- Why build system?
• who are users?
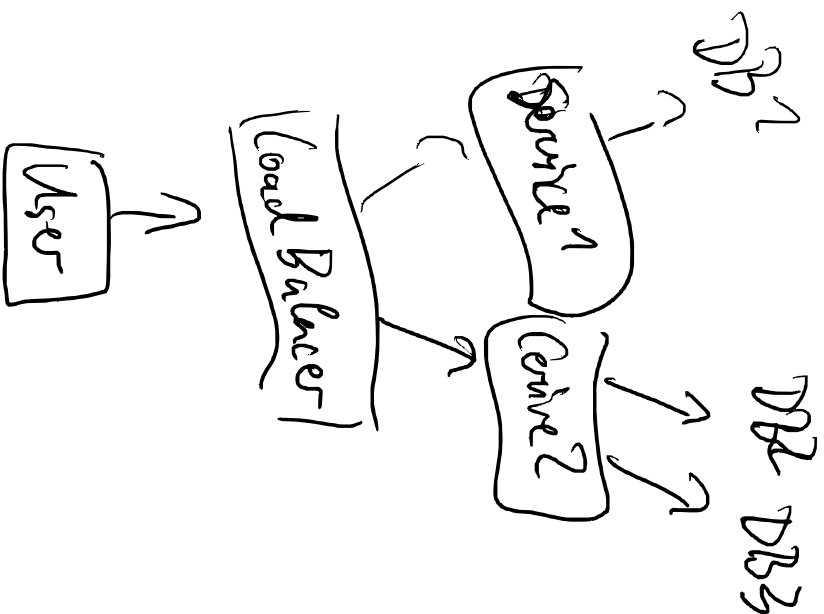• what features?

} Get Top 5
features

=> Specify N-F-Requirements => Capacity

(=> Bottlenecks?)

1. Users => API [=> back-end]
↳ REST
(microservices)
↳ servers?

User

Load Balancer

Service 1

Service 2

DB1   DB2   DB3

Session Pricing → don't change

OUT OF SCOPE → Webe zwische Stopps ?!

○ Antbuilding on Palmer Arenas

(○ Session spec of Pluggreize ?!) → Wi,e Probe?

○ Flights (From →> To) [○ in internate steps?]

○ Bottlenecks → many fetch requests / lot of requests (users online)

⇒ OVERBOOKING?

↳ Increase "computational" power every hour for few minutes

↳ Cache

↳ Load Balancer

Reads higher than writes (=> less bookings)

⇒ DB → Leader — Follower

Services

→ Data Ingestion (→ Process flight data from airlines)

→ Reservation Service (→ Manages reservations & enforces consistency)
   (=> Cache Prices?)

→ Search Service (→ Find searched flight)

External APIs

→ Websocket/batch-job stream periodic updates

(=> Payment API → service?!)

Partitioning
[→ AIRPORT
 → Fluggesellschaft]

For writes

↗ high read-write ratio

Single-leader, potentially
Multi-leader
   ↳ complex vs speed?
      → consensus speed

Search Service

Flight DB

Flight DB

Flights DB

External Flight Updates Service

Writes