DITEC
Data Information Technology & Expert Consulting

Špecifikácia SCA
D.Signer/XAdES .NET

Copyright

Popisné charakteristiky dokumentu

Projekt	GOV_ZEP		A3019_002
Dokument	Špecifikácia SCA		
Podnázov	D.Signer/XAdES .NET		
Ref. číslo	GOV_ZEP.4	Verzia	10

Vypracoval	Róbert Vittek	Podpis	Dátum 14.4.2014
Preveril	Major Marián	Podpis	Dátum
Schválil		Podpis	Dátum

Formulár	Dokument		
Ref. číslo	Fo 11	Dátum poslednej aktualizácie	Dátum 18.5.2005

Projekt	GOV_ZEP		A3019_002
Dokument	Špecifikácia SCA		
Referencia	GOV_ZEP.4	Verzia	10

Záznamy o zmenách

Autor	Popis zmien	Dátum	Verzia

Pripomienkovanie a kontrola

Autor	Stanovisko	Dátum	Verzia

Rozdeľovník

	Priezvisko Meno	Firma, Funkcia
Originál		
Kópia		
Kópia		
Kópia		

Obsah

1. Úvod

Cieľom tohto dokumentu je špecifikácia SCA aplikácie D.Signer/XAdES .NET pre vytváranie ZEP podľa profilu XAdES_ZEP [24][25][26] (ďalej len XAdES_ZEP) a v súlade s definovanou podpisovou politikou, teda:

- definovanie katalógu požiadaviek aplikácie D.Signer/XAdES .NET,
- špecifikácia architektúry aplikácie D.Signer/XAdES .NET,
- funkčná špecifikácia aplikácie D.Signer/XAdES .NET,
- špecifikácia API aplikácie D.Signer/XAdES .NET.

Prílohou tejto špecifikácie aplikácie D.Signer/XAdES .NET sú dokumenty špecifikácií jednotlivých prípojných modulov (pluginov) pre jpodporované typy vstupných dokumentov (XML, PDF, RTF atď.)

2. Zoznam použitých skratiek

DHC – Data Hashing Component

DTBS – Data To Be Signed

DTBSF – Data To Be Signed Formatted, resp. komponent Data To Be Signed Formatter

DTBSR – Data To Be Signed Representation

NBÚ – Národný bezpečnostný úrad

SAC – Signer's Authentication Component

SCA – Signature Creation Application

SCDev – Signature Creating Device

SCVA – Signature Creation and Validation Application

SDOC – Signed Data Object Composer

SDP – Signer's Document Presentation

SIC – Signer Interaction Component

SLC – Signature Logging Component

SSA – SCDev/SCA Authenticator

SSC – SCDev/SCA Communicator

SSCD – Secure Signature Creating Device

SVA – Signature Validation Application

XML – eXtended Markup Language

XSD – XML Schema Definition

XSL – eXtensible Stylesheet Language

XSLT – XSL Transformation

XAdES – XML Advanced Electronic Signatures

XAdES_ZEP – profil špecifikácie formátu elektronického podpisu XAdES pre ZEP

ZEP – Zaručený elektronický podpis

3. Referencie

[1] W3C/IETF Recommendation: "XML-Signature Syntax and Processing" v2002-02-12 (XMLDSIG)

[2] ETSI TS 101 733 – CMS Advanced Electronic Signatures (CAdES) v1.6.3

[3] ETSI TS 101 903 – XML Advanced Electronic Signatures (XAdES) v1.4.2

[4] RFC 3125 – Electronic Signature Policies

[5] RFC 3161 – Internet X.509 Public Key Infrastructure Time-Stamp Protocol

[6] RFC 3279 – Algorithms and Identifiers for the Internet X.509 PKI

[7] RFC 5280 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

[8] RFC 3548 – The Base16, Base32, and Base64 Data Encodings

[9] RFC 3852 – Cryptographic Message Syntax (CMS)

[10] RFC 4051 – Additional XML Security Uniform Resource Identifiers

[11] Zákon č. 215/2002 Z.z. o elektronickom podpise a o zmene a doplnení niektorých zákonov v znení neskorších predpisov

[12] Vyhláška NBÚ č. 131/2009 Z.z. o certifikátoch a kvalifikovaných certifikátoch

[13] Vyhláška NBÚ č. 134/2009 Z.z. o produktoch elektronického podpisu

[14] Vyhláška NBÚ č. 135/2009 Z.z. o vyhotovení a overovaní elektronického podpisu a časovej pečiatky

[15] Vyhláška NBÚ č. 136/2009 Z.z. o spôsobe a postupe používania elektronického podpisu v obchodnom styku a administratívnom styku

[16] NBÚ Formáty certifikátov a kvalifikovaných certifikátov, v3.0 (2009-06-30)

[17] NBÚ Formáty zoznamu zrušených kvalifikovaných certifikátov, v1.2 (2005-11-06)

[18] NBÚ Formáty zaručených elektronických podpisov, v3.0 (2009-08-12)

[19] NBÚ Upresnenia obsahu a formálne špecifikácie formátov dokumentov pre ZEP, v1.0 (2007-07-24)

[20] CWA 14170:2004 E – Security requirements for signature creation applications

[21] CWA 14171:2004 D/E/F – General guidelines for electronic signature verification

[22] XMLENC – XML Encryption Syntax and Processing", J. Reagle, D. Eastlake, December 2002. http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

[23] Koncepcia všeobecného formátu XML podpisu a aplikácie SCVA, DITEC, a.s., 2006

[24] Profil XAdES_ZEP – formát ZEP na báze XAdES, v1.0, DITEC, a.s., 2008

[25] Profil XAdES_ZEP – formát ZEP na báze XAdES, v1.1, DITEC, a.s., 2009

[26] Profil XAdES_ZEP – formát ZEP na báze XAdES, v2.0, DITEC, a.s., 2011

[27] Rozhodnutie komisie 2011/130/EU, ktorým sa ustanovujú minimálne požiadavky na cezhraničné spracovanie dokumentov elektronicky podpísaných príslušnými orgánmi v zmysle smernice Európskeho parlamentu a Rady 2006/123/ES o službách na vnútornom trhu

[28] ETSI TS 103 171 – Electronic Signatures and Infrastructures (ESI) XAdES Baseline Profile, v2.1.1

4. Katalóg požiadaviek

4.1. Požiadavky vyplývajúce z legislatívy

Legislatívne požiadavky pre vytváranie zaručeného elektronického podpisu špecifikuje

- zákon o elektronickom podpise č. 215/2002 Z.z. v znení neskorších predpisov,
- vyhláška NBÚ SR č. 135/2009 Z.z.

V zmysle zákona č. 215/2002 Z.z. o elektronickom podpise musia bezpečné zariadenia na vyhotovovanie elektronického podpisu a postupy používané na vyhotovovanie zaručeného elektronického podpisu:

- spoľahlivo zabezpečiť, že podpisovaný elektronický dokument pri vyhotovovaní zaručeného elektronického podpisu sa nemení,
- umožniť, aby sa elektronický dokument, ktorý sa bude elektronicky podpisovať, zobrazil podpisovateľovi ešte predtým, ako sa spustí procedúra na vyhotovenie zaručeného elektronického podpisu,
- zaručiť, že pravdepodobnosť toho, že sa nejaký súkromný kľúč vyhotoví viac než raz, bude zanedbateľná.

Na overovanie zaručených elektronických podpisov sa musia používať také produkty a postupy, ktoré zabezpečia, že:

- podpísaný elektronický dokument sa pri overovaní zaručeného elektronického podpisu nezmení,
- zaručený elektronický podpis sa spoľahlivo overí a výsledok overovania sa správne zobrazí,
- možno určiť, či podpísaný elektronický dokument je zhodný s elektronickým dokumentom, ku ktorému bol zaručený elektronický podpis vyhotovený,
- overovateľ môže určiť osobu, ktorej zaručený elektronický podpis patrí a použitie pseudonymu je jasne vyznačené.

Podrobnosti procesu vytvorenia a overenia ZEP upravuje vyhláška NBÚ SR č. 135/2009 Z.z. Ďalšie požiadavky na zariadenia pre vyhotovovanie elektronického podpisu (SSCD) stanovuje vyhláška NBÚ SR č. 134/2009 Z.z.

Popis procesu vytvárania a overovania certifikačnej cesty pri overovaní ZEP podľa platnej legislatívy SR, štandardov X.509, ETSI a RFC požiadaviek kladených na kvalifikované certifikáty a ZEP je obsiahnutý v dokumente NBÚ SR – Kontrola certifikačnej cesty, v1.4.

Metodický postup a požiadavky pre vykonanie auditu aplikácií pre ZEP stanovuje dokument NBÚ SR – Metodika auditu SCA a SVS pre ZEP, OID: 1.3.158.36061701.0.0.1.3.2.6.

Povolené formáty podpisovaných elektronických dokumentov v administratívnom styku definuje vyhláška NBÚ č. 136/2009 Z.z. Požiadavky na formát a obsah podpisovaných dát stanovuje dokument NBÚ SR – Upresnenia obsahu a formálne špecifikácie formátov dokumentov pre ZEP, v1.0.

Požiadavky NBÚ SR na vytváraný formát ZEP upravuje dokument – Formáty zaručených elektronických podpisov, v3.0. Minimálne požiadavky EÚ na cezhraničné spracovanie dokumentov elektronicky podpísaných príslušnými orgánmi v zmysle smernice Európskeho parlamentu a Rady 2006/123/ES o službách na vnútornom trhu stanovuje rozhodnutie komisie 2011/130/EU.

