

Standarde de integrare Dosarul Electronic de Sanatate

CONTROLUL DOCUMENTELOR

Istoric Modificări

Nr crt.	Data	Versiune	Justificare modificare
1	26.02.2014	1	Versiunea initiala
2	4.03.2014	2	Actualizare pentru publicare
3	28.03.2014	3	Precizari privind schemele CDA revision 2 Precizari privind sursa fiecarui codesystem si utilizarea urlencode pentru elementele code din CDA. Indicatii privind algoritmul de compresie. Includerea in pachet a wsdl-ului pentru ManageMedicalFile.wsdl
4	24.07.2014	4	Explicatii suplimentare privind compunerea tokenului de autentificare software Explicatii suplimentare privind generarea semnaturii in format pkcs7 Explicatii privind particula caracter de identificare a unui mesaj de eroare Explicatii suplimentare privind utilizarea matricei de securitate si a serviciilor web prin care poate fi utilizata Alte corectii privind structura mesajelor SOAP

Nota: Modificarile efectuate in ultima revizie sunt evidentiatare cu **font albastru**.

CUPRINS

Controlul Documentelor	2
Istoric Modificări	2
Introducere	5
Obiectiv document	5
Aplicabilitate	5
Continutul pachetului de documentatie	5
Funcționalitățile suportate prin Interfața Automată	6
Inregistrare apartinator	6
Transmiterea documentelor medicale	6
Anulare document medical	8
Interogare document medical	8
Interogare trimiteri medicale ale pacientului pentru o specialitate	9
Interogare listă documente emise de un medic	9
Interogare documente vechi	10
Interogare date consolidate (Date medicale relevante – DMR)	10
Sumarul de urgenta (EmergencySummaryOutDocument)	10
Istoricul medical (MedicalHistoryOutDocument)	10
Antecedente declarate de pacient (ReportedMedicalHistoryOutDocument)	11
Documente medicale (MedicalEventsOutDocument)	11
Date personale (PatientInformationOutDocument)	11
Genereaza pozitie sugerata din matricea de securitate	11
Export cataloage	11
Autorizare	11
Autentificare FSM	12
Autentificare medic	12
Autenticitate	13
CoAutentificarea pacient	13
CoAutentificare pacient utilizand CEAS	14
CoAutentificare pacient utilizand matricea de securitate	15
Autentificare client software	16
Servicii web	19
Securitatea operatiilor	19
Tipuri de date des utilizate	21

Implementarea CDA in Dosarul Electronic de Sanatate.....	21
Tipul de surse de date pentru dosarul de sanatate.....	23
Operatiile.....	23
Inregistrare apartinator.....	25
Transmitere document medical	28
Anulare document medical	30
Interogare document medical pe baza de ID.....	31
Interogare trimiteri medicale ale pacientului pentru o specialitate	33
Documente emise	35
Interogare documente vechi	38
Interogare date consolidate	40
Sumar de urgenta	40
Istoric medical	41
Preluare antecedente	42
Documente medicale	43
Date personale	44
Genereaza pozitie sugerata din matricea de securitate.....	45
Export cataloage	48
Sumar cataloage exportabile	48
Export catalog dupa nume.....	51
Exceptii SOAP.....	53
Cod exemplificativ si informatii privind abordarea integrarii.....	57
Exemple .NET	58
Generarea claselor proxy din WSDL.....	58
Transmitere document medical CDA către DES	59
Consultare date medicale relevante din DES	59
Consultare documente medicale emise de medic existente în DES.....	60
Clasă utilitară pentru autorizarea aplicației de raportare.....	61
Exemplu de realizare a semnăturii digitale.....	61
Exemple JAVA	62
Autentificare Software	62
Compresia date (constructia campului <archivedDocument>).....	63
Decompresie date	63
Generare semnatura pkcs7.....	64
Generare header HTTP „Authorization”.....	66

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 separat.

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 implementarea datelor medicale relevante (folosit pentru 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.

Conținutul pachetului de documentație

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 în wsdl).

xslt – scheme de transformare în HTML a documentelor CDA ce respecta ghidul de implementare DES. Aceste transformări sunt furnizate ca atare, cu titlu exemplificativ, spre a fi utilizate ca punct de plecare în afisarea dosarului medical și nu își propun să respecte formularistica aflată în vigoare sau să o înlocuiască.

