

20/10

PROJECT
Decoder

Microservices Patterns e Spring Projects

Overview Completo

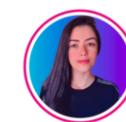
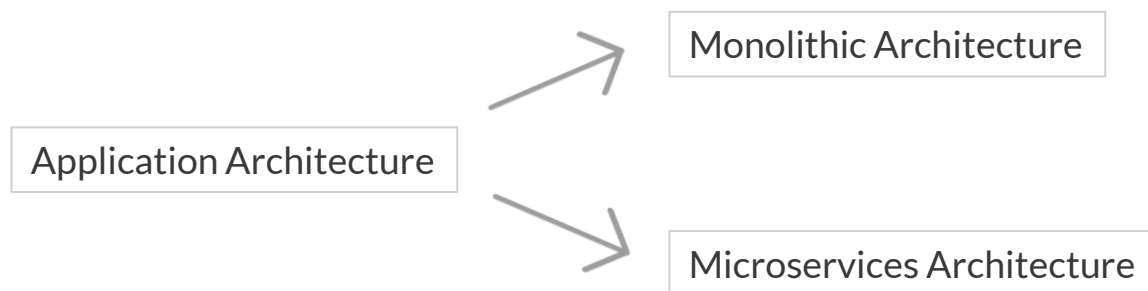
O que você verá:

- Overview do ecossistema Spring para a implementação de Arquiteturas de Microservices;
- Comunicação entre Microservices;
- Comunicação Síncrona One-To-One Request/Response;
- API Gateway Pattern;
- Service Registry Discovery Pattern;
- Circuit Breaker Pattern;
- API Composition Pattern;
- Comunicação Assíncrona One-To-One via Mensageria;
- Comunicação Assíncrona One-To-Many via Mensageria;

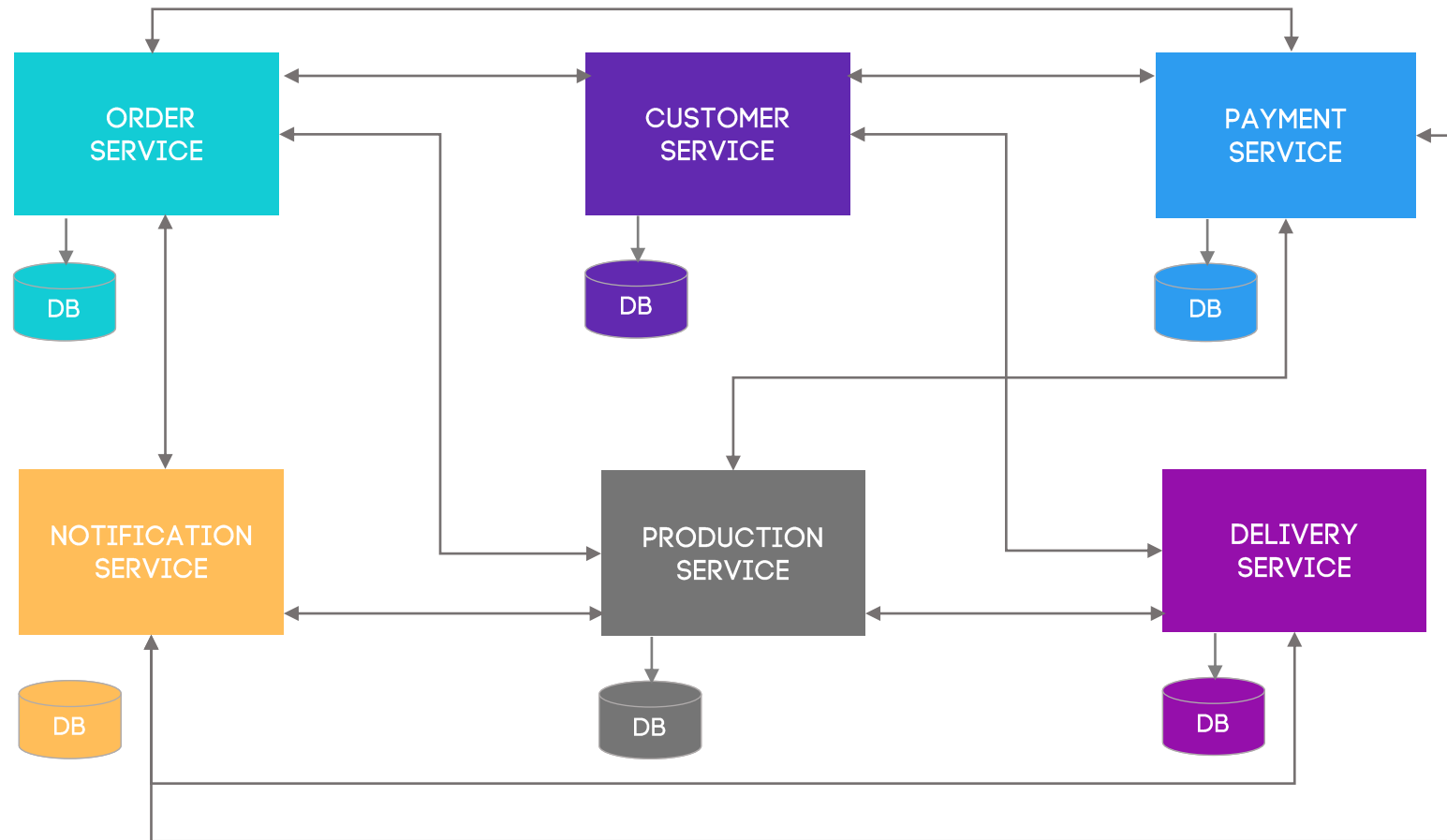
- Broker Pattern;
- Mediator Pattern;
- Event Driven com Event Notification Pattern;
- Event Carried State Transfer Pattern;
- Saga Pattern;
- Externalized Configuration Pattern;
- Observability: Distributed Tracing;
- Observability: Metrics;
- Log Aggregation Pattern;
- Etc.



