenta de monitorizare cu sau fara agenti
- Componenta de gestionare a incidentelor.
- Componenta de monitorizare a traficului

Componenta Tablou de Bord va respecta urmatoarele cerinte minimale:

- Componenta trebuie să pună la dispoziţie o interfaţa web cu tablouri de bord individuale pe baza rolurilor, asigurând perspective tehnice şi de business corespunzătoare necesităţilor specifice rolurilor
- Componenta trebuie să permită consolidarea evenimentelor şi a datelor de performanţă în cadrul unei console unice de management al evenimentelor, consolă ce poate consolida date atât din componente interne cât şi din componente third party, în cadrul aceleaşi vederi
- Componenta trebuie să afişeze statusul operațional al serviciilor dintr-o perspectivă geografică
- Componenta trebuie să asigure operatorilor posibilitatea de a aloca un eveniment unui operator/unui grup de operatori şi de a adăuga comentarii evenimentelor
- La atingerea unui anumit prag de evenimente asignate unui operator, soluția trebuie să

  permită asocierea unui punctaj acestuia în vederea evaluării activităţii operatorului;

  asocierea punctajului trebuie să poată fi comunicată operatorului prin mesaje de tip pop-up
- Componenta trebuie să permită utilizatorilor autentificarea prin integrarea cu un serviciu de tip director (LDAP)
- Componenta trebuie să permită încărcarea unei imagini de profil (poza) specifică operatorului

- Pentru operatorii înregistraţi, soluţia trebuie să permită adăugarea unei adrese de e-mail, adresă poate fi vizibilă sau invizibilă altor utilizatori şi poate sa fie folosită pentru notificări sau pentru resetarea parolei de acces in aplicaţie

- Componenta trebuie să includă o Bază de Date a Elementelor de Configuraţie actualizată şi populată automat, aproape în timp real prin intermediul componentelor de monitorizare. Aceasta bază de date trebuie să ofere o vedere asupra topologiei serviciilor, inclusiv asupra elementelor dinamice de infrastructură, cum este virtualizarea.
- Baza de date a elementelor de configuraţie trebuie să permită federarea (integrarea cu alte surse, astfel încât sursele în cauză menţin controlul asupra datelor) cu alte componente de tip bază de date a elementelor de configuraţie externe, produse de acelaşi furnizor sau de furnizori diferiţi.
- Componenta trebuie să asigure reconcilierea datelor
- Modelul de date al componentei trebuie să permită implementarea proceselor ITIL
- Componenta trebuie să permită configurarea rolurilor şi a permisiunilor într-un mod flexibil, pe baza nivelelor de acces şi să ofere un control granular al accesului la rapoarte şi metrici
- Componenta trebuie să poată creea setări personalizate pentru utilizatori sau grupuri diferite, de exemplu personalizarea paginii de start, a ratei de actualizare a paginilor, restricţionarea accesului la anumite tab-uri din aplicaţie.

- Componenta trebuie să permită Single Sign-On (SSO)
- Componenta trebuie să facă auditarea modificărilor efectuate asupra configuraţiei componentei: orice acţiune efectuată cu drepturi de administrator, utilizatorul care a efectuat acţiunea, data şi ora acţiunii, accesul la orice înregistrare, precum şi tipul de acţiune efectuat (citire, tipărire sau prezentare)
- Componenta trebuie să permita modelarea vederilor prezentate in tablourile de bord
- Componenta, pe baza unor hărţi/imagini personalizate, trebuie să prezinte statusul operațional al serviciilor de business
- Componenta trebuie să asigure modalităţi de suprimare a mesajelor duplicat şi a mesajelor parte dintr-un "message storm". Aceste funcţionalităţi trebuie să fie configurabile şi să poată avea loc cel puţin la nivelul serverului de management. Deduplicarea mesajelor trebuie să se se realizeze cu incrementarea unui contor. Astfel, atunci cand este recepţionat un nou mesaj, acesta este comparat cu mesajele existente şi dacă este identificat un duplicat, sistemul trebuie doar să incrementeze un contor în cadrul consolei de management al evenimentelor şi să ignore noul mesaj. Pe baza textului mesajului, a anumitor atribute ale mesajului, pe baza intervalului de timp între evenimente şi pe baza numărului de evenimente duplicat trebuie să se poată defini evenimente duplicat
- Componenta trebuie sa permită alocarea unui punctaj, operatorului ce a definit un nou

  eveniment duplicat, în vederea evaluării performanţei in operare.

- Componenta trebuie să ofere o reprezentare grafică a statusului operaţional al elementelor de configurație. Modificările descoperite la nivelul elementelor de configurație/ relațiilor dintre ele sunt actualizare automat în reprezentarea grafica a statusului operaţional. Componenta trebuie să poată relaţiona evenimentele recepţionate cu elementele de configurație din reprezentarea grafică

- Componenta trebuie să permită nativ evenimentelor conţinute în consola de monitorizare exportarea acestora în format Excel şi CSV
- Componenta trebuie să asigure comunicarea bidirecţională cu componenta de gestionare a incidentelor, in vederea obtinerii următoarelor capabilități:
    - crearea automată sau manuală de incidente la momentul producerii unei alerte şi respectiv închiderea automată a alertei la momentul închiderii incidentului asociat;
    - oprirea manuală a funcţionării unui echipament din consola unică de evenimente va produce în componenta de gestionare a incidentelor o solicitare de schimbare în vederea implicării şi urmăririi echipelor de specialişti ce vor gestiona procesul de întrerupere a funcţionării echipamentului;

- vizibilitatea în cadrul consolei unice de evenimente a tuturor schimbărilor planificate şi a incidentelor înregistrate în componenta de gestionare a incidentelor.

Componenta de monitorizare cu sau fără agenţi va respecta următoarele cerinţe minimale:

- Soluţia trebuie sa poată face monitorizare cu sau fără agenţi a infrastructurii IT, a serviciilor IT si a performantei aplicaţiilor
- Soluţia trebuie sa monitorizeze implicit cu sau fără agenţi echipamente de reţea, sistemele de operare si aplicaţii
- Monitorizarea trebuie sa se realizeze cu sau fară instalarea fizică a agenţilor pe sisteme
- Cu aceeaşi aplicaţie se oferă posibilitatea monitorizării parametrilor echipamentelor de reţea. Aceşti parametri trebuie sa fie, cel puțin, disponibilitate si performanta (lăţime de banda)
- Vendorul trebuie să deţină şabloane standard pentru monitorizarea de aplicaţii enterprise care pot fi achiziţionate ulterior
- Soluţia, pentru aplicaţii non-standard sau pentru metode de monitorizare conform proceselor organizaţiei, trebuie sa permită configurarea de şabloane noi
- Soluţia permite iniţierea unor acţiuni corective cum ar fi restartarea echipamentelor, curăţarea spaţiului de pe disk si executarea comenzilor
- Soluţia trebuie sa pună la dispoziţia operatorilor o interfaţa grafica Web ce poate fi personalizată în funcţie de rol
- Soluţia permite notificarea operatorilor folosind metode ca: e-mail, SNMP trap, pager, POST, si alerte in baze de date
- Soluţia este capabilă să monitorizete sisteme cum ar fi: Windows, Linux si UNIX precum si metodele de conectare specifice cum ar fi TELNET, rlogin, HTTP, SSH, şi NetBIOS

- Soluţia suportă monitorizarea mai multor parametrii de sistem, dar cel puţin a următorilor parametrii de sistem: CPU utilization, Database, DHCP, Disk space, Memory, Network, NT event log, UNIX resources, Windows resources

- Soluţia trebuie să permită setarea pragurilor de monitorizare utilizând baseline-uri (praguri de referinţă) şi posibilitatea impunerii unor astfel de praguri dependente de anumite perioade de timp sau date calendaristice
- Soluţia trebuie sa suporte native monitorizarea Web: Link check, URL, Web Server, Web service
- Soluţia este capabilă să monitorizeze următoarelor servicii de reţea: DNS, FTP, Network bandwidth, Ping, Port, SNMP by MIB, SNMP, SNMP trap;
- Componenta de monitorizare cu sau fară agenţi permite integrarea cu componenta tabloul de bord in vederea obţinerii unei console unice în care operatorii pot vizualiza centralizat evenimentele transmise.
- Pentru a permite monitorizarea parametrilor si a serviciilor de mai sus componenta trebuie licențiată pentru toate echipamentele hardware ofertate.

Componenta de gestionare a incidentelor va respecta următoarele cerinţe minimale:

- Să ofere fluxuri predefinite pentru managementul apelurilor, incidentelor, problemelor, cererilor, schimbărilor şi a nivelurilor de servicii, conform bunelor practici ITIL v3
- Să permită înregistrarea unică a apelurilor, incidentelor, schimbărilor, problemelor și cererilor
- Să permită accesul a minim 10 utilizatori concurenți cu drepturi de a gestiona (deschide, actualiza, închide) înregistrărilor de apeluri, incidente, probleme, cereri

- Să permită gestionarea de apeluri prin următoarele mijloace – email, web sau telefon
- Să permită idenditificarea rapidă a unui client și a serviciilor de care beneficiază acesta
- Să suporte o bază de date compatibilă ODBC
- Să suporte definirea de active într-o bază de date unică, accesibilă și în alte înregistrări (apeluri, incidente, schimbări, probleme, cereri)
- Să permită gestionarea de dispoziții de lucru
- Să permită definirea de tipuri de roluri de utilizator; în funcție de rolul asociat, un utilizator are acces la anumite înregistrări și poate să efectueze anumite acțiuni
- Să permită importul de informații din alte surse externe
- Să permită analizarea și identificarea serviciilor cu probleme
- Să permită configurarea interfețelor folosite de utilizatorii finali
- Să permită configurarea de rapoarte
- Să permită definirea de reguli automate în procesele de gestionare a apelurilor, incidentelor, problemelor sau schimbărilor
- Să permită gestionarea proactivă a incidentelor
- Să permită prioritizarea înregistrărilor, prin asocierea de niveluri de impact și urgență.
- Să permită definirea de niveluri de servicii pentru înregistrările de incidente; nivelurile de servicii trebuie să fie de tip timp de răspuns și timp total de rezolvare
- Să ofere un flux predefinit de definire a unui nivel de serviciu, astfel încât procesul de definire a nivelurile de servicii să fie facilă
- Să permită analizarea stării unui incident în funcție de nivelul de serviciu asociat
- Să permită definirea de rapoarte pe nivele de servicii
- Să ofere nivele de servicii predefinite pentru incidente, probleme și schimbări
- Să permită definirea de niveluri serviciu de tip răspuns și disponibilitate
- Să permită definirea de nivele de servicii care țin cont de un anumit orar de lucru și a unui fus orar

- Să permită posibilitatea de suspendare a unui nivel de serviciu
- Să permită definirea de notificări pe mail atunci când sunt îndeplinite anumite condiții, de exemplu – trimitere de mail atunci când se schimbă persoana asignată sau când n-a mai fost actualizată de un anumit timp
- Să permită definirea de reguli automate de asociere a unui incident către un anumit grup, atunci când sunt îndeplinite anumite condiții
- Să permită definirea de șabloane cu câmpuri predefinite pentru categorii de incidente repetitive, astfel încât să se scurteze timpul de înregistrare de incidente
- Să permită identificarea de incidente deschise pe același serviciu, astfel încât să se evite duplicarea de incidente
- Să ofere posibilitatea de alertare și notificare a unui coordonator de incidente atunci când se apropie timpul de expirare a unui nivel de serviciu (când mai este 25% din timpul de răspuns al nivelului de serviciu)

Componenta de monitorizare a traficului va respecta următoarele cerinţe minimale:

- Sa asigure posibilitatea de captura şi analiza a pachetelor de date de la nivelul infrastructurii de retea precum si inspectia detaliata a fluxurilor de trafic de date
- Sa permita urmarirea fluxurilor de date între aplicaţii, sisteme şi clienţi
- Sa permita urmarirea traficului la nivel de networking în centre de date dedicate sau partajate

- Sa sustina verificarea configuraţiilor actualizate la nivelul echipamentelor de comunicaţie precum si verificarea politicilor de tip calitate a serviciului.

- Sa permita monitorizarea traficului de retea, prin intermediul unor grafice generate la cerere privind utilizarea echipamentelor de comunicaţie utilizând criterii de clasificare bazate pe grupuri DSCP, aplicaţii, servere, sau tipuri de încapsulări

- Sa permita analiza in vederea determinarii zonelor critice şi furnizarea de informatii necesare alocărilor optime de resurse de reţea si identificarea marilor consumatori de resurse,

- Sa asigure analiza tipului de utilizare a echipamentelor de comunicaţie de către aplicaţii, sisteme sau maşini virtuale

- Sa permita urmarirea la nivel de cadru şi pachet, inclusiv cu o metoda de captura a pachetelor pentru analiza offline, pentru a permite analiza problemelor complexe de performanţă prin capturi declanşate de erori, evenimente, sau depasiri ale unor limite de performanţă predefinite.

- Sa permita analiza problemelor de performanţă ce pot apare în infrastructuri fizice sau virtuale.
- Sa suporte cel putin urmatoarele protocoale:

- HTTP şi HTTPS

- protocoale SAN si protocoale specifice bazelor de date si din categoria Peer-to-peer
- protocoale SIGTRAN si protocoale suportate de echipamente de tip switch/router: VRRP/HSRP, LACP,
- TCP si UDP over IP inclusiv IPv6
- Voice over IP (VoIP): Session Initiation Protocol (SIP), Media Gateway Control Protocol (MGCP), Skinny Client Control Protocol (SCCP), Real-Time Transport Protocol/Real-Time Control Protocol (RTP/RTCP)

- Sa poata functiona ca maşină virtuală cel putin în VMware vSphere 5.1 (ESXi 5.1) si

  sa suporte cel putin browserele Microsoft Internet Explorer 9.0+ sau Firefox ESR 10.0+
- .Sa aiba o capacitate de analiza a traficului de peste 3.8 Gbps si o capacitate de capturare pachete pe disk de pana la 1.9 Gbps.

- Sa permita urmarirea performanţei aplicaţiilor si izolarea problemelor de performanţă

  la nivel de reţea, server sau aplicaţie.

3.1.10.4.2. Managementul utilizatorilor şi accesul la sistem

Sistemul informatic va asigura securitatea datelor printr-un sistem de limitări ale accesului bazat pe drepturi şi parole, defalcat pe mai multe niveluri. Drepturile de acces ale utilizatorilor vor putea fi configurate din interfaţa aplicaţiei de către administratorul sistemului.

Utilizatorii pot accesa anumite funcţionalităţi disponibile în cadrul sistemului informatic în funcţie de matricea de drepturi ataşată profilului său de utilizator.

Autentificarea utilizatorilor trebuie să se poată realiza cu oricare din următoarele metode:

- prin nume de utilizator şi parolă pentru utilizatorii interni, angajaţi ai CAS
- prin folosirea certificatelor digitale înregistrate în sistem, asociate cu un cont de utilizator extern PIAS, pentru medici/furnizori.

Utilizatorii interni se autentifică pe baza numelui de utilizator şi a parolei asignate şi au acces la funcţionalităţile disponibile în cadrul sistemului.

Drepturile de acces se pot da atât individual, cât şi la nivel de grup. Utilizatorii, pot aparţine mai multor grupuri, iar drepturile lor constau în suma dintre drepturile individuale şi cele de grup. În funcţie de drepturile utilizatorilor, fiecare dintre aceştia va accesa o anumita configuraţie de meniu, cea la care are dreptul, şi va putea efectua operaţiuni doar pentru operaţiunile la care are drept de scriere.

Utilizatorii externi se autentifică prin intermediul aplicaţiei de raportare în baza unui cont de utilizator, cu nume şi parolă, dar şi prin utilizarea unui certificat digital calificat înregistrat în sistemul central.

Utilizatorii externi nu au asociate roluri propriu-zise, accesul acestora fiind restricţionat pe baza drepturilor ce reies din contractul furnizorului, în baza certificatelor digitale asociate cu furnizorul.

Cerinţe pentru componenta de management al utilizatorilor

Stocarea profilelor de utilizatori se va realiza în mod centralizat, folosind un director LDAP standardizat. Având în vedere numărul de utilizatori estimaţi ai sistemului, soluţia de stocare a profilelor de aplicaţie pentru aceştia trebuie să utilizeze metode eficiente de stocare –

nivelul de stocare a datelor trebuie să fie o baza de date interna sau externa produsului, dar inclusă în soluţie.

Informaţiile stocate în directorul LDAP trebuie să asigure redundanţă, disponibilitate şi securitate ridicată atât la nivelul aplicativ (server LDAP de prezentare) cât şi la nivel de stocare a datelor, după cum urmează:

- Accesul la baza de date trebuie permis doar pentru utilizatorii care au acest drept; filtrarea accesului trebuie să se poată face cel puţin prin nume de utilizator, parolă şi rol de acces (care sa definească operaţiunile pe care utilizatorul are dreptul să le realizeze asupra datelor)
- Accesul la nivel aplicativ (server LDAP) trebuie să fie filtrat în funcţie de utilizator şi reguli de acces (liste de acces) definite la nivel de atribut (de exemplu un utilizator privilegiat să poată căuta, adăuga, vizualiza date de profil ale altor utilizatori, dar doar utilizatorul să îşi poată modifica propriile informaţii din profil)
- Întreaga soluţie pentru directorul de utilizatori trebuie să suporte – la toate nivelele ei, redundanţa şi înaltă disponibilitate prin posibilitatea implementarii unui cluster activ-activ, cu balansare hardware sau software internă (prin mecanisme optimizate specifice arhitecturii soluţiei director)
- Datele cu caracter special (de exemplu parolele) trebuie să fie protejate la acces şi modificare prin mecanisme superioare autentificării – de exemplu criptare.

Componenta de management al utilizatorilor trebuie să asigure funcţionalităţi de virtualizare a surselor de identităţi; accesarea datelor despre utilizatori trebuie să fie posibila atât din baze de date cât şi din directoare LDAP, cu posibilitatea de agregare selectivă a profilelor şi expunerea acestor informaţii în format LDAP către alte sisteme (de exemplu către portalul intern), pentru a putea acomoda uşor surse de profile de utilizatori fără duplicarea acestora, inclusiv reutilizarea profilelor de utilizatori existente în actualele sisteme informatice ale CNAS. Subsistemul trebuie să permită integrarea cu celelalte componente ale sistemului general astfel încât să existe o singură sursă de utilizatori pentru toate nivelele (server de aplicaţie, subsistem de fluxuri de lucru şi procese, subsistem de raportare).

Soluţia trebuie să permită utilizarea clienţilor de tip LDAP provenind de la alţi producători software, dar să includă în acelaşi timp un client de tip web pentru modificarea parametrilor

de funcţionare a soluţiei şi editarea datelor stocate în director (cu aplicarea restricţiilor de acces specifice utilizatorului autentificat)

Componenta de tip director LDAP trebuie să permită definirea şi impunerea politicilor de parole la nivel de ramură a organizaţiei (organizaţie, grup, subgrup); fiecare astfel de ramură trebuie să poată utiliza politici de parole diferite; modulul va oferi extinderea claselor utilizate pentru definirea atributelor utilizatorilor

Subsistemul de administrare utilizatori va trebui să permită integrarea cu alte directoare LDAP care respectă acest standard.

Directorul LDAP trebuie sa suporte posibilitatea implementarii în arhitectură cu înaltă disponibilitate (cluster activ-activ). Se solicita cel putin 4 licente pentru aceasta componenta.

Cerinţe pentru componenta de monitorizare a activităţii utilizatorilor

Componenta trebuie să suporte următoarele funcţionalităţi:

- Componenta trebuie sa perimtă monitorizarea performanţei si a disponibilităţilor aplicaţiilor din perspectiva utilizatorilor finali
- Componenta va dispune de o interfaţă web pentru adiminitrare.
- Componenta va oferi rapoarte detaliate de performanţă la nivelul aplicaţiilor măsurând atât timpii de răspuns pentru pagini individuale cât şi cumulat la nivel de tranzacţie definită.
- Componenta va urmări fiecare pas tranzacţional din cadul unui proces, putând să prezinte grafic interacţiunile dintre toate componentelor sistemului.
- Componenta trebuie să ofere informaţii despre impactul asupra procesului de business.
- Componenta va realiza diagrame de topologie cu scopul identificării arhitecturii mesajelor.
- Componenta trebuie să permită detecţia automată a tranzacţiilor
- Componenta trebuie să poată monitoriza în timp real cererile şi răspunsurile de tip HTTP/HTTPS folosind tehnici pasive TCP/IP de monitorizare a traficului de reţea
- Componenta trebuie să permită captarea sesiunilor create de utilizatorii reali.

- Componenta va oferi posibilitatea protejării datelor cu caracter sensibil, de tip parola, astfel aceste informaţii nu vor fi salvate in baza de date.
- Componenta trebuie să permită definirea de notificări în cazul tranzacţiilor critice, aceste notificări trebuie să fie trimise pe email.
- Componenta va permite identificare sesiunilor utilizatorilor pe baza identificatorilor unici permiţând urmărirea tranzacţiilor end to end.

3.1.11. Securitatea sistemului

Atacurile asupra unui sistem informatic sunt o ameninţare reală în ziua de azi. Tot mai multe tehnologii şi unelte avansate sunt disponibile pe Internet la îndemâna oricui. Pentru a preveni aceste atacuri, sistemul informatic trebuie analizat riguros, identificate punctele slabe şi luate măsurile corespunzătoare. Modalităţile de asigurare a securităţii sistemului informatic propus sunt următoarele:

- acces autentificat la sistem
- sisteme firewall / security gateway
- protocoale de comunicare criptate

Principalele obiective ale mecanismelor de protecţie şi securitate a datelor sunt corelate cu segmente ale sistemului de acces la sistemul global şi local, fiind necesară respectarea următoarelor cerinţe minimale obligatorii:

- proceduri de autorizare validate global;
- mecanisme şi proceduri pentru managementul autentificării şi autorizării utilizatorilor;
- metode înglobate în sistem pentru asigurarea şi oferirea unui înalt nivel de securitate a datelor, proceselor şi tranzacţiilor;
- politica aprobată a controlului şi verificării accesului;
- preventie a intruziunilor.

Subsistemul de securitate va include componente specializate pentru accesul autentificat la sistem pentru resursele de tip aplicatii web si servicii web si o componentă de colectare şi analiză a evenimentelor de securitate.

Cerinte pentru componenta de acces autentificat la resursele de tip aplicatii web

Datorita faptului ca sistemul trebuie sa asigure ergonomie sporita pentru utilizatori componenta de acces la aplicatii web trebuie sa puna la dispozitie o serie de facilitati astfel incat sa implementeze mijloace moderne de utilizare a tehnologiilor de tip “mobile”, pe scara larga, ca mijloc de acces in sistem sau ca factor de autentificare.

Componenta de acces la aplicatii web va trebui sa exploateze capabilitatile in domeniul stocarii, integrarii si utilizarii de credentiale de tip certificate digitale pe care smartphone-urile si tabletele le detin, atat in domeniul software (de tip repository) cat si in domeniul hardware (integrare cu dispozitive criptografice externe de stocare a certificatelor digitale).

Trebuie sa poata fi utilizate, cu echipamentele “mobile”, cel putin doua tipuri de dispozitive externe de stocare a certificatelor utilizatorilor din cele enumerate (minim unul certificat FIPS 140-2 Level 3):

- Smart card cu interfata PKCS#11 cu contact
- Smart card “contactless” (eg. NFC).
- Secure SD card criptografic.

Pentru a se asigura unificarea experientei de utilizare dar si pentru a creste nivelul de securizare si a scadea costurile operatiunilor administrative legate de controlul accesului la aplicatiile web, se consideră necesară utilizarea unei solutii care sa satisfaca cel putin urmatoarele cerinte functionale si tehnice:

- Sa protejeze resursele de tip web impotriva acceselor neautorizate – atat din interiorul cat si din exteriorul retelei
- Nici o resursa web din interiorul sistemului nu trebuie sa poata fi accesata direct din exterior
- Sa ceara utilizatorilor sa introduca date de identificare pentru accesul la aplicatii
- Sa permita impunerea unor filtre de acces (operatiuni de autorizare) – cel putin interval orar si locatie de retea de unde s-a initiat cererea de acces
- Sa permita administratorului sistemului sa aleaga mai metode de autentificare si autorizare diferite pentru fiecare grup de resurse in parte

- Sa ofere o interfata de administrare de tip web pentru accesul facil la configurari, care sa poata fi accesata doar de catre administratorii de securitate ai solutiei
- Sa ofere SSO – autentificare unica pentru accesul la resurse; pe parcursul unei singure sesiuni de lucru utilizatorul fi autentificat o singura data , dupa care va putea accesa fara reautentificare toate aplicatiile web pentru care are drept de acces.
- Fiecare utilizator sa fie identificat de sistem pe baza unei sesiuni
- Sistemul sa permita administratorilor terminarea manuala a sesiunilor utilizatorilor
- Dupa un timp configurabil de inactivitate sesiunile utilizatorilor trebuie sa fie terminate in mod automat
- Numarul de sesiuni pe care un utilizator le poate deschide trebuie sa poata fi limitat de catre administratori
- Toate evenimentele de acces – autentificari reusite, autentificari nereusite, autorizari reusite, autorizari nereusite trebuie sa poata fi auditate
- Datele colectate prin auditarea accesului trebuie sa fie stocata intr-o baza de date pe care sa poata fi rulate in caz de nevoie rapoarte
- Toate componentele software ale solutiei de control acces trebuie sa suporte posbilitatea rularii in mod disponibilitate ridicata folosind functionalitati native
- Stocarea configuratiilor si a politicilor de acces la resursele web sa se realizeze in baza de date a sistemului, fara a exista nevoia unui depozitar proprietar de date
- Sa permita accesarea simultana a mai multor surse de identitati pentru realizarea autentificarii si autorizarii, fara a implica duplicarea informatiei sau crearea unui metadirector
- Toate politicile de control al accesului trebuie sa poata fi definite utilizand interfata web a solutiei, fara a necesita cunostinte de programare sau rularea de scripturi pe server
- Sa ofere autentificare multi nivel combinand orice metode de autentificare disponibile (de exemplu nume utilizator si parola pentru nivel privat, certificat digital pentru nivel confidential, etc)
- Sa suporte cel putin urmatoarele metode de autentificare:

    - Nume de utilizator si parola
    - Certificate digitale x.509
    - Smart card
    - Token-uri fizice cu PIN
    - Bazata pe formulare web
    - API-uri de autentificare pentru dezvoltari
- Schimbarea comportamentului standard (refuza acces sau permite acces pentru resursele neprotejate)
- Nivelul de auditare trebuie sa fie configurabil (succes, nereusita, etc)
- Sa realizeze criptarea informatiei transferata intre componentele sistemului si clienti (HTTPS, LDAPS, etc)
- Solutia de control acces sa ofere integrare cu solutia de stocare a profilelor de utilizatori si cu cea de administrare unitara
- Solutia de control acces trebuie sa suporte utilizarea într-o arhitectura pe mai multe niveluri:
    - Nivel server de acces- server central de control acces, care primeste si trateaza cererile de autentificare, autorizare si auditare
    - Nivel proxy - integrare cu serverele web de tip proxy pentru blocarea tentativelor de acces la resursele protejate
    - Nivel de integrare – solutia de control acces trebuie sa foloseasca directorul centralizat de utilizatori al solutiei (LDAP)
    - Nivel de stocare – datele sistemului de control acces (politici de acces, date de auditare) trebuie sa fi stocate intr-o componenta specializata de tip baza de date, separat de serverele de acces pentru a asigura ca toate serverele de acces au acces la aceleasi informatii
- Sesiunea utilizatorului trebuie sa nu fie afectata in cazul in care unul din serverele de control acces este oprit.

- Suport pentru acces federalizat la resursele expuse de organizatii partenere si toate standardele urmatoare:
    - SAML 1.1, 2.0
    - OpenID 2.0
    - Suport dedicat sau integrare cu componente de securizare a serviciilor web care suporta standardul WS-Federation
- Solutia trebuie sa fie capabila sa converteasca token-uri de securitate, astfel incat sa asigure compatibilitatea cu alte solutii de control acces. Astfel, trebuie asigurata conversia pentru token-uri nume utilizator, X.509 (certificate digitale), Kerberos, SAML 1.1/2.0

Sistemul va trebui sa fie capabil sa utilizeze mecanisme de autentificare de tip OOB (Out-of-Band) care isi bazeaza securitatea pe canale de comunicare in afara celui din serverul web si browser. Stabilirea unui al doilea canal de comunicatie direct cu sistemul imbunatateste semnificativ securitatea sistemului impotriva vulnerabilitatilor prezente in aplicatiile de tip browser (de exemplu atacurile de tip “man-in-the-middle)”.

Prin urmare sistemul trebuie sa detina mecanisme de autentificare puternice care utilizeaza autentificare de tip 2 - factor (bazate pe certificate digitale) și permite utilizatorilor să se autentifice prin intermediul tehnologiei mobile existente, cum ar fi smartphone-uri și tablete, prin semnarea digitala de coduri QR (Quick Response Code). Aceasta tehnologie reprezinta o versiune substantial imbunatatita a tehnologiei de tip OTP (One-Time-Password), permitand utilizatorilor finali sa-si utilizeze mijloacele tehnologice personale ca si dispozitive de autentificare.

Avand in vedere importanta proiectului, utilizatorii tinta si aria de aplicabilitate sunt obligatorii urmatoarele:

- Aplicatiile ce vor fi livrate trebuie sa fie de tip COTS (Commercial off the shelf) cu capabilitati de personalizare;
- Interfetele aplicatiilor trebuie sa fie disponibile minim in limba romana.

Autentificarea utilizatorilor la sistem

Sistemul trebuie sa fie capabil sa permita accesul la DES al utilizatorilor de pe PC/laptop prin scanare/semnare electronica de QR Code-uri utilizand smartphone-uri și/sau tablete si accesul

utilizatorilor direct de pe aceste tipuri de dispozitive, in mod unitar, prin expunerea acelorasi interfete.

Sistemul trebuie sa fie capabil sa:

- Permita definirea si configurarea (prin completarea detaliilor necesare) listei de Autoritati de certificare considerate de incredere;
- Permita definirea si configurarea listei utilizatorilor care au drept de acces in sistem (la nivel de lista de acces);
- Permita realizarea de Single Sign On la diverse aplicatii web expuse in cadrul sistemului, si in acest sens trebuie sa fie capabil sa realizeze managementul credentialelor utilizatorilor si al metodelor de autentificare la acestea (minim LDAP, autentificare de tip basic access, autentificare de tip form-based);
- Valideze, anterior autentificarii, toate certificatele digitale utilizate pentru autentificare prin protocol OCSP (RFC 6960) si sa permita, ca alternative, validarea prin CRL-uri (Certificate Revocation List) stocate local sau pe un server de tip LDAP;

Autentificarea utilizatorilor la sistem utilizand QR Code-uri

Aceasta modalitate de implementare va fi asigurata printr-o aplicatie dedicata instalata pe smartphone-uri și/sau tablete care, ca si in cazul aplicatiei de inrolare, va fi pusa la dispozitia utilizatorilor finali (prin download direct sau prin intermediul Google Play/Apple App Store).

Fluxul de autentificare trebuie sa fie urmatorul:

- Utilizatorul introduce URL-ul sistemului (in browser-ul de pe laptop sau PC);
- Sistemul prezinta pagina de login, inclusiv QR Code-ul semnat electronic;
- Utilizatorul scaneaza QR Code-ul, il decodifica, il semneaza electronic si-l transmite sistemului;
- Sistemul permite logarea utilizatorului.

Aplicatia de autentificare trebuie sa fie capabila sa utilizeze in acest proces atat containere criptografice software (de tip PKCS#12), cat si containere criptografice hardware (de tip smartcard/token criptografic cu contact – PKCS#11 sau contactless – de ex. NFC).

Toate semnaturile electronice utilizate in cadrul acestui flux se vor crea in baza standardelor existente, iar toate certificatele utilizate trebuie sa fie in format standard x509v3. Totodata toate operatiunile de verificare a semnaturilor electronice vor implica verificarea validitatii certificatelor digitale utilizate pentru crearea semnaturilor prin OCSP (Online Certificate Status Protocol), in conformitate cu RFC 6960.

Accesul utilizatorilor la sistem direct de pe smartphone-uri și/sau tablete

- Sistemul trebuie sa fie capabil sa permita accesul utilizatorilor de pe smartphone-uri și/sau tablete prin protocol HTTPS, cu autentificare mutuala si ulterior verificarii validitatii certificatelor digitale prin protocol OCSP.

Se solicita cel putin 4 buc licente pentru aceasta componenta.

Cerinte pentru componenta de acces autentificat la resursele de tip servicii web

Aceasta componenta trebuie să permită integrarea, accelerarea si securizarea sistemelor de tip SOA putand rula pe sisteme de operare de tip Microsoft Windows, Linux sau pe platforma proprie. Componenta trebuie să fie uşor de implementat şi să dispună de administrare facilă, centralizată; componenta de interfatare pe baza de servicii web trebuie să se bazeze pe o arhitectură scalabilă si sa suporte posibilitatea rularii in mod inalta disponibilitate de tip activ-activ.

Componenta trebuie sa suporte utilizarea în conjuncţie cu componente software specializate care sa ofere integrare nativa cu modulele de control acces, stocare profile de utilizatori, fluxuri de lucru si procese de integrare.

Componenta trebuie sa suporte integrarea in timp real cu alte sisteme si aplicatii, deci are nevoie de capabilitati de monitorizare a activitatilor si de limitare a tranzactiilor per client pentru a evita supraincarcarea sistemului.

Din punct de vedere al securizarii, avand in vedere ca aceasta componenta trebuie sa suporte utilizarea ca punct de intrare in sistem pentru cererile externe adresate prin servicii web, componenta de interfatare va trebui sa permita definirea de politici de acces, securizare servicii web (autentificare, autorizare, criptare, auditare), conexiuni criptate, precum si detectarea si blcoarea mesajelor duplicate sau corupte. Componenta trebuie să se asigure că mesajele recepţionate nu conţin atacuri de tip SQL Injection sau coduri maliţioase.

Pentru usurinta in utilizare, crearea si aplicarea politicilor se va realiza in mediu grafic fara a necesita cunostinte de programare.

Componenta trebuie să permită o implementare de tip gateway (poarta de acces), astfel incat orice interactiune pe baza de servicii web sa fie controlata de acest modul. Pentru a asigura

functionarea in cazul in care in sistem apar disfunctionalitati ale echipamentelor fizice, componenta va permite rularea pe cel putin doua echipamente fizice distincte.

Componenta trebuie sa aiba urmatoarele caracteristici:

- Medoda de integrare cu aplicaţiile existente trebuie să fie non destructiv (să nu necesite modificare aplicaţiilor existente).
- Să fie uşor de instalat configurat oferind pentru aceasta o intefaţă web, interfaţa grafica dedicată bazată pe standarde deschisă cât şi interpretor pentru linia de comandă.
- Trebuie să ofere parametri de performanţă ridicată pentru funcţiile de transformare a mesajelor, putând fi folosită pentru decongestionarea serverelor de aplicaţii prin preluarea funcţiilor de procesare XSLT, rutare Xpath, conversie a mesajelor XML şi a altor funcţii consumatoare de resurse pentru a reduce capacitatea de transfer şi pentru a asigura un timp de răspuns ridicat.
- Să suporte un trafic cumulat intre 200 Mbits/sec şi 700 Mbits / sec, şi până la 25000 de conexiuni TCP (acestea includ conexiunile utilizatorilor, aplicatiilor integrate din intranet si internet).
- Să ofere suport pentru validarea din punct de vedere tehnic a mesajelor (verificari de sintaxa, XML schema);
- Să ofere suport pentru transformarea documentelor XML folosind foi XSLT
- Să ofere suport pentru interpretoare şi validarea mesajelor transmise în format SOAP şi JSON
- Să ofere suport pentru a monitoriza apariţia unor greşeli în timpul prelucrării şi poate reacţiona la ea (de exemplu, să permite să formatarea mesajelor de eroare)

- Să ofere suport pentru transformarea documentelor în alte formate decât XML (de exemplu, fişiere plate, delimitate binare sau structura de lungime fixă)

- Să ofere funcţii de securitate la nivel de mesaj şi control al accesului astfel încât mesajele să poată fi: filtrate, validate (validare de schema XML, validare conform WSDL a serviciilor web, filtrare SOAP), criptate (criptarea trebuie să poată fi aplicată la nivel de mesaj la la nive de camp al mesajului), semnate
- Să permită interconectarea cu sisteme de tip antivirus ce folosesc standardul ICAP (Internet Content Adaptation Protocol).

- Să includă accelerator hardware ce oferă o mai mare eficienţă şi siguranţă.

- Să nu stocheze informaţii de business sau de transfer de date.

- Sa fie de clasa HSM si certificată Common Criteria EAL4
- Să accepte standardul de securitate WS-Security pentru autentificare, autorizare, semnături digitale (semnarea şi verificarea semnăturilor) pentru a cripta conţinutul

- Să ofere suport pentru versiuni SAML standard: 2.0, 1.1, 1.0, standardele WS-Trust, XKMS.
- Să ofere suport pentru criptare XML

- Să ofere suport pentru a genera o auto-semnat (auto-semnat) un set de cheie / certificat
- Să ofere suport pentru următoarele infrastructuri de chei publice (PKI): XKMS, RSA, 3DES, DES, AES, SHA, X.509, PKCS, CRLs, OCSP

- Să ofere un depozit sigur pentru certificate şi chei publice

- Să ofere suport pentru lungime de chei RSA de 1024, 2028 şi 4096.

- Să permită determinarea dimensiunii maxime a mesajului de intrare în KB

- Să permită validarea documentelor XML, în conformitate cu definiţia XSD

- Să permită descărcarea de fişiere XSD externe la validarea documentelor XML. (De exemplu, validarea documentelor XML, XSD numai pentru structuri aflate la nivel local), în special, pentru a preveni descărcarea XSD de la serverele HTTP la distanţă

- Să permită specificarea numărului maxim de atribute pentru documentul XML prelucrate şi permit o eroare răspuns adecvat pentru documente salvate în exces criteriilor

- Să permită specificarea numărului maxim de elemente care urmează să fie prelucrate documentul XML şi vă permite să reacţioneze la o eroare în documentele salvate în plus faţă de criteriile
- Să permită specificarea dimensiunii de nod XML (nod) pentru documentul XML şi reacţionarea la o greşeală pe documente
- Să permită protecţie împotriva următoarelor riscuri cauzate de utilizarea tehnologiei XML:
    - Incarcaturi Jumbo (documente foarte mari XML)
    - Elemente recursive
    - MegaTags (nume, de mult timp)
    - Injecţie XPath
    - Injecţie SQL
    - Atac schemă,

- Încapsulare XML (se introduce sistemul de comenzi secţiune CDATA)

- Virusul XML (SOAP ataşamente cu viruşi)

- Să permită filtrarea de documente în raport cu un conţinut fix de documente în orice format

- Să permită tăierea automată de ataşamente SWA (SOAP cu ataşamente)

- Să ofere suport pentru următoarele tehnologii:
  - WS-SecureConversation
  - WS-Policy
  - WS-ReliableMessaging
  - WS-SecurityPolicy
  - WS-SecureConversation
  - WS-Trust
  - SAML
  - LDAP
  - SOAP
- Sa ofere suport pentru implementarea mecanismului de SSO (Single Sign On) pentru servicii web si să ofere interoperabilitate cu regiştri de descriere universală, descoperire si integrare a serviciilor web (UDDI).
- Să ofere posibilitatea configurării ca reverse proxy in DMZ.
- Mecanismul Reverse Proxy trebuie să ofere suport pentru Split SSL Certificate.

- Serverul reverse proxy trebuie să suporte următoarele metode de autentificare:
    - Nume de utilizator/parola;
    - Certificate Client x.509 V3;
    - Autentificare cu token;
    - LDAP.
- Serverul reverse proxy trebuie să ofere suport pentru autentificare cu doi actori de autentificare, prin hardware tokens si combinarea a doua metode de autentificare cu factor singular precum certificatele X.509 si nume de utilizator/parola.
- Să ofere metode grafice de definire a funcţiilor de procesare a mesajelor făcand posibilă implementarea următoarelor scenarii:
    - Rutare complexa de mesaje in mai mulţi paşi – pe baza sursei, destinatiei şi a conţinutului mesajului
    - Filtrare
    - Algoritmi de procesare folosind decizii logice, bucle (loops), procesare paralela, mecanisme de reincercare în caz de eroare.
- Trebuie să ofere suport pentru următoarele protocoale de transport:
    - HTTP
    - HTTPs
    - JMS
    - MQ
    - ODBC
    - FTP
    - IMS Connect
    - NFS
    - TIBCO EMS
    - WebServices proxy (SOAP/HTTP(s), SOAP/JMS)
- Să ofere suport pentru prelucrarea de control (rutare), pe baza conţinutului mesajului de intrare
- Să permită prelucrarea pe baza header-ului de intrare a mesajelor
- Suport pentru jurnalizare, incluzând iSCSI pentru conectivitate către discuri externe.

- Să permită implementarea de funcţii de tip AAA (Autentificare, Autorizare, Auditare).
- Să permită implementarea de arhitecturi de cluster de tip activ – activ cu balansare de trafic
- Din motive de securitate si protectie, nu trebuie să permita instalarea de alte componente software sau alte aplicaţii altele decat cele autorizate de producator.
- Să permită interoperabilitate cu sisteme de monitorizare operatională.

- Să ofere suport nativ pentru protocolul WMQ (versiunea 7.0.1), fără aplicaţii intermediare ca

  JMS şi altele

- Să poate modifica câmpurile de antet MQMD pentru intrare şi mesaje de ieşire

- Să permită o cale de comunicare, precum şi protocolul de cerere-răspuns folosind WMQ

- Să permită comunicare folosind cozi şi publică, abona mecanism

- Să permită controlul numărului de conexiuni la sesiuni WMQ, (de la 1 la n), în special pentru a sprijini prelucrarea mesajelor din coada de intrare FIFO şi FIFO pentru a menţine coada de intrare la coada de ieşire (pentru neschimbat de mesaje între intrare şi cozile de ieşire)

- Să permită procesare paralelă a mesajelor (in sesiuni / fire conectate) de coada de intrare la un mesaj de FIFO ordonate de la coada de intrare la coada de ieşire

- Să accepte conexiunea la server WMQ folosind un protocol securizat SSL

- Să permită modificarea setărilor de PMO (opţiune de mesaje) şi OMG (Get Opţiunea Mesaj)

- Să permită conectarea la WMQ pentru multi utilizatori diferiţi

- Să permită pentru a creşte în mod automat nivelul de securitate pentru a genera sprijin pentru criptarea şi decriptarea mesajelor pentru fluxuri în WebSphere broker de mesaje şi pentru a asigura Proxy Securitate
- Să poată bloca funcţiile http respective, cum ar fi GET, POST, PUT, DELETE
- Să accepte versiunea de protocol HTTP 1.1
- Să permită modificarea (ştergere, modificare, plus) câmpuri header atât pentru cerere şi răspuns
- Să permită emiterea protocolulului HTTP de la orice port
- Să permită schimbarea URL-ului (rescriera adresei URL)
- Să permită integrarea cu serverele UDDI conţine definiţii de servicii
- Să permită crearea de Proxy pentru Servicii Web
- Să permită import de fişiere WSDL pentru generarea de Proxy de Servicii Web
- Să suporte standardul WS-Policy

- Să suporte standardul WS-SecurityPolicy
- Să suporte standardul WS-ReliableMessagingPolicy
- Să permită comunicarea cu următoarele baze de date folosind SQL: Oracle, DB2, Sybase,
- Să permită măsurarea numărului de cereri specifice şi să creeze limitele pe unitate de timp.

  Rata de limitare să poată fi parametrizată pe bază de adrese IP, zi şi oră.
- Să permită colectarea informaţiilor despre erori în funcţionare şi să permită gruparea pe

  categorii şi nivel de criticitate (severitate)
- Să ofere un jurnal pentru a înregistra erori sau diferinte evenimente
- Să permită trimiterea de e-mail-uri cu informaţii despre eroarea in vedera informării

  administratorului de apariţia lor

- Să permită configurarea şi administrarea prin consolă şi conexiuni ssh si Telnet

Se solicita cel putin 8 buc licente pentru aceasta componenta.