4.1.1. Podpisová politika

Podpisová politika predstavuje množinu pravidiel pre vytváranie a overovanie elektronického podpisu a vzhľadom ku ktorej môže byť elektronický podpis vyhodnotený ako platný, resp. neplatný. Požiadavky na obsah podpisovej politiky sú dané právnym alebo obchodným kontextom, v rámci ktorého je potrebné implementovať elektronický podpis.

Podpisová politika musí byť pre účely vyhodnotenia naplnenia požiadaviek, (daných uvedeným právnym, resp. obchodným kontextom) k dispozícii v čitateľnej forme.

Pre účely automatického spracovania elektronických podpisov, tie časti podpisovej politiky, ktoré špecifikujú elektronické pravidlá pre vytváranie a overovanie elektronického podpisu, musia byť k dispozícii v počítačovo spracovateľnej forme.

Podpisová politika môže byť jednoznačne určená implicitne národnou legislatívou alebo zmluvou (ktorá uvádza, aká podpisová politika musí byť v danom kontexte použitá) alebo môže byť definovaná explicitne v rámci elektronického podpisu. V takom prípade musí mať podpisová politika jedinečný identifikátor, ktorý musí byť zviazaný s vytvoreným elektronickým podpisom. V takom prípade musí tiež pre danú explicitnú podpisovú politiku existovať práve jedna definitívna forma s jedinečnou binárne kódovanou reprezentáciou.

Pre vytvorenie ZEP musí byť aplikácia použitá len v súlade s platnou podpisovou politikou pre ZEP, ktorá bola schválená NBÚ SR. Používateľ je pred vytvorením podpisu povinný presvedčiť sa, že podpisová politika, ktorú aplikácia používa, je stále platná a nebola zo strany vydavateľa predčasne zrušená. Výrobca, resp. integrátor aplikácie D.Signer/XAdES .NET je povinný zabezpečiť také nastavenie konfigurácie aplikácie a parametrov volania metód rozhrania aplikácie, aby aplikácia vytvárala podpis v súlade so špecifikovanou podpisovou politikou.

4.2. Požiadavky vyplývajúce z koncepcie riešenia aplikácií pre ZEP

4.2.1. Architektúra SCVA aplikácií a formát ZEP

Cieľom koncepcie riešenia aplikácií pre ZEP je návrh:

- profilu XAdES_ZEP (formátu elektronického podpisu na báze XAdES, [24][25][26])
- a architektúry SCVA aplikácií pre vytváranie a overovanie ZEP, [23]

nad množinou rôznych formátov dokumentov, resp. typov dát (XML dokumenty, HTML stránky, PDF súbory, atď.), ktoré sú definované v rámci dokumentov [15][19] tak, aby bolo možné v rámci konkrétnych business aplikácií:

- vytvárať elektronický podpis nad komplexnými dátovými štruktúrami, v rámci ktorých je možné kombinovať ľubovoľné typy dát z podporovanej množiny typov.
- priebežne rozširovať SCVA o podporu ďalších typov dátových objektov,
- v rámci danej business aplikácie nasadiť len relevantné komponenty SCVA aplikácie pre zabezpečenie podpory požadovaných typov dátových objektov,
- riadiť činnosť jednotlivých komponentov SCVA pomocou konfiguračných dát (pričom musí byť splnená nutná podmienka zabezpečenia ochrany integrity konfiguračných dát).

Dodržaním princípov komponentovej architektúry bude môcť aplikácia pre vytvorenie ZEP (SCA) pomocou samostatných komponentov (pluginov pre jednotlivé typy dát) pripraviť na podpis a zobraziť používateľovi pred vytvorením ZEP všetky podpisované dátové objekty. Podobne aplikácia pre overenie ZEP (SVA) bude pomocou rovnakého mechanizmu pluginov pre každý typ aplikačných dát schopná overiť platnosť všetkých referencií v rámci overovaného ZEP, ako aj platnosť samotného ZEP.

Aplikácia D.Signer/XAdES .NET bude vytvárať ZEP v súlade so schválenými formátmi pre zaručený elektronický podpis:

- XAdES_ZEP, v1.0 (http://www.ditec.sk/ep/signature_formats/xades_zep/v1.0)
- XAdES_ZEP, v1.1 (http://www.ditec.sk/ep/signature_formats/xades_zep/v1.1)
- XAdES_ZEP, v2.0 (http://www.ditec.sk/ep/signature_formats/xades_zep/v2.0).

Príslušné XML schémy pre

- XML Signature,
- XAdES,
- XAdES_ZEP, v1.0
- XAdES_ZEP, v1.1

- XAdES_ZEP, v2.0
- a ďalšie XML štruktúry (napr. verifikačné údaje pre XML dokumenty)

budú tvoriť súčasť aplikácie.

4.2.2. Bezpečnostné požiadavky

Bezpečnostné požiadavky na systémy a aplikácie pre vytváranie (zaručeného) elektronického podpisu definuje dokument CWA 14170:2004 E – Security requirements for signature creation applications [20].

Bezpečnostné požiadavky na systémy a aplikácie pre overovanie (zaručeného) elektronického podpisu definuje dokument CWA 14171:2004 D/E/F – General guidelines for electronic signature verification [21].

4.3. Požiadavky vyplývajúce z potrieb cieľových systémov

Aplikácia D.Signer/XAdES .NET pre vytváranie ZEP bude nasadená ako súčasť informačných systémov pre elektronickú výmenu dokumentov medzi rôznymi subjektami, v rámci ktorej je potrebné zabezpečiť:

- jednoznačnú identifikáciu pôvodcu dokumentu a neodmietnuteľnosť autorstva,
- integritu prenášaných dokumentov.

Aplikácia D.Signer/XAdES .NET bude poskytovať:

- rozhranie pre integráciu v rámci klientskej webovej alebo .Net aplikácie,
- GUI rozhranie pre jednoznačnú prezentáciu podpisovaných dát a ďalších atribútov elektronického podpisu používateľovi predtým, ako sa spustí procedúra na vyhotovenie zaručeného elektronického podpisu,
- možnosť smart inštalácie bezpečným spôsobom.

Systémové požiadavky:

- operačný systém MS Windows 2003 Server, 2008 Server, 2012 Server, Vista, Windows 7, Windows 8 a .Net framework, verzia 2.0-3.5,
- certifikované SSCD zariadenie pre generovanie kľúčových párov a vytváranie elektronického podpisu,
- web prehliadač – MS Internet Explorer, v6.0 a vyššia alebo prehliadač s podporou NP API: Firefox v3.x, Google Chrome v12.x, Opera v10.x, Safari 5.1 alebo viac.

Ďalšie požiadavky na prevádzkové prostredie aplikácie D.Signer/XAdES .NET a na systémy, v rámci ktorých bude aplikácia nasadená, budú špecifikované v samostatnom dokumente. Kontrola naplnenia týchto požiadaviek sa bude riadiť príslušnými nariadeniami, prípadne potrebami daného projektu.

4.4. Požiadavky na otvorenosť systému

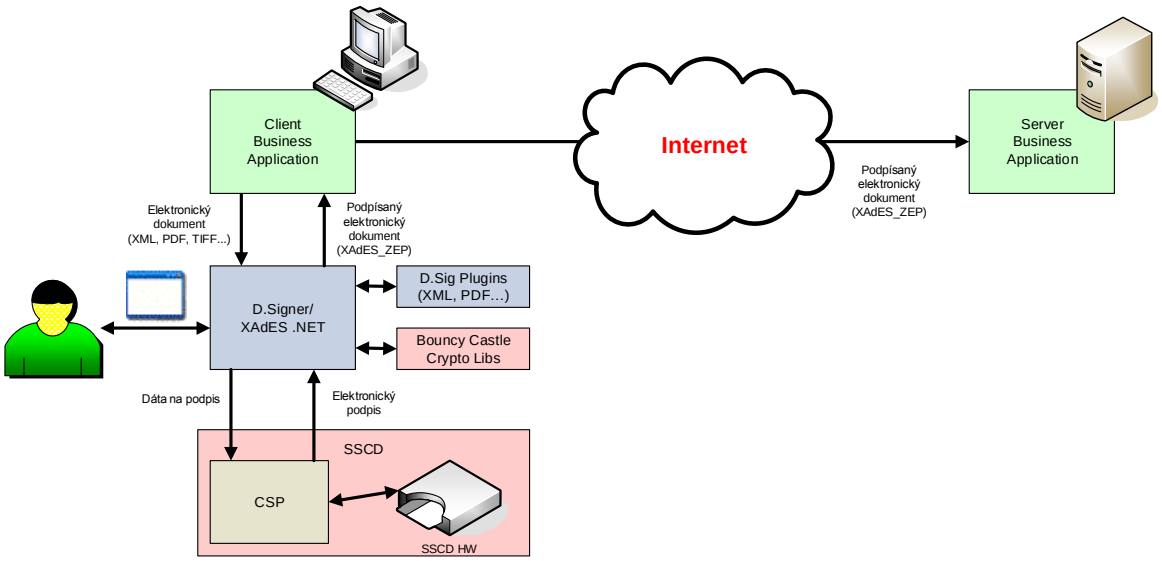
Pri návrhu a implementácii aplikácie D.Signer/XAdES .NET pre vytváranie ZEP sa autori aplikácie riadili dokumentami, normami a odporúčaniami, ktoré sú uvedené v rámci referencií, pozri kapitolu 3.

5. Architektúra aplikácie D.Signer/XAdES .NET

V rámci tejto kapitoly je popísaný návrh komponentovej architektúry aplikácie D.Signer/XAdES .NET, ktorý vychádza z dokumentov:

- Koncepcia všeobecného formátu XML podpisu a aplikácie SCVA [23]
- CWA14170:2004 E – Security requirements for signature creation applications [20].

