ÚJ SZÉCHENYI TERV

Magyar Posta Zrt.

„Hibrid kézbesítési és konverziós rendszer”

Elektronikus Kormányzat Operatív Program 1.2.23.

Biztonságos Kézbesítési Szolgáltatás

Felhasználói kézikönyv II.

Nagyobb forgalmú felhasználóknak

2019. február 19.

V.2.1

Dokumentum adatok

Dokumentum címe	Felhasználói kézikönyv II.
Projekt neve	Biztonságos elektronikus kézbesítési szolgáltatás
Szerző	Szittner Károly
Felelős	
Cég	Magyar Posta Zrt.
Elektronikus állomány neve	BKSZ_Felhasználói kézikönyv_II_v2_1.pdf
Dokumentum verzió	2.1 kiadás
Státusz	Kiadásra

Jóváhagyó

Név	*Beosztás*	*Aláírás*	*Dátum*
Váradi András	projektvezető		

Dokumentum életrajz

Kiadás	*Dátum*	*Szerző*	*Leírás*
0.1	2013.06.27	Molnár Gábor	A dokumentum első verziója
0.2	2013.07.05	Molnár Gábor	Kiegészítések
1.0	2013.12.30	Molnár Gábor	Kiegészítések
1.1	2014.12.05	Szittner Károly	A végleges rendszer szolgáltatásaihoz aktualizálás
1.2	2017.05.20	Szittner Károly	Újraírás a tesztelések alapján
2.0	2017.10.31	Szittner Károly	Újraírt verzió véglegesítése
2.01	2017.11.29	Szittner Károly	sendMessageResponse üzenet pontosítása
2.1	2019.02.19	Váradi András	A dokumentum egyszerűsítése, valamint a 84/2012 (IV.21) Vhr. 2019. január 1-től hatályba lépett változásainak átvezetése

Kapcsolódó dokumentumok

Cím	*Tárgy*	*Elektronikus állomány neve*

A dokumentum célja

Ez a dokumentum a Magyar Posta Zrt. biztonságos kézbesítési szolgáltatása felhasználói szempontú leírását tartalmazza. Tartalmazza a leírás az alacsonyabb hitelességi követelmények kielégítésére

alkalmas kézbesítési szolgáltatás a Magyar Posta Zrt. által nyújtott szolgáltatáscsomagjának leírását is.

A dokumentum célja a nagyobb forgalmú felhasználók részére a Kézbesítési és biztonságos kézbesítési szolgáltatás használatához alapot nyújtson, rendszer működésének és kezelésének bemutatásával. Nem foglalkozik részleteiben a hibrid kézbesítési és konverziós rendszer más szolgáltatásaival, csak ahol közvetlenül érintkeznek. Nem tárgyalja a regisztrációs eljárás részleteit, és részletesen foglalkozik a webszerviz alapú küldéssel.

Tartalomjegyzék

A dokumentum célja 2

1 Bevezetés 7

2 Fogalmak, rövidítések 9

3 A rendszer használatának előfeltételei 12

3.1 A küldemények küldésének és elérésének csatornái 12

3.1.1 Webszervizen keresztüli üzenettovábbítás 12

3.1.2 Hivatali kapun keresztüli üzenettovábbítás 12

3.2 Elektronikus levelezési cím 12

3.3 SMS képes telefonszám 13

3.4 Elektronikusan előzetesen azonosított felhasználók 13

3.4.1 Webszervizen keresztüli hozzáférés esetén 13

3.4.2 Hivatali kapun keresztüli hozzáférés esetén 13

3.5 Elektronikus aláírás 14

3.6 Regisztráció 14

3.6.1 Gépi kapcsolatok regisztrálása 14

3.6.1.1 A webszerviz regisztrálása 15

3.6.1.2 Hivatali kapu regisztrálása 20

3.7 Financiális feltételek 20

3.8 A kommunikációs csatornák sajátosságai 21

3.8.1 Webszervizen keresztüli kommunikáció 21

3.8.2 Hivatali kapun keresztüli kommunikáció 22

4 A kézbesítési/ biztonságos kézbesítési szolgáltatások alapelemei 24

4.1 A szolgáltatások kommunikációs modellje 24

4.1.1 A kézbesítési szolgáltatás kommunikációs modellje 24

4.1.2 A biztonságos kézbesítési szolgáltatás kommunikációs modellje 26

4.2 A rendszer által készített igazolások, tanúsítványok 28

4.2.1 Feladási igazolás 29

4.2.2 Kézbesítési igazolás 31

4.2.3 Átvételi elismervény 32

4.2.4 Letöltési igazolás 34

4.2.4.1 Sikeres letöltés igazolása 34

4.2.4.2 Letöltés elmaradásának igazolása 36

4.3 Küldemények küldése, fogadása, megtekintése, ellenőrzése 38

4.3.1 Elektronikus küldemény feladásának folyamata 38

4.3.1.1 Elektronikus dokumentum küldése a webAutomata használatával 38
4.3.1.2 Elektronikus dokumentum küldése hivatali kapun keresztül 43
4.3.2 Az igazolások keresése és megismerése a küldő oldalán 44
4.3.2.1 A küldő számára biztosított igazolások kezelése webAutomata használatával 45
4.3.2.2 A küldő számára biztosított igazolások kezelése hivatali kapu használatával 53
4.3.3 Kommunikáció a fogadó oldalon 53
4.3.3.1 A küldemény, illetve az értesítések címzett általi átvétele a webAutomata használatával 53
4.3.3.2 Hivatali kapu használata címzetti oldalon 62
5 A webszerviz funkciói és használatuk 63
5.1 Kommunikáció a webAutomatával 64
5.2 A webAutomata beállítását segítő eljárások 65
5.2.1 A szolgáltatás paramétereinek lekérdezése getClientConfiguration 65
5.2.2 Válasz a szolgáltatás paramétereinek lekérdezésére getClientConfigurationResponse 67
5.2.3 A szolgáltatás tesztelése: probe 68
5.2.4 Válasz a probe parancsra: probeResponse 69
5.3 A küldemények küldését és fogadását szolgáló eljárások 70
5.3.1 Üzenet továbbítása a sendMessage parancs használatával 70
5.3.2 Válasz az üzenet továbbítására: sendMessageResponse 74
5.3.3 Üzenet lekérdezése: getMessage 76
5.3.4 Válasz az üzenet lekérdezésére: getMessageResponse 78
5.3.5 Az üzenet fogadásának visszaigazolása: releaseMessage 81
5.3.6 Válasz az üzenet fogadásának visszaigazolására: releaseMessageResponse 84
5.4 Az átvételi elismervény kezelésének eljárásai 84
5.4.1 Átvételi elismervény kérése getAcceptanceCertificate 85
5.4.2 Válasz az átvételi elismervény kérésére getAcceptanceCertificateResponse 87
5.4.3 Aláírt átvételi elismervény feltöltése sendSignedAcceptanceCertificate 88
5.4.4 Válasz az aláírt átvételi elismervény feltöltésére sendSignedAcceptanceCertificateResponse 90
5.4.5 Visszautasítási elismervény kérése: getNonAcceptanceCertificate 91
5.4.6 Válasz a visszautasítási elismervény kérésére: getNonAcceptanceCertificateResponse 92
5.4.7 Aláírt visszautasítási elismervény feltöltése: sendSignedNonAcceptanceCertificate 93
5.4.8 Válasz az aláírt visszautasítási elismervény feltöltésére: sendSignedNonAcceptanceCertificateResponse 95
5.5 A kivételek kezelése 96
1. sz. függelék: A webszerviz XML alapú leírása webAutomata.wsdl 97
2. sz. függelék: A webAutomata websecurity leírása 104
2.1 A leírásban használt fogalmak magyarázata 104

3. sz. függelék: Az átvételi elismervény sémája 106

4. sz. függelék: A kötött elemek listája és értelmezésük.....109

4.1 deliveryType 109
4.2 messageType.....109
4.3 signatureType 110
4.4 IdentificationMethodType 111

5. sz. függelék: A webAutomata kliens (WebAPI) bemutatása.....112

1 Bevezetés

A biztonságos kézbesítési szolgáltatás az elektronikus kézbesítési szolgáltatások magasabb hitelességi és bizonyítási követelményeket kielégítő, hivatalos küldemények bizonyító erejű továbbítására, a feladás és átvétel körülményeinek bizonyítására alkalmas, illetve a továbbított küldemény tartalmának bizonyítását lehetővé tevő megoldása.

A megvalósított biztonságos kézbesítési szolgáltatás fontosabb feladatai az elektronikus ügyintézés és bizalmi szolgáltatások általános szabályairól szóló 2015. évi CCXXII. törvény (a továbbiakban Eüsztv.) szerint következők:

- elvégzi a feladó illetve a címzett azonosítását minden egyes tevékenységnél
- a feladónak megfelelő bizonyító erejű (elektronikus aláírással, illetve bélyegzővel ellátott) igazolást küld a feladott küldeményről annak alapvető jellemzőivel,
- a feladót és az azonosítás szintjét jelzi a címzettnek az átvételi elismervény részeként
- a küldeményt csak az átvételi elismervény aláírása után bocsátja a címzett rendelkezésére,
- az átvételi elismervény aláírása alapján ellenőrzi az aláíró átvételi jogosultságát, nem megfelelő aláírás esetén ismételten felhív az átvételre
- megfelelő bizonyító erejű átadási igazolást juttat el a feladónak az átvételi elismervény elkészülte és ellenőrzése után, amely tartalmazza az aláírt átvételi elismervényt
- tárolja és hozzáférést biztosít a kapott üzentekhez és az elkészített bizonyítékokhoz, mind a feladó, mind a címzett számára.

A kézbesítési szolgáltatás esetében a fentiekhez képest a következő egyszerűsítés érvényesül:

- az aláírt átvételi elismervény helyett a szolgáltató által aláírt kézbesítési igazolást juttat el a feladónak, amely csak a küldemény rendelkezésre bocsátását, annak körülményeit igazolja,
- nincs aláírt átvételi elismervény.

Fontos szem előtt tartani, hogy a biztonságos kézbesítési szolgáltatás esetében nem postai (elektronikus) szolgáltatás lát el eddig postai szolgáltatásként megismert feladatot. A jelen leírásban – a könnyebb érthetőség kedvéért – az elektronikus folyamatok útján létrejött dokumentumokra, igazolásokra esetenként a postai szolgáltatások megfelelő megnevezéseit alkalmazzuk. Ez a funkcionális hasonlóság, vagy akár azonosság miatt lehetséges és célravezető az elektronikus szolgáltatások esetére is. Ugyanakkor a postai fogalmak (pl. tértivevény, feladóvevény) használata esetén is azokon a rendszerben minden esetben a létrejövő **elektronikus igazolások**at (hiteles elektronikus dokumentumokat) kell érteni.

A jelen dokumentum tervezett felhasználói azon, a szolgáltatást igénybe vevő szervezetek, amelyek a küldemények nagyobb számára tekintettel gépi csatlakozáson, web szervizen keresztül kívánják a szolgáltatást használni.

A szolgáltatás jelenleg a Magyar Posta Hibrid és Inverz hibrid konverziós szolgáltatásához kapcsolódóan érhető el. Ennek megfelelően Hibrid konverzió esetén a küldemény címzettje az a

speciális „postafiók”, amely a konverziós szolgáltatás bemenete. Inverz hibrid konverzió esetén a feladó maga a konverziós szolgáltatás, a címzett pedig az Inverz hibrid szolgáltatást igénybe vevő szervezet.

2 Fogalmak, rövidítések

A fejezet ABC sorrendben, röviden bemutatja a legfontosabb fogalmakat annak érdekében, hogy segítsen a folyamatok megértésében. A folyamatok, ahol indokolt, részletesebben is bemutatásra, értelmezésre kerülnek, azonban itt a kereszthivatkozások megértéséhez remélhetően elégséges információ érhető el.

Átvételi elismervény (Acceptance Receipt vagy Acceptance Certificate): A címzett által elektronikusan aláírt (bélyegzővel ellátott), a küldemény fogadását elismerő PDF formátumú dokumentum, melyet a biztonságos kézbesítési szolgáltató állít elő és a címzett rendelkezésére bocsát. A címzett az igazolást aláírva visszaküldi a szolgáltatónak, aki azt beépíti (beágyazza) csatolt állományként a szolgáltató minősített bélyegzőjével és időbélyegzővel hitelesített **Letöltési igazolás**ba és a feladó rendelkezésére bocsátja. Az Átvételi elismervény aláírása a feltétele a tértivevényes (BKSZ) küldemény a címzett számára történő hozzáférhetővé tételének.

Azonosításra Visszavezetett Dokumentumhitelesítés (AVDH): olyan KEÜSZ, amelynek keretében a jogszabályban kijelölt központi szolgáltató az ügyfél által rendelkezésre bocsátott dokumentumot az általa igazolt személyhez rendeli, majd a személyhez rendelésről kiállított igazolást elektronikus dokumentumba vagy az elektronikus dokumentumhoz kapcsolt záradékba foglalja, és azt - a hitelesítendő nyilatkozattal együtt - minősített vagy minősített tanúsítványon alapuló fokozott biztonságú elektronikus bélyegzővel, valamint minősített időbélyegzővel hitelesíti.

Biztonságos Kézbesítési Szolgáltatás (BKSZ): A szolgáltatás a Hivatalos irat tértivevényes levél kézbesítésének megfelelő zártságú elektronikus kézbesítési szolgáltatás (a vonatkozó követelmények összefoglalása a bevezetésben). Ezen szolgáltatás során a Feladóvevény mellett a címzett értesítése és az átvétel elektronikus aláírással történő igazolása után a címzett számára letölthetővé teszi a küldött dokumentumot, és a feladó számára a címzett által aláírt átvételi igazolást biztosítja (Tértivevény). Az átvétel megtagadható, illetve az értesítéstől számított 5 munkanap elteltével újabb értesítés történik, amely után 5 munkanappal a hagyományos esettel azonos („címzett nem kereste”) igazolás kerül kiállításra.

Contract (szerződés): Az azonos feltételek mellett biztosítandó szolgáltatásra vonatkozó megállapodás. Egy természetes vagy jogi személynek több szerződése is lehet, és kivételesen az is előfordulhat, hogy egy szerződés mögött több természetes vagy jogi személy áll.

Elektronikus kézbesítési szolgáltatás: A kézbesítési és biztonságos kézbesítési szolgáltatás együttes megnevezéseként használt fogalom.

Elektronikus ügyintézési törvény (Eüsztv): Az elektronikus ügyintézés és bizalmi szolgáltatások általános szabályairól szóló 2015. évi CCXXII. törvény.

Entitás: Elektronikus ügyintézést biztosító szervezet (Eüsztv. 1. § 17. pont) és a gazdálkodó szervezet (Eüsztv. 1. § 23. pont), mint logikai kategóriák együttes jelölése (minden olyan szervezeti forma együttese, amelyhez természetes személy képviselők kapcsolódnak. Amennyiben egy Szerződés (Contract) Entitáshoz kapcsolódik, akkor ennek a Szerződésnek a tranzakciói a Hatóságok

és szervezetek kezelőfelületén érhetők el. Entitáshoz nem kötött Szerződésekhez a Magánszemélyek kezelőfelületén lehet hozzáférni.

Értesítés (Notification): A rendszer a feladó számára különböző figyelmeztetéseket küld az előzetesen beregisztrált e-mail címre, illetve a szolgáltatás külön megfizetése esetén SMS üzenetben is. A címzett szintén értesítés útján kap tájékoztatást arról, hogy számára a biztonságos kézbesítési szolgáltatás használatával átvehető elektronikus üzenet érkezett.

Feladási igazolás (Dispatch Receipt vagy Dispatch Certificate): A biztonságos kézbesítési szolgáltató által készített és elektronikusan aláírt (bélyegzővel ellátott), PDF formátumú, beágyazott XML állományt tartalmazó igazolás a feladó által küldött üzenet a szolgáltató általi átvételéről. Az igazolást a rendszer elektronikus üzenet formájában visszaküldi a feladónak. A feladás folyamat csak a feladási igazolás átvételével fejeződik be, addig a küldemény nem számít feladottnak.

Hibrid küldemény: olyan küldemény mely elektronikusan, biztonságos kézbesítési szolgáltatás vagy kézbesítési szolgáltatás igénybe vételével kerül feladásra, de hagyományos postai úton kerül kézbesítésre. Ennek lehetővé tétele érdekében a Magyar Posta, mint kijelölt KEÜSZ szolgáltató hiteles átalakítást végez, az elektronikus dokumentumot papíralapú dokumentummá alakítja át.

Kézbesítési igazolás (Delivery Receipt vagy Delivery Certificate): a kézbesítési szolgáltató által készített és elektronikusan aláírt (bélyegzővel ellátott), PDF formátumú, beágyazott XML állományt tartalmazó igazolás a feladó által küldött üzenet elhelyezéséről a címzett postafiókjába. Az igazolást a rendszer elektronikus üzenet formájában visszaküldi a feladónak.

Kézbesítési Szolgáltatás (KSZ): A szolgáltatás a postai gyakorlat szerinti ajánlott (könyvelt) küldemények kézbesítésének megfelelő elektronikus szolgáltatás. Az igazolások a szolgáltató által a befogadásról (Feladóvevény – feladási igazolás), illetve a címzett kézbesítési tárhelyére történő elhelyezéséről (Kézbesítési igazolás) készülnek.

Központi Azonosítási Ügynök (KAÜ): Az Eüsztv 38. § (1) bekezdés j) pontjában központi biztosításra kijelöl szolgáltatás, mely keretében a szolgáltató az elektronikus ügyintézést biztosító szerv számára elérhetővé teszi a vele együttműködő azonosítási szolgáltatásokat, ideértve *az* azonosítási mód az azonosítandó személy általi megválasztásának lehetővé tételét és az így kiválasztott azonosítási szolgáltatónál az azonosítás végrehajtását, illetve az információ közlését az azonosítást kérő szervvel.

Központi Elektronikus Ügyintézési Szolgáltatások (KEÜSZ): Az Eüsztv. VII. fejezetében leírt szolgáltatások, melyeket az elektronikus ügyintézés hozzáférhetősége érdekében a kormány biztosít jogszabályban kijelölt szolgáltató útján.

Küldemény: A KSZ és BKSZ alapvetően komplett dokumentumok hiteles továbbítását biztosító elsődlegesen gép-gép közötti szolgáltatás, ennek megfelelően csak speciális formában képes a csatolt dokumentumok mellett külön üzeneteket vagy egyéb leíró adatokat továbbítani.

Letöltési igazolás (Download Receipt vagy Download Certificate): a biztonságos kézbesítési szolgáltató által készített és elektronikusan aláírt (bélyegzővel ellátott), PDF formátumú, beágyazott

XML állományt és sikeres kézbesítés esetén az aláírt átvételi elismervényt is beágyazva tartalmazó igazolás, mely tanúsítja, hogy a címzett aláírta az **átvételi elismervényt** és a küldemény a szolgáltató a címzett rendelkezésére bocsátotta.

SMTP: Elektronikus levelek továbbításának kommunikációs protokollja (az RFC 5321 írja le)

Szabályozott elektronikus ügyintézési szolgáltatások (SZEÜSZ): Az elektronikus ügyintézés és bizalmi szolgáltatások általános szabályairól szóló 2015. évi CCXXII. törvény VI. fejezetében szereplő olyan szolgáltatások, amelyek felett az állam közvetlen felügyeletet gyakorol, és ezen keresztül felelősséget vállal az elektronikus ügyintézés biztonsága érdekében

Szerződés (Contract): A hibrid kézbesítési és konverziós rendszerben létrehozott egység (Contract), amely min. 4 számjegyű azonosítóval rendelkezik. A szerződés a tulajdonságai, beállítandó paraméterei a Szerződés Kezelés (Contract Management) felületen érhetők el. A szerződés (Contract) nem azonos a papír alapú szerződéssel, amely a szolgáltatás igénybevételének szerződéses feltételeit jogilag megalapozza. A Szerződés (Contract) a Hibrid kézbesítési és konverziós rendszerben a szerződés dokumentumnak az üzenetek technikai kezelése szempontjából lényeges elemeit rögzíti, és a rendszer ezek figyelembe vételével és a szerződésazonosító használatával biztosítja a szolgáltatásokat

Szerződés címe: Egy szerződéshez egy vagy több cím rendelhető. Ezek a címek használhatók a biztonságos kézbesítési szolgáltatás során, az átvételre alapesetben jogosult címzett megjelölésére.

Szerződés felhasználója: Elméletileg lehetséges felhasználó(k) rögzítése egy Contract-hoz. A rendszer jelenlegi használati esetében természetes személy felhasználó nem rendelhető, mivel ilyen felhasználói hozzáférés nem biztosított. .

Szerződés aláírója: Ez a mező a rendszerben a küldemények átvételre használt aláírások, illetve bélyegzők lenyomatait hordozza. A csatlakozó rendszer a küldemények átvételi igazolásának aláírására használja. Ezt az aláírást a rendszer érvényesség szempontjából ellenőrzi.

Tértivevény: a tértivevény elnevezés használatát kerülni igyekszünk, mert keveredést okozhat a hagyományos postai levelezésben használatos formanyomtatvánnyal. A dokumentumban csak a tértivevényes küldemény fogalmat használjuk, amelynél a feladó igénye alapján a címzett csak akkor kaphatja meg a küldeményt, ha aláírta az átvételi elismervényt. Az átvételről (át nem vételről) és annak körülményeiről a feladó **letöltési igazolás** formájában értesül.

Ügyfélkapu (ÜK): A kormány által kötelezően biztosított elektronikus azonosítási szolgáltatás. (KEÜSZ)

3 A rendszer használatának előfeltételei

3.1 A küldemények küldésének és elérésének csatornái

A KSZ és BKSZ küldeményeinek és üzeneteinek fogadásához a továbbítandó küldemények mennyiségtől függően különböző megoldások használata indokolt. (az ügyfél választhatja meg, hogy melyikre tud felkészülni). Természetes követelmény itt is, hogy a kapcsolódó rendszer is rendelkezzen teszt környezettel is, ahogy ez a BKSZ esetében is biztosított, azaz lényegében a fenti követelményeket kétszer szükséges teljesíteni, a teszt környezetnél kisebb egyszerűsítésekkel.

3.1.1 Webszervizen keresztüli üzenettovábbítás

A hibrid kézbesítési és konverziós rendszer részeként működő KSZ és BKSZ szolgáltatások esetében szükséges kommunikáció lebonyolításához a rendszer egy webszervizt biztosít, melynek megszólításához szükséges program kialakítása az igénybe vevő feladata. A szolgáltató – elsősorban demonstrációs célból, de nem zárva ki annak rendszerszerű használatát sem – egy java nyelven fejlesztett klienst is biztosít a webszervizzel való kommunikáció céljára. Ennek bemutatása az 5. sz. függelékben található meg (a részletes leírás külön dokumentumban érhető el). A webszerviz funkcióinak leírása az **Hiba! A hivatkozási forrás nem található.**. fejezetben olvasható.

3.1.2 Hivatali kapun keresztüli üzenettovábbítás

Jelenleg elsődlegesen a konverziós szolgáltatások kommunikációjának biztosítására van lehetőség, de távlatilag a két biztonságos kézbesítési szolgáltatás között bármely szolgáltatást lehetővé tevő kapcsolatot szeretnénk megvalósítani, hogy a Magyar Posta által üzemeltetett BKSZ és KSZ korlátozás nélkül képes legyen a NISZ által működtetett biztonságos kézbesítési rendszerből küldemények fogadására és oda küldésére.

3.2 Elektronikus levelezési cím

A BKSZ és a KSZ használatakor, mind a feladó, mind a címzett a küldemények feldolgozásának különböző fázisaiban értesítéseket kaphat. Ezek közül a kiemelendő a küldemény érkezéséről szóló értesítés. Ez indítja a küldemény fogadás folyamatát a biztonságos kézbesítési szolgáltatás és a kézbesítési szolgáltatás esetén is. De érkezhetnek értesítések a feldolgozási folyamat esetleges hibáiról, meghiúsulásáról is

Ezen kommunikáció kezeléséhez az Igénybe vevőnek a szerződés megkötésekor meg kell adnia egy (és csak egy, de értelemszerűen ez lehet csoportcím is) érvényes e-mail címet. E cím megadása nélkül is működik a rendszer, de ebben az esetben nem teljes értékű a kommunikáció, például a címzett nem kap értesítést a küldemény érkezéséről. Gépi kommunikáció esetén ez nem feltétlenül kritikus, hiszen ott a rendszer amúgy is rendszeresen bekérdez, hogy van-e számára küldemény és a kérdésre válaszként megkapja a kezelendő üzenet típusát is.

3.3 SMS képes telefonszám

SMS képes telefonszám megadása nem kötelező, ez egy másodlagos, kényelmi szolgáltatás, amely az e-mailben is megkapott értesítéseket továbbítja az Igénybevevő számára. Az SMS szolgáltatás díja a megfizetett díj részét képezi.

3.4 Elektronikusan előzetesen azonosított felhasználók

A KSZ és BKSZ gépi csatlakozással történő használata nem tartalmaz természetes személy felhasználókat.

3.4.1 Webszervizen keresztüli hozzáférés esetén

Webszervizen keresztüli hozzáférése esetén az hibrid rendszer magát a kapcsolatot tartó rendszert azonosítja, ehhez a WS security szabvány szerinti aszimmetrikus kulcsú azonosítást és titkosítást használ. A biztonsági modellt részletesen leíró szabványos XML állományt a 2. számú függelék tartalmazza.

A kapcsolat létrehozásához előzetesen az adott szerződéshez be kell regisztrálni a kommunikációhoz használt tanúsítványokat. A WS security szabvány használatával megvalósított kapcsolat biztosítja, hogy kommunikáció csak megfelelően azonosított partnerek között védett tartalommal jöhessen létre. Így a kapcsolat hitelessége és a kommunikáció sértetlensége is biztosított. A címzett felelőssége, hogy a kapcsolat azonosítási megoldása mögé a kapcsolódó kliens (amely a küldemények küldését és fogadását esetleg nem is egy alkalmazásban biztosítja) rendszer oldalán olyan felhasználói azonosítást és naplózást valósítson meg, amely a küldés és átvétel felhasználói oldalán a számára megfelelő kezelést (azonosítást) igazolni tudja. A hibrid kézbesítési és konverziós rendszer oldalán a kommunikáció megfelelő naplózása biztosított és a küldeményeket tételesen, a küldő szolgáltatás a fentieknek megfelelő magas megbízhatóságú azonosításával, a tartalom azonosítását lehetővé tevő lenyomattal igazolja vissza. Értelemszerűen a hitelesség forrása itt a rendszer szempontjából az aláírás, a kliens oldali mögöttes eljárásokra nem terjed ki a rendszer kontrollja.

3.4.2 Hivatali kapun keresztüli hozzáférés esetén

A rendszer jelenleg a hibrid konverzióra küldött, illetve inverz hibrid konverzió nyomán keletkezett küldemények fogadását illetve továbbítását biztosítja. Tervezett a szolgáltatás általánosítása, azaz tisztán elektronikus célú kommunikáció támogatásának megvalósítása is.

Ebben az esetben a hivatali kapuval rendelkező intézmény azonosítása történik meg, a rendszer arra épít, hogy a hivatali kapuk biztosítása megfelelő regisztrációs eljárás alapján, elégséges hitelességgel történik. Ennek megfelelően a hivatali kapun keresztüli kapcsolatot létesíteni kívánó szervezetnek az általános szabályok szerint magának kell gondoskodnia arról, hogy a hivatali kapu hozzáférése az arra vonatkozó szabályok szerint történjen. A hivatali kapu hozzáférésére vonatkozóan az ott érvényes hozzáférési szabályok érvényesülnek, és ezek elvileg megfelelnek a biztonságos kézbesítés követelményeinek. Mivel a hivatali kapun keresztüli kommunikáció önmagában is naplózott és

biztosít igazolást a kommunikációról, itt az ellenőrzés kétszeres. A BKSZ ebben az esetben csak a hivatali kaput azonosítja a megfelelő webszerviz meghívásával. Itt is fel kell hívni azonban a figyelmet arra az alapelvre, hogy a biztonságos kézbesítési szolgáltatás esetében csak az a küldemény tekinthető feladottnak, amelyet a rendszer már saját maga visszaigazolt.

A már létező hivatali kapun keresztüli kapcsolat létesítéséhez elégséges a hivatali kapu azonosítójának beregisztrálása a szerződés megkötésekor. Ha valamely szervezet még nem rendelkezik hivatali kapuval, abban az esetben először a hivatali kapu a rendszertől független csatlakozási folyamatát kell megvalósítania.

3.5 Elektronikus aláírás

A BKSZ küldemények átvételéhez vagy elektronikus ügyintézéshez alkalmas elektronikus aláírás, illetve elektronikus bélyegző vagy az AVDH KEÜSZ használata szükséges. Az elektronikus ügyintézésre alkalmas aláírás követelményeit a 137/2016 (VI.13.) Korm. rendelettel és az eIDAS követelményekkel összhangban kell értelmezni. A lényege az, hogy az aláírásnak, illetve bélyegzőnek vagy minősítettnek kell lennie, vagy olyan regisztráció alapján kibocsátott fokozott biztonságú aláírásnak, amely egy, a személyazonosságot közhitelesen tanúsító igazolvány és egy közhiteles nyilvántartásban történt ellenőrzés alapján került kiadásra. Bélyegzők esetében szintén a közhiteles nyilvántartásban történt ellenőrzés a követelmény.

Technikai értelemben, amennyiben webAutomatával történik a kommunikáció, hasonló követelmény áll fenn a webszerviz használatához szükséges tanúsítványok vonatkozásában is, ezekkel kerülnek aláírásra a SOAP üzenetek WS security szabványnak megfelelően.

3.6 Regisztráció

A biztonságos kézbesítési szolgáltatást és ezen keresztül a teljes hibrid kézbesítési és konverziós rendszert az Európai Parlament és a Tanács 2014. július 23-i 910/2014/EU rendelete 44. cikk (1) bekezdés b) és c) pontjaiból következően csak regisztrált felhasználók használhatják. A regisztráció web service alkalmazása esetén a csatlakozó szervezet azonosítóinak rögzítését jelenti. Ezt a regisztrációt a szolgáltatást nyújtó szervezet (Magyar Posta) részéről a rendszer adminisztrációját végző munkatársak végzik el.

3.6.1 Gépi kapcsolatok regisztrálása

Gépi kapcsolat regisztrálására a kommunikáció jellegéből következően kizárólag szervezetekhez van lehetőség. Ez minden esetben a Magyar Posta Zrt. regisztrátorainak közreműködését igényli.

A nagyobb küldeménymennyiség kezelését lehetővé tevő gép-gép kapcsolat megvalósítására a hibrid kézbesítési és konverziós rendszer két megoldást biztosít, egy a rendszer által biztosított webszerviz igénybe vételét és a hivatali kapun keresztüli kommunikációt

3.6.1.1 A webszerviz regisztrálása

A webszerviz igénybe vételéhez a csatlakozni kívánó szervezetnek kell rendelkeznie olyan saját alkalmazással, amely képes a hibrid kézbesítési és konverziós rendszer e célra kialakított webszervizével (amit a továbbiakban webAutomata-ként fogunk jelölni) kommunikálni. A webszerviz áttekintő specifikációját az 5. fejezet tartalmazza.

A webAutomata úgy hoz létre egy biztonságos kommunikációs csatornát a kliens (az alkalmazást, amely megszólítja a hibrid rendszer webAutomata interfészét, ezt a továbbiakban „kliensnek" nevezzük) és a Magyar Posta hibrid kézbesítési és konverziós rendszere (a továbbiakban szerver) között, hogy az transzparens a kliens alkalmazás fejlesztője számára.

A megvalósított biztonsági paradigma a következő:

- A kliens és a szerver közötti kommunikációs csatorna, csatornaszintű biztonságát SSL/TLS protokoll biztosítja egy hibrid kézbesítési és konverziós rendszer oldalán (szerver) telepített X.509 magánkulcsának felhasználásával, amely egy kliens oldali TrustStore-ban telepített nyilvános tanúsítvány párjával hozza létre a megfelelően biztonságos csatornát a TLS protokoll (RFC 5246) használatával;
- A kliens üzeneteinek hitelesítése (aláírása) egy kliens oldali kulcstárba (KeyStore) telepített X.509 magánkulccsal történik;
- Ezzel a kliens magánkulccsal történik az egyes webszolgáltatás-hívások SOAP üzenetei fejében a törzsben szereplő információk aláírása a WS security és a W3C XML signature (aláírás) szabványok szerint. A biztonsági modellre vonatkozó részletes XML alapú és a vonatkozó szabványhivatkozásokat is tartalmazó leírás a 2. sz. függelékben található;
- A kliens által készített és a SOAP üzenetben elküldött aláírás érvényességét a hibrid kézbesítési és konverziós rendszer ellenőrzi.

E biztonsági modell használhatóságához a kliens (Igénybe vevői) oldalnak az alábbi követelményeknek kell megfelelnie:

- Rendelkezésre kell állnia egy az Igénybe vevő részére kibocsátott tanúsítvány állománynak, amely tartalmaz egy X.509 magánkulcsot aláírva egy, a megbízható EU közzétevők listáján szereplő magyar és/vagy európai minősített hitelesítés szolgáltató által, illetve a szolgáltató számára elérhetőnek kell lennie az előbbihez tartozó nyilvános tanúsítványnak és a visszavonási listának vagy az OCSP szolgáltatásnak;
- A nyilvános tanúsítványt a Magyar Posta rendelkezésére kell bocsátani, mivel azt hozzá kell rendelni a hibrid szolgáltatási szerződéshez, ugyanis ezt használva történik majd a küldeményeket tartalmazó SOAP üzenetek feladása (aláírása és indítása) a WebAutomata szolgáltatás felé. Ehhez a kibocsátó hitelesítés szolgáltatót és a tanúsítványt regisztrálni kell a szerver oldali hozzáférés-védelmi rendszerben. Tesztelési célokra a Magyar Posta kérés esetén biztosít a saját CA-ja által kibocsátott teszt tanúsítványt, ezt azonban az éles rendszerbe nem lehet regisztrálni;

- A webAutomata, webszolgáltatást biztosító szervernek elérhetőnek kell lennie a kliens felől, amihez el kell végezni a tűzfalak beállítását, tehát a kapcsolat kialakításához meg kell adni a kliens rendszer IP címét.
- A SSL/TLS protokoll használatához szükséges azonosító tanúsítványt (a benne levő nyilvános kulccsal) egy, a hibrid kézbesítési és konverziós rendszer üzemeltetőjéhez küldött, szabványos (PKCS # 10) tanúsítványkérés útján kaphatja meg a kliens a Magyar Postától, de a csatlakozni kívánó is beszerezheti független szolgáltatótól.

Az éles rendszerhez tartozó nyilvános IP címet a sikeres csatlakozási tesztet követően adja meg a Magyar Posta. Mivel IP szűrés biztosított, így természetesen csak megadott IP címről lehet azt majd elérni.
A webAutomata címe a teszt környezetben: https://webautomata.hibrid.uat.posta.hu port:8888
A hozzá tartozó nyilvános IP cím: 194.88.45.250, de ezt is csak az arra feljogosítottak érhetik el.
A webautomata használata esetén, mivel az elég alacsony szinten fér hozzá a rendszer szolgáltatásaihoz, arra is figyelni kell, hogy a rendszerben vannak rögzített címek, amelyek bizonyos szolgáltatásokhoz tartoznak. Ilyen például a hibrid szolgáltatás címe, hybrid_conversion@hmdacs.posta.hu, amit nem webAutomatával történő küldés esetén nem is kell ismerni, mivel a rendszer a kézbesítési utasításból értelmezi, csak mivel a webAutomata címértelmezése megelőzi a kézbesítési utasítás feldolgozását, ezért ki kell tölteni minden egyes ilyen célú *sendMessage* parancsnál. A kliens rendszerekbe éppúgy be lehet ezt az adatot paraméterezni, mint a hivatali kapu címét vagy magának a webAutomatának az IP címét.

A szolgáltatás összehangolása, üzembiztossá tétele, mivel itt egyedi alkalmazásokról van szó, nagy valószínűséggel jelentősebb tesztidőszakot igényel.

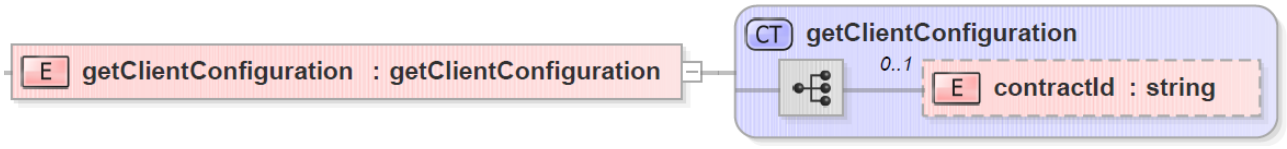
A webAutomata a kapcsolat beállításainak ellenőrzésére és a kapcsolat létezésének ellenőrzésére a webAutomata két eljárás-párt biztosít.