Cerinţe pentru componenta de colectare şi analiză a evenimentelor de securitate

Componenta trebuie să dispuna de capabilităţi de monitorizare a evenimentelor de securitate, de management a logurilor şi va cuprinde toate componentele hardware necesare funcţionării depline, precum şi licentele software, incluzând aici şi licenţele terţilor producători, cu suport şi mentenanţă pe 3 ani. Componenta va indeplini şi următoarele cerinţe:

- Să permită colectarea logurilor si monitorizarea unui număr minim de 700 dispozitive de reţea/securitate (servere, firewall-uri, IDS/IPS-uri, routere etc).
- Să ofere posibilitatea dezvoltării ulterioare a soluţiei într-un mod cât mai facil (să fie scalabilă).
- Să ofere posibilitatea identificării de acţiuni repetitive sau tiparuri de evenimente, pe baza cărora se pot stabili reguli de alertare şi se pot adopta politici de securitate.
- Să transforme logurile colectate într-un format comun (normalizat) şi să permită categorisirea acestora în vederea efectuării unei analize ulterioare cât mai facile.
- Să coreleze relaţiile dintre evenimente, deducând semnificaţia acestora şi prioritizându-le.

- Să ofere o varietate de instrumente flexibile de monitorizare, permiţând investigarea şi remedierea potenţialelor ameninţări.
- Să ofere o structură personalizabilă de niveluri de escaladare ale fluxului de lucru pentru a se asigura că evenimentele de interes ajung la personalul avizat în intervalul de timp util.
- Să ofere instrumente de analiză ce permit detalierea evenimentelor şi pentru a apela funcţii, cum ar fi nslookup, Ping, PortInfo, Traceroute, WebSearch, şi Whois.
- Să permită raportarea cu privire la statusul securităţii de reţea prin rapoarte personalizate, realizate manual sau automat.
- Să dispună de un motor de corelare care să permită identificarea elementelor comune din două sau mai multe evenimente aparent fără nici o legătură.

- Să colecteze, normalizeze şi să filtreze cel puţin 250 EPS (evenimente pe secundă).

- Să permită accesul la aplicaţie unui număr minim de 10 utilizatori prin intermediul unei interfeţe de tip web.
- Să ofere posibilitatea identificării incidentelor de securitate IT în timp real, pe baza regulilor prestabilite şi să permită prioritizarea evenimentelor în funcţie de importanţă.

- Să aibă o consolă de management cu o interfaţă intuitivă în scopul administrării sistemului.
- Monitorizarea evenimentelor să se facă cu ajutorul tablourilor de bord.
- Să existe posibilitatea utilizării tablourilor de bord grafice predefinite, precum şi posibilitatea personalizării acestora pentru o imagine cât mai elocventă asupra nivelului de securitate; acestea trebuie să poate afişa informaţiile în timp real.
- Motorul de căutare să micsoreze timpul de răspuns al regăsirii evenimentelor.

- Să fie disponibile capabilităţi de escaladare privitoare la alerte, precum şi configurarea acţiunilor pentru evenimente critice.

- Să fie disponibilă o consola de management pentru alerte ce permite vizualizarea evenimentelor în timp real şi într-un format comun, inteligibil.

- Să dispună de template-uri de notificare configurabile.

- Să ofere funcţionalităţi încorporate de management a cazurilor precum: cazuri şi fluxuri pentru verificarea conformităţii, adnotări, ataşamente, mesaje text, pager sau e-mail, alerte SNMP.
- Să dispună de un pachet de reguli de corelare şi alertare care să trateze cele mai întâlnite ameninţări asupra securităţii unei reţele, precum şi posibilitatea modificării acestor reguli în funcţie de cerinţe/necesităţi.

- Să permită exportarea rapoartelor în următoarele formate: PDF, XML, HTML.
- Să dispună de posibilitatea filtrării log-urilor bazată pe orice criteriu legat de informaţiile conţinute în log-urile respective.
- Să pună la dispoziţie un API (Application Programming Interface) pentru a permite normalizarea şi managementul log-urilor provenite de la surse de evenimente/aplicaţii proprietare sau care nu sunt suportate în mod implicit.
- Să fie capabilă să colecteze evenimente de la minim următoarele surse:
    - Anti-Virus/ AntiSpam

- Aplicaţii de tip ERP
- Aplicaţii de securitate
- Aplicaţii de securitate a conţinutului
- Aplicaţii de securitate a bazelor de date
- Baze de date
- Prevenire a scurgerii de date
- Securitatea datelor
- Firewall
- IDS/IPS
- Managementul identităţii
- Management de operatiuni IT
- Filtrare de mail
- Server de mail
- Management de reţea

- Monitorizare reţea
- Analiza a traficului de retea
- Managment al traficului de reţea
- Sisteme de operare
- Router
- Switch
- Virtualizare
- VPN
- Web Cache
- Filtrare Web

3.1.12. Confidenţialitatea datelor

Sistemul propus asigură confidenţialitatea informaţiilor necesare pentru operare, accesul la interfaţa de administrare făcându-se pe baza de nume de utilizator şi parolă. Totodată sistemul asigură integritatea datelor transmise, actualizate, vizualizate sau înregistrate.

Toate informaţiile despre utilizatori vor fi confidenţiale în limitele stabilite prin politica de securitate. Aceste limite sunt stabilite în funcţie de rolul pe care îl are fiecare utilizator în cadrul sistemului informatic propus. De asemenea se vor respecta legislaţia şi reglementările internaţionale privind protecţia intimităţii şi a datelor personale.

Prin intermediul unei componente specializate de administrare, persoanele acreditate (administratori de sistem) vor putea restricţiona accesul în anumite zone ale sistemului informatic, la anumite documente sau date, după cum va fi necesar, pentru a acorda drepturi doar anumitor utilizatori sau grupuri de utilizatori.

Cu ajutorul acestei politici, utilizatorii vor putea vizualiza, modifica sau adăuga documente/înregistrări numai în limita drepturilor de acces asociate, asigurându-se confidenţialitatea datelor.

Din motive de securitate parolele utilizatorilor nu vor fi păstrate în baza de date, ci se vor păstra criptate într-un director LDAP centralizat.

Cerinţe pentru componenta de mascare dinamică a datelor

Aceasta componenta reprezinta o platforma de mascare a datelor care furnizeaza capabilitati in timp real de prevenire accesul utilizatorilor neautorizati la informatiile sensibile. Solutia permite organizatiilor IT sa aplice reguli flexibile, sofisticate de mascare a datelor, pe baza nivelului de autentificare a utilizatorului. Acest modul mascheaza dinamic informatia sensibila si blocheaza, auditeaza si alerteaza asupra utilizatorilor finali, personalului IT si echipelor externalizate care acceseaza informatia sensibila, asigurand in acelasi timp o conformitate rapida cu regulamentele interne.

Avantaje:

- Diminueaza dramatic riscul accesului neautorizat la date sensibile
- Permite personalizarea facila pentru cerinte variate, regulamentare si de business
- Protejeaza informatiile sensibile si cele personale, calificand in acest mod initiativele de externalizare si de cloud

Functionalitati:

- Securizeaza si monitorizeaza in timp real a bazele de date, pe baza de politici
- Aplica o multitudine de actiuni de securitate in timp real: mascare dinamica, amestecare, ascundere, blocare, auditare si alertare asupra accesului neautorizat
- Restrictioneaza utilizatorii finali si personalul IT la nivel de ecran, tabela, coloana, linie si celula
- Este usor de instalat si configurat
    - Aplica rapid algoritmi de mascare a datelor in orice format, referitor la orice informatie sensibila
    - Accelereaza termenele de proiecte, prin seturi predefinite de reguli pentru aplicatiile de business comune
- Prezinta o solutie unitara, scalabila
    - Poate fi scalata pentru a suporta sute de baze de date printr-o singura instalare

    - Permite restrictionarea accesului rapid si consistent asupra uneltelor, aplicatiilor si mediilor de lucru prin definirea unica de politici de mascare a datelor si aplicarea lor multipla
- Este o solutie versatila si neintruziva pentru aplicatii sau baze de date:
    - Previne accesul neautorizat la aplicatii personalizate, aplicatii impachetate, depozite de date si instante operationale, fara a impacta datele de substrat sau performanta aplicatiilor
    - Suporta medii virtualizate si de cloud
- Se integreaza deplin cu platformele de autentificare
    - Limiteaza livrarea informatiilor critice de business numai catre persoanele autorizate, in baza unor reguli de securitate aplicate selectiv
    - Aliniaza platformele de management al identitatii pentru a accelera timpul de implementare si creste amprenta de securitate asupra aplicatiilor si uneltelor
- Mascarea si blocarea datelor in timp real
    - Mentine functionalitatea, consitenta si integritatea datelor mascate in mediile enterprise complexe, prin sincronizarea valorilor datelor in interiorul si peste linii si tabele
    - Selecteaza automat tehnica aplicabila pe baza de politici, cum ar fi amestecarea numelor sau ascunderea informatiilor relative la carduri de credit si salarii

Modulul va masca dinamic pe baza unor reguli flexibile si complexe datele sensibile din bazele de date.

Sunt urmarite urmatoarele avantaje:

- Diminuarea riscului accesului neautorizat la date sensibile
- Personalizarea politicilor de mascare pentru cerinte variate
    - Trebuie mascate datele pe baza utilizatorului
    - Pe baza IP-ului hostului utiizatorului
    - Pe baza momentului cand se produce cererea SQL

- Modulul nu va depinde de aplicatie, trebuie mascate datele indiferent de modul de conectare la baza de date (prin aplicatie, sau direct prin SQL Plus, prin Excel, sau alt tip de client)

Functionalitati:

- Modulul trebuie sa poata aplica urmatoarele actiuni de securitate in timp real: mascare dinamica, amestecare, ascundere, blocare, auditare si alertare asupra accesului neautorizat
- Modulul trebuie sa poata restrictiona utilizatorii finali si personalul IT la nivel de ecran, tabela, coloana, linie si celula
- Modulul trebuie sa fie usor de instalat si configurat:
    - Aplicare rapida de algoritmi de mascare a datelor in orice format, referitor la orice informatie sensibila
    - Accelereaza termenele de proiecte, prin seturi predefinite de reguli pentru aplicatiile de business comune
- Modulul trebuie sa fie scalabil
    - Sa poata fi scalata pentru a suporta minim 100 de baze de date printr-o singura instalare
    - Sa permita restrictionarea accesului rapid si consistent asupra uneltelor, aplicatiilor si mediilor de lucru prin definirea unica de politici de mascare a datelor si aplicarea lor multipla
    - Modulul trebuie sa suporte cel putin 3,000 de fraze SQL pe secunda
- Modulul trebuie sa fie neintruziv pentru aplicatii sau baze de date:
    - Nu trebuie sa aiba impact asupra bazelor de date
    - Trebuie instalat pe o masina separata cu rol de proxy, astfel incat sa nu existe consumuri suplimentare pentru baza de date
- Modulul trebuie sa permita segregarea atributiilor.
    - Administratorii de baze de date sa nu aiba acces la modulul de mascare a datelor

    - Administratorii modulului de mascare sa nu aiba acces la baza de date (cel putin nu cu drepturi de DBA)

- Modulul trebuie sa permita executii multi-threading
    - In conformitate cu numarul conexiunilor si cu volumul frazelor SQL, modulul trebuie sa fie capabil sa deschida multiple thread-uri pentru a asigura scalabilitatea
- Modulul trebuie sa ofere mascarea si blocarea datelor in timp real
    - Mascarea trebuie realizata in timp real
    - Tabelele relationate trebuie sa poata pastra consistenta datelor prin sincronizarea valorilor mascate
- Failover – Modulul trebuie sa poata fi instalat pe un cluster de tip HA si sa beneficieze de functionalitatile native de failover ale clusterului.
- Modulul trebuie sa suporte cel putin baze de date Oracle, SQL Server si DB2.

4. CERINŢE PRIVIND SOLUŢIA TEHNICĂ

Soluţia tehnică integrată care va permite funcţionarea sistemului informatic propus se referă la componentele constituente care, împreună, contribuie la realizarea serviciilor publice online pentru furnizorii de servicii medicale, pe de o parte, dar şi la eficientizarea activităţilor de la nivelul CNAS şi al CJAS care asigură furnizarea acestor servicii.

Sistemele informatice propuse au ca scop principal furnizarea de servicii online specifice administraţiei publice centrale în beneficiul cetăţenilor, al mediul de afaceri şi al altor instituţii şi organizaţii la nivel local şi/sau la nivel central. În plus, aceste sisteme permit desfăşurarea în mod eficient şi în anumite cazuri automat a activităţilor specifice interfeţei dintre organismele administraţiei centrale şi cetăţeni/mediul de afaceri. Activitatea specifică a instituţiei şi serviciile publice oferite de către aceasta pentru cetăţeni şi mediul de afaceri, în cazul de faţă fiind vorba de asiguraţi şi furnizorii de servicii medicale, vor fi puse la dispoziţie prin intermediul componentei aplicative sub forma de servicii web publice online. Acest lucru va asigura apropierea CNAS de cetăţeni şi mediul de afaceri şi va oferi acestora servicii sofisticate, moderne şi electronice.

4.1. Cerinţe generale

Soluţia implementată trebuie să asigure inter-operabilitatea sistemului propus cu alte sisteme existente în Platforma Informatică a Asigurărilor de Sănătate (PIAS).

Din perspectiva colaborării inter-instituţionale, comunicarea şi colaborarea joacă un rol esenţial. Sistemele informatice propuse sunt instrumente moderne, actuale, care asigură legătura directă între CNAS şi publicul larg şi conduc către o mai bună transparenţă şi eficienţă a activităţilor efectuate pentru îndeplinirea obiectivelor proprii prin punerea la dispoziţie a unor servicii accesibile online.

Obiectivul principal al *Sistemului* este de a verifica şi valida corect, coerent şi la timp documentele medicale gestionate, fapt ce va conduce la creşterea transparenţei şi a prestigiului instituţiei CNAS, precum şi la dezvoltarea unor instrumente importante de informare şi suport operaţional pentru utilizatori.

Sistemele informatice propuse vor respecta atât politicile şi reglementările interne privind tehnologia informaţiei cât şi legislaţia în vigoare privind protecţia datelor cu caracter personal, protecţia informaţiilor clasificate şi alte acte normative care referă tehnologia informaţiei.

Interfaţa utilizator a sistemului va fi intuitivă (facilă), informativă, fiabilă, atractivă şi stabilă, ea fiind accesibilă doar utilizatorilor interni, angajaţi ai CNAS/CJAS. Interfaţa utilizator poate fi accesată utilizând ultimele versiuni ale *browser*-elor Mozilla Firefox / Internet Explorer / Google Chrome şi este optimizată pentru o rezoluţie de minim 1024x768. Prin utilizarea şabloanelor de afişare se va obţine o interfaţă grafică unitară. Aceasta va fi realizată conform ultimelor versiuni ale standardelor HTML, CSS, XML.

Furnizorii de servicii vor avea la dispoziţie servicii web pentru transmiterea şi validarea documentelor medicale în format electronic care vor fi accesate prin intermediul unor aplicaţii informatice proprii. Validarea se face în baza de date a noului sistem informatic pentru documentele medicale gestionate de acesta, dar şi în celelalte sistemele informatice ale PIAS prin intermediul unor servicii de inter-operabilitate.

Se vor pune la dispoziţia utilizatorilor interni de la CNAS/CJAS-uri toate instrumentele necesare administrării nomenclatoarelor proprii sistemului, precum şi a regulilor de validare. De asemenea se vor pune la dispoziţia furnizorilor care nu utilizează aplicaţii informatice proprii, un set de aplicaţii care să permită accesul la serviciile expuse de noul sistem, prin extinderea funcţională a aplicaţiilor existente, utilizate de furnizorii de servicii medicale pentru conectarea la PIAS, conform prevederilor din secţiunea 3.1.6..

Accesul la serviciile web pentru furnizori sau la aplicaţia specifică CNAS se va face utilizând aceleaşi mijloace de securizare a accesului ca şi pentru celelalte sisteme constituente ale PIAS, pentru a evita necesitatea creării unor seturi paralele de conturi de utilizatori pentru utilizatorii externi sau interni şi a reduce astfel dificultăţile de accesare a unor sisteme informatice diferite.

4.2. Cerinţe de interoperabilitate

Pentru a asigura interoperabilitatea şi integrarea între noul sistem şi sistemele existente în cadrul platformei PIAS a CNAS, precum şi cu aplicaţiile de raportare ale furnizorilor de servicii medicale, dar şi cu alte sisteme informatice din Uniunea Europeană, este necesar ca noul sistem, sau componentele constitutive ale acetuia, să adere la standarde internaţionale comun acceptate şi larg utilizate.

Una dintre cele mai importante iniţiative la nivel internaţional în domeniul standardizării sistemelor informatice din domeniul sănătăţii este IHE (*Integrating the Healthcare Enterprise*) – o organizaţie internaţională a profesioniştilor şi instituţiilor medicale cu scopul

declarat de a îmbunătăţi şi accelera utilizarea sistemelor informatice facilitând colectarea şi accesul la informaţiile medicale.

IHE nu reprezintă un nou standard, ci promovează utilizarea standardelor deja încetăţenite, cum ar fi HL7 sau DICOM pentru a îndeplini la nivel optim anumite necesităţi în procesul de îngrijire medicală a pacienţilor. Sistemele informatice dezvoltate în conformitate cu IHE comunică mai uşor unele cu celelalte, sunt mai uşor de implementat şi permit furnizorilor de servicii medicale să colecteze şi să utilizeze informaţiile mai eficient. IHE susţine îmbunătăţirea calităţii, eficienţei şi siguranţei actului medical prin facilitarea accesului la informaţii relevante atât pentru pacienţi, cât şi pentru personalul medical, dar şi pentru factorii de decizie în domeniul sanitar.

IHE defineşte un set de „profile" care oferă un limbaj comun care permite producătorilor şi consumatorilor de soluţii informatice medicale să înţeleagă nevoile şi capabilităţile de integrare ale produselor sau soluţiilor, oferind direcţii bine definite în implementarea precum şi standarde de comunicare internaţional recunoscute şi cu un grad ridicat de adopţie.

Pentru asigurarea dezideratelor de mai sus, următoarele profile de integrare IHE vor fi considerate pentru implementarea în cadrul proiectului:

1 PIX - Patient Identifier Cross-referencing (HL7v3)

   Interoperabilitatea sistemelor informatice medicale începe cu identificarea pacientului. Profilul de integrare PIX este definit pentru a rezolva identificarea unică a pacientului în cadrul mai multor sisteme informatice interconectate, bazat pe standardul HL7 v3.

   PIX permite tranzacţii care gestionează crearea, actualizarea sau unificarea informaţiilor despre pacienţi în cadrul unui sistem informatic local, precum şi tranzacţii de interogare inter-sisteme a listei de identificatori ai pacienţilor, folosind un identificator local cunoscut.

2 PDQ - Patient Demographics Query (HL7v3)

   Profilul de integrare PDQ este definit pentru a facilita obţinerea identificatorilor pacienţilor care îndeplinesc un set dat de criterii demografice, bazat pe standardul HL7 v3.

   PDQ permite tranzacţii de interogare a mai multor sisteme informatice pentru a obţine o listă de identificatori de pacienţi în baza unui set dat de criterii demografice.

3 XDS.b - Cross-Enterprise Document Sharing

Profilul de integrare XDS.b este o specificaţie pentru schimbul de documente clinice între sisteme informatice medicale. Schimbul de documente este gestionat prin intermediul unor baze de date şi registre cu documente care permit crearea unui dosar ce conţine înregistrările unui pacient sau a unui sau index cu referinţe către aceste înregistrări.

XDS.b permite gestionarea înregistrărilor medicale electronice (EHR – Electronic Health Records) prin facilitarea înregistrării, distribuţiei şi accesului la înregistrărilor medicale ale pacienţilor la nivel inter-instituţional. În acelaşi timp permite definirea de standarde pentru partajarea documentelor clinice între instituţiile medicale, începând cu cabinete medical individuale şi până la policlinici sau spitale şi chiar instituţii guvernamentale de specialitate.

4 XDR - Cross-Enterprise Document Reliable Interchange

Profilul de integrare XDR permite implementarea unui sistem de mesagerie pentru schimbul de documente. Acest sistem permite schimbul direct de documente între sisteme EHR sau alte sistem informatice medicale, chiar şi în absenţa unei baze de date centrale sau a unui registru central.

XDR oferă un mecanism automatic şi sigur pentru transferul documentelor medicale şi a meta-datelor unui pacient între mai multe sistem EHR, chiar şi în absenţa unei infrastructuri XDS (vezi mai sus).

5 XUA - Cross-Enterprise User Assertion

Profilul de integrare XUA oferă mecanisme de comunicare a cererilor de validare a identităţii unei entităţi autorizate (utilizator, aplicaţie, sistem...) în cadrul unor tranzacţii inter-sisteme. Pentru a permite identificarea şi a oferi mijloace de asigurare a responsabilităţii este necesară crearea unui director de entităţi autorizate conţinând toate atributele necesare pentru identificarea ulterioară.

XUA permite solicitarea şi obţinerea unei acreditări de la autoritate de certificare care este adăugat la mesajul transmis în cadrul tranzacţiei IHE, care va fi verificată la primirea cererii prin mijloace specifice.

6 ATNA - Audit Trail and Node Authentication

Profilul de integrare ATNA stabileşte măsuri de securitate care conferă confidenţialitatea informaţiilor despre pacient, integritatea datelor, precum şi responsabilitatea utilizatorilor. Următoarele mecanisme sunt specificate de acest profil: autentificarea utilizatorilor (necesară doar la nivel local), autentificarea

conexiunii (prin utilizarea certificatelor digitale) şi piste de audit (care permit monitorizarea activităţii utilizatorilor).

ATNA permite implementarea unor politici de confidenţialitate prin autentificarea şi autorizarea utilizatorilor şi prin înregistrarea activităţilor realizate de aceştia.

7 SVS - Sharing Value Sets

Profilul de integrare SVS oferă mijloace prin care sistemele informatice medicale care produc date medicale clinice sau date administrative, cum ar fi echipamente de laborator, sisteme de raportarea pentru medici, sau chiar sisteme naţionale ce gestionează înregistrări medicale să poată obţine şi să utilizeze un set comun, gestionat central de nomenclatoare.

SVS permite utilizarea şi sincronizarea unui set comun şi uniform de nomenclatoare între sistemele informatice medicale, facilitând încărcarea automată a seturilor de valori ale nomenclatoarelor, reducând astfel operaţiile manuale şi reducând erorile de operare.

8 CT – Consistent Time

Profilul de integrare CT defineşte mecanisme de sincronizare a unităţilor de măsură a timpului între diferiţi utilizatori sau calculatoare, dat fiind că informaţiile medicale sunt sensibil dependente de momentul în care s-au întâmplat sau au fost colectate.

CT permite sincronizarea ceasurilor interne ale diferitelor calculatoare conectate într-o reţea, pe baza unor servere de timp.

4.3. Prevederi de securitate

Componentele sistemului propus trebuie să fie protejate împotriva încercărilor deliberate sau accidentale de acces neautorizat la datele pe care acesta le înmagazinează. Astfel, sistemul informatic trebuie să asigure:

- Securitatea datelor printr-un sistem de limitări ale accesului la aplicaţie bazat pe drepturi şi parole, defalcat pe mai multe niveluri. Drepturile de acces ale utilizatorilor vor putea fi configurate de administratorii sistemului din interfaţa aplicaţiei
- Autentificarea utilizatorilor externi trebuie să fie permisa de la orice staţie de lucru conectata la Internet, atât timp cât aceasta nu se află într-o zonă pentru care accesul a fost restricţionat din motive de securitate

- Împiedicarea utilizatorilor de a se conecta la sistem dacă acesta este în incapacitate temporară de a asigura securitatea datelor sau există suspiciuni că mecanismele de protecţie au fost compromise
- Închiderea automata a sesiunilor de lucru ale utilizatorilor în caz de inactivitate pe o anumita durata predeterminata de timp, pentru a proteja dezvăluirea accidentală a informaţiilor către alte persoane care nu sunt autorizate să le primească
- Jurnalizarea operaţiilor zilnice la nivelul aplicaţiei, individual pentru fiecare utilizator cu drept de acces la modificarea înregistrărilor, cu marcarea orei la care a fost executata fiecare operaţie precum şi a identităţii utilizatorului care a iniţiat-o
- Stabilirea unei sesiuni de lucru va consta în operaţiunea de autentificare (*login*) a utilizatorului curent în aplicaţie; autentificarea unică a utilizatorilor şi autorizarea acestora se vor realiza o singura data pe sesiune, prin intermediul rolurilor şi privilegiilor; autentificarea şi asocierea permisiunilor/privilegiilor funcţie de rolurile prestabilite vor fi realizate cu ajutorul unor instrumente specializate
- Securitate de perimetru - prin implementarea unui sistem de tip *firewall* care va proteja reţeaua interna de trafic nedorit.

Soluţia de securitate proiectată trebuie să asigure confidenţialitatea transferului de informaţii. Informaţia dintr-un astfel de sistem trebuie protejată împotriva ameninţărilor în orice situaţie, fie când este stocată, fie când este transportată.

Instrumentele proiectate pentru asigurarea confidenţialităţii datelor trebuie să asigure accesul utilizatorilor sistemului prin intermediul protocolului securizat HTTPS, folosind certificate digitale calificate, pentru a elimina posibilele încercări de interceptare a datelor când sunt transmise prin mediile de comunicaţie.

La nivelul fizic al sistemului vor trebui implementate un set de mecanisme astfel incat sa existe un nivel de securizare a suportului fizic informational. Astfel, toate echipamentele livrate in cadrul proiectului vor trebui instalate intrun ansamblu unitar de rack-are. Ansamblul va fi format din cabineti de tip rack de dimensiunea standard 42U (unitati de rack-are) si vor permite instalarea unui sistem de control acces fizic, monitorizare si alertare, astfel incat accesul personalului sa fie controlat si monitorizat. Solutia propusa trebuie sa fie una unitara, cu un management centralizat, care sa permita definirea granulara a drepturilor de acces fizic la echipamente de la nivelul ansamblului pana la nivelul fiecarui echipament rack.

Echipamentele rack vor fi prevazute cu sisteme de blocare automata a deschiderii usilor, care sa fie actionat prin intermediul unei autentificari cu card perimetral (RFID). In acest sens, sistemul va dispune de toate elementele necesare functionarii cat si de interfata de management centralizata ce va functiona peste retea TCP/IP.

De asemenea, avand in vedere natura datelor vehiculate si cele mai bune practici in domeniu, pentru solutia de backup se va dimenesiona numarul de casete astfel incat sa poata exista o copie de siguranata a bazei de date care sa poata fi scoasa din locatie lunar si depozitata in cadrul sediului CNAS intro incinta securizata. Pentru a se putea audita actiunea de scoatere si depozitare a mediilor magnetice ce contin informatii relevante ale bazei de date, fiecare caseta de banda va fi marcata cu un tag RFID (sub forma de eticheta). La iesirea din camera tehnica unde se vor instala echipamentele va fi instalat un cititor de proximitate RFID care va permite identificarea casetelor ce pleaca spre sediul CNAS. La sediul CNAS se va instala un dulap securizat cu acces securizat unde vor fi depozitate casetele scoase din centrul de date. Ofertantii vor propune un dulap securizat potrivit cu nevoile proiectului si in concordanta cu acesta. De asemenea, dulapul va fi dotat cu un cititor de proximitate RFID care va permite auditarea actiunilor de depozitare a casetelor.

4.4. Cerinţe suplimentare

4.4.1. Disponibilitate ridicată

Sistemul informatic propus trebuie să fie disponibil online în permanenţă 24 de ore, 7 zile pe săptămână. Orice întrerupere accidentală va fi tratata cu maxima urgenta, iar opririle programate pentru mentenanţă hardware şi software necesare vor trebui să fie anunţate în prealabil şi să se încadreze în afara intervalului orar 6:00 - 22:00.

În situaţiile de supra-încărcare a sistemului, sau pe perioadele de inactivitate a sistemelor PIAS care furnizează date privind solicitarea utilizatorului, procesarea cererilor se va putea face cu un decalaj de timp, în intervalul de timp neprioritar, situaţie care trebuie adusă la cunoştinţa solicitantului, împreună cu estimarea timpului în care se va transmite răspunsul sistemului.

Operaţiunile de realizare a copiilor de siguranţă trebuie incluse tot în intervalul de timp neprioritar. Salvarea datelor se va realiza periodic utilizând medii de stocare specifice.

Datele care vor fi salvate sunt cele din baza de date de utilizatori şi configuraţiile acestora.

În cazul unui incident se vor putea restaura rapid datele de pe unitatea de siguranţă pentru oricare din serverele de baze de date.

4.4.2. Administrare şi monitorizare

Sistemele informatice propuse trebuie să pună la dispoziţia administratorilor o componentă pentru realizarea funcţionalităţilor necesare administrării sistemului precum şi pentru monitorizarea funcţionării acestuia în vederea urmăririi îndeplinirii obiectivelor de performanţă şi disponibilitate.

Aceasta componentă trebuie să răspundă următoarelor cerinţe generale:

- Definirea şi documentarea procedurilor şi proceselor necesare pentru operarea soluţiei
- Minimum următoarele cerinţe vor fi acoperite de aceste proceduri şi definiţii de procese:
    - Operarea şi administrarea soluţiei în mod proactiv şi eficient
    - Monitorizarea permanenta a funcţionării sistemului cu alertarea anomaliilor – erori sau avertizări legate de funcţionalitate
    - Readucerea sistemului în parametrii normali de operare
    - Persoane cu nivel mediu de cunoştinţe IT şi a produselor soluţiei să poată aplica procedurile definite.
- Toate componentele soluţiei vor înregistra principalele evenimente de succes de eroare în jurnale specializate care îndeplinesc următoarele cerinţe:
    - pot fi securizate pentru a limita accesul la aceste informaţii
    - permit consultarea lor directa de către un operator uman
    - permit interpretarea prin metode programatice – sunt organizate intr-un mod consistent şi structura este documentata.
- Toate componentele hardware şi software ale soluţiei respectă cerinţele de suportabilitate emise de producător.

De asemenea sistemul va trebui sa includa o componenta de mesagerie electronica sigura ce va fi utilizata de administratorii sistemului in vederea schimbului de informatii confidentiale. Intreaga solutie va fi instalata la client (on premises), nefiind permise solutii de tip „as-a-service”. Infrastructura hardware aferenta functionarii componentei va trebui sa fie asigurata

de catre ofertant. Aceasta componenta va utiliza certificate digitale instalate pe dispozitivele mobile ale administratorilor, oferind urmatoarele functionalitati:

1. Transmiterea de mesaje text (tip SMS) criptate peste conexiunile de date (GPRS, 3G, 4G şi Wi-Fi)
2. Conexiune criptată SSL între utilizatorii solutiei
3. Pentru fiecare mesaj transmis este necesara generarea unei chei de sesiune unice
4. Este necesara utilizarea minim a algoritmului AES cu chei de cel puțin 256 de biți
5. Trebuie sa utilizeze chei criptografice RSA cu lungimea de minim 2048 de biți
6. La fiecare conectare este necesara verificarea starii certificatului digital a utilizatorului
7. Trebuie sa permita configurarea duratei de viață a mesajelor cu cel puțin următoarele opțiuni: afișare o singură dată, pentru citire, și durată de viață nelimitată
8. Mesajele cu durată de viață nelimitată trebuie păstrate permanent exclusiv în formă criptată pe dispozitiv
9. Nu trebuie sa permita copierea de text din mesajele transmise sau recepționate
10. Asigură configurarea unei liste de contacte independentă de agenda telefonică a dispozitivului
11. Numai expeditorul și destinatarul trebuie sa poata avea acces la conținutul mesajelor, acestea nefiind disponibile serverelor de comunicații prin care trece transmisia de date.
12. Solutia trebuie să asigure inclusiv criptarea comunicatiilor de voce intre utilizatorii sistemului.

4.4.3. Autentificare şi autorizare

Sistemele informatice propuse trebuie să pună la dispoziţia administratorilor o componentă pentru controlul accesului utilizatorilor interni sau externi la funcţiile aplicative ale sistemului informatic, pe baza drepturilor de acces specifice pentru fiecare categorie sau grup de utilizatori.

Este necesar ca soluţia tehnică să implementeze cel puţin următoarele funcţionalităţi:

- Posibilitatea restricţionării accesului utilizatorilor privilegiaţi la datele manipulate de aplicaţiile de business, prin segregarea responsabilităţii
- Soluţia va permite autentificarea furnizorilor de servicii medicale şi farmaceutice pe baza certificatelor digitale calificate
- Certificatele digitale calificate ale furnizorilor trebuie validate la momentul accesului în sistem prin protocolul OCSP

- Autentificarea persoanelor asigurate la informatiile din dosar pe baza cardurilor electronice de asigurat (CEAS) din momentul introducerii acestora sau prin sistemul de autentificare utilizand certificatele digitale emise de autoritatea de certificare existenta pentru emiterea certificatelor digitale aferente CEAS pentru terminale mobile inteligente (smartphone, tabletă)
- Certificatele digitale utilizate in cadrul solutiei pentru autentificare prin intermediul tehnologiei mobile vor fi emise de catre Autoritatea de certificare existenta la sediul Beneficiarului, si utilizata pentru emiterea Cardurilor Electronice de Asigurari de Sanatate.
- In vederea emiterii acestor certificate digitale, utilizatorilor finali li se va pune la dispozitie o aplicatie (prin download direct sau prin intermediul Google Play/Apple App Store) care se va putea instala pe smartphone-uri şi/sau tablete. Rolul acestei aplicatii este de a asigura generarea cheilor criptografice si a cererii de certificat (standard PKCS#10) direct la nivelul echipamentului pe care ruleaza si de a transmite aceasta cerere Autoritatii de certificare. Prin urmare, aceasta aplicatie trebuie sa permita, prin configurare, integrarea cu aceasta Autoritate de certificare (prin adresa publica a autoritatii si un cod de emitere a certificatului). Subsecvent aprobarii cererii (si implicit emiterii certificatului) aplicatia trebuie sa permita stocarea certificatului in containerul dedicat de la nivelul smartphone-ului/tabletei.
- Aplicatia de inrolare trebuie sa fie capabila sa utilizeze in acest proces atat containere criptografice software (de tip PKCS#12), cat si containere criptografice hardware (de tip smartcard/token criptografic cu contact – PKCS#11 sau contactless – de ex. NFC).
- Aplicatia trebuie sa fie personalizabila din punct de vedere grafic, in sensul afisarii elementelor grafice imprimate pe Cardul Electronic de Asigurari de Sanatate sau alte elemente de identitate vizuala ale autoritatii contractante.
- Avand in vedere importanta proiectului, manipularea de date cu caracter medical, utilizatorii tinta si aria de aplicabilitate este obligatoriu ca interfata aplicatiilor sa fie disponibila minim in limba romana

4.4.4. Audit şi control

Sistemele informatice propuse trebuie să pună la dispoziţia administratorilor o componentă care va îndeplini atât funcţiile de audit informatic cât şi funcţiile de control al accesului la informaţii.

Soluţia de audit şi control va îndeplini următoarele cerinţe generale:

- Se va păstra un istoric de tip log al activităţii utilizatorilor aplicaţiei
- Va permite includerea informaţiilor despre momentul în care au fost modificate anumite seturi de date de către utilizatori

5. Cerințe de servicii

5.1. Cerințe de instruire

În utilizarea sistemului informatic propus se identifică 2 categorii principale de utilizatori:

- furnizorii de servicii medicale
- personalul Caselor de Asigurări de Sănătate.

În cadrul proiectului se va asigura instruirea diferenţiată a următoarelor categorii de beneficiari astfel:

- administratorii şi personalul de întreţinere a sistemului informatic de la CNAS. Pentru această categorie proiectul va asigura instruirea a 5 persoane de la CNAS;
- 45 de persoane din personalul CJAS, cu accent pe modalităţile de răspuns la solicitările şi sesizările furnizorilor, precum şi pe valorificarea potenţialului datelor colectate. De asemenea, personalul CJAS şi echipele desemnate de CNAS vor fi instruite privind modificările şi suplimentările de funcţionalităţi determinate prin implementarea *Sistemului* alături de celelalte sisteme ale PIAS. O echipa desemnată de CNAS de 5 persoane va fi instruită de furnizorul soluţiei informatice pentru asigurarea Nivelului 1 de suport pentru utilizatorii sistemului propus.

Prin instruire se va asigura realizarea cel puţin a următoarelor obiective:

- cunoaşterea sistemului integrat în ansamblul său
- învăţarea modului de operare a funcţionalităţilor sistemului propus
- învăţarea modului de rezolvare a problemelor curente de folosire a sistemelor componente ale PIAS relaţionate cu *Sistemul*
- înţelegerea implicaţiilor sistemului propus şi a avantajelor acestuia asupra modului de informare şi de rezolvare a problemelor curente ale asiguraţilor şi furnizorilor de servicii.

Sesiunile de instruire vor fi realizate de furnizorul soluţiei informatice. De asemenea, furnizorul soluţiei informatice va elabora şi pune la dispoziţia beneficiarului manuale de utilizare şi suport de curs în limba română, pentru toate categoriile de utilizatori ai sistemului.

La terminarea cursului, cursanţii din categoriile administrator de sistem şi personal CJAS/CNAS vor primi de la furnizor certificate de instruire individuale. Certificarea se va face diferenţiat pentru cele două categorii.

Celelalte categorii de utilizatori vor beneficia de materiale de prezentare şi de instruire individuală (*self-training*) în format electronic, furnizorii de servicii medicale vor avea la dispoziţie materiale complete de instruire electronică pentru operarea funcţionalităţilor puse la dispoziţie de *Proiect*, care complementează funcţionalităţile aplicaţiilor de raportare utilizate în mod curent pentru raportarea în SIUI şi conexiunea la sistemele din platforma PIAS.

Furnizorul soluţiei va face instruirea utilizatorilor sistemului prin livrarea de documentaţie şi organizarea de cursuri de instruire la nivelul CNAS şi al fiecarui CJAS. Logistica necesara instruirii (sali, calculatoare, videoproiector, etc) va fi asigurata de catre CNAS.

Instruirea utilizatorilor sistemului se va efectua la finalizarea implementării proiectului pe baza manualelor de utilizare în limba română, care vor fi disponibile în format fizic şi electronic. Se vor realiza manuale distincte în funcţie de tipurile de utilizatori ai sistemului. Aceste materiale vor fi puse la dispoziţia beneficiarului înainte de punerea în producţie a sistemului informatic propus.

Instruirea personalului care va utiliza/administra sistemul propus va fi realizată în cadrul a două categorii de cursuri specifice organizate, în funcţie de tipul de utilizatori (personal CNAS/CJAS şi administratori) şi se va face pe modelul de tipul “train the trainers” astfel încât fiecare CJAS să poată instrui un număr corespunzător de reprezentanţi. Astfel sunt propuse următoarele:

- instruirea dedicată a utilizatorilor sistemului propus şi a funcţionalităţilor sistemelor platformei PIAS determinate de implementarea sistemului propus
- instruirea dedicată a persoanelor care vor asigura administrarea sistemului propus.

Limba folosită în activităţile de instruire este limba româna.

La sfârşitul fiecărei sesiuni de instruire se vor elabora documentele:

- Prezenţa la curs
- Raport de şcolarizare realizat de către instructor

- Evaluare curs (se va completa de către cursanţi).

5.1.1. Platforma pentru instruirea utilizatorilor

Furnizorul va asigura accesul beneficiarului pe durata perioadei de implementare a proiectului la o platforma pentru instruire ce va fi utilizată pentru utilizatorii sistemului, in special personalul Caselor de Asigurari in vederea obtinerii celor mai bune rezultate privind utilizarea noilor servicii electronice intr-un mod cat mai rapid la costuri minime. Instruirea

urmăreşte o pregătire continuă a utilizatorilor astfel încât acestia sa poata utiliza la maxim

benificiile noului sistem. Platforma va pune la dispozitia cursantilor materialul de curs din cadrul proiectului in format electronic si va dispune de facilitati de invatare online interactive.

5.1.1.1. Caracteristici tehnice ale platformei de instruire on-line

Platforma de instruire va trebui sa aiba următoarele capabilităţi:

- Sa functioneze pe tehnologie de tip cloud
- Sa permita instruirea utilizând instrumente moderne de predare cum ar fi ecrane touch, table interactive cu rolul de ”tabla clasei”
- Sa permita instruirea participanţilor în aceeaşi locaţie cu instructorul (aceeaşi

încăpere)

- Sa faca posibilă instruirea participanţilor atunci când aceştia nu se află în aceeaşi încăpere cu instructorul. Participantii pot fi:
    - În alte încăperi aflată în altă locaţie
    - În faţa computerului personal conectaţi la internet
    - O parte în aceeaşi încăpere cu instructorul, o parte în faţa computerelor personale, o parte în alte încăperi (fără prezenţa fizică a instructorului)

- Instruirea participanţilor sa fie posibilă în mod similar indiferent de locul unde

aceştia sunt prezenţi în sensul că:

  - Toţi pot interacţiona cu instructorul (pe care în văd şi îl aud)

  - Toţi pot interacţiona unii cu alţii (se pot auzi şi vedea)

  - ”Tabla clasei”, ca instrument de instruire poate fi utilizată în mod

    comun (toţi văd acelaşi conţinut şi pot interacţiona cu acest conţinut)

  - Instruirea sa fie condusa de un instructor certificat de autoritatea care pune la dispozitie platforma.
- Platforma sa poata permite prin intermediul ”tablei clasei”:
  - Scrierea, desenarea

- Partajarea (utilizarea în comun) de filme, animaţii, imagini
- Transmiterea de imagini a unor obiecte fizice pe care instructorul doreşte să le utilizeze (documente, obiecte specifice instruirii de urgenţă cum ar fi: extinctoare, măşti de oxigen, în general echipament specific de intervenţie)

• Platforma sa permita autoinstruire prin parcurgerea cursurilor predate anterior.

Aceste cursuri sa poata fi parcurse de utilizatori fără a fi necesară prezenţa unui instructor.

• Platforma sa permita examinarea cunoştinţelor acumulate de participanţi prin:

- Teste în scris
- Teste orale

- Testele efectuate indiferent de locul unde se află cei testaţi

• Platforma sa fie uşor de utilizat şi sa necesite o curbă de învăţare foarte mică.

Astfel instrumentele ce se vor pune la dispoziţie trebuie să fie similare cu cele din lumea reală şi nu cu cele specifice IT. În acest sens, platforma:

- Sa poata fi operata cunoscand noţiuni cum ar fi tablă, cretă, burete virtuale care sa se utilizeze similar ca cele din lumea reală
- Sa aiba o intruziune minimă în activitatea de instruire – nu se solicită instructorilor şi participanţilor să facă operaţiuni tehnice complicate sau care

să interfereze cu activitatea de predare cum ar fi vorbitul la microfon, comutare de imagini, comutare de sunet

- Platforma sa poata genera salvari ale cursurilor atat in format PDF cat si intr-un format propriu care sa poata fi apoi revizuit de cursanti ca un film.
- Participantii la curs sa poata intra intr-o sesiune de training oricand, avand instantaneu acces la cursul deja predat pana la momentul accesarii lui de catre participant. Accesarea unei sesiuni deja initiate sa se faca cu un cod PIN care este generat initiatorului cursului si care ulterior va fi furnizat tuturor cursantilor.

5.1.1.2. Functionalitati

”Tabla clasei” va fi o aplicaţie din cadrul platformei optimizată pentru dispozitive touch (table, tablete, ecrane, telefoane etc) care va fi utilizată intensiv în cadrul sesiunilor de instruire fie că participanţii sunt în aceeaşi încăpere cu instructorul fie că o parte dintre aceştia se află la distanţă.

Funcţionalităţile pe care această aplicaţie trebuie să le ofere celor care o utilizează:

- **Scrierea**

Pe tablă sa se poata scrie utilizând un instrument tip ”pen” sau utilizând un obiect oarecare (chiar şi degetul). Scrierea sa se faca ca în modul clasic, prin atingerea suprafeţei.

Pentru situaţia în care sunt implicaţi participanţi din alte locaţii, scrierea se va transmite către aceştia pe măsură ce se produce. Aceştia trebuie sa vada pe dispozitivul propriu (altă tablă interactivă sau ecranul PC-ului propriu sau al tabletei) scrierea celorlalţi ca şi când aceştia ar scrie chiar pe dispozitivul lor – în mod continuu.

