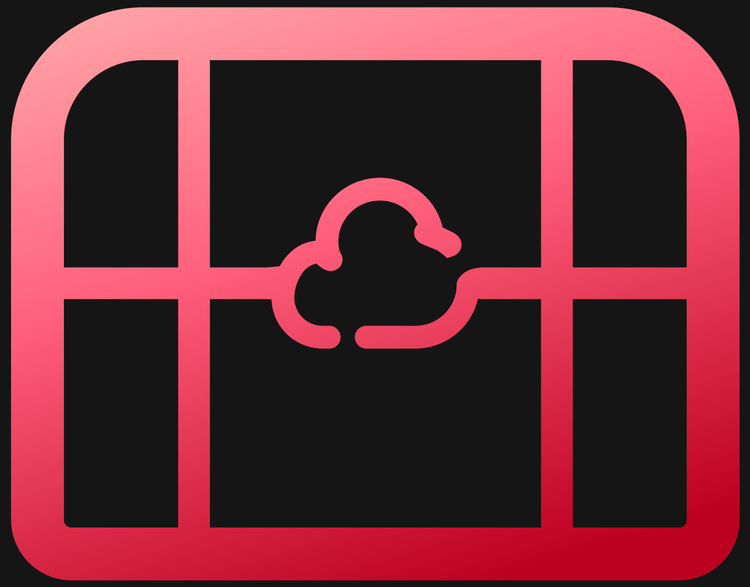
SeharaCloud

privatni cloud streaming servis

Analiza backend dijela projekta



Predstavljanje tima:

Tašić Tajra

UI / UX designer

TajraT

Tajra Tašić

tajra.tasic.24@size.ba

Kolasević Samer

Backend developer

SamerKolasevic29

Samer Kolasević

samer.kolasevic.24@size.ba

Kolasević Amel

Mobile developer

amell555k

Amel Kolasević

amel.kolasevic.24@size.ba

0. korak:

Telefon u džep.
Kada se radi, radi se.
Rasporedi vrijeme.

Napredak? Neminovan.

Analiza zahtjeva – stvari na papiru

Metoda razvoja sistema

Agilni pristup – iterativni sprintovi (skateboard → bicikl → motorcikl → automobil ...)

Funkcionalni zahtjevi

- **Streaming videa, muzike, slika i dokumenata, te prikladan prikaz istih**
- **Pristup sa bilo koje lokacije** (Cloudflare tunnel)
- **Pretraga i filtriranje po tipu, žanru...**

Nefunkcionalni zahtjevi

- **Latencija < 200ms za metadata upite**
- **Podrška za HTTP Range request** (streaming bez punog download-a)
- **Izolacija pristupa (clouduser, UFW)**

Open Source alternative?

Postoje, naravno. Međutim...

Jellyfin

odlično rješenje ali gotov produkt. Cilj projekta je inženjersko učenje, ne deployment tuđeg koda.

Nextcloud

PHP monolith, preteška infrastruktura, za naš slučaj upotrebe nema fine kontrole nad streaming logikom

