HRONOLOGIJA RAZVOJA BAZA PODATAKA

Od datoteka i hijerarhijskih struktura do distribuiranih i AI-orijentisanih baza



Predavač: dr Dušan Stefanović

ZAŠTO SU SE BAZE PODATAKA MENJALE?

Razvoj je vođen zahtevima aplikacija, ne samo tehnologijom.

KOLIČINA

Od KB i MB ka TB i PB

STRUKTURA

Od tabela ka JSON-u, grafovima, vektorima

BRZINA

Od periodične obrade ka realnom vremenu

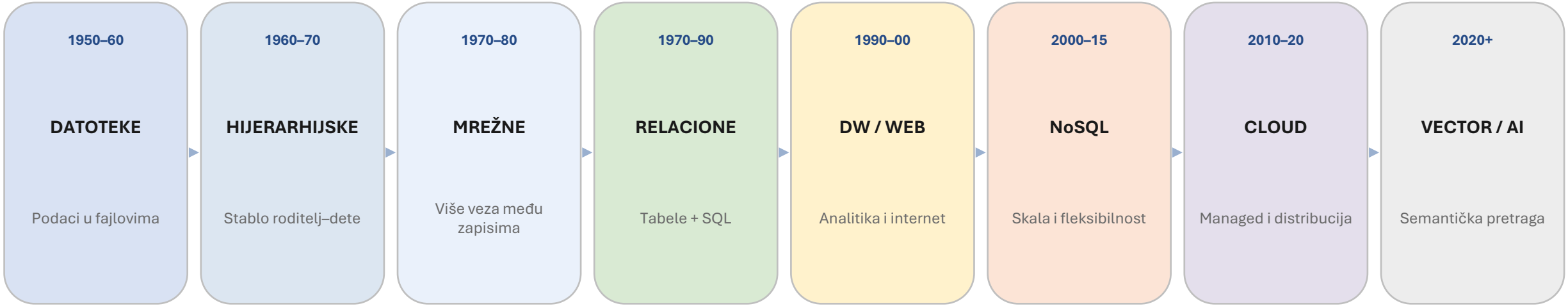
DISTRIBUCIJA

Od jednog računara ka cloud regionima

Ključna ideja: jedan model baze ne rešava podjednako dobro sve probleme.

HRONOLOGIJA U JEDNOM POGLEDU

Vizuelna mapa razvoja — od lokalnih datoteka do distribuiranih i AI baza.

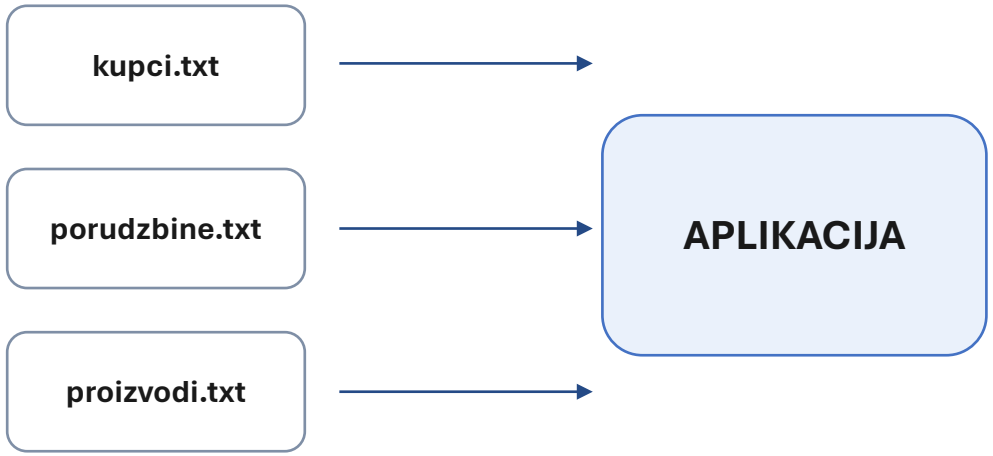


Važno: periodi se preklapaju. Novi modeli nisu potpuno zamenili stare — već su proširili izbor.

1. SISTEMI ZASNOVANI NA DATOTEKAMA

Pre DBMS-a aplikacije su direktno čitale i menjale fajlove.

1950–1960



- Podaci su vezani za program koji ih koristi
- Mnogo dupliranih podataka
- Teško istovremeno ažuriranje i deljenje
- Pretraga se programira ručno