5.1. Postavenie aplikácie v rámci prevádzkového prostredia



Aplikácia D.Signer/XAdES .NET bude realizovaná ako hlavná aplikácia (modul) a sada komponentov (pluginov) pre jednotlivé podporované dátové typy, ktoré môžu byť v súlade s požiadavkami zákazníka nasadené ako súčasť rozsiahlejších aplikácií a informačných systémov napr. pre elektronickú výmenu dokumentov medzi rôznymi subjektami, v rámci ktorých je potrebné zabezpečiť:

- jednoznačnú identifikáciu pôvodcu dokumentu a neodmietnuteľnosť autorstva,
- integritu (prenášaných) dokumentov.

Aplikácia D.Signer/XAdES .NET bude poskytovať pre klientské aplikácie nasledujúce integračné rozhrania – API:

- .Net API – umožňuje volanie služieb komponentu D.Signer/XAdES .NET priamo z .Net prostredia,

- COM API – wrapper nad .Net API, ktorý umožňuje volanie služieb komponentu D.Signer/XAdES .NET z iných prostredí (kontainerov),
- ATL COM API – wrapper nad COM API (primárne pre Internet Explorer),
- NP API – wrapper pre volanie služieb komponentu D.Signer/XAdES .NET z prehliadačov s podporou NPAPI rozhrania.

Pre interakciu s podpisovateľom bude aplikácia D.Signer/XAdES .NET poskytovať GUI rozhranie, v rámci ktorého bude realizované:

- zobrazenie obsahu podpisovaných dokumentov ako aj všetkých relevantných parametrov ZEP pred spustením procedúry vytvorenia ZEP,
- výber kvalifikovaného certifikátu pre vytvorenie ZEP,
- štandardné ovládacie prvky – potvrdenie procedúry vytvorenia ZEP, zrušenie procedúry vytvárania ZEP apod.

Pre kryptografické operácie spojené s výpočtami digitálnych odtlačkov a samotného elektronického podpisu bude aplikácia využívať:

- kód knižníc Bouncy Castle Crypto,
- certifikované SSCD zariadenie pre generovanie kľúčových párov a vytváranie elektronického podpisu, ku ktorému bude pristupovať pomocou CSP implementácie MS CryptoAPI.

5.2. Vnútorná architektúra aplikácie

5.2.1. Funkčný pohľad

Vnútorná architektúra aplikácie D.Signer/XAdES .NET vychádza a je v súlade s funkčným komponentovým modelom dokumentu CWA14170:2004 E – Security requirements for signature creation applications [20]. Jednotlivé súčasti aplikácie D.Signer/XAdES .NET je teda možné rozdeliť do dvoch skupín:

- dôveryhodné komponenty – povinné komponenty zabezpečujúce základnú požadovanú funkcionalitu SCA,
- aplikačne závislé komponenty – komponenty, ktorých existencia, architektúra a funkcionalita je aplikačne závislá.

Z pohľadu funkčného komponentového modelu SCA sú v rámci aplikácie D.Signer/XAdES .NET implementované nasledujúce dôveryhodné komponenty:

- SDP – Signer's Document Presentation Component – zabezpečuje zobrazenie podpisovaných dokumentov podpisovateľovi,
- SAV – Signature Attributes Viewer – zabezpečuje zobrazenie atribútov vytváraného ZEP podpisovateľovi,
- DTBSF – Data To Be Signed Formatter – zabezpečuje sformátovanie a tranformáciu vstupných dokumentov a ďalších parametrov podpisu do kanonickej formy a vytvorenie štruktúry DTBSF,

- SIC – Signer Interaction Component – rozhranie pre interakciu medzi podpisovateľom a aplikáciou D.Signer/XAdES .NET,
- DHC – Data Hashing Component – umožňuje vytvorenie DTBSR z DTBSF pomocou príslušnej hashovacej funkcie,

Aplikácia D.Signer/XAdES .NET obsahuje nasledujúce aplikačne závislé komponenty všeobecnej architektúry SCA:

- SDOC – Signed Data Object Composer – modul pre vytvorenie dokumentu elektronického podpisu vo formáte XAdES_ZEP zo vstupných dokumentov, ďalších vstupných parametrov, DTBSF a vypočítanej hodnoty elektronického podpisu,

Medzi ďalšie implementované súčasti komponentu D.Signer/XAdES .NET patria:

- Config Reader – modul pre načítanie konfiguračných údajov aplikácie D.Signer/XAdES .NET z MS Windows Registry,

Nasledujúce komponenty netvoria súčasť aplikácie D.Signer/XAdES .NET:

- SAC – Signer's Authentication Component – umožňuje autentifikáciu podpisovateľa pre použitie SSCD zariadenia, je realizovaný v rámci CSP príslušného certifikovaného SSCD zariadenia,
- SSA – SCDev/SCA Authenticator – voliteľný modul pre vytvorenie dôveryhodnej cesty medzi aplikáciou D.Signer/XAdES .NET a SSCD, je realizovaný v rámci CSP príslušného certifikovaného SSCD zariadenia,
- SSC – SCDev/SCA Communicator – rozhranie pre komunikáciu medzi aplikáciou D.Signer/XAdES .NET a SSCD, je realizovaný v rámci CSP príslušného certifikovaného SSCD zariadenia,
- SDC – Signer's Document Composer – umožňuje podpisovateľovi vytvoriť podpisované dokumenty, bude realizovaný v rámci klientskej aplikácie,
- SLC – Signature Logging Component. – zabezpečuje vytváranie auditných záznamov o činnosti aplikácie D.Signer/XAdES .NET, voliteľný komponent – nie je realizovaný,
- SHI – SCDev Holder Indicator – umožňuje zobraziť meno vlastníka SCDev (SSCD) zariadenia, voliteľný komponent – nie je realizovaný.

Aplikácia D.Signer/XAdES .NET umožňuje vytváranie ZEP nad komplexnými dátovými štruktúrami, ktoré môžu zahŕňať rôzne typy dátových objektov (XML, PDF, atď.), pričom aplikácia musí byť schopná rozširovania podpory pre nové typy dátových objektov a jej architektúra musí byť prísne komponentová tak, aby v rámci cieľového prostredia mohli byť nasadené len komponenty (pluginy) s podporou pre relevantné typy dátových objektov.

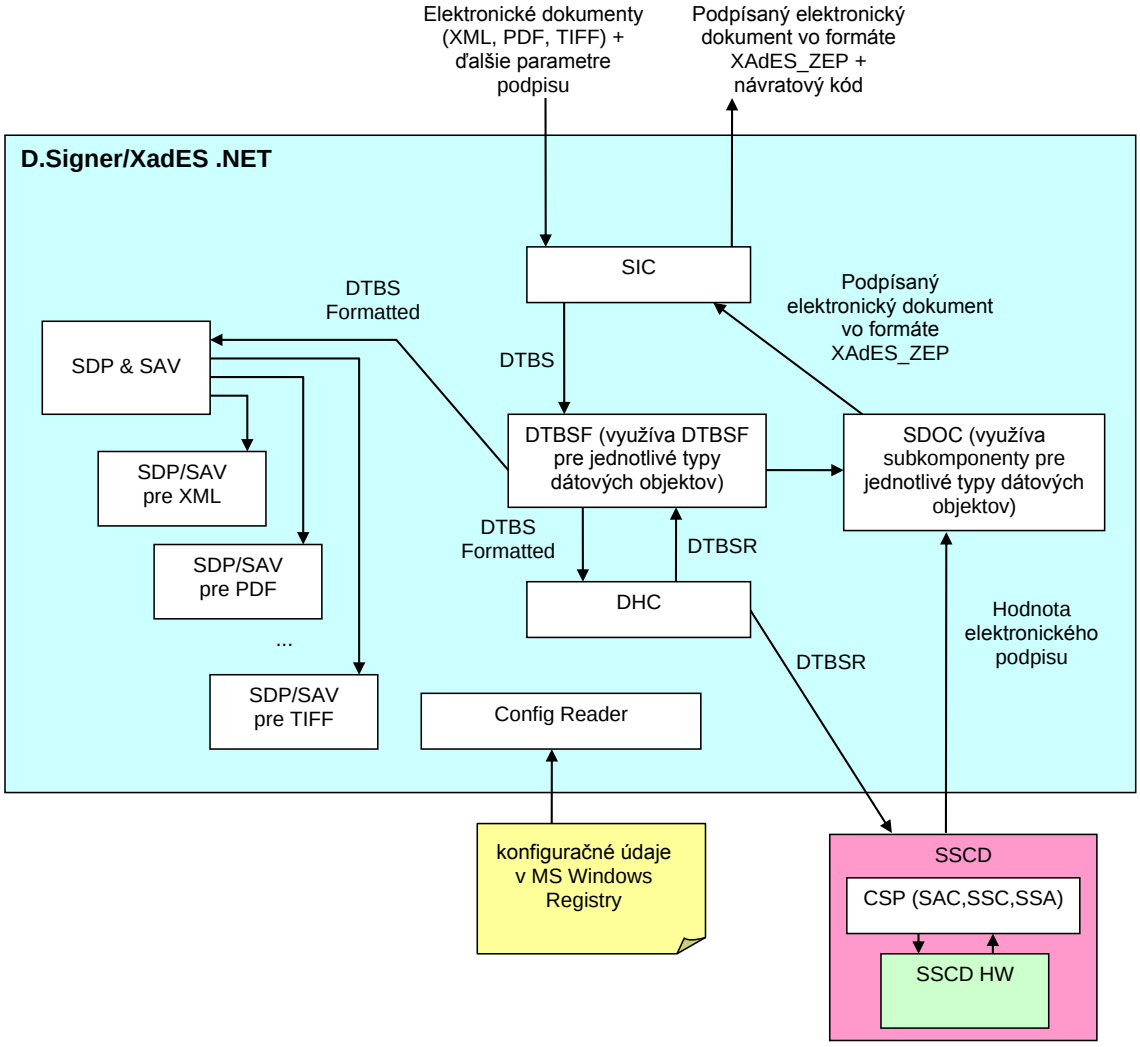
Z pohľadu rozdelenia funkcionality SCA do samostatných modulov, ktoré je možné pri nasadení aplikácie D.Signer/XAdES .NET kombinovať podľa požiadaviek zákazníka, bude aplikácia tvorená nasledujúcimi komponentami:

