**1st round**： user🡺 API gateway (Routing, Authentication, Authorization, rate limiting)

API (写明功能), 选择 microservices architecture, 一般都是REST API 架构的。 要不要：rate limiting 🡺

如何选择: SQL DB, data can be related? ACID property needed?

Non-SQL DB, 还有 要不要加 **Object Storage** (AWS S3), object storage不需要replicate 或者 sharding, 可以直接scaling

**2nd round**: Check Latency

加 Database Index; 加 Cache (Redis key-value DB), eviction policies: LRU or LFU

要不要加 CDN?

**3rd round**: how to scale:

讨论需不需要 divide API?(need api gateway);

Horizontal Scaling API server (need load balancer and consistency hashing or Rendezvous hashing)

Config: Scaling setup: CPU usage stays above 70%, we add 1 server; when CPU stays below 30%, remove 1 server.

set a **cooldown period**, like 5 minutes

Load Balancer: consistency Hashing or Rendezvous hashing

**4th round**: check Availability

Check 999 for SLA; Redundancy for DB (active or passive), Snapshot for DB?  
API gateway和 LB，云服务商会自动配置 redundancy，不需担心

要不要加：

Rate limiting 一般是加在 API gateway

log server (Time series DB) + Monitor to setup actions to do when trigger alert

Config: Scaling setup (K8s or Auto Scaling group); Dyna config