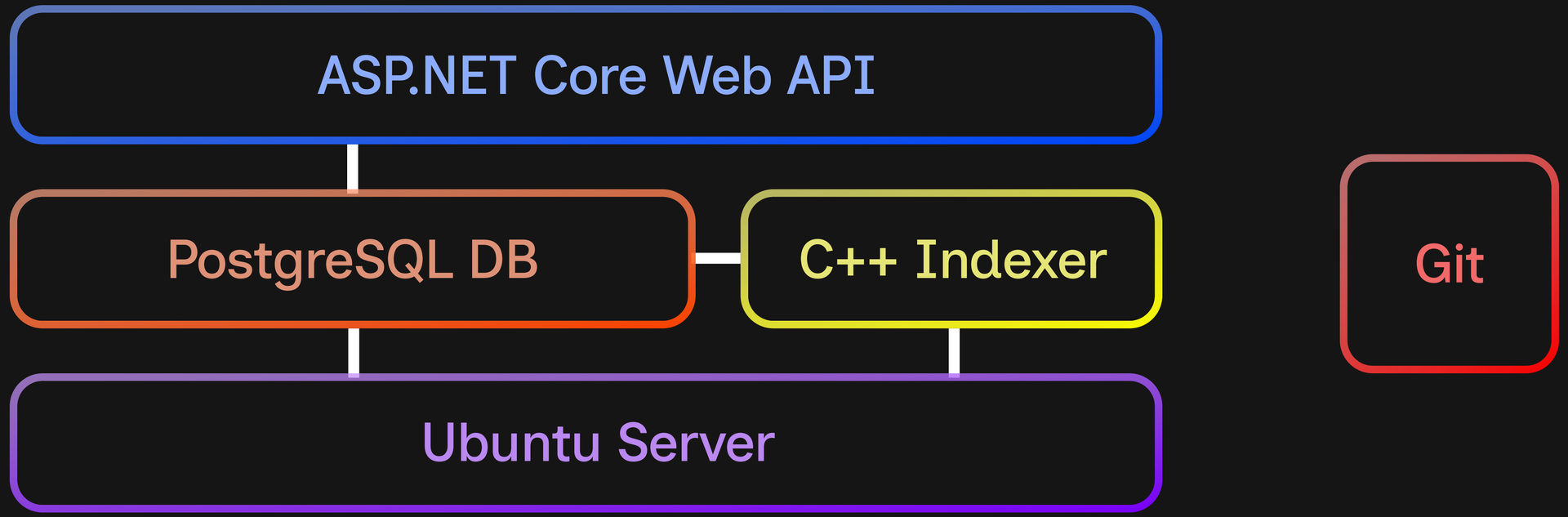
Vastiti razvoj daje:

- ✓ Potpunu kontrolu nad arhitekturom
- ✓ Prilagođenje tačno našim formatima i workflow-u
- ✓ Hands-on iskustvo sa svakim slojem sistema
- ✓ Portfolio projekat koji možemo objasniti svaki red koda

Uvod - pogled na problem:

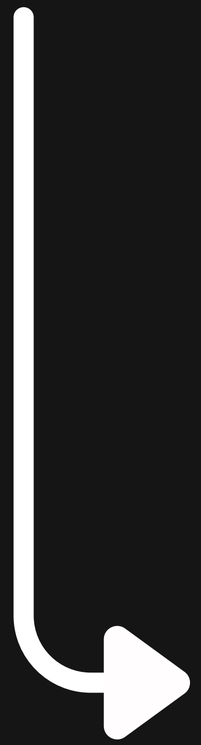
Ili stari laptop, struja i znanje...

Arhitektura backend dijela:



Primjer toka jednog zahtjeva:

GET /api/music/artists

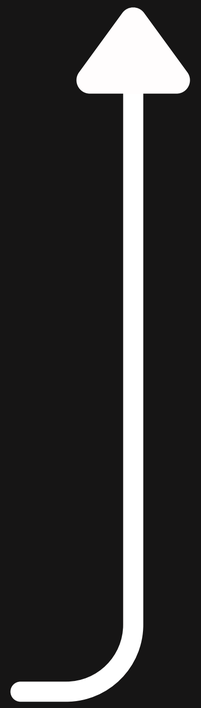


MusicController.GetArtists()