FUNCȚIONALITĂȚILE SUPORTATE PRIN INTERFAȚA 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 apartinător

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 reprezentantul fără a-i crea dosar și apoi legătura între cele două persoane.

Notă: la crearea 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** se transmit 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 etc.

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 următoarele condiții sunt respectate:

- dacă semnatarul electronic al acestuia se regăsește și printre authenticators sau legalAuthenticator din documentul de înlocuit
- dacă prin schimbarea specialității medicale în baza carei se face documentul nu reflecta o schimbare de profil medic în sensul medic spital sau medic de familie.

- dacă se referă la același CID, dacă specialitatea 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.

Această 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 authenticator" 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

Această funcționalitate permite unei aplicații 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:
 - o Are acces la dosar conferit de politicile de securitate alese de pacient
 - o Are acces la dosar conferit direct de pacient
 - o Este documentul său (este autorul acestuia)
 - o Este unul dintre semnatarii documentului original declarat ca autenticator sau legalAutheticator în CDA
 - o Are acces conferit de pacient pe loc (prin coautentificare 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

Funcționalitatea permite unei aplicații externe să obțină lista de trimiteri medicale pe o anumită specialitate și pentru un anumit pacient.

Pentru ca DES să furnizeze aceste date medicul care face solicitarea trebuie să aibă specialitatea 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

Funcționalitatea 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 primește vreo atenționare, dar intern serviciul limitează perioada maximă de cautare la $N(=30)$ de zile relativ la data de start. N – reprezintă un număr exprimat în zile și 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 autenticată 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.

Antecedente 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.

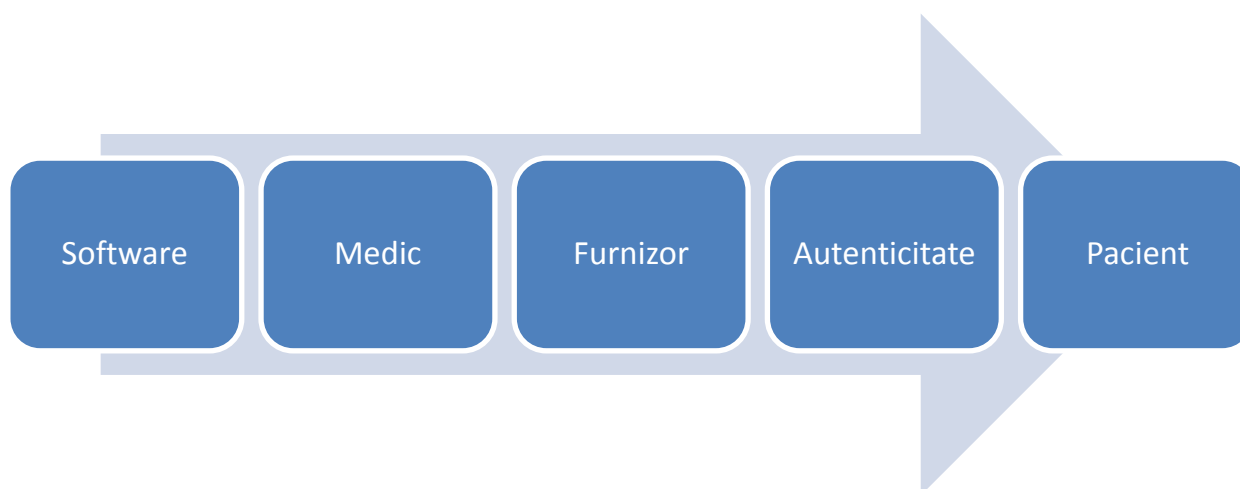
Export cataloage

Funcționalitatea permite unei aplicații terțe să obțină din DES cataloagele utilizate.

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 obținerea unui nou jeton.

