System requirements - Summary

- Identifying both functional and non-functional requirements is an integral part of any system design.
- Write requirements on the board.
- Use non-functional requirements as a tiebreaker.
- To identify functional requirements, we start with the customers and move backwards.
- Be ready to convert identified use cases into API.

System requirements - Summary

Availability	Scalability	Performance	Durability	Consistency
System uptime, the percentage of time the system has been working and available.	The property of a system to handle a growing load.	The time required to process something and/or the rate at which something is processed.	Once data is successfully submitted to the system, it is not lost.	Consistency of data across distributed copies.
• high availability • SLO and SLA • fault tolerance • resilience • chaos engineering • reliability	• vertical and horizontal scaling • elasticity • autoscaling	• latency • percentiles • throughput • bandwidth	• backup • RAID • replication • data corruption and checksum	• strong consistency • weak consistency • consistency model • linearizability • CAP • eventual consistency • monotonic reads • read-your-writes • consistent prefix reads

System requirements - Summary

Maintainability	Security	Cost
The ease with which a product can be maintained.	Degree to which the system protects against threats.	How to design systems with the most effective use of resources.
• failure modes and mitigations • monitoring • testing • deployment	• CIA triad • identity and permissions management • infrastructure protection • data protection	• engineering cost • maintenance cost • hardware cost • software cost