

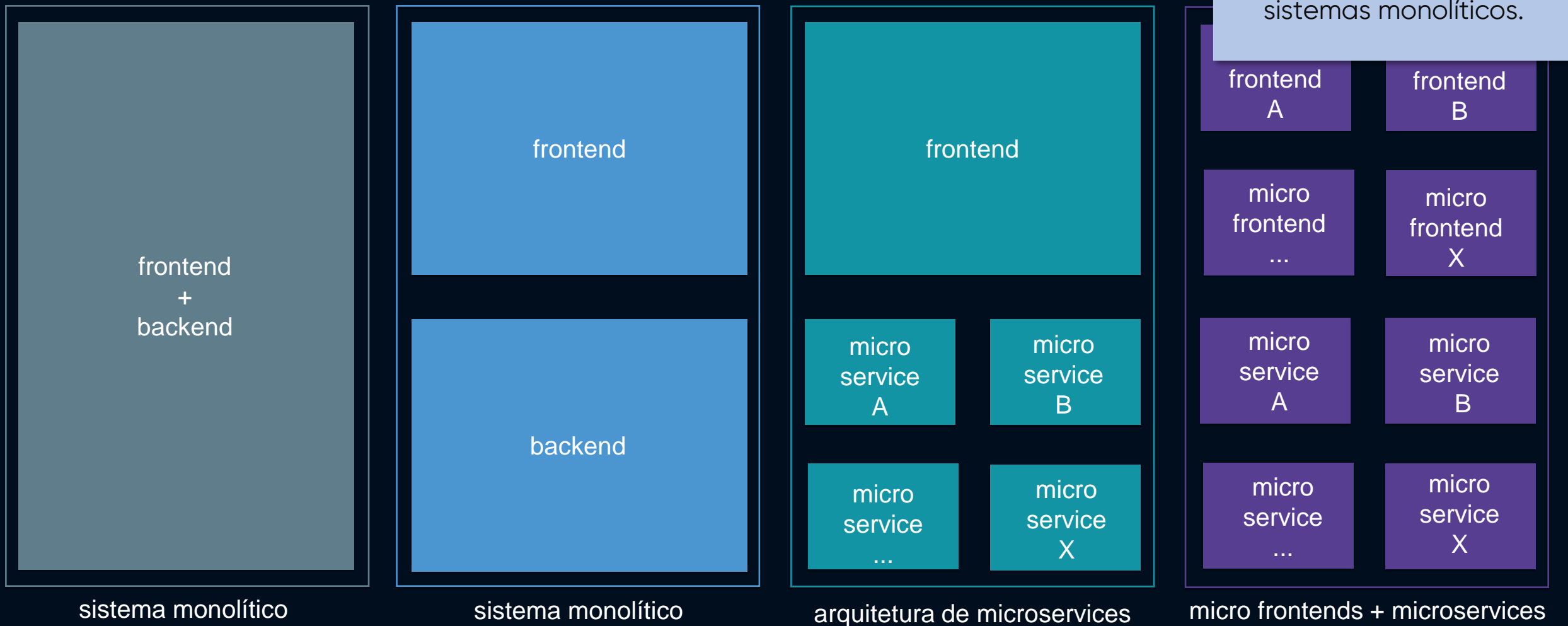


DECODER WEEK

Arquitetando o sistema Food4You

DIA 1

Evolução Arquitetural



evolução arquitetural



youtube.com/michellibrito



@brito_michelli

#decoderweek

COMPLEMENTANDO

Principais Benefícios dos Microservices

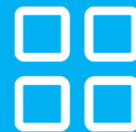
Manutenibilidade



Alta Disponibilidade

Flexibilidade
TecnológicaIndependência
das EquipesMelhor
Performance

Divisão da Complexidade do Negócio



Isolamento a Falhas



Alta Escalabilidade



Baixo Acoplamento

Melhor
TestabilidadeAgilidade nas
MudançasIsolamento
Modelagem DadosAumento
Resiliência

Arquitetura de Microservices é um modelo evolutivo ...

