The top-left portion of the slide features a complex, abstract pattern of thin, black, overlapping lines. These lines form various geometric shapes, including triangles and polygons, creating a sense of depth and movement. The lines are scattered across the upper-left quadrant, with some extending towards the center of the slide.

IZBOR BAZE PODATAKA PRILIKOM DIZAJNIRANJA SISTEMA

AGENDA

Skaliranje

CAP Teorema

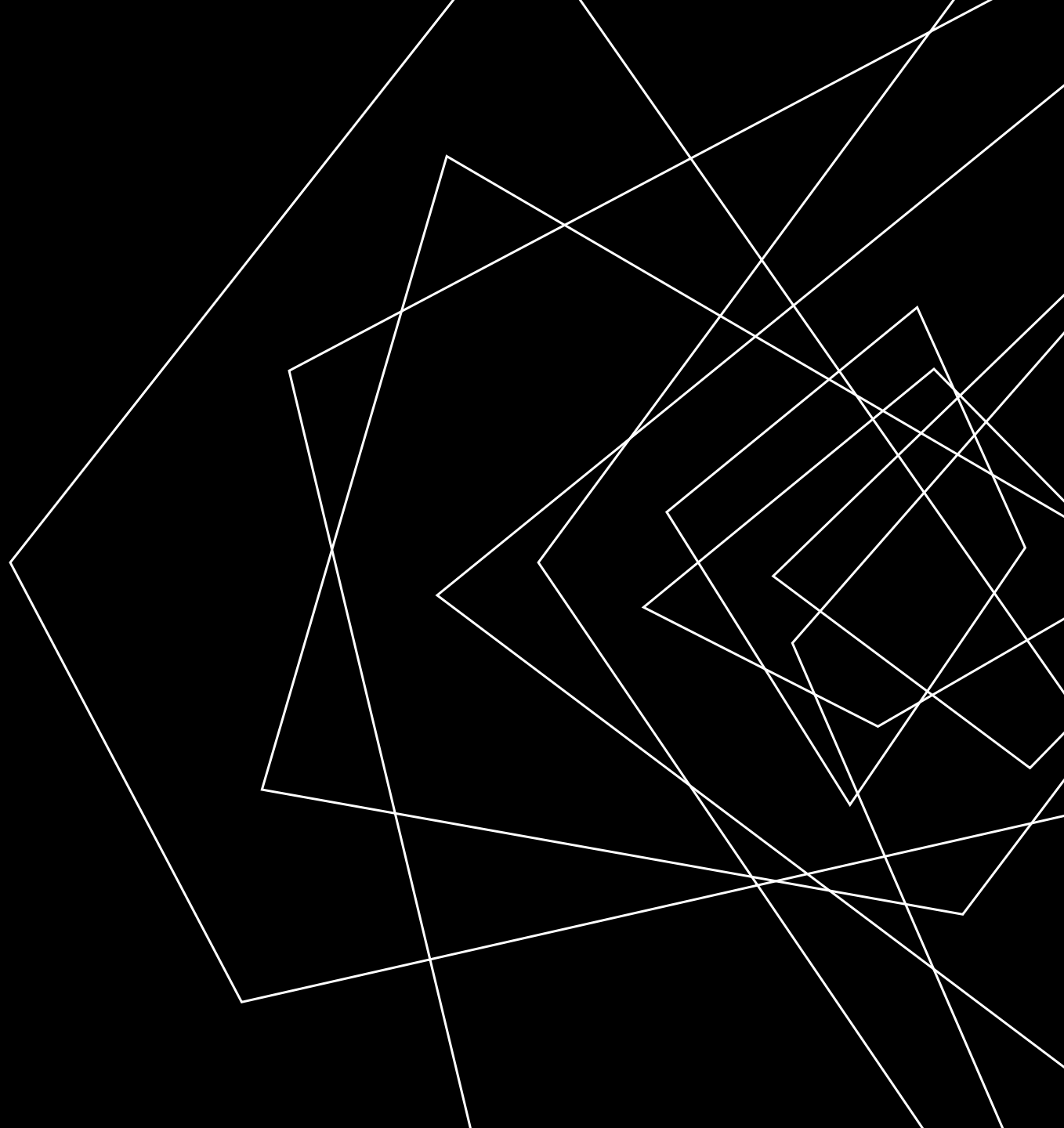
Tipovi baza

Kako odabrati

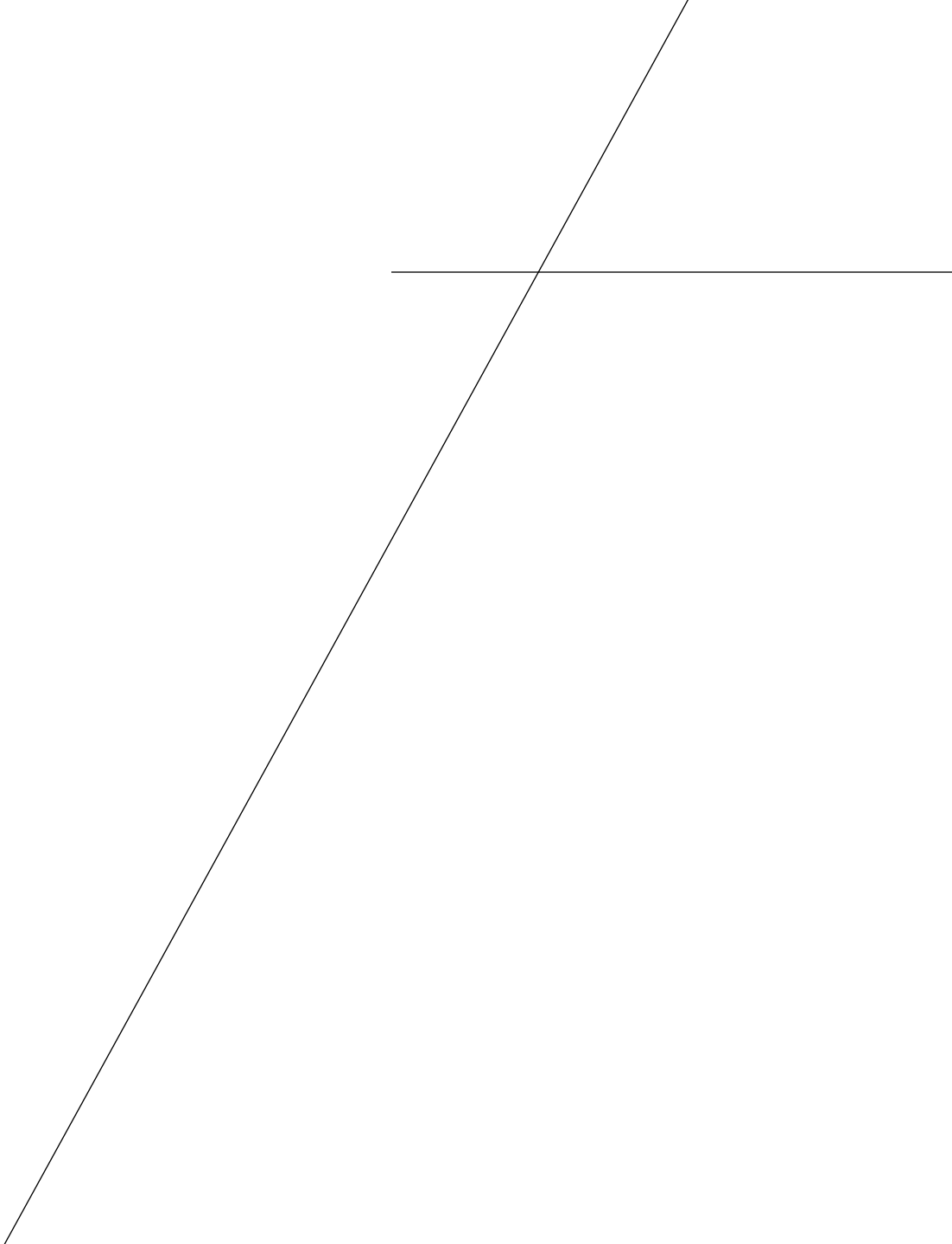
Dizajn velikog sistema

Linkovi

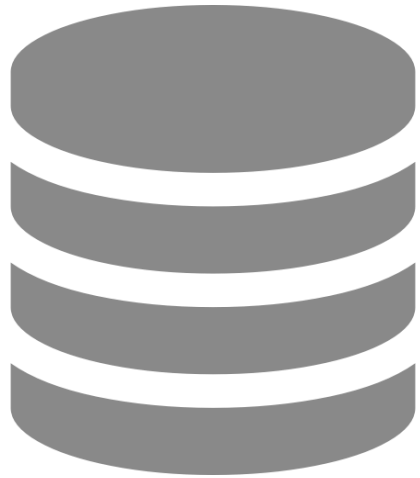
MongoDb / Kreiranje, korišćenje i održavanje
klastera (Sledeći termin)



ODABIR BAZE PODATAKA

- Najveći broj aplikacija koristi baze podataka
 - Data – najznačajniji resurs
