

5/13/2025

Live Streaming Service

Hallgató: Buga Péter (G50RDF)

Konzulensek:

Dr. Maliosz Markosz

Dr. Simon Csaba

5/13/2025



Bevezetés



Feladat

A egy élő videó közvetítő (streaming) szolgáltatás fejlesztése konténerizált mikroszolgáltatás-architektúrában, Kubernetes környezetben. A hallgatónak ki kell alakítania a rendszer fő komponenseit (pl. stream kezelés, felhasználókezelés, nézőszámlálás), és biztosítania kell azok együttműködését konténerekben.



Streaming Szolgáltatások Jelentősége

- Mindennapos használat
- Szórakoztatás, Oktatás, Ipar
- Felhasználók milliói
- Internet forgalma



Informatikai Háttér

Komplex Rendszerek

Hullámzó erőforrásigény

Felhőalapú megoldás

Kubernetes



5/13/2025

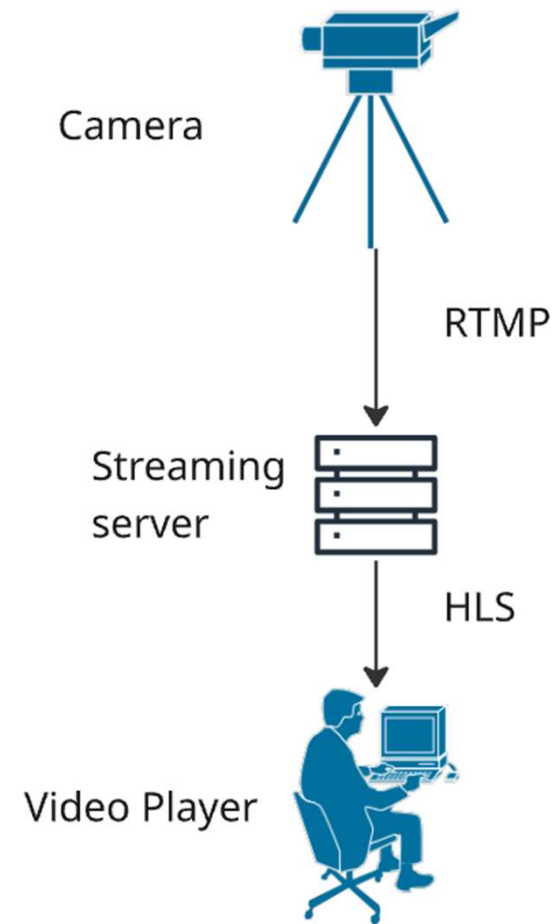


Elméleti Háttér



Közvetítő protokollok

- RTMP
 - + Tartalom feltöltése
- HLS
 - + Tartalom letöltés
 - + Szegmensek
 - + HTTP



Microservice architektúra és kubernetes

- Microservice architektúra
 - + Lazán kapcsolt modulok
 - + Feladat szeparáció
 - + Rugalmas
- Kubernetes
 - + Skálázhatóság
 - + Ingress
 - + Könnyű telepíthetőség



