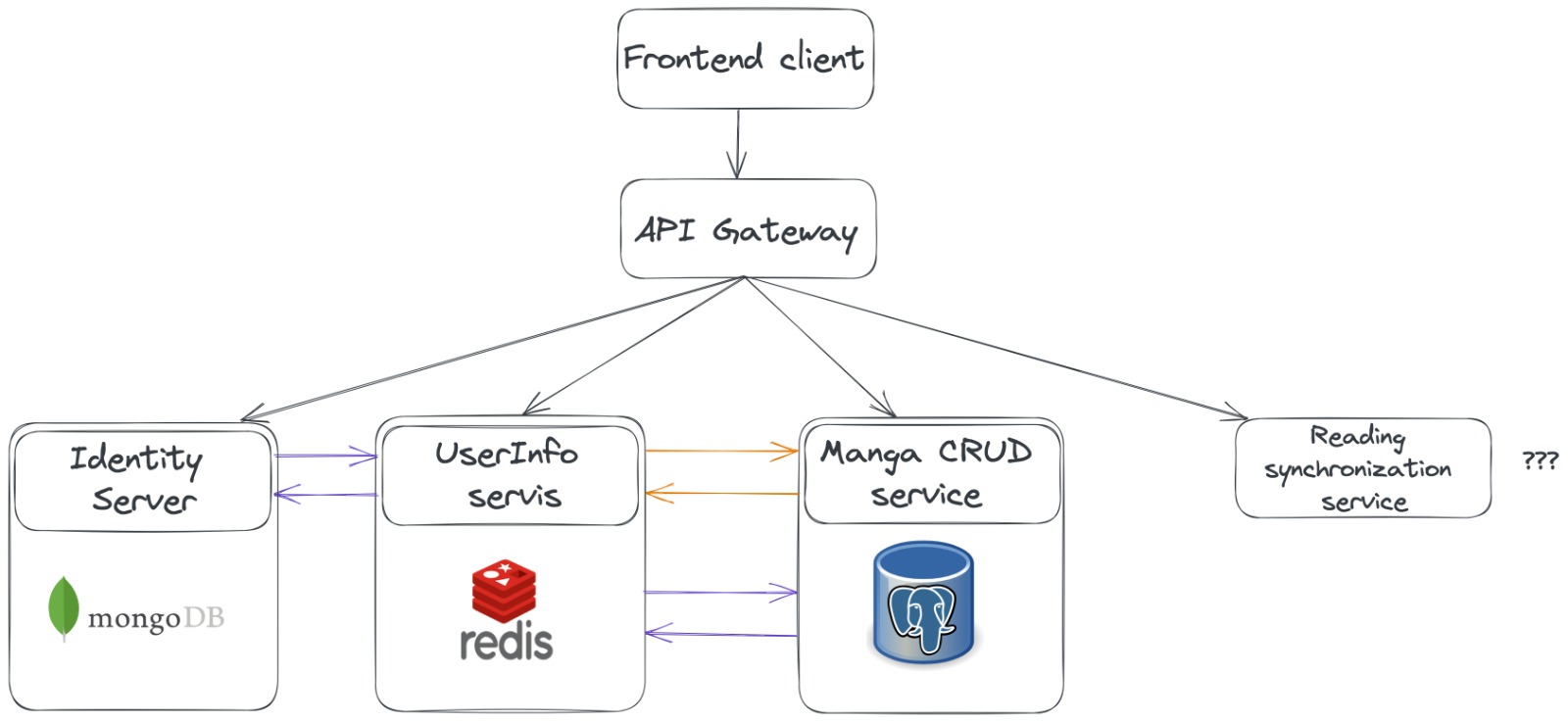
**Mikroservisna arhitektura**

Početna ideja arhitekture mikroservise aplikacije sa kojim se kreće u razvoj, prikazana je na slici ispod. Moguće je da će doći do odredjenih izmena tokom rada.



Aplikacija se sastoji od 5 mikroservisa, a to su API Gateway servis, IdentityServer servis, UserInfo servis, Manga CRUD servis i Reading synchronization servis.

