ELECTRA SIGNATURE ÜGYFÉLKAPU, PDF/A XMP ALAPÚ DOKUMENTUM HITELESÍTŐ SZOLGÁLTATÁS

Szerzők:

Roskó Tibor
Debreceni Egyetem

Lektorok:

Bajalinov Erik
Nyíregyházi Egyetem

Feketéné Nyíri Lívia
Jókai Mór ált. isk. és Alapfokú műv. isk.

Mező Ferenc
Eszterházy Károly Egyetem

Első szerző e-mail címe:
r.tibor92@gmail.com

Roskó, Tibor (2017): Electra Signature: Ügyfélkapu, PDF/a XMP alapú dokumentum hitelesítő szolgáltatás Különleges Bánásmód, III. évf. 2017/2. szám, 65-85. DOI 10.18458/KB.2017.2.65

Absztrakt

Tanulmányunkban egy Ügyfélkapu, kormányzati autentikáción alapuló elektronikus aláírás megoldást szeretnénk ismertetni, mely a szemantikus Web alapjait figyelembe véve került megtervezésre. Bevezetésként az e-aláírás kialakulásának folyamatáról készítettünk egy Elektronikus aláírás című tanulmányt (Roskó, 2017), mely részletezi a főbb mérföldköveket, amelyek elvezettek a ma ismert elektronikus aláírás kialakulásáig. E tanulmányunkban, magát a kidolgozott Electra Signature rendszert és szolgáltatásait részletezzük. Köszönhetően a szemantikus Web biztosította infrastruktúráknak, a rendszer képes együttműködni más szemantikus Web alkalmazásokkal, például a FOAF, mint digitális névjegykártya szolgáltatás. A szigorú jogszabályi megfelelőség érdekében részletesen megismertük az ide vonatkozó jogszabályokat, melyeket az olvasóval is megismertettünk a már említett bevezető tanulmányunkban.

Zárásként itt is külön ki szeretnénk emelni a különleges bánásmódot igénylő embertársaink segítését. Projektünk keretében lehetőség nyílik a már korábban megismert, sokrétű lehetőségeket biztosító Ügyfélkapu használatára, ezzel egy, a tanúsítványon alapuló megoldástól egyszerűbb, viszont azonos biztonságú e-aláírás létrehozására, mely nem csak a különleges bánásmódot igénylő személyeknek jelenthet könnyebbséget, hanem minden felhasználónak. Továbbá az Ügyfélkapu szerepet kaphat még az e- és m-learning alapú oktatási rendszerekben (Mező és Psenáková, 2009, 2010) is, mely rendszerek részeként indirekt módon az esélyegyenlőség biztosításához szintén nagyban hozzájárulhat.

Kulcsszavak: elektronikus aláírás, szemantikus Web, elektronikus ügyintézés, IoT

Diszciplína: Informatikai tudományok

Abstract

An Ügyfélkapu, government authentication based electronic signature service will be detailed in our paper. This service was designed to accomplish the rules of semantic Web and component based design. The important milestones of the birth process of e-signature are described in our other paper, Electronic signature. The services of Electra Signature are described in this study. Thanks to the semantic Web infrastructures Electra can cooperate with other semantic Web based applications, for example FOAF using as a digital business card. We also inspected the laws of e-signature and we shared these with you in the previously mentioned study to help creating well formed agreements.

We would like to absolutely highlight the importance of special needed people' help options such as Electra service. Thanks to the Ügyfélkapu they can easier use e-signature than general certificate based solution.

Keywords: electronic signature, semantic Web, electronic official administration services, IoT

Disciplines: Computer Science

BEVEZETÉS

Elektronikus aláírás című tanulmányunkban (Roskó, 2017) megismerhették az elektronikus aláírás kialakulásának főbb mérföldköveit, illetve jelen tanulmányunk alapját adó Electra Signature rendszer alternatíváiként tekinthető e-aláírás szolgáltatásokat.

Cikksorozatunk második tanulmányában, az Electra Signature Ügyfélkapu azonosításra épülő elektronikus dokumentum aláíró megoldásunk infrastruktúráit, szolgáltatásait kívánjuk ismertetni.

Elöljáróban tekintsük át a technológia tól-ig folyamatát, mely alapjaiban írja le az aláírási metódust.

A folyamat elindítása egy PDF/a típusú fájl létrehozásával kezdődik, mely az általunk fejlesztett online felületen összerendelődik az aláírók, Ügyfélkapuban regisztrált e-mail címével és az aláírás egymás utáni sorrendjét jelölő szinttel és mindezek egy RDF metaadat struktúrában kerülnek a fájl binárisába, valamint aláírásra beküldés révén a rendszer adattárában is létrejön a dokumentum hivatkozása az aláírók adataival. Ezt követően szintén az általunk megalkotott online felület igénybevételével elindításra kerül az aláírás folyamata. Elsőként az Ügyfélkapu bejelentkezés kerül ellenőrzésre, amennyiben nincs aktuálisan Ügyfélkapu munkamenetünk, a magyarorszag.hu rendszere felkér minket az azonosításra. Ennek sikeres lefutását követően visszakerülünk az Electra felületére, ahol már emberi beavatkozás nélkül kerül át a vezérlés a háttérrendszerhez, amely rögzíti az aláírás tényét. Ha minden sikeresen végbement, visszajelzést kapunk az aláírás megtörténtéről. Egy fájl aláírási információi hasonló módon érhetőek el, szintén az online felületet igénybe véve, a kérdéses fájlt feltöltve visszakapjuk a rendelkezésre álló aláírás információkat.

A fent leírt folyamatokból egyértelműen kitűnik, hogy nincs szükség semmilyen befektetésre, illetve internet elérésen kívül egyéb eszköz használatára, például chipkártya, tanúsítvány.

Bízunk benne, hogy a tanulmányban bemutatásra kerülő megoldás sokak számára nyújthat majd segítséget az elektronikus ügyintézés szemantikus Web alapokra való felkészítésével és mielőbbi elterjedésében, illetve, ahogy korábban is jeleztük, e projektünk is kiválóan alkalmas a hátrányos helyzetben lévő személyek segítésére. Az Ügyfélkapu rendszer révén egyszerűsödhet az e-aláírás létrehozása és ellenőrzése is. A tanúsítványok és egyéb

célalkalmazások elhagyásával egy kevésbé hozzáértő személy is hibamentesen képes lehet ellenőrizni az elektronikusan aláírt dokumentumot, illetve nem elhanyagolható szempont, hogy a hosszú távú megőrzéshez nincs szükséges folyamatos után követésre, időbélyegzésekre.

ELECTRA SIGNATURE

E, fő fejezetben szeretnénk részletesen kifejteni az általunk megtervezett és implementált dokumentum hitelesítő szolgáltatás alapjait, működésének folyamatait, jövőbeni szerepét.

Rendszerünk alapját két infrastruktúra építi fel, ezek közül mindkettő térítésmentesen elérhető, ez nagyban hozzájárul a szolgáltatás nonprofit jellegének kialakításához, fenntarthatóságához. A következő két alfejezetben e bázisokat ismerheti meg az olvasó.

Ügyfélkapu

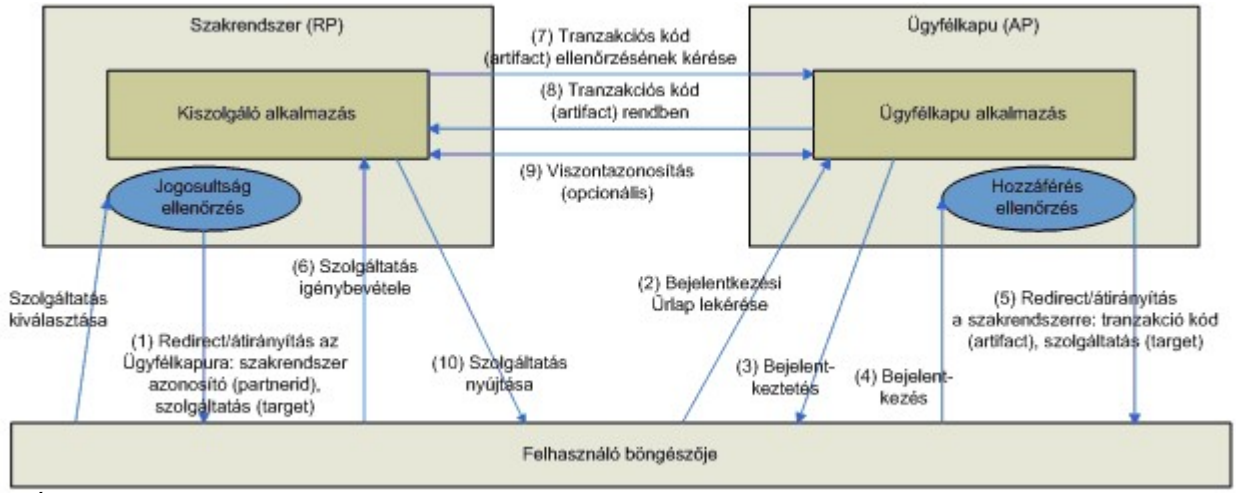
A magyar kormány által üzemeltetett, elektronikus ügyintézést támogató infrastruktúra összefogó online rendszere, mely Ügyfélkapu néven ismert. Elérhető a www.magyarorszag.hu címen, minden személyi azonosításra alkalmas okmánnyal rendelkező személy igényelhet hozzáférést, nem csak magyar állampolgár. A papírmentes irány képviseletében több ügytípus is végigvihető papír felhasználást mellőző folyamatban.

A hozzáférés két irányból elérhető: hivatali szakrendszer vagy általános felhasználó. Az utóbbi lényegében a lakosságot fedi le, míg a hivatali rendszerek ügyintézői, illetve háttérrendszerek az ügykezelést végzik.

