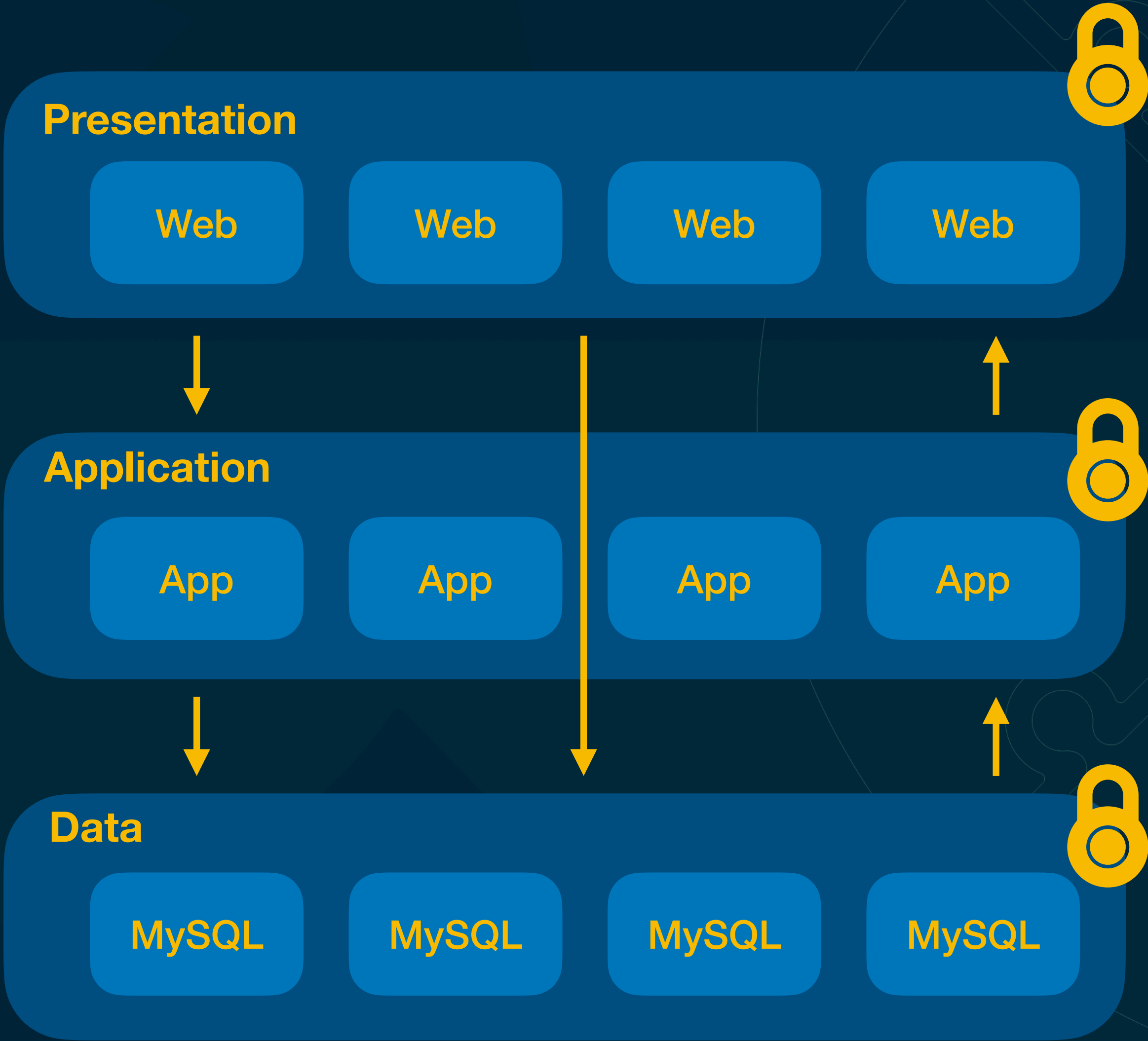


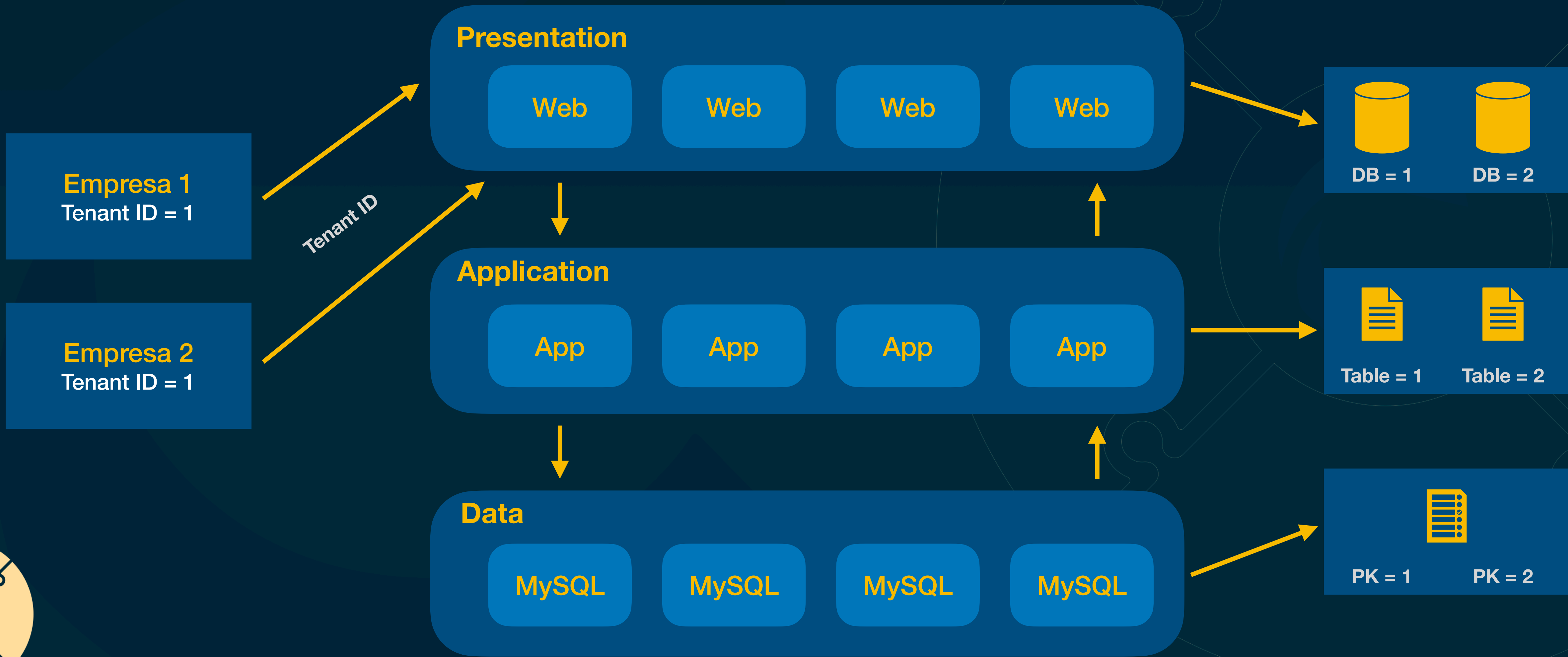
Design patterns

Soluções comuns para diversos tipos de problemas