3.6.1.1.1 A szolgáltatás paramétereinek lekérdezése getClientConfiguration

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	getClientConfiguration	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getClientConfiguration	Összetett	A szerződéshez tartozó kliens működési paramétereinek lekérdezése
contractId	Karaktersorozat	A kapcsolatot megvalósító szerződés a rendszerben kapott azonosítója

1. táblázat: getClientConfiguration parancs

A kérdés adatcsomag egyetlen paramétert tartalmaz a szerződés azonosítóját



1. ábra: getClientConfiguration adatcsomagjának szerkezete

Ennek megfelelően a kérést tartalmazó SOAP üzenet is viszonylag egyszerű, azonban az üzenet biztonsága érdekében a fej lényegesen terjedelmesebb, mint maga az üzenet érdemi tartalma:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
         <SignatureValue>
            MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
            ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
            cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
            B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
            M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
         </SignatureValue>
         <KeyInfo>
            <X509Data>
               <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi
                  Főigazgatóság,L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
               </X509SubjectName>
               <X509Certificate>
                  MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
                  dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
                  VQQKDAdQcm9nZXNpMRswGQYDVQQDDBJNYXJjbyBDb25mYWxvbmllcmkwHhcNMTQwOTI1MTMyMDI2
                  <!-- itt folytatódik a tanúsítvány base64 kódolással -->
                  BjpHOh3wUAUGKwBYVrTL1icezubmC6oCOZqyHeEKYjiH+0C1Ic06RZ7Hzt0z860XeqzUA9T6qEYA
                  nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
                  RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
               </X509Certificate>
            </X509Data>
         </KeyInfo>
      </Signature>
   </S:Header>
   <S:Body Id="Body">
      <ns2:getClientConfiguration xmlns:ns2="http://selexes.com/hmdacs">
         <contractId>3047</contractId>
      </ns2:getClientConfiguration>
   </S:Body>
</S:Envelope>
```

A getClientConfiguration SOAP üzenet egy szerkezeti példája

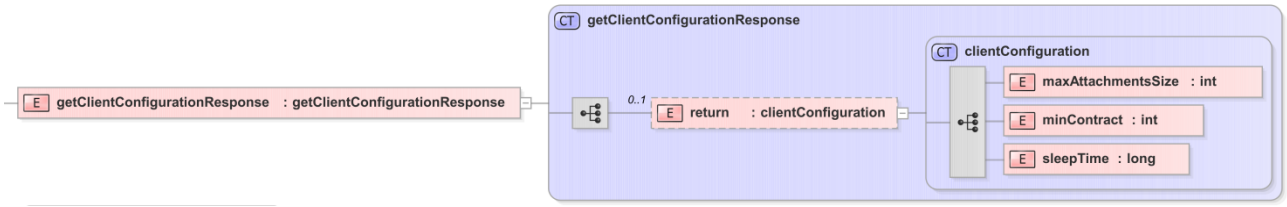
3.6.1.1.2 Válasz a getClientConfiguration parancsra

Válasz adatcsomag elemei

Elem név	Típus	Leírás
getClientConfigurationResponse	Összetett	A szerződéshez tartozó kliens működési paramétereinek visszaadása
return	Összetett	A visszaadott üzenet
clientConfiguration	Összetett	A konfiguráció elemei

2. táblázat: A getClientConfigurationResponse adatcsomag elemei

A válasz adatcsomag három adatot ad vissza:



2. ábra: A getClientConfigurationResponse adatcsomag szerkezete

Elem név	Típus	Leírás
maxAttachmentSize	Integer	A kliens által küldhető legnagyobb üzenet mérete MB-ban
minContract	integer	A legkisebb kezelhető contract azonosító
sleepTime	long	Az az idő, amennyi idő után, ha az adott szerződéshez a getMessage kérdésre nincs válasz, a kapcsolat alvó állapotba kerül

3. táblázat: A clientConfiguration összetett adat elemeinek értelmezése

A fenti paraméterek kiinduló beállításai a JBoss *configuration\configuration\esbapp* könyvtárban a web-automata.properties fájlban találhatók meg. A maximális csatolmányméret alapértelmezésben 20 MB, a szerződések azonosítószámai 1000-től kezdődnek és az alvási idő alapértéke 30000.

A getClientConfigurationResponse SOAP üzenet szerkezete az aláírás hiányában lényegesen egyszerűbb:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:getClientConfigurationResponse xmlns:ns2="http://selexes.com/hmdacs">
         <return>
            <maxAttachmentSize>20</maxAttachmentSize>
            <minContract>1000</minContract>
            <sleepTime>30000</sleepTime>
         </return>
      </ns2:getClientConfigurationResponse>
   </soap:Body>
</soap:Envelope>
```

A getClientConfigurationResponse SOAP üzenet szerkezeti mintája

Ezzel az üzenetváltással tehát meg lehet győződni egyrészt a kapcsolat működőképességéről, másrészt le lehet kérdezni az aktuális beállításokat

3.6.1.1.3 A szolgáltatás tesztelése: probe

A webszolgáltatások alapműködéséhez tartozik egy olyan kérés-válasz pár beépítése, amely minden belső tartalom nélkül, pusztán az üzenetváltás képességét teszteli. Ennek általánosan elfogadott elnevezése a probe. Ez itt is megvalósításra került.

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	probe	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
probe	Összetett	egyetlen üres sequence tagot tartalmaz

4. táblázat: A probe parancs

3. ábra: A probe adatcsomag szerkezete

Mivel itt a SOAP üzenet szerkezetének megismétlése már nem adna többletet, kizárólag az adatcsomag XML sémáját adjuk meg:

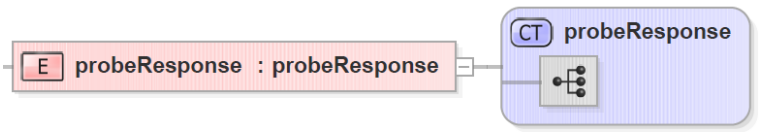
```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
      xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="probe" type="tns:probe"/>
   <xs:complexType name="probe">
      <xs:sequence/>
   </xs:complexType>
</xs:schema>
```

A probe kérés xml sémája

3.6.1.1.4 Válasz a probe parancsra

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
probeResponse	Összetett	egyetlen üres sequence tagot tartalmaz

5. táblázat: probe válasz adatcsomagjának elemei



4. ábra: A probe válasz szerkezete

A válasz xml sémája lényegében a kéréssel megegyező:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
      xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="probeResponse" type="tns:probeResponse"/>
   <xs:complexType name="probeResponse">
      <xs:sequence/>
   </xs:complexType>
</xs:schema>
```

A probeResponse XML sémája

Ez a kérés-válasz pár tehát első sorban arra alkalmas, hogy meggyőződhessen az igénybe vevő, hogy a kapcsolat él.

3.6.1.2 Hivatali kapu regisztrálása

Hivatali kapuval csak akkor célszerű kapcsolódni a biztonságos kézbesítési rendszerhez, ha van már olyan jól működő szolgáltatás, amely ezen az úton hatékonyan kommunikál. Az egyedi üzenetek kiküldésére a hivatali kapu kézi kezelőfelülete alkalmazható.

A regisztrációhoz értelemszerűen a szervezet által használt hivatali kapu szükséges. Ennek biztosítása és működtetése a NISZ Zrt. útján történik. A hivatali kapun keresztüli szolgáltatás igénybe vételéhez a csatlakozni kívánó szervezet hivatali kapu azonosítójának megadása szükséges. Ezt a regisztrátorok egyeztetik a NISZ Zrt által folyamatosan frissített hivatali kapu listával és annak alapján rögzítik.

Ebben az esetben a bejövő kommunikáció címzettje a Magyar Posta Zrt. hivatali kapuja, melynek HK azonosítója az éles környezetben 506341775, a teszt környezetben 404184391 a rövid neve MPZRT. Ugyanezen azonosítójú feladótól érkeznek a továbbított, vagy a hibrid rendszer által generált küldemények is.

E kapcsolati forma esetén külön, a kapcsolatra vonatkozó tesztelésre nincs szükség, ha a szervezet megfelelően működteti a saját hivatali kapu kapcsolatát, akkor a másik oldalon a Magyar Posta már stabil kapcsolattal rendelkezik. Ugyanakkor fontos itt is jelezni, hogy ez a csatorna kizárólag KRX formátumú küldemények kezelésére van felkészítve, itt nincs lehetőség „szerkezet nélküli” üzenetek küldésére

3.7 Financiális feltételek

Mint minden bizalmi szolgáltatás, a kézbesítési szolgáltatás, biztonságos kézbesítési szolgáltatás is azonos feltételekkel vehető igénybe, mint minden más SZEÜSZ illetve KEÜSZ. A vonatkozó

jogszabályban díjmentességre jogosult igénybevevők számára a szolgáltatás díjfizetés nélkül történik. Amennyiben a felhasználó nem jogosult a díjmentességre, a szerződésnek rendeznie kell a fizetés feltételeit is.

A nagyobb forgalmú feladók esetében van lehetőség az utólagos elszámolásra, amikor a Magyar Posta meghatározott időközönként számlázza utólag az adott időszakban igénybe vett szolgáltatást. Értelemszerűen ennek az igénybevételi módnak előzetes feltétele az előzetesen erre vonatkozóan megkötött szerződés.

A címzettek oldalán a szolgáltatásnak nincs külön díja, de az azonosítás követelménye miatt a fogadásnak is feltétele az érvényes szolgáltatási szerződés.

3.8 A kommunikációs csatornák sajátosságai

Az alapfunkcionalitás minden esetben azonos, azonban minden megoldásnak vannak olyan sajátosságai, amelyeket érdemes figyelembe venni az alkalmazandó kommunikációs csatorna kiválasztásakor, kialakításakor. Mivel a kézbesítési rendszer nem értelmezi az egyes csatolmányokat, így azok formátumára nincs megkötés, azonban makrókat, szkripteket a víruskereső rendszer nem engedi tovább.
A küldött dokumentumok minden esetben vírus- és spamszűrésen esnek keresztül, és a felismert fertőzött küldeményeket még felvétel előtt visszautasítja a rendszer, ennek ellenére a küldemények vírusmentességéért a küldő felel.
Az egyes csatolmányok neve a küldeményben különböző kell legyen, mivel az itt alkalmazott logika nem képes egy küldeményen belül két azonos nevű állományt kezelni. Az elküldhető küldemény tesztelt méretkorlátja 500 MB.

3.8.1 Webszervizen keresztüli kommunikáció

A webszerviz esetében a rendszer képességei jelentős mértékben függenek a kliens megvalósításától. Értelemszerűen az ügyfél által megvalósított kliens megoldástól függ, hogy az manuális és/vagy gép-gép kapcsolat lehetőségét biztosítja. A gyakorlatban az így megvalósított manuális kapcsolatot csak tesztelési célra ajánljuk. Maximum 10 csatolmány megengedett egy küldeményben, és tiltott két azonos nevű csatolmány küldése egy küldeményen belül.

A webszervizt megszólító alkalmazásnak értelmezni, használni kell tudni a webAutomata kapcsolat útján elérhető parancsokat, és az azokban meghatározott adatcsomagok-sémáknak megfelelő szerkezetű SOAP üzenetekben kell továbbadnia az információt a rendszernek. A rendszer válaszai és a kommunikáció bizonyítékai is ezen a csatornán keresztül érkeznek. Értelemszerűen van lehetőség ebben az esetben is mailben vagy SMS-ben kérni egyes figyelmeztetéseket. A rendszer által a kommunikációhoz biztosított webszerviz részeként elérhető szolgáltatások és az azok megszólításához szükséges, illetve a válaszok fogadása során megkapott adatok, adatcsomagot az alkalmazott parancsok köré csoportosított leírása az 5. fejezetben olvasható. Alapesetben az

erőforrásokkal való takarékosság érdekében a webszerviz kliensekhez kisebb maximális fájlméret van beállítva, de ez a képesség paraméterezhető a rendszerben.

Amennyiben egy ügyfél egyidejűleg több szálon próbál meg küldeményeket küldeni és fogadni, ebben az esetben a kliens rendszernek kell gondoskodni az egy üzenetek tartozó adatcsomagok összerendezéséről. Ugyanakkor minden adatcsomag biztosít az összerendezéshez elégséges kulcsinformációt.

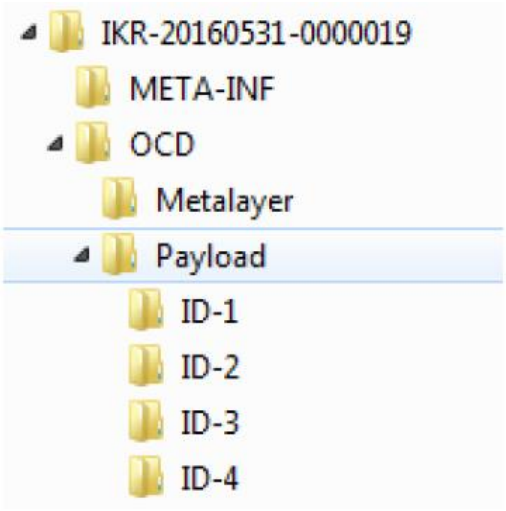
A gépi kommunikáció sajátosságából következően a küldemény visszautasításának eljárásrendje eltér az általánosan alkalmazott megoldástól, ahol erre a célra egy kattintás elengedő. Itt a küldemény elutasítására vonatkozó külön átvételi elismervénytípust kell letölteni és aláírni, mivel ezzel a megoldással biztosítható a homogén gépi kommunikáció lehetősége valamennyi esetre.

3.8.2 Hivatali kapun keresztüli kommunikáció

A hivatali kapun keresztüli kommunikáció a hivatali kapuk, illetve iratkezelő rendszerek közötti kommunikációra már széles körben használt KRX formátum alkalmazásával történik. Ebben az esetben valamennyi a hivatali kaputól érkező, illetve oda címzett üzenet egységesen ezt a formátumot használja, a küldő, illetve fogadó rendszereknek az ebben a formátumban érkező, illetve indított küldemények kezelésére kell felkészülniük. A KRX formátum tulajdonképpen egy konténer, amely meghatározott könyvtárszerkezetben tárolja az információt, azaz a küldemény mellékleteit, illetve az azok leírásához, kezeléséhez szükséges meta-adatokat.

A küldemény csatolmányait ebben az esetben a **Payload** (tartalom) könyvtár tartalmazza. Ebben helyezkednek el ténylegesen továbbítandó dokumentum(ok) az alatta kialakított alkönyvtárakban. Az ID-1, ID-2, ID-3, ID-4 (maximum 10) könyvtárak tartalmazzák küldemény egyes mellékleteit, és ebből a logikából következően a különböző alkönyvtárak azonos nevű csatolmányokat, például ugyanazon dokumentum különböző verzióit, állapotait is tartalmazhatják.

Metalayer könyvtár tartalmazza a leíró adatokat, amelyek az információcseréhez szükséges kiegészítő adatokat adják meg. Itt található egy állomány, a kuldemeny_meta.xml, amely tartalmazza a feladót, címzettet, a küldemény készítési dátumát és tárgyát.



5. ábra: Egy jellemző KRX szerkezete a kicsomagolást követően

Ezen túlmenően is van lehetőség különféle információk tárolására a szerkezetben, azonban ezeket a küldemény kézbesítési/biztonságos kézbesítési szolgáltatás útján történő továbbításánál nem vizsgáljuk, hasznosítjuk. Magát a kuldemeny_meta-ra vonatkozó séma dokumentációt a Nova iratkezelő rendszer fejlesztői gondozzák, az azzal kapcsolatos információkat tőlük lehet beszerezni.

A rendszer értelemszerűen nem vizsgálja, hogy a hivatali kaput a küldő/fogadó oldalon gépi vagy manuális interfésszel érik el, a kommunikáció egységes formában történik.

4 A kézbesítési/ biztonságos kézbesítési szolgáltatások alapelemei

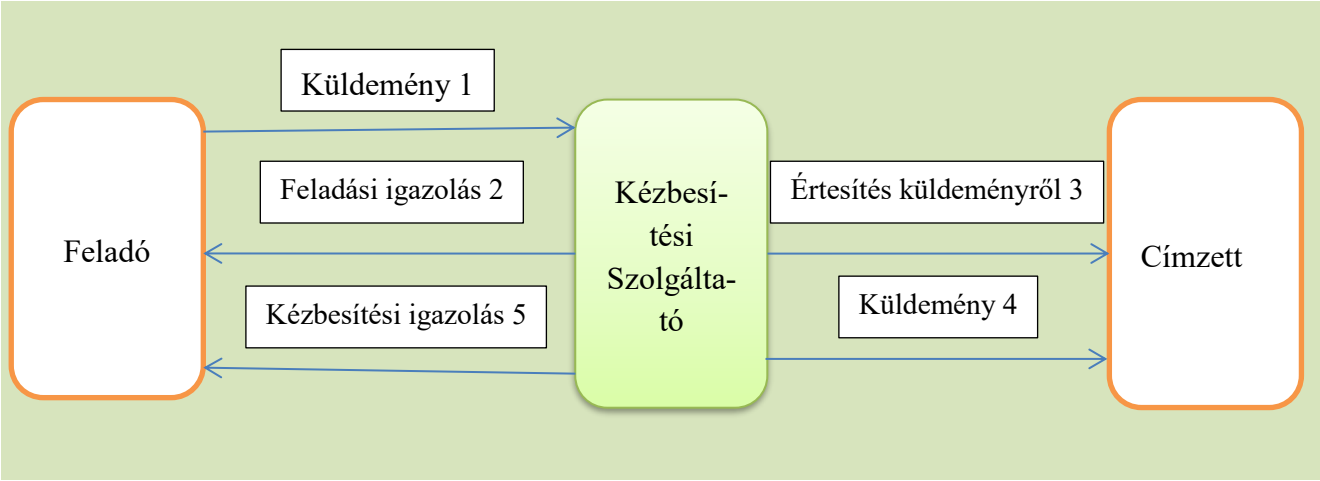
A biztonságos kézbesítési szolgáltatás, illetve a kézbesítési szolgáltatás alapfunkcionalitását meghatározás szinten már az 1. fejezet bemutatta, itt magát a küldemény összeállításával, elküldésével kapcsolatos folyamatot mutatjuk be annak érdekében, hogy a webszerviz által nyújtott funkciók értelmezhetőek legyenek.

4.1 A szolgáltatások kommunikációs modellje

Ebben a szolgáltatások három szereplője a küldő, a szolgáltató és a címzett tevékenységét vizsgáljuk a két szolgáltatás során. Alapvető fontosságú, hogy a feladó és a címzett közötti kommunikációba egy harmadik fél, a szolgáltató független megbízható félként kapcsolódik be, azaz az ő állításai (amelyek hitelességét technikai eszközök is alátámasztják) a kommunikáció körülményeinek tanúsításával alátámasztják annak tartalmi és időbeli hitelességét.

4.1.1 A kézbesítési szolgáltatás kommunikációs modellje

A Magyar Posta által biztosított kézbesítési szolgáltatás a hagyományos postai szolgáltatások közül a könyvelt (ajánlott) küldemény szolgáltatási (hitelességi) szintjének felel meg. A szolgáltató e szolgáltatásban azt tanúsítja, hogy egy adott küldeményt az adott (változatlan) tartalommal egy adott időpontban a címzett rendelkezésére bocsátotta. Természetesen a szolgáltatás nyújtásának az is feltétele, hogy mind a feladó, mind a címzett előzetesen regisztrált felhasználója legyen a szolgáltatásnak. Ebből következően a nem létező címzett esete nem értelmezhető. A regisztráció követelményeiről az előző fejezetben a többi előfeltétellel együtt írtunk.



6. ábra: A kézbesítési szolgáltatás elvi kommunikációs modellje

A küldemény feldolgozásának, illetve a hitelesítésnek a lépései egy hibátlan kézbesítési folyamat során a következők:

1. A feladó elküldi küldeményét a címzettnek, azaz átadja a kézbesítési szolgáltatónak kézbesítésre. [Küldemény 1]
2. A kézbesítési szolgáltató fogadja a küldeményt és átmeneti tárba helyezi, *feladási igazolást* (Dispatch receipt) – postai terminológiával feladóvevényt – készít, mely a küldemény és a bizonyíték azonosítószámával ellátott, általa elektronikusan lebélyegzett PDF dokumentum. Ez tartalmazza egyrészt olvasható formában a feladó és címzett adatai mellett az eredeti küldemény lenyomatát (hash) és időbélyeggel van ellátva. Az időbélyeget minősített időbélyeg szolgáltató készíti. Az igazolás emellett egy beágyazott XML állomány formájában gépi feldolgozásra alkalmas formában is tartalmazza ezeket az információkat.
3. A kézbesítési szolgáltató visszaküldi a feladási igazolást a feladónak. [Feladási igazolás 2]
4. A kézbesítési szolgáltató létrehoz egy (a szerződéstől függően e-mail vagy SMS) *értesítést,* amellyel értesíti a címzettet, hogy számára egy küldemény érkezett, amelyet a tárhelyén átvehet. [Értesítés küldeményről 3]
5. A kézbesítési szolgáltató a kézbesítendő küldeményt elhelyezi a címzett kizárólagos felügyelete alatt álló tárhelyén. [Küldemény 4.]
6. A kézbesítés tényéről a kézbesítési szolgáltató *kézbesítési igazolást* **(**Delivery Receipt) készít, itt is egy elektronikus bélyegzővel ellátott PDF dokumentum formájában, mely a küldemény és a bizonyíték azonosítószáma valamint a feladó és a címzett adatai mellett a kézbesített küldemény lenyomatát tartalmazza. Tartalmaz emellett időbélyeget, amelyet minősített időbélyeg szolgáltató készített. Az igazolás emellett egy beágyazott XML állomány formájában gépi feldolgozásra alkalmas formában is tartalmazza ezeket az információkat.
7. Végezetül a kézbesítési szolgáltató a kézbesítési igazolást visszaküldi a Feladónak. [Kézbesítési igazolás 5]

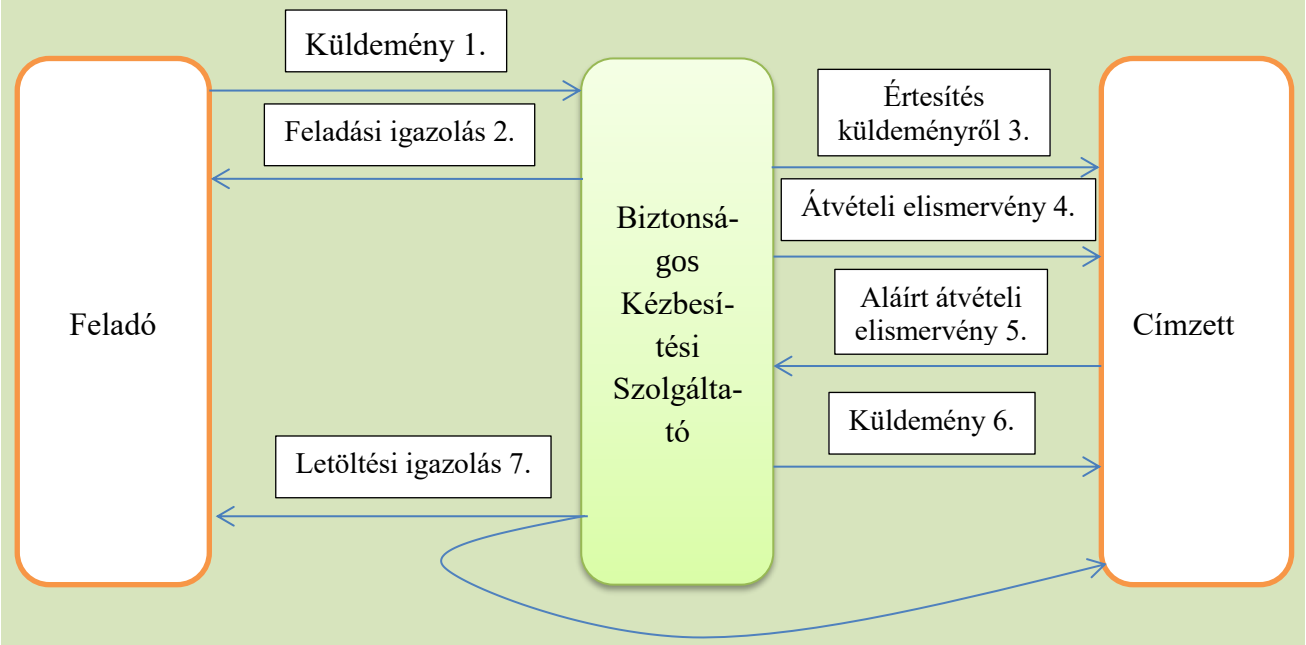
Amennyiben a küldemény nem jut el a szolgáltatóig, akkor a küldés nem értelmezhető, amennyiben az átvitel során sérül a küldemény vagy a vírus- és spamellenőrzés alapján továbbíthatatlan a küldemény a címzett a feladási igazolás helyett kap negatív faladási igazolást. Mivel itt rendelkezésre bocsátás történik, egyéb hibajelenség csak abban az erősen elméleti esetben fordulhat elő, ha a küldemény címzettje által megadott értesítési cím érvénytelen. Ekkor a rendszer a kézbesítés meghiúsulásáról szóló értesítést küld a feladónak, és a küldeményt – a bizonyíték megőrzése mellett – visszavonhatatlanul törli.

A kézbesítés a küldéssel kapcsolatos események tanúsítására alkalmas naplóbejegyzései és az elkészített bizonyítékok öt évig megőrzésre kerülnek és azokból a szolgáltató szükség esetén információt szolgáltat az érintetteknek és az adatkérésre jogszabály szerint jogosultaknak.

A feladó és a címzett tárhelye az érintettek kizárólagos felügyelete alatt áll, az alapszerződés 100 MB-os tárkapacitást biztosít, az ennél nagyobbra külön díj ellenében szerződni szükséges. A szolgáltató itt értelemszerűen csak a rendszer rendelkezésre állásáért szavatol, az egyes ott tárolt dokumentumok vonatkozásában hozzáférés hiányában nem értelmezhető a felelőssége.

4.1.2 A biztonságos kézbesítési szolgáltatás kommunikációs modellje

A Magyar Posta által biztosított biztonságos kézbesítési szolgáltatás a hagyományos postai szolgáltatások közül a hivatalos irat tértivevényes küldeménytípus szolgáltatási (hitelességi) szintjének felel meg. A szolgáltató e szolgáltatásban azt tanúsítja, hogy egy adott küldeményt az adott (változatlan) tartalommal egy adott időpontban a címzett vagy az általa megbízott helyettese aláírásával igazoltan átvette. Ezen túlmenően e kézbesítési módnál az átvétel esetleges meghiúsulását is igazolja a szolgáltató. Értelemszerűen szolgáltatás nyújtásának itt is feltétele, hogy mind a feladó, mind a címzett előzetesen regisztrált felhasználója legyen a szolgáltatásnak és a regisztráció részeként rögzítsék az átvételre szolgáló elektronikus aláírásuk, bélyegzőjük (vagy az azt helyettesítő AVDH szolgáltatás) lenyomatát. A regisztráció követelményeiről a 3.6 fejezetben írtak itt is érvényesek.



7. ábra: A biztonságos kézbesítési szolgáltatás kommunikációs modellje

Az alább leírás a sikeres folyamat elemeit mutatja be (nincs hiba a kommunikációban és a címzett átveszi).

1. A feladó elküldi küldeményét a címzettnek, azaz átadja a biztonságos kézbesítési szolgáltatónak kézbesítésre. [Küldemény 1]
2. A biztonságos kézbesítési szolgáltató fogadja a küldeményt és átmeneti tárba helyezi, feladási igazolást (Dispatch receipt) – postai terminológiával feladóvevényt – készít, mely a küldemény és a bizonyíték azonosítószámával ellátott, általa elektronikusan lebélyegzett PDF dokumentum. Ez tartalmazza egyrészt olvasható formában a feladó és címzett adatai mellett az eredeti küldemény lenyomatát (hash) és időbélyeggel van ellátva. Az időbélyeget minősített időbélyeg szolgáltató készíti. Az igazolás emellett egy beágyazott XML állomány formájában gépi feldolgozásra alkalmas formában is tartalmazza ezeket az információkat.

3. A biztonságos kézbesítési szolgáltató visszaküldi a feladási igazolást a feladónak. [Feladási igazolás 2]
4. A biztonságos kézbesítési szolgáltató létrehoz egy (a szerződéstől függően e-mail vagy SMS) *értesítést*, amellyel értesíti a címzettet, hogy számára egy küldemény érkezett, és tegye meg az átvételhez szükséges intézkedéseket. [Értesítés küldeményről 3]
5. A biztonságos kézbesítési szolgáltató létrehoz egy átvételi elismervényt (Acceptance receipt) amely a küldemény és a bizonyíték azonosítószámával ellátott, PDF dokumentum. Ez tartalmazza olvasható formában a feladó és címzett adatai mellett az eredeti küldemény lenyomatát (hash).
6. A biztonságos kézbesítési szolgáltató a létrehozott elismervényt a címzett rendelkezésére bocsátja. [Átvételi elismervény 4.]
7. A címzett (vagy az erre feljogosított helyettese) aláírja az átvételi elismervényt.
8. A címzett visszaküldi (feltölti) az aláírt átvételi elismervényt. [Aláírt átvételi elismervény 5.]
9. A biztonságos kézbesítési szolgáltató ellenőrzi, hogy arra jogosult írta-e alá az átvételi elismervényt.
10. A biztonságos kézbesítési szolgáltató elérhetővé teszi a küldeményt a címzett számára.
11. A biztonságos kézbesítési szolgáltató létrehozza a *letöltési igazolást* (Download receipt) – ez felel meg a postai gyakorlatban a tértivevény nyomtatványnak – mely a küldemény és a bizonyíték azonosítószámával ellátott, a szolgáltató által elektronikusan lebélyegzett PDF dokumentum. Ez tartalmazza egyrészt olvasható formában a feladó és címzett adatai mellett az eredeti küldemény lenyomatát (hash), az átvétel időpontját és időbélyeggel van ellátva. Az időbélyeget minősített időbélyeg szolgáltató készíti. Az igazolás emellett beágyazott állomány formájában tartalmazza az aláírt átvételi elismervényt, és gépi feldolgozásra alkalmas XML formájában is a küldeménnyel és átvételével kapcsolatos információkat.
12. A biztonságos kézbesítési szolgáltató a feladó és a címzett rendelkezésére bocsátja a letöltési igazolást [Letöltési igazolás 7.].

Az összetettebb folyamat és az ebben az esetben megállapítható kézbesítési vélelem miatt itt a lehetséges meghiúsulási esetek száma lényegesen nagyobb, és az említett két, a szolgáltató által kiadott igazolásnak ezen esetekben is megfelelő információt kell szolgáltatnia a küldemény sorsáról. Itt is igaz, mivel a rendszer csak regisztrált felhasználók közötti kommunikációt támogat, nem fordulhat elő, hogy nem létező címzettnek indul a küldemény és ezért hiúsulna meg a továbbítás.

1. Amennyiben a küldemény nem jut el a szolgáltatóig pl. a méretkorlát túllépése miatt, akkor a küldés maga nem értelmezhető. Ebben az esetben a feladó semmiféle igazolást nem kap, azaz a feladás sem történt meg.
2. Amennyiben az átvitel során sérül a küldemény vagy a vírus- és spamellenőrzés alapján továbbíthatatlan, a címzett a feladási igazolás helyett a küldemény be nem fogadásáról kap igazolást. Ebben az esetben is csak a küldés megismétlésével juthat megfelelő feladási bizonyítékhoz. [nemleges feladóvevény]

A további esetekben a feladó már rendelkezik egy (pozitív) feladási igazolással, de különböző okok miatt mégsem jut el a küldemény a címzetthez. A meghiúsulás körülményeire vonatkozó információt ekkor a (negatív) letöltési igazolás tartalmazza, amelynek meghatározott eseteihez kézbesítési vélelem kapcsolódhat.

1. Itt is előfordulhat elméletileg, hogy a levelező rendszer, illetve a távközlési szolgáltató azt jelzi a kiküldött értesítésre [Értesítés küldeményről 3], hogy maga az értesítési cím nem létező. Ekkor a nemleges letöltési igazolás (amit ebben az esetben értelemszerűen csak a feladónak lehet megküldeni), azt a tény tartalmazza, hogy a címzett az adott címen nem elérhető. Ez a hagyományos postai esetekben az „elköltözött” esetének felel meg.
2. A címzett jogosult megtagadni a küldemény átvételét (helyettes átvevőnek, meghatalmazottnak nincs ilyen jogosultsága). Ebben az esetben a feladó és a címzett is az átvétel megtagadásának időpontját tartalmazó (negatív) letöltési igazolást kap, amely értelemszerűen nem tartalmaz aláírt átvételi elismervényt. A kézbesítési vélelem ebben az esetben az átvétel megtagadásának időpontjában áll be. Értelemszerűen a küldemény a továbbiakban már nem elérhető, azt a rendszer törli.
3. Ha a címzett öt munkanapig nem reagál a megküldött értesítésre (de a levelező rendszer, illetve a távközlési szolgáltató nem jelzi a cím alkalmatlanságát), akkor a rendszer ismét kiküldi az értesítést.
4. Ha a címzett (vagy valaki, aki hozzáfér a rendszerhez) letölti az átvételi elismervényt, és egy rendelkezésére álló aláírással vagy bélyegzővel ellátva feltölti a rendszerbe, de az nem szerepelt az átvételre feljogosított (regisztrált) aláírások között, a rendszer ismételten értesítésben kéri az aláírt átvételi elismervény megküldését. Ebben az esetben az időtartam mérése tovább folytatódik, mintha nem történt volna nem megfelelő átvételi kísérlet.
5. Ha az újabb öt munkanap is átvétel nélkül (sikertelenül) telik el, a rendszer negatív letöltési igazolást állít elő és küld ki mind a feladó, mind a címzett részére. Ez a rendszer által előállított, a küldemény és a bizonyíték azonosítószámával ellátott, a szolgáltató által elektronikusan lebélyegzett PDF dokumentum. A (negatív) letöltési igazolás tartalmazza egyrészt olvasható formában a feladó és címzett adatai mellett az eredeti küldemény lenyomatát (hash), az eredeti küldés és az igazolás kiállításának időpontját és ez utóbbi időpontban időbélyeggel van ellátva. Az időbélyeget minősített időbélyeg szolgáltató készíti. Az igazolás emellett beágyazott állomány formájában tartalmazza gépi feldolgozásra alkalmas XML formájában is a küldeménnyel kapcsolatos információt. Az ilyen negatív letöltési igazolás kiállításának időpontját tekintik a kézbesítési vélelem beállása időpontjának. Az igazolás kiállítását követően a küldemény már nem elérhető, azt a rendszer törli.

4.2 A rendszer által készített igazolások, tanúsítványok

A kommunikációs folyamatban összesen három igazolástípus fordulhat elő, mindhárom pozitív és negatív tartalommal egyaránt elvileg létezhet, bár az előfordulási gyakoriságuk lényegesen eltérő. Ezek formáját és tartalmát mutatja be ez a fejezet. A feladási igazolás a kézbesítési szolgáltatásra és a biztonságos kézbesítési szolgáltatásban egyaránt használt, a kézbesítési szolgáltatás esetében a

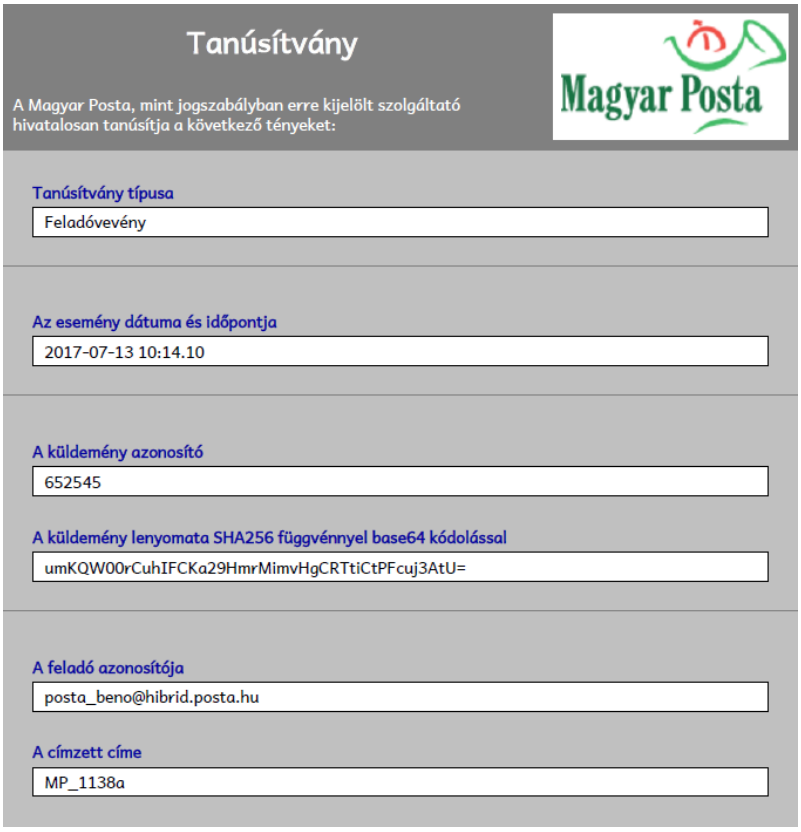
szolgáltató kézbesítési igazolással tanúsítja, hogy a küldeményt a címzett rendelkezésére bocsátotta, a biztonságos kézbesítési szolgáltatási esetében viszont letöltési igazolással tanúsítja a küldemény átvételét. Ebben az esetben a címzettnek rendszer által generált átvételi elismervény – azaz még egy űrlaptípus – aláírásával kell igazolnia az átvételt. Az átvétel megtagadása webAutomata használata esetén egy speciális, a visszautasítást tanúsító űrlap aláírásával történik a visszautasítás annak érdekében, hogy a folyamatot egységesen lehessen megvalósítani. Gépi kommunikáció esetén maga a visszautasítás egy meglehetősen elméleti konstrukció, nincs példa a használatára, hiszen a címzett pozíciója ebben az esetben lényegesen kedvezőtlenebb lenne.

A feladó és a címzett az igazolások hitelességét és sértetlenségét, azok elektronikus aláírásának és időbélyegzőjének ellenőrzésével kontrollálhatja.

4.2.1 Feladási igazolás

A feladónak a rendszer a Magyar Posta által elektronikusan lebélyegzett és időbélyegzővel ellátott feladási igazolással (a hagyományos postai feladóvevénynek felel meg) igazolja vissza, hogy a KSZ, BKSZ küldeményt befogadta. Az igazolás egy PDF dokumentum, amely tartalmazza a küldemény és az igazolás a rendszerben kapott azonosítószámát, a küldemény lenyomatát a feladó és a címzett a rendszerben használt azonosítóját. A PDF dokumentum emellett beágyazva tartalmaz két XML állományt, amelyek a küldeménnyel kapcsolatos információkat tartalmazzák ebben a formátumban, és így az információ gépi feldolgozásra is alkalmas.

A feladási igazolást a rendszer az adott csatornának megfelelő formátumban állítja elő. A webAutomata esetén ez közvetlenül egy PDF állomány, míg a hivatali kapun keresztüli eljuttatás esetén a PDF állomány egy KRX konténerbe csomagolja és ellátja a szükséges kiegészítő adatokkal, ahogy azt már a 3.8.2 fejezet bemutatta.



8. ábra: A feladási igazolás küldemény képi megjelenése

A dokumentum tartalmaz két beágyazott (tehát az alapállománnyal együtt aláírt) XML állományt, így azok hitelesítése is biztosított az egész állományon elhelyezett elektronikus bélyegzővel és időbélyegzéssel. Ezek körül az első, a DispatchNonDispatchCertificate.xml, a küldemény felvételét tanúsító alapvető információkat tartalmazza. Egy mintapéldány az alábbiakban látható

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:evidence xmlns:ns2="http://selexes.com/hmdacs"
      evidenceId="674542"
      evidenceType="DispatchCertificate"
      consignmentId="652615"
      refEvidenceId="0"
      extConsignmentId="0"
      consignmentHash="75784fd16149da00bf5fd01319727bcb8890e3410953540f8ff0a18ef3e48169"
      senderIdentificationMethod="KAU"
      recipientIdentificationMethod="KAU"
      senderAddress="posta_beno@hibrid.posta.hu"
      eventTime="2017-07-20T16:04:10.731+02:00">
      <referenceRecipient electronicAddress="1138_MP"/>
      <recipients>
            <recipient electronicAddress="1138_MP"/>
      </recipients>
</ns2:evidence>
```

A feladási igazolásban található DispatchNonDispatchCertificate.xml állomány mintája

A megjelenített adatok remélhetően önmagukat magyarázzák. Mivel az igazolások nem kizárólag a KSZ, BKSZ esetén használatosak, tartalmaznak olyan elemeket is, amelyek itt tulajdonképpen szükségtelenek lennének, de az egységes kezelhetőség érdekében itt is megjelenítésre kerülnek

(refEvidenceId, extConsignmentId, referenceRecipient), ezek a feldolgozás során is figyelmen kívül hagyhatók. Értelemszerűen amennyiben nem sikerül a küldeményt felvenni, mert olyan hiányosságban vagy hibában szenved, hogy már fizikailag megérkezik, de a kezelése lehetetlen, akkor az <evidenceType> értéke "NonDispatchCertificate".