PRIMER UPOTREBE

Mala lokalna evidencija

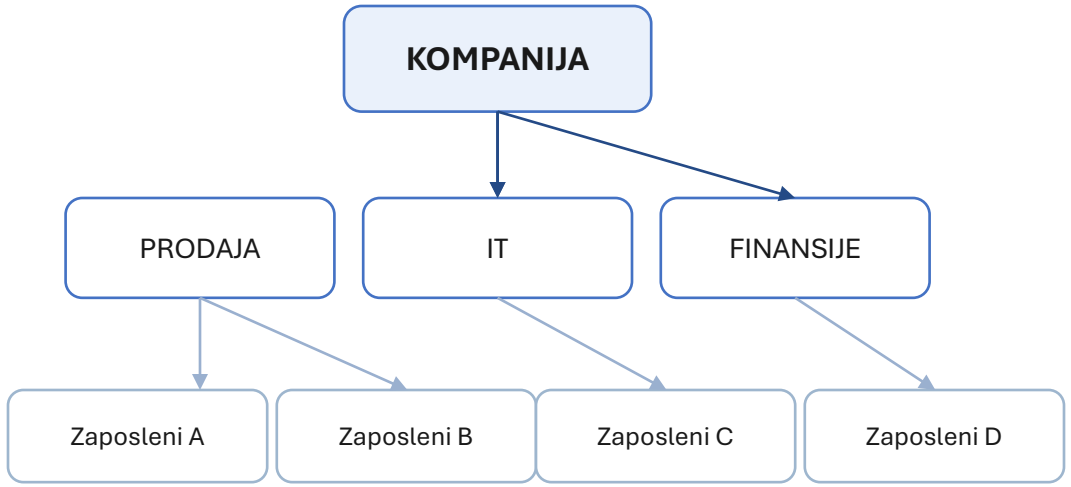
CSV datoteka sa kontaktima ili inventarom u maloj aplikaciji bez složenih veza.

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | NoSQL | CLOUD / AI

2. HIJERARHIJSKE BAZE

Podaci su organizovani kao stablo: jedan roditelj može imati više potomaka.

1960-e



PRIMER UPOTREBE

IBM IMS

Veliki transakcioni sistemi, npr. tradicionalni sistemi rezervacija i poslovne evidencije sa stabilnom hijerarhijom.

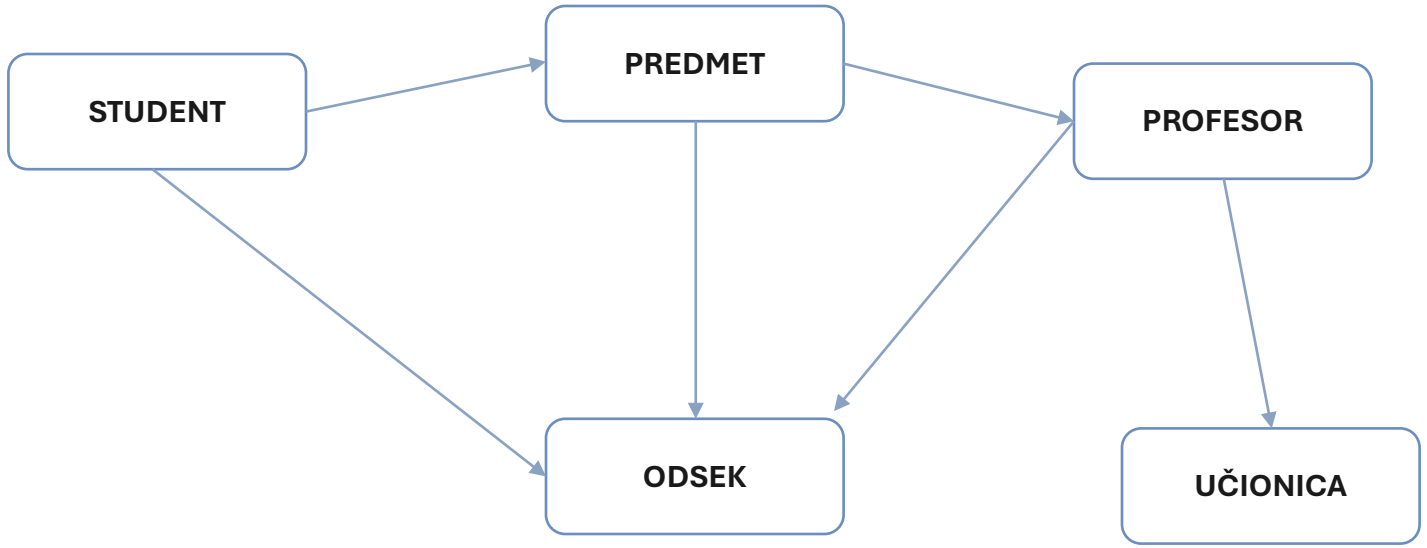
- Brza navigacija kroz poznatu hijerarhiju
- Jednostavno kada je odnos prirodno 1:N
- Problem kada jedan zapis treba više roditelja

DATOTEKE | **HIJERARHIJSKE** | MREŽNE | RELACIONE | NoSQL | CLOUD / AI

3. MREŽNE BAZE

Zapis može imati više roditelja — struktura više nije strogo stablo.

1960–1970-e



PRIMER UPOTREBE

CODASYL pristup

Sistemi sa složenim unapred poznatim vezama: proizvodnja, telekomunikacije i veliki poslovni sistemi.

PREDNOST više veza nego kod hijerarhijskog modela

SLABOST aplikacija mora da zna put kroz mrežu zapisa

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | NoSQL | CLOUD / AI

4. RELACIONI MODEL — PREKRETNICA

Coddov model iz 1970. organizuje podatke u relacije — praktično, u tabele.