Oricare dintre participanţi sa poata utiliza dispozitivul propriu pentru a scrie pe ”tabla clasei” chiar şi în paralel cu ceilalţi. Scrisul trebuie sa se transmita către toţi în aceeaşi manieră continuă şi firească ca mai sus.

- **Ştergerea**

Pe tablă sa se poata şterge similar ca pe o tablă clasică. În acest sens sa fie utilizat un burete virtual – efectul de ştergere trebuie sa fie cât mai asemănător cu cel clasic.

Participanţii la o sesiune de instruire care vor avea acces la ”tabla clasei” prin dispozitive touch proprii vor vedea ştergerea ca şi când ar fi făcută pe dispozitivul lor. Mai mult, aceştia trebuie sa poata şterge chiar în paralel cu alţi participanţi sau cu instructorul.

- **Desenarea**

La tablă sa se poata desena similar cu desenarea pe o tablă clasică (prin atingere). Aplicaţia trebuie sa transmita desenarea similar cu scrierea şi ştergerea către toţi participanţii.

Aceştia sa poata desena în acelaşi timp, acţiunea fiecăruia fiind vizibilă, în mod continuu tuturor.

- **Schimbare instrumente de scris , de ştergere, a fundalului**

Aplicaţia sa poata permite alegerea instrumentului virtual de scris (exemplu creta, pix, pensulă) precum şi culoarea, grosimea sau tipul liniei produs de instrument.

Fundalul tablei trebuie sa fie customizabil – sa se poata modifica textura şi culoarea.

- **Paginarea**

În timpul unei sesiuni de instruire sa se poata crea ”pagini” noi. De asemenea sa se poata naviga la paginile anterioare sau oricare pagină, sa se poata modifica ordinea paginilor, sa se poata şterge pagini sau conţinutul unei pagini.

Trebuie sa existe o pagină specială al cărei conţinut sa poata fi afişat rapid şi oricând – acesta sa poata fi utilizată de exemplu pentru scrierea structurii cursului sau pentru orice alt scop

care implică necesitatea de a reveni la acel conţinut în mod repetat de-a lungul sesiunii de instruire. Pozitionarea acesteia trebuie sa fie intr-un mod neintruziv, usor de accesat.

- **Înregistrarea şi redarea cursului**

Sesiunea de predare (tot ceea ce se întâmplă pe tablă) trebuie sa poata fi înregistrată în mod propriu marcata temporal sub forma de video când sunt participanţi la distanţă.

Obiectivul înregistrării este de a permite ulterior:

- vizionarea cursului în scop didactic de către voluntari care nu au participat live la curs
- reutilizarea cursului de către instructor cu posibilitatea de a-l modifica şi de a crea un nou curs

- **Undo şi redo**

Aplicaţia va trebui sa permita functii de undo şi redo (anularea sau revenirea asupra unei comenzi şi efectului acesteia) în numar de 10.

Functiile de undo şi redo vor fi specifice unui dispozitiv în sensul că vor afecta numai comenzile efectuate de pe acel dispozitiv. Astfel un utilizator sa poata face undo sau redo numai comenzilor proprii în cazul în care acel utilizator foloseşte un dispozitiv touch propriu.

Instructorul sa poata face undo şi redo numai comenzilor date către dispozitivul utilizat de acesta (exemplu tablă inteligentă touch)

- **Recunoaştere scris**

Aplicaţia va trebui sa recunoasca scrisul de mână. Efectul recunoaşterii este vizibil către toţi participanţii indiferent cine îl iniţiază.

- **Căutare online**

Aplicaţia va trebui sa permita căutari online pornind de la scrisul recunoscut prin functiile programate in aplicatie.

Căutarea pe internet se va face folosind motorul Google sau altul precum şi folosind motoare specifice unei materii cum ar fi Wolfram Alpha pentru matematică sau Wikipedia.

Căutarea sa poata fi executată şi direct prin introducerea de text de la o tastatură virtuală.

Efectul căutării sa fie vizibile către toţi participanţii indiferent cine îl iniţiază. Mai mult, oricare dintre participanţi sa poata continua căutarea sau navigarea pe internet – ceilalţi sa vada în mod continuu efectul acţiunilor acestuia.

- **Play multimedia**

Aplicaţia sa permita redarea de conţinut multimedia care este afişat tuturor perticipanţilor. Controlul privind: play, pauză, stop, înapoi, înainte este disponibil oricărui participant. Formatul multimedia sa fie jpg pentru poze si mp4 pentru video.

- **Partajarea informatiei de pe device-ul propriu**

Aplicaţia va trebui sa ofere posibilitatea utilizatorilor sa aduca sub forma de poze orice continut deja disponibil pe device-ul propriu. Pentru a aduce informatii de pe device-ul propriu aplicatia trebuie sa permita accesul la sistemul calculatorului fara a intrerupe cursul.

- **Spatiu de stocare propriu**

In aplicatie, fiecare utilizator va trebui sa aiba un spatiu de stocare propriu care va trebui sa stea in cloud. In acest spatiu, utilizatorii sa poata urca materiale multimedia cum ar fi poze in format jpg sau filme in format mp4.

5.1.1.3. Examinarea

Platforma de instruire trebuie să permită examinarea voluntarilor atât în aceeaşi sală de clasă

cât şi de la distanţă.

Examinarea implică crearea testelor:

- crearea de teste clasice
- crearea de teste grilă
- crearea de teste orale, interactive

Examinarea implică planificarea testelor şi a participanţilor la teste.

Examinarea implică desfăşurarea testelor:

- clasice, în care participanţii scriu şi argumentează răspunsurilor

- grilă, în care participanţii aleg răspasurile corecte

- orale, în care participanţii răspund oral şi utilizând ”tabla clasei” fie că sunt în aceeaşi încăpere cu examinatorul fie că sunt la distanţă. Testele orale pot fi înregistrate similar sesiunii de predare pentru a fi evaluate ulterior.

Examinarea implică corectarea testelor şi stocarea rezultatelor în dosarul voluntarului.

Corectarea testelor va fi făcută de examinator.

5.1.1.4. Autoinstruire

Platforma de instruire va permite utilizarea instrumentelor puse la dispoziţie de aplicaţia ”tabla clasei” pentru a fi create cursuri destinate autoinstruirii.
În acest sens se solicită furnizarea a unui curs de prim ajutor creat cu aceste instrumente care va fi inclus în platforma livrată.

Cursurile pentru autoinstruire vor fi accesibile prin portalul informatic şi pot fi parcurse utilizând o platforma „tabla clasei” pentru redarea acestora.

5.1.1.5. Infrastructura asociata

In cadrul proiectului furnizorul va asigura echiparea unei singure locatii pentru facilitatea de instruire a utilizatorilor. Aceasta va fi echipata minimal cu sistem de videoconferinta, tabla

interactiva multitouch, videoproiector, dispozitive USB pentru cursanti, aplicatii specifice. Aceasta locatie va fi folosita de catre specialistii ce vor sustine cursurile.
Caracteristicile minimale ale tablei interactive ofertate vor fi:

- Diagonala minim 80 inch
- Input: senzor de imagine cu infrarosu
- Interfata de conectare: USB 1.1/2.0
- Rezolutie: 0.05 mm
- Sistemul de video conferinta ofertat trebuie sa suporte definitie inalte HD 1080p, posibilitate de transmitere prezentari in timpul transmisiei, sa suporte protocoale de VoIP tip H323, SIP, sa permita conectarea simultana a doua camere 720p/60fps, sa permita microfon cu functii de reducere zgomot si sa fie dotat interfata Ethernet.

5.2. Cerinte privind serviciile de testare software si management a calitatii sistemului

Asigurarea funcţionării la un nivel ridicat de calitate şi disponibilitate a sistemului impune

asigurarea unei funcţionări corecte a acestuia, în condiţii de încărcare maxima.
Pentru aceasta, pe durata proiectului, implementatorul va trebui sa puna la dispozitie si sa livreze serviciile de testare functionala si de performanta a sistemului utilizand o platforma

software integrată pentru managementul calităţii software, testare functionala automata si

testare de stress si de performanta automata (prin simularea unui numar de minim 500 utilizatori concurenti ai sistemului).
Testarea sistemului se va realiza in mai multe etape, dupa cum urmeaza:

- Testare functionala;
- Testare pentru identificarea problemelor, cu mentionarea zonei de cod care trebuie optimizata (clasa, functie, procedura, instructiune);

- Testare de performanta si testare de stres (numar utilizatori concurenti, timp de raspuns);

Principalele standarde utilizate in cadrul metodologiei de testare din cadrul proiectului vor fi cele incluse in metodologiile de lucru ale ISEB (BS7925-2 Software Component Testing Standard) sau ISTQB. Strategia de testare, definita conform standardelor mentionate mai sus, va contine scopul si obiectivele testarii, criteriile de intrare si iesire, mediul de testare, modalitatea de executie a testelor, tipurile de teste, management-ul defectelor, management-ul release-urilor, niveluri de testare, roluri si responsabilitati.

Pentru optimizarea procesului de testare a aplicatiilor software ce urmeaza a fi livrate in cadrul proiectului, se va alege un nivel dorit conform Test Process Maturity Matrix (TPI®) si se va modela un proces de calitate pe baza ariilor cheie din care este format.

Procesul de testare va avea la baza un flux de lucru, proceduri, strategie de testare, modele pentru planul de test si cazuri de test, tipuri de rapoarte dorite si indicatori de performanta.

Platforma de testare ce urmeaza a fi pusa la dispozitie proiectului de catre ofertant trebuie sa fie compatibila cu sistemul implementat, sa fie modularizata, cu integrari native intre diferitele componente si module. De asemenea, va permite automatizarea de funcţii şi procese pe parcursul procesului de testare, pentru a răspunde dezideratului de eficientizare.

Testele vor fi definite de catre ofertant astfel incat sa se asigure ca aplicatiile functioneaza corect.

Testarea se va face iterativ (in cicluri de testare), cand au loc schimbari in sistem, cauzate de corectia defectelor identificate pe durata testarii.

5.2.1. Metodologia de testare

Metodologia de testare ce va fi utilizata de implementator pentru testarea sistemului va indeplini urmatoarele cerinte minime:

- Se va crea un model pentru planul general de testare, care va contine scopul planului, zonele testate ale sistemului, rolurile si responsabilitatile echipei de testare in cadrul procesului de testare, date de intrare pentru efectuarea testelor, cazurile de testare;
- Se va implementa o strategie automatizata de gestionare a specificatiilor functionale si tehnice, a defectelor identificate, care va include si un flux de lucru;

- Se solicita crearea unui model pentru cazurile de testare. Cazul de testare reprezinta un set de actiuni executate de utilizator pentru a verifica obtinerea unui rezultat asteptat din partea modulului/subsistemului/sistemului testat;
- Se va dezvolta si implementa un flux de lucru optimizat pentru procesul de testare;
- Se va crea si pune la dispozitie un set de proceduri de lucru pentru testeri, care sa descrie actiunile de urmat in anumite situatii intalnite in cadrul procesului de testare;
- Se va efectua managementul automat al cazurilor de testare;
- Se vor utiliza modele automatizate pentru analize de risc si impact;
- Se va efectua implementarea intr-un instrument software dedicat al rapoartelor, KPI-urilor, fluxurilor de lucru din cadrul procesului si a tuturor modelelor;

5.2.2. Instruirea utilizatorilor pentru metodologia si platforma de testare

Implementatorul va instrui un numar de 5 testeri atat pentru metodologia de testare, cat si pentru platforma de testare software ce va fi utilizata in cadrul proiectului. Scopul pregatirii utilizatorilor este de a asigura faptul ca acestia asimileaza terminologia folosita (promovata de ISEB prin standardul BS7925-1 sau mentiunile din ISTQB din vocabularul folosit) si sunt in masura sa utilizeze diferitele notiuni, modele, elemente automatizate si platforma software folosita pentru management-ul procesului de testare.

Sesiunile de instruire vor fi adaptate necesitatilor si proceselor specifice clientului si vor fi sustinute de un instructor certificat tehnic pentru metodologia si platforma software de testare ce urmeaza a fi utilizata in cadrul proiectului.

5.2.3. Livrabilele procesului de testare

Prin metodologia si platforma de testare ce va fi utilizata in cadrul proiectului, ofertantul va asigura realizarea cel putin a urmatoarelor livrabile:

Livrabilele Serviciilor de testare:		**Cerinte minime obligatorii**
Strategia testarii	Metodologie si/sau Strategie de testare	• Se va crea o strategie de testare conform standardelor ISO29119 urmand metodologiile ISEB si ISTQB, aceasta urmand sa contina scopul si obiectivele testarii, criteriile de intrare si iesire, mediul de testare, modalitatea de executie a testelor, tipurile de teste, management-ul defectelor, management-ul release-urilor, niveluri de testare, roluri si responsabilitati. • Se va propune un flux de lucru optimizat pentru procesul de testare care sa se integreze cu fluxurile de lucru propuse de celelelte echipe din cadrul proiectului; • Se solicita o testare software standardizata si rularea de procese de calitate standard prin folosirea unei platforme de management al calitatii unice, cu mecanisme de workflow si alerte;

		• Se va creea un model pentru planul de testare general (High Level Test Plan) care trebuie sa contina scopul planului, zonele testate, roluri si responsabilitati, date de intrare, cazuri de test; • Se solicita crearea unui model pentru cazurile de testare, un caz de testare (test case) reprezentand un set minimal de actiuni executate de tester pentru a verifica ca se obtine un anumit rezultat asteptat ; • Se doreste implementarea unei strategii automatizate de gestionare a specificatiilor functionale si tehnice ; • Se solicita implementarea unei strategii automatizate de gestionare a defectelor gasite care sa contina si un flux de lucru ; • Trebuie sa se permita utilizarea de modele automatizate pentru analize de risc si impact ; • Pentru optimizarea procesului de testare a aplicatiilor software ce urmeaza a fi livrate in cadrul proiectului, se va alege un nivel dorit conform Test Process Maturity Matrix (TPI®) si se va modela un proces de calitate pe baza ariilor cheie din care este format. • Se solicita creearea si descrierea de indicatori de performanta (KPI) in mod grafic sau lista ; • Trebuie sa se defineasca indicatori de performanta adaptati cerintelor proiectului (KPI) care sa poate fi integrati in platforma de testare si asigurare a calitatii propusa;
	Fluxuri de lucru si proceduri automatizate	• Trebuie sa se creeze un set de proceduri de lucru pentru testeri care sa descrie actiunile de urmat in anumite situatii intalnite in cadrul procesului de testare; • Se va utiliza o platforma de gestionare a testarii si asigurare a calitatii in care sa se poata automatiza fluxurile de lucru si procedurile din strategia de testare. Aceasta platforma trebuie sa poata gestiona intregul ciclu de viata al unei aplicatii software. Se solicita utilizarea unei platforme COTS cu module integrate care sa poata satisface cerintele de mai jos:. o Sa se ofere access pentru minim 5 utilizatori concurenti atat pentru functionalitatile de management al testarii cat si pentru modulul de testare functionala automata ; o Este necesar ca platforma de testare utilizata sa poata fi configurata conform metodologiilor internationale in domeniu, cum ar fi: BS7925; ISO29119; ITIL; ASAP Program Methodology; OUM (Oracle Unified Methodology);. o In cadrul acestei platforme sa se automatizeze si functiile de administrare si management pentru a reduce efortul necesar rularii procesului de calitate; o Trebuie sa se permita administrerea activitatilor de testare in asa fel incat inginerii de calitate, dezvoltatorii si analistii sa poata accesa activitatile inregistrate pe proiect de pe sisteme si din locatii diferite. o Platforma propusa trebuie sa gestioneze intreg ciclul de viata al aplicatiilor software dezvoltate in cadrul acestui proiect. Va trebui sa sustina creearea de specificatii functionale, testarea automata functionala, testarea manuala functionala, testarea de performanta, asigurarea calitatii, gestionarea procesului de testare, raportare pe intreg ciclul de viata si gestionarea defectelor;

Planificarea si Documentarea testarii	Planul de testare	Se va documenta modul in care se abordeaza testarea sistemului, obiectul efortului de testare, activitatile necesare pregatirii si efectuarii nivelurilor de testare, mediile de testare, livrabilele, rolurile si responsabilitatile pentru testare, procedurile de testare si metoda de raportare. Documentatia transmisa va contine minim urmatoarele informatii: • obiectele supuse testarii • obiectivele si perimetrul testelor • cerintele mediului de testare • functiile de testat si rezultatele asteptate • abordarea de testare si tipurile de teste prevazute • abordarea folosita in crearea/gestionarea datelor de test • succesiunea testelor din matricea testelor, cu dependentele corespunzatoare modelate in instrumentele de testare • responsabilitatile in procesul de testare • riscurile si actiunile de preintampinare a defectelor, cu determinarea impactului, probabilitatii si responsabilului cu preintampinarea lor • criterii de intrare/iesire, care sa asigure ca sunt pregatite conditiile de incepere a testelor planificate, respectiv finalizarea testelor planificate si eliminarea defectelor • livrabilele implicate • criterii de validare. Planul de testare atat pentru testele manuale, cat si pentru cele automate va fi definit in platforma de testare si asigurare a calitatii utilizate pentru acest proiect.
	Specificatii detaliate de testare	Trebuie sa contina cel putin: • cazurile de test pentru testarea functionala manuala; • cazurile de testare trebuie gestionate in mod automatizat prin implementarea unui flux de lucru si notificari automate. • descrierea datelor de test, cu referire la datele de intrare si la baza de date peste care se executa testele • scenariile de test (lanturi de executie a cazurilor de test, pentru a simula procese end-to-end) • matricea cerintelor functionale/non-functionale, matricea testelor de acoperire a cerintelor (mapeaza cazurile de test cu cerintele) • matricea conditiilor de test, matricea testelor de acoperire a conditiilor de test (mapeaza cazurile de test cu conditiile de test) Platforma in care vor fi gestionate specificatiile de testare trebuie sa permita importul de specificatii functionale direct din aplicatia utilizatorului, urmand ca ulterior sa poata fi automat convertite in cazuri de testare Pentru a prioritiza corect specificatiile de testare, platforma utilizata trebuie sa poata gestiona specificatii functionale oferind analize de acoperire (coverage), analiza de risc si impact. Platforma trebuie sa fie capabila sa genereze in mod automat documentatia de testare pe masura ce echipa de testare creaza testele. Documentatia trebuie sa poata fi exportata in formate compatibile cu procesoarele de text, pentru a fi personalizata.

Executia testarii	Testarea Functionala Automata si Manuala	• In cadrul platformei de testare si asigurare a calitatii propuse trebuie sa se permita automatizarea testelor functionale si a celor de regresie. Solutia trebuie sa fie usor de folosit, permitand crearea de teste sofisticate cu cat mai putine cunostinte. Acest lucru trebuie sa fie posibil prin captarea activitatilor procesului desfasurat direct din ecranul aplicatiei si generarea automata a testelor. Solutia trebuie sa ajute utilizatorul sa identifice mai usor defectele si activitati duplicate, sa genereze documentatia si sa semnaleze defectele dezvoltatorilor; • Pentru a eficientiza testarea functionala, platforma trebuie sa evite testarea duplicata si inutila datorata informatiilor incomplete, a comunicarii ineficiente si a lipsei de procese consistente si repetitive. Platforma trebuie sa permita vizionarea in timp real a cerintelor si a defectelor detectate, pentru a avea o perspectiva mai clara asupra riscurilor. • Solutia trebuie sa permita in mod usor expertilor sa adauge, modifice, ruleze si sa elimine pasi de testare. • Solutia trebuie sa se poata actualiza automat cand se recompileaza o aplicatie iar actualizarea trebuie sa fie propagata pentru toate testele care utilizeaza o anumita componenta. • Solutia trebuie sa permita testarea functionala atat manuala, cat si automata, creearea testelor, mentenanta si executia lor cat si managementul datelor de test. Solutia trebuie sa permita reducerea timpului necesar ciclurilor de test prin implicarea in procesul de creeare a testelor si optimizare a procesului de calitate a expertilor in procesele ce urmeaza a fi implementate, prin automatizarea creerii de planuri de testare si eficientizarea mentenantei testelor atunci cand se modifica aplicatiile. • Solutia propusa trebuie sa permita executia de teste functionale manuale si automate. • Solutia trebuie sa permita colaborarea usoara intre grupuri de lucru prin administrarea si partajarea definitiilor si obiectelor aplicatiei. • Trebuie sa se poata combina testarea functionala manuala cu cea automata pentru ca expertii care cunosc detalii despre dezvoltarea si testarea aplicatiei sa poata participa la procesul de optimizare a calitatii. • Utilizatorii trebuie sa poata defini teste manuale sau automate care sa execute diferite activitati pentru fiecare componenta si trebuie sa poata converti componentele manuale in componente automate, asociindu-le cu anumite zone ale aplicatiei. Aceste functionalitati trebuie implemntate in cadrul platformei propuse. • Trebuie sa se permita generarea de teste automate intr-un mod flexibil, prin combinarea automatizarii testelor cu documentatia, astfel permitand masurarea calitatii din definitii abstracte. Utilizatorul trebuie sa poata defini teste manuale sau automate care sa faca diferite activitati pentru fiecare componenta si trebuie sa poata converti acele componente in componente automate, asociindu-le cu anumite zone ale aplicatiei. • Solutia trebuie sa permita administrarea centralizata a cazurilor de test, a testelor si a planului de testare. • Trebuie sa se permita asamblarea proceselor care urmeaza a fi testate folosind componente reutilizabile. Acest lucru trebuie sa poata fi efectuat prin actiuni de tip „drag-and-drop” de componente in script, dintr-o lista arborescenta;. • Serviciile propuse trebuie sa includa un minim de 80 script-uri automate de testare functionala si 300 de cazuri de testare functionala manuala create pentru componenta aplicativa specifica;

		• Se va putea genera in mod automat documentatia de testare in timp ce echipa de testare creeaza testele. Documentatia trebuie sa poata fi exportata in formate compatibile cu procesoarele de text, pentru a fi personalizata;. • Solutia trebuie sa ofere automat si in timp real functionalitati de control al acoperirii cu teste a specificatiilor functionale si de agregare a rezultatelor testelor la nivel de specificatie functionala • Solutia utilizata trebuie sa afiseze automat si in timp real rezultatele testarii la nivelul fiecarei actiuni din fluxul de lucru indiferent de granularitatea acestuia si de cate specificatii functionale compun acea activitate • Trebuie sa se poata optimiza testarea manuala astfel incat timpul petrecut de tester cu citirea si manipularea cazurilor de testare manuale sa fie minim • Solutia trebuie sa fie optimizata pentru a fi folosita de utilizatori fara cunostinte tehnice de testare automata sau manuala si sa permita resurselor non-it rularea de teste automate si manuale fara sa necesite cunostinte tehnice. • Solutia trebuie sa permita administrarea centralizata a cazurilor de test, a testelor si a planului de testare. • Solutia trebuie sa permita creearea de teste automate si utilizatorilor (testeri) care nu au notiuni de scripting sau programare. • Se va permite administrarea şi editarea simultana a mai multor teste şi librării, reducând timpul petrecut de testeri la depanarea scripturilor şi facilitând astfel procesul de testare. • Se va permite, în cazul unui test format din mai multe operaţii, rularea oricărei dintre operaţii fără a fi necesară rularea întregului test, reducând considerabil timpul de testare. • Trebuie sa se permita compararea directă a fişierelor PDF şi rularea checkpointurilor (verificarilor) pe acestea • Se vor putea actualiza automat toate scenariile de testare in momentul in care o componenta a testarii manuale este modificata • Se va oferi un utilitar care usureaza munca de testare manuala. Acest utilitar trebuie sa fie intuitiv, usor de folosit de catre utilizatori neexperimentati si sa nu necesite sedinte de instruire. Utilitarul de testare manuala trebuie sa prezinte capacitati avansate de editare

		a testelor manuale inclusiv abilitatea de a creea automat pasi in cadrul cazului de testare in timpul navigarii prin aplicatia de testat; • In cadrul procesului de testare manuala utilitarul folosit trebuie sa fie capabil sa injecteze automat date in campurile aplicatiei de testat, reducand astfel timp si posibilitatea de a introduce eroari datorate introducerii manuale.. • In urma executiei testelor in cadrul platformei propuse, vor trebui urmarite si rezolvate defectele descoperite.
	Testarea Automata de Performanta	Asigurarea functionarii la un nivel ridicat de calitate al sistemului propus impune functionarea corecta in conditii de incarcare maxima fara erori semnificative. In acest scop, trebuie furnizate servicii de testare de performanta cu ajutorul unei o solutii automate care sa asigure verificarea corecta a capacitatii sistemului si robustetea acestuia. Platforma de testare de performanta utilizata in acest scop va avea urmatoarele caracteristici minime si obligatorii: • Trebuie sa permita validarea performantei aplicatiilor ce se vor implementa. Solutia utilizata trebuie sa poata genera probleme de performanta si sa diagnosticheze provenienta acestora premergator punerii in productie a sistemului;. • Se va putea prezice comportamentul sistemului si performantele aplicatiilor prin verificarea faptului daca noile dezvoltari sau actualizari intrunesc cerintele de performanta definite, permitand identificarea si eliminarea problemelor de performanta pe durata etapei de dezvoltare a sistemului;. • Se va putea reduce timpul necesar procesului de creare a scripturilor prin inregistrarea scripturilor la nivel de interfata cu utilizatorul, prin click-uri pe ecran, astfel incat scripturile sa poata fi generate in mod automat; • Solutia utilizata in cadrul proiectului va trebui sa fie capabila sa genereze in mod virtual un numar de 500 de utilizatori concurenti pentru a simula activitati reale, folosind resurse hardware minimale. • Vor trebui dezvoltate si furnizate minim 30 script-uri de testare de performanta si sa se ruleze minim 3 teste de performanta (3 iteratii) pentru component aplicativa specifica. • Se va putea determina dimensionarea optima a platformelor hardware si software in functie de pasii proceselor ce vor face obiectul testarii;. • Se va permite cresterea incarcarii virtuale pentru a permite simularea incarcarii de varf pentru procesele selectate pentru testare.; • Se vor putea masura rezultatele performantei simulate cu indicatorii de performanta ai procesului, astfel incat sa se poata recomanda modificarile necesare a fi efectuate asupra aplicatiei software testate;. • Se va putea permite re-rularea testelor folosind mediul de lucru modificat pentru a valida eficacitatea modificarilor;. • Ttrebuie sa se puna la dispozitie module avansate de analiza si diagnosticare;. • Se vor putea izola problemele legate de performanţa şi se va putea reduce timpul mediu până la soluţionarea blocajelor de performanţa ale aplicaţiei testate;. • Se va putea verifica daca noile dezvoltari sau actualizari indeplinesc criteriile specifice de performanta pre-definite. • Se vor putea semnala cu usurinta blocajele la nivel de utilizatori finali, sistem si cod; • Trebuie sa se ofere un motor pentru scanarea datelor despre utilizatorul final,

		sistem şi diagnostice şi sa ofere cele mai probabile 10 cauze ale încetinirii sistemului;. • Se va testa anduranţa sistemului pe intreaga arhitectura a acestuia, aplicând fluxuri consistente, măsurabile şi repetabile si sa utilizeze datele pentru a identifica problemele de scalabilitate ce pot afecta utilizatorii reali în producţie. • Se vor capta timpii de răspuns ai utilizatorului final pentru procesele de afaceri şi tranzacţiile cheie, pentru a determina dacă se pot îndeplini condiţiile asumate privind nivelul calităţii serviciilor oferite. • Trebuie sa ofere un modul distinct dedicat analizei rezultatelor care sa poata fi folosit pe alte statii de lucru (in afara de platforma de testare) pentru o interpretare si analiza ulterioara a rezultatelor. • Se va permite dezvoltarea de script-uri intr-un modul care poate fi folosit separat de server-ul de testare de performanta pentru a facilita dezvoltarea sau mentenanta de script-uri in paralel cu testele de performanta • Trebuie sa permita dezvoltarea usoara a script-urilor preferabil prin tehnologii de tip „click & script"; • Se va oferi un maximum de flexibilitate in conceperea script-urilor pentru a le putea adapta celor mai complexe procese ce urmeaza a fi testate; • Solutia trebuie sa permita creearea de script-uri cu minimum de cunostinte de scripting sau programare; • Se vor oferi functionalitati predefinite care sa permita creearea de componente / actiuni distincte care la randul lor sa se poata recombina in orice ordine in acelasi script pentru a simula procese complexe cu elemente aleatorii; • Se va permite simularea de procese care sa contina activitati repetate de un numar variabil si aleator de ori in cadrul aceluiasi script; • Solutia propusa trebuie sa contina un numar sufficient de monitoare care sa analizeze toate componentele sistemulu;i • Se vor pute agrega toate rezultatele furnizate de monitoare, sa le trimita catre un instrument de analiza in care sa poata fi suprapuse si editate diferite grafice • Instrumentul sau modulul de analiza a rezultatelor de performanta trebuie sa contina functionalitati de tip drill down pentru a analiza in detaliu rezultatele • Instrumentul sau modulul de analiza trebuie sa permita efectuarea de insemnari direct pe graficele analizate • Instrumentul sau modulul de analiza trebuie sa permita exportul rezultatelor prin template-uri de rapoarte in formate uzuale cum ar fi docx, html sau pdf • Se vor putea izola problemele de performanta ale aplicatiilor implementate si sa reduca MTTR al acestora. Solutia trebuie sa asigure informatii privind actiunile posibile pentru a rezolva problemele de performanta. • Solutia trebuie sa contina elemente neintruzive de monitorizare a performanţei care sa obţina şi sa afişeaze în timp real datele despre performanţă la fiecare nivel, server şi componentă a sistemului si sonde de diagnosticare care adună date la nivel de cod pentru a izola blocajele de la nivel de declaraţie sau metodă SQL. • Se va putea trece prin tranzacţiile lente ale utilizatorului final ajungând la componenta, metoda sau instrucţiunea SQL blocată, ajutând la rezolvarea problemelor de memorie, excepţii şi alte probleme similar

		• Se va putea detecta în mod automat care componente sunt „active” atunci când este efectuată o anumită tranzacţie şi sa colecteze date cu privire la acestea pentru analiză. • Solutia de testare de performanta utilizata trebuie sa fie capabila sa sa verifice primele trei dintrea cele mai defavorabile scenarii din activitatile de business actuale cu maxim 500 de utilizatori concurenti
	Raportare	Pentru a asigura o raportare eficienta trebuie indeplinite minim urmatoarele conditii: • Se va permite crearea de indicatori de performanta (KPI-uri) personalizati in functie de necesitatile proiectului, astfel incat sa se ofere un modul de vizualizare al acestora usor de folosit si adaptabil pentru fiecare utilizator; • Platforma trebuie sa ofere posibilitatea de configurare de rapoarte personalizate si posibilitatea de export a rezultatelor testarii; • Platforma trebuie sa poata fi accesibila de oricare dintre membrii echipei de testare prin intermediul unei interfete web. • Trebuie sa se monitorizeze calitatea pe parcursul ciclurilor de testare, permitand vizualizarea starii aplicatiei si compararea acestei stari cu cea din ciclul anterior in cadrul unui release.. • Platforma trebuie sa permita masurarea in procente a numarului de zile scurs din perioada alocata testarii in ciclul respectiv. • Trebuie sa se poata ordona prioritatile de testare in functie de gradul de risc. • Platforma trebuie sa permita raportarea si construirea unui grafic cu procesul de calitate al Beneficiarului si sa permita exportarea datelor catre un editor de text sau grafic pentru a permite o manipulare imbunatatita a datelor. • Trebuie sa se permita raportarea referitor la ciclul de viata al proiectului astfel incat datele sa fie colectate din surse multiple, sa permita crearea de perpective ad-hoc a starii proiectului in orice moment, de tendinte si de rapoarte cuprinzatoare care sa acopere intregul ciclu de viata al procesului. Raportarea trebuie sa fie facuta atat la nivelul intregului ciclu de viata al proiectului cat si la nivelul release-urilor sau a ciclurilor de dezvoltare/testare cuprinse intr-un release.

5.3. Servicii de tip CSIRT (Computer Security Incident Response Team)

Ofertantul trebuie să furnizeze servicii de tip CSIRT (Computer Security Incident Response Team) la un nivel ridicat prin furnizarea tuturor activităților necesare prin care să poată fi identicate, analizate și remediate incidentele de securitate informatică. Astfel ofertantul va trebui:

- Să furnizeze servicii de tip „preventive security monitoring” prin care să fie identificate atacurile cibernetice.
- Să furnizeze un singur punct de contact pentru raportarea și comunicarea incidentelor de Securitate.

- Să furnizeze un timp de reacție redus pentru a răspunde la incidente de Securitate
- Să dețină capabilități de identificare a atacurilor bazate pe vectori de infecție/propagare interni.
- Să dețină capabilități de rezolvare și investigare a incidentelor de securitate informatică.
- Să dețină echipamente dedicate de analiză a traficului de rețea, a log-urilor generate de echipamente de rețea și de protecție, și de corelare avansată a evenimentelor.
- Să dețină capabilități de recomandare a unor soluții temporare de remediere a problemelor.

5.3.1. Servicii de monitorizare securitate

Ofertantul trebuie:

- Să dețină echipamente și aplicații dedicate de tip SIEM prin care să se realizeze analiza traficului de retea și a evenimentelor generate de echipamentele din rețeaua Beneficiarului. Echipamentul utilizat trebuie să dețină capabilități de detecție și corelare a evenimentelor la nivel avansat. Echipamentul utilizat trebuie să dețină capabilități de integrare cu scanere de vulnerabilități astfel încât să permită o corelare avansată între evenimentele de rețea și vulnerabilitățile identificate.
- Să dețină capabilități de corelare și în baza feed-urilor de threat intelligence primite de la furnizori din domeniu.
- Să dețină și să opereze un sistem de generare a alertelor și de tracking a acestora.
- Să dețină experiență în configurarea unui sistem SIEM astfel încât nivelul de corelare să fie la un nivel ridicat.
- Să dețină capabilități de înțelegere a arhitecturii de rețea a Beneficiarului și de realizare a modului optim de implementare a senzorilor/colectorilor din rețeaua monitorizată.
- Să dețină capabilități de analiza preliminară a incidentelor prin funcționalități de drill-down și stocare a pachetelor de date din traficul de rețea aferente alertelor generate.
- Să asigure serviciile de monitorizare în regim 24/7.

- Timp de semnalare a incidentului catre beneficiar de maxim 1 oră de la generarea alertei în sistemul SIEM.
- Analiștii care monitorizează sistemul trebuie să dețină certificări pentru soluția SIEM utilizată pentru asigurarea serviciului.

5.3.2. Servicii de digital forensics

Ofertantul trebuie:

- Să furnizeze serviciul de ”digital forensics” prin care să permită identificarea fișierelor, proceselor, acțiunilor, sesiunilor de date, web site-urilor accesate, care au fost utilizate în compromiterea sistemului informatic tinta și să releve modul în care acestea au fost utilizate și breșa de securitate utilizată.
- Să dețină capabilități de restaurare și analiza a mediilor de stocare formatate.
- Să dețină capabilități de identificare a datelor și informațiilor compromise, exfiltrate, restaurarea sesiunilor de comunicații de tip chat.
- Să dețină un laborator specializat de ”digital forensics” în cadrul căruia să realizeze analizele echipamentelor de tip laptop, computer sau smartphones.
- Să dețină echipamente specializate care să permită achiziția de probe digitale din cel puțin echipamente de tip: computer, laptop, smartphones, tablete.
- Să dețină echipamente/aplicații specializate care permit achiziția în mod securizat/criptat de probe digitale de la distanță.

5.3.3. Servicii de Analiză malware

Ofertantul trebuie:

- Să dețină capabilități de înțelegere a modului de acțiune a unei aplicații malware prin identificarea fișierelor utilizate, a proceselor generate, a conexiunilor de date generate, a modificărilor realizate la nivel de sistem de operare, metode de obfuscare a prezenței.
- Să dețină capabilități de generare de indicatori de compromitere care pot fi utilizați în identificarea altor sisteme infectate.

5.3.4. Serviciul de evaluarea vulnerabilităților

Ofertantul trebuie:

- Să dețină echipamente și aplicații dedicate pentru identificarea și obținerea informațiilor despre sistemele informatice țintă, identificarea de vulnerabilități și formularea unor recomandări de remediere.
- Să dețină expertiza în realizarea analizei de risc pe baza specificului companiei, a nivelui de impact al vulnerabilităților identificate și a importanței sistemului informatic vulnerabil.
- Să dețină experiență în realizarea scanărilor de vulnerabilități și în analiza acestora
- Să dețină mecanisme de eliminare a vulnerabilităților ”false-positive”.
- Să aibă capabilitatea de realizare de recomandări de remediere a vulnerabilităților identificate.

5.3.5. Serviciul de ”penetration testing”

- Să dețină echipamente și aplicatii pentru realizarea de teste de penetrare la nivel de rețea, sistem de operare, aplicații, inclusiv cele web, servicii, rețea wireless.
- Să dețină proceduri de lucru conforme cu standardele în domeniu: NIST, ENISA, SANS Institute, prin care este redus riscul de a afecta rețeaua evaluată și prin care toate activitățile auditorilor sunt înregistrate.
- Să dețină certificări în domeniul ”penetration testing”.
- Să dețină mecanisme de validare a vulnerabilităților identificate.
- Să dețină capacitate de a realiza exploit-uri customizate pentru vulnerabilitățile particulare identificate în infrastructura clientului.
- Să dețină experiență în evaluarea rezultatelor testelor de penetrare prin evidențierea nivelului de risc și a gradului de impact.
- Să dețină mecanisme de reducere/eliminare a riscului de afectare a rețelei auditate.

5.4. Resurse materiale

În calitate de solicitant al investiţiei, Casa Naţională de Asigurări de Sănătate înţelege că pentru bunul mers al proiectului ce face obiectul investiţiei trebuie să pună la dispoziţie toate resursele necesare îndeplinirii cu succes a proiectului.

Întregul proiect se va realiza şi implementa la sediul CNAS sau într-o locaţie desemnată de CNAS.

În cadrul proiectului se va asigura dotarea hardware necesară pentru realizarea, testarea şi operarea curentă a sistemului propus. Astfel, vor fi achiziţionate de furnizorul soluţiei

informatice şi vor rămâne în proprietatea CNAS resursele materiale descrise în secţiunea de componente hardware din cadrul arhitecturii funcţionale a sistemului.

Aplicatiile rezultate in urma proiectului devin proprietatea exclusiva a CNAS, respectandu-se regimul licentierii aplicatiilor terte.

În acelaşi timp, beneficiarul va pune la dispoziţie pentru buna desfăşurare a fazelor proiectului, dar şi pentru exploatarea curentă a acestuia după implementare, dotări şi resurse materiale existente.

Dintre aceste resurse disponibile, enumeram:

Instituţia	**Adresa de implementare**	**Resurse materiale puse la dispoziţie de către instituţie**
Casa Naţională de Asigurări de Sănătate (CNAS)	Sediul CNAS sau locaţia desemnată de CNAS	• Sediul propriu situat la adresa: Calea Călăraşi 248, Bl. S19, Sector 3, 030634, Bucureşti • Calculatoarele personalului tehnic implicat, împreună cu echipamentele periferice alocate fiecărui calculator în parte (imprimante, scanere.) • Dotările permanente ale CNAS – mobilier, linii de comunicaţii.

5.5. Cerinte privind serviciile de implementare

Se doreste ca echipamentele si produsele software furnizate sa fie insotite de servicii de implementare si de project management de calitate, care sa garanteze atingerea cu succes a obiectivelor *Sistemului*.

Furnizorul va presta toate servicile necesare pentru implementarea si dezvoltarea solutiei in conformitate cu recomandarile producatorului si cerintele functionale si tehnice prezentate in Caietul de sarcini, precum si rezultate in urma etapei de analiza a proiectului. Toate costurile asociate trebuiesc cuprinse in oferta financiara.

Implementarea sistemului va include in mod obligatoriu:

- Livrarea si configurarea echipamentelor hardware si de comunicatie
- Analiza necesitatilor, a proceselor legate de activitatea de baza a beneficiarului, cu documentarea specificatiilor detaliate si a cazurilor de utilizare.

Este obligatoriu ca etapa de analiza a proceselor sa se desfasoare conform unei metodologii recunoscute care sa asigure Beneficiarul ca toata aria de acoperire a proiectului va fi tratata in mod corespunzator. Ofertantul trebuie sa includa in oferta

dovada compententei in domeniul analizei proceselor de business prin prezentarea metodologiei pe care o aplica in proiectele

sale. Este obligatoriu ca metodologia propusa pentru analiza proceselor sa faca parte din sistemul de management al calitatii implementat in cadrul organizatiei ofertantului si sa fie certificata in conformitate cu prevederile standardului ISO:9001.

Pentru realizarea *Sistemului* furnizorul va folosi o metodologie unica pentru toate componentele, pentru a asigura premisele unei implementari de succes.

Livrarea echipamentelor hardware si a licentelor software, instalarea lor si punerea in functiune a sistemelor de operare, va fi realizata de catre personalul furnizorului impreuna cu reprezentantii beneficiarului. Beneficiarul va asigura conditiile tehnice pentru desfasurarea in bune conditiuni a acestor activitati pe amplasamente.

5.5.1. Graficul de implementare

Durata de implementare a proiectului este de maxim 9 luni de la data încheierii contractului**.** In faza de inceput (maximum 10 zile de la data semnării contractului) a proiectului furnizorul va prezenta un plan de proiect actualizat (inclusiv diagrama GANTT), conform metodologiei de Project Management propuse, care va fi analizat si agreat de beneficiar.

5.5.2. Cerinte privind serviciile de management de proiect

Ofertantul solutiei informatice trebuie sa prezinte in cadrul propunerii tehnice descrierea detaliata a metodologiei de management de project pe care o va utiliza in cadrul proiectului. Metodologia de management a proiectului va fi detaliata in oferta tehnica si va confine cel putin evidentierea urmatoarelor aspecte: controlul fazelor, activitatilor, atributiilor, planificarea in timp, alocarea resurselor, continutul si rezultatul etapelor, confirmarea rezultatetor si documentarea procesului de implementare.

Toate procedurile de lucru prezentate pentru toate fazele de proiect vor fi certificate ISO 9001, iar in cadrul ofertei trebuie prezentate toate procesele de lucru pentru fiecare activitate identificata de catre ofertant.

Planul tehnic de proiect prezentat de ofertant trebuie sa acopere urmatoarele puncte:

- Organizarea proiectului
- Diagrama Gantt a proiectului conține planificarea activitatilor, timpul de desfasurare
- Resursele implicate, livrabilele fazelor de implementare

- Organizarea echipei de proiect — include rolurile si responsabilitatile necesare persoanelor care vor efectua implementarea si operarea.

Ofertantul trebuie sa prezinte un plan detaliat, coerent si etapizat care sa acopere fazele de analiza si implementare. Ofertantul va prezenta in cadrul solutiei propuse un plan detaliat pozitionat in timp si spatiu care sa cuprinda elemente esentiale incluzand cel putin: analiza, identificarea si evaluarea proceselor, designul aplicatiei, dezvoltarea/customizarea sistemului integrat, asistenta tehnica, alte aspecte considerate de importanta pentru proiect. Furnizorul va configura sistemul propus in conformitate cu fluxurile identificate, analizate si evaluate.

Initierea proiectului

Ofertantul va prezenta in detaliu modalitatea in care proiectul va fi organizat, incluzand cel putin urmatoarele elemente: Comitetul de conducere al proiectului, managerul de project, sefii de echipa si alte roluri importante din cadrul echipei tehnice de project.

Se va prezenta modalitatea de escaladare a problemelor in interiorul organizatiei ofertantului.

Planificarea proiectului

Ofertantul va prezenta modalitatea in care propune sa aplice procesul de planificare in cadrul proiectului.

Pentru toate serviciile incluse in bugetul proiectului (in oferta financiara) se vor prezenta livrabilele care vor rezulta in urma prestarii serviciilor.

Executia proiectului

Managerul de proiect din partea furnizorului va fi responsabil cu executia proiectului. De asemenea acesta va fi responsabil cu intretinerea unui Registru al Problemelor de project pe intreaga durata a derularii acestuia.

Ofertantul va prezenta in cadrul propunerii tehnice modul in care se vor rezolva problemele care pot sa apara pe parcursul proiectului.