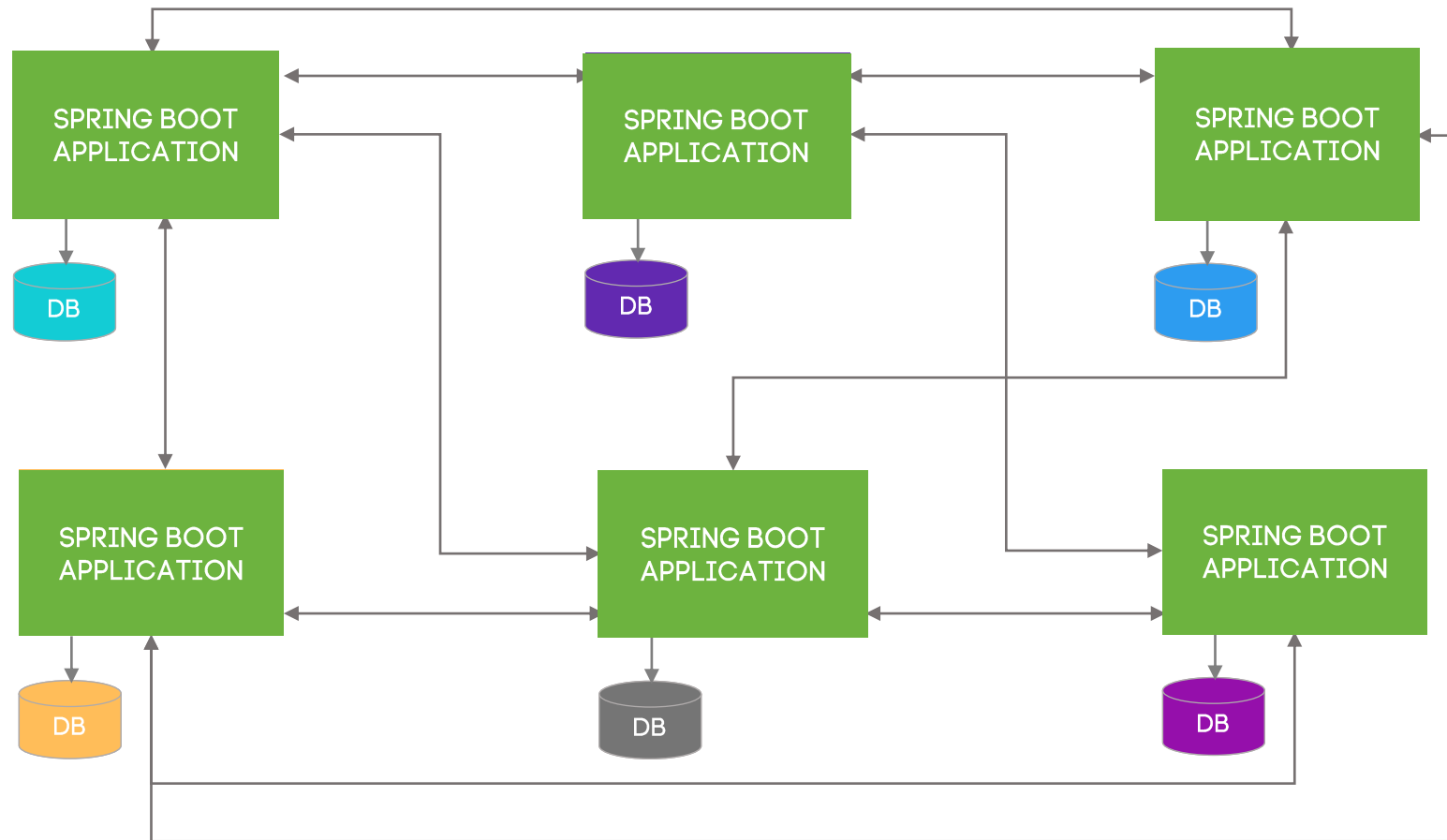
Arquitetura da Aplicação



Arquitetura de Microservices



Arquitetura de Microservices



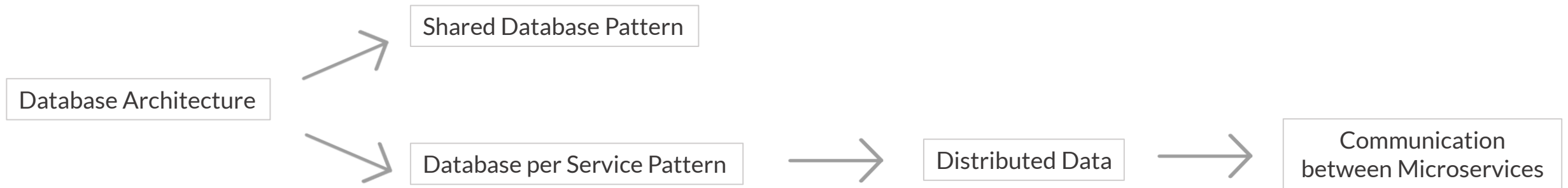
Spring Projects

Spring Boot
Spring Data

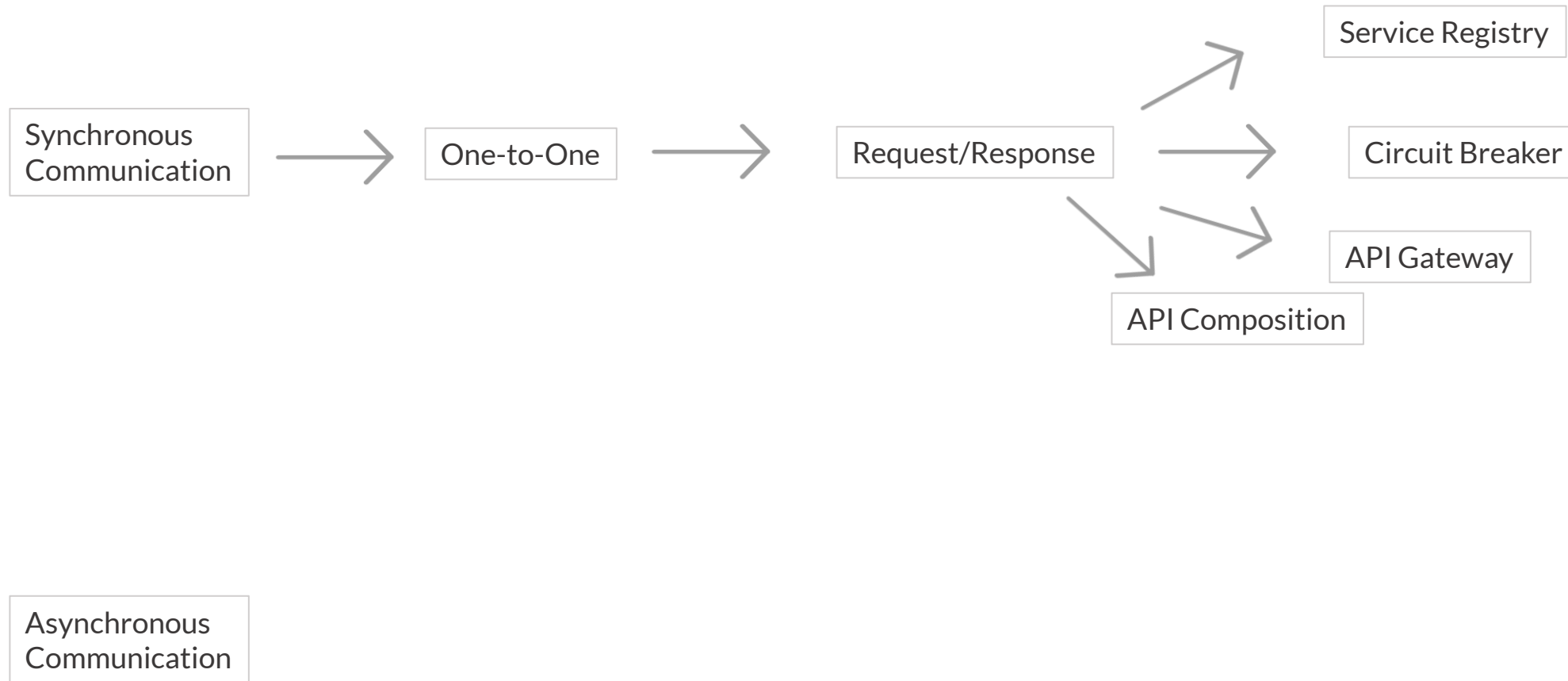


Michelli Brito
@brito_michelli

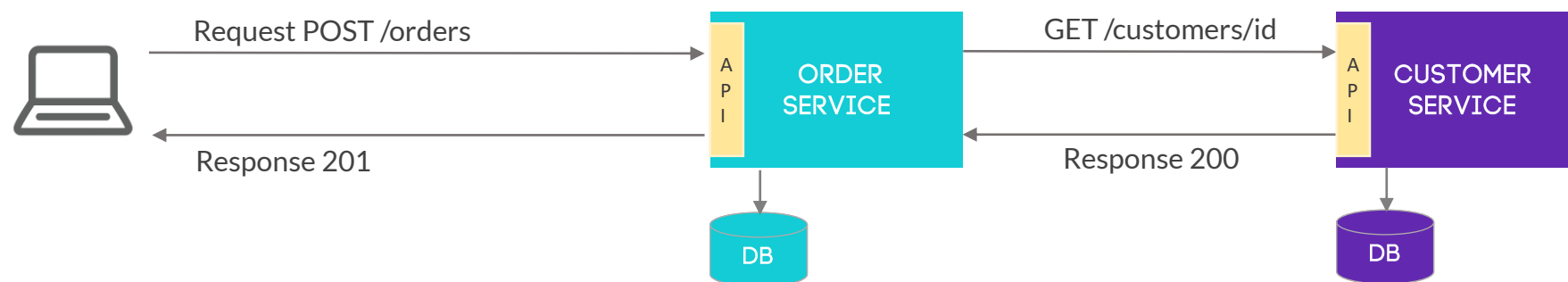
Database Pattern



Comunicação entre Microservices

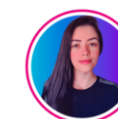


Comunicação Síncrona via REST API

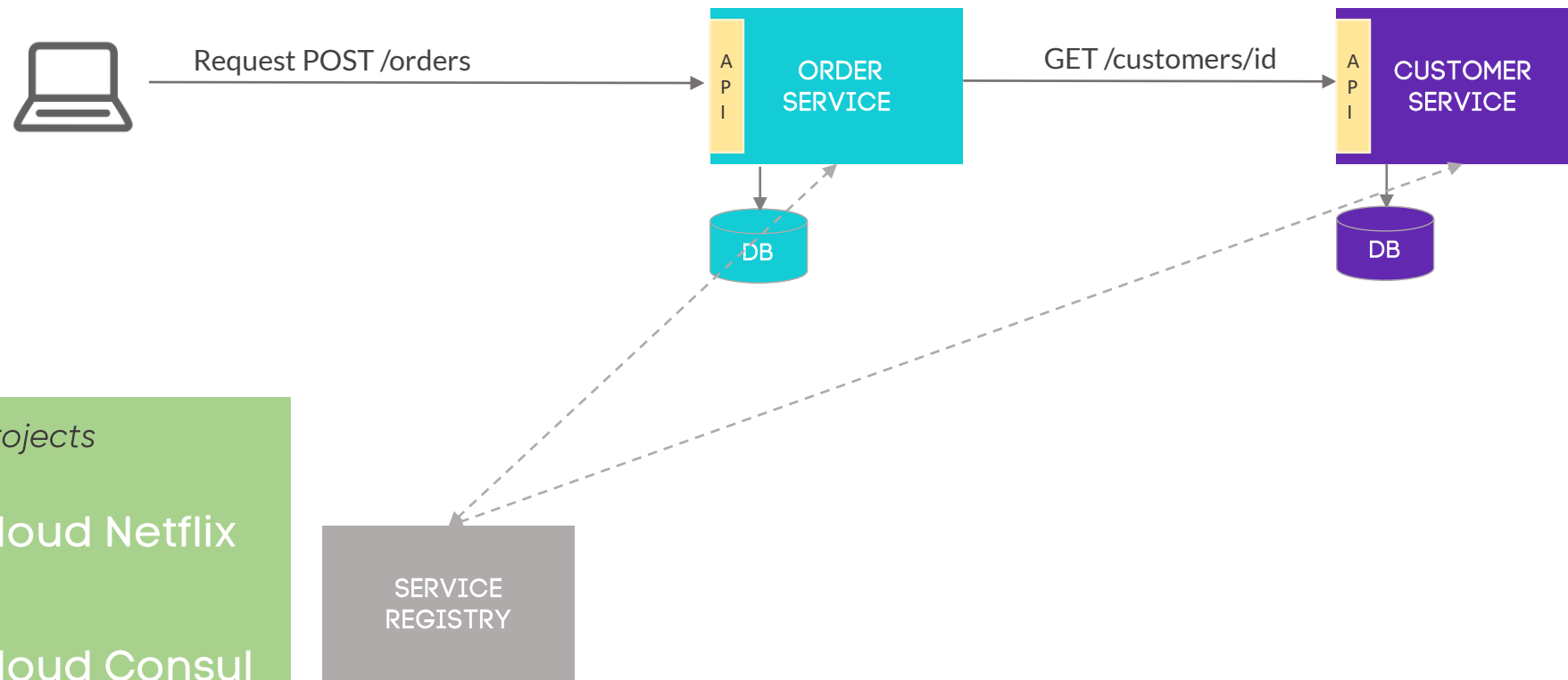


Spring Projects

Spring Web: RESTful and Spring MVC
Spring Reactive Web: Spring Webflux
Spring HATEOAS
Spring Validation



