KKSZB magas szintű szoftverarchitektúra

Publikus – IdomSoft Zrt.

Verzió v1.3.0, 2023-06-12

Tartalomjegyzék

Dokumentum kontroll

Dokumentum jellemzők

Projekt hivatalos neve	Közigazgatási szakrendszerek egységes eléréséhez és interoperabilitásához központi alkalmazás szintű szolgáltatások biztosítása
Projekt rövid neve	KAK SW
Projektazonosító	KÖFOP-1.0.0-VEKOP-15-2016-00025
Dokumentum címe	KKSZB magas szintű szoftverarchitektúra
Dokumentum azonosítója	kkszb-docs-$Id:$
Verziószám	v1.3.0
Állapot	Átadott
Kiadás kelte	2023-06-12
Utolsó mentés kelte	2023-06-12
Készítette	Juhász Erzsébet, Pató István
Fájlnév	kkszb_magas_szintu_szoftverarchitektura-v1.3.0.pdf
Dokumentum célja	A KKSZB rendszer magas szintű szoftverarchitektúrájának bemutatása.

Változások jegyzéke

Verzió	Dátum	Változtatás rövid leírása
v1.3.0	2023.06.12	Mongo adatbázisra áttéréssel bevezetett új funkciók kapcsán dokumentum frissítése
v1.2.0	2023.02.01	Dokumentum aktualizálása
v1.1.0	2020.09.03	Bevezetésre kerül a KKSZB rate limit funkció KKSZB rate limit
v1.0.3	2018.12.18	Útvonalválasztás fejezettel bővült, magyarázat a Szolgáltatás URL-jének legfontosabb adatairól és verziózásának kérdésköréről
v1.0.2	2017.11.11	Csatlakozás fejezet kiegészítve a KKSZB környezetekkel, amelyhez csatlakozni lehet.
v1.0.1	2017.03.20	A Szolgáltatás Audit rendszer fejezet bővítése: nem szükséges teljesíteni az elvárásokat jelenleg. A Tanúsítványok függelék bővítése: a Szolgáltatás nyújtónak kell beszereznie a szolgáltatásához a webszerver tanúsítványát.

v1.0.0	2016.12.15	Első, publikus kiadás.

Kapcsolódó dokumentumok

Dokumentum neve	Kapcsolat tartalma - helye

Dokumentum használata

A dokumentum verziószámmal és készítési dátummal rendelkezik, kísérje figyelemmel a verziók változását.

A 0.x formájú verziók **munkaverziót** jelentenek, így azok kidolgozás alatt állnak, csak saját felelősségre használja!

Amennyiben a dokumentumban ezt a jelzést látja  úgy azon a ponton a dokumentum kidolgozás vagy egyeztetés alatt áll. Jellemzően a v0.x verziójú munkaanyagokban talál ilyet.

A kék színben megjelenő, aláhúzott szavak referenciák a dokumentumon belül.

A dokumentáció tartalmaz Definíciókat, [Hivatkozások]at és Függelékeket. A dokumentumban szereplő fogalmak jobb megértéséhez a Definíciókban talál rövid fogalom magyarázatot. A Hivatkozások és Függelékek bővebb információt adnak egy-egy fogalomról.

Használt jelzések:

Megjegyzés, további információ.

Tanács, tipp.

Fontos figyelmeztetés.

A dokumentumban használt angol szavak a megértést segítik. Ott használjuk, ahol a magyar változat félreérthető lenne, vagy nem honosult meg annak használata a szakmai nyelvben.

A dokumentumban az alábbi értelemben használunk bizonyos szavakat:

kell, kötelező: az adott mondatban szereplő feltételek, elvárások teljesítése mindenképpen szükséges a KKSZB használatához, üzemszerű működéséhez.

ajánlott, javasolt: az adott mondatban szereplő feltételek, elvárások teljesítése nem szükséges a KKSZB használatához, üzemszerű működéséhez, de nagyban megkönnyíti, vagy egyszerűsíti az egyéb folyamatokat, vagy elvégzendő feladatokat (pl.: hibakeresés).

opcionális, igény szerint: az adott mondatban szereplő feltételek, elvárások teljesítése a felhasználó saját döntésétől függ.

Tárgy

Jelen dokumentum a KKSZB rendszer magas szintű szoftverarchitektúráját mutatja be.

Szoftver *architect-ek* és rendszerszervezők, rendszertervezőknek és fejlesztőknek készült azért, hogy a KKSZB rendszerhez történő csatlakozás magasszintű áttekintésével segítse a csatlakozást.

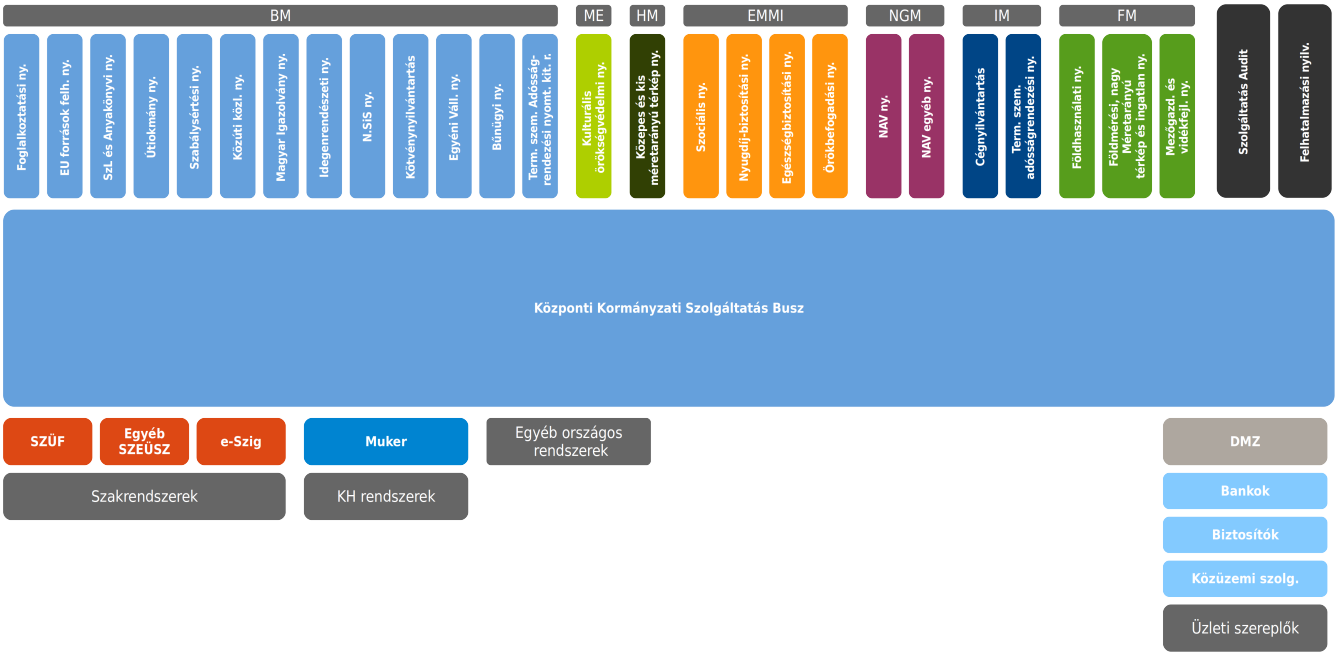
A különálló **Csatlakozás a KKSZB rendszerhez** dokumentum részletes útmutatót ad a csatlakozás technikai kérdéseiről.

Bevezető

A KKSZB rendszer szoftverarchitektúra segítségével bemutatható a KKSZB rendszert felépítő szoftverkomponensek közötti kapcsolat. Az architektúra segítséget nyújt a fogalmak meghatározásához, megértéséhez és a működés bemutatásához, ezzel segíti a projekttel kapcsolatba kerülő személyek kommunikációját. Az architektúra terven szereplő komponensek definiálása segíti a fejlesztés részekre bontását, így a fejlesztések párhuzamosíthatók. A követelmények és ezen architektúra terv alapján megalkothatók a további más típusú, vagy részletezettségű architektúra tervek is (pl.: fizikai elhelyezés architektúrája).

Üzleti követelmények

A Közigazgatás- és Közszolgáltatás-fejlesztés Operatív Program 2015. évre szóló éves fejlesztési keretének megállapításáról szóló 1004/2016. (I. 18.) Korm. határozatban kiemelt projektként került nevesítésre a "Közigazgatási szakrendszerek egységes eléréséhez és interoperabilitásához központi alkalmazás szintű szolgáltatások biztosítása" című projekt, mely projekt keretein belül valósul meg a Központi Kormányzati Szolgáltatás Busz (továbbiakban: KKSZB). Ennek célja, hogy a nemzeti adatvagyon részét képező nagy állami nyilvántartások és szakrendszerek a KKSZB keretrendszeren keresztül történő, szolgáltatás orientált szabványos összekapcsolását biztosítsa, az interoperabilitás megteremtésével a rendszerek közötti hatékony kommunikáció egységes szintre emelésével. A KKSZB segítségével lehetőség nyílik a jelenleg különböző technológiai, integráltsági, üzemeltetési szinten működő információs rendszerek egymással való összekapcsolására, a redundáns adattárolás visszaszorítására, és az ebből eredő adatintegritási hibák csökkentésére. Az alábbi ábrán a KKSZB rendszer elhelyezkedése látható a közigazgatási szakrendszerek, nyilvántartások valamint az üzleti szereplők rendszereihez képest. Ennek a megoldásnak előnye, hogy az az alkalmazás, amely egyszer már csatlakozott mint szolgáltatást igénybe vevő (kliens) a KKSZB rendszerre, úgy a továbbiakban - a megfelelő jogosultság birtokában - könnyedén tudja elérni az összes egyéb szolgáltatást. A szolgáltatást nyújtók szemszögéből hasonló előny származik abból, hogy ha már publikálták a szolgáltatásukat a KKSZB-n, úgy egyéb - szoftverfejlesztési, azonosítás, fizikai csatlakozási -, teendőjük nincs, csak a kliensek által benyújtott jogosultsági kérések elbírálása.



Ábra 1. KKSZB logikai architektúra

A KKSZB csatlakozáshoz a csatlakoztatandó alkalmazást fel kell készíteni a KKSZB igényeinek megfelelően. A KKSZB tervezésekor szempont volt, hogy a csatlakoztatandó alkalmazások módosítási igénye a lehető legkisebb legyen, ezért a KKSZB alaptechnológiaként a HTTP protokollt használja.

Alapkövetelmények

Az üzleti követelmények mellet a KKSZB rendszerrel szemben alapkövetelményeket fogalmaztak meg. Az alapkövetelmények azok, amelyeket kötelezően teljesítenie kell a rendszernek.

A KKSZB rendszerrel szembeni alap követelmények:

- **minimálisan érintse a jelenlegi rendszereket**: a KKSZB sikerének fontos kérdése, hogy milyen gyorsan és mennyiért lehet csatlakozni hozzá. A jelenlegi rendszerek csatlakozás miatti módosításának elkerülése vagy minimalizálása fontos cél.
- **könnyen csatlakoztatható legyen bármely rendszer:** a könnyű csatlakozás feltétele, hogy minél kevesebb előírást kelljen betartani és implementálni. Lehetőleg amit be kell tartani, teljes mértékben szabványosnak és széles körben elterjedtnek kell lenni. Emiatt a HTTP jelenleg az egyetlen megkötése a rendszernek, nincs üzenetformátum, vagy más egyedi szoftver vagy egyedi interfész (pl.: Message Queue) igénye.
- **a jövőbeni, modernebb rendszereket ne fogja vissza:** csak a HTTP a megkötés, bármely jövőben megjelenő új üzenetformátum, titkosítás, vagy módszer (ami HTTP kompatibilis, pl.: REST), képes legyen működni rajta.
- **nagy teljesítményű és magas rendelkezésre állású legyen:** a cél, hogy a *komplett közigazgatási szakrendszer és nyilvántartás* ezen keresztül működjön, így szükséges, hogy akár több tízezer vagy százezer kérés kiszolgálása másodpercenként. Jelenleg a tömeges adatlekérdezések generálnak nagy terheléseket, illetve olyan szakrendszerek, amelyeken gyorsító tárak jelennek meg, így akár több ezer kérést is ki tudnak szolgálni másodpercenként, tipikusan ilyenek az automatizált adatgyűjtő (pl.: rendszám felismerő) rendszerek. A KKSZB rendelkezésre állásának a lehető legmagasabbnak kell lennie az összes közigazgatási rendszer közül: az év minden napján 24 órában elérhetőnek kell lennie.
- **széles körben elterjedt nyílt és ingyenes szabványokon és technológiákon alapuljon:** a széleskörűen elterjedt megoldások növelik az értékállóságot, mivel nem függnek egy-egy gyártó döntéseitől (pl.: termék megszüntetése), ugyanakkor nagyobb tömegben érhetők el fejlesztők ezekhez. Másik fontos tényező a biztonság. Az elterjedt, elfogadott, napi szinten használt technológiák és szabványok magasabb biztonsági szintet jelenthetnek mint a zárt, belül ismeretlen megoldások. Az ingyenes megoldásokkal jelentősen csökkenthető a TCO, valamint javítható a szoftver minősége, hiszen a feladathoz legjobban megfelelő szoftverkomponens használható a megoldásra, nem csak a beszerzett termék.
- **időtálló megoldás legyen:** az elvárt életciklus legalább 15 év, így ennek megfelelő technológiákat és szabványokat kell választani. A jövőbeli technológiai bővíthetőség és az inkrementális megújíthatóság fontos szempont, mert a KKSZB az év minden napján, 24 órában rendelkezésre kell hogy álljon. Ennek megfelelően a monolitikus megoldás nem alkalmazható, csak a részenkénti megújíthatóságot támogató microservice architektúra az amely biztosítja ezt.
- **felhő kompatibilis legyen:** a gazdaságos üzemeltetés mellett meg kell tartani a skálázhatóságot, és a magas rendelkezésre állást is, ezért célszerű a rendszert a felhő technológiákra építeni.
- **megbízható és biztonságos legyen:** nagy terhelés esetén is megbízhatóan üzemeljen. A KKSZB tervezését elosztott rendszerként végezzük, így adott meghibásodási fajtákkal szemben érzéketlen lesz a rendszer. Kiemelt figyelmet igényel a biztonság, így a tervezés és fejlesztés

minden lépésében ezt figyelembe kell venni.

