# **Requirements fázis**

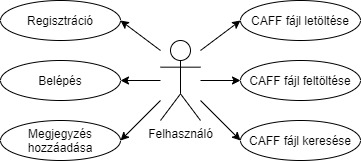
## **Funkcionális követelmények**

· A felhasználóknak kell tudni regisztrálni és belépni

· A felhasználóknak kell tudni CAFF fájlt feltölteni, letölteni, keresni

· A felhasználóknak kell tudni CAFF fájlhoz megjegyzést hozzáfűzni

A rendszer legfontosabb felhasználási szcenárióit az 1. ábra foglalja magában.



1. ábra A rendszer felhasználási szcenáriói

## **Szerver oldali funkciókövetelmények**

· A szerver képes legyen feldolgozni a CAFF fájlformátumot

· A szoftver C/C++ nyelven legyen implementálva

· A webshop megjelenítésénél generálni kell CAFF fájlból egy előnézetet

## **Kliens oldali követelmények**

· A szoftvernek működnie kell webes felületen

# **Biztonsági követelmények és célok**

A 2. ábrán látható a rendszer és annak környezete. Ezt a funkcionális követelmények ismeretében tudjuk felvázolni. A webshop rendszert regisztrált felhasználók használják, velük kerülhet interakcióba a rendszer. A tőlük érkező kéréseket nem tudjuk kontrollálni, ezért az interakció bizalmi kérdéseket vet fel. A 2. ábrán ezt a szaggatott piros vonal jelzi. A rendszer működéséhez szükség van a felhasználók adatainak tárolására, valamint tárolni kell a feltöltött fájlokat és a hozzá tartozó megjegyzéseket is.

A webshop rendszer biztonsági követelményeit hat kategóriába osztjuk: bizalmasság, integritás, elérhetőség, autentikáció, autorizáció, auditálás. A kategóriákhoz a következő követelményeket fogalmazhatjuk meg.

· Bizalmasság

o A regisztrált felhasználók adatait csak ők maguk láthatják.

o A felhasználók láthatják az összes feltöltött státuszú elemet.

o A felhasználók csak a saját letöltött elemeiket tekinthetik meg.

· Integritás

o A felhasználók csak a saját személyes adataikat módosíthatják.

o Csak regisztrált felhasználók tölthetnek fel és le elemet.

o A felhasznált regisztráló csak a saját eleméhez fűzhet megjegyzést és szerkesztheti azt.

· Elérhetőség

o Bármikor lehet regisztrálni felhasználót.

o Elemet fel-, letölteni bármikor lehet.

o Megjegyzést feltöltés közben kell megadni és utána bármikor szerkeszthetőnek kell lennie.

· Autentikáció

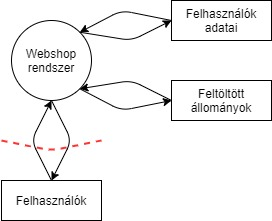
o A webshopot elérni kívánók csak regisztráció és bejelentkezés után használhatják a rendszert.

· Autorizáció

o Elemek feltöltése és a megjegyzés szerkesztése egy adott személyhez tartozó tevékenység lehet.

· Auditálás

o A felhasználók tevékenységét naplózni kell.



2. ábra A webshop rendszer és környezete

# **Threat assessment**

A threat assessment két részből áll össze: az assetek azonosítása és az azokat veszélyeztető működések (threat) azonosítása.

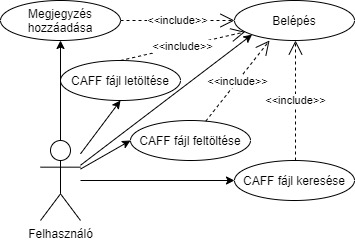
## **Assetek megállapítása**

A Threat, Risk, Vulnerability Analysis során három különböző típusú assetet különítünk el:

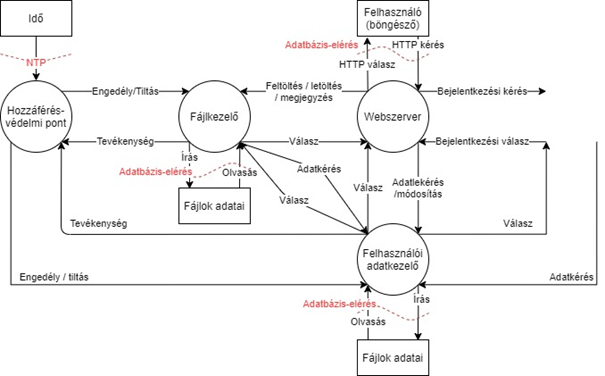
· fizikai assetek

· emberi assetek

· logikai assetek



3. ábra Felhasználókhoz köthető use casek



## **Támadó modell kidolgozása**

Megszemélyesítéssel kapcsolatos veszélyek forrása lehet bármelyik külső szereplő, akivel interakcióba lépünk.