Monitorizare si control

Ofertantul trebuie sa descrie cum va realiza monitorizarea evolutiei proiectului precum si criteriile de calitate urmarite pe durata de viata a proiectului.

Ofertantul va descrie tipul si frecventa rapoartelor de monitorizare a evolutiei proiectului.

Finalizarea proiectului

Ofertantul va propune documentele care se intocmesc la finalizarea proiectului.

Ofertantul trebuie sa isi dimensioneze echipa de project astfel incat, pe toata durata contractului, persoanele responsabile de derularea acestei activitati sa fie disponibile on-site in vederea derularii in conditii optime a proiectului.

5.5.3. Implementarea proiectului

Ofertantul trebuie sa prezinte metodologia de implementare pe care o va folosi in desfasurarea intregului project. Metodologia trebuie sa fie bazata pe metodologiile standard folosite in proiecte IT de complexitate ridicata.

Ofertantul trebuie sa prezinte in cadrul proiectului modalitatea prin care se va realiza comunicarea intre participantii la proiect.

Ofertantul va prezenta in cadrul propunerii tehnice si modalitatea de tratare a schimbarilor in cadrul proiectului (in limitele Caietului de Sarcini). Se va prezenta descrierea procedurii de management al schimbarilor precum si formularele care vor fi utilizate in cadrul acestui proces pe durata proiectului.

Ofertantul trebuie sa isi dimensioneze echipa de conducere a proiectului astfel incat, pe toata durata contractului, persoanele responsabile de derularea acestei activitati sa fie disponibile on-site in vederea derularii in conditii optime a proiectului.

Oferta trebuie sa includa descrierea la nivel inalt a activitatilor, modalitatea in care aceste activitati vor fi duse la indeplinire si livrabilele produse in urma activitatilor pe parcursul urmatoarelor etape:

- Analiza
- Proiectare
- Testare
- Implementare
- Instruire
- Livrare, instalare si configurare hardware si pachet software
- Mentenanță și suport tehnic
- Asistenta tehnica

Planul initial care va fi prezentat impreuna cu oferta trebuie sa acopere toate etapele mentionate mai sus.

5.5.4. Garanția sistemului și servicii de mentenanță

Pentru toate echipamentele şi pentru produsele software de bază se va acorda suport tehnic până la finalizarea implementării proiectului, conform contractului încheiat de instituţia beneficiară cu furnizorul soluţiei informatice.

Se solicita ca perioada de garantie pentru componentele hardware ofertate sa fie:

- Echipamente hardware: 3 ani (servere), 3 ani (sistem de stocare) de la data finalizarii etapei de constructie a sistemului.
- Echipamente de comunicatii: 3 ani de la data finalizarii etapei de constructie a sistemului.

Pentru componenta software de baza si de aplicație specifica perioada de garanție este de 3 ani de la data finalizarii etapei de constructie a sistemului, iar apoi costurile de depanare defecte aplicative şi realizare de versiuni noi ale aplicaţiilor informatice vor face obiectul unui contract de service şi suport tehnic.

Pe întreaga perioadă de garanţie furnizorul soluţiei informatice va asigura obligativitatea funcţionării sistemului în perioada de post-implementare, va presta servicii de suport pentru toate sistemele software furnizate, iar această activitate va fi monitorizată de către Responsabilul de proiect.

Activităţile de mentenanţă şi suport din aceasta perioadă vor realiza prevenirea şi remedierea defecţiunilor şi anomaliilor apărute la produsele software din cadrul soluţiei informatice.

Remedierea defecţiunilor pe perioada garanţiei se va face la sediul beneficiarului proiectului sau prin intervenţie de la distanţă (*remote maintenance*), iar în cazul unor defecte mai grave, echipamentele se vor transporta la sediul furnizorului de către acesta.

Fiecare intervenţie în perioada de garanţie va fi documentată cu ajutorul unei fişe de intervenţie care va conţine următoarele detalii: data intervenţiei, descrierea intervenţiei, modalitatea de rezolvare a intervenţiei (reparaţie/înlocuire), durata de intervenţie şi confirmarea recepţiei prin semnăturile furnizorului şi beneficiarului.

5.5.5. Urmarirea incidentelor

Persoana desemnata ca punct de contact din partea beneficiarului va lansa un incident, Administratorul de Servicii al furnizorului primind o notificare. Fiecare incident va avea atasat un nivel de prioritate care sa reflecte impactul problemei asupra functionarii sistemului. Initial atasarea nivelului de prioritate se va face cu ajutorul Administratorului de Servicii al furnizorului pentru a facilita rezolvarea incidentului in timp util.

Nivelul de prioritate poate fi modificat cu acordul partilor in functie de evolutia incidentului.

Furnizorul poate sa isi rezerve dreptul de a modifica un nivelul de prioritate a incidentului, dar cu anuntarea in avans a echipei beneficiarului

Serviciile de Suport vor fi furnizate sub incidenta Clauzelor de Confidentialitate.

Asistenta poate fi de doua tipuri:

- on site (numai la sediul central al beneficiarului)
- remote (beneficiarul va asigura un acces VPN consultantilor care vor remedia problema)

Niveluri de Prioritate

Definitiile nivelelor de prioritate sunt cele de mai jos:

Denumire	**Descriere**
Urgent	Impact Major asupra functionarii sistemului Problema impiedica desfasurarea activitatii în sistem, procesul de activitate este serios afectat si nu mai poate continua pierderea functionalitatilor devenind critică.
Critic	Impact Semnificativ asupra functionarii sistemului Problema impiedica desfasurarea in conditii normale a activitatii utilizatorilor. Nicio solutie alternativa nu este disponibila, dar activitatea utilizatorilor poate totusi continua, insa limitata la componentele neafectate.
Major	Impact Mediu asupra functionarii sistemului. Problema afecteaza minor functionalitatile sistemului. Impactul reprezinta un inconvenient care necesita solutii alternative pentru refacerea functionalitatilor.

Minor	Impact Minim asupra functionarii sistemului Problema nu afecteaza functionalitatile sistemului. Rezultatul este o eroare minora care nu impiedica desfasurarea in bune conditii a activitatii utilizatorilor.

In cazul incidentelor cu nivel de prioritate ”urgent” asistenta va fi asigurata 24x7, fiind disponibila pana cand problema va fi rezolvata. Pentru aceasta beneficiarul va furniza o persoana de contact, disponibila 24x7, care sa furnizeze informatii, sa testeze solutii si sa aplice solutiile furnizate.

Timpi de raspuns si rezolvare

Pentru incidente critice Ofertantul va asigura urmatorii timpi de raspuns si de remediere:

Componenta sistemului	Timp de raspuns	Timp solutie provizorie/temporara	Timp de remediere
Echipamente Hardware	2 ore	6 ore	6 ore
Aplicatii software specifice	2 ore	12 ore	24 ore

Timpii de mai sus sunt calculati din momentul in care furnizorul a fost instiintat de aparitia problemelor.

La sfarsitul fiecarui caz deschis Furnizorul va efectua o analiza a cauzelor care au dus la producerea incidentului si va fi inclusa in recomandarea finala.

Definitiile, descrise mai jos se vor aplica la Service Level Agreement:

- *Timp de Raspuns*: Timpul scurs de la contactul initial dintre beneficiar si echipa de helpdesk a furnizorului si raspunsul primit de la echipa furnizorului catre beneficiar. Aceasta actiune se va desfasura prin intermediul e-mail.
- *Timp de Remediere*: Durata de timp pana la oferirea solutiei finale
- *Remediere Temporara*: O modificare in cadrul procedurilor sau datelor care sa evite erorile fara folosirea defectuoasa a produselor.

În cadrul timpilor de mai sus nu se contorizează perioadele de timp necesare pentru obținerea unor răspunsuri de clarificare a incidentului, în cadrul unui dialog purtat prin e-mail între echipa furnizorului și beneficiar.

5.5.6. Sustenabilitate

Prin realizarea activităţilor prevăzute şi îndeplinirea obiectivelor propuse, proiectul contribuie la soluţionarea pe termen lung a problemelor cu care se confruntă CNAS. După finalizarea implementării, rezultatele proiectului vor fi menţinute prin activităţile specifice desfăşurate în cadrul Departamentului Tehnologia Informaţiei al CNAS, prin personalul propriu.

Sustenabilitatea din punctul de vedere al asigurării capacităţii tehnice

Soluţia tehnică trebuie să fie dimensionată pentru a asigura scalabilitatea software şi hardware a aplicaţiei pentru o perioadă de minimum 5 ani după implementarea proiectului, iar solicitantul va aloca resurse tehnice (spaţii, accesul la reţeaua internă proprie de date şi accesul la Internet) pentru desfăşurarea optimă a procesului de implementare a sistemului.

De asemenea, sustenabilitatea tehnică a proiectului va fi asigurată prin activităţile de mentenanţă care vizează administrarea echipamentelor de calcul necesare bunei funcţionări a soluţiei software implementate, asigurarea suportului tehnic intern şi extern, ceea ce se va face de către furnizorii de echipamente hardware care vor asigura asistenţa tehnică pe o perioadă specificată în contractul de achiziţie.

În ceea ce priveşte sistemul software, asistenţa tehnică va fi prevăzută în contract pentru o perioadă de minim 3 ani din momentul în care începe executarea obligaţiilor contractuale, urmând ca pentru următorii ani serviciile de mentenanţă şi suport să fie contractate separat.

Sustenabilitatea din punctul de vedere al resurselor umane

Resursele umane alocate proiectului sunt suficiente atât din punct de vedere numeric cât şi din punct de vedere al experienţei. În situaţia apariţiei fluctuaţiei de personal, se va desfăşura instruirea noilor angajaţi de către personalul pregătit în cadrul proiectului. Persoanele implicate în proiect au experienţa în domeniul implementării de proiecte. Echipa va fi alcătuită din specialişti cu pregătire în diverse domenii aferente activităţilor desfăşurate, asigurând astfel interdisciplinaritatea necesară realizării unui astfel de proiect.

6. Cerinte privind formatul ofertelor

Limba de redactare a ofertei este limba romana. Ofertantul va suporta toate costurile asociate elaborarii si prezentarii ofertei sale, precum si documentelor care o insotesc, iar beneficiarul nu va fi responsabil sau raspunzator pentru costurile respective. Documentele emise de institutii/organisme oficiale din tara in care ofertantii straini sunt rezidenti pot fi prezentate in alta limba, cu conditia ca acestea sa fie insotite de o traducere autorizata in limba romana.

Toate cerinţele din prezentul caiet de sarcini, sunt minime şi obligatorii, nerespectarea oricăreia dintre cerinţe duce automat la declararea ofertei ca fiind neconforma. Nu se accepta oferte parţiale ci doar oferte complete şi care satisfac toate cerinţele prezentei documentaţii. Propunerea tehnică va fi prezentată astfel încât să fie posibilă maparea cu uşurinţă a corespondentei cu specificaţiile tehnice din acest caiet de sarcini. Fiecare cerință va fi tratată în propunerea tehnică prin descrierea modalității de îndeplinire însoțită de documente probatoare/referințe publice la documente probatoare. Simpla confirmare a îndeplinirii cerinței nu va fi acceptată pentru demonstrarea conformității propunerii tehnice. Este obligația ofertantului de a prezenta toate informațiile necesare demonstrării conformității propunerii tehnice iar ofertele care nu conţin aceste informaţii ci doar simpla confirmare a îndeplinirii cerinței vor fi declarate neconforme.

Ofertantul are obligatia de a mentine oferta valabila pe toata perioada de valabilitate stabilita de autoritatea contractanta. Orice oferta valabila pentru o perioada mai mica decat cea stabilita de autoritatea contractanta va fi respinsa de comisia de evaluare ca fiind necorespunzatoare. Perioada de valabilitate a ofertei este de 90 de zile. Autoritatea organizatoare/contractanta are dreptul de a solicita ofertantilor, in circumstante exceptionale, inainte de expirarea perioadei de valabilitate a ofertei, prelungirea acestei perioade. In cazul extinderii perioadei de valabilitate a ofertei, perioada de valabilitate a garantiei pentru participare va fi prelungita in mod corespunzator. Ofertantul are obligatia de a comunica autoritatii organizatoare/contractante daca este sau nu de acord cu prelungirea perioadei de valabilitate a ofertei. Ofertantul care nu este de acord cu prelungirea perioadei de valabilitate a ofertei se considera ca si-a retras oferta, fara ca acest fapt sa atraga pierderea garantiei pentru participare.

Desi nu se impune un format standard al ofertelor, partea tehnica va acoperi urmatoarele aspecte (ofertantul va adauga orice informatie considerata de interes relativ la oferta proprie):

1. Rezumat executiv
2. Prezentarea companiei
3. Intelegerea cerintelor proiectului precum si viziunea asupra realizarii *Sistemului*
4. Modul de prezentare a ofertei tehnice trebuie sa faciliteze urmarirea indeplinirii cerintelor si sa includa fara a se limita, urmatoarele:
    - Arhitectura tehnica detaliata a soluţiei ofertate
    - Descrierea tehnica a componentelor solutiei ofertate
    - Documentatia tehnica a produselor incluse in cadrul ofertei
    - Matricea de conformitate cu toate cerintele proiectului; se va include documentatie care sa faca dovada conformitatii cu cerintele tehnice (documentatie tehnica producatori, brosuri produse, etc.). Oferta tehnica va include toate documentele necesare pentru a dovedi caracteristicile fiecărei componente ofertata (producător, produs, documentatie tehnica de specialitate) datele de la producator care identifica produsele oferite. Se va raspunde punctual la fiecare dintre cerintele exprimate in caietul de sarcini si se va mentiona in clar daca cerinta este indeplinita si se vor da detalii privind modul de realizare a cerintei respective. Nu este acceptată simplă confirmare a îndeplinirii cerinţei fără o detaliere a modului de îndeplinire.
    - Plan de testare de securitate (descrierea sumara a tipurilor de teste de securitate a sistemului)
5. Detalierea scopului proiectului
6. Metodologiile de lucru pentru realizarea *Sistemului*
7. Descrierea activitatilor, a livrabilelor si criteriile de finalizare ale activitatilor
8. Planul de Management al Proiectului initial confom cerintelor din Caietul de sarcini
9. Planul de instruire
10. Lista echipa proiect furnizor cu documentele doveditoare cu privire la cerintele minime solicitate in Caietul de Sarcini
11. Anexe

Specificațiile tehnice definite in cadrul prezentului caiet de sarcini corespund necesităților și exigențelor CNAS. Având în vedere specificul acestui proiect, autoritatea a descris operatorilor economici interesați necesarul de livrabile și servicii la nivel de detaliu suficient, permițând astfel identificarea cu ușurință a obiectului acestei achiziții publice. Toate specificațiile, serviciile și cerințele menționate și solicitate in cadrul acestui caiet de sarcini sunt însoțite de mențiunea „sau echivalent". Oriunde în caietul de sarcini se întâlnesc nume, mărci, denumiri pentru anumite produse se va considera implicit adăugată menţiunea „sau echivalent". Ofertele care vor stipula "echivalent" vor fi evaluate numai în baza informaţiilor furnizate de ofertanţi. Comisia de evaluare nu este responsabilă pentru obţinerea oricăror alte informaţii ajutătoare care nu sunt conţinute în ofertă.

7. Alte cerințe

Soluția propusa de Ofertant trebuie să indeplinească următoarele cerințe minime obligatorii:

Nr. Crt	Descrierea cerintei
1	Ofertantul trebuie sa implementeze funcționalitățile cerute prin integrarea cu sistemele din platforma PIAS astfel încât să nu se duplice funcționalitățile și informațiile deja existente în acestea.
2	Soluția propusă trebuie să extindă funcțional, arhitectural și din punctul de vedere al infrastructurii informatice sistemele din platforma PIAS.
3	Soluția trebuie să respecte mecanismele de comunicare utilizate de sistemele din platforma PIAS.
4	Soluția trebuie să extindă aplicațiile locale, destinate uzului furnizorilor de servicii medicale și farmaceutice pentru conectarea la PIAS. Pentru producătorii de aplicații de pe piața IT conectabile la sistemele PIAS, furnizorul soluției informatice va publica specificațiile funcționale complete pentru conectarea la noul sistem informatic.
5	Soluția propusă trebuie să utilizeze registrele naționale și nomenclatoarele utilizate de sistemele din platforma PIAS.
6	Soluția propusă trebuie să permită autentificarea cetățenilor cu cardurile CEAS si a furnizorilor de servicii cu certificatul digital calificat.
7	Codul sursa al componentelor non-COTS (nu se aplica pentru produsele Customer Of The Shelf – sisteme şi aplicaţii software disponibile în mod comercial)

ANEXA 1 - Serviciile Web expuse de SIUI/SIPE/CEAS

În acest capitol sunt prezentate pe larg metodele expuse de interfaţa serviciului-web al SIUI. Prezentarea constă în descrierea semnăturii metodelor, adică a numelui, a parametrilor şi a tipului întors pentru fiecare metodă, urmate de o scurtă descriere a modului de folosire.

Accesul prin serviciul-web la SIUI se face în mod securizat prin autorizarea apelului pe bază de nume de utilizator şi parolă. În acest scop în SIUI trebuie înregistrat în prealabil un utilizator pentru fiecare furnizor de servicii medicale care doreşte să raporteze electronic datele în sistem.

Pentru accesul la sistem, în urma încheierii contractului dintre furnizor şi casa de asigurări, se eliberează o convenţie de utilizare care conţine codul de acces al utilizatorului autorizat sub forma unei serii de licenţă ce conţine o sumă de control. Această serie de licenţă este creată aleator de către sistem la cerere prin intermediul interfeţei de operare de la nivelul casei judeţene de asigurări.

OBSERVAŢIE

Prin convenţie numele acestui utilizator este chiar codul unic de identificate al acestuia (CUI sau CNP, dup caz) la care se adaugă codul SIUI ai casei de asigurări cu care s-a încheiat contractul de prestare servicii, respectiv convenţia de utilizare a aplicaţiei, iar parola este seria de licenţă de mai sus.

Prezentăm mai jos un exemplu practic de nume de utilizator şi parolă:

- Nume: 123456789_CODCAS
- Parolă: AB012-C345-D678-E910

Pentru fiecare serviciu Web vor fi prezentate în anexele acestui document structurile de date ale fişierelor XML, sub forma unor fişiere XSD (XML Schema Definition), precum şi fişierele WSDL de definiţie a metodelor expuse.

Sistemul SIUI foloseşte patru fişiere WSDL corespunzătoare funcţionalităţilor majore expuse:

- ***SiuiWS.wsdl*** pentru serviciile pentru sincronizarea nomenclatoarelor, fişierelor de personalizare, transmiterea de raportări şi preluarea rezultatelor prelucrării raportărilor, precum şi alte servicii conexe, expuse în secțiunile următoare. Toate aceste servicii expun online funcționalitățile oferite până acum de sistem în mod offline, prin transferul fișierelor folosind medii de stocare mobile.

- ***SiuiInsuredWS.wsdl*** pentru serviciul-web de verificare online a calităţii de asigurat al unei persoane/pacient. Acest serviciu este o funcţionalitate nouă expusă de SIUI începând cu anul 2011.

- ***SiuiValidateWS.wsdl*** pentru serviciile-web de pre-validare online a eligibilităţii la decontare a serviciilor prestate de furnizori. Acest serviciu este o funcţionalitate nouă introdusă în SIUI începând din decembrie-2011.

- ***SiuiEInvoiceWS.wsdl*** pentru serviciile-web destinate facturii electronice. Acest serviciu este o funcţionalitate nouă introdusă în SIUI începând din decembrie-2013.

Sistemul Informatic pentru Prescripţia Electronică (SIPE) foloseşte un singur fişier WSDL care expune funcţionalităţile specifice destinate medicilor prescriptori, dar şi farmaciilor care eliberează reţete compensate şi gratuite:

- ***EPrescriptionWS.wsdl*** pentru serviciile-web de procesare online a eligibilităţii la decontare a serviciilor prestate de furnizori. Acest serviciu este o funcţionalitate nouă introdusă începând din iulie-2012.

Sistemul CEAS (Cardul Electronic de Asigurări de Sănătate) foloseşte un singur fişier WSDL care expune funcţionalităţile specifice, care expune funcţionalităţile specifice destinate medicilor de familie, care vor realiza activarea cardurilor electronice de asigurare, precum şi alte operaţiuni de inscripţionare pe cardul electronic conform legislaţiei în vigoare:

- ***CeasMedicalDataWS.wsdl*** pentru serviciile-web de procesare a cererilor de transmitere online informaţiilor medicale care vor fi inscripţionate pe cardul electronic, precum şi a altor informaţii cum ar fi datelor de contact ale medicului de familie, sau declararea unui card cu date incorect. Acest serviciu este o funcţionalitate nouă introdusă începând din decembrie-2012.

Adresele serviciilor-web expuse de SIUI sunt următoarele:

```
https://www.siui.ro/svapntws/services/SiuiWS
https://www.siui.ro/svapntws/services/SiuiValidateWS
https://www.siui.ro/svapntws/services/SiuiInsuredWS
https://www.siui.ro/svapntws/services/SiuiEInvoiceWS
```

Adresa serviciilor-web expuse de Sistemul Informatic pentru Prescripţia Electronică:

```
https://sipe.siui.ro/svapntws/services/EPrescriptionWS
```

Adresa serviciilor-web expuse de Sistemul Informatic pentru Cardul Electronic de Asigurări de Sănătate:

```
https://ceas.siui.ro/svapntws/services/CeasMedicalDataWS
```

Adresa serviciului de autentificare şi validare OCSP a certificatelor digitale este următoarea:

```
https://www.siui.ro/OCSP/validator
```

A se nota că adresa pentru OCSP corespunde serviciilor expuse de SIUI; accesul la serviciile expuse de Prescripţia Electronică fiind realizat folosind aceleaşi certificate digitale şi credenţiale de acces (*utilizator/parolă*) ca şi pentru SIUI.

Serviciul de autentificare transmite aplicaţiei client un jeton de sesiune care trebui adăugat de către aplicaţie în antetul cererii HTTP pentru a putea accesa serviciile web din lista anterioară. Jetonul de sesiune este generat de serviciul de autorizare pe baza certificatului digital al utilizatorului SIUI.

OBSERVAŢIE

Pentru a putea obţine jetonul de sesiune serviciul de autentificare necesită transmiterea ca parametru a numelui utilizatorului SIUI care se solicită accesul, ca în exemplul următor: https://www.siui.ro/OCSP/validator?username=nnnnnn_CODCAS.

De notat că acest jeton are o perioadă de valabilitate limitată, după care expiră, fiind necesară obţinerea unui nou jeton.

Serviciul pentru sincronizarea nomenclatoarelor

Acest serviciu se foloseşte pentru descărcarea fişierului de nomenclatoare specifice pentru furnizorii de servicii medicale şi farmaceutice.

Metoda *getCatalogues*

```
String[] getCatalogues (
        String partnerCategory,
        DateTime start )
```

Metoda are doi parametri de intrare :

- parametrul ***partnerCategory*** de tip şir de caractere reprezintă codul categoriei de furnizor pentru care se cere versiunea actuală de nomenclatoare, lista valorilor permise fiind prezentată mai jos;
- parametrul ***start*** de tip dată calendaristică reprezintă data de la care se caută în sistem existenţa unei noi versiuni.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat. URL-ul va expira la momentul publicării unei noi versiuni de nomenclatoare pentru a nu permite aplicaţiilor de raportare să descarce accidental un fişier de nomenclatoare mai vechi folosind un URL din *cache*.

Dacă nu există o versiune mai nouă de nomenclatoare metoda întoarce ***null***.

Cel de-al doilea parametru poate fi folosit pentru a evita transferul inutil de date prin stocarea în aplicaţia client a datei la care s-a efectuat sincronizare anterioară şi prin folosirea acestei date ca dată de început pentru căutare a unei versiuni mai noi a nomenclatoarelor.

Fişierul XML va conţine în nodul rădăcină un câmp care va reprezenta data la care a fost generat. Această data va fi utilizată de aplicaţia client pentru a memora data valabilităţii nomenclatoarelor care va fi folosită ca valoare pentru parametrul al doilea.

Este recomandat ca aplicaţiile de raportare să nu permită importul unor nomenclatoare mai vechi decât cele deja încărcate în aplicaţie.

Instrucţiuni de folosire

Aplicaţia client trebuie să folosească URL-ul rezultat pentru a descărca fişierul cu nomenclatoarele. Dimensiunea fişierului poate fi folosită pentru a verifica completitudinea fişierului descărcat. Fişierul descărcat este o arhivă ZIP care conţine un fişier XML de nomenclatoare SIUI.

Schema de validare pentru acest fişier este detaliată în anexele corespunzătoare fiecărui tip de furnizor.

Prezentăm în continuare lista de valori admise pentru parametrul ***partnerCategory***:

Valoare parametru	Tip de furnizor corespunzător
MF	Medicină primară şi de familie
PHM	Farmacii (circuit deschis / circuit închis)
CLIN	Specialităţi clinice
PARA	Specialităţi paraclinice
STOM	Specialităţi stomatologice
AMB	Ambulanţe
MD	Dispozitive medicale
HC	Îngrijire la domiciliu
REC	Recuperare - ambulatoriu şi sanatorii
SPT	Spitale
NHP	P.N.S.
DIA	Hemodializă
SICK	Convenţii pentru concedii medicale
CBRET	Convenţii pentru reţete compensate

Un exemplu tipic de algoritm pentru actualizarea nomenclatoarelor este:

```
Se apelează metoda getCatalogues cu parametrii corespunzători.
Dacă apelul întoarce null atunci:
   - Se afişează mesajul "Nu există o versiune mai nouă".
Dacă se întoarce un vector de şiruri de caractere de lungime 2 atunci:
   - Se consideră primul şir ca fiind url-ul pentru descărcarea
fişierului.
   - Se descarcă fişierul (care este o arhivă ZIP).
   - Dacă dimensiunea fişierului descărcat coincide cu valoarea celui de-
al doilea element din vector atunci:
         - Se dezarhivează arhiva descărcată şi rezultă un fişier XML.
         - Se validează fişierul XML cu schema de validare XSD
corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se parcurge fişierul şi se actualizează valorile din
nomenclatoarele din baza de date.
         - Altfel se afişează mesaj de eroare "Fişier invalid".
   - Altfel se afişează mesaj de eroare de comunicaţie.
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

De asemenea, accesul prin URL la arhivă este securizat, folosindu-se aceeaşi nume de utilizator şi parolă ca pentru accesul la metoda Web, precum şi certificatul digital pentru deschiderea conexiunii SSL.

Serviciul pentru sincronizarea datelor de personalizare

Acest serviciu este folosit pentru descărcarea fişierului cu datele de personalizare specifice pentru furnizorii de servicii medicale şi farmaceutice.

Serviciul expune două metode, prima destinată furnizorilor care au contract cu Casa de Asigurări, iar a doua medicilor care au încheiat convenţii de prescriere a reţetelor compensate şi activează în instituţii care nu au contract direct cu Casa de Asigurări.

Metoda *getProviderInfo*

```
String[] getProviderInfo (
      String partnerCategory,
      DateTime start,
      DateTime stop,
      String uic )
```

Metoda are patru parametri de intrare :

- parametrul ***partnerCategory*** de tip şir de caractere reprezintă codul categoriei de furnizor, lista valorilor permise fiind prezentată mai jos;

- parametrul ***start*** de tip dată calendaristică reprezintă data de început a perioadei pentru care se caută datele furnizorului în sistem;

- parametrul ***stop*** de tip dată calendaristică reprezintă data de sfârşit a perioadei pentru care se caută datele furnizorului în sistem;

- parametrul ***uic*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de personalizare în format XML, care se validează cu schema de validare corespuspunzătoare fiecărui tip de furnizor, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

URL-ul va fi generat pentru fiecare cerere și are perioadă de valabilitate limitată după trecerea căreia nu va mai fi disponibil pentru a nu permite aplicaţiilor de raportare să descarce accidental un fişier de personalizare mai vechi folosind un URL memorat.

Metoda *getPartnerInfo*

```
String[] getPartnerInfo (
        String partnerCategory,
        DateTime start,
        DateTime stop,
        String uic,
        String subUnitCode )
```

Metoda are patru parametri de intrare :

- parametrul ***partnerCategory*** de tip şir de caractere reprezintă codul categoriei de furnizor, lista valorilor permise fiind prezentată mai jos;

- parametrul ***start*** de tip dată calendaristică reprezintă data de început a perioadei pentru care se caută datele furnizorului în sistem;

- parametrul ***stop*** de tip dată calendaristică reprezintă data de sfârşit a perioadei pentru care se caută datele furnizorului în sistem;

- parametrul ***uic*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;

- parametrul ***subUnitCode*** de tip şir de caractere reprezintă codul unic de identificare al subunităţii în sistem (valoarea *partnerCode* din fişierul de personalizare).

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de personalizare în format XML,

care se validează cu schema de validare corespuspunzătoare fiecărui tip de furnizor, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

URL-ul va fi generat pentru fiecare cerere şi are perioadă de valabilitate limitată după trecerea căreia nu va mai fi disponibil pentru a nu permite aplicaţiilor de raportare să descarce accidental un fişier de personalizare mai vechi folosind un URL memorat.

Parametrul ***subUnitCode*** se transmite cu valoarea ***null*** pentru cazul în care contractul se realizează direct cu o unitate medicală cu personalitate juridică, dar trebuie transmis pentru cazul unităţilor medicale care activează în instituţii şcolare sau instituţii de îngrijire a bătrânilor, care nu au contract direct cu Casa de Asigurări ci întocmesc convenţii de eliberare a reţetelor compensate.

Metoda *getProviderInfoForPhysician*

```
String[] getProviderInfo (
        String partnerCategory,
        DateTime start,
        DateTime stop,
        String uic,
        String stencilNo )
```

Metoda are patru parametri de intrare :

- parametrul ***partnerCategory*** de tip şir de caractere reprezintă codul categoriei de furnizor, lista valorilor permise fiind prezentată mai jos;
- parametrul ***start*** de tip dată calendaristică reprezintă data de început a perioadei pentru care se caută datele furnizorului în sistem;
- parametrul ***stop*** de tip dată calendaristică reprezintă data de sfârşit a perioadei pentru care se caută datele furnizorului în sistem;
- parametrul ***uic*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;
- parametrul ***stencilNo*** de tip şir de caractere reprezintă numărul de parafă al medicului titular de listă.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de personalizare în format XML, care se validează cu schema de validare corespuspunzătoare fiecărui tip de furnizor, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

URL-ul va fi generat pentru fiecare cerere și are perioadă de valabilitate limitată după trecerea căreia nu va mai fi disponibil pentru a nu permite aplicaţiilor de raportare să descarce accidental un fişier de personalizare mai vechi folosind un URL memorat.

Metoda este destinată în exclusivitate aplicaţiilor pentru medicii de familie, în special pentru cabinetele medicale cu contract pentru mai mulţi medici titulari de listă. În acest context, pentru validare se utilizează schema de validare specifică, şi anume ***PersonalizedFileMF.xsd***

Instrucţiuni de folosire

Fişierul de personalizare conţine date de identificare ale furnizorului, datele de contract, date legate de medicii angajaţi şi specialităţile acestora, precum şi, acolo unde este cazul, valorile tarifelor, plafoanelor sau altor sume contractate.

Schema de validare pentru fişierul de personalizare este detaliată în anexele corespunzătoare fiecărui tip de furnizor.

Prezentăm în continuare lista de valori admise pentru parametrul ***partnerCategory***:

Valoare parametru	**Tip de furnizor corespunzător**
MF	Medicină primară şi de familie
FARMD	Farmacii (circuit deschis)
FARMI	Farmacii (circuit închis)
CLIN	Specialităţi clinice
PARA	Specialităţi paraclinice
STOM	Specialităţi stomatologice
AMB	Ambulanţe
MD	Dispozitive medicale
HC	Îngrijire la domiciliu
RECA	Recuperare - ambulatoriu
RECS	Recuperare - sanatorii
SPT	Spitale
NHP	P.N.S.
DIA	P.N.S. / Dializă publică
FSD	Dializă privată
SICK	Convenţii pentru concedii medicale
CBRET	Convenţii pentru reţete compensate

Un exemplu tipic de algoritm pentru actualizarea datelor de contract este:

```
Se apelează metoda getProviderInfo cu parametrii corespunzători.
Dacă se întoarce un vector de şiruri de caractere de lungime 2 atunci:
   - Se consideră primul şir ca fiind url-ul pentru descărcarea
fişierului.
   - Se descarcă fişierul (care este o arhivă ZIP).
   - Dacă dimensiunea fişierului descărcat coincide cu valoarea celui de-
al doilea element din vector atunci:
         - Se dezarhivează arhiva descărcată şi rezultă un fişier XML.
         - Se validează fişierul XML cu schema de validare XSD
corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se parcurge fişierul şi se actualizează datele de
contractare din baza de date.
         - Altfel se afişează mesaj de eroare "Fişier invalid".
   - Altfel se afişează mesaj de eroare de comunicaţie.
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

De asemenea, accesul prin URL la arhivă este securizat, folosindu-se aceeaşi nume de utilizator şi parolă ca pentru accesul la metoda Web, precum şi certificatul digital pentru deschiderea conexiunii SSL.

Fişierul XML va conţine în nodul rădăcină un câmp care va reprezenta data la care fişierul a fost generat. Această data va fi utilizată de aplicaţia client pentru a memora data valabilităţii fişierului de personalizare.

Este recomandat ca aplicaţiile de raportare să nu permită importul unui fişier de personalizare mai vechi decât cel deja încărcat în aplicaţie.

Serviciul pentru trimiterea raportărilor periodice

Acest serviciu se foloseşte pentru trimiterea unui fişier de raportare periodic către SIUI. La momentul trimiterii se realizează validarea formei şi conţinutului fişierului, precum şi verificarea existenţei unui contract valid şi a unei perioade de raportare deschisă pentru furnizorul respectiv.

Metoda *sendReport*

```
Boolean sendReport (
        String reportType,
        String reportXml )
```

Metoda are doi parametri de intrare :

- parametrul ***reportType*** de tip şir de caractere reprezintă codul tipului de raportare, lista valorilor permise fiind prezentată mai jos;

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare semnat electronic, arhivat în formatul ***ZIP (JavaZip)*** şi codat ulterior în formatul ***Base64***.

Dacă metoda întoarce valoarea „*adevărat”*, atunci trimiterea raportului s-a făcut cu succes, altfel trimiterea s-a terminat cu erori. Pe baza mesajului primit în cazul unei erori se poate determina cauza respingerii raportării.

Instrucţiuni de folosire

Numele fișierului XML de raportare trebuie sa respecte formatul:

```
{Prefix} + "_" + {Cod} + "_" + {Data} + "_" + {Ora} + ".xml"
```

{Prefix} reprezintă un cod de identificare pentru tipul de furnizor, lista completă a acestor coduri fiind prezentată în tabelul de mai jos.

{Cod} reprezintă codul unic de identificare al furnizorului în sistem, codul fiscal, CUI sau CNP, după caz.

Parametrii ***{Data}*** şi ***{Ora}*** reprezintă data şi ora la care a fost efectuată raportarea şi trebuie să apară în formatul "AAAALLZZ" pentru dată şi "OOMM", fără nici un separator.

Schema de validare pentru acest fişier este detaliată în anexele corespunzătoare fiecărui tip de furnizor.

Prezentăm în continuare lista de valori admise pentru parametrul ***reportType***:

Valoare parametru	**Valoare prefix fişier**	**Tip de furnizor corespunzător / Tip de raportare**
MF	MF	Medicină primară şi de familie
PRM	PRM	Centre de permanenţă
FARMD	FARMD	Farmacii (circuit deschis) – reţete cu regim special (tipizate)
FARME	FARME	Farmacii (circuit deschis) – reţete electronice (online şi offline)
FARMI	FARMI	Farmacii (circuit închis)
CLIN	CLIN	Specialităţi clinice
PARA	PARA	Specialităţi paraclinice
STOM	STOM	Specialităţi stomatologice
AMB	AMB	Ambulanţe
MD	MD	Dispozitive medicale
HC	HC	Îngrijire la domiciliu
RECA	RECA	Recuperare - ambulatoriu
RECS	RECS	Recuperare - sanatorii
SICK	SICK	Certificate de concediu medicale
FSD	DIA	Dializă privată
NHPDIA	DIA	P.N.S. / Dializă publică
NHPREP	NHPREP	P.N.S. / Raportare de indicatori P.N.S.

Valoare parametru	Valoare prefix fişier	Tip de furnizor corespunzător / Tip de raportare
NHPCJ	NHPCJ	P.N.S. / Cereri justificative (facturi şi ordine de plată)
SPT_ACUT	SPT_ACUT	Spitale / Raportare de cazuri acute (internări)
SPT_CHR	SPT_CHR	Spitale / Raportare de cazuri cronice
SPT_DRG	SPT_DRG	Spitale / Raportare D.R.G.
SPT_SPZ	SPT_SPZ	Spitale / Raportare spitalizare de zi
SPT_PAL	SPT_PAL	Spitale / Raportare paliative

Un exemplu tipic de algoritm pentru generarea fişierelor XML de raportare este:

```
Se pregătesc datele pentru raportare:
   - Se generează fişierul de raportare XML corespunzător perioadei
selectate.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
   - Se semnează electronic fişierul XML, folosind standardul CMS
(RFC5652).
   - Se arhivează fişierul XML folosind algoritmul ZIP.
   - Se codifică conţinutul arhivei folosind codarea Base64.
Se apelează metoda sendReport cu parametrii corespunzători.
Dacă metoda întoarce valoarea true se afişează mesaj de succes.
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Pentru semnarea digitală a unui fişier în vederea procesării în SIUI este necesară deținerea unui certificat digital calificat *X.509* emis de unul din furnizorii acreditați de servicii de certificare din România. Perechea de chei aferentă certificatului trebuie să fie de tip *RSA*.

Fişierele semnate cu certificatul digital *X.509*, folosind algoritmul *SHA-1*, se transmit către SIUI folosind formatul *CMS* („*Cryptographic Message Syntax*”) publicat în *RFC-5652* de către *IETF* („*Internet Engineering Task Force”*) (vezi http://tools.ietf.org/html/rfc5652).

Descrierea algoritmului *SHA* („*Secure Hash Algorithm*”) este publicată de către National Institute of Standards and Technology (NIST) în *Digital Signature Standard FIPS 186-2.*

Majoritatea sistemelor de operare permit realizarea unei astfel de semnături digitale folosind biblioteci și/sau framework-uri disponibile în sistem, dar și prin produse adiționale.

Semnarea electronică a fişierului XML este necesară doar în cazul transmiterii electronice online a acestuia către SIUI, fişiere aduse la CAS de către furnizor pe suport electronic nu trebuie semnate

NOTĂ

Pentru furnizorii cu mai multe contracte pe aceeaşi perioadă de raportare trebuie generat câte un fişier pentru fiecare contract. Excepţie face aplicaţie de raportare pentru PNS unde se generează câte un fişier pentru fiecare PNS.

Raportări speciale

Pentru anumite categorii de furnizori există raportări speciale, care nu sunt în vederea decontării serviciilor, ci pentru trimiterea în sistem a unor informaţii auxiliare, de exemplu:

- structura organizatorică a unităţii (departamente, secţii, angajaţi)
- oferte de preţuri pentru servicii în vederea contractării

Valoare parametru	Valoare prefix fişier	Tip de furnizor corespunzător / Tip de raportare
RECA_OFFER	RECAMB_OFFER	Recuperare - ambulatoriu / Ofertă de preţuri pentru servicii
MD_OFFER	MEDDEV_OFFER	Dispozitive medicale / Ofertă de preţuri pentru dispozitive medicale
PARA_OFFER	PARA_OFFER	Paraclinice (Laboratoare) / Ofertă de preţuri pentru servicii
SPT_E	SPT_E	Spitale / Structura organizatorică (departamente, secţii, angajaţi)
SPT_I	SPT_I	Spitale / Raportare indicatori statistici
HBDG	HBDG	Spitale / Structură şi indicatori bugetari

Serviciul pentru preluarea rezultatelor raportărilor periodice

Acest serviciu se foloseşte pentru preluarea fişierului de răspuns pentru o raportare trimisă anterior către SIUI pentru prelucrare. Pentru ca fişierul de răspuns să poate fi descărcat acesta trebuie să fie salvat pe server, lucru care se efectuează automat în urma prelucrării fişierului de raportare.

Metoda *getReportFeedback*

```
String[] getReportFeedback ( String fileName )
```

Metoda are un singur parametru de intrare :

- parametrul ***fileName*** de tip şir de caractere reprezentă numele fişierului de raportare trimis de aplicaţie pentru care se cere răspunsul procesării.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

Dacă nu există un fişier de raportare procesat cu numele dat, metoda întoarce ***null***.

Instrucţiuni de folosire

Aplicaţia client trebuie să folosească URL-ul rezultat pentru a descărca fişierul cu nomenclatoarele. Dimensiunea fişierului poate fi folosită pentru a verifica completitudinea fişierului descărcat. Fişierul descărcat este o arhivă ZIP care conţine un fişier XML cu rezultatul procesării raportării în SIUI.

Schema de validare pentru acest fişier este detaliată în anexele corespunzătoare fiecărui tip de furnizor.

Un exemplu tipic de algoritm pentru actualizarea nomenclatoarelor este:

```
Se apelează metoda getReportFeedback cu parametrii corespunzători.
Dacă se întoarce un vector de şiruri de caractere de lungime 2 atunci:
   - Se consideră primul şir ca fiind url-ul pentru descărcarea
fişierului.
   - Se descarcă fişierul (care este o arhivă ZIP).
   - Dacă dimensiunea fişierului descărcat coincide cu valoarea celui de-
al doilea element din vector atunci:
         - Se dezarhivează fişierul ZIP descărcată şi rezultă un fişier
XML.
         - Se validează fişierul XML cu schema de validare XSD
corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se parcurge fişierul şi se actualizează tabela de erori
din baza de date.
         - Altfel se afişează mesaj de eroare "Fişier invalid".
   - Altfel se afişează mesaj de eroare de comunicaţie.
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

Numele fişierului de raportare identifică în mod unic o raportare efectuată, astfel încât alţi parametrii, cum ar fi tipul de furnizor, nu sunt necesari pentru această metodă. Aplicaţia client trebuie să ţină evidenţa fişierelor de raportare trimise pentru a putea cere răspunsurile procesate ale acestor fişiere.

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Serviciul pentru preluarea decontului

Acest serviciu este folosit pentru obţinerea fişierului de decont aferent unei perioade de raportare sau unei anumite raportări (în baza numărului de factură). În acest sens, serviciul va expune două metode, una pentru preluarea decontului pentru o perioadă de raportare, și alta pentru preluarea decontului pentru o anumită factură. Interogarea pe bază de factură este folosită în mod particular de furnizorii de medicamente (farmacii) sau de dispozitive medicale.

Datele vor fi disponibile după finalizarea procedurii de decontare din cadrul SIUI. Decontul este un raport (în format PDF) și este destinat consultării de către furnizor. Datele de pe raport nu vor fi preluate în aplicație.

Metoda *getRefund*

```
String[] getRefund (
        String partnerCategory,
        DateTime start,
        DateTime stop,
        String uic )
```

Metoda are patru parametri de intrare:

- parametrul ***partnerCategory*** de tip şir de caractere reprezintă codul categoriei de furnizor, lista valorilor permise fiind prezentată mai jos;
- parametrul ***start*** de tip dată calendaristică reprezintă data de început a perioadei pentru care se doreşte fişierul de decont;
- parametrul ***stop*** de tip dată calendaristică reprezintă data de sfârşit a perioadei pentru care se doreşte fişierul de decont;
- parametrul ***uic*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de decont, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

Dacă nu există un fişier de decont generat pentru furnizorul respectiv, metoda întoarce ***null***.

Metoda *getRefundForInvoice*

```
String[] getRefundForInvoice (
        String partnerCategory,
        String invoiceNumber,
        DateTime invoiceDate,
        String uic )
```

Metoda are patru parametri de intrare:

- parametrul ***partnerCategory*** de tip şir de caractere reprezintă codul tipului de furnizor, acelaşi ca pentru metoda de preluare decont dintr-o perioadă;
- parametrul ***invoiceNumber*** de tip şir de caractere reprezintă numărul de serie al facturii pentru care se doreşte fişierul de decont;

- parametrul ***invoiceDate*** de tip dată calendaristică reprezintă data facturii pentru care se doreşte fişierul de decont;

- parametrul ***uic*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de decont, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

Dacă nu există un fişier de decont generat pentru furnizorul respectiv, metoda întoarce ***null***.

