Agenda:

1) What is a monolith.

2) Advantages & Disadvantages of Monolith.

3) What is a Microservice.

4) Advantages & Disadvantages of a Microservice.

5) Communication b/w Microservices.

6) When to use Monolith vs Microservice.

———————— * ————————

Sachin.   Shipkart.com

(2008.) ←

⟹ Ecommerce.

→ Product Service   → Payment

→ Authentication.   → Logistics

→ Inventory   → Notification.

→ User Service

⟹ frameworks.

⇒ PHP : Magento.

ROR : Spree.

⇒

2008.

10-20 orders/Day

⇒ Domain.

Magento/Spree ⇒ Single Server.

4GB.
100GB.

| Payment | Notⁿ |
| Order | Logist |
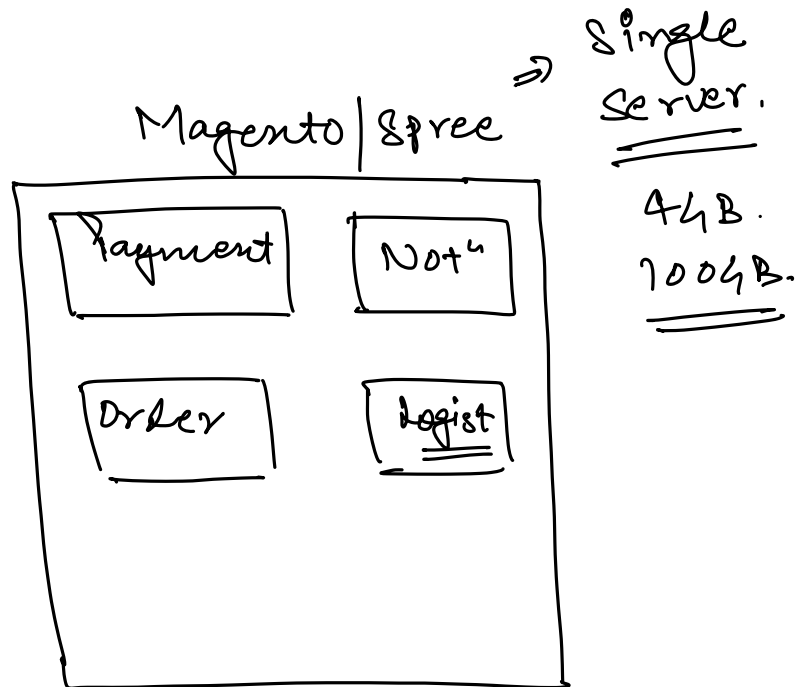
⇒ 2010.

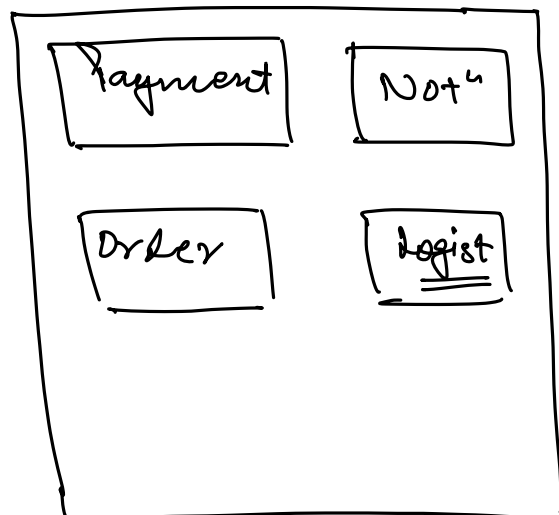↳ 2000 orders/Day.

2000| ⟶

Payment	Not^n
Order	Logist
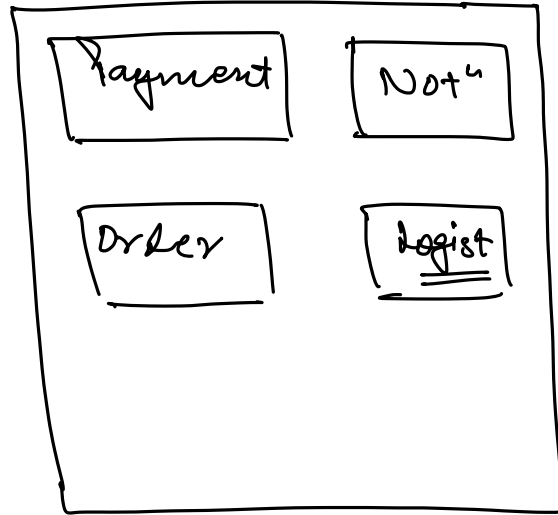
⇒ 16GB RAM.

⇒ 2013.

20000 orders/Day.

