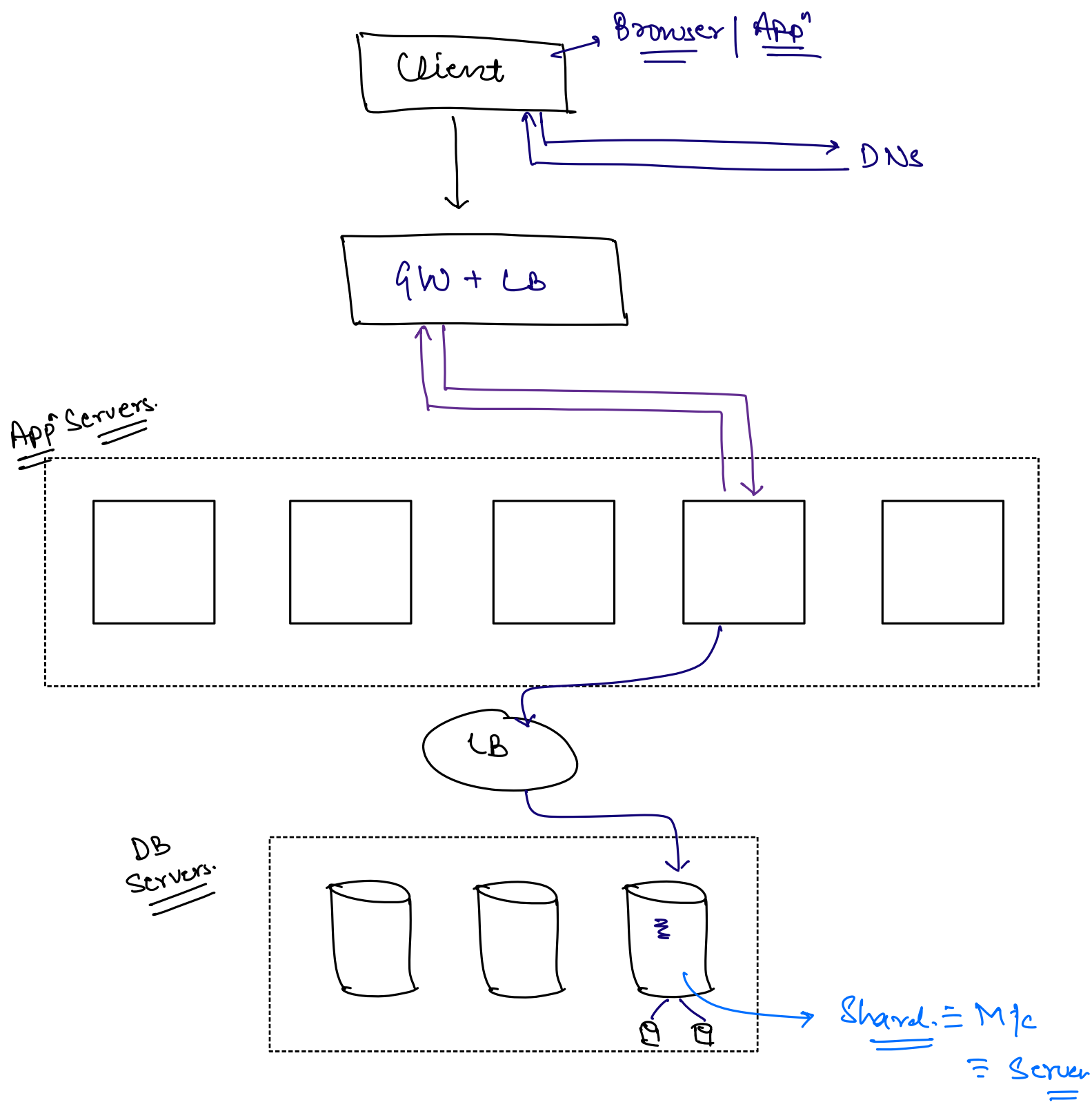


→ Recap.

→ Consistent Hashing

→ App server layer & storage layer should be separate.