Service Registry / Discovery



Spring Projects

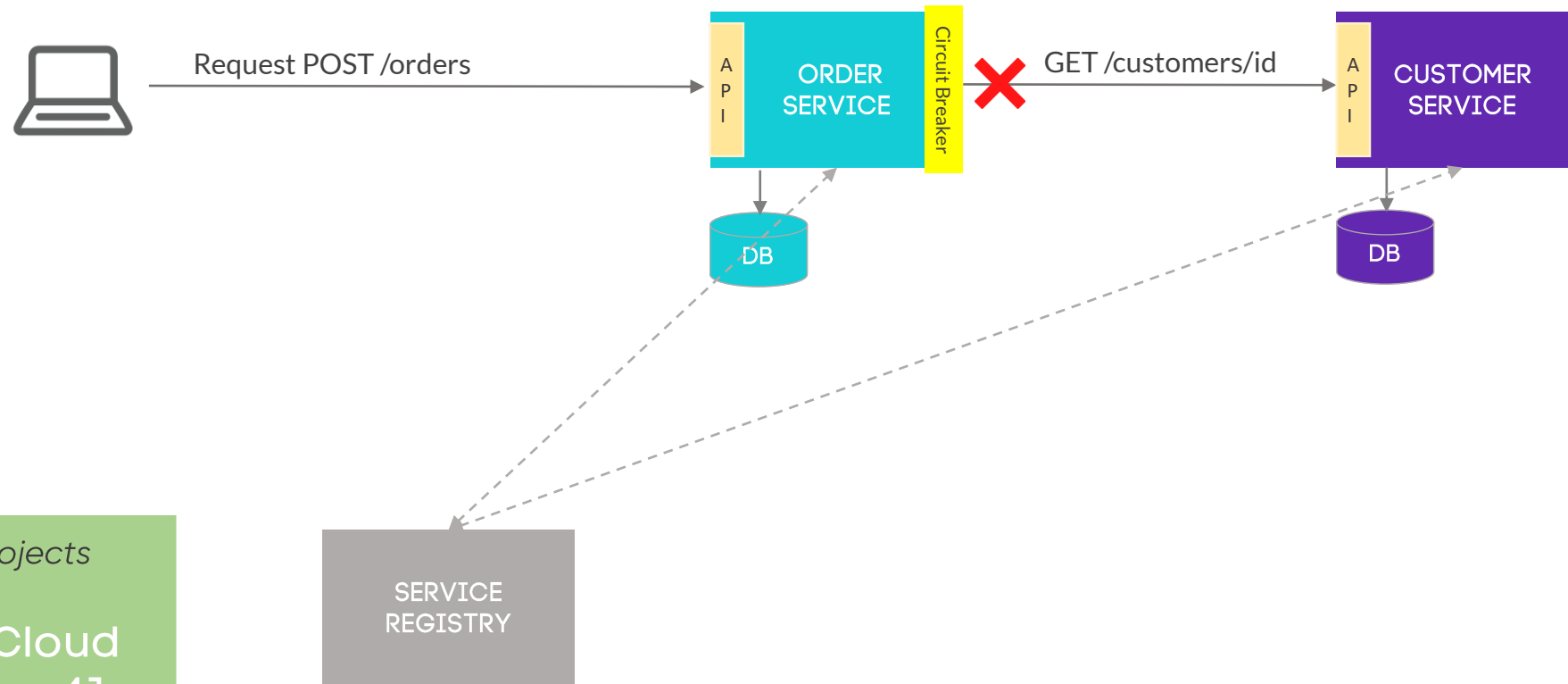
Spring Cloud Netflix
Eureka

Spring Cloud Consul



Michelli Brito
@brito_michelli

Circuit Breaker



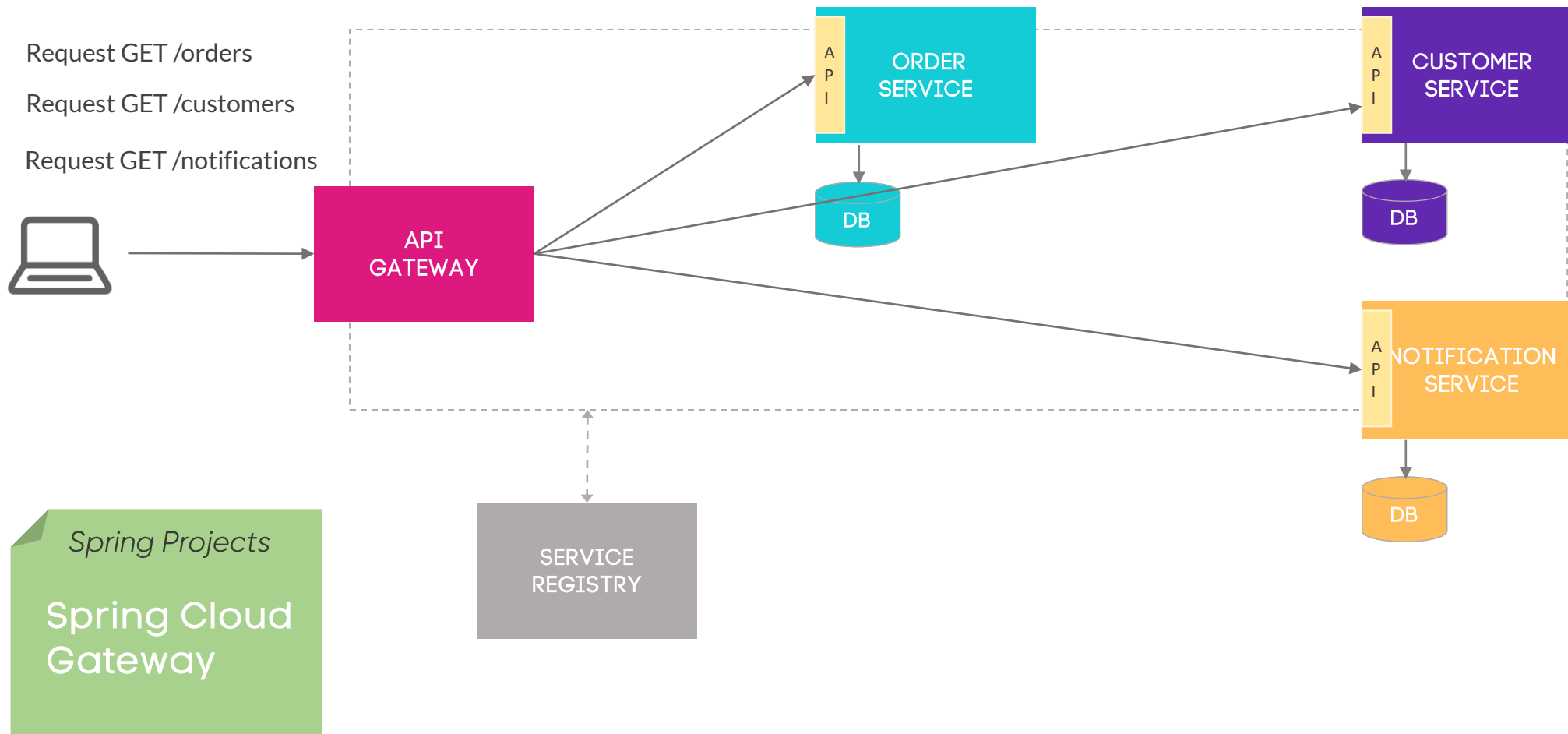