1970+



PRIMER UPOTREBE

Banka / ERP / e-commerce

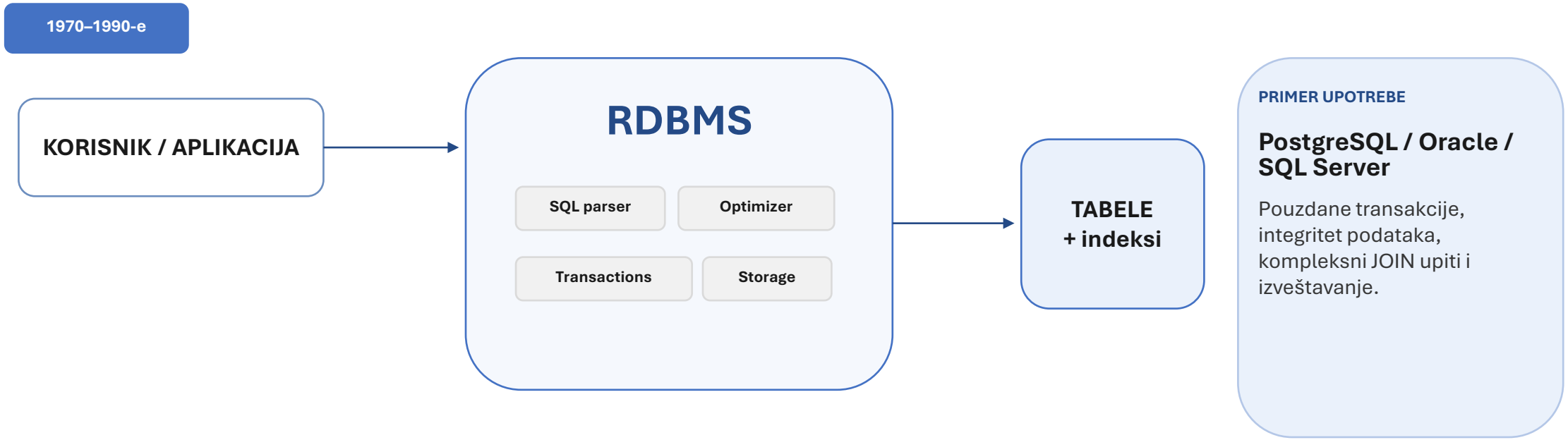
Transakcije, korisnici, računi, narudžbine i drugi podaci sa jasnim pravilima i vezama.

- Podaci se opisuju logički, bez navigacije kroz fizičke putanje
- Veze se izražavaju ključevima
- Upiti mogu da kombinuju tabele

DATOTEKE | HIJERARHIJSKE | MREŽNE | **RELACIONE** | NoSQL | CLOUD / AI

5. SQL I RDBMS — STANDARDIZOVAN PRISTUP

Relacijski model postaje praktičan kroz SQL i sisteme za upravljanje bazama.



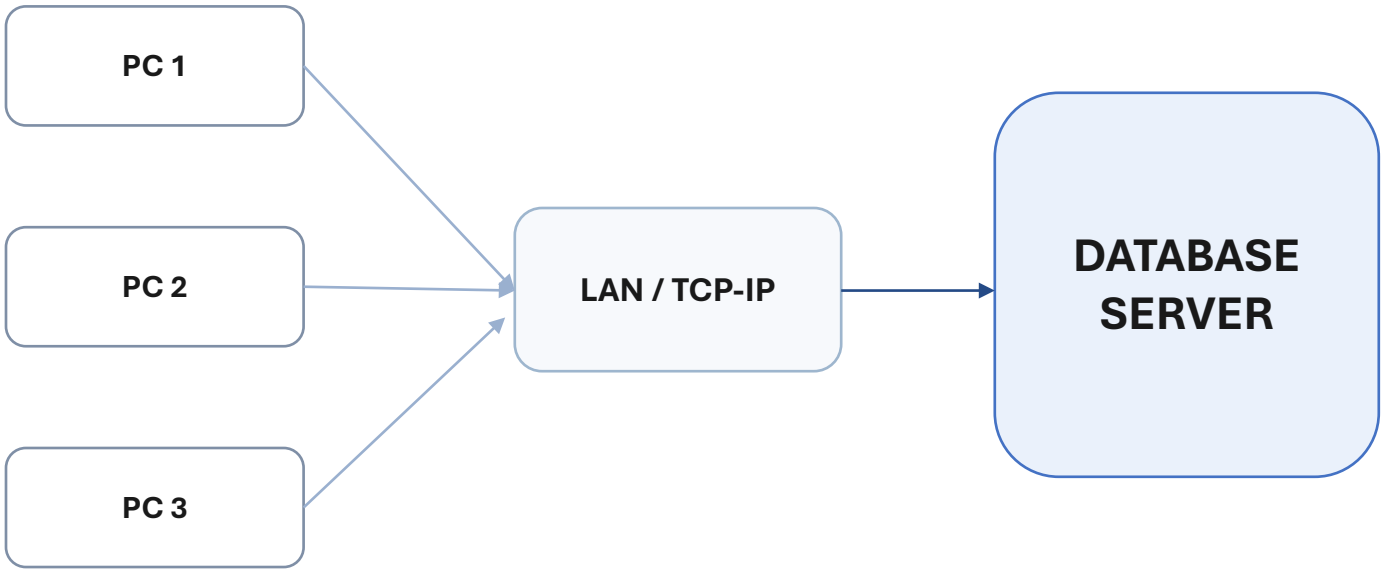

```
SELECT Kupac.Ime, Porudzbina.Datum  FROM Kupac JOIN Porudzbina ...
```

DATOTEKE | HIJERARHIJSKE | MREŽNE | **RELACIONE** | NoSQL | CLOUD / AI

6. KLIJENT–SERVER ERA

Baza se odvaja od korisničke aplikacije i postaje centralni servis.

