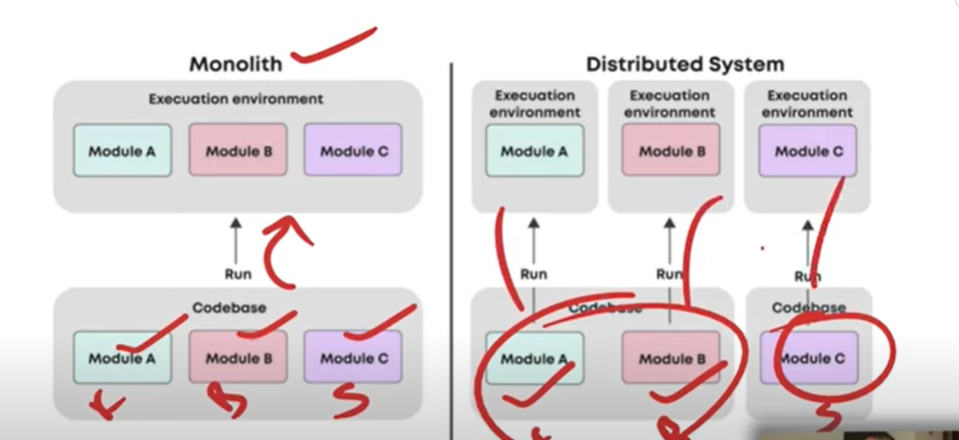
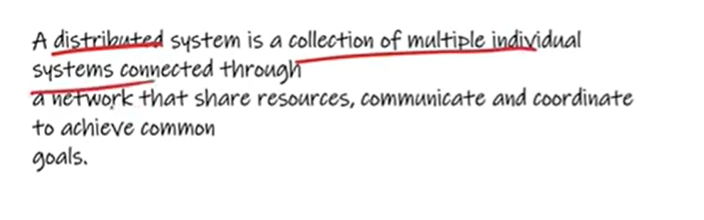
# Distributed Systems

Codebases are different, deployments are different, execution environment same.





There are different modules that are connected with each other.

Suppose we took 5 machines and distributed 2 GB data on each one making 10 GB data. If one machine gets fail, others will run. Single point of Failure is (spof) eliminated. In order to prevent data loss of machine, we have copies of each machine (replica/copy can be either 2,3,4) like each machine having 3 copies. If machine gets fail, control transfer to replica.

# Advantages:

* Scalable
* Low Latency (servers can be placed at different places)

# Disadvantages:

* Difficult to secure
* Complex, More nodes, more time, more resources, more knowledge.  
  Message lost risk
* More management

# Latency:

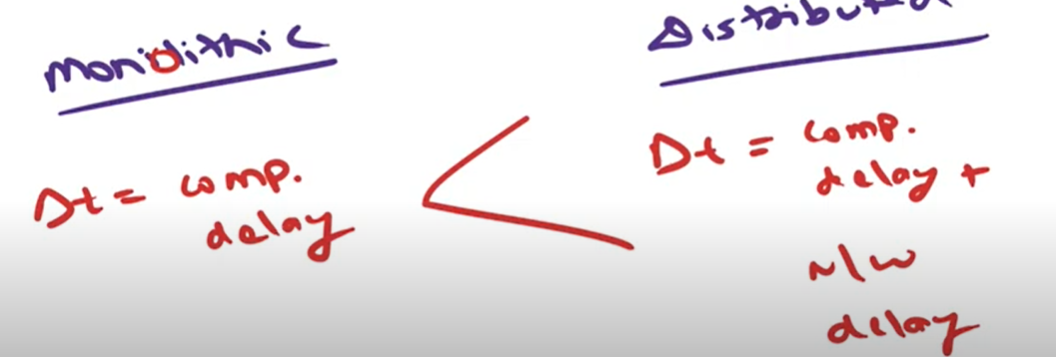
We type facebook.com and request to access. The time taken by client to take our request to server and then comeback to us that cycle is basically Time taken. Time taken for requestion as

T1 -> client to server, T2 -> server to client, T3 -> Computational Time (searching user account)

It is network delay + computational delay.

Network delay -> Request gone + receive time

Computation delay -> searching user for authentication, post etc.



## Reduce Latency:

1. Cache (most frequently data access)
2. CDN (content delivery network) (used for static data from server -> data from usa)

(Tooking data from USA into neighbour server)

Caching can be set on server

CDN can be set geographically

Upgrading hardware is third method

1. 