- **rugalmasan bővíthető legyen:** a buszra kerülő szolgáltatások plusz funkciókat tudnak a rendszerhez adni, maga a KKSZB ennek a lehetőségét biztosítja. Ezzel gyakorlatilag a rendszer képessége korlátlanul bővíthető anélkül, hogy az a többi alapkövetelmény rovására menne.
- **támogassa az üzemeltetést, a vezetői információk kinyerését:** a KKSZB képes lesz minden rákapcsolt kliens és szolgáltatás esetén a kommunikációról kimutatásokat készíteni. Ezzel a klienstől és szakrendszertől függetlenül egy harmadik fél által (KKSZB) biztosítható akár a szolgáltatási szint (SLA) értékelése, vagy a szolgáltatások egymáshoz történő relatív értékelése is.
- **az üzeneteket nem olvashatja, rögzítheti vagy módosíthatja:** a KKSZB nem avatkozhat a kliensek és szolgáltatások üzeneteibe, azokat nem olvashatja, nem értelmezheti, nem rögzítheti. Ezzel a fontos kitétellel garantálható az, hogy mind a kliens mind a szolgáltató biztos lehessen abban, hogy az általa kért, vagy küldött üzenet úgy ér célba, ahogy azt ő feladta.

Főbb funkcionális követelmények

- A KKSZB a kliens és szolgáltatás között teremt egy *elérési jogosultság* ellenőrzés után kapcsolatot.
- A KKSZB adatokat gyűjt a kapcsolatról, amelyeket a BigData rendszerének továbbit elemzés, megfigyelés céljából.
- A KKSZB nem tárol és nem elemez semmilyen üzenet tartalmat, csak a KKSZB saját HTTP fejléceit kezeli.
- A KKSZB rendszerre való csatlakozás feltételei:
  - a csatlakozó a KKSZB műszaki követelményeit betartsa, ehhez a csatlakozási dokumentum ad egyértelmű, fejlesztői szintű leírást
  - a KKSZB rendszerre vonatkozó *nem műszaki követelményeket* betartsa, ilyen például: a tanúsítványok telepítése, kezelése, vagy szerződéses feltételek, stb.
- A KKSZB rendszer biztosítja, hogy bármilyen szolgáltatás kapcsolódni tudjon rá, amely a műszaki követelményeket teljesíti.
- A KKSZB biztosít olyan szolgáltatásokat is, amelyek a projekt keretében valósulnak meg:
  - Aszinkron üzenet továbbító rendszer HTTP push technológiával
- A KKSZB támogatja a kliensek és szolgáltatások csatlakozási folyamatát, pl.: igény bejelentés, kezelés, teszt felületek, teszt adatok, minta program kódok.
- A KKSZB támogatja az üzemeltetést: folyamatos adat publikálás (BigData) és konténerizált szállítási környezet (Docker).
- A KKSZB publikálja a rajta elérhető szolgáltatások leírását a Szolgáltatás katalógusán keresztül, így azok online, naprakészen elérhetőek a kliensek számára.
- A KKSZB biztosítja az adminisztrátori, üzemeltetési és biztonsági felügyeletet nyújtó felületeket.

Elemzett megoldások és technológiák

A KKSZB koncepció kidolgozási fázisában megvizsgáltunk számos megoldást, rendszert és technológiát azért hogy az ottani tapasztalatokat felhasználhassuk. A vizsgálatok során a KKSZB rendszerrel szemben támasztott üzleti és alapkövetelmények, valamint a mai, modern technológiai megoldások alapján értékeltük ezeket. A KKSZB architektúra kialakításának megértéséhez segítséget nyújthat az, ha már meglévő megvalósítások előnyeit és hátrányait kiemeljük és röviden bemutatjuk.

X-ROAD

Az észtországi X-ROAD rendszer *hasonló* céllal készült, mint a KKSZB. Fontos kiemelni, hogy már maga az X-ROAD elkészítéséhez definiált célok sem egyeznek meg a KKSZB rendszer céljaival, a követelmény rendszere pedig még jobban eltér.

Az X-ROAD rendszer tervezése és fejlesztés 2000-ben kezdődött és 2003-ban helyezték üzembe. A KKSZB tervezésénél nagy segítséget nyújtott az, hogy egy korai (jelenleg 13-16 éves) rendszert tudtunk mai szemmel elemezni és értékelni a saját követelményeink szerint.

X-ROAD jellemzők:

- Az X-ROAD rendszer az XML (SOAP) üzenet szintaxist választotta. Ennek következménye, hogy a jelenlegi észt igényeket már nem tudja kiszolgálni, csak kerülő megoldásokkal, nevezetesen a JSON üzeneteket XML-be csomagolva tudja csak kezelni.
- Az X-ROAD beleavatkozik az üzenetek szintaxisába, amely a KKSZB esetén nem megengedett.
- Az X-ROAD rendszer az azonosítást üzenet szintaxis függően és az üzenet részeként kezeli, ezért a modernebb, egyszerűbb és kevésbé erőforrás igényes azonosítási módszereket nem lehet átvezetni rajta (csak minden rendszer átírásával lehetne).
- Mivel az X-ROAD rendszer az azonosítást üzenetben tartalmazza, ezért szükség szerűen az üzenetet olvassa és bizonyos esetekben módosítja, amely a KKSZB esetén nem megengedett.
- Az X-ROAD által választott üzenet szintaxis kötelezően XML, amely feldolgozása erőforrás igényes.
- Az X-ROAD a kor technológiai szintjén készült el.
- Semmilyen módon nem támogatja a felhő technológiákat (akkor ez nem létező fogalom volt).
- Az X-ROAD rendszer architektúrája szerint a szolgáltatások és a kliensek elérési jogosultságai központilag vannak kezelve.
- Az X-ROAD rendszer architektúrája szerint nem monolitikus felépítésű rendszer, a microservicek és az elosztott rendszerek jellemzőit mutatja.
- HTTP protokollt használ a kommunikációban.
- Nyílt és ingyenes termékekre épít, amelyek az elmúlt 15 évben folyamatosan fejlődtek, így az X-ROAD-nak jelenleg már a 6. verzióját tudták kiadni.
- Működő rendszer, ismert kérés számokkal és korábbi biztonsági eseményekkel.

Következtetések:

- Bármely üzenet szintaxis melletti döntés a meglévő rendszerek jelentős módosítását igényelné, a jövőbeli rendszereket pedig az idő előrehaladtával egyre elavultabb megoldásra korlátozná le. Ezért a KKSZB nem definiál kötelezően használandó üzenet szintaxist, mindössze a HTTP protokoll használatát igényli.
- A KKSZB az azonosítást nem az üzenetbe, hanem a HTTP protokollra bízza.
- Az X-ROAD által használt, mára elavult megoldásokat a KKSZB nem fogja használni.
- A KKSZB a microservice architektúrára fog épülni.
- Megerősítést nyert a nyílt és ingyenes technológiák használatának előnye az X-ROAD-ban.
- A KKSZB szoftver- és üzembiztonsága a legmodernebb és széleskörűen elterjedt megoldásokra épül.

A KKSZB esetén hasznosítottuk az X-ROAD előnyeit és kiküszöböltük annak hátrányait. Ugyanakkor a módosításokat figyelembe véve értékeltük annak visszahatását a rendszerre, és értékeltük esetleges jövőbeli következményeit is.

Monolitikus ESB rendszerek

Az 1990-es évek végén, 2000-es évek elején megjelent az ESB fogalma, amely 2005 körül a nagyvállalatok körében népszerű volt. Ekkor a néhány 100 esetleg 1000 felhasználó által használt rendszerek esetén, az adott szervezetnél platformot nyújtott az integrációhoz. Ekkor alakultak ki a monolitikus rendszerek, amelyek kezdeti előnyük mellett később jelentős hátrány is okoztak a használójuknak.

Nagy felhasználó számú rendszerek esetén nem tudták már vertikálisan skálázni ezen rendszereket, így a horizontális skálázás irányába mozdultak el. Az X-ROAD rendszer fejlesztéséhez az akkori teljesítményű számítógépek és a felhasználó szám miatt (összes állampolgár, cégek, stb.) vélhetően fel sem vetődött ilyen rendszer üzembe helyezése.

A monolitikus rendszerek hátrányai a következők:

- Gyakran szállító függő megoldás, amely a szállító, támogató üzleti döntésének kockázatát a vásárlóra hárítja, pl.: JBoss ESB támogatás 2014-ben megszűnt, így az utolsó elérhető verziója 2013 márciusi.
- Platform függő: a monolitikus rendszer egy szoftver fejlesztési platformot támogat, ezzel korlátozva a megoldásra felhasználható eszközök körét, valamint a fejlesztők számát.
- Monolitikus: az összes ESB-re helyezett alkalmazást meg kell újítani, ha az alrendszer cseréje megtörténik (nagyobb változásokkal járó verzióváltás).
- Alacsony rendelkezésre állást biztosít: mivel a folyamatos megújítás a monolitikus felépítés miatt nem lehetséges, ezért egyszerre kell a megújítást elvégezni, így ez gyakran jelent nagyobb tervezett leállást.
- Horizontálisan nem skálázható, vagy ha igen, akkor újabb licencek beszerzését, így többlet költséget jelent.

Következtetések:

- A KKSZB rendszer nem használhatja a monolitikus felépítést, mivel microservice architektúrára alapszik.
- A felhő megoldás, a rendelkezésre állás, a megbízhatóság és a megújíthatóság fontos szempont a KKSZB esetén, amely nem egyeztethető össze egy monolitikus rendszerrel.
- A nagy teljesítmény igény szükségessé teszik a könnyű és gyors horizontális skálázhatóságot.

A KKSZB esetén a monolitikus ESB használata megalapozatlan, annak működése, célja és szemlélete nem felel meg a követelményeknek és alap elvárásoknak.

Modern informatikai rendszerek, megoldások

A KKSZB igényei szerint elemeztük a ma rendelkezésre álló legmodernebb és legelterjedtebb megoldásokat és módszereket. Figyelembe vettünk a saját területükön piacvezető cégek informatikai megoldásait és módszereiket.

- **REST API:** jó pár éve terjed ez az interfész programozási szemlélet, amelynek elméletét 2000-ben alkották meg. Segítségével egyszerűbbé tehető az interfészek elérése és használata, valamint csökkenthető a tervezésre és fejlesztésre fordított idő mindezt úgy, hogy a HTTP protokollra épül.
- **JSON üzenet szintaxis:** az interneten elérhető interfészek közel 90%-a ma már ezt az üzenet szintaxist használja. A JSON közvetlenül felhasználható a böngészőkben vagy mobil telefon alkalmazásokban. Mivel feldolgozása egyszerű és kevésbé erőforrás igényes mint például az XML-é, ezért mára a rendszer-rendszer kapcsolatok egyik építő eleme lett.
- **Microservice architektúra:** a robbanásszerű felhasználói szám növekedéssel együtt terjedt el. Segítségével küszöbölték ki azokat a hiányosságokat, amelyre a monolitikus rendszerek nem tudtak megoldást adni.
- **HTTP és HTTPS (TLS):** a mai modern rendszerek a HTTP-re épülnek, a kapcsolat biztonságát az interneten szinte kizárólag a HTTPS és VPN rendszerek biztosítják.
- **Non-Blocking IO:** a nagy teljesítmény eléréséhez a korábbi megoldások nem voltak megfelelőek vagy gazdaságosak, ezért a magas felhasználószám miatt elterjedtek a "Non-Blocking IO" megoldások. A korábbi többszálú rendszerek helyett az esemény alapú rendszerek tervezhetővé tették és radikálisan csökkentették a hardver igényt. Eközben nőtt a rendszerek megbízhatósága, mivel a rendszer akár két nagyságrenddel is több párhuzamos kapcsolatot tud fenntartani a korábbi megoldásokhoz képest.
- **Cloud (felhő) technológiák:** a gazdaságos működés, gyors skálázás, valamint az üzemeltetési és rendszer környezet standardizálását és szoftveres meghatározását megvalósító rendszereket hoztak létre, amelyek bár gyakran eltérő eszközökkel, de azonos elvek mellett jöttek létre. Ezek összefoglaló néven *felhő* megoldásként kerültek a köztudatba.