N-tier / N-layer



Multi-tenant



Stateless vs Stateful

Application

Session

Assets

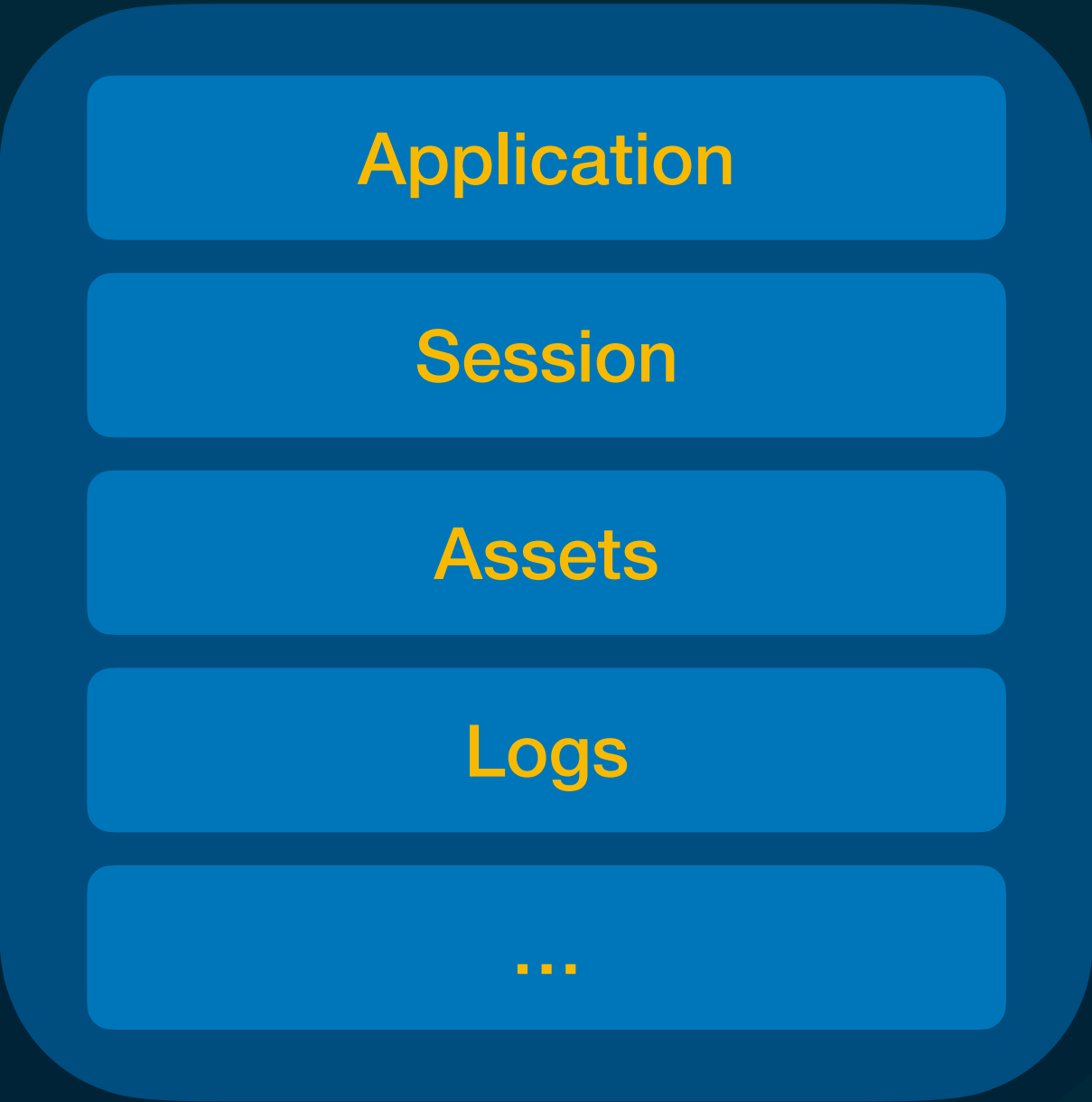
Logs

...