MusicService - podaci i poslovna logika

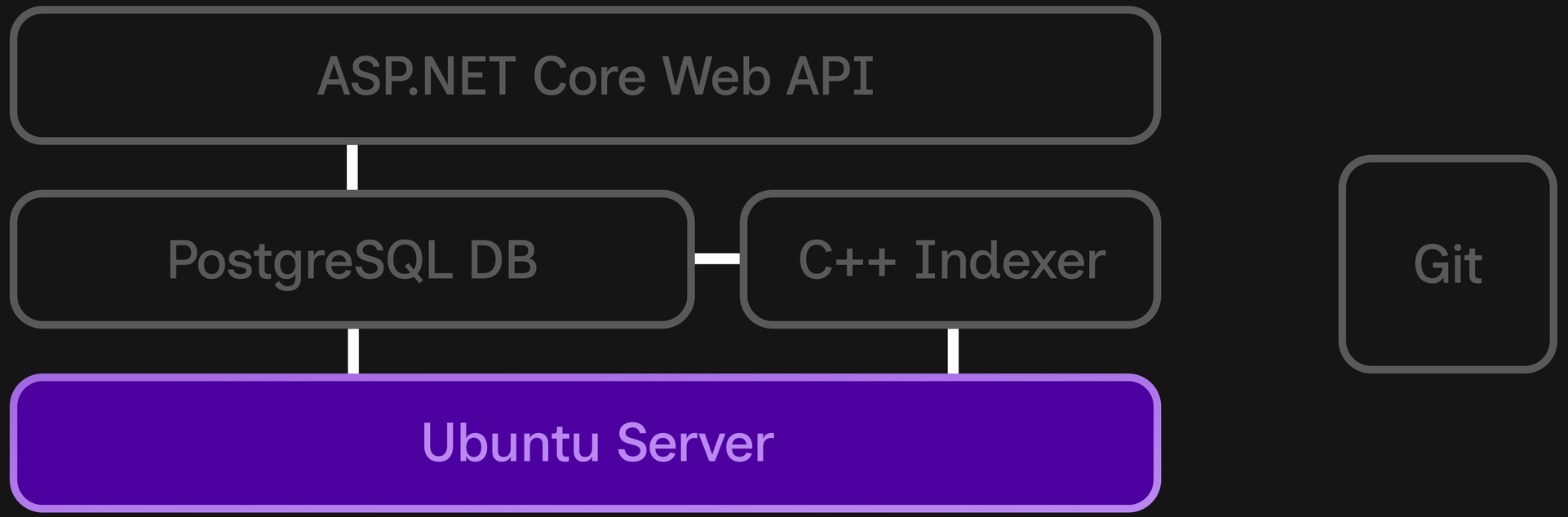
MusicRepository - SQL upit

Vraća redove iz tabela



Grid prikaz izvođača

Jedno po jedno



Ubuntu kao server OS?

Definitivno! A Zašto?

- Hardverski prelagan
- Besplatan
- Industrijski standard
- SSH pristup nativan

Daljnje postavljanje servera

ext4 particija 100GB

/mnt/cloud/ struktura

clouduser (izolovan)