A fent említett technológiákat számos piacvezető cég használja:

- PayPal: piacvezető, internetes pénzügyi rendszereket fejleszt és üzemeltet
- Walmart: piacvezető, kiskereskedelmi rendszert fejleszt és üzemeltet

- GoDaddy: piacvezető, internetes domain név és hosting szolgáltató, fejlesztő

Következtetések:

A KKSZB rendszernek a fenti technológiák megfelelőek, biztosítják az üzleti és alapkövetelmények elérését. Lehetővé teszik egy modern, biztonságos, gyors, rugalmasan megújítható és időtálló megoldás létrejöttét.

Technológiák és módszerek

A használt technológiák az üzleti- és alapkövetelményekkel nem állhatnak ellentétben, összhangban kell állniuk azokkal a következtetésekkel, amelyeket más rendszerek és megoldások elemzésénél megtettünk.

A kiválasztásnál a fenti alapkövetelményeknek való megfelelőséget kell megvizsgálni, több alternatíva esetén pedig azt kell választani, amelyik jobban megfelel ezeknek.

A kiválasztott eszközök mindegyike elterjedt, nyílt és ingyenes, így a szoftver licencelésből eredő gazdasági, jogi kockázatok és költségek kiküszöbölhetők.

Használt fontosabb technológiák, eszközök, módszerek:

- **NodeJS:** nagy teljesítményű, magas rendelkezésre állású, széles körben elterjedt, ingyenes, nyílt forráskódú, "Non-Blocking IO" alkalmazás fejlesztést lehetővé tevő rendszer. Fejlesztése a Linux Foundation felügyelete mellett zajlik.
- **Ansible:** támogatja a szoftver definiált (automatikus) rendszer telepítést és konfigurálását.
- **Docker:** alkalmazás futtató konténer, amely biztosítja a reprodukálható futtatást, a gyors frissítéseket és csökkenti az üzemeltetés terhelését.
- **Linux:** alkalmazás fejlesztési, tesztelési és futtatási platform és operációs rendszer. Kernel fejlesztése a Linux Foundation felügyelete mellett zajlik.
- **HA-Proxy:** magas rendelkezésre állású, nagy teljesítményű, elterjedt proxy alkalmazás
- **F5:** fizikai eszköz, magas rendelkezésre állású, nagy teljesítményű fizikai terheléselosztó
- **Git:** a legelterjedtebb és legmodernebb verziókövető rendszer
- **NoSQL (CouchDB/MongoDB):** magas rendelkezésre állású, megbízható, kiemelkedő replikációs képességekkel rendelkező adatbázis.
- **AsciiDoctor:** dokumentációs szintaxis és infrastruktúra, amely a verziózható, és média független dokumentációkészítést támogatja.
- **Fluentd:** nagy teljesítményű, elterjedt, megbízható, egységesített üzenet log közvetítő rendszer.
- **Elasticsearch:** BigData feldolgozást lehetővé tevő, nagy teljesítményű alkalmazás.
- **Kibana:** az Elasticsearch vizualizációs felülete.
- **VueJS:** asztali és mobil eszközöket támogató webes kliens alkalmazás fejlesztő keretrendszer.
- **Bootstrap:** asztali és mobil eszközöket támogató webes megjelenítést támogató CSS keretrendszer
- **REST API:** tervezési módszer, amellyel HTTP-n keresztül lehet pontosan definiált interfészeket létrehozni.
- **Jenkins:** automatikus tesztek futtatását végző alkalmazás, kulcs eleme a CI/CD folyamatoknak.

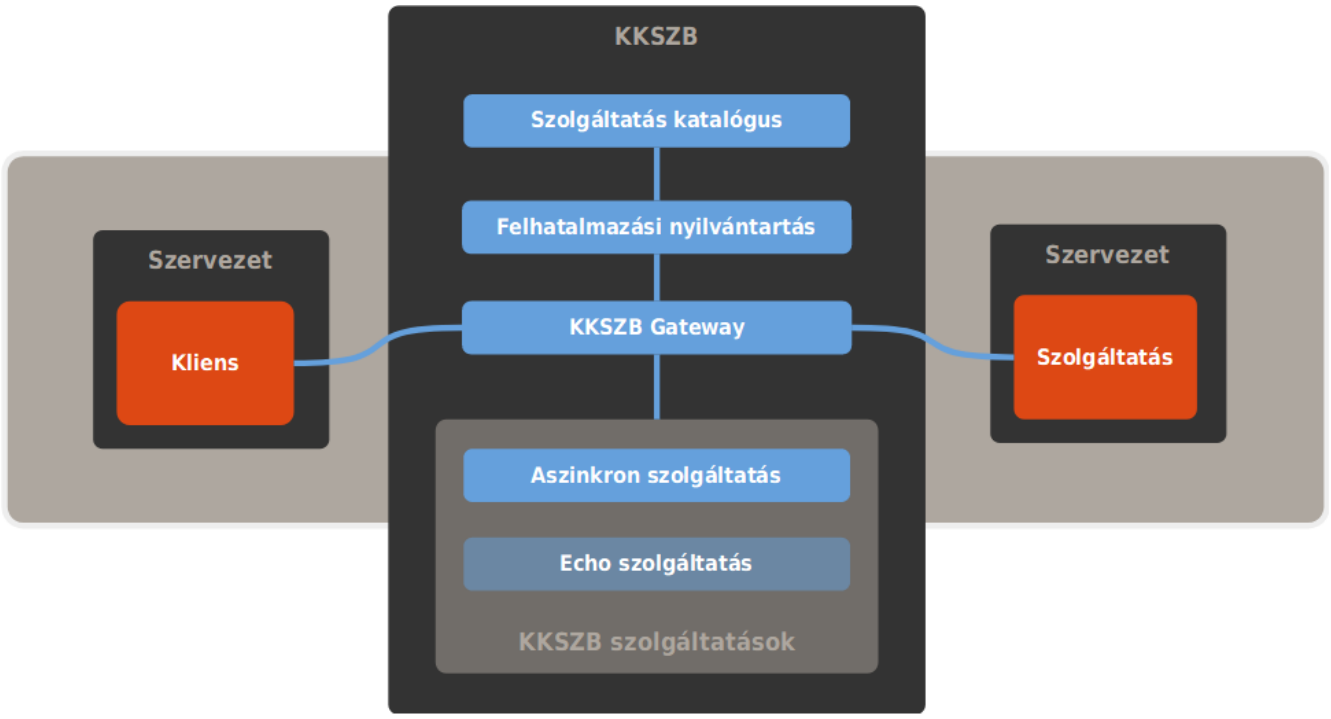
A fent felsorolt technológiák, rendszerek, alkalmazások mellett számos olyan módszert használunk, amely a modern alkalmazás fejlesztés alapja. Ezek célja, hogy emeljék a minőséget, növeljék a szoftverek teljesítményét, csökkentsék a fejlesztési és üzemeltetési ráfordítást, és biztosítsák az

átláthatóságot.

- **Continuous Integration (CI):** a KKSZB szoftver komponensek folyamatosan és automatikusan tesztelve vannak. Az emberi beavatkozást kiküszöbölve, reprodukálhatóan és minden egyes változtatáskor lefutnak a tesztek, ezzel könnyen kiküszöbölhetők a fejlesztés során jelentkező hibák.
- **Test-Driven Development (TDD):** a tesztek és a programok együtt készülnek. A tesztek automatikusan futnak le a CI szerveren és a fejlesztő gépén, így a tesztek azonnal felfedik a lehetséges hibákat. A tesztek egyben a *valós* működést is definiálják, így *önleíró módon dokumentálják* magát a tesztelt szoftvert is.
- **Folyamatos minőség mérés:** a minőséget nem csak a szoftver funkcionális tesztekkel mérjük, hanem a statikus kód elemző eszközökkel. Ezek segítik a kód karbantarthatóságának és áttekinthetőségének növelését a teszt lefedettség méréssel és egyben a tesztelés minőségéről is kapunk információt. A fejlesztés közben használt kód elemző eszközök segítségével elkerülhetjük a leggyakoribb programhibákat.
- **Folyamatos sérülékenység ellenőrzés:** a szoftver folyamatosan és automatikusan sérülékenység vizsgálat alatt van, a használt kódok és függőségek sérülékenységéről még fejlesztési időben információt kapunk.

KKSZB magas szintű szoftverarchitektúra

A KKSZB szoftverarchitektúra kidolgozásánál a korábban felsorolt, ismertetett követelményeket és lehetőségeket vettük figyelembe. Az alábbi ábra alapján bemutatjuk a *fő komponenseket és azok kapcsolatait.*



Ábra 2. KKSZB magas szintű szoftverarchitektúra

Az ábrán található komponensek definícióját a Definíciók fejezet tartalmazza.

A leggyakoribb kérés fajta a szinkron kérés, melyben a kliens alkalmazás a feltett kérésre megvárja a Szolgáltatás válaszát.

A kapcsolódó fél (kliens) a kérését a KKSZB Szolgáltatás cégpontra küldi, amely a KKSZB Gateway alkalmazásra érkezik be. A KKSZB Gateway alkalmazás elvégzi a megfelelő ellenőrzéseket: azonosíttatja a kliens alkalmazást a Rendszer Felhatalmazási Nyilvántartás felé küldött kérés alapján. Amennyiben ez sikeres, úgy egy eléréshez használható, rövid lejáratú tokent kap vissza, amely tartalmazza a kliens alkalmazás elérési jogosultságait. A KKSZB Gateway megvizsgálja az elérési jogosultságok érvényességét a meghívott Szolgáltatás tekintetében, ha van megfelelő elérési jogosultság, akkor a Szolgáltatás felé a kérés továbbításra kerül.

A Szolgáltatás részére a HTTP fejlécben információk kerülnek átadásra a szolgáltatás igénybe vevőjéről, amelyet a Szolgáltatás igényei szerint felhasználhat. A kapott kérést a Szolgáltatás saját logikája szerint kiszolgálja.

Mivel a KKSZB elvégzi az azonosítás és az **elérési** jogosultságok kezelését, így minden kliens alkalmazás és Szolgáltatás mentesül az egyedi, *alkalmazás szintű* azonosítási és jogosultság kezelési rendszerek fejlesztése alól. Ráadásul a csatlakozott felek bármely más Szolgáltatást is potenciálisan elérhetnek (megfelelő elérési jogosultság birtokában) anélkül, hogy bármilyen azonosítási réteget újra implementálniuk kellene.

A biztonság fokozása miatt a KKSZB tanúsítványkezelést követel meg, amely részletes bemutatását a Tanúsítványok függelék tartalmazza.

Csatlakozás

A csatlakozási folyamat során a Kapcsolódó fél hozzáférést kap a KKSZB rendszerhez, ekkor még nem dől el, hogy miként fogja azt igénybe venni. A kapcsolódó fél a csatlakozás eredményeképpen hozzáférést kap a Rendszerfelhatalmazási Nyilvántartás (RFNY) és a Szolgáltatás Katalógus (SZK) webes felületéhez.

Az RFNY felület alapvető művelete, amelyhez megfelelő jogosultság szükséges:

- Szolgáltatás Létrehozásának Kérelmezése a KKSZB rendszerben
- Szolgáltatás Elérési Jogosultságának Kérelmezése egy adott, KKSZB rendszeren elérhető Szolgáltatásra

Amennyiben Szolgáltatást akar kiajánlani a KKSZB rendszeren keresztül, úgy a Szolgáltatás Katalógusba kell feltölteni az adott szolgáltatáshoz szükséges interfész tervet.

Amennyiben Szolgáltatást akar igénybe venni akkor előtte a Szolgáltatás Katalógusból böngészheti a szolgáltatásokat és itt érheti el az interfésztervet.

A KKSZB rendszerhez kétféleképpen lehet csatlakozni:

- **kliens alkalmazással:** olyan alkalmazás, amely a KKSZB rendszeren elérhető szolgáltatásokat - megfelelő elérési jogosultság esetén - igénybe veheti.
- **szolgáltatással:** olyan alkalmazás, amely a KKSZB rendszeren HTTP felületen publikálja az elérését. A végpont tetszőleges típusú lehet; végezhet karbantartást (adatgyűjtő), vagy kiszolgálhat lekérdezéseket (adat szolgáltató).

A szolgáltatás - mint *adat szolgáltató* -, képes a *közadatok* szolgáltatására is, amennyiben ez számára kötelezettség, ez technikailag nem tér el más jellegű adat szolgáltatásoktól.

A szolgáltatások csatlakoztatása a KKSZB rendszerre folyamatos, bővebb információt a szolgáltatás birtokosa tud adni a várható csatlakozási időpontról.

A szolgáltatások KKSZB rendszerre történő csatlakozásánál **figyelembe kell venni a jogszabályi hátteret és irányelveket** (pl.: *once only principle:* egyszeri adatbekérés elve, IBTV).