Mint szolgáltatási rendszer nagyszerű lehetőségeket rejt magában, ez elsősorban a hiteles, megbízható személy azonosító funkciójából adódik, melynek felhasználásával számos autentikációs feladat elvégezhető. Egyetlen hátránya, hogy kizárólag nonprofit vagy állami szerv kérhet hozzáférést a központi rendszerhez. Esetünkben, az egyetem, mint állami szervezet, bekapcsolódhat az azonosítást végző SSO modulba, ezáltal megnyílik a lehetőség az Ügyfélkapu nyújtotta autentikáció felé.

Tekintettel arra, hogy a lakosság nagy számban használja (Roskó, Adamkó, 2016), illetve széles körben ismert, nem kell túlzottan bizonygatni, valóban megbízható és hiteles forrása lehet egy autentikációs folyamatnak. Ezen szempontokat szem előtt tartva döntöttünk mi is a csatlakozás és integrálás mellett.

Az alábbi 1. ábra bemutatja az Ügyfélkapura épülő azonosítási folyamatot. Első lépésként a csatlakozott szakrendszerben elinduló esemény közben ellenőrzi, hogy azonosította-e már magát a felhasználó, amennyiben nem, átirányítja az Ügyfélkapu központi bejelentkezési felületére, ahonnan sikeres azonosítást követően visszatér az igényelt szakrendszer szolgáltatásához, immár azonosított ügyfélként. A bejelentkezés során az SSO egy token és egy target paramétert küld vissza a szakrendszer felé, a token segítségével tudja ellenőrizni a felhasználót, a target pedig a kiválasztott szolgáltatást jelzi a szakrendszer központi kiválasztó rendszerének. Erre azért van szükség, mert a csatlakozott szolgáltató egyetlen visszatérési URL-t tud megadni, így szükséges a szolgáltatások megkülönböztetése és egy központosított döntési-navigálási felület beépítése, mely a target-nek megfelelő alcímre navigál, elérve a kiválasztott szolgáltatást.



1. Ábra: Szakrendszer – ÜK kommunikáció (forrás: KIB.21.ajánlás)

Adobe Extensible Metadata Platform

Az Adobe által kidolgozott RDF struktúrájú metaadat beágyazási lehetőség, mely 2012-től ISO 16684-1 szabványként elfogadott. Lehetőséget biztosít szemantikus Web alapoknak megfelelő metainformációk rögzítésére egy PDF dokumentumban, elsősorban a PDF archív fájlokhoz ajánlott, de ettől függetlenül a standard PDF is képes alkalmazni. (Adobe) (Gasiorowski-Denis, 2012) (TechNote 0003., 2008)

Rendkívül széles körben alkalmazható tulajdonsága a sokrétű információ tároló képességéből adódik. A szöveges tartalom mellett beágyazhatunk bélyegképeket, logókat is, illetve mindent, ami XML formátumba konvertálható.

Magát az XMP csomagot is beágyazhatjuk széles spektrumú fájlkiterjesztésekbe, ezzel nem csak a tartalma, de a felhasználási lehetőségei is nagy teret biztosítanak. A PDF-en kívül elsősorban képfájlokban terjedt el a használata, tekintettel arra, hogy számos készülékgyártó alkalmazza e szabványt a metaadatok rögzítésére. Nagy előnye, hogy az XML-re támaszkodva a saját igényeinkhez igazodó tag-ek implementálhatók, ezáltal szinte korlátlan alkalmazhatóságot kapunk. (Adobe)

A PDF-ből kiindulva, saját rendszerünkön szeretnénk ismertetni az implementálás, beágyazás menetét. Magát a csomagot egy PDF objektumban kell elhelyezni egy xpacket konténerben, az objektum típusának meg kell adni, hogy XML metaadat leíró struktúrát tartalmaz. Az általunk létrehozott rendszerhez nem állt rendelkezésre megfelelő XSD, ezért egy sajátot hoztunk létre, amely leírja az aláírásban résztvevők adatainak formátumát. (Adobe)

Leíró XSD az Electra Signature-höz

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:sign="http://w2.inf.unideb.hu/~maszatweb/signature/" elementFormDefault="qualified"
targetNamespace="http://w2.inf.unideb.hu/~maszatweb/signature/" xmlns:xs="http://www.w3.org/2001/XMLSchema">
                        <xs:annotation>
                            <xs:documentation>Electra sign signature schema - 2016</xs:documentation>
                        </xs:annotation>

                        <xs:simpleType name="email_type">
                            <xs:restriction base="xs:string">
                                <xs:pattern value="[^@]+@[^\.]+\..+"/>
                            </xs:restriction>
                        </xs:simpleType>

                        <xs:element name="signatures">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="signature" minOccurs='1' maxOccurs="unbounded">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="email" type="sign:email_type" minOccurs='1'
maxOccurs="1"></xs:element>
                                                <xs:element name="level" type="xs:positiveInteger" minOccurs='1'
maxOccurs="1"></xs:element>
                                                <xs:element name="vices" minOccurs='0' maxOccurs="1">
                                                    <xs:complexType>
                                                        <xs:sequence>
                                                            <xs:element name="vice" minOccurs='1' maxOccurs="5">
                                                                <xs:complexType>
                                                                    <xs:sequence>
                                                                        <xs:element name="email"
type="sign:email_type" minOccurs='1' maxOccurs="1"></xs:element>
                                                                    </xs:sequence>
                                                                </xs:complexType>
                                                            </xs:element>
                                                        </xs:sequence>
                                                    </xs:complexType>
                                                </xs:element>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
</xs:schema>
```

A signatures konténer tartalmazza az aláírók e-mail címét és aláírási szintjét hordozó signature blokkot, melyből kötelezően kell legalább egynek lenni. Ezen kívül a signature blokk tartalmazza a továbbfejlesztés eredményeként megvalósuló aláírás helyettes kijelölését biztosító konténert, a vices-t, melyben minden egyes helyettes egy vice elemként jelenik meg, ez hordozza az e-mail címet. A rendszer nem XSD validációval, de ellenőrzi, hogy egy dokumentum csak egy érvényes signatures blokkot tartalmazzon, vagyis az aláírók módosításakor felüldefiniálja a meglévőt. A PDF olvasó mindig az utolsó objektumot tekinti érvényesnek, ha ugyanabból több is szerepel egy binárisban. Ezt úgy érhetjük el, hogy az adott objektumot felüldefiniáljuk, vagyis létrehozunk egy ugyanolyan azonosítóval rendelkező új objektumot.

A következő kód egy példa aláírás struktúrát mutat be, melyet az Electra rendszer ágyazott be a PDF-be:

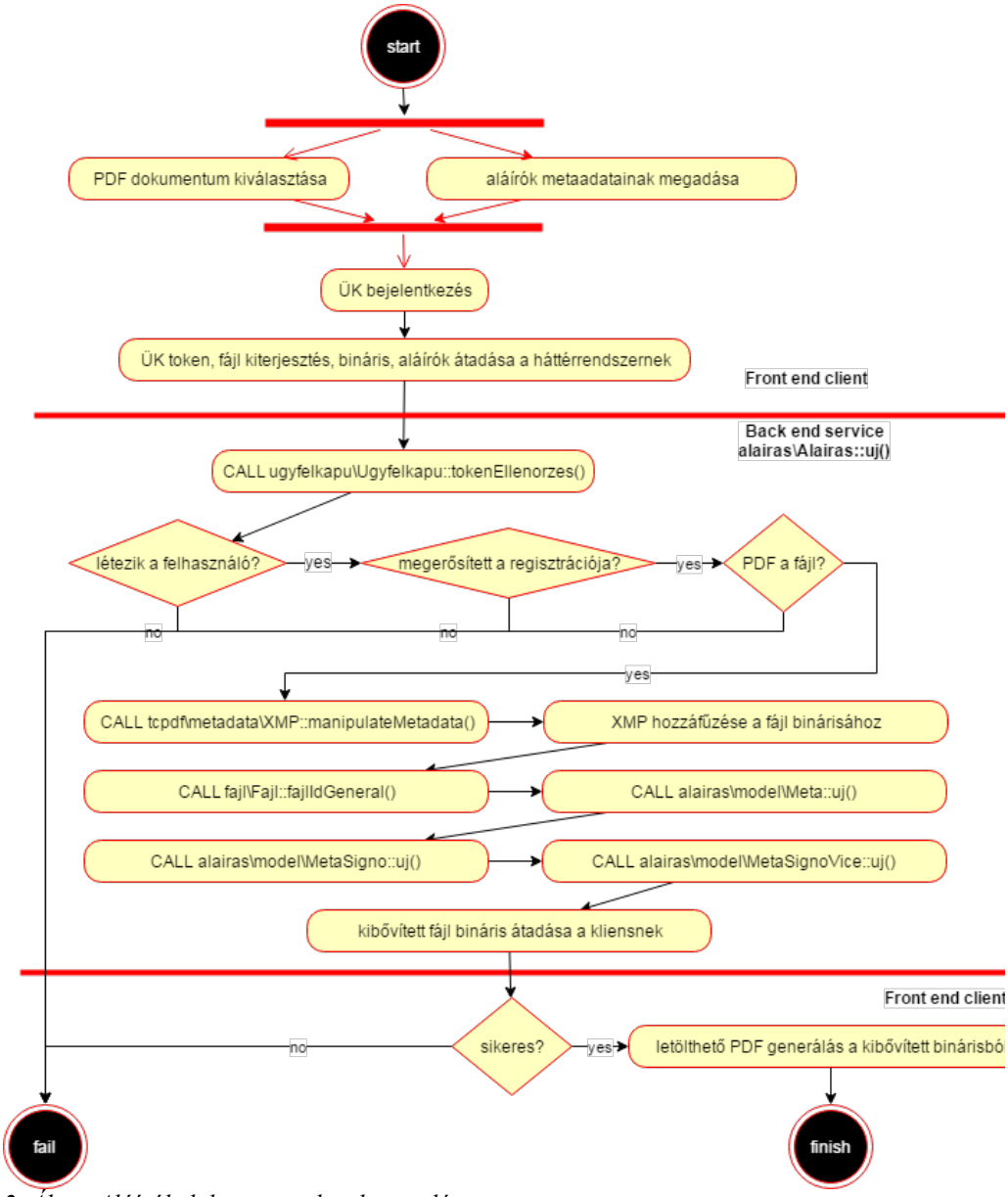
```
7 0 obj
<</Subtype/XML/Type/Metadata/Length 1124>>
stream
<?xml version="1.0"?>
<?xpacket begin="ď»ż" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/">
	<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
		<rdf:Description xmlns:dc="http://purl.org/dc/elements/1.1/" rdf:about="">
			<dc:format>application/pdf</dc:format><dc:title><rdf:Alt><rdf:li xml:lang="x-default">T-
4800/901</rdf:li></rdf:Alt></dc:title>
		</rdf:Description>
		<rdf:Description xmlns:pdf="http://ns.adobe.com/pdf/1.3/" rdf:about="">
			<pdf:Producer>ABBYY FineReader 12</pdf:Producer><pdf:Keywords/>
		</rdf:Description>
		<rdf:Description xmlns:xmp="http://ns.adobe.com/xap/1.0/" rdf:about="">
			<xmp:CreatorTool/><xmp:CreateDate>2016-10-
17T19:53:50Z</xmp:CreateDate><xmp:ModifyDate>2016-10-17T19:53:50Z</xmp:ModifyDate>
		</rdf:Description>
		<rdf:Description xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/" rdf:about="">
			<xmpMM:DocumentID>uuid:{1E55B97A-A61A-4917-AE1C-
428AFC6C43E8}</xmpMM:DocumentID>
		</rdf:Description>
		<rdf:Description xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/" rdf:about="" pdfaid:part="1"
pdfaid:conformance="A"/>