UFW firewall

Port 5000 otvoren

PostgreSQL 16

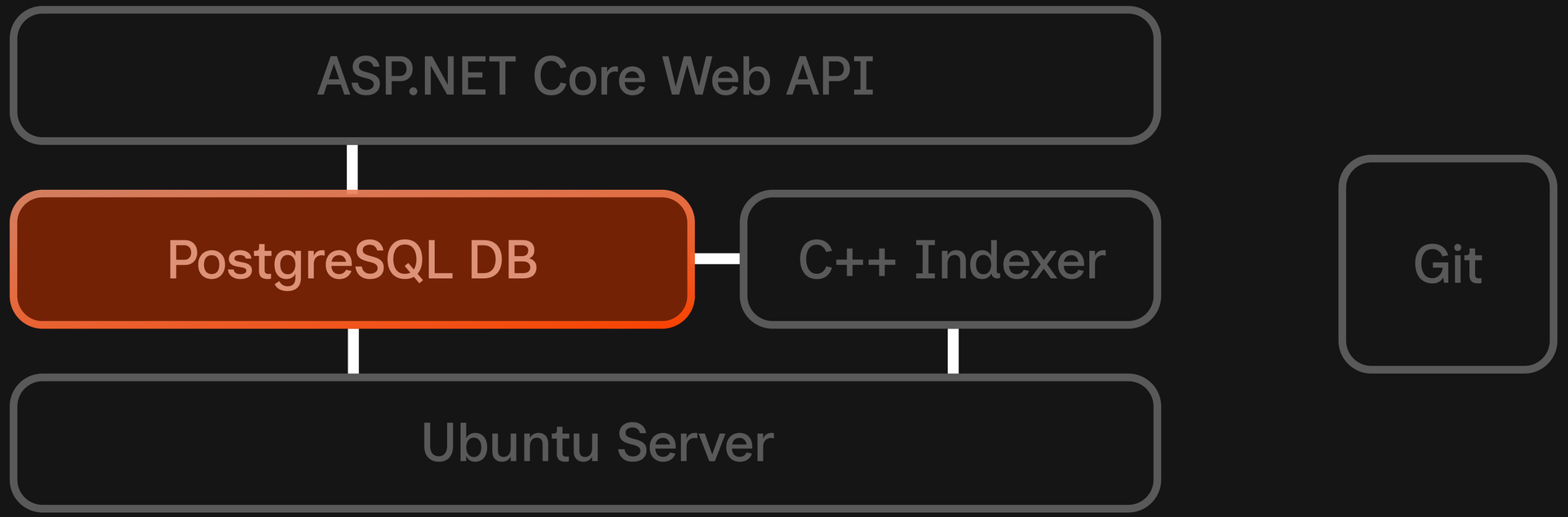
.NET 9 runtime

SHH remote kontrola

Šta bi bilo bolje?

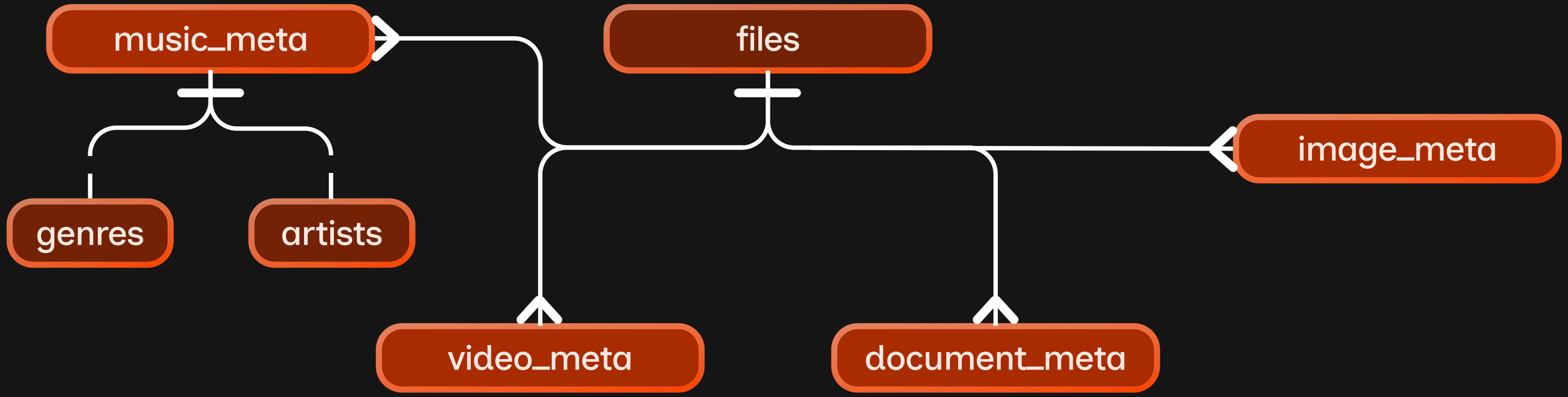
Mini PC (Thinkcentre, Dell Optiplex, HP Elitedesk) umjesto fat PC-a na kojem je SeharaCloud.
Mali potrošači struje, tiši, velika toleracija na 24/7 dostupnost, SSD umjesto HDD-a (10x brži)...