Spring Projects

Spring Cloud
Resilience4J

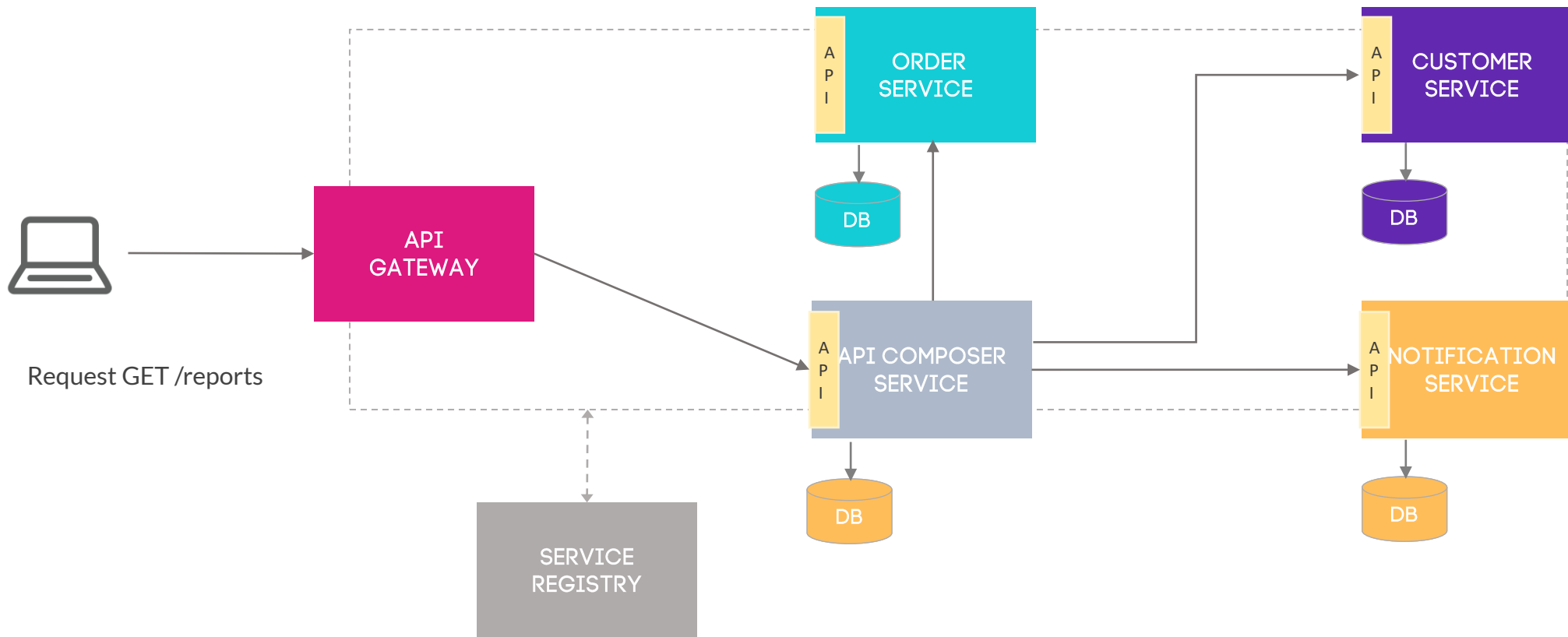


Michelli Brito
@brito_michelli

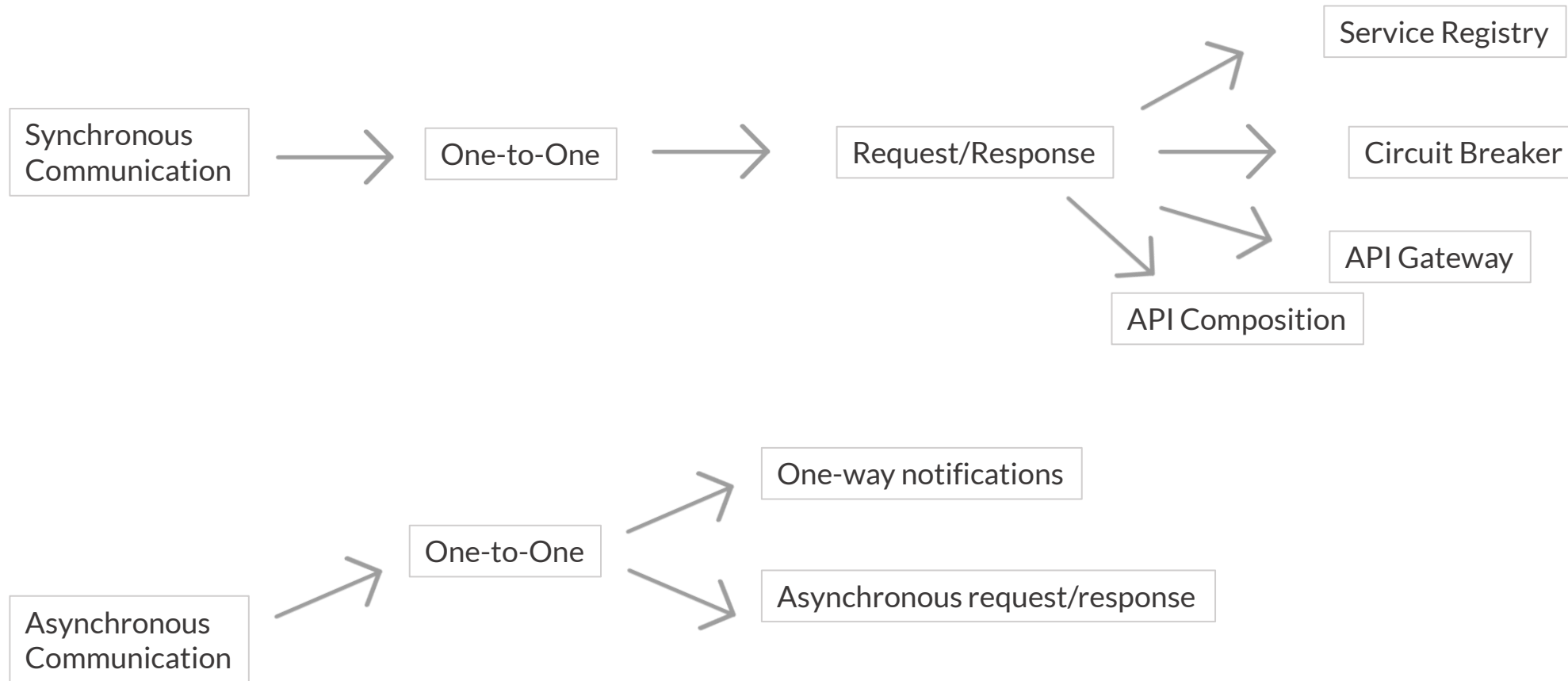
API Gateway Pattern



