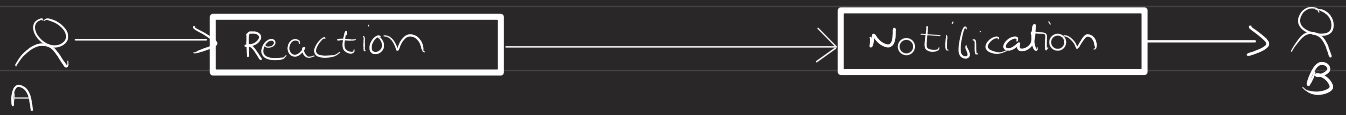


Sync Vs Asyn Communication

Let's take an example



We have a reaction service and whenever a user puts a reaction it sends the notification to other user.

Now we have 2 scenarios in here :-

1 Monolith

In this case we don't have much problem because of following reasons :-

- 1 It is a function call away
- 2 It is guaranteed to work

2 Distributed Systems

Here things become interesting due to the following scenarios:

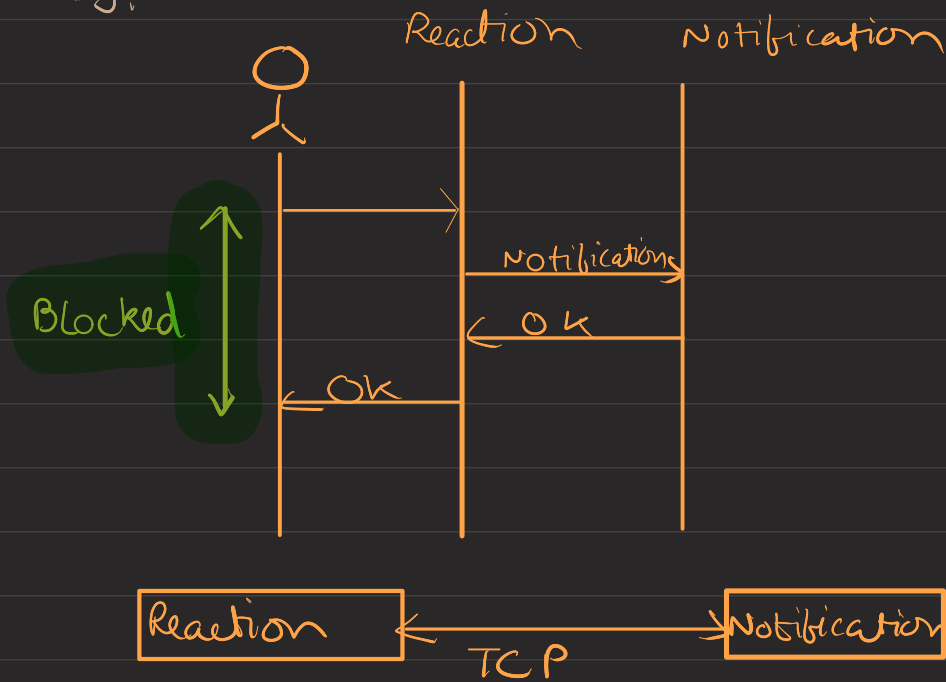
- 1 One of the services may be down
- 2 The receiving service is getting high traffic
- 3 The receiving service might even not get data

For this we have 2 modes of communication synchronous & asynchronous.

Synchronous Commⁿ

In this form of Commⁿ, the 2 services are in direct contact over a reliable networking protocol

Eg:-



In the above example there is a TCP commⁿ b/w the 2 services. Whenever the reaction service is invoked, it invokes Notification service and this returns ACK. After it ACKs, reaction service ACKs the user.

But here as you can see, there is a small catch of blocking. We will dig into this further.