A második beágyazott XML a küldemény egészének teljességének tanúsítására, illetve ellenőrzésére szolgáló IndexFile.xml állomány. Egy mellékletet tartalmazó küldemény esetében ennek használata tulajdonképpen szükségtelen lenne, hiszen a küldemény egészének lenyomata megegyezne az egyetlen csatolmány lenyomatával. Itt azonban az egységes kezelés érdekében a rendszer a DispatchNonDispatchCertificate.xml-ben is azt az algoritmust használja, amit több csatolmány esetén kell, azaz ismételten lenyomatot készít az eredeti állományról (több állomány esetén itt az egyes lenyomatok konkatenációja szerepel) készített lenyomatról. A másik különbség a nyomtatott formához képest az ábrázolásban van. Míg a megjelenített PDF állományban a tömörebb forma érdekében base64 kódolással (44 karakteren) szerepel a lenyomat, addig az XML állományban bájtonként két hexadecimális karakterrel 64 karakter (32 bájt, azaz 256 bit) hosszúságú az SHA256 függvénnyel előállított, 256 bites állomány. A fájl neveként látható 16 bájtos bájtsorozatnak nincs jelentősége ebben az esetben, az a belső munkaközi tárolási helyet azonosítja.

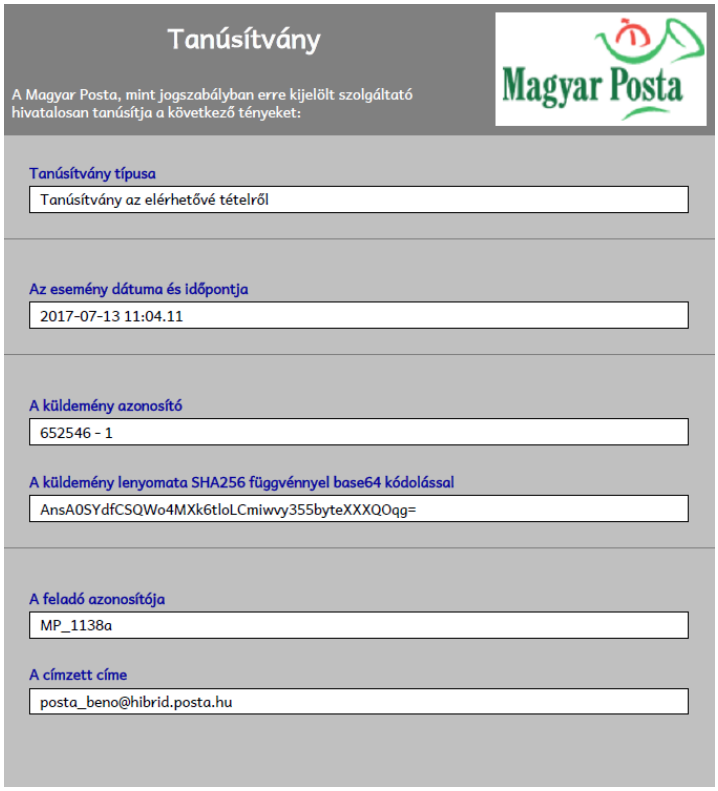
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConsignmentIndex xmlns="http://www.hmdacs.posta.hu/ConsignmentIndex">
    <ConsignmentFile>
        <FileName>5FD09E600ABF9F540000000055709CA7</FileName>
        <Prog>1</Prog>
        <Hash>e9fc2ae3a95c1f4675cbc78eeea8fd2d63aea4ce10eca82ded928021e291a6ab</Hash>
        <HashAlgorithm>SHA256</HashAlgorithm>
    </ConsignmentFile>
</ConsignmentIndex>
```

A feladási igazolásban található IndexFile.xml állomány mintája

A bizonylatot a feladó kapja meg (a beküldési csatornának megfelelő formátumban), és a rendszer a teljes megőrzési időszakra (alapesetben 5 év) biztosítja a hiteles megőrzését, azaz megfelelő megkeresés esetén ismételten a feladó vagy a megkereső és azt jogszabály vagy bíró döntés alapján megismerni jogosult szerv vagy személy részére képes rendelkezésre bocsátani.

4.2.2 Kézbesítési igazolás

A feladónak a rendszer a Magyar Posta által elektronikusan lebélyegzett és időbélyegzővel ellátott kézbesítési igazolással igazolja vissza címzettenként, hogy mint kézbesítési szolgáltató küldeményt a címzett rendelkezésére bocsátotta. (a hagyományos postai szolgáltatásban nincs szorosan vett megfelelője). Az igazolás egy PDF dokumentum, amely tartalmazza a küldemény és az igazolás a rendszerben kapott azonosítószámát, a küldemény lenyomatát a feladó és a címzett a rendszerben használt azonosítóját. A PDF dokumentum emellett beágyazva tartalmaz egy XML állományt, amely a küldeménnyel kapcsolatos információkat hordozza ebben a formátumban, és így az igazolás gépi feldolgozásra is alkalmas.



9. ábra: A kézbesítési igazolás küldemény képi megjelenése

A dokumentum tartalmaz egy beágyazott (tehát az alapállománnyal együtt aláírt) XML állományt, így annak hitelesítése is biztosított az egész állományon elhelyezett elektronikus bélyegzővel és időbélyegzéssel. Ez a DeliveyNonDdeliveryCertificate.xml, amely a küldeményre vonatkozó, illetve a rendelkezére bocsátást tanúsító alapvető információkat tartalmazza. Egy mintapéldány az alábbiakban látható

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:evidence xmlns:ns2="http://selexes.com/hmdacs"
            evidenceId="674408"
            evidenceType="DeliveryCertificate"
            consignmentId="652546"
            refEvidenceId="0"
            extConsignmentId="1"
            ConsignmentHash="027b00d1261d7c24905a8e0c5e4ead9682c29a2c2fcb7e796f2b5e5d75d03aa8"
            senderIdentificationMethod="KAU"
            recipientIdentificationMethod="KAU"
            senderAddress="MP_1138a"
            eventTime="2017-07-13T11:04:11.548+02:00">
      <referenceRecipient electronicAddress="posta_beno@hibrid.posta.hu"/>
      <recipients>
            <recipient electronicAddress="posta_beno@hibrid.posta.hu"/>
      </recipients>
</ns2:evidence>
```

A kézbesítési igazolásban található DeliveryNonDeliveryCertificate.xml állomány mintája

A kézbesítési igazolásban található adatok szerkezete és adattartalma megegyezik a feladási igazolásnál a 4.2.1 fejezetben már bemutatott adatokkal. Itt is ugyanazokra a sajátosságokra kell figyelemmel lenni.

Mivel ennek a tanúsítványnak a kiállítása nem követeli meg a címzett közreműködését, így itt a negatív igazolás csak abban a kivételes esetben fordulhat elő, ha a feldolgozás (néhány másodpercnyi) időszaka alatt a címzett megszünteti a szerződését (egyébként már a feladás is meghiúsult volna), vagy valamennyi értesítési címről visszapattan a küldemény érkezéséről szóló értesítés.
Ezt a bizonylatot is a feladó kapja meg (a beküldési csatornának megfelelő formátumban), és a rendszer a teljes megőrzési időszakra (alapesetben 5 év) biztosítja a hiteles megőrzését, azaz megfelelő megkeresés esetén ismételten a feladó vagy a megkereső és azt jogszabály vagy bíró döntés alapján megismerni jogosult szerv vagy személy részére képes rendelkezésre bocsátani.

4.2.3 Átvételi elismervény

A címzett (több címzett esetében valamennyi címzett) a küldemény érkezéséről először e-mailben, illetve amennyiben a szerződése ezt tartalmazza, SMS-ben kap értesítést. Ez az értesítés mind kézbesítési, mind biztonságos kézbesítési szolgáltatás esetében kiküldésre kerül. Kézbesítési szolgáltatás esetén a címzettnek ezek után már csak az a feladata, hogy a tárhelyén, a csatornának megfelelő eljárással átvegye a küldeményt.

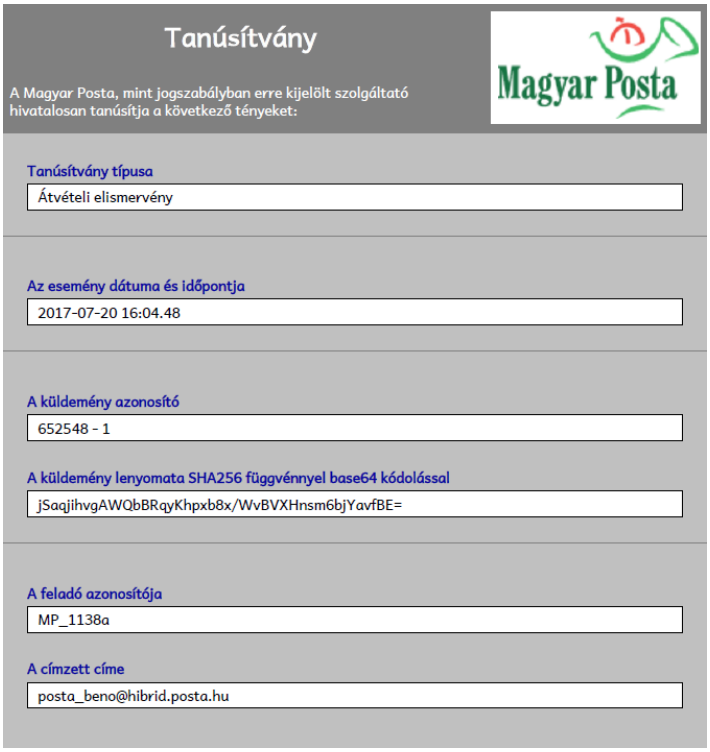
Biztonságos kézbesítés esetén azonban egyrészt van lehetősége a kézbesítést visszautasítani, ha viszont át kívánja venni, akkor először alá kell írnia, vagy az AVDH útján hitelesítenie kell az átvételi elismervényt. Ez a folyamat azonban már minden egyes címzett vonatkozásában önállóan történik. Egy címzett esetében is van lehetőség több értesítendő megjelölésére a szerződésben. Az átvétel megtagadása webautomata használatával történő kommunikáció esetére, annak érdekében, hogy ne kelljen másik csatornát igénybe venni, a rendszer egy speciális nemleges átvételi elismervény letöltésére és aláírására is lehetőséget ad, de ez erősen egy elméleti opció gyakorlati jelentőség nélkül. Hivatali kapun keresztül jelenleg az átvételi protokoll hivatali kapu oldali megvalósítatlansága miatt nincs lehetőség BKSZ küldemények fogadására.

A rendszer BKSZ esetén figyeli, hogy az értesítés kiküldését követően 5 munkanapon belül letöltésre kerül-e a letöltési igazolás. Amennyiben nem, akkor öt munkanap után az értesítés a küldemény érkezéséről a szerződésben szereplő értesítési csatornákon megismétlésre kerül.

Maga az átvételi elismervény egy a Magyar Posta által elektronikusan lebélyegzett és időbélyegzővel ellátott dokumentum. A tervek szerint ezt az aláírást és időbélyegzést törölni fogjuk annak érdekében, hogy az Igénybe vevők az AVDH használatával a legegyszerűbb PDF formátumú aláírt átvételi elismervényt készíthessenek, és azt tölthessék vissza. Az átvételi igazolás változatlanságának (azonosságának) ellenőrzését ekkor a rendszer másképp fogja biztosítani.

Az átvételi elismervény aláírásával a címzett elismeri, hogy a küldeményt átveszi. (Ennek feltöltése és az alább leírt pozitív aláírás ellenőrzés után a biztonságos kézbesítési szolgáltató a küldeményt a címzett rendelkezésére bocsátja – lásd a letöltési igazolásnál leírtakat.) Hagyományos postai szolgáltatás esetében ennek az elismervénynek az aláírása felel meg a tértivevény aláírásának. Az átvételi elismervény egy PDF dokumentum, amely tartalmazza a küldemény és az igazolás a rendszerben kapott azonosítószámát, a küldemény lenyomatát a feladó és a címzett a rendszerben

használt azonosítóját. A PDF dokumentum emellett beágyazva tartalmaz egy XML állományt, amely a küldeménnyel kapcsolatos információkat hordozza ebben a formátumban, és így az igazolás gépi feldolgozásra is alkalmas.



10. ábra: Az átvételi elismervény képi megjelenése

Az átvételi elismervény PDF állományába ágyazott AcceptanceNonAcceptanceCertificate.xml állományban található adatok szerkezete és adattartalma jelentős rokonságot mutat a feladási igazolásnál a 4.2.1 fejezetben már bemutatott adatokkal. Itt is ugyanazokra a sajátosságokra kell figyelemmel lenni. Annyiban egyszerűbb a helyzet, hogy itt a refEvidenceId tag már nem szerepel.

A beágyazott XML állományra vonatkozó séma állományt a 3. sz. melléklet tartalmazza

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:evidence xmlns:ns2="http://selexes.com/hmdacs"
            evidenceId="674544"
            evidenceType="AcceptanceCertificate"
            consignmentId="652548"
            extConsignmentId="1"
            consignmentHash="8d26aa8e286f8005906c146ac8a869c5bf31fd6bc15571e7b26e9b8d86af7c11"
            senderIdentificationMethod="KAU"
            recipientIdentificationMethod="KAU"
            senderAddress="MP_1138a"
            eventTime="2017-07-20T16:04:48.018+02:00">
      <referenceRecipient electronicAddress="posta_beno@hibrid.posta.hu"/>
      <recipients>
            <recipient electronicAddress="posta_beno@hibrid.posta.hu"/>
      </recipients>
</ns2:evidence>
```

Az átvételi igazolásba beágyazott AcceptanceNonAcceptanceCertificate.xml állomány mintája

Ezt az állományt kell aláírva, illetve az AVDH használatával hitelesítve visszatölteni ahhoz, hogy a rendszer letölthetővé tegye a biztonságos kézbesítési szolgáltatás útján küldött eredeti küldeményt. A rendszer ellenőrzi, hogy megfelelő dokumentum és az előzetesen bejelentett aláírások egyike jelenjen meg az átvételi elismervényen. Amennyiben az aláírás nem megfelelő, úgy a rendszer ismételten értesíti a szerződés szerinti címen és csatornán az Igénybe vevőt, hogy számára küldemény érkezett BKSZ útján. (az átvételre nyitva álló időablak pedig változatlanul értelmezett, mintha semmi nem történt volna)
Ahogy már jeleztük az elutasításra vonatkozó NonAcceptanceCertificate csak a webAutomata útján megvalósított kapcsolat esetében és elméleti jelentőséggel értelmezett, azt külön nem is tárgyaljuk. Szerkezete és adattartalma – az <evidenceType> tag értékének kivételével, amely ebben az esetben „NonAcceptanceCertificate” – megegyezik az átvétel elismervénnyel.

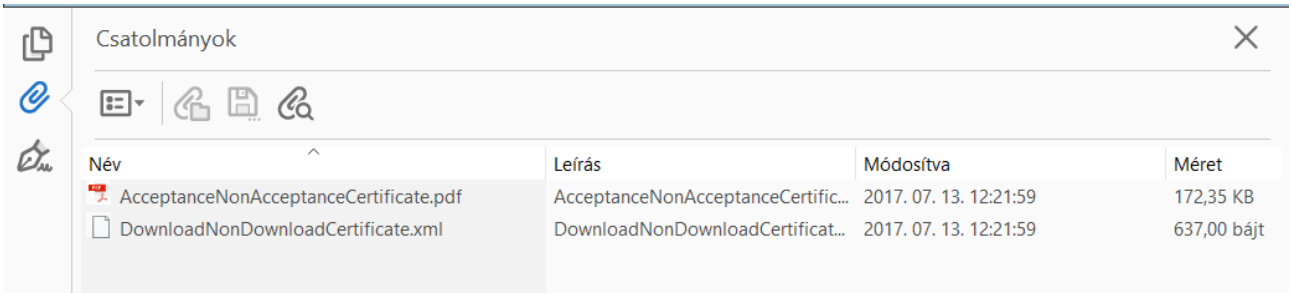
4.2.4 Letöltési igazolás

A letöltési igazolás alapesetben mind a címzett, mind a feladó felé kiküldött dokumentum. A sikeres és sikertelen letöltések igazolása mind szerkezetében, mind adattartalmában lényegesen eltér, ezért azokat külön tárgyaljuk. Értelemszerűen, ennél az igazolás-típusnál, ha esetleg több címzett van, akkor ez a dokumentum is már címzettenként külön folyamatban, a többi címzettől függetlenül jön létre. Ennek megfelelően előfordulhat az is, hogy egy küldeményről párhuzamosan sikeres és sikertelen kézbesítésre vonatkozó igazolás érkezik a feladóhoz (a címzett oldalán természetesen csak vagy sikeres, vagy sikertelen lehet egy adott küldemény kézbesítése).

4.2.4.1 Sikeres letöltés igazolása

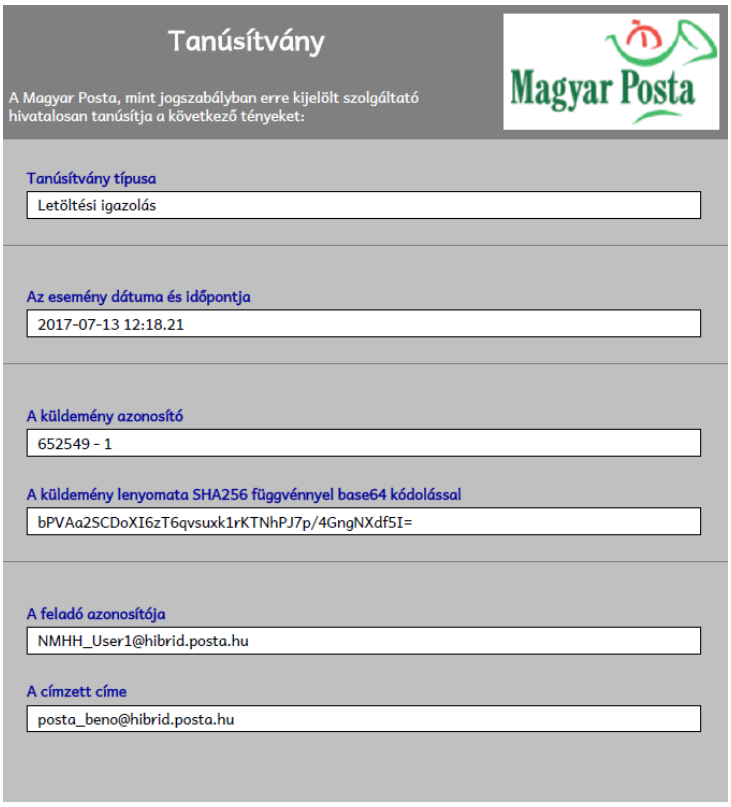
A feladónak és a címzettnek a rendszer a Magyar Posta által elektronikusan lebélyegzett és időbélyegzővel ellátott kézbesítési igazolással igazolja vissza, hogy mint biztonságos kézbesítési szolgáltató, az átvételi elismervény aláírása után a küldeményt a címzett rendelkezésére bocsátotta. Ez a lépés felel meg a hagyományos postai folyamatban a tértivevény visszajuttatásának azzal a különbséggel, hogy itt mind a feladó, mind a címzett kap belőle egy-egy egyaránt hiteles példányt.

Az igazolás egy PDF dokumentum, amely tartalmazza a küldemény és az igazolás a rendszerben kapott azonosítószámát, a küldemény lenyomatát a feladó és a címzett a rendszerben használt azonosítóját. A PDF dokumentum emellett beágyazva tartalmazza egyrészt a címzett által aláírt (vagy AVDH útján hitelesített) átvételi elismervényt teljes egészében (beleértve az eredetileg beleágyazott AcceptanceNonAcceptanceCertificate.xml állományt), valamint egy DownloadNonDownloadCertificate.xml állományt, amely a küldeménnyel kapcsolatos információkat hordozza ebben a formátumban, és így az igazolás gépi feldolgozásra is alkalmas.



11. ábra: A sikeres letöltési igazolás beágyazott állományai az Acrobat Reader-ben megjelenítve

Ez a dokumentum szolgál bizonyítékul, illetve az ellenőrzés alapjául mind a feladó, mind a címzett számára.



12. ábra: A sikeres letöltési igazolás képi megjelentése

A letöltési igazolás aláírt PDF állományába ágyazott DownloadNonDownloadCertificate.xml állományban található adatok szerkezete és adattartalma megegyezik a feladási igazolásnál a 4.2.1 fejezetben már bemutatott adatokkal. Itt is ugyanazokra a sajátosságokra kell figyelemmel lenni. Az adattartalma közel azonos az AcceptanceNonAcceptanceCertificate.xml tartalmával. Az esemény időpontja (eventTime) minden ilyen esetben az, amikor az átvételi elismervény visszaérkezését követően a Szolgáltató az Igénybe vevő rendelkezésére bocsátotta a küldeményt.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:evidence xmlns:ns2="http://selexes.com/hmdacs"
        evidenceId="674413"
        evidenceType="DownloadCertificate"
        consignmentId="652549"
```

```
        extConsignmentId="1"
        consignmentHash="6cf5406b64820e85c8eb34faaafb2ec64d6b2933613c9ee9ff81a780d5dd7f92"
        senderIdentificationMethod="KAU"
        recipientIdentificationMethod="KAU"
        senderAddress="NMHH_User1@hibrid.posta.hu"
        eventTime="2017-07-13T12:18:21.919+02:00">
    <referenceRecipient electronicAddress="posta_beno@hibrid.posta.hu"/>
    <recipients>
        <recipient electronicAddress="posta_beno@hibrid.posta.hu"/>
    </recipients>
</ns2:evidence>
```

A letöltési igazolásba beágyazott sikeres DownloadNonDownloadCertificate.xml állomány mintája

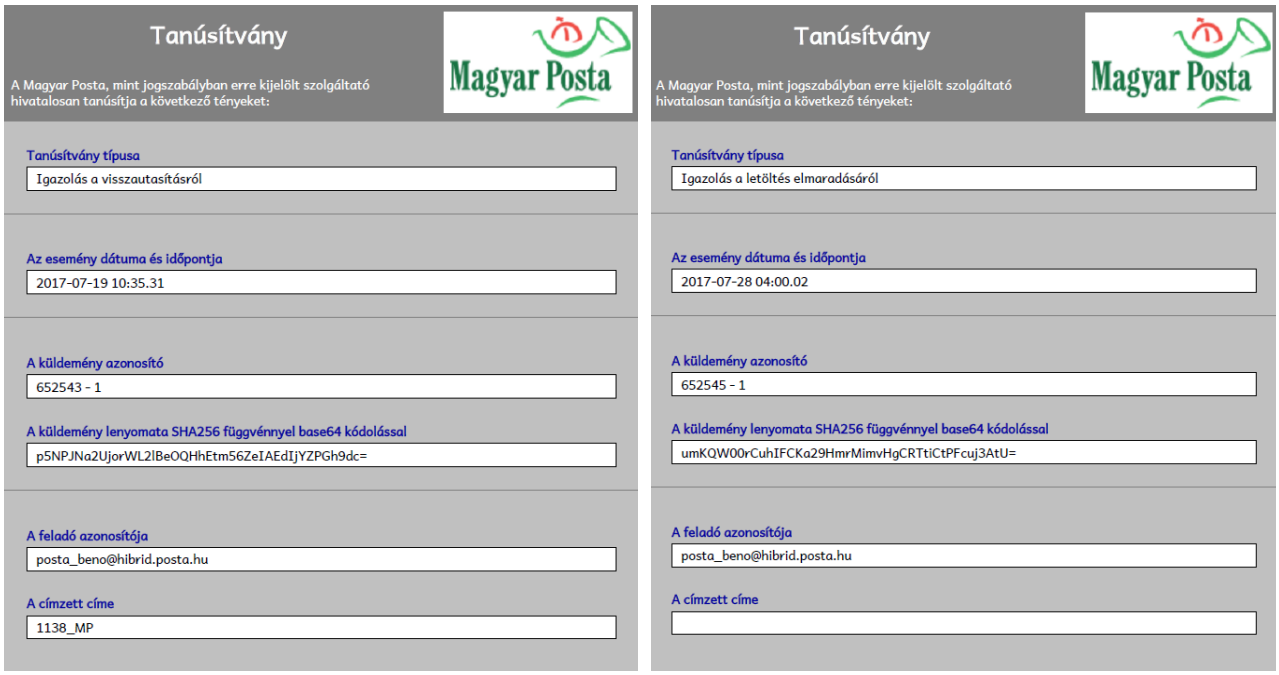
Egy kivétel van a főszabályhoz képest. Amennyiben egy webAutomatán keresztüli megtagadás történne (ami, ahogy jeleztük erősen elméleti lehetőség) akkor a sikeres kézbesítési igazolás szerkezetével megegyezően készül el az igazolás, de az igazolás típusa (evidenceType) "NonDownloadCertificate" lesz.

4.2.4.2 Letöltés elmaradásának igazolása

A feladónak a Magyar Posta által elektronikusan lebélyegzett és időbélyegzővel ellátott negatív kézbesítési igazolással igazolja vissza, hogy a biztonságos kézbesítési szolgáltatás teljesítése nem volt sikeres. A hagyományos postai folyamatban a tértivevény visszajuttatásának folyamatától ez annyiban tér el, hogy itt minden esetben a tanúsított esemény időpontja egybeesik a kézbesítési vélelem beállásának időpontjával.

- ha az adott címzett által megjelölt összes értesítési címről visszapattan az értesítés, illetve a kézbesítési folyamat során szűnik meg a szerződés, akkor ennek időpontját tartalmazza, (ez felel meg a hagyományos kézbesítésben a címzett ismeretlen, elköltözött, elhunyt eseteknek)
- ha a címzett megtagadja a küldemény átvételét, akkor ennek az időpontja szerepel az igazolásban
- ha a címzett a második értesítés ellenére sem veszi át a küldeményt, akkor a jogszabályi határidő lejárata szerepel az igazolásban.

Az igazolás egy PDF dokumentum, amely tartalmazza a küldemény és az igazolás a rendszerben kapott azonosítószámát, a küldemény lenyomatát a feladó és a címzett a rendszerben használt azonosítóját. Itt tehát a biztonságos kézbesítési szolgáltató egyoldalú igazolása igazolja az címzett közreműködésének hiányában az eseményt. A PDF dokumentum emellett beágyazva tartalmaz egy DownloadNonDownloadCertificate.xml állományt, amely a küldeménnyel kapcsolatos információkat hordozza ebben a formátumban, és így az igazolás gépi feldolgozásra is alkalmas.



13. ábra: A kézbesítés meghiúsulásának különböző eseteiben kiadott igazolások képei

A letöltés elmaradásának igazolása esetén az aláírt PDF állományába ágyazott DownloadNonDownloadCertificate.xml állományban található adatok szerkezete és adattartalma megegyezik a feladási igazolásnál a 4.2.1 fejezetben már bemutatott adatokkal. Itt is ugyanazokra a sajátosságokra kell figyelemmel lenni. Az adattartalma lényegében megegyezik az AcceptanceNonAcceptanceCertificate.xml tartalmával. Értelemszerűen itt az igazolás típusa (evidenceType) "NonDownloadCertificate" lesz.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:evidence xmlns:ns2="http://selexes.com/hmdacs"
        evidenceId="674578"
        evidenceType="NonDownloadCertificate"
        consignmentId="652545"
        refEvidenceId="0"
        extConsignmentId="1"
        consignmentHash="ba62905b4d2b0ae84814229adbd1e6acc8a6bc7802453b620ad3c572e8f702d5"
        senderIdentificationMethod="KAU"
        recipientIdentificationMethod="KAU"
        senderAddress="posta_beno@hibrid.posta.hu"
        eventTime="2017-07-28T04:00:02.230+02:00">
    <recipients>
        <recipient electronicAddress="MP_1138a"/>
    </recipients>
</ns2:evidence>
```

A letöltési igazolásba beágyazott nemleges DownloadNonDownloadCertificate.xml állomány mintája

Ezek tehát az igazolások, azok adattartalma és a kibocsátásuk körülményei. Ezek után tekintsük át a folyamat alaplépéseit, hogy utána a tömeges kommunikáció eseteit vizsgálhassuk

4.3 Küldemények küldése, fogadása, megtekintése, ellenőrzése

4.3.1 Elektronikus küldemény feladásának folyamata

A KSZ és BKSZ szolgáltatás igénybevétele során a dokumentumok hagyományos továbbítási logikájához illeszkedő módon biztosítja az azonosított, ügyfelek közötti kézbesítést, a címezhetőséget, továbbá az ügyfelek eltérő adatvédelmi igényéhez igazodva nyújt választási lehetőséget a kézbesítési módok közül, garantálva a szükséges bizonyító erőt.

Web szervizen keresztül kétfajta küldemény feladására van lehetőség:

- Kézbesítési szolgáltatás
- Biztonságos kézbesítési szolgáltatás, a biztonságos kézbesítési szolgáltatást azonban jelenleg a visszaigazolási folyamat támogatásának hiánya miatt nem lehet hivatali kapuval rendelkező címzettek felé megvalósítani

A hivatali kapun keresztüli a Magyar Postával, mint BKSZ szolgáltatóval szerződéses kapcsolatban álló ügyfelek felé BKSZ küldemények továbbítása még fejlesztési fázisban van, a fejlesztés befejezése utána dokumentum kiegészítésre kerül.

A rendszeren keresztül továbbított levelek tartalmát a Magyar Posta kizárólag vírus illetve spamszűrés szempontjából vizsgálja, küldeményre vonatkozó érdemi korlátozás kizárólag a csatolmányok összméretére vonatkozóan van, melyek küldeményenként nem lépheti túl az 500Mb-ot. Az egyéb technikai előfeltételeket a 3. fejezet tárgyalja, itt azokat nem ismételjük.

4.3.1.1 Elektronikus dokumentum küldése a webAutomata használatával

A webAutomata működésének részletei nem képezik a jelen felhasználói kézikönyv tárgyát. Egy rövid összefoglalás található a működéséről az 5. fejezetben, illetve az 1. és 2. számú függelékekben. Itt most csak közvetlenül a küldemény elküldésével kapcsolatos lépéseket mutatjuk csak be.

Amint arról már volt szó, a webAutomata egy webszerviz, amely különböző rögzített szerkezetű adatcsomagokkal valósítja meg a küldemények elküldéséhez, fogadásához, illetve a bizonyítékok kezeléséhez kommunikációt a SOAP specifikáció 1.1. verziója szerinti üzenetformátumokkal. Ugyanakkor annak érdekében, hogy az adatcsomag-típusok száma se legyen túl nagy a SOAP üzenetekben található adatcsomagokat leíró XML sémák elég általánosak, nem minden esetben kell minden egyes a sémában szereplő adatot használni.

A kommunikáció hitelességét a küldőtől a hibrid kézbesítési és konverziós rendszer irányában a SOAP üzenetek fejének aláírásával biztosítjuk, fordított irányban, mivel maguk az igazolások, illetve üzenetek önmagukban is aláírtak és időbélyeggel ellátottak, nem használunk ilyen jelentős számításigényű megoldást. Ennek megfelelően a küldött és fogadott üzenetek struktúrája eltér. A megvalósított szolgáltatáscsomagban minden üzenet egy kérésből és válaszból áll. (de nem mindegyik tartalmaz return elemet). Ezeket az adatcsomagokat, azok szerkezetét és használatát fogjuk a továbbiakban tárgyalni.

4.3.1.1.1 Üzenet továbbítása a sendMessage parancs használatával

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	sendMessage	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
sendMessage	Összetett	A szerződéshez tartozó egy üzenet továbbítása a hibrid kézbesítési és konverziós rendszerbe a feldolgozás jellemzőinek meghatározásával

6. táblázat: A sendMessage parancs alapvető jellemzői

A <sendMessage> összetett adatcsomag két elemből áll:

Elem név	**Típus**	**Leírás**
contractId	Karaktersorozat	A küldő szerződésének azonosítója
message	Összetett	A teljes továbbítandó üzenet az leíró adataival

7. táblázat: A sendMessage adatcsomag szerkezeti elemei

A <message> tag maga is egy összetett elem, amelynek elemei között vannak kötött értékkészletűek és összetett elem is. Mivel a <message>, mint elem több parancsban, illetve adatcsomagban is előfordul, vannak olyan elemeit, amelyek csak bizonyos esetekben kapnak értéket (a séma önmagában minden elem kitöltetlenségét vagy elhagyását megengedi, azonban bizonyos elemek hiánya a feladat természetéből adódóan a megvalósulás ellehetetlenüléséhez vezet.)

Elem név	**Típus**	**Leírás**
uid	Karaktersorozat	Az állomány egyedi azonosítója
subject	Karaktersorozat	A küldemény tárgya
requestId	Karaktersorozat	Az igénybe vevő által adott azonosító
recipients	Karaktersorozat	Címzett
notificationEMail	Karaktersorozat	Az e-mail értesítés címe
messageType	Kötött értékkészlet egy eleme	A küldemények típusait felsoroló lista eleme (lásd 28. táblázat)
messageDateTime	Dátum-idő	A küldeménnyel kapcsolatos esemény időpontja
deliveryType	Kötött értékkészlet egy eleme	A küldés (szolgáltatás) típusait felsoroló lista eleme (lásd 27. táblázat)

consignmentId	Karaktersorozat	A küldemény a rendszerben kapott azonosítója
body	Karaktersorozat	A küldemény törzse (ha az nem csatolmány)
attachments	Összetett	Csatolmányok, azok jellemzőivel

8. táblázat: A message tag szerkezete

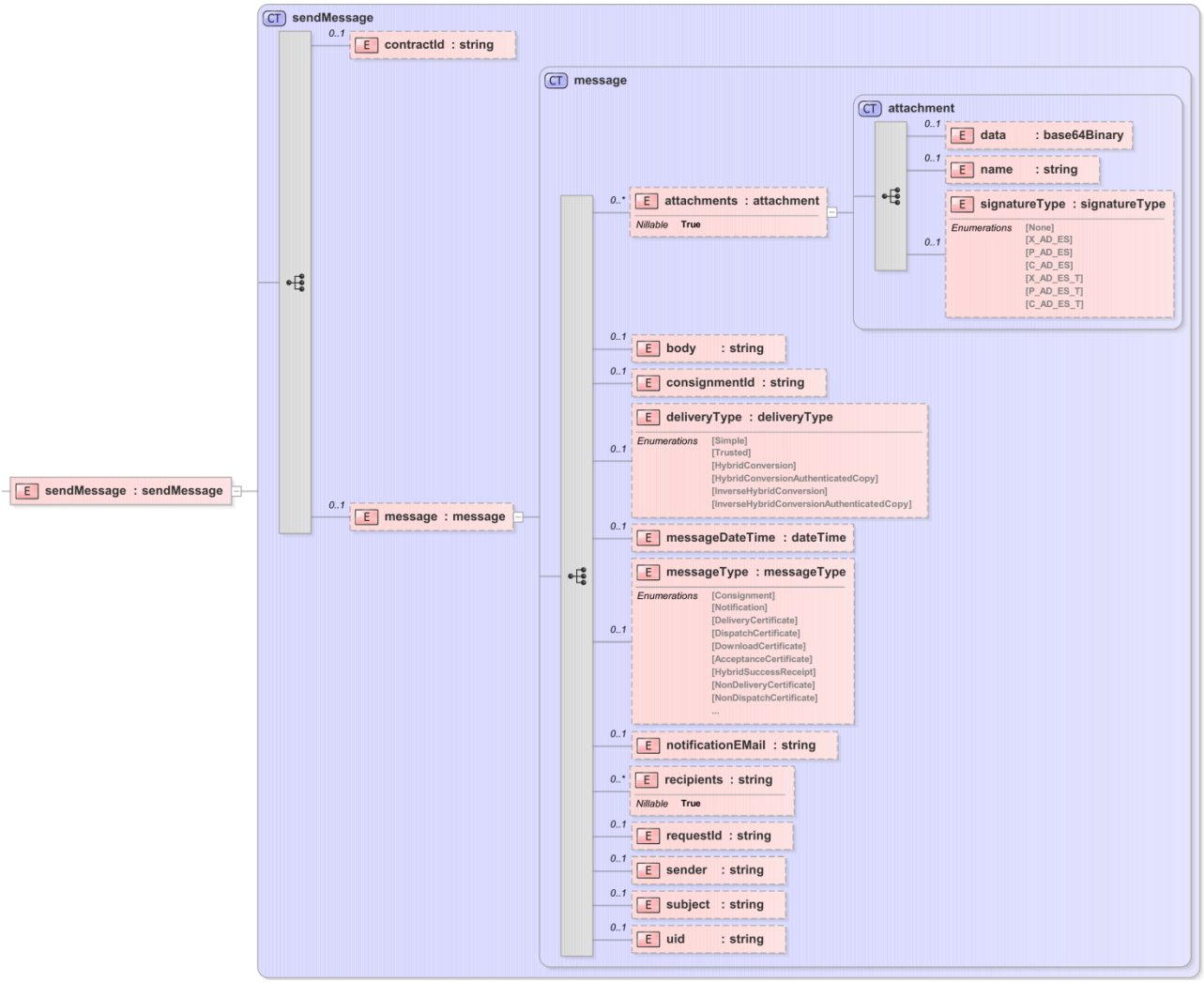
Az <attachments> tag (amely egyes esetekben el is maradhat) többször is ismételhető a <message> tagban. Ez is egy összetett tag, amely tartalmazza az egyes csatolmányok elemeit, jellemzőit.

Elem név	Típus	Leírás
data	Karaktersorozat	Maga az állomány base64 kódolással
name	Karaktersorozat	Az állomány neve
signatureType	Kötött értékkészlet egy eleme	A dokumentum aláírásának típusát azonosítja (lásd 29. táblázat)

9. táblázat: Az attachments tag szerkezete

A wsdl-ben meghatározott kötött értékkészletű állományok felvehető értékeit azok értelmezésével a 4. sz. függelék tartalmazza.

A fenti struktúrában megadott elemek közül a küldemény típusának megfelelő adatokat kell megadni. Esetünkben a kimenő küldemény <messageType> értéke csak „Consignment” lehet. Mivel jelen esetben kézbesítési szolgáltatásról, illetve biztonságos kézbesítési szolgáltatásról lehet csak szó, ennek megfelelően a <deliveryType> értéke csak „Simple” vagy „Trusted” lehet. A csatolmányok adatait értelemszerűen ki kell tölteni, éppúgy, mint a feladó és címzett adatait, hiszen ezek hiányában a küldés nem értelmezhető. A tárgy és törzs, az értesítési e-mail cím, valamint a felhasználó által adott azonosító megadása értelemszerűen a feladó igényétől függ. Az esemény időpontja, a küldemény azonosítószáma és az egyedi azonosító adatok viszont akár megadásra kerülnek, akár nem, a rendszer új értékeket fog megadni ezekre.



14. ábra: A sendMessage adatcsomag szerkezete

Ennek megfelelően a *sendMessage* SOAP üzenet szerkezete a következőképpen épül fel:

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
            <ds:Reference URI="#SBody">
               <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha256"/>
               <ds:DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</ds:DigestValue>
            </ds:Reference>
         </ds:SignedInfo>
         <ds:SignatureValue>
            mnDG3aeahjRqQLx19PKJUuRT/85hTtFxMWTmL13CF70KH1i7wlueUQJhRwD1EqH6XjxjJfTysexp
            tkGMw9fHRKnqQxuJH8RcNKxxUWQTKJjolBBc/971ZON9WAekTFkq9WdPwTOuDDQN5bN4yYPA7ZRC
            66F57Lz/GFRJGucV+GAjbm0xSdB89ZsD0g6p6uumJTMRbsuBpn5qbngzfwKd6dQ9B5kI5/zY+ONa
            oJsbDtCPSPJdEzCufv6K9hMsvKsKEU/A/Kg4qGqAJa6+8G/kCjXkpJSpwCCYd4r8sgxoYUV/S0+O
            E38VLHcxSURvfjVb11YWCCWmwYOE+T3gcIDZBw==
         </ds:SignatureValue>
         <ds:KeyInfo>
            <ds:X509Data>
               <ds:X509Certificate>
```

```
                MIIHHTCCBgWgAwIBAgIIK2IB5TTiq3owDQYJKoZIhvcNAQELBQAwgaYxCzAJBgNVBAYTAkhVMREw
                DwYDVQQHDAhCdWRhcGVzdDE8MDoGA1UECgwzTklTWiBOZW16ZXRpIEluZm9rb21tdW5pa8OhY2nD
                <!-- itt folytatódnak a tanúsítvány adatai base64 kódolással-->
                lszAlk+6Ll8wd9BySS9JjjvdjtJijqb9LF7jhaM44j6mVNQg68MYblqfTuVbhPXCprLjZjbUDJKYU
                0pfFyRd8UwoUX6j5DcZp0HwWNm+vuqKTZFcolXw845omExs5/jFUnh9mjW8Ewv2DWCDCmpZ/VXAxfQ==
              </ds:X509Certificate>
            </ds:X509Data>
          </ds:KeyInfo>
        </ds:Signature>
      </SOAP-ENV:Header>
      <S:Body Id="SBody">
        <ns2:sendMessage xmlns:ns2="http://selexes.com/hmdacs">
          <contractId>4063</contractId>
          <message>
            <attachments>
              <data>
                JVBERi0xLjQKJeLjz9MKMyAwIG9iago8PC9Db2xvclNwYWNlL0RldmljZUdyYXkvU3VidHlwZS9JbWFnZS9IZWlnaHQg
                <!-- itt folytatódik a base 64 kódolt szöveg -->
                Nz5dCi9Sb290IDQwIDAgUgovU2l6ZSA0NwovUHJldiAyMTAzOTYKPj4Kc3RhcnR4cmVmCjI3NzY0NwolJUVPRgo=
              </data>
              <name>proba118_00_9_00986_01_2017.pdf</name>
              <signatureType>P_AD_ES_T</signatureType>
            </attachments>
            <deliveryType>Trusted</deliveryType>
            <messageType>Consignment</messageType>
            <requestId>ST000000000000005</requestId>
          </message>
        </ns2:sendMessage>
      </S:Body>
    </S:Envelope>
```

