Raport stiintific si tehnic in extenso pentru proiectul *Tehnologii de procesare si garantare a continutului electronic - TAPE*

Etapa III-2014

Proiectarea prototipurilor si elaborarea documentatiei tehnice de realizare

Cuprins

1 Introducere

Proiectul „*Tehnologii de procesare si garantare a continutului electronic – TAPE*" isi propune sa dezvolte tehnologii care sa stea la baza unor produse sau servicii ce vor permite acoperirea necesitatilor aflate la convergenta pietelor de tip eDiscovery combinate cu arhivare electronica pe termen lung si garantare de continut. In aceasta etapa a proiectului (etapa numarul trei) – „*Proiectarea prototipurilor si elaborarea documentatiei tehnice de realizare*" – au fost prevazute urmatoarele activitati:

3.1 Proiectarea prototipului pentru componentele tehnice de colectare continutului electronic.

3.2 Proiectarea prototipului pentru componentele tehnice de arhivare si garantare a continutului electronic.

3.3 Proiectarea prototipului pentru serviciul de tip TTP pentru garantarea continutului web.

3.4 Diseminarea rezultatelor.

CertSIGN SA isi propune realizarea si promovarea pe piata a acestor noi tehnologii ce urmaresc procesarea si garantarea continutului electronic, ca element principal al strategiei de diversificare a portofoliului de servicii si produse in domenii noi, inovative folosind in acest sens experienta si maturitatea dobandite in domenii tehnologice specifice.

2 Descrierea etapei si a activitatilor

A doua etapa a proiectului a avut ca obiectiv „ Proiectarea prototipurilor si elaborarea documentatiei tehnice de realizare". Etapa a cuprins urmatoarele activitati si livrabilele aferente:

1. Activitate III.1 Proiectarea prototipului pentru componentele tehnice de colectare continutului electronic, cu livrabilul:
    a. *Specificatiile de proiectare pentru forma de prototip a componentelor de colectare si centralizare de continut.*

2. Activitate III.2 Proiectarea prototipului pentru componentele tehnice de arhivare si garantare a continutului electronic, cu livrabilul:
    a. *Specificatiile de proiectare pentru prototipul componentelor de arhivare si garantare de continut electronic.*

3. Activitate III.3 Proiectarea prototipului pentru serviciul de tip TTP pentru garantarea continutului web, cu livrabilul:
    a. *Specificatiile de proiectare pentru prototipul serviciului de garantare a continutului web.*

4. Activitatea III.4 - Diseminarea rezultatelor, cu urmatoarele livrabile:
    a. *Articol*
    b. *Participare la conferinta internationala*
    c. *Actualizare site web de prezentare a proiectului*
    d. *Document cu arhitectura propusa, publicare in site web dedicat proiectului*

3 Activitatea III.1 Proiectarea prototipului pentru componentele tehnice de colectare continutului electronic

In cadrul activitatii au fost realizate specificatiile de prototip pentru componente tehnice de colectare, structurare, arhivare si garantare a continutului electronic.

Prototipul sistemului de colectare a continutului (eCollectorSystem) este format din urmatoarele componente:

1) e-CollectorAgent

- Va fi dezvoltat ca aplicatii serviciu pentru Windows si Linux.
- Ca arhitectura este un client pentru aplicatia server e-CollectorCenter.
- Realizeaza identificarea si colectarea continutului electronic la nivelul unei statii monitorizate.
- Fiecare instanta va avea o configuratie de identificare a resurselor electronice. Configuratiile sunt sub forma de fisiere XML ce vor contine elemente de filtrare bazate pe numele fisierelor sau a folderelor targetate, cuvinte cheie ce vor fi cautate in documente, data ultimei modificari a fisierelor si dimensiunea fisierelor.
- Fiecare agent va detine un modul specializat pe diverse tipuri de continut pentru a permite cautarea orientata pe tipul continutului (PDF, word, text).
- Realizeaza colectarea si expedierea continutului catre eCollectorCenter
- Pe langa continutul documentelor se vor trimite informatii de tip metadata (owner/author, date range and file type, source OS type, source station id).
- Comunicatia cu centrul de colectare se va face pe canal TCP securizat SSL cu autentificare mutual bazata pe certificate X.509.
- Pentru fiecare resursa colectata este creata o arhiva de tip ZIP care contine documentul colectat si un fisier XML ce contine informatiile de tip metadata.
- Va avea un modul dedicat care va fi activat din configuratie pentru realizeaza semnarii si marcarii temporale a arhivelor inainte de fi trimise catre e-CollectorCenter. Pentru semnatura foloseste formatul de semnatura CAdES-T in forma detasata. Pentru generarea semnaturilor in format CAdES-T foloseste SDK –ul proiectat descris in capitolul urmator.
- Semnarea documentelor colectate se va face optional in functie de configuratia existenta la nivelul agentului. In acest caz, la pornire serviciului acesta va cere autorizarea de acces la o cheie de semnare.

2) e-CollectorCenter

- Arhitectura aplicatiei este de tip serviciu ce ruleaza un server TCP pentru clientii e-CollectorAgent.
- Va fi dezvoltata pentru a fi implementata pe sisteme Linux.
- Va contine un modul pentru comunicatia securizata cu clientii e-CollectorAgent folosind protocolul SSL.
- Va contine un modul responsabil cu receptionarea continutului primit de la clientii e-CollectorAgents.
- Va contine un modul de validare a informatiilor de garantare a continutului electronic (validarea semnaturilor electronice si a marcilor temporale). Acest modul va fi dezvoltat pe baza SDK –ului descris in acest raport.
- Va contine un modul de semnare si marcare a documentelor daca aceasta operatie nu este realizata la client.