1980–1990-e



PRIMER UPOTREBE

Poslovna mreža

Više računara u firmi pristupa istoj centralnoj bazi za prodaju, skladište i računovodstvo.

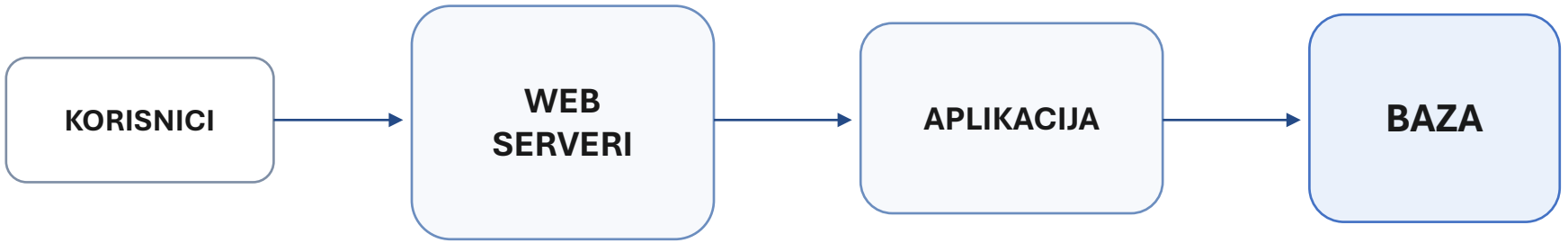
Dobit: jedna verzija podataka + centralna kontrola pristupa

DATOTEKE | HIJERARHIJSKE | MREŽNE | **RELACIONE** | NoSQL | CLOUD / AI

7. WEB MENJA ZAHTEVE

Internet donosi milione korisnika, 24/7 rad i nagle skokove opterećenja.

1995–2010



PRIMER UPOTREBE

E-commerce / društvene mreže

Aplikacije moraju da skaliraju brzo i da ostanu dostupne i kada broj zahteva naglo poraste.