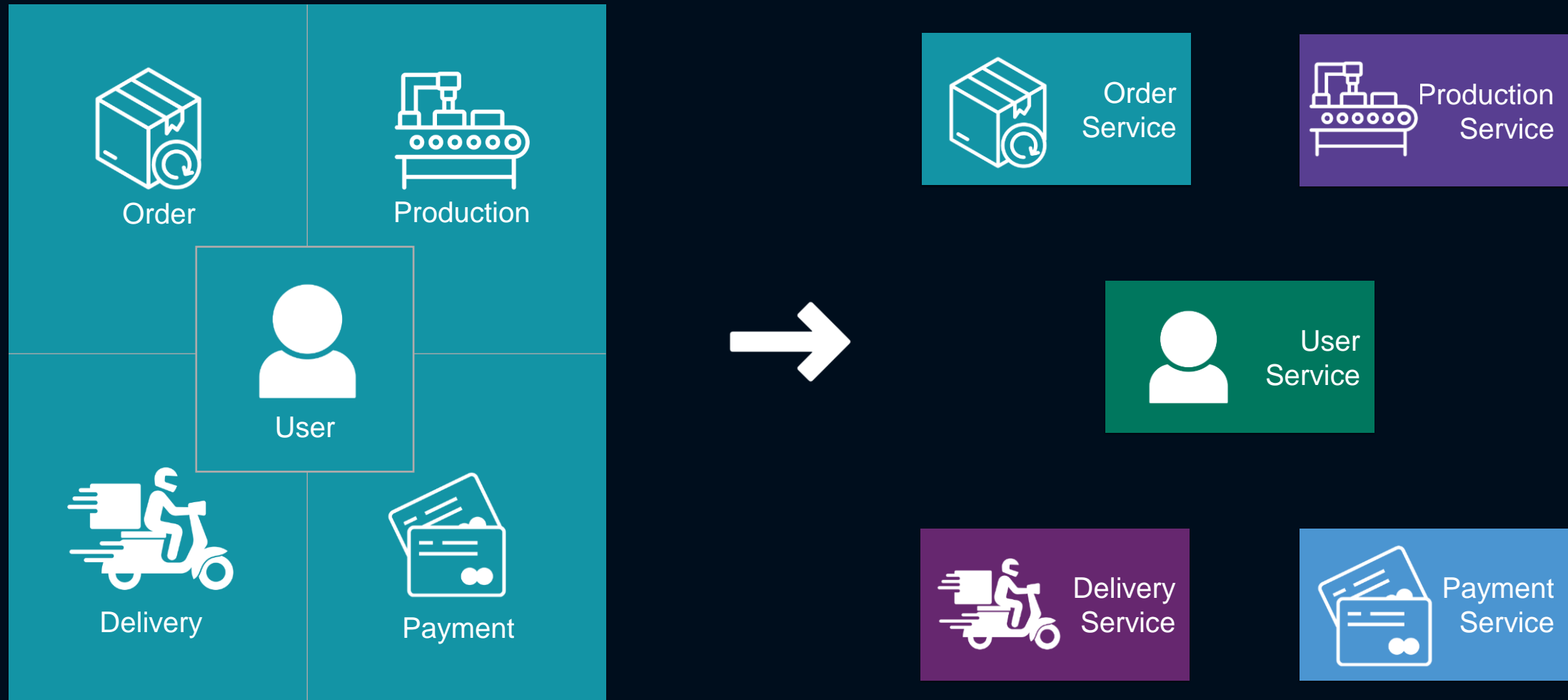
Modelo que mais facilmente se adapta as mudanças dos negócios e as modernizações tecnológicas.

Boas práticas e Padrões na modelagem de uma Arquitetura de Microservices com Spring