- Va avea un modul dedicat pentru realizarea indexarii continutului colectat folosind algoritmi moderni de indexare dupa criterii configurabile.
- Va avea un mdoul de structurare si clasificare a continutului electronic pe baza unor configuratii
- Se interfateaza cu component de arhivare (e-CollectorArchiver) folosind un sistem de fisiere comun si notificari pe baza de fisiere special.

3) e-CollectorArchiver

- Realizeaza arhivarea continutului colectat de perechea e-CollectorAgent si e-CollectorCenter.
- Va avea un modul de adaugare de informatii de garantare pe termen lung folosind formatul de semnatura CAdes-A.
- Realizeaza arhivarea continutului colectat impreuna cu date suplimentare (semnaturile CAdES-A, informatii de indexare, structurare).

Pentru implementarea serviciilor PKI necesare, a fost propusa o arhitectura bazata pe 3 componente:

- Un serviciu PKI de gestiune a certificatelor digitale necesare pentru asigurarea comunicatiei si a operatiilor de generare a semnaturilor electronice
- Un serviciu de validare a certificatelor bazat pe un server OCSP, necesar in operatiile de validare a semnaturilor de componentele propuse in cadrul sistemului
- Un serviciu de marca temporala compatibil cu specificatiile din RFC 3161.

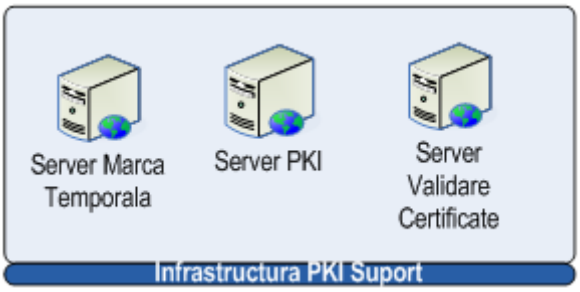

Figure 1 – Infrastructura PKI de suport folosita in cadrul proiectului TAPE

Arhitectura PKI definita are urmatoarele responsabilitati:

- Emiterea certificatelor digitale pentru operatorii care au dreptul de solicitare a informatiilor colectate
- Introducerea de extensii private in certificate pentru a granulariza accesul la informatiile colectate
- Revocarea certificatelor digitale pe masura ce nevoia de acces a disparut
- Actualizarea periodica si la intervale de timp foarte scurte (maxim 24h) a listelor de certificate revocate
- Emiterea de certificate digitale pentru securizarea comunicatiilor intre componentele de tip client-server. Chiar daca nu este posibila utilizarea unor canale securizate de comunicatii end-to-end in toate situatiile este de dorit stabilirea unui canal de comunicatii cu autentificare a serverului (https)
- Implementarea unui serviciu de marcare temporala pentru adaugarea unei astfel de marci pe schemele de hash implementate la nivelul sistemului de stocare si arhivare pe termen lung.

4 Activitatea III.2 Proiectarea prototipului pentru componentele tehnice de arhivare si garantare de continut electronic

Pentru generarea semnaturilor digitale pentru garantarea continutului pe termen lung se vor folosi formatele de semnatura avansata CAdES. Acestea sunt propuse si standardizate de ETSI – *European Telecommunication Standards Institute* in specificatia TS 101 733.

Pentru dezvoltarea componentelor software de garantare a continutului a fost proiectat un SDK structurat pe 2 nivele:

1) Un nivel de baza ce va fi implementat sub forma unei biblioteci (va fi denumita in continuare SSEAPI). Acest API va fi responsabil cu implementarea formatului de baza CMS – *Cryptographic Message Syntax* pentru semnaturile digitale, managementul cheilor de semnare si gestiunea mecanismelor criptografice necesare.
2) Un nivel suplimentar – construit pe functionalitatile oferite de SSEAPI – responsabil cu generarea formatelor de semnatura avansata de tip CAdES.

Componenta SSEAPI (parte a SDK –ului de garantare a continutului pe termen lung) va fi implementata sub forma unei biblioteci DLL/LIB, si va fi dezvoltata folosind limbajul de programare C++. SDK-ul a fost proiectat pentru a putea asigura urmatoarele functionalități:

- Generarea si gestiunea semnăturilor electronice în format CMS – Cryptographic Message Syntax (definit prin standardul RFC 5652).
- Crearea de co-semnături (semnaturi paralele) și contra-semnături (semnături seriale).
- Suport pentru crearea semnăturilor atașate și detașate.
- Suport pentru validarea semnăturilor, co-semnăturilor și contra-semnăturilor conforme CMS.
- Adaugarea de marci temporale conforme cu standardul RFC 3161.
- Suport pentru validarea marcilor temporale.
- Validarea certificatelor digitale X 509v3 și a lantului de certificare folosind protocolul OCSP(Online Certificate Status Protocol) și listele de tip CRL – *Certificate Revocation List*.
- Suport pentru validarea stării certificatelor pe baza informațiile continute în certificatele digitale: extensia AIA (Authority Information Access) și CRLDP(CRL Distribution Point).
- Integrarea cu servicii de directoare de tip LDAPv2.
- Suport pentru lucrul cu chei și certificate stocate pe dispozitive criptografice specializate, de tip PKCS#11 (tokenuri criptografice sau HSM – *Hardware Secure Module*) sau software in fisiere de tip container PKCS#12.
- Algoritmi cu chei publice suporțati: RSA.
- Algoritmi de hash suportați: SHA-512, SHA-256, SHA-1.