⇒ 20-30. Engineers.

Payment	Not^n
Order	Logist

⇒ 64 RAM.

2015.

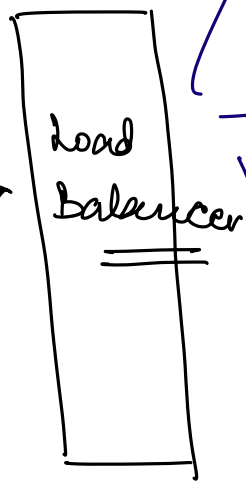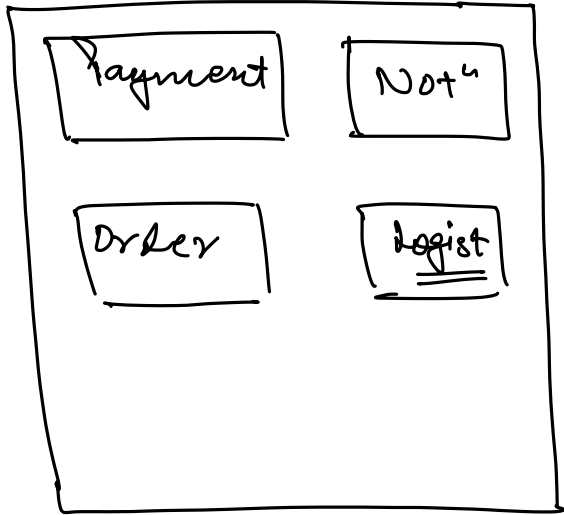$\Rightarrow$ 200,000 | Day.

3 servers.

2015

Load
Balancer

Payment   Not⁴

Order   Logist   4X

Payment   Not⁴

Order   Logist   4X

Payment   Not⁴
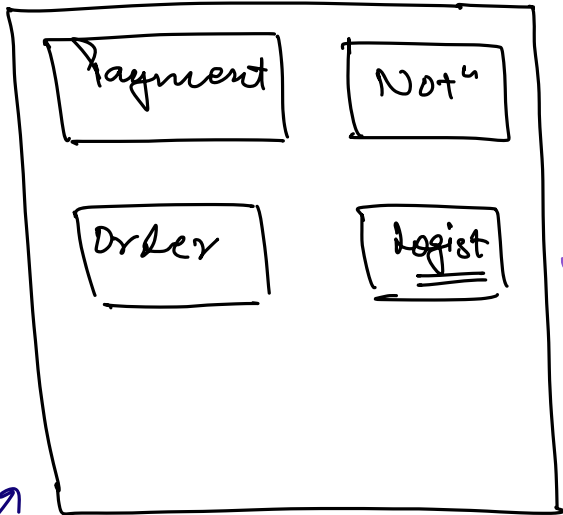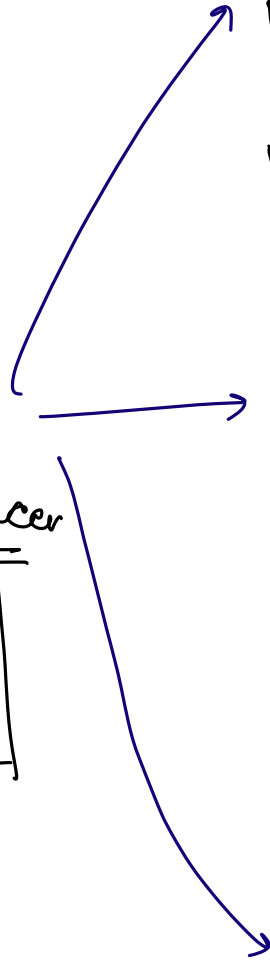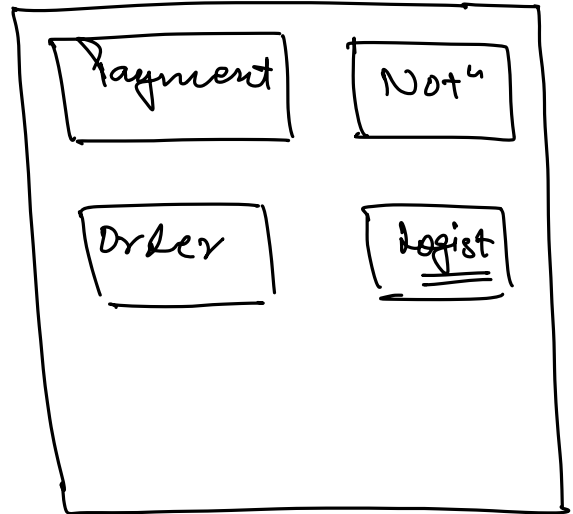
Order   Logist   4X.