A sendMessage SOAP üzenet szerkezeti példája küldemény biztonságos kézbesítési szolgáltatás útján történő küldése esetére

4.3.1.1.2 Válasz az üzenet továbbítására a sendMessageResponse

A rendszer az üzenet elküldésére a *sendMessageResponse* adatcsomaggal válaszol, ami alapesetben a küldemény azonosítóját (consignmentID) tartalmazza, vagy egy esetleges hibaüzenetet.

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
sendMessageResponse	Összetett	egy return elemet tartalmazhat
return	Karaktersorozat	Az elküldött küldemény azonosítója, kivételesen a fogadó rendszer esetleges hibaüzenete

10. táblázat: A sendMessageResponse adatcsomag elemei

A struktúrája igen egyszerű:

15. ábra: A sendMessageResponse adatcsomag szerkezete

Ez tehát SOAP üzenet formájában is igen egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:sendMessageResponse xmlns:ns2="http://selexes.com/hmdacs">
         <return>itt adja vissza alapesetben a küldemény azonosítóját (consignmentID), esetleg a hibaüzenetet</return>
      </ns2:sendMessageResponse>
   </soap:Body>
</soap:Envelope>
```

A sendMessageResponse SOAP üzenet egy szerkezeti példája

A fogadó rendszernek itt alapesetben az elküldött küldemény a rendszer által adott azonosítóját kell megkapnia, a továbbiakban ezzel hivatkozhat rá. Emellett az esetleges kivételekből származó hibaüzenetek kezelésére is fel kell készülnie. Szerencsére ezek tulajdonképpen csak a rendszer összehangolásának időszakában relevánsak, hiszen a helyesen megformázott üzenetek esetében nem ezen a szinten kell a hibajelzéseknek előállniuk, azok a különböző igazolások, illetve a felhasználóknak küldött rendszerüzenetek formájában jelentkeznek.

A sendMessage parancsra kapott válasznál az „IllegalMessageTypeException” fordul elő. Ez a gyakorlatban azt jelenti, hogy nem „Consignment” volt az elküldött üzenet <messageType> elemének értéke. Más küldeménytípus a küldő oldalán nem szerepelhet a küldött adatcsomagban. A többi, a 28. táblázatban szereplő adatcsomag csak fogadási oldalon a getMessage parancsnál fordulhat elő.

A kivételekre a küldő rendszernek kell valamilyen módon reagálnia. Értelemszerűen bizonyítékokban vagy értesítésekben szereplő <notification> és <error> üzenetek kezelése jellemzően nem gépi feldolgozást kíván, hanem humán kezelők informálását szolgálja.

A fentiek alapján látható, hogy egy küldemény elküldését a webszerviz használatával is teljes értékűen lehet biztosítani. A feladó (később majd fogadó) csatlakozó rendszernek, rendszereknek az itt megadott sémáknak megfelelő tartalommal kell hozzá előállítani adatcsomagokat.

4.3.1.2 Elektronikus dokumentum küldése hivatali kapun keresztül

Amint azt a 3.8.2 fejezetben már leírtuk a hivatali kapun keresztüli kommunikáció KRX formátumú állományok használatával valósul meg. Ennek megfelelően ahhoz, hogy egy a Magyar Posta kézbesítési/biztonságos kézbesítési rendszerében regisztrált címzettnek elektronikus üzenetet lehessen küldeni, a címzésre vonatkozó információkat megfelelően el kell helyezni a KRX állományban.
Jelenleg a címzési információk hordozására egy a message.properties állomány szolgál, amelyet a KRX állományon belül a Metalayer könyvtárban kell elhelyezni. (a KRX állomány szerkezetét a 5. ábra mutatja be). Ezt egységesíteni kívánjuk a jövőben a kuldemeny_meta.xml adataival, de jelenleg még külön kell megcímezni, amennyiben valaki hivatali kapuról akar BKSZ/KSZ üzenetet küldeni. A címzés állomány egy példája alább látható.

```
sender_contract=3001
sender_address=cadastre@hmdacs.posta.hu
request_id=Prova TED
service=T
recipient_addresses=police@hmdacs.posta.hu,education_department@hmdacs.posta.hu
```

Egy minta message.properties állomány

Mint látható. az adatállomány formátuma az XML-nél is egyszerűbb, az egyes értékek kötött sorrendű neve után egyenlőségjellel következik az adott érték. Az egyes címzési tulajdonságokat és felvehető értékeiket az alábbi táblázat tartalmazza:

Az adat megnevezése	Az adat típusa	A használat jellege	Az adat leírása, jellemzői
sender_contract	integer	kötelező	A küldő szerződésének azonosítószáma a postai rendszerben
sender_address	BKSZ cím	kötelező	A feladó címe a postai rendszerben. Elvileg válaszcím lenne, de jelenleg ez még nem működik.
request_id	karaktersor	opcionális	A küldő által adott azonosító
service	1 karakter	kötelező	A két lehetséges szolgáltatásnak megfelelően T=Biztonságos kézbesítési szolgáltatás D=Kézbesítési szolgáltatás
recipient_addresses	BKSZ cím	kötelező	A címzett vagy címzettek címe a postai rendszerben. Az egyes címzetteket vesszővel kell elválasztani

11. táblázat: A message.properties állomány elemei és jellemzőik

A fenti módon megcímzett üzenet elküldését követően a rendszer szintén KRX csomagban megküldi a felvételi igazolást (feladóvevényt), majd a küldemény jellegének megfelelően a kézbesítési vagy letöltési igazolást.

Mindez azt jelzi, hogy hivatali kapuról teljes értékűen lehet a postai BKSZ/KSZ szerződött partnereinek megfelelő bizonyító erővel dokumentumokat küldeni. A fordított irány sajnos nem működik, mivel a hivatali kapun, illetve szélesebben a NISZ által működtetett biztonságos kézbesítési szolgáltatásban nincs lehetőség az átvételi elismervény aláírattatására.

4.3.2 Az igazolások keresése és megismerése a küldő oldalán

Egy küldemény elindítását követően meg kell győződnünk arról, hogy a rendszer valóban befogadta-e a küldeményt, hiszen a küldés sikeressége még nem jelenti azt, hogy a rendszer valóban be is fogadta a küldeményünket. A küldemény sikeres feladásának bizonyítéka a 4.2.1 fejezetben már ismertetett tartalmú feladási igazolás vagy feladóvevény. Az esetleges sikertelen feladást tanúsító bizonyíték elérésének eljárásrendje teljes mértékbe megegyezik a sikeressel, kizárólag az igazolás tartalma tér el. Sikertelen feladás eseté a folyamat itt le is zárul.

Amennyiben a feladó pozitív feladóvevényt kapott, számára ezek után már csak a kézbesítésről (sikeres vagy sikertelen) szóló igazolás letöltése marad feladatként, amit ugyanúgy kell végrehajtani,

mint az előző dokumentumnál. Különbség itt is csak a letöltendő dokumentum tartalmában van, magának a letöltésnek az eljárásrendje azonos.

4.3.2.1 A küldő számára biztosított igazolások kezelése webAutomata használatával

A webAutomata egységes logikával kezeli a feladónak, illetve a címzettnek küldött különböző típusú üzenetek lekérését, mivel maga a fogadás független attól, hogy a fogadott üzenet egy érdemi küldemény vagy egy igazolás. Ez az üzenet-válasz pár a *getMessage – getMessageResponse* pár. Ez az egy üzenetpár azonban egy nem megbízható környezetben nem lenne elégséges a küldemény vagy üzenet biztonságos eljuttatásához, ezért egy külön eljárás került implementálásra, amely az üzenet „elengedését” biztosítja, azaz amivel az adott üzenet fogadója visszaigazolja a küldőnek, hogy az átvitel teljesedésbe ment, az üzenetet már nem kell rendelkezésre tartania. Ez a *releaseMessage – releaseMessageResponse* pár. A következőkben ezek használatával bemutatjuk, hogyan kérdezhet le akár a küldő, akár a fogadó fél üzeneteket. (ennek megfelelően a címzett oldali üzenetek fogadásánál már csak hivatkozni fogjuk az üzenettípusokat, és csak a specialitásokra mutatunk rá).

4.3.2.1.1 Üzenet lekérdezése getMessage

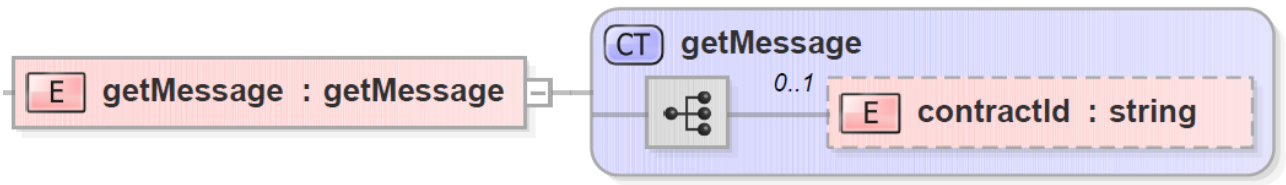
Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	getMessage	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getMessage	Összetett	A szerződéshez tartozó egy üzenet lekérdezése a hibrid kézbesítési és konverziós rendszerből

12. táblázat: getMessage parancs alapvető jellemzői

Maga a <getMessage> kérés rendkívül egyszerű, mindössze a kérdező, fogadni kívánó fél szerződésének azonosítóját tartalmazza

Elem név	**Típus**	**Leírás**
contractId	Karaktersorozat	A küldő szerződésének azonosítója

13. táblázat: A getMessage tag szerkezete



16. ábra: A getMessage adatcsomag szerkezete

Ennek megfelelően egyszerű a kérdés tartalmának XML sémája is

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
           xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="getMessage" type="tns:getMessage"/>
    <xs:complexType name="getMessage">
        <xs:sequence>
            <xs:element minOccurs="0" name="contractId" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

A getMessage adatcsomag XML sémája

A SOAP üzenet a biztonsági követelmények miatt összetettebb, de a tartalom itt is igen egyszerű. Itt már látszik, hogy ennek a kommunikációs modellnek az a gyengéje, hogy a hasznos tartalomnál az üzenet biztonságos továbbítását szolgáló üzenetrész jelentősen több lehet.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
        <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
        <Reference URI="#Body">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          </Transforms>
          <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
          <DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
        </Reference>
      </SignedInfo>
      <SignatureValue>
        MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
        ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
        cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
        B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
        M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
      </SignatureValue>
      <KeyInfo>
        <X509Data>
          <X509SubjectName>
            CN=Marco Confalonieri,O=Progesi,L=Genova,ST=Genova,C=IT,
            1.2.840.113549.1.9.1=#160d6d6172636f40746573742e6974
          </X509SubjectName>
          <X509Certificate>
            MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
            dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
            <!-- itt folytatódik a tanúsítvány tartalma base 64 kódolással -->
            nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
            RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
          </X509Certificate>
        </X509Data>
      </KeyInfo>
    </Signature>
  </S:Header>
  <S:Body Id="Body">
    <ns2:getMessage xmlns:ns2="http://selexes.com/hmdacs">
      <contractId>3047</contractId>
    </ns2:getMessage>
  </S:Body>
</S:Envelope>
```

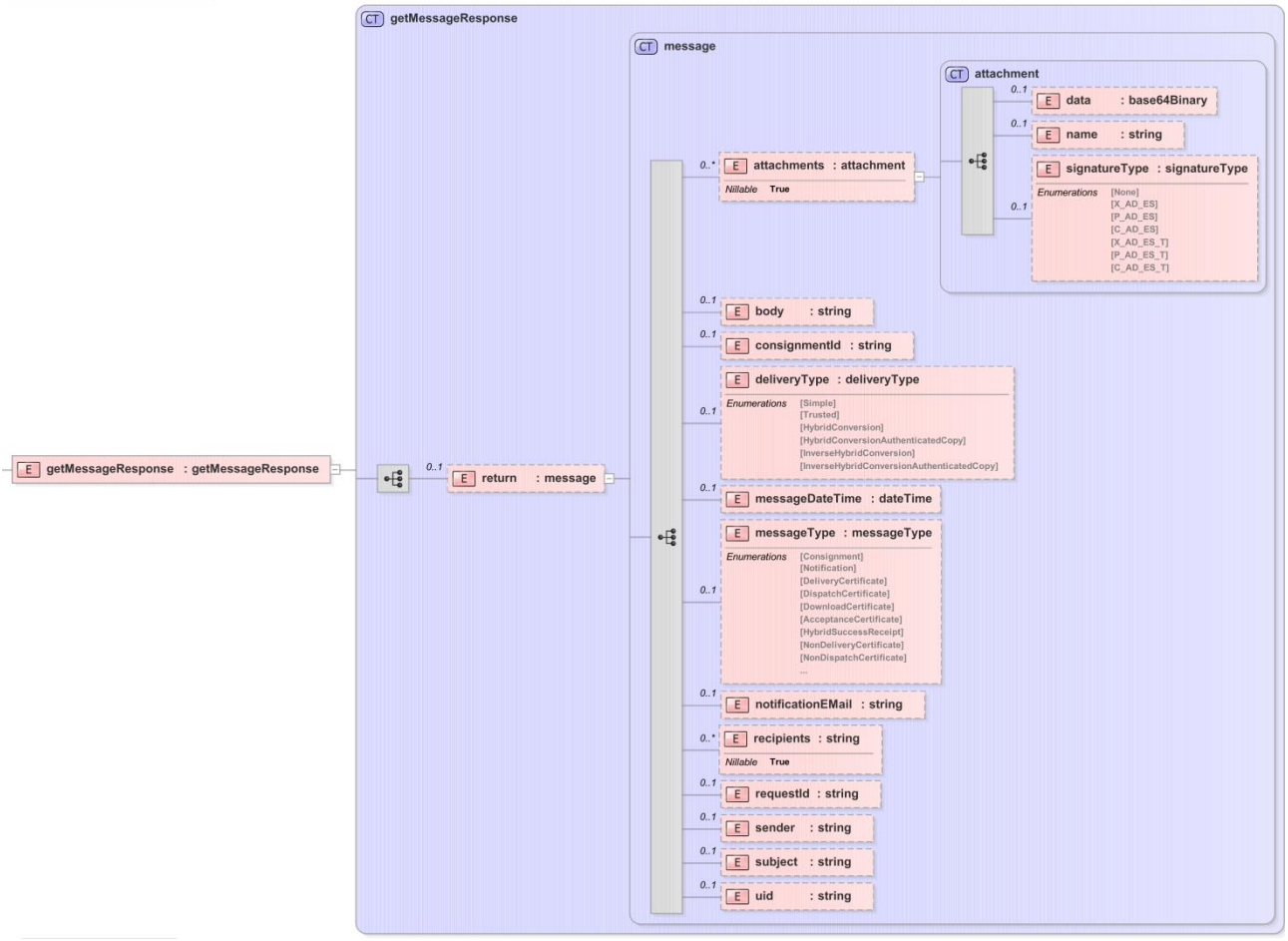
A getMessage SOAP üzenetének szerkezete

4.3.2.1.2 Válasz az üzenet lekérdezésére getMessageResponse

A rendszer a lekérdezésre a <getMessageResponse> üzenettel válaszol, ami a feladat általános jellegéből adódóan, hogy minden, a rendszer által akár a küldő, akár a címzett számára elérhetővé tett üzenetet át kell tudnia vinni, eléggé összetett. A válasz jellegből adódóan a <getMessageResponse> összetett elem egy return összetett típust tartalmaz, amelyben egy (és csak egy) már korábban, a 4.3.1.1.1 fejezetben és a 8. táblázatban ismertetett szerkezetű <message> összetett elem található. Amennyiben nincs visszaadható üzenet a <getMessageResponse> válaszüzenet üres. Ez jelzi, hogy a rendszer várhat az ismételt lekérdezéssel. Az ismételt lekérdezés ütemezése itt értelemszerűen a kérdező oldal feladata.

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getMessageResponse	Összetett	A válasz tartalmát hordozó összetett elem
return	Összetett	A hibrid kézbesítési és konverziós rendszer válaszüzenete
message	Összetett	A 8. táblázatban részletezett tartalommal

14. táblázat: A getMessageResponse adatcsomag szerkezeti elemei



17. ábra: A getMessageResponse üzenet adatainak szerkezete

Itt már a <message> összetett adatcsomag valamennyi, az adott üzenettípus esetén értelmezhető tagja értéket kap, és azokat a fogadó fél (esetünkben a küldő) fel is tudja használni a küldeménye sorsának követéséhez.

A küldő fél alapesetben a getMessage parancsra válaszként először egy „DispatchCertificate” <messageType>-pal jellemzett küldeményt kap. Ez jelzi a küldemény sikeres befogadását. Amennyiben a küldemény esetleg olyan hibában szenvedett, amely már a küldemény további feldolgozásra való befogadását is meghiúsította (vírusfertőzés, nem megfelelő kódolás, struktúra), akkor az elsőként (és az adott küldeményre vonatkozóan egyben utolsó) kapott üzenet típusa „NonDispatchCertificate” lesz.

Itt hívjuk fel a figyelmet arra, hogy az üzenet típusának meghatározása minden esetben a <messageType> tag által történik. Így például minden a hibrid kézbesítési és konverziós rendszer által kiadott igazolás neve az <attachments> tagon belül „Certificate.pdf”. A „Certificate.pdf” maga beágyazottan már az igazolás jellegének megfelelő megnevezésű XML állományt tartalmaz. Azt azonban tudni kell, hogy a beágyazott XML állományok megnevezése nem minden esetben egyezik meg a <messageType> tag értékével. A két információ között ugyanakkor kölcsönösen egyértelmű a megfeleltetés.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="getMessageResponse" type="tns:getMessageResponse"/>
   <xs:complexType name="getMessageResponse">
      <xs:sequence>
         <xs:element minOccurs="0" name="return" type="tns:message"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="message">
      <xs:sequence>
         <xs:element minOccurs="0" maxOccurs="unbounded" name="attachments" nillable="true" type="tns:attachment"/>
         <xs:element minOccurs="0" name="body" type="xs:string"/>
         <xs:element minOccurs="0" name="consignmentId" type="xs:string"/>
         <xs:element minOccurs="0" name="deliveryType" type="tns:deliveryType"/>
         <xs:element minOccurs="0" name="messageDateTime" type="xs:dateTime"/>
         <xs:element minOccurs="0" name="messageType" type="tns:messageType"/>
         <xs:element minOccurs="0" name="notificationEMail" type="xs:string"/>
         <xs:element minOccurs="0" maxOccurs="unbounded" name="recipients" nillable="true" type="xs:string"/>
         <xs:element minOccurs="0" name="requestId" type="xs:string"/>
         <xs:element minOccurs="0" name="sender" type="xs:string"/>
         <xs:element minOccurs="0" name="subject" type="xs:string"/>
         <xs:element minOccurs="0" name="uid" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="attachment">
      <xs:sequence>
         <xs:element minOccurs="0" name="data" type="xs:base64Binary"/>
         <xs:element minOccurs="0" name="name" type="xs:string"/>
         <xs:element minOccurs="0" name="signatureType" type="tns:signatureType"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="messageType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Consignment"/>
         <xs:enumeration value="Notification"/>
         <xs:enumeration value="DeliveryCertificate"/>
         <xs:enumeration value="DispatchCertificate"/>
         <xs:enumeration value="DownloadCertificate"/>
         <xs:enumeration value="AcceptanceCertificate"/>
         <xs:enumeration value="HybridSuccessReceipt"/>
         <xs:enumeration value="NonDeliveryCertificate"/>
         <xs:enumeration value="NonDispatchCertificate"/>
         <xs:enumeration value="NonDownloadCertificate"/>
         <xs:enumeration value="NonAcceptanceCertificate"/>
         <xs:enumeration value="HybridFailureReceipt"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="signatureType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="None"/>
         <xs:enumeration value="X_AD_ES"/>
         <xs:enumeration value="P_AD_ES"/>
         <xs:enumeration value="C_AD_ES"/>
         <xs:enumeration value="X_AD_ES_T"/>
         <xs:enumeration value="P_AD_ES_T"/>
         <xs:enumeration value="C_AD_ES_T"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="deliveryType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Simple"/>
         <xs:enumeration value="Trusted"/>
         <xs:enumeration value="HybridConversion"/>
         <xs:enumeration value="HybridConversionAuthenticatedCopy"/>
         <xs:enumeration value="InverseHybridConversion"/>
         <xs:enumeration value="InverseHybridConversionAuthenticatedCopy"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

A getMessageResponse XML sémája

A sikeres felvételt követően egy későbbi lekérdezés eredménye a küldemény továbbítására igénybe vett szolgáltatásnak megfelelően egy „DeliveryCertificate” vagy egy „DownloadCertificate” <messageType>-pal rendelkező küldemény lehet, amennyiben a küldemény továbbítása sikeres volt. A sikertelen továbbítás megfelelő esetei a "NonDeliveryCertificate", illetve "NonDownloadCertificate" elküldését indítják a küldő felé. A küldő ezeket az üzeneteket minden esetben a rendszer által adott <consignmentId> értéke alapján tudja egymáshoz, illetve a saját nyilvántartásához rendelni. Segíthet még a szintén minden esetben továbbított <requestId> tag is, amely a küldő rendszer azonosítóját is hozzáférhetővé teszi.

Bár ez az üzenet-szerkezet elég sok adminisztrációs feladatot hárít a küldő-fogadó rendszerre, a kompakt üzenetcsomagok miatt van lehetőség többszálas kezelésre is, a szálak nem zavarják egymást.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <soap:Body>
        <ns2:getMessageResponse xmlns:ns2="http://selexes.com/hmdacs">
            <return>
                <attachments>
                    <data>JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9JbnRlbnQvUGVyY2VwdHVhbC9EZWNvZGVQYXJtczw8
                        <!-- itt folyatódik magának a csatolmánynak a base64 kódolt szövege -->
                        byAxNiAwIFIvU2l6ZSAyMj4+CiVpVGV4dC01LjUuNQpzdGFydHhyZWYKNTg3NDkKJSVFT0YK</data>
                    <name>Certificate.pdf</name>
                </attachments>
                <consignmentId>653161</consignmentId>
                <messageDateTime>2017-10-13T12:59:40.398+02:00</messageDateTime>
                <messageType>DispatchCertificate</messageType>
                <recipients>posta2@hibrid.posta.hu</recipients>
                <sender>posta2@hibrid.posta.hu</sender>
                <uid>S_675306</uid>
            </return>
        </ns2:getMessageResponse>
    </soap:Body>
</soap:Envelope>
```

A getMessageResponse SOAP üzenetének szerkezete (egy Dispatch üzenettel)

4.3.2.1.3 Az üzenet fogadásának visszaigazolása releaseMessage

A lekérdezésre érkező válaszüzenet nyomán vissza kell igazolni az adott üzenet fogadását, különben a rendszer egy időtúllépési értéket követően ismételten elküldi ugyanazt az üzenetet, ami különösen a többszálú kezelés esetén – de egyszeres kapcsolatnál is – szükségtelen erőforrás lekötést okozna.

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	releaseMessage	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**

releaseMessage	Összetett	A lekérdezett és sikeresen rögzített üzenet visszaigazolása és ezzel „elengedésének” kérése a hibrid kézbesítési és konverziós rendszerből

15. táblázat: A releaseMessage parancs alapvető jellemzői

Maga a <releaseMessage> összetett adatcsomag szerkezete is egyszerű, mindössze a kérdező, az adott üzenetet előzetesen fogadó fél szerződésének azonosítóját, a rendszer által getMessage parancs végrehajtása során adott azonosítót (lásd a getMessageResponse adatcsomagjának szerkezetét) és a visszaigazolni kívánt üzenet típusát tartalmazza. Az utóbbi tulajdonképpen feleslegesnek tűnik, valószínűleg azért került ide, hogy az adatcsomag ne legyen túl rövid a lenyomatképzés egyértelműségéhez.

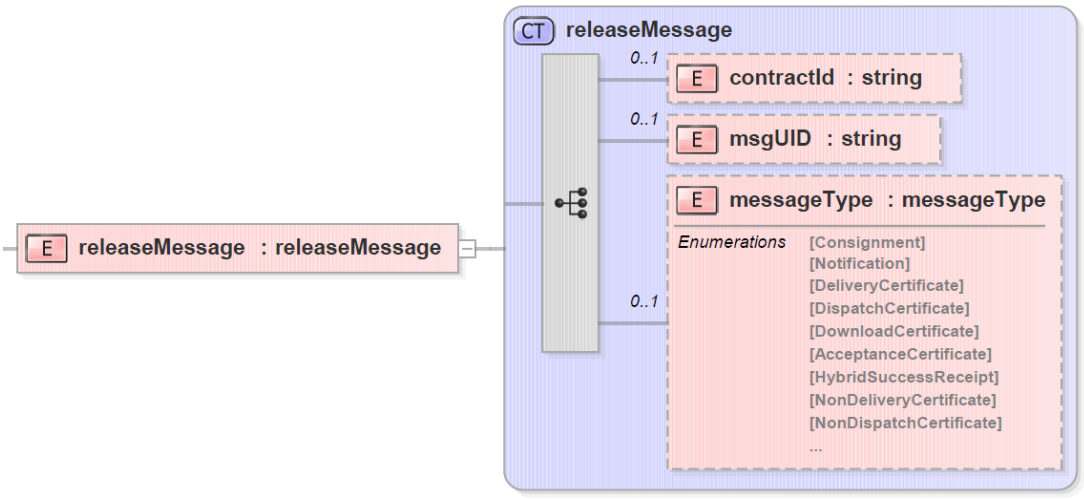
Elem név	Típus	Leírás
contractId	Karaktersorozat	A fogadó fél szerződésének azonosítója
msgUID	Karaktersorozat	a getMessageResponse üzenetben megkapott egyedi azonosító
messageType	Kötött értékkészlet egy eleme	A visszaigazolt üzenet típusa, a 28. táblázat egy eleme

16. táblázat: A releaseMessage adatcsomag elemei

Ennek megfelelően az üzenet XML sémája is könnyen áttekinthető:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="releaseMessage" type="tns:releaseMessage"/>
   <xs:complexType name="releaseMessage">
      <xs:sequence>
         <xs:element minOccurs="0" name="contractId" type="xs:string"/>
         <xs:element minOccurs="0" name="msgUID" type="xs:string"/>
         <xs:element minOccurs="0" name="messageType" type="tns:messageType"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="messageType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Consignment"/>
         <xs:enumeration value="Notification"/>
         <xs:enumeration value="DeliveryCertificate"/>
         <xs:enumeration value="DispatchCertificate"/>
         <xs:enumeration value="DownloadCertificate"/>
         <xs:enumeration value="AcceptanceCertificate"/>
         <xs:enumeration value="HybridSuccessReceipt"/>
         <xs:enumeration value="NonDeliveryCertificate"/>
         <xs:enumeration value="NonDispatchCertificate"/>
         <xs:enumeration value="NonDownloadCertificate"/>
         <xs:enumeration value="NonAcceptanceCertificate"/>
         <xs:enumeration value="HybridFailureReceipt"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

A releaseMessage XML sémája



18. ábra: A releaseMessage adatcsomag szerkezete

Itt is látható, hogy maga az üzenet teljes egyértelműséggel rögzíti, hogy melyik üzenetet lehet – mint már továbbítottat – kivenni az átvételre várakozó üzenetek közül és ezzel viszonylag kis kockázatúvá tenni a többszálas megvalósítást. A minta az előző példában szereplő DispatchCertificate üzenet visszaigazolásának adattartalmát mutatja.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
            <SignedInfo>
                <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
                <Reference URI="#Body">
                    <Transforms>
                        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                    </Transforms>
                    <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
                </Reference>
            </SignedInfo>
            <SignatureValue>MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==</SignatureValue>
            <KeyInfo>
                <X509Data>
                    <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi
                        Főigazgatóság,L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=
                        DO20141223-1DO3</X509SubjectName>
        <X509Certificate>MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
<!-- itt folytatódik a tanúsítvány base64 kódolással -->
nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==</X509Certificate>
                </X509Data>
            </KeyInfo>
        </Signature>
    </S:Header>
    <S:Body Id="Body">
        <ns2:releaseMessage xmlns:ns2="http://selexes.com/hmdacs">
            <contractId>3047</contractId>
            <msgUID>S_675306</msgUID>
            <messageType>DispatchCertificate</messageType>
```

```
        </ns2:releaseMessage>
    </S:Body>
</S:Envelope>
```

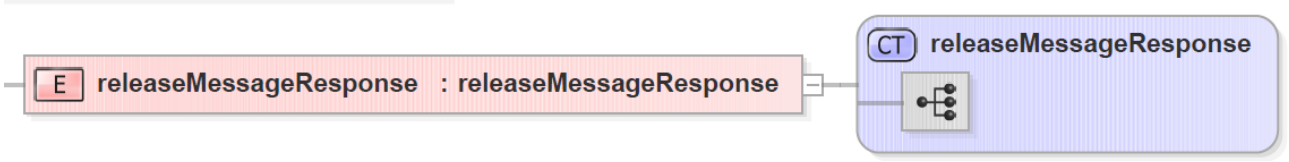
A releaseMessage SOAP üzenetének szerkezete

4.3.2.1.4 *Válasz az üzenet fogadásának visszaigazolására releaseMessageResponse*

A <releaseMessage> parancs visszaigazolása lényegében egy üres üzenet, az üzenet törzse nem tartalmaz elemet:

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
releaseMessageResponse	Összetett	A válasz ténye, maga a visszaigazolás