În SSEAPI fiecare element din arhitectură urmeaza sa fie modelat de o secventa de trei clase C++. Aceste clase sunt de tip interfața, factory și implementare. Clasele de tip interfața si factory sunt accesibile utilizatorului spre deosebire de clasele de tip implementare care sunt invizibile. In proiectarea acestor clase, se vor folosi urmatoarele conventii: pentru numele interfețelor de forma "ISS_*Element*", pentru numele claselor factory de forma "SS_*Element*", iar pentru numele claselor implementare, de forma "_SS_*Element*". De exemplu, pentru modelarea entitații de tip certificat se vor folosi urmatoarele trei clase: *ISS_Cert, SS_Cert* si *_SS_Cert*.

Clasele de tip factory conțin doar funcții pentru crearea si distrugerea obiectelor entitații respective. Funcțiile pentru creare unui obiect sunt de forma "Create_Instance" iar cele pentru distrugerea obiectelor

se numesc “Destroy_Instance”. Funcțiile “Create_Instance” se ocupa de instanțierea claselor de implementare corespunzatoare și de inițializarea acestora.

Figura 2 – Structura generala de tip factory pentru obiecte

Clasele de tip interfața vor expune funcții publice corespunzatoare operațiile oferite de element. Avand in vedere necesitatile existente in cadrul proiectului, de a lucra cu secvente mari de date (fisiere de dimensiuni mari, arhive, etc.) la nivelul API –ului s-au proiectat o serie de extensii la anumite functii, orientate pentru lucrul cu stream-uri de date. Suportul pentru lucrul cu stream-uri aduce numeroase avantaje printre care si posibilitatea folosirii fișierelor de dimensiune mare. Teoretic, limita maximă pentru dimensiunea fisierelor este limita maximă stabilită de bibliotecta STL – *Standard Template Library*.

O parte din functionalitățile din SSEAPI urmeaza a fi implementate la nivelul unui element sau a unei colectii de elemente. De exemplu operația de creare semnatură electronică implica un numar relativ mare de elemente, comparativ cu operația de afisare a unui atribut dintr-un certificat care implică doar folosirea entității de tip certificat.

SSEAPI a fost proiectat sub forma unei colectii de clase. Mai jos sunt prezentate o parte din aceste clase:

- **SS_P7Signature**. Clasa asigura suportul de nivel inalt pentru gestiunea (crearea, validarea, parsarea actualizarea) semnaturilor digitale in format de baza CMS.
- **SS_P7SignerInfo**. Clasa asigura suportul de nivel inalt pentru gestiunea informatiilor specifice unui semnatar.
- **SS_KeyStore**. Clasa asigura gestiunea cheilor criptografice de semnare la nivelul SDK –ului.
- **SS_Cert**. Clasa este responsabila cu gestiunea certificatelor digitale la nivelul SDK –ului.
- **SS_CertList**. Clasa responsabila cu gestiunea unei liste de certificate.
- **SS_CertValidator**. Clasa este responsabila cu implementarea operatiilor de validare a unui semnatar.

SS_P7Signature

Implementarea va asigura suportul pentru crearea și validarea semnăturilor conforme standardului CMS (Cryptographic Message Syntax) definit în RFC 5652. Acest standard definește sintaxa mesajelor criptografice, inclusiv a semnăturilor electronice, iar mesajele de tip *SignedData* asigura integritatea și autenticitatea datelor prin intermediul semnăturilor electronice. Sintaxa mesajelor criptografice foloseste notația ASN.1.

Codificarea structurilor SS_P7Signature se realizează folosind regulile BER – Basic Encoding Rules și DER – Distinguished Encoding Rules, definite în standardul ITU-T X.690. Elementul *P7Signature* propune modelarea unui mesaj criptografic de tip *SignedData*. Secvența de clase folosite pentru modelare este: *ISS_P7Signature, SS_P7Signature* si *_SS_P7Signature*. Clasa factory ofera funcții de instantiere folosind un

buffer de memorie, un fișier sau un stream de date, precum și funcție necesară pentru distrugerea instanțelor create.

Figura 3 – Diagrama de clase entitati SS_P7Signature

Un obiect de tipul SS_P7Signature este un container de una sau mai multe semnături. Pentru a adauga o semnătura noua sau o contra-semnătura la o semnătura existenta se folosesc funcțiile *Add_NewSigner* si *Add_NewCounterSigner*. Aceste funcții returneaza o entitate de tipul SS_P7SignerInfo care are ca scop modelarea unei semnaturi propriu-zisă. Pentru parcurgerea listei de entitați SS_P7SignerInfo, sunt disponibile funcțiile *Get_SignerCounter* și *Get_Signer*.

SS_P7SignerInfo

Elementul *P7SignerInfo* reprezintă modelarea structurii *SignerInfo* prezenta in format CMS. Secvența de clase folosită este: *ISS_P7SignerInfo, SS_ P7SignerInfo* și *_SS_ P7SignerInfo*. Clasa *factory* nu va pune la dispozitie funcții de instanțiere pentru că nu are sens existența unui semnatar în afara entitatii *P7Signature*. Instanțierea se va realiza in fapt prin intermediul funcțiilor *Add_NewSigner* și *Add_NewCounterSigner* iar obiectele se distrug odată cu obiectul *P7Signature* corespunzător.

Figura 4 – Diagrama de clase entitati SS_P7SignerInfo

SS_Cert si **SS_CertList**

Clasa *Cert* modeleaza entitati de tip certificate digital, folosite in contextual generarii sau verificarii semnaturilor electronice. Secvența de clase pentru modelarea elementului Cert este: ISS_Cert, SS_Cert si _SS_Cert. Clasa de tip factory ofera funcții de instanțiere folosind un buffer de memorie sau un fișier precum și funcție necesară pentru distrugerea instanțelor create. Elementul *CertList* este folosit pentru modelarea unei liste de certificate.

