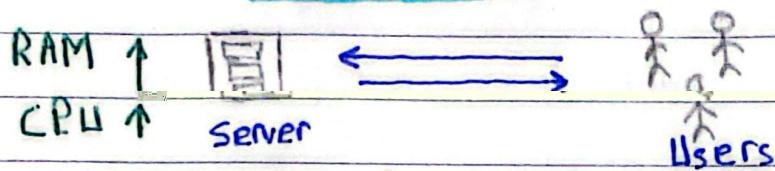


→ Todo - List:

- + System design [NeetCode]
- + Microservices [5 minor part]
- + 8 design patterns [NeetCode]

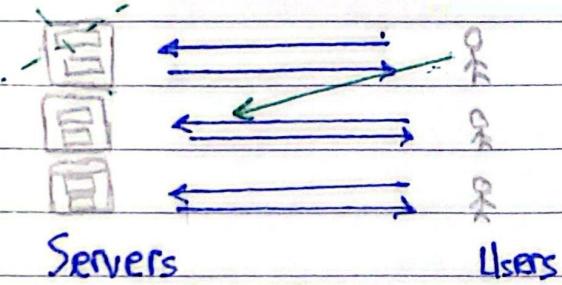
→ System Design:

→ Vertical Scaling:



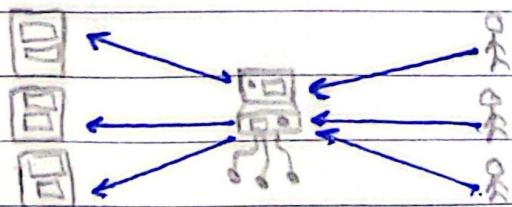
* vertical scaling consiste à améliorer les performances d'un serveur en augmentant sa puissance.

→ Horizontal Scaling:



* horizontal scaling consiste à améliorer les performances d'un système en ajoutant de nouveaux serveurs.

→ Load Balancers:



Servers Load Balancer Users

* Load Balancer répartit le trafic entre plusieurs serveurs pour éviter la surcharge et optimiser les performances des applications

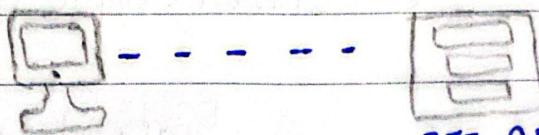
→ Content Delivery Networks:

- * Static Content : des éléments fixes sur un site web, comme des imgs, les fichiers CSS et JS
- * CDN : système de serveurs distribués géographiquement qui collaborent pour fournir du contenu web de manière rapide.

→ Caching:

- * Caching : stocke temporairement des données pour les récupérer plus rapidement.

→ IP Address:



216.58.216.238

255.255.255.255

- * IP : est une série de chiffres qui identifie de manière unique un appareil connecté à un réseau.

→ TCP/IP

- * TCP : Il assure le transfert fiable des données et la transmission des données entre les appareils.

→ DNS

- * DNS : système qui traduit les noms de domaine en adresse IP.

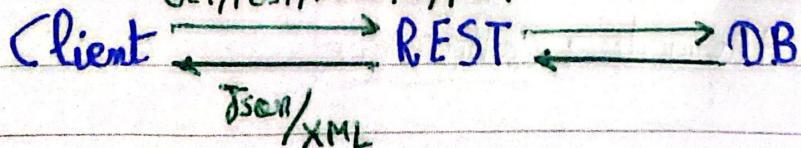
→ HTTP:

- * HTTP : Protocole de communication pour transférer des données.

- * Request → Header : info sur la requête
Body : les données

→ REST:

GET/POST/DELETE/PUT



→ Graph QL

* Graph QL : permet de récupérer toutes les données nécessaires en une seul requête et permet aux clients de spécifier les données dont ils ont besoin.