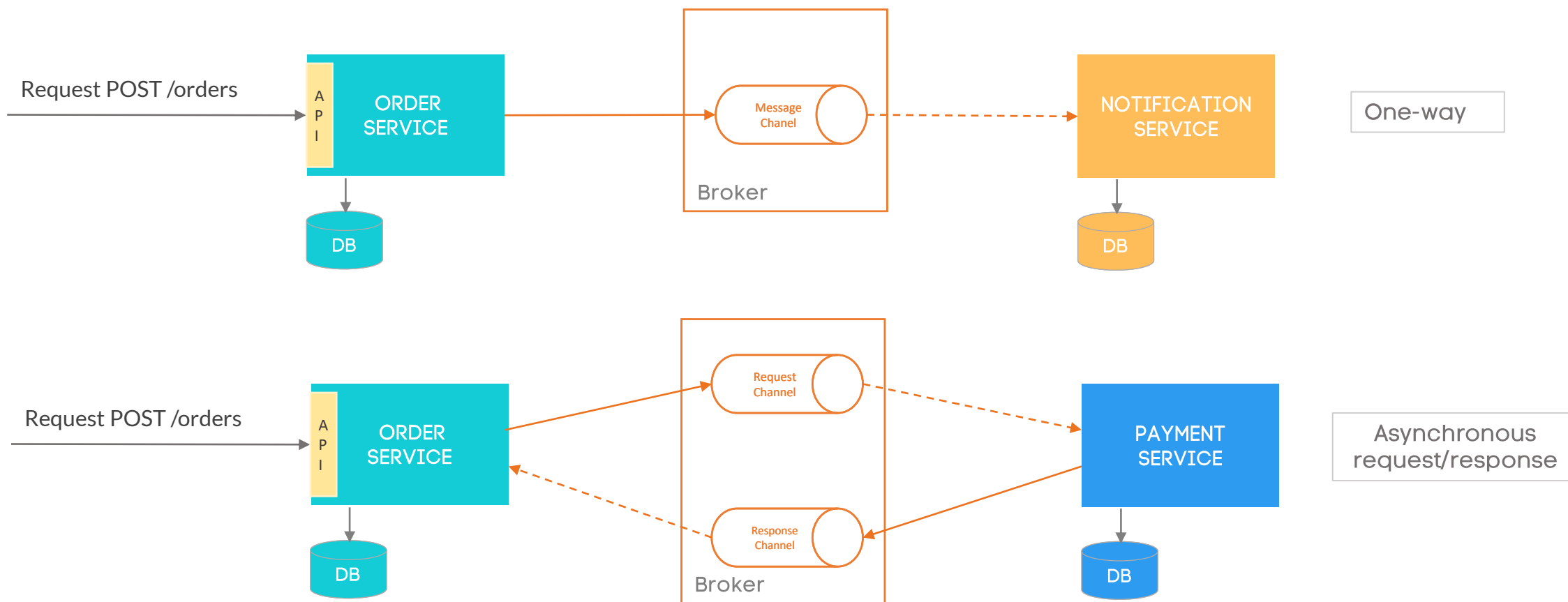
API Composition Pattern



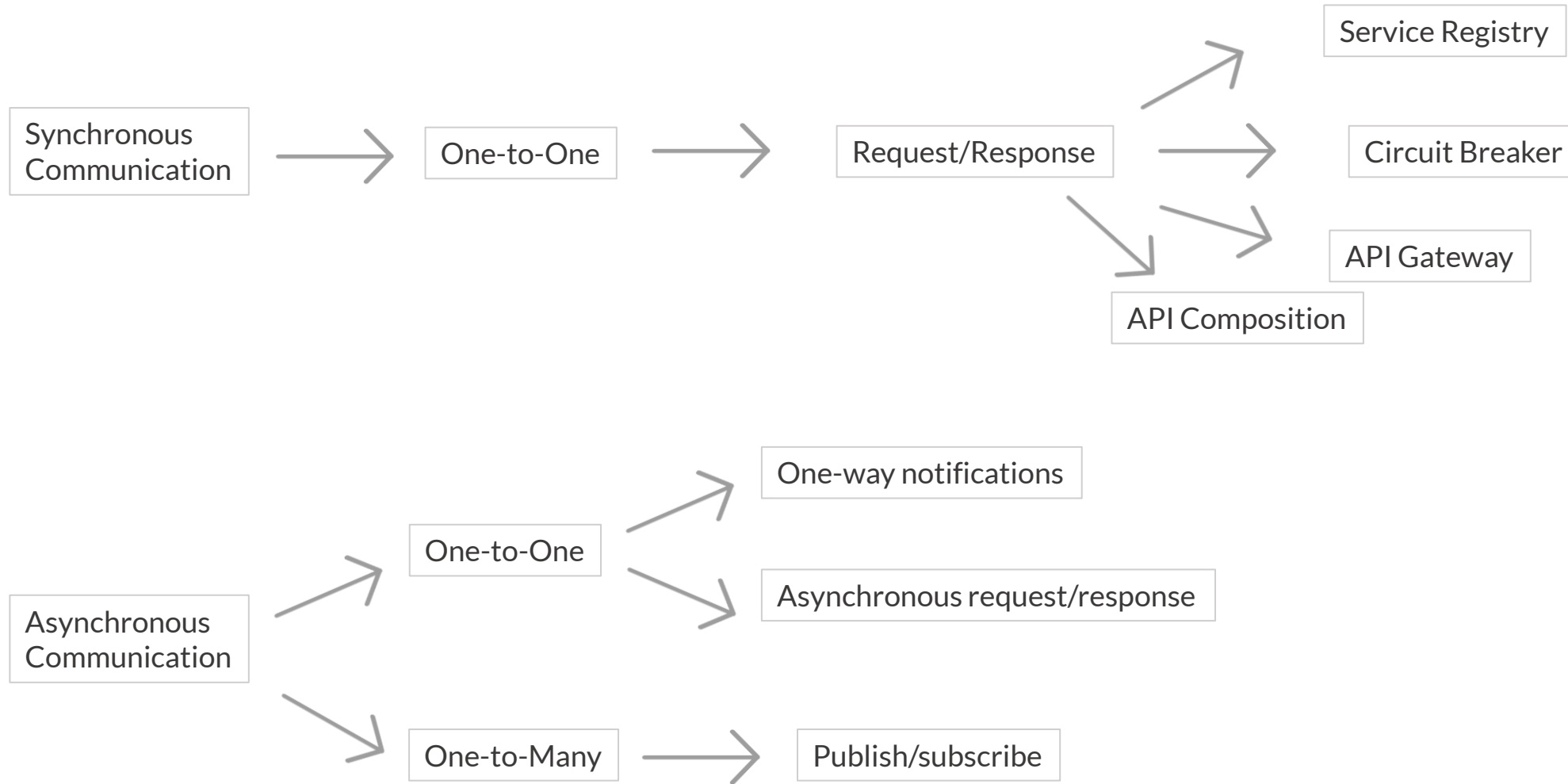
Comunicação entre Microservices



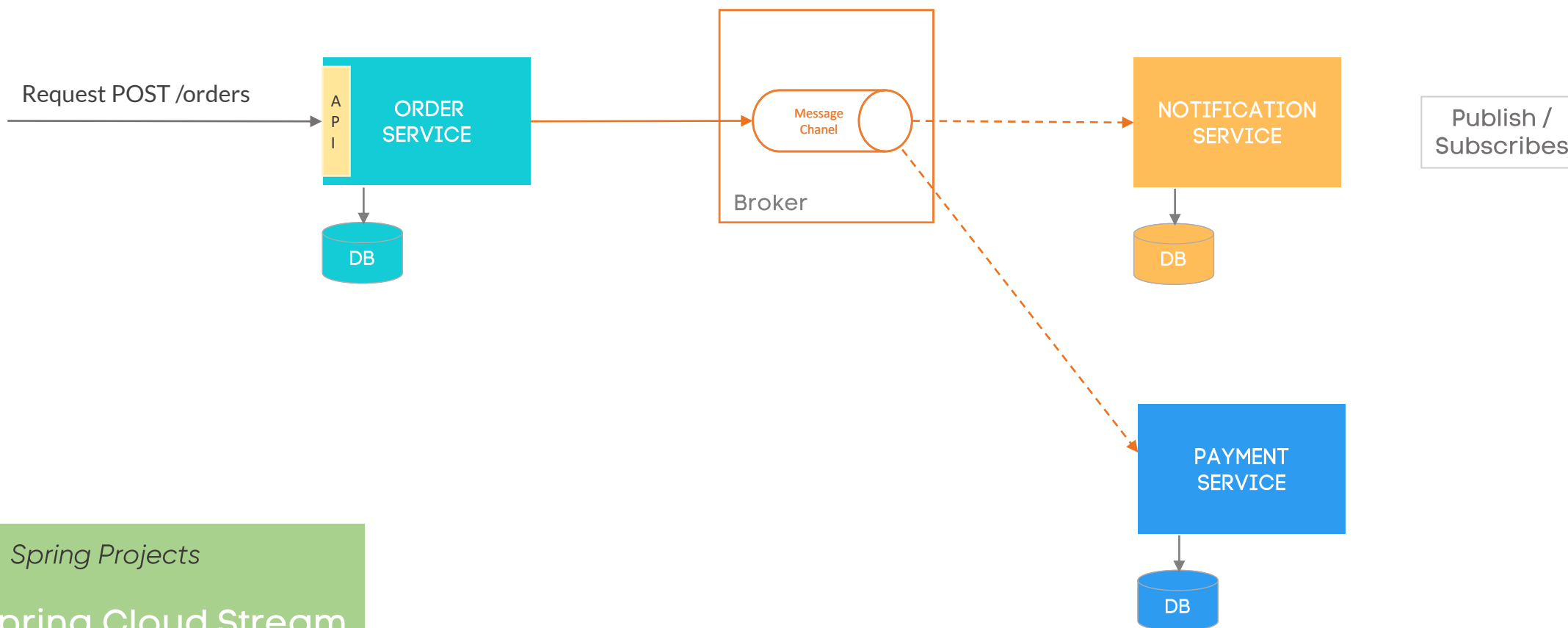
Comunicação Assíncrona via Mensageria: One-To-One