		<s:signatures xmlns:s="http://w2.inf.unideb.hu/~maszatweb/signature/">
			<s:signature>
				<s:email>rtibor@unideb.hu</s:email>
				<s:level>1</s:level>
				<s:vices>
					<s:vice><s:email>adamko@unideb.hu</s:email></s:vice>
				</s:vices>
			</s:signature>
		</s:signatures>
	</rdf:RDF>
</x:xmpmeta>
<?xpacket end='w'?>
endstream
endobj
```

A szolgáltatás tartalma, működése

Az Electra rendszer által biztosított három alszolgáltatás elsődleges szemléletben egymásra épülve lett kialakítva. A komponens alapú tervezés révén ettől elrugaszkodva mindhárom részmodul alkalmazható egymástól függetlenül is. A felépítésnek köszönhetően némi újratervezést, feladatra optimalizálást követően akár teljesen eltérő feladatokra is alkalmazhatók. A következő három alfejezetben e szolgáltatásokat kívánjuk részleteiben bemutatni, megismertetni az olvasóval, leendő felhasználóval.

A szolgáltatás célja: a dokumentumot aláírók metaadatainak rögzítése. A folyamat során az Ügyfélkapu e-mail címek, aláírási szintek és szükség esetén az aláíró helyettesek írhatók le szemantikus Web formában, RDF-ben. Ez az aláírási folyamat első alappillére.



2. Ábra: Aláírók dokumentumhoz kapcsolása

A 2. ábra lépésről-lépésre szemlélteti a folyamatot. A megosztott architektúra működése jól kitűnik, a böngészőben futó kliens és a háttérkiszolgáló egymással együttműködve valósítja meg a
szolgáltatást

Első lépésként az aláírandó dokumentum kijelölése és az aláírók metaadatainak megadása szükséges, majd Ügyfélkapu azonosítás a feltöltő részéről, ezt követi az információk átadása a háttérrendszer felé.

A tényleges munka elvégzése itt történik meg, az általunk implementált metódus szerint, első lépésben az Ügyfélkapu token alapján ellenőrzésre kerül a feltöltő: létezik, a regisztrációja megerősített, valamint PDF fájlt küldött be. Ha mindent rendben talál az

algoritmus, folytatódik a lánc, immár a szintén általunk -különálló projekt keretében- továbbfejlesztett TCPDF komponens létrehozza és hozzáfűzi az aláírók metaadatait tartalmazó struktúrát az eredeti fájlhoz. Ha minden hiba nélkül végbemegy, a kiegészített fájl kap egy egyedi azonosítót, mely ettől kezdve a rendszerben azonosítja. A kiszolgáló utolsó feladataként rögzíti a fájl és aláírói, illetve ezek helyetteseinek minden metaadatát az adatbázisba. Az aláírás létrehozását ismertető alfejezetben részletesen megismerhetjük ezen információkat, illetve, ezek alapján miként jöhet létre aláírás. Záró lépésként a metodika alapján visszakerül a vezérlés a klienshez -ha nem volt semmilyen hiba a korábbi lépéseknél- a kiegészített fájl binárisával, melyből letölthető PDF készül. Ez lesz az a fájl, amelyet az aláírók részére kell átadni, aláírásra. A szabály alapján ez a dokumentum már nem változtatható anélkül, hogy az aláírás lehetősége el ne vesszen. Ezt hivatott biztosítani az említett egyedi azonosító.

Az azonosítót létrehozó *fajlIdGeneral()* függvényt szeretnénk részletesen is bemutatni, tekintettel az általa létrehozott azonosító kulcsfontosságú szerepére: biztosítja az aláírásra beküldést követő módosíthatatlanságot. A folyamat során a fájl binárisáról egy sha-256 hash-t képez, a lenyomat minden esetben egyedi, a legkisebb módosítás is nagyfokú eltérést okoz a hash-ben. Ezáltal azonnal kiszűrhető, a fájl módosítva lett. A rendszer viszont nem azt vizsgálja, hogy módosították-e a fájlt, erre nem sok esély lenne, még szabálytalan adattárolással sem. Ha eltárolnánk a fájlok binárisait -ami természetesen nem lenne etikus, szabálykövető- még akkor sem mondhatná azt a rendszer, hogy igen ez a fájl módosítva lett, mert előfordul, hogy csak az aláírók körében történt módosítás. Ettől egy jóval egyszerűbb folyamat deríti ki a csalási kísérletet:

1. az aláírók hozzárendelésekor menti a fájl metaadatokat is a rendszer, ezáltal az ezt követő lépésekben minden módosítást követően újként érzékelné
2. ha második vagy azt követő szintű aláíróhoz kerül illegálisan manipulált dokumentum, a rendszer nem is engedi aláírni, hiszen az új dokumentumként való értelmezés során a megelőzők még nem írták alá

ezen elvek alapján nincs lehetőség úgy manipulálni egy dokumentumot, hogy az kárt okozhasson bármelyik aláíró fél számára. *(A két folyamatbeli okozat a teljes aláírási folyamat áttekintését követően válhat egyértelművé, tekintettel arra, hogy nincs még minden lépés ismertetve)*

A hash képzés során használt algoritmus egyirányú karaktersorozatot állít elő a bemenő szövegből, azaz nincs rá mód, hogy rekonstruáljuk az eredeti tartalmat, ez elsősorban adatvédelmi szempontok figyelembe vétele miatt releváns: a rendszerből nem lehet előállítani az aláírt dokumentumot, nem kerül tárolásra. A keletkezett string minden esetben 64 karakter hosszú, 256 bites "szót" eredményez.

Példa az sha-256 algoritmus működésére:

bemenet ez egy bemenő szöveg

kimenet 73cd9b16119447e76740a6929857d371ccc0cbcc6ce1861cf8921f90f3b2aa8e

A megelőző fejezetekben már ismertettük az XMP működését, miként kerül feldolgozásra a PDF-ben. Ennek tükrében szeretnénk betekintést nyújtani a generáló metódus működésébe. A folyamat első lépése, hogy ellenőrizzük létezik-e már XMP objektum a beküldött fájlban, egyszerűbb eset, ha még nem létezik. Ekkor ugyanis csupán a csatolt tartalmat felhasználva előállítjuk az új objektumot. Amennyiben már korábban kapcsoltak szemantikus metaadatot a fájlhoz, felül kell definiálnunk a tartalmát, természetesen a meglévő információk megtartása mellett. E művelet már több munkát igényel a komponens részéről, az XML szabványos függvényeit felhasználva az általunk definiált XSD-re épülő csomópontokat hozunk létre,

melyek a hozzákapcsolni kívánt aláírók adatait írják le. Függetlenül melyik ágat kell végrehajtani, egyaránt többszörösen validált műveletsorok vezetnek el a végeredmény elkészültéig, ezzel biztosítva a szabványos és megbízható munkafolyamatokat.

Az eddigi ismeretek alapján egyértelmű, hogy az architektúra kiválasztása a szemantikus Webhez való közvetlen kötődés révén történt. Az XML RDF alapjaiban járul hozzá a gépi információfeldolgozás támogatásához.