- D.Signer/XAdES .NET Main – hlavný modul:

- ⇒ poskytuje integračné API pre klientské aplikácie,
- ⇒ spracovanie tých parametrov vytvorenia ZEP, ktoré nie sú závislé na typoch podpisovaných dátových objektov,
- ⇒ poskytuje hlavné prezentačné GUI pre podpisovateľa,
- ⇒ má na starosti vytvorenie ZEP a formátu podpisu podľa profilu XAdES_ZEP,
- ⇒ pre svoju činnosť využíva rozhranie pluginov pre jednotlivé typy dátových objektov (vizualizácia, vytvorenie príslušných DTBSF apod.),

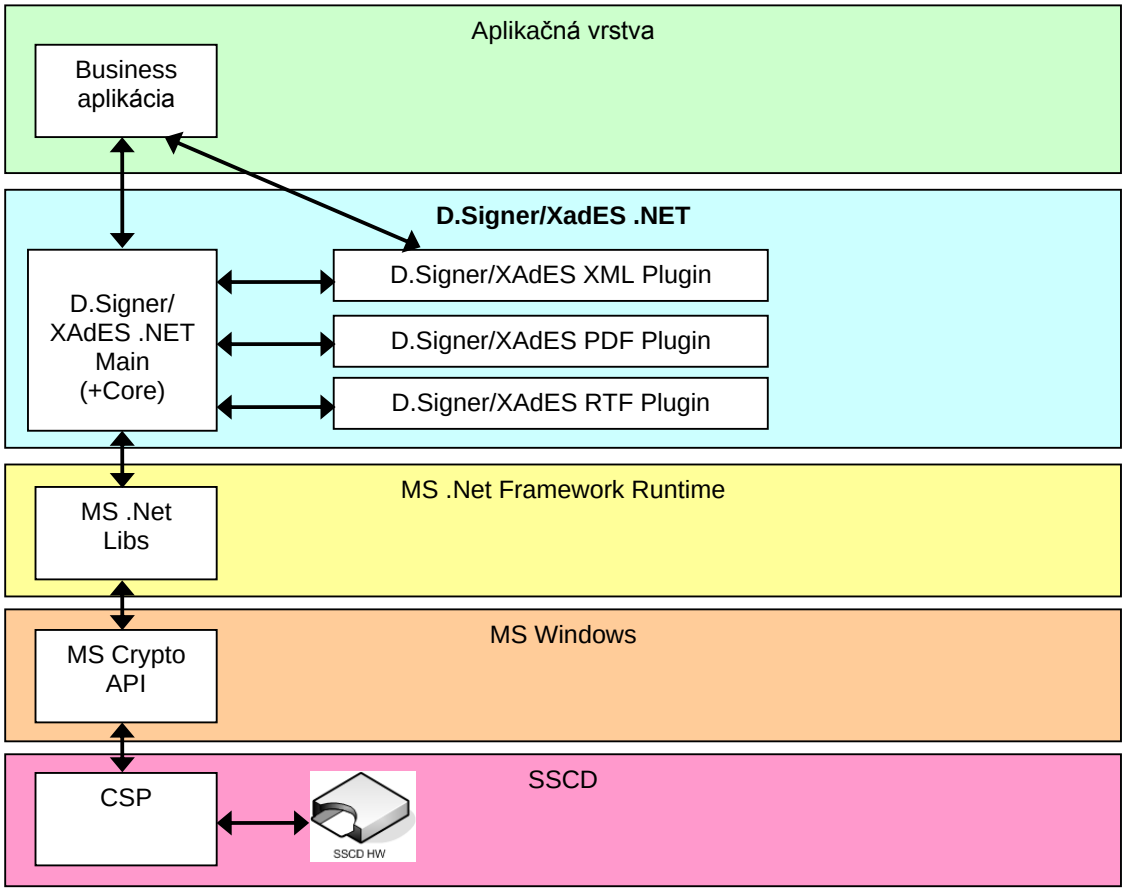
- pluginy pre jednotlivé typy dátových objektov – poskytujú funkcie:
  - ⇒ pre spracovanie tých parametrov vytvorenia ZEP, ktoré sú závislé od typu podpisovaného dátového objektu,
  - ⇒ pre vytvorenie dátových objektov pre podpisované dáta a príslušné verifikačné parametre,
  - ⇒ pre vytvorenie príslušných XML štruktúr pre jednotlivé spracovávané dátové objekty v rámci vytváraného ZEP podľa profilu XAdES_ZEP,
  - ⇒ pre vizualizáciu daného typu dátového objektu,
- D.Signer/XAdES .NET Core – poskytuje funkcie, ktoré sú spoločné pre hlavnú aplikáciu a jednotlivé pluginy (kanonikalizácia XML, výpočet digitálnych odtlačkov apod.)

Na nasledujúcom obrázku je zobrazená bloková schéma dekompozície aplikácie D.Signer/XAdES .NET na jednotlivé popísané súčasti a tok informácií medzi jednotlivými komponentami aplikácie.



5.2.2. Pohľad na vrstvy architektúry

Na nasledujúcom obrázku je zobrazený pohľad na jednotlivé vrstvy architekúry aplikácie, ktorá využíva služby vytvárania ZEP aplikácie D.Signer/XAdES .NET, a postavenie komponentu D.Signer/XAdES .NET v rámci tejto architektúry.



D.Signer/XAdES .NET poskytuje integračné API rozhranie pre aplikačnú vrstvu, teda pre aplikácie, ktoré potrebujú vytvárať ZEP. Pre svoju činnosť využíva knižnice prostredia MS .Net Framework Runtime a prostredníctvom nich pristupuje k MS Windows API a implementácii CSP príslušného SSCD zariadenia.

6. Špecifikácia funkčnosti

6.1. Popis činnosti

Aplikácia (modul) D.Signer/XAdES .NET bude nasadená ako súčasť klientských systémov a aplikácií, v rámci ktorých je potrebné implementovať vytváranie ZEP. Ak chce klientská aplikácia využívať služby modulu D.Signer/XAdES .NET, musí vytvoriť jeho inštanciu. V rámci vytvorenia inštancie modulu prebehne zároveň jeho inicializácia (pozri ďalej).

Následne môže klientská aplikácia pomocou metód integračného API predať modulu D.Signer/XAdES .NET vstupné dokumenty a ďalšie parametre, potrebné pre vytvorenie ZEP. Výsledok procesu vytvorenia ZEP je klientskej aplikácii prístupný cez návratové premenné modulu D.Signer/XAdES .NET: ErrorMessage a SignedXmlWithEnvelope.

Činnosť aplikácie (modulu) D.Signer/XAdES .NET pre vytváranie ZEP je možné popísať nasledovne:

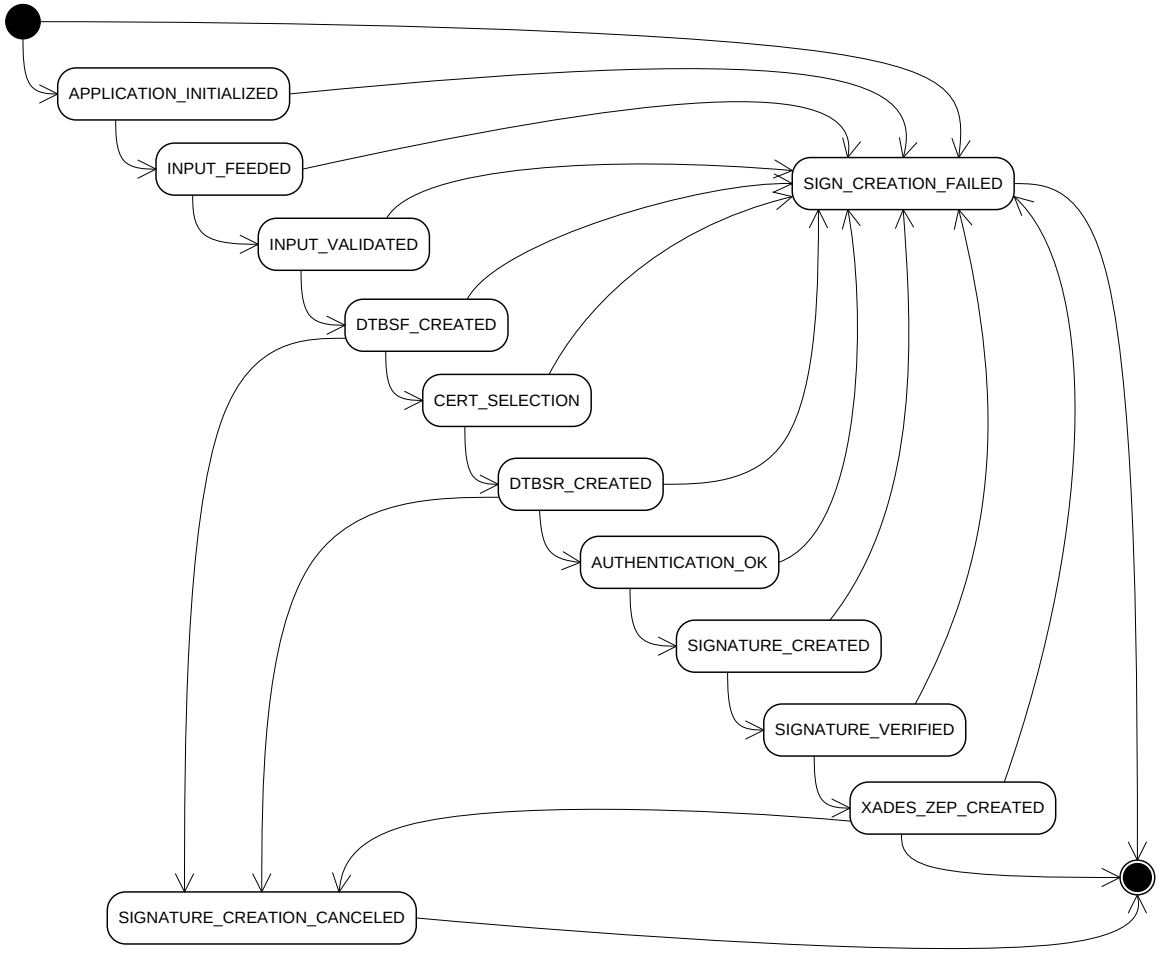
- po vytvorení inštancie modulu D.Signer/XAdES .NET klientskou aplikáciou modul načíta na základe svojich konfiguračných dát z MS Windows Registry zoznam nainštalovaných pluginov D.Signer/XAdES .NET pre typy dátových objektov,
- klientská aplikácia ďalej vytvorí inštancie jednotlivých pluginov pre požadované dátové typy a pomocou volaní metód pluginov CreateObject vytvorí príslušné dátové objekty pre jednotlivé vstupné dokumenty, ktoré majú byť podpísané,
- následne klientská aplikácia zavolá pre jednotlivé vytvorené dátové objekty metódu hlavného modulu addObject, ktorá pridá jednotlivé vstupné dátové objekty do kolekcie dátových objektov na podpísanie (DTBS),
- keď sú pripravené všetky dátové objekty na podpis, klienská aplikácia zavolá metódu Sign (resp. Sign11, Sign20, ďalej len Sign) hlavného modulu, ktorá vykoná validáciu vstupných dokumentov a ich spracovanie v rámci jednotlivých pluginov na DTBSF (aplikovanie príslušných transformácií, napr. kanonikalizácia)
- zobrazí sa hlavné okno aplikácie D.Signer/XAdES .NET, pričom vizualizácia jednotlivých podpisovaných dátových objektov je realizovaná prostredníctvom príslušných funkcií pluginov pre jednotlivé typy dátových objektov,
- používateľ má možnosť si cez GUI aplikácie D.Signer/XAdES .NET prezrieť podpisované dátové objekty a ďalšie parametre podpisu,
- v ďalšom kroku používateľ vyberie pomocou GUI podpisový certifikát,
- po výbere certifikátu, modulu pripraví vstupné dáta (ds:SignedInfo) pre výpočet DTBSR a sprístupní objekt zvoleného poskytovateľa pre výpočet digitálneho odtlačku,

- v ďalšom kroku prebehne autentifikácia používateľa pre použitie príslušného SSCD zariadenia, na ktorom je uložený privátny kľúč pre zvolený podpisový certifikát. Autentifikácia prebehne podľa nastavení daného zariadenia,
- ak je autentifikácia pre použitie SSCD úspešná, SSCD výpočíta a vráti modulu D.Signer/XAdES .NET hodnotu elektronického podpisu,
- modul D.Signer/XAdES .NET následne algoritmicky overí hodnotu elektronického podpisu pomocou kódu z knižníc Bouncy Castle Crypto, čím sa zároveň overí dôveryhodná cesta medzi D.Signer/XAdES .NET a SSCD,
- modul D.Signer/XAdES .NET nakoniec vytvorí XML štruktúru podľa profilu XAdES_ZEP a uloží ju do návratovej premennej SignedXmlWithEnvelope,
- v prípade, že došlo pri vytváraní ZEP k chybe, modul D.Signer/XAdES .NET nastaví návratovú premennú ErrorMessage (hodnota návratovej premennej SignedXmlWithEnvelope bude nastavená na prázdny reťazec),
- používateľ následne potvrdí (tlačidlo Ok) alebo zruší (tlačidlo Zrušiť) vytvorenie ZEP a modul D.Signer/XAdES .NET vráti riadenie klientskej aplikácii.

Klientská aplikácia môže následne získať informáciu o výsledku vytvorenia ZEP pomocou modulu D.Signer/XAdES .NET a samotný ZEP z návratových premenných komponentu ErrorMessage a SignedXmlWithEnvelope.

6.2. Stavový diagram

V rámci procesu vytvorenia ZEP nad vstupnými dokumentami prechádza aplikácia D.Signer/XAdES .NET rôznymi stavmi. Jednotlivé prechody popisuje nasledujúci stavový diagram.



Počiatočný stav – klientská aplikácia vytvorila inštanciu modulu D.Signer/XAdES .NET. V rámci svojej inicializácie modul na základe konfiguračných dát z MS Windows Registry načíta zoznam nainštalovaných pluginov D.Signer/XAdES .NET pre typy dátových objektov,

Nasledujúce možné stavy sú:

- SIGN_CREATION_FAILED (v prípade zlyhania inicializácie modulu),
- APPLICATION_INITIALIZED.

APPLICATION_INITIALIZED – inicializácia modulu bola úspešne vykonaná. V nasledujúcom kroku klientská aplikácia zavolá postupne pre každý pripravený dátový objekt (ktorý bol vytvorený volaním metódy CreateObject príslušného pluginu) metódu hlavného modulu addObject, ktorá pridá vstupný dátový objekt do kolekcie dátových objektov na podpísanie (DTBS),

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- INPUT_FEEDED.

INPUT_FEEDED – po vytvorení kolekcie dátových objektov na podpísanie (DTBS) a zavolaní metódy Sign klientskou aplikáciou vykoná modul

D.Signer/XAdES .NET v rámci jednotlivých pluginov validáciu vstupných dátových objektov.

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- INPUT_VALIDATED.

INPUT_VALIDATED – ak prebehla validácia vstupných dokumentov úspešne, modul vykoná v rámci jednotlivých pluginov spracovanie vstupných dátových objektov na DTBSF (aplikovanie príslušných transformácií, napr. kanonikalizácia).

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- DTBSF_CREATED.

DTBSF_CREATED – v nasledujúcom kroku modul zobrazí hlavné okno aplikácie, pričom vizualizácia jednotlivých podpisovaných dátových objektov je realizovaná prostredníctvom príslušných funkcií pluginov pre jednotlivé typy dátových objektov. Používateľ má možnosť si cez GUI modulu D.Signer/XAdES .NET prezrieť podpisované dátové objekty a ďalšie parametre podpisu, zrušiť vytváranie ZEP alebo pristúpiť k výberu podpisového certifikátu.

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- SIGNATURE_CREATION_CANCELED,
- CERT_SELECTION.

CERT_SELECTION – v ďalšom kroku používateľ vyberie pomocou GUI podpisový certifikát. Po výbere podpisového certifikátu, modul pripraví vstupné dáta (ds:SignedInfo) pre výpočet DTBSR a sprístupní objekt zvoleného poskytovateľa pre výpočet digitálneho odtlačku,

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- DTBSR_CREATED.

DTBSR_CREATED – v ďalšom kroku prebehne autentifikácia používateľa pre použitie príslušného SSCD zariadenia, na ktorom je uložený privátny kľúč pre zvolený podpisový certifikát. Autentifikácia prebehne podľa nastavení daného SSCD zariadenia. Používateľ má možnosť opäť zrušiť procedúru vytvárania ZEP.

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- SIGNATURE_CREATION_CANCELED,
- AUTHENTICATION_OK.

AUTHENTICATION_OK – ak je autentifikácia pre použitie SSCD úspešná, SSCD výpočíta pomocou príslušného objektu poskytovateľa digitálny odtlačok

a pomocou SSCD HW hodnotu elektronického podpisu a výsledok vráti modulu D.Signer/XAdES .NET.

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- SIGNATURE_CREATED.

SIGNATURE_CREATED – modul D.Signer/XAdES .NET následne algoritmicky overí hodnotu elektronického podpisu pomocou kódu z knižníc Bouncy Castle Crypto, čím sa zároveň overí dôveryhodná cesta medzi D.Signer/XAdES .NET a SSCD.

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- SIGNATURE_VERIFIED.

SIGNATURE_VERIFIED – modul D.Signer/XAdES .NET nakoniec vytvorí XML štruktúru podľa profilu XAdES_ZEP a uloží ju do návratovej premennej SignedXmlWithEnvelope.

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- XADES_ZEP_CREATED.

XADES_ZEP_CREATED – ak užívateľ potvrdí vytvorenie ZEP stlačením tlačidla Ok, modul D.Signer/XAdES .NET uloží do návratovej premennej SignedXmlWithEnvelope vytvorenú štruktúru XAdES_ZEP. Používateľ má tiež možnosť zrušiť procedúru vytvárania ZEP stlačením tlačidla Zrušiť.

Nasledujúce možné stavy:

- SIGN_CREATION_FAILED,
- SIGNATURE_CREATION_CANCELED,
- Koncový stav.

