High Level Design - [Distributed Systems]

Monday, April 15, 2024 10:55 AM

High Level System Design - COM [Complexity, Overload and Mishap]

Is a mechanism to lay down the architecture of system in high level for example, How to tackle all the high level problems for example Overload, Mishap and Complexity.
Here are the solutions to think around.

High Level Design of the LIFT?

So the problem we are solving is to make sure, we have LIFT in our building that can handle both people and goods and both can go from one level of building to another level of building.
Now the First problem we have to solve is think around the
Is there any other item other than people or goods to think of? If yes then separate the concerns by making multiple LIFTS.

What will happen if you have those days when there is an event in the building and many people show up?
This answer will define the budget we have and capacity of single LIFT along with how many LIFTS we can build more?
Vertical / Horizontal Scaling.

How LIFT will deal when there is an failure
a. Failure in the electricity or any **technical fault**?
i. In such case having **more lift** can solve our purpose of moving the people/goods from one level to another level.

What if there Electricity **failure** in the City?
Have a Backup, So Back up in this case is **stairs case**, see solution to the problem is making sure, LIFT is helping move people and good from one location to another.

CRON Job - How will we handle the overload situation, for example we can put a board in front of the LIFT that states that LIFT can only be **used for heavy goods after 6 pm or before 8 am.**

Caching - How can we solve the problem of person standing outside of the lift wait very less for lift to open?
Keep the LIFT close to the level many people are located, it's more of probability.

Here are the key things you consider while designing the High Level systems.

1. Vertical Scaling.
2. Horizontal Scaling.
3. Microservice Architecture.
4. CRON Jobs.
5. Caching.
6. Distributed System (Partitions).
7. Load Balancer.
8. Decoupling the concerns.
9. Logging for better later analysis.
10. **Extensible** - You are able to extend any system.

Low Level System Design

1. Low level design is actual implementation of those individual high level components

References

System Design Primer ⭐: How to start with distributed systems?