A működés során a legnagyobb kockázatot a PDF dokumentum binárisának feldolgozása jelenti, ennek csökkentésére a szabvány szerint szükséges elemek meglétének vizsgálatával igyekszünk reagálni. A legnagyobb problémát az XREF kódolt formában történő tárolása jelenti, nem minden esetben volt képes az alkalmazott modul dekódolni az offset táblát, emiatt a további műveletek elvégzése már nem lehetséges. (Adobe) (TechNote 0003., 2008) Pozitívum, hogy a frekventáltan PDF előállításra használt szoftverek nem kódolt formában tárolják e táblát, ezáltal csekély mértékben léphet fel olyan eset, amikor visszautasítja a rendszer a metaadatok hozzáadását. Ha nem képes kiolvasni az offset tábla tartalmát egy kivétel keletkezik, mely szabályos folyamatleállítást eredményez, azaz a komponens működése hibamentesen fejeződik be, ezzel nem kockáztatva a teljes rendszer üzemelését. A kivétel feldolgozását minden esetben a háttérkiszolgáló API-ja végzi, mely kapcsolódási pontot biztosít az Electra kiszolgáló komponenseihez. Az API-ról és a komponensek működéséről az Üzemeltetés és biztonság fejezetben adunk részletesebb leírást.

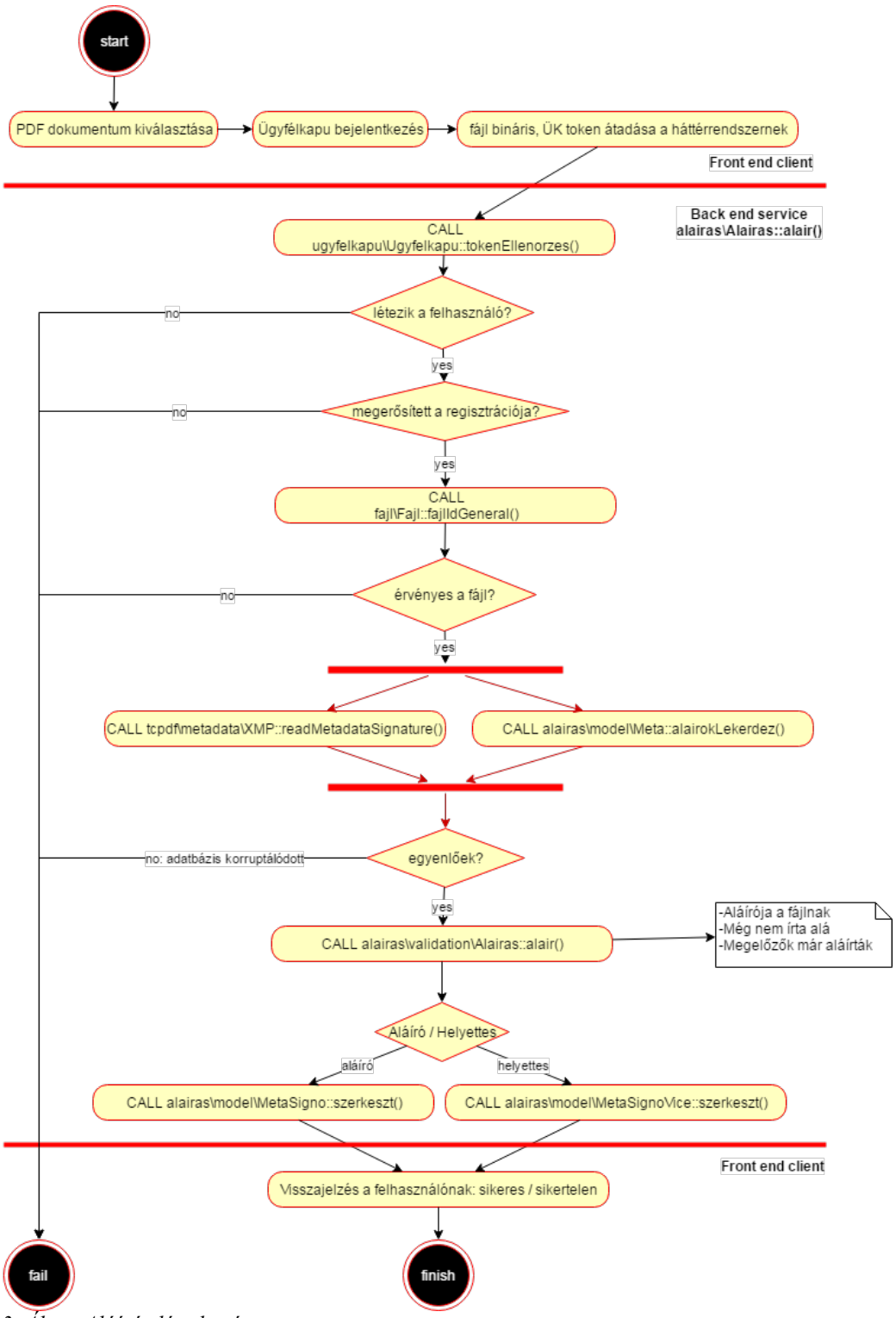
Az újrafelhasználhatóság jegyében létrehozott komponensek esetén nem az univerzális, mindent megoldó függvények megalkotása a cél. Robert C. Martin Clean Code könyvében az alábbi idézet ezt tökéletesen megfogalmazza: (Martin, R. C., 2009)

"*FUNCTIONS SHOULD DO ONE THING. THEY SHOULD DO IT WELL. THEY SHOULD DO IT ONLY.*" Azaz minden metódus egy feladatot végezzen, azt jól és csak azt.

Az alfejezetben ismertetett metaadat kezelésre szolgáló modul feladatra optimalizálást követően alkalmas bármilyen tartalmú XMP előállítására, valamint beolvasására. Az átalakítás elengedhetetlen, elsősorban a szofisztikus működés megteremtése, másodsorban pedig az említett univerzalitás beidegződésének kivédése végett. Nem lehet ma olyan komponenst implementálni, ami képes minden jövőbeni igényre felkészülni, pontosabban fogalmazva, nem is szabad.

2. Aláírás létrehozása

A szolgáltatás célja: létrehozni a dokumentum aláírásának tényét a háttéradatbázisban. A komponens, Ügyfélkapu által azonosított személy részére, lehetőséget biztosít olyan dokumentum aláírására, melyhez aláíróként hozzáadták Ügyfélkapu e-mail címét. Az ellenőrzés folyamatában adatokat szolgáltat a megjelenítő rendszer felé. Ezáltal kettős funkciót lát el: aláírás, ellenőrzés.



3. Ábra: Aláírás létrehozása

A 3. ábrán részleteiben megismerhetjük az aláírás rögzítésének lépéseit, melynek java részét ismét a back end kiszolgáló végzi. A kliens feladata csupán annyi, hogy beküldje a feltöltött fájl binárisát és az Ügyfélkapu által azonosított felhasználó token-jét.

Ezt követően az aláírás komponens *alair()* metódusa megkezdi az ellenőrzéseket, elsőként a legfontosabb lépés az Ügyfélkapu rendszerben ellenőrizni az aláírásra bejelentkezett felhasználó token-jét, lekérni a nevét, regisztrációjának kódját. Két esetet vizsgál a rendszer:

létezik-e a felhasználó, ha igen, véglegesített regisztrációval rendelkezik-e. Minden egyéb esetben megtagadja az aláírás létrehozását.

A második lépés során a beküldött fájl alapján létrehozza annak azonosítóját, mely az aláírók hozzárendelése alfejezetben ismertetésre került. Ezzel ellenőrzi, a beküldött fájl érvényes-e, azaz korábban aláírása jelölték az Electra rendszerben, ha nem érvényes, nem kerülhet aláírásra.

A következő lépésben az adatbázis illegális módosítása ellen tett óvintézkedés végett összeveti a beküldött fájlhoz rendelt aláírók és az adatbázisban a fájlhoz rendelt aláírók minden eleme egyezik-e. Amennyiben eltérés mutatkozik, feltételezhető az adatbázis manipulációja, az aláírás lehetőségét elutasítja.

Az aláírás tényét rögzítő utolsó lépésben aláíró vagy helyettes megjelölés függvényében a megfelelő helyen tárolja az aláírás időbélyegét, valamint az aláíró Ügyfélkapu rendszerben szereplő nevét.

Zárásként ismét a kliens kap szerepet, visszajelzést küld a felhasználó részére: sikeres vagy sikertelen az aláírás.

A következőkben, az adatbázisban tárolt attribútumokat szeretnénk ismertetni, amelyek ténylegesen rögzítik az aláírást, eltérően a tanúsítvány alapú megoldástól, ez esetben nem a fájl hordozza az aláírási információkat, hanem egy központi adattárban kerül letárolásra.

alairas komponens Meta modellhez kapcsolt adatbázis attribútumok

(A modell a beküldött fájlt írja le.)

1. fajl_id
    1. A fájlt egyedileg azonosítja, a *fajlIdGeneral()* függvény hozza létre.
2. feltolto_email
    1. Az aláírókat kijelölő és a fájlt aláírásra beküldő Ügyfélkapu felhasználói fiókkal rendelkező e-mail címe, mely az Ügyfélkapu azonosítás alapján kerül lekérdezésre a központi rendszerből.
3. feltolto_nev
    1. A 2. attribútumhoz tartozó, Ügyfélkapu rendszerben tárolt teljes név, mely szintén a központi rendszerből érkezik.
4. feltoltes_idobelyeg
    1. A későbbiekben bemutatásra kerülő módon meghatározott hiteles időbélyeg, mely igazolja, a fájl mikor lett beadva aláírásra.

alairas komponens MetaSigno modellhez kapcsolt adatbázis attribútumok
(A modell a beküldött fájlhoz rendelt aláírókat írja le.)

- fajl_id
  - A Meta rekord azonosítóra hivatkozik.
- email
  - Az aláíró Ügyfélkapu e-mail címe, mely egyedileg azonosítja a központi rendszerben.
- nev
  - Alapértelmezetten NULL, mely a fájl aláírásra beküldésekor inicializálódik. Az aláírás megtörténtekor, az aláíróhoz rendelt, Ügyfélkapu rendszerből származó teljes név.
- szint
  - Az aláíró sorrendiségét meghatározó egész szám, mely megadja az aláírási sorban ki után következik. Példa: szerződő felek 1. szint -> tanúk 2. szint -> ügyvéd 3. szint
- alairas_idobelyeg
  - Alapértelmezetten NULL, mely a fájl aláírásra beküldésekor inicializálódik. Az aláírás megtörténtekor hitelesen előállított időbélyeg kerül rögzítésre.

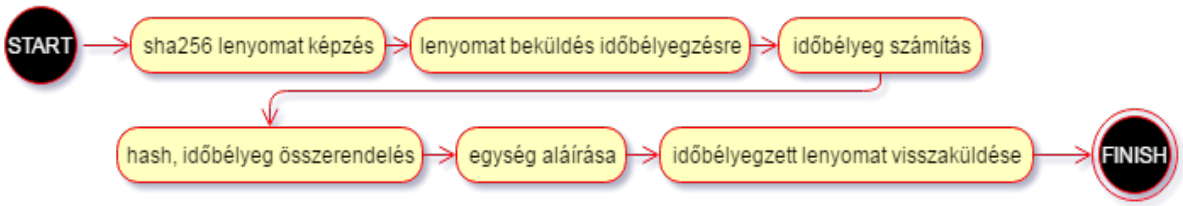
alairas komponens MetaSignoVice modellhez kapcsolt adatbázis attribútumok
(A modell az aláírókhoz rendelt helyetteseket írja le.)

1. fajl_id
   1. A Meta rekord azonosítóra hivatkozik.
2. email_helyett
   1. A MetaSigno azon aláíró személy rekordjára hivatkozik, amely az adott fájlhoz rendelt. Azaz a helyettes ki helyett fogja aláírni a megadott fájlt.