Figura 5 – Diagrama de clase entitati SS_Cert si SS_CertList

SS_KeyStore

Responsabila cu gestiunea cheilor criptografice in cadrul SDK –ului. Oferă suport pentru realizarea operațiilor criptografice folosind chei private stocate pe dispozitive hardware (token criptografic si HSM) compatibile cu standardul PKCS11 sau chei private stocata într-un fișier de tip PKCS12.

Diagrama de clase din figura urmatoare arată relatia de asociere între clasele implicate în modelarea elementului KeyStore.

Figura 6 – Diagrama de clase entitati KeyStore

Validarea unei semnături electronice presupune atât verificarea integritatii informatiei criptografice de semnatura propriu-zisa dar și validarea stării certificatului digital al semnatarului.

Validarea stării certificatelor implica:

- verificarea caii de certificare și
- verificarea starii de revocare a certificatelor

Verificarea căii de certificare presupune verificarea certificatului din punct de vedere al structurii sale cât și a încrederii oferite. In contextual SSEAPI, verificarea se va face recursiv pentru certificatele autorităților emitente pană când se identifică o autoritate de încredere sau până când certificatul autorității este autosemnat.

Verificarea stării de revocare presupune in general fie interogarea unor servere de OCSP – Online Certificate Status Protocol sau analizarea unor liste de tip CRL – Certificate Revocation List în vederea determinării faptului că certificatul era revocat sau valid (nerevocat) la momentul creării semnăturii. SSEAPI a fost proiectat sa ofere suport pentru ambele metode iar entitatea de baza in contextual unor functionalitați de validare este *CertValidator*.

SS_CertValidator

Elementul CertValidator, modelat prin clasele ISS_CertValidator, SS_CertValidator si _SS_CertValidator, oferă suport atât pentru validarea căii de certificare cât și pentru validarea stării certificatelor folosind OCSP sau CRL. Funcțiile propuse sunt *Validate_CertPath* și respectiv *Validate_RevStatus*. Datele de configurare, precum adresa serverului de OCSP și lista de CRL-uri, La nivelul claselor vor putea fi specificate explicit sau pot fi preluate automat din extensiile dedicate – AIA si CRLDP – existente la nivelul certificatelor.

Figura 7 – Diagrama de clase entitati CertValidator

Pașii necesari pentru validarea unei semnături electronice CMS in contextual lui SSEAPI sunt (în paranteze sunt menționate elementele implicate):

- Pasul 1 : Se crează o instanța a unui obiect P7Signature (*P7Signature*).
- Pasul 2 : Se apelează una din funcțiile de verificare corespunzatoare obiectului P7Signature sau P7SignerInfo(*P7Signature,P7SignerInfo*).
- Pasul 3: Se obține certificatul sau o referintă la certificatul semnatarului *(P7SignerInfo*).
- Pasul 4: Se adauga autoritațile de încredere (*CertValidator*).
- Pasul 5: Se constuiește calea de certificare (*CertValidator*).
- Pasul 6: Se verifică starea de revocare a certificatului semnatarului (*CertValidator*).

Specificaţia tehnică ETSI defineşte şapte profile diferite pentru semnăturile electronice de tip CAdES:

- CAdES-BES (Basic Electronic Signature) – asigură cerinţele de bază pentru o semnătură electronică avansată.

- CAdES-EPES (Explicit Policy-based Electronic Signature) – completează CAdES-BES şi permite adăugarea unei politici de semnare.

- CAdES-T (Time-stamped) – completează CAdES-BES sau CAdES-EPES şi permite adăugarea unei mărci temporale pentru a asigura non-repudiabilitatea. În acest mod se poate proba că semnătura a fost creată înainte de un moment de timp dat.

- CAdES-C (Complete) – completează CAdES-T şi permite adăugarea informaţiilor de validare (referinţe la certificatele din calea de certificare precum şi la informaţiile de revocare) pentru a asigura valabilitatea pe termen lung a semnăturii electronice.

- CAdES-X (Extended) – completează CAdES-C şi permite protejarea informaţiilor de validare prin marcarea temporală a referinţelor acestora.

- CAdES-XL (Extended Long) – completează CAdES-X şi permite adăugarea informaţiilor **complete** de validare (căile de certificare şi informaţiile de revocare necesare).

- **CAdES-A (Archive)** – completează CAdES-XL şi permite protejarea periodică a informaţiilor complete de validare prin (re)marcarea lor temporală. Permite asigurarea valabilităţii semnăturii electronice pe termen foarte lung.

In vederea arhivarii si garantarii pe termen lung a continutului electronic se va folosi formatul de semnatura avansata de tip CAdES-A, propus de ETSI in acest sens. Pentru implementarea acestuia sunt necesare implementarile pentru toate celelalte formate intermediare.

Nivelul 2 al SDK-ului ce va fi folosit in componentele de garantare a continutului pe termen lung implementeaza formatelor de semnatura avansata propuse de ETSI. El va fi dezvoltat de asemenea ca un API (SSEAPI_CAdES) peste SSEAPI si va contine urmatoarele clase:

- SS_CAdESBes
- SS_CAdEST
- SS_CAdESC
- SS_CAdESX1 si SS_CAdESX2
- SS_CAdESXL
- **SS_CAdESA**

Arhitectura claselor proiectata pentru aceasta componenta este prezentata in Figura 8. In figura sunt evidentiate doar o parte din clasele proiectate pentru acest nivel al SDK –ului.