 - Jedna od najbitnijih odluka tokom dizajna sistema
 - Migracija je skupa i rizična procedura
 - Svaki tip baze podataka ima prednosti i nedostatke
- 

KRENIMO OD



Jedna instanca

Porast broja korisnika



- CPU/Memory overloaded
- Out of storage
- Network overloaded

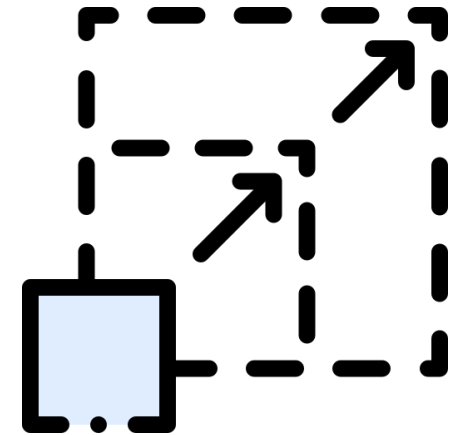
Kako proveravamo?



Optimizacija

- Aplikacija (cache)
- Baza (index)

Dostignut limit



Skaliranje

Kako testiramo?

VERTIKALNO SKALIRANJE

Single instance – Vertical scaling

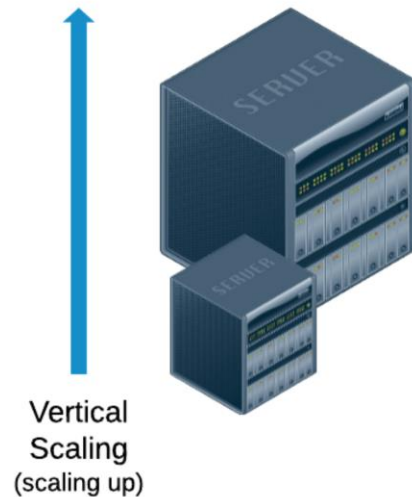
- Dodavanje više CPU-a i memorije

Prednosti:

- Infrastruktura baze se ne menja osim hardvera
- Lakše za održavanje

Nedostaci:

- Cena (ne raste linearno)
- Fizičko ograničenje CPU I Memorije na jednoj mašini
- Migracija hardvera - downtime