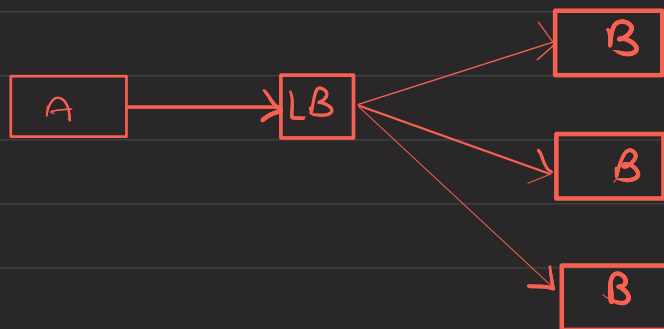
Generally, the commⁿ protocol is HTTP based like REST, RPC, GraphQL.

Advantages of Synchronous Commⁿ

- 1 Commⁿ happens in real time
- 2 Its simple and intuitive.

Disadvantages of Synchronous Commⁿ

- 1 Caller is blocked until the response is received
↳ may span to ms / s / min
- 2 Servers need to be proactively provisioned to handle load



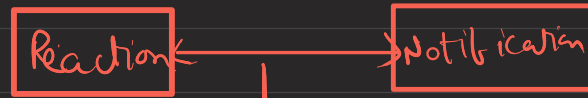
let's say if B is already being overwhelmed and more load comes from A. Then if we don't have provisioning for more servers, system will slow down and latency will ↑

3 Cascading of Failures



If D fails C will get affected and in turn B & A will also get affected.

4 It generates strong coupling



versioning
strong contract
backward compatibility

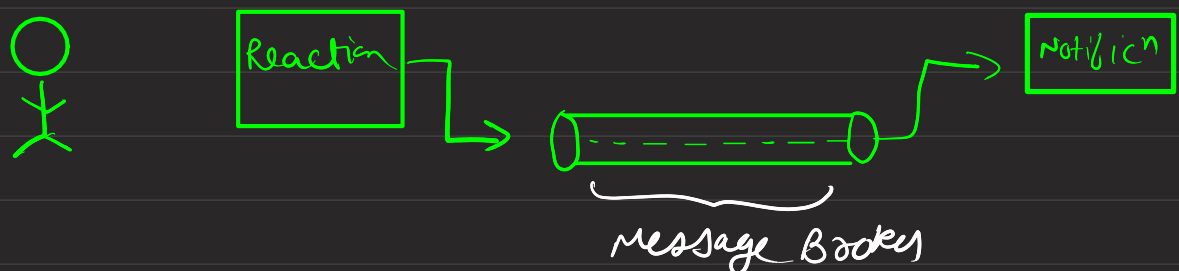
→ This bond / connection right here introduces strong coupling b/w 2 & both of them need to know about changes in other

When to use Synchronous Commⁿ

- 1 When there is no other option left - Eg - DB queries, API resp
- 2 When the user interaction is realtime. Eg - Chat, checkout
- 3 When it will take relatively less time to compute & respond.

Asynchronous Commⁿ

In asynchronous commⁿ the calling service send the message request to a broker which eventually gives it to receiver.



Above, whenever a user reacts to the post, the reaction service will send the data to message broker, and then this data until reaches to notification service, will remain in message broker.