Comunicação entre Microservices

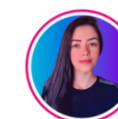


Comunicação Assíncrona via Mensageria: One-To-Many

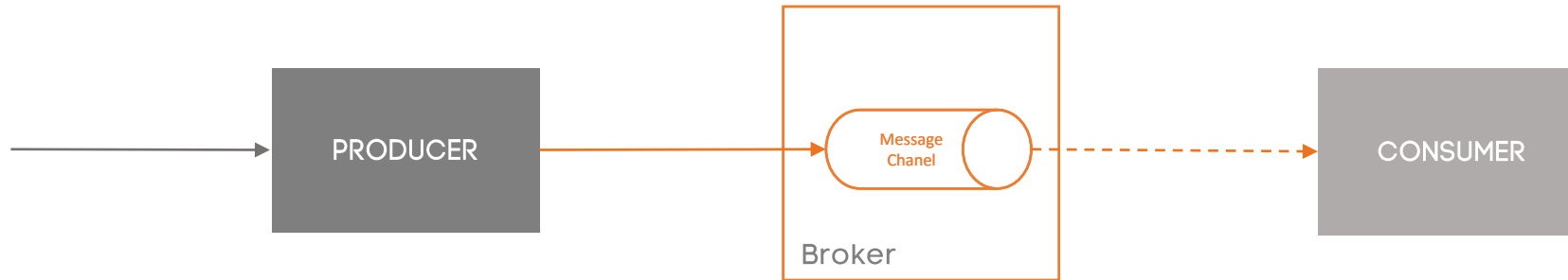


Spring Projects

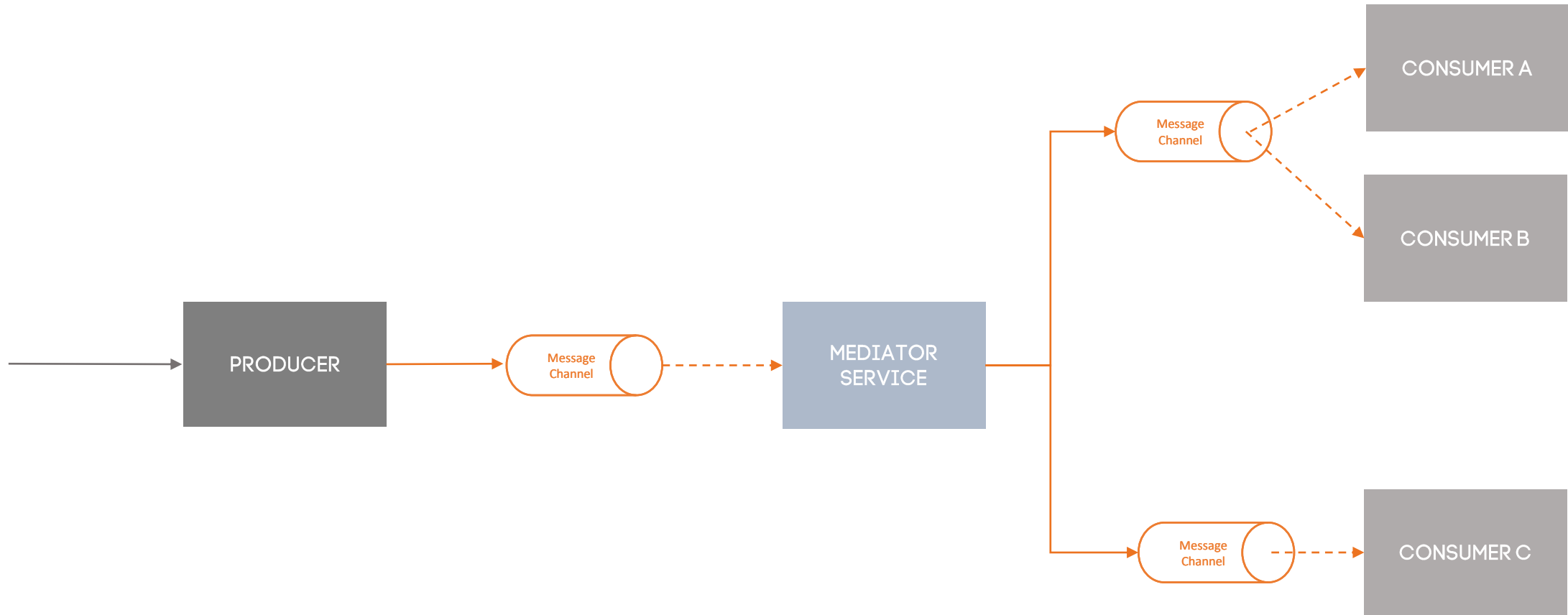
Spring Cloud Stream
Spring AMQP



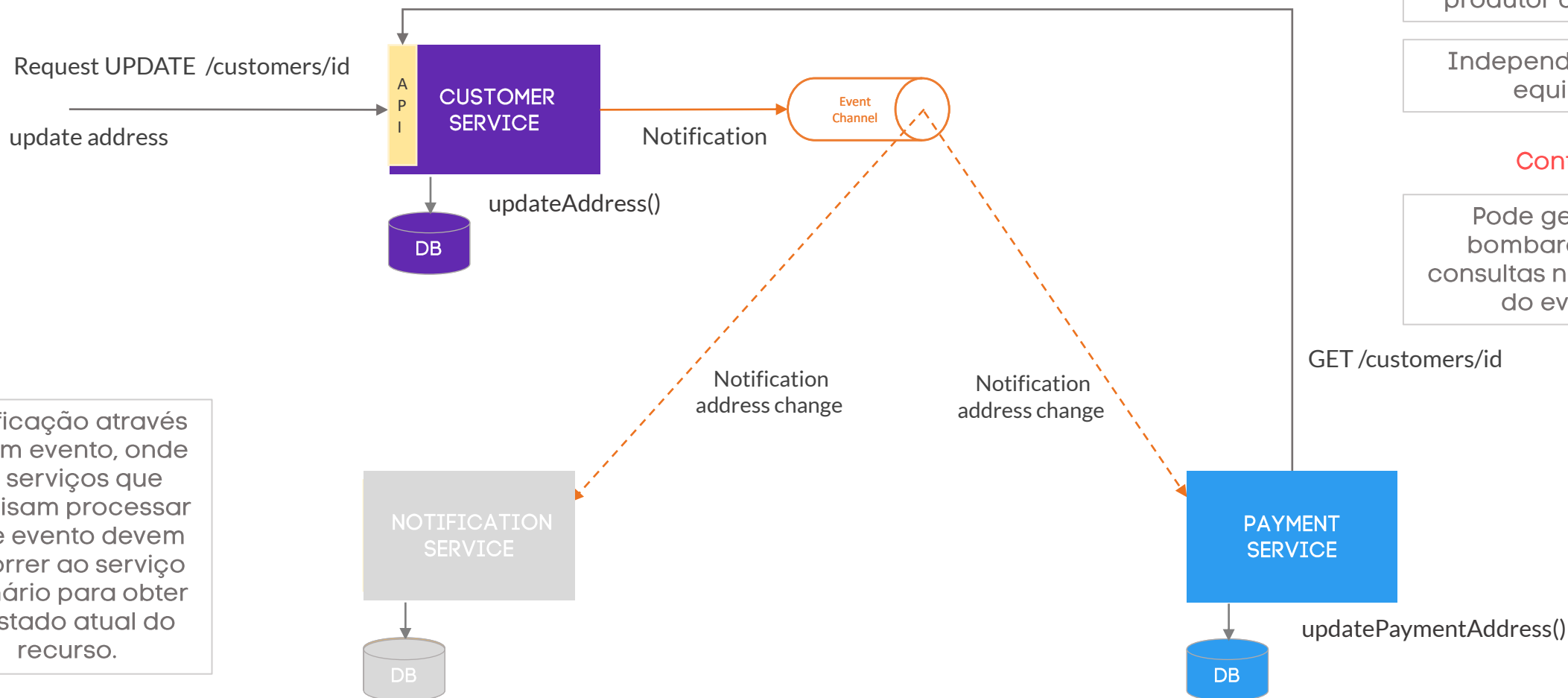
Broker Pattern



Mediator Pattern



Event Notification Pattern



Notificação através de um evento, onde os serviços que precisam processar esse evento devem recorrer ao serviço primário para obter o estado atual do recurso.