Ajánlott, hogy a Szolgáltatás belső hálózati kialakítása vegye figyelembe az IBTV által támasztott követelményeket.

A Szolgáltatónak lehetősége van eldöntenie, hogy az adott, alacsonyabb IBTV (lásd [it-biztonsag]) osztályba sorolt rendszertől érkező kérést kiszolgálja vagy nem szolgálja ki, ezt **az RFNY webes felületén a szolgáltatás elérési jogosultság engedélyezési folyamata során teheti meg.**

Amennyiben adott Szolgáltatáshoz egy hálózati ponton keresztül kezeli az eltérő biztonsági szinteket - egy Szolgáltatás URL van az összes több IBTV szinthez -, úgy a **beérkező kérés feldolgozásakor** az **x-kk-security-class** HTTP fejléc alapján tudja az útvonalválasztást biztosítani.

KKSZB környezetek

A KKSZB az integrációs tesztekhez (INT) és az éles működéshez (PROD) biztosít környezeteket.

A kifejlesztett szolgáltatásokat az integrációs környezetben (INT) lehet alávetni az integrált tesztnek, ezért ebben a környezetben már kitesztelt alkalmazással kell csatlakozni, amely a KKSZB és a tervezett interfész igényeit teljesíti.

A KKSZB integrációs (INT) környezetbe az éles működésre szánt alkalmazás stabil verzióját kell csatlakoztatni. Az integrációs környezet nem fejlesztői környezet, azt a fejlesztőnek a megfelelő módon (pl.: mock) magának kell biztosítani.

Az éles (PROD) környezetbe a valós, éles rendszerek kerülhetnek csak felcsatlakoztatásra!

A **Csatlakozás a KKSZB rendszerhez** dokumentum részletes útmutatót, önálló fejezetet tartalmaz a tesztelésről.

Rendszer Felhatalmazási Nyilvántartás

A Rendszer Felhatalmazási Nyilvántartás a KKSZB rendszer elérési jogosultság és szolgáltatás kezelő rendszere. Segítségével a felhasználók megfelelő jogosultság segítségével webes felületen keresztül elláthatják feladataikat:

- **csatlakozott féllel kapcsolatos menedzsment:** autentikációs tokenek menedzsmentje, szolgáltatáshoz való csatlakozási igény bejelentése, stb.
- **csatlakoztatott szolgáltatással kapcsolatos menedzsment:** saját szolgáltatások menedzsmentje, új szolgáltatási végpont iránti kérelem, szolgáltatás elérési jogosultságok menedzsmentje, stb.
- **KKSZB rendszer menedzsment:** kapcsolódó felek kérelmeinek menedzsmentje, általános folyamat felügyelet, stb.

A Rendszer Felhatalmazási Nyilvántartás szolgáltatását használja fel a KKSZB Gateway a kliens alkalmazások azonosítására, valamint az elérhető Szolgáltatásokhoz tartozó útvonal (routing) tábla letöltésére.

Szolgáltatás Katalógus

A Szolgáltatás Katalógus tartalmazza minden egyes Szolgáltatás interfész leírását, valamint a KKSZB rendszerhez történő csatlakozás információit is. A Szolgáltatás Katalógus leírások lehetnek szabadon elérhetők (publikus), vagy azonosításhoz kötöttek (bizalmas).

A Szolgáltatás Katalógus tartalmának naprakészen tartásáért a szolgáltatás nyújtója a felelős. Azonosítás után a tartalmat frissítheti, vagy új verziót készíthet. A Szolgáltatás Katalógus tartalmának összhangban kell lennie a KKSZB rendszerben elérhető szolgáltatással, különösen a verziók módosítása esetén kell erre ügyelni.

Aszinkron szolgáltatás

A kliens alkalmazás ebben az esetben mint *Üzenet küldő* a Szolgáltatás mint *Üzenet fogadó* jelenik meg. A kliens alkalmazás elküldi az üzenetet a Szolgáltatás felé, de nem várja meg a válaszát, hanem folytatja saját tevékenységét.

Ebben az esetben a kliens alkalmazás a KKSZB Aszinkron Szolgáltatás végpontját hívja meg egy sima HTTP kéréssel (szinkron hívás). Az üzenetet tetszőleges szintaxissal adhatja fel, a KKSZB specifikus címzettlista és egyéb információk a HTTP fejlécben kerülnek átadásra. Az üzenetet az Aszinkron Szolgáltatás mindig befogadja, amennyiben a KKSZB követelményeknek az megfelel.

Az Üzenet fogadó (Szolgáltatás) az Aszinkron szolgáltatástól kapja meg az üzenetet HTTP push megoldással, így a Szolgáltatás számára ez sima HTTP kérésként jeléinek meg. Amennyiben a Szolgáltatás nem elérhető, úgy az Aszinkron Szolgáltatás újra próbálkozik.

A Szolgáltatás implementációban gondoskodni kell az idempotens működésről, hogy az esetlegesen többször beérkező ugyanazon kérésekre is megfelelő feldolgozás történjen.

A fenti működési minta *értesítés jellegű*, amikor a kliens alkalmazást nem érdekli, hogy mi lett az üzenet feldolgozás eredménye.

A fenti aszinkron küldési logikákat a kliens alkalmazás és Szolgáltatás tetszőleges logikával állíthatja egymás mögé, például a kliens alkalmazás által küldött aszinkron üzenetre akár a Szolgáltatás is válaszolhat aszinkron üzenettel, ekkor a Szolgáltatás lesz az *Üzenet küldő* és a kliens alkalmazás lesz az *Üzenet fogadó* fél. Az Aszinkron Szolgáltatás segítségével így tetszőleges kommunikációs logika valósítható meg a kliens alkalmazás és a Szolgáltató között.

Az Aszinkron szolgáltatás használatával gyártó és egyedi megoldás független aszinkron üzenetküldés valósítható meg, csupán a HTTP protokoll használatával. Ennek eredménye az egyszerűbb programszerkezet, gyorsabb implementáció és a költség megtakarítás a gyártó specifikus megoldások licenc díjának és üzemeltetésének tekintetében.

Echo Szolgáltatás

Az Echo szolgáltatás a beüzemelés és az üzemi állapot ellenőrzésére szolgál a kliens alkalmazások irányából, a beküldött üzenetet visszaküldi. Minden csatlakozott kliens alkalmazásnak van hozzá elérési jogosultsága.

Ha a kliens alkalmazásnak problémája merülne fel egy Szolgáltatással vagy akár a saját működésével kapcsolatban, akkor az Echo szolgáltatáson keresztül gyorsan és megbízhatóan ellenőrizheti saját és a KKSZB állapotát.

Útvonalválasztás (routing)

A KKSZB rendszer egyik feladata, hogy az útvonalválasztást megoldja a Kliens és a Szolgáltatás közt úgy, hogy:

- a Szolgáltatás végponton történő fizikai áthelyezés (pl.: IP cím változás) ne befolyásolja a Klienseket
- transzparens legyen a Kliens számára az Útvonalválasztás: standard URL-ekkel működjön HTTP protokollon
- a Szolgáltatást leíró végpont elnevezése értelmezhető legyen akár számítógép nélkül is
- támogassa a Szolgáltatás verziózását
- támogassa a Szolgáltatások csoportosíthatóságát (Szolgáltatás névtér)

A fenti megoldással az eddigi gyakorlattal szemben egy áttekinthető, értelmezhető és egységes Szolgáltatás elérési végpont leíró rendszer jön létre, amely segíti a Klienseknek és a Szolgáltatásoknak a hibamentesebb működést és gyorsabb üzembeállítást.

KKSZB Szolgáltatás végpont

Ezen a végponton érhető el egy adott Szolgáltatás a Kliens számára. A végpont egy URL, amelynek egy adott részét a KKSZB rendszerben, a Rendszer Felhatalmazási Nyilvántartás webes felületén kell beállítani. Az URL további részét a Szolgáltatás interfész definíció szerint lehet használni.

Az alábbi egy példa egy Szolgáltatás végpontot mutat, ahol egy **jarmu** *Szolgáltatás névtérbe* sorolt Szolgáltatás érhető el.

```
https://gw.kkszb.gov.hu/jarmu/rsz/v1/
```

A Kliens a fenti URL-en például rendszám alapján a járművet le tudja kérdezni az éppen aktuális időpontra:

```
https://gw.kkszb.gov.hu/jarmu/rsz/v1/rsz=AAA111
```

Táblázat 1. Szolgáltatás URL részei, https://gw.kkszb.gov.hu/jarmu/rsz/v1

https:	Állandó része a szolgáltatás eléréseknek, a Szolgáltatás a KKSZB rendszeren keresztül lesz kiszolgálva.
/jarmu/rsz/v1	Szolgáltatás azonosítója. A Szolgáltatás a **jarmu** névtérben található Szolgáltatások közül lesz kiválasztva.

Szolgáltatás névtér

A *Szolgáltatás névtér* fontos több ütköző Szolgáltatás név feloldása esetén. Például személyt lehet lekérdezni egy azonosítóval, akkor két eltérő rendszer is rendelkezhet a **szaz** Szolgáltatással

(Személyi Azonosító Jel alapú lekérdezés).

Szolgáltatás névtér nélkül mindkettőnek egyező URL-je lenne: https:/szaz/v1

A Szolgáltatás névtér bevezetésével értelmezhetővé és elkülöníthetővé válik a két Szolgáltatás:

- https://gw.kkszb.gov.hu/jarmu/szaz/v1/szaz=17201010001 : Jármű Nyilvántartó rendszerből szaz alapú lekérdezés, eredménye jármű lista.
- https://gw.kkszb.gov.hu/szl/szaz/v1/szaz=17201010001 : Személy és Lakcím Nyilvántartó rendszerből szaz alapú lekérdezés, eredménye személy.

A Szolgáltatás névtér azonosító akár több alkalmazást vagy akár több nyilvántartást is összefoghat.

A Szolgáltatás névtér képzését a KKSZB rendszer felelőse végzi.

A **kkszb** Szolgáltatás névtér a KKSZB rendszer részére van fenntartva, amely alatt a KKSZB saját szolgáltatásait publikálja.

Szolgáltatás azonosítója

Az adott szolgáltatás azonosítója akár több részből is állhat, például érvényesek az alábbiak:

- **/jarmu/leksz/rsz/v1** ahol a **leksz/rsz** a szolgáltatás azonosítóban a leksz például jelentheti az alkalmazás nevét, amely a kérést kiszolgálja.
- **/jarmu/rendorseg/rsz/v1** ahol a **rendorseg/rsz** a szolgáltatás azonosítóban a *rendorseg* a szakrendszeri jogosultságra is utalhat (nem keverendő össze a KKSZB rendszerben értelmezett elérési jogosultsággal).

A Szolgáltatás azonosító nem tartalmazhat verzió értéket. Invalid szolgáltatás URL a *v122* miatt: https://gw.kkszb.gov.hu/jarmu/v122/rsz/v1

Szolgáltatás verzió

A Szolgáltatás verzió száma a Szolgáltatás végpontban kötelező.

A Szolgáltatás felelős döntése a *Szolgáltatás verziószámának* beállítása.

A *Szolgáltatás verziószáma* olyan érték, amelynek változása magával vonja, hogy az azt használó Klienseknek is be kell állítani. Ezért kiemelt figyelmet érdemel az interfészek verziózása.

Javasolt, hogy új verziójú interfészek esetén, az új verzió bevezetésének hatásait a Szolgáltatás felelős mérlegelje!

Az interfészek verziózásánál két fő módszer lehetséges:

- **párhuzamosan működő főverziók:** ekkor például a https://gw.kkszb.gov.hu/jarmu/rsz/v1/ és a https://gw.kkszb.gov.hu/jarmu/rsz/v2/ végpont is ad választ, fokozatos átállást tesz lehetővé.

- **azonnali átállás:** mindig csak egy érvényes és elérhető verziója van a Szolgáltatásnak, a Kliensek részéről azonnali átállást kell végezni, a régi verzió nem használható.

Az interfész verzióváltásánál a fenti két stratégia közül kell választani, mérlegelve az előnyöket és hátrányokat.

A Szolgáltatás verzió számát a szemantikus verziózás szabályai szerint kell kezelni az interfészekre vonatkozó kitétel figyelembe vételével, bővebben: http://semver.org

A Szolgáltatás végpontok verziója nem egyezik meg az azt szolgáltató szoftver verziójával!

A Szolgáltatás végpontok verziószáma csak *MAJOR* verziószám lehet 1 vagy annál nagyobb, a "Semantic Versioning 2.0.0" leírása szerint.

Az interfészen végzett hibajavítások, vagy új funkciók hozzáadása nem emeli a verziószámot, csak akkor, ha azok a Kliens oldal számára változást jelentenek. **Mindig mérlegelni kell a változás hatását a Kliensekre.**

A fentiek alapján érvényes verziószámok:

- https://gw.kkszb.gov.hu/jarmu/rsz/v1
- https://gw.kkszb.gov.hu/jarmu/rsz/v896

Érvénytelen verziószámok, például:

- https://gw.kkszb.gov.hu/jarmu/rsz/v0
- https://gw.kkszb.gov.hu/jarmu/rsz/v1.2
- https://gw.kkszb.gov.hu/jarmu/rsz/v1.2.3

A Szolgáltatás végpont RFNY-beli definíciója mindig a verziószámra végződik és nem tartalmazza a "/" jelet.

