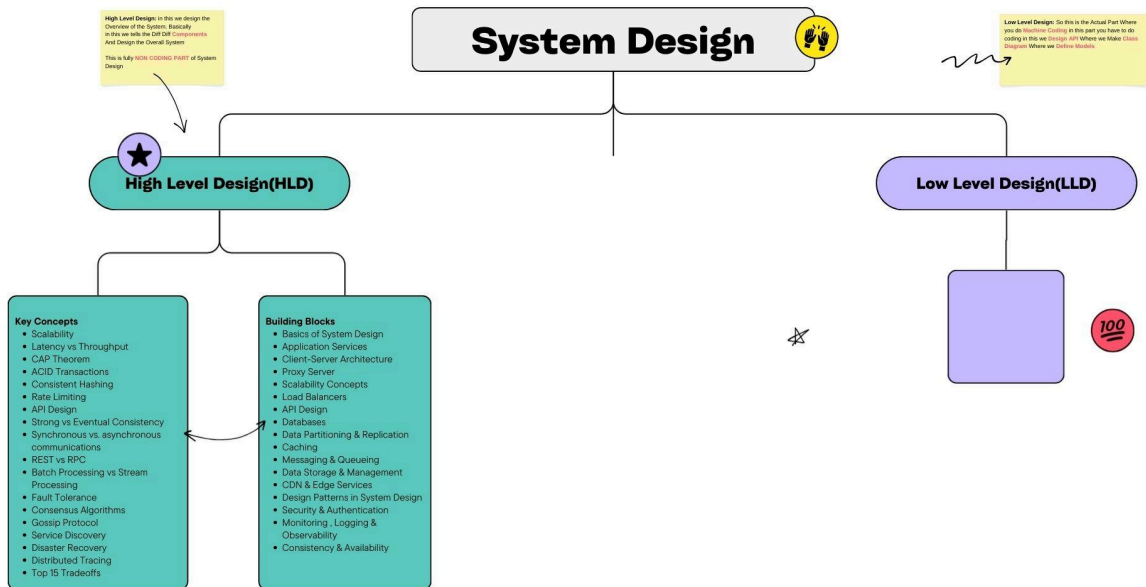


# System Design



# High Level Design

## Key Concepts:

- Scalability
- Latency vs Throughput
- CAP Theorem
- ACID Transactions
- Consistent Hashing
- Rate Limiting
- API Design
- Strong vs Eventual Consistency
- Synchronous vs. asynchronous communications
- REST vs RPC
- Batch Processing vs Stream Processing
- Fault Tolerance
- Consensus Algorithms
- Gossip Protocol
- Service Discovery
- Disaster Recovery
- Distributed Tracing
- Top 15 Tradeoffs

# Building Blocks

## 1. Basics of System Design

- What is system design?
- HLD vs LLD
- Functional vs Non-functional requirements
- Latency, throughput, availability

## 2. Scalability Concepts

- Horizontal vs Vertical scaling
- Load testing basics
- Caching overview (briefly introduce before deep dive)
- Load balancer overview

## 3. Load Balancers

- L4 vs L7 Load Balancing
- Reverse Proxy (Nginx, HAProxy)
- Consistent hashing
- Global Load Balancing (GSLB)

## 4. API Design

- RESTful APIs
- gRPC (optional advanced)
- Rate limiting
- API Versioning
- idempotency

## **5. Databases**

- SQL vs NoSQL
- Relational DBs: MySQL, PostgreSQL
- NoSQL DBs: MongoDB, Cassandra, Redis, DynamoDB
- Indexing, Joins, ACID
- CAP Theorem

## **6. Data Partitioning & Replication**

- Sharding
- Replication
- Partitioning Strategies
- Consistent Hashing (again, in DB context)

## **7. Caching**

- Caching strategies: write-through, write-around, write-behind
- Cache Invalidation
- Redis, Memcached
- CDN overview (Cloudflare, Akamai)

## **8. Messaging & Queueing**

- Message Brokers: Kafka, RabbitMQ, SQS
- Asynchronous processing
- Pub/Sub architecture
- Eventual consistency

## **9. Data Storage & Management**

- File, Block, Object storage
- Blob storage (AWS S3, Google Cloud)
- Data lakes vs Data warehouses

## **10. CDN & Edge Services**

- CDN architecture
- Caching static content
- Edge computing & latency reduction

## **11. Design Patterns in System Design**

- Microservices vs Monolith vs SOA
- Event-driven architecture
- CQRS
- Saga pattern
- Circuit breaker, API Gateway

## **12. Security & Authentication**

- OAuth 2.0, JWT
- SSL/TLS, HTTPS
- API Gateways for security
- Rate Limiting

## **13. Monitoring, Logging & Observability**

- Metrics vs Logs vs Traces
- Prometheus, Grafana
- ELK Stack
- Health checks, alerts

## **14. Consistency & Availability**

- Strong vs Eventual consistency
- Quorum-based systems
- Paxos, Raft (conceptually)
- BASE vs ACID