PROMENLJIV SAOBRAĆAJ

npr. flash sale

GLOBALNI KORISNICI

niska latencija

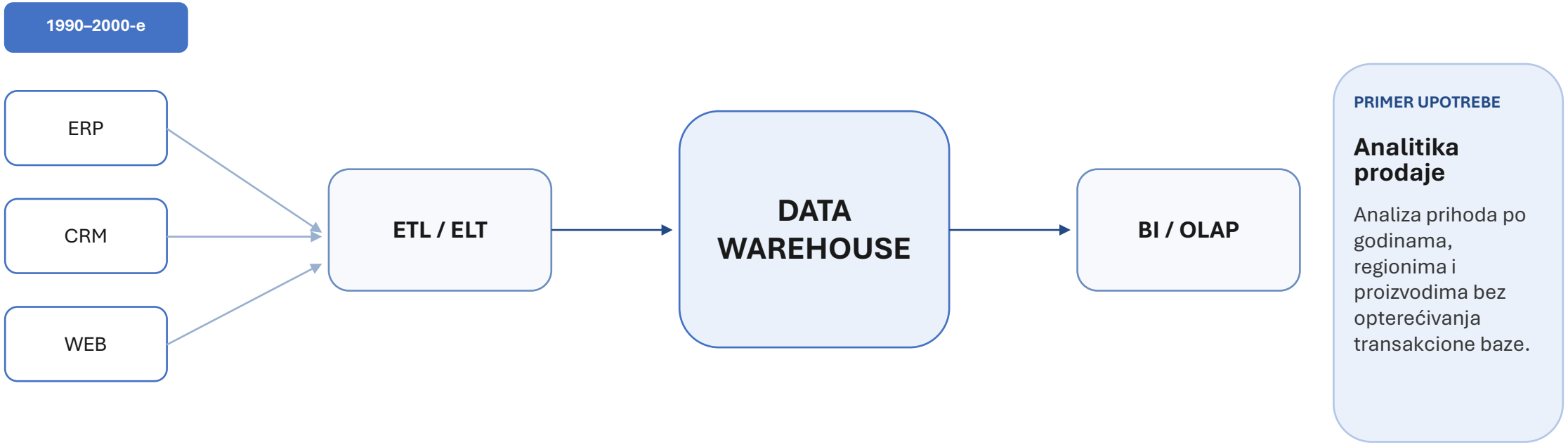
24/7 DOSTUPNOST

bez dužeg prekida

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | **NoSQL** | CLOUD / AI

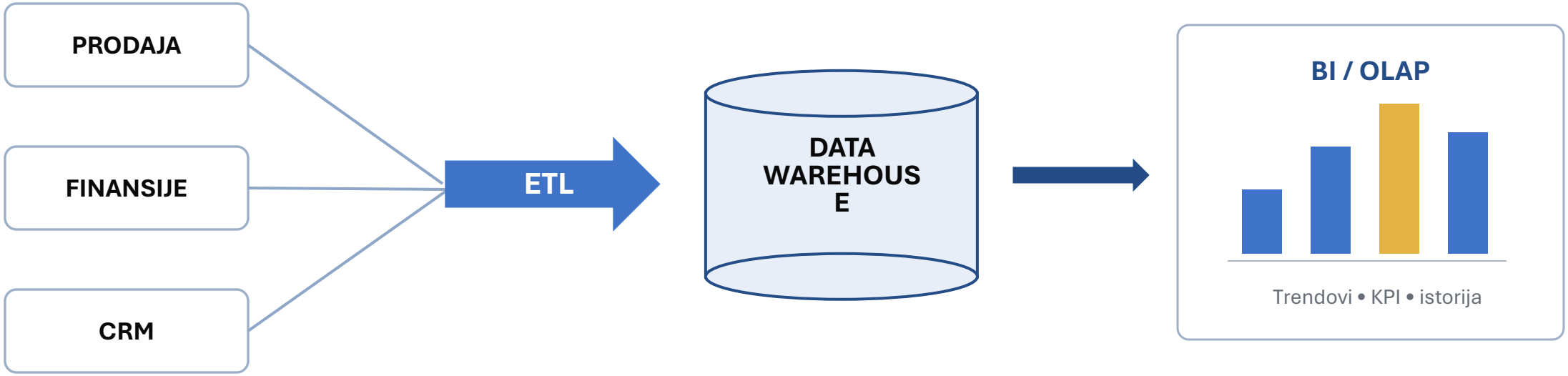
8. DATA WAREHOUSE I OLAP

Operativna baza odgovara „šta se dešava sada“, skladište podataka „šta se dešavalo kroz vreme“.



Operativni podaci → integracija → istorijski podaci → analiza i odluke

8. DATA WAREHOUSE I OLAP



PRIMER **Maloprodaja: analiza prodaje po godini, regionu i kategoriji**

9. BIG DATA — KADA JEDAN SERVER VIŠE NIJE DOVOLJAN

Volume, velocity i variety guraju obradu ka klasterima.

2000-e

VOLUME

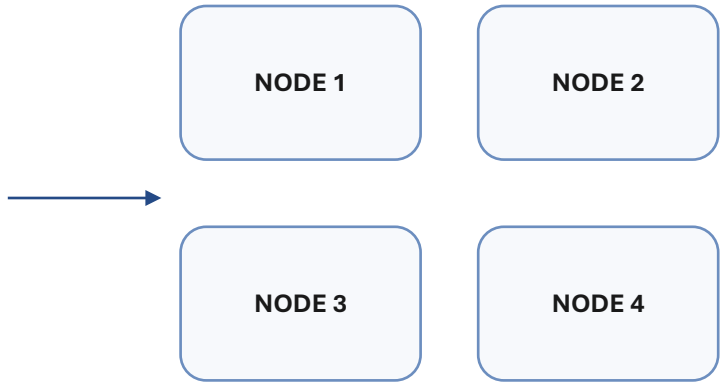
milijarde zapisa

VELOCITY

stalni tok podataka

VARIETY

logovi, tekst, JSON, senzori



Nova ideja: skalirati horizontalno — dodavanjem servera.

PRIMER UPOTREBE

Logovi i telemetrija

Milioni događaja sa servera ili IoT uređaja raspodeljuju se i obrađuju na više mašina.

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | **NoSQL** | CLOUD / AI

10. NoSQL — VIŠE MODELA ZA RAZLIČITE SLUČAJEVE

NoSQL nije jedna baza, već porodica modela koji odstupaju od klasičnog tabelarnog pristupa.