Modelagem na Prática

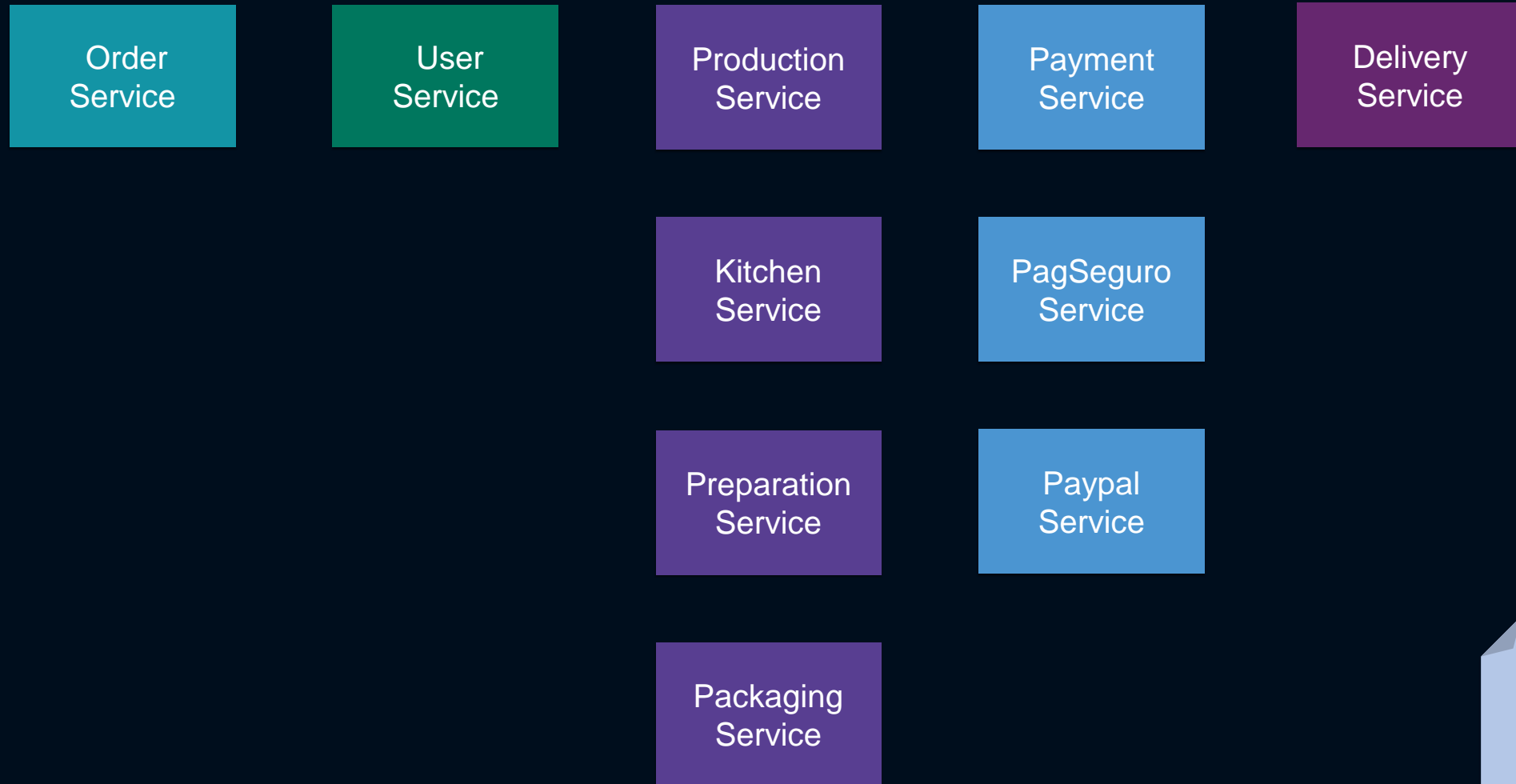


Divisão do Negócio em Microservices



Food4You domain

Granularidade dos Microservices



É essencial o conhecimento profundo do negócio.