Metoda *getRefundForPhysician*

```
String[] getRefundForPhysician (
        String partnerCategory,
        DateTime start,
        DateTime stop,
        String uic
        String stencil )
```

Metoda are cinci parametri de intrare:

- parametrul ***partnerCategory*** de tip şir de caractere reprezintă codul tipului de furnizor, lista valorilor permise fiind prezentată mai jos;

- parametrul ***start*** de tip dată calendaristică reprezintă data de început a perioadei pentru care se doreşte fişierul de decont;

- parametrul ***stop*** de tip dată calendaristică reprezintă data de sfârşit a perioadei pentru care se doreşte fişierul de decont;

- parametrul ***uic*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz.

- parametrul ***stencil*** de tip şir de caractere reprezintă codul de parafă al medicului pentru care se dorește fișierul de decont, în cazul cabinetelor medicale cu mai mulți medici titulari de contract, pentru care se calculează decontul separat.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de decont, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

Dacă nu există un fişier de decont generat pentru furnizorul respectiv, metoda întoarce ***null***.

Instrucțiuni de folosire

Aplicaţia client trebuie să folosească URL-ul rezultat pentru a descărca fişierul de decont. Valoarea celui de-al doilea parametru poate fi folosită pentru a verifica completitudinea

fişierului descărcat. Fişierul descărcat este o arhivă ZIP care conţine un fişier PDF cu sumele care vor fi decontate de casa de asigurări.

Prezentăm în continuare lista de valori admise pentru parametrul ***partnerCategory***:

Valoare parametru	Tip de furnizor corespunzător
MF	Medicină primară şi de familie
PRM	Centre de permanenţă
FARM	Farmacii (circuit deschis)
CLIN	Specialităţi clinice
PARA	Specialităţi paraclinice
STOM	Specialităţi stomatologice
AMB	Ambulanţe
MD	Dispozitive medicale
HC	Îngrijire la domiciliu
RECA	Recuperare - ambulatoriu
RECS	Recuperare - sanatorii
SPT	Spitale
NHP	P.N.S.
DIA	P.N.S. / Dializă publică
FSD	Dializă privată

NOTĂ

Nu toate categoriile de furnizori pot descărca un fişier de decont, de exemplu farmaciile cu circuit închis sau medicii cu convenţie de eliberare a certificatelor de concediu medical, deoarece fluxul de lucru specific nu implică decontări.

Un exemplu tipic de algoritm pentru preluarea fişierului de decont este:

```
Se apelează metoda getRefund cu parametrii corespunzători.
Dacă se întoarce un vector de şiruri de caractere de lungime 2 atunci:
   - Se consideră primul şir ca fiind url-ul pentru descărcarea
fişierului.
   - Se descarcă fişierul (care este o arhivă ZIP).
   - Dacă dimensiunea fişierului descărcat coincide cu valoarea celui de-
al doilea element din vector atunci:
         - Se dezarhivează arhiva descărcată şi rezultă un fişier PDF.
         - Se afişează conţinutul fişierului PDF folosind aplicaţia de
vizualizare instalată (ex. Acrobat Reader).
   - Altfel se afişează mesaj de eroare de comunicaţie.
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

De asemenea, accesul prin URL la arhivă este securizat, folosindu-se aceeaşi nume de utilizator şi parolă ca pentru accesul la metoda Web, precum şi certificatul digital pentru deschiderea conexiunii SSL.

Serviciul pentru sincronizarea deciziilor de acordare

Acest serviciu este folosit pentru sincronizarea informaţiilor referitoare la deciziile de aprobare ale unor categorii de servicii.

Metoda *getDecisions*

```
String[] getDecisions (
        String partnerCategory,
        String requestXml )
```

Metoda are doi parametri de intrare:

- parametrul ***partnerCategory*** de tip şir de caractere reprezintă codul categoriei de furnizor, lista valorilor permise fiind prezentată mai jos;
- parametrul ***requestXml*** de tip şir de caractere reprezintă conţinutul fişierului de cerere arhivat în formatul ***ZIP (JavaZip)*** şi codat ulterior în formatul ***Base64***.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de răspuns, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

Dacă nu există un fişier de raportare procesat cu numele dat, metoda întoarce ***null***.

Instrucţiuni de folosire

Aplicaţia client trebuie să folosească URL-ul rezultat pentru a descărca fişierul de răspuns. Dimensiunea fişierului poate fi folosită pentru a verifica completitudinea fişierului descărcat. Fişierul descărcat este o arhivă ZIP care conţine un fişier XML cu datele referitoare la deciziile cerute din SIUI.

Numele fişierului XML de cerere trebuie sa respecte formatul:

```
{Prefix} + "_" + {Cod} + "_" + {Data} + "_" + {Ora} + ".xml"
```

{Prefix} reprezintă un cod de identificare pentru tipul de furnizor, lista completă a acestor coduri fiind prezentată în tabelul de mai jos.

{Cod} reprezintă codul unic de identificare al furnizorului în sistem, CUI sau CNP, după caz.

Parametrii ***{Data}*** şi ***{Ora}*** reprezintă data şi ora la care a fost efectuată raportarea şi trebuie să apară în formatul "AAAALLZZ" pentru dată şi "OOMM", fără nici un separator.

Schema de validare pentru acest fişier, dar şi pentru fişierul de răspuns care conţine deciziile, este detaliată în anexele corespunzătoare fiecărei categorii de furnizor:

Valoare parametru	Valoare prefix fişier	Tip de furnizor corespunzător
MD	MD_SYNC	Dispozitive medicale
HC	HC_SYNC	Îngrijire la domiciliu

Un exemplu tipic de algoritm pentru preluarea şi sincronizarea deciziilor este:

```
Se pregătesc datele pentru raportare:
   - Se generează fişierul cerere în format XML corespunzător deciziei
selectate.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
   - Se arhivează fişierul XML folosind algoritmul ZIP.
   - Se codifică conţinutul arhivei folosind codarea Base64.
Se apelează metoda getDecisions cu parametrii corespunzători.
Dacă se întoarce un vector de şiruri de caractere de lungime 2 atunci:
   - Se consideră primul şir ca fiind url-ul pentru descărcarea
fişierului.
   - Se descarcă fişierul (care este o arhivă ZIP).
   - Dacă dimensiunea fişierului descărcat coincide cu valoarea celui de-
al doilea element din vector atunci:
         - Se dezarhivează arhiva descărcată şi rezultă un fişier XML.
         - Se validează fişierul XML cu schema de validare XSD
corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se parcurge fişierul şi se actualizează tabela de
decizii din baza de date.
         - Altfel se afişează mesaj de eroare "Fişier invalid".
   - Altfel se afişează mesaj de eroare de comunicaţie.
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

Această metodă are implementări doar pentru două categorii de furnizori, cei de dispozitive medicale şi servicii de îngrijire la domiciliu, pentru care este necesară obţinerea unei aprobări speciale (decizie) din partea casei de asigurări în vederea eliberării dispozitivului sau acordării serviciului de îngrijire la domiciliu.

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie). De asemenea, accesul prin URL la arhivă este securizat, folosindu-se aceeaşi nume de utilizator şi parolă ca pentru accesul la metoda Web.

Serviciul pentru verificarea calităţii de asigurat

Acest serviciu este folosit pentru verificarea online a calităţii de asigurat pe baza CNP-ului unui beneficiar de servicii medicale sau farmaceutice.

Aplicaţiile de raportare vor folosi acest serviciu pentru a verifica starea de asigurat a unui beneficiar, care vor putea astfel asista operatorul recompletând informaţiile corespunzătoare sau vor afişa mesaje de avertizare în cazul în care se înregistrează servicii pentru persoane neasigurate.

Metoda *getInsured*

```
String getInsured (
        String pid,
        Date requestDate )
```

Metoda are doi parametri de intrare:

- parametrul ***pid*** de tip şir de caractere reprezintă CNP-ul unui beneficiar;
- parametrul ***requestDate*** de tip dată calendaristică reprezintă data la care se dorește verificarea calității de asigurat, de exemplu data curentă sau data efectuării serviciului.

Metoda întoarce ca răspuns un şir de caractere reprezentând conţinutul unui fişier în format XML care conţine următoarele informaţii:

- Un cod numeric de răspuns indicând dacă beneficiarul este asigurat sau nu, dacă figurează ca decedat în sistem, dacă nu este înregistrat în sistem sau dacă CNP-ul nu este corect.
- Lista categoriilor active la data interogării

Observaţie: În cazul unei erori întâlnite în sistem la procesarea cererii se va întoarce un cod numeric de răspuns (-1) precum şi o descriere a erorii.

Metoda *getInsuredByCID*

```
String getInsuredByCID (
        String cid,
        Date requestDate)
```

Metoda are doi parametri de intrare:

- parametrul ***cid*** de tip şir de caractere reprezintă codul de asigurat al unui beneficiar;
- parametrul ***requestDate*** de tip dată calendaristică reprezintă data la care se dorește verificarea calității de asigurat, de exemplu data curentă sau data efectuării serviciului.

Metoda întoarce ca răspuns un şir de caractere reprezentând conţinutul unui fişier în format XML care conţine următoarele informaţii:

- Un cod numeric de răspuns indicând dacă beneficiarul este asigurat sau nu, dacă figurează ca decedat în sistem, dacă nu este înregistrat în sistem sau dacă CNP-ul nu este corect.
- Lista categoriilor active la data interogării

Observaţie: În cazul unei erori întâlnite în sistem la procesarea cererii se va întoarce un cod numeric de răspuns (-1) precum şi o descriere a erorii.

Instrucţiuni de folosire

Aplicaţia de raportare trebuie să proceseze fişierul de răspuns şi să afişeze un mesaj sugestiv pentru utilizator cu privire la starea de asigurat a persoanei respective. Dacă este cazul aplicaţia va pre-completa câmpurile corespunzătoare categoriei de asigurat, selectând categoria cea mai favorabilă pacientului din lista transmisă din SIUI.

Este de preferat ca aplicaţia de raportare să realizeze validarea de corectitudine a CNP-ului, algoritmul fiind arhicunoscut, pentru a nu supraîncărca sistemul cu cereri inutile.

Schema de validare pentru fişierul de răspuns este detaliată în anexele corespunzătoare fiecărei categorii de furnizor.

Un exemplu tipic de algoritm pentru verificarea categoriei de asigurat este:

```
Utilizatorul introduce CNP-ului unui pacient sau selectează un pacient
dintr-o listă derulantă.
Aplicaţia validează corectitudinea CNP-ului:
   - Dacă CNP-ul este incorect se afişează un mesaj de avertizare.
   - Altfel se continuă verificarea online:
Aplicaţia apelează metoda getInsured folosind CNP-ul respectiv şi data
serviciului ca parametri.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se parcurge fişierul şi se afişează un mesaj
corespunzător stării de asigurat.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Metoda poate fi apelată de orice categorie de furnizor, motiv pentru care nu apare parametrul de apel corespunzător prezent în celelalte metode ale serviciilor Web SIUI.

Serviciul pentru validarea mişcărilor de capitaţie

Acest serviciu este folosit pentru validarea unei cereri de înscriere sau ieşire a unui pacient pe lista unui medic de familie, însoțită de motivația operației.

Metoda *validateEnlisted*

```
String validateEnlisted ( String enlistedXml )
```

Metoda are un singur parametru de intrare:

- parametrul ***enlistedXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML.

Metoda întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine următoarele informaţii:

- O structură similară cu cea raportată, conţinând fiecare identificator de înregistrare transmisă însoţit de starea validării (validat/nevalidat)
- Lista erorilor sau avertizărilor pentru fiecare înregistrare raportată, în caz că acestea au fost depistate
- Ștampila de timp la momentul emiterii răspunsului

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru validarea mişcărilor de capitaţie este:

```
Utilizatorul adaugă sau elimină un pacient din lista de înscrişi:
   - Aplicaţia generează fişierul cerere în format XML conţinând
informaţiile referitoare la pacient, operaţia efectuată şi motivul
acesteia.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
Aplicaţia apelează metoda validateEnlisted trimiţând conţinutul
fişierului.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se parcurge fişierul şi se afişează un mesaj
corespunzător rezultatului validării.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Serviciul pentru validarea serviciilor și investigațiilor medicale

Acest serviciu este folosit pentru validarea serviciilor prestate în aplicaţie pe măsură ce acestea sunt înregistrate, înainte de încheierea perioadei re raportare.

Metoda *validateReport*

```
String validateReport (
        String reportXml,
        String reportType,
        String requestType )
```

Metoda are trei parametri de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML.
- parametrul ***reportType*** de tip şir de caractere reprezintă codul tipului de furnizor, lista valorilor permise fiind prezentată mai jos;
- parametrul ***requestType*** de tip şir de caractere reprezintă codul tipului de cerere de validare transmisă, lista valorilor permise fiind prezentată mai jos;

Metoda întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine următoarele informaţii:

- O structură similară cu cea raportată, conţinând fiecare identificator de înregistrare transmisă însoţit de starea validării (validat/nevalidat)
- Lista erorilor sau avertizărilor pentru fiecare înregistrare raportată, în caz că acestea au fost depistate
- Ştampila de timp la momentul emiterii răspunsului

Conţinutul şi formatul datelor transmise este specific fiecărui tip de furnizor şi va fi descris în detaliu în anexele care însoțesc acest document. Ca regulă generală, datele transmise din aplicaţia de raportare către SIUI vor fi validate iar serviciul Web va întoarce un răspuns cu privire la rezultatul validării serviciului medical raportat.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru validarea unei rețete este:

```
Utilizatorul adaugă o rețetă în baza de date:
   - Aplicaţia generează fişierul cerere în format XML conţinând
informaţiile referitoare la rețetă și medicamentele prescrise.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
Aplicaţia apelează metoda validateReport trimiţând conţinutul fişierului
însoțit de tipul raportării.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează un mesaj
corespunzător rezultatului validării.
```

```
                - Aplicaţia asociază şi păstrează rezultatul validării,
afişând înregistrarea respectivă în mod distinct.
          - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepţia returnată sau un mesaj de eroare de
comunicaţie.
```

Prezentăm în continuare lista de valori admise pentru parametrul ***reportType***:

Valoare parametru	**Valoare prefix fişier**	**Tip de furnizor corespunzător**
MF	MF	Medicină primară şi de familie
PRM	PRM	Centre de permanenţă
FARMD	FARMD	Farmacii (circuit deschis)
FARMI	FARMI	Farmacii (circuit închis)
CLIN	CLIN	Specialităţi clinice
PARA	PARA	Specialităţi paraclinice
STOM	STOM	Specialităţi stomatologice
AMB	AMB	Ambulanţe
MD	MD	Dispozitive medicale
HC	HC	Îngrijire la domiciliu
RECA	RECA	Recuperare - ambulatoriu
RECS	RECS	Recuperare - sanatorii
DIA	DIA	Dializă privată
NHPDIA	DIA	P.N.S. / Dializă publică
NHPREP	NHPREP	P.N.S. / Raportare de indicatori P.N.S.
NHPCJ	NHPCJ	P.N.S. / Cereri justificative (facturi şi ordine de plată)
SPT_ACUT	SPT_ACUT	Spitale / Raportare de cazuri acute (internări)
SPT_CHR	SPT_CHR	Spitale / Raportare de cazuri cronice
SPT_DRG	SPT_DRG	Spitale / Raportare D.R.G.
SPT_SPZ	SPT_SPZ	Spitale / Raportare spitalizare de zi
SPT_PAL	SPT_PAL	Spitale / Raportare paliative
CM	CM	Concedii medicale

Prezentăm în continuare lista de valori admise pentru parametrul ***requestType***:

Valoare parametru	**Tip de cerere de validare**
RQ_PRESCRIPTION	Cerere de validare reţetă prescrisă de medic
RQ_REFERRAL	Cerere de validare bilet de trimitere emis
RQ_CHRONICS	Cerere de validare a bolnavilor cronici aflaţi în evidenţă
RQ_MF_ENLISTED	Cerere de validare a mişcării înscrişilor pe listele medicilor de familie
RQ_MF_SERVICES	Cerere de validare a serviciilor prestate de medicii de familie
RQ_SPT_ACUTE	Cerere de validare a cazurilor acute de spitalizare
RQ_SPT_CHRONIC	Cerere de validare a cazurilor cronice de spitalizare
RQ_SPT_ANALYTIC	Cerere de validare a raportării analitice a spitalelor
RQ_SPT_DRG	Cerere de validare a raportării DRG a spitalelor
RQ_SPT_HC_REC	Cerere de validare a recomandărilor de servicii de îngrijire la domiciliu
RQ_SPT_MD_REC	Cerere de validare a recomandărilor de acordare de dispozitive medicale

Valoare parametru	Tip de cerere de validare
RQ_SPT_PAL	Cerere de validare a cazurilor paliative de spitalizare
RQ_PHM_PRESCRIPTION	Cerere de validare a unei reţete eliberate în farmacie
RQ_PHM_HR	Cerere de validare a unei foi de condică din farmacii cu circuit închis
RQ_PRM	Cerere de validare a serviciilor prestate în centre de permanenţă
RQ_MD	Cerere de validare a dispozitive medicale acordate
RQ_HC	Cerere de validare a servicii de îngrijire la domiciliu
RQ_AMB_SRV	Cerere de validare a servicii de urgență sau transport cu ambulanța
RQ_AMB_WT	Cerere de validare/raportare a timpilor de așteptare a ambulanțelor
RQ_DIA_SRV	Cerere de validare a serviciilor de dializă privată
RQ_NHP_DIA_SRV	Cerere de validare a serviciilor de dializă publică
RQ_RECA_SRV	Cerere de validare a serviciilor de recuperare ambulatorie
RQ_RECS_SRV	Cerere de validare a serviciilor de recuperare în sanatorii
RQ_CLIN_SRV	Cerere de validare a serviciilor clinice de specialitate
RQ_PARA_SRV	Cerere de validare a serviciilor paraclinice (investigații de laborator)
RQ_STOM_SRV	Cerere de validare a serviciilor stomatologice şi dentare
RQ_NHP_GC	Cerere de validare a consumului de materiale decontate din PNS
RQ_NHP_IND	Cerere de validare/raportare a indicatorilor PNS
RQ_NHP_TS	Cerere de validare a schemelor terapeutice recomandate în cadrul PNS
RQ_NHP_INV	Cerere de validare a facturilor decontate din PNS
RQ_NHP_OP	Cerere de validare a plăților decontate din PNS
RC_CM	Cerere de validare certificat medical prescris de medic

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Orice serviciu pre-validat poate fi modificat ulterior de către furnizor, în intervalul de timp alocat raportărilor, conform legislației în vigoare, dar nu mai târziu de întocmirea deconturilor către furnizori. Pentru re-validarea după modificarea a serviciului medical efectuat aplicaţia de raportare va trebui să transmită acelaşi identificator de serviciu, în caz contrar operaţia va tratată ca o adăugare şi va fi invalidată (serviciul medical efectuat îşi păstrează identificatorul unic indiferent de câte ori este modificat).

Serviciul pentru validarea rețetelor prescrise

Acest serviciu permite raportarea reţetelor prescrise de către medici. Medicul va completa datele referitoare la rețetă în aplicaţia de raportare, la salvarea reţetei se va apela serviciul Web prin care se va transmite pentru validare către SIUI rețeta.

NOTĂ

Acest serviciu este destinat validării reţetelor clasice pe formulare cu regim special şi va fi păstrat pentru compatibilitate până la eliminarea completă a acestor reţete. Pentru reţetele electronice trebuie folosite serviciile specifice expuse de SIUI+PE.

Metoda *validatePrescription*

```
String validatePrescription (
        String reportXml,
        String reportType )
```

Metoda are doi parametri de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML;
- parametrul ***reportType*** de tip şir de caractere reprezintă codul tipului de furnizor, lista valorilor permise fiind prezentată mai jos.

Metoda întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine rezultatul operaţiunii de validare.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru validarea unei rețete prescrise de medic este:

```
Utilizatorul adaugă o rețetă în baza de date:
   - Aplicaţia generează fişierul cerere în format XML conţinând
informaţiile referitoare la rețetă şi medicamentele prescrise.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
Aplicaţia apelează metoda validatePrescription trimiţând conţinutul
fişierului generat.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează un mesaj
corespunzător rezultatului validării.
               - Aplicaţia asociază şi păstrează rezultatul validării,
afişând înregistrarea respectivă în mod distinct.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepția returnată sau un mesaj de eroare de
comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Structura fişierul permite transmiterea mai multor înregistrări simultan, de exemplul la cerea utilizatorului, după ce acesta a finalizat operarea mai multor rețete, sau în mod automat la revenirea conexiunii online după o perioadă de lucru offline.

Modificarea unei reţete prescrise se poate face doar de către medicul prescriptor atât timp cât reţeta nu a fost eliberată de către furnizorul de servicii farmaceutice. În cazul în care un medic prescriptor aflat on-line va dori să modifice o reţetă care a fost eliberată, nu va putea salva modificările şi va primi un mesaj care îl va avertiza că reţeta a fost eliberată.

Metoda ***validareReport*** se poate folosi în locul aceste metode, dacă se utilizează parametrii corespunzători, rezultatul validării fiind acelaşi.

Serviciul pentru validarea biletelor de trimitere

Acest serviciu permite raportarea biletelor de trimitere pentru specialităţi clinice sau investigaţii de laborator de către un medic emitent. Medicul va completa datele biletului de trimitere în aplicaţia de raportare, la salvarea biletului de trimitere se va apela serviciul Web prin care se va transmite pentru validare către SIUI biletul de trimitere.

Metoda *validateClinicReferral*

```
String validateClinicReferral (
        String reportXml,
        String reportType )
```

Metoda are doi parametri de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML;
- parametrul ***reportType*** de tip şir de caractere reprezintă codul tipului de furnizor, lista valorilor permise fiind prezentată mai jos.

Metoda întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine rezultatul operaţiunii de validare.

Metoda *validateLabReferral*

```
String validateLabReferral (
        String reportXml,
        String reportType )
```

Metoda are doi parametri de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML;

- parametrul ***reportType*** de tip şir de caractere reprezintă codul tipului de furnizor, lista valorilor permise fiind prezentată mai jos.

Metoda întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine rezultatul operaţiunii de validare.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru validarea unui bilet de trimitere este:

```
Utilizatorul adaugă un bilet de trimitere în baza de date:
   - Aplicaţia generează fişierul cerere în format XML conţinând
informaţiile referitoare la biletul de trimitere şi diagnosticul
prezumtiv.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
Aplicaţia apelează metoda validateClinicReferral sau validateLabReferral,
după caz, trimiţând ca parametru conţinutul fişierului generat.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează un mesaj
corespunzător rezultatului validării.
               - Aplicaţia asociază şi păstrează rezultatul validării,
afişând înregistrarea respectivă în mod distinct.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepţia returnată sau un mesaj de eroare de
comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Structura fișierul permite transmiterea mai multor înregistrări simultan, de exemplul la cerea utilizatorului, după ce acesta a finalizat operarea mai multor bilete de trimitere, sau în mod automat la revenirea conexiunii online după o perioadă de lucru offline.

Metoda ***validareReport*** se poate folosi în locul acestor metode, dacă se utilizează parametrii corespunzători, rezultatul validării fiind acelaşi.

Serviciul pentru validarea certificatelor medicale

Acest serviciu permite unui medic prescriptor sa raporteze concediile medicale prescrise. Serviciul va valida concediul medical și va informa medicul prescriptor despre rezultatul

validării. Certificatele medicale astfel raportate vor fi stocate într-o bază de date pentru realizarea verificărilor de unicitate a certificatelor medicale și a verificărilor încrucișate conform normelor în vigoare.

Metoda *validateSickLeave*

```
String validateSickLeave ( String reportXml )
```

Metoda are un singur parametru de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML.

Metoda întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine rezultatul operaţiunii de validare.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru validarea unui certificat medicale este:

```
Utilizatorul adaugă un certificat medical în baza de date:
   - Aplicaţia generează fişierul cerere în format XML conţinând
informaţiile referitoare la certificatul medical, perioadă și diagnostic.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
Aplicaţia apelează metoda validateSickLeave trimiţând ca parametru
conţinutul fişierului generat.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează un mesaj
corespunzător rezultatului validării.
               - Aplicaţia asociază şi păstrează rezultatul validării,
afişând înregistrarea respectivă în mod distinct.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepția returnată sau un mesaj de eroare de
comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepție).

Structura fișierul permite transmiterea mai multor înregistrări simultan, de exemplul la cerea utilizatorului, după ce acesta a finalizat operarea mai multor certificate medicale, sau în mod automat la revenirea conexiunii online după o perioadă de lucru offline.

Metoda ***validareReport*** se poate folosi în locul aceste metode, dacă se utilizează parametrii corespunzători, rezultatul validării fiind același.

Serviciul pentru validarea reţetelor emise de farmacii

Acest serviciu permite unei farmacii să verifice compatibilitatea dintre medicamentele prescrise si cele eliberate (calitativ şi cantitativ) precum şi validarea încadrării în plafonul de decontare contractat cu Casa de Asigurări. SIUI va returna un mesaj prin care farmacistul este înștiințat despre rezultatul operaţiunii de validare a eliberării medicamentelor.

NOTĂ

Acest serviciu este destinat validării reţetelor clasice pe formulare cu regim special şi va fi păstrat pentru compatibilitate până la eliminarea completă a acestor reţete. Pentru reţetele electronice trebuie folosite serviciile specifice expuse de SIUI+PE.

Metoda *validateFarmacyDrugs*

```
String validateFarmacyDrugs ( String reportXml )
```

Metoda are un singur parametru de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML.

Metoda întoarce un şir de caractere reprezentând fişierul de răspuns în format XML.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru validarea unei reţete eliberate în farmacie este:

```
Utilizatorul adaugă un certificat medical în baza de date:
   - Aplicaţia generează fişierul cerere în format XML conţinând
informaţiile referitoare la reţetă, pacient şi medicamentele eliberate.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
Aplicaţia apelează metoda validateFarmacyDrugs trimiţând ca parametru
conţinutul fişierului generat.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează un mesaj
corespunzător rezultatului validării.
               - Aplicaţia asociază şi păstrează rezultatul validării,
afişând înregistrarea respectivă în mod distinct.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepţia returnată sau un mesaj de eroare de
comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Structura fișierul permite transmiterea mai multor înregistrări simultan, de exemplul la cerea utilizatorului, după ce acesta a finalizat operarea mai multor reţete eliberate, sau în mod automat la revenirea conexiunii online după o perioadă de lucru offline.

O rețetă poate fi eliberată, total sau parţial, de o singură farmacie. După eliberare reţeta trece în starea „*eliberată*" şi nu mai este disponibilă pentru alte farmacii. Orice modificare a unei reţete eliberate de către o farmacie poate fi făcută exclusiv de farmacia în cauză. Toate aceste modificări sunt salvate într-un log pentru posibilitatea auditării ulterioare.

Serviciul pentru consultarea reţetelor prescrise

Acest serviciu este folosit pentru a permite unei farmacii să vizualizeze reţetele prescrise de medici şi să elibereze medicamentele aferente reţetei.

NOTĂ

Acest serviciu este destinat validării reţetelor clasice pe formulare cu regim special şi va fi păstrat pentru compatibilitate până la eliminarea completă a acestor reţete. Pentru reţetele electronice trebuie folosite serviciile specifice expuse de SIUI+PE.

Metoda *getPrescription*

```
String getPrescription (
        String serial,
        String number,
        String pid,
        String stencil )
```

Metoda are patru parametri de intrare:

- parametrul ***serial*** de tip şir de caractere reprezintă seria rețetei;
- parametrul ***number*** de tip şir de caractere reprezintă numărul rețetei;
- parametrul ***pid*** de tip şir de caractere reprezintă CNP-ul beneficiarului rețetei;
- parametrul ***stencil*** de tip şir de caractere reprezintă numărul de parafă al medicului emitent;

Metoda întoarce şir de caractere reprezentând fişierul de răspuns în format XML.

Aplicațiile de raportare vor avea posibilitatea de implementare a unor funcționalităţi de preluare automată a conținutului acestor documente în format electronic către SIUI. Astfel o farmacie poate apela serviciul Web pentru a descărca o rețetă prescrisă în scopul de a elibera medicamentele aferente.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru consultarea unei reţete prescrise este:

```
Utilizatorul introduce datele necesare (vezi lista de parametri):
Aplicaţia apelează metoda getPrescription trimiţând ca parametri datele
introduse ce utilizator.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează conţinutul
reţetei cerute, eventual se precompletează datele în ecranul de
introducere reţete.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepţia returnată sau un mesaj de eroare de
comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Numai reţetele raportate ca validate de SIUI vor fi disponibile pentru interogare de către furnizorii de servicii farmaceutice, aceştia vor identifica reţetele prescrise în vederea eliberării medicaţiei după combinaţia de câmpuri: serie şi număr reţetă, CNP pacient şi parafă medic prescriptor.

Serviciul pentru consultarea biletelor de trimitere

Acest serviciu este folosit pentru consultarea biletelor de trimitere pentru specialități clinice sau investigaţii de laborator validate de SIUI de către furnizorii de servicii medicale care prestează servicii în baza unui bilet de trimitere.

Metoda *getClinicReferral*

```
String getClinicReferral (
      String serial,
      String number,
      String pid,
      String stencil )
```

Metoda are patru parametri de intrare:

- parametrul ***serial*** de tip şir de caractere reprezintă seria biletului de trimitere;
- parametrul ***number*** de tip şir de caractere reprezintă numărul biletului de trimitere;

- parametrul ***pid*** de tip şir de caractere reprezintă CNP-ul beneficiarului biletului de trimitere;

- parametrul ***stencil*** de tip şir de caractere reprezintă numărul de parafă al medicului emitent;

Metoda întoarce şir de caractere reprezentând fişierul de răspuns în format XML.

Metoda *getLabReferral*

```
String getLabReferral (
        String serial,
        String number,
        String pid,
        String stencil )
```

Metoda are patru parametri de intrare:

- parametrul ***serial*** de tip şir de caractere reprezintă seria biletului de trimitere;

- parametrul ***number*** de tip şir de caractere reprezintă numărul biletului de trimitere;

- parametrul ***pid*** de tip şir de caractere reprezintă CNP-ul beneficiarului biletului de trimitere;

- parametrul ***stencil*** de tip şir de caractere reprezintă numărul de parafă al medicului emitent;

Metoda întoarce şir de caractere reprezentând fişierul de răspuns în format XML.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru consultarea unei reţete prescrise este:

```
Utilizatorul introduce datele necesare (vezi lista de parametri):
Aplicaţia apelează metoda getClinicReferral sau getLabReferral, după caz,
trimiţând ca parametri datele introduse ce utilizator.
Dacă SIUI întoarce un şir de caractere ca răspuns, aplicaţia îl salvează
într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează conţinutul
biletului de trimitere cerut, eventual se precompleteaza datele în ecranul
de introducere servicii.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepţia returnată sau un mesaj de eroare de
comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Serviciul pentru validarea reţetelor electronice

Acest serviciu este folosit pentru validarea reţetelor electronice prescrise de medici sau eliberate de farmacişti.

Utilizatorul (medic sau farmacist) va completa datele referitoare la rețeta electronică în aplicaţia de raportare, la salvarea reţetei aplicaţia va apela serviciul Web prin care se va transmite pentru validare către sistemul central rețeta reprezentată în format XML.

Serviciul expune două metode, una specifică medicului, iar cealaltă farmacistului. Ambele metode validează reţeta din punct de vedere medical pentru ca pacientul să poată beneficia în acest fel de eventualele avertizări pe care sistemul le-ar putea emite cu privire la interacţiuni între medicamentele prescrise sau între acestea şi anumite alergii sau boli cronice ale pacientului. În acest mod, chiar dacă medicul a prescris reţeta offline, farmacistul va avea acces la setul de reguli specific medicului pentru a informa pacientul despre posibile contraindicaţii.

Metoda *processPrescribedPrescription*

```
String processPrescribedPrescription (
        String reportXml )
```

Metoda are un singur parametru de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML, care se validează cu schema ***PhysicianDrugPERequest.xsd***.

Metoda este destinată procesării reţetelor prescrise de medici şi întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine rezultatul operaţiunii de validare online, şi se validează cu schema ***PhysicianDrugPEResponse.xsd***. Fişierul are un conţinut similar celui trimis de aplicaţie spre procesare, dar conţine la nivel de reţetă (date pacient) dar şi la nivel de detalii (medicamente) eventuale avertizări emise de sistem în cazul nerespectării unor norme de prescriere.

Metoda *processIssuedPrescription*

```
String processIssuedPrescription (
        String reportXml )
```

Metoda are un singur parametru de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML, care se validează cu schema ***PharmacyDrugsPERequest.xsd***.

Metoda este destinată procesării reţetelor eliberate de farmaciile cu circuit deschis şi întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine rezultatul operaţiunii de validare online, şi se validează cu schema ***PharmacyDrugsPEResponse.xsd***. Fişierul are un conţinut similar celui trimis de aplicaţie spre procesare, dar conţine la nivel de reţetă (date pacient) dar şi la nivel de detalii (medicamente) eventuale avertizări emise de sistem în cazul nerespectării unor norme de eliberare.

Metoda *processHospitalPrescription*

```
String processHospitalPrescription (
        String reportXml )
```

Metoda are un singur parametru de intrare:

- parametrul ***reportXml*** de tip şir de caractere reprezintă conţinutul fişierului de raportare în format XML, care se validează cu schema ***HospitalRegisterPERequest.xsd***.

Metoda este destinată procesării reţetelor eliberate de farmaciile cu circuit închis şi întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine rezultatul operaţiunii de validare online, şi se validează cu schema ***HospitalRegisterPEResponse.xsd***. Fişierul are un conţinut similar celui trimis de aplicaţie spre procesare, dar conţine la nivel de reţetă (date pacient) dar şi la nivel de detalii (medicamente) eventuale avertizări emise de sistem în cazul nerespectării unor norme de eliberare.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru validarea unei rețete electronice prescrise de medic este:

```
Utilizatorul (medic) adaugă o rețetă în baza de date:
   - Aplicaţia generează fişierul cerere în format XML conţinând
informaţiile referitoare la rețetă și medicamentele prescrise.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
Dacă medicul are semnătură electronică extinsă (certificat digital
calificat):
   - Se semnează electronic fişierul XML, folosind standardul CMS
(RFC5652).
   - Se codifică rezultatul folosind codarea Base64.
Se apelează metoda processPrescribedPrescription trimiţând conţinutul
fişierului generat.
Dacă sistemul întoarce un şir de caractere ca răspuns, aplicaţia îl
salvează într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează un mesaj
corespunzător rezultatului validării.
```

```
                - Aplicaţia asociază şi păstrează rezultatul validării,
afişând înregistrarea respectivă în mod distinct.
          - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepţia returnată sau un mesaj de eroare de
comunicaţie.
```

Un exemplu tipic de algoritm pentru validarea unei rețete electronice eliberate în farmacie este:

```
Utilizatorul (farmacist) adaugă o reţetă în baza de date, prin scanarea
codului de bare 2D sau prin completare manuală:
   - Aplicaţia generează fişierul cerere în format XML conţinând
informaţiile referitoare la reţetă, pacient şi medicamentele eliberate.
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
Dacă medicul are semnătură electronică extinsă (certificat digital
calificat):
   - Se semnează electronic fişierul XML, folosind standardul CMS
(RFC5652).
   - Se codifică rezultatul folosind codarea Base64.
Se apelează metoda processIssuedPrescription trimiţând ca parametru
conţinutul fişierului generat.
Dacă sistemul întoarce un şir de caractere ca răspuns, aplicaţia îl
salvează într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul XML şi se afişează un mesaj
corespunzător rezultatului validării.
               - Aplicaţia asociază şi păstrează rezultatul validării,
afişând înregistrarea respectivă în mod distinct.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează excepţia returnată sau un mesaj de eroare de
comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepție).

Pentru semnarea digitală a unui fișier în vederea procesării în SIUI este necesară deținerea unui certificat digital calificat *X.509* emis de unul din furnizorii acreditați de servicii de certificare din România. Perechea de chei aferentă certificatului trebuie să fie de tip *RSA*.

Fișierele semnate cu certificatul digital *X.509*, folosind algoritmul *SHA-1*, se transmit către SIUI folosind formatul *CMS* („*Cryptographic Message Syntax*”) publicat în *RFC-5652* de către *IETF* („*Internet Engineering Task Force”*) (vezi http://tools.ietf.org/html/rfc5652).

Descrierea algoritmului *SHA* („*Secure Hash Algorithm*”) este publicată de către National Institute of Standards and Technology (NIST) în *Digital Signature Standard FIPS 186-2.*

Semnarea electronică a fișierului XML este necesară doar în cazul transmiterii electronice online a acestuia către SIUI, și este opțională în prima fază de implementare a Sistemului

Informatic pentru Prescripţia Electronică, aşa cum este prevăzut în normele metodologice ale CNAS.

NOTĂ

*Reţetele offline se transmit spre procesare folosind aceleaşi metode, dar completând atributul **reportedOnline="0"**, cu semnificaţia "OFFLINE: Prescrisă offline de către medic. Validată online de farmacie".*

Modificarea unei reţete prescrise se poate face doar de către medicul prescriptor atât timp cât reţeta nu a fost încă tipărită de acesta, ulterior ea poate fi anulată, dar nu şi după ce a fost eliberată de către furnizorul de servicii farmaceutice. În cazul în care un medic prescriptor aflat on-line va dori să modifice o reţetă care a fost eliberată, nu va putea salva modificările şi va primi un mesaj care îl va avertiza că reţeta a fost eliberată.

O rețetă electronică poate fi eliberată, total sau parţial, de una sau mai multe farmacii. După eliberarea completă reţeta trece în starea „*eliberată*” şi nu mai este disponibilă pentru alte farmacii. Dacă reţeta nu a fost eliberată complet atunci ea trece în starea „*eliberată parţial*”, medicamentele neeliberate încă fiind disponibile pentru alte farmacii, cu condiţia ca pacientul să se prezinte la aceste farmacii pentru a ridica medicamentele.

NOTĂ

Nu se poate elibera parţial un anumit medicament, ci doar medicamente diferite (poziţii diferite de pe reţetă). Eliberarea parţială este valabilă doar pentru reţetele electronice prescrise online care poartă semnătură electronică.

Serviciul pentru anularea reţetelor electronice

Acest serviciu este folosit pentru anularea reţetelor electronice prescrise de medici, în cazul în care se constată ulterior o greşeală de întocmire sau o schimbare în starea de sănătate a pacientului, ceea ce necesită emiterea unei noi reţete.

Este permisă anularea unei reţete doar de către medicul care a prescris-o, acesta selectând rețeta electronică introdusă anterior în aplicaţia de raportare iar apoi opţiunea de anulare. Aplicaţia transmite către sistemul central cererea de anulare.

În cazul în care medicul a prescris reţeta electronică *offline*, iar farmacistul constată la raportarea în sistem o neregulă care poate duce la anularea reţetei, atunci va contacta medicul sau îl va îndruma pe pacient către acesta, deoarece nu va putea elibera medicamente în baza reţetei respective.

Serviciul poate fi utilizat prin apelarea metodelor ***cancelPrescribedPrescription***, ***cancelReleasedPrescription*** sau ***cancelReleasedHospitalPrescription*** specifice medicilor, respectiv farmaciilor cu circuit deschis şi cu circuit închis, aşa cum este descris în continuare:

Metoda *cancelPrescribedPrescription*

```
Integer cancelPrescribedPrescription (
        String providerCode,
        String physicianStencilNo,
        String contractNo,
        String contractType,
        String insuranceHouseCode,
        String series,
        String no,
        Date prescriptionDate,
        String cancellationReason )
```

Metoda are nouă parametri de intrare:

- parametrul ***providerCode*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;
- parametrul ***physicianStencilNo*** de tip şir de caractere reprezintă numărul de parafă al medicului emitent;
- parametrul ***contractNo*** de tip şir de caractere reprezintă numărul contractului dintre furnizor şi Casa de Asigurări;
- parametrul ***contractType*** de tip şir de caractere reprezintă codul tipului de contract, lista valorilor permise fiind prezentată mai jos;
- parametrul ***insuranceHouseCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări.
- parametrul ***series*** de tip şir de caractere reprezintă seria rețetei;
- parametrul ***no*** de tip şir de caractere reprezintă numărul rețetei;
- parametrul ***prescriptionDate*** de tip dată calendaristică reprezintă data la care reţeta a fost prescrisă de medic;
- parametrul ***cancellationReason*** de tip şir de caractere reprezintă motivaţia anulării (text liber).

Metoda întoarce o valoare întreagă indicând dacă cererea a fost procesată cu succes, caz în care valoarea este ***1***, altfel apelul întorcându-se cu eroare.

Metoda *cancelReleasedPrescription*

```
Integer cancelReleasedPrescription (
        String providerCode,
        String workplaceCode,
        String insuranceHouseCode,
        String series,
        String no,
        Integer fractionNo,
        String cancellationReason )
```

Metoda are opt parametri de intrare:

- parametrul ***providerCode*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;
- parametrul ***workplaceCode*** de tip şir de caractere reprezintă codul punctului de lucru al farmaciei (acest parametru trebuie sa aibă valoarea din contractul încheiat cu CAS, iar dacă farmacia nu are punct de lucru se va trimite string vid ***""***);
- parametrul ***insuranceHouseCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări.
- parametrul ***series*** de tip şir de caractere reprezintă seria rețetei;
- parametrul ***no*** de tip şir de caractere reprezintă numărul rețetei;
- parametrul ***fractionNo*** de tip număr întreg reprezintă numărul de ordine al fracţiei, în cazul rețetelor eliberate fracţionat farmacia/punctul de lucru (pentru o eliberare integrală acest parametru va avea valoarea ***1***);
- parametrul ***cancellationReason*** de tip şir de caractere reprezintă motivaţia anulării (text liber).

Metoda întoarce o valoare întreagă indicând dacă cererea a fost procesată cu succes, caz în care valoarea este ***1***, altfel apelul întorcându-se cu eroare.

Metoda *cancelReleasedHospitalPrescription*

```
Integer cancelReleasedHospitalPrescription (
        String providerCode,
        String workplaceCode,
        String insuranceHouseCode,
        String series,
        String no,
        Integer fractionNo,
        String cancellationReason )
```

Metoda are opt parametri de intrare:

- parametrul ***providerCode*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;

- parametrul ***workplaceCode*** de tip şir de caractere reprezintă codul punctului de lucru al farmaciei (acest parametru trebuie sa aibă valoarea din contractul încheiat cu CAS, iar dacă farmacia nu are punct de lucru se va trimite string vid ***""***);

- parametrul ***insuranceHouseCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări.

- parametrul ***series*** de tip şir de caractere reprezintă seria rețetei;

- parametrul ***no*** de tip şir de caractere reprezintă numărul rețetei;

- parametrul ***fractionNo*** de tip număr întreg reprezintă numărul de ordine al fracţiei, în cazul rețetelor eliberate fracţionat farmacia/punctul de lucru (pentru o eliberare integrală acest parametru va avea valoarea ***1***);

- parametrul ***cancellationReason*** de tip şir de caractere reprezintă motivaţia anulării (text liber).

Metoda întoarce o valoare numerică indicând dacă cererea a fost procesată cu succes, caz în care valoarea este ***1***, altfel apelul întorcându-se cu eroare.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru anularea unei reţete prescrise este:

```
Utilizatorul (medic sau farmacist) doreşte să anuleze o reţetă
electronică:
Aplicaţia apelează metoda cancelPrescribedPrescription, respectiv
cancelReleasedPrescription cu parametrii corespunzători (vezi lista de mai
sus).
Sistemul central validează cererea şi întoarce un şir de caractere ca
răspuns.
   - Dacă cererea a fost procesată cu succes (valoarea întoarsă este "1"):
         - atunci aplicaţia afişează un mesaj care indică succesul
operaţiei;
   - Altfel se afişează un mesaj de eroare de comunicaţie sau mesajul de
eroare transmis de sistemul central cu privire la motivul din care
anularea nu a reuşit.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Serviciul pentru consultarea reţetelor electronice

Acest serviciu este folosit pentru a permite medicilor şi farmaciilor să consulte reţetele electronice prescrise pentru verificarea datelor transferate în sistemul centaral sau pentru a elibera medicamentele prescrise de medic către pacient, verificând în acelaşi timp autenticitatea reţetei respective.