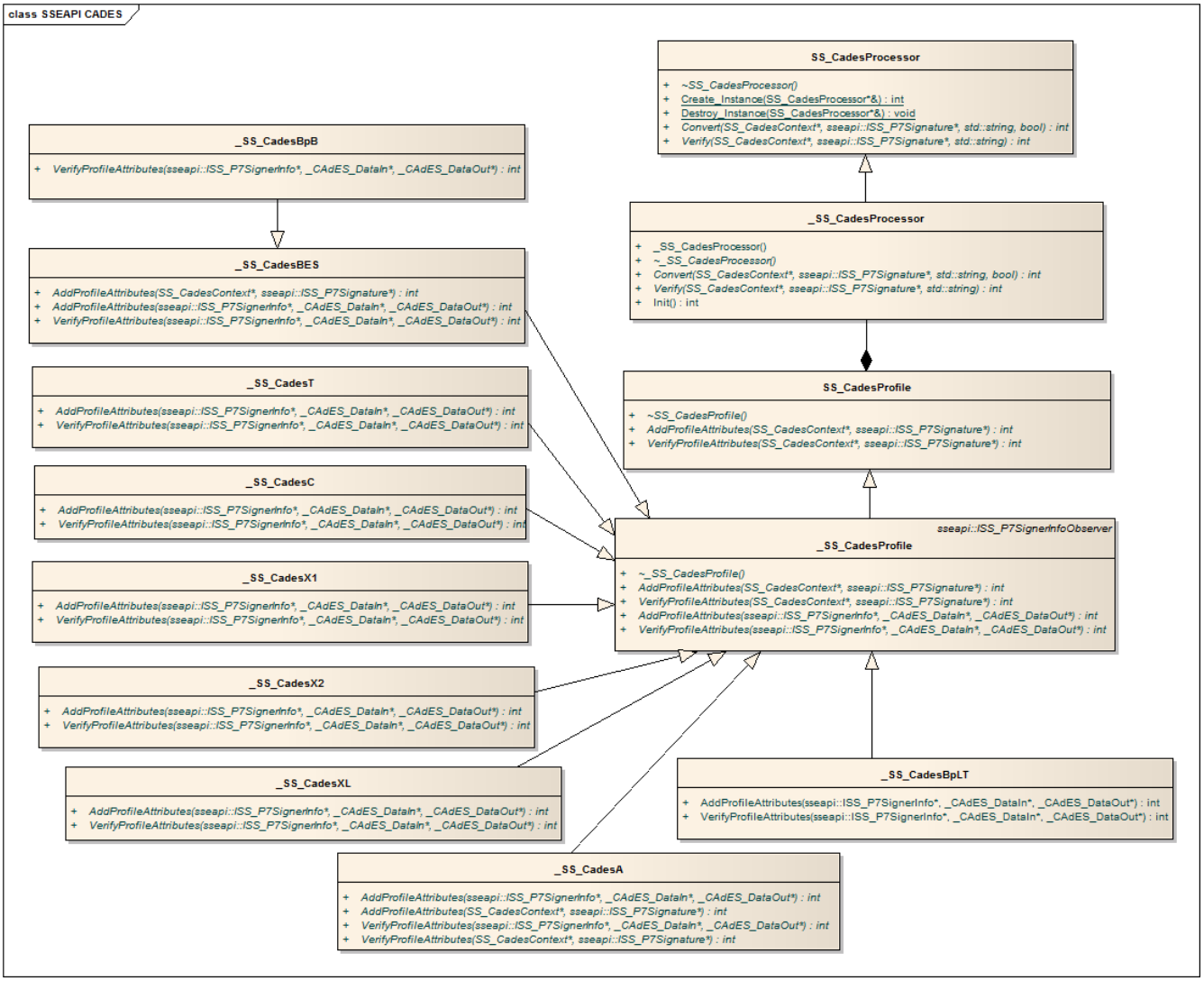

Figura 8 – Diagrama de clase componente SSEAPI_CADES

5 Activitatea III.3 Proiectarea prototipului pentru serviciul de tip TTP pentru garantarea continutului web, cu livrabilul

In aceasta activitate a fost realizata proiectarea pentru prototipul serviciului TTP responsabil cu garantarea continutului Web. In urma analizei realizate in etape anterioare din cadrul proiectului a rezultat necesitatea ca acest serviciu sa permita doua scenarii de utilizare:

1) Scenariul de tip *Site-Request* in care furnizorul continutului web foloseste serviciul TTP pentru ca acesta sa garanteze la un moment de timp existent, formatul si autenticitatea continutului furnizat.
2) Scenariul de tip *Client-Request* in care un client al unui continut web (site) foloseste serviciul TTP pentru a garanta pe termen lung existenta si formatul unui unui continut vizitat.

Avand in vedere cele doua scenarii au rezultat doua solutii arhitecturale, ce vor fi integrate in acelasi serviciu TTP:

1) Solutia de tip *Site-Request*

Solutia propune garantarea continutului web publicat prin metode cum ar fi site–urile web ale organizatiilor (corporaţii, agentii ale statului, instituţii publice, agenţii de presă, etc.), bloguri, continut de tip social-media, etc. Pe langa colectarea si arhivarea continutului site-urilor web, sistemul isi propune abordarea si rezolvarea unor provocari specifice, datorate pe de o parte dinamismului informatiei publicate si, pe de alta parte a diferentelor structurale intre modul de prezentare a informatiei catre utilizatori si modul de stocare la nivelul bazelor de date din spatele serverelor de publicare. Multitudinea tehnologiilor folosite in acest moment pentru publicarea informatiei web, caracaterul interactiv al paginilor si site –urilor de prezentare constituie elemente de dificultate in colectarea acestor informatii in vederea arhivarii lor pentru a putea fi apoi reproduse si garantate in acelasi format.

Componentele din cadrul sistemului vor asigura urmatoarele cerinte:

- Certitudinea că informațiile publicate provin de la o anumită sursă.
- După publicarea informațiilor web în formă agregată acestea să poată fi prezentate, în aceeasi formă unitară, la orice moment de timp ulterior si sa garanteze autenticitatea si integritatea informatiilor respective.
- În cazul unui litigiu informațiile să poată fi prezentate în justiție într-o formă agregată, acceptată din punct de vedere legal.