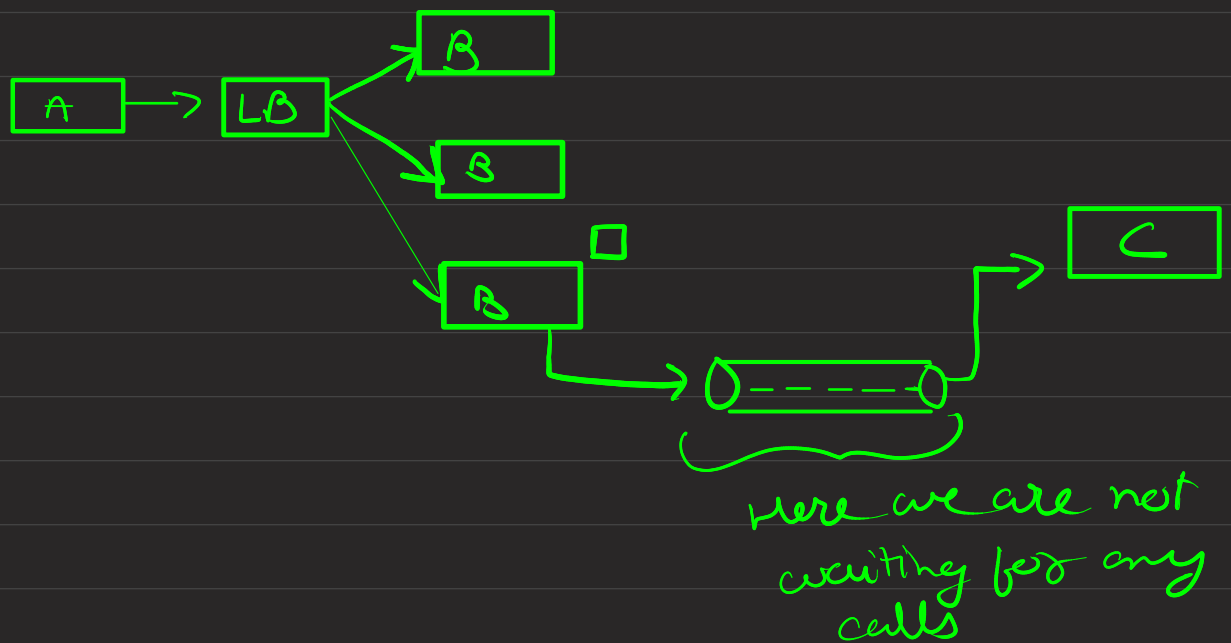
Advantages

- 1 There is less delay on the user facing service after sending data to broker



- 2 No need to pro-actively scale the servers. As we can have a monitoring on broker for scaling the server
- 3 Services are truly decoupled. As there is no constant dependency.

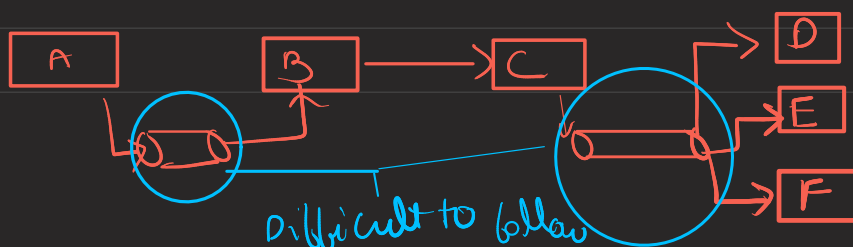
4 no need of extra load Balancer



5 No request loss or data loss as the messages are maintained in broker until recipient service ACKs

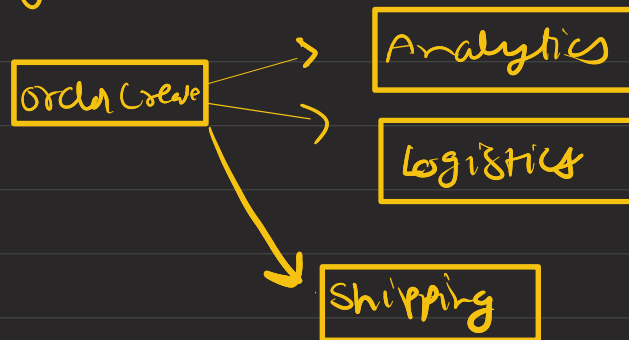
Disadvantages

- 1 Eventual Consistency:- Since the recipient services take up their own time to process request, not ideal for real-time commⁿ
- 2 Broker SPOF:- We need to ensure that message broker is up and running.
- 3 Harder to track flow of commⁿ



When to use Async Commⁿ

- 1 When delay is OK - Eg - notification, reporting, analytics
- 2 When there are long running processes - Eg - provisioning server, order tracking, DB backup
- 3 When many services are related to one event



- 4 When it is okay to allow failures - Eg - notificⁿ failed, retry