Idemo dalje


ASP.NET Core Web API
PostgreSQL DB
C++ Indexer
Git
Ubuntu Server

Kreiranje sheme baze

- Jednostavna, lako proširiva shema koja ima sve dovoljno za UI prikaz. Posjeduje osnovne informacije, metapodatke i putanje do thumbnail-ova. Također, indeksiranje određenih kolona daje na velikoj brzini dohvata odgovarajućih redova



Primjer SQL upita

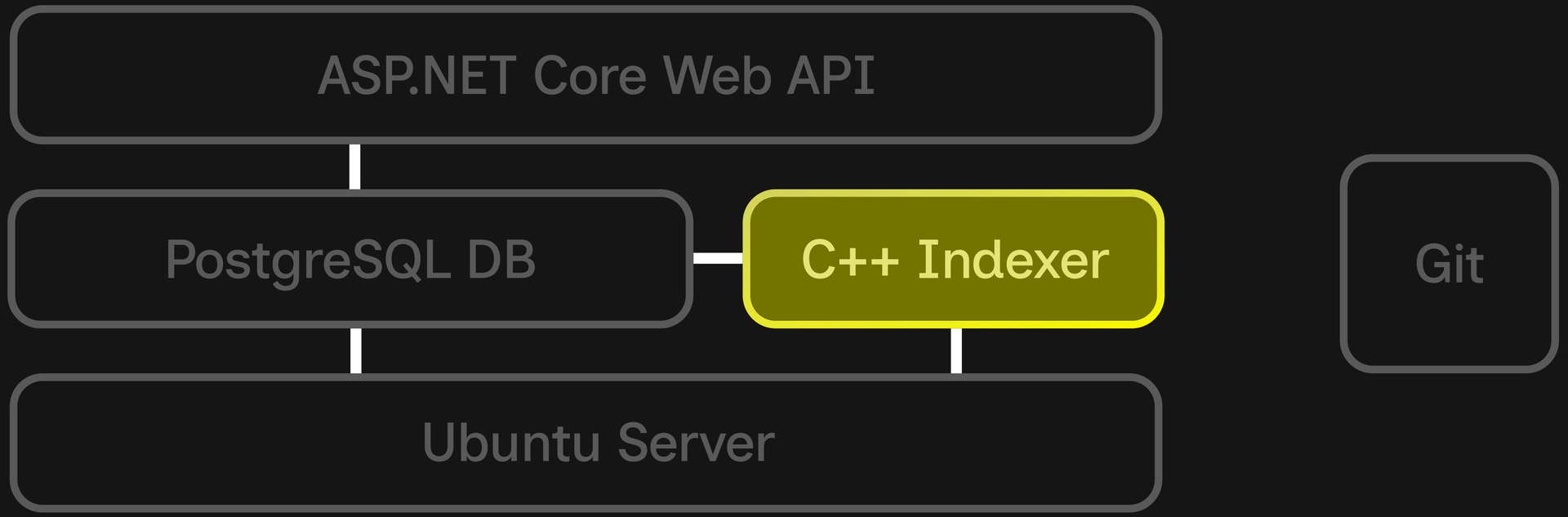
```
SELECT
    f.id,
    f.thumbnail_path,
    m.title,
    a.name      AS artist,
    g.name      AS genre,
    m.duration_sec
FROM files f
JOIN music_meta m ON f.id = m.file_id
LEFT JOIN artists a ON m.artist_id = a.id
LEFT JOIN genres  g ON m.genre_id  = g.id
ORDER BY m.title;
```

Prikaži mi ID, putanju thumbnail-a, naziv pjesme, ime izvođaća, naziv žanra i dužinu u sekundama
iz tabele files
naziv, dužina u sek. iz tabele music_meta preko tabele files
ime izvođača iz tabele artists preko tabele music_meta
naziv žanra iz tabele genres preko tabele music_meta

Šta bi bilo bolje?

Na ovom nivou iskustva i više nego dovoljno, ali postoji šansa sa boljom preraspodjelom baze podataka skupa sa većim ograničenjima ili mogućnostima.

Sljedeći na redu...



Jedan korak automatizacije

Ti kopiraš nova_pjesma.mp3 na server

Linux kernel (inotify event), indekser SAMO sluša

Indekser se budi

- TagLib čita ID3 tag (artist, duration..)
- SHA256 hash (duplikat?)
- INSERT u PostgreSQL

```
// TagLib - 5 linija da dobiješ sve što trebaš
TagLib::FileRef f("/mnt/cloud/music/smoke.mp3");

std::string title  = f.tag()->title().to8Bit(true);
std::string artist = f.tag()->artist().to8Bit(true);
int duration = f.audioProperties()->lengthInSeconds();
// → "Brate Čuvaj Se", "Smoke Mardeljano", 165
```

A što baš C++?