Sistemul proiectat va rula sub forma unui terţ de încredere independent de proprietarii de drept ai datelor ce vor fi arhivate iar garantarea integritatii continutului arhivat se face prin mecanisme criptografice bazate pe semnatura digitală si marcare temporala pe termen lung. Prin intermediul serviciului, ulterior se poate proba existenta, autenticitatea si integritatea continutului respectiv la un anumit moment de timp iar detinatorii site-ului pe care informatiile au fost publicate vor putea proba, chiar in justitie daca este cazul, faptul ca informatia a fost publicata pe respectivul site la un anumit moment de timp. In acest sens printre alte cerinte de proiectare mai mentionam urmatoarele:

- Serviciul va avea functionalitatea specifica unui web-crawler capabil sa colecteze si sa arhiveze sigura o serie de resurse web puse la dispozitie de furnizorul de continut web.
- Serviciul va face arhivarea continutului web in mod automat si direct de pe site –ul furnizorului web detectand modificarile aparute.
- La arhivarea informatiei web, serviciul TTP va pastra forma de prezentare a continutului pe site –ul furnizorului.

- Serviciul va putea garanta autenticitatea informatiei captate de pe site –urile furnizorilor apeland in acest sens la autentificarea acestora prin certificate digitale.

Sistemul *Site-Request* va fi dezvoltat modular si va cuprinde componente cu functii specifice bine definite ce se vor conecta intre ele prin definirea si implementarea interfetelor de comunicatie. Solutia va fi astfel compusa din urmatoarele componente software:

- **WebCrowler** – responsabila cu colectarea continutul site-ului si a unor metadate specifice.
- **LongTimeSignature** – responsabila cu generarea informatiilor necesare pentru garantarea pe termen lung a continutului (existenta, autenticitatea, integritate).
- **Archive** – responsabila cu arhivarea informatiilor colectate.
- **SignatureUpdater** – responsabila cu actualizarea periodica a informatiei de garantare.
- **ContentPresentation** – responsabila cu operatiile de cautare, identificare si prezentare a continutului arhivat.

Figura 9 contine o diagrama in care sunt evidentiate componentele sistemului de arhivare on-site si modul in care acestea comunica intre ele:

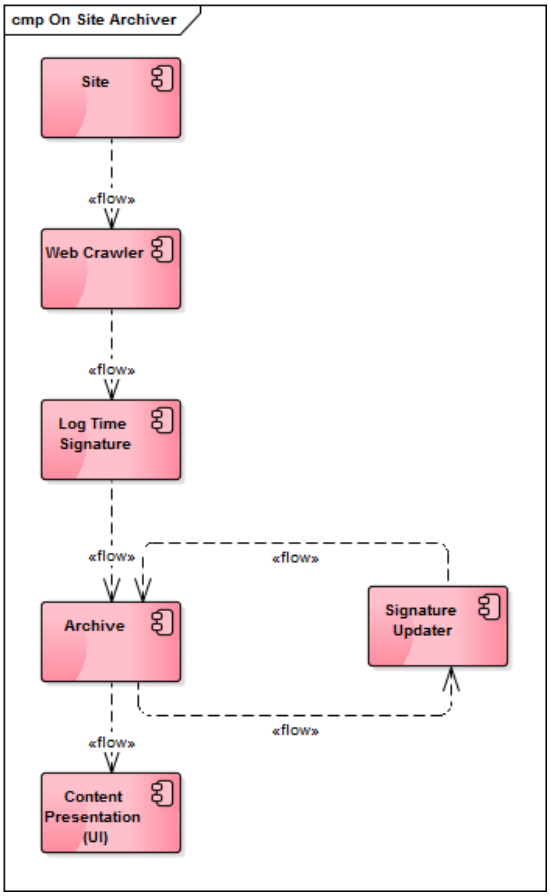

Figura 9 – Structura sistemului de garantare a continutului web

Fiecare componenta comunica pe lantul din diagrama prezentata in Figura 9 cu urmatoarea componenta intr-un mod standardizat. Astfel componenta care ofera datele va expune o interfata iar componenta care consuma datele va implementa un conector la aceasta interfata. Site-ul va expune datele prin protocolul HTTP iar *WebCrawler* -ul le va colecta datele folosind acest protocol. Prin urmare datele obtinute in urma procesului de colectare (colectia de pagini web) vor fi in forma in care aceste se vad in mod normal intr-un browser web. Intre celelalte componente: *WebCrawler* si *LongTimeSignature*, *Archive* si *SignatureUpdater* interfatarea este prin fisiere.

Componenta *WebCrowler* asigura colectare si aducerea datelor de pe site-urile tinta ce urmeaza a fi arhivate si actualizarea acestor date cand acestea se schimba pe site. In etapele anterioare au fost parcurse o serie de experimente si documentari pe marginea solutiilor de tip web crowler existente pe piata. In urma acestei analize, in faza de proiectare s-a decis folosirea ca web crowler in cadrul prototipului a solutiei ***Heritrix***. Acesta permite colectarea resurselor si stocarea acestora in arhive web de tipul ARC sau WARC. Este dezvoltata integral in Java si se instaleaza ca o aplicatie server de sine statatore fara a necesita rularea intr-un server web precum Tomcat sau JBoss. Este compusa din doua parti: cadrul general si module externe. Cadrul general cuprinde motorul central al aplicatiei si modul de interactiune cu acesta: interfata cu utilizatorul, managementul proceselor interne si setarile. Setarile sunt transmise folosind un fisier de configurare de tip xml care contine definirea de bean-uri corespunzatoare modulelor aditionale cat si proprietatile acestora. Ca detalii de proiectare a configuratiilor ce urmeaza a fi folosite in cadrul prototipului:

- In optiunile de configurare se vor avea in vedere politica de selectie, politica de revizitare, politica de politete si politica de paralelizare.
- Va fi configurat sa comunice cu site –urile vizitate folosind protocoalele HTTP/HTTPS.
- Va fi configurat sa permita implementarea urmatoarelor scenarii (politica de selectie):
    - Colectarea unei singure resurse (pagina) si a resurselor direct referite de aceasta (css, javascripturi, img-uri, doc, pdf). Se vor folosi regulile bazate pe utilizarea de expresii regulate (regula *MatchesListRegexDecideRule).*
    - Colectarea in intregime a unui site si a resurselor referite, aflate in acelasi domeniu de nume.
    - Colectarea partiala a unui site (adancime configurabila, folosind regula *TooManyHopsDecideRule*).
- In vederea monitorizarii continue a unui site, in secventa de procesoare disponibile *DispositionChain*, va fi configurat un procesor de tip ReschedulingProcessor pentru a permite revizitarea periodica a unui continut.
- Vor fi folosite proprietatile *sendIfNoneMatch* si *sendIfModifiedSince* corespunzatoare procesorului *FetchHTTP* pentru a obtine o politica de revizitare eficienta bazata pe identificarea modificarilor aparute la nivelul continutului colectat.
- Datele rezultate din procesul de colectare si oferite de WebCrawler vor fi agregate sub forma unor arhive web de tip WARC. Aceste arhive vor putea fi apoi transferate usor catre componentele de garantare si arhivare.
- Va fi configurat sa realizeaze adaugarea la arhiva a unor informatii de tip metadata.
- Intervalul de revizitare va fi configurat la nivelul web crowlerului in functie de fiecare site pentru a obtine un procent bun privind gradul de incarcare in raport cu frecventa modificarilor posibile la nivelul site-ului. In acest sens, se va folosi un procesor dedicat de tip *DispositionProcessor*.
- Se va folosi beanul de tip CrawlController pentru a configura numarul de threaduri active prin politica de paralelizare.

Componenta *LongTimeSignature* este responsabila cu aplicarea de semnaturi digitale si de marca temporala (prin intermediul atributelor *signature-time-stamp* si *archive-time-stamp-v3*) asupra datelor colectate si expuse de *Web Crawler* sub forma de arhive war. In acest sens:

- *LongTimeSignature* va functiona ca serviciu fiind un consumator pentru componenta WebCrowler.
- Pentru interfatarea cu aplicatia WebCrowler se va folosi un sistem comun de fisiere impreuna cu un mecanism de comunicare bazat pe monitoprizarea unor fisiere speciale de notificare generate de aplicatia de colectare.
- Pentru generarea informatiei de semnatura componenta va folosi functionalitati expuse de SDK –ul proiectat in Activitatea III.2 (descris in acest raport in capitolul precedent).
- Formatul de semnatura folosit va fi CAdES-A. Acest format permite actualizarea periodica a mecanismelor de protectie (algoritmi, mecanisme , lungimea cheilor criptografice, etc.) si poate asigura in acest fel proprietati de garantare pe termen lung a continutului arhivat.

- Pentru gestiunea (crearea si protectia) cheilor de semnare componenta va folosi un dispozitiv de tip HSM – *Hardware Secure Module* (se va folosi o instanta obtinuta din SSEAPI printr-un apel *SS_KeyStore::CreateNFHSMInstance*() capabil sa gestioneze o instanta de lucru cu un HSM de tip nCipher).
- Pentru gestiunea informatiilor de garantare se va folosi formatul de semnatura elelctronica detasata (detached). In acest format, datele pot fi pastrate separat fata de informatiile de garantare iar acest lucru permite optiunea de a numai replica datele la urmatoarele sesiuni de actualizare a informatiei de garantare.

Componenta SignatureUpdater are rolul de a urmari periodic datele pastrate in *Arhiva*, de a le extrage si de a actualiza semnaturile digitale atunci cand este cazul dupa care le va furniza componentei *Archive* pentru a le depozita inapoi in *Arhiva*. Detalii de proiectare:

- Componenta va fi implementata ca serviciu.
- Executa actiuni periodice de *refresh* la nivelul elementelor arhivate. La fiecare *refresh* realizeaza identificarea elementelor ce trebuie actualizate si realizeaza actualizarea semnaturilor.
- Componenta constituie un model redus al componentei de semnare, realizand actualizarea semnaturii prin adaugarea unui nou atribut *archive-time-stamp-v3* specific formatului CAdES-A. SDK –ul ce urmeaza a fi dezvoltat este proiectat sa asigure aceasta functionalitate pentru formatul CAdES-A.
- Realizeaza inlocuirea semnaturilor vechi cu cele actualizate.
- Pentru comunicatia si interfatarea cu un sistem de marca temporala, componenta va avea implementat un modul propriu de client HTTP, capabil sa realizeze expedierea cererilor de tip TimestampRequest si receptia raspunsurilor de tip TimestmapResponse definite de RFC 3161.
- Politicile de actualizare a semnaturii vor fi implementate prin fisiere de configurare dedicate ce vor mentiona pentru fiecare item in parte detaliile privind algorimii ce urmeaza a fi folositi.
- Necesitatile de interfatare cu componenta de arhivare pentru a rearhiva datele de semnatura se va face prin sistemul de fisiere.