SIGNATURE_CREATION_CANCELED – ak užívateľ zruší vytvorenie ZEP stlačením tlačidla Zrušiť, modul D.Signer/XAdES .NET vymaže obsah návratovej premennej SignedXmlWithEnvelope.

Nasledujúce možné stavy:

- Koncový stav.

SIGN_CREATION_FAILED – v prípade, že došlo pri vytváraní ZEP k chybe, modul D.Signer/XAdES .NET nastaví návratovú premennú ErrorMessage (hodnota návratovej premennej SignedXmlWithEnvelope bude nastavená na prázdny reťazec).

Nasledujúce možné stavy:

- Koncový stav.

Koncový stav – modul D.Signer/XAdES .NET vráti riadenie volajúcej klientskej aplikácii. Klientská aplikácia môže následne získať informáciu o výsledku vytvorenia ZEP pomocou modulu D.Signer/XAdES .NET a samotnú štruktúru XAdES_ZEP z návratových premenných komponentu ErrorMessage a SignedXmlWithEnvelope.

7. Špecifikácia API

Funkcionalita SCA je v rámci aplikácie D.Signer/XAdES .NET rozdelená do samostatných modulov, ktoré je možné pri nasadení aplikácie kombinovať podľa požiadaviek zákazníka. Aplikáciu D.Signer/XAdES .NET bude tvoriť sada DLL knižníc, ktoré budú poskytovať pre klientské aplikácie nasledujúce integračné rozhrania:

- .Net API – pre .Net aplikácie,
- COM API – wrapper nad .Net API pre iné ako .Net aplikácie,
- COM API (prostredníctvom ATL knižnice) – wrapper nad COM API (primárne pre Internet Explorer)
- NP API – wrapper pre volanie služieb komponentu D.Signer/XAdES .NET z prehliadačov s podporou NPAPI rozhrania (bude odvodené z .Net API komponentu D.Signer/XAdES .NET a jeho pluginov).

Aby bolo možné postupne budovať podporu pre ďalšie typy dátových objektov, medzi hlavným modulom D.Signer/XAdES .NET a pluginmi je navrhnuté abstraktné API, ktoré musí každý plugin implementovať. Hlavný modul bude komunikovať s jednotlivými pluginmi prostredníctvom tohto rozhrania.

Každý plugin musí navyše definovať triedu pre typ dátového objektu, pre ktorý je určený. Metódy a atribúty tejto triedy sú závislé na type dátového objektu a musia byť definované v samostatnom dokumente špecifikácie daného pluginu.

V nasledujúcich kapitolách sú popísané jednotlivé rozhrania.

7.1. Integračné API hlavnej aplikácie

7.1.1. .Net API

Pre .Net aplikácie bude hlavný modul aplikácie D.Signer/XAdES .NET publikovať:

Triedu:

```
Ditec.Zep.DsigXades.XadesSig
```

Metódy a premenné:

```
void SetWindowSize(int width, int height);
void SetSigningTimeProcessing(bool displayGui, bool includeSigningTime);
int Sign
(   string signatureId
,   string digestAlgUri
,   string signaturePolicyIdentifier
);
int Sign11
(   string signatureId
,   string digestAlgUri
,   string signaturePolicyIdentifier
,   string dataEnvelopeId
,   string dataEnvelopeURI
,   string dataEnvelopeDescr
);
int Sign20
(   string signatureId
,   string digestAlgUri
,   string signaturePolicyIdentifier
,   string dataEnvelopeId
,   string dataEnvelopeURI
,   string dataEnvelopeDescr
);
int AddObject(object obj);
string ErrorMessage { get;}
string SignedXmlWithEnvelope { get;}
string SignedXmlWithEnvelopeBase64 { get;}
string SignedXmlWithEnvelopeGZipBase64 { get;}
DateTime SigningTimeUtc { get;}
string SigningTimeUtcString { get;}
string SignerIdentification { get;}
```

7.1.2. COM API

Pre iné prostredia ako .Net bude hlavný modul aplikácie D.Signer/XAdES .NET publikovať nasledujúce COM rozhranie:

ProgId:

```
DSig.XadesSig
```

Funkcie a premenné:

```
HRESULT SetWindowSize([in] LONG width, [in] LONG height);
HRESULT SetSigningTimeProcessing([in] VARIANT_BOOL displayGui, [in]
VARIANT_BOOL includeSigningTime);
long Sign
(   [in] BSTR signatureId
,   [in] BSTR digestAlgUri
,   [in] BSTR signaturePolicyIdentifier
);
long Sign11
(   [in] BSTR signatureId
,   [in] BSTR digestAlgUri
,   [in] BSTR signaturePolicyIdentifier
,   [in] BSTR dataEnvelopeId
,   [in] BSTR dataEnvelopeURI
,   [in] BSTR dataEnvelopeDescr
);
long Sign20
(   [in] BSTR signatureId
,   [in] BSTR digestAlgUri
,   [in] BSTR signaturePolicyIdentifier
,   [in] BSTR dataEnvelopeId
,   [in] BSTR dataEnvelopeURI
,   [in] BSTR dataEnvelopeDescr
);
long AddObject ( [in] VARIANT obj);
[propget] BSTR ErrorMessage();
[propget] BSTR SignedXmlWithEnvelope();
[propget] BSTR SignedXmlWithEnvelopeBase64 ();
[propget] BSTR SignedXmlWithEnvelopeGZipBase64 ();
[propget] BSTR SigningTimeUtcString();
[propget] BSTR SignerIdentification();
```

7.1.3. COM API prostredníctvom ATL knižnice

Aby bolo možné v rámci MS Internet Explorer identifikovať v rámci AddOns výrobcu aplikácie D.Signer/XAdES .NET, bude hlavný modul aplikácie D.Signer/XAdES .NET publikovať popísané COM rozhranie aj prostredníctvom ATL knižnice.

ProgId:

```
DSig.XadesSigAtl
```

Funkcie a premenné:

```
HRESULT SetWindowSize([in] LONG width, [in] LONG height);
HRESULT SetSigningTimeProcessing([in] VARIANT_BOOL displayGui, [in]
VARIANT_BOOL includeSigningTime);
long Sign
(   [in] BSTR signatureId
,   [in] BSTR digestAlgUri
,   [in] BSTR signaturePolicyIdentifier
);
long Sign11
(   [in] BSTR signatureId
,   [in] BSTR digestAlgUri
,   [in] BSTR signaturePolicyIdentifier
,   [in] BSTR dataEnvelopeId
,   [in] BSTR dataEnvelopeURI
,   [in] BSTR dataEnvelopeDescr
);
long Sign20
(   [in] BSTR signatureId
,   [in] BSTR digestAlgUri
,   [in] BSTR signaturePolicyIdentifier
,   [in] BSTR dataEnvelopeId
,   [in] BSTR dataEnvelopeURI
,   [in] BSTR dataEnvelopeDescr
);
long AddObject ( [in] VARIANT obj);
[propget] BSTR ErrorMessage();
[propget] BSTR SignedXmlWithEnvelope();
[propget] BSTR SignedXmlWithEnvelopeBase64 ();
[propget] BSTR SignedXmlWithEnvelopeGZipBase64 ();
[propget] BSTR SigningTimeUtcString();
[propget] BSTR SignerIdentification();
```

7.1.4. Popis funkcií a premenných API hlavnej aplikácie

7.1.4.1. SetWindowSize

Nastavuje veľkosť okna aplikácie D.Signer/XAdES .NET.

Štandardná veľkosť okna aplikácie D.Signer/XAdES .NET je 600x450 bodov. Metóda umožňuje programovo nastaviť inú veľkosť okna aplikácie. Metóda však nedovolí nastaviť veľkosť okna menšiu ako 450x350 bodov a väčšiu ako je rozlíšenie obrazovky používateľa.

7.1.4.2. SetSigningTimeProcessing

Nastavuje spracovanie elementu xades:SigningTime pri vytváraní elektronického podpisu podľa profilu XAdES_ZEP, v1.1 [25].

Parametre:

- displayGui – ak displayGui = True, používateľ bude mať k dispozícii štandardné Windows GUI, ktoré mu zobrazí aktuálny systémový dátum a čas pre overenie aktuálnej hodnoty systémového času PC a jej prípadnú korekciu pred zahrnutím elementu xades:SigningTime do štruktúry vytváraného elektronického podpisu.
- includeSigningTime – v prípade, že podpisová politika definuje tento element ako povinný alebo ak includeSigningTime = True, tak element xades:SigningTime bude zahrnutý do štruktúry vytváraného elektronického podpisu a nastavený na aktuálnu hodnotu systémového času PC. V tomto prípade bude hodnota elementu xades:SigningTime zobrazená používateľovi takisto v rámci parametrov podpisu.

Informácia o povinnosti alebo voliteľnosti elementu xades:SigningTime v rámci podpisovej politiky bude používateľovi zobrazená v rámci parametrov podpisu.

Nastavenie includeSigningTime nemá žiadny význam pri vytváraní elektronického podpisu podľa profilu XAdES_ZEP, v1.0 [24] a v2.0 [26], pretože v rámci týchto profilov je element xades:SigningTime povinný.

7.1.4.3. Sign

Metóda Sign spúšťa samotnú procedúru vytvorenia ZEP podľa profilu XAdES_ZEP, v1.0 [24]. Pri zavolaní metódy Sign sa vykonajú nasledujúce činnosti:

- zobrazenie GUI aplikácie D.Signer/XAdES .NET,
- spracovanie a vizualizácia všetkých dátových objetov, ktoré boli pridané do kolekcie dátových objektov na podpis, pomocou funkcií príslušných pluginov pre príslušné typy dátových objektov,
- spracovanie a vizualizácia ostatných parametrov vytvárania ZEP (napr. verifikačných údajov),
- umožnenie výberu podpisového certifikátu,
- po výbere podpisového certifikátu používateľom spustenie procedúry pre výpočet hodnoty eletronického podpisu,
- matematické overenie elektronického podpisu pomocou kódu z knižníc Bouncy Castle Crypto,
- vytvorenie XML štruktúry ZEP podľa profilu XAdES_ZEP, v1.0 [24].