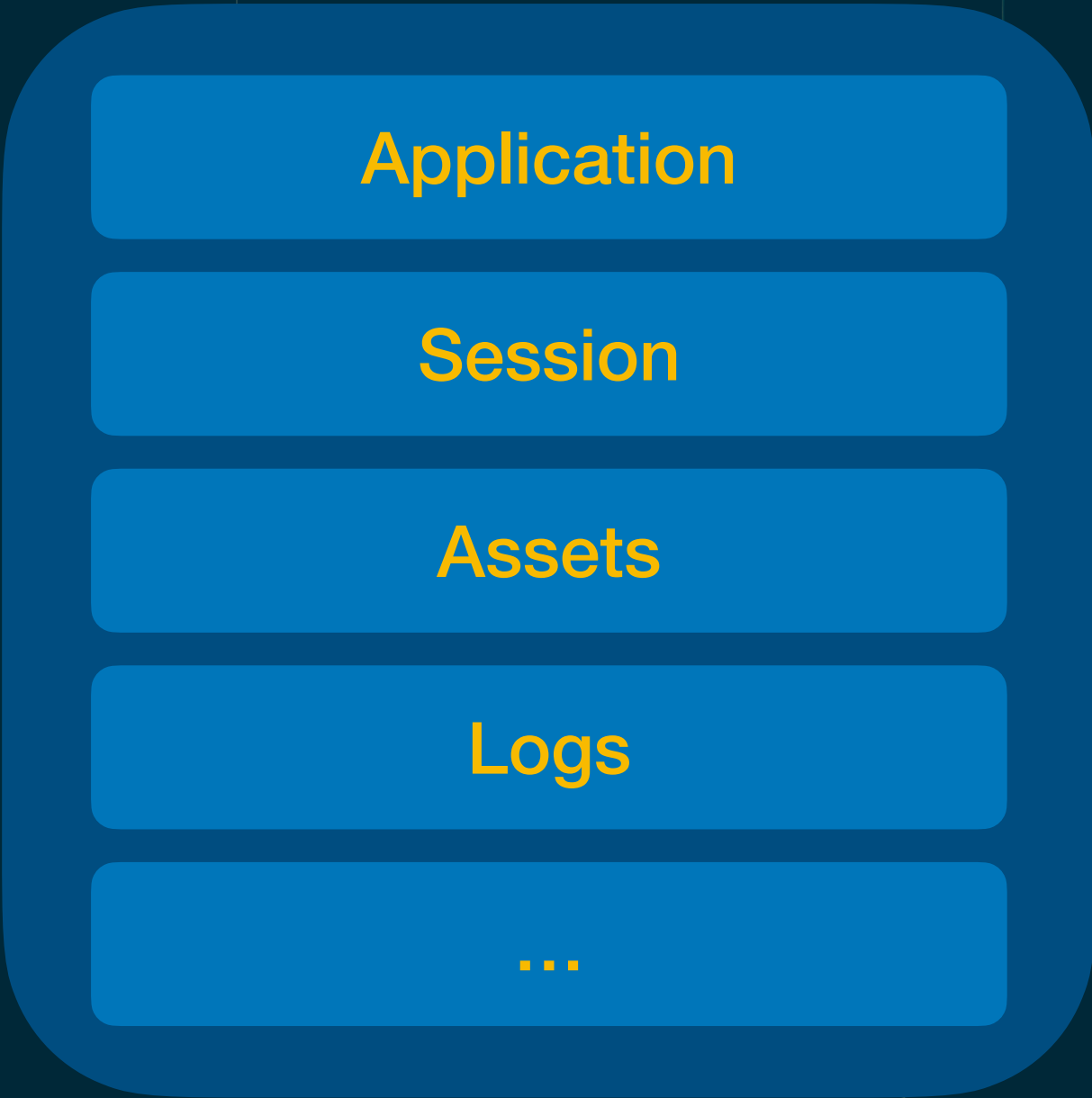
Stateless vs Stateful

ASG = Auto scaling group

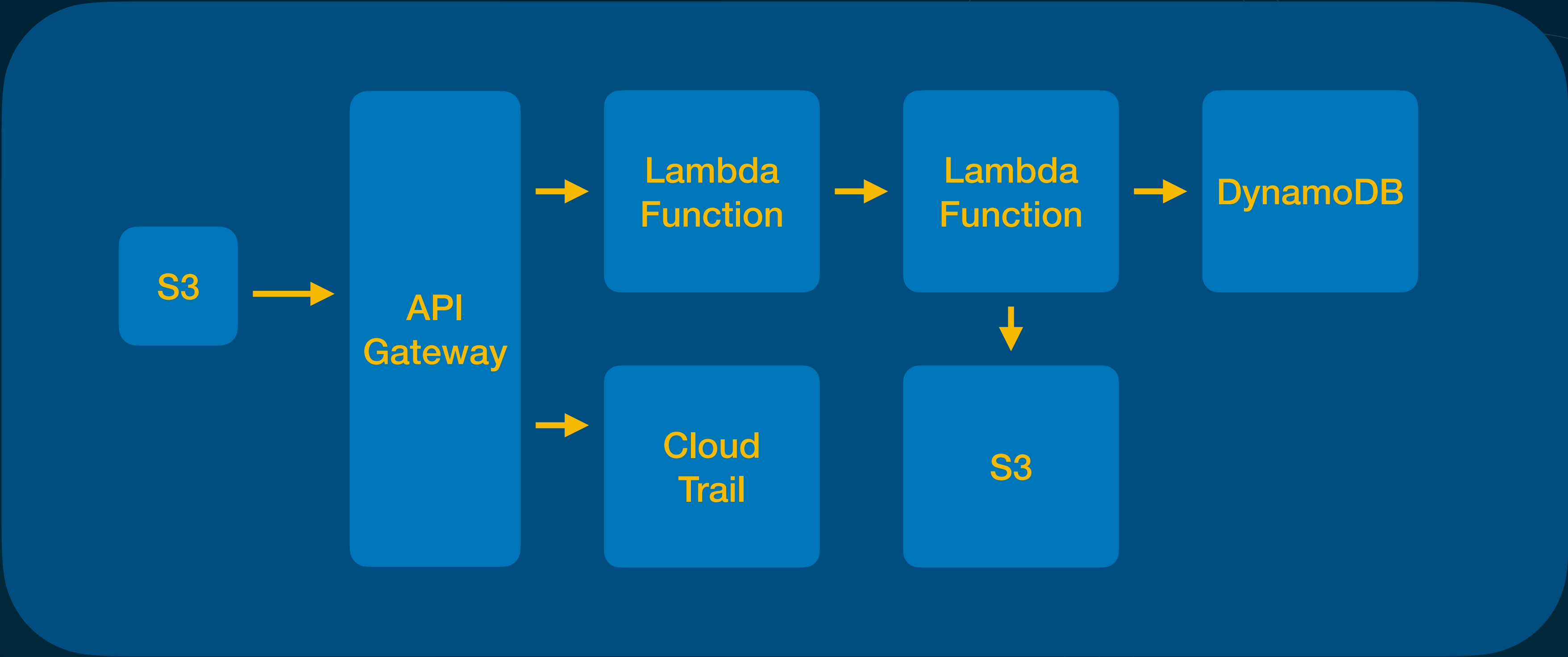
Instance 1



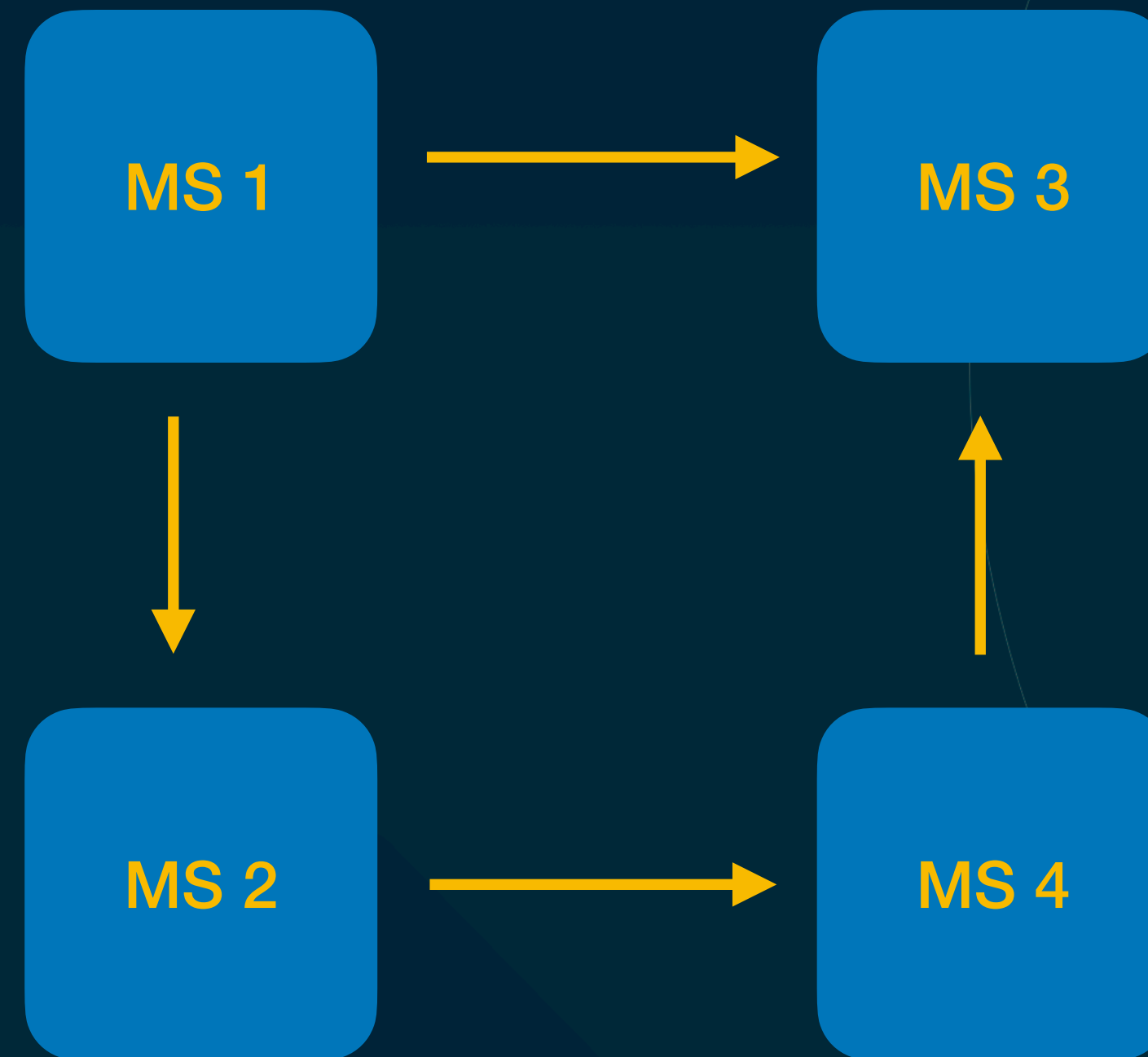
Instance 2



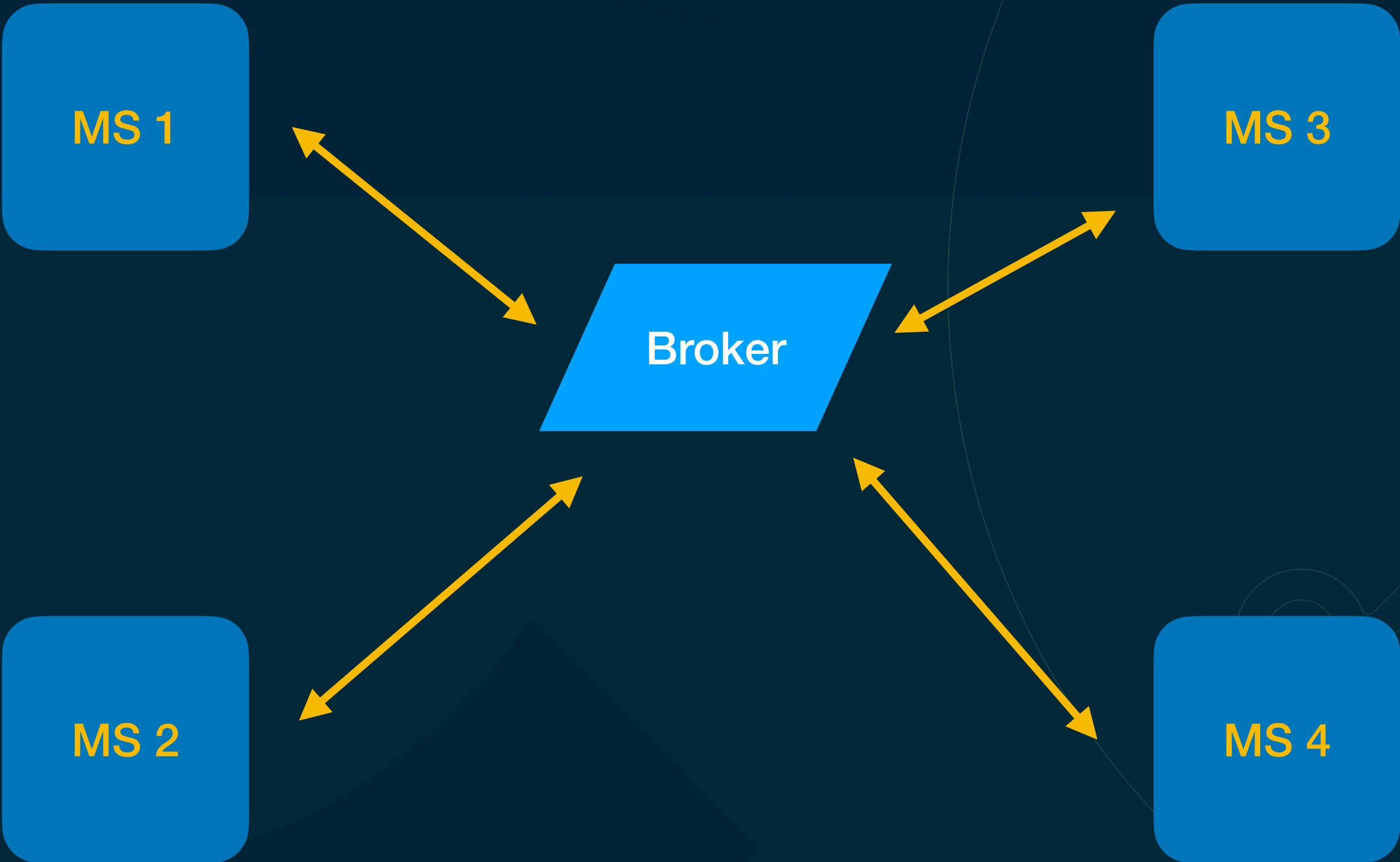
Serverless



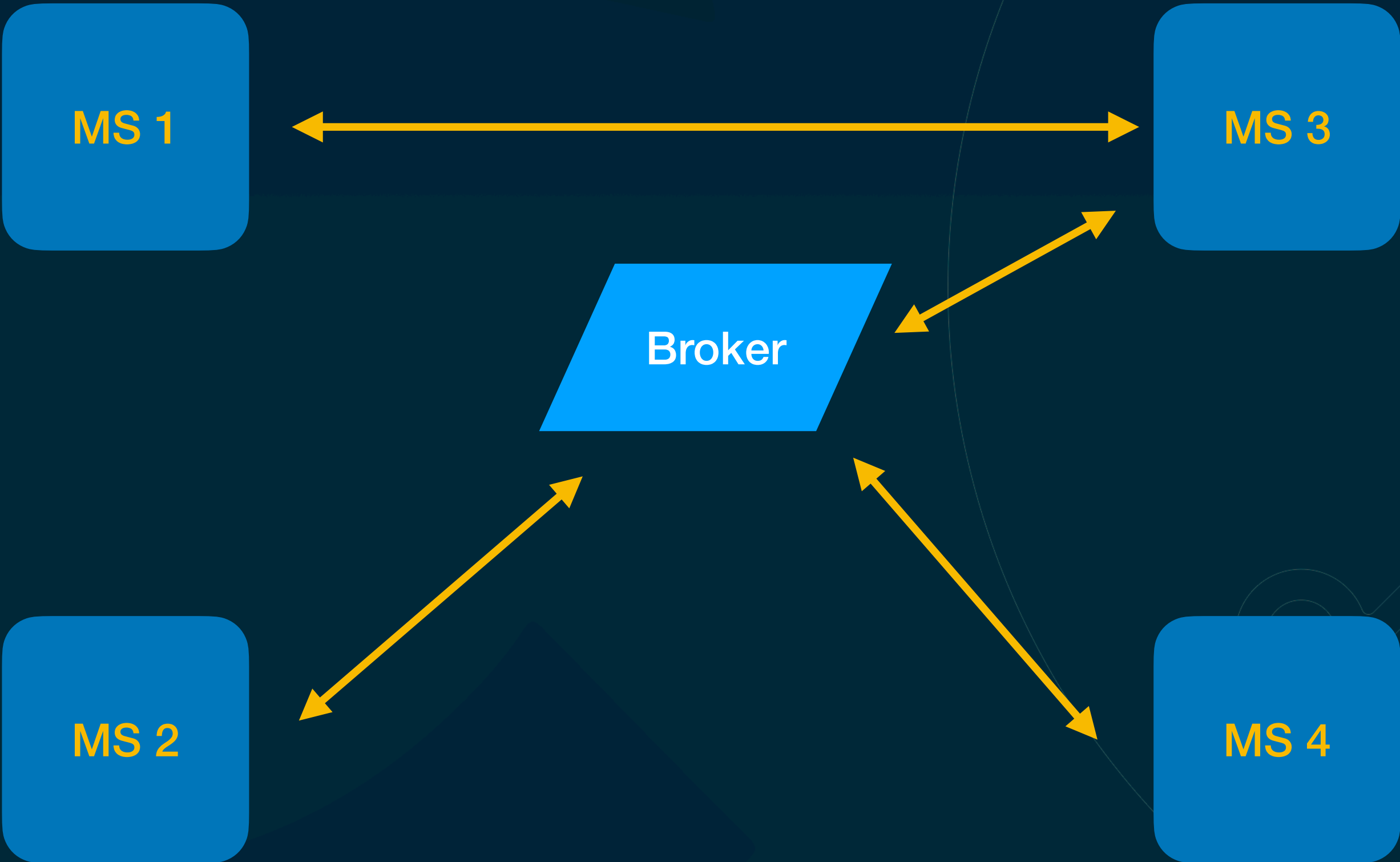
Microserviços



Microserviços



Microserviços



Microserviços

- Principal motivação: **Organizacional. Equipes.**