17. táblázat: A releaseMessageResponse adatcsomag alapstruktúrája



19. ábra: A releaseMessageResponse szerkezete

Ennek megfelelően az XML séma és a SOAP üzenet is igen egyszerű

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:releaseMessageResponse xmlns:ns2="http://selexes.com/hmdacs">
      </ns2:releaseMessageResponse>
   </soap:Body>
</soap:Envelope>
```

A releaseMessageResponse SOAP üzenetének szerkezete

A fentiek alapján látható, hogy a küldeményhez kapcsolódó igazolások kezelését a webszerviz is képes teljes értékűen biztosítani feladói oldalon, ha a csatlakozó rendszer képes bizonyítékok, illetve a rendszer által küldött üzenetetek feldolgozására, valamint képes a megfelelő parancsok adattartalmát a hozzáféréshez előállítani.

4.3.2.2 A küldő számára biztosított igazolások kezelése hivatali kapu használatával

Ahogy azt már a 3.8.2 fejezetben jeleztük a hivatali kapun keresztül a kommunikáció minden esetben KRX formátumú üzenetek használatával történik.

Ennek megfelelően a rendszer az elkészülő igazolásokat is KRX struktúrában helyezi el, az igazolás PDF állomány a Payload alkönyvtár ID-1 alkönyvtárában fog elhelyezkedni. Maga az igazolás teljesen megegyezik a 4.2 fejezetben leírtakkal, csak a formátum megkövetelte kiegészítő információkat kapja meg, így felöltésre kerül a kuldemeny_meta.xml állomány is, amelyet a fogadó iratkezelő rendszerek értelmezhetnek.

Az elkészült igazolásokat a küldő a hivatali kapu általános kezelési gyakorlatának megfelelően kézi vagy gépi úton szedheti le és dolgozhatja fel.

4.3.3 Kommunikáció a fogadó oldalon

Az elektronikus kézbesítési szolgáltató a Feladótól érkező küldemény befogadása és az átvétel visszaigazolása után eltárolja a küldeményt, és értesíti a Címzettet az általa meghatározott értesítési csatornán (SMS vagy e-mail), és a saját hozzáférési felületén is, hogy átvehető küldeménye érkezett. Biztonságos kézbesítési szolgáltatással küldött küldemény esetén az értesítéssel egyidejűleg a rendszer elkészíti a Címzett által aláírandó átvételi elismervényt és azt letölthetővé teszi a rendszerben.

4.3.3.1 A küldemény, illetve az értesítések címzett általi átvétele a webAutomata használatával

A címzett egy küldemény érkezéséről webAutomata használata esetén egy. a *getMessage* parancsra kapott (lásd 4.3.2.1.1 – 4.3.2.1.4fejezet) olyan *getMessageResponse* válasz adatcsomagból (lásd 4.3.2.1.2 fejezet) szerez tudomást, amelyben a <message> tag <messageType> elemének értéke „Notification”.

Amennyiben a *getMessage* parancsra kapott adatcsomagban a <deliveryType> elem értéke „Simple”, a <messageType> elem értéke pedig az alábbiak egyike

- „Consignment”,
- „DispatchCertificate”,
- „Non DispatchCertificate”,
- „DeliveryCertificate”, vagy
- „NonDeliveryCertificate”

a *getMessage* parancsra érkezett küldeményt közvetlenül kell feldolgozni, értelmezni, ahogy azt már a 4.3.2.1.1 – 4.3.2.1.4 fejezetekben részletesen bemutattuk. Ugyanez érvényes abban az esetben is, ha a *getMessage* parancsra kapott adatcsomagban a <deliveryType> értéke „Trusted”, a <messageType> értéke pedig az alábbiak egyike:

- „Consignment”,
- „DispatchCertificate”,
- „Non DispatchCertificate”,
- „DownloadCertificate”, vagy
- „NonDownloadCertificate”.

Értelemszerűen az egyes igazolások nem önkényes sorrendben jelennek meg az egyes küldemények szempontjából, csak akkor, ha a küldemény hozzáférésének, illetve az igazolások elkészültének feltételei előzetesen teljesültek.

A <messageType> elem „Notification” értéke esetén először magát az adatcsomagot kell feldolgozni. Nem automatikus működés esetén azt meg is kell tekinteni a további lépések meghatározásához. Automatizált feldolgozás esetén ezen adatcsomag <consignmentId> elemének értéke jelenti azt a küldeményazonosítót, amely meg fog érkezni. Attól függően, hogy az adatcsomagban a <deliveryType> értéke „Simple” vagy „Trusted”, kell az átvételi elismervényt letölteni és aláírni vagy sem.

Ezt követően minden esetben szükséges az üzenet vételének haladéktalan visszaigazolása, azaz a *releaseMessage* parancs elküldése (lásd 4.3.2.1.3 fejezet) a megkapott üzenet három paraméterével. Ennek a webAutomata általi *releaseMessageResponse* útján (lásd 4.3.2.1.4 fejezet) történő visszaigazolását követően kezdődhet meg a biztonságos kézbesítési szolgáltatással érkezett üzenet visszaigazolásának folyamata.

A továbbiakban már csak a biztonságos kézbesítési szolgáltatás részeként a webAutomatában a küldemények fogadásának visszaigazolásához kialakított speciális üzenetváltási megoldásokkal foglalkozunk.

4.3.3.1.1 Az átvételi elismervény dokumentum elérésének kérése

Amennyiben a *getMessage* útján kapott adatcsomagban a <messageType> elem értéke „Notification” és a <deliveryType> értéke „Trusted”, akkor a *releaseMessage* parancs végrehajtását követően el kell végezni az átvételi elismervény dokumentum lekérését.

Ez a dokumentum megfelelő azonosítási információkkal egyértelműen azonosítja a kézbesítésre felajánlott küldeményt, megadja a kézbesítési folyamatra vonatkozó fontosabb adatokat és bizonyíthatóvá teszi a küldemény tartalmát. Az átvételi elismervény lekérésére a *getAcceptanceCertificate* parancs szolgál.

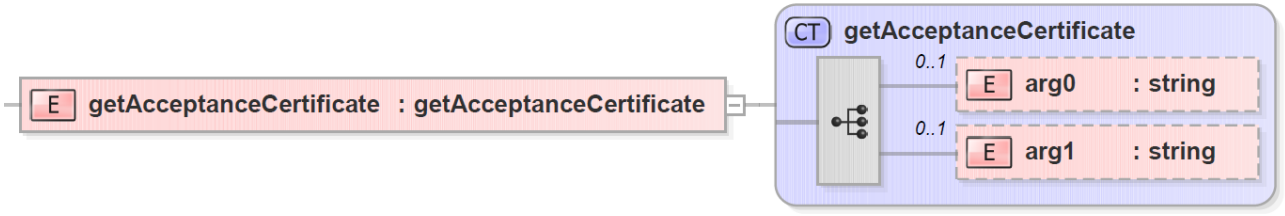
Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	getAcceptanceCertificate	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getAcceptanceCertificate	Összetett	Az azonosított biztonságos kézbesítési szolgáltatással érkező küldemény átvételi elismervényének lekérdezésére irányuló kérés adatai
arg0	Karaktersorozat	A szerződés azonosítója
arg1	Karaktersorozat	Az értesítésben megkapott küldeményazonosító <consignmentId>

18. táblázat: A getAcceptanceCertificate parancs alapvető jellemzői

A getAcceptanceCertificate parancs szerkezete is egyszerű, mindössze két paramétere van, a szerződés azonosítója és az előzőleg, a „Notification” részeként megkapott küldemény-azonosító, a <consignmentId> elem értéke. Sajnálatos, hogy itt a parancs paramétereinek elnevezése nem beszédes. Az adatok XML sémája is egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="getAcceptanceCertificate" type="tns:getAcceptanceCertificate"/>
   <xs:complexType name="getAcceptanceCertificate">
      <xs:sequence>
         <xs:element minOccurs="0" name="arg0" type="xs:string"/>
         <xs:element minOccurs="0" name="arg1" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A getAcceptanceCertificate xml sémája



20. ábra: A getAcceptanceCertificate adatcsomagjának szerkezete

Ennek megfelelően a SOAP üzenetben is ismételten az aláírással kapcsolatos információ jelenti a küldemény túlnyomó részét:

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYI7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
         <SignatureValue>
            MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
            ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
            cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
            B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
            M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
         </SignatureValue>
         <KeyInfo>
            <X509Data>
               <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi
                  Főigazgatóság,L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
               </X509SubjectName>
               <X509Certificate>
                  MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
                  dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
                  <!-- itt folytatódik a tanúsítvány base64 kódolással -->
                  nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
```

```
                RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
              </X509Certificate>
            </X509Data>
          </KeyInfo>
        </Signature>
      </S:Header>
      <S:Body Id="Body">
        <ns2:getAcceptanceCertificate xmlns:ns2="http://selexes.com/hmdacs">
          <arg0>3047</arg0>
          <arg1>675306</arg1>
        </ns2:getAcceptanceCertificate>
      </S:Body>
</S:Envelope>
```

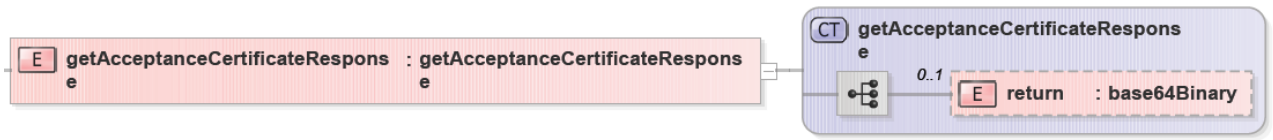
A getAcceptanceCertificate SOAP üzenet szerkezeti mintája

4.3.3.1.2 Az átvételi elismervény dokumentum letöltése

A kérésre érkező válaszüzenet helyes adatmegadás esetén tartalmazza a kért átvételi elismervény dokumentumot a 4.2.3 fejezetben már ismertetett tartalommal.

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getAcceptanceCertificateResponse	Összetett	Válasz az átvételi elismervény-kérésre
return	base64 kódolású bitfolyam	Az átvételi elismervény állomány

19. táblázat: A getAcceptanceCertificateResponse adatcsomag alapstruktúrája



21. ábra: A getAcceptanceCertificateResponse adatcsomag szerkezete

A *getAcceptanceCertificateResponse* szerkezete a lehető legegyszerűbb, kizárólag a kért átvételi elismervény dokumentumot tartalmazza mindenfajta további azonosító nélkül, ennek megfelelően itt az üzenetváltás csak kérdés-válaszként értelmezhető. Természetesen az átvételi elismervény maga tartalmaz a kéréssel összevethető, azonosításra alkalmas adatokat, de ahhoz magát a megkapott átvételi elismervény dokumentumot kell először értelmezni. Az adatcsomag XML sémája is igen egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="getAcceptanceCertificateResponse" type="tns:getAcceptanceCertificateResponse"/>
  <xs:complexType name="getAcceptanceCertificateResponse">
    <xs:sequence>
      <xs:element minOccurs="0" name="return" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A getAcceptanceCertificateResponse adatcsomag XML sémája

Az egyszerű szerkezetnek és a válaszüzenet jellegnek megfelelően itt a SOAP üzenet szerkezetileg igen egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:getAcceptanceCertificateResponse xmlns:ns2="http://selexes.com/hmdacs">
         <return>
            JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9JbnRlbnQvUGVyY2VwdHVhbC9EZWNvZGVQYXJtczw8L0NvbG9ycyAzL1By
            <!-- itt folyatódik maga az átvételi elismervény base64 kódolt szövege -->
            SW5mbyAxNiAwIFIvU2l6ZSAyMj4+CiVpVGV4dC01LjUuNQpzdGFydHhyZWYKNTg3NDkKJSVFT0YK
         </return>
      </ns2:getAcceptanceCertificateResponse>
   </soap:Body>
</soap:Envelope>
```

A getAcceptanceCertificateResponse SOAP üzenet szerkezeti mintája

Ha esetleg olyan adatok szerepelnek a kérésben, amelyre vonatkozóan nem létezik átvételi elismervény, akkor a return tag egy hibaüzenetet fog tartalmazni. A hibaüzenet ekkor „NoCertificateFoundException”. Ennek akkor van lehetősége, ha a megtagadásra vonatkozó átvételi elismervényt már előzetesen lekérte a címzett, vagy természetesen, ha az azonosítók nem létező küldeményre mutatnak. A hibát akkor a fogadó rendszernek kell lekezelnie.

4.3.3.1.3 Az átvételi elismervény dokumentum aláírása

A megkapott átvételi elismervény dokumentumot, amely egy PDF állomány, a küldemény átvétele érdekében a Hibrid kézbesítési és konverziós rendszerben előzetesen már regisztrált, legalább fokozott biztonságú elektronikus ügyintézésre alkalmas elektronikus aláírással alá kell írni vagy az előbbi követelményeknek megfelelő elektronikus bélyegzővel kell ellátni. Ez egy külső PDF aláíró alkalmazással történhet meg, ami értelemszerűen a kliens rendszer része is lehet. Az igényelt aláírás formátum ennek megfelelően PAdES. Időbélyegző elhelyezése nem követelmény, de lehetséges.

A fogadó rendszer ellenőrzi az aláírás érvényességét, és hogy valóban jogosult (előzetesen regisztrált) aláírással történt az átvételi elismervény aláírása. Ezek biztosítása a sikeres átvétel érdekében a fogadó (címzett) fél feladata.

4.3.3.1.4 Az aláírt átvételi elismervény dokumentum felöltése

Az így elkészített, a regisztrált tanúsítvánnyal aláírt átvételi elismervényt fel kell tölteni a webAutomatába, hogy az elvégezhesse annak ellenőrzését, és azok sikeres esetén az erre épülő műveletsort elindíthassa.

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	sendSignedAcceptanceCertificate	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**

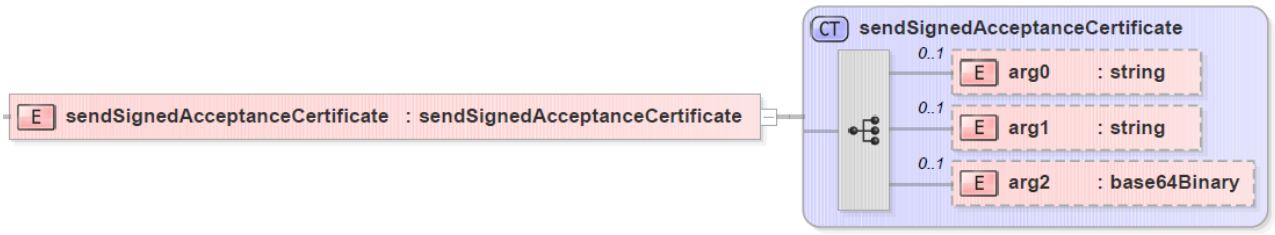
sendSignedAcceptanceCertificate	Összetett	A megkapott és aláírt átvételi elismervény feltöltéséhez szükséges adatcsomag
arg0	Karaktersorozat	A szerződés azonosítója
arg1	Karaktersorozat	Az értesítésben megkapott küldeményazonosító <consignmentId>
arg2	base64 kódolású bitfolyam	Az aláírt átvételi elismervény állomány

20. táblázat: A sendSignedAcceptanceCertificate parancs alapvető jellemzői

Mint látható a parancs a getAcceptanceCertificate rokona, mindössze egy adattal bővült, kiegészült magával az aláírt átvételi elismervénnyel. Ennek megfelelően az XML séma is hasonló szerkezetű az előbbihez:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="sendSignedAcceptanceCertificate" type="tns:sendSignedAcceptanceCertificate"/>
   <xs:complexType name="sendSignedAcceptanceCertificate">
      <xs:sequence>
         <xs:element minOccurs="0" name="arg0" type="xs:string"/>
         <xs:element minOccurs="0" name="arg1" type="xs:string"/>
         <xs:element minOccurs="0" name="arg2" type="xs:base64Binary"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A sendSignedAcceptanceCertificate XML sémája



22. ábra: A sendSignedAcceptanceCertificate adatcsomagjának szerkezete

Ennek megfelelően a SOAP üzenet szerkezete is nagyon hasonló:

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
         <SignatureValue>
            MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
            ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
            cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
```

```
            B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
            M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
         </SignatureValue>
         <KeyInfo>
            <X509Data>
               <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi Főigazgatóság,
                  L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
               </X509SubjectName>
               <X509Certificate>
                  MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
                  dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
                  <!-- itt folytatódik a tanúsítvány base64 kódolással -->
                  nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
                  RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
               </X509Certificate>
            </X509Data>
         </KeyInfo>
      </Signature>
   </S:Header>
   <S:Body Id="Body">
      <ns2:sendSignedAcceptanceCertificate xmlns:ns2="http://selexes.com/hmdacs">
         <arg0>3047</arg0>
         <arg1>675306</arg1>
         <arg2>
            JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9JbnRlbnQvUGVyY2VwdHVhbC9EZWNvZGVQYXJtczw8L0NvbG9ycyAzL1By
            <!-- itt folyatódik az aláírt átvételi elismervény base64 kódolt szövege -->
            SW5mbyAxNiAwIFIvU2l6ZSAyMj4+CiVpVGV4dC01LjUuNQpzdGFydHhyZWYKNTg3NDkKJSVFT0YK
         </arg2>
      </ns2:sendSignedAcceptanceCertificate>
   </S:Body>
</S:Envelope>
```

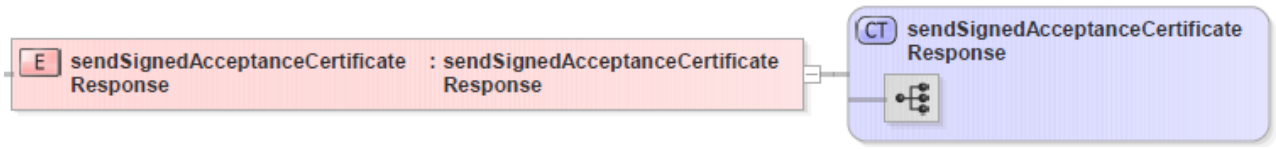
A sendSignedAcceptanceCertificate SOAP üzenet szerkezeti mintája

4.3.3.1.5 *Az aláírt átvételi elismervény dokumentum feltöltésének visszaigazolása*

A *sendSignedAcceptanceCertificate* parancs s webAutomata által adott visszaigazolása lényegében egy üres üzenet, az üzenet törzse nem tartalmaz elemet:

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
sendSignedAcceptanceCertificateResponse	Összetett	A válasz ténye, maga a küldés visszaigazolása

21. táblázat: A sendSignedAcceptanceCertificateResponse adatcsomag alapstruktúrája



23. ábra: A sendSignedAcceptanceCertificateResponse adatcsomag szerkezete

Ennek megfelelően az adatcsomag XML sémája és az annak nyomán kialakított SOAP üzenet is rendkívül egyszerű. Ebben az esetben kizárólag az üzenet kliens általi megkapásának ténye hordozza az információt.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="sendSignedAcceptanceCertificateResponse" type="tns:sendSignedAcceptanceCertificateResponse"/>
```

```
  <xs:complexType name="sendSignedAcceptanceCertificateResponse">
    <xs:sequence/>
  </xs:complexType>
</xs:schema>
```

A sendSignedAcceptanceCertificateResponse adatcsomag XML sémája

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
          xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <soap:Body>
    <ns2:sendSignedAcceptanceResponse xmlns:ns2="http://selexes.com/hmdacs">
    </ns2:sendSignedAcceptanceResponse>
  </soap:Body>
</soap:Envelope>
```

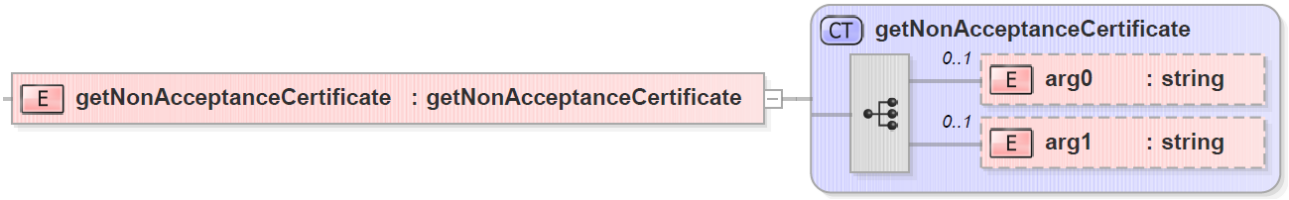
A sendSignedAcceptanceCertificateResponse SOAP üzenet mintájának szerkezete

4.3.3.1.6 A nemleges átvételi elismervény lekérése

Amennyiben a 4.3.3.1 fejezet végén leírt döntési helyzetben bármilyen okból olyan döntés születik, hogy a küldemény átvétele helyett a visszautasítást kell megvalósítani, akkor az előzőekben leírt folyamatot annyi eltéréssel kell végigvinni a webAutomatát megszólító klienssel, hogy az átvételt igazoló elismervény helyett az átvétel megtagadását igazoló át nem vételi elismervényt kell letölteni, értelemszerűen a *getNonAcceptanceCertificatet* parancs használatával.

Ahogy jeleztük már ez a megoldás egyáltalán nem életszerű, de ahhoz, hogy a webAutomatán keresztül, teljes mértékben gépesített módon meg lehessen oldal ezt a kommunikációs helyzetet is, implementálni kellett ezt a megoldást. Így akár a kezelőnek egyedi döntéssel, akár a rendszernek bármilyen a fogadó oldalon megállapított döntési jellemző alapján van lehetősége automatizáltan is az érkező küldemény visszautasítására. Természetesen a kiválasztási mechanizmus kialakítása és programozása a kliens oldal feladata.

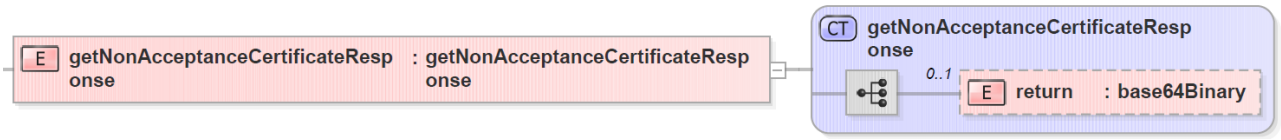
A parancs mind szerkezetében mind logikájában megegyezik a *getAcceptanceCertificate* paranccsal. Éppen ezért csak a szerkezetét mutatjuk be, ami azonnal összevethető a 20. ábra21. ábra, illetve erre építve a teljes 4.3.3.1.1 fejezet tartalmával.



24. ábra: A getNonAcceptanceCertificate adatcsomagjának szerkezete

4.3.3.1.7 Nemleges átvételi elismervény dokumentum letöltése

A helyzet itt is megegyezik a párhuzamos pozitív eljárással, az alábbi adatcsomag szerkezeti ábrája tartalmát, szerkezetét tekintve megegyezik a 21. ábra tartalmával, ennek megfelelően a 4.3.3.1.2 fejezetben leírtak itt is érvényesek.



25. ábra: A getNonAcceptanceCertificateResponse adatcsomagjának szerkezete

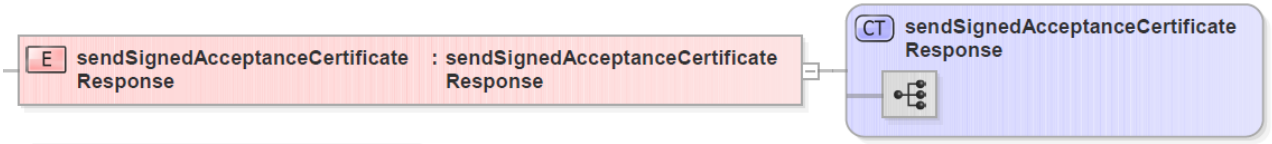
4.3.3.1.8 Az aláírt nemleges átvételi elismervény dokumentum feltöltése

A helyzet itt is megegyezik a párhuzamos pozitív eljárással, az alábbi adatcsomag szerkezeti ábrája tartalmát, szerkezetét tekintve megegyezik a 22. ábra tartalmával, ennek megfelelően a 4.3.3.1.3 és 4.3.3.1.4 fejezetekben leírtak itt is érvényesek.

26. ábra: A sendSignedNonAcceptanceCertificate adatcsomagjának szerkezete

4.3.3.1.9 Az aláírt nemleges átvételi elismervény dokumentum feltöltésének visszaigazolása

A helyzet itt is megegyezik a párhuzamos pozitív eljárással, az alábbi adatcsomag szerkezeti ábrája tartalmát, szerkezetét tekintve megegyezik a 23. ábra tartalmával, ennek megfelelően a 4.3.3.1.5 fejezetben leírtak itt is érvényesek.



27. ábra: A sendSignedNonAcceptanceCertificateResponse adatcsomagjának szerkezete

Azaz összefoglalóan is megállapítható, hogy valamennyi lépésben párhuzamos a feldolgozás, csak azt kell eldönteni a folyamat indításakor, hogy elfogadni vagy elutasítani akarjuk a küldeményt és annak megfelelően végigmenni a négy lépésen.

4.3.3.1.10 Az aláírt átvételi elismervények feldolgozása

Az aláírt vagy bélyegzővel ellátott, átvételt vagy visszautasítást igazoló átvételi elismervények a webAutomata általi befogadást követően az általános szabályoknak megfelelően kerülnek feldolgozásra, illetve ennek nyomán a kézbesítésre vonatkozó igazolások kiküldésre. Ebből a szempontból tehát a feldolgozás során oda kell figyelni, hogy előfordulhatnak olyan „NonDownloadCertificate”-ek is, amelyek tartalmaznak beágyazva visszautasítást. Ez azonban az általános kezelhetőségi szabályokat, illetve követelményeket nem érinti.

Amennyiben az átvételi elismervényt fel tudta dolgozni a rendszer, úgy a következő *getMessage* parancsok egyikénél a címzett oldalán meg fog jelenni, mint elérhető, egy „Consignment” értékű <MessageType> elemmel rendelkező üzenet, amelyet letöltve megkapjuk magát a már átvett küldeményt. Majd pedig mind a feladóhoz, mind a címzetthez megérkezik egy-egy „DownloadCertificate” értékű <MessageType> elemmel rendelkező üzenet, ami a mindkét félnek megküldött letöltési igazolás (tértivevény) beleágyazva az aláírt átvételi elismervénnyel. Ezzel a kommunikáció befejeződik.

A visszautasítást tartalmazó átvételi elismervény feldolgozását követően pedig a feladó és a címzett is kap egy-egy „NonDownloadCertificate” értékű <MessageType> elemmel rendelkező üzenet, ami a mindkét félnek megküldött letöltési igazolás (tértivevény), amely jelen esetben a visszautasítást tanúsítja, beleágyazva az aláírt visszautasító elismervénnyel. Ezzel párhuzamosan az üzenet maga törlődik a rendszerből, és a bizonyítékok, illetve a naplózási adatok tartós megőrzésre kerülnek.

Megállapítható, hogy amennyiben a kliens szoftver ki tudja használni a webAutomata biztosította lehetőségeket, akkor a kézbesítési szolgáltatás és a biztonságos kézbesítési szolgáltatás valamennyi eleme megvalósítható, sőt az üzenetek (message) szerkezete, összetettebb struktúrájú közlések továbbítására is alkalmas.

4.3.3.2 Hivatali kapu használata címzetti oldalon

A rendszer jelenleg nem biztosít lehetőséget arra, hogy általános esetben a postai BKSZ/KSZ igénybe vevője hivatali kapu címét adja meg kézbesítési címként.

Ez alól egy kivétel működik jelenleg is. Amikor a szerződő fél inverz hibrid szolgáltatást vesz igénybe, akkor megadhatja kézbesítési címként a hivatali kapu címét, ahová az átalakított vagy hitelesen átalakított elektronikus küldemények a KSZ szabályinak megfelelően megérkeznek. Mivel a hivatali kapu nincs felkészítve az átvételi elismervények aláírattatására, így BKSZ-t ezen az úton jelenleg nem lehet igénybe venni.

5 A webszerviz funkciói és használatuk

A hibrid kézbesítési és konverziós rendszer webszerviz használatával közvetlenül elérhető. Ez a megoldás elsősorban abban az esetben célravezető, ha a hibrid kézbesítési és konverziós rendszer szolgáltatásait gép-gép interfésszel kívánja valamely kapcsolódó szervezet igénybe venni. Ennek megfelelően a webszerviz igénybe vételéhez a csatlakozni kívánó szervezetnek kell rendelkeznie olyan saját alkalmazással, amely képes a hibrid kézbesítési és konverziós rendszer külső kommunikációra kialakított webszervizével (amit a továbbiakban WebAutomata-ként fogunk jelölni) kommunikálni.

A webszerviz szolgáltatásait a rendszer egy szabványos wsdl formájában ajánlja ki. A webAutomata.wsdl egyrészt az erre vonatkozó szabályok szerint a később megadott címen történő meghívással érhető el, másrészt leírva az 1. sz. függelék tartalmazza egy strukturális ábrával.

A hozzá kapcsolódó kliens program feladata, hogy értelmezze a webAutomata.wsdl-t, és ezen keresztül biztosítsa a szerveren rendelkezésre álló parancsok megfelelő használatát. Az adatcserében felhasználható összes speciális adattípus meghatározása a wsdl állományba van beágyazva. A leírás logikája a wsdl séma 1.1 verziójának megfelelő formátumot követi. (lásd http://schemas.xmlsoap.org/wsdl/soap/) A kliensnek a wsdl-ből megkapott információk alapján kell megformáznia saját SOAP üzeneteit, illetve kell fogadnia a szintén SOAP formátumban érkező válaszokat a wsdl-ben kiajánlott funkciók használatához. A rendszer a SOAP W3C specifikáció 1.1. verzióját használja http://www.w3.org/TR/2000/NOTE-SOAP-20000508/ a kommunikációhoz.

A webAutomata.wsdl-ben kiajánlott valamennyi szolgáltatás request-response (kérés-válasz) jellegű, azaz a minden egyes kommunikáció egy-egy üzenetváltásra épül. A webAutomata fogad egy a klienstől érkező kérést, és választ fog küldeni. Az már nem törvényszerű, hogy a válasz adatot is továbbítson. A kérdés-válasz párokkal a kézbesítési szolgáltatás és biztonságos kézbesítési szolgáltatás valamennyi műveletét meg lehet valósítani.

Mivel a 4. fejezetben részletesen bemutatásra került a webAutomata használatának lehetősége a kézbesítési szolgáltatás/biztonságos kézbesítési szolgáltatás egyes műveleteinek megvalósítása szemszögéből, ezért ebben a fejezetben kifejezetten a webAutomata.wsdl állományban megadott kérdés-válasz párokra koncentrálva mutatjuk be az azok használatához szükséges információt. A megértés elősegítése érdekében az előző fejezetekben már ezek funkciókkal kapcsolatban bemutatott táblázatokat, ábrákat az eredeti számozással meg is ismételjük, hogy a használathoz ne kelljen visszalapozni.

Nem tárgya ugyanakkor ennek a leírásnak a webAutomata más célokra (hibrid, inverz hibrid) való felhasználásához kapcsolódó folyamati specialitások leírása. Ezek azonban ugyanazokkal a parancsokkal mind megvalósíthatók, esetenként a feldolgozási folyamatok, a küldemény, igazolás típusok eltérőek, bár a 4. sz. mellékletben a felsorolások az érintett állományok által felvehető értékek szempontjából teljesek.

5.1 Kommunikáció a webAutomatával

A webszerviz https protokoll felett kommunikál.

A webAutomata úgy hoz létre egy biztonságos kommunikációs csatornát a kliens (az alkalmazást, amely megszólítja a hibrid rendszer webAutomata interfészét, a továbbiakban „kliensnek” nevezzük) és a Magyar Posta hibrid kézbesítési és konverziós rendszere (a továbbiakban szerver) között, hogy az transzparens a kliens alkalmazás fejlesztője számára.

A megvalósított biztonsági paradigma a következő:

- A kliens és a szerver közötti kommunikációs csatorna, csatornaszintű biztonságát SSL/TLS protokoll biztosítja egy hibrid kézbesítési és konverziós rendszer oldalán (szerver) telepített X.509 magánkulcsának felhasználásával, amely egy kliens oldali TrustStore-ban telepített nyilvános tanúsítvány párjával hozza létre a megfelelően biztonságos csatornát a TLS protokoll (RFC 5246) használatával;
- A kliens üzeneteinek hitelesítése (aláírása) egy kliens oldali kulcstárba (KeyStore) telepített X.509 magánkulccsal történik,
- Ezzel a kliens magánkulccsal történik az egyes webszolgáltatás-hívások SOAP üzenetei fejében a törzsben szereplő információk aláírása a WS security és a W3C XML signature (aláírás) szabványok szerint. A biztonsági modellre vonatkozó részletes XML alapú és a vonatkozó szabványhivatkozásokat is tartalmazó leírás a 2. sz. függelékben található.
- A kliens által készített és a SOAP üzenetben elküldött aláírás érvényességét a hibrid kézbesítési és konverziós rendszer ellenőrzi.

A kommunikáció hitelességét a küldőtől a rendszer irányában a SOAP üzenetek fejének aláírásával biztosítjuk, fordított irányban, mivel maguk az üzenetek önmagukban is aláírtak és időbélyeggel ellátottak, nem használunk ilyen jelentős számításigényű megoldást. Ennek megfelelően a küldött és fogadott üzenetek struktúrája jelentősen eltér. Az alkalmazott megoldásban minden üzenet egy kérésből és válaszból áll, ezeket fogjuk itt is tárgyalni.

E biztonsági modell használhatóságához a kliens (Igénybe vevői) oldalnak az alábbi követelményeknek kell megfelelnie:

- Rendelkezésre kell állnia egy az Igénybe vevő részére kibocsátott tanúsítvány állománynak, amely tartalmaz egy X.509 magánkulcsot aláírva egy, a megbízható EU közzétevők listáján szereplő magyar és/vagy európai minősített hitelesítés szolgáltató által, illetve a szolgáltató számára elérhetőnek kell lennie az előbbihez tartozó nyilvános tanúsítványnak és a visszavonási listának vagy az OCSP szolgáltatásnak;
- A nyilvános tanúsítványt a Magyar Posta rendelkezésére kell bocsátani, mivel azt hozzá kell rendelni a hibrid szolgáltatási szerződéshez, ugyanis ezt használva történik majd a küldeményeket tartalmazó SOAP üzenetek feladása (aláírása és indítása) a WebAutomata szolgáltatás felé. Ehhez a kibocsátó hitelesítés szolgáltatót és a tanúsítványt regisztrálni kell a szerver oldali hozzáférés-védelmi rendszerben. Tesztelési célokra a Magyar Posta kérés

esetén biztosít a saját CA-ja által kibocsátott teszt tanúsítványt, ezt azonban az éles rendszerbe nem lehet regisztrálni;

- A webAutomata, webszolgáltatást biztosító szervernek elérhetőnek kell lennie a kliens felől, amihez el kell végezni a tűzfalak beállítását, tehát a kapcsolat kialakításához meg kell adni a kliens rendszer IP címét.
- A SSL/TLS protokoll használatához szükséges azonosító tanúsítványt (a benne levő nyilvános kulccsal) egy, a hibrid kézbesítési és konverziós rendszer üzemeltetőjéhez küldött, szabványos (PKCS # 10) tanúsítványkérés útján kaphatja meg a kliens a Magyar Postától, de a csatlakozni kívánó is beszerezheti független szolgáltatótól. Ekkor a tanúsítványt megfelelően regisztrálni kell a hibrid kézbesítési és konverziós rendszerben.

A webAutomata címét és a hozzá tartozó nyilvános IP címet a sikeres csatlakozási tesztet követően adja meg a Magyar Posta. Mivel IP szűrés biztosított, így természetesen csak megadott IP címről lehet azt majd elérni.
A webAutomata címe a teszt környezetben: https://webautomata.hibrid.uat.posta.hu port:8888
A hozzá tartozó nyilvános IP cím: 194.88.45.250, de ezt is csak az arra feljogosítottak érhetik el.
A szolgáltatás összehangolása, üzembiztossá tétele, mivel itt egyedi alkalmazásokról van szó, nagy valószínűséggel jelentősebb tesztidőszakot igényel.

5.2 A webAutomata beállítását segítő eljárások

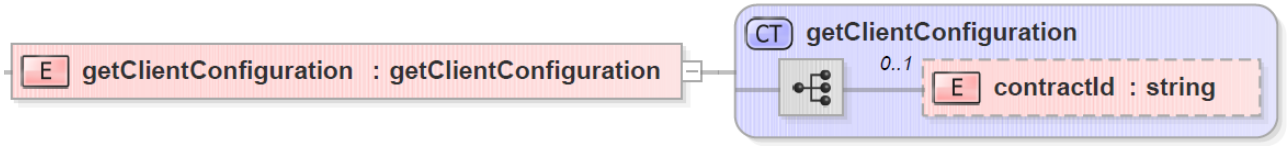
A webAutomata a kapcsolat beállításainak ellenőrzésére és a kapcsolat létezésének ellenőrzésére a webAutomata két eljárás-párt biztosít.

5.2.1 A szolgáltatás paramétereinek lekérdezése getClientConfiguration

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	getClientConfiguration	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getClientConfiguration	Összetett	A szerződéshez tartozó kliens működési paramétereinek lekérdezése
contractId	Karaktersorozat	A kapcsolatot megvalósító szerződés a rendszerben kapott azonosítója

1. táblázat: getClientConfiguration parancs

A kérdés adatcsomag egyetlen adatot tartalmaz a szerződés azonosítóját:



1. ábra: getClientConfiguration adatcsomagjának szerkezete

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://selexes.com/hmdacs" version="1.0" xmlns:tns="http://selexes.com/hmdacs">
   <xs:element name="getClientConfiguration" type="tns:getClientConfiguration"/>
   <xs:complexType name="getClientConfiguration">
      <xs:sequence>
         <xs:element minOccurs="0" name="contractId" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

getClientConfiguration adatcsomagjának XML sémája

Ennek megfelelően a kérést tartalmazó SOAP üzenet is viszonylag egyszerű, azonban az üzenet biztonsága érdekében a fej lényegesen terjedelmesebb, mint maga az üzenet érdemi tartalma:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYI7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
         <SignatureValue>
            MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
            ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
            cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYIMJE1XsI
            B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
            M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
         </SignatureValue>
         <KeyInfo>
            <X509Data>
               <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi
                  Főigazgatóság,L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
               </X509SubjectName>
               <X509Certificate>
                  MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
                  dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
                  VQQKDAdQcm9nZXNpMRswGQYDVQQDDBJNYXJjbyBDb25mYWxvbmllcmkwHhcNMTQwOTI1MTMyMDI2
                  <!-- itt folytatódik a tanúsítvány base64 kódolással -->
                  BjpHOh3wUAUGKwBYVrTL1icezubmC6oCOZqyHeEKYjiH+0C1Ic06RZ7Hzt0z860XeqzUA9T6qEYA
                  nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
                  RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
               </X509Certificate>
            </X509Data>
         </KeyInfo>
      </Signature>
   </S:Header>
   <S:Body Id="Body">
      <ns2:getClientConfiguration xmlns:ns2="http://selexes.com/hmdacs">
         <contractId>3047</contractId>
      </ns2:getClientConfiguration>
```

```
    </S:Body>
</S:Envelope>
```

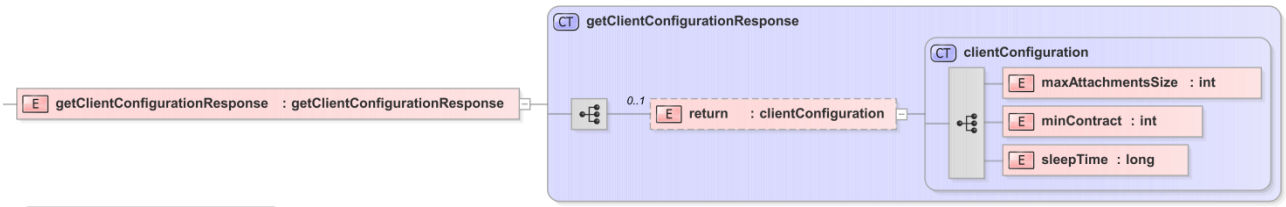
A getClientConfiguration SOAP üzenet egy szerkezeti példája

5.2.2 Válasz a szolgáltatás paramétereinek lekérdezésére getClientConfigurationResponse

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getClientConfigurationResponse	Összetett	A szerződéshez tartozó kliens működési paramétereinek visszaadása
return	Összetett	a válasz elem
clientConfiguration	Összetett	a konfiguráció leírói

2. táblázat: A getClientConfigurationResponse adatcsomag elemei

A válaszüzenet három adatot ad vissza:



2. ábra: A getClientConfigurationResponse adatcsomag szerkezete

Elem név	**Típus**	**Leírás**
maxAttachmentSize	Integer	A kliens által küldhető legnagyobb üzenet mérete MB-ban
minContract	integer	A legkisebb kezelhető contract azonosító
sleepTime	long	Az az idő, amennyi idő után, ha az adott szerződéshez a getMessage kérdésre nincs válasz, a kapcsolat alvó állapotba kerül

3. táblázat: A clientConfiguration összetett adat elemeinek értelmezése

A fenti paraméterek kiinduló beállításai a JBoss *configuration\configuration\esbapp* könyvtárban a web-automata.properties fájlban találhatók meg. Maximális csatolmányméret alapértelmezésben 20 MB, a szerződések azonosítószámai 1000-től kezdődnek és az alvási idő alapértéke 30000.

A SOAP üzenet szerkezete, mivel a fejet a webAutomatától a klienshez irányuló üzenetnél nem írjuk alá, lényegesen egyszerűbb:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

```
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:getClientConfigurationResponse xmlns:ns2="http://selexes.com/hmdacs">
         <return>
            <maxAttachmentSize>20</maxAttachmentSize>
            <minContract>1000</minContract>
            <sleepTime>NonDispatchCertificate</sleepTime>
         </return>
      </ns2:getClientConfigurationResponse>
   </soap:Body>
</soap:Envelope>
```

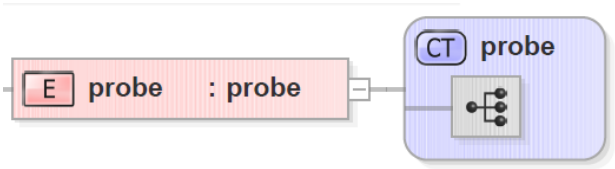
Ezzel az üzenetváltással tehát meg lehet győződni egyrészt a kapcsolat működőképességéről, másrészt le lehet kérdezni az aktuális beállításokat

5.2.3 A szolgáltatás tesztelése: probe

A webszolgáltatások alapműködéséhez tartozik egy olyan kérés-válasz pár beépítése, amely mindenbelső tartalom nélkül, pusztán az üzenetváltás képességét teszteli. Ennek általánosan elfogadott elnevezése a probe. Ez itt is megvalósításra került.

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	probe	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
probe	Összetett	egyetlen üres sequence tagot tartalmaz

4. táblázat: A probe parancs



3. ábra: A probe adatcsomag szerkezete

A parancs ebben az esetben az eljárás nevén kívül semmit nem tartalmaz, ennek megfelelően az adatcsomag XML sémája is egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
      xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="probe" type="tns:probe"/>
   <xs:complexType name="probe">
      <xs:sequence/>
   </xs:complexType>
</xs:schema>
```

A probe adatcsomag XML sémája

Ebben az esetben a SOAP üzenet szerkezete is nagyon egyszerű, lényegében csak az aláírás szintaktikai helyességét kell biztosítani.