Parametre:

signatureId – jednoznačné XML Id elementu ds:Signature, povolené znaky: a..z, A..Z, 0..9, „.“ (bodka), „-“ (pomlčka), „_“ (podčiarkovník),

digestAlgUri – identifikátor algoritmu pre výpočet digitálnych odtlačkov v rámci vytváraného elektronického podpisu,

signaturePolicyIdentifier – jednoznačný identifikátor podpisovej politiky použitej pri vytváraní elektronického podpisu.

7.1.4.4. Sign11

Metóda Sign11 spúšťa samotnú procedúru vytvorenia ZEP podľa profilu XAdES_ZEP, v1.1 [25]. Pri zavolaní metódy Sign11 sa vykonajú nasledujúce činnosti:

- zobrazenie GUI aplikácie D.Signer/XAdES .NET,
- spracovanie a vizualizácia všetkých dátových objetov, ktoré boli pridané do kolekcie dátových objektov na podpis, pomocou funkcií príslušných pluginov pre príslušné typy dátových objektov,
- spracovanie a vizualizácia ostatných parametrov vytvárania ZEP (napr. verifikačných údajov),
- umožnenie výberu podpisového certifikátu,
- po výbere podpisového certifikátu používateľom spustenie procedúry pre výpočet hodnoty eletronického podpisu,
- matematické overenie elektronického podpisu pomocou kódu z knižníc Bouncy Castle Crypto,
- vytvorenie XML štruktúry ZEP podľa profilu XAdES_ZEP, v1.1 [25].

Parametre:

signatureId – jednoznačné XML Id elementu ds:Signature, povolené znaky: a..z, A..Z, 0..9, „.“ (bodka), „-“ (pomlčka), „_“ (podčiarkovník),

digestAlgUri – identifikátor algoritmu pre výpočet digitálnych odtlačkov v rámci vytváraného elektronického podpisu,

signaturePolicyIdentifier – jednoznačný identifikátor podpisovej politiky použitej pri vytváraní elektronického podpisu,

dataEnvelopeId – jednoznačné XML Id elementu xzep:DataEnvelope, povolené znaky: a..z, A..Z, 0..9, „.“ (bodka), „-“ (pomlčka), „_“ (podčiarkovník),

dataEnvelopeURI – URI atribút elementu xzep:DataEnvelope,

dataEnvelopeDescr – Description atribút elementu xzep:DataEnvelope.

7.1.4.5. Sign20

Metóda Sign20 spúšťa samotnú procedúru vytvorenia ZEP podľa profilu XAdES_ZEP, v2.0 [26]. Pri zavolaní metódy Sign20 sa vykonajú nasledujúce činnosti:

- zobrazenie GUI aplikácie D.Signer/XAdES .NET,
- spracovanie a vizualizácia všetkých dátových objetov, ktoré boli pridané do kolekcie dátových objektov na podpis, pomocou funkcií príslušných pluginov pre príslušné typy dátových objektov,
- spracovanie a vizualizácia ostatných parametrov vytvárania ZEP (napr. verifikačných údajov),
- umožnenie výberu podpisového certifikátu,

- po výbere podpisového certifikátu používateľom spustenie procedúry pre výpočet hodnoty eletronického podpisu,
- matematické overenie elektronického podpisu pomocou kódu z knižníc Bouncy Castle Crypto,
- vytvorenie XML štruktúry ZEP podľa profilu XAdES_ZEP, v2.0 [26].

Parametre:

signatureId – jednoznačné XML Id elementu ds:Signature, povolené znaky: a..z, A..Z, 0..9, „.“ (bodka), „-“ (pomlčka), „_“ (podčiarkovník),

digestAlgUri – identifikátor algoritmu pre výpočet digitálnych odtlačkov v rámci vytváraného elektronického podpisu,

signaturePolicyIdentifier – jednoznačný identifikátor podpisovej politiky použitej pri vytváraní elektronického podpisu,

dataEnvelopeId – jednoznačné XML Id elementu xzep:DataEnvelope, povolené znaky: a..z, A..Z, 0..9, „.“ (bodka), „-“ (pomlčka), „_“ (podčiarkovník),

dataEnvelopeURI – URI atribút elementu xzep:DataEnvelope,

dataEnvelopeDescr – Description atribút elementu xzep:DataEnvelope.

7.1.4.6. AddObject

Umožňuje pridať dátový objekt vytvorený pomocou metódy CreateObject príslušného pluginu pre daný dátový typ do kolekcie dátových objektov určených na podpis.

7.1.4.7. ErrorMessage

V prípade výsytu chyby v rámci procesu vytvárania ZEP bude obsahovať príslušnú chybovú správu.

7.1.4.8. SignedXmlWithEnvelope

V prípade úspešného vytvorenia ZEP bude obsahovať výslednú XML štruktúru podľa profilu XAdES_ZEP.

7.1.4.9. SignedXmlWithEnvelopeBase64

V prípade úspešného vytvorenia ZEP bude obsahovať výslednú XML štruktúru podľa profilu XAdES_ZEP zakódovanú do Base64.

7.1.4.10. SignedXmlWithEnvelopeGZipBase64

V prípade úspešného vytvorenia ZEP bude obsahovať výslednú XML štruktúru podľa profilu XAdES_ZEP zakódovanú do Base64 a skomprimovanú algoritmom gzip v súlade s RFC 1952.

7.1.4.11. SigningTimeUtc

V prípade úspešného vytvorenia ZEP bude vracať hodnotu elementu xades:SigningTime, teda deklarovaný čas vytvorenia podpisu v UTC.

7.1.4.12. SigningTimeUtcString

V prípade úspešného vytvorenia ZEP bude vracať hodnotu elementu xades:SigningTime, teda deklarovaný čas vytvorenia podpisu v UTC ako reťazec.

7.1.4.13. SignerIdentification

V prípade úspešného vytvorenia ZEP bude vracať Common Name z položky Subject podpisového certifikátu.

7.2. Integračné API pluginu

7.2.1. .Net API

Pre .Net aplikácie musí každý plugin aplikácie D.Signer/XAdES .NET publikovať:

Triedu:

```
<názov_triedy_pluginu>
```

kde <názov_triedy_pluginu> je skutočný názov triedy, napr.

```
Ditec.Zep.DsigXades.Plugins.XmlPlugin
```

Metódy a premenné:

```
object CreateObject(<parametre>);
string ErrorMessage { get;}
```

kde <parametre> sú skutočné parametre metódy CreateObject pre daný typ dátového objektu. Parametre tejto metódy sú závislé na type dátového objektu, pre ktorý je plugin určený a musia byť definované v samostatnom dokumente špecifikácie daného pluginu.

7.2.2. COM API

Pre iné prostredia ako .Net musí každý plugin aplikácie D.Signer/XAdES .NET publikovať nasledujúce COM rozhranie:

ProgId:

```
<progid>
```

kde <progid> je programový identifikátor príslušného COM objektu, napr. DSig.XmlPlugin.

Funkcie a premenné:

```
VARIANT CreateObject ( <parametre> );
[propget] BSTR ErrorMessage();
```

kde <parametre> sú skutočné parametre funkcie CreateObject pre daný typ dátového objektu, odvodené pre COM prostredie z parametrov tej istej funkcie pre .Net prostredie.

7.2.3. COM API prostredníctvom ATL knižnice

Plugin aplikácie D.Signer/XAdES .NET musí publikovať popísané COM rozhranie aj prostredníctvom ATL knižnice.

ProgId:

```
<progid>
```

kde <progid> je programový identifikátor príslušného ATL COM objektu, napr. DSig.XmlPluginAtl.

Funkcie a premenné:

```
VARIANT CreateObject ( <parametre> );
[propget] BSTR ErrorMessage();
```

kde <parametre> sú skutočné parametre funkcie CreateObject pre daný typ dátového objektu, odvodené pre ATL COM prostredie z parametrov tej istej funkcie pre COM prostredie.

7.2.4. Popis funkcií a premenných API pluginu

7.2.4.1. CreateObject

Umožňuje vytvoriť dátový objekt pre daný dátový typ. Parametre tejto metódy sú závislé na type dátového objektu, pre ktorý je plugin určený a musia byť definované v samostatnom dokumente špecifikácie daného pluginu.

7.2.4.2. ErrorMessge

V prípade výsytu chyby v rámci procesu vytvárania dátového objektu bude obsahovať príslušnú chybovú správu.