HORIZONTALNO SKALIRANJE

Multi instance (Distributed) - Horizontal scaling

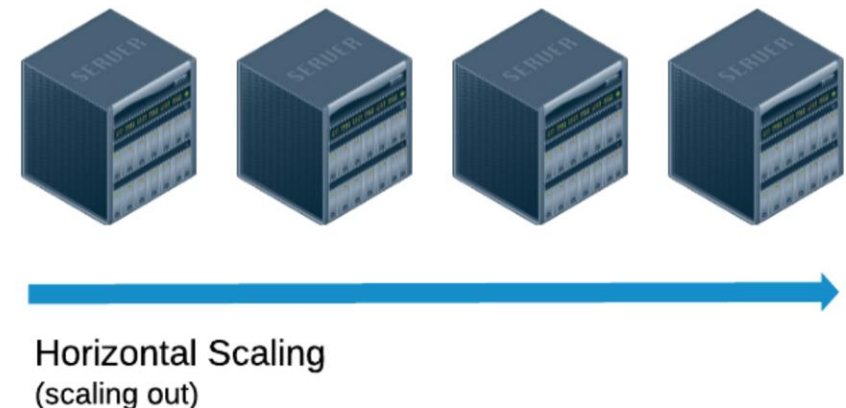
- Dodavanje novih instanci

Prednosti:

- Podržava veći load

Nedostaci:

- Kompleksnije za održavanje



HOSTING

- Lokalno - pogodno za development
- Data Center
- Cloud Providers (Infrastructure as a Service – IaaS): Google Cloud Provider (GCP), Amazon Web Services (AWS), Azure...
 - Self-hosted: Sami kreiramo instance I brinemo od dostupnosti I skalabilnosti baze – koristimo Virtuelne mašine ili kontejnere. Zahteva znanje arhitekture baze
 - Managed solutions: Postoji sistem koji brine od kreiranju I menadzmentu baze



CAP Theorem

Distribuirane baze mogu da zadovolje dve od sledeće tri stavke (AP, CP): ???

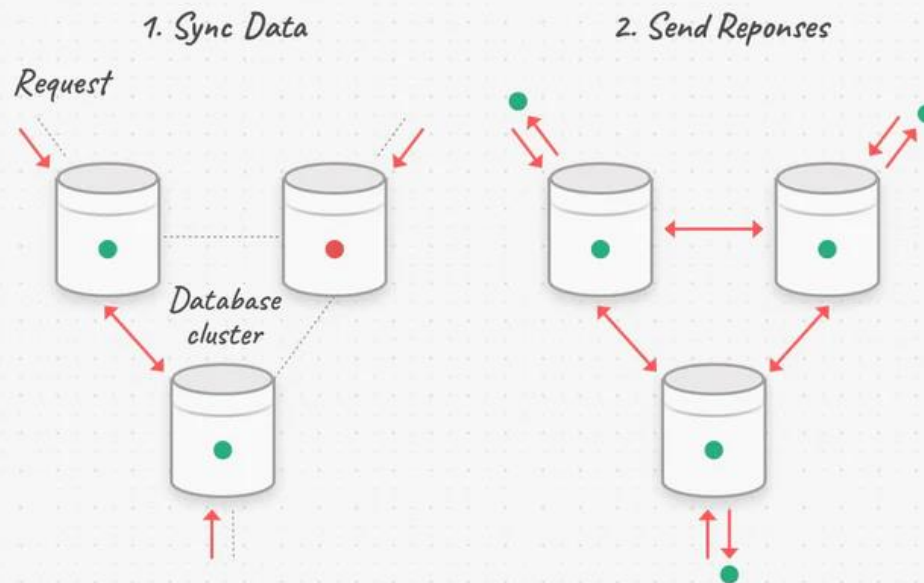
- **Consistency**: svaka instanca sadrži poslednju verziju podataka / svi klijenti vide iste podatke (≠ ACID Consistency)
- **Availability**: sistem radi iako sve instance nisu dostupne / uvek dobijamo odgovor na zahtev
- **Partition Tolerance**: način na koji sistem radi kada je komunikacija između bilo koje dve instance prekinuta (problemi na mreži)

Strong Consistency – Reduced Availability

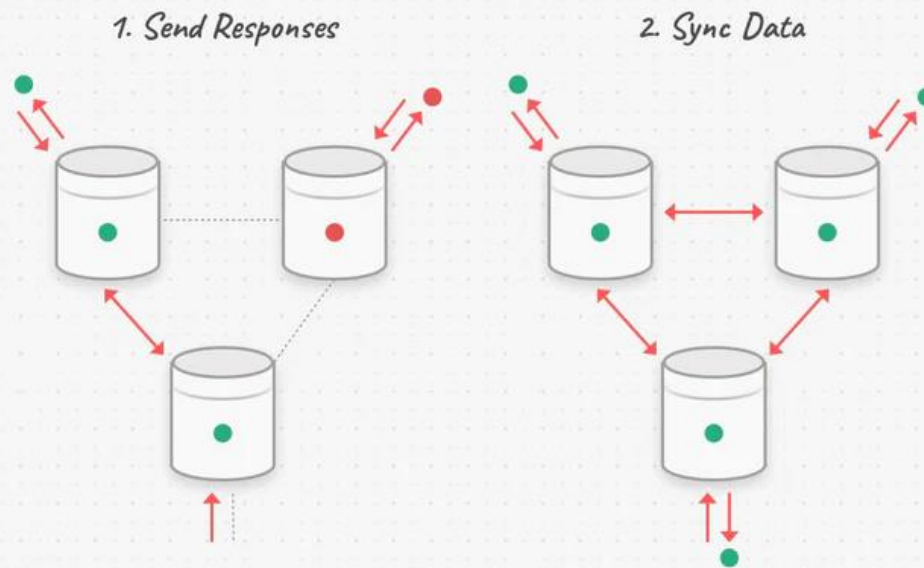
High Availability – Eventual Consistency

