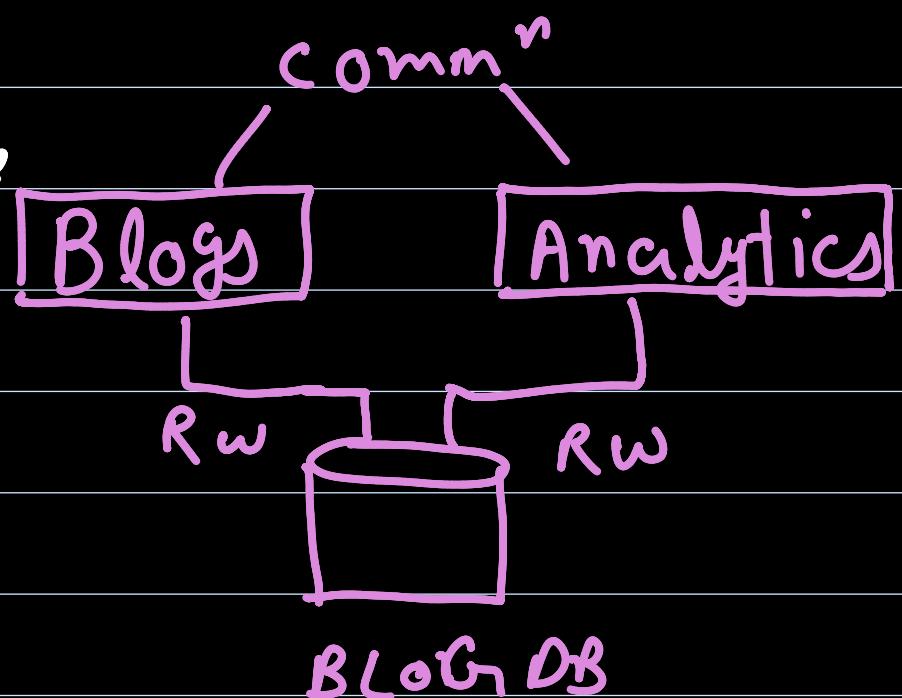


SHARED DB PATTERN

One of the simplest way to integrate a couple of microservices is to let them Share a database



Advantages

- No inter-service commⁿ
- No network hopping / latency
- Faster development and deployment times
- Simpler operations

Challenges

1 External Parties get internal details

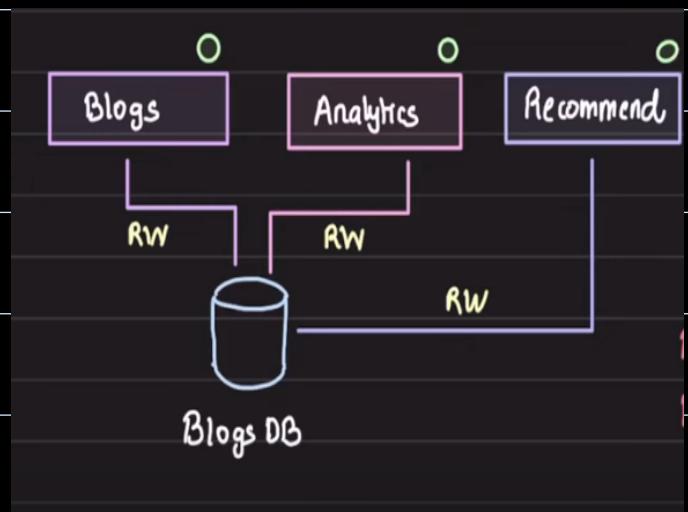
When using this pattern a lot of information is being shared with external parties. For eg:- In above eg, we share following things

- Database schema
- Design decisions
 - soft / hard delete
 - normalization
 - redundancy

Now let's say if the Blog team wants to change the DB or schema, then they also needs to inform this to analytics service. Leading to

Tight Coupling. And interdependency increases.

2 Sharding DB = Sharding Business Logic



lets say for some purpose Blogs gets data from tables T_1, T_2, T_3, T_4 . Now the Analytics and Recommendⁿ service also needs to do some thing for their purpose.

This brings in sharing of business logic. Now let's say the Blog service changes its way to get data to T_1, T_2, T_3, T_4 . Now other 2 services also needs to change these. This way we lose cohesion (diff services in same services)

3. Deletion & Corruption of data .

Now all the services have R+W access & this can cause imp data to be deleted and may impact user. Thus we need to change ACLs

4 Abuse of DB

Sometimes it may happen that Services run a heavy query. Now if there are multiple such queries running, it may cause delay to other svcs

1. Sharing a DB is a Quick solution, when crunched on time



Doing this requires efforts and co-ordination from multiple teams

2. When Schema does not change often

Schema or business logic do not change often, so why create that unnecessary dependency

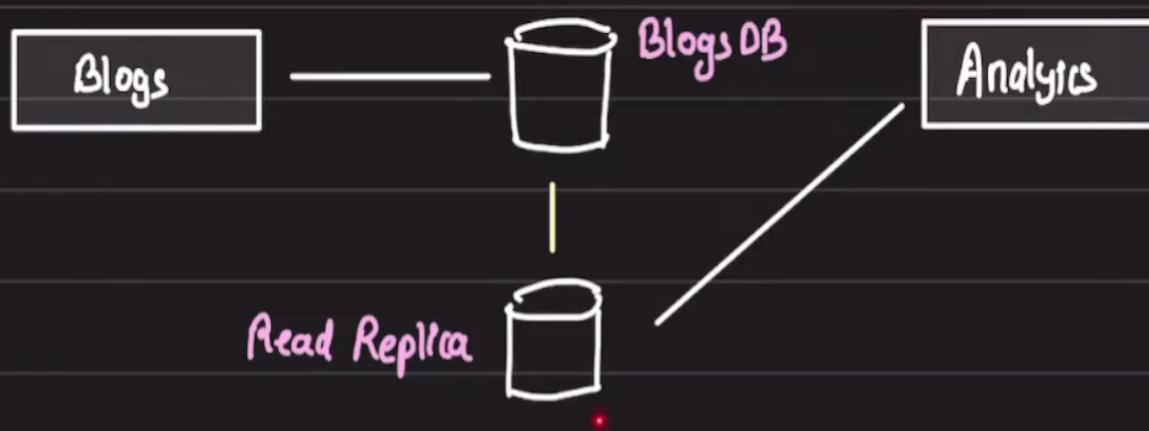
3. Read load can be moved to a Replica

Heavy analytics query can run on a separate replica of



3. Read load can be moved to a Replica

Heavy analytics query can run on a separate replica of the Blogs DB



This way dependent system does not put a high
read load on the main DB