Arquitetura de Microservices – **Food4You App**



Microservices com Spring Boot

Arquitetura do Food4You utilizando Spring Boot

Spring Boot

Spring Boot

Spring Boot

Spring Boot

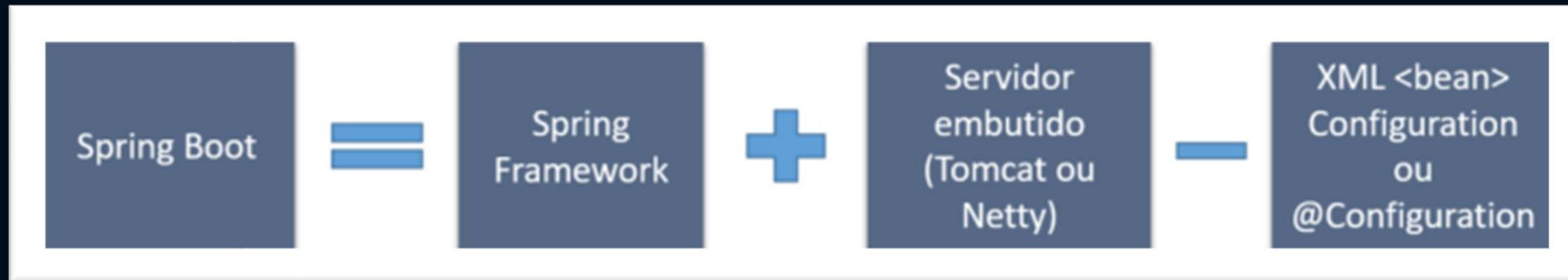
Spring Boot

Spring Projects

Spring Boot



Spring Boot



Spring Projects

Spring Boot



Spring Security



Spring Cloud



Spring Data



Spring Web



Spring AMQP



Spring HATEOAS