2017.   500,000 orders/Day.

~100-200 Engineers.

Monolothic.

Payment | Not^n
Order | Logist   ~x. → 100GB.

Payment | Not^n
Order | Logist   ~x.

Load Balancer

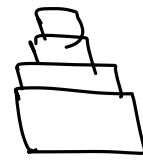Payment | Not^n
Order | Logist   ~x.

Payment | Not^n
Order | Logist   ~x.
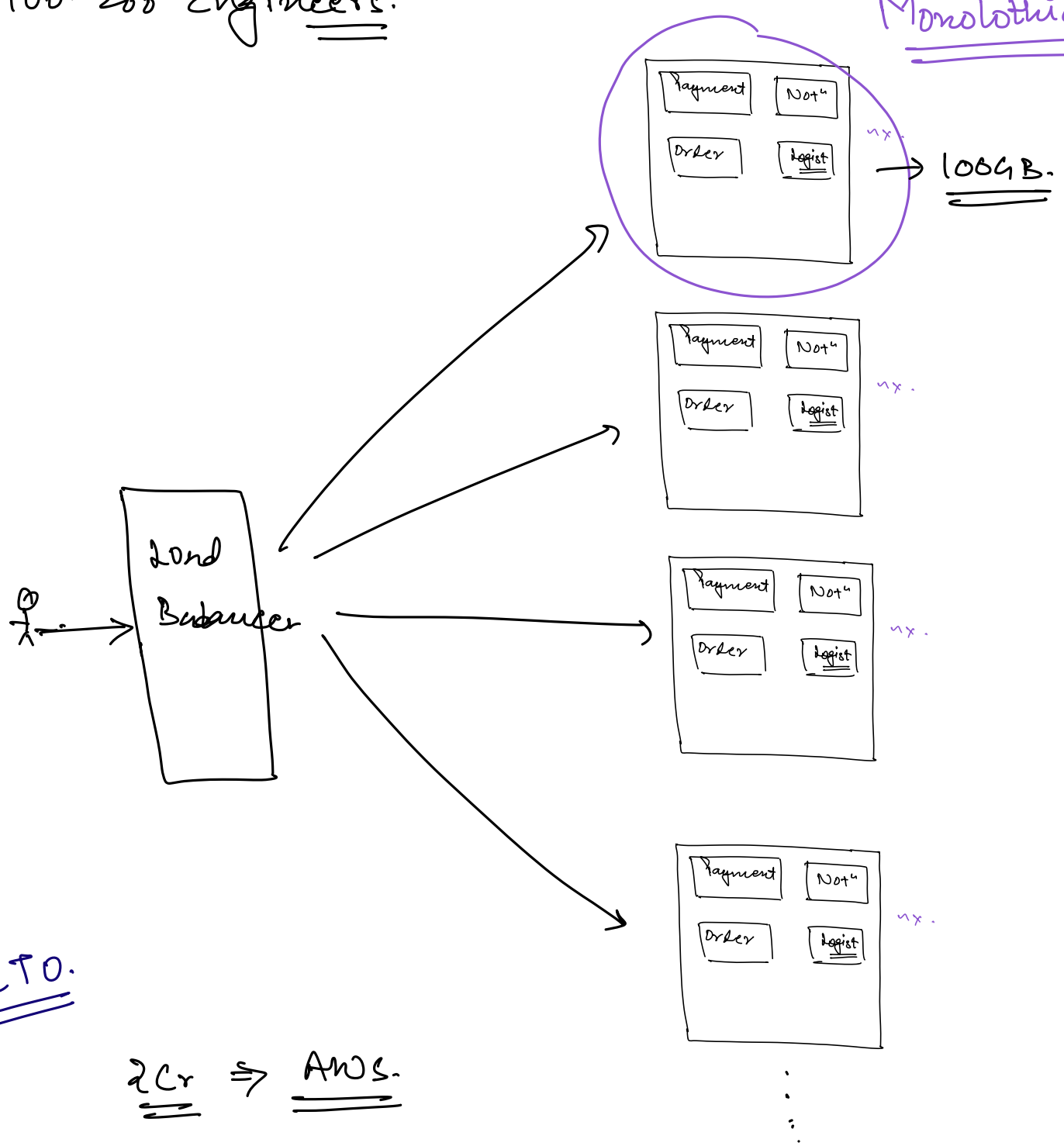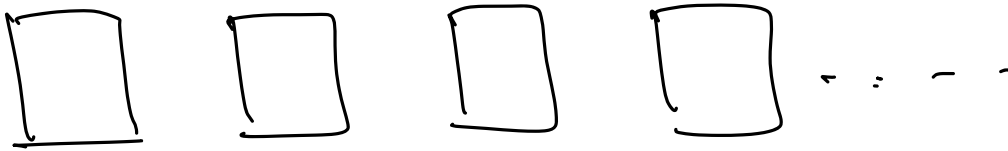
CTO.