Consultarea sistemului este obligatorie înainte de eliberarea medicamentelor pentru ca farmacistul să se asigure că reţeta este valabilă şi este înregistrată în sitemul central, iar medicamentele prescrise nu au fost deja eliberate integral sau parţial de altă farmacie.

Metoda *getPrescribedPrescription*

```
String getPrescribedPrescription (
        String providerCode,
        String physicianStencilNo,
        String contractNo,
        String contractType,
        String insuranceHouseCode,
        String series,
        String no,
        Date prescriptionDate )
```

Metoda are opt parametri de intrare:

- parametrul ***providerCode*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;
- parametrul ***physicianStencilNo*** de tip şir de caractere reprezintă numărul de parafă al medicului emitent;
- parametrul ***contractNo*** de tip şir de caractere reprezintă numărul contractului dintre furnizor şi Casa de Asigurări;
- parametrul ***contractType*** de tip şir de caractere reprezintă codul tipului de contract (acest parametru este opţional);
- parametrul ***insuranceHouseCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări;
- parametrul ***series*** de tip şir de caractere reprezintă seria rețetei;
- parametrul ***no*** de tip şir de caractere reprezintă numărul rețetei;
- parametrul ***prescriptionDate*** de tip dată calendaristică reprezintă data la care reţeta a fost prescrisă de medic.

Metoda întoarce şir de caractere reprezentând fişierul de răspuns în format XML, care se validează cu schema ***PhysicianDrugPEResponse.xsd***.

NOTĂ

*Fişierul de răspuns respectă schema de validare **PhysicianDrugPEResponse.xsd**, având aceeaşi structură ca fişierul de răspuns primit de medic la cererea de validare a unei reţete prescrise, deoarece conţine practic acelaşi set de informaţii transmise de către sistem medicului pentru validare, bineînţeles, cu condiţia ce reţeta prescrisă de către medic să fi fost validată cu succes şi în starea emisă (tipărită).*

Această metodă permite aplicațiilor pentru farmacii să implementeze funcționalităţi de preluare automată a conținutului rețetelor electronice în format electronic din sistemul central. Astfel o farmacie poate descărca o rețetă prescrisă de un medic în scopul de a elibera medicamentele corespunzătoare.

Metoda *getPrescribedPrescriptionsForCid*

```
String getPrescribedPrescriptionsForCid (
        String requestXml )
```

Metoda are un singur parametru de intrare:

- parametrul ***requestXml*** de tip şir de caractere reprezintă conţinutul fişierului de cerere în format XML, care se verifică cu schema ***getPrescribedPrescriptionsForCidRequest.xsd***.

Metoda întoarce un şir de caractere reprezentând fişierul de răspuns în format XML care conţine reţetele prescrise şi neeliberate încă, inclusiv fracţiile din reţetele eliberate parţial, şi se validează cu schema ***getPrescribedPrescriptionsForCidResponse.xsd***. Fişierul de răspuns conţine şi eventuale mesaje de alertă emise de sistem către medici în cazul nerespectării unor norme de prescriere.

Structrura celor două fişiere de validare este prezentată în anexa corespunzătoare aplicaţiilor pentru farmacii cu circuit deschis.

NOTĂ

*Această metodă poate fi utilizată doar împreună cu cardul electronic de asigurat de sănătate (CEAS) şi **eCard.SDK** pentru realizarea semnăturii electronice a asiguratului şi completarea câmpurilor corespunzătoare din XML, ceea ce verifică prezenţa asiguratului în farmacie şi certifică exprimarea acordului pentru consultarea datelor personale din sistem prin introducerea PIN-ului pe teminal.*

Această metodă permite aplicațiilor pentru farmacii să implementeze funcționalităţi de preluare automată a conținutului rețetelor electronice în format electronic din sistemul

central. Astfel o farmacie poate descărca o rețetă prescrisă de un medic în scopul de a elibera medicamentele corespunzătoare.

Metoda *getReleasedPrescription*

```
String getReleasedPrescription (
        String providerCode,
        String workplaceCode,
        String insuranceHouseCode,
        String series,
        String no,
        Integer fractionNo )
```

Metoda are şase parametri de intrare:

- parametrul ***providerCode*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (*cod fiscal*) sau CNP, după caz;
- parametrul ***workplaceCode*** de tip şir de caractere reprezintă codul punctului de lucru care a raportat reţeta (acest parametru trebuie sa aibă valoarea din contractul încheiat cu CAS, iar dacă farmacia nu are punct de lucru se va trimite string vid ***""***);
- parametrul ***insuranceHouseCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări;
- parametrul ***series*** de tip şir de caractere reprezintă seria rețetei;
- parametrul ***no*** de tip şir de caractere reprezintă numărul rețetei;
- parametrul ***fractionNo*** de tip număr întreg reprezintă numărul de ordine al fracţiei, în cazul rețetelor eliberate fracţionat farmacia/punctul de lucru (pentru o eliberare integrală acest parametru va avea valoarea ***1***).

Metoda întoarce şir de caractere reprezentând fişierul de răspuns în format XML, care se validează cu schema ***PharmacyDrugsPEResponse.xsd***.

NOTĂ

*Fişierul de răspuns respectă schema de validare **PharmacyDrugsPEResponse.xsd**, având aceeaşi structură ca fişierul de răspuns primit de farmacie la cererea de validare a unei reţete eliberate, deoarece setul de informaţii transmise din sistem este practic acelaşi, bineînţeles, cu condiţia ce reţeta eliberată de către farmacie să fi fost validată cu succes şi în starea finalizată (tipărită).*

Această metodă permite aplicațiilor de raportarepentru farmacii cu circuit deschis să implementeze funcţionalităţi de preluare automată a conținutului reţetelor electronice în

format electronic din sistemul central. Astfel o farmacie poate descărca o rețetă transmisă anterior în sistem, dar pentru care nu a putut prelua din motive tehnice rezultatul validării.

Metoda *getReleasedHospitalPrescription*

```
String getReleasedHospitalPrescription (
        String providerCode,
        String workplaceCode,
        String insuranceHouseCode,
        String series,
        String no,
        Integer fractionNo )
```

Metoda are şase parametri de intrare:

- parametrul ***providerCode*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (*cod fiscal*) sau CNP, după caz;
- parametrul ***workplaceCode*** de tip şir de caractere reprezintă codul punctului de lucru care a raportat reţeta (acest parametru trebuie sa aibă valoarea din contractul încheiat cu CAS, iar dacă farmacia nu are punct de lucru se va trimite string vid ***""***);
- parametrul ***insuranceHouseCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări;
- parametrul ***series*** de tip şir de caractere reprezintă seria rețetei;
- parametrul ***no*** de tip şir de caractere reprezintă numărul rețetei;
- parametrul ***fractionNo*** de tip număr întreg reprezintă numărul de ordine al fracţiei, în cazul rețetelor eliberate fracţionat farmacia/punctul de lucru (pentru o eliberare integrală acest parametru va avea valoarea ***1***).

Metoda întoarce şir de caractere reprezentând fişierul de răspuns în format XML, care se validează cu schema ***HospitalRegisterPEResponse.xsd***.

NOTĂ

*Fişierul de răspuns respectă schema de validare **HospitalRegisterPEResponse.xsd**, având aceeaşi structură ca fişierul de răspuns primit de farmacie la cererea de validare a unei reţete eliberate, deoarece setul de informaţii transmise din sistem este practic acelaşi, bineînţeles, cu condiţia ce reţeta eliberată de către farmacie să fi fost validată cu succes şi în starea finalizată (tipărită).*

Această metodă permite aplicațiilor de raportare pentru farmacii cu circuit închis să implementeze funcționalităţi de preluare automată a conținutului rețetelor electronice în

format electronic din sistemul central. Astfel o farmacie poate descărca o rețetă transmisă anterior în sistem, dar pentru care nu a putut prelua din motive tehnice rezultatul validării.

Metoda *getStatusForPrescriptions*

```
String getStatusForPrescriptions (
        String insuranceHouseCode,
        String providerCode,
        String contractNo,
        String contractType,
        Date startFrom,
        Date endTo,
        String workPlaceCode )
```

Metoda are şapte parametri de intrare, fiind destinată în exclusivitate farmaciilor:

- parametrul ***insuranceHouseCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări;
- parametrul ***providerCode*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;
- parametrul ***contractNo*** de tip şir de caractere reprezintă numărul contractului dintre furnizor şi Casa de Asigurări;
- parametrul ***contractType*** de tip şir de caractere reprezintă codul tipului de contract (acest parametru este opţional);
- parametrul ***startFrom*** de tip dată calendaristică reprezintă data de început a intervalului în care se caută reţetele în sistem;
- parametrul ***endTo*** de tip dată calendaristică reprezintă data de sfârşit a intervalului în care se caută reţetele în sistem;
- parametrul ***workplaceCode*** de tip şir de caractere reprezintă codul punctului de lucru care a raportat rețeta (acest parametru trebuie sa aibă valoarea din contractul încheiat cu CAS, iar dacă farmacia nu are punct de lucru se va trimite string vid **””**).

Metoda întoarce şir de caractere reprezentând fişierul de răspuns în format XML, care se validează cu schema ***ImportPrescriptionStatusResponse.xsd***.

NOTĂ

*Fişierul de răspuns este conform structurii definită de schema de validare **ImportPrescriptionStatusResponse.xsd**, şi conţine informaţii despre datele de facturare, numărul de referinţă sau stări asociate (finalizată, transmisă în SIUI, eliberare parţială/integrală) pentru reţetele transmise în sistem de către farmacie pentru perioada de interogare specificată, identificate unic prin serie şi număr.*

Această metodă permite aplicațiilor de raportare pentru farmacii cu circuit deschis să implementeze funcţionalităţi de preluare automată a informaţiilor despre starea reţetele transmise în sistemul central SIPE întru-un anumit interval. Aceste informaţii pot fi utilizate pentru verificarea acurateţii datelor transmise şi, eventual, pentru retransmiterea corectă a acestora, pregătind astfel datele pentru a înlesni procesul de raportare în vederea decontării în SIUI.

Metoda *downloadNotReportedPrescriptionsReport*

```
String downloadNotReportedPrescriptionsReport (
        String insuranceHouseCode,
        Date startFrom,
        Date endTo,
        String providerCode,
        String workPlaceCode )
```

Metoda are cinci parametri de intrare, fiind destinată în exclusivitate medicilor:

- parametrul ***insuranceHouseCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări;
- parametrul ***startFrom*** de tip dată calendaristică reprezintă data de început a intervalului în care se caută reţetele în sistem;
- parametrul ***endTo*** de tip dată calendaristică reprezintă data de sfârşit a intervalului în care se caută reţetele în sistem;
- parametrul ***providerCode*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;
- parametrul ***workplaceCode*** de tip şir de caractere reprezintă codul punctului de lucru care a raportat reţeta (acest parametru trebuie sa aibă valoarea din contractul încheiat cu CAS, iar dacă farmacia nu are punct de lucru se va trimite string vid **””**).

Metoda întoarce şir de caractere reprezentând fişierul de răspuns în format PDF, arhivat utilizând algoritmul *JavaZIP* şi codificat *base64*, care poate fi salvat local şi afişat cu un utilitar specializat. Fişierul PDF conţine un raport desfăşurătogenerat de sistemul central SIPE.

Această metodă permite aplicațiilor de raportare pentru medici să implementeze funcţionalităţi de preluare şi prezentare a raportului de reţete netransmise online, pentru consultare şi verificare a datelor introduse în aplicaţie, în vederea respectării obligaţiilor contractuale de transmitere în sistemul central a reţetelor electronice.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru consultarea unei reţete prescrise este:

```
Utilizatorul (medic sau farmacist) doreşte să consulte starea unei reţete
prescrise anterior de medic sau eliberate anterior de farmacie:
Aplicaţia apelează metoda getPrescribedPrescription, respectiv metoda
getReleasedPrescription, cu parametrii corespunzători (vezi lista de mai
sus).
Sistemul central validează cererea şi întoarce un şir de caractere ca
răspuns, pe care aplicaţia îl salvează într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se procesează fişierul şi se afişează reţeta
electronică.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează un mesaj de eroare de comunicaţie sau mesajul de eroare
transmis de sistemul central.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Metoda de consultarea a raportului face excepţie de la fluxul prezentat anterior, fişierul PDF nefiind verificat cu schema de validare XSD, ci fiind afişat direct, prin intermediului unei aplicaţii utilizatare specializate.

Numai reţetele raportate online ca tipărite/finalizate vor fi disponibile pentru interogare de către furnizorii de servicii farmaceutice, aceştia vor identifica reţetele prescrise în vederea eliberării medicaţiei după combinaţia de câmpuri: serie şi număr reţetă, parafă medic prescriptor şi CUI unitate emitentă.

Serviciul pentru generarea seriilor de reţete electronice

Prin intermediul acestui serviciu medicului prescriptor poate genera online calupuri noi de reţete în cazul epuizării „stocului" existent. Serviciul expune două metode complementare, una pentru generarea unui calup nou, iar cealaltă pentru descărcarea calupurilor generate anterior.

Metoda *generatePrescriptionSeries*

```
String[] generatePrescriptionSeries (
        String categoryCode,
        String orgUnitCode,
        String uic,
        String subUnitCode,
        Date validFrom,
        Boolean isOnLine,
        Integer quantity )
```

Metoda are şapte parametri de intrare :

- parametrul ***categoryCode*** de tip şir de caractere reprezintă codul categoriei de furnizor, lista valorilor permise fiind prezentată mai jos;
- parametrul ***orgUnitCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări.
- parametrul ***uic*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;
- parametrul ***subUnitCode*** de tip şir de caractere reprezintă codul unic de identificare al subunităţii în sistem (valoarea *partnerCode* din fişierul de personalizare);
- parametrul ***validFrom*** de tip dată calendaristică reprezintă data de la care seriile vor valabile;
- parametrul ***isOnLine*** de tip boolean (*true/false*) indică dacă seriile generate sunt pentru modul de lucru online sau offline (reţete pre-tipărite);
- parametrul ***quantity*** de tip număr întreg reprezintă numărul de reţete din calup.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de răspuns, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

URL-ul va fi generat pentru fiecare cerere şi are o perioadă de valabilitate limitată după trecerea căreia nu va mai fi disponibil pentru a nu permite aplicaţiilor de raportare să descarce accidental un fişier mai vechi folosind un URL memorat.

Metoda *getPrescriptionSeriesInfo*

```
String[] getPrescriptionSeriesInfo (
        String categoryCode,
        String orgUnitCode,
        String uic,
        String subUnitCode )
```

Metoda are patru parametri de intrare:

- parametrul ***categoryCode*** de tip şir de caractere reprezintă codul categoriei de furnizor, lista valorilor permise fiind prezentată mai jos;
- parametrul ***orgUnitCode*** de tip şir de caractere reprezintă codul casei de asigurări cu care furnizorul are contract, valoare din nomenclatorul de case de asigurări.
- parametrul ***uic*** de tip şir de caractere reprezintă codul unic de identificare al furnizorului în sistem, CUI (cod fiscal) sau CNP, după caz;
- parametrul ***subUnitCode*** de tip şir de caractere reprezintă codul unic de identificare al subunităţii în sistem (valoarea *partnerCode* din fişierul de personalizare).

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de răspuns, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

URL-ul va fi generat pentru fiecare cerere şi are o perioadă de valabilitate limitată după trecerea căreia nu va mai fi disponibil pentru a nu permite aplicaţiilor de raportare să descarce accidental un fişier mai vechi folosind un URL memorat.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru generarea seriilor de reţete electronice este:

```
Utilizatorul doreşte să genereze un nou calup de reţete:
Aplicaţia apelează metoda generatePrescriptionSeries cu parametrii
corespunzători (vezi lista de mai sus).
Sistemul central validează cererea şi întoarce un şir de caractere ca
răspuns, pe care aplicaţia îl salvează într-un fişier XML:
   - Se validează fişierul XML cu schema de validare XSD corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se parcurge fişierul şi se importă seria generată.
         - Altfel se afişează mesaj de eroare "Fişier de răspuns
invalid".
Altfel se afişează un mesaj de eroare de comunicaţie sau mesajul de eroare
transmis de sistemul central.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată sau parametrii nu întrunesc condiţiile necesare pentru procesare, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Parametrul ***subUnitCode*** se transmite cu valoarea ***null*** pentru cazul în care contractul se realizează direct cu o unitate medicală cu personalitate juridică, dar trebuie transmis pentru cazul unităţilor medicale care activează în instituţii şcolare sau instituţii de îngrijire a bătrânilor, care nu au contract direct cu Casa de Asigurări ci întocmesc convenţii de eliberare a reţetelor compensate.

Serviciul pentru completarea datelor de facturare

Acest serviciu este folosit pentru a permite unei farmacii să completeze datele de facturare aferente unei reţete electronice eliberate anterior.

Completarea detelor de facturare este necesară înainte de raportarea periodică pentru a putea fi transferate reţetele electronice în SIUI pentru decontare. Astfel aplicaţiile de raportare ale farmaciştilor trebuie să apeleze această metodă pentru a transmite în sistem seria şi numărul facturii, precum şi poziţia de pe borderoul de reţete eliberate.

Metoda *updateInvoices*

```
Integer updateInvoices (
        String requestXml )
```

Metoda are un parametru de intrare:

- parametrul ***requestXml*** de tip şir de caractere reprezintă conţinutul fişierului de cerere în format XML.

Metoda întoarce o valoare întreagă indicând faptul dacă cererea a fost procesată cu succes, caz în care valoarea este *1*, altfel se întoarce valoarea *0*, iar pentru a se determina eroarea se consultă excepţiile returnate.

Fişierul de cerere în format XML, care se validează cu schema ***UpdateInvoicesPERequest.xsd***.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru consultarea unei reţete prescrise este:

```
Utilizatorul (farmacist) doreşte să transmită datele de facturare pentru
reţetele electronice înainte de raportarea periodică pentru decontare:
Aplicaţia apelează metoda updateInvoices cu parametrii corespunzători
(vezi lista de mai sus).
Sistemul central validează cererea şi întoarce o valoare întreagă ca
răspuns.
   - Dacă cererea a fost procesată cu succes (valoarea întoarsă este 1):
         - atunci aplicaţia afişează un mesaj care indică succesul
operaţiei;
   - Altfel se afişează un mesaj de eroare de comunicaţie sau mesajul de
eroare transmis de sistemul central cu privire la motivul din care
anularea nu a reuşit.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Numai pentru reţetele raportate online ca validate şi tipărite, iar pentru reţetele eliberate parţial – doar pentru medicamentele eliberate, vor putea fi completate datele de facturare.

Serviciul pentru preluarea borderourilor cu valori admise la plată

Acest serviciu este folosit pentru a permite unei farmacii să preia din sistemul central valorile acceptate şi respinse la plată corespunzătoare uneia sau mai multor raportări.

În baza valorilor preluate şi prin asocierea acestora cu reţetele de pe care au provenit o farmacie poate emite factura/facturile de decontare către Casa de Asigurări, împreună cu borderourile însoţitoare.

Metoda *getRegisterFeedback*

```
String[] getRegisterFeedback ( String filename )
```

Metoda are un parametru de intrare:

- parametrul ***fileName*** de tip şir de caractere reprezentă numele fişierului de raportare trimis de aplicaţie pentru care se cere borderoul de decontare.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

Dacă nu există un fişier de raportare procesat cu numele dat, metoda întoarce ***null***.

Numele fişierului de raportare identifică în mod unic o raportare efectuată, astfel încât alţi parametrii, cum ar fi tipul de furnizor, nu sunt necesari pentru această metodă. Aplicaţia client trebuie să ţină evidenţa fişierelor de raportare trimise pentru a putea cere borderourile corespunzătoare acestor fişiere.

Metoda *getRegisterFeedbackAggregated*

```
String[]getRegisterFeedbackAggregated (
        String partnerUic,
        String partnerWorkplace,
        DateTime start,
        DateTime stop )
```

Metoda are patru parametri de intrare:

- parametrul ***partnerUic*** de tip şir de caractere reprezintă codul unic de identificare fiscală al furnizorului (farmaciei);
- parametrul ***partnerWorkplace*** de tip şir de caractere reprezintă codul punctului de lucru al farmaciei (acest parametru trebuie sa aibă valoarea din contractul încheiat cu CAS, iar dacă farmacia nu are punct de lucru se va trimite string vid ***""***).
- parametrul ***start*** de tip dată calendaristică reprezintă data de început a perioadei pe care se face agregarea borderourilor;

- parametrul ***stop*** de tip dată calendaristică reprezintă data de sfârşit a perioadei pe care se face agregarea borderourilor.

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

Dacă nu există un fişier de raportare procesat cu numele dat, metoda întoarce ***null***.

Această metodă permite preluarea datelor de pe mai multe raportări dintr-o perioadă calendaristică şi agregarea borerourilor pentru reţetele de pe toate aceste raportatări, spre deosebire de prima metodă care permite prealuare borderourilor pentru o singură raportare.

Instrucţiuni de folosire

Aplicaţia client trebuie să folosească URL-ul rezultat pentru a descărca fişierul cu nomenclatoarele. Dimensiunea fişierului poate fi folosită pentru a verifica completitudinea fişierului descărcat. Fişierul descărcat este o arhivă ZIP care conţine un fişier XML cu rezultatul procesării raportării în SIUI.

Schema de validare pentru fişierul transmis din sistemul central este prezentată în anexa corespunzătoare furnizorilor se servicii farmaceutice cu circuit deschis, şi se numeşte ***ExportPharmacyInv.xsd***.

Un exemplu tipic de algoritm pentru actualizarea nomenclatoarelor este:

```
Se apelează metoda getRegisterFeedback cu parametrii corespunzători.
Dacă se întoarce un vector de şiruri de caractere de lungime 2 atunci:
   - Se consideră primul şir ca fiind url-ul pentru descărcarea
fişierului.
   - Se descarcă fişierul (care este o arhivă ZIP).
   - Dacă dimensiunea fişierului descărcat coincide cu valoarea celui de-
al doilea element din vector atunci:
         - Se dezarhivează fişierul ZIP descărcată şi rezultă un fişier
XML.
         - Se validează fişierul XML cu schema de validare XSD
corespunzătoare.
         - Dacă fişierul este valid atunci:
               - Se parcurge fişierul şi se actualizează tabela de erori
din baza de date.
         - Altfel se afişează mesaj de eroare "Fişier invalid".
   - Altfel se afişează mesaj de eroare de comunicaţie.
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Serviciul pentru activarea cardurilor electronice de asigurări de sănătate

Acest serviciu este folosit pentru a permite unei medicilor de familie să activeze cardurile electronice de asigurări de sănătate şi să completeze date cu caracter medical pe aceste carduri. Completarea datelor medicale se face numai cu acordul pacientului, în baza unui document semnat de acesta.

Acest serviciu expune două metode, una pentru transmiterea propriu-zisă a datelor medicale după activarea cu succes a unui card, şi cealaltă pentru semnalarea unui card inscripţionat cu date incorecte, care nu poate fi activat în consecinţă.

Metoda *sendMedicalData*

```
Integer sendMedicalData (
        String cid,
        String medicalDataRequest )
```

Metoda are doi parametri de intrare:

- parametrul ***cid*** de tip şir de caractere reprezentă numărul (codul) de asigurat înscris pe card activat cu succes pentru care se transmit datele medicale.
- parametrul ***medicalDataRequest*** de tip şir de caractere reprezintă conţinutul fişierului de cerere în format XML, care se validează cu schema ***ActivateSmartCard.xsd***.

Metoda întoarce o valoare întreagă indicând dacă cererea a fost procesată cu succes, caz în care valoarea este 1, altfel apelul întorcându-se cu eroare.

Metoda se apelează pentru transmiterea în sistemul central a datelor medicale insctipţionate cu acordul pacientului de către medic pe cardul electronic după activarea cu succes a acestuia.

Metoda *setInvalidCardData*

```
Integer setInvalidCardData (
        String cid,
        String cardNo,
        String motivation )
```

Metoda are trei parametri de intrare:

- parametrul ***cid*** de tip şir de caractere reprezentă numărul (codul) de asigurat înscris pe card pentru care se transmite cererea de semnalare a datelor incorecte.
- parametrul ***cardNo*** de tip şir de caractere reprezentă numărul cardului de asigurări de sănătate pentru care se transmite cererea de semnalare a datelor incorecte.

- parametrul ***motivation*** de tip şir de caractere reprezentă motivaţia declarată de medic pentru care se transmite cererea de semnalare a datelor incorecte.

Metoda întoarce o valoare întreagă indicând dacă cererea a fost procesată cu succes, caz în care valoarea este 1, altfel apelul întorcându-se cu eroare.

Metoda se apelează pentru semnalarea unui card inscripţionat cu date incorecte, care nu poate fi activat în aceste condiţii.

Serviciul pentru transmiterea facturilor electronice

Prin intermediul acestui serviciu furnizorii pot transmite online facturi electronice pentru decontarea serviciilor medicale sau farmaceutice prestate într-un anumit intervat. Acest interval se suprapune de regulă cu o perioadă de raportare. Serviciul expune o singură metodă pentru transmiterea facturii în format electronic, care preia un fişier XML care trebuie să respecte structura descrisă în anexa specifică fiecărui tip de furnizor.

Metoda *sendEInvoice*

```
Integer sendEInvoice (
        String invoiceFile )
```

Metoda un singur parametru de intrare :

- parametrul ***invoiceFile*** de tip şir de caractere reprezintă conţinutul unui fişier XML semnat electronic conform ***CMS (RFC3852)***, arhivat în formatul ***ZIP (JavaZip)*** şi codat apoi utilizând formatul ***Base64***.

Metoda întoarce o valoare numerică ce reprezintă valoarea numărul de înregistrare alocat unic de sistemul central pentru cererea de transmitere a facturii electronice. Acest număr este returnat doar pentru cererile ce conţin un fişier valid pentru un furnizor aflat în relaţii contractuale cu CAS, în caz contract fiind întors un mesaj de eroare corespunzător.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru transmiterea facturilor electronice este:

```
Utilizatorul completează factura electronică asistat de calculator, iar
apoi iniţiază procedura de transmitere online:
Aplicaţia apelează metoda sendEInvoice cu parametrii corespunzători (vezi
lista de mai sus).
Sistemul central validează cererea, iar dacă cererea este validă ,întoarce
o valoare ca răspuns, pe care aplicaţia o asociază facturii electronice,
pentru identificarea ulterioară.
Altfel se afişează un mesaj de eroare de comunicaţie sau mesajul de
avertizare la validarea cererii în sistemul central.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată sau parametrii nu întrunesc condiţiile necesare pentru procesare, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Fişierul XML trebuie să respecte structura descrisă în anexele specifice fiecărui tip de furnizor, aceasta fiind identică pentru toate tipurile de furnizor, diferenţele constând în tipurile de articole care pot fi facturate de fiecare tip de furnizor.

Serviciul pentru anularea facturilor electronice

Prin intermediul acestui serviciu furnizorii pot transmite online cereri de anulare a unei facturi electronice transmisă anterior în sistemul central, identificată prin serie şi număr, unice la nivelul furnizorului conform Codului Fiscal. Serviciul expune o singură metodă conform specificaţiei de mai jos:

Metoda *cancelEInvoice*

```
Integer cancelEInvoice (
        String serialCode,
        String serialNumber,
        String providerFiscalCode,
        DateTime cancellationDate,
        String cancellationReason )
```

Metoda are trei parametri de intrare:

- parametrul ***serialCode*** de tip şir de caractere reprezintă seria de identificarea a facturii electronice;
- parametrul ***serialNumber*** de tip şir de caractere reprezintă numărul de identificarea a facturii electronice;
- parametrul ***providerFiscalCode*** de tip şir de caractere reprezintă codul fiscal al furnizorului (CUI);
- parametrul ***cancellationDate*** de tip dată calendaristică reprezintă data cererii de anulare a facturii;
- parametrul ***cancellationReason*** de tip şir de caractere reprezintă motivaţia anulării facturii.

Metoda întoarce o valoare numerică indicând dacă cererea a fost procesată cu succes, caz în care valoarea este ***1***, altfel apelul întorcându-se cu eroare.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru anularea unei facturi electronice este:

```
Utilizatorul doreşte să anuleze o reţetă electronică:
Aplicaţia apelează metoda cancelEInvoice cu parametrii corespunzători
(vezi lista de mai sus).
Sistemul central validează cererea şi întoarce o valoare numerică ca
răspuns.
   - Dacă cererea a fost procesată cu succes (valoarea întoarsă este 1):
         - atunci aplicaţia afişează un mesaj care indică succesul
operaţiei;
   - Altfel se afişează un mesaj de eroare de comunicaţie sau mesajul de
eroare transmis de sistemul central cu privire la motivul din care
anularea nu a reuşit.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată sau parametrii nu întrunesc condițiile necesare pentru procesare, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

Anularea este posibilă doar dacă nu a fost deja iniţiat procesul de plată al facturii electronice.

Serviciul pentru preluarea notelor de refuz ale facturilor

Prin intermediul acestui serviciu furnizorii pot prelua online notele de refuz asociate facturilor electronice după ce acestea au fost procesate în sistemul central. Serviciul expune o singură metodă conform specificaţiei de mai jos:

Metoda *getEInvoiceClearingNote*

```
String[] getEInvoiceClearingNote (
      String serialCode,
      String serialNumber,
      String providerFiscalCode )
```

Metoda are trei parametri de intrare:

- parametrul ***serialCode*** de tip şir de caractere reprezintă seria de identificarea a facturii electronice;
- parametrul ***serialNumber*** de tip şir de caractere reprezintă numărul de identificarea a facturii electronice;
- parametrul ***providerFiscalCode*** de tip şir de caractere reprezintă codul fiscal al furnizorului (CUI).

Metoda întoarce un vector de şiruri de caractere de lungime doi. Primul şir din acest vector reprezintă URL-ul de la care se face descărcarea fişierului de răspuns, iar cel de-al doilea şir reprezintă dimensiunea fişierului care trebuie descărcat.

URL-ul de mai sus va fi generat pentru fiecare cerere și are o perioadă de valabilitate limitată după trecerea căreia nu va mai fi disponibil pentru a nu permite aplicaţiilor de raportare să descarce accidental un fişier mai vechi folosind un URL memorat.

Instrucţiuni de folosire

Un exemplu tipic de algoritm pentru descărcarea notei de refuz asociate uneri facturi electronice este:

```
Utilizatorul doreşte preluarea notei de refuz pentru o factură
electronică.
Aplicaţia apelează metoda getEInvoiceClearingNote cu parametrii
corespunzători (vezi lista de mai sus).
Dacă se întoarce un vector de şiruri de caractere de lungime 2 atunci:
   - Se consideră primul şir ca fiind url-ul pentru descărcarea
fişierului.
   - Se descarcă fişierul (care este o arhivă ZIP).
   - Dacă dimensiunea fişierului descărcat coincide cu valoarea celui de-
al doilea element din vector atunci:
         - Se dezarhivează arhiva descărcată şi rezultă un fişier PDF.
         - Se afişează conţinutul fişierului PDF folosind aplicaţia de
vizualizare instalată (ex. Acrobat Reader).
   - Altfel se afişează mesaj de eroare de comunicaţie.
Altfel se afişează un mesaj de eroare de comunicaţie.
```

Observaţii

În cazul în care conexiunea nu a putut fi efectuată sau parametrii nu întrunesc condițiile necesare pentru procesare, rezultatul apelului metodei Web va fi un mesaj de eroare (o excepţie).

ANEXA 2 - Standarde de integrare DES

1. Introducere

Obiectiv document

Acest document conține descrierea standardelor de interfațare și integrare dintre sistemul Dosarul Electronic de Sănătate (DES) și sisteme medicale informatice externe.
Standardele de interfațare sunt bazate pe servicii WEB – descrise în cadrul acestui document.
Pentru realizarea integrării este necesară însă înțelegerea conceptelor DES și compatibilizarea sistemelor medicale externe cu aceste concepte.
Conceptele medicale (cum ar fi: document medical, boli, proceduri, medici, pacienți etc) sunt modelate în sistemul DES conform standardului HL7.
Datele vehiculate între sistemul DES și sistemele externe sunt modelate sub forma de ”document medical” mai precis sub forma ClinicalDocument conform cu arhitectura acestuia descrisă de HL7 (Clinical Document Architecture – CDA release 2). Specificațiile pentru implementarea documentelor medicale conform DES sunt furnizate sub formă de ”Ghid implementare” – documentație furnizată seaparat.
Sunt furnizate două ghiduri de implementare:

- Pentru implementarea documentelor medicale (folosit la transmitere din aplicații externe către DES precum și la primirea în aplicații externe a anumitor documente medicale)
- Pentru implemenatrea datelor medicale relevante (folosit pentr primirea în aplicații externe a conținutului unui dosar medical anume)

Aplicabilitate

Documentul de față se aplică interfețelor expuse de către sistemul DES altor sisteme medicale.

Continutul pachetului de documentatie

Standarde de integrare.pdf – prezentul document
Anexa Ghid implementare CDA – anexa tehnica privind implementarea CDA in DES
Offline Codesystems – extras de valori pentru sisteme de codificare DES la data publicarii
wsdl – descrierea serviciilor web DES
xsd – descrierea mesajelor utilizate de serviciile web DES (doar mesaje ce nu sunt deja cuprinse in wsdl).
xslt – scheme de transformare in HTML a documentelor CDA ce respecta ghidul de implementare DES. Aceste transformari sunt furnizate ca atare, cu titlu exemplificativ, spre a

fi utilizate ca punct de plecare in afisarea dosarului medical si nu isi propun sa respecte formularistica aflata in vigoare sau sa o inlocuiasca.

2. Funcţionalităţile suportate prin Interfaţa Automată

Sistemul DES implementează o serie de funcționalități și printr-o interfață automată destinată sistemelor medicale terțe. Aceste funcționalități sunt oferite și în cadrul portalului extern DES direct utilizatorilor umani (medici, pacienți)

Inregistrare apartinator

Funcționalitatea permite solicitarea de adăugare a unui reprezentant pentru un pacient. Reprezentantul astfel adăugat va putea apoi accesa dosarul de sănătate al reprezentaților. Reprezentantul poate fi adăugat numai de către un medic.

La solicitarea de adăugare reprezentant următoarele situații sunt acceptate de sistemul DES.

1. Atât reprezentantul cât și reprezentatul au deja dosar de sănătate – sistemul DES creează legătura de reprezentare între cele 2 persoane
2. Numai reprezentantul are dosar de sănătate – sistemul DES creează dosarul reprezentatului și apoi legătura cu reprezentantul.
3. Numai reprezentatul are dosar de sănătate – sistemul DES adaugă ca persoană reprezentantul dar nu-i creează și dosar de sănătate. Sistemul DES creează legătura între cele două persoane
4. Nici una dintre cele 2 persoane nu are dosar de sănătate - sistemul DES creează dosarul reprezentatului, adaugă repreentantul fără a-i crea dosar și apoi legătura între cele două persoane.

Notă: la creearea oricărui dosar sistemul DES importă datele medicale istorice din sistemul SIUI și din SIPE dacă sunt regăsite astfel de date. Perioada de timp din urmă este reglată printr-un parametru de sistem (de exemplu 6 luni)
Dosarul de sănătate este creat în mod automat la primul document medical transmis pentru un pacient sau atunci când este declarat un reprezentant al său.

Transmiterea documentelor medicale

Funcționalitatea permite transmiterea de către o aplicație externă a documentelor medicale referitoare la un pacient.

Lista documentelor medicale care pot fi transmise:

1. **Fișa de observație – internare continuă**
    a. se transmite parțial la internare **(HospitalAdmissionDocument)**

b. și completă la externare **(InpatientDischargeDocument)**

2. **Fișa de observație - internare de zi,** se transmite după fiecare vizită **(OutpatientDischargeDocument)**
3. **Fișa de consultație – specialist,** se transmite după consultație **(ConsultationDocument)**
4. **Fișa de consultație – medic familie,** se transmite după consultație **(PrimaryCareConsultationDocument)**
5. **Trimitere clinică,** se transmite după emitere **(ClinicalReferralDocument)**
6. **Trimitere paraclinică,** se transmite după emitere **(LaboratoryReferralDocument)**
7. **Recomandare îngrijire la domiciliu,** se transmite după emitere **(HomeCareReferralDocument)**
8. **Recomandare proteze și dispozitive,** se transmite după emitere **(MedicalDevicesReferralDocument)**
9. **Rețete prescrise** setransmit direct în DES, eliberate se afișează din SIPE **(MedicationPrescriptionDocument)**

Pentru fiecare dintre aceste documente medicale ”Ghidul de implementare” specifică conținutul, forma, regulile de validare, cardinalități, exemple.

Funcționalitatea permite:

A. Trimiterea documentului inițial

B. Trimiterea unor versiuni noi. Acest lucru se face prin trimiterea unui nou document care îl va înlocui pe cel de dinainte. Așadar, documentul curent indică, atunci când este cazul, identificatorul documentului înlocuit.

Efectul transmiterii unui document care înlocuiește alt document este următorul:

- Datele consolidate în Dosarul de Sănătate din documentul înlocuit sunt șterse
- Datele din documentul care îl înlocuiește sunt adăugate

Documentul înlocuit este acceptat de sistem dacă urmatoarele conditii sunt respectate:

- daca semnatarul electronic al acestuia se regăsește și printre autheticators sau legalAuthenticator din documentul de înlocuit
- daca prin schimbarea specialitatii medicale in baza carei se face documentul nu reflecta o schimbare de profil medic in sensul medic spital sau medic de familie.

- dacă se referă la același CID, dacă specilitatea medicului din ”encompassingEncounter > assignedEntity” este aceeași cu cea din documentul de înlocuit și dacă specialitatea medicului ”author” a rețetei este aceeași cu cea din documentul de înlocuit.

Dacă aceste reguli nu pot fi îndeplinite documentul de înlocuit va trebui anulat și trimis unul nou.

Documentele înlocuite sunt menținute în sistem în scop de audit.

Anulare document medical

Funcționalitatea permite transmiterea de către o aplicație externă a unei cereri de anulare a unui document medical deja transmis.

Această funcționalitate se utilizează atunci când documentul trimis a fost complet greșit.

Acestă funcționalitate se utilizează și atunci când documentul trimis avea CID-ul greșit sau specialitatea medicului care a făcut actul medical greșită sau a medicului autor al rețetei. În acest caz:

- Se anulează documentul cu datele greșite (de exemplu cu CID-ul greșit)
- Se trimite un document nou cu CID-ul corect

Pentru cazul în care alte date erau greșite se utilizează funcționalitatea de Transmitere documente medicale cu indicația de ”înlocuire”.

Efectul anulării în DES:

- Datele consolidate în Dosarul de Sănătate din documentul anulat precum și din versiunile anterioare ale lui (dacă este cazul) sunt șterse

Pentru ca cererea de anulare să fie acceptată:

- Autorul cererii trebuie să fie unul dintre ”authenticators” sau ”legal autheticator” din documentul de anulat
- Instituția medicală de unde se trimite cererea să fie aceeași cu cea care a transmis documentul original
- Documentul indicat pentru anulare să fie cel curent (ultima versiune în cazul în care au fost mai multe versiuni)

Documentele anulate sunt menținute în sistem în scop de audit.

Interogare document medical

Acestă funcționalitate permite unei aplicați externe să obțină din sistemul DES un anumit document medical.

Ca și în cazul transmisiei documentul este furnizat în format HL7 CDA.

Ca regulă generală, pentru a putea primi un document medical aplicația și medicul care îl solicită trebuie să îndeplinească criteriile de securitate impuse de DES și să fie autorizate să primească acel document.

Pe scurt:

- Aplicația trebuie să fie certificată și conectată la DES
- FSM-ul să fie autentificat
- Medicul să fie autentificat
- Medicul să aibă dreptul de a primi acel document:
    - Are acces la dosar conferit de politicile de securitate alese de pacient
    - Are acces la dosar conferit direct de pacient
    - Este documentul său (este autorul acestuia)
    - Este unul dintre semnatarii documentului original declarat ca autheticator sau legalAutheticator în CDA
    - Are acces conferit de pacient pe loc (prin coautentifcare cu card de sănătate sau matrice de securitate)

Nu pot fi obținute documentele care au fost înlocuite și nici cele anulate.

Interogare trimiteri medicale ale pacientului pentru o specialitate

Funcționalitatetea permite unei aplicații externe să obțină lista de trimteri medicale pe o anumită speciliatate și pentru un anumit pacient.

Pentru ca DES să furnizeze aceste date medicul care face solicitarea trebuie să aibă specilitatea din trimitere și pacientul să activeze politica de securitate prin care acordă acest gen de permisiune.

După obținerea listei aplicația poate solicita o anumită trimitere prin funcționalitatea de ”Interogare document medical”. DES o va furniza însă numai dacă se îndeplinesc condițiile descrise mai sus.

Interogare listă documente emise de un medic

Funcționalitatetea permite unei aplicații externe să obțină lista de documente medicale emise de un anumit medic.

Medicul care face cererea va obține lista documentelor transmise în DES pentru care a fost declarat ”author” în header-ul CDA

Perioada pentru care pot fi cerute documente per cerere este reglată printr-un parametru de sistem (de exemplu maxim 30 de zile incepand cu perioada de start a cautarii). In cazul in care intervalul de interogare dataStart – dataStop > N (=30) zile, utilizatorul nu primeste vreo atentionare, dar intern serviciul limiteaza perioada maxima de cautare la N(=30) de zile relativ la data de start. N – reprezinta un numar exprimat in zile si configurabil la nivel de sistem.

Interogare documente vechi

Funcționalitatea permite unei aplicații terțe să obțină liste de documente vechi de un anumit tip, pentru o anumită perioadă și pentru un anumit pacient care au fost transmise în DES.
”Documente vechi” sunt acele documente transmise în DES dar care nu mai apar în ”Interogare date consolidate/Istoric evenimente”.
Vechime este reglată printr-un parametru de sistem (de exemplu mai vechi de 6 luni). Obiectivul este de a menține în datele consolidate, pentru interogare rapidă acele date medicale considerate ca fiind relevante la momentul accesării dosarului.
Perioada pentru care pot fi cerute per cerere este reglată printr-un parametru de sistem (de exemplu maxim 1 lună).
Cererea va fi respinsă dacă se încearcă obținerea de documente care sunt în datele consolidate sau nu se respectă perioada maximă. Documentele din datele consolidate se obțin prin metoda de interogare date consolidate.

Interogare date consolidate (Date medicale relevante – DMR)

Această funcționalitate permite unei aplicații externe să obțină datele medicale din dosarul de sănătate al unui pacient.
Interogarea se face pentru fiecare secțiune în parte.
Pentru a primi o anumită secțiune aplicația și medicul care face solicitarea trebuie să îndeplinească criteriile de securitate DES și să fie autorizate să primească acele date.

Sumarul de urgenta (EmergencySummaryOutDocument)

Sumarul de urgență poate fi interogat din orice aplicație authetificată și autorizată de sistemul DES și de către orice medic autentificat și autorizat de sistemul DES.

Istoricul medical (MedicalHistoryOutDocument)

La interogarea istoricului medical trebuie furnizate și datele de coautentificare ale pacientului (card de sănătate sau matrice).

Atunci când medicul are drept de acces la întreg dosarul, conferit de pacient prin politicile de acces sau direct, nu mai este necesară furnizarea coautentificării pacientului.

La coautentificare, poziția furnizată din matricea de securitatea a unui pacient anume poate fi utilizată un timp limitat de către un medic (reglat prin parametru de sistem – de exemplu 20 minute). După acest timp trebuie furnizată o altă poziție din matricea de securitate și nu se mai poate reutiliza vechea poziție).

Dacă se încearcă obținerea acstor date fără coautentificare și fără ca medicul să aibă dreptul sistemul DES va respinge cererea cu un mesaj de eroare specific.

Antecedenete declarate de pacient (ReportedMedicalHistoryOutDocument)

Se obțin în aceleași condiții ca istoricul medical.

Documente medicale (MedicalEventsOutDocument)

Se obțin în aceleași condiții ca istoricul medical.

Date personale (PatientInformationOutDocument)

Se obțin în aceleași condiții ca istoricul medical.

Genereaza pozitie sugerata din matricea de securitate

Această funcționalitate permite unei aplicații terțe să obțină din DES o sugestie de poziție din matricea de securitate a unui pacient care să poată fi folosită de un medic.