5/13/2025



Architektúra és Implementáció



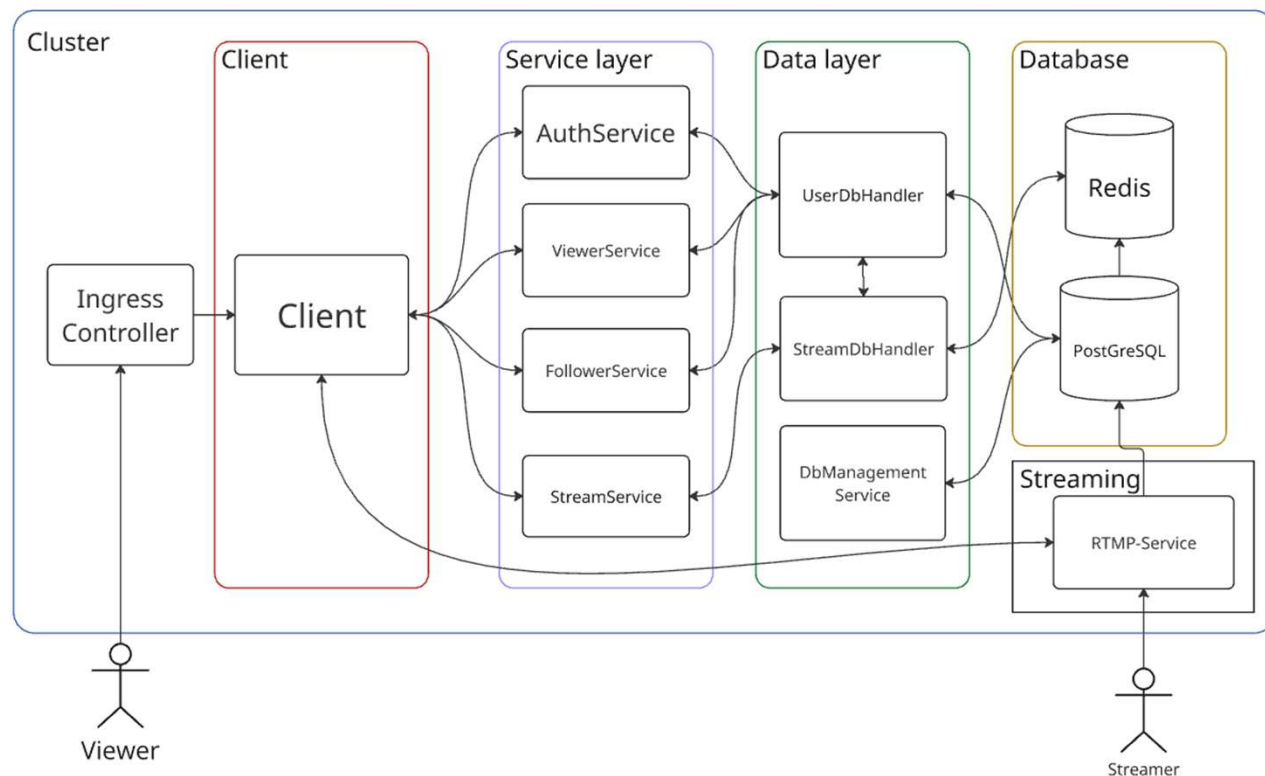
Tervezett funkciók

- Felhasználói regisztráció és bejelentkezés (JWT alapú hitelesítés)
- Élő stream indítása egyedi stream kulccsal
- Streamek böngészése, nézése valós idejű nézőszámlálással
- Stream ajánlások (népszerűség, követések alapján)
- Követés funkció (streamerek követése, élő státusz megjelenítése)