How different databases respond to network failures

CP Database

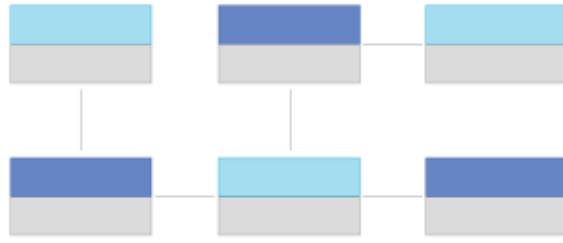


AP Database



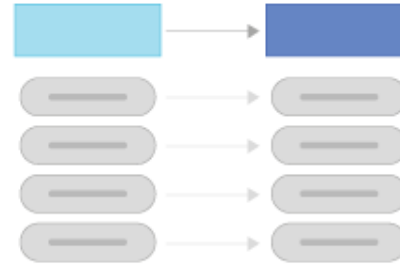
SQL

Relational

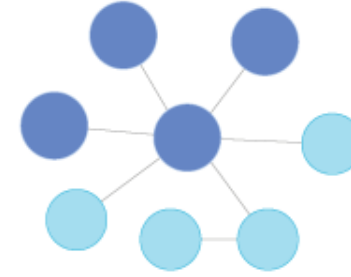


NoSQL

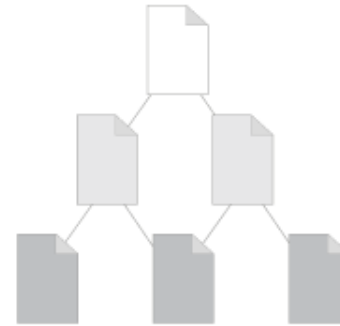
Key - Value



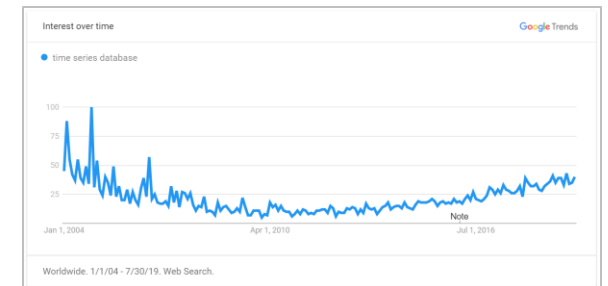
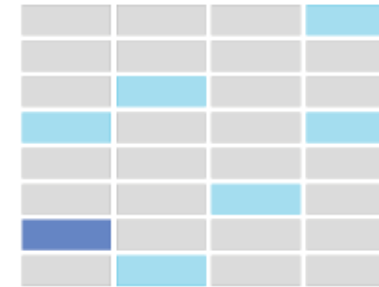
Graph



Document



Wide Column



Time series

RELACIONE BAZE

- Ogranizacija u tabelama
- Transakcije - ACID
- SQL
- Index



ORACLE®



Kada koristimo

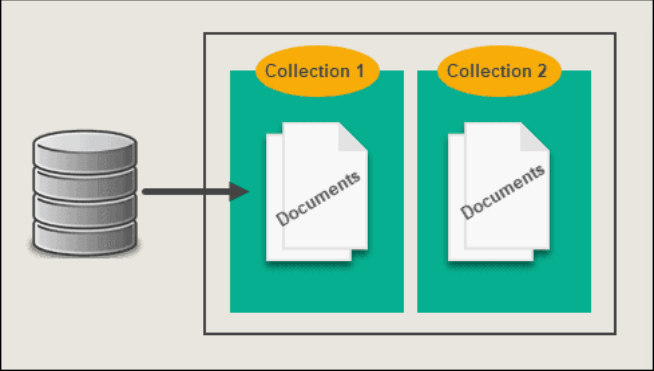
- Jasno definisana struktura
- Many-to-many relationships
- Veze između entiteta uvek moraju da budu tačne
- Strong consistency
- Primer: Finansije, Zdravstvene ustanove

Nedostaci

- Teže je postići horizontalno skaliranje zbog veza, joina / Partitioning, Sharding/ aplikacija je svesna instanci baze
- Schema overhead – Provera korektnosti šeme usporava upis u bazu
- Teška promena šeme

Document baze

- JSON objekti / BSON
- Document – Set podataka, organizovan hijerarhijski (isti podaci bi u relacionoj bazi bili podeljeni u više tabela)
- Fleksibilna šema - dodavanje novog property-a ne utiče na ostale dokumente
- Denormalizovani podaci
- Data locality



```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  }
  "hobbies": ["surfing", "coding"]
}
```

Kada koristimo

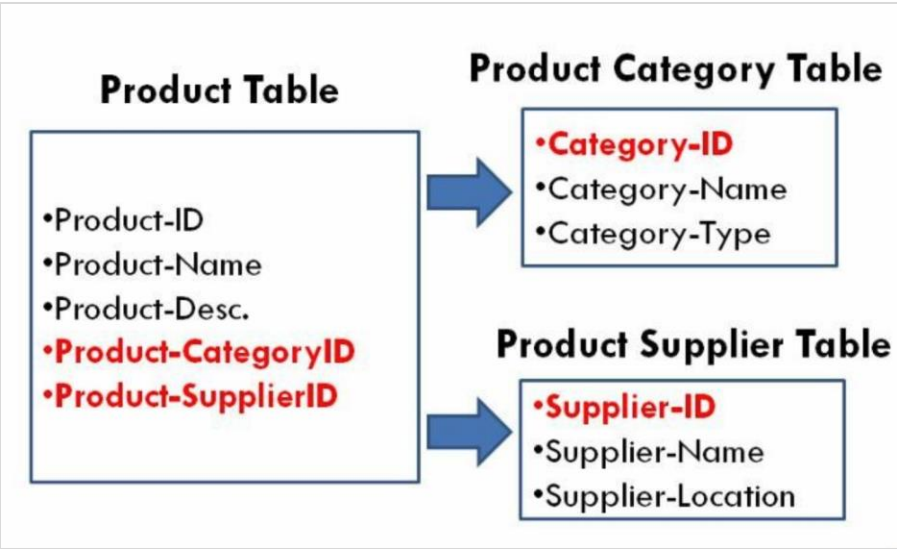
- Slabo definisana šema
- Horizontalno skaliranje, visoke performanse
- CRUD aplikacije
- User Profiles