Érvénytelen Szolgáltatás végpont definíció: https://gw.kkszb.gov.hu/jarmu/rsz/v1/

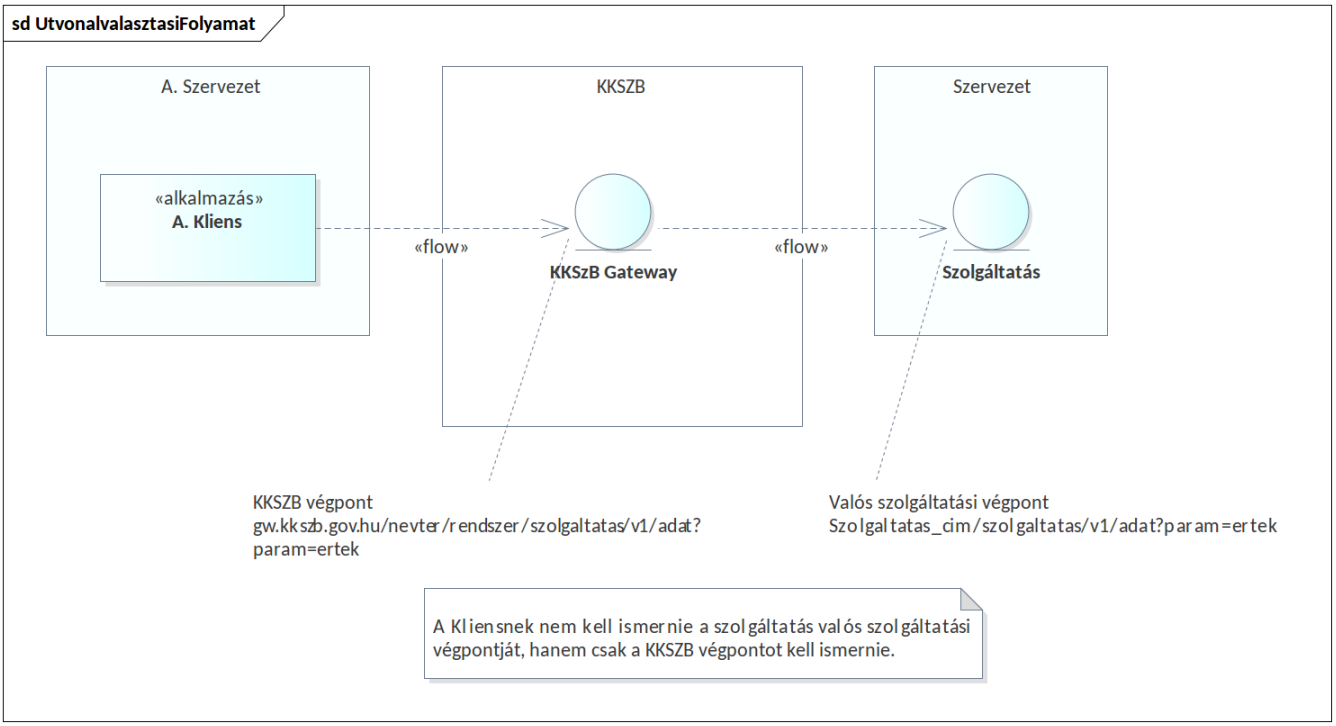
Annak ellenére, hogy maga a Szolgáltatás végpont definíció érvénytelen ha zár "/" jelet tartalmaz, maga a Szolgáltatás hívás tartalmazhatja, akár a szolgáltatás definiálhat rá működést is az interfész leírásban.

Például a *https://gw.kkszb.gov.hu/jarmu/szin/v1* szolgáltatást hívva a https://gw.kkszb.gov.hu/jarmu/szin/v1/ formában az összes járműrendszerben definiált színkódot visszaadja.

Mivel a verziószámmal bővített URL önálló Szolgáltatási végpontot jelent, ezért a Szolgáltatás két verziója között jelentős méretű eltérés is lehet.

Útvonalválasztási folyamat

Az útvonalválasztások esetén a *routing table* tartalmazza a KKSZB szolgáltatás végpontok és a valós szolgáltatási végpont közötti megfeleltetést. Minden egyes Szolgáltatás végpont egy darab valós végponttal rendelkezik.



Ábra 3. Útvonalválasztás folyamata

Az útvonalválasztás esetén az *A. Kliens* a KKSZB gatewayen keresztül éri el a *Szervezet Szolgáltatását.*

KKSZB rate limit

A KKSZB rendszer képes arra, hogy az adott Szolgáltatást elérő kérések számát korlátozza (rate-limiting). A korlát (rate-limit) beállítása a Szolgáltatás Felelős feladata, alapértelmezetten nincs korlátozás.

A korlátozás beállításával ugyan csökkenthető a Szolgáltatásra jutó terhelés, de ettől még a kérést küldő fél – amennyiben ezt túllépi –, célja nem teljesül, nem fog válaszhoz jutni, így a felhasználók kiszolgálási minősége sérül. Javasoljuk, hogy a korlátozást csak átmenetileg, vagy végső esetben használja, és helyette a Szolgáltatás felskálázásával, vagy gyorstárak használatával próbálja teljesíteni a felhasználói igényeket.

A korlátozást kérések számával lehet megadni az adott percre vonatkozóan, például a 60 érték jelentése: 1 perc alatt maximum 60 kérés érheti el a Szolgáltatást, ha ettől több kérés érkezik, akkor **HTTP429 Too Many Requests** üzenetet kap válaszként a kérdező fél, a HTTP fejlécben a beállított korlátozás értékét visszakapja a **x-kk-rate-limit** mezőben.

A Szolgáltatás Felelős a korlátozást az engedélyezett Szolgáltatás Elérési Jogosultságnál tudja megadni, és bármikor, igénye szerint módosítani.

Ezzel a korlátozás hatása három paramétertől függ:

- a beállított korlát értékétől, 0 esetén nincs korlátozás (alapértelmezett)
- a kliens rendszer azonosítójától, amely a SZEK-ben szerepel
- az elérendő Szolgáltatás azonosítójától, amely a SZEK-ben szerepel
- a jogalaptól, amely a SZEK-ben szerepel

A fentiek alapján például lehetséges a szabályozást úgy beállítani, hogy ugyanannak a lekérdező félnek, ugyanarra a Szolgáltatásra, a jogalapjától függően eltérő korlátozást alkalmazunk.

A KKSZB rate-limit szolgáltatása felmenő rendszerben kerül bevezetésre, a jelenleg működő rendszereket nem befolyásolja, változtatás nem szükséges részükről, amennyiben továbbra sem veszik igénybe ezt a szolgáltatást.

A KKSZB rate-limit szolgáltatás – amennyiben be van kapcsolva –, érinti a Szolgáltatást és érinti az azt igénybe vevő kliens rendszert is, ezért az alábbiakban összefoglaljuk azt, hogy mit kell figyelembe vennie ezen szereplőknek.

Szolgáltatás esetén

A KKSZB rendszer a rate-limiting funkcióval korlátozza a Szolgáltatás terhelését, így **a korlátot meghaladó kérésszám esetén, a kérés nem éri el a Szolgáltatást,** ezért ott log bejegyzés a kérésről nem történik!

Szolgáltatás felelősként bizonyosodjon meg arról, mielőtt bekapcsolja a korlátozást

(0-tól eltérő értéket ad meg), hogy a kapcsolódó fél képes a **HTTP429 Too Many Requests** üzenetet kezelésére.

Szolgáltatás felelősként vegye figyelembe, hogy milyen megállapodás (szerződés, törvényi, egyéb előírás) vonatkozik a Szolgáltatás elérésére, mielőtt bekapcsolja a korlátozást (0-tól eltérő értéket ad meg) – praktikusan a szerződés, vagy megállapodás, vagy engedély tartalmazza a kérés/perc értéket.

Szolgáltatás felelősként bármikor lehetősége van módosítani a korlátozás értékét (csökkenteni, növelni, vagy 0 értékre állítva kikapcsolni), amely legfeljebb 2 percen belül életbe lép.

A beállított korlátozás 1 percre vonatkoztatott kérés számot jelent. Ez azt is jelenti, hogy szélsőséges esetben akár 1-2 másodperc alatt is beérkezhet ez a kérés szám, az ez utáni időszakban viszont újabb kérés nem fogja elérni már a Szolgáltatást.

Kliens rendszer esetén

Amennyiben nincs korlátozva a kérés szám az adott elérésében, úgy nincs tennivalója.

Ha a korlátozás bekapcsolásra kerül, akkor szükséges implementálni a kérést indító alkalmazásban a **HTTP429 Too Many Requests** üzenetet kezelését.

Ha lehetősége van, akkor úgy implementálja a kérést küldő alkalmazást, hogy a **HTTP429 Too Many Requests** üzenetet kezelje. Ha biztosan tudja, hogy mennyi kérést intézhet az adott Szolgáltatás elé, akkor célszerű önkorlátozást bevezetni, amennyiben az technikailag megoldható.

Amennyiben a kérések száma túllépi a korlátot, akkor a **HTTP429 Too Many Requests** üzenetet kapja mindaddig, amíg a kérések száma az utolsó percre vonatkoztatva le nem csökken a korlát alá.

A **HTTP429 Too Many Requests** üzenetet a KKSZB rendszer küldi, a Szolgáltatás oldalon ezt nem fogják érzékelni, nem kapja meg a Szolgáltatás a kérést, így ott a kérésről semmilyen információ nem keletkezik.

Meg lehet különböztetni azt, hogy a HTTP429 válasz a KKSZB-től, vagy a mögötte lévő Szolgáltatástól (vagy egyéb KKSZB mögötti eszköztől, szoftvertől) érkezett-e. **Ha a KKSZB rendszertől érkezik a válasz,** akkor a **x-kk-gw-status-message** és a **x-kk-rate-limit** HTTP fejlécek szerepelnek a válaszban.

Az **x-kk-rate-limit** HTTP fejléc értéke a beállított korlátozás (rate-limit) értékét tartalmazza.

Egyéb korlátozások

Előfordulhat, hogy a KKSZB rendszeren kívül más komponens – akár maga a Szolgáltatás –, szintén (vagy csak az) tartalmaz korlátozó funkciót, amely ugyanúgy a **HTTP429 Too Many Requests** választ adhatja a KKSZB rendszertől függetlenül.

Amennyiben problémája adódik a korlátozással kapcsolatban, akkor vegye fel a kapcsolatot a

Szolgáltatás Felelősével.

Definíciók

KKSZB

Központi Kormányzati Szolgáltatás Busz

[Szerződő fél]

Egy tetszőleges (piaci vagy közigazgatási) szervezet, amely csatlakozik a KKSZB rendszerhez.

[Kapcsolódó fél]

Egy tetszőleges kapcsolódó fél, mely csatlakozik a KKSZB-hez, és az ott lévő szolgáltatásait (szolgáltató), vagy az ott igénybe vett szolgáltatások (kliens) tekintetében a kapcsolódó félhez rendelt személyek eljárhatnak.

[Kliens]

(Kliens) Kapcsolódó fél, aki a KKSZB rendszerben nyújtott szolgáltatásokat használja HTTP protokollon keresztül.

[Szolgáltató]

(Szolgáltató) Kapcsolódó fél, aki a KKSZB rendszerben szolgáltatásokat publikál HTTP protokollon keresztül.

[Kapcsolattartó]

Olyan természetes személy, akit a kapcsolódó fél felhatalmazott, hogy a KKSZB rendszerben a kapcsolódó felet képviselje.

[Szolgáltatás felelős]

Az a személy, aki a KKSZB rendszerben a Szolgáltatás felett teljes jogkörrel rendelkezik: szolgáltatásokat futtat, publikálja a KKSZB rendszerben, a Kliensek Szolgáltatáshoz történő hozzáférését szabályozza, stb.

[Szolgáltatás Katalógus Kezelő]

Az a személy, aki a KKSZB rendszerben a *Kapcsolódó fél* által a Szolgáltatás interfész leírásának menedzsmentjével meghatalmazott személy, szolgáltatás létrehozását vagy megszüntetését nem kezdeményezheti.

[RFNY admin]

Az a személy, aki a KKSZB Rendszer Felhatalmazási Nyilvántartás rendszerében admin szerepkörrel meghatalmazott, a rendszerhez csatlakozó feleket kezeli, és a szolgáltatás kérelmeket elbírálja, felülvizsgálja, felfüggeszti amennyiben szükséges.

[Szolgáltatás]

(KKSZB szolgáltatás) Alkalmazás, amely a KKSZB rendszerben szolgáltatást nyújt HTTP protokollon keresztül. Szolgáltatásnak nevezünk minden KKSZB-n igénybe vehető, HTTP-n elérhető webes szolgáltatást.

[KKSZB felügyeleti szerv]

Az a szervezet, amely a KKSZB rendszer felügyeletét ellátja, rendelkezik felette és működésére

hatással lehet.

[KKSZB Gateway]

(röv.: gateway vagy GW) Olyan KKSZB architekturális elem, amely az adott kliens vagy szolgáltatás csatlakozását lehetővé teszi a KKSZB rendszerbe.