Tokenul astfel 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	<p>Cuvantul „Basic” urmat de un token obtinut prin encodarea Base64 a valorilor user si parola concatenate prin „:”</p> <p>Exemplu:</p> <p>User: user1 Parola: pass</p> <p>Se concateneaza valorile: user1:pass</p> <p>Se aplica Base64(„user:pass”) -></p> <p>dXNlcjE6cGFzcw==</p> <p>Valorea trimisa este:</p> <p>Basic dXNlcjE6cGFzcw==</p>

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 legalAuthenticator

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
- Interogările 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 accesibil oricarui medic
- Dosarul integral sau documentele medicale ce constituie dosarul sunt accesibile 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 primește token software (session-id-hash) de la serverul de autentificare OSCP denumit în continuare **ocspToken**
2. Se formează un șir de caractere de forma: `cid|cardNo|evidenceDate|ocspToken`, unde `evidenceDate` este în format `XmlDateTime`.
3. Șirul 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](#)
- Interogările de date consolidate – mai puțin sumarul de urgență

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>GPI50F0FNTU4G3VxWGGMebgfdRFGzwWxXkMfrQ08xpEscJIuftg+...<challengeResponse>
      <clientCode>testAppl</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>11111111111111</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
```

Denumirile 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 -> preluare 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};  
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+xVTQTcyYBUTkuY39lCm</challengeResponse>  
  
    <clientCode>testApp1</clientCode>  
  
  </ns2:desClientSoftwareAuthentication>
```

</S:Header>

SERVICII WEB

DES expune un set de servicii web ce 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 trimeri medicale ale pacientului pentru o specialitate	getRelevantReferrals	Nu	Nu	Nu
Documente emise de un medic	getPhysicianClinicalDo cuments	Nu	Nu	Nu
Interogare documente vechi	getMedicalFileOlderDoc uments	Nu	Da	Da
Preluare antecedente	getConsolidatedAntece dents	Nu	Da	Da
Istoric medical	getConsolidatedMedica lHistory	Nu	Da	Da
Date personale	getConsolidatedPatient Identity	Nu	Da	Da
Sumarul de urgenta	getConsolidatedSumm ary	Nu	Da	Nu
Documente medicale	getConsolidatedEvents History	Nu	Da	Da
Sugereaza pozitie neutilizata din matricea de securitate	suggestMatrixCoordina tes	Nu	Nu	Nu
Sumar cataloage exportabile	exportCatalogsSumma ry	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 encodeate 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 ABO	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	Diagnostic 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 domiciliu	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	Casa 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	DES	2.16.840.1.113883.3.3368.6.21

	consultatie MF		
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 documentele folosite ca sursa de informatie	DES	2.16.840.1.113883.3.3368.6.37

Tipul de surse de date pentru dosarul de sanatate

In toate interogările 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	getClinicalDocumentsS
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 pozitii sugerate din matricea de	/desws/SecurityMatrix/SecurityMatrix.wsdl	SecurityMatrix	SecurityMatrixPort	suggestMatrixCoordinates

securitate				
Interogare trimiteri medicale ale pacientului pentru o specialitate	/desws/ClinicalDocu ment/ClinicalDocum ent.wsdl	ClinicalDocume nt	ClinicalDocum entPort	getRelevantR eferrals
Documente emise	/desws/ClinicalDocu ment/ClinicalDocum ent.wsdl	ClinicalDocume nt	ClinicalDocum entPort	getPhysicianCl inicalDocume nts
Interogare documente vechi	/desws/ClinicalDocu ment/ClinicalDocum ent.wsdl	ClinicalDocume nt	ClinicalDocum entPort	getMedicalFile OlderDocume nts
Sumar cataloage exportabile	/desws/ExportCodin gSystem/ExportCodi ngSystem.wsdl	ExportCodingSy stem	ExportCoding SystemPort	getCatalogsSu mmmary
Export catalogdupa nume	/desws/ExportCodin gSystem/ExportCodi ngSystem.wsdl	ExportCodingSy stem	ExportCoding SystemPort	exportCodeSy stem

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+HqQ45gRlwWfLdK3BApfA6yqZFpNAtR</ch
      allengeResponse>
      <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>
```

```
</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+xVTQTcyYBUTkuY39lCm</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 formularului de initializare dosar:

```
<archivedDocument>eJzFUtFggzAU/...</archivedDocument>
```

Explicatie:

Se construiesc 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>ZkZyfRvBcDkG77xCAelZJ0jWR8ycCamj+eOTmT4TwlE64GVq2R3ZOjU8j+HgvXIo</challengeResponse>
      <clientId>testApp1</clientId>
    </ns2:desClientSoftwareAuthentication>
  </S:Header>
  <S:Body>
    <ns2:storeClinicalDocumentS xmlns:ns2="core.des.uti.ro">
      <storeClinicalDocument>
        <archivedDocument>eJwryi8tSVVITelRMDQ30j001DPQM1DITSzOVjAyNdU...</archivedDocument>
      </storeClinicalDocument>
    </ns2:storeClinicalDocumentS>
  </S:Body>
</S:Envelope>
```