Spring Batch



Spring Framework



Maturidade do ecossistema Spring

Microservices



Cloud



Event Driven



Batch



Reactive



Database



Modelando Base de Dados em Microservices

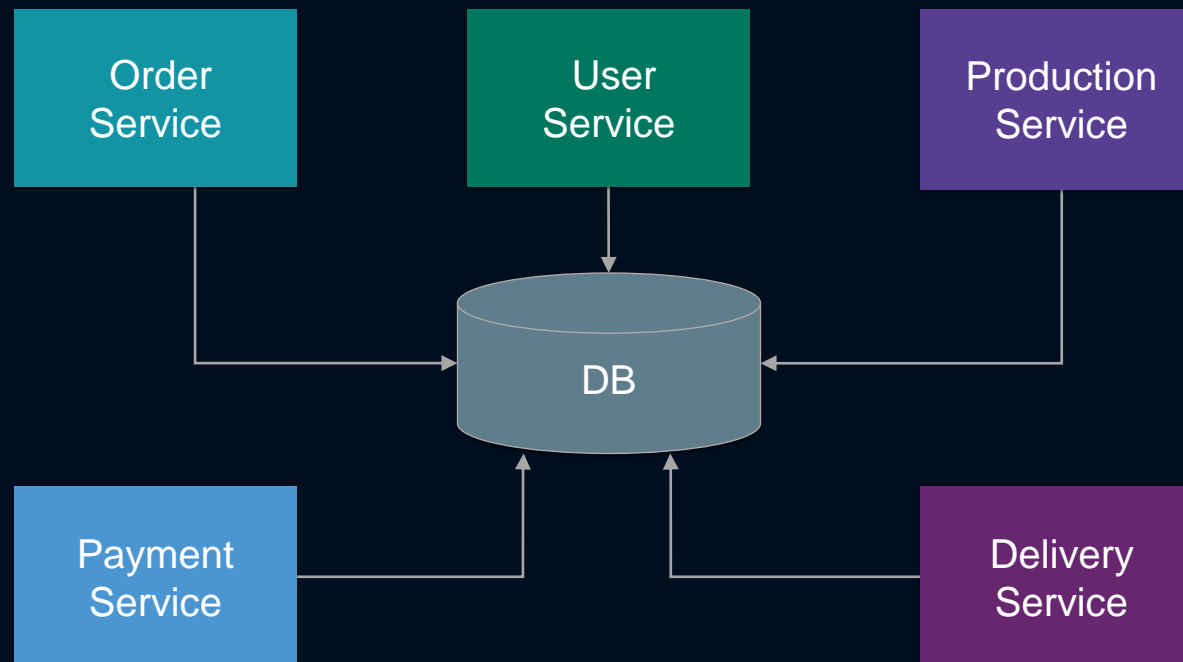
Base de dados
compartilhada

X

Base de dados
por Microservice

Base de Dados Compartilhada

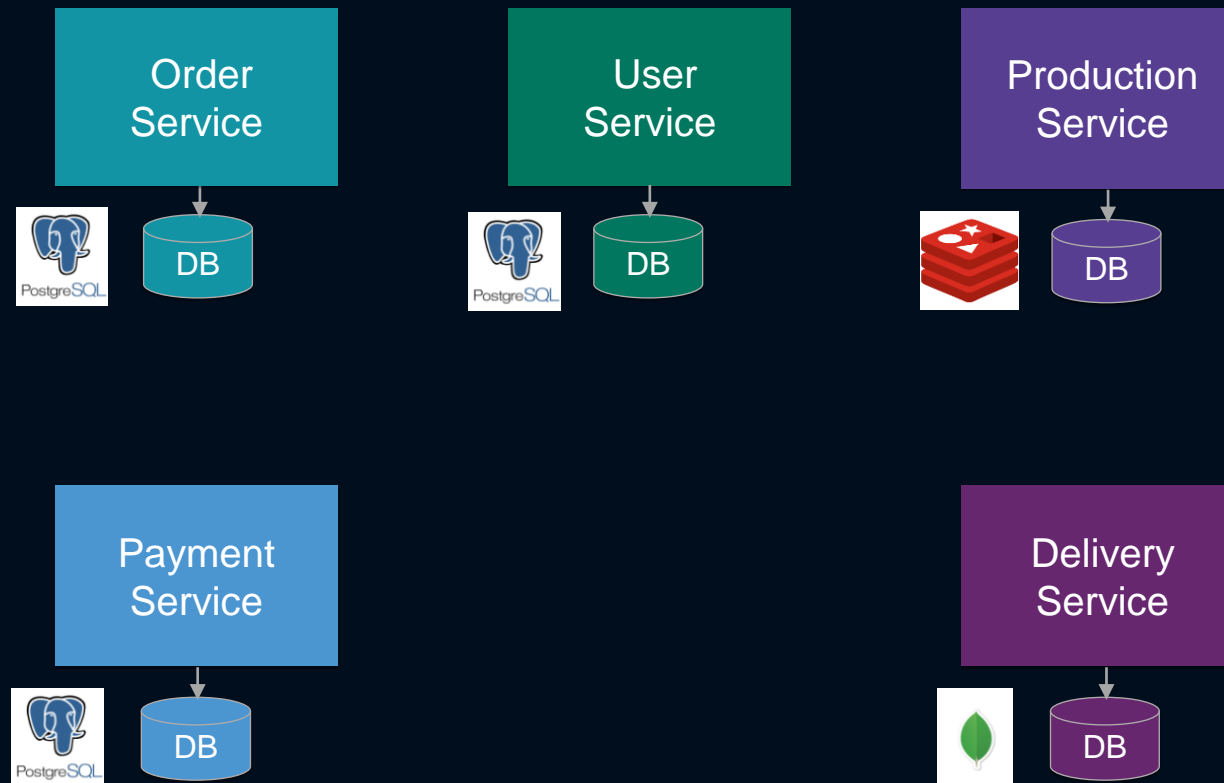
Arquitetura do Food4You com base de dados compartilhada