[Rendszer Felhatalmazási Nyilvántartás]

(röv. *RFNY*) Rendszer Felhatalmazási Nyilvántartás, olyan Webes felülettel rendelkező rendszer, amely a KKSZB rendszer kliens, szolgáltatás és gateway (átjáró) alkalmazásainak a hitelesítését és jogosultság kiosztását biztosítja az arra felhatalmazott személyek által.

[Szolgáltatás Katalógus]

Szolgáltatás Katalógus, amely a KKSZB rendszer összes elérhető szolgáltatásáról információt nyújt: *KKSZB végpont*, interfész leírás és egyéb adatok.

[Green-Page]

Green-Page, a KKSZB rendszerhez való csatlakozást segítő technikai, fejlesztői ismeretanyagok, cikkek megosztását, ismeretanyagok közzétételét szolgáló alkalmazás.

[Statisztika]

Statisztika, olyan Webes felülettel rendelkező rendszer, amely a KKSZB RFNY adatbázisából, és a BigData log rendszeréből statisztikákat, kimutatásokat készít, az admin, a SZF és a KT felhasználók számára.

[Aszinkron Szolgáltatás]

Aszinkron Szolgáltatás, amely a KKSZB rendszerben biztosítja a HTTP protokollon történő aszinkron üzenet küldés lehetőségét.

[Echo Szolgáltatás]

Az Echo szolgáltatás a beüzemelés és az üzemi állapot ellenőrzésére szolgál a Kliensek irányából, a beküldött üzenetet visszaküldi.

[Próba kliens]

Olyan kliens, amely a KKSZB rendszert felügyelő szervezet saját, kliensként csatlakozó alkalmazása, amely segíti a valós kliens és szolgáltatás kapcsolat ellenőrzését a Kliens irányából.

[szolgáltatás létrehozási kérelem]

A szolgáltatók, ha létre akarnak hozni egy új szolgáltatást, szolgáltatás létrehozási kérelmet hoznak létre, melyet a KKSZB felügyeleti szerv megfelelő jogosultsággal rendelkező felhasználója elbírál.

[szolgáltatási kérelem]

A szolgáltatók, ha létre akarnak hozni egy új szolgáltatást, vagy meg akar szüntetni, akkor a szolgáltatásra vonatkozó kérelmet tölt ki és küld be, melyet a KKSZB felügyeleti szerv megfelelő jogosultsággal rendelkező felhasználója elbírál.

[szolgáltatás elérési jogosultság kérelem]

Amennyiben egy kapcsolódó fél el kíván érni egy szolgáltatást, a szolgáltatás elérési jogosultság

megszerzéséhez egy szolgáltatás elérési jogosultság kérelmet kell létrehoznia, melyet az adott szolgáltatást nyújtó Szolgáltatás felelőse elbírál.

[szolgáltatás elérési jogosultság]

A kliensnek rendelkeznie kell az adott KKSZB szolgáltatás eléréséhez szükséges jogosultsággal. Nem keverendő össze az egyéb, *szakrendszeri jogosultságokkal.*

[kliens autentikációs token]

Olyan Json Web Token (JWT), amely az RFNY által kiállított, kizárólag egy adott kapcsolódó fél, egy adott *Szolgáltatás elérési jogosultságához* tartozó, KKSZB rendszerben történő hitelesítésére szolgál. Érvényessége a kiállítástól számított maximum 12 hónap. Biztonságos, megbízható módon kezelendő, titkos információ, csak a KKSZB és a kliens ismerheti.

[kliens access token]

Olyan Json Web Token (JWT), amely az RFNY által kiállított, rövid lejáratú (néhány perc), kizárólag egy adott kliens által a KKSZB szolgáltatások elérésére szolgál. Csak a kliens autentikációs token birtokában szerezhető meg a KKSZB kliens gateway által. A szolgáltatások számára megismerhető információ.

[KKSZB szolgáltatás végpont]

Olyan URL, amely a KKSZB rendszerben értelmezett HTTP szolgáltatási végpont, például: https://gw.kkszb.gov.hu/jarmu/leksz/rsz/v1 Ez az adott szolgáltatás előtt álló gatewayen értelmezett elérési pont. A szolgáltatás azonosítóval felírva a *KKSZB szolgáltatás végpont:* `https://gw.kkszb.gov.hu/szolgáltatás-azonosító`

[valós szolgáltatási végpont]

Olyan URL, amely az adott szolgáltató eszköz környezetében értelmezett, a szolgáltatás oldali KKSZB gatewayről elérhető.

[szolgáltatás névtér]

(KKSZB szolgáltatás névtér) A KKSZB-n elérhető szolgáltatások névterekbe vannak szervezve. A névteret a szolgáltatás végpont URL útvonal részének első tagja azonosítja. Ez egyben a névtér egyedi azonosítója is, pl.: a *jarmu* névtér a következőképpen jelenik meg a szolgáltatás végpont URL-ben: *https://gw.kkszb.gov.hu/jarmu*

[szolgáltatás azonosító]

A KKSZB rendszerben értelmezett szolgáltatás azonosítója, például: **/jarmu/leksz/rsz/v1** vagy **/jarmu/private/leksz/eucaris/rsz/v1** A KKSZB szolgáltatás végpont és a **szolgáltatás azonosító** a protokoll (https) és a domain név (kkszb.gov.hu) kivételével megegyezik.

[KKSZB azonosítási rendszer]

A KKSZB átvállalja az alkalmazás szintű azonosítást a szolgáltatástól. A szakrendszeri felhasználók azonosítása és a szakrendszeri jogosultság kezelése továbbra is a szolgáltatás felelőssége marad.

[kliens alkalmazás]

Olyan alkalmazás, amely a KKSZB rendszerre, mint kliens csatlakozik és Szolgáltatást vesz igénybe.

[állapot információ végpont]

Olyan szolgáltatás végpont, amely 200 OK HTTP státus kóddal válaszol, amennyiben a szolgáltatás üzemszerűen használható.

[polimorf szolgáltatás]

Azokat a szolgáltatásokat, amelyek **ugyanazon lekérdezésre, ugyanazon végponton más választ adnak** attól függően, hogy mely rendszertől (vagy felhasználótól) érkezett a kérés, polimorf szolgáltatásoknak nevezzük.

függelék A: Kliens autentikációs token

A kiadott tokent kezelje bizalmasan. Azt csak a kliens szoftver és a KKSZB kliens gateway ismerheti meg az RFNY rendszeren kívül.

A kliens autentikációs token azonosít egy kliensként kapcsolódó rendszert (alkalmazást) a KKSZB rendszerben. A token kibocsájtója az RFNY, amely JWT formájú, és amelyet az RFNY digitálisan aláír.

A tokent csak az a kliens használhatja, amely részére kiadták, és csak a KKSZB rendszerben szabad azt felhasználni.

Minden jogosult kliens példány ugyanazt a tokent használja, nem szükséges alkalmazás példányonként újabb tokent igényelni.

Token kiadása

A token létrehozását és letöltését **a kapcsolódó fél kliens kapcsolattartó szereppel rendelkező felhasználói végzik** az RFNY webes felületén keresztül.

Token verziók

2023-ban, az RFNY CouchDB-ről MongoDB-re való átállásakor, a token egyes verzióját a kettes verzió váltotta fel. Az új verzió bevezetését az entitások azonosítójának a korábbi CouchDB-sről MongoDB-sre való módosulása indokolta.

Az adatbázis migrációjáig a token verziója egyes, CouchDB-ben generált ID-kal töltött serviceId, sapId, legalbasisId mezőkkel, az után kiállítottak verziója kettes, MongoDB ID-kal generált.

Ez a változás, mivel az újonnan generált tokenek értékeit megváltoztatta, a token ellenőrzési eljárásának módosítását követelte.

Az első verzió használata, a tokenek éves lejáratása miatt 2024-től folyamatosan megszűnik.

Token szerkezete

Minta kliens autentikációs token - version 1

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjEifQ.eyJqdGkiOiIyZDgyNWY2ZC0xZmFiLTRmY2E
tOGIxMi0wZWYxNDU4ZTBkMDAiLCJpc3MiOiJ1cm46c3lzOmtrc3piOmZueSIsInN1YiI6InVybjpwaWQ6a2tze
mI6cGVlcjEiLCJhdWQiOiJ1cm46c3lzOmtrc3piOmdhdGV3YXkiLCJ0eXBlIjoidXJuOnRva2VuOmtrc3piOmN
saWVudDphdXRoIiwiaWF0IjoxNDkzMTEzNjUzLCJuYmYiOjE0OTMxMTM2NTMsImV4cCI6MTUyNDY0OTY1Mywic
2VydmljZUlkIjoiTDJwaGNtMTFMM05sY25acFkyVTFMM1l4Iiwic2VydmljZVVyaSI6Ii9qYXJtdS9zZXJ2aWN
lNS92MSIsInNhcElkIjoiNDFmNzVlOTYtZWI0Ni00NTBjLTk0NmUtNTE3YmVjZjVlYzlhIiwic2FwTmFtZSI6I
mRlZmF1bHQiLCJsZWdhbEJhc2lzSWQiOiJkY2RmMWU4MiIsIm5hbWUiOiJUb2tlbjEiLCJsZWdhbEJhc2lzQ29
kZSI6IkpBUjEyMDJBIiwic2VjdXJpdHlDbGFzcyI6NCwidmVyc2lvbiI6MX0.i6NZ3G4zik1V8RJvdMphSMbwj
```

```
dGxSwR3Bim4X608DtXWjWQKie-ymrwaFstvn9lGv1o1jLwpDgc-
Ub9gOYzrEbl4WMxFVo5Bz9uWA0vJMDwBcKO1QhH0u3sBhYqtVuQD_TchUt61cLBobESVaa1a9QSGRAB-
UNVlnVXEbJBJvF3nXxVNB7RZ-dUix4r2jooGo8pqaaBQocyUJgIHUqGqJuyJeGVV0LtrIoMWxNkmHNSPaIpL-
ns_XrumzVZZqYOhGbuODMrmCWJRzYl0CsPZo8BtUnRzEhK1kgrBkaSwlKlPEKxv-
S9fKxOnhiVp6KBbNZhLj75Y0rdzSy3K5eol3Q
```

Minta kliens autentikációs token - version 2

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjMifQ.eyJqdGkiOiIyZDgyNWY2ZC0xZmFiLTRmY2E
tOGIxMi0wZWYxNDU4ZTBkMDAiLCJpc3MiOiJ1cm46c3lzOmtrc3piOmZueSIsInN1YiI6InVybjpwaWQ6a2tze
mI6cGVlcjEiLCJhdWQiOiJ1cm46c3lzOmtrc3piOmdhdGV3YXkiLCJ0eXBlIjoidXJuOnRva2VuOmtrc3piOmN
saWVudDphdXRoIiwiaWF0IjoxNDkzMTEzNjUzLCJuYmYiOjE0OTMxMTM2NTMsImV4cCI6MjIyNDY0OTk5OSwic
2VydmljZUlkIjoiNjM5YjZhNDIzNmQ2NWMwNmY2ODg4YTBlIiwic2VydmljZVVyaSI6Ii9qYXJtdS9zZXJ2aWN
lNS92MSIsInNhcElkIjoiNjM5YjZhNDIzNmQ2NWMwNmY2ODg4YTBmIiwic2FwTmFtZSI6ImRlZmF1bHQiLCJsZ
WdhbEJhc2lzSWQiOiI2MzliNmE0MjM2ZDY1YzA2ZjY4ODhhMGciLCJuYW1lIjoiVG9rZW4xIiwibGVnYWxCYXN
pc0NvZGUiOiJKQVIxMjAyQSIsInNlY3VyaXR5Q2xhc3MiOjQsInZlcnNpb24iOjJ9.CcxpOhnShARbIq3QnTJV
Rjgt1rmynx3J6R5IXKC77Lw3Tk_lEOcGoW9Kz09pC4riAK3yCMC0BtOomhlG3OGZCvlZSHSjG7v2EmrTeS8qs-
iNIfPPArew0U2eNCOoOoPvGuuyBQYxQ9mg5i3rPdM8nxzNnyC-
xVSbIyxkdQ803eONqLRuObgvOCubqh20rLiyZUqzZ83yuCDsWR_WYYB3nikoIiKJLWqfTkUPbrtlhVEBAEd68n
sWGVT0K0Ie3oEqNs03Z_eXrBJLnwJnReZaEhZ5gqhNztXtWDFIdAdfNyW3V2PXpz5MmyDpu9rSRspYlK_A0Qvk
2j6n9mVL1ToSFQ
```

Az autentikációs token egy [JWT], amely az RFNY által alá van írva. Kódolása [Base64url], amely olvasható, tartalma JSON.