## SHARDING.

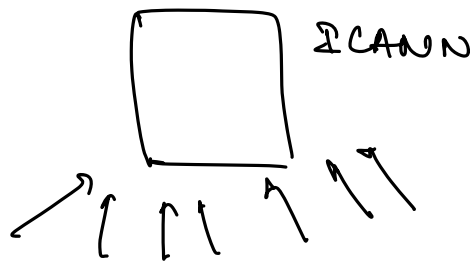
↳ Distributing data across multiple mysql.

⇒ To enable scaling of our App, we have decoupled the app servers and DB mysql but the tradeoff here is we have introduced an extra n/w hop & because of this overall latency of the system will take a hit.

App<sup>n</sup> Servers can be Stateful or Stateless depending on the usecase.

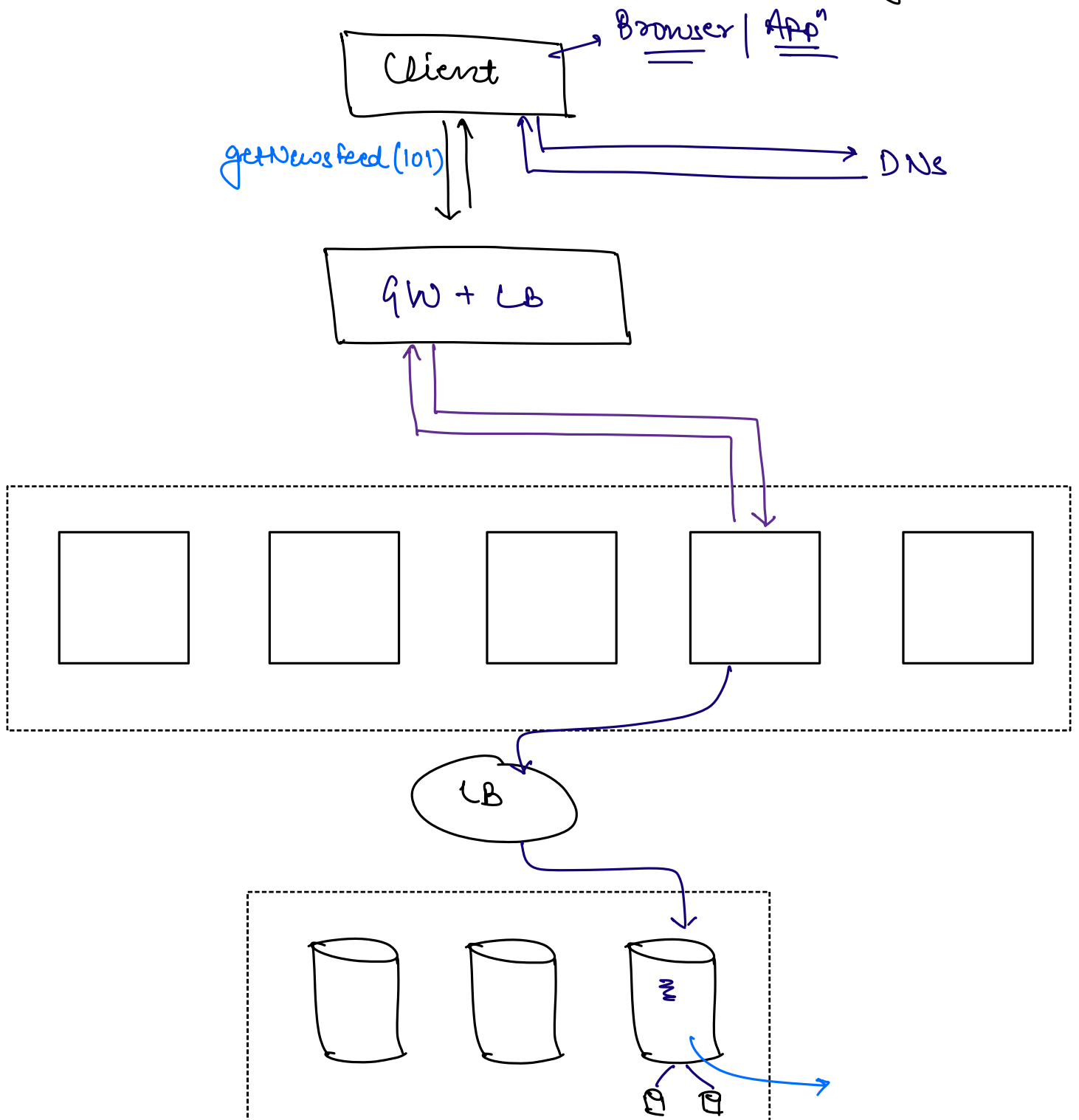
DB Servers should always be Stateful.