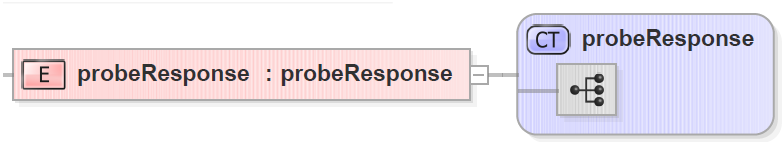
```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
         <SignatureValue>
            MPS9JHilRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
            ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
            cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
            B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
            M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
         </SignatureValue>
         <KeyInfo>
            <X509Data>
               <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi Főigazgatóság,
                  L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
               </X509SubjectName>
               <X509Certificate>
                  MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
                  dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
                  <!-- itt folytatódik a tanúsítvány base64 kódolással -->
                  nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
                  RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
               </X509Certificate>
            </X509Data>
         </KeyInfo>
      </Signature>
   </S:Header>
   <S:Body Id="Body">
      <ns2:probe xmlns:ns2="http://selexes.com/hmdacs">
      </ns2:probe>
   </S:Body>
</S:Envelope>
```

A probe parancs SOAP üzenetének szerkezeti mintája

5.2.4 Válasz a probe parancsra: probeResponse

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
probeResponse	Összetett	egyetlen üres sequence tagot tartalmaz

5. táblázat: probe válasz adatcsomagjának elemei



4. ábra: A probe válasz szerkezete

A válasz XML sémája lényegében a kéréssel megegyező:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
           xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="probeResponse" type="tns:probeResponse"/>
    <xs:complexType name="probeResponse">
        <xs:sequence/>
    </xs:complexType>
</xs:schema>
```

A probeResponse XML sémája

E válasz SOAP üzenet formájában is egyszerű, mivel ebben az irányban nem használunk aláírást a SOAP üzenet fejében.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
           xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <soap:Body>
    <ns2:probeResponse xmlns:ns2="http://selexes.com/hmdacs">
    </ns2:probeResponse>
  </soap:Body>
</soap:Envelope>
```

5.3 A küldemények küldését és fogadását szolgáló eljárások

A webszerviz egyik alapvető feladata a kliens által küldött üzenetek befogadása, aminek a párja a rendszer által generált vagy továbbított üzenetek fogadása. Ahogy már jeleztük a szolgáltatáskészlet minél egyszerűbb, ugyanakkor hatékony kialakítása érdekében az üzenetek küldését illetve fogadását általánosítottan, mindössze három üzenetpárral oldja meg a webAutomata. E fejezetben ezek használatát mutatjuk be.

5.3.1 Üzenet továbbítása a sendMessage parancs használatával

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	sendMessage	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
sendMessage	Összetett	A szerződéshez tartozó egy üzenet továbbítása a hibrid kézbesítési és

		konverziós rendszerbe a feldolgozás jellemzőinek meghatározásával

6. táblázat: A sendMessage parancs alapvető jellemzői

A <sendMessage> üzenet két elemből áll:

Elem név	Típus	Leírás
contractId	Karaktersorozat	A küldő szerződésének azonosítója
message	Összetett	A teljes továbbítandó üzenet az leíró adataival

7. táblázat: A sendMessage adatcsomag szerkezeti elemei

A <message> tag maga is egy összetett elem, amelynek elemei között vannak kötött értékkészletűek és összetett elem is. Mivel a <message>, mit elem a kérdésekben és válaszokban egyaránt előfordul, vannak olyan elemeit, amelyek csak bizonyos esetekben kapnak értéket (a séma önmagában minden elem kitöltetlenségét megengedi, azonban bizonyos elemek hiánya a feladat természetéből adódóan a megvalósulás ellehetetlenüléséhez vezet.)

Elem név	Típus	Leírás
uid	Karaktersorozat	Az állomány a hibrid rendszer által adott egyedi azonosítója
subject	Karaktersorozat	Az üzenet tárgya
requestId	Karaktersorozat	Az igénybe vevő által adott azonosító
recipients	Karaktersorozat	Címzett
notificationEMail	Karaktersorozat	Az e-mail értesítés címe
messageType	Kötött értékkészlet egy eleme	A küldemények típusait felsoroló lista eleme (lásd 28. táblázat)
messageDateTime	Dátum-idő	A küldeménnyel kapcsolatos esemény időpontja
deliveryType	Kötött értékkészlet egy eleme	A küldés (szolgáltatás) típusait felsoroló lista eleme (lásd 27. táblázat)
consignmentId	Karaktersorozat	A küldemény a rendszerben kapott azonosítója
body	Karaktersorozat	A küldemény törzse
attachments	Összetett	Csatolmányok, azok jellemzőivel

8. táblázat: A message tag szerkezete

Az <attachments> tag (amely egyes esetekben el is maradhat) többször is ismételhető a <message> tagban. Ez is egy összetett tag, amely tartalmazza a csatolmány egyes elemeit, jellemzőit.

Elem név	Típus	Leírás
data	Karaktersorozat	Maga az állomány base64 kódolással

name	Karaktersorozat	Az állomány neve
signatureType	Kötött értékkészlet egy eleme	A dokumentum aláírásának típusát azonosítja (lásd 29. táblázat)

9. táblázat: Az attachments tag szerkezete

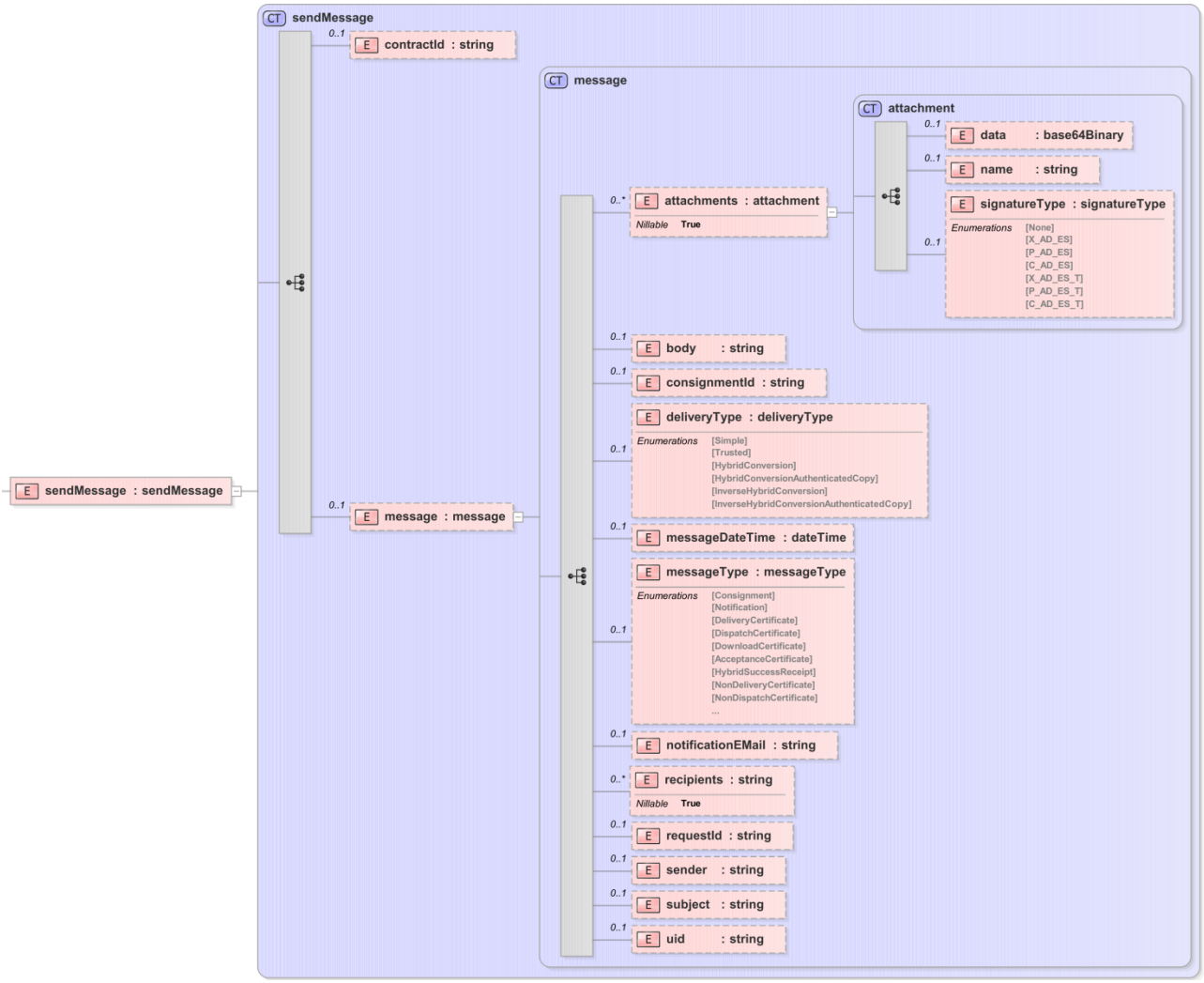
A wsdl-ben meghatározott kötött értékkészletű állományok felvehető értékeit azok értelmezésével a 4. sz. függelék tartalmazza.

A fenti struktúrában megadott elemek közül a küldemény típusának megfelelő adatokat kell megadni. Esetünkben a kimenő küldemény <messageType> értéke csak „Consignment” lehet. Mivel jelen esetben kézbesítési szolgáltatásról, illetve biztonságos kézbesítési szolgáltatásról lehet csak szó, ennek megfelelően a <deliveryType> értéke csak „Simple” vagy „Trusted” lehet. A csatolmányok adatait értelemszerűen ki kell tölteni, éppúgy, mint a feladó és címzett adatait, hiszen ezek hiányában a küldés nem értelmezhető. A tárgy és törzs, az értesítési e-mail cím, valamint a felhasználó által adott azonosító megadása értelemszerűen a feladó igényétől függ. Az esemény időpontja <messageDateTime>, a küldemény azonosítószáma <consignmentId> és az egyedi azonosító <uid> adatok viszont akár megadásra kerülnek, akár nem, a rendszer új értékeket fog megadni ezekre. A teljes adatcsomag XML sémája alább látható:

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:tns="http://selexes.com/hmdacs" elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs"
version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="sendMessage" type="tns:sendMessage"/>
  <xs:complexType name="sendMessage">
    <xs:sequence>
      <xs:element name="contractId" type="xs:string" minOccurs="0" />
      <xs:element name="message" type="tns:message" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="message">
    <xs:sequence>
      <xs:element name="attachments" type="tns:attachment" minOccurs="0" maxOccurs="unbounded" nillable="true" />
      <xs:element name="body" type="xs:string" minOccurs="0" />
      <xs:element name="consignmentId" type="xs:string" minOccurs="0" />
      <xs:element name="deliveryType" type="tns:deliveryType" minOccurs="0" />
      <xs:element name="messageDateTime" type="xs:dateTime" minOccurs="0" />
      <xs:element name="messageType" type="tns:messageType" minOccurs="0" />
      <xs:element name="notificationEMail" type="xs:string" minOccurs="0" />
      <xs:element name="recipients" type="xs:string" minOccurs="0" maxOccurs="unbounded" nillable="true" />
      <xs:element name="requestId" type="xs:string" minOccurs="0" />
      <xs:element name="sender" type="xs:string" minOccurs="0" />
      <xs:element name="subject" type="xs:string" minOccurs="0" />
      <xs:element name="uid" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="attachment">
    <xs:sequence>
      <xs:element name="data" type="xs:base64Binary" minOccurs="0" />
      <xs:element name="name" type="xs:string" minOccurs="0" />
      <xs:element name="signatureType" type="tns:signatureType" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="messageType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Consignment" />
      <xs:enumeration value="Notification" />
      <xs:enumeration value="DeliveryCertificate" />
      <xs:enumeration value="DispatchCertificate" />
      <xs:enumeration value="DownloadCertificate" />
```

```
            <xs:enumeration value="AcceptanceCertificate" />
            <xs:enumeration value="HybridSuccessReceipt" />
            <xs:enumeration value="NonDeliveryCertificate" />
            <xs:enumeration value="NonDispatchCertificate" />
            <xs:enumeration value="NonDownloadCertificate" />
            <xs:enumeration value="NonAcceptanceCertificate" />
            <xs:enumeration value="HybridFailureReceipt" />
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="signatureType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="None" />
            <xs:enumeration value="X_AD_ES" />
            <xs:enumeration value="P_AD_ES" />
            <xs:enumeration value="C_AD_ES" />
            <xs:enumeration value="X_AD_ES_T" />
            <xs:enumeration value="P_AD_ES_T" />
            <xs:enumeration value="C_AD_ES_T" />
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="deliveryType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Simple" />
            <xs:enumeration value="Trusted" />
            <xs:enumeration value="HybridConversion" />
            <xs:enumeration value="HybridConversionAuthenticatedCopy" />
            <xs:enumeration value="InverseHybridConversion" />
            <xs:enumeration value="InverseHybridConversionAuthenticatedCopy" />
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

A sendMessage adatcsomag XML sémája



14. ábra: A sendMessage adatcsomag szerkezete

Ennek megfelelően a sendMessage SOAP üzenet szerkezete a következő:

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
            <ds:Reference URI="#SBody">
               <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha256"/>
               <ds:DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</ds:DigestValue>
            </ds:Reference>
         </ds:SignedInfo>
         <ds:SignatureValue>
            mnDG3aeahjRqQLx19PKJUuRT/85hTtFxMWTmL13CF70KH1i7wlueUQJhRwD1EqH6XjxjJfTysexp
            tkGMw9fHRKnqQxuJH8RcNKxxUWQTKJjolBBc/971ZON9WAekTFkq9WdPwTOuDDQN5bN4yYPA7ZRC
            66F57Lz/GFRJGucV+GAjbm0xSdB89ZsD0g6p6uumJTMRbsuBpn5qbngzfwKd6dQ9B5kI5/zY+ONa
            oJsbDtCPSPJdEzCufv6K9hMsvKsKEU/A/Kg4qGqAJa6+8G/kCjXkpJSpwCCYd4r8sgxoYUV/S0+O
            E38VLHcxSURvfjVb11YWCCWmwYOE+T3gcIDZBw==
         </ds:SignatureValue>
         <ds:KeyInfo>
            <ds:X509Data>
               <ds:X509Certificate>
```

```
                MIIHHTCCBgWgAwIBAgIIK2IB5TTiq3owDQYJKoZIhvcNAQELBQAwgaYxCzAJBgNVBAYTAkhVMREw
                DwYDVQQHDAhCdWRhcGVzdDE8MDoGA1UECgwzTklTWiBOZW16ZXRpIEluZm9rb21tdW5pa8OhY2nD
                <!-- itt folytatódnak a tanúsítvány adatai base64 kódolással-->
                lszAlk+6Ll8wd9BySS9JjjvdjtJijqb9LF7jhaM44j6mVNQg68MYblqfTuVbhPXCprLjZjbUDJKYU
                0pfFyRd8UwoUX6j5DcZp0HwWNm+vuqKTZFcolXw845omExs5/jFUnh9mjW8Ewv2DWCDCmpZ/VXAxfQ==
              </ds:X509Certificate>
            </ds:X509Data>
          </ds:KeyInfo>
        </ds:Signature>
      </SOAP-ENV:Header>
      <S:Body Id="SBody">
        <ns2:sendMessage xmlns:ns2="http://selexes.com/hmdacs">
          <contractId>4063</contractId>
          <message>
            <attachments>
              <data>
                JVBERi0xLjQKJeLjz9MKMyAwIG9iago8PC9Db2xvclNwYWNlL0RldmljZUdyYXkvU3VidHlwZS9JbWFnZS9IZWlnaHQg
                <!-- itt folytatódik a base 64 kódolt szöveg -->
                Nz5dCi9Sb290IDQwIDAgUgovU2l6ZSA0NwovUHJldiAyMTAzOTYKPj4Kc3RhcnR4cmVmCjI3NzY0NwolJUVPRgo=
              </data>
              <name>proba118_00_9_00986_01_2017.pdf</name>
              <signatureType>P_AD_ES_T</signatureType>
            </attachments>
            <deliveryType>Trusted</deliveryType>
            <messageType>Consignment</messageType>
            <requestId>ST000000000000005</requestId>
          </message>
        </ns2:sendMessage>
      </S:Body>
    </S:Envelope>
```

A sendMessage SOAP üzenet egy szerkezeti példája

A küldeményt indító kliens rendszernek az itt megadott tartalomnak megfelelően és szerkezetben kell előállítani az adatcsomagot.

5.3.2 Válasz az üzenet továbbítására: sendMessageResponse

A rendszer az üzenet elküldésére a sendMessageResponse üzenettel válaszol, ami vagy üres, vagy egy esetleges hibaüzenetet tartalmazhat.

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
sendMessageResponse	Összetett	egy return elemet tartalmazhat
return	Karaktersorozat	A fogadó rendszer válasza, alapesetben az elküldött küldemény azonosítója (ConsignmentID), kivételesen hibaüzenet

10. táblázat: A sendMessageResponse adatcsomag elemei

A struktúrája igen egyszerű:

15. ábra: A sendMessageResponse adatcsomag szerkezete

Ennek megfelelően az XML séma is igen egyszerű

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="sendMessageResponse" type="tns:sendMessageResponse"/>
   <xs:complexType name="sendMessageResponse">
      <xs:sequence>
         <xs:element minOccurs="0" name="return" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A sendMessageResponse adatcsomag XML sémája

Mivel ez válaszüzenet a rendszertől a kliens felé, tehát SOAP üzenet formájában is igen egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:sendMessageResponse xmlns:ns2="http://selexes.com/hmdacs">
         <return>Itt alapesetben a küldeményazonosító, (consignmentID) kivételként a hibaüzenet szövege szerepelhet</return>
      </ns2:sendMessageResponse>
   </soap:Body>
</soap:Envelope>
```

A sendMessageResponse SOAP üzenet szerkezeti mintája

A fogadó rendszernek itt fogadnia kell a küldemény azonosítóját, ezzel tud a későbbiekben hivatkozni rá. Ezen túlmenően az esetleges kivételekből származó hibaüzenetek lekezelésére kell felkészülnie. Szerencsére ezek tulajdonképpen csak a rendszer összehangolásának időszakában relevánsak, hiszen a helyesen megformázott üzenetek esetében nem ezen a szinten kell a hibajelzéseknek előállniuk, azok a különböző igazolások, illetve a felhasználóknak küldött rendszerüzenetek formájában jelentkeznek.

A sendMessage parancsra kapott válasznál az „IllegalMessageTypeException” fordulhat elvileg elő. Ez a gyakorlatban azt jelenti, hogy nem „Consignment” a <messageType> értéke a küldött adatcsomagban. A küldendő üzenettípusként ugyanis a többi, a 28. táblázatban szereplő üzenettípus nem fordulhat elő.

Ezekre a kivételekre kell a küldő rendszernek valamilyen módon reagálnia. Értelemszerűen bizonyítékokban vagy értesítésekben szereplő <notification> és <error> üzenetek kezelése jellemzően nem gépi feldolgozást kíván, hanem humán kezelők informálását szolgálja.

5.3.3 Üzenet lekérdezése: getMessage

A webAutomata egységes logikával kezeli a feladónak, illetve a címzettnek küldött különböző típusú üzenetek lekérését, mivel maga a fogadás független attól, hogy a fogadott üzenet egy érdemi küldemény vagy egy igazolás, vagy csak egy figyelmeztetés. Ez a parancs-válasz pár a *getMessage – getMessageResponse* pár. Ez az egy üzenetpár azonban egy nem megbízható környezetben nem lenne elégséges az üzenet biztonságos eljuttatásához, ezért egy külön eljárás került implementálásra,

amely az üzenet „elengedését” biztosítja, azaz amivel az adott üzenet fogadója visszaigazolja a küldőnek, hogy az átvitel teljesedésbe ment, az üzenetet már nem kell rendelkezésre tartania. Ez a *releaseMessage – releaseMessageResponse* pár. Ha a rendszer nem kapja meg az adott adatcsomagra vonatkozóan a *releaseMessage* parancsot, akkor megfelelő időtúllépési (timeout) időszak után ismételten fel fogja kínálni az üzenetet.

A következőkben e parancsok használatával bemutatjuk, hogyan kérdezhet le akár a küldő, akár a fogadó fél üzeneteket.

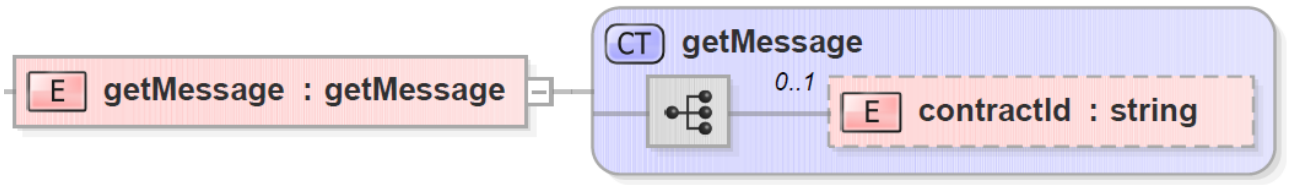
Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	getMessage	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getMessage	Összetett	A szerződéshez tartozó egy üzenet lekérdezése a hibrid kézbesítési és konverziós rendszerből

12. táblázat: getMessage parancs alapvető jellemzői

Maga a <getMessage> összetett elem rendkívül egyszerű, mindössze a kérdező, fogadni kívánó fél szerződésének azonosítóját tartalmazza

Elem név	**Típus**	**Leírás**
<contractId>	Karaktersorozat	A küldő szerződésének azonosítója

13. táblázat: A getMessage tag szerkezete



16. ábra: A getMessage adatcsomag szerkezete

Ennek megfelelően egyszerű a kérdés tartalmának XML sémája is

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
        xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="getMessage" type="tns:getMessage"/>
    <xs:complexType name="getMessage">
        <xs:sequence>
            <xs:element minOccurs="0" name="contractId" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

A getMessage sémája

A SOAP üzenet a biztonsági követelmények miatt összetettebb, de a tartalom itt is igen egyszerű. Itt már látszik, hogy ennek a kommunikációs modellnek ez a gyengéje, hogy a hasznos tartalomnál a biztonsági üzenetrész jelentősen több lehet.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYI7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
         <SignatureValue>
            MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
            ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
            cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYIMJE1XsI
            B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
            M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
         </SignatureValue>
         <KeyInfo>
            <X509Data>
               <X509SubjectName>
                  CN=Marco Confalonieri,O=Progesi,L=Genova,ST=Genova,C=IT,
                  1.2.840.113549.1.9.1=#160d6d6172636f40746573742e6974
               </X509SubjectName>
               <X509Certificate>
                  MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
                  dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
                  <!-- itt folytatódik a tanúsítvány tartalma base 64 kódolással -->
                  nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
                  RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
               </X509Certificate>
            </X509Data>
         </KeyInfo>
      </Signature>
   </S:Header>
   <S:Body Id="Body">
      <ns2:getMessage xmlns:ns2="http://selexes.com/hmdacs">
         <contractId>3047</contractId>
      </ns2:getMessage>
   </S:Body>
</S:Envelope>
```

A getMessage SOAP üzenetének szerkezete

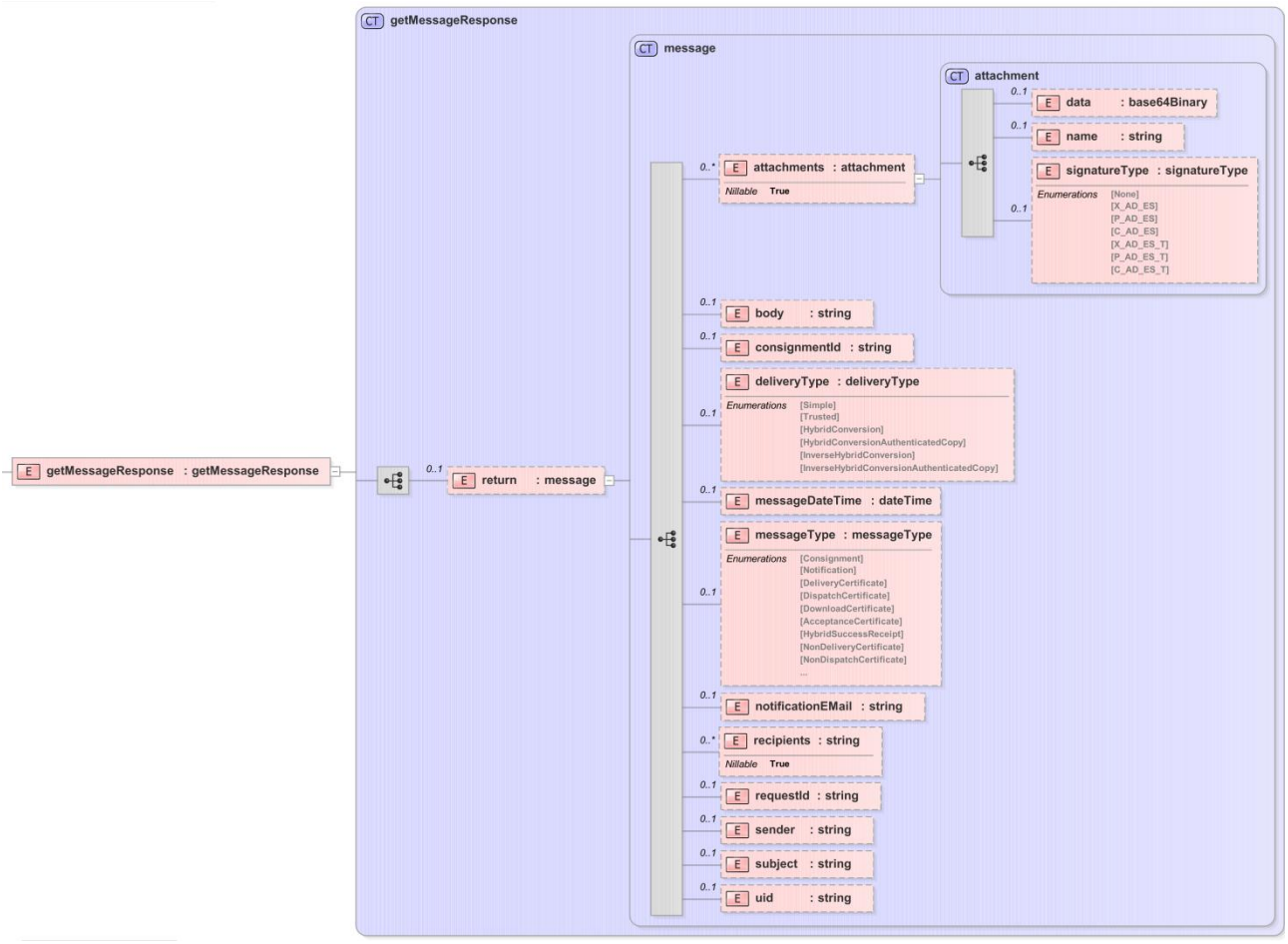
5.3.4 Válasz az üzenet lekérdezésére: getMessageResponse

A rendszer a lekérdezésre a <getMessageResponse> adatcsomaggal válaszol, ami a feladat általános jellegéből adódóan, hogy minden, a rendszer által akár a küldő, akár a fogadó felé elérhetővé tett adatcsomagot le kell tudnia kezelni, eléggé összetett. A válasz jellegből adódóan a <getMessageResponse> összetett elem egy return összetett típust tartalmaz, amelyben egy (és csak egy) már korábban, az 5.3.1 fejezetben és a 8. táblázatban ismertetett szerkezetű <message> összetett elem található. Amennyiben nincs visszaadható üzenet a <getMessageResponse> válaszüzenet üres.

Ez jelzi, hogy a kérdező rendszer várhat az ismételt lekérdezéssel. Az ütemezés itt értelemszerűen a kérdező oldal feladata.

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getMessageResponse	Összetett	A válasz tartalmát hordozó összetett elem
return	Összetett	A hibrid kézbesítési és konverziós rendszer válaszüzenete
message	Összetett	A 8. táblázat**Hiba! A hivatkozási forrás nem található.**ban részletezett tartalommal

14. táblázat: A getMessageResponse adatcsomag szerkezeti elemei



17. ábra: A getMessageResponse üzenet adatainak szerkezete

Itt már a Message üzenet valamennyi az adott üzenettípus esetén értelmezhető tagja értéket kap és azokat a fogadó fél (esetünkben a küldő) fel is tudja használni, a küldeménye sorsának követéséhez.

A küldő fél alapesetben először egy „DispatchCertificate” <messageType>-pal jellemzett küldeményt kap a getMessage kérdésére válaszként. Ez jelzi a küldemény sikeres befogadását. Amennyiben a küldemény esetleg olyan hibával volt terhelt, amely már a küldemény további feldolgozásra való befogadását is meghiúsította (vírusfertőzés, nem megfelelő kódolás, struktúra), akkor az üzenet típusa „NonDispatchCertificate” lesz. Itt kell felhívjuk a figyelmet arra, hogy az üzenet típusának meghatározása minden esetben a <messageType> tag által történik. Így például minden a hibrid kézbesítési és konverziós rendszer által kiadott igazolás a neve az <attachments> tagon belül „Certificate.pdf”. A „Certificate.pdf” maga beágyazottan már az igazolás jellegének megfelelő megnevezésű XML állományt tartalmaz, azt azonban látni kell, hogy a beágyazott XML állományok megnevezése nem minden esetben egyezik meg a <messageType> tag értékével. A két információ ugyanakkor egyértelműen megfeleltethető egymásnak.

A fogadó (címzett) fél esetében először egy „Notification” típusú üzenet kerül továbbításra, amely az átveendő üzenetre, illetve az ehhez szükséges lépésekre hívja fel a figyelmet. A „Notification” feldolgozása után a fogadó oldalnak a küldemény elfogadásával kapcsolatos lépéseket kell megtennie, amiről a következő fejezet szól. Ez az üzenet és az elfogadással kapcsolatos lépések a kézbesítési szolgáltatás esetében elmaradnak.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="getMessageResponse" type="tns:getMessageResponse"/>
   <xs:complexType name="getMessageResponse">
      <xs:sequence>
         <xs:element minOccurs="0" name="return" type="tns:message"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="message">
      <xs:sequence>
         <xs:element minOccurs="0" maxOccurs="unbounded" name="attachments" nillable="true" type="tns:attachment"/>
         <xs:element minOccurs="0" name="body" type="xs:string"/>
         <xs:element minOccurs="0" name="consignmentId" type="xs:string"/>
         <xs:element minOccurs="0" name="deliveryType" type="tns:deliveryType"/>
         <xs:element minOccurs="0" name="messageDateTime" type="xs:dateTime"/>
         <xs:element minOccurs="0" name="messageType" type="tns:messageType"/>
         <xs:element minOccurs="0" name="notificationEMail" type="xs:string"/>
         <xs:element minOccurs="0" maxOccurs="unbounded" name="recipients" nillable="true" type="xs:string"/>
         <xs:element minOccurs="0" name="requestId" type="xs:string"/>
         <xs:element minOccurs="0" name="sender" type="xs:string"/>
         <xs:element minOccurs="0" name="subject" type="xs:string"/>
         <xs:element minOccurs="0" name="uid" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="attachment">
      <xs:sequence>
         <xs:element minOccurs="0" name="data" type="xs:base64Binary"/>
         <xs:element minOccurs="0" name="name" type="xs:string"/>
         <xs:element minOccurs="0" name="signatureType" type="tns:signatureType"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="messageType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Consignment"/>
         <xs:enumeration value="Notification"/>
         <xs:enumeration value="DeliveryCertificate"/>
         <xs:enumeration value="DispatchCertificate"/>
         <xs:enumeration value="DownloadCertificate"/>
         <xs:enumeration value="AcceptanceCertificate"/>
         <xs:enumeration value="HybridSuccessReceipt"/>
```

```
        <xs:enumeration value="NonDeliveryCertificate"/>
        <xs:enumeration value="NonDispatchCertificate"/>
        <xs:enumeration value="NonDownloadCertificate"/>
        <xs:enumeration value="NonAcceptanceCertificate"/>
        <xs:enumeration value="HybridFailureReceipt"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="signatureType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="X_AD_ES"/>
        <xs:enumeration value="P_AD_ES"/>
        <xs:enumeration value="C_AD_ES"/>
        <xs:enumeration value="X_AD_ES_T"/>
        <xs:enumeration value="P_AD_ES_T"/>
        <xs:enumeration value="C_AD_ES_T"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="deliveryType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="Simple"/>
        <xs:enumeration value="Trusted"/>
        <xs:enumeration value="HybridConversion"/>
        <xs:enumeration value="HybridConversionAuthenticatedCopy"/>
        <xs:enumeration value="InverseHybridConversion"/>
        <xs:enumeration value="InverseHybridConversionAuthenticatedCopy"/>
      </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

A getMessageResponse sémája

Később a lekérdezés eredménye mind a küldő, mind a fogadó oldalon a küldemény továbbítására igénybe vett szolgáltatásnak megfelelően egy „DeliveryCertificate” vagy egy „DownloadCertificate” <messageType>-pal rendelkező küldemény lehet, amennyiben a küldemény továbbítása sikeres volt. A sikertelen továbbítás megfelelő esetei a "NonDeliveryCertificate", illetve "NonDownloadCertificate" elküldését indítják minden esetben csak a küldő felé. A küldő ezeket az üzeneteket minden esetben a rendszer által adott <consignmentId> értéke alapján tudja egymáshoz, illetve a saját nyilvántartásához rendelni. Segíthet még a szintén minden esetben kezelt <requestId> is, amely a küldő rendszer azonosítóját viszi végig.

Bár ez az üzenet-szerkezet elég sok adminisztrációs feladatot hárít a küldő-fogadó rendszerre, a kompakt üzenetcsomagok miatt van lehetőség többszálas kezelésre is, a szálak nem zavarják egymást.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <soap:Body>
        <ns2:getMessageResponse xmlns:ns2="http://selexes.com/hmdacs">
            <return>
                <attachments>
                    <data>JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9JbnRlbnQvUGVyY2VwdHVhbC9EZWNvZGVQYXJtczw8
                        <!-- itt folyatódik magának a csatolmánynak a base64 kódolt szövege -->
                        byAxNiAwIFIvU2l6ZSAyMj4+CiVpVGV4dC01LjUuNQpzdGFydHhyZWYKNTg3NDkKJSVFT0YK</data>
                    <name>Certificate.pdf</name>
                </attachments>
                <consignmentId>653161</consignmentId>
                <messageDateTime>2017-10-13T12:59:40.398+02:00</messageDateTime>
                <messageType>DispatchCertificate</messageType>
                <recipients>posta2@hibrid.posta.hu</recipients>
                <sender>posta2@hibrid.posta.hu</sender>
                <uid>S_675306</uid>
            </return>
```

```
        </ns2:getMessageResponse>
    </soap:Body>
</soap:Envelope>
```

A getMessageResponse SOAP üzenetének szerkezete (egy Dispatch üzenettel)

5.3.5 Az üzenet fogadásának visszaigazolása: releaseMessage

A lekérdezésre érkező válaszüzenet nyomán vissza kell igazolni a fogadott üzenet hibátlan megérkezését, különben a rendszer a továbbiakban is ugyanazt az üzenetet fogja ismételten a következő kérésre is elküldeni, ami különösen a többszálú kezelés esetén – de egyszeres kapcsolatnál is – szükségtelen erőforrás lekötést okozna. E válasz nyomán az üzenet törlésre kerül a továbbításra sorban állók közül.

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	releaseMessage	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
releaseMessage	Összetett	A lekérdezett és sikeresen rögzített üzenet visszaigazolása és ezzel „elengedésének” kérése a hibrid kézbesítési és konverziós rendszerből

15. táblázat: A releaseMessage parancs alapvető jellemzői

Maga a <releaseMessage> parancs is egyszerű, mindössze a parancsot kiadó, az adott üzenetet már fogadó fél szerződésének azonosítóját, a rendszer által az adott üzenetnek adott azonosítót (lásd a getMessage adatcsomag szerkezetét) és az üzenet (message) típusát tartalmazza. Az utóbbi tulajdonképpen feleslegesnek tűnik, valószínűleg azért került ide, hogy az üzenet ne legyen túl rövid a lenyomatképzés egyértelműségéhez.