2Cr ⇒ AWS.

Vertical Scaling

⇒ We increased the capacity of same m/c.

⇒ Can't be done infinitely

Horizontal Scaling

⇒ Add more servers to our Platform.

⇒ Can be done infinitely

⇒ We aren't utilising the resources efficiently.

Payment Service

AuthService

OrderService.

SearchService

Payment Service $\Rightarrow$ Low Scale
$\Rightarrow$ Moderate CPU

Search Service $\Rightarrow$ High Scale $\quad\Big\}$ 64 Cores -
$\Rightarrow$ High CPU. $\quad$ 128 GB RAM.

Authentication. $\Rightarrow$ High Scale $\Big\}$ 2 cores
$\quad$ Less CPU. $\quad$ 16 GB RAM.

$\Rightarrow$ Selective Scaling.

Advantages & DisAdvantages of Monolith.

Advantages:

1) Single Deployment.
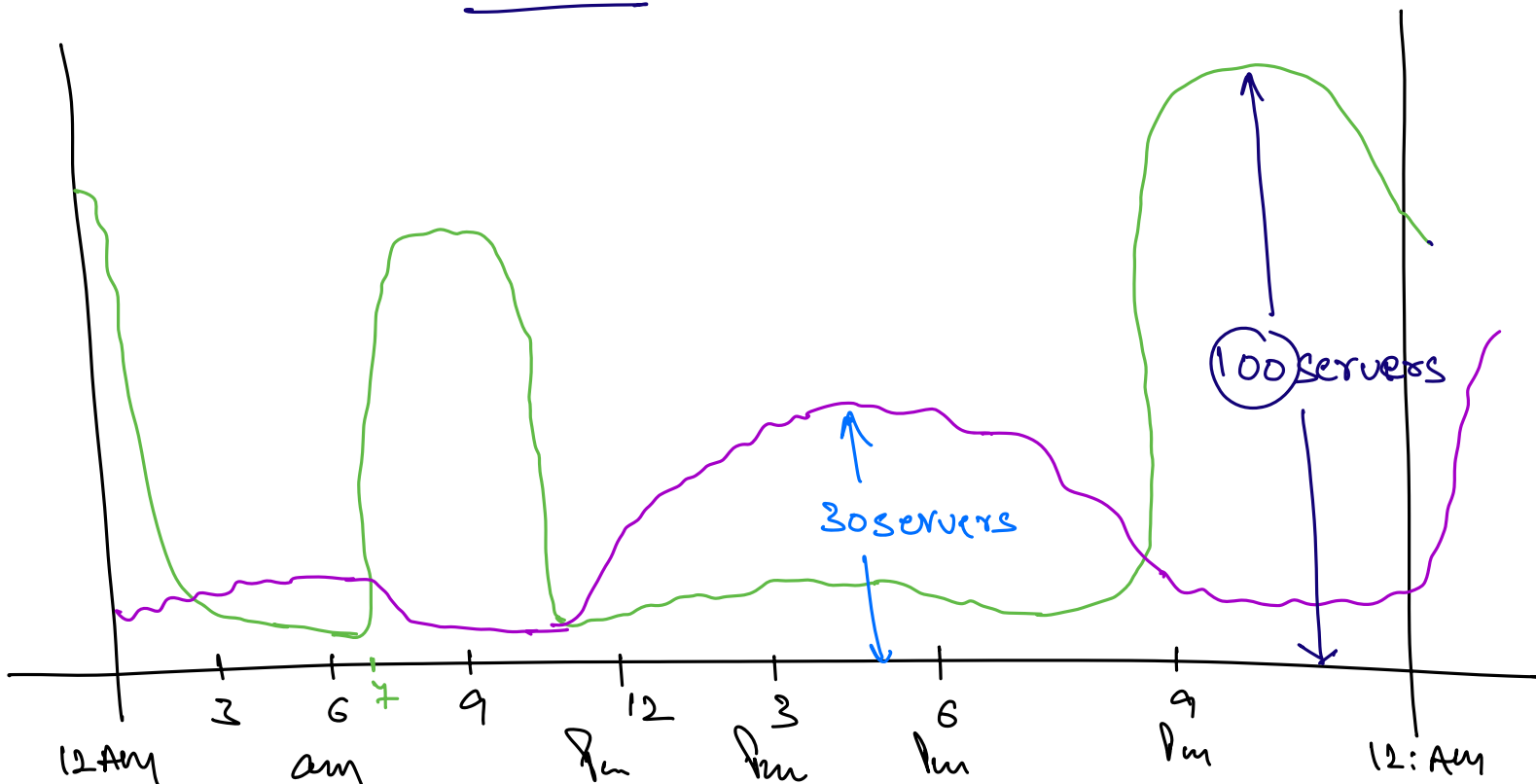2) E2E testing is easy.
3) Easy to debug
4) Single team.