⇒ DNS.



Caching: Process of storing frequently accessed data at the place which is near to user to optimize the overall data access time.

I) Client Side Caching | Browser Caching



⇒ Pagination.

⇒

Chrome  
Browser.

mic

microsoft

microscope

michael jackson

microservice

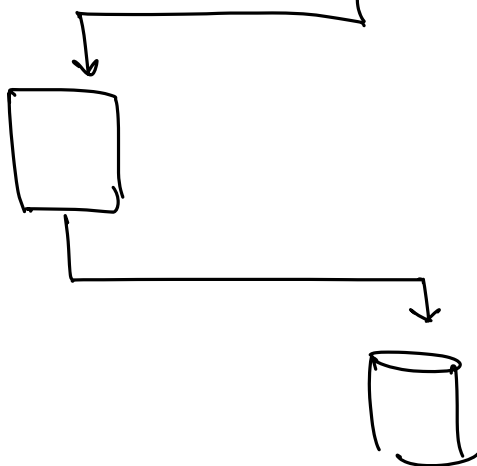
≡  
≡  
≡  
≡  
≡

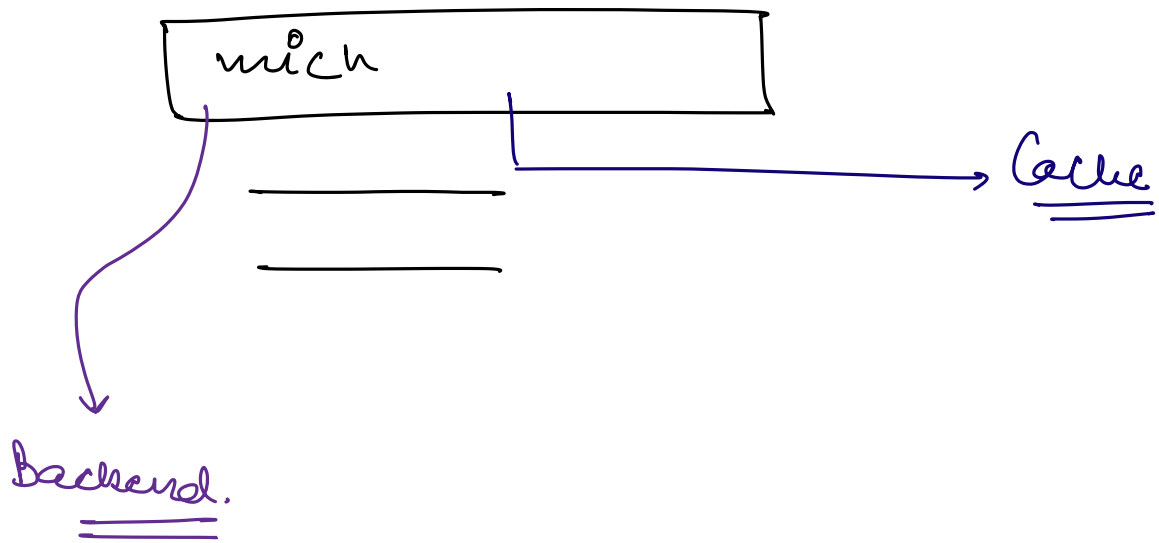
cached  
in Browser

Client

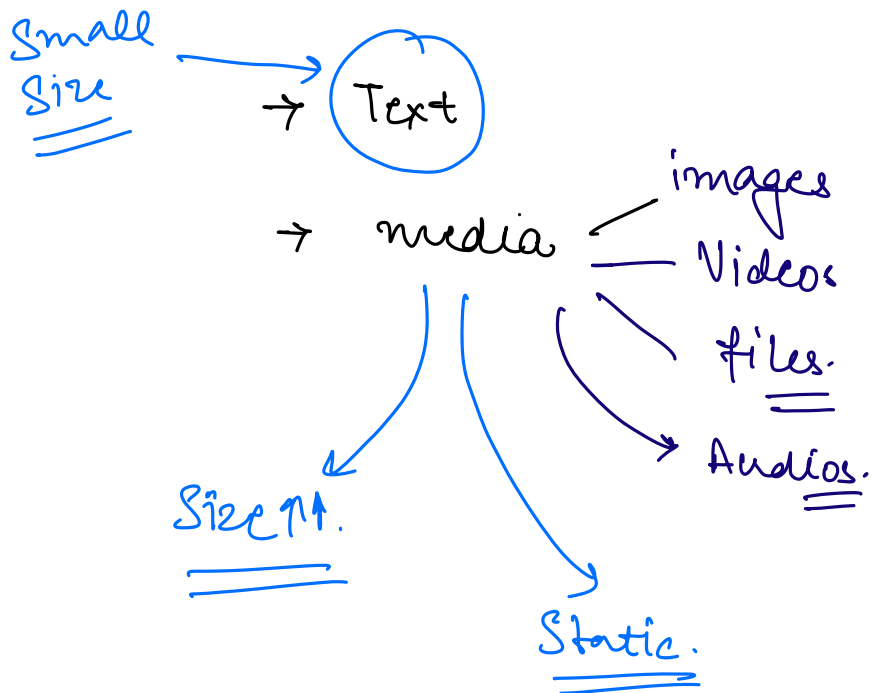
↑↓ get suggestions ("mic")