Nedostaci

- Slaba podrška za join – zahteva prebacivanje posla na nivo aplikacije što je sporije



Primer: Product Catalog



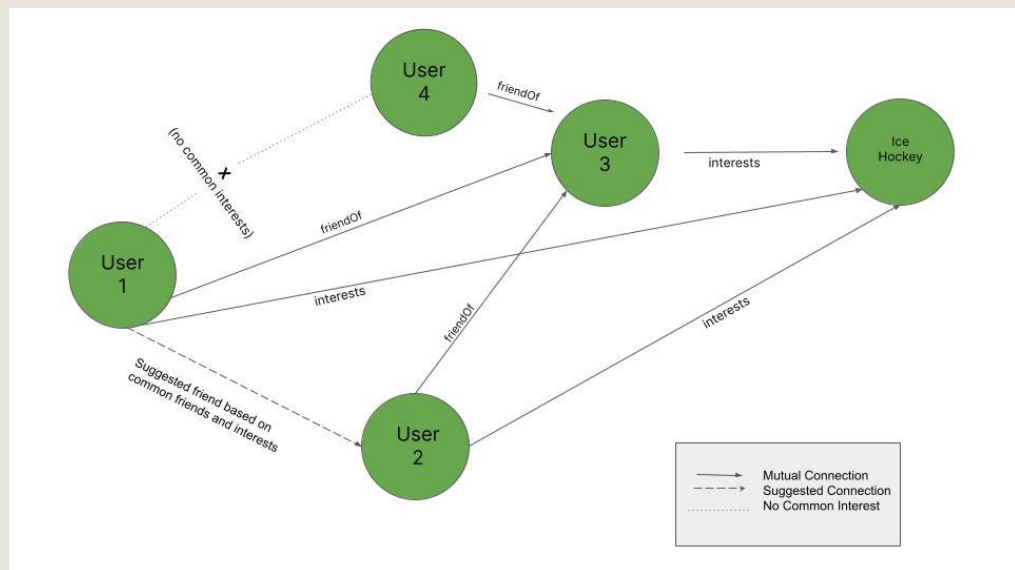
Relational



Document

Graph baze

- Koriste koncept grafa kao model podataka
- Sastoji se od čvorova i grana
- **Čvor (node)** - sadrži podatke
- **Grana (edge)** - veza između čvorova
- Ugrađeni algoritmi za pretragu u grafu

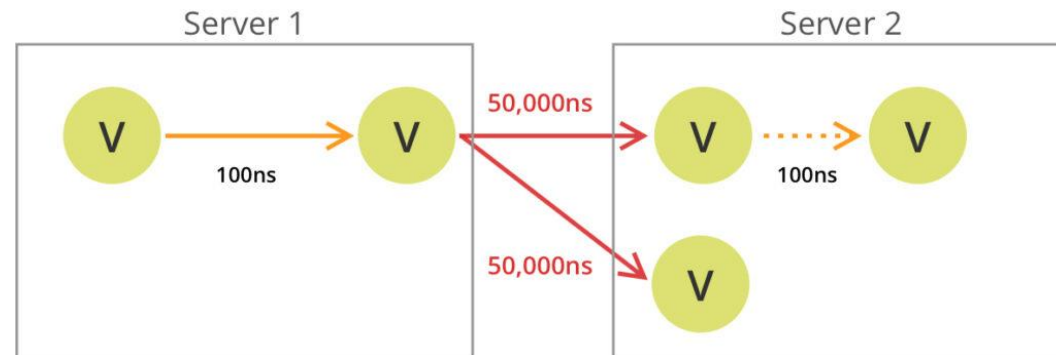


Kada koristimo

- Many-to-many relationships
- Potreba za fleksibilnošću modela
- Real-time recommendation engines
- Identity and access management (IAM)

Nedostaci

- Ne postoji standardizovani jezik
- Teže horizontalno skaliranje
- Manje korisnika - teže za podršku
- Slaba podrška za transakcije
- Supernodes



Graph vs Relational – many relationships

Graph baza: neo4j

Relaciona baza: MySql

Primer društvene mreže: pretraga prijatelja u dubinu

Broj korisnika: 1 000 000

Rezultati pretrage za: 1 000 korisnika

Vreme: sekunde

Depth	Execution Time – MySQL	Execution Time –Neo4j
2	0.016	0.010
3	30.267	0.168
4	1,543.505	1.359
5	Not Finished in 1 Hour	2.132

Key-value baze

- Jedinstveni string key
- Vrednost

Phone directory

Key	Value
Paul	(091) 9786453778
Greg	(091) 9686154559
Marco	(091) 9868564334

MAC table

Key	Value
10.94.214.172	3c:22:fb:86:c1:b1
10.94.214.173	00:0a:95:9d:68:16
10.94.214.174	3c:1b:fb:45:c4:b1



Kada koristimo

- aplikacija treba da obrađuje puno malih kontinuiranih čitanja i pisanja
- skladištenje osnovnih informacija, kao što su detalji o klijentima; čuvanje veb stranica sa URL-om kao ključem i veb stranicom kao vrednosti
- Aplikacije koje ne izvršavaju kompleksne upite
- Sessions, cache
- Upravljanje sesijom svakog igrača u masovnim multiplayer igricama