Minta kliens autentikációs token header Base64url kódolás nélkül (JWT header)

```
{
  "alg": "RS256",
  "typ": "JWT",
  "kid": "3"
}
```

Minta kliens autentikációs token body Base64url kódolás nélkül (JWT claim) - version 1

```
{
  "jti": "2d825f6d-1fab-4fca-8b12-0ef1458e0d00",
  "iss": "urn:sys:kkszb:fny",
  "sub": "urn:pid:kkszb:peer1",
  "aud": "urn:sys:kkszb:gateway",
  "type": "urn:token:kkszb:client:auth",
  "iat": 1493113653,
  "nbf": 1493113653,
  "exp": 2224649999,
  "serviceId": "L2phcm11L3NlcnZpY2U1L3Yx",
  "serviceUri": "/jarmu/service5/v1",
  "sapId": "41f75e96-eb46-450c-946e-517becf5ec9a",
  "sapName": "default",
  "legalBasisId": "dcdf1e82",
```

```
  "name": "Token1",
  "legalBasisCode": "JAR1202A",
  "securityClass": 4,
  "version": 1
}
```

Minta kliens autentikációs token body Base64url kódolás nélkül (JWT claim) - version 2

```
{
  "jti": "2d825f6d-1fab-4fca-8b12-0ef1458e0d00",
  "iss": "urn:sys:kkszb:fny",
  "sub": "urn:pid:kkszb:peer1",
  "aud": "urn:sys:kkszb:gateway",
  "type": "urn:token:kkszb:client:auth",
  "iat": 1493113653,
  "nbf": 1493113653,
  "exp": 2224649999,
  "serviceId": "639b6a4236d65c06f6888a0e",
  "serviceUri": "/jarmu/service5/v1",
  "sapId": "639b6a4236d65c06f6888a0f",
  "sapName": "default",
  "legalBasisId": "639b6a4236d65c06f6888a0g",
  "name": "Token1",
  "legalBasisCode": "JAR1202A",
  "securityClass": 4,
  "version": 2
}
```

Táblázat 2. Az autentikációs token header részében szereplő kulcsok és leírásuk

Kulcs	Leírás
alg	A token aláírásához használt algoritmus (JWA). Jelenleg két algoritmus támogatott: ***RS256*** és ***ES256***
typ	A token típusa. ***JWT***
kid	Az aláíráshoz használt kulcs azonosítója, pl.: **1**

Táblázat 3. Az autentikációs token body részében szereplő kulcsok és leírásuk

Kulcs	Leírás
jti	Az autentikációs token egyedi azonosítója.
iss	A kibocsájtó egyedi azonosítója. ***[urn:sys:kkszb:fny]***
sub	A kapcsolódó fél (kliens alkalmazás) egyedi azonosítója, aki a szolgáltatást igénybe veszi, pl.: **urn:pid:kkszb:peer1**
type	A token típusa ***[urn:token:kkszb:client:auth]***
aud	A token célközönségének azonosítója ***[urn:sys:kkszb:gateway]***
iat[1]	Kibocsájtás időpontja (POSIX time).

Kulcs	Leírás
nbf[1]	Ezen időponttól érvényes a token (POSIX time).
exp[1]	Ebben az időpontban jár le a token. Megjegyzés: előfordulhat az az eset, hogy az *exp* időpont a szolgáltatáshoz történő beérkezéskor már elmúlt. Ez adódhat abból, hogy a szolgáltatás szerveren és az RFNY szerver időpontja eltér, valamint abból, hogy a KKSZB rendszeren történő áthaladás időt vett igénybe, és az ellenőrzéskor még nem érte el az *exp* időpontot Az előzőek miatt a szolgáltatáson ezt az értéket nem kell ellenőrizni, arról a KKSZB gondoskodik.
serviceId	Az autentikációs tokenhez tartozó szolgáltatás egyedi azonosítója. (v1: CouchDB ID, v2: MongoDB ID)
serviceUri	Az autentikációs tokenhez tartozó szolgáltatás URI-ja.
sapId	A szolgáltatás elérési jogosultság egyedi azonosítója. (v1: CouchDB ID, v2: MongoDB ID)
sapName	A szolgáltatás elérési jogosultság neve, amely a könnyebb azonosíthatóságot és nyomonkövethetőséget segíti.
name	Az autentikációs token neve, például: az alkalmazás címkéje, amelyben felhasználásra kerül.
legalBasisId	Jogalap azonosító. (v1: CouchDB ID, v2: MongoDB ID)
legalBasisCode	Jogalapkód értéke, amelyet a Szolgáltatás felelős adott meg az RFNY felületén ehhez a jogalaphoz.
securityClass	A szolgáltatást hívó fél kérésének IBTV biztonsági osztály értéke, numerikus érték 1-5 között (implicit). Értéke a Szolgáltatás Elérési Jogosultság Kérelem kitöltésekor kerül megadásra a Kapcsolódó fél által.
version	A token verziója.

[1] Az iat, nbf és exp értéke 'POSIX time', másodperc pontossággal.

függelék B: Kliens access token

A kliens access token felhasználója a KKSZB gateway, a kliens oldalon nem használt, ugyanakkor a kliens access token megismerhető a szolgáltatás számára is, így az ott igény szerint felhasználható.

A kliens access token tartalmazza a kliensről szóló információkat egy adott kérésben. A token kibocsájtója az RFNY, [JWT] formájú, amelyet az RFNY digitálisan aláír.

Token kiadása

A token kiadását az RFNY automatikusan végzi a Kliens autentikációs token alapján. Csak a KKSZB Gateway képes a token megszerzésére.

A token a HTTP fejlécben az **x-kk-access-token** mezőben elérhető a szolgáltatás oldalon.

Token verziók

2023-ban, az RFNY CouchDB-ről MongoDB-re való átállásakor, a token egyes verzióját a kettes verzió váltotta fel. Az új verzió bevezetését az entitások azonosítójának a korábbi CouchDB-sről MongoDB-sre való módosulása indokolta.

Az adatbázis migrációjáig a token verziója egyes, CouchDB-ben generált ID-kal töltött serviceId, sapId, legalbasisId mezőkkel, az az után kiállítottak verziója kettes, MongoDB ID-kal generált.

Ez a változás, mivel a generált token értékeit megváltoztatta, a token ellenőrzési eljárásának módosítását követelte.

Token szerkezete

Minta kliens access token - version 1


```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjEifQ.eyJqdGkiOiJkNzNlYzBmYy1lZTU2LTQwZDk
tYjVhYS1iZDE3NDZlMmJhM2MiLCJpc3MiOiJ1cm46c3lzOmtrc3piOmZueSIsInN1YiI6InVybjpwaWQ6a2tze
mI6cGVlcjEiLCJ0eXBlIjoidXJuOnRva2VuOmtrc3piOmNsaWVudDphY2Nlc3MiLCJpYXQiOjE0OTMxMTM2NTM
sIm5iZiI6MTQ5MzExMzY1MywiZXhwIjoxNDkzMTE0MjUzLCJzZXJ2aWNlSWQiOiJMMnBoY20xMUwzTmxjblpwW
TJVMUwzWXgiLCJzZXJ2aWNlVXJpIjoiL2phcm11L3NlcnZpY2U1L3YxIiwiYXV0aHRva2VuSnRpIjoiMmQ4MjV
mNmQtMWZhYi00ZmNhLThiMTItMGVmMTQ1OGUwZDAwIiwic2FwSWQiOiI0MWY3NWU5Ni1lYjQ2LTQ1MGMtOTQ2Z
S01MTdiZWNmNWVjOWEiLCJzYXBOYW1lIjoiZGVmYXVsdCIsImF1dGh0b2tlbk5hbWUiOiJUb2tlbjEiLCJsZWd
hbEJhc2lzSWQiOiJkY2RmMWU4MiIsImxlZ2FsQmFzaXNDb2RlIjoiSkFSMTIwMkEiLCJzZWN1cml0eUNsYXNzI
jo0LCJ2ZXJzaW9uIjoxfQ.UUnMFj2oyG3tsDRHQCKB5yFL2rSX-
SZpHsAVie522FSAhAUXePZmJJNCYq8wVqkzrBvQ_VmlM3X5jyKuDR_TrKeNAOWVrCG4B0VwsRc9U0hBNuaxYLe
jktH_z-8SFdgIF7nlBUoQ2JnFOxZQLlpKQY7-OHObUj6aJBDbnqt8O731Jga2QjY-
1HM9e9dbONLXBY_FrhXYpQZWDvokRNndAqZwNQ2YeIy8ONds6cUhQTh_8-
wRjOBvkqvQAjXvpraJMG5_WtWYWY2n_wMQtZgdaszxjdnT191KbQoLqoO-P-
YVAmR7FrF6yXdJvvXZrnNZM4kIs4gCcXn_xH3njomB-Q
```

Minta kliens access token - version 2

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjMifQ.eyJqdGkiOiJkNzNlYzBmYy1lZTU2LTQwZDk
tYjVhYS1iZDE3NDZlMmJhM2MiLCJpc3MiOiJ1cm46c3lzOmtrc3piOmZueSIsInN1YiI6InVybjpwaWQ6a2tze
mI6cGVlcjEiLCJ0eXBlIjoidXJuOnRva2VuOmtrc3piOmNsaWVudDphY2Nlc3MiLCJpYXQiOjE0OTMxMTM2NTM
sIm5iZiI6MTQ5MzExMzY1MywiZXhwIjoxNDkzMTE0MjUzLCJzZXJ2aWNlSWQiOiI2MzliNmE0MjM2ZDY1YzA2Z
jY4ODhhMGUiLCJzZXJ2aWNlVXJpIjoiL2phcm11L3NlcnZpY2U1L3YxIiwiYXV0aHRva2VuSnRpIjoiMmQ4MjV
mNmQtMWZhYi00ZmNhLThiMTItMGVmMTQ1OGUwZDAwIiwic2FwSWQiOiI2MzliNmE0MjM2ZDY1YzA2ZjY4ODhhM
GYiLCJzYXBOYW1lIjoiZGVmYXVsdCIsImF1dGh0b2tlbk5hbWUiOiJUb2tlbjEiLCJsZWdhbEJhc2lzSWQiOiI
2MzliNmE0MjM2ZDY1YzA2ZjY4ODhhMGciLCJsZWdhbEJhc2lzQ29kZSI6IkpBUjEyMDJBIiwic2VjdXJpdHlDb
GFzcyI6NCwidmVyc2lvbiI6Mn0.Ms1KnXmSkCcMkD_eLyuqGV-
GS6LKMQz6DPP0T2LCpHOUPjBJxHAglqmo3bznvv7vtjeOQf7yW3rQeJs1QZ6A0hY_5QLbtNzZOLgH6sTR0_Zde
jQtS1V3qnccubIf_QOhpTh2NHnNSI5Tci7_kLGrfy-
SQH4DQDhkq_IEtixwJuxPvNWHAmbazENnY_1lxeecjC3euZHszPohKoJx0AhY39SqTfjgao__-
w_p_eILmEYE2n4yEWLv8JNdzpPbUK1sbhQf7hiBz7e20LIrdNWiDH7_yYXmmSAzWoQM6R4b4OPV8Bv5r20Gs9A
sVlVSk2pv6dAW_obCxXhA25hAEhylRA
```

Az access token egy [JWT], amely az RFNY által alá van írva. Kódolása [Base64url], amely olvasható, tartalma JSON.

Minta kliens access token header Base64url kódolás nélkül (JWT header)

```
{
  "alg": "RS256",
  "typ": "JWT",
  "kid": "3"
}
```

Minta kliens access token body Base64url kódolás nélkül (JWT claim) - version 1

```
{
  "jti": "d73ec0fc-ee56-40d9-b5aa-bd1746e2ba3c",
  "iss": "urn:sys:kkszb:fny",
  "sub": "urn:pid:kkszb:peer1",
  "type": "urn:token:kkszb:client:access",
  "iat": 1493113653,
  "nbf": 1493113653,
  "exp": 2224649999,
  "serviceId": "L2phcm11L3NlcnZpY2U1L3Yx",
  "serviceUri": "/jarmu/service5/v1",
  "authtokenJti": "2d825f6d-1fab-4fca-8b12-0ef1458e0d00",
  "sapId": "41f75e96-eb46-450c-946e-517becf5ec9a",
  "sapName": "default",
  "authtokenName": "Token1",
  "legalBasisId": "dcdf1e82",
  "legalBasisCode": "JAR1202A",
  "securityClass": 4,
  "version": 1
}
```

Minta kliens access token body Base64url kódolás nélkül (JWT claim) - version 2