Python

RAM: ~45MB
Start: ~800ms
CPU idle: 2%

VS

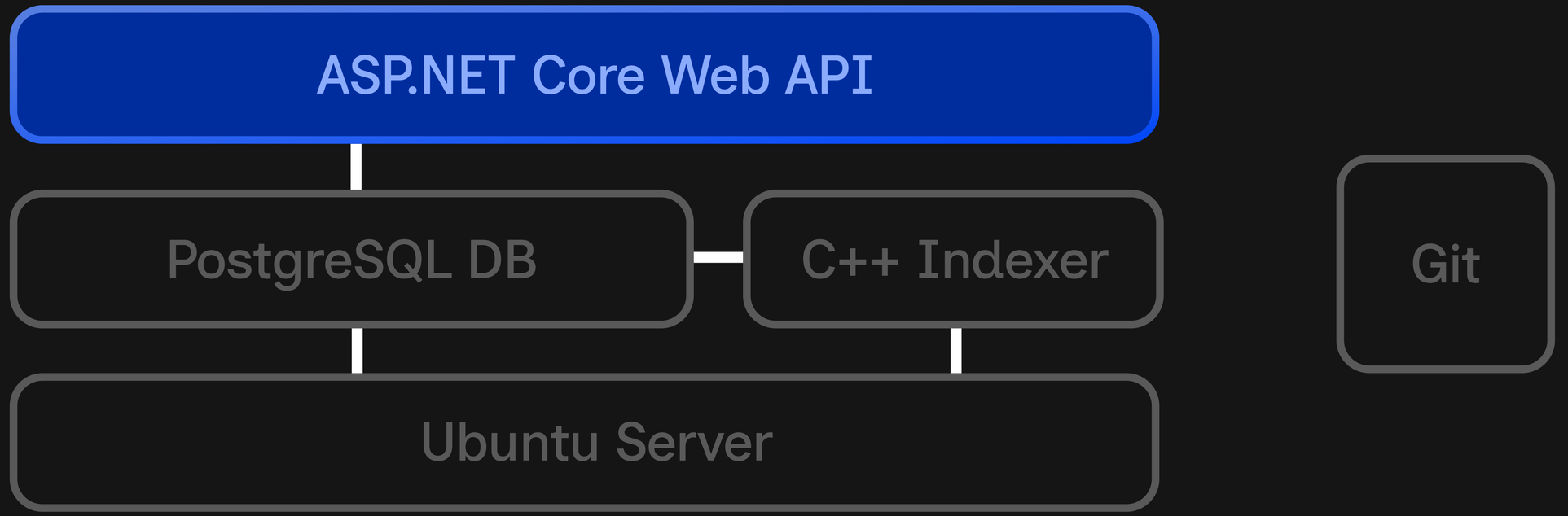
C++

RAM: ~3MB
Start: ~20ms
CPU idle: 0%

Šta bi bilo bolje?

Inkonzistentnost metapodataka unutar fajlova raznih izvora (YouTube, mp3 Converteri, Fizičke verzije itd.). Uvesti Python skriptu za formatiranje imena fajla → ručna provjera ...

Nakon indeksera...



Zašto višeslojno? Kad pukne, znaš gdje je

HTTP Request: GET /api/music/artists

GET-only API, za sada dovoljno!

Sprint 1:

Zaokružena ideja streaming platforme koja nudi podržku van lokalne mreže, razvijeno iskustvo u fazi razvoja i planiranja, kao i timskog rada!