KEY–VALUE

ključ → vrednost

Redis

Primer: sesije / cache

DOKUMENT

JSON dokument

MongoDB

Primer: katalog proizvoda

COLUMN-FAMILY

particije + kolone

Cassandra

Primer: IoT / događaji

GRAF

čvorovi + veze

Neo4j

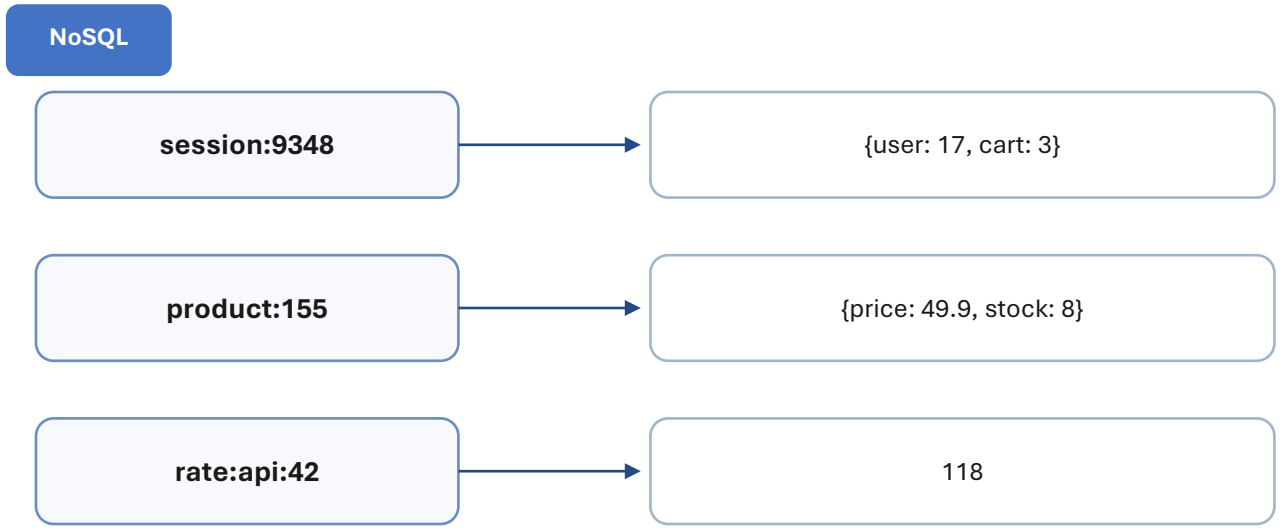
Primer: preporuke / prevare

Izbor modela prati oblik podataka i obrazac pristupa.

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | **NoSQL** | CLOUD / AI

11. KEY–VALUE BAZE

Najjednostavniji NoSQL model: jedinstven ključ vodi direktno do vrednosti.



PRIMER UPOTREBE

Redis

Session store, cache, rate limiting i veoma brza privremena stanja aplikacije.

- Vrlo brz pristup po ključu
- Model je jednostavan i lako se distribuira
- Nije idealan za složene odnose i JOIN upite

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | **NoSQL** | CLOUD / AI

12. DOKUMENT BAZE

Podaci se čuvaju kao samostalni JSON/BSON dokumenti sa fleksibilnom strukturom.

NoSQL

```
{
  "productId": 155,
  "name": "Smart Sensor",
  "price": 49.90,
  "tags": ["IoT", "WiFi"],
  "spec": { "range": 30, "battery": "2y" }
}
```

PRIMER UPOTREBE

MongoDB

Katalog proizvoda, profil korisnika ili CMS gde različiti objekti mogu imati različita polja.

- Prirodno se uklapa sa objektima web API-ja
- Lakše menja strukturu dokumenta
- Povezani podaci se često modeluju drugačije nego u tabelama

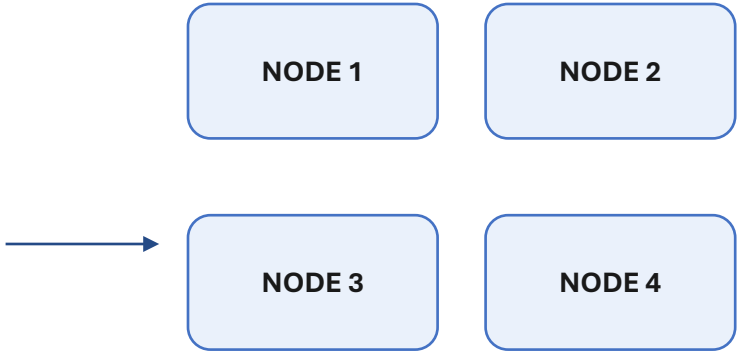
DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | **NoSQL** | CLOUD / AI

13. WIDE-COLUMN BAZE

Podaci se organizuju oko redova i porodica kolona, optimizovanih za distribuciju i veliki obim.

NoSQL

device_id	time	temp	humidity
A17	10:01	22.4	48
A17	10:02	22.5	49
B04	10:01	19.7	55
B04	10:02	19.8	55



PRIMER UPOTREBE

Apache Cassandra

Telemetrija, clickstream i servisi sa velikim brojem upisa i raspodelom podataka na mnogo čvorova.

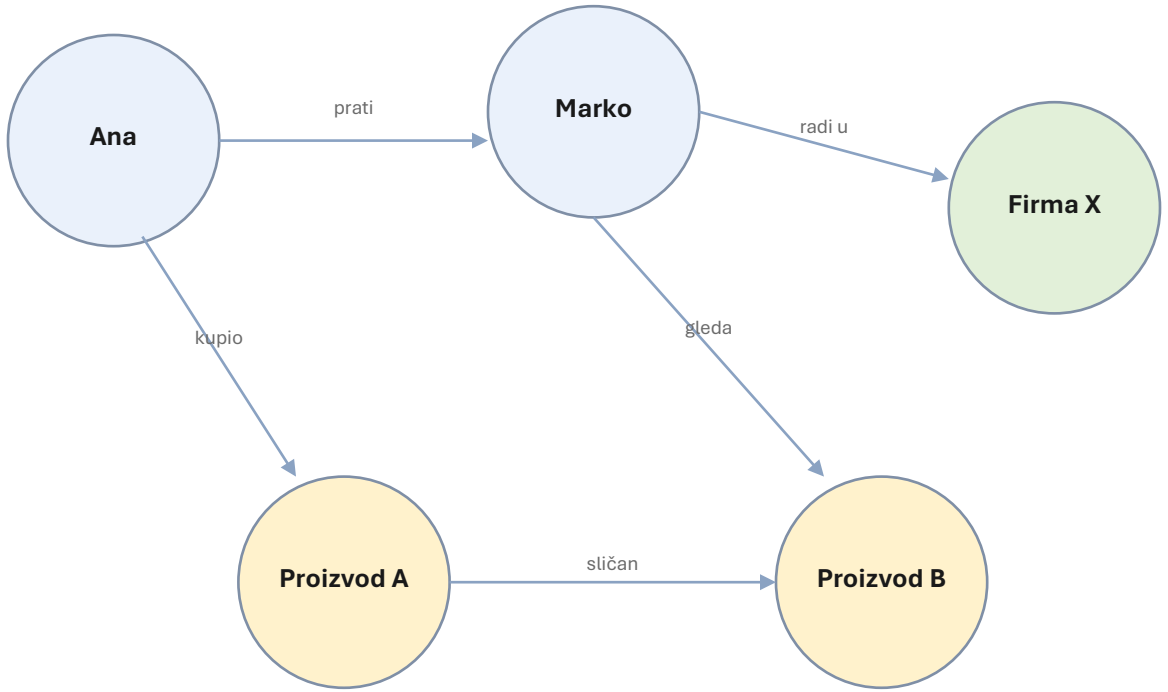
Model se često dizajnira prema upitima koje aplikacija mora brzo da izvršava.