Componenta *Archive* asigura pastrarea datelor pe o perioada lunga de timp intr-o forma structurata si indexata dupa cuvinte cheie si alte tipuri de metadata. Aceasta componenta preia datele colectate si a celor generate de componentele de semnare (*LongTimeSignature* si *SignatureUpdater*) si le pastreaza intr-o *Arhiva*. Pentru arhiva se va folosi sistemul de arhivare electronic existent deja la CertSIGN, iar la nivelul componentei Archive va fi dezvoltat un conector capabil sa realizeaza interfatarea cu acest sistem. O solutie alternativa ce va fi luata in considerare este de a se dezvolta un sistem separat de arhivare dedicat acestui serviciu.

Content Presentation (UI) este componenta care, la cerere, va extrage datele din arhiva si le va prezenta in forma in care acestea au existat in original pe site. Functionalitatile ce vor fi asigurate de aceasta componenta sunt:

- Permite cautarea dupa metadata si identificarea unui content arhivat
- Va contine un modul de verificare a semnaturile CAdES-A (cu tot cu lantul de atribute adaugate de componenta de actualizare). Acest modul va folosi functionalitatile puse la dispozitie de SDK ul dezvoltat.
- Va detine un modul de validare a certificatelor folosite la semnare.
- Descarca la cerere intr-un folder a continutului asociat si a informatiilor de garantare pentru ca acestea sa poata fi folosite ca probe.
- Componenta lucreaza direct cu sistemul de arhivare iar in acest sens s eva dezvolta un conector de interfatare cu acest sistem.

2) Solutia de tip *Client-Request*

Solutia propune garantarea pe termen lung a optiunilor de vizitare de continut web de catre un client. Clientul viziteaza diferite site –uri, vizualizeaza pagini web si realizeaza cereri explicite la momente pe care el le decide, prin care cere serviciului TTP sa-i arhiveze anumite pagini (impreuna cu toate informatiile necesare pentru garantarea pe termen lung).

Din punct de vedere arhitectural, solutia va fi similara cu cea anterioara cu exceptia primei aplicatii. Componenta de colectare (bazata pe WebCrowler in solutia anterioara) va fi dezvoltata in acest caz sub forma unei aplicatii dedicate, de tip gateway, implementata ca portal web. Aceasta va permite "browsarea" clientului pe site –uri si generarea la cererea acestuia a unor screenshot. Fiecare screenshot presupune randarea paginii si apoi generarea imaginii corespunzatoare. Fluxul asigurat de componenta de colectare este urmatorul:

a) Clientul se autentifica cu un certificat digital la serviciul gateway.
b) Fiecare client va avea creat un userspace creat la nivelul TTP -ului
c) Clientul are posibilitatea de a introduce un URL.
d) Serviciu prezinta pagina (resursa) clientului.
e) La cererea clientului, serviciul capteaza pagina: randare pagina, generare png, salvare pagina intr-un folder creat in userspace-ul clientului .
f) Solutia de randare va tine cont de tipul de browser folosit de client.

6 Activitatea III.4 - Diseminarea rezultatelor

In aceasta activitate a fost realizate mai multe actiuni de diseminare asupra rezultatelor proiectului:

- Actualizare site -ului proiectului (www.proiect-tape.ro) cu specificatii privind arhitectura sistemului si a componentelor sale.
- In mai 2014, in contextul unei intalniri cu reprezentanti ai organizatiilor guvernamentale si ai mediului de afaceri, a fost efectuata o prezentare privind arhitectura prototipului TTP de garantare a continutului web si a serviciilor ce pot fi dezvoltate pe baza acestor prototipuri. In urma acestei intalniri am constatat interesul acestor organizatii in special pentru utilizarea serviciilor oferite tertilor pentru garantarea continutului publicat de acestia.
- In iulie 2014 a avut loc o intalnire cu reprezentanti ai mediului de afaceri din SUA la Bruxelles carora le-au fost prezentate variantele de prototipuri (*Site-Request* si *Client-Request*) privind garantarea continutului web si mecanismele avute in vedere pentru asigurarea nerepudierii si integritatii. In urma feedback-ului obtinut in cadrul acestor discutii s-a ajuns la varianta curenta care acomodeaza si cerinte generate din zona de business.
- Realizarea unei prezentari in cadrul evenimentului "*Security Exhibition*" care a avut loc in Republica Moldova in perioada 17-20 septembrie 2014. In prezentare au fost atinse urmatoarele aspecte: necesitatea unui serviciu TTP pentru garantarea de continut web, o solutie de arhitectura pentru serviciul TTP propus in acest sens in cadrul proiectului TAPE si mecanismele de securitate bazate pe formatele de semnatura avansata rezistente pe termen lung ce vor fi utilizate in cadrul solutiei.
- In cadrul conferintei internationale "*CyberSecurity in Romania*", desfasurata la Sibiu in perioada 2-4 octombrie 2014 a fost realizata o prezentare (in panelul „*Mobile and Cloud Security*") privind prototipurile solutiei ce adreseaza garantarea continutului web si provocarile existente. Accentul s-a pus in principal pe scenariile de utilizare, arhitectura sistemului si mecanismele de garantare pe termen lung a integritatii si nepudierii.
- In cadrul evenimentului "*Cyber Security Day*" (https://www.cybersecurityday.ro/) organizat in 29 octombrie 2014, in panelul "*Modern approaches for cybersecurity challenges – technologies and services*", in contextul provocarilor actuale de securitate cibernetica, au fost promovate discutii privind necesitatea si utilitatea unui serviciu de securitate orientat pe garantarea autenticitatii continuturilor web. Au fost discutate aspecte tehnice privind securitatea datelor arhivate pe termen lung, scenarii practice de utilizare precum si elemente de reglementare specific unui astel de serviciu.

Virgiliu Mihail TOGAN

DIRECTOR PROIECT