Néhány lehetséges támadás:

* Egy vásárló megpróbál egy másik vásárló személyes adataihoz hozzáférni, pedig ezt csak adminisztrátorok tehetnék meg.
* Egy vásárló megpróbálja átírni a termékek árát ingyen vásárlás céljából, miközben árat csak adminisztrátorok írhatják át.

# **Szükséges biztonsági funkcionalitások**

A biztonsági követelmények kielégítéséhez többféle biztonsági funkcionalitást kell megterveznünk, implementálnunk és tesztelnünk. A szükséges biztonsági funkcionalitások listáját a biztonsági követelmények és abues case-ek elemzésével kaphatjuk meg.

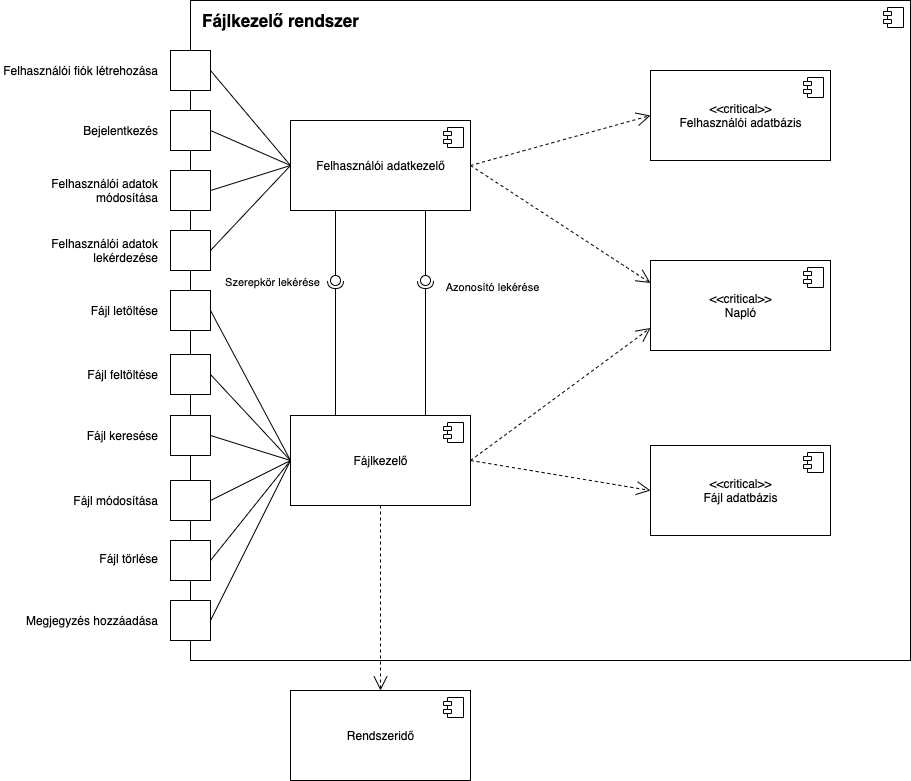
Először is, a webshop használatához az autentikációt kell megvalósítanunk. A tervezett webshoppal böngészőn keresztül léphetnek interakcióba a felhasználók, ezért jelszó alapú autentikációt választottunk.

A felhasználó lehet vásárló vagy adminisztrátor. A felhasználó tud CAFF fájlt letölteni, keresni, belépni és megjegyzést hozzáfűzni. Az adminisztrátor ezen felül CAFF fájlokat tud törölni és módosítani. Ehhez szerep alapú autorizációs mechanizmust kell implementálnunk.

A felhasználók személyes adatait és az autentikációhoz szükséges jelszót védenünk kell szivárgás és illetéktelen hozzáférés ellen. Azért a személyes adatokat titkosítanunk kell tárolás és átvitel során, a jelszavakat pedig biztonságosan kell tárolnunk. A jelszóhoz policy-t dolgozunk ki (legalább 8 karakter, kis- és nagybetűt kell tartalmaznia).

Architektúra tervek

A követelmények ismeretében két oldalról közelítjük meg az architektúrális tervezést. Első körben a rendszer struktúráját **komponensdiagramon** szemléltetjük a különböző biztonsági követelményekkel együttvéve *UMLsec* segítségével. Ezt követően a rendszer működését **szekvencia diagramokon** keresztül mutatjuk be a különböző szerepkörök szemszögéből.



