

# High-Level Design Checklist

Architecture validation points for system design interviews.

---

## Core Components

- Clients, entry points, and main request path are clear.
- Load balancer plan defined (L4/L7, health checks).
- App tier is stateless and horizontally scalable.
- Datastore + cache responsibilities are explicit (what lives where).

## Data Flow Validation

- Write flow (create) is end-to-end and idempotent.
- Read flow (redirect) includes cache hit/miss behavior.
- Key entities + indexes match query patterns.
- Failure behavior on the hot path is stated (timeouts, retries, fallback).

## Scalability Readiness

- Bottleneck-first scaling path: 1x → 10x → 100x.
- DB scaling approach: replicas, partitioning/sharding (as needed).
- Cache scaling: eviction policy + sizing assumptions.
- Async where appropriate (queue/workers) to protect latency.

## Technology Choices Justification

- Major choices tie to constraints (latency, availability, cost).
- Consistency model stated (strong vs eventual) per feature.
- Observability included for critical flows (metrics, logs, tracing).
- Security basics acknowledged if in scope (auth, rate limits, abuse).

## Abstraction Level Check

- Diagram shows only the most important 4–6 components (no clutter).
- Interfaces are clear (APIs + inputs/outputs), not implementation detail.
- One deep-dive uses: Problem → Options → Trade-offs → Decision.
- Assumptions are explicit and consistent (scale, retention, SLA).

Tip: If you can't check an item confidently, clarify assumptions or simplify the design.