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | **NoSQL** | CLOUD / AI

14. GRAF BAZE

Kada su veze podjednako važne kao i sami podaci.

NoSQL



PRIMER UPOTREBE

Neo4j

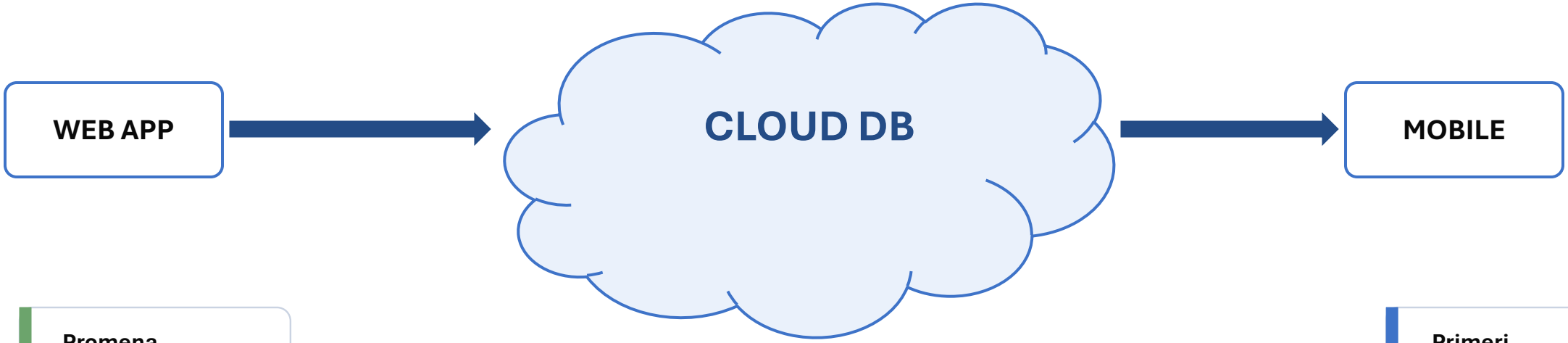
Preporuke, društvene mreže, fraud detection i analiza veza između korisnika, uređaja i transakcija.

Relacijski JOIN može biti dovoljan za nekoliko veza; graf postaje prirodan kada su višestruke veze srž problema.

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | **NoSQL** | CLOUD / AI

15. CLOUD BAZE I DBaaS

Baza postaje upravljani servis: provisioning, backup, replike i skaliranje



Promena

Tim više ne mora da održava svaki detalj infrastrukture baze.

automatski backup

replikacija

monitoring

skaliranje

Primeri

AWS RDS • Azure SQL • Cloud SQL

PRIMER **SaaS aplikacija koja raste bez sopstvenog DBA server tima**

Cloud

16. MODERNE SPECIJALIZOVANE BAZE

Jedna aplikacija danas često koristi više baza — svaku za ono u čemu je najbolja.

2020+

TIME-SERIES

vreme + merenje

Primer tehnologije

Prometheus / Timescale

IoT, monitoring

SPATIAL

geometrija + lokacija

Primer tehnologije

PostGIS

mape, saobraćaj

VECTOR

embedding vektori

Primer tehnologije

pgvector / Milvus

semantička pretraga, RAG

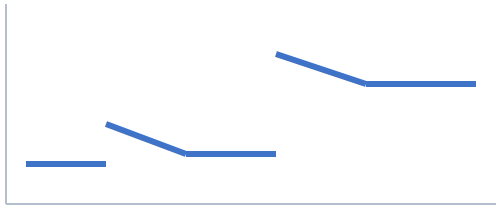
Poliglot persistence: kombinovanje više baza u istom sistemu prema nameni.