3. email
   1. Az aláíró Ügyfélkapu e-mail címe, mely egyedileg azonosítja a központi rendszerben.
4. nev
   1. Alapértelmezetten NULL, mely a fájl aláírásra beküldésekor inicializálódik. Az aláírás megtörténtekor, az aláíróhoz rendelt, Ügyfélkapu rendszerből származó teljes név.
5. alairas_idobelyeg
   1. Alapértelmezetten NULL, mely a fájl aláírásra beküldésekor inicializálódik. Az aláírás megtörténtekor hitelesen előállított időbélyeg kerül rögzítésre.

Az elektronikus aláírás letagadhatatlanságát időbélyeggel lehet biztosítani, mely hitelesített szolgáltatótól szerezhető be, általában elektronikus aláírással együtt alkalmazzák a dokumentum hosszú távú archiválásának biztosítására, valamint az aláíró tanúsítvány visszavonása esetén fellépő aláírás megkérdőjelezhetőségének elkerülése érdekében. Időbélyeg szolgáltatásra az NMHH által engedélyezett szolgáltató jogosult, melyet a hatóság folyamatosan ellenőriz a szabályos működés érdekében. Általánosságban a rendelkezésre álló megoldások együttesen, csomagban érhetők el egy-egy szolgáltatónál, például az elektronikus aláíró tanúsítvány és időbélyegzés egy helyről igénybe vehető. Két nagy szolgáltató van jelen Magyarországon: NetLock (Netlock. (b)) és Microsec e-Szignó (Microsec. (b)). Mindkét vállalat széles termékpalettát kínál az e-ügyintézés megvalósításához: SSL, e-aláíró, titkosító tanúsítványok és időbélyeg szolgáltatás.

Az elektronikus időbélyegzés során a dokumentumról egy sha-256 hash lenyomat készül, ez kerül átküldésre az időbélyeg szolgáltató felé. A kapott hash-t összerendelik egy pontos idővel és a kapott egységet a szolgáltató elektronikus aláírással ellátva visszaküldi az igénylő részére. A pontos idő előállítása egy megbízható forrásra épülve állítódik elő, UTC+időzóna formában, másodperc pontos megadással. A forrás lehet atomóra, GPS műhold, példa időbélyegre: 2015/04/01 14:26:29 +01'00'.

Egy e-aláírás hitelességét a hozzákapcsolt tanúsítvány érvényessége határozza meg, ez azt jelenti, ha a tanúsítványt bármilyen okból kifolyóan visszavonják, az aláírt dokumentumok hitelessége megkérdőjelezhető. Ennek elkerülésére szükséges időbélyeget is alkalmazni az aláírás mellett, ez bizonyítja, az adott időpontban a dokumentum létezett, az aláíró tanúsítvány érvényes volt. Egy nehézség, hogy az időbélyeg is tanúsítványra támaszkodik, érvényessége a hitelesítő tanúsítvány idejére korlátozódik. Emiatt minden lejárat előtt újra időbélyegezni szükséges a további érvényesség fenntartása végett. E folyamat hiteles archiválást nyújtó szolgáltatás igénybevételével elkerülhető, ez esetben az érvényesség fenntartását a szolgáltató garantálja. (Microsec. (a)) (Netlock. (a)) (NMHH)
A fent bemutatott folyamatot a 4. ábra szemlélteti.



4. Ábra: Hiteles időbélyeg létrehozás : szolgáltató

Az általunk fejlesztett rendszer, hasonlóan az időbélyeg szolgáltatóhoz, UTC időbélyeget alkalmaz, viszont azt nem külső szolgáltatótól, hanem helyben üzemeltetett NTP kiszolgálótól kéri be. Ezáltal a fent ismertetett folyamat jelentősen egyszerűsödik, csupán az időbélyeg elkérés lépése marad. Ahhoz, hogy hiteles időbélyeget tudjunk kiadni az aláíráshoz, szükséges egy NTP kiszolgáló üzemeltetése, mely hiteles NTP kiszolgálóktól kéri be a pontos időt és ehhez igazítva biztosítja a mindig pontos UTC időt.
A folyamat működése igen összetett, ezért a szakszerű, megbízható üzemeltetés miatt hozzáértő személynek kell végeznie az üzemeltetést. Erre az egyetem ISZK egysége biztosít erőforrást, a következőkben szeretnénk ismertetni magát az NTP architektúrát, illetve a kiszolgáló működési vázát.

Az NTP egy TCP/IP protokoll, mely lehetőséget ad központi időszerverekkel való kommunikációra a pontos idő lekérdezése céljából. A lekérdezett időadatokat felhasználva Linux operációs rendszer alatt lehetőség van szinkronizálni a gép óráját az UTC atomórák idejével. Ezáltal mindig pontos időbélyeg állítható elő, nem csak aláírás céljára, hanem például log fájlok bejegyzéséhez.

Az időkiszolgálók az alábbi hierarchikus rendszerben épülnek fel:

1. Stratum0: atomóra
    1. Stratum1: időszerverek, melyek az atomórákhoz kapcsolódnak
        1. Stratum2: a következő szint kiszolgálói, ezen a szinten olyan számítógépek kapcsolódnak hálózatba, melyek több Stratum1 időszervertől is kérnek pontos időt
            1. Stratum3...n: itt hasonlóan az előző szinthez összekapcsolódott számítógépek kérnek és szolgáltatnak időadatokat

A mi esetünkben az Ubuntu rendszerhez letölthető *ntpd* szoftver, egy NTP-démon, mely napi szinten folyamatosan szinkronizálja az időt a megadott NTP kiszolgálókkal, melyek az */etc/ntp.conf* fájlban adhatók meg. Számos megbízható NTP Stratum1 időszerver érhető el nyílt forrásban, ezáltal biztosított a pontos és hiteles időbélyegzés az Electra rendszerben. Ezen felül, ahogy korábban ismertettük, napi szinten naplózásra kerül, hogy a gépi óra mindig az UTC pontos időt szolgáltatja, így visszamenőlegesen is ellenőrizhető a rendszer.

Az időbélyeg tényleges létrehozása a PHP *time()* metódussal történik, mely a Unix Epoch (January 1 1970 00:00:00 GMT) óta eltelt másodpercek számával adja meg az UTC időt. Ezáltal az időbélyegre nem hat ki az időzónabeli eltérés, minden időzóna szerint megjeleníthető az időzóna szerinti értéke. Minden időzónában ugyanannyi az eltelt másodpercek száma, viszont a megjelenített dátum-idő konverzió Magyarországon és Ausztráliában nem ugyanazt mutatja, ezért lényeges, hogy ilyen formában rögzítsük az aláírás idejét.

A 32 bites rendszerek korában realizálódott egy nagyméretű probléma, mely a 64 bites processzor megjelenésével kiküszöbölésre került, ez nem más, mint az integer túlcsordulás. A probléma Unix Millennium Bug néven vált ismertté, a 03:14:07 UTC on 19 January 2038 időpontot követően 32 bites rendszerben túlcsordul a megjeleníthető integer szám, emiatt nem lehet jövőbeni időadatokat megadni. Ez a 64 bites rendszerben a maximálisan megadható 9999-12-31 dátumig működőképes, ezáltal megoldódott a 2038 probléma.

Az aláírási folyamat lezárásaként a kliens visszajelzést ad a felhasználó felé a folyamat sikeres vagy sikertelen mivoltáról. Ez egyben egy nyugta az aláírás megtörténtéről vagy meghiúsulásáról.