Nedostaci

- Ne možemo da filtriramo vrednosti

Column-family base

- Čuva podatke po kolonama umesto redovima
- Optimizovana za čuvanje kolona podataka
- principle of locality
- Compression

RowId	EmpId	Lastname	Firstname	Salary
001	10	Smith	Joe	60000
002	12	Jones	Mary	80000
003	11	Johnson	Cathy	94000
004	22	Jones	Bob	55000

```
001:10,Smith,Joe,60000;  
002:12,Jones,Mary,80000;  
003:11,Johnson,Cathy,94000;  
004:22,Jones,Bob,55000;
```

Row-oriented

```
55000:004;  
60000:001;  
80000:002;  
94000:003;
```

Row-oriented: index

```
10:001,12:002,11:003,22:004;  
Smith:001,Jones:002,Johnson:003,Jones:004;  
Joe:001,Mary:002,Cathy:003,Bob:004;  
60000:001,80000:002,94000:003,55000:004;
```

Column-oriented



Kada koristimo

- analytical applications
- Big data processing
- Scalable

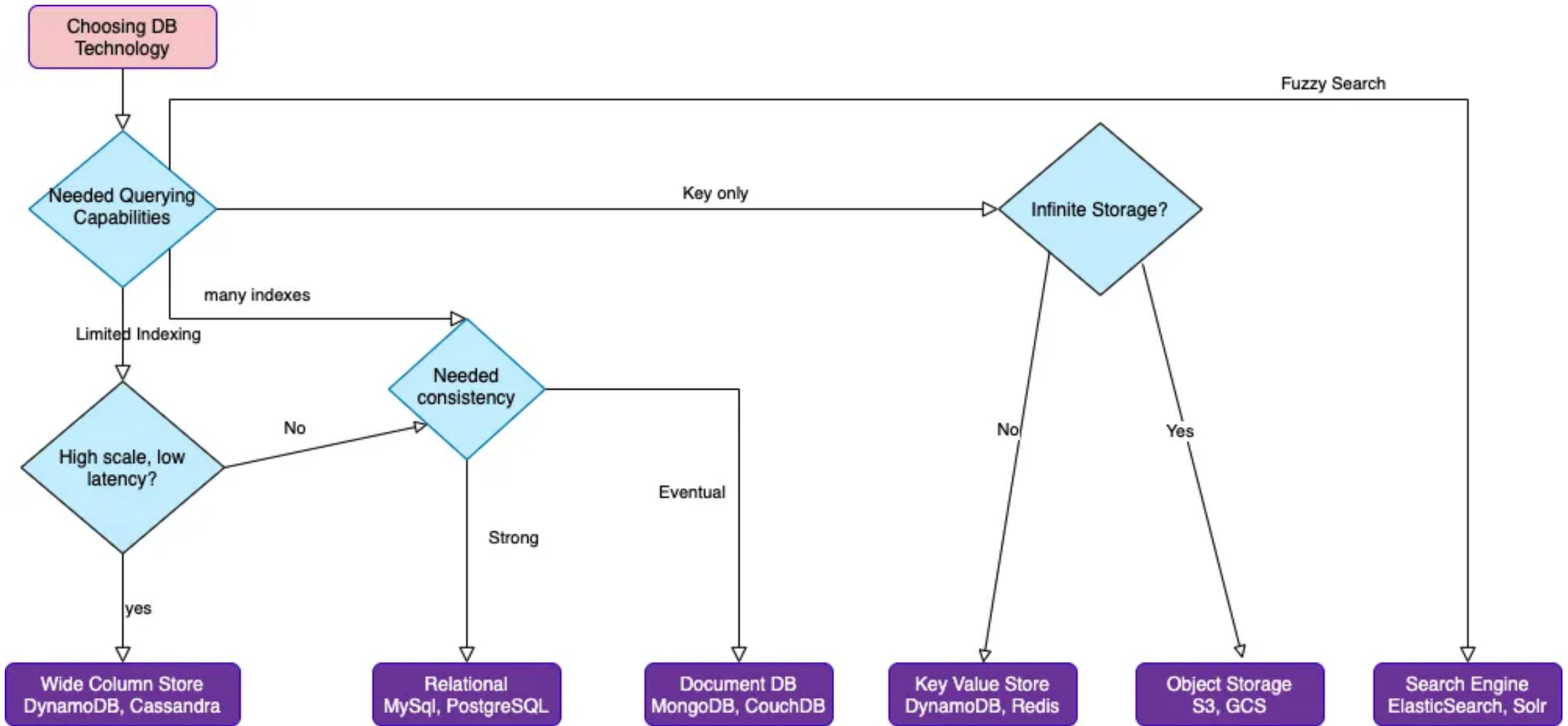
Nedostaci

- Upiti specifični za redove su sporiji
- Nema podršku za transakcije

PITANJA PRILIKOM DIZAJNIRANJA SISTEMA

Odgovorom na sledeća pitanja o tome šta nam je potrebno u sistemu olakšavamo izbor odgovarajuće baze:

- Struktura podataka (Schema)
 - Strogo definisana / fleksibilna
- Skaliranje (Scalability), broj korisnika
- Consistency vs Availability (Strong, eventual consistency)
- Veze (Relationships)
 - Potreba za Many-to-one I Many-to-many vezama
- Način pretrage (Query pattern)
 - Pretraga po ključu - kompleksni upiti
- Writes vs reads
 - Više čitanja ili više upisa
- Cena



DA LI JE DOVOLJNA JEDNA BAZA?