Funcționalitatea este necesară deoarece o aplicație nu poate cunoaște pozițiile deja utilizate de acel medic (”arse”) dacă medicul a accesat dosarul pacientului și din alte aplicații.

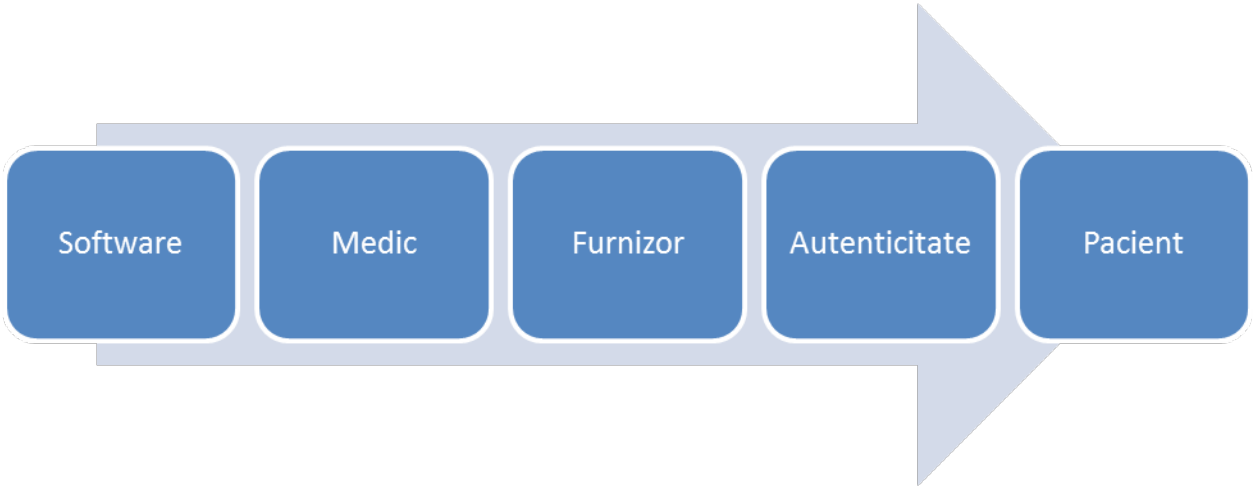
Export cataloage

Funcționalitatea permite unei aplicații terțe să obțină din DES cataloagele utilizate.

3. Autorizare

Autorizarea in DES este realizata sub forma unui pipe-line de verificari ce impreuna definesc contextul operatiei apelate.

Ca nota de ordin general apelurile serviciilor DES vor contine datele de identificare ale pacientului sau ale medicului, chiar daca acestea sunt furnizate in contextul de autentificare iar DES va corela si verifica informatiile.



Autentificare FSM

Se realizeaza prin transmiterea unui mesaj de la modulul OCSP al PIAS ce contine certificatul, user, parola.Clientul va furniza credentialele asociate furnizorului de servicii medicale (user si parola) precum si token-ul obtinut de la OCSP.

Pentru autorizare tokenul poate fi obtinut de la oricare din serviciile OCSP oferite de platforma PIAS (SIUI, SIPE, CEAS, DES).

Pentru a putea obține jetonul de sesiune serviciul de autentificare necesită transmiterea ca parametru a numelui utilizatorului SIUI care se solicită accesul, ca în exemplul următor: https://www.siui.ro/OCSP/validator?username=1234567_CODCAS

De notat că acest token are o perioadă de valabilitate limitată, după care expiră, fiind necesară obţinerea unui nou jeton.

Tokenul asfel obtinut va fi pus pe headerul http la crearea apelului catre DES dupa cum urmeaza:

Header HTTP	Valoare
OCSP_RESPONSE	token-ul obtinut de la serviciul web OCSP al oricarora din sistemele SIUI, SIPE, CEAS, DES.

Authorization	Cuvantul „**Basic**” urmat de un token obtinut prin encodarea Base64 a valorilor user si parola concatenate prin „**:**” Exemplu: User: **user1** Parola: **pass** Se concateneaza valorile: **user1:pass** Se aplica Base64(„user:pass”) ->**dXNlcjE6cGFzcw==** Valorea trimisa este: **Basic dXNlcjE6cGFzcw==**

Autentificare medic

Se realizeaza prin utilizarea unui certificat digital calificat pentru stabilire conexiune SSL mutual.

DES utilizeaza certificatele digitale al medicilor inrolate in SIUI.

Autenticitate

Se realizeaza prin semnarea mesajului relevant utilizand standardul PKCS7, mecanismul de semnatura detasata.

La sosirea unui mesaj ce are semnatura digitala (doar documentele medicale care afecteaza istoricul pacientului sau afecteaza aspectele administrative ale contului – spre exemplu adaugare apartinator) se despacheteaza documentul, se valideaza semnatura, se valideaza certificatul folosit pentru semnare, si se verifica revocarea acestuia la OCSP STS.

Se utilizeaza la apelurile serviciilor web DES ce afecteaza istoricul pacientului sau inroleaza un apartinator pentru un dosar.

Se vor transmite:

- archivedDocument: documentul CDA se compreseaza cu algoritm ZLIB (RFC 1950)
    - Framework-ul WS folosit va face mai departe encodarea in base64 pentru transmisie
    - Atentie: nu se transmite un fisier ZIP continand la randul lui un fisier. Se transmite continutul CDA arhivat in format ZIP
- pkcs7signature: semnatura digitala asupra vectorului de octeti obtinut dupa compresia ZIP

- Semnatura include doar certificatul utilizat pentru semnare (nu intregul lant de certificate)

```
<pkcs7signature>...</pkcs7signature>
```

Pe baza datelor din certificate se determina identitatea persoanei semnatare.

In cazul transmiterii documentelor CDA catre DES se va verifica ca semnatura sa fie a unui dintre urmatorii actori:

- Unul din medicii enumerati in authenticator
- Medicul precizat in legalAuthenticatior

Semnatura va include timestamp.

CoAutentificarea pacient

Co-autentificarea este mecanismul prin care se demonstreaza acceptul pacientului pentru executia unei operatii in DES.

Utilizarea unui challenge inclus in mesajul transmis pe care il poate furniza doar pacientul

- Un cod din matricea de securitate
- Un token semnat cu cardul sa sanatate CEAS

Apelurile ce necesita coautentificare sunt:

- Interogare document medical pe baza de ID
- Interogarile de date consolidate
- Interogare lista documente medicale din dosar pacient mai vechi decat cele prezente in sectiunea „Documente medicale”

Recomandare implementare: in cazul utilizarii CEAS recomandam transmiterea challenge-ului de fiecare data. Altfel, daca interogarea fara nici un challenge intoarce SOAP fault continand codul de eroare “ACCES-2” in elementul detail (vezi capitolul Exceptii SOAP), se solicita co-autentificare.

Accesul la dosar este guvernat, din punct de vedere al acceptului pacientului, de urmatoarele reguli:

- Sumarul de urgenta este acesibil oricarui medic
- Dosarul integral sau documentele medicale ce constituie dosarul sunt acesibile fara restrictii unui medic doar daca interogarea face obiectului unei politici de securitate la care pacientul a ales sa adere sau daca pacientul a conferit drepturi de acces complete pentru codul de parafa al medicului utilizand portalul dedicat pacientilor
- Documentele medicale pot fi accesate de medic fara restrictii daca medicul este unul dintre participanti in documentul transmis: author, autenticator sau legalAuthenticator.

Pacientul poate adera sa nu la urmatoarele politici:

- Medicul de familie poate accesa dosarul in lipsa pacientului
- Medicul specialist ce are pacientul pe lista, intr-un program de sanatate national integrat in DES
- Pe durata internarii orice medic din spital are acces la dosar
- Medicul specialist poate vedea o trimitere medicala catre propria specialitate cunoscand CID-ul pacientului

CoAutentificare pacient utilizand CEAS

Documentul de mai jos reprezinta descrierea caracteristicilor de interfatare cu SDK-ul implementat in cadrul CEAS pentru cititoarele de smart-card:

- http://siui.casan.ro/cnas/siui_3.5/docs/specificatii/Specificatie%20Interfatare%20SIUI%20-%20Anexa%20102%20-%20Specificatii_eCard_SDK.pdf

Operatia de semnare electronică se poate realiza numai cu un card activ si se face numai prin apelul metodei ComputeHash expusa de SDK.

Metoda are urmatoare definitie:

```
byte[] ComputeHash( byte[] buffer )

buffer - byte array to compute

returns - computed hash code
```

Scenariul de utilizare este următorul:

1. Se primeşte token software (session-id-hash) de la serverul de autentificare OCSP denumit in continuare ***ocspToken***
2. Se formează un şir de caractere de forma: `cid|cardNo|evidenceDate|ocspToken`, unde *evidenceDate* este in format *XmlDateTime*.
3. Şirul de caractere se transformă într-un array de bytes.
4. Se apelează metoda *computeHash* expusă de SDK, iar rezultatul în format base64 se completează în atributul *evidenceHash* la apelul către DES.

```
<patientCoAuthentication>

      <ns2:ceasChallenge>

           <cardNo></cardNo>

           <evidenceDate>...</evidenceDate>

           <evidenceHash>...</evidenceHash>

      </ns2:ceasChallenge>

      <coAuthenticatorCID>...</coAuthenticatorCID>
```

```
</patientCoAuthentication>
```

CoAutentificare pacient utilizand matricea de securitate

Se realizeaza prin utilizarea unui challenge inclus in mesajul transmis pe care il poate furniza doar pacientul – fie un cod din matricea de securitate.

Apelurile ce necesita coautentificare sunt:

- Interogare document medical pe baza de ID
- Interogare lista documente medicale mai vechi
- Interogarile de date consolidate – mai putin sumarul de urgenta

Elementul de coautentificare are forma:

- Pentru autentificare cu matrice de securitate

```
<patientCoAuthentication>
       <coAuthenticatorCID>….</coAuthenticatorCID>
       <ns2:matrixChallenge>
            <x>..</x>
            <y>..</y>
            <value>..</value>
       </ns2:matrixChallenge>
</patientCoAuthentication>
```

Pozitiile transmise in momentul coautentificarii se obtin invocand operatia **Genereaza pozitie sugerata din matricea de securitate**

Astfel un apel securizat de preluare a istoricului medical cu coautentificare are forma:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
       <S:Header>
            <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">
            <challengeResponse>GPI50F0FNTU4G3VxWGGMebgfdRFGzwWxXkMfrQO8xpEscJIuftg+…
       <challengeResponse>
                   <clientCode>testApp1</clientCode>
            </ns2:desClientSoftwareAuthentication>
       </S:Header>
       <S:Body>
            <ns2:getConsolidatedMedicalHistoryS xmlns:ns2="core.des.uti.ro">
```

```
                    <consolidatedMedicalHistory>
                           <patientCid>12345678901234567890</patientCid>
                                  <patientCoAuthentication>
                                         <ns2:matrixChallenge>
                                                <value>44</value>
                                                <x>5</x>
                                                <y>7</y>
                                         </ns2:matrixChallenge>

            <coAuthenticatorCID>111111111111111</coAuthenticatorCID>
                                  </patientCoAuthentication>
                           <physicianStencilNo>243234</physicianStencilNo>
                    </consolidatedMedicalHistory>
            </ns2:getConsolidatedMedicalHistoryS>
        </S:Body>
</S:Envelope>
```

Autentificare client software

Se realizeaza prin transmiterea codului de furnizor si al unui hash obtinut prin aplicarea AES cu o cheie specifica furnizorului asupra unui token cu valabilitate limitata.

Tokenul este format din codul operatiei SOAP si data curenta, in acest fel un cod poate fi utilizat pentru un anumit tip de operatie o perioada limitata de timp configurata la nivelul sistemului. Valabilitatea este stabilita de data transmisa in tokenul de autentificare si trebuie sa fie incadrata intr-un interval de incredere +/- minute configurat fata de ora serverului catre care se transmite cererea (ex: +/- 60 minute).

Exemplu de token valid pentru operatia de salvare document:

```
storeClinicalDocument#2013-11-23T12:34:23
```

Denumiriti de operatii ce se vor folosi pentru apelurile DES sunt asadar urmatoarele:

```
getClinicalDocument -> preluare document medical

storeClinicalDocument -> salvare document medical

removeDocumentSet -> anulare document medical

initializeMedicalFile -> initializare dosar

getPhysicianClinicalDocuments -> documentele mele
```

```
getRelevantReferrals -> trimiteri pe specialitate

getMedicalFileOlderDocuments -> preluare documente mai vechi

getConsolidatedAntecedents -> preluare antecedente

getConsolidatedEventsHistory -> preluare evenimente medicale

getConsolidatedMedicalHistory -> preluare istoric medical

getConsolidatedPatientIdentity -> preluate identitate pacient

getConsolidatedSummary -> preluare sumar de urgenta

suggestMatrixCoordinates -> sugestie coordonate matrice

exportSystemCodesSummary -> export nomenclator cataloage

exportCodeSystem -> exportul unui catalog
```

Numele operatiei poate contine si sufixul „S” pentru a permite ca numele serviciului sa corespunda cu numele metodei. Spre exemplu se poate folosi *removeDocumentSetS* in loc de *removeDocumentSet*, sau *storeClinicalDocumentS* in loc de *storeClinicalDocument.*
Exemplu de constructie a hash-ului trimis catre DES (Java):

```
//Initialization Vector preluat din certificat

byte iv[] = new byte[] {0,0, 0, 0, 0,0, 0, 0, 0,0, 0, 0, 0, 0, 0, 0};

String keyBase64 = "some key"; //cheia encodata base64 preluata din certificat

Byte[] key = Base64.decodeBase64(keyBase64);

String message = "storeClinicalDocument#2013-11-23T12:34:23";

byte[] symKeyData = key;//DatatypeConverter.parseHexBinary(key);

SecretKeySpec symKey = new SecretKeySpec(symKeyData, "AES");

//initializeaza cifrul

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

cipher.init(Cipher.ENCRYPT_MODE, symKey, iv);

// cripteaza mesaj

byte[] encrypted = cipher.doFinal(message.getBytes());
```

Se foloseste **CBC** la criptare astfel incat este nevoie de un **Initialization Vector** pentru a evita criptarea in mod identic a blocurilor identice.

Rezultatul este pus in headerul mesajului SOAP impreuna cu identificatorul software-ului extern

```
<S:Header>

       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">
```

```
        <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY39l
        Cm</challengeResponse>

             <clientCode>testApp1</clientCode>

        </ns2:desClientSoftwareAuthentication>

</S:Header>
```

4. Servicii web

DES expune un set de servicii webce pot fi utilizate pentru a transmite date medicale in DES, pentru a interoga datele din dosarele de sanatate sau pentru a executa operatii administrative. Portalul DES este client al acestor servicii. Serviciile web sunt destinate aplicatiilor pentru medici.

DES se va folosi in regim on-line, ideal documentele medicale sunt transmise imediat dupa intalnirea medicala.

La baza proiectarii DES a stat analiza unei comisii de medici ce au stabilit Datele Medicale Relevante. Acestea reprezinta un subset din documentele medicale. Relevanta acestor date are in unele situatii caracter temporar - unele informatii isi “pot pierde” relevanta in acceptiunea acestei proiectari.

DES este optimizat pentru a furniza sectiunile DMR, dar permite cautarea documentelor medicale ce nu mai sunt cuprinse in DMR.

Securitatea operatiilor

Operatie logica	Token operatie construire token autentificare software	Autenticitate (Semnatura digitala) cerere	Autenticitate (Semnatura digitala) raspuns	Co-Autentificare
Inregistrare apartinator	initializeMedicalFile	Da	Nu	Nu

Salvare document	storeClinicalDocument	Da	Nu	Nu
Anulare document medical	removeDocumentSet	Da	Nu	Nu
Preluare document pe baza de id	getClinicalDocument	Nu	Da	Da
Interogare trimiteri medicale ale pacientului pentru o specialitate	getRelevantReferrals	Nu	Nu	Nu
Documente emise de un medic	getPhysicianClinicalDocuments	Nu	Nu	Nu
Interogare documente vechi	getMedicalFileOlderDocuments	Nu	Da	Da
Preluare antecedente	getConsolidatedAntecedents	Nu	Da	Da
Istoric medical	getConsolidatedMedicalHistory	Nu	Da	Da
Date personale	getConsolidatedPatientIdentity	Nu	Da	Da

Sumarul de urgenta	getConsolidatedSummary	Nu	Da	Nu
Documente medicale	getConsolidatedEventsHistory	Nu	Da	Da
Sugereaza pozitie neutilizata din matricea de securitate	suggestMatrixCoordinates	Nu	Nu	Nu
Sumar cataloage exportabile	exportCatalogsSummary	Nu	Nu	Nu
Export catalog dupa nume	exportCatalog	Nu	Nu	Nu

Tipuri de date des utilizate

Implementarea CDA in Dosarul Electronic de Sanatate

DES utilizeaza standardul CDA pentru a modela documentele medicale ce fac obiectul DES dar si sectiunile dosarelor medicale electronice consolidate.

Documentele CDA, pentru a fi valide trebuie sa respecte urmatoarele reguli:

- Sa respecte structura descrisa de schema XSD din directorul Schema CDA rev2\infrastructure\cda\CDA.xsd
- Sa respecte constrangerile de multiplicitate si regulile de compozitie din ghidul de implementare al CDA in DES din directorul Guides
- Sa utilizeze nomenclatoarele (codesystem) indicate in GHID cu versiunea valabila la data documentului

Schema CDA cuprinsa in acest pachet de documentatie este extrasa de pe site-ul www.hl7.org. Pachetul de documentatie complet al CDA poate fi descarcat de la adresa http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7, documentul "CDA

Release 2". Pentru a descarca aceste fisiere trebuie sa va inregistrati gratuit pe site-ul www.hl7.org.

Ghidul de implementare CDA prezent in pachetul de documentatie prezinta atat informatii generale despre standard cat si cum este acesta folosit pentru a modela datele ce fac obiectul DES.

Important

Unele sisteme de codificare (codesystem) utilizate in documentele CDA au fost proiectate in alte sisteme ale CNAS si in unele cazuri nu respecta regulile exprimate in CDA.xsd.

Pentru a fi utilizate in DES este obligatoriu ca toate codurile utilizate in elemente CE (coded element), CD (coded description), CV (coded value) sa fie in prealabil encodate utilizand algoritmul de encodare RFC 2396 (cunoscut ca url-encode).

Urmatorul tabel prezinta sursa codificarii sistemelor de codificare utilizate in implementarea CDA pentru DES.

Codesystem	Descriere	Codificat de	OID
ActiveSubstances	Substante active	SIUI	2.16.840.1.113883.3.3368.6.11
AdministrativeGender	Sexul pacientului	HL7	2.16.840.1.113883.5.1
AdmissionTypes	Tip internare	DES (via DRG)	2.16.840.1.113883.3.3368.6.20
BloodABO	Grupa sanguina AB0	HL7	2.16.840.1.113883.3.3368.6.33
BloodRh	Tip Rh	HL7	2.16.840.1.113883.3.3368.6.34
CANAMED	Catalogul national medicamentele de uz uman eliberate cu prescriptie medicala, autorizate de punere pe piata	SIUI	2.16.840.1.113883.3.3368.6.24
Cities	Catalogul de localitati al SIUI	SIUI	2.16.840.1.113883.3.3368.6.4
ClinicalServices	Servicii clinice	SIUI	2.16.840.1.113883.3.3368.6.7

ClinicalServicesMF	Servicii clinice MF	SIUI	2.16.840.1.113883.3.3368.6.43
Concentrations	Concentratii medicamente	SIUI	2.16.840.1.113883.3.3368.6.13
DeathCause	Tip deces	DES (via DRG)	2.16.840.1.113883.3.3368.6.30
Diag999	Diagnostice 999	SIUI	2.16.840.1.113883.3.3368.6.1
DischargeStatus	Stare la externare	DES	2.16.840.1.113883.3.3368.6.16
DischargeType	Tip externare	DES	2.16.840.1.113883.3.3368.6.19
Districts	Catalogul de judete al SIUI	SIUI	2.16.840.1.113883.3.3368.6.3
DocumentTypeSources	Tipuri de documente folosite ca sursa de informatie	DES	2.16.840.1.113883.3.3368.6.36
Drugs	Nomenclator de medicamente SIUI	SIUI	2.16.840.1.113883.3.3368.6.5
FamilyMemberRelationType	Lista de relatii familiale pentru specificarea istoricului de boli familiale (4 concepte): mama naturala, tata natural, frate natural, sora naturala	HL7	2.16.840.1.113883.3.3368.6.25
HomeCareMedicalServices	Servicii de ingrijire la domiciuliu	SIUI	
HospitalServices	Servicii spitalicesti	SIUI	2.16.840.1.113883.3.3368.6.42

ICD10AM	Clasificarea statistica internationala a bolilor si a problemelor de sanatate inrudite, modificarile australiene (ICD-10-AM)	SIUI	2.16.840.1.113883.3.3368.6.18
Immunizations	Imunizari	DES (via RENV)	2.16.840.1.113883.3.3368.6.29
InsuranceHouses	Case de asigurari de sanatate	SIUI	2.16.840.1.113883.3.3368.6.35
LaboratoryServices	Servicii paraclinice	SIUI	2.16.840.1.113883.3.3368.6.8.1
LocationType	Tip locatie pentru consultatie MF	DES	2.16.840.1.113883.3.3368.6.21
MedicalDeviceLaterality	Tipuri de dispozitive medicale	DES	2.16.840.1.113883.3.3368.6.14
MedicalDevices	Dispozitive medicale	SIUI	2.16.840.1.113883.3.3368.6.2
MedicalProcedures	Proceduri medicale	SIUI	2.16.840.1.113883.3.3368.6.17
MedicalProcedureTypes	Tipuri de proceduri medicale	DES	2.16.840.1.113883.3.3368.6.41
MedicalSpecialities	Specialitati medicale	SIUI	2.16.840.1.113883.3.3368.6.9
NationalHealthProgrammes	Nomenclator Programe Nationale de Sanatate	SIUI	2.16.840.1.113883.3.3368.6.6
OtherSectionCodes	Extensie LOINC	DES	2.16.840.1.113883.3.3368.6.26
PharmaceuticalForms	Forme farmaceutice	SIUI	2.16.840.1.113883.3.3368.6.12
PrescriptionTypes	Tipuri de retete	SIUI	2.16.840.1.113883.3.3368.6.10
ProsthesisTypes	Tipuri de proteze	DES	2.16.840.1.113883.3.3368.6.15
SectionTypeSources	Tipuri de sectiuni din	DES	2.16.840.1.113883.3.3368.6.37

	documentele folosite ca sursa de informatie

Tipul de surse de date pentru dosarul de sanatate

In toate interogarile campul „documentType” poate lua una din urmatoarele valori:

```
67852-4 -> HospitalAdmission  //PrezentareLaInternare

34108-2 -> OutpatientDischarge //FisaDeSpitalizareDeZiPrezentareVizita

46458-1 -> OutpatientDischarge //ExtrasFisaSpitalizareContinua

11488-4 -> Consultation //FisaConsultatie

29551-9 -> MedicationPrescription //Reteta emisa in sistemul DES

57133-3 -> ClinicalReferral //TrimitereClinica

57133-2 -> LaboratoryReferral //TrimitereParaclinica

57133-5 -> HomeCareReferral //RecomandarePentruIngrijiriLaDomiciliu

57133-4 -> MedicalDevicesReferral //RecomandarePentruDispozitiveMedicale

68834-1 -> PrimaryCareConsultation //FisaConsultatieMedicinaDeFamilie

SIPE -> SIPE (atunci cand este vorba de o reteta emisa in SIPE)
```

Operatiile

Operatie logica	WSDL 1.1	Serviciu	Port	Operatie
Inregistrare apartinator	/ desws/ManageMedicalFile/ManageMedicalFile.wsdl	ManageMedicalFile	ManageMedicalFilePort	initializeMedicalFileS
Salvare document	/ desws/StoreClinicalDocument/StoreClinicalDocument.wsdl	StoreClinicalDocument	StoreClinicalDocumentPort	storeClinicalDocumentS

Anulare document medical	/desws/StoreClinicalDocument/StoreClinicalDocument.wsdl	StoreClinicalDocument	StoreClinicalDocumentPort	removeDocumentSetS
Preluare document pe baza de id	/desws/ClinicalDocument/ClinicalDocument.wsdl	ClinicalDocument	ClinicalDocumentPort	getClinicalDocumentS
Preluare antecedente	/desws/ConsolidatedClinicalDocument/ConsolidatedClinicalDocument.wsdl	ConsolidatedClinicalDocument	ConsolidatedClinicalDocumentPort	getConsolidatedAntecedentsS
Date personale	/desws/ConsolidatedClinicalDocument/ConsolidatedClinicalDocument.wsdl	ConsolidatedClinicalDocument	ConsolidatedClinicalDocumentPort	getConsolidatedMedicalHistoryS
Identitate Pacient	/desws/ConsolidatedClinicalDocument/ConsolidatedClinicalDocument.wsdl	ConsolidatedClinicalDocument	ConsolidatedClinicalDocumentPort	getConsolidatedPatientIdentityS
Sumarul de urgenta	/desws/ConsolidatedClinicalDocument/ConsolidatedClinicalDocument.wsdl	ConsolidatedClinicalDocument	ConsolidatedClinicalDocumentPort	getConsolidatedSummaryS
Documente medicale	/desws/ConsolidatedClinicalDocument/ConsolidatedClinicalDocument.wsdl	ConsolidatedClinicalDocument	ConsolidatedClinicalDocumentPort	getConsolidatedEventsHistoryS

Genereaza pozitiesugerata din matricea de securitate	/desws/SecurityMatrix/SecurityMatrix.wsdl	SecurityMatrix	SecurityMatrixPort	suggestMatrixCoordinates
Interogare trimiteri medicale ale pacientului pentru o specialitate	/desws/ClinicalDocument/ClinicalDocument.wsdl	ClinicalDocument	ClinicalDocumentPort	getRelevantReferrals
Documente emise	/desws/ClinicalDocument/ClinicalDocument.wsdl	ClinicalDocument	ClinicalDocumentPort	getPhysicianClinicalDocuments
Interogare documente vechi	/desws/ClinicalDocument/ClinicalDocument.wsdl	ClinicalDocument	ClinicalDocumentPort	getMedicalFileOlderDocuments
Sumar cataloage exportabile	/desws/ExportCodingSystem/ExportCodingSystem.wsdl	ExportCodingSystem	ExportCodingSystemPort	getCatalogsSummary
Export catalogdupa nume	/desws/ExportCodingSystem/ExportCodingSystem.wsdl	ExportCodingSystem	ExportCodingSystemPort	exportCodeSystem

Inregistrare apartinator

Exemplu apel SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

       <S:Header>

            <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">
```

```
        <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlwWfLdK3BApfA6yqZFpNA
        tR</challengeResponse>

                    <clientCode>testApp1</clientCode>

            </ns2:desClientSoftwareAuthentication>

        </S:Header>

        <S:Body>

            <ns2:initializeMedicalFileS xmlns:ns2="core.des.uti.ro">

                    <initMedicalFileRequest>

                            <archivedDocument>eJzFUtFqgzAU/...</archivedDocument>


            <pkcs7signature>MIAGCSqGSIb3DQEHAqCAMIACAQExD...</pkcs7signature>

                    </initMedicalFileRequest>

            </ns2:initializeMedicalFileS>

        </S:Body>
</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>

        <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">



        <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY391
        Cm</challengeResponse>

            <clientCode>testApp1</clientCode>

        </ns2:desClientSoftwareAuthentication>

</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
initializeMedicalFile#2013-11-23T12:34:23
```

2) **O versiune arhivata a forumularului de initializare dosar:**

```
<archivedDocument>eJzFUtFqgzAU/...</archivedDocument>
```

Explicatie:

Se construieste un xml de forma:

```
<initMedicalFileRequest>
```

```
        <physician>
            <fullName>Georgescu Ion</fullName>
            <stencilNo>424334</stencilNo>
        </physician>
        <relationType>CARER</relationType>
        <requestPerson>
            <icExpiration>2014-01-31T00:00:00.000+02:00</icExpiration>
            <icNumber>24534323</icNumber>
            <icSeries>rt</icSeries>
            <personData>
                    <birthDate>2012-11-06T00:00:00.000+02:00</birthDate>
                    <cid>12345678901234567890</cid>
                    <firstName>Ion</firstName>
                    <lastName>Popescu</lastName>
            </personData>
        </requestPerson>
        <subject>
            <birthDate>2014-01-31T00:00:00.000+02:00</birthDate>
            <cid>12345678901234567890</cid>
            <firstName>Vasile</firstName>
            <lastName>Ionescu</lastName>
        </subject>
</initMedicalFileRequest>
```

Observatie: Detalii despre structura XML-ul care trebuie impachetat si semnat pentru a face aceasta operatie se regasesc in anexa atasata documentatiei (InitMedicalFileRequest_schema.xsd)

2.1)Doctorul ce solicita operatia:

```
<physician>
<fullName>Georgescu Ion</fullName>
<stencilNo>424334</stencilNo>
</physician>
```

2.2) Tipul de relatie dintre solicitant si subiect:

```
<relationType>CARER</relationType>
```

pentru relatie de custodie, sau

```
<relationType>CHILD</relationType>
```

pentru legatura parinte-copil.

2.3) Persoana ce solicita cererea (reprezentantul)

```
<requestPerson>

<icExpiration>2014-01-31T00:00:00.000+02:00</icExpiration>

<icNumber>24534323</icNumber>

<icSeries>rt</icSeries>

<personData>

       <birthDate>2012-11-06T00:00:00.000+02:00</birthDate>

<cid>23443432</cid>

<firstName>Ion</firstName>

<lastName>Popescu</lastName>

</personData>

</requestPerson>
```

Descriere date solicitant:

- icExpiration – data exiprarii CI
- icSeries – serie CI
- icNumber – numar CI
- birthDate – data nasterii
- cid – cid
- firstName – prenume
- lastName – nume

2.4)Persoana pentru care se face cererea (reprezentatul)

```
<subject>
```

```
        <birthDate>2014-01-31T00:00:00.000+02:00</birthDate>

        <cid>23443432</cid>

        <firstName>Vasile</firstName>

        <lastName>Ionescu</lastName>

</subject>
```

Descriere date subiect pentru care se solicita crearea unui dosar:

- birthDate –data nasterii
- cid – cid
- firstName – prenume
- lastName – nume

XML-ul este apoi comprimat (zip) si encodat Base64

3) Semnatura continutului:

```
<pkcs7signature>MIAGCSqGSIb3DQEHAqCAMIACAQExD...</pkcs7signature>
```

- pkcs7signature este semnatura detasata base64 pt sirul de octeti prezentat de archivedDocument

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

Transmitere document medical

Exemplu apel SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

        <S:Header>

            <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">



        <challengeResponse>ZkZyfRvBcDkG77xCAe1ZJ0jWR8ycCamj+eOTmT4TwlE64GVq2R3ZOjU8j+HgvX
        Io</challengeResponse>

                    <clientCode>testApp1</clientCode>

            </ns2:desClientSoftwareAuthentication>

        </S:Header>

        <S:Body>

            <ns2:storeClinicalDocumentS xmlns:ns2="core.des.uti.ro">

                    <storeClinicalDocument>
```

```
        <archivedDocument>eJwryi8tSVVITElRMDQ30jO01DPQM1DITSzOVjAyNdU...</archivedDocumen
        t>


        <pkcs7signature>MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFA...</pkcs7signat
        ure>

                    </storeClinicalDocument>

                </ns2:storeClinicalDocumentS>

        </S:Body>

</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>

        <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">


        <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY39l
        Cm</challengeResponse>

            <clientCode>testApp1</clientCode>

        </ns2:desClientSoftwareAuthentication>

</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
storeClinicalDocument#2013-11-23T12:34:23
```

2) **O versiune arhivata a documentului:**

```
<archivedDocument>eJzFUtFqgzAU/...</archivedDocument>
```

Explicatie:

Se arhiveaza documentul (zip) si se encodeaza base64.

3) **Semnatura continutului:**

```
<pkcs7signature>MIAGCSqGSIb3DQEHAqCAMIACAQExD...</pkcs7signature>
```

- pkcs7signature este semnatura detasata base64 pt sirul de octeti prezentat de archivedDocument

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

    <soapenv:Body>

        <dlwmin:storeClinicalDocumentSResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

            <return/>

        </dlwmin:storeClinicalDocumentSResponse>

    </soapenv:Body>

</soapenv:Envelope>
```

Anulare document medical

Exemplu apel SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

       <S:Header>

            <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">



       <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlwWfLdK3BApfA6yqZFpNA
       tR</challengeResponse>

                   <clientCode>testApp1</clientCode>

            </ns2:desClientSoftwareAuthentication>

       </S:Header>

       <S:Body>

            <ns2:removeDocumentSetSxmlns:ns2="core.des.uti.ro">

                   <removeClinicalDocumentRequest>

                          <archivedDocument>eJzFUtFqgzAU/...</archivedDocument>


            <pkcs7signature>MIAGCSqGSIb3DQEHAqCAMIACAQExD...</pkcs7signature>

                   </removeClinicalDocumentRequest >

            </ns2:removeDocumentSetS>

       </S:Body>

</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>
       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">

       <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY391
       Cm</challengeResponse>
            <clientCode>testApp1</clientCode>
       </ns2:desClientSoftwareAuthentication>
</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
removeDocumentSet#2013-11-23T12:34:23
```

2) O versiune arhivata a forumularului de initializare dosar:

```
<archivedDocument>eJzFUtFqgzAU/...</archivedDocument>
```

Explicatie:

Se construieste un xml de forma:

```
<removeClinicalDocumentRequest>
       <authorName>Georgescu Ion</authorName>
<authorStencilCode>424334</authorStencilCode>
<dateTime>2012-11-06T00:00:00.000+02:00</dateTime>
<documentType>67852-4</documentType>
<uicMedicalServiceProvided>123456</uicMedicalServiceProvided>
<patientCID>12345678900987654321</patientCID>
<removedDocumentID>f47ac10b-58cc-4372-a567-0e02b2c3d479</removedDocumentID>
</removeClinicalDocumentRequest>
```

Observatie:

Detalii despre structura XML-ul care trebuie impachetat si semnat pentru a face aceasta operatie gasiti in anexa atasata documentatiei (RemoveClinicalDocumentRequest_schema.xsd)

Descriere:

authorName – Numele doctorului

authorStencilCode – Parafa doctor

dateTime – Data solicitarii anularii documentului

uicMedicalServiceProvided – CUI furnizor de servicii medicale

removedDocumentID – Id-ul documentului ce va fi anulat

documentType – Tipul documentului ce va fi anulat

patientCID – CID-ul pacientului pentru care se solicita anularea documentului

XML-ul este apoi comprimat (zip) si encodat Base64

3) **Semnatura continutului:**

```
<pkcs7signature>MIAGCSqGSIb3DQEHAqCAMIACAQExD...</pkcs7signature>
```

- pkcs7signature este semnatura detasata base64 pt sirul de octeti prezentat de archivedDocument

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

Interogare document medical pe baza de ID

Exemplu apel SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

       <S:Header>

            <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">



       <challengeResponse>j1Wgkn4svSyV5HbbqhaONFBJvqUgyXFnrQqmwa0DEijURec2jKV8sOtniDsQ/b
       5W</challengeResponse>

                   <clientCode>testApp1</clientCode>

            </ns2:desClientSoftwareAuthentication>

       </S:Header>

       <S:Body>

            <ns2:getClinicalDocumentS xmlns:ns2="core.des.uti.ro">

                   <clinicalDocumentRequest>

                          <desDocumentUUID>4312321</desDocumentUUID>

                          <patientCid>12345678901234567890</patientCid>

                          <physicianStencilNo>123123</physicianStencilNo>

                   </clinicalDocumentRequest>

            </ns2:getClinicalDocumentS>

       </S:Body>

</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>

       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">



       <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY39l
       Cm</challengeResponse>

            <clientCode>testApp1</clientCode>

       </ns2:desClientSoftwareAuthentication>

</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
getClinicalDocument#2013-11-23T12:34:23
```

2) **Parametrii necesari aducerii documentului:**

```
<clinicalDocumentRequest>

<desDocumentUUID>4312321</desDocumentUUID>

<patientCid>12345678901234567890</patientCid>

<physicianStencilNo>123123</physicianStencilNo>

</clinicalDocumentRequest>
```

Descriere:

- desDocumentUUID – uuid document pentru care se face cererea
- patientCid – cid pacient

physicianStencilNo – parafa doctorului ce efectueaza cererea

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

    <soapenv:Body>

        <dlwmin:getClinicalDocumentSResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

            <return>

                <archivedDocument>KIESA76ASDIUSADI...</archivedDocument>

            </return>

        </dlwmin:getClinicalDocumentSResponse>

    </soapenv:Body>
```

```
</soapenv:Envelope>
```

Continutul decomprimat reprezinta documentul clinic

Interogare trimiteri medicale ale pacientului pentru o specialitate

Exemplu:

```
<?xml version="1.0" encoding="UTF-8"?><soap:Envelope
       xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <soap:Header>

        <desClientSoftwareAuthentication xmlns="core.des.uti.ro">

                          <challengeResponse>
       LpEqQq/0u2C3hIFEq+d8a0+J6IQXbz67wZA8tRC8hcB0xj+cYHBorKHqetcnybxZ</challengeRespon
       se>

            <clientCode>testApp1</clientCode>

        </desClientSoftwareAuthentication>

    </soap:Header>

    <soap:Body>

        <getRelevantReferrals xmlns="core.des.uti.ro">

            <clinicalDocumentRequest xmlns="">

                <patientCid>12345678901234567890</patientCid>

                <physicianSpecialtyCode>Medicina Familie</physicianSpecialtyCode>

                <physicianStencilNo>324234</physicianStencilNo>

            </clinicalDocumentRequest>

        </getRelevantReferrals>

    </soap:Body>

</soap:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<soap:Header>

        <desClientSoftwareAuthentication xmlns="core.des.uti.ro">

                          <challengeResponse>
       LpEqQq/0u2C3hIFEq+d8a0+J6IQXbz67wZA8tRC8hcB0xj+cYHBorKHqetcnybxZ</challengeRespon
       se>

            <clientCode>testApp1</clientCode>

        </desClientSoftwareAuthentication>
```

```
    </soap:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
getRelevantReferrals#2013-11-23T12:34:23
```

2) Parametrii necesari pentru cerere:

```
        <soap:Body>
       <getRelevantReferrals xmlns="core.des.uti.ro">
           <clinicalDocumentRequest xmlns="">
               <patientCid>12345678901234567890</patientCid>
               <physicianSpecialtyCode>Medicina Familie</physicianSpecialtyCode>
               <physicianStencilNo>324234</physicianStencilNo>
           </clinicalDocumentRequest>
       </getRelevantReferrals>
   </soap:Body>
```

Descriere:

- patientCid – cid pacient
- physicianSpecialtyCode – codul specialitatii
- physicianStencilNo – parafa doctorului ce efectueaza cererea

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<ns2:getRelevantReferralsResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<return xmlns:ns2="core.des.uti.ro">
           <relevantReferralsResponse>
                  <documents>
                         <documentReferralMetadata>
                                <authorStencilNo>23423423</authorStencilNo>

           <medicalServiceProdiderName>324235325</medicalServiceProdiderName>
```

```
                                <documentUUID>londsas-dsk-asdasd-sad</documentUUID>

                                <documentType>67852-4</documentType>

                                <createTime>2014-02-15T00:00:00.000+02:00</createTime>

                                <effectiveTime>2014-02-
      15T00:00:00.000+02:00</effectiveTime>


          <medicalServiceProdiderUic>324234</medicalServiceProdiderUic>

                                <patientCid>12345678901234567890</patientCid>

                                <documentNumber>76</documentNumber>

                                <documentSeries>1</documentSeries>

                        </documentReferralMetadata>

                        <documentReferralMetadata>

                                ...

                        </documentReferralMetadata>

                        ...

                </documents>

          </relevantReferralsResponse>

      </return>

      </ns2:getRelevantReferralsResponse>

</soapenv:Body>

</soapenv:Envelope>
```

Raspunsul reprezinta o lista de metadate pentru documente

Fiecare intrare contine:

authorStencilNo -> parafa doctor

medicalServiceProdiderName ->nume spital

documentUUID -> UUID document

documentType -> tip document

createTime -> data crearii

effectiveTime -> data

medicalServiceProdiderUic ->UIC spital

patientCid -> cid pacient

documentNumber ->numar document

documentSeries ->serie document

Documente emise

Exemplu:

```
<?xml version="1.0" encoding="UTF-8"?>

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

<S:Header>

<ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">

<challengeResponse>JcARv7JRPhrnWxaS952Zy15wjbGr9RQs5Wp6Ru1p...<challengeResponse>

<clientCode>testApp1</clientCode>

</ns2:desClientSoftwareAuthentication>

</S:Header>

<S:Body>

<ns2:getPhysicianClinicalDocuments xmlns:ns2="core.des.uti.ro">

            <clinicalDocumentRequest>

                <documentType>68834-1</documentType>

                <patientCid>40114157747268213406</patientCid>

                <startDate>2014-01-01T00:00:00.000+02:00</startDate>

                <endDate>2014-07-01T00:00:00.000+02:00</endDate>

                <uicMedicalServiceProvider>20576955</uicMedicalServiceProvider>

                <physicianStencilNo>221628</physicianStencilNo>

            </clinicalDocumentRequest>

</ns2:getPhysicianClinicalDocuments>

</S:Body>

</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>

       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">



       <challengeResponse>LpEqQq/0u2C3hIFEq+d8a0+J6IQXbz67wZA8tRC8hcB0xj+cYHBorKHqetcnyb
       xZ</challengeResponse>

            <clientCode>testApp1</clientCode>

       </ns2:desClientSoftwareAuthentication>
```

```
</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
getPhysicianClinicalDocuments#2013-11-23T12:34:23
```

2) Parametrii necesari pentru cerere:

```
<S:Body>
<ns2:getPhysicianClinicalDocuments xmlns:ns2="core.des.uti.ro">
            <clinicalDocumentRequest>
                <documentType>68834-1</documentType>
                <patientCid>40114157747268213406</patientCid>
                <startDate>2014-01-01T00:00:00.000+02:00</startDate>
                <endDate>2014-07-01T00:00:00.000+02:00</endDate>
                <uicMedicalServiceProvider>20576955</uicMedicalServiceProvider>
                <physicianStencilNo>221628</physicianStencilNo>
            </clinicalDocumentRequest>
</ns2:getPhysicianClinicalDocuments>
</S:Body>
```

Descriere:

- patientCid – cid pacient
- documentType – tipul documentului
- uicMedicalServiceProvider – UIC servicu medical
- startDate – data inceput
- endDate – data stop
- physicianStencilNo – parafa doctor

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<ns2:getPhysicianClinicalDocumentsResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<return xmlns:ns2="core.des.uti.ro">
                    <clinicalDocuments>
                           <documentMetadata>
                                  <authorStencilNo>23423423</authorStencilNo>

            <medicalServiceProdiderName>324235325</medicalServiceProdiderName>
                                  <documentUUID>londsas-dsk-asdasd-sad</documentUUID>
                                  <documentType>67852-4</documentType>
                                  <createTime>2014-02-15T00:00:00.000+02:00</createTime>
                                  <effectiveTime>2014-02-
       15T00:00:00.000+02:00</effectiveTime>

            <medicalServiceProdiderUic>324234</medicalServiceProdiderUic>
                                  <patientCid>12345678901234567890</patientCid>
                                  <documentNumber>76</documentNumber>
                                  <documentSeries>1</documentSeries>
                           </documentMetadata>
                           <documentMetadata>
                                  ...
                           </documentMetadata>
                           ...
                    </clinicalDocuments>
            </return>
       </ns2:getPhysicianClinicalDocumentsResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Raspunsul reprezinta o lista de metadate pentru documente