Kockázat szempontjából itt már több tényezőt is figyelembe kell venni, eltérően az előző szolgáltatástól.

Fontos a megbízható személy azonosítás az aláírás során, erre az Ügyfélkapu szolgáltatása nagy biztonsággal alkalmas. A folyamat kezdetekor a központi rendszer kéri a felhasználót, hogy azonosítsa magát, ezt követően az Electra rendszer megkapja a felhasználóhoz tartozó token-t, amely a művelet befejezéséig azonosítja. A feladat elvégzését követően átirányítjuk az Ügyfélkapu felületére, ahol lehetősége nyílik kijelentkezni. Fontos, hogy mi nem szakítjuk meg a session-t, több folyamatban lévő eseménye is lehet a miénk mellett, viszont jelezni kell részére, hogy aktív a bejelentkezése, erre legalkalmasabb, ha az Ügyfélkapu központi nyitólapjára navigáljuk. Annak érdekében, hogy ne kelljen ismét bekérni az Electra felületét a navigálás új lapon nyílik meg, így amennyiben tovább szeretné használni a mi rendszerünket, elegendő cselekvés nélkül bezárnia az Ügyfélkapu oldalt.

Másik fő szabály elektronikus aláírás tekintetében a későbbi azonosítás, erre is megfelelő alapot biztosít az Ügyfélkapu hitelesítés. A művelet során regisztrált név és e-mail cím tárolása nem csak aktuális, de jövőbeni visszakövethetőséget tesz lehetővé. A központi rendszerben regisztrált felhasználókhoz kapcsolt e-mail cím bármikor egyértelműen hozzákapcsolható egy személyhez, mivel kikötés, hogy az e-mail cím egyedi kell, legyen,

egyedileg azonosít egy adott felhasználót. Ez teljesíti a letagadhatatlanság fogalmát is, a megadott e-mail címhez köthetően csak egy adott személy írhat alá.

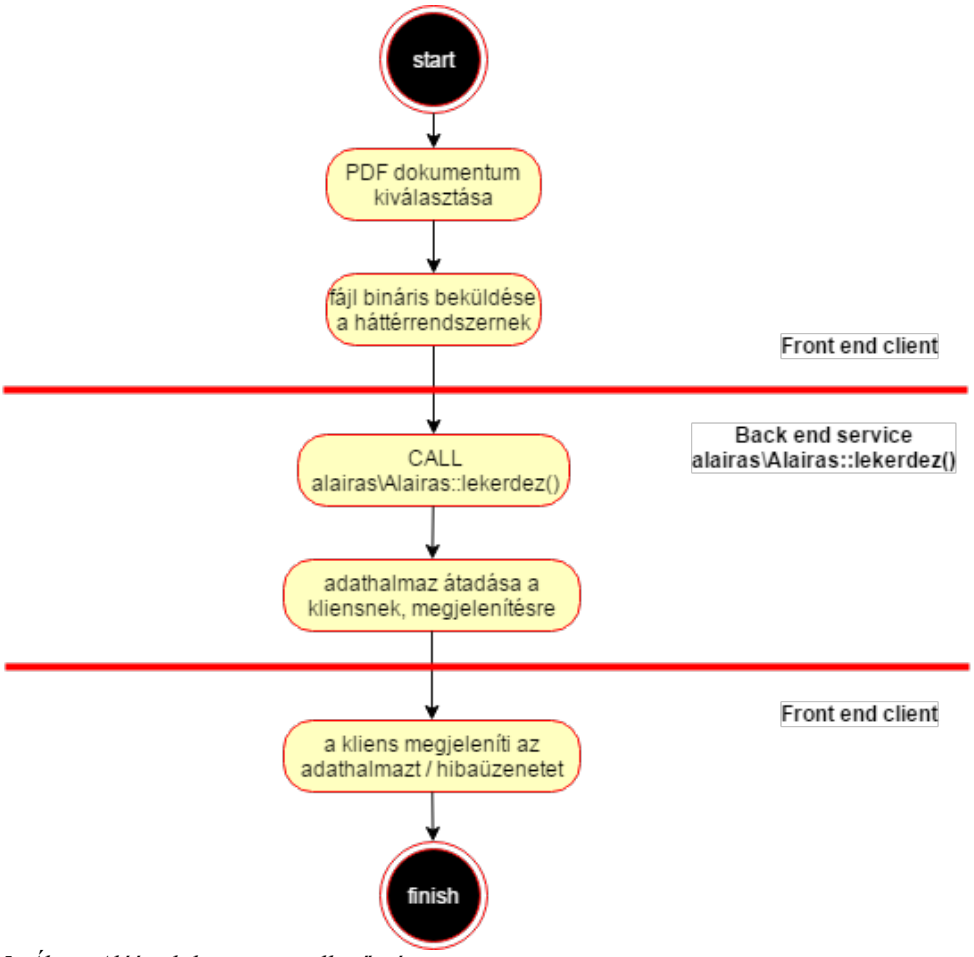
Más oldalról felmerül a manipuláció kockázata, bárki megszerezheti az e-mail/jelszó párost, bárki nevében írhatunk így alá. Ez minden rendszerben kockázat, természetesen ezt kivédeni nem lehet 100%-os eredménnyel. Ettől eltekintve vélelmezni kell, hogy az adott dokumentumot az adott személy írta alá. A jövőben a biztonság növelésére bevezethető például sms azonosítás egy második lépcsőként, ahogyan az a Google esetén már elérhető is. Ez növelheti a fiókkal való visszaélés elleni erőfeszítések hatékonyságát, de 100%-os megszüntetést így sem lehet elérni.

A jövőbeni felhasználhatóság tekintetében, hasonlóan az előző fejezetben bemutatott komponens alapú tervezés eredményeként itt is lehetőséget ad az egyes összetevők újrafelhasználására. Az Ügyfélkapu token ellenőrzését megvalósító komponens változtatás nélkül, teljes egészében használható erősen eltérő környezetben is, bárhol ahol Ügyfélkapura épülő személyazonosítás van használatban.

Ezen túlmenően az aláírás komponens is felhasznál meglévő komponensek által megvalósított funkciókat: metaadat kezelés, fájl azonosító generálás. Az első a korábban ismertetett TCPDF által nyújtott szolgáltatás, míg a második egy fájlkezelő modulban implementált metódus révén kerül integrálásra. Ezzel nem csak az aláírás eszköz használható újra, hanem maga ez is újrafelhasznált egységekből épül fel, részben.

3. Dokumentum ellenőrzése

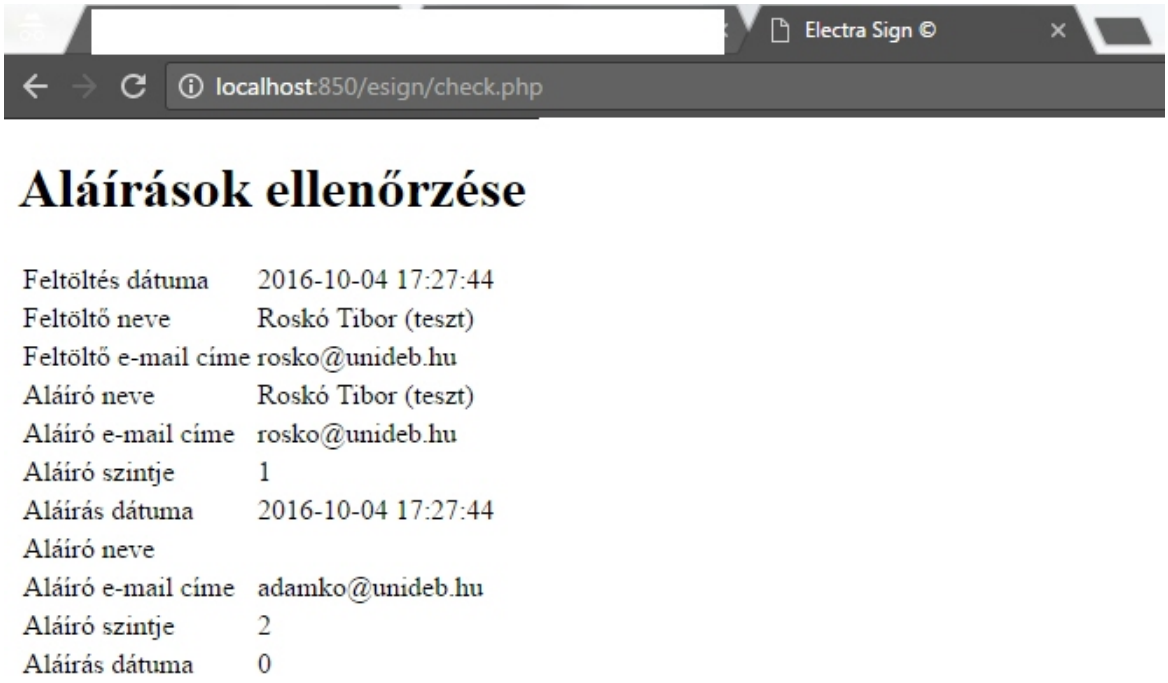
A szolgáltatás célja: lehetővé tenni az Electra rendszerben aláírt dokumentumok érvényességének ellenőrzését.



5. Ábra: Aláírt dokumentum ellenőrzése

Az ellenőrzés folyamata az 5. ábrán szemléltetett néhány eseménysorból tevődik össze, mint korábban megszokhattuk, itt is kliens-kiszolgáló architektúra mentén dolgozik a rendszer. Ennek köszönhető az újrafelhasználás lehetősége és a feladatok jó elkülönülése is

A művelet a kliens megszokott feladatával indul, beküldésre kerül a feltöltött PDF binárisa a kiszolgáló komponens felé. A tényleges működés az aláírás komponens *lekerdez()* metódusában történik. Elsőként elkészül a fájl azonosító a beküldött bináris alapján, mely azonosítja a fájlt, illetve ha megváltozott az aláírások alatt, a korábban ismertetett formában visszajelzésre kerül. Ezt követően lekérdezésre kerül a fájl alapvető metaadatait, aláírók és helyetteseik adatait tartalmazó információcsomag. Az alábbi 6. ábra a demó rendszerben végrehajtott ellenőrzés során készült:



6. Ábra: Dokumentum ellenőrzése: az aláírási információk megtekintése

A háttérrendszer ezt követően továbbítja az adatcsomagot a kliens felé, amely formázottan megjeleníti a releváns információkat.

Egy dokumentum akkor tekinthető hitelesnek, ha minden aláírója aláírta, ezt az éles rendszer külön ellenőrzést követően ki is jelzi majd. Az ellenőrzés során csupán azt kell megvizsgálni, hogy tartozik-e NULL időbélyegű aláíró a dokumentumhoz, amennyiben nem, a dokumentumot mindenki aláírta. Ezen ellenőrzést a kliens végzi el, a kapott, aláírók információit tartalmazó adatcsomagból.

Az aláírás időbélyegét rendkívül egyszerűen, időzónától függetlenül meg tudjuk jeleníteni, köszönhetően az integer időbélyeg tárolásnak. A PHP *date()* függvény két paramétert vár: formátum, időbélyeg, mely lehet opcionális, ha nincs megadva, az aktuális dátum-idő jelenik meg. A mi esetünkben ide kerül megadásra a tárolt időbélyeg, mely a kliens időzónájának megfelelően jelenik meg, ami a korábban említett okokból például Magyarországon és Ausztráliában eltérő időadatokat jelenít meg. Egy konkrét példán szeretnénk ezt illusztrálni:

Legyen a tárolt időbélyeg a következő: 1475683200.

Ez Magyarországon: October 05, 2016 18:00:00 időpontnak felel meg, GMT+01:00 időzónában

Ausztráliában pedig: October 06, 2016 03:00:00 időpontnak, GMT+11:00 időzónában.

Jól látható, hogy Ausztráliában már nem is az adott napon jelenik meg az időbélyeg, ezért fontos időzóna függetlenül eltárolni az időbélyeget.

Az újrafelhasználhatóság szemszögéből a már korábban is ismertetett lehetőségek merülnek fel e szolgáltatás során is, tekintettel arra, hogy a szolgáltatást biztosító metódus az aláírás komponens része. A fájl azonosítót előállító metódus szinte minden folyamatban megjelenik, mint újrafelhasznált egység. Ez jól mutatja, egy körültekintően implementált modul vagy metódus számtalan esetben felhasználható új modulok, komponensek építéséhez.

MySQL vs Neo4j