- Escalabilidade
- Separação de responsabilidades
- Diferentes tecnologias
- Baixo acoplamento

Microserviços vs Complexidades

- Maturidade da organização
- Maturidade dos times
- Deployment
- Observabilidade
- Troubleshooting

CQRS (Command Query Responsibility Segregation)

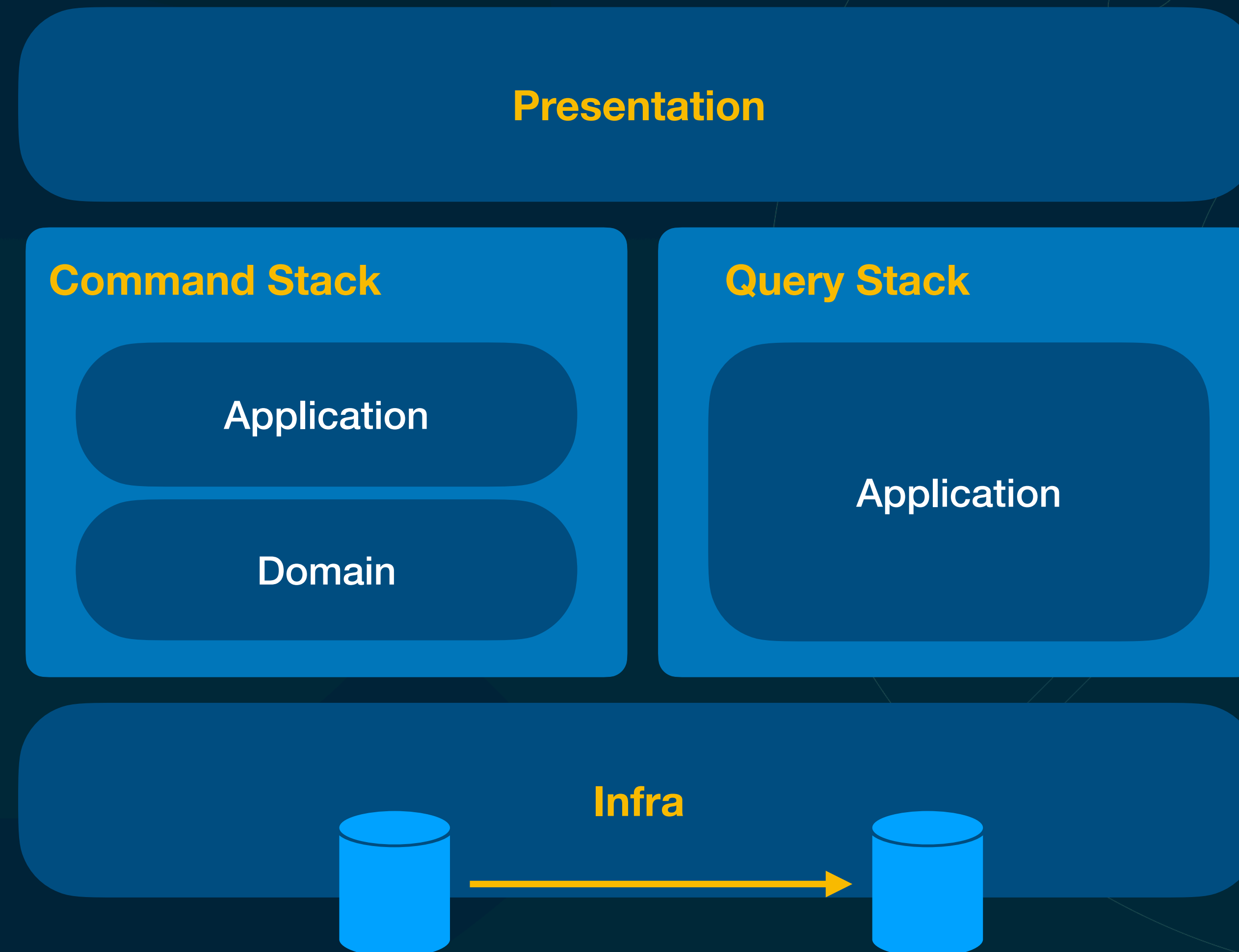
Presentation

Application

Domain

Infra

CQRS (Command Query Responsibility Segregation)



Caching



900ms

Caching



Caching

- Time-based invalidation
- Least Recently Used (LRU)
- Most Recently Used (MRU)
- Least Frequently Used (LFU)
- TTL-based invalidation
- Write-through invalidation
- Write-back invalidation

Caching - LRU

D*
C
B
A

Caching - MRU

C*

C

Caching - LFU

D - 10

C - 5

B - 6

A - 7

Caching - Write-through invalidation

Sempre quando há alteração no disco, o cache é atualizado em conjunto.