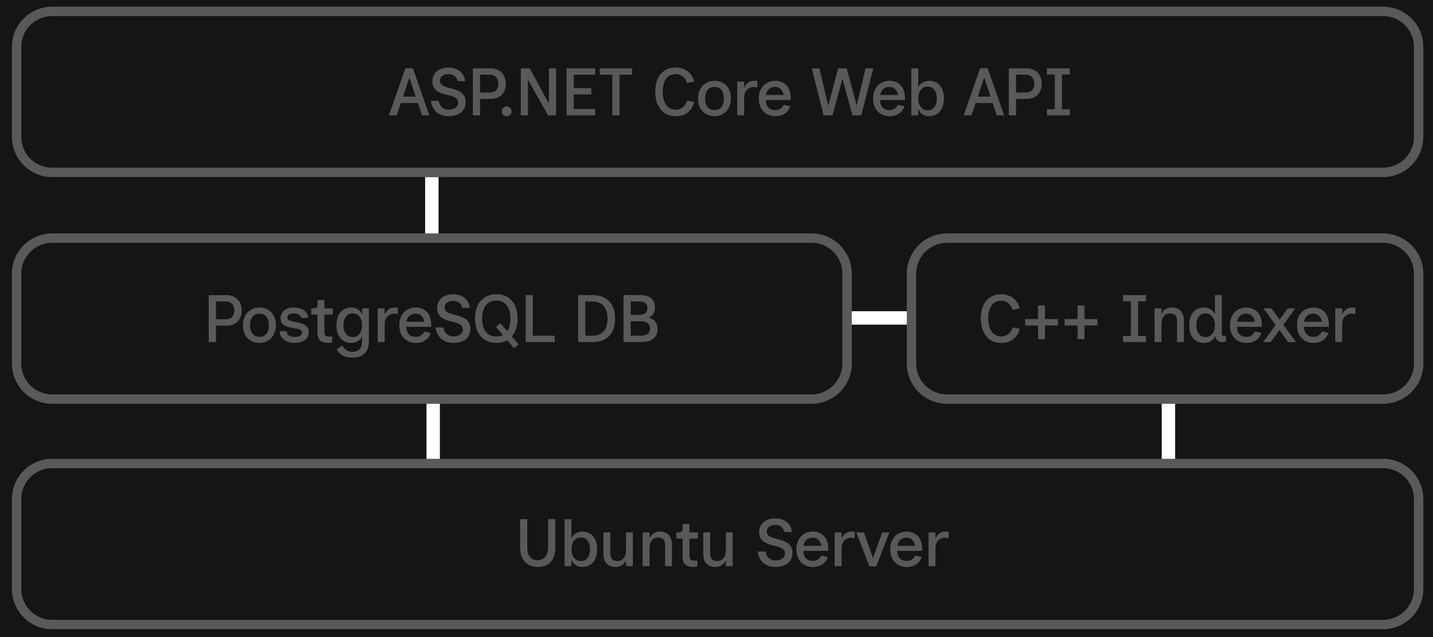
Ostali sprintovi:

Profilizacija, kreiranje liste omiljenog sadržaja, auto backup...

Šta bi bilo bolje?

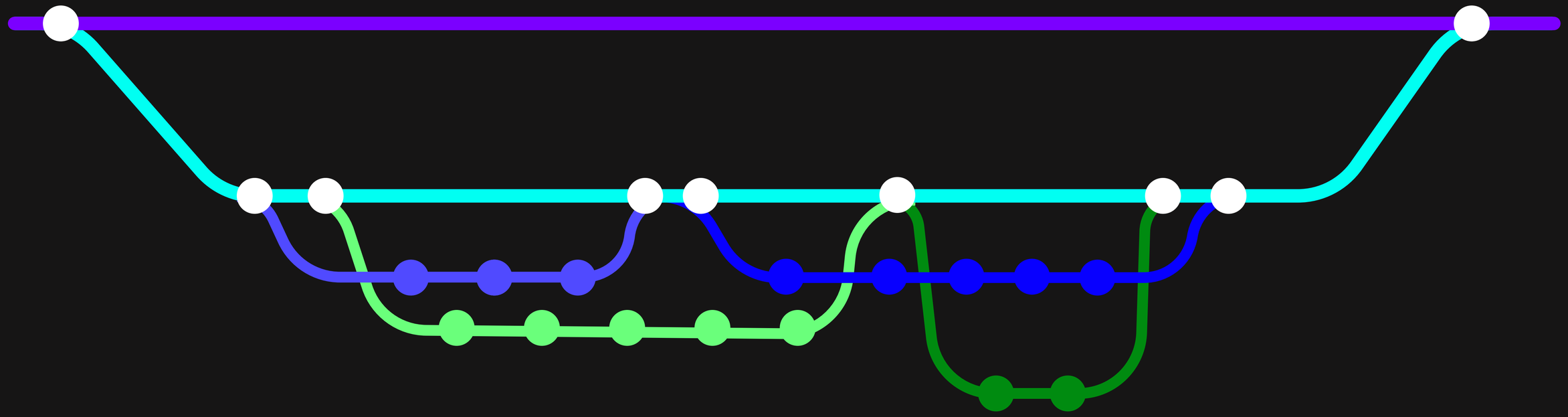
Nije razložen dovoljan broj metoda na Repository sloju kada je u pitanju ThumbnailRepository, korištenje jedne metode sa UNION ALL upitom je neefikasan, viđeno u toku razvijanja..