Elem név	**Típus**	**Leírás**
contractId	Karaktersorozat	A fogadó fél szerződésének azonosítója
msgUID	Karaktersorozat	a getMessageResponse üzenetben megkapott egyedi azonosító
messageType	Kötött értékkészlet egy eleme	A visszaigazolt üzenet típusa, a 28. táblázat egy eleme

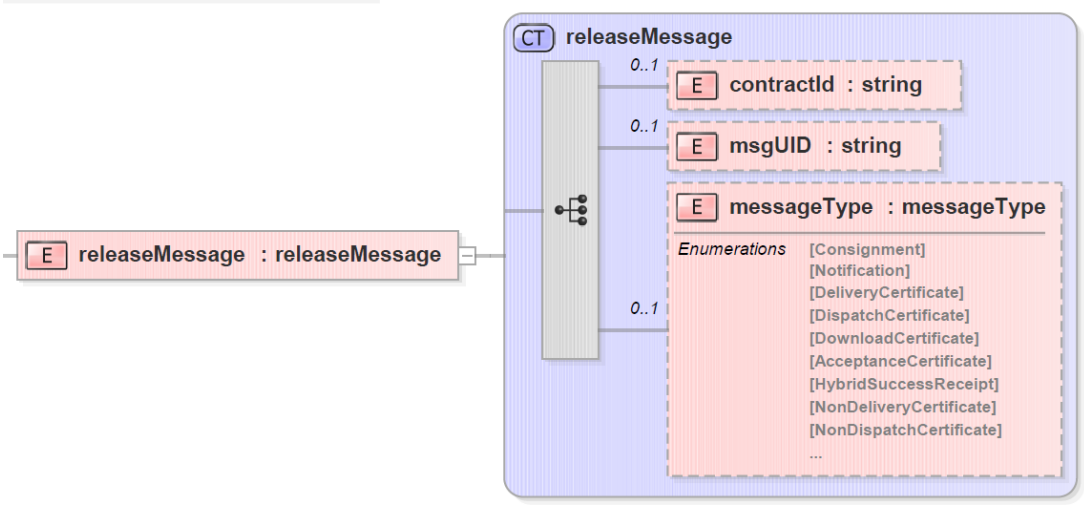
16. táblázat: A releaseMessage adatcsomag elemei

Ennek megfelelően az üzenet sémája is könnyen áttekinthető:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="releaseMessage" type="tns:releaseMessage"/>
   <xs:complexType name="releaseMessage">
      <xs:sequence>
         <xs:element minOccurs="0" name="contractId" type="xs:string"/>
         <xs:element minOccurs="0" name="msgUID" type="xs:string"/>
         <xs:element minOccurs="0" name="messageType" type="tns:messageType"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="messageType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="Consignment"/>
         <xs:enumeration value="Notification"/>
         <xs:enumeration value="DeliveryCertificate"/>
         <xs:enumeration value="DispatchCertificate"/>
         <xs:enumeration value="DownloadCertificate"/>
         <xs:enumeration value="AcceptanceCertificate"/>
         <xs:enumeration value="HybridSuccessReceipt"/>
         <xs:enumeration value="NonDeliveryCertificate"/>
         <xs:enumeration value="NonDispatchCertificate"/>
         <xs:enumeration value="NonDownloadCertificate"/>
         <xs:enumeration value="NonAcceptanceCertificate"/>
         <xs:enumeration value="HybridFailureReceipt"/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

A releaseMessage sémája



18. ábra: A releaseMessage adatcsomag szerkezete

Itt is látható, hogy maga az üzenet teljes egyértelműséggel rögzíti, hogy melyik üzenetet lehet – mint már továbbítottat – kivenni az átvételre várakozó üzenetek közül és ezzel viszonylag kis kockázatúvá tenni a többszálas megvalósítást. A minta az előző példában szereplő DispatchCertificate üzenet visszaigazolásának adattartalmát mutatja.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Header>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
            <SignedInfo>
                <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
                <Reference URI="#Body">
                    <Transforms>
```

```
                        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                    </Transforms>
                    <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <DigestValue>r0xGIyYZPYI7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
                </Reference>
            </SignedInfo>
            <SignatureValue>MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==</SignatureValue>
            <KeyInfo>
                <X509Data>
                    <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi
                        Főigazgatóság,L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=
                        DO20141223-1DO3</X509SubjectName>
        <X509Certificate>MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
<!-- itt folytatódik a tanúsítvány base64 kódolással -->
nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==</X509Certificate>
                </X509Data>
            </KeyInfo>
        </Signature>
    </S:Header>
    <S:Body Id="Body">
        <ns2:releaseMessage xmlns:ns2="http://selexes.com/hmdacs">
            <contractId>3047</contractId>
            <msgUID>S_675306</msgUID>
            <messageType>DispatchCertificate</messageType>
        </ns2:releaseMessage>
    </S:Body>
</S:Envelope>
```

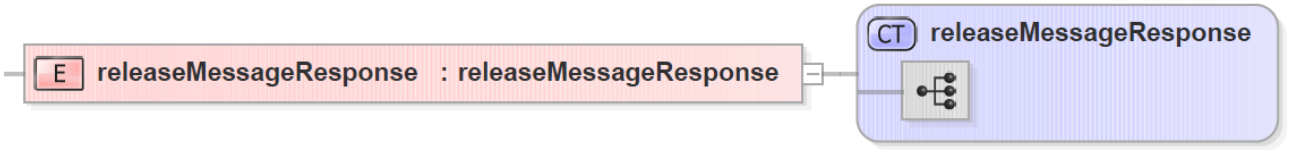
A releaseMessage SOAP üzenetének szerkezete

5.3.6 Válasz az üzenet fogadásának visszaigazolására: releaseMessageResponse

A <releaseMessage> parancs visszaigazolása lényegében egy üres üzenet, az üzenet törzse nem tartalmaz elemet, csak maga a válasz adatcsomag elküldése jelzi, hogy a parancsot vette a rendszer:

Válasz üzenet elemei		
Elem név	**Típus**	**Leírás**
releaseMessageResponse	Összetett	A válasz ténye, maga a visszaigazolás

17. táblázat: A releaseMessageResponse adatcsomag alapstruktúrája



19. ábra: A releaseMessageResponse szerkezete

Ennek megfelelően a séma és a SOAP üzenet is igen egyszerű

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="releaseMessageResponse" type="tns:releaseMessageResponse"/>
  <xs:complexType name="releaseMessageResponse">
    <xs:sequence/>
```

```
   </xs:complexType>
</xs:schema>
```

releaseMessageResponse adatcsomag XML sémája

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:releaseMessageResponse xmlns:ns2="http://selexes.com/hmdacs">
      </ns2:releaseMessageResponse>
   </soap:Body>
</soap:Envelope>
```

A releaseMessageResponse SOAP üzenetének szerkezete

5.4 Az átvételi elismervény kezelésének eljárásai

Az általános *getMessage releaseMessage* parancsokkal történő érkező üzenet kezelés és *sendMessage* paranccsal történő üzenetküldéshez képest a webAutomata esetében külön, specializált eljárással valósították meg a biztonságos kézbesítési szolgáltatáshoz kapcsolódó, az üzenetek átvételének igazolását biztosító, az rendszer által készített átvételi elismervények le- és feltöltését.

Ezek az eljárások kizárólagosan az átvételi elismervények életútjának támogatását szolgálják. Az egyébként elvileg alkalmazható általános érvényű *getMessage, releaseMessage,* illetve *sendMessage* parancsok e célra való használata helyett a külön céleljárások használatának oka minden bizonnyal ennek a megoldásnak a lényegesen kisebb erőforrás-igénye volt.

A rendszer és ennek megfelelően a webAutomata nem ad közvetlen támogatást a letöltött elismervények elektronikus aláírással, bélyegzővel történő ellátásához, azt minden esetben a fogadó rendszernek önállóan kell megoldania.

5.4.1 Átvételi elismervény kérése getAcceptanceCertificate

Amennyiben a *getMessage* útján kapott adatcsomagban a <messageType> elem értéke „Notification” és a <deliveryType> értéke „Trusted”, akkor a *releaseMessage* parancs végrehajtását követően el kell végezni az átvételi elismervény dokumentum lekérését, hogy utóbb megkaphassuk magát a biztonságos kézbesítési szolgáltatással érkezett üzenetet.

Az átvételi elismervény dokumentum megfelelő azonosítási információkkal egyértelműen azonosítja a kézbesítésre felajánlott küldeményt, megadja a kézbesítési folyamatra vonatkozó fontosabb adatokat és bizonyíthatóvá teszi a küldemény tartalmát. Az átvételi elismervény lekérésére a *getAcceptanceCertificate* parancs szolgál.

Parancs definíció	
Név	Érték
Rendszer	**Hibrid kézbesítési és konverziós rendszer**
Modul	webAutomata
Szolgáltatás	getAcceptanceCertificate

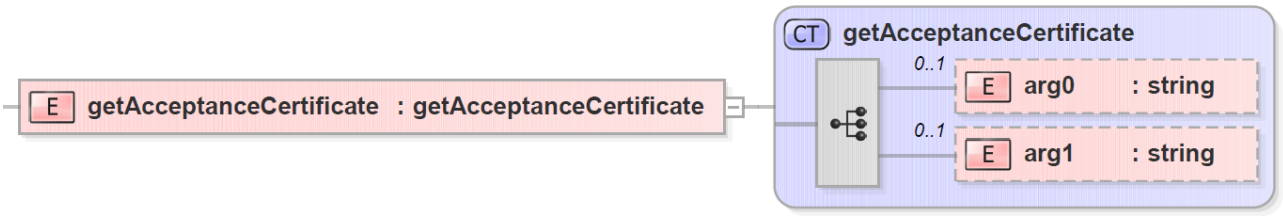
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getAcceptanceCertificate	Összetett	Az azonosított biztonságos kézbesítési szolgáltatással érkező küldemény átvételi elismervényének lekérdezésére irányuló kérés adatai
arg0	Karaktersorozat	A szerződés azonosítója
arg1	Karaktersorozat	Az értesítésben megkapott küldeményazonosító <consignmentId>

18. táblázat: A getAcceptanceCertificate parancs alapvető jellemzői

A getAcceptanceCertificate parancs szerkezete is egyszerű, mindössze két paramétere van, a szerződés azonosítója és az előzőleg, a „Notification” részeként megkapott küldemény-azonosító, a <consignmentId> elem értéke. Sajnálatos, hogy itt a parancs paramétereinek elnevezése nem beszédes. Az adatok XML sémája is egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="getAcceptanceCertificate" type="tns:getAcceptanceCertificate"/>
   <xs:complexType name="getAcceptanceCertificate">
      <xs:sequence>
         <xs:element minOccurs="0" name="arg0" type="xs:string"/>
         <xs:element minOccurs="0" name="arg1" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A getAcceptanceCertificate XML sémája



20. ábra: A getAcceptanceCertificate adatcsomagjának szerkezete

Ennek megfelelően a SOAP üzenetben is ismételten az aláírással kapcsolatos információ jelenti a küldemény túlnyomó részét:

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
```

```
      </SignedInfo>
      <SignatureValue>
        MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
        ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
        cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
        B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
        M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
      </SignatureValue>
      <KeyInfo>
        <X509Data>
          <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi
            Főigazgatóság,L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
          </X509SubjectName>
          <X509Certificate>
            MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
            dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
            <!-- itt folytatódik a tanúsítvány base64 kódolással -->
            nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
            RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
          </X509Certificate>
        </X509Data>
      </KeyInfo>
    </Signature>
  </S:Header>
  <S:Body Id="Body">
    <ns2:getAcceptanceCertificate xmlns:ns2="http://selexes.com/hmdacs">
      <arg0>3047</arg0>
      <arg1>675306</arg1>
    </ns2:getAcceptanceCertificate>
  </S:Body>
</S:Envelope>
```

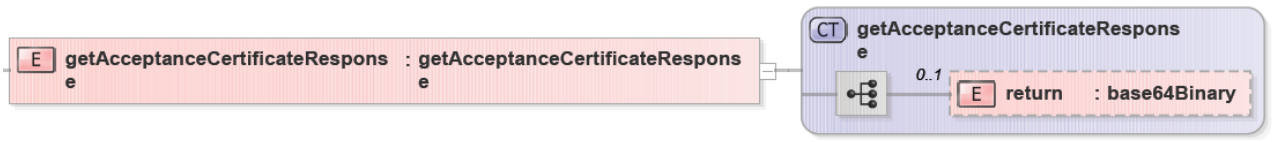
A getAcceptanceCertificate SOAP üzenet szerkezeti mintája

5.4.2 Válasz az átvételi elismervény kérésére getAcceptanceCertificateResponse

A kérésre érkező válaszüzenet helyes adatmegadás esetén tartalmazza a kért átvételi elismervény dokumentumot a 4.2.3 fejezetben már ismertetett tartalommal.

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getAcceptanceCertificateResponse	Összetett	Válasz az átvételi elismervény-kérésre
return	base64 kódolású bitfolyam	Az átvételi elismervény állomány

19. táblázat: A getAcceptanceCertificateResponse adatcsomag alapstruktúrája



21. ábra: A getAcceptanceCertificateResponse adatcsomag szerkezete

A *getAcceptanceCertificateResponse* szerkezete a lehető legegyszerűbb, kizárólag a kért átvételi elismervény dokumentumot tartalmazza mindenfajta további azonosító nélkül, ennek megfelelően itt az üzenetváltás csak kérdés-válaszként értelmezhető. Természetesen az átvételi elismervény maga tartalmaz a kéréssel összevethető, azonosításra alkalmas adatokat, de ahhoz magát a megkapott

átvételi elismervény dokumentumot kell először értelmezni. Az adatcsomag XML sémája is igen egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="getAcceptanceCertificateResponse" type="tns:getAcceptanceCertificateResponse"/>
   <xs:complexType name="getAcceptanceCertificateResponse">
      <xs:sequence>
         <xs:element minOccurs="0" name="return" type="xs:base64Binary"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A getAcceptanceCertificateResponse adatcsomag XML sémája

Az egyszerű szerkezetnek és a válaszüzenet jellegnek megfelelően itt a SOAP üzenet szerkezetileg igen egyszerű:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
          xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:getAcceptanceCertificateResponse xmlns:ns2="http://selexes.com/hmdacs">
         <return>
            JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9JbnRlbnQvUGVyY2VwdHVhbC9EZWNvZGVQYXJtczw8L0NvbG9ycyAzL1By
            <!-- itt folyatódik maga az átvételi elismervény base64 kódolt szövege -->
            SW5mbyAxNiAwIFIvU2l6ZSAyMj4+CiVpVGV4dC01LjUuNQpzdGFydHhyZWYKNTg3NDkKJSVFT0YK
         </return>
      </ns2:getAcceptanceCertificateResponse>
   </soap:Body>
</soap:Envelope>
```

A getAcceptanceCertificateResponse SOAP üzenet szerkezeti mintája

Ha esetleg olyan adatok szerepelnek a kérésben, amelyre vonatkozóan nem létezik átvételi elismervény, akkor a return tag egy hibaüzenetet fog tartalmazni. A hibaüzenet ekkor „NoCertificateFoundException”. Ez akkor következhet be, ha a megtagadásra vonatkozó átvételi elismervényt már előzetesen lekérte a címzett, vagy természetesen, ha az azonosítók nem létező küldeményre mutatnak. A hibát akkor a fogadó rendszernek kell lekezelnie.

5.4.3 Aláírt átvételi elismervény feltöltése sendSignedAcceptanceCertificate

Az elkészített, regisztrált tanúsítvánnyal aláírt átvételi elismervényt fel kell tölteni a webAutomatába, hogy az elvégezhesse annak ellenőrzését, és azok sikere esetén az erre épülő műveletsort elindíthassa.

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	sendSignedAcceptanceCertificate	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**

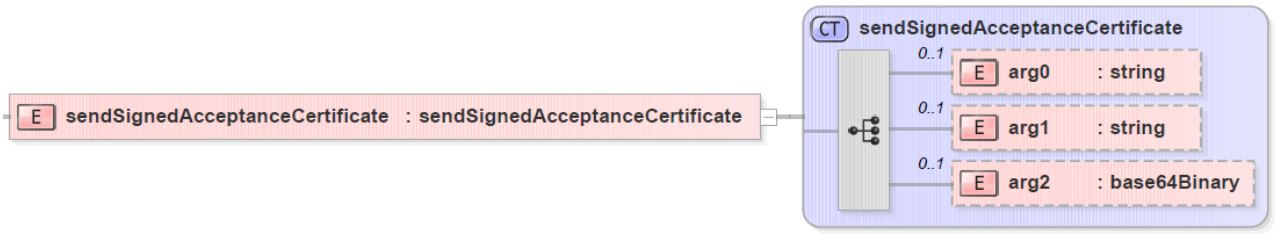
sendSignedAcceptanceCertificate	Összetett	A megkapott és aláírt átvételi elismervény feltöltéséhez szükséges adatcsomag
arg0	Karaktersorozat	A szerződés azonosítója
arg1	Karaktersorozat	Az értesítésben megkapott küldeményazonosító <consignmentId>
arg2	base64 kódolású bitfolyam	Az aláírt átvételi elismervény állomány

20. táblázat: A sendSignedAcceptanceCertificate parancs alapvető jellemzői

Mint látható a parancs a *getAcceptanceCertificate* rokona, mindössze egy adattal bővült, kiegészült magával az aláírt átvételi elismervénnyel. Ennek megfelelően az XML séma is hasonló szerkezetű az forráshoz

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="sendSignedAcceptanceCertificate" type="tns:sendSignedAcceptanceCertificate"/>
   <xs:complexType name="sendSignedAcceptanceCertificate">
      <xs:sequence>
         <xs:element minOccurs="0" name="arg0" type="xs:string"/>
         <xs:element minOccurs="0" name="arg1" type="xs:string"/>
         <xs:element minOccurs="0" name="arg2" type="xs:base64Binary"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A sendSignedAcceptanceCertificate XML sémája



22. ábra: A sendSignedAcceptanceCertificate adatcsomagjának szerkezete

Ennek megfelelően a SOAP üzenet szerkezete is nagyon hasonló:

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYI7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
         <SignatureValue>
            MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
            ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
            cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
```

```
            B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
            M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
         </SignatureValue>
         <KeyInfo>
            <X509Data>
               <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi Főigazgatóság,
                  L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
               </X509SubjectName>
               <X509Certificate>
                  MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
                  dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
                  <!-- itt folytatódik a tanúsítvány base64 kódolással -->
                  nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
                  RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
               </X509Certificate>
            </X509Data>
         </KeyInfo>
      </Signature>
   </S:Header>
   <S:Body Id="Body">
      <ns2:sendSignedAcceptanceCertificate xmlns:ns2="http://selexes.com/hmdacs">
         <arg0>3047</arg0>
         <arg1>675306</arg1>
         <arg2>
            JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9JbnRlbnQvUGVyY2VwdHVhbC9EZWNvZGVQYXJtczw8L0NvbG9ycyAzL1By
            <!-- itt folyatódik az aláírt átvételi elismervény base64 kódolt szövege -->
            SW5mbyAxNiAwIFIvU2l6ZSAyMj4+CiVpVGV4dC01LjUuNQpzdGFydHhyZWYKNTg3NDkKJSVFT0YK
         </arg2>
      </ns2:sendSignedAcceptanceCertificate>
   </S:Body>
</S:Envelope>
```

A sendSignedAcceptanceCertificate SOAP üzenet szerkezeti mintája

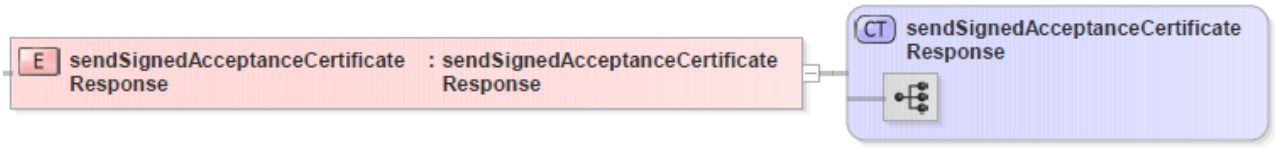
5.4.4 Válasz az aláírt átvételi elismervény feltöltésére sendSignedAcceptanceCertificateResponse

A *sendSignedAcceptanceCertificate* parancs a webAutomata által adott visszaigazolása lényegében egy üres üzenet, az üzenet törzse nem tartalmaz elemet:

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
sendSignedAcceptanceCertificateResponse	Összetett	A válasz ténye, maga a küldés visszaigazolása

21. táblázat: A sendSignedAcceptanceCertificateResponse adatcsomag alapstruktúrája

Ebben a kommunikációban maga az válaszüzenet kliens általi megkapásának ténye hordozza az információt.



23. ábra: A sendSignedAcceptanceCertificateResponse adatcsomag szerkezete

Ennek megfelelően az adatcsomag XML sémája is rendkívül egyszerű,

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="sendSignedAcceptanceCertificateResponse" type="tns:sendSignedAcceptanceCertificateResponse"/>
   <xs:complexType name="sendSignedAcceptanceCertificateResponse">
      <xs:sequence/>
   </xs:complexType>
</xs:schema>
```

A sendSignedAcceptanceCertificateResponse adatcsomag XML sémája

Mivel a SOAP üzenet fejét ebben az esetben nem kell aláírni, annak szerkezete is igen egyszerű

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:sendSignedAcceptanceResponse xmlns:ns2="http://selexes.com/hmdacs">
      </ns2:sendSignedAcceptanceResponse>
   </soap:Body>
</soap:Envelope>
```

A sendSignedAcceptanceCertificateResponse SOAP üzenet mintájának szerkezete

Az aláírt átvételi elismervény sikeres feltöltése és az aláírás megfelelő volta esetén a következő *getMessage* parancsok egyikénél a címzett oldalán meg fog jelenni, mint elérhető, egy „Consignment” értékű <MessageType> elemmel rendelkező üzent, amelyet letöltve megkapjuk magát, a már átvett küldeményt. Majd pedig mind a feladóhoz, mind a címzetthez megérkezik egy-egy „DownloadCertificate” értékű <MessageType> elemmel rendelkező üzenet, ami a mindkét félnek megküldött letöltési igazolás (tértivevény) beleágyazva az aláírt átvételi elismervénnyel. Ezzel a kommunikáció befejeződik.

5.4.5 Visszautasítási elismervény kérése: getNonAcceptanceCertificate

A biztonságos kézbesítési szolgáltatással érkezett küldemények gépi átvétele esetén a rendszer speciális módon biztosít lehetőséget a küldemény visszautasítására.

Annak érdekében, hogy a visszautasítást valamilyen paraméterek mellett vezérelve gépesítetten is meg lehessen oldani, a rendszer egy külön megoldást biztosít, amivel lehetőséget ad arra, hogy a *getAcceptanceCertificate* parancs helyett egy *getNonAcceptanceCertificate* parancs meghívásával olyan tanúsítvány kerüljön letöltésre, amely a küldemény visszautasítását tanúsítja. Ennek aláírásával és visszatöltésével, illetve a „NonDownloadCertificate”-be történő beágyazásával egy teljes értékű bizonyíték jön létre az átvétel visszautasításáról.

Mivel a folyamat eljárásrendi, folyamati megközelítésben teljesen megegyezik az átvételi elismervény webAutomata által támogatott munkafolyamataival (a tanúsítvány tartalmának kivételével), így ennél a négy folyamatlépésnél csak az adatokat mutatjuk be, nem ismételjük a párhuzamos, a küldemény elfogadását jelentő eljárásoknál már leírt megjegyzéseket, magyarázatokat.

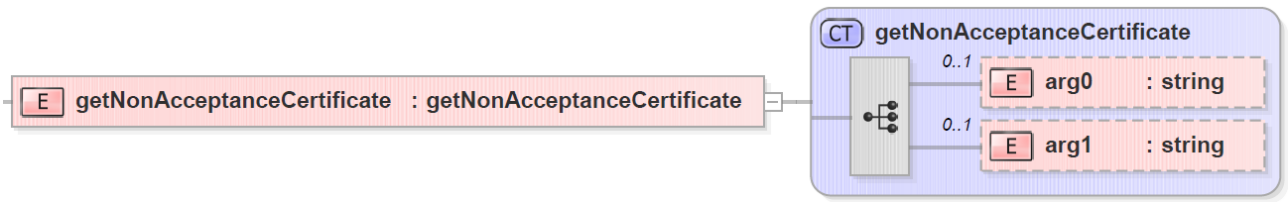
Maga az eljárás igen nagy valószínűséggel nem kerül ténylegesen alkalmazásra, ennek ellenére a folyamat teljességéhez hozzá tartozik.

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	getNonAcceptanceCertificate	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getNonAcceptanceCertificate	Összetett	Az azonosított biztonságos kézbesítési szolgáltatással érkező küldemény nemleges (visszautasító) átvételi elismervényének lekérdezésére irányuló kérés adatai
arg0	Karaktersorozat	A szerződés azonosítója
arg1	Karaktersorozat	Az értesítésben megkapott küldeményazonosító <consignmentId>

22. táblázat: A getNonAcceptanceCertificate parancs alapvető jellemzői

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="getNonAcceptanceCertificate" type="tns:getNonAcceptanceCertificate"/>
   <xs:complexType name="getNonAcceptanceCertificate">
      <xs:sequence>
         <xs:element minOccurs="0" name="arg0" type="xs:string"/>
         <xs:element minOccurs="0" name="arg1" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A getNonAcceptanceCertificate adatcsomag XML sémája



24. ábra: A getNonAcceptanceCertificate adatcsomagjának szerkezete

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYI7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
```

```
        <SignatureValue>
          MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
          ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
          cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
          B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
          M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
        </SignatureValue>
        <KeyInfo>
          <X509Data>
            <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi Főigazgatóság,
                L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
            </X509SubjectName>
            <X509Certificate>
              MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
              dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
              <!-- itt folytatódik a tanúsítvány base64 kódolással -->
              nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
              RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
            </X509Certificate>
          </X509Data>
        </KeyInfo>
      </Signature>
    </S:Header>
    <S:Body Id="Body">
      <ns2:getNonAcceptanceCertificate xmlns:ns2="http://selexes.com/hmdacs">
        <arg0>3047</arg0>
        <arg1>675312</arg1>
      </ns2:getNonAcceptanceCertificate>
    </S:Body>
</S:Envelope>
```

A getNonAcceptanceCertificate SOAP üzenete mintájának szerkezete

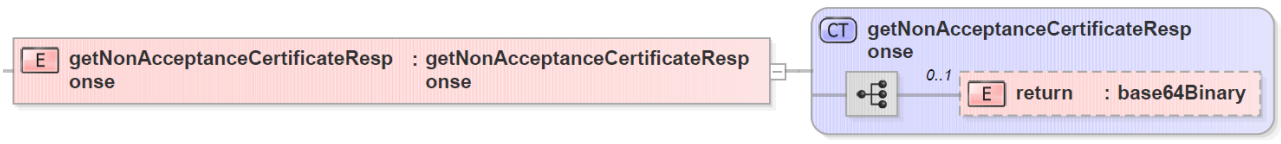
5.4.6 Válasz a visszautasítási elismervény kérésére: getNonAcceptanceCertificateResponse

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
getNonAcceptanceCertificateResponse	Összetett	Válasz a nemleges átvételi elismervény-kérésre
return	base64 kódolású bitfolyam	Az átvételi elismervény állomány

23. táblázat: A getNonAcceptanceCertificateResponse adatcsomag alapvető jellemzői

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="getNonAcceptanceCertificateResponse" type="tns:getNonAcceptanceCertificateResponse"/>
  <xs:complexType name="getNonAcceptanceCertificateResponse">
    <xs:sequence>
      <xs:element minOccurs="0" name="return" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A getNonAcceptanceCertificateResponse adatcsomag XML sémája



25. ábra: A getNonAcceptanceCertificateResponse adatcsomagjának szerkezete

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:getNonAcceptanceCertificateResponse xmlns:ns2="http://selexes.com/hmdacs">
         <return>
            JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9JbnRlbnQvUGVyY2VwdHVhbC9EZWNvZGVQYXJtczw8L0NvbG9ycyAzL1By
            <!-- itt folyatódik maga az elutasítást rögzítő átvételi elismervény base64 kódolt szövege -->
            SW5mbyAxNiAwIFIvU2l6ZSAyMj4+CiVpVGV4dC01LjUuNQpzdGFydHhyZWYKNTg3NDkKJSVFT0YK
         </return>
      </ns2:getNonAcceptanceCertificateResponse>
   </soap:Body>
</soap:Envelope>
```

A getNonAcceptanceCertificateResponse SOAP üzenetének egy mintája

Ezzel a paranccsal tehát megkapjuk magát az elutasítást tartalmazó átvételi elismervényt, amit utána aláírva kell visszajuttatni a webAutomatán keresztül.

5.4.7 Aláírt visszautasítási elismervény feltöltése: sendSignedNonAcceptanceCertificate

Parancs definíció		
Név	Érték	
Rendszer	**Hibrid kézbesítési és konverziós rendszer**	
Modul	webAutomata	
Szolgáltatás	sendSignedNonAcceptanceCertificate	
Kérés adatcsomag elemei		
Elem név	**Típus**	**Leírás**
sendSignedNonAcceptanceCertificate	Összetett	A megkapott és aláírt visszautasítást tanúsító elismervény feltöltéséhez szükséges adatcsomag
arg0	Karaktersorozat	A szerződés azonosítója
arg1	Karaktersorozat	Az értesítésben megkapott küldeményazonosító <consignmentId>
arg2	base64 kódolású bitfolyam	Az aláírt átvételi elismervény állomány

24. táblázat: A sendSignedNonAcceptanceCertificate parancs alapvető jellemzői

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="sendSignedNonAcceptanceCertificate" type="tns:sendSignedNonAcceptanceCertificate"/>
   <xs:complexType name="sendSignedNonAcceptanceCertificate">
      <xs:sequence>
         <xs:element minOccurs="0" name="arg0" type="xs:string"/>
         <xs:element minOccurs="0" name="arg1" type="xs:string"/>
         <xs:element minOccurs="0" name="arg2" type="xs:base64Binary"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

A sendSignedNonAcceptanceCertificate adatcsomag XML sémája

26. ábra: A sendSignedNonAcceptanceCertificate adatcsomagjának szerkezete

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Header>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
         <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256"/>
            <Reference URI="#Body">
               <Transforms>
                  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               </Transforms>
               <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
               <DigestValue>r0xGIyYZPYl7+gFx0a6/KCLvM24ChV8qQFXl2BQr6Jg=</DigestValue>
            </Reference>
         </SignedInfo>
         <SignatureValue>
            MPS9JHiIRIm9VRRfniXLd8TlGcqrh9eZZX+oTCiln9x4dTR1YmRajS8g2gFhKv03oHX8FaR7WAuQ
            ISI/pFakdIu8ORiF5CWL7DZUWLbqEc+gIrrYVZLsaCxemoC+CLXt692qhdX2BZSipjIuWV8lmezI
            cJY4Ad2K1PIRBEyklKffwaYDSj4oK0m73AXbs1mqP7aWkwxgHoKCimq+3tsL0CYCTFJYlMJE1XsI
            B703zcxKUN/xjbtWpN9qZ85fXRx/yitW9UsBdw9UmSwLRiZOFu364jTSy+hDJ/kFb5rocX3ucYX5
            M+Ejk8aYGcyxjcUuvcqjsPj1x1Gzq/09pGWR2A==
         </SignatureValue>
         <KeyInfo>
            <X509Data>
               <X509SubjectName>CN=Bélyegző BM OKF,O=Belügyminisztérium Országos Katasztrófavédelmi Főigazgatóság,
                  L=Budapest, C=HU, 2.5.4.97=VATHU-15722720-2-51, serialNumber=DO20141223-1DO3
               </X509SubjectName>
               <X509Certificate>
                  MIIDbzCCAlcCBFQkFpowDQYJKoZIhvcNAQELBQAwfDEcMBoGCSqGSIb3DQEJARYNbWFyY29AdGVz
                  dC5pdDELMAkGA1UEBhMCSVQxDzANBgNVBAgMBkdlbm92YTEPMA0GA1UEBwwGR2Vub3ZhMRAwDgYD
                  <!-- itt folytatódik a tanúsítvány base64 kódolással -->
                  nQdPeCYIoKPSOXXf2v1X5mrlXCVrTGBSYglgEVup8pgAHOoSrp5P7xz0VfX1daEZjr+A0q0dUWAH
                  RMul/ZZ28RGt3wv/ZhZKvSAYyAIoo6k6Bm8T/g==
               </X509Certificate>
            </X509Data>
         </KeyInfo>
      </Signature>
   </S:Header>
   <S:Body Id="Body">
      <ns2:sendSignedNonAcceptanceCertificate xmlns:ns2="http://selexes.com/hmdacs">
         <arg0>3047</arg0>
         <arg1>675306</arg1>
         <arg2>
            JVBERi0xLjQKJeLjz9MKMiAwIG9iago8PC9JbnRlbnQvUGVyY2VwdHVhbC9EZWNvZGVQYXJtczw8L0NvbG9ycyAzL1By
            <!-- itt folyatódik az aláírt visszautasítást tartalmazó elismervény base64 kódolt szövege -->
            SW5mbyAxNiAwIFIvU2l6ZSAyMj4+CiVpVGV4dC01LjUuNQpzdGFydHhyZWYKNTg3NDkKJSVFT0YK
         </arg2>
      </ns2:sendSignedNonAcceptanceCertificate>
   </S:Body>
</S:Envelope>
```

A sendSignedNonAcceptanceCertificate SOAP üzenetének egy szerkezeti mintája

5.4.8 Válasz az aláírt visszautasítási elismervény feltöltésére: sendSignedNonAcceptanceCertificateResponse

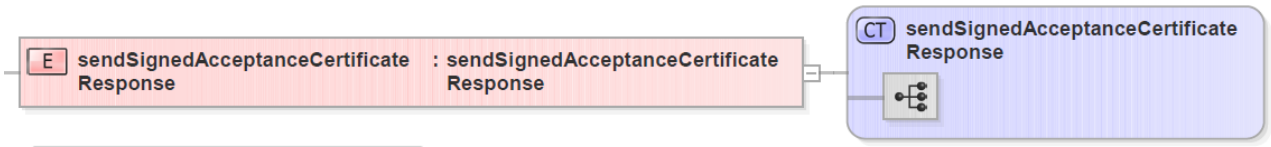
A *sendSignedNonAcceptanceCertificate* parancs a webAutomata által adott visszaigazolása lényegében egy üres üzenet, az üzenet törzse nem tartalmaz elemet:

Válasz adatcsomag elemei		
Elem név	**Típus**	**Leírás**
sendSignedNonAcceptanceCertificateResponse	Összetett	A válasz ténye, maga a küldés visszaigazolása

25. táblázat: A sendSignedNonAcceptanceCertificateResponse adatcsomag alapvető jellemzői

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://selexes.com/hmdacs" version="1.0"
           xmlns:tns="http://selexes.com/hmdacs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="sendSignedNonAcceptanceCertificateResponse" type="tns:sendSignedNonAcceptanceCertificateResponse"/>
   <xs:complexType name="sendSignedNonAcceptanceCertificateResponse">
      <xs:sequence/>
   </xs:complexType>
</xs:schema>
```

A sendSignedNonAcceptanceCertificateResponse adatcsomagjának XML sémája



27. ábra: A sendSignedNonAcceptanceCertificateResponse adatcsomagjának szerkezete