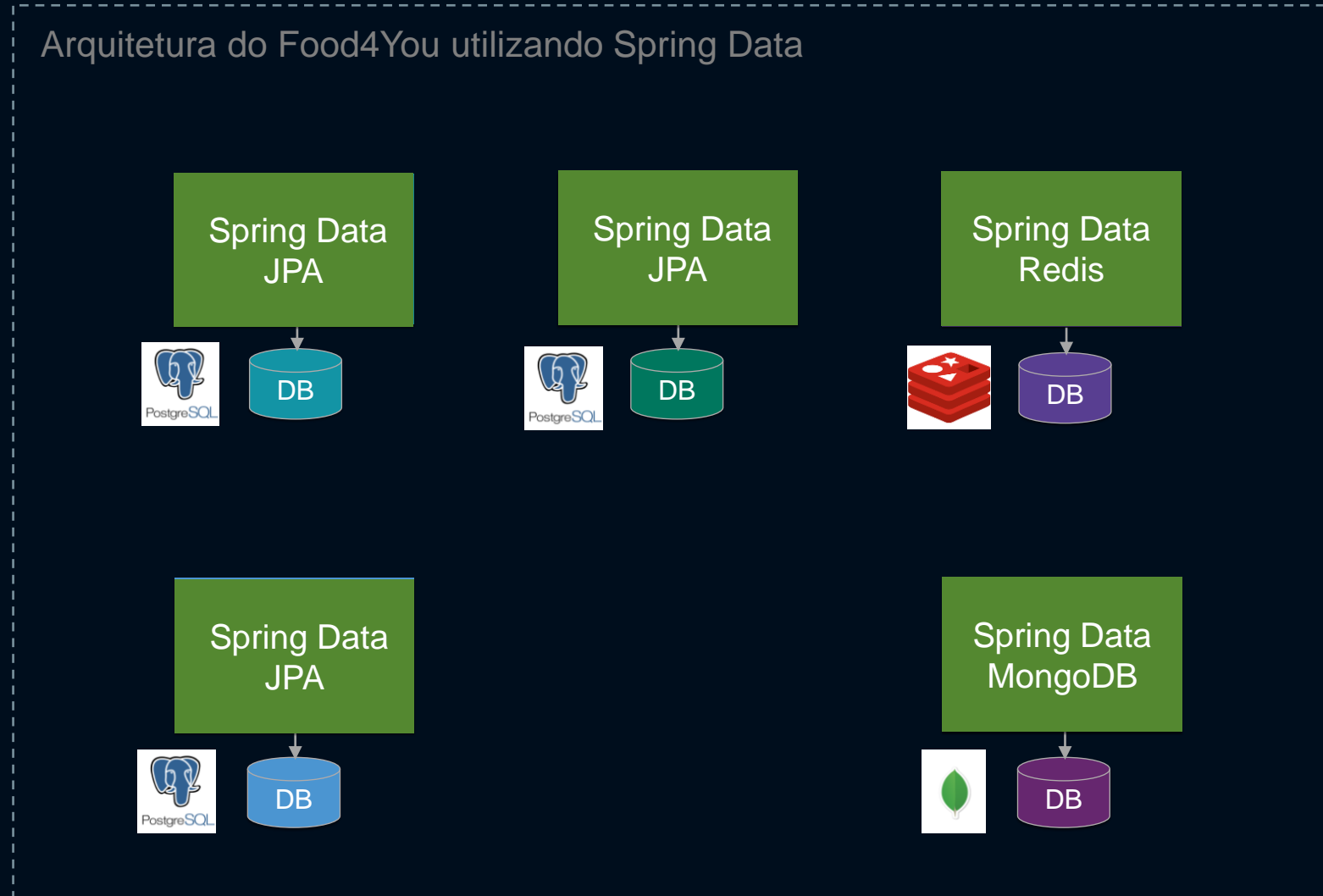
**Já temos várias vantagens de uma
Arquitetura de Microservices mesmo
com a base de dados compartilhada.**

Base de Dados por Microservices

Arquitetura do Food4You com base de dados por microservices



Base de Dados por Microservices com Spring Data



Spring Projects

Spring Data

Spring Data

Spring Data JPA

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-jpa</artifactId>  
</dependency>
```

Spring Data MongoDB

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-mongodb</artifactId>  
</dependency>
```

Spring Data JDBC

Spring Data Redis

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-mongodb-reactive</artifactId>  
</dependency>
```

Spring Data Cassandra

Spring Data Elasticsearch

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-jdbc</artifactId>  
</dependency>
```

Com base de dados por Microservices temos os dados distribuídos pela arquitetura.

Com os dados distribuídos entre os Microservices temos que lidar com a sincronia e replicação de dados, com o desafio de manter a consistência com a alta disponibilidade.

Dia 2

Gestão de dados distribuídos, teorema CAP...



Precisamos utilizar de Identificadores Universais (UUIDs) para Arquiteturas Distribuídas.