```
{
  "jti": "d73ec0fc-ee56-40d9-b5aa-bd1746e2ba3c",
  "iss": "urn:sys:kkszb:fny",
  "sub": "urn:pid:kkszb:peer1",
  "type": "urn:token:kkszb:client:access",
  "iat": 1493113653,
  "nbf": 1493113653,
  "exp": 1493114253,
  "serviceId": "639b6a4236d65c06f6888a0e",
  "serviceUri": "/jarmu/service5/v1",
  "authtokenJti": "2d825f6d-1fab-4fca-8b12-0ef1458e0d00",
  "sapId": "639b6a4236d65c06f6888a0f",
  "sapName": "default",
  "authtokenName": "Token1",
  "legalBasisId": "639b6a4236d65c06f6888a0g",
  "legalBasisCode": "JAR1202A",
  "securityClass": 4,
  "version": 2
}
```


Táblázat 4. Az access token header részében szereplő kulcsok és leírásuk

Kulcs	Leírás
alg	A token aláírásához használt algoritmus (JWA). Jelenleg két algoritmus támogatott: ***RS256*** és ***ES256***
typ	A token típusa. ***JWT***
kid	Az aláíráshoz használt kulcs azonosítója, pl.: **1**

Táblázat 5. Access tokenben szereplő kulcsok és leírásuk

Kulcs	Leírás
jti	Az access token egyedi azonosítója, UUIDv4.
iss	A kibocsájtó egyedi azonosítója, , sztring, max. 100 karakter, URN formában, pl.: urn:sys:kkszb:fny
sub	A kérést küldő egyedi azonosítója, sztring, max. 100 karakter, URN formában, pl.: urn:pid:kkszb:xxxx.
type	A token típusa, , sztring, max. 100 karakter, URN formában, pl.: urn:token:kkszb:client:access
iat[1]	Kibocsájtás időpontja (POSIX time), numerikus, pl.: 1504703686
nbf[1]	Ezen időponttól érvényes a token (POSIX time), numerikus, pl.: 1504703686

Kulcs	Leírás
exp[1]	Ebben az időpontban jár le a token (POSIX time), numerikus, pl.: 1504703686 Megjegyzés: előfordulhat az az eset, hogy az *exp* időpont a szolgáltatáshoz történő beérkezéskor már elmúlt. Ez adódhat abból, hogy a szolgáltatás szerveren és az RFNY szerver időpontja eltér, valamint abból, hogy a KKSZB rendszeren történő áthaladás időt vett igénybe, és az ellenőrzéskor még nem érte el az *exp* időpontot. Az előzőek miatt a szolgáltatáson ezt az értéket nem kell ellenőrizni, arról a KKSZB gondoskodik.
serviceId	Az access token által elérhető szolgáltatás egyedi azonosítója (hash), string, max. 500 karkater. (v1: CouchDB ID, v2: MongoDB ID)
serviceUri	Az access token által elérhető szolgáltatás URI-ja, sztring, max. 201 karakter.
authtokenJti	Annak a kliens autentikációs tokennek az egyedi azonosítója, ami alapján ez az access token ki lett adva, UUIDv4.
sapId	A szolgáltatás elérési jogosultság egyedi azonosítója, sztring. (v1: CouchDB ID, v2: MongoDB ID)
sapName	A szolgáltatás elérési jogosultság neve, amely a könnyebb azonosíthatóságot és nyomon követhetőséget segíti, sztring, max. 30 karakter.
authtokenName	Az autentikációs token neve, szabadszöveges string, max. 20 karakter, például: az alkalmazás címkéje, amelyben felhasználásra kerül.
legalBasisId	Jogalap azonosító, mely v1 (CouchDB) esetén 8 karakter hosszú hexadecimális sztring, amelyet a KKSZB generál, sztring, hexadecimális érték, pontosan 8 karakter, pl.: a70d2dfd, v2 (MongoDB) esetén, Mongo által generált Id.
legalBasisCode	Jogalapkód, opcionális (ha nincs értéke, akkor a fejlécben nem szerepel), **amelyet az RFNY webes felületén a Szolgáltatás felelős ad meg, így közvetlenül használható a Szolgáltatást nyújtó alkalmazásban**, a következő karaktereket tartalmazhatja: a-z, A-Z, 0-9, -, _, / és . (pont) karakterek, max. 20 karakter.
securityClass	A szolgáltatást hívó fél kérésének IBTV biztonsági osztály értéke, numerikus érték 2-5 között (implicit), az AccessToken securityClass mezőjével egyezik meg, a Szolgáltatás fel tudja használni arra, hogy a kérést a biztonsági szintnek megfelelő útvonalra terelje. Értéke a Szolgáltatás Elérési Jogosultság Kérelem kitöltésekor kerül megadásra a Kapcsolódó fél által.
version	A token verziója, numerikus érték.

[1] Az iat, nbf és exp értéke 'POSIX time', másodperc pontossággal.

A HTTP fejlécek közül az **x-kk-sap-name** (sapName) és az **x-kk-token-name** (authtokenName) értékét a HTTP szabvány követelménye szerint *escape-elni* kell,

amelyre a KKSZB a javascript **encodeURI()** függvényét használja. A megadott karakterhosszúságok a dekódolt értékre vonatkoznak, a **kódolt értékek hosszabbak lehetnek.**

függelék C: x-kk-client-id szerkezete

A Szolgáltatások a HTTP kérés fejlécben megkapják az kliens access tokenben található **sub** mező értékét.

Ez a mező az alábbi formában épül fel:

Táblázat 6. Példa: x-kk-client-id = urn:pid:kkszb:bm szerkezete

urn:pid:kkszb	A KKSZB rendszerben a *peer-id* típus jelzője.
bm	A kapcsolódó fél egyedi azonosítója a KKSZB rendszerben, jelen esetben Belügyminisztérium.

függelék D: Tanúsítványok

A KKSZB rendszer segítségével egyszerűsíthető és biztonságosabbá tehető a rendszerek közötti kapcsolat, mivel nem kell minden egyes szolgáltatási végpont tanúsítványát felvenni a kliensekhez, hanem elég csak a KKSZB rendszer **webszerver** tanúsítványát biztonságosként elfogadni.

Ugyanígy a szolgáltatás oldalon is jelentősen egyszerűsödik a tanúsítványok kezelése, mert nem kell minden egyes kliens alkalmazás tanúsítványát felvenni és menedzselni, elégséges a KKSZB rendszer **kliens tanúsítványát** felvenni.

A Szolgáltatásoknak kell beszerezni a GovCA tanúsítványokat és az emiatt szükséges domain nevet a saját valós szolgáltatási végpontjukhoz, amelyet a KKSZB elér, mint hívó fél.

A tanúsítványok elfogadása és megfelelő kezelése szerves része a biztonságnak.

A csatlakozott alkalmazások **szolgáltatás igénybe vevő** oldalán (kliens) el **kell** fogadni biztonságos tanúsítványnak a KKSZB rendszer szerver tanúsítványát **és csak azt.**

A valós szolgáltatási végponton el **kell** fogadni, **és csak azt,** a KKSZB rendszer kliens tanúsítványát biztonságosnak.

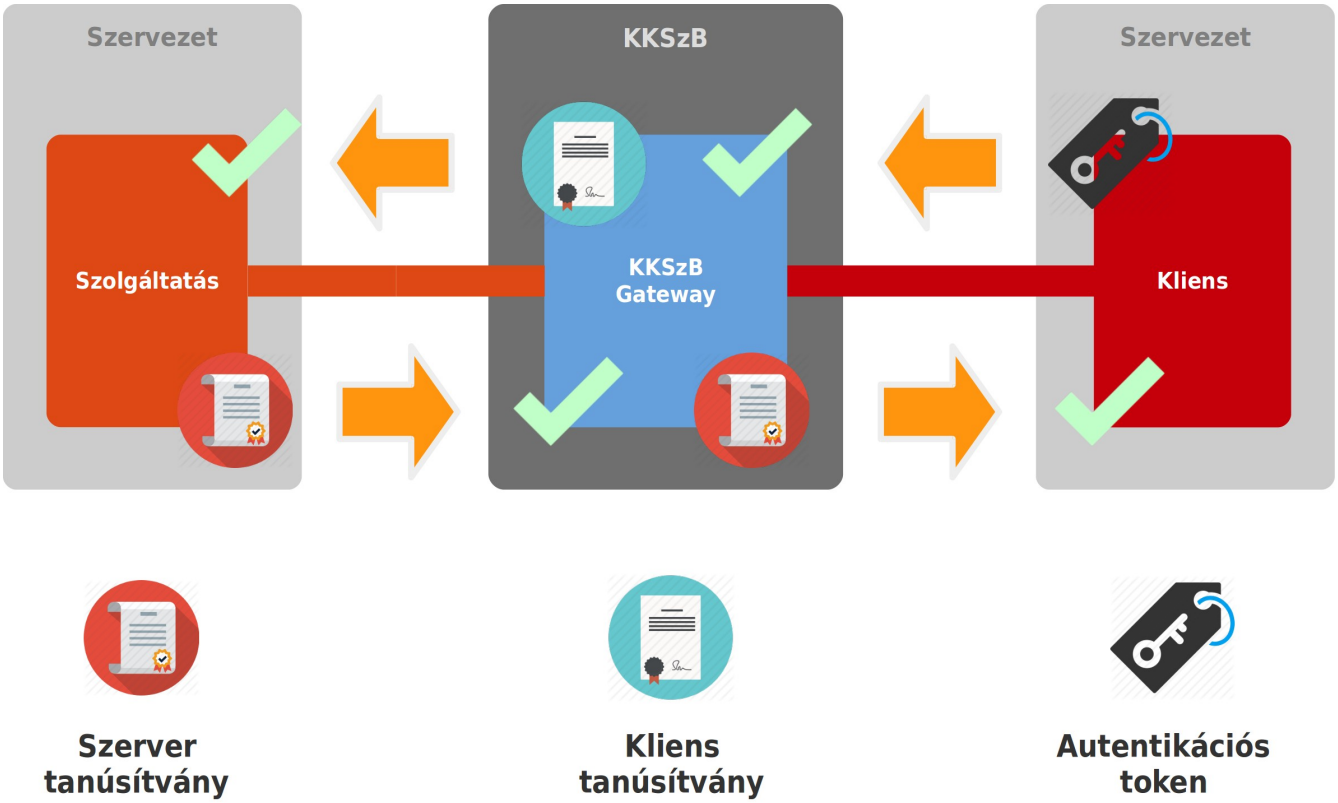
A KKSZB rendszerben (kkszb-gateway) a szolgáltatások részére kiadott szerver tanúsítványokat és **csak azt, kell** elfogadni biztonságosnak.

A KKSZB nem kér vagy vár el tanúsítványt a Kliensektől, hanem ezeket az kliens autentikációs token segítségével azonosítja. Ezzel rugalmasan kezelhetőek a Kliens alkalmazás oldali azonosítások.

A KKSZB rendszer a "Production" (PROD - éles) és a "Integration" (INT - integrációs teszt) környezetben is minősített tanúsítványokat szolgáltat.

A tanúsítvány kiadó a NISZ Gov CA, amely alapértelmezetten nincs az operációs rendszerekben és böngészőkben telepítve, így azokat - ha szükséges - fel kell ott venni és megbízhatónak kell jelölni (böngésző).

Az alábbi ábra szemlélteti a tanúsítványok elhelyezését.



Ábra 4. KKSZB rendszer tanúsítványai és elfogadásuk

A biztonságos kapcsolat lebontását hardver eszközök végzik, amelyek a tanúsítvány adatait a HTTP fejlécbe helyezik.

Amennyiben a tanúsítvány kezelés nem megfelelően kezelt úgy az alábbi veszélyek állnak fent:

- **DNS név eltérítés támadás esetén:** a gw.kkszb.gov.hu a támadó oldalára mutat, így - **amennyiben a csatlakozó kliens alkalmazás nem fogadja el a KKSZB és csak a KKSZB tanúsítványát** - úgy a hamisított gw.kkszb.gov.hu oldalra küldi el az autentikációs tokenjét a kérésével együtt, amelyet tokent így a támadó megszerez és felhasználhat a nevében más lekérdezés indításához. Mivel a KKSZB név alapon kezeli a routingot, ezért a támadó - mivel eltérítette a DNS-t -, nem tudja azonnal ezt az információt felhasználni, csak akkor, ha visszaállította az eredeti DNS működést és az ismét jó helyre mutat. Amennyiben a KKSZB IP címen lenne elérhető, úgy valós időben, azonnal tudna lekérdezéseket indítani a támadó.
- **Hibás konfiguráció a Kliens alkalmazás oldalon:** Ha például az adatküldő fél rossz konfiguráció miatt nem az éles környezetet szólítja meg, hanem a teszt KKSZB-t, akkor az éles autentikáció tokenjét kompromittálja, **amennyiben a csatlakozó kliens alkalmazás nem fogadja el a KKSZB és csak a KKSZB tanúsítványát.** Még rosszabb esetben, ha a konfigurációban a teszt autentikációs tokent használja a teszt URL eléréssel, akkor például a teszt rendszerbe akár adatot is rögzíthet, vagy a teszt rendszerből adatot kérdezhet le (összecseréli a teszt és éles környezeteket). **Ez az eset a szolgáltató oldalán is előfordulhat:** a KKSZB éles rendszerre a teszt környezetét köti rá véletlen, ebben az esetben csak a megfelelő tanúsítvány menedzsment képes ezt megakadályozni.
- **Nem ellenőrzött KKSZB kliens tanúsítvány a szolgáltatás oldalon:** A KKSZB irányából a belső támadó HTTPS kérést indít egy szolgáltatás felé, a KKSZB szerverről indítható akár egy *curl* alkalmazással is HTTPS kérés. Amennyiben a támadó nem fér hozzá a KKSZB kliens

tanúsítványához - nem rendelkezik *root* joggal - úgy nem tud érvényes kérést indítani még a szerverről sem, amennyiben a szolgáltatás végez tanúsítvány ellenőrzést. Ha a szolgáltatás nem végez ellenőrzést, akkor a belső támadó le tudja kérdezni a szolgáltatást