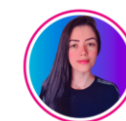
Prós

Isolamento de responsabilidade do produtor do evento

Independência de equipes

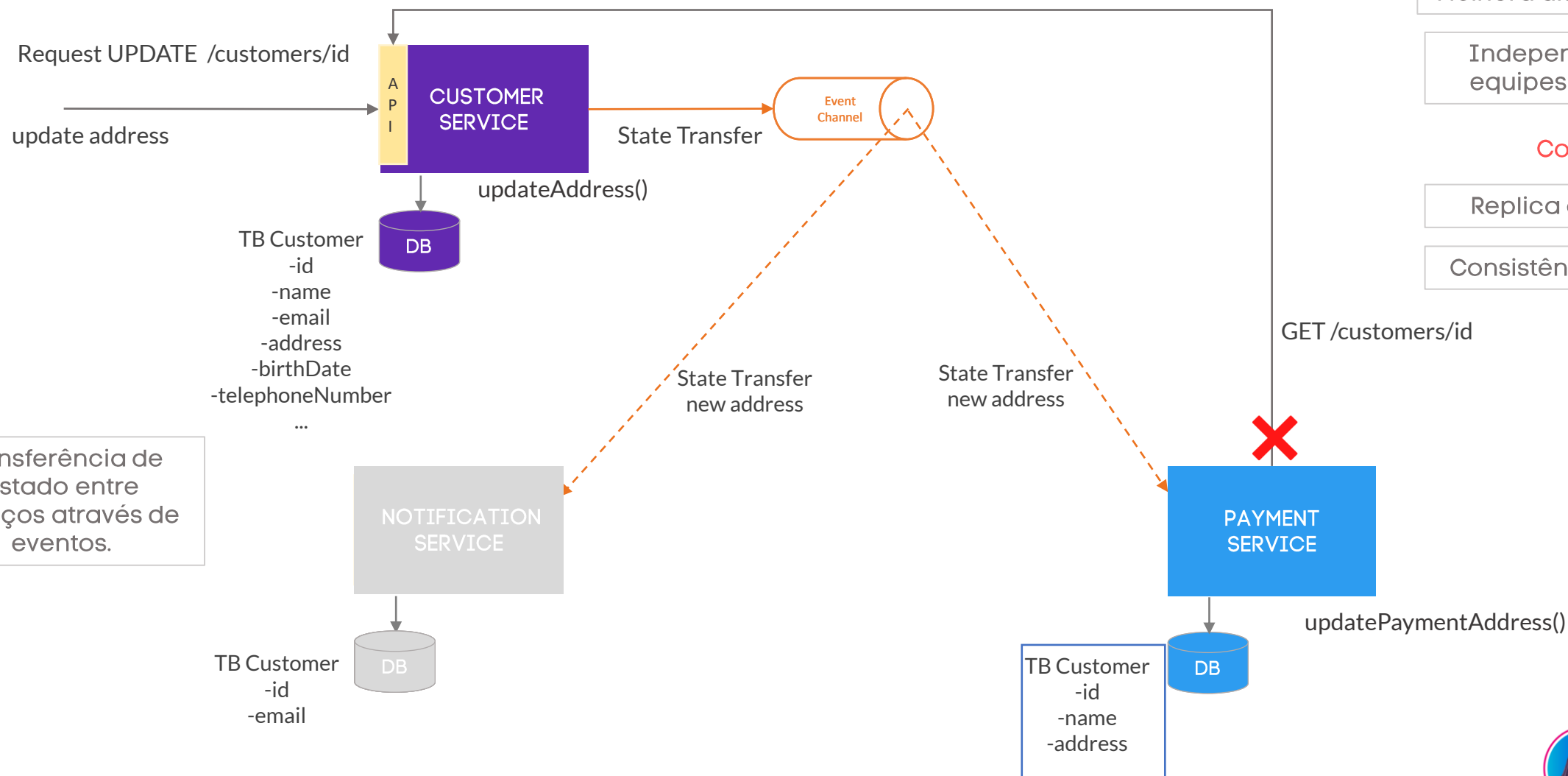
Contras

Pode gerar um bombardeio de consultas no produtor do evento



Michelli Brito
@brito_michelli

Event-Carried State Transfer Pattern



Prós

Melhora performance

Melhora disponibilidade

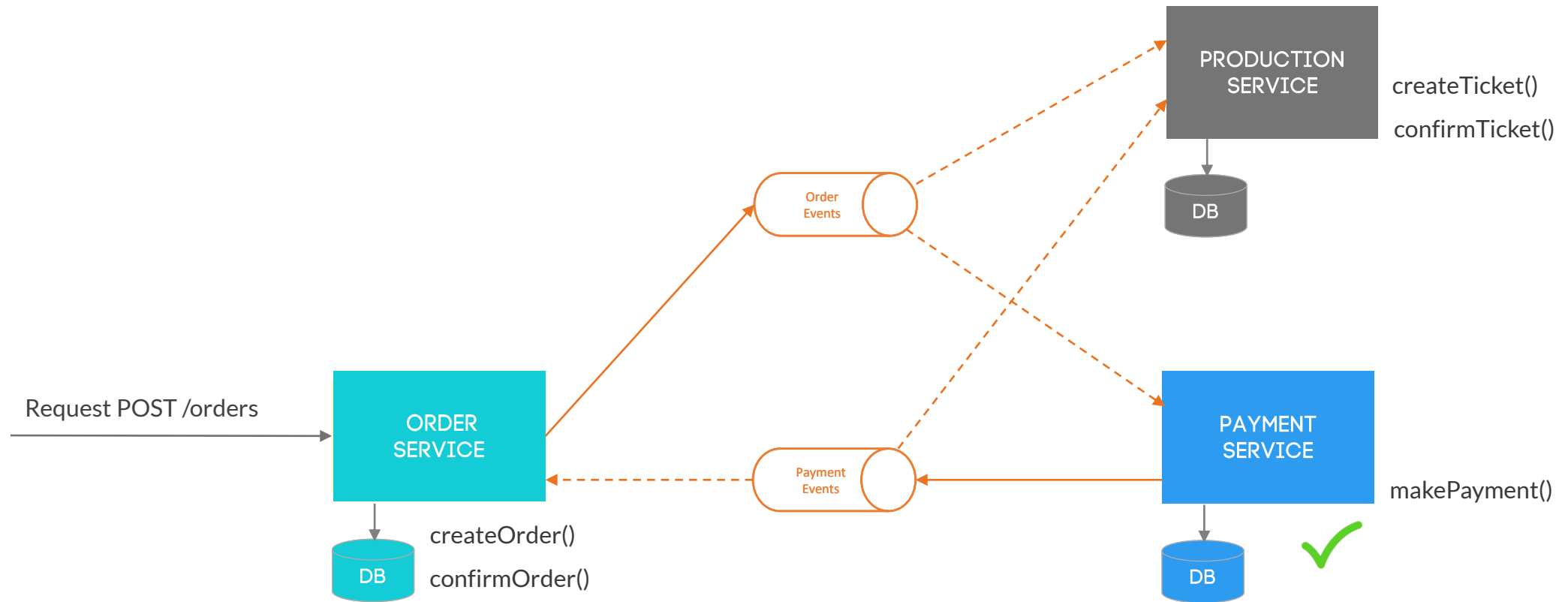
Independência de