Last but not least...

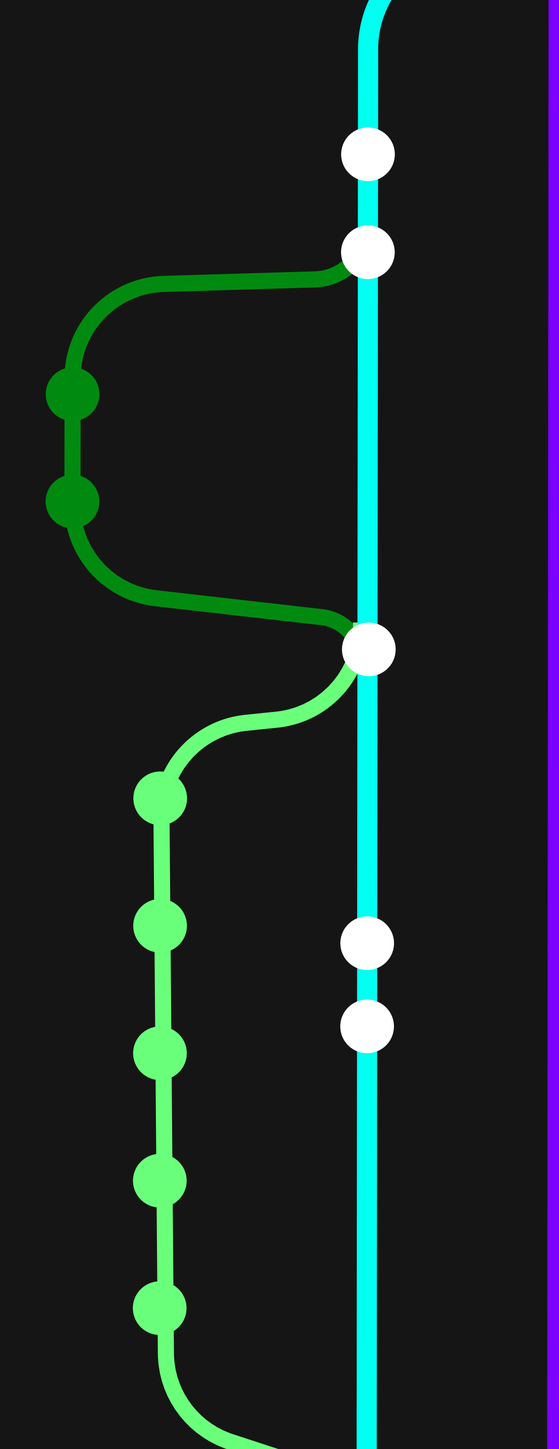


Git

Danas sutra, biće nas više na projektu



Main Dev Frontend Backend



```
# Jutro - uvijek
git checkout dev
git pull origin dev

# Nova funkcionalnost
git checkout -b feature/naziv

# Tokom rada
git add .
git commit -m "feat: opis što si napravio"
git push origin feature/naziv


# Završio → Pull Request na GitHubu
# feature/naziv → dev → review → merge
```