Mint ahogy a tanulmány bevezetőjében ismertettük, kutatásunk fő területe a szemantikus Web, ezáltal e projekt keretében nem csak a metaadatok felhasználási lehetőségét, de a gráf adatbázis alkalmazhatóságát is megvizsgáltuk.

A rendszer alapját MySQL háttéradatbázis adja, viszont készült egy Neo4j megoldást alkalmazó verzió is. Ezáltal lehetőség nyílik a két, alapjaiban is jól elkülönülő adatbázis hatékonyságának vizsgálatára, majd az eredmények összehasonlítására. A perdikció, hogy nagy hatékonyságbeli eltérésnek nem kell jelentkezni, tekintettel a szerény adatkapcsolatra. A rendszer adattárában mindössze egyetlen összekapcsolás jelenik meg: a fájl és a hozzá tartozó aláírók rekordjai között.

A MySQL rendszer egy relációs adatbázis, mely széles körben elterjedt, a legtöbb webes alkalmazás épül rá. Előnye az előre definiált sémában keresendő, melynek alkalmazása egy stabil alapot képez az adattároláshoz. Ezzel szemben a gráf adatbázis, a nevéből adódóan egy gráf struktúrában tárolja el az adatokat. Ez a szemantikus Web alapját képező RDF struktúrához hasonló adatkapcsolatokat állít elő, ezáltal jól illeszthető a szemantikus környezetbe. Fő alkalmazási területe a sok, újrafelhasznált adatok között felépülő kapcsolatokkal rendelkező információhalmazok tárolása. Egyetlen hátránya a sémamentesség lehet, mely nagyobb szabadságot adhat az inkonzisztencia megjelenésének. Ennek ellenére az előnye, hogy letisztultabb és jóval egyszerűbb formában felépíthető kapcsolatok létrehozását teszi lehetővé, maximálisan ellensúlyozza a sémamentességből fakadó hátrányt, mely alkalmazás szinten hatékonyan kezelhető. Fontos szem előtt tartani, hogy a komponens alapú fejlesztés területén nem hagyományos szemléletben kell egy adatbázisra tekinteni, hanem csak egy funkciómentes tárolóként. Erre azért van nagy szükség, mert a komponens újrafelhasználhatóságához hozzátartozik a függetlenség, amely leggyakrabban adatbázis felől jelentkezhet, ennek elkerülésére célszerű az adatbázist kizárólag tárolóként implementálni, azaz kerülni kell a trigger-ek és más beépített metódusok használatát, ezeket mind a komponensben kell megvalósítani.

Összegezve, a relációs és gráf adatbázis között jelentős különbség a tárolás módjában mutatkozik, mely mindkét megoldás előnyét jelenti. Ezen előny a feladat és környezet függvényében érvényesül, gráf adatbázis hatékonyan újrafelhasznált, többszörösen összekapcsolt információk tárolásában alkalmazható.

Felhasználás lehetőségei, jövőbeli értékteremtés

Elsődleges célunk megalkotni egy elektronikus aláírás biztonsági szintet teljesítő dokumentum hitelesítő szolgáltatást, mely alkalmas elsősorban e-szerződések megkötéséhez. A korábban ismertetett jogszabályok tükrében erre minden lehetőség adott, csupán két

sarkalatos pontnak kell megfelelni: aláíró személyének megdönthetetlen azonosítása, hiteles időbélyeg alkalmazása. (eIDAS 910/2014/EU)

Természetesen ennyire nem egyszerű megvalósítani az elképzeléseinket, illetve nem is feltétlen lenne kedvező hatása, ha bármilyen rendszerre kimondhatnánk, hogy holnaptól fogadjuk el, mint e-aláíró megoldás. Ez rendkívül felelőtlen döntés lenne, már csak biztonsági szempontok alapján is, ettől viszont még nem lehetetlen vállalkozás, csak teljesíteni kell a megadott követelményeket. Az elemzett jogi szabályozásokon túl meg kell felelni minőségi követelményeknek is, jelenleg EAL 4+ minősítettségű kártyák és rendszerek biztosítják az e-aláírás megfelelő védelmét. Ezt a minősítést egy audit folyamat során lehet megszerezni, melynek során egy 7 fokozatú skálából álló minősítési rendszerben, az elérni kívánt szinthez tartozó szigorú követelményeknek kell, megfeleljen a vizsgált szoftver.

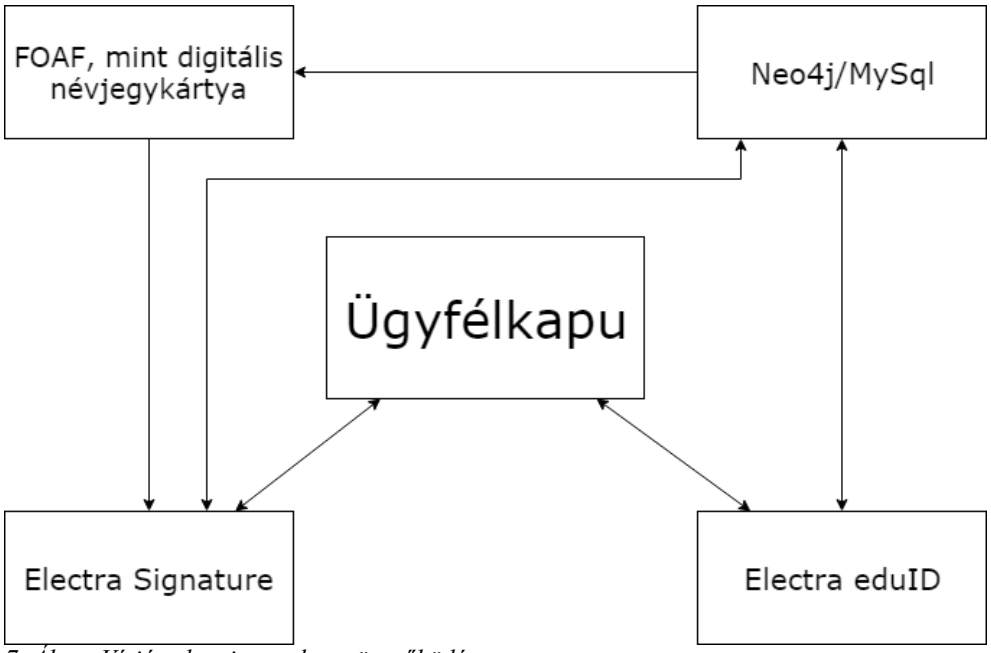
Az EAL szintjei:

1. EAL1: Funkcionálisan tesztelt
2. EAL2: Strukturálisan tesztelt
3. EAL3: Módszertanilag tesztelt és ellenőrzött
4. EAL4: Módszertanilag tervezett, tesztelt és auditált
    1. Az elkészített rendszer jól dokumentált, az ellenőrzés során a tesztelési dokumentáció szerint, a specifikációknak való megfeleltetés történt.
5. EAL5: Fél formális módszerrel tervezett és tesztelt
6. EAL6: Fél formális módon ellenőrzött tervezés és tesztelés
7. EAL7: Formálisan ellenőrzött tervezés és tesztelés

Rövid tájékozódást követően megtudtuk, hogy nem alapfeltétele egy EAL minősítés az elektronikus aláírást biztosító szoftvernek, tekintettel arra, hogy ezt jogszabály jelenleg nem is írja elő. Emiatt fordul elő az is, hogy nem minden piacon lévő megoldás rendelkezik tanúsítvánnyal. Ettől eltérően az esetleges szolgáltató vagy megrendelő saját döntése mellett lehet elvégeztetni egy audit folyamatot. Ez azért is lényeges számunkra, mert egy ilyen tesztelés költsége -megfelelő támogatók hiányában- nem tenné lehetővé a nonprofit szolgáltatás biztosítását.

Ezen szempontok mentén levonható egy olyan következtetés, mely szerint az általunk elkészített rendszer biztonsággal alkalmazható, például az egyetem e-ügyintézésének alapjaként.

Mint azt már a bevezetőkben is felvázoltuk, a szemantikus Web infrastruktúrák stabil lehetőséget biztosítanak különböző alkalmazások együttműködésére. Végső célunk a bemutatott különálló projektjeinkben implementált algoritmusok kooperatív alkalmazása, melynek folyamatát a 7. ábra szemlélteti.



7. Ábra: Vízió: algoritmusok együttműködése

A bázis infrastruktúra az Ügyfélkapu személy azonosító szolgáltatása, erre épül az Electra projekt mindkét megoldása: aláírás, oktatási azonosító. A rendszerek adatkiszolgálását MySQL és Neo4j adatbázisok együttesen, a feladatra optimalizáltan végzik. A FOAF struktúrára épülő információ megosztó algoritmusunk az adattárban elhelyezett információt továbbítja az aláíró szolgáltatásnak, gépi feldolgozható formában, szemantikus Web alapokon.

Felismerhető, hogy az egyes rendszerek szigetszerű, redundáns adattárolás helyett egy központi adatkiszolgálót használnak, illetve egymásnak információcsomagokat adnak át. A metodika leképezésével elérhető, például IoT rendszerek bekapcsolása közös hálózatokba, ezzel csökkenhet a szenzorok száma, és a tárolni szükséges adatmennyiség is.

ÖSSZEFOGLALÁS