APIGW + LB

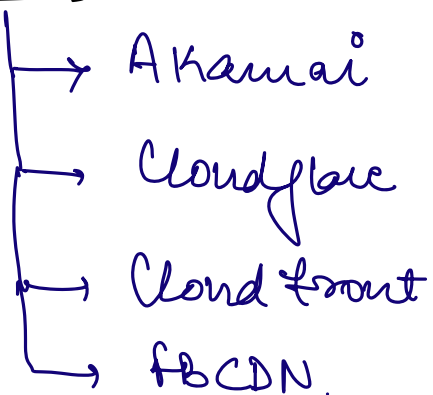




# Content Delivery Network.  $\equiv$  CDN.

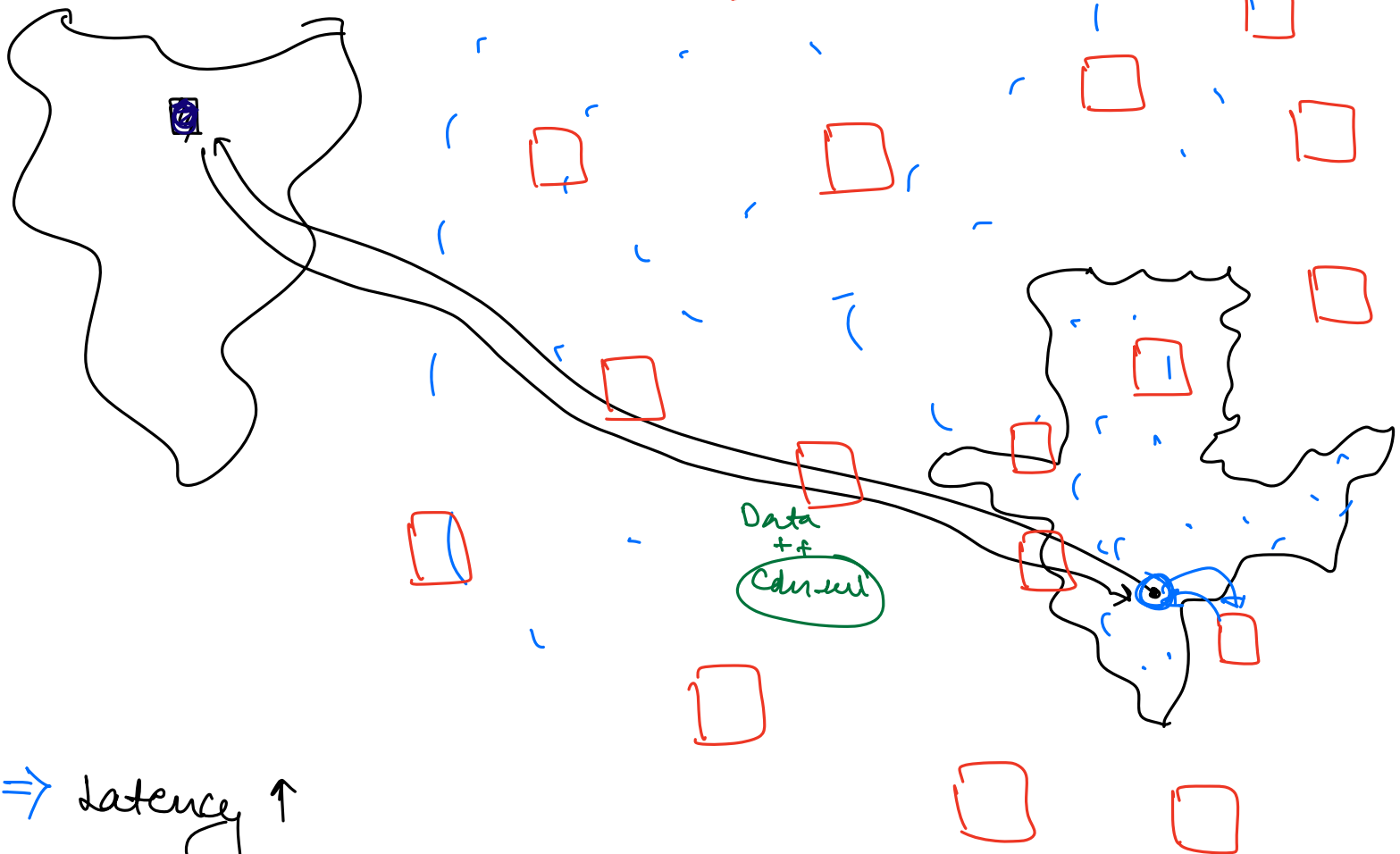


# CDN.



=

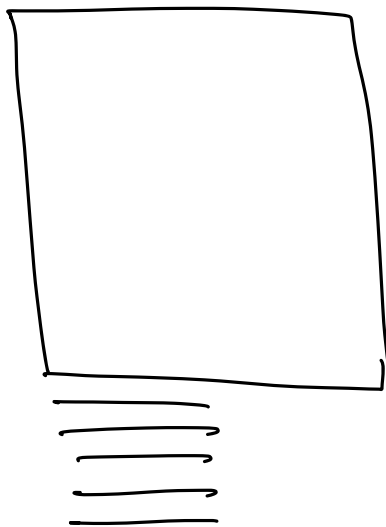
Us



⇒ Latency ↑

⇒ N/w congestion

⇒ App & DB server will get overloaded.



Virat Kohli

Client

Client (Raghu)

createPost(==)

getNewsfeed(701)

CDN  
URL

API GW + LB

Akemi

701

Virat Kohli

~~S3 url~~ cdn-url

url

posts

id	...	<del>S3 url</del>	cdn-url
1074	...		(x42)

File Storage |  
AWS S3 |  
Blob Storage  
==

(DB for Media)  
Content

=

⇒ CDN.

↳ Geographically distributed n/w of m/c  
to store frequently accessed data.

→ Edge m/c (or) Edge nodes (or)

Geo Nodes.

———— \* ————