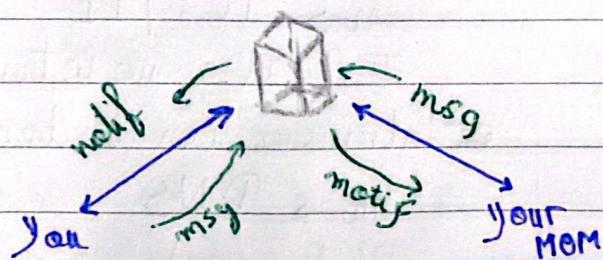
→ gRPC:

* gRPC : Framework open source by Google qui facilite la communication entre application réparties. Il utilise Protocol Buffer.

* Protocol Buffer : est un format de sérialisation efficace et simple pour transmettre les données

Protocol
Buffer >>>> Json

→ Websocket

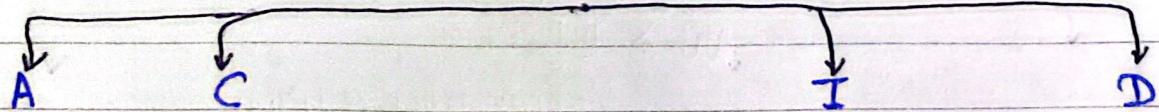


→ Bidirectionnel ; Temps réel ; Rés surcharge

→ SQL

→ ACID

* propriétés fondamentales d'une transaction dans une db

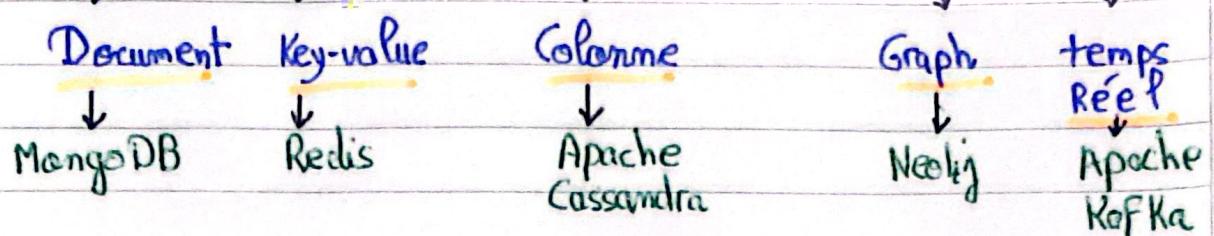


Atomicité ; Cohérence

isolation ; Durabilité

ACID ⇒ garantie la fiabilité des transactions

→ NOSQL



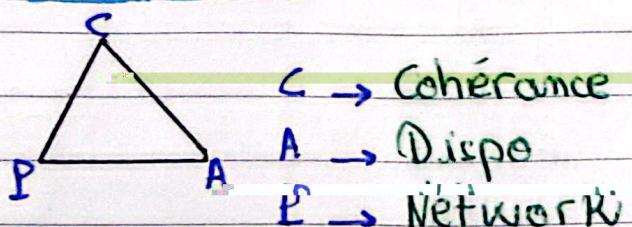
→ Sharding:

* Sharding : technique de gestion de base de données qui divise une base volumineuse en plusieurs morceaux appelés «shards» à fin de faciliter la gestion et améliorer la performance.

→ RéPLICATION:

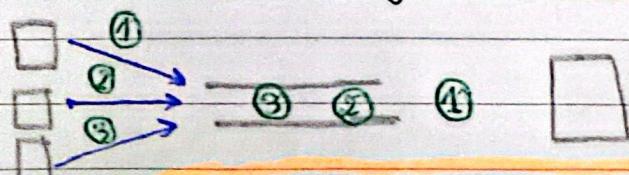
* RéPLICATION : consiste à reproduire les données d'une db dans plusieurs endroits → assurer la dispo et améliorer la perfo.

→ CAP Theorem:



→ un système ne peut garantir simultanément C, A et P.

→ Message Queues.



→ Facilitent la communication asynchrone entre les parties d'un system

→ stockage temp
→ transmission msg