- Veliki sistemi često koriste više tipova baza
- Zahteva održavanje više klastera
- Multi-model baze:
 - ArangoDb (Document, Graph, Key-value)
 - MongoDb (Document, Graph)
 - PosgreSql podržava JSON

Ne zaboraviti Backup
podataka

Availability nikada nije 100%

—

DIZAJN INSTAGRAM SOCIJALNE MREŽE



FUNKCIONALNI ZAHTEVI

- Upload slika
- Pregled slika
- Praćenje korisnika (follow) - neograničeno
- Početna strana (news feed)
- Comments, likes

NE-FUNKCIONALNI ZAHTEVI

- Preko 2.3 milijarde korisnika
- Minimum latency za početnu stranu i pregled slika
- High Availability, Eventual Consistency?
- Reliability – slike koje uploadujemo ne smeju biti izgubljene
- Read-Heavy
- Efficient Storage – ne postoji ograničenje za broj slika

PROCENA KAPACITETA

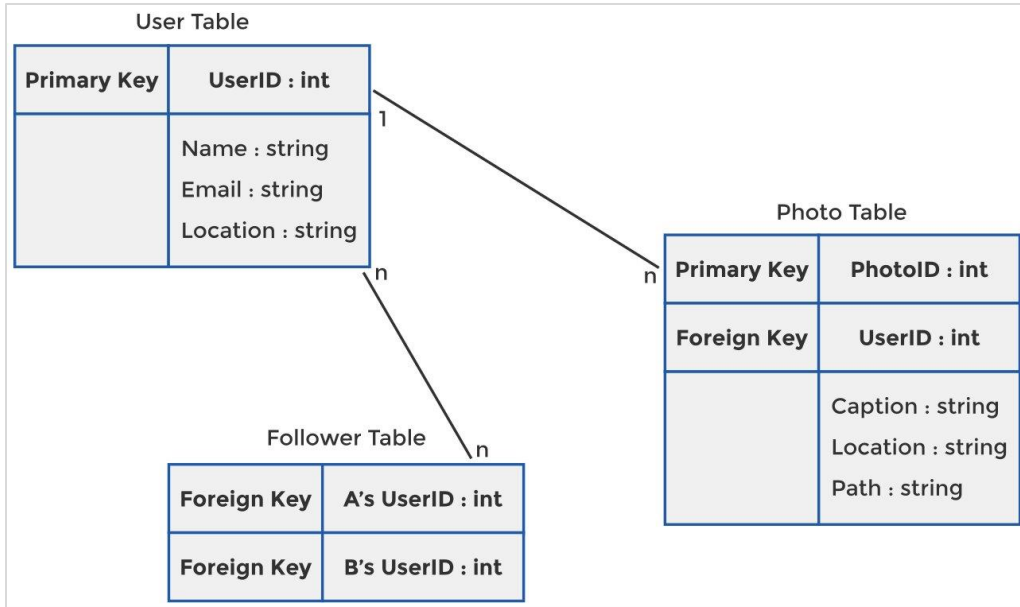
Pretpostavimo:

- 500 miliona aktivnih korisnika dnevno
- Dnevni upload slika je **100 miliona** – **1160/s** (100mil/86400)
- Ako je slika veličine **150KB**, trebaće nam dnevno: **15TB (100mil * 150KB)**
- Za godinu dana: **5475TB = 5.475PT (Petabyte)**
- Treba da čuvamo metadata za slike, komentare, lajkove, tagove...

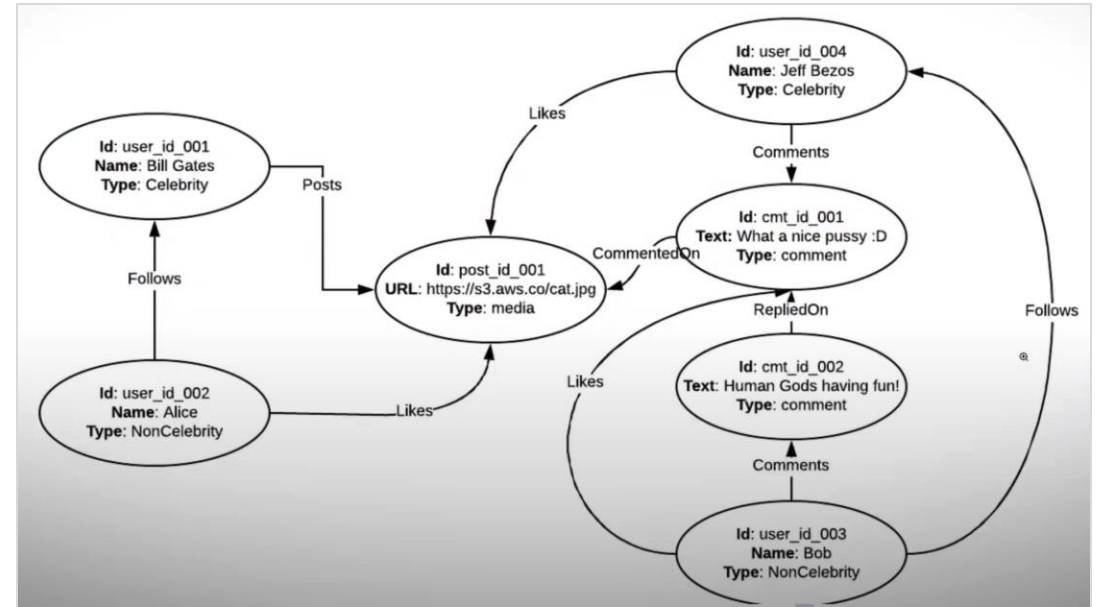
Gde čuvati slike?