Fiecare intrare contine:

authorStencilNo -> parafa doctor

medicalServiceProviderName -> nume spital

documentUUID -> UUID document

documentType -> tip document

createTime -> data crearii

effectiveTime -> data

medicalServiceProdiderUic -> UIC spital

patientCid -> cid pacient

documentNumber -> numar document

documentSeries -> serie document

Interogare documente vechi

Exemplu:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Header>
        <desClientSoftwareAuthentication xmlns="core.des.uti.ro">
            <challengeResponse xmlns="">
                LpEqQq/0u2C3hIFEq+d8a0+J6IQXbz67wZA8tRC8hcB0xj+cYHBorKHqetcnybxZ
            </challengeResponse>
            <clientCode xmlns="">testApp1</clientCode>
        </desClientSoftwareAuthentication>
</S:Header>
<S:Body>
        <getMedicalFileOlderDocuments xmlns="core.des.uti.ro">
            <olderDocumentsRequest xmlns="">
                <endDate>2013-11-15T00:00:00+02:00</endDate>
                <patientCid>40876229893134684887</patientCid>
                <startDate>2013-10-16T00:00:00+03:00</startDate>
                <physicianStencilNo>985632</physicianStencilNo>
            </olderDocumentsRequest>
        </getMedicalFileOlderDocuments>
</S:Body>
</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>
        <desClientSoftwareAuthentication xmlns="core.des.uti.ro">
```

```
            <challengeResponse xmlns="">
                LpEqQq/0u2C3hIFEq+d8a0+J6IQXbz67wZA8tRC8hcB0xj+cYHBorKHqetcnybxZ
            </challengeResponse>
            <clientCode xmlns="">testApp1</clientCode>
        </desClientSoftwareAuthentication>
</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
getMedicalFileOlderDocuments#2013-11-23T12:34:23
```

2) Parametrii necesari pentru cerere:

```
<S:Body>
        <getMedicalFileOlderDocuments xmlns="core.des.uti.ro">
            <olderDocumentsRequest xmlns="">
                <endDate>2013-11-15T00:00:00+02:00</endDate>
                <patientCid>40876229893134684887</patientCid>
                <startDate>2013-10-16T00:00:00+03:00</startDate>
                <physicianStencilNo>985632</physicianStencilNo>
            </olderDocumentsRequest>
        </getMedicalFileOlderDocuments>
</S:Body>
```

Descriere:

- patientCid – cid pacient
- documentType – tipul documentului
- startDate – data inceput
- endDate – data stop
- physicianStencilNo – codul de parafa al medicului care face interogarea

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
```

```
<ns2: getMedicalFileOlderDocumentsResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<return xmlns:ns2="core.des.uti.ro">

                  <clinicalDocuments>

                         <documentMetadata>

                                <authorStencilNo>23423423</authorStencilNo>


           <medicalServiceProdiderName>324235325</medicalServiceProdiderName>

                                <documentUUID>londsas-dsk-asdasd-sad</documentUUID>

                                <documentType>67852-4</documentType>

                                <createTime>2014-02-15T00:00:00.000+02:00</createTime>

                                <effectiveTime>2014-02-
       15T00:00:00.000+02:00</effectiveTime>


           <medicalServiceProdiderUic>324234</medicalServiceProdiderUic>

                                <patientCid>12345678901234567890</patientCid>

                                <documentNumber>76</documentNumber>

                                <documentSeries>1</documentSeries>

                         </documentMetadata>

                         <documentMetadata>

                                ...

                         </documentMetadata>

                         ...

                  </clinicalDocuments>

           </return>

       </ns2:getMedicalFileOlderDocumentsResponse>

</soapenv:Body>

</soapenv:Envelope>
```

Raspunsul reprezinta o lista de metadate pentru documente

Fiecare intrare contine:

authorStencilNo -> parafa doctor

medicalServiceProdiderName -> nume spital

documentUUID -> UUID document

documentType -> tip document

createTime -> data crearii

effectiveTime -> data

medicalServiceProdiderUic -> UIC spital

patientCid -> cid pacient

documentNumber -> numar document

documentSeries -> serie document

Interogare date consolidate

Sumar de urgenta

La fel ca **Preluare antecedente** doar body-ul difera:

```
<S:Body>
       <ns2:getConsolidatedSummaryS xmlns:ns2="core.des.uti.ro">
            <consolidatedSummaryRequest>
                   <patientCid>12345678901234567890</patientCid>
                   <physicianStencilNo>234234</physicianStencilNo>
            </consolidatedSummaryRequest>
       </ns2:getConsolidatedSummaryS>
</S:Body>
```

precum si numele operatiei folosit la creare tokenului pentru autentificarea software:

```
getConsolidatedSummary#2013-11-23T12:34:23
```

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <dlwmin:getConsolidatedSummarySResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <return>
                <archivedDocument>KIESA76ASDIUSADI...</archivedDocument>
            </return>
        </dlwmin:getConsolidatedSummarySResponse>
    </soapenv:Body>
```

```
</soapenv:Envelope>
```

Continutul decomprimat reprezinta documentul clinic

1.1.1.1 Istoric medical

La fel ca **Preluare antecedente** doar body-ul difera:

```
<S:Body>
       <ns2:getConsolidatedMedicalHistoryS xmlns:ns2="core.des.uti.ro">
           <consolidatedMedicalHistory>
                  <patientCid>12345678901234567890</patientCid>
                  <physicianStencilNo>234234</physicianStencilNo>
           </consolidatedMedicalHistory>
       </ns2:getConsolidatedMedicalHistoryS>
</S:Body>
```

precum si numele operatiei folosit la creare tokenului pentru autentificarea software:

```
getConsolidatedMedicalHistory#2013-11-23T12:34:23
```

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <dlwmin:getConsolidatedMedicalHistorySResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
           <return>
               <archivedDocument>KIESA76ASDIUSADI...</archivedDocument>
           </return>
        </dlwmin:getConsolidatedMedicalHistorySResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Continutul decomprimat reprezinta documentul clinic

1.1.1.2 Preluare antecedente

Exemplu apel SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
       <S:Header>
            <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">

       <challengeResponse>Cq6m8MZJIvNMn3JS/ZGzKN3r6xE2VefYru9jvb8enLNwHId5sOTIauWYmjbvza
       Aq</challengeResponse>
                   <clientCode>testApp1</clientCode>
            </ns2:desClientSoftwareAuthentication>
       </S:Header>
       <S:Body>
            <ns2:getConsolidatedAntecedentsS xmlns:ns2="core.des.uti.ro">
                   <consolidatedAntecedentsRequest>
                          <patientCid>12345678901234567890</patientCid>
                          <physicianStencilNo>234234</physicianStencilNo>
                   </consolidatedAntecedentsRequest>
            </ns2:getConsolidatedAntecedentsS>
       </S:Body>
</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>
       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">

       <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY39l
       Cm</challengeResponse>
            <clientCode>testApp1</clientCode>
       </ns2:desClientSoftwareAuthentication>
</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
getConsolidatedAntecedents#2013-11-23T12:34:23
```

2) **Datele necesare interogarii**

```
<consolidatedAntecedentsRequest>
```

```
<patientCid>12345678901234567890</patientCid>

<physicianStencilNo>234234</physicianStencilNo>

</consolidatedAntecedentsRequest>
```

Descriere:

- patientCid – cid pacient
- physicianStencilNo – parafa doctor

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

    <soapenv:Body>

        <dlwmin:getConsolidatedAntecedentsSResponse xmlns:dlwmin="core.des.uti.ro"


       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

            <return>

                <archivedDocument>KIESA76ASDIUSADI...</archivedDocument>

            </return>

        </dlwmin:getConsolidatedAntecedentsSResponse>

    </soapenv:Body>

</soapenv:Envelope>
```

Continutul decomprimat reprezinta documentul clinic

1.1.1.3 **Documente medicale**

La fel ca **Preluare antecedente** doar body-ul difera:

```
<S:Body>

       <ns2:getConsolidatedEventsHistoryS xmlns:ns2="core.des.uti.ro">

            <consolidatedEventsHistory>

                   <patientCid>12345678901234567890</patientCid>

                   <physicianStencilNo>234234</physicianStencilNo>

            </consolidatedEventsHistory>

       </ns2:getConsolidatedEventsHistoryS>

</S:Body>
```

precum si numele operatiei folosit la creare tokenului pentru autentificarea software:

```
getConsolidatedEventsHistory#2013-11-23T12:34:23
```

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <dlwmin:getConsolidatedEventsHistorySResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <return>
                <archivedDocument>KIESA76ASDIUSADI...</archivedDocument>
            </return>
        </dlwmin:getConsolidatedEventsHistorySResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Continutul decomprimat reprezinta documentul clinic

1.1.1.4 **Date personale**

La fel ca **Preluare antecedente** doar body-ul difera:

```
<S:Body>
       <ns2:getConsolidatedPatientIdentityS xmlns:ns2="core.des.uti.ro">
            <consolidatedPatientIdentityRequest>
                   <patientCid>1234567890123456789O</patientCid>
                   <physicianStencilNo>234234</physicianStencilNo>
            </consolidatedPatientIdentityRequest>
       </ns2:getConsolidatedPatientIdentityS>
</S:Body>
```

precum si numele operatiei folosit la creare tokenului pentru autentificarea software:

```
getConsolidatedPatientIdentity#2013-11-23T12:34:23
```

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <dlwmin:getConsolidatedPatientIdentitySResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <return>
                <archivedDocument>KIESA76ASDIUSADI...</archivedDocument>
            </return>
        </dlwmin:getConsolidatedPatientIdentitySResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Continutul decomprimat reprezinta documentul clinic

Genereaza pozitie sugerata din matricea de securitate

Exemplu apel SOAP:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
       <S:Header>
            <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">

       <challengeResponse>LpEqQq/0u2C3hIFEq+d8a0+J6IQXbz67wZA8tRC8hcB0xj+cYHBorKHqetcnyb
       xZ</challengeResponse>
                   <clientCode>testApp1</clientCode>
            </ns2:desClientSoftwareAuthentication>
       </S:Header>
       <S:Body>
            <ns2:suggestMatrixCoordinates xmlns:ns2="core.des.uti.ro">
                   <patientCid>12345678901234567890</patientCid>
                   <physicianStencilNo>C55428</physicianStencilNo>
            </ns2:suggestMatrixCoordinates>
       </S:Body>
</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>

       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">



       <challengeResponse>LpEqQq/0u2C3hIFEq+d8a0+J6IQXbz67wZA8tRC8hcB0xj+cYHBorKHqetcnyb
       xZ</challengeResponse>

            <clientCode>testApp1</clientCode>

       </ns2:desClientSoftwareAuthentication>

</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
suggestMatrixCoordinates#2013-11-23T12:34:23
```

2) **Parametrii necesari pentru a genera coordonatele:**

```
<ns2:suggestMatrixCoordinates xmlns:ns2="core.des.uti.ro">

<patientCid>12345678901234567890</patientCid>

<physicianStencilNo>C55428</physicianStencilNo>

</ns2:suggestMatrixCoordinates>
```

Descriere:

- patientCid – cid pacient
- physicianStencilNo – parafa doctorului ce efectueaza cererea

Raspuns:

Servicul intoarce o exceptie SOAP in caz de esec.

In caz de succes se va intoarce un raspuns:

```
<?xml version="1.0" encoding="UTF-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

<soapenv:Body>

<ns2:suggestMatrixCoordinatesResponse xmlns:dlwmin="core.des.uti.ro"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<return xmlns:ns2="core.des.uti.ro">

            <matrixPos>

                  <XCoord>3</Xcoord>

                  <YCoord>4</Ycoord>

            </matrixPos>
```

```
        </return>

        </ns2:suggestMatrixCoordinatesResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Raspunsul reprezinta coordonatele unei pozitii din matrice ce pot fi folosite in procesul de autentificare

- **Xcoord** reprezinta numarul liniei din matrice: 1, 2, 3, etc
- **Ycoord** reprezinta numarul coloanei din matrice, valoarea primita corespunde unei litere asociata coloanei din matrice. 1 ->A, 2 ->B, 3 ->C.

Exemplu:

X=1 y=1(A)	X=1 y=2(B)						X=1 y=8(H)
X=2 y=1(A)	X=2 y=2(B)						X=2 y=8(H)
X=8 y=1(A)	X=8 y=2(B)						X=8 y=8(H)

Dupa introducerea valorii corecte corespunzatoare pozitiei, acea pozitie este „arsa” adica nu mai poate fi refolosita decat un anumit interval de timp configurat pe server (de exemplu 10 minute). Daca se doreste efectuarea altor operatii dupa trecerea acestui interval de timp trebuie utilizata alta pozitie „nearsa” care poate fi obtinuta prin mecanismul de sugestie pozitii.

Fiecare pozitie este „arsa” in functie de medic si nu pentru toti medicii. Exemplu: daca Medic 1 utilizeaza pozitia (X,Y):(1,A) acesta dupa trecerea celor N minute nu o mai poate utiliza, dar un alt medic, Medic 2, o poate utiliza fara probleme, pentru acesta din urma fiind „nearsa”.

In procesul de sugestie de pozitii precum si in cel de autentificare pot aparea urmatoarele evenimente/exceptii:

1. Sugestie pozitii:

    a. NO_MORE_DIGITS – inseamna ca pentru medicul curent toate pozitiile sunt arse si este nevoie de o regenerare a matricei de securitate care se face conform protocolului stabilit, descris la capitolul corespunzator. Aceasta exceptie mai poate insemna ca pacientul nu are generata o matrice de securitate inca, care conduce in acelasi proces de generare a matricei de securitate pentru pacientul in cauza.

    b. StencilNo inexistent: se incearca sugestia de pozitie pentru o parafa inexistenta

    c. cid inexistent: se incearca sugestia unei pozitii din matrice folosind un CID inexistent.

2. Raspunsuri la autentificare:

    a. SecurityMatrixBussinessException: Parametrii de apel metoda gresiti – daca unul din parametrii de autentificare lipsesc: pacinet,medic sau valoarea corespunzatoare din matricea de securitate.

    b. SecurityMatrixBussinessException: Coordonatele sunt gresite – daca una sau amandoua coordonatele sunt gresite.

    c. BRUTE_FORCE: in caz ca toate autentificarile incercate consecutive (in numar de 5) intr-un timp de N(=10) minute au fost gresite. In acest caz trebuie sa se astepte N(=10) min pentru a putea face o noua incercare. Ori ce incercare facuta intr-un timp mai scurt, chiar daca datele sunt corecte, este respinsa, codul returnat fiind BRUTE_FORCE.

d. INCORRECT: este returnat in cazul cand datele introduse pentru autentificare sunt incorecte.

d. DUPLICATE: in caz ca se incearca refolosirea aceleiasi pozitii dincolo de cele N(=10) minute permise dupa ce aceasta a fost arsa (prima autentificare corecta pentru acea pozitie din matrice)

e. CORRECT: este returnat in momentul cand toate datele introduse sunt corecte, dupa acest moment aceasta pozitie poate fi refolosita timp de N(=10) minute.

Export cataloage

1.1.1.5 Sumar cataloage exportabile

Exemplu:

```
<?xml version="1.0" encoding="UTF-8"?>

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

<S:Header>

<ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">

<challengeResponse>oDLxRAB0gpm5Q5LIrxL70gvuuLtxnBTVindSbZ14NqzdjrYj9KPI6ANCr+cBpOwl</cha
       llengeResponse>
```

```
<clientCode>testApp1</clientCode>

</ns2:desClientSoftwareAuthentication>

</S:Header>

<S:Body>

       <ns2:exportSystemCodesSummary xmlns:ns2="core.des.uti.ro"/>

</S:Body>

</S:Envelope>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>

       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">

<challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY39lCm</cha
       llengeResponse>

            <clientCode>testApp1</clientCode>

       </ns2:desClientSoftwareAuthentication>

</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
exportSystemCodesSummary#2013-11-23T12:34:23
```

Raspuns:

- In caz de exceptie serviciul intoarce o eroare SOAP
- In caz de succes este intors raspunsul:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

    <soapenv:Body>

        <dlwmin:exportSystemCodesSummaryResponse xmlns:dlwmin="core.des.uti.ro"


       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

            <return>

                <dlwmin:systemCodesSummary>

                    <dlwmin:summaryDate>2014-04-10 17:53:26</dlwmin:summaryDate>

                    <dlwmin:systemCodes>

                        <dlwmin:codeSystem>


       <dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.25</dlwmin:codeSystem>
```

```
<dlwmin:codeSystemName>FamilyMemberRelationType</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>


<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.26</dlwmin:codeSystem>


<dlwmin:codeSystemName>OtherSectionCodes</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>


<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.19</dlwmin:codeSystem>

                    <dlwmin:codeSystemName>DischargeType</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2010-11-13
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>


<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.20</dlwmin:codeSystem>


<dlwmin:codeSystemName>AdmissionTypes</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2000-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>


<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.14</dlwmin:codeSystem>


<dlwmin:codeSystemName>MedicalDeviceLaterality</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>
```

```
<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.15</dlwmin:codeSystem>


<dlwmin:codeSystemName>ProsthesisTypes</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2014-02-27
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>


<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.29</dlwmin:codeSystem>

                    <dlwmin:codeSystemName>VaccineTypes</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>

                    <dlwmin:codeSystem>2.16.840.1.113883.5.1</dlwmin:codeSystem>


<dlwmin:codeSystemName>AdministrativeGender</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>


<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.33</dlwmin:codeSystem>

                    <dlwmin:codeSystemName>BloodABO</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>


<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.34</dlwmin:codeSystem>

                    <dlwmin:codeSystemName>BloodRH</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>
```

```
<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.41</dlwmin:codeSystem>

<dlwmin:codeSystemName>MedicalProcedureTypes</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-02-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>

<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.37</dlwmin:codeSystem>

<dlwmin:codeSystemName>SectionTypeSources</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2014-03-11
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>

<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.21</dlwmin:codeSystem>

                    <dlwmin:codeSystemName>LocationType</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2013-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>

<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.16</dlwmin:codeSystem>

<dlwmin:codeSystemName>DischargeStatus</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2000-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>

                <dlwmin:codeSystem>

<dlwmin:codeSystem>2.16.840.1.113883.3.3368.6.30</dlwmin:codeSystem>

                    <dlwmin:codeSystemName>DeathCause</dlwmin:codeSystemName>

                    <dlwmin:lastModifyDate>2000-01-01
00:00:00</dlwmin:lastModifyDate>

                </dlwmin:codeSystem>
```

```
                    </dlwmin:systemCodes>

                </dlwmin:systemCodesSummary>

            </return>

        </dlwmin:exportSystemCodesSummaryResponse>

    </soapenv:Body>

</soapenv:Envelope>
```

1.1.1.6 Export catalog dupa nume

Exemplu:

```
<?xml version="1.0" encoding="UTF-8"?>

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

<S:Header>

<ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">

<challengeResponse>vLuTpnZhO3yM4lVqQzpS+42ZHf3j2nYOiaYbWDyTbBGiXfMDRfRT3fMwfo+306UB</cha
       llengeResponse>

<clientCode>testApp1</clientCode>

</ns2:desClientSoftwareAuthentication>

</S:Header>

<S:Body>

<ns2:exportCodeSystem xmlns:ns2="core.des.uti.ro">

<exportCodeSystemRequest>

<codeSystemName>AdministrativeGender</codeSystemName>

</exportCodeSystemRequest>

</ns2:exportCodeSystem>

</S:Body>
```

Descriere:

1) **Headerul SOAP ce contine valorile necesare autentificarii software:**

```
<S:Header>

       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">



       <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY39l
       Cm</challengeResponse>

            <clientCode>testApp1</clientCode>
```

```
        </ns2:desClientSoftwareAuthentication>
</S:Header>
```

Exemplu de token pentru challengeResponse ce trebuie criptat:

```
exportCodeSystem#2013-11-23T12:34:23
```

2) Parametrii necesari pentru identificarea catalogului:

```
<S:Body>
<ns2:exportCodeSystem xmlns:ns2="core.des.uti.ro">
<exportCodeSystemRequest>
<codeSystemName>AdministrativeGender</codeSystemName>
</exportCodeSystemRequest>
</ns2:exportCodeSystem>
</S:Body>
```

Descriere:

CodeSystemName – nume catalog

Raspuns:

In caz de exceptie serviciul intoarce o eroare SOAP

In caz de succes este intors raspunsul:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<ns2:exportCodeSystemResponse xmlns:dlwmin="core.des.uti.ro"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<return xmlns:ns2="core.des.uti.ro">
<archivedFile>eJy9k01uwyAQha...</archivedFile>
</return>
</ns2:exportCodeSystemResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Raspunsul reprezinta continutul comprimat al unui catalog

Continutul decomprimat are forma:

```
<?xml version="1.0" ?>
<codeSystem>
        <codeSystemMetadata>
```

```
            <codeSystem>2.16.840.1.113883.3.3368.6.37</codeSystem>

            <codeSystemName>SectionTypeSources</codeSystemName>

            <description>SectionTypeSources</description>

            <exportDate>2014-02-26 16:16:56</exportDate>

            <lastModifyDate>2014-02-25 00:00:00</lastModifyDate>

        </codeSystemMetadata>

        <values>

            <value validFrom="2014-01-04 00:00:00.0" description="testtrecv"
        code="1"></value>

            <value validFrom="2014-02-25 00:00:00.0" description="SectÃunea 1"
        code="1889-1"></value>

            <value validFrom="2014-01-04 00:00:00.0" description="testtrecv"
        code="1"></value>

            <value validFrom="2014-01-04 00:00:00.0" description="testtrecv"
        code="1"></value>

            <value validTo="2014-01-16 15:34:02.0" validFrom="2014-01-01 00:00:00.0"
        description="test" code="2"></value>

            <value validTo="2014-01-01 00:00:00.0" validFrom="2014-01-01 00:00:00.0"
        description="test" code="111"></value>

            <value validTo="2014-01-01 00:00:00.0" validFrom="2014-01-01 00:00:00.0"
        description="test" code="111"></value>

            <value validFrom="2014-02-25 00:00:00.0" description="Sectiunea 1"
        code="1889-1"></value>

        </values>

</codeSystem>
```

Observatie: Detalii despre structura XML-ul care trebuie impachetat si semnat pentru a face aceasta operatie se regasesc in anexa atasata documentatiei (CatalogExport_schema.xsd)

Exceptii SOAP

Sunt de doua feluri:

- **DesException** – aruncata in momentul in care datele sunt incorecte/invalide (contine motivul pentru care apelul a fost respins)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
           <soapenv:Body>
                  <soapenv:Fault>
                         <faultcode>soapenv:Server</faultcode>
                         <faultstring>Medicul autentificat nu...</faultstring>
                         <detail>
                                <ns2:DesRuntimeException xmlns:ns2="core.des.uti.ro">
                                       <code>REQ-VAL-ERROR-1</code>
                                       <message>Medicul autentificat nu este
      corespunde cu medicul din request.{msgId=36357P}
                                       </message>
                                </ns2:DesRuntimeException>
                         </detail>
                  </soapenv:Fault>
           </soapenv:Body>
      </soapenv:Envelope>
```

- **DesRuntimeException** – aruncata in momentul in care are loc o exceptie pe server (nu a putut fi accesata o resursa, probleme tehince...).

```
<?xml version="1.0" encoding="UTF-8"?>

<soapenv:Envelope xmlns:soapenv = "http://schemas.xmlsoap.org/soap/envelope/">

    <soapenv:Body>

        <soapenv:Fault>

            <faultcode>soapenv:Server</faultcode>

            <faultstring>Eroare interna pe server</faultstring>

            <detail>

                <ns2:DesRuntimeException xmlns:ns2 = "core.des.uti.ro">

                    <code>INTERNAL-ERROR</code>

                    <message>Operatia nu a putut fi efectuata din cauza unor erori
      interne. Contactati echipa de suport: suport.des@casan.ro.
      {msgId=46674C}</message>

                </ns2:DesRuntimeException>

            </detail>

        </soapenv:Fault>

    </soapenv:Body>

</soapenv:Envelope>
```

Observatie:

Fiecare identificator de mesaj este sufixat de o particula care indica mediul pe care s-a inregistrat problema:

P - productie

C- certificare

Structural exceptia este compusa din:

- faultcode: codul ce descrie eroarea
- faultstring: contine un preview al mesajului de eroare(numarul de caractere este limitat)
- detail
    - code: contine codul de eroare intern al aplicatiei (Ex: DMR_AGGREG-4, CDA_STORE-2, ...)
    - message: contine intrega descriere a exceptiei

Important

Elementul msgId plasat la sfarsitul mesajului de eroare din „detail” este utilizat in DES pentru a asigura trasabilitatea erorii.

Este obligatoriu ca acesta sa fie furnizat in cazul raportarii unor probleme catre suportul DES.

Valori posibile pentru faultcode sunt:

Cod	Descriere
VersionMismatch	Namespace invalid in mesajul SOAP
Client	Mesajul a fost format incorect sau contine informatie incorecta
Server	Serverul nu a putut procesa mesajul

Valorilele posibile ale detail/code sunt:

Prefix Cod	Descriere
DMR-AGGREG	Exceptie la agregare
DMR-CONSOL	Exceptie la consolidare
ACCESS	Exceptie la autorizare
CDA-VLD-ENT	Exceptie la validarea entitatii din document

CDA-VLD-SEM	Exceptie la validarea semantica a documentului
EHR-INIT	Exceptie la initializarea dosarului
CDA-VLD-CAT	Exceptie la validarea catalogului
CDA-VLD-STR	Exceptie la validarea structurala
CDA-TPL	Sablon gresit
EHR-STAT	Fisier medical invalid
SEC	Exceptie de securitate
AUDIT	Exceptie la auditarea datelor
AUTH	Exceptie la autentificare
EHR-DATA	Date medicale gresite
SGN-VLD	Semnatura invalida
SGN	Exceptie la utilizarea semnaturii
CDA-STORE	Exceptie la salvarea documentului
MATRIX	Exceptie la folosirea matricei
MATRIX-SEC	Exceptie la operatiile ce implica matricea de securitate
PKI	Exceptie la validarea certificatului
PERS-REL	Relatie gresita intre persoane
MATRIX-VLD	Nu mai exista pozitii valide in matricea de securitate
EHR-SIPE	Eroare la apelarea serviciului SIPE
EHR-INIT-EXT	Exceptie la initializarea inregistrarilor medicale
GENERAL-ERROR	Exceptie generala DES
APP-CERT-ERROR	Exceptie la verificarea certificatului software
REQ-VAL-ERROR	Exceptie la validarea cererii
MSG_STORE_ERROR	Exceptie la salvarea cererii/raspunsului

CAT-EXPORT	Exceptie la exportul de cataloage
CEAS	Exceptie la coautentificarea cu CEAS

5. Cod exemplificativ si informatii privind abordarea integrarii

DES se bazeaza pe doua tehnologii importante:

- Servicii web ca mecanism de interactiune si transport de date
- Documente CDA revision 2 (HL7 v3), in fapt documente XML guvernate de restrictii si conventii specifice documeniului medical

Tehnlogiile de dezvoltare ce permit lucrul cu documentele CDA se bazeaza pe cateva paradigme:

- Message Driven
    - presupuneutilizareaunorlibrariiceasista in constructia CDA-ului
    - automatizeazaintr-o masuraconstructia
- Model Driven
    - presupuneproiectarearestrictiilorasupraschemei CDA in UML (definirere-strictiipe template-uri, reguli de cardinalitate)
    - modelsintacticsisemnaticsuprapuspeste CDA.xsd
- Existasiimplementarimaicomplexe, tip middleware
    - atat open source (eventual cu suportcomercial) cat sicomerciale

Exemple de astfel de tehnologii sunt:

- Message Driven
    - Class generation based on XSD
        - Ex JAXB (Java), XSD.exe (.net)
    - Marc Everest
        - Platforma Microsoft .NET
- Model Driven (MIF sau UML)
    - MDHT
        - Platforma Java
    - JavaSIG

- RIMBAA
- Platforma Java

- Middleware
  - Mirth
  - Majoritateavendorilormari au solutiisauadaptoare

Urmatoareledouacapitoleexpunexemple de cod ceasigurainvocareaoperatiilorimportanteexpuse de sistemul DES.

Exemple .NET

În continuare prezentăm câteva exemple cod sursă .NET/C# pentru conectarea la DES şi transmiterea către acesta a documentelor medicale, dar şi pentru consultarea datelor medicale relevante din dosarul unui pacient.

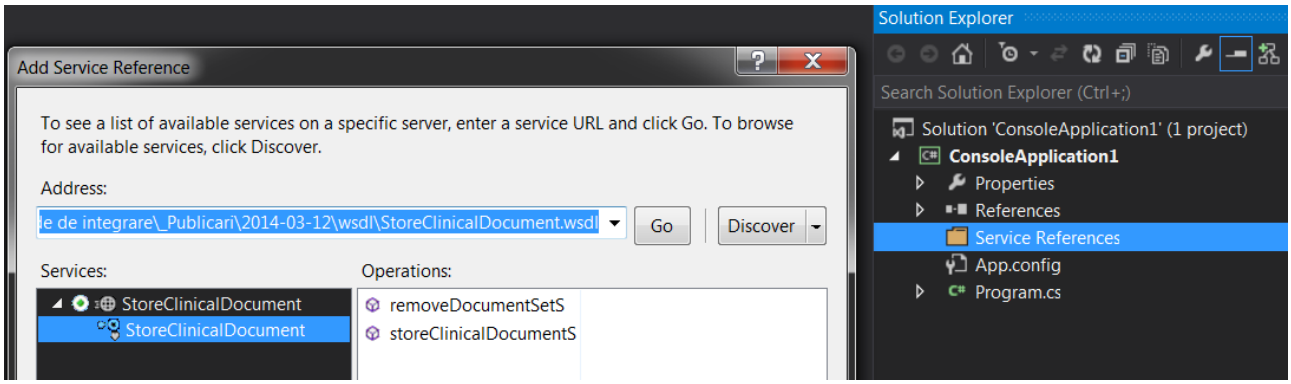
Exemplele folosesc tipurile de date definile în WSDL-urile publice ale DES.

La sfârşitul secţiuni sunt prezentate câteva exemple de cod ajutător pentru realizarea unor operaţii specifice, cum ar fi semnarea electronică a documentului medical, obţinerea codului de autorizare a aplicaţiei şi al *challange*-ului de autorizare a aplicaţiei.

Generarea claselor proxy din WSDL

Clasele proxy pentru serviciile web pot fi generate utilizand wsdl.exe sau wizard-ul din Visual Studio.

Utilizarea Visual Studio 2008/2010/2012/2013



Utilizarea wsdl.exe

Nota: pentru generare trebuie sa furnizati toate fisierele wsdl si xsd necesare. In cazul acesta se vor furniza wsdl-ul si xsd-ul ce are un nume similar.

Spre exemplu

Wsdl.exe ClinicalDocument.wsdl ClinicalDocument_schema1.xsd

Transmitere document medical CDA către DES

```
string CallWebServiceMethod( StoreClinicalDocument webService,
                             string cdaFilePath,
                             X509Certificate2 certificate )
{
    // set auth header
    webService.desClientSoftwareAuthenticationValue = new
       desClientSoftwareAuthentication
        {
            clientCode = DesKeyGen.AppKey(),
            challengeResponse = DesKeyGen.ComputeAuth( "storeClinicalDocument" )
        };
    // prepare method params
    var zipBuffer = ZipArchive.ZipBuffer( File.ReadAllBytes( cdaFilePath ) );
    var signBuffer = DigitalSignature.ApplySignature( certificate, zipBuffer, true );
    // init method params
    var storeClinicalData = new storeClinicalDocumentS
        {
            storeClinicalDocument = new storeClinicalDocumentRequest
                {
                    archivedDocument = zipBuffer,
                    pkcs7signature = signBuffer
                }
        };
    // call ws method
    var response = webService.storeClinicalDocumentS( storeClinicalData );
    // return warnings, if any
    return response.@return.warnings;
}
```

Consultare date medicale relevante din DES

```
string CallWebServiceMethod( ConsolidatedClinicalDocument webService,
                             string dmrFilePath )
```

```
{
    // set auth header
    webService.desClientSoftwareAuthenticationValue = new
       desClientSoftwareAuthentication
        {
            clientCode = DesKeyGen.AppKey(),
            challengeResponse = DesKeyGen.ComputeAuth( "getConsolidatedSummary" )
        };
    // init method params
    var webServiceInput = new getConsolidatedSummaryS
        {v
            consolidatedSummaryRequest = new getConsolidatedSummaryRequest
                {
                    physicianStencilNo = ”123456”, // id of physician
                    patientCid = ”40251225216312512354”, // id of patient
                    patientCoAuthentication = GetPatientCoAuthentication()
                }
        };
    // call ws method
    var response = webService.getConsolidatedSummaryS( webServiceInput );
    // process response
    var response = ZipArchive.UnzipBuffer( response.@return.archivedDocument );
    File.WriteAllBytes( dmrFilePath, response );
    // return contents
    return Encoding.UTF8.GetString( response );
}
```

Consultare documente medicale emise de medic existente în DES

```
documentMetadata[] CallWebServiceMethod( ClinicalDocument webService )
{
    // set auth header
    webService.desClientSoftwareAuthenticationValue = new
       desClientSoftwareAuthentication
```

```
        {
            clientCode = DesKeyGen.AppKey(),
            challengeResponse = DesKeyGen.ComputeAuth( "getPhysicianClinicalDocuments" )
        };
    // init method params
    var clinicalDocumentsRequest = new getPhysicianClinicalDocuments
        {
            clinicalDocumentRequest = new getClinicalDocumentsRequest
                {
                    physicianStencilNo = ”123456”,
                    uicMedicalServiceProvider = ”987654321”,
                    startDate = new DateTime( 2014, 1, 1 ),
                    startDateSpecified = true,
                    endDate = new DateTime( 2014, 1, 31 ),,
                    endDateSpecified = true,
                }
            };
    // call ws method
    var result = webService.getPhysicianClinicalDocuments( clinicalDocumentsRequest );
    // return documents array
    return result.@return.clinicalDocuments;
}
```

Clasă utilitară pentru autorizarea aplicaţiei de raportare

```
public static class DesKeyGen
{
    public static string AppKey()
    {
        return “SIUIMF”; // se completează cu valoarea specifică fiecărei aplicaţii
    }

    public static byte[] ComputeAuth( string methodName )
```

```
    {
        string seed = string.Format( "{0}#{1:yyyy-MM-ddTHH:mm:ss}", methodName,
       DateTime.Now );
        return EncryptDataAES( Encoding.UTF8.GetBytes( seed ) );
    }

    private static byte[] EncryptDataAES( byte[] toEncrypt )
    {
        using( var aes = SymmetricAlgorithm.Create() )
        {
            aes.Mode = CipherMode.CBC;
            aes.Key = Convert.FromBase64String( key ); // key este o constantă de tip
       string
            aes.IV = Convert.FromBase64String( iv ); //iv este o constantă de tip string
            aes.Padding = PaddingMode.PKCS7;
            using( var mStream = new MemoryStream() )
            {
                using( var cStream = new CryptoStream( mStream, aes.CreateEncryptor(),
       CryptoStreamMode.Write ) )
                {
                    cStream.Write( toEncrypt, 0, toEncrypt.Length );
                    cStream.FlushFinalBlock();
                    return mStream.ToArray();
                }
            }
        }
    }
}
```

Exemplu de realizare a semnăturii digitale

```
byte[] ApplySignature( X509Certificate2 certificate,
                                    byte[] message,
                                    bool detached = false )
{
```

```
    // create content info wrapper

    var contentInfo = new ContentInfo( message );

    // create non-detached CMS signed message (to include original content)

    var signedCms = new SignedCms( contentInfo, detached );

    // create CMS signer, using certificate issuer and serial number

    var signer = new CmsSigner( SubjectIdentifierType.IssuerAndSerialNumber, certificate
       );

    // include only subject certificate info (not the whole trust chain)

    signer.IncludeOption = X509IncludeOption.EndCertOnly;

    if( detached ) signer.SignedAttributes.Add( new Pkcs9SigningTime() );

    // sign message with certificate / silent-mode false, to allow propt for password

    signedCms.ComputeSignature( signer, false );

    // encode the CMS message, original content is included in this byte array.

    return signedCms.Encode();

}
```

Notă: această metodă poate fi utilizată şi pentru realizarea semnăturii digitale pentru celelalte sisteme ale platformei PIAS, şi anume SIUI, SIPE şi CEAS folosind parametrul ***detached*** cu valoarea ***false***. Pentru DES, se foloseşte parametrul ***detached*** cu valoarea ***true***, aşa cum se vede în exemplele anterioare.

Exemple de librarii cu care se poate realiza o compresie compatibila ZLIB sunt: DotNetZip - http://dotnetzip.codeplex.com sau #ziplib http://www.icsharpcode.net.

Exemple JAVA

Autentificare Software

Exemplu de constructie a hashului trimis catre DES (Java):

```
//Initializare Vector preluat din certificat

byte iv[] = new byte[] {0,0, 0, 0, 0,0, 0, 0, 0,0, 0, 0, 0, 0, 0, 0};


String keyBase64 = "some key"; //cheia encodata base64 preluata din certificat

Byte[] key = Base64.decodeBase64(keyBase64);


String message = "storeClinicalDocument#2013-11-23T12:34:23";

byte[] symKeyData = key;//DatatypeConverter.parseHexBinary(key);
```

```
SecretKeySpec symKey = new SecretKeySpec(symKeyData, "AES");


//initializeaza cifrul
Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
cipher.init(Cipher.ENCRYPT_MODE, symKey, iv);


// cripteaza mesaj
byte[] encrypted = cipher.doFinal(message.getBytes());
```

Algoritmul de criptare este **AES128**, cypher se realizeaza cu AES/CBC/PKCS5Padding. PKCS7Padding, fiind un subset al PKCS5, poate fi deasemenea folosit.

Se foloseste **CBC** la criptare astfel incat este nevoie de un **Vector de initializare** peru a evita criptarea in mod identic a blocurilor identice.

Rezultatul este pus in headerul mesajului SOAP impreuna cu identificatorul software-ului extern

```
<S:Header>
       <ns2:desClientSoftwareAuthentication xmlns:ns2="core.des.uti.ro">


       <challengeResponse>fPJ91Eg0GXtZ0mSJHyjqTFuQggOzUNfJY+HqQ45gRlxC+xVTQTcyYBUTkuY391
       Cm</challengeResponse>
            <clientCode>testApp1</clientCode>
       </ns2:desClientSoftwareAuthentication>
</S:Header>
```

Compresia date (constructia campului <archivedDocument>)

```
import java.io.ByteArrayOutputStream;
import java.util.zip.DataFormatException;
import java.util.zip.Deflater;
import java.util.zip.Inflater;


public static byte[] compress(byte[] data) {
        Deflater deflater = new Deflater();
        deflater.setInput(data);
        ByteArrayOutputStream outputStream = new ByteArrayOutputStream(data.length);
```

```
        deflater.finish();
        byte[] buffer = new byte[1024];
        while (!deflater.finished()) {
            int count = deflater.deflate(buffer);
            outputStream.write(buffer, 0, count);
        }
        byte[] output = outputStream.toByteArray();
        return output;
}
```

Decompresie date

```
import java.io.ByteArrayOutputStream;
import java.util.zip.DataFormatException;
import java.util.zip.Deflater;
import java.util.zip.Inflater;


public static byte[] decompress(byte[] data) throws DataFormatException {
        Inflater inflater = new Inflater();
        inflater.setInput(data);
        ByteArrayOutputStream outputStream = new ByteArrayOutputStream(data.length);
        byte[] buffer = new byte[1024];
        while (!inflater.finished()) {
            int count = inflater.inflate(buffer);
            outputStream.write(buffer, 0, count);
        }
        byte[] output = outputStream.toByteArray();
        return output;
    }
```

Generare semnatura pkcs7

Semnatura poate fi creata cu unul din algoritmii de hash-ing: sha1, sha256, sha384, sha512. Conform legii trebuie folosit alg de criptare RSA cu unul din alg de hash de mai sus. Semnatura bruta (raw) este adaugata intr-un mesaj alaturi de alte informatii utile pentru verificare, cum ar fi: algoritm semnare, algoritm hash, data semnare, cheia publica. Rezultatul final este un CMS (cryptographic message syntax).

Atentie!
Unele probleme care pot interveni in construirea semnaturii care pot fi cauzate de alterarea datelor (extrem de rar) sau encoding de caractere.
Mesajul CMS trebuie codat base64. In generarea mesajului base64 trebuie folosit charsetul utf8.
Functie de limbajul de programare folosit (Java, C# etc) sa luam exemplul de mai jos:
...
1. byte[] cms = ... // genereaza semnatura
2. String str = new String(cms);
3. String base64 = Base64.encode(str); // semnatura codata base64
...
In acest exemplu, la linia 2 se creeaza un string cu octetii semnaturii, dar constructorul String va folosi charsetul default, care poate fi ISO-8859-1.
Aceasta poate fi o cauza in care semnatura nu poate fi validata, pentru ca la decodare se extrag octetii folosind charsetul UTF-8 cand se decodeaza base64.
Din cauza asta se obtine un alt CMS decat cel transmis si apare o eroare la validarea semnaturii.

```
import java.io.FileInputStream;

import java.io.InputStream;

import java.security.KeyStore;

import java.security.PrivateKey;

import java.security.Security;

import java.security.cert.Certificate;

import java.security.cert.X509Certificate;

import java.util.ArrayList;

import java.util.List;

import org.bouncycastle.cert.jcajce.JcaCertStore;

import org.bouncycastle.cms.CMSProcessableByteArray;

import org.bouncycastle.cms.CMSSignedData;

import org.bouncycastle.cms.CMSSignedDataGenerator;

import org.bouncycastle.cms.CMSTypedData;

import org.bouncycastle.cms.jcajce.JcaSignerInfoGeneratorBuilder;

import org.bouncycastle.jce.provider.BouncyCastleProvider;

import org.bouncycastle.operator.ContentSigner;

import org.bouncycastle.operator.jcajce.JcaContentSignerBuilder;
```

```
import org.bouncycastle.operator.jcajce.JcaDigestCalculatorProviderBuilder;

import org.bouncycastle.util.Store;

import org.bouncycastle.util.encoders.Base64;


public final class Signer {

    private static final String PATH_TO_KEYSTORE = "/path/to/keyStore";

    private static final String KEY_ALIAS_IN_KEYSTORE = "My_Private_Key";

    private static final String KEYSTORE_PASSWORD = "MyPassword";

    private static final String SIGNATUREALGO = "SHA1withRSA";


    public Signer() {

    }

    KeyStore loadKeyStore() throws Exception {


        KeyStore keystore = KeyStore.getInstance("JKS");

        InputStream is = new FileInputStream(PATH_TO_KEYSTORE);

        keystore.load(is, KEYSTORE_PASSWORD.toCharArray());

        return keystore;

    }

    CMSSignedDataGenerator setUpProvider(final KeyStore keystore) throws Exception {

        Security.addProvider(new BouncyCastleProvider());

        Certificate[] certchain = (Certificate[])
       keystore.getCertificateChain(KEY_ALIAS_IN_KEYSTORE);

        final List<Certificate> certlist = new ArrayList<Certificate>();

        for (int i = 0, length = certchain == null ? 0 : certchain.length; i < length;
       i++) {

            certlist.add(certchain[i]);

        }


        Store certstore = new JcaCertStore(certlist);

        Certificate cert = keystore.getCertificate(KEY_ALIAS_IN_KEYSTORE);

        ContentSigner signer = new
       JcaContentSignerBuilder(SIGNATUREALGO).setProvider("BC").
```

```
                build((PrivateKey) (keystore.getKey(KEY_ALIAS_IN_KEYSTORE,
       KEYSTORE_PASSWORD.toCharArray())));

        CMSSignedDataGenerator generator = new CMSSignedDataGenerator();

        generator.addSignerInfoGenerator(new JcaSignerInfoGeneratorBuilder(new

       JcaDigestCalculatorProviderBuilder().setProvider("BC").build()).build(signer,
       (X509Certificate) cert));



        generator.addCertificates(certstore);

        return generator;

    }



    byte[] signPkcs7(final byte[] content, final CMSSignedDataGenerator generator)
       throws Exception {

        CMSTypedData cmsdata = new CMSProcessableByteArray(content);

        CMSSignedData signeddata = generator.generate(cmsdata, true);

        return signeddata.getEncoded();

    }

}
```

Generare header HTTP „Authorization”

```
import org.apache.commons.io.IOUtils;

import org.apache.commons.codec.binary.Base64;

import java.io.ByteArrayInputStream;



public String getAuthorizationHeader(String username, String password) {

       String basic = username + ":" + password;

       String headerBasicAuth = "Basic " + IOUtils.toString(new
       ByteArrayInputStream(Base64.encodeBase64(basic.getBytes())));

       return headerBasicAuth;

}

//rezultatul se pune pe headerele HTTP cu cheia "Authorization"
```