Essa estratégia funciona bem com sistema que não possuem um alto índice de escrita.

Caching - Write-back invalidation

Quando há alteração, o cache primeiramente é atualizado e depois o dado em disco é atualizado. Muitas vezes o dado em disco é atualizado quando o cache já está para expirar de alguma forma.

Distributed Locking

- Consistência de dados
- Contenção de recursos
- Evita dead locks
- Garante mais eficiência dos recursos
- Ferramentas: Apache Zookeeper, ETCD, Redis, Consul

Distributed Locking

```
1  import * as zookeeper from 'node-zookeeper-client';
2
3  const zkClient = zookeeper.createClient('localhost:2181');
4  const lockPath = '/product-id';
5
6  // faz lock em um registro
7  function lock() {
8    return new Promise<void>((resolve, reject) => {
9      zkClient.create(lockPath, zookeeper.CreateMode.EPHEMERAL, (error, path) => {
10        if (error) {
11          reject(error);
12        } else {
13          resolve();
14        }
15      });
16    });
17  }
18
19  // faz unlock em um registro
20  function unlock() {
21    return new Promise<void>((resolve, reject) => {
22      zkClient.remove(lockPath, -1, (error) => {
23        if (error) {
24          reject(error);
25        } else {
26          resolve();
27        }
28      });
29    });
30  }
```


Configuration

- Configurações de uma aplicação mudam a qualquer momento
- Como mudar uma senha de banco de dados, credenciais de email, etc.. Sem ter que reiniciar totalmente a aplicação ou refazer o deploy?

Configuration



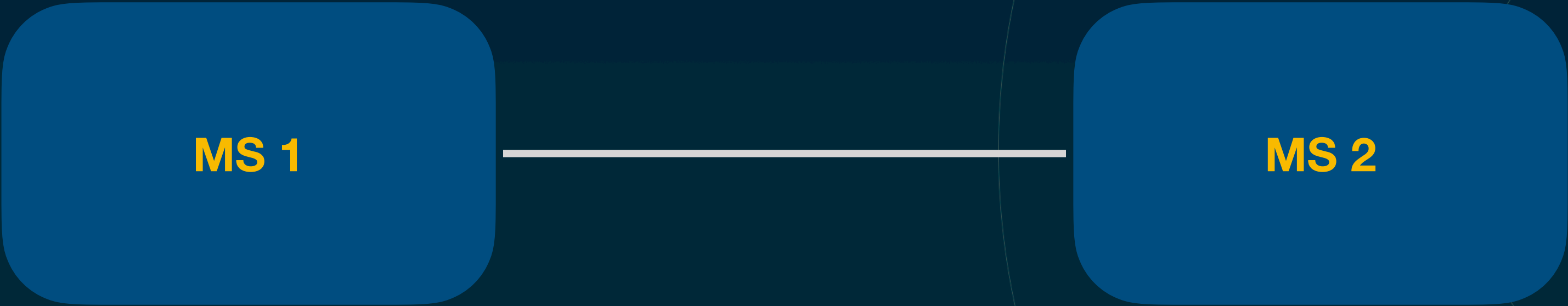
Secret Management

- Credenciais não podem ficar “voando” na empresa
- Processos para rotacional credenciais são importantes
- Serviços gerenciados e soluções ajudam nessa tarefa
 - Hashicorp Vault
 - AWS Secret Manager
 - Armazenamento de secrets
 - Rotacionamento automáticos de secrets nos serviços como RDS.
 - SDK para recuperação dos secrets em tempo de execução