equipes e serviços

Contras

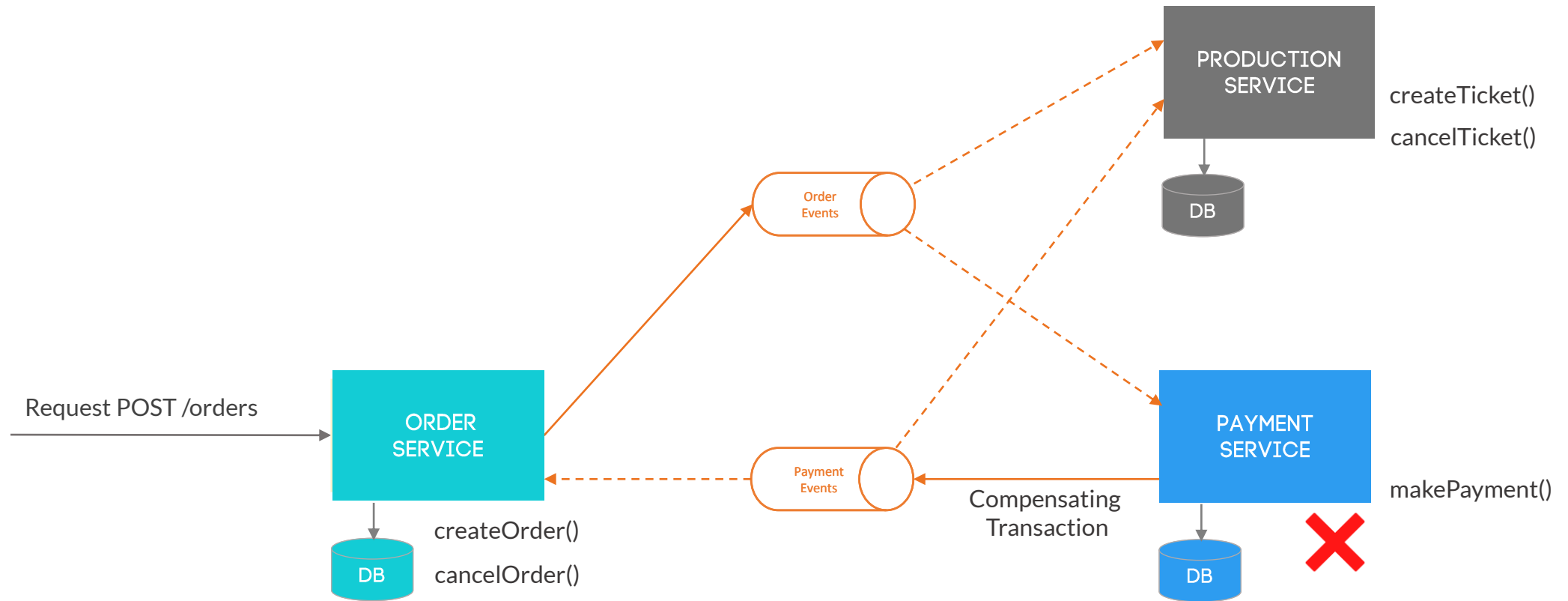
Replica dos dados

Consistência Eventual

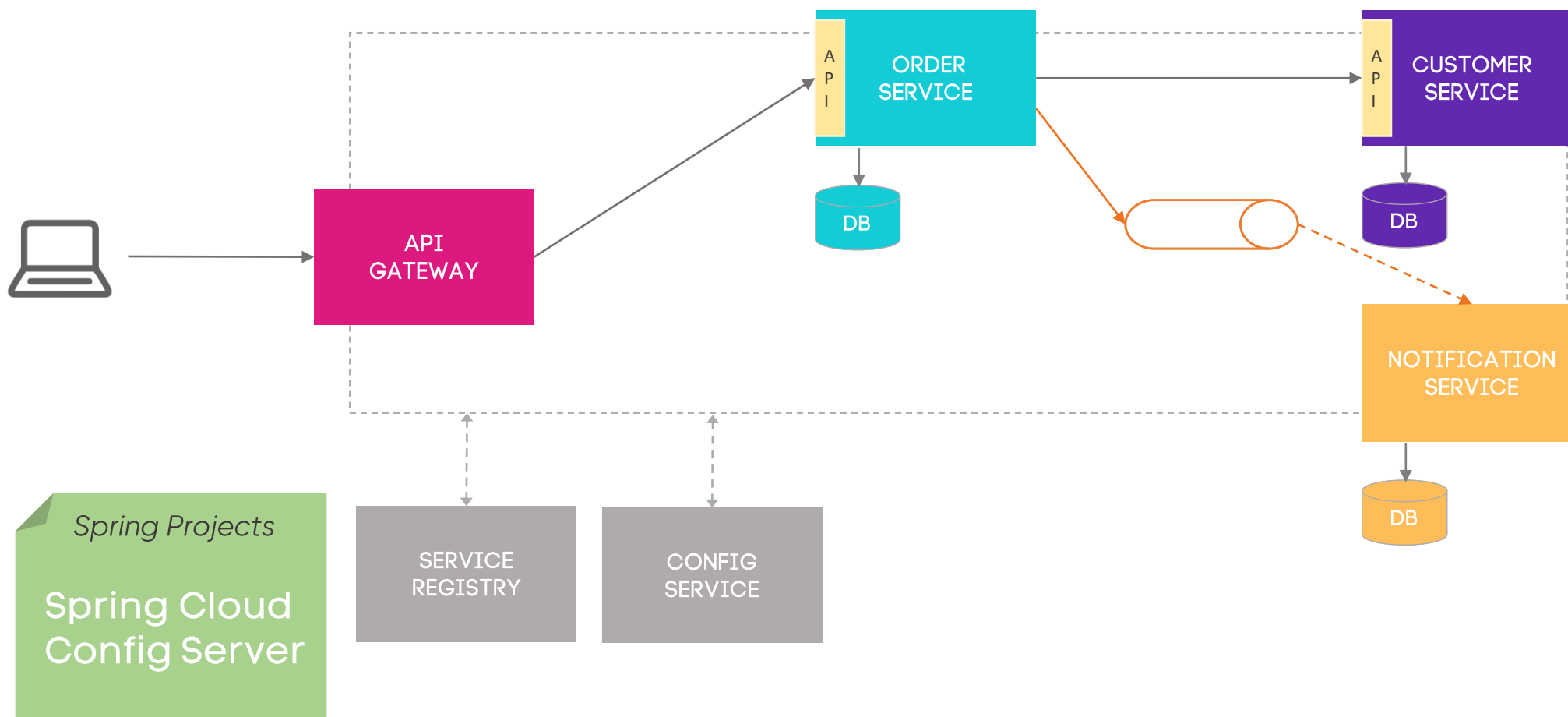
Saga Pattern



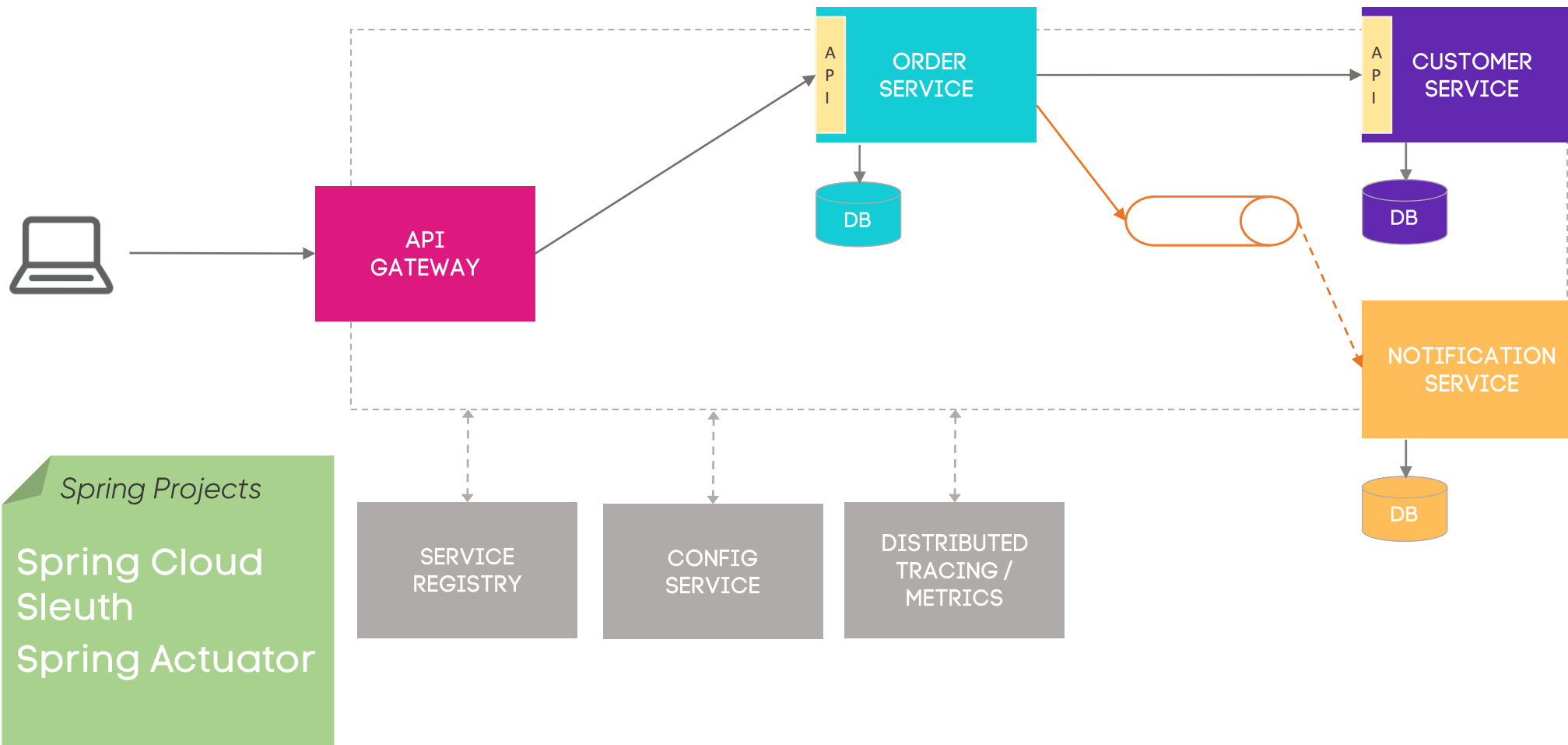
Saga Pattern



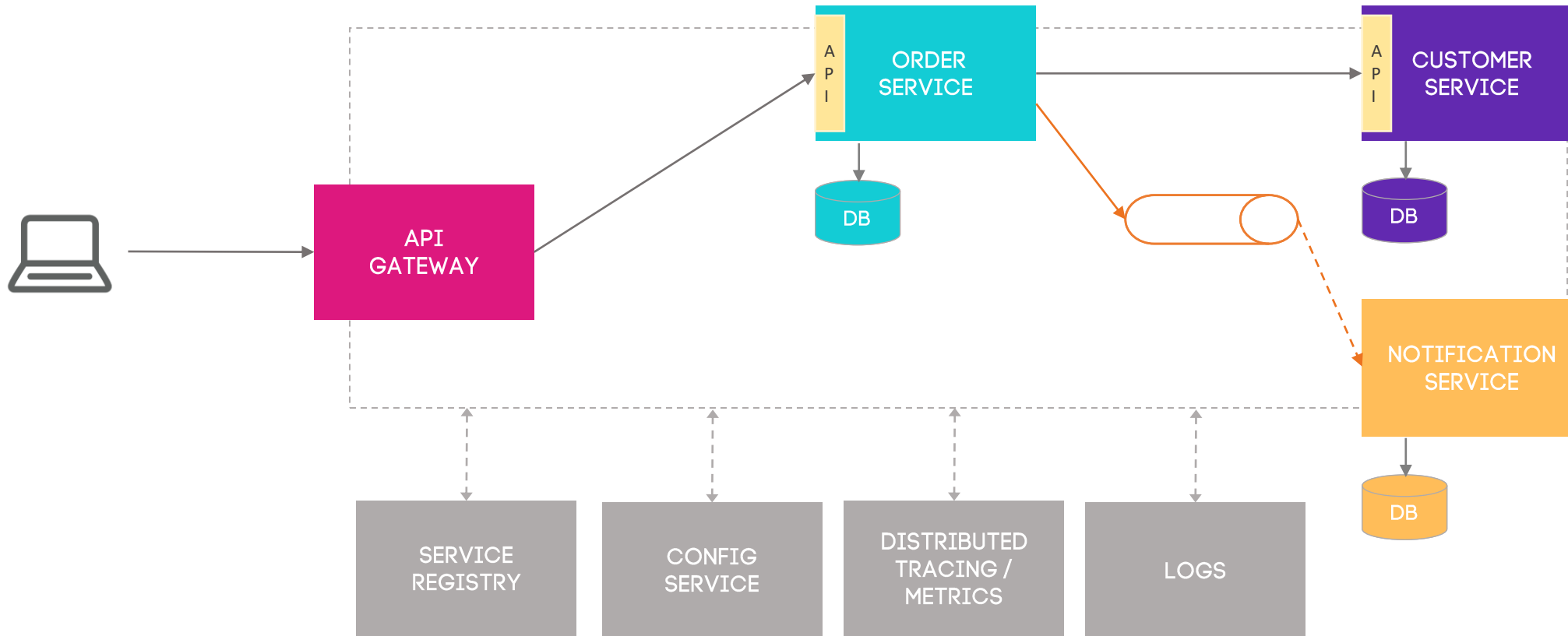
Externalized Configuration



Observability: Distributed Tracing / Metrics



Observability: Log Aggregation



Obrigada!

Não perca o próximo dia: Projeto Decoder