DisAdvantages:

⇒ A small issue can make the entire system down.

⇒ Tight coupling

⇒ No tech stack feasibility

⇒ High Deployment time.

SCALER.

→ Video Streaming Service ≈   9PM
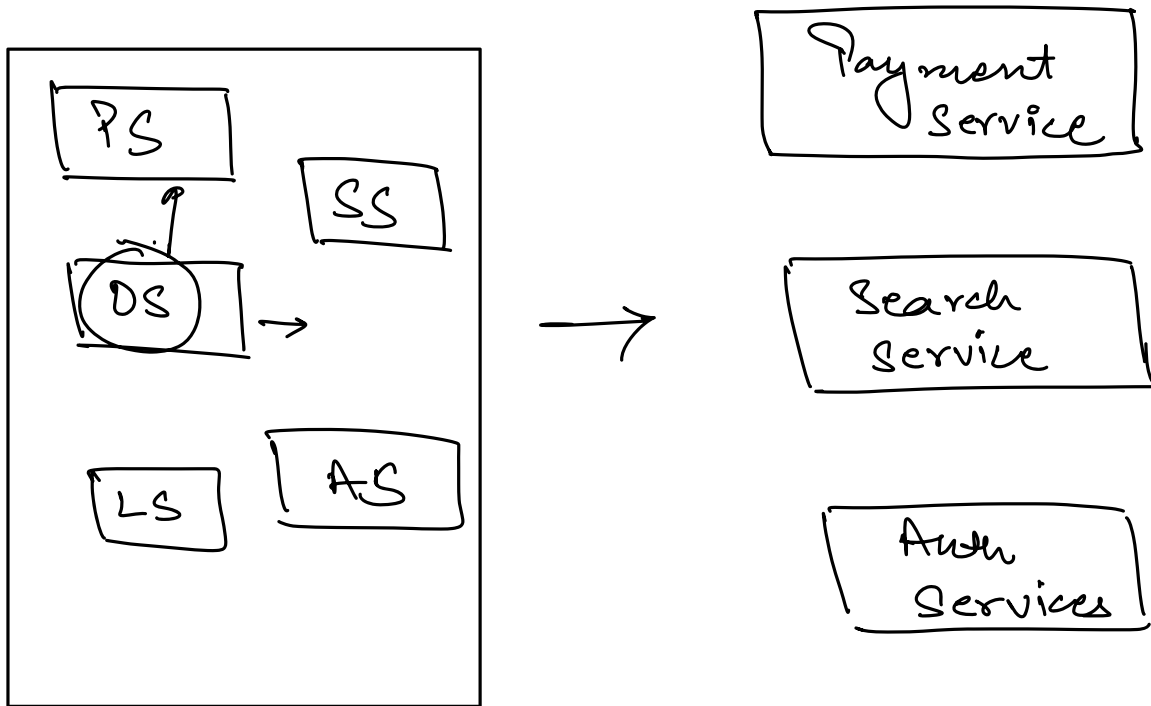
→ Assignment Service —   7AM.

→ Auth Service

MICROSERVICES.

⇒ Instead of having only one app", split our application into individual components.

Each component ⇒ Microservice

⇒ One service might need to communicate with other services in order to get the work. done.