- Object Storage (AWS S3, GCS)

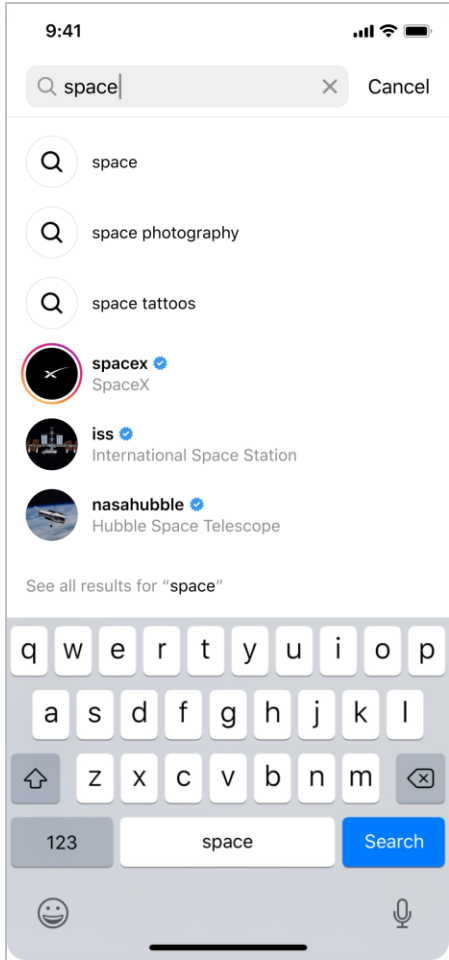
User data, photos, tags, comments, likes...



VS



Instagram koristi: PostgreSQL (12 extra-large memory instanci / sharded)



Search: hashtags, locations, users and media

Elasticsearch, custom solution

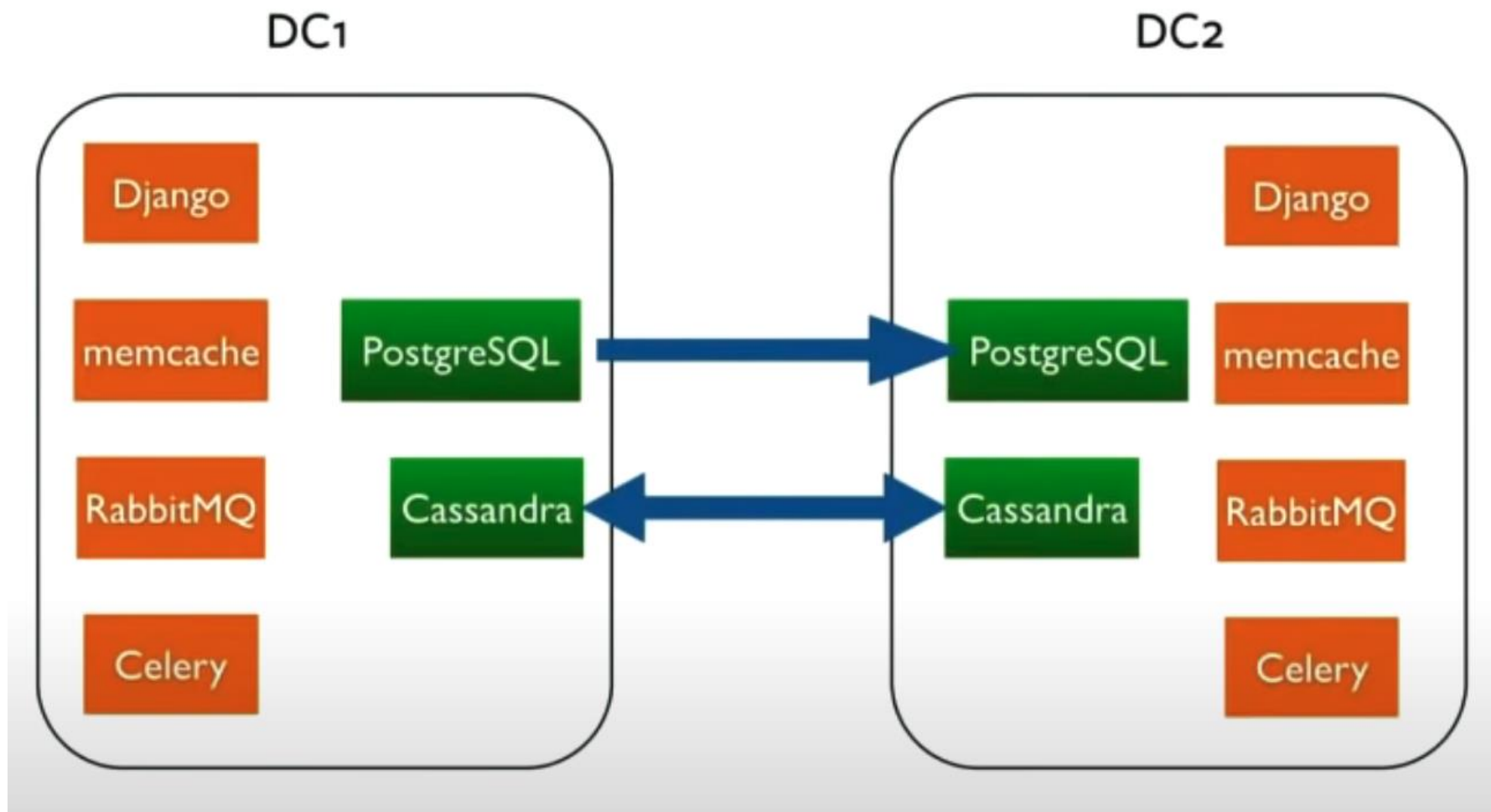


Pre-compute news feed

Latency hurts



Veća redundantnost -> veća sigurnost -> veća kompleksnost



KORISNI LINKOVI



<https://discord.com/blog/how-discord-stores-trillions-of-messages>

- <https://instagram-engineering.com>
- <https://www.youtube.com/@hnasr>
- Martin Kleppmann: Designing Data Intensive Applications