Mikroservis IdentityServer pruža usluge autorizacije i autentikacije korisnika. Autentikacija se zasniva na JWT tokenima (access token, refresh token). Takodje, ovaj servis upravlja korisničkim nalogom, tj. omogućava promenu podataka u korisničkom nalogu. Koristi Mongo bazu podataka.

UserInfo servis je servis koji ima više funkcionalnosti. Kako samo ime kaže, ovaj servis obezbedjuje razne informacije o korisniku. Za konkretnog korisnika ovaj servis čuva informacije o svim mangama koje ikada čitao, koja je njegova lista želja za čitanje (tj. korisnik može da napravi listu koje mange želi sledeće da čita, kao podsetnik), kao i informaciju koja je poslednja manga koju je korisnik čitao. Na osnovu tih informacija ovaj servis daje preporuke mangi korisniku na osnovu različitih kriterijuma koje korisnik može da unese, na osnovu autora čije je mange čitao, na osnovu žanra koji je već čitao i slično. Koristi se Redis baza podataka.

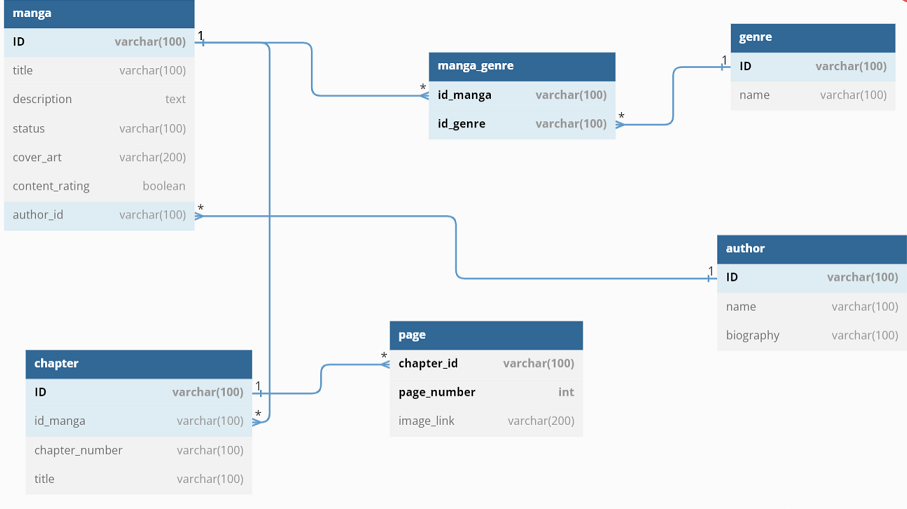
Komunicira sa IdentityServer servisom, kada se kreira novi korisnik, IdentityServis posalje informaciju UserInfo servisu o tom dogadjaju, pa se i u UserInfo servisu kreira novi objekat za tog korisnika sa praznim svim informacijama koje ovaj servis čuva, koje će se tokom vremena kako korisnik čita razne mange biti dopunjavan. Opisana komunikacija je asinhrona.

Manga CRUD servis je servis koji skladišti same mange I omogućava manipulaciju njima. Omogućava dodavnje, brisanje, izmenu i čitaje mangi. Koristi PostgreSQL bazu podataka. Komunicira sa UserInfo servisom na dva načina, sinhrono i asinhrono. Asinhrono komunicra sa USerInfo servisom kada mu šalje informacije o radnjama korisnika, u smislu, kada korisnik pročita neku mangu, on asinhrono pošalje UserInfo servisu informaciju o tome da bi UserInfo servis zabeležio to i dodao tu mangu u spisak ikad čitanih mangi tok korisnika i slične akcije. Dok se sinhrona komunikacija vrđi kada UserInfo servis od Manga CRUD servisa traži informacije o konkretnim mangama kako bi mogao da odredi razne preporuke koje mu korisnik zahteva.

Tokom rada plan za Reading synchronization servis će biti detaljnije razvijen.

**Arhitektura baza podataka**

**Manga database - PostGresql**



**UserInfo database - Redis**



Polje reading\_journal se može deserijalizovati u:

{

string user\_id,

string last\_read\_manga\_id,

List<string> all\_read\_manga\_ids,

List<string> wish\_list

}

**IdentityServer database - Mongo**