Order Service {

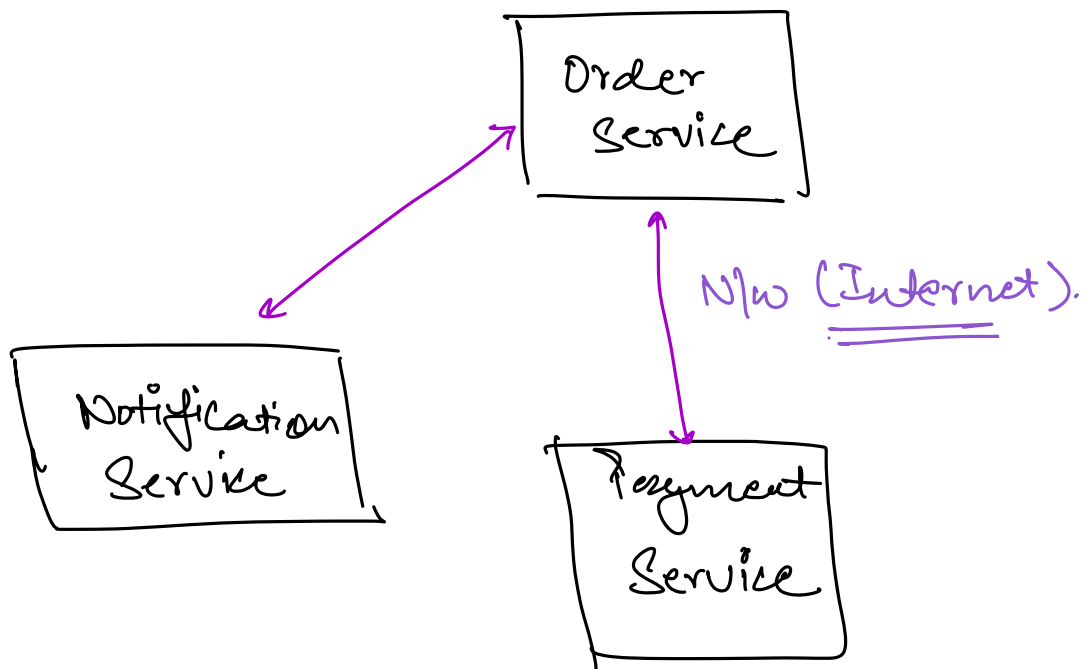 onOrderPlaced() {

   Notif^n Service
   Payment Service   } PS.doPayment(—)
   Logistic =
            =   =

  3   3
  =   =

=> Microservices.
         =

OrderService {

    onOrderPlaced() {

        ||HTTP.
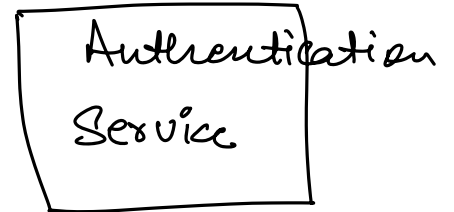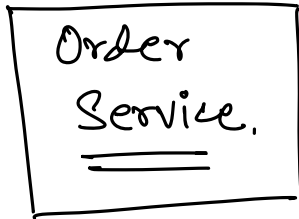
3

3

⇒

token

T/F.

Order Service.

Authentication Service

Advantages & DisAdvantages of Microservices.

Advantages

1) Tech Stack flexibility.

2) faster Deployments.
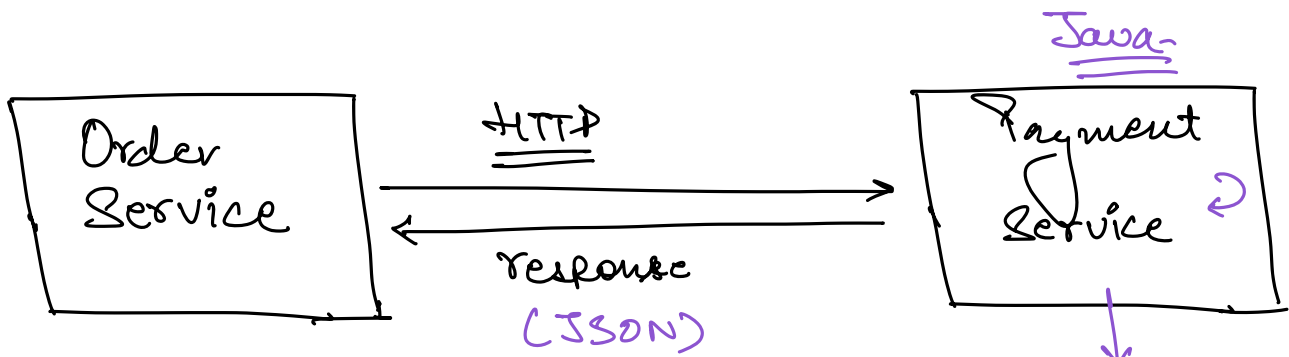
3) Selective Scaling

4) Easy onboarding of new Engineers.

DisAdvantages.

1) lot of teams.

2) High latency because of n/w calls.

3) E2E testing.

⇒ Communication b/w Microservices.