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
           xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <soap:Body>
      <ns2:sendSignedNonAcceptanceResponse xmlns:ns2="http://selexes.com/hmdacs">
      </ns2:sendSignedNonAcceptanceResponse>
   </soap:Body>
</soap:Envelope>
```

A sendSignedNonAcceptanceCertificateResponse SOAP válaszüzenetének szerkezete

A visszautasító üzenet sikeres aláírását és visszajuttatását követően a feladó és a címzett is kap egy-egy „NonDownloadCertificate” értékű <MessageType> elemmel rendelkező üzenet, ami a mindkét félnek megküldött letöltési igazolás (tértivevény), amely jelen esetben a visszautasítást tanúsítja, beleágyazva az aláírt visszautasító elismervénnyel. Ezzel párhuzamosan az üzenet maga törlődik a rendszerből, és a bizonyítékok, illetve a naplózási adatok tartós megőrzésre kerülnek.

5.5 A kivételek kezelése

A fogadó rendszernek fel kell készülnie az esetleges kivételekből származó üzenetek kezelésére is. Szerencsére ezek tulajdonképpen csak a rendszer összehangolásának időszakában jelentenek komolyabb kihívást, hiszen a helyesen megformázott üzenetek esetében nem ezen a szinten kell a hibajelzéseknek előállniuk, azok a különböző igazolások, illetve a felhasználóknak küldött

rendszerüzenetek formájában jelentkeznek. A jelzett kivételek lényegében valamennyi üzenetváltás esetén előfordulhatnak, ezért itt összefoglalva jelenítjük meg azokat. Ez alól az utolsó hibaüzenet jelent kivételt, ez ugyanis az átvételi elismervények lekérésénél fordulhat csak elő. (5.4.2 és 5.4.6 fejezetek)

Négy kivétel-típust deklarál a rendszer

Kivételtípus	A kivétel értelmezése
com.selexes.hmdacs.wa.ContractNotFoundException	Az üzenetben megjelölt szerződésazonosító <contractId> nem létezik
com.selexes.hmdacs.wa.IllegalAddressException	Az üzenetben megjelölt cím nem megfelelő. Ez jelentheti akár a <recipients> egyikének akár a <sender> adatok hibáját a kérésben
com.selexes.hmdacs.wa.IllegalMessageTypeException	Az üzenetben használt üzenet-típus megjelölés nem megfelelő. Az üzenettípus nem felel meg 28. táblázatban leírt <messageType> értékeknek
com.selexes.hmdacs.wa.NoCertificateFoundException	A kért (átvételi) elismervény nem létezik. Akkor fordulhat elő, ha vagy hibás volta kérés címzése, vagy pedig a másik típusú átvételi igazolást már lekérdezték az adott küldeményhez.

26. táblázat: A webAutomata kivételei

Ezekre a kivételekre kell a küldő rendszernek valamilyen módon reagálnia. Értelemszerűen az igazolásokban, üzenetekben található <notification> és <error> üzenetek kezelése jellemzően nem gépi feldolgozást kíván, hanem humán kezelők informálását szolgálja.

Ezen eljárásokkal tehát a webAutomata szolgáltatáskészlete bemutatásra került és láthatóan ez a szolgáltatáscsomag a normál működés valamennyi folyamatát képes támogatni, miután a kezdeti beállítások és összehangolás megtörtént.

1. sz. függelék: A webszerviz XML alapú leírása webAutomata.wsdl

A függelék a teszt környezet webAutomata.wsdl-jét mutatja be, az éles környezet, a cím kivételével megegyezik ezzel

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:ns1="http://schemas.xmlsoap.org/soap/http"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:tns="http://selexes.com/hmdacs"
        xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        name="webAutomata"
        targetNamespace="http://selexes.com/hmdacs">
    <wsdl:types>
        <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified"
                targetNamespace="http://selexes.com/hmdacs" version="1.0">
            <xs:element name="getAcceptanceCertificate" type="tns:getAcceptanceCertificate"/>
            <xs:element name="getAcceptanceCertificateResponse" type="tns:getAcceptanceCertificateResponse"/>
            <xs:element name="getClientConfiguration" type="tns:getClientConfiguration"/>
            <xs:element name="getClientConfigurationResponse" type="tns:getClientConfigurationResponse"/>
            <xs:element name="getMessage" type="tns:getMessage"/>
            <xs:element name="getMessageResponse" type="tns:getMessageResponse"/>
            <xs:element name="getNonAcceptanceCertificate" type="tns:getNonAcceptanceCertificate"/>
            <xs:element name="getNonAcceptanceCertificateResponse" type="tns:getNonAcceptanceCertificateResponse"/>
            <xs:element name="probe" type="tns:probe"/>
            <xs:element name="probeResponse" type="tns:probeResponse"/>
            <xs:element name="releaseMessage" type="tns:releaseMessage"/>
            <xs:element name="releaseMessageResponse" type="tns:releaseMessageResponse"/>
            <xs:element name="sendMessage" type="tns:sendMessage"/>
            <xs:element name="sendMessageResponse" type="tns:sendMessageResponse"/>
            <xs:element name="sendSignedAcceptanceCertificate" type="tns:sendSignedAcceptanceCertificate"/>
            <xs:element name="sendSignedAcceptanceCertificateResponse"
                type="tns:sendSignedAcceptanceCertificateResponse"/>
            <xs:element name="sendSignedNonAcceptanceCertificate" type="tns:sendSignedNonAcceptanceCertificate"/>
            <xs:element name="sendSignedNonAcceptanceCertificateResponse"
                type="tns:sendSignedNonAcceptanceCertificateResponse"/>
            <xs:complexType name="sendSignedAcceptanceCertificate">
                <xs:sequence>
                    <xs:element minOccurs="0" name="arg0" type="xs:string"/>
                    <xs:element minOccurs="0" name="arg1" type="xs:string"/>
                    <xs:element minOccurs="0" name="arg2" type="xs:base64Binary"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="sendSignedAcceptanceCertificateResponse">
                <xs:sequence/>
            </xs:complexType>
            <xs:complexType name="releaseMessage">
                <xs:sequence>
                    <xs:element minOccurs="0" name="contractId" type="xs:string"/>
                    <xs:element minOccurs="0" name="msgUID" type="xs:string"/>
                    <xs:element minOccurs="0" name="messageType" type="tns:messageType"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="releaseMessageResponse">
                <xs:sequence/>
            </xs:complexType>
            <xs:complexType name="sendSignedNonAcceptanceCertificate">
                <xs:sequence>
                    <xs:element minOccurs="0" name="arg0" type="xs:string"/>
                    <xs:element minOccurs="0" name="arg1" type="xs:string"/>
                    <xs:element minOccurs="0" name="arg2" type="xs:base64Binary"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="sendSignedNonAcceptanceCertificateResponse">
                <xs:sequence/>
```

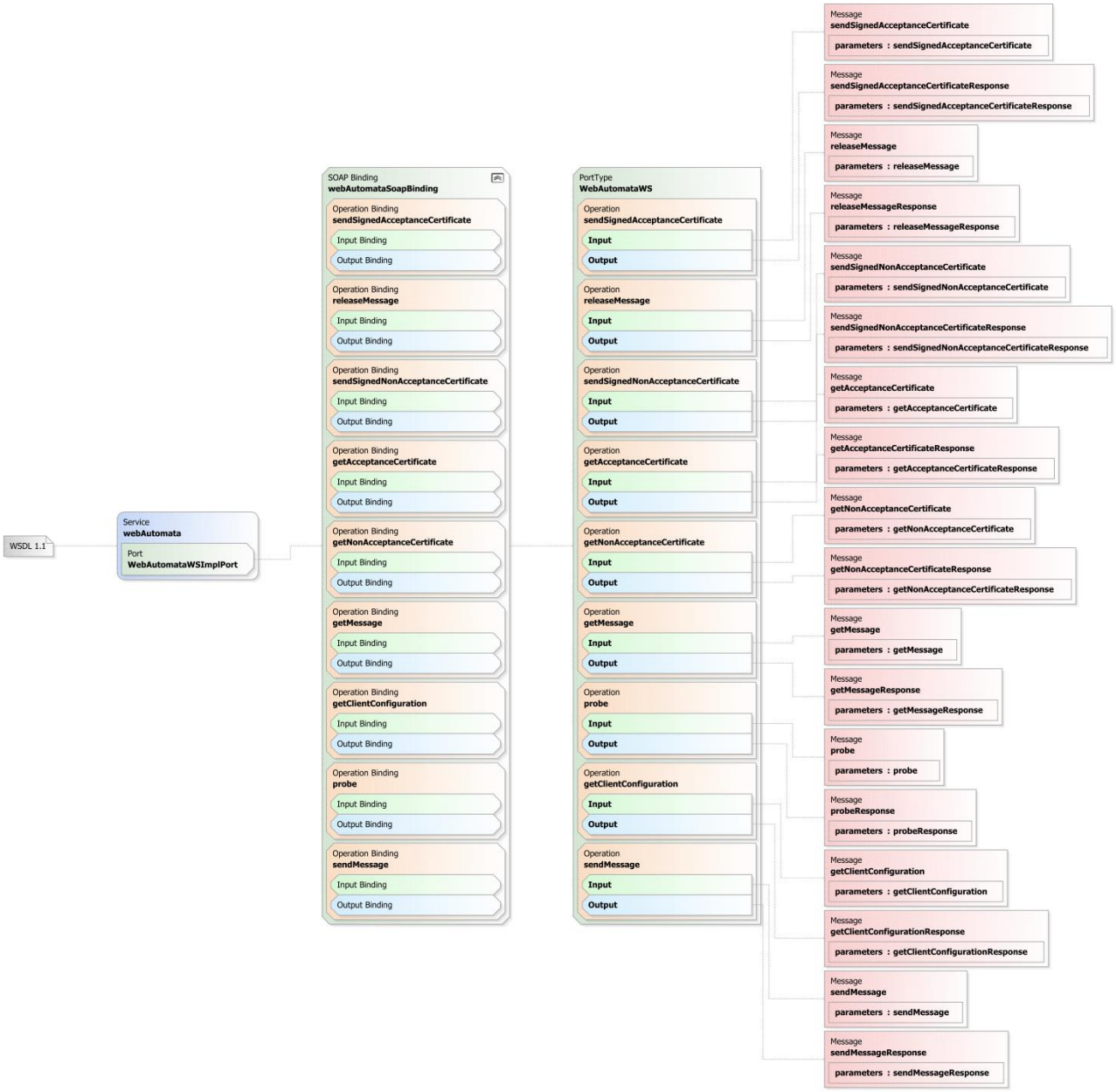
```
</xs:complexType>
<xs:complexType name="getAcceptanceCertificate">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="xs:string"/>
        <xs:element minOccurs="0" name="arg1" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getAcceptanceCertificateResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:base64Binary"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getNonAcceptanceCertificate">
    <xs:sequence>
        <xs:element minOccurs="0" name="arg0" type="xs:string"/>
        <xs:element minOccurs="0" name="arg1" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getNonAcceptanceCertificateResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="xs:base64Binary"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMessage">
    <xs:sequence>
        <xs:element minOccurs="0" name="contractId" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getMessageResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="tns:message"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="message">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="attachments" nillable="true" type="tns:attachment"/>
        <xs:element minOccurs="0" name="body" type="xs:string"/>
        <xs:element minOccurs="0" name="consignmentId" type="xs:string"/>
        <xs:element minOccurs="0" name="deliveryType" type="tns:deliveryType"/>
        <xs:element minOccurs="0" name="messageDateTime" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="messageType" type="tns:messageType"/>
        <xs:element minOccurs="0" name="notificationEMail" type="xs:string"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="recipients" nillable="true" type="xs:string"/>
        <xs:element minOccurs="0" name="requestId" type="xs:string"/>
        <xs:element minOccurs="0" name="sender" type="xs:string"/>
        <xs:element minOccurs="0" name="subject" type="xs:string"/>
        <xs:element minOccurs="0" name="uid" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="attachment">
    <xs:sequence>
        <xs:element minOccurs="0" name="data" type="xs:base64Binary"/>
        <xs:element minOccurs="0" name="name" type="xs:string"/>
        <xs:element minOccurs="0" name="signatureType" type="tns:signatureType"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="probe">
    <xs:sequence/>
</xs:complexType>
<xs:complexType name="probeResponse">
    <xs:sequence/>
</xs:complexType>
<xs:complexType name="getClientConfiguration">
    <xs:sequence>
        <xs:element minOccurs="0" name="contractId" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="getClientConfigurationResponse">
    <xs:sequence>
        <xs:element minOccurs="0" name="return" type="tns:clientConfiguration"/>
```

```
            </xs:sequence>
        </xs:complexType>
        <xs:complexType name="clientConfiguration">
            <xs:sequence>
                <xs:element name="maxAttachmentsSize" type="xs:int"/>
                <xs:element name="minContract" type="xs:int"/>
                <xs:element name="sleepTime" type="xs:long"/>
            </xs:sequence>
        </xs:complexType>
        <xs:complexType name="sendMessage">
            <xs:sequence>
                <xs:element minOccurs="0" name="contractId" type="xs:string"/>
                <xs:element minOccurs="0" name="message" type="tns:message"/>
            </xs:sequence>
        </xs:complexType>
        <xs:complexType name="sendMessageResponse">
            <xs:sequence>
                <xs:element minOccurs="0" name="return" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
        <xs:simpleType name="messageType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="Consignment"/>
                <xs:enumeration value="Notification"/>
                <xs:enumeration value="DeliveryCertificate"/>
                <xs:enumeration value="DispatchCertificate"/>
                <xs:enumeration value="DownloadCertificate"/>
                <xs:enumeration value="AcceptanceCertificate"/>
                <xs:enumeration value="HybridSuccessReceipt"/>
                <xs:enumeration value="NonDeliveryCertificate"/>
                <xs:enumeration value="NonDispatchCertificate"/>
                <xs:enumeration value="NonDownloadCertificate"/>
                <xs:enumeration value="NonAcceptanceCertificate"/>
                <xs:enumeration value="HybridFailureReceipt"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="signatureType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="None"/>
                <xs:enumeration value="X_AD_ES"/>
                <xs:enumeration value="P_AD_ES"/>
                <xs:enumeration value="C_AD_ES"/>
                <xs:enumeration value="X_AD_ES_T"/>
                <xs:enumeration value="P_AD_ES_T"/>
                <xs:enumeration value="C_AD_ES_T"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="deliveryType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="Simple"/>
                <xs:enumeration value="Trusted"/>
                <xs:enumeration value="HybridConversion"/>
                <xs:enumeration value="HybridConversionAuthenticatedCopy"/>
                <xs:enumeration value="InverseHybridConversion"/>
                <xs:enumeration value="InverseHybridConversionAuthenticatedCopy"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:schema>
</wsdl:types>
<wsdl:message name="probeResponse">
    <wsdl:part element="tns:probeResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="sendMessage">
    <wsdl:part element="tns:sendMessage" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="sendMessageResponse">
    <wsdl:part element="tns:sendMessageResponse" name="parameters">
    </wsdl:part>
```

```
</wsdl:message>
<wsdl:message name="sendSignedAcceptanceCertificate">
    <wsdl:part element="tns:sendSignedAcceptanceCertificate" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="sendSignedAcceptanceCertificateResponse">
    <wsdl:part element="tns:sendSignedAcceptanceCertificateResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="sendSignedNonAcceptanceCertificateResponse">
    <wsdl:part element="tns:sendSignedNonAcceptanceCertificateResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="releaseMessageResponse">
    <wsdl:part element="tns:releaseMessageResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="sendSignedNonAcceptanceCertificate">
    <wsdl:part element="tns:sendSignedNonAcceptanceCertificate" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="getClientConfigurationResponse">
    <wsdl:part element="tns:getClientConfigurationResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="getMessageResponse">
    <wsdl:part element="tns:getMessageResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="getAcceptanceCertificateResponse">
    <wsdl:part element="tns:getAcceptanceCertificateResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="releaseMessage">
    <wsdl:part element="tns:releaseMessage" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="getAcceptanceCertificate">
    <wsdl:part element="tns:getAcceptanceCertificate" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="getMessage">
    <wsdl:part element="tns:getMessage" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="getClientConfiguration">
    <wsdl:part element="tns:getClientConfiguration" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="getNonAcceptanceCertificateResponse">
    <wsdl:part element="tns:getNonAcceptanceCertificateResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="getNonAcceptanceCertificate">
    <wsdl:part element="tns:getNonAcceptanceCertificate" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="probe">
    <wsdl:part element="tns:probe" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:portType name="WebAutomataWS">
    <wsdl:operation name="sendSignedAcceptanceCertificate">
        <wsdl:input message="tns:sendSignedAcceptanceCertificate" name="sendSignedAcceptanceCertificate">
        </wsdl:input>
        <wsdl:output message="tns:sendSignedAcceptanceCertificateResponse"
            name="sendSignedAcceptanceCertificateResponse">
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="releaseMessage">
```

```
            <wsdl:input message="tns:releaseMessage" name="releaseMessage">
            </wsdl:input>
            <wsdl:output message="tns:releaseMessageResponse" name="releaseMessageResponse">
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="sendSignedNonAcceptanceCertificate">
            <wsdl:input message="tns:sendSignedNonAcceptanceCertificate" name="sendSignedNonAcceptanceCertificate">
            </wsdl:input>
            <wsdl:output message="tns:sendSignedNonAcceptanceCertificateResponse"
                name="sendSignedNonAcceptanceCertificateResponse">
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getAcceptanceCertificate">
            <wsdl:input message="tns:getAcceptanceCertificate" name="getAcceptanceCertificate">
            </wsdl:input>
            <wsdl:output message="tns:getAcceptanceCertificateResponse" name="getAcceptanceCertificateResponse">
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getNonAcceptanceCertificate">
            <wsdl:input message="tns:getNonAcceptanceCertificate" name="getNonAcceptanceCertificate">
            </wsdl:input>
            <wsdl:output message="tns:getNonAcceptanceCertificateResponse" name="getNonAcceptanceCertificateResponse">
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getMessage">
            <wsdl:input message="tns:getMessage" name="getMessage">
            </wsdl:input>
            <wsdl:output message="tns:getMessageResponse" name="getMessageResponse">
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="probe">
            <wsdl:input message="tns:probe" name="probe">
            </wsdl:input>
            <wsdl:output message="tns:probeResponse" name="probeResponse">
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getClientConfiguration">
            <wsdl:input message="tns:getClientConfiguration" name="getClientConfiguration">
            </wsdl:input>
            <wsdl:output message="tns:getClientConfigurationResponse" name="getClientConfigurationResponse">
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="sendMessage">
                <wsdl:input message="tns:sendMessage" name="sendMessage">
            </wsdl:input>
            <wsdl:output message="tns:sendMessageResponse" name="sendMessageResponse">
            </wsdl:output>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="webAutomataSoapBinding" type="tns:WebAutomataWS">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="sendSignedAcceptanceCertificate">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="sendSignedAcceptanceCertificate">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="sendSignedAcceptanceCertificateResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="releaseMessage">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="releaseMessage">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="releaseMessageResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="sendSignedNonAcceptanceCertificate">
```

```
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="sendSignedNonAcceptanceCertificate">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="sendSignedNonAcceptanceCertificateResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getAcceptanceCertificate">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="getAcceptanceCertificate">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="getAcceptanceCertificateResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getNonAcceptanceCertificate">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="getNonAcceptanceCertificate">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="getNonAcceptanceCertificateResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getMessage">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="getMessage">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="getMessageResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getClientConfiguration">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="getClientConfiguration">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="getClientConfigurationResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="probe">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="probe">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="probeResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="sendMessage">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input name="sendMessage">
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="sendMessageResponse">
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="webAutomata">
        <wsdl:port binding="tns:webAutomataSoapBinding" name="WebAutomataWSImplPort">
            <soap:address location="http://webautomata.hibrid.uat.posta.hu:8888/web-automata-ws-server/webAutomata"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```



28. ábra: A webszerviz strukturális ábrája

2. sz. függelék: A webAutomata websecurity leírása

```
<wsp:Policy xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:sec="http://schemas.xmlsoap.org/ws/2002/12/secext" >
    <wsp:ExactlyOne>
        <wsp:All>
            <sp:AsymmetricBinding xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
                <wsp:Policy>
                    <sp:InitiatorToken>
                        <wsp:Policy>
                            <sp:X509Token sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient">
                                <wsp:Policy>
                                    <sp:WssX509V3Token10/>
                                </wsp:Policy>
                            </sp:X509Token>
                        </wsp:Policy>
                    </sp:InitiatorToken>
                    <sp:RecipientToken>
                        <wsp:Policy>
                            <sp:X509Token sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/Never">
                                <wsp:Policy>
                                    <sp:WssX509V3Token10/>
                                </wsp:Policy>
                            </sp:X509Token>
                        </wsp:Policy>
                    </sp:RecipientToken>
                    <sp:AlgorithmSuite>
                        <wsp:Policy>
                            <sp:Basic256Sha256/>
                        </wsp:Policy>
                    </sp:AlgorithmSuite>
                    <sp:SignedParts>
                        <sp:Body/>
                    </sp:SignedParts>
                </wsp:Policy>
            </sp:AsymmetricBinding>
        </wsp:All>
    </wsp:ExactlyOne>
</wsp:Policy>
```

2.1 A leírásban használt fogalmak magyarázata

Az itt használt fogalmak az OASIS szabványosító csoport által 2005 júliusában kiadott Web Services Security Policy Language 1.1 verziójának felhasználásával kerültek leírásra, amely a http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf címen érhető el.

Az "*asymmetric binding policy*" (aszimmetrikus hozzárendelési rend) a SOAP üzenetek védelmét aszimmetrikus kulcsú algoritmusok használatával (nyilvános és magánkulcsok kombinációjával) kívánja megoldani. Ennek a modellnek megfelelően alakul az egyes elemek használata. (Más megoldások is léteznek, de a szállító ezt a megoldást választotta)

A "*WssX509V3Token10*" azt mutatja. hogy egy X509 Version 3 szerinti tokent kell alkalmazni a WSS: X509 Token Profile 1.0. szerint, amely dokumentum elérhető a https://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf címen.

A kezdeményező oldali tokenekre (sp:InitiatorToken) vonatkozó http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient attribútum az "sp:X509Token" tag "sp:IncludeToken" attribútumában azt mutatja, hogy az X.509 tanúsítványt minden a kezdeményezőtől (jelen esetben a küldő klienstől) a fogadóhoz (jelen esetben

a fogadó szerverhez, a hibrid rendszerhez) küldött üzenethez hozzá KELL kapcsolni. Ezzel biztosítja a rendszer minden esetben a fogadott üzenet hitelességét.

Ezzel szemben a fogadó oldali tokenekre (*sp:RecipientToken*) vonatkozó http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/Never attribútum az "*sp:X509Token*" tag "*sp:IncludeToken*" attribútumában azt mutatja, hogy az X.509 tanúsítványt a fogadótól (jelen esetben a fogadó szervertől, a hibrid rendszerből) a kezdeményezőhöz (jelen esetben a kapcsolódó kliensnek)a küldött üzenethez nem kell hozzákapcsolni. Itt az üzenetek elégséges biztonságát, az üzenetet amúgy is kísérő tanúsítványok megfelelően biztosítják

Az "*sp:AlgorithmSuite*" tagban a "*Basic256SHA256*" érték azt mutatja, hogy a kommunikáció során a lenyomatképző algoritmus az *SHA256*. (az első tagban szereplő AES 256, itt nem kerül használatba)

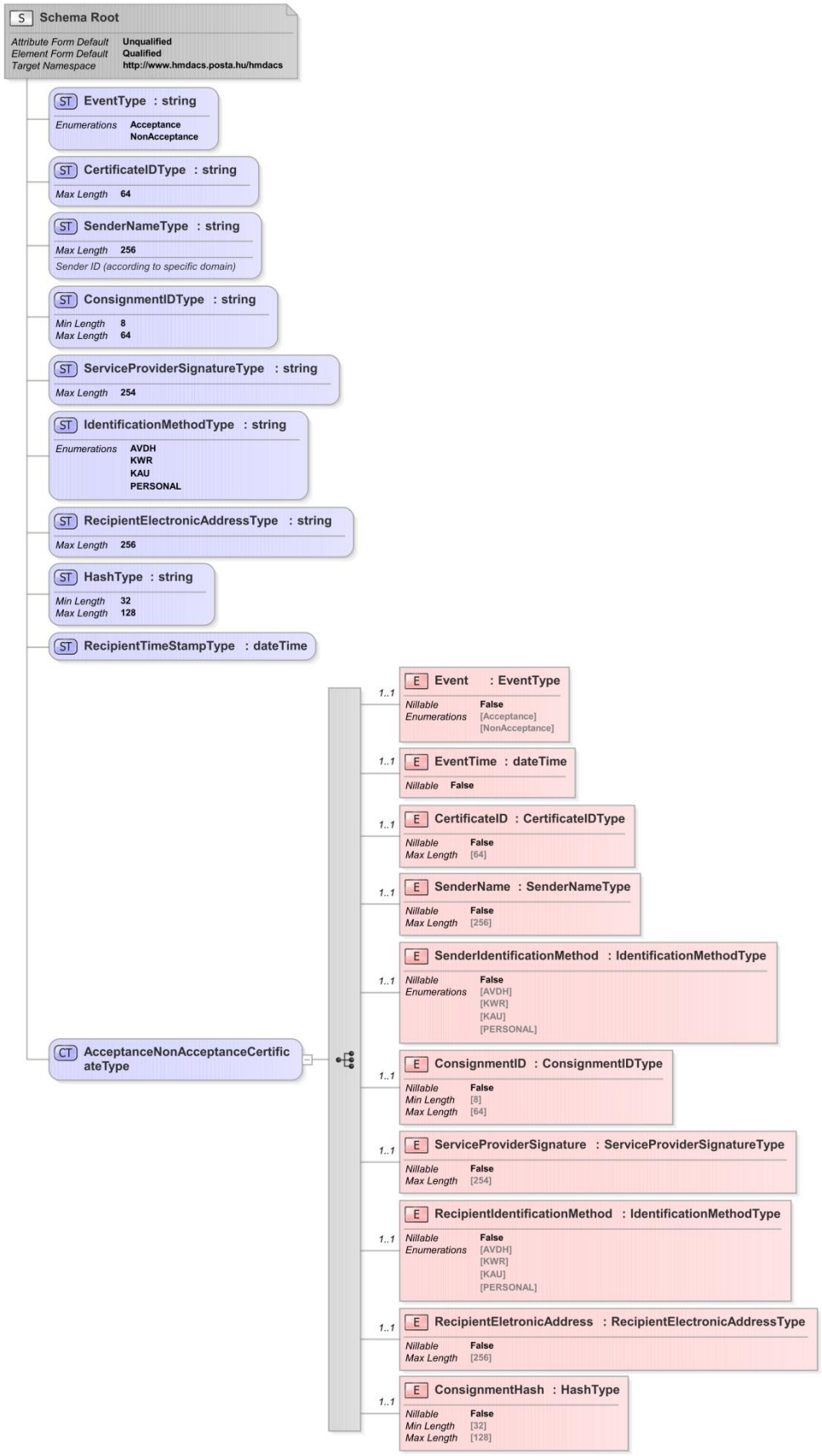
Az "*sp:SignedParts*" tagban szereplő állítás határozza meg, hogy az üzenet mely részei igényelnek integritásvédelmet az üzenet fején kívül. Esetünkben ez a SOAP üzenet "*Body*" (törzs) része, amelyet az aláírásnak védenie kell, és ezt a hivatkozást jelzi is az üzenet aláírást tartalmazó feje.

3. sz. függelék: Az átvételi elismervény sémája

AcceptanceNonAcceptanceCertificate.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns:hmdacs="http://www.hmdacs.posta.hu/hmdacs"
                xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
                attributeFormDefault="unqualified"
                elementFormDefault="qualified"
                targetNamespace="http://www.hmdacs.posta.hu/hmdacs">
    <xs:simpleType name="EventType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Acceptance"/>
            <xs:enumeration value="NonAcceptance"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="CertificateIDType">
        <xs:restriction base="xs:string">
            <xs:maxLength value="64"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="SenderNameType">
        <xs:annotation>
            <xs:documentation>Sender ID (according to specific domain)</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:maxLength value="256"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ConsignmentIDType">
        <xs:restriction base="xs:string">
            <xs:minLength value="8"/>
            <xs:maxLength value="64"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ServiceProviderSignatureType">
        <xs:restriction base="xs:string">
            <xs:maxLength value="254"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="IdentificationMethodType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="AVDH"/>
            <xs:enumeration value="KWR"/>
            <xs:enumeration value="KAU"/>
            <xs:enumeration value="PERSONAL"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="RecipientElectronicAddressType">
        <xs:restriction base="xs:string">
            <xs:maxLength value="256"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="HashType">
        <xs:restriction base="xs:string">
            <xs:minLength value="32"/>
            <xs:maxLength value="128"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="RecipientTimeStampType">
        <xs:restriction base="xs:dateTime"/>
    </xs:simpleType>
    <xs:complexType name="AcceptanceNonAcceptanceCertificateType">
        <xs:sequence>
            <xs:element name="Event" type="hmdacs:EventType" minOccurs="1" maxOccurs="1" nillable="false"/>
            <xs:element name="EventTime" type="xs:dateTime" minOccurs="1" maxOccurs="1" nillable="false"/>
            <xs:element name="CertificateID" type="hmdacs:CertificateIDType" minOccurs="1" maxOccurs="1" nillable="false"/>
            <xs:element name="SenderName" type="hmdacs:SenderNameType" minOccurs="1" maxOccurs="1" nillable="false"/>
```

```
            <xs:element name="SenderIdentificationMethod" type="hmdacs:IdentificationMethodType" minOccurs="1"
                maxOccurs="1" nillable="false"/>
            <xs:element name="ConsignmentID" type="hmdacs:ConsignmentIDType" minOccurs="1" maxOccurs="1" nillable="false"/>
            <xs:element name="ServiceProviderSignature" type="hmdacs:ServiceProviderSignatureType" minOccurs="1"
                maxOccurs="1" nillable="false"/>
            <xs:element name="RecipientIdentificationMethod" type="hmdacs:IdentificationMethodType" minOccurs="1"
                maxOccurs="1" nillable="false"/>
            <xs:element name="RecipientEletronicAddress" type="hmdacs:RecipientElectronicAddressType" minOccurs="1"
                maxOccurs="1" nillable="false"/>
            <xs:element name="ConsignmentHash" type="hmdacs:HashType" minOccurs="1" maxOccurs="1" nillable="false"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```


Schema Root
Attribute Form Default Unqualified
Element Form Default Qualified
Target Namespace http://www.hmdacs.posta.hu/hmdacs
EventType : string
Enumerations Acceptance
NonAcceptance
CertificateIDType : string
Max Length 64
SenderNameType : string
Max Length 256
Sender ID (according to specific domain)
ConsignmentIDType : string
Min Length 8
Max Length 64
ServiceProviderSignatureType : string
Max Length 254
IdentificationMethodType : string
Enumerations AVDH
KWR
KAU
PERSONAL
RecipientElectronicAddressType : string
Max Length 256
HashType : string
Min Length 32
Max Length 128
RecipientTimeStampType : dateTime
AcceptanceNonAcceptanceCertificateType
1..1 Event : EventType
Nillable False
Enumerations [Acceptance]
[NonAcceptance]
1..1 EventTime : dateTime
Nillable False
1..1 CertificateID : CertificateIDType
Nillable False
Max Length [64]
1..1 SenderName : SenderNameType
Nillable False
Max Length [256]
1..1 SenderIdentificationMethod : IdentificationMethodType
Nillable False
Enumerations [AVDH]
[KWR]
[KAU]
[PERSONAL]
1..1 ConsignmentID : ConsignmentIDType
Nillable False
Min Length [8]
Max Length [64]
1..1 ServiceProviderSignature : ServiceProviderSignatureType
Nillable False
Max Length [254]
1..1 RecipientIdentificationMethod : IdentificationMethodType
Nillable False
Enumerations [AVDH]
[KWR]
[KAU]
[PERSONAL]
1..1 RecipientEletronicAddress : RecipientElectronicAddressType
Nillable False
Max Length [256]
1..1 ConsignmentHash : HashType
Nillable False
Min Length [32]
Max Length [128]

29. ábra: Az átvételi elismervény sémájának struktúrája

4. sz. függelék: A kötött elemek listája és értelmezésük

A 4.1-4.3 alcímeken elérhető táblázatok a webAutomata.wsdl-ben található felsorolások értelmezését hivatottak biztosítani

4.1 deliveryType

Elem név	Leírás
Simple	Kézbesítési szolgáltatás
Trusted	Biztonságos kézbesítési szolgáltatás
HybridConversion	Elektronikus dokumentum átalakítása papír alapú dokumentummá és erre vonatkozó igény esetén postai feladása
HybridConversionAuthenticatedCopy	Elektronikus dokumentum átalakítása hiteles papíralapú dokumentummá és erre vonatkozó igény esetén postai feladása
InverseHybridConversion	Papíralapú dokumentum átalakítása elektronikus dokumentummá és annak továbbítása elektronikus úton
InverseHybridConversionAuthenticatedCopy	Papíralapú dokumentum hiteles átalakítása elektronikus dokumentummá és annak továbbítása elektronikus úton a hitelesség biztosításával

27. táblázat: A deliveryType felvehető értékei és értelmezésük

4.2 messageType

Elem név	Leírás
Consignment	Küldemény – az elküldendő ügyfél üzenet típusa
Notification	Értesítés – a rendszer által küldött, nem bizonyíték jellegű közlés
DeliveryCertificate	Kézbesítési igazolás a kézbesítés szolgáltatás esetén a dokumentum a címzett rendelkezésére bocsátásáról kiadott elektronikus dokumentum
DispatchCertificate	Felvételi igazolás – a feladóvevénynek megfelelő elektronikus dokumentum
DownloadCertificate	Letöltési igazolás a biztonságos kézbesítési szolgáltatásnál a kézbesítést igazoló, az aláírt átvételi elismervényt tartalmazó elektronikus dokumentum, a tértivevény megfelelője
AcceptanceCertificate	Átvételi elismervény – az aláírandó elektronikus dokumentum biztonságos kézbesítési szolgáltatás esetén
HybridSuccessReceipt	Hibrid küldemény sikeres gyártása esetén kiküldött elektronikus dokumentum
NonDeliveryCertificate	Kézbesítési szolgáltatás esetén a felvett küldemény továbbításának meghiúsulását és annak okát tanúsító elektronikus dokumentum

Elem név	Leírás
NonDispatchCertificate	A felvétel meghiúsulását és annak okát tanúsító elektronikus dokumentum
NonDownloadCertificate	Biztonságos kézbesítési szolgáltatás esetén a küldemény kézbesítésének meghiúsulást és annak okát, illetve a kézbesítési vélelem megállapításához szükséges információt tartalmazó elektronikus dokumentum
NonAcceptanceCertificate	Kizárólag gép-gép kapcsolat esetén a küldemény átvételének megtagadását igazoló elektronikus dokumentum
HybridFailureReceipt	A hibrid gyártás meghiúsulását és annak okát tanúsító elektronikus dokumentum

28. táblázat: A messageType felvehető értékei és értelmezésük

4.3 signatureType

Elem név	Leírás
None	Nincs aláírás vagy bélyegző
X_AD_ES	A Bizottság (EU) 2015/1506 végrehajtási határozata követelményeinek megfelelő, legalább az ETSI TS 103 171 v2.1.1 követelményeinek eleget tevő XAdES formátumú elektronikus aláírás vagy bélyegző
P_AD_ES	A Bizottság (EU) 2015/1506 végrehajtási határozata követelményeinek megfelelő, legalább az ETSI TS 103 172 v2.2.2 követelményeinek eleget tevő PAdES formátumú elektronikus aláírás vagy bélyegző
C_AD_ES	A Bizottság (EU) 2015/1506 végrehajtási határozata követelményeinek megfelelő, legalább az ETSI TS 103 173 v2.2.1 követelményeinek eleget tevő CAdES formátumú elektronikus aláírás vagy bélyegző
X_AD_ES_T	A Bizottság (EU) 2015/1506 végrehajtási határozata követelményeinek megfelelő, legalább az ETSI TS 103 171 v2.1.1 követelményeinek eleget tevő XAdES formátumú elektronikus aláírás vagy bélyegző és időbélyegzés
P_AD_ES_T	A Bizottság (EU) 2015/1506 végrehajtási határozata követelményeinek megfelelő, legalább az ETSI TS 103 172 v2.2.2 követelményeinek eleget tevő PAdES formátumú elektronikus aláírás vagy bélyegző és időbélyegzés
C_AD_ES_T	A Bizottság (EU) 2015/1506 végrehajtási határozata követelményeinek megfelelő, legalább az ETSI TS 103 173 v2.2.1 követelményeinek eleget tevő CAdES formátumú elektronikus aláírás vagy bélyegző és időbélyegzés

29. táblázat: A signatureType felvehető értékei és értelmezésük

A 4.4 alcímen elérhető táblázat az AcceptanceNonAcceptanceCertificate.xsd-ben található felsorolás értelmezését segíti:

4.4 IdentificationMethodType

Elem név	Leírás
AVDH	Azonosításra Visszavezetett Dokumentumhitelesítés (származtatott azonosítás)
KWR	A Magyar Posta Zrt. saját azonosítási rendszere útján történt azonosítás
KAU	A Központi Azonosítási Ügynök használatával történt azonosítás
PERSONAL	Személyesen történt azonosítás

30. táblázat: Az IdentificationMethodType felvehető értékei és értelmezésük

5. sz. függelék: A webAutomata kliens (WebAPI) bemutatása

A Selex a rendszer részeként leszállított egy Java-ban fejlesztett API-t, ami alkalmas a webAutomata megszólításával (lényegében annak a parancsainak az interpretálásával) a kézbesítési és biztonságos kézbesítési üzenetváltás egyes lépéseit megvalósítani. Az API leírását a szállító az RJ4A18303412 számú dokumentumban adta át. Ennek ismerete a használathoz elengedhetetlen, azonban igen hasznos a programból elérhető javadoc állomány is, amely sok kérdésben támpontot tud adni (angolul)
A Web API bármely olyan operációs rendszer alatt telepíthető, amelyen telepítve van és fut az Oracle Java Virtual Machine 1.7-es vagy újabb (felülről kompatibilis) verziója. (A Java 9. verziójának támogatása még nem biztosított)

A webAutomata által egyébként is megkövetelt biztonsági környezethez a kliens telepítéséhez az alábbi követelményeknek kell megfelelnie:

- Rendelkezésre kell állnia egy JKS állománynak, amelyben a megbízható EU közzétevők listáján szereplő magyar és/vagy európai minősített hitelesítés szolgáltató által kibocsátott egy X.509 v3 szabvány szerinti tanúsítványt (benne magánkulcsot) kell tartalmazzon a hitelesítés-szolgáltató által aláírva. Ezt kliens oldalon egy KeyStore állományban kell elhelyezni. A KeyStore helye egyike a web API által igényelt paramétereknek;
- Az aláírás nyilvános kulcsát a Magyar Posta rendelkezésére kell bocsátani, mivel azt hozzá kell rendelni a hibrid szolgáltatási szerződéshez, ugyanis ezt használva történik majd a kapott SOAP üzenetek hitelességének ellenőrzése a webAutomata szolgáltatásainak igénybe vételével, amihez a kibocsátó hitelesítés szolgáltatót és a tanúsítványt konfigurálni kell a rendszerben;
- Az igénybe vevő által telepített API-nak el kell érnie a webAutomata szolgáltatásait (Magyar Posta megfelelő HMDACS-szerverét), amihez a kapcsolatot be kell állítani a Magyar Posta tűzfalain. (ehhez ismerni kell a kapcsolódni kívánó szolgáltatás IP címét.)
- A Magyar Posta HMDACS-szerverét azonosító X.509 tanúsítványt tartalmazó TrustStore JKS állománynak rendelkezésre kell állnia. Egy tipikus konfigurációban a TrustStore fájlneve `cacerts`, a teljes elérési útvonal pedig

  ```
  $JAVA_HOME/lib/security/cacerts
  ```

  de a fájl helye módosítható a `javax.net.ssl.trustStore` rendszer-tulajdonság módosításával.

A WebAPI egy `web-automata-client-jars.zip` elnevezésű zip fájlban kerül kiadásra, amely tartalmazza a kliens számítógép operációs rendszerének `classpath` paraméteréhez hozzáadandó következő jar fájlokat:

- `web-automata-client.jar`

- `Web-automata-objectmodel.jar`
- `Web-automata-client-javadoc.jar` opcionális
- `Web-automata-ws-client.jar`
- `Web-automata-interface.jar`

A zip fájl egy teszt könyvtárat is tartalmaz, amelyben a Web Automata API használatát bemutató forráskód minta található.