Architektúra

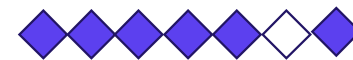
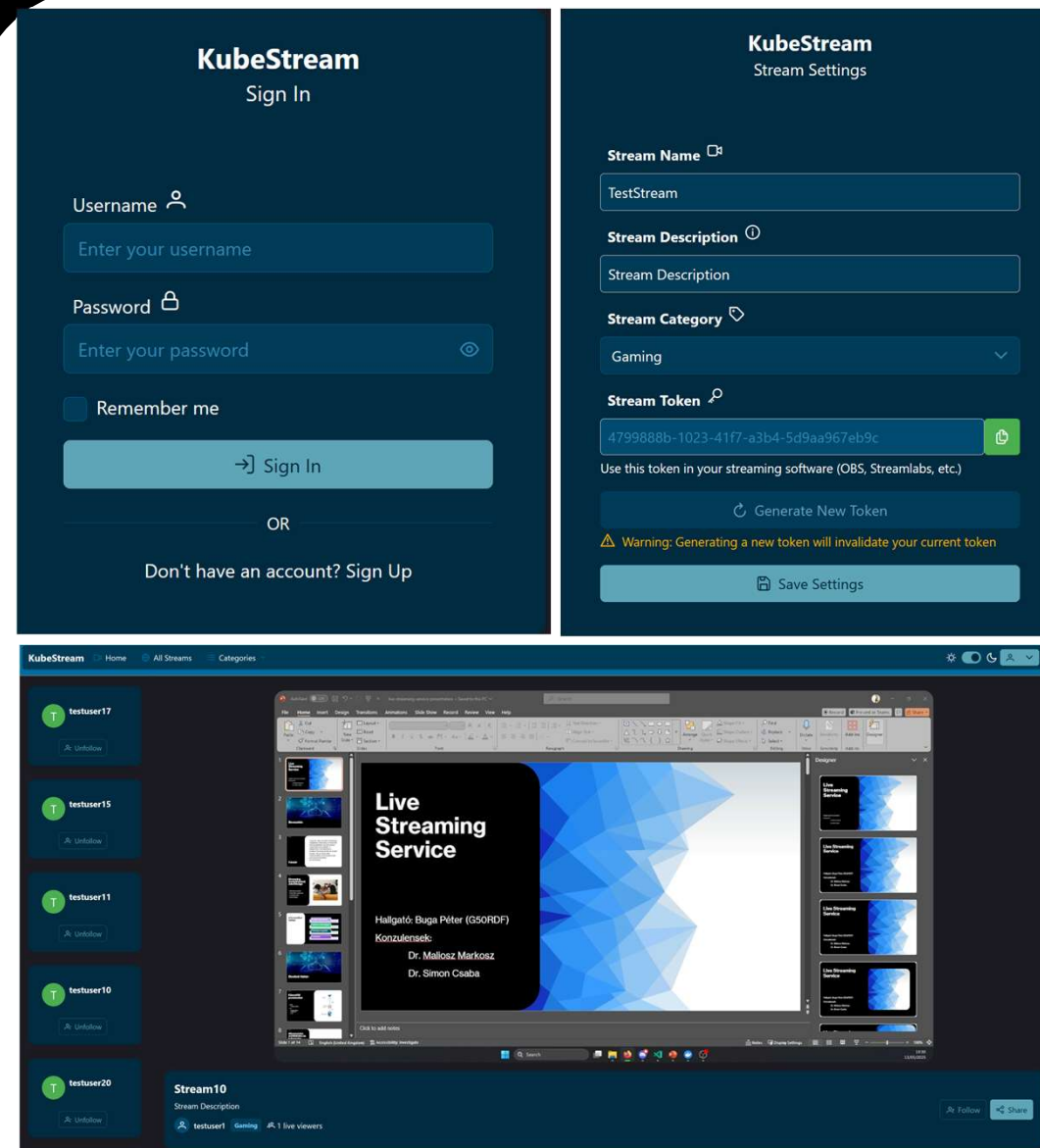
- Mikroszolgáltatás-alapú backend (különálló szolgáltatások)
- REST API hívások komponensek között
- Streaming szerver:
 - RTMP fogadás,
 - HLS továbbítás
- PostgreSQL (tartós adatok), Redis (gyorsítótár)
- Ingress controller a külső forgalom irányítására
- Angular alapú webalkalmazás, HLS lejátszás
- Skálázható, bővíthető, biztonságos rendszerfelépítés



Streaming működése

Stream Indítása:

1. Regisztrálás majd belépés
2. Stream kulcs generálása
3. Stream indítás



Nehézségek

- Mikroszolgáltatások közötti adat-szinkronizáció (pl. nézőszám frissítése)
- Streaming szerver (Nginx-RTMP) és backend szolgáltatások integrációja (webhookok, API hívások)
- Hibakezelés a szolgáltatások közötti kommunikációban (pl. elérhetetlen komponensek)
- JWT tokenek érvényességének egységes ellenőrzése minden szolgáltatásban (közös token handler fejlesztése)
- Gyors adatfrissítés megvalósítása Redis cache segítségével
- Fejlesztői környezet (Docker, Kubernetes, VS Code) helyes beállítása és használata



Összefoglalva

- Modern, mikroszolgáltatás-alapú streaming platform fejlesztése Kubernetes környezetben
- Fő funkciók: élő közvetítés, nézőszámlálás, felhasználó- és követéskezelés, ajánlórendszer
- Konténerizált architektúra: könnyű bővíthetőség, skálázhatóság, megbízható működés
- Valós idejű adatok kezelése Redis cache segítségével
- Biztonságos hozzáférés JWT token alapú hitelesítéssel
- Gyakorlati tapasztalat szerzése felhőalapú technológiákban és konténer-orkesztrációban
- A rendszer további funkciókkal (pl. chat, értesítések) bővíthető