1) HTTP

2) gRPC + Protobufs
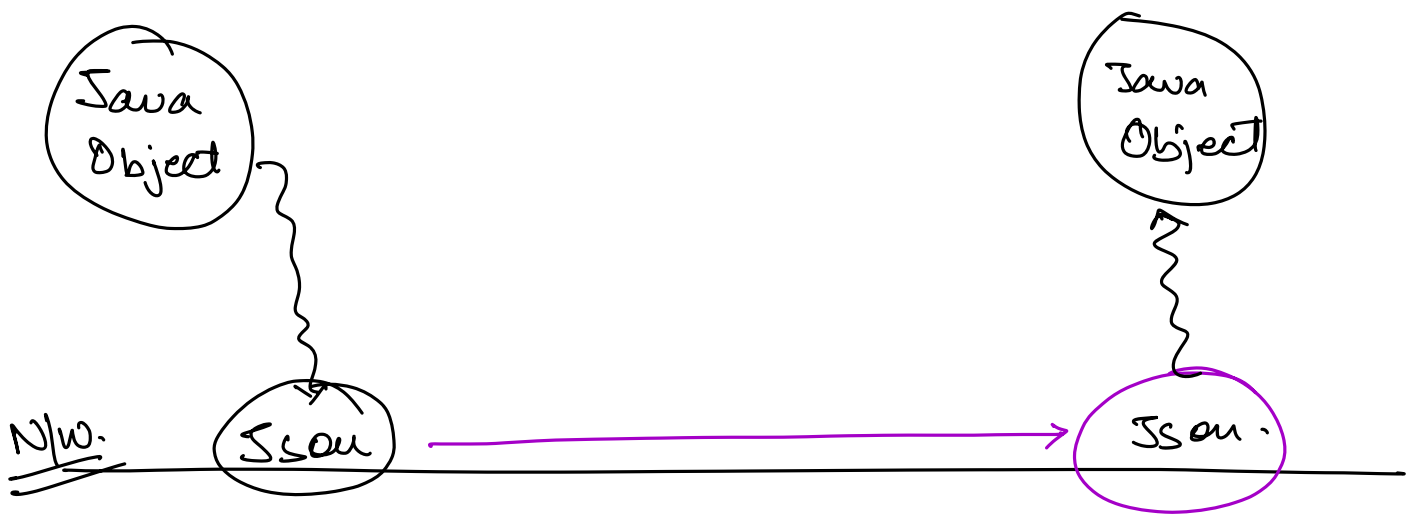
3) Event Driven Systems.

⇒ (HTTP.)
      ↓
   N/W

HTTP →

Order Service ⇄ Payment Service (Java)

response (JSON)

→ Java Object.

JSON. ⇒ key value.
{
   "id" : ——
   "name" : ——
   "desc" : ——
}

⇒ Easily readable

Java Object ———— Java Object

N/W. Json ⟶ Json.

Serialization   DeSerialization
Marshalling   UnMarshalling

⟹ Time consuming

2) gRPC + Protobufs.

⟶ Data format developed to make N/w Calls faster.

⟹ language agnostic.

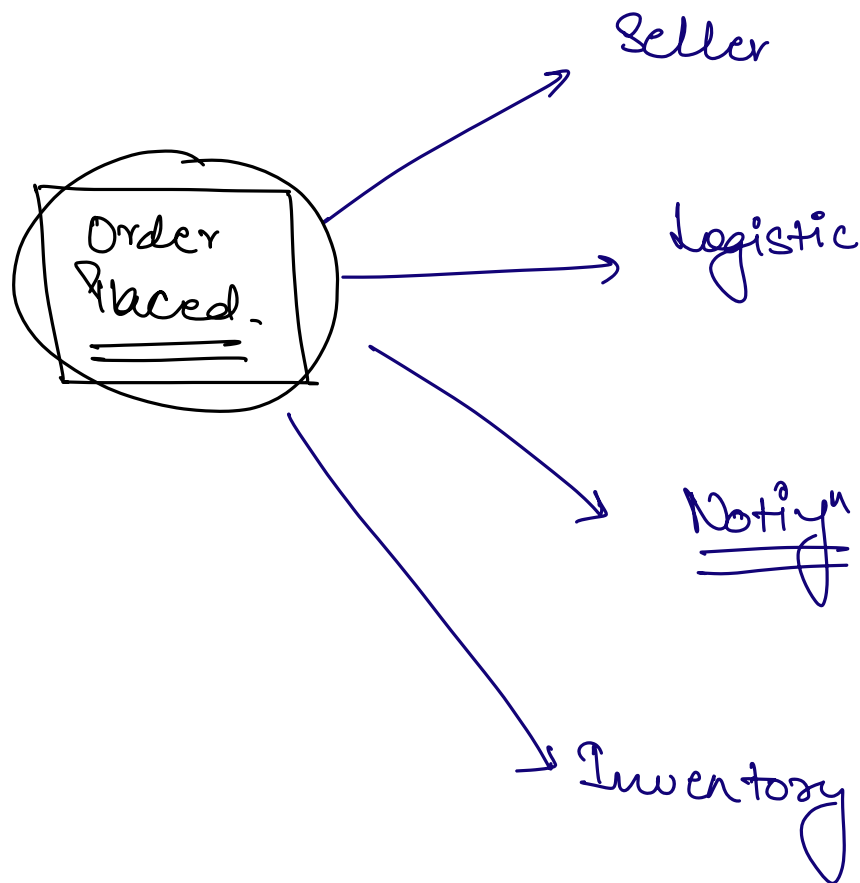⟹ Binary.