```
<pkcs7signature>MIAGCSqGSib3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFA...</pkcs7signature>
</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+xVTQTcyYBUTkuY39lCm</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>eJzFUtFggzAU/...</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+HqQ45gRlwWfLdK3BApfA6yqZFpNatR</challengeResponse>
      <clientCode>testApp1</clientCode>
    </ns2:desClientSoftwareAuthentication>
  </S:Header>
  <S:Body>
    <ns2:removeDocumentSetS xmlns:ns2="core.des.uti.ro">
      <removeClinicalDocumentRequest>
        <archivedDocument>eJzFUtFggzAU/...</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+xVTQTcyYBUTkuY39lCm</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 formularului de initializare dosar:

```
<archivedDocument>eJzFUtFggzAU/...</archivedDocument>
```

Explicatie:

Se construiesc 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>j1Wgkn4svSyV5HbbqhaONFBJvqUgyXFnrQqmw0DEijUREc2jKV8sOtniDsQ/b5W</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+xVTQTcyYBUTkuY39lCm</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 decompimat 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+cYHBorKHqetcnbybZ</challengeResponse>
      <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+cYHBorKHqetcnbybZ</challengeResponse>
    <clientCode>testApp1</clientCode>
  </desClientSoftwareAuthentication>
</soap:Header>
```

```
</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+cYHBorKHqetcnbyxZ</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 serviciu 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 metadata 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="">testAppl</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:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:dlwmin="core.des.uti.ro"
    >
      <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>
          </documentMetadata>
        </clinicalDocuments>
      </return>
    </ns2:
      getMedicalFileOlderDocumentsResponse
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:dlwmin="core.des.uti.ro"
    >
  </soapenv:Body>
</soapenv:Envelope>
```

```
<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 metadata 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 decompimat reprezinta documentul clinic

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 decompimat reprezinta documentul clinic

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/ZGzKN3r6xE2VefYru9jvb8enLNwHid5sOTIauWYmjbvzaAq</ch
allengeResponse>

    <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+xVTQTcyYBUTkuY39lCm</ch
allengeResponse>

  <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

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 decompimat reprezinta documentul clinic

Date personale

La fel ca **Preluare antecedente** doar body-ul difera:

```
<S:Body>
  <ns2:getConsolidatedPatientIdentityS xmlns:ns2="core.des.uti.ro">
    <consolidatedPatientIdentityRequest>
      <patientCid>12345678901234567890</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>
```

```
</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+cYHBorKHqetcnybxZ</ch  
allengeResponse>  
      <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+cYHBorKHqetcnybxZ</ch  
allengeResponse>  
    <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 etc.

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:

- NO_MORE_DIGITS – inseamna ca pentru medicul curent toate pozitiiile 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.
- StencilNo inexistent: se incearca sugestia de pozitie pentru o parafa inexistentă
- cid inexistent: se incearca sugestia unei pozitii din matrice folosind un CID inexistent.

2. Raspunsuri la autentificare:

- SecurityMatrixBussinessException: Parametrii de apel metoda gresiti – daca unul din parametri de autentificare lipsesc: pacinet, medic sau valoarea corespunzatoare din matricea de securitate.
- SecurityMatrixBussinessException: Coordonatele sunt gresite – daca una sau amandoua coordonatele sunt gresite.
- 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 re folosita timp de N(=10) minute.

Export cataloage

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</challengeResponse>
      <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</challengeResponse>
    <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>

```

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</chal
lengeResponse>
<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+xVTQTcyYBUTkuY39lCm</ch
    allengeResponse>