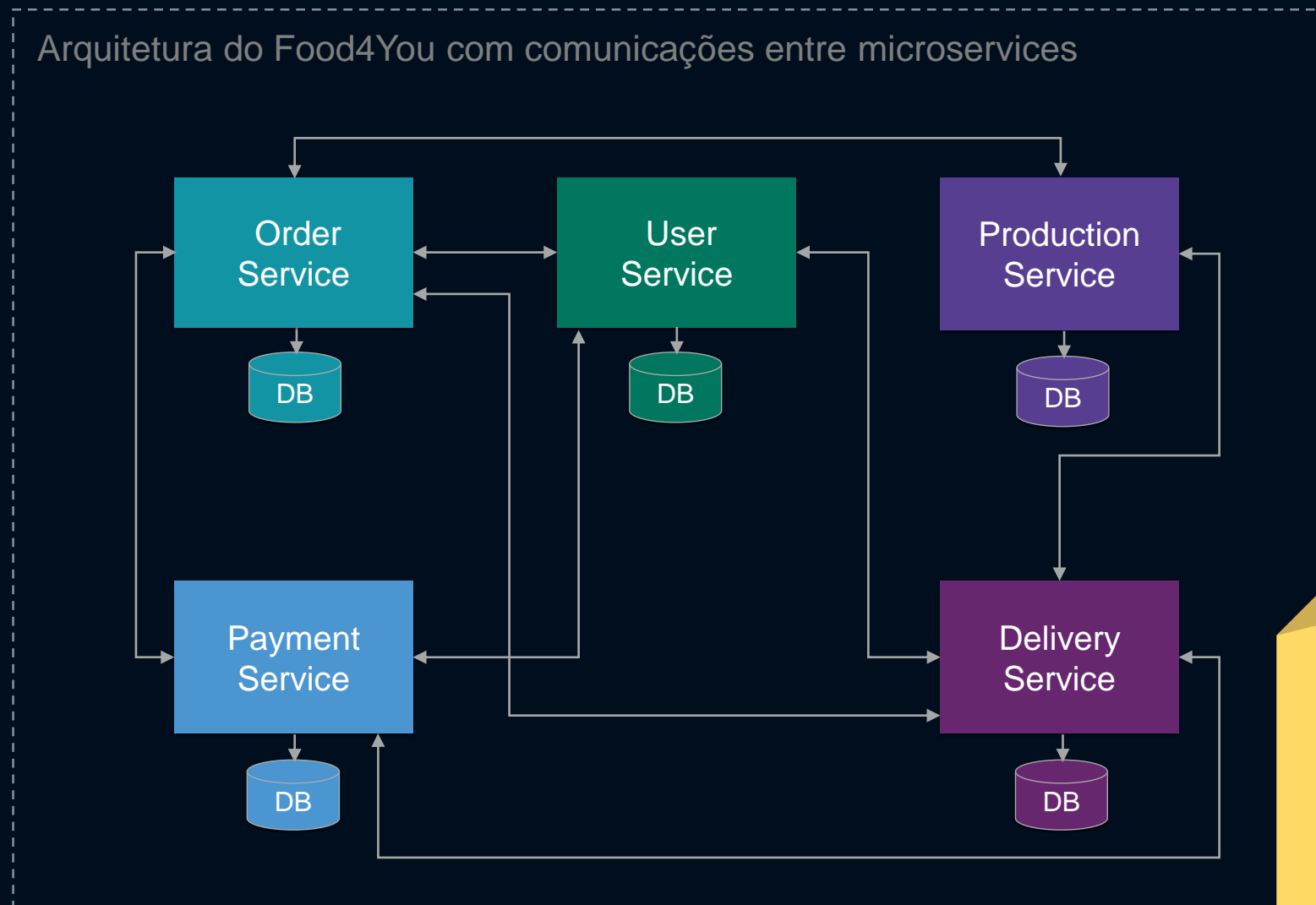
Base de Dados por
Microservices



Sincronia e Replicação
dos Dados Distribuídos



Comunicação entre
Microservices



Dia 2

Modelos de
comunicações e
sincronia e replicação
de dados.

Não existe desacoplamento absoluto entre Microservices...

A ideia é buscar o maior desacoplamento possível.

O alto acoplamento nem sempre é um problema de modelagem arquitetural, mas sim pode ser inerente ao próprio negócio, pode ser uma regra do negócio.

Microservices de Configurações

API
Gateway