⇒ Communication via Protobufs is faster as No marshalling & unmarshalling is required.
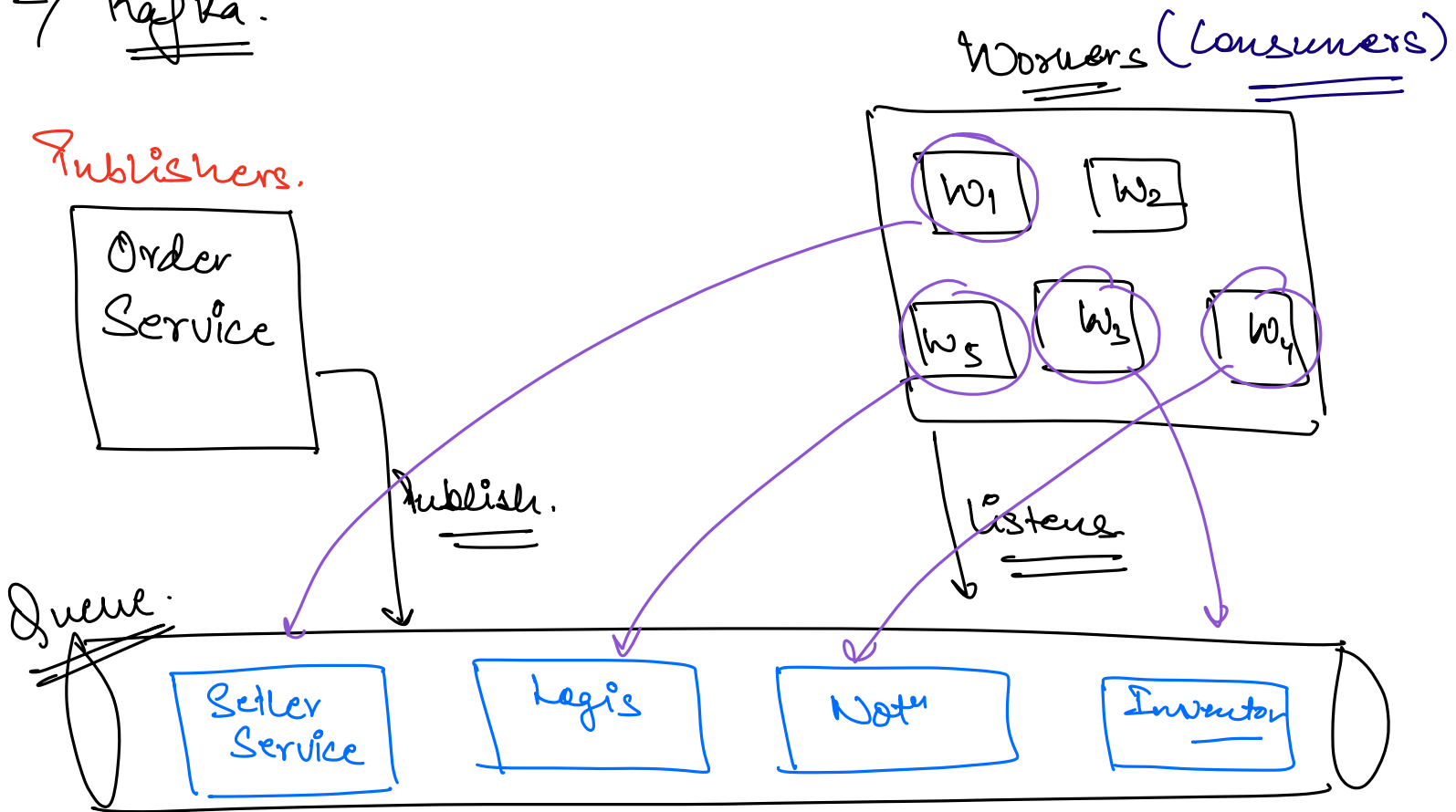
③ Event Driven Systems.

⇒ Asynchronous. (Parallel).

Seller

Order Placed. → Logistic

→ Notifⁿ

→ Inventory

⇒ All these services can be called asynchronsly

⇒ Message Queues.
Kafka
Rabit MQ

⇒ Kafka.

Workers (Consumers)

Publishers.

Order Service

W₁ W₂

W₅ W₃ W₄

Publish.

Listens

Queue.

| Seller Service | Logis | Not⁴ | Inventor |

⇒ PubSub Model.