    <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>eJy9k0luwyAQha...</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="SectĂ-unea 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 tehnice...).

```
<?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
 - o code: contine codul de eroare intern al aplicatiei (Ex: DMR_AGGREG-4, CDA_STORE-2, ...)
 - o 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

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 documentului medical

Tehnologiile de dezvoltare ce permit lucrul cu documentele CDA se bazeaza pe cateva paradigme:

- Message Driven
 - o presupune utilizarea unor librării ce asista in constructia CDA-ului
 - o automatizeaza intr-o masura constructia
- Model Driven
 - o presupune proiectarea restrictiilor asupra schemei CDA in UML (definirea restrictiilor template-uri, reguli de cardinalitate)
 - o model sintactic si semnificativ suprapus peste CDA.xsd
- Există si implementari mai complexe, tip middleware
 - o atat open source (eventual cu suport comercial) cat si comerciale

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
 - Majoritatea vendorilor mari au soluții sau adaptoare

Următoarele două capitole expun exemple de cod ce asigură invocarea operațiilor importante expuse de sistemul DES.

Exemple .NET

În continuare prezentăm câteva exemple cod sursă .NET/C# pentru conectarea la DES și transmiterea către acesta a documentelor medicale, dar și pentru consultarea datelor medicale relevante din dosarul unui pacient.

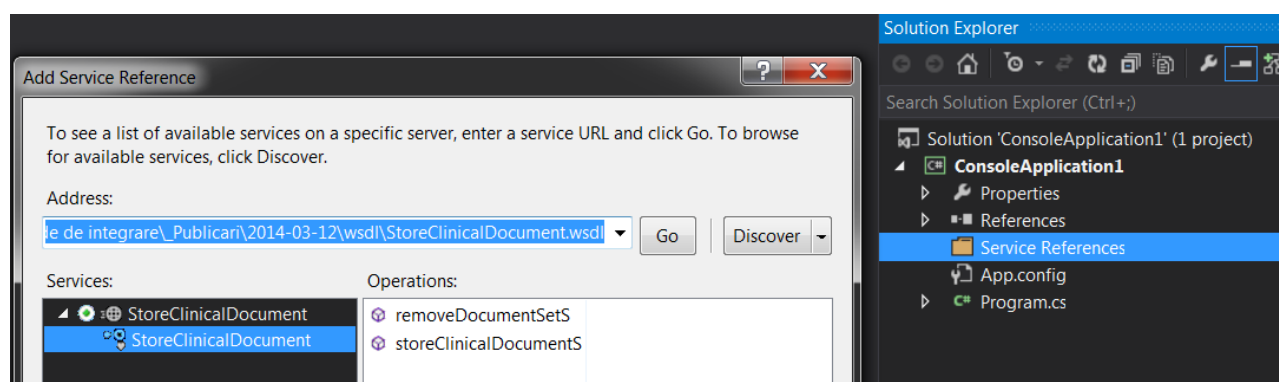
Exemplele folosesc tipurile de date definite în WSDL-urile publice ale DES.

La sfârșitul secțiunii sunt prezentate câteva exemple de cod ajutător pentru realizarea unor operații specifice, cum ar fi semnarea electronică a documentului medical, obținerea codului de autorizare a aplicației și al *challenge*-ului de autorizare a aplicației.

Generarea claselor proxy din WSDL

Clasele proxy pentru serviciile web pot fi generate utilizând `wSDL.exe` sau wizard-ul din Visual Studio.

Utilizarea Visual Studio 2008/2010/2012/2013



Utilizarea `wSDL.exe`

Nota: pentru generare trebuie să furnizați toate fișierele `wSDL` și `xSD` necesare. În cazul acesta se vor furniza `wSDL`-ul și `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 aplicației de raportare

```
public static class DesKeyGen
{
    public static string AppKey()
    {
        return "SIUIMF"; // se completează cu valoarea specifică fiecărei aplicații
    }

    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 și pentru realizarea semnăturii digitale pentru celelalte sisteme ale platformei PIAS, și anume SIUI, SIPE și CEAS folosind parametrul **detached** cu valoarea **false**. Pentru DES, se folosește parametrul **detached** cu valoarea **true**, așa cum se vede în exemplele anterioare.

Exemple de librării cu care se poate realiza o compresie compatibilă ZLIB sunt:
DotNetZip - <http://dotnetzip.codeplex.com> sau #ziplib <http://www.icsharpcode.net>.

Exemple JAVA

Autentificare Software

Exemplu de construcție a hashului trimis către 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};

String keyBase64 = "some key"; //cheia encodată base64 preluată 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+xVTQTcyYBUTkuY39lCm</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 algoritmi 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 etc.

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"
```