A rendszer struktúrája

A rendszer több balkomponensből épül fel és nyolc interfészt nyújt a felhasználók számára:

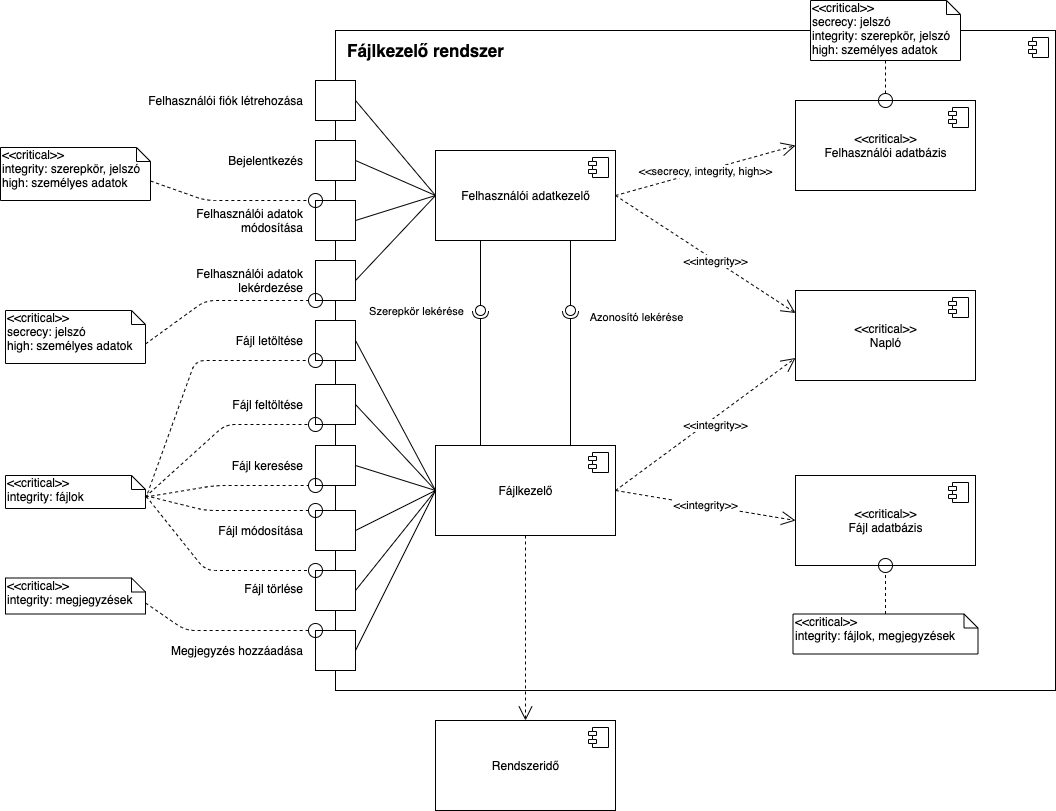
1. **Felhasználói fiók létrehozása:** Ezen az interfészen keresztül tudják a felhasználók létrehozni a saját felhasználó fiókjukat.
2. **Bejelentkezés:** Egy email cím, jelszó párossal ezen az interfészen keresztül tudnak bejelentkezni a felhasználók a rendszerbe.
3. **Felhasználói adatok módosítása:** A felhasználók tudják módosítani a bejelentkezési adataikat és jelszavukat ezen az interfészen keresztül.
4. **Felhasználói adatok lekérdezése:** Ennek az interfésznek a segítségével tudják lekérdezni a saját adataikat a felhasználók.
5. **Fájl letöltése:** A fájlok megvásárlásához és letöltéséhez ez az interfész nyújt szolgáltatásokat a felhasználók számára.
6. **Fájl feltöltése:** A különböző fájlok feltöltéséhez ezt az interfészt tudják használni a *user*ek.
7. **Fájl keresése:** A fájlok közötti kereséshez ez az interfész nyújt szolgáltatásokat.
8. **Fájl módosítása:** Az admin jogosultságú felhasználóknak lehetősége van módosítani az egyes fájlokat.
9. **Fájl törlése:** Az admin jogosultságú felhasználóknak lehetősége törölni az egyes fájlokat.
10. **Megjegyzés hozzáadása:** A felhasználóknak lehetőségük van megjegyzéseket fűzni az egyes fájlokhoz. Ehhez ezt az interfészt tudják használni.

A személyes adatokat a *felhasználói adatbázis*ban fogjuk tárolni, amiknek a lekéréséhez egy különálló, *felhasználói adatkezelő* komponenst használunk. A különböző fájlok tárolása hasonló módon egy *fájl adatbázis*ban történik majd, amiknek a kezeléséért egy *fájlkezelő* komponens lesz a felelős.

A rendszerrel elvárt követelmény, hogy a felhasználó tudjon fájlt letölteni, feltölteni, illetve keresni, ezért ISP objektunorientál elvnek megfelelően külön interfészeket fogunk készíteni az egyes use-casekhez.

A *fájlkezelő*nek a feladata még az egyes szerepkörökhöz tartozó tevékenységek felülvizsgálata és ezek alapján történő engedélyezése vagy tiltása. Az admin jogosultságú felhasználók kezelhetik az oldalt úgy, hogy módosíthatják és törölhetik az egyes fájlokat.

Mind a felhasználói, mind a fájlkezelő komponens biztonsági okokból naplózni fogja az egyes eseményeit.



A ..-es diagramon láthatóak a biztonsági követelmények, amit UMLsec segítségével ábrázoltunk. A rendszer szempontjából biztonságkritikus komponenseknél látható a *<<critical>>* sztereotípia. Az adatbázisoknál j