DATOTEKE | HIJERARHIJSKE | MREŽNE | RELACIONE | NoSQL | **CLOUD / AI**

16. SPECIJALIZOVANE MODERNE BAZE

Jedna aplikacija može koristiti različite baze za različite vrste podataka

TIME-SERIES

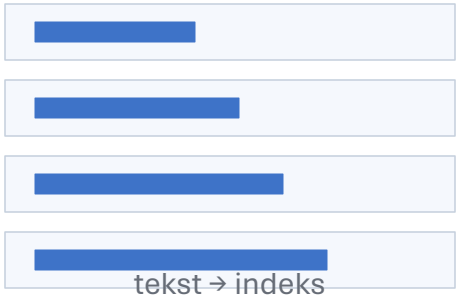


vreme → merenje

InfluxDB / TimescaleDB

Primer: IoT i monitoring

SEARCH

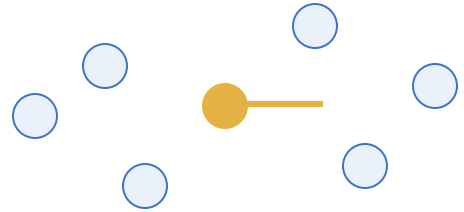


tekst → indeks

Elasticsearch / OpenSearch

Primer: logovi i full-text pretraga

VECTOR



vektor → sličnost

pgvector / Milvus / Pinecone

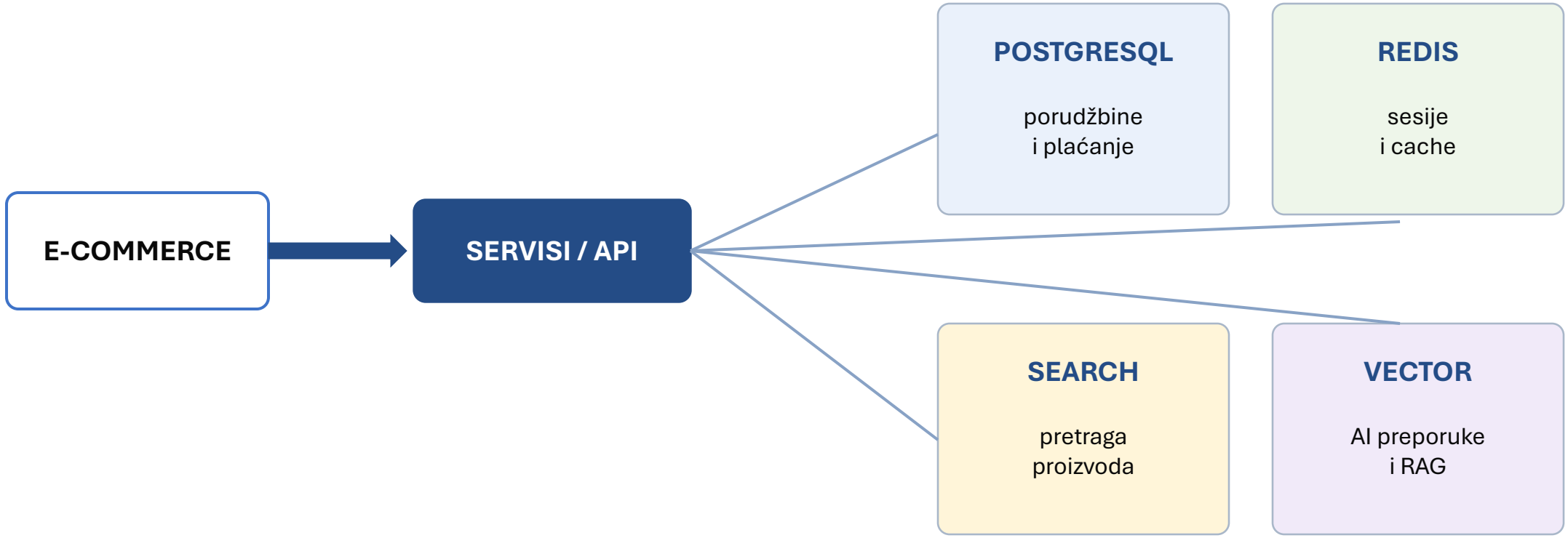
Primer: semantička pretraga i RAG

Moderni sistem često koristi „polyglot persistence“ — više baza, svaka za posao koji najbolje radi.

AI/vekt.

17. JEDNA APLIKACIJA — VIŠE BAZA

Bazu ne biramo po trendu, već po obrascu podataka i upita



Ne postoji „najbolja baza“ — postoji baza koja najbolje odgovara konkretnom delu sistema.

AI/vekt.

OD TRADICIONALNIH DO MODERNIH BAZA — ŠTA TREBA ZAPAMTITI

Ne postoji univerzalno „najbolja baza“ — postoji baza koja odgovara konkretnom problemu.

MODEL / ERA	GLAVNA IDEJA	TIPIČAN PRIMER
DATOTEKE	jednostavno čuvanje	mala lokalna evidencija
HIJERARHIJSKE / MREŽNE	navigacija kroz veze	stabilne, unapred poznate strukture
RELACIONE	integritet + transakcije	banke, ERP, e-commerce
NoSQL	skala + fleksibilna struktura	web, cache, telemetrija, grafovi
CLOUD / DISTRIBUIRANE	dostupnost + globalna skala	SaaS i globalne aplikacije
SPECIJALIZOVANE / VECTOR	optimizacija za specifičan tip	IoT, GIS, AI / RAG

Pitanje pri izboru baze: Kako su podaci organizovani? Kako im pristupamo? Koliko moraju da skaliraju? Koju konzistentnost zahtevamo?