7.3. Abstraktné API pre pluginy

Každý plugin, ktorý má byť integrovaný ako súčasť aplikácie D.Signer/XAdES .NET musí implementovať nasledujúce abstraktné API.

```
public interface IPlugin
{
    /// <summary>
    /// get visualizationm control
    /// </summary>
    /// <returns></returns>
    Control GetVisualizer();

    /// <summary>
    /// get error message
    /// </summary>
    string ErrorMessage {get;}

    /// <summary>
    /// set object data
    /// </summary>
    /// <param name="data"></param>
    bool SetData(object data, Core.DigestAlgs hashAlg,
        string envelopeNS);

    /// <summary>
    /// get full type name of data object
    /// </summary>
    string TypeName { get;}

    /// <summary>
    /// get version of plugin
    /// </summary>
    string PluginVersion { get;}

    /// <summary>
    /// get objectId of data object
    /// </summary>
    string GetObjectId(object data);

    /// <summary>
    /// get objectDescription of data object
    /// </summary>
    string GetObjectDescription(object data);

    /// <summary>
    /// get string array of objects
    /// </summary>
    /// <returns>list of objects, or empty list</returns>
    List<string> GetDSObjects();

    /// <summary>
    /// get sring array of manifests
    /// </summary>
    /// <returns>list of Manifests, or empty list</returns>
    List<string> GetDSManifests();

    /// <summary>
    /// get string array of dataobjectformat
    /// </summary>
```

```
    /// <returns>list of DataObjectFormats, or empty list</returns>
    List<string> GetXadesDataObjectFormats();

    /// <summary>
    /// get sring array of references
    /// </summary>
    /// <returns>list of References, or empty list</returns>
    List<string> GetDSReferences();

    /// <summary>
    /// clean up of memory
    /// </summary>
    /// <returns>void</returns>
    void CleanUp();
}
```

Trieda, ktorá implementuje definované abstraktné rozhranie, musí mať zároveň definovaný nasledujúci atribút:

```
[AttributeUsage(AttributeTargets.Class)]
public class PluginDescriptionAttribute : Attribute
{
    public PluginDescriptionAttribute(string description)
        {
            this.description = description;
        }

    private string description;

    public string Description
        {
            get { return this.description; }
            set { this.description = value; }
        }
}
```

7.3.1. Popis metód abstraktného API pre pluginy

7.3.1.1. GetVisualizer

Vráti GUI ovládač pre vizualizáciu dát a verifikačných parametrov pre daný typ dátového objektu (typu Control).

7.3.1.2. ErrorMessage

V prípade výskytu chyby v rámci vykonávania metódy pluginu bude obsahovať príslušnú chybovú správu (typu string).

7.3.1.3. SetData

Pridá dátový objekt do kolekcie dátových objektov na podpis, spracuje dátový objekt (aplikovanie príslušných transformácií, vytvorenie DTBSF). V prípade úspechu vráti true, inak false.

Parametre:

data – dátový objekt (typ object),

hashAlg – algoritmus pre výpočet digitálneho odtlačku (typ Core.DigestAlgs),

envelopeNS – namespace obálky vytváranej XML štruktúry podpisu, teda XAdES_ZEP v1.0, v1.1, resp. XAdES_ZEP v2.0 (typ string).

7.3.1.4. TypeName

Vráti úplný názov dátového objektu pre dáta a verifikačné parametre pre daný dátový typ (typu string).

7.3.1.5. PluginVersion

Vráti informáciu o verzii pluginu (typu string).

7.3.1.6. GetObjectId

Vráti objectId (typu string) príslušného dátového objektu – data.

Parametre:

data – dátový objekt (typ object).

7.3.1.7. GetObjectDescription

Vráti objectDescription (typu string) príslušného dátového objektu – data.

Parametre:

data – dátový objekt (typ object).

7.3.1.8. GetDSObjects

Vráti zoznam XML štruktúr (typu List<string>) ds:Object pre jednotlivé dátové objekty a príslušné verifikačné údaje z kolekcie dátových objektov daného typu, ktoré sú určené na podpis, podľa profilu XAdES_ZEP.

7.3.1.9. GetDSManifests

Vráti zoznam XML štruktúr (typu List<string>) ds:Manifest pre jednotlivé dátové objekty a príslušné verifikačné údaje z kolekcie dátových objektov daného typu, ktoré sú určené na podpis, podľa profilu XAdES_ZEP.

7.3.1.10. GetXadesDataObjectFormats

Vráti zoznam XML štruktúr (typu List<string>) xades:DataObjectFormat pre jednotlivé dátové objekty a príslušné verifikačné údaje z kolekcie dátových objektov daného typu, ktoré sú určené na podpis, podľa profilu XAdES_ZEP.

7.3.1.11. GetDSReferences

Vráti zoznam XML štruktúr (typu List<string>) ds:Reference do ds:SignedInfo pre jednotlivé dátové objekty a príslušné verifikačné údaje z kolekcie dátových objektov daného typu, ktoré sú určené na podpis, podľa profilu XAdES_ZEP.

7.3.1.12. CleanUp

Obsahom implementácie na strane pluginu by malo byt vyčistenie pamäte od zdrojov, ktoré nemusia byť uvoľnené .Net garbage collectorom (napríklad niektoré COM objekty, cyklické referencie, atď.)

8. Konfiguračné parametre

Činnosť aplikácie D.Signer/XAdES .NET je riadená pomocou konfiguračných parametrov, ktoré sú uložené v rámci MS Windows Registry. Konfiguračné parametre aplikácie D.Signer/XAdES .NET tvoria zoznam podporovaných podpisových politík a konfigurácia filtra pre podpisové certifikáty.

Výrobca, resp. integrátor aplikácie D.Signer/XAdES .NET je povinný zabezpečiť také nastavenie konfigurácie aplikácie a parametrov volania metód rozhrania aplikácie, aby aplikácia vytvárala podpis v súlade so špecifikovanou podpisovou politikou. Informácie o podporovaných podpisových politikách budú distribuované spolu s aplikáciou D.Signer/XAdES .NET.

Význam jednotlivých konfiguračných parametrov pre podporované podpisové politiky je nasledujúci:

- SignaturePolicies – zoznam podporovaných podpisových politík, v súlade s ktorými je možné pomocou aplikácie D.Signer/XAdES .NET vytvárať elektronický podpis
    - ⇒ key name = jednoznačný identifikátor podpisovej politiky,
        - ♦ DigestAlgUri – Identifikátor algoritmu pre výpočet odtlačku tejto podpisovej politiky, napr. "http://www.w3.org/2000/09/xmldsig#sha1"
        - ♦ DigestValue – hodnota odtlačku tejto podpisovej politiky, vypočítaná pomocou algoritmu, ktorý je špecifikovaný v DigestAlgUri a kódovaná do base64,
        - ♦ Identifier – jednoznačný identifikátor tejto podpisovej politiky,
        - ♦ NotBefore – dátum a čas začiatku platnosti podpisovej politiky,
        - ♦ NotAfter – dátum a čas konca platnosti podpisovej politiky,
        - ♦ URL – URL, na ktorom je táto podpisová politika k dispozícii, prípadne na ktorom je možné overiť, či podpisová politika nebola predčasne zrušená,
        - ♦ SigningTime – povinnosť zahrnutia elementu xades:SigningTime do vytváraného elektronického podpisu, 1 = xades:SigningTime je v rámci tejto podpisovej politiky povinný element, 0 = xades:SigningTime je v rámci tejto podpisovej politiky voliteľný element, teda bude zahrnutý do podpisu na základe parametrov volania metódy SetSigningTimeProcessing.

Aplikácia D.Signer/XAdES .NET bude primárne slúžiť na vytvorenie zaručeného elektronického podpisu. Zákon č. 215/2002 Z.z. o elektronickom podpise a o zmene a doplnení niektorých zákonov v znení neskorších predpisov však umožňuje použiť v styku s orgánmi verejnej moci aj obyčajný elektronický podpis. Používateľ môže mať na svojom SSCD zariadení vygenerovaných viacero

kľúčových párov, na ktoré má vystavené kvalifikované alebo nekvalifikované certifikáty.

Aby používateľ pri vytvorení (zaručeného) elektronického podpisu omylom nepoužil nesprávny typ certifikátu, aplikácia D.Signer/XAdES .NET bude umožňovať konfiguráciu filtra pre podpisové certifikáty, ktoré sa majú používateľovi zobraziť pri výbere podpisového certifikátu. Predpokladá sa, že filter certifikátov bude nastavený integrátorom aplikácie D.Signer/XAdES .NET do portálu príslušného orgánu verejnej moci a bude sa teda distribuovať spolu s aplikáciou D.Signer/XAdES .NET. Preto aplikácia samotná nebude poskytovať žiadne GUI pre nastavenie filtra certifikátov.

Aplikácia D.Signer/XAdES .NET bude len umožňovať používateľovi zapnúť alebo vypnúť filter certifikátov v okne pre výber podpisového certifikátu. Nastavenia pre filter certifikátov budú distribuované spolu s aplikáciou D.Signer/XAdES .NET.

Význam jednotlivých konfiguračných parametrov pre prístupné SSCD zariadenia a podpisové certifikáty je nasledujúci:

- Name – názov filtra certifikátov,
- Default – indikátor, či má byť v rámci GUI aplikácie D.Signer/XAdES Java filter certifikátov zapnutý/vypnutý; povolené hodnoty: true/false,
- CertRules – množina pravidiel pre vyhodnotenie certifikátov; jednotlivé pravidlá (elementy <CertRule>) sa vyhodnocujú cez logické OR,
    - ⇒ CertRule – jedno pravidlo pre vyhodnotenie certifikátov; jednotlivé položky pravidla sa vyhodnocujú cez logické AND,
        - ♦ KeyUsage – certifikát musí mať v rámci KeyUsage nastavenú príslušnú hodnotu (digitalSignature, nonRepudiation, atď.), viď [7]
        - ♦ CertificatePolicyOID – certifikát musí mať v rámci zoznamu certifkačných politík uvedenú certifikačnú politiku s daným OID,
        - ♦ QCStatementOID – certifikát musí mať v rámci položky QCStatements uvedené OID príslušného QCStatementu,
        - ♦ SubjectAttrValue – v rámci poľa Subject certifikátu prevedeného do textovej formy sa musí nachádzať výraz: <názov/OID attribútu DN>=<regulárny výraz>; regulárny výraz sa bude vyhľadávať v hodnote daného atribútu; vyhľadávanie bude case-insensitive.