A tanulmányban lehetőség nyílt megismerni az Electra rendszer részletes felépítése mellett a háttérben működő technológiákat, az Ügyfélkapu rendszerét és Adobe XMP architektúráját. Bízunk benne, hogy kellő részletességgel sikerült átadni az információkat a teljes kép áttekintéséhez.

Az Elektronikus aláírás című tanulmányunkban (Roskó, T., 2017) az elérhető e-ügyintézést támogató megoldásokat mutattuk be, egyfajta alternatívaként az általunk fejlesztett rendszer mellett, illetve bevezetőként jelen tanulmányunkhoz. Itt két elérhető szolgáltatást vizsgáltunk meg részleteiben: DigitDoc biometrikus aláírás és NISZ AVDH. Az első rendszer tanúsítvány alkalmazása nélkül biztosít lehetőséget e-aláírás létrehozására, viszont a hitelesség végett tanúsított időbélyeggel kell ellátni az aláírt dokumentumot, mely nem része az alapszolgáltatásnak. A második alternatíva az Ügyfélkapura épülő dokumentum hitelesítés, mely egy elektronikus dokumentum tartalmát hitelesíti egy központi -NISZ által biztosított- e-aláírással. Ezzel lehetőség nyílik egy papír alapon megkötött szerződés digitalizált változatának hitelesítésére, vagy egy megrendelés elektronikus úton történő hiteles benyújtására.

A tanulmány fő fejezete az Electra rendszer részletes ismertetését tartalmazza, mely több alfejezeten keresztül mutatja be a konkrét szolgáltatás működése mellett a rendszermagot

felépítő architektúrák használati eseteit, paramétereit: Ügyfélkapu, Adobe XMP, MySQL-Neo4j. Az utolsó fejezetben a jövőbeni felhasználás lehetőségeit, előnyeit vettük górcső alá.

Fontos hangsúlyozni, hogy egy ilyen volumenű fejlesztés során nem lehet figyelmen kívül hagyni az érvényben lévő törvényi előírásokat, ajánlásokat ahhoz, hogy megfelelő és biztonságos alkalmazást legyünk képesek implementálni. A projekt lezárultával, eredményként elkönyvelhetjük egy olyan megoldás megszületését, amely a megfelelő szabályozás mellett képes biztosítani, például az egyetem e-ügyintézés folyamatai során e-szerződések megkötését.

Bízunk benne, hogy tanulmányunk és a projekt keretében megvalósuló e-aláíró szolgáltatás, valamint terv szerint egy oktatási segédlet pozitív előrelépést jelent majd az elektronikus ügyintézés minél szélesebb körben való elterjedésében. Az oktatási segédlet révén a kiemelt figyelmet igénylő személyek mellett a teljes lakosság e-ügyintézés tudását, tudatosságát is fejleszthetjük, mely nagyban hozzájárulhat a mindennapi életvitel megkönnyítéséhez, különösen a bevezetőben említett, valamilyen korlát révén akadályozottak tekintetében.

Cikksorozatunk következő, egyben utolsó tanulmányából az e-ügyintézés, az Electra rendszer bevezetésének előkészítését, az üzemeltetés főbb mérföldköveit, illetve a különleges bánásmódot igénylő személyek segítését lehetővé tévő előnyöket ismerhetik meg.

IRODALOM

Adobe. Extensible Metadata Platform guide. (Letöltés: 2016.10.05.). (Web: http://www.adobe.com/products/xmp.html).

eIDAS 910/2014/EU rendelet. (Letöltés: 2016.10.05.). (Web: http://eur-lex.europa.eu/legal-content/HU/TXT/?uri=CELEX%3A32014R0910).

Gasiorowski-Denis, E.. (2012). Adobe Extensible Metadata Platform (XMP) becomes an ISO standard. (Letöltés: 2016.10.05.). (Web: http://www.iso.org/iso/home/news_index/news_archive/news.htm?refid=Ref1525).

Martin, R. C.. (2009). Clean Code. Prentice Hall.

Mező, F. és Psenáková, I. (2009). A képességfejlesztés lehetőségei az e-learning és m-learning révén. In: Ildikó Psenáková, Ferenc Mező, Ildikó Viczayová (szerk.): Teória a prax II. Nitra: Univerzita Konstantina Filozofa v Nitre. pp. 119-129.

Mező, F. és Psenáková, I. (2010). Measuring and development of cognitive abilities of Guilford's SoI-Theory by E- and M-learning. In: Zuzana Nagyová-Lehocká (szerk.): Collection of Psychological Studies. Nitra: Constantine the Philosopher University in Nitra, Faculty of Central European Studies. pp. 27-31.

Microsec. (a). e-Szignó Időbélyegzés. (Letöltés: 2016.10.05.). (Web: https://e-szigno.hu/tudasbazis/idobelyegzes.html).

Microsec. (b). e-Szignó tanúsítvány fajták. (Letöltés: 2016.10.05.). (Web: https://e-szigno.hu/hitelesites-szolgaltatas/tanusitvanyok/tanusitvany-fajtak.html).

Netlock. (a). Időbélyeg. (Letöltés: 2016.10.05.). (Web: https://www.netlock.hu/html/idobelyeg.html).

Netlock. (b). Minősített tanúsítvány. (Letöltés: 2016.10.05.). (Web: https://www.netlock.hu/html/minositett.html).

NMHH. Mi az időbélyeg?. (Letöltés: 2016.10.05.). (Web: http://nmhh.hu/cikk/3964/Mi_az_idobelyeg_Mi_az_idobelyegzesszolgaltato).

Roskó, T., Adamkó, A.. (2016). Ügyfélkapu azonosítás használata oktatási környezetben, tanulmány. Debreceni Egyetem. (Letöltés: 2016.10.05.). (Web: https://drive.google.com/file/d/0B2coYzWnBXE6ZTV2bTBKV2ZoUlE/view).

Roskó, T. (2017). Elektronikus aláírás. Debreceni Egyetem. Különleges Bánásmód 2017/2.
TechNote 0003. (2008). Metadata in PDF/A-1. PDF/a Competence Center. (Letöltés: 2016.10.05.). (Web: https://www.pdfa.org/wp-content/until2016_uploads/2011/08/tn0003_metadata_in_pdfa-1_2008-03-182.pdf).

FÜGGELÉK

Rövidítések feloldása

1. API: Application Programming Interface
2. DE ISZK: Debreceni Egyetem Informatikai Szolgáltató Központ
3. EAL: Evaluation Assurance Level
4. FOAF: Friend Of A Friend
5. GMT: Greenwich Mean Time
6. GPS: Global Positioning System
7. NISZ AVDH: Nemzeti Infokommunikációs Szolgáltató Zrt. Azonosításra Visszavezetett Dokumentum Hitelesítés
8. NMHH: Nemzeti Média- és Hírközlési Hatóság
9. NTP: Network Time Protocol
10. OO (P): Object Oriented (Programming)
11. PDF/a: Portable Document Format / archivable
12. PHP: Hypertext Preprocessor
13. RDF: Resource Description Framework
14. sha-256: Secure Hash Algorithm 2 (256 bites változat)
15. SSL: Secure Sockets Layer
16. SSO: Single Sign-On
17. target: szakrendszer kiválasztott szolgáltatását jelölő kulcsszó
18. TCP/IP: Transmission Control Protocol/Internet Protocol
19. token: Ügyfélkapu azonosítást követően a bejelentkezett felhasználó azonosítója
20. URL: Uniform Resource Locator
21. UTC: Coordinated Universal Time
22. XML: Extensible Markup Language
23. XMP: Extensible Metadata Platform
24. Xref: PDF offset tábla, mely az objektumok kezdőcímeit jegyzi
25. XSD: XML Schema Definition