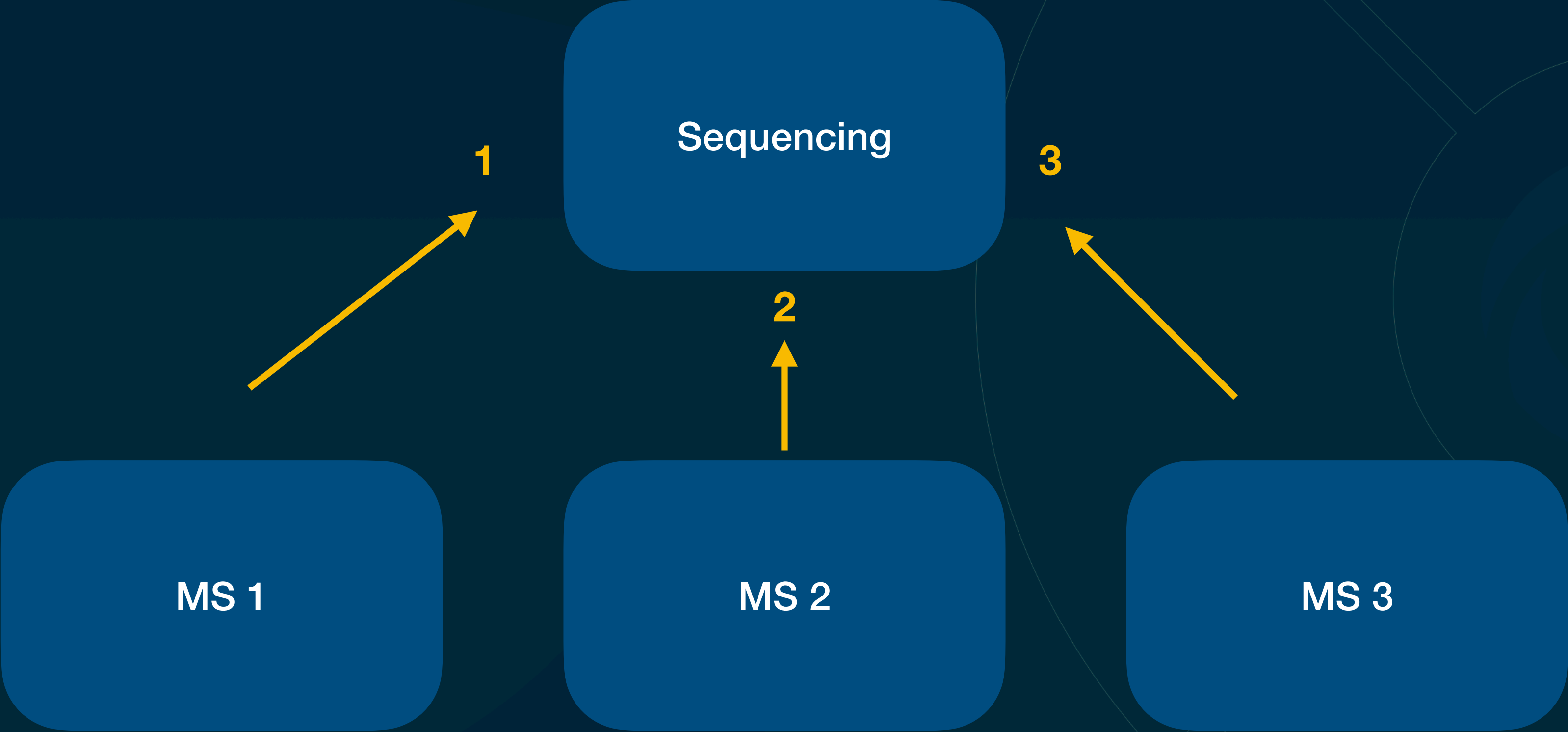
Circuit breaker



Circuit breaker



Sequencing



API Gateway

- Centralizador de requisições
- Roteamento
- Autenticação
- Conversão de dados
- Cabeçalhos
- Throttling
- Rate Limit

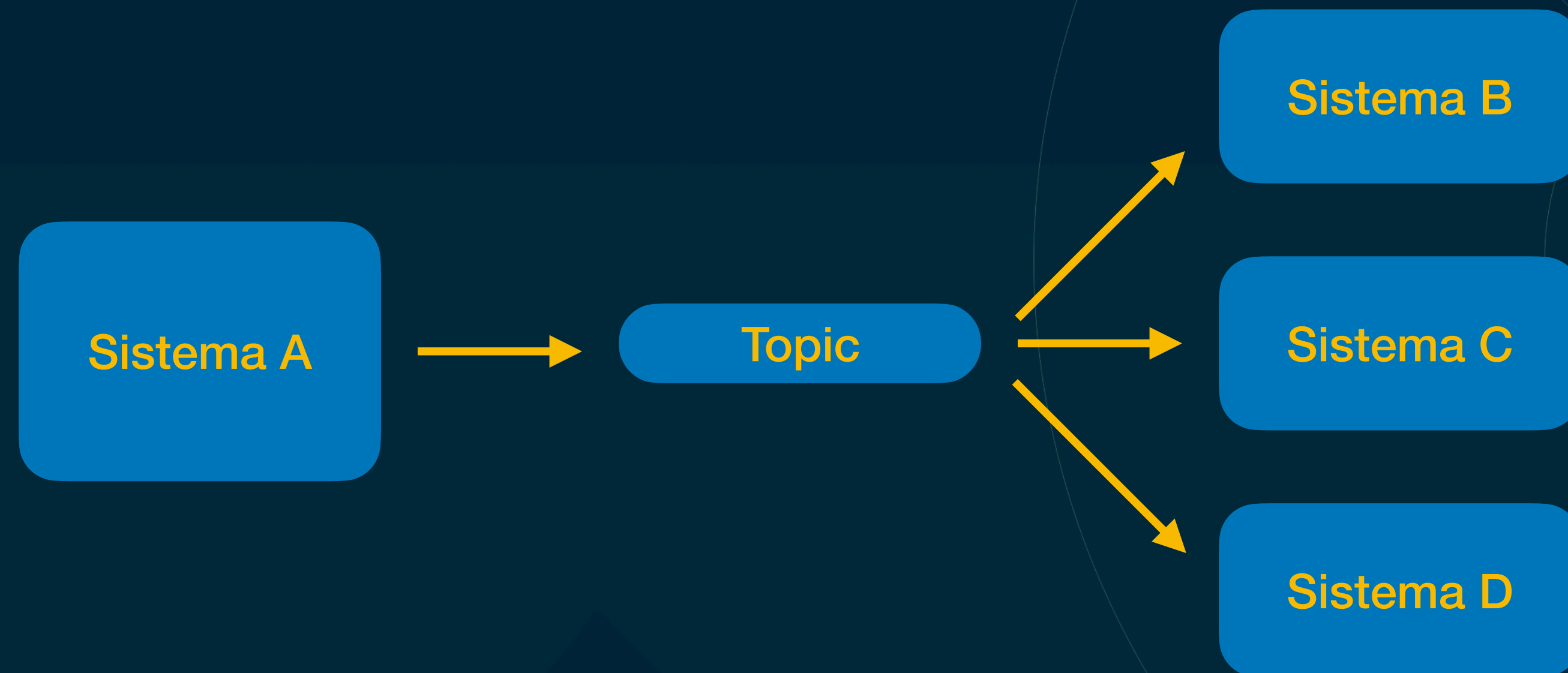
API Gateway



Event Driven Architecture

- Evento acontecem no passado
 - Event Notification
 - Event Carried State Transfer
 - Event Sourcing
- Um evento emitido pode ser o gatilho de entrada para um outro sistema
- Coreografia vs Orquestração

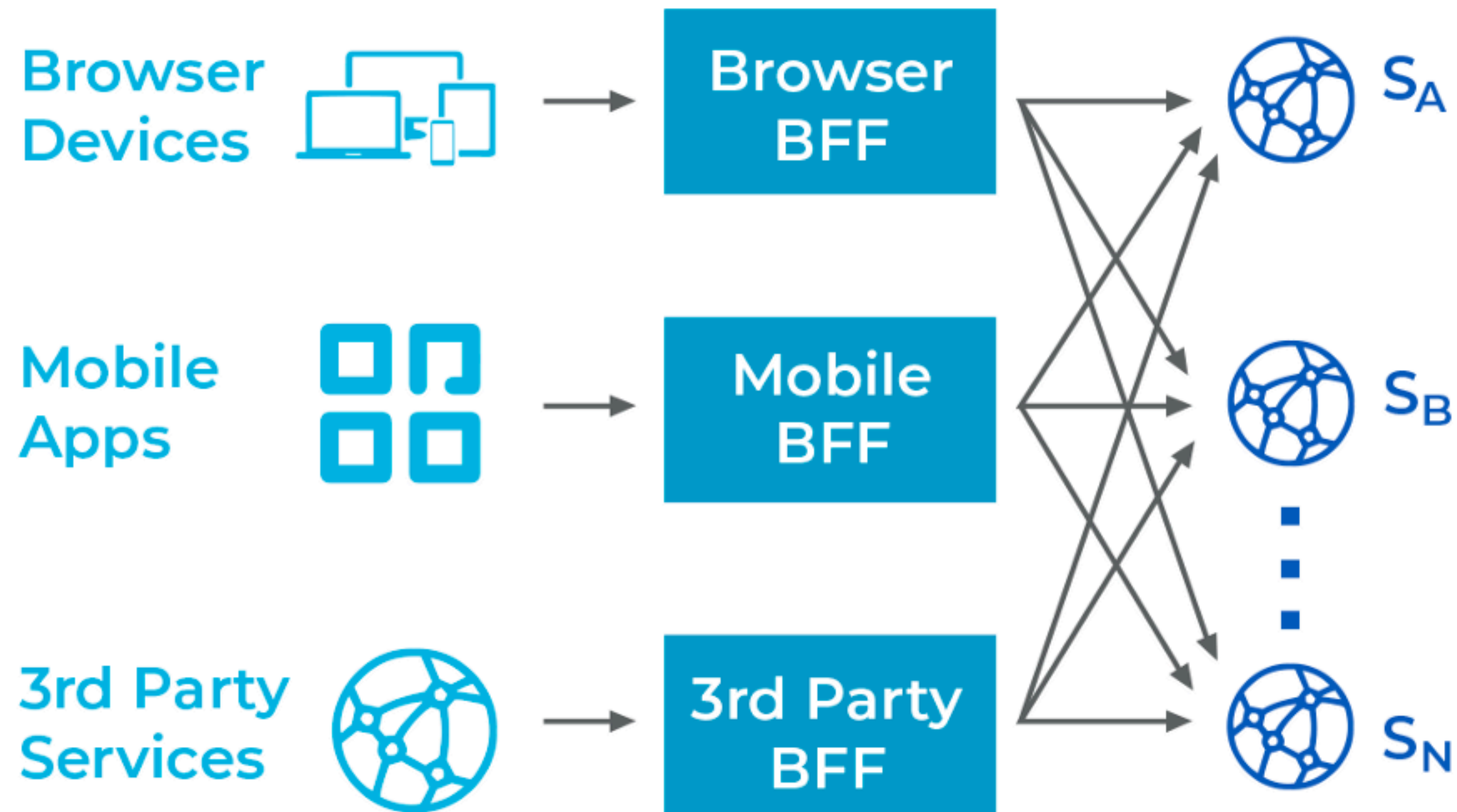
Publish-Subscribe



BFF - Backend for frontend

BFF Implementation

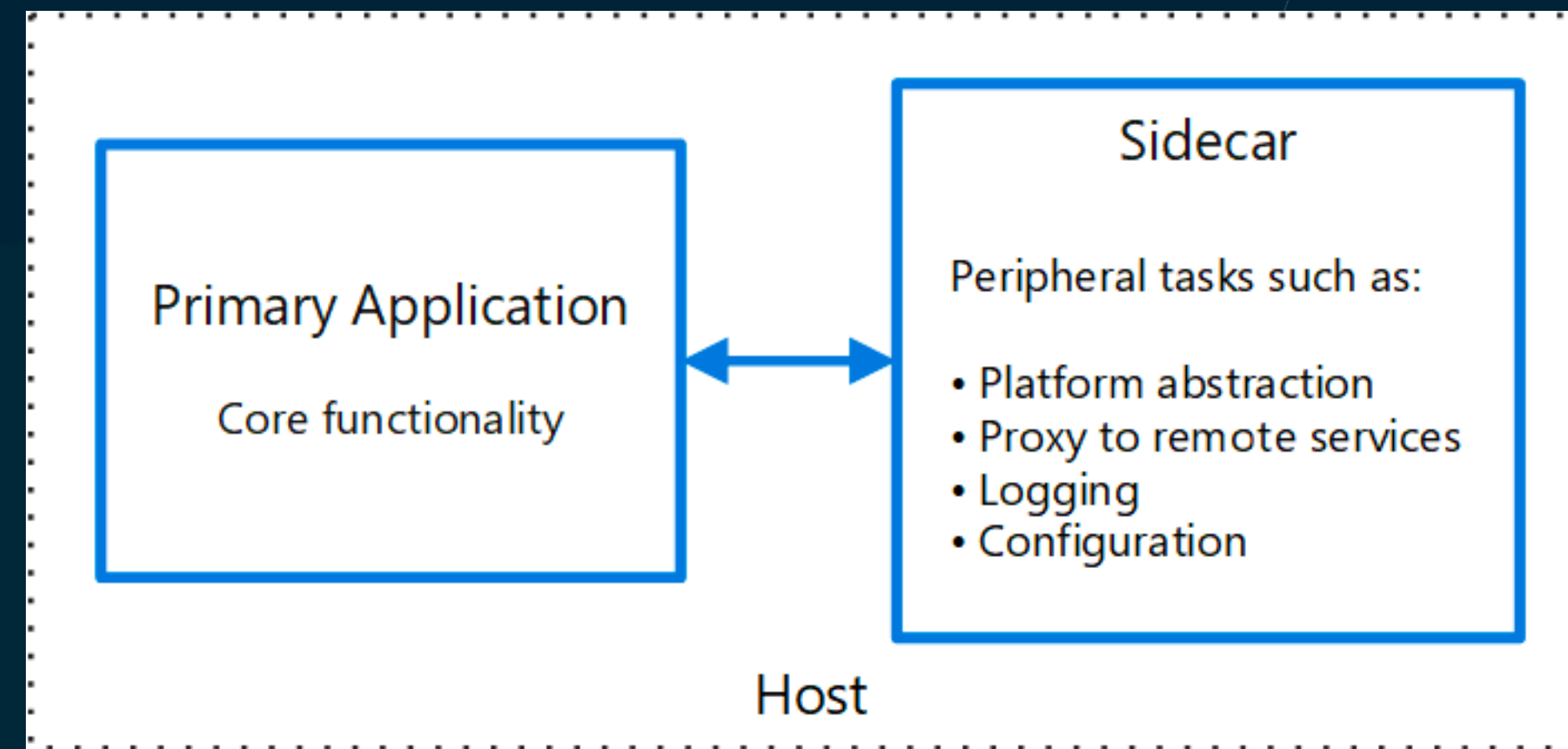
One BFF Backend -
Single Application
Interface



Sidecars applications

- Aplicações auxiliares na aplicação principal
- Coleta de logs
- mTLS
- Controle de tráfego

Sidecars applications



<https://learn.microsoft.com/en-us/azure/architecture/patterns/sidecar>

Service Mesh

"Uma malha de serviço é uma camada de infraestrutura dedicada que você pode adicionar às suas aplicações. Ele permite adicionar recursos de forma transparente, como observabilidade, gerenciamento de tráfego e segurança, sem adicioná-los ao seu próprio código"

<https://istio.io/latest/about/service-mesh/>

Service Mesh - Istio

- Gerenciamento de tráfego
- Segurança
- Policy enforcement
- Observabilidade
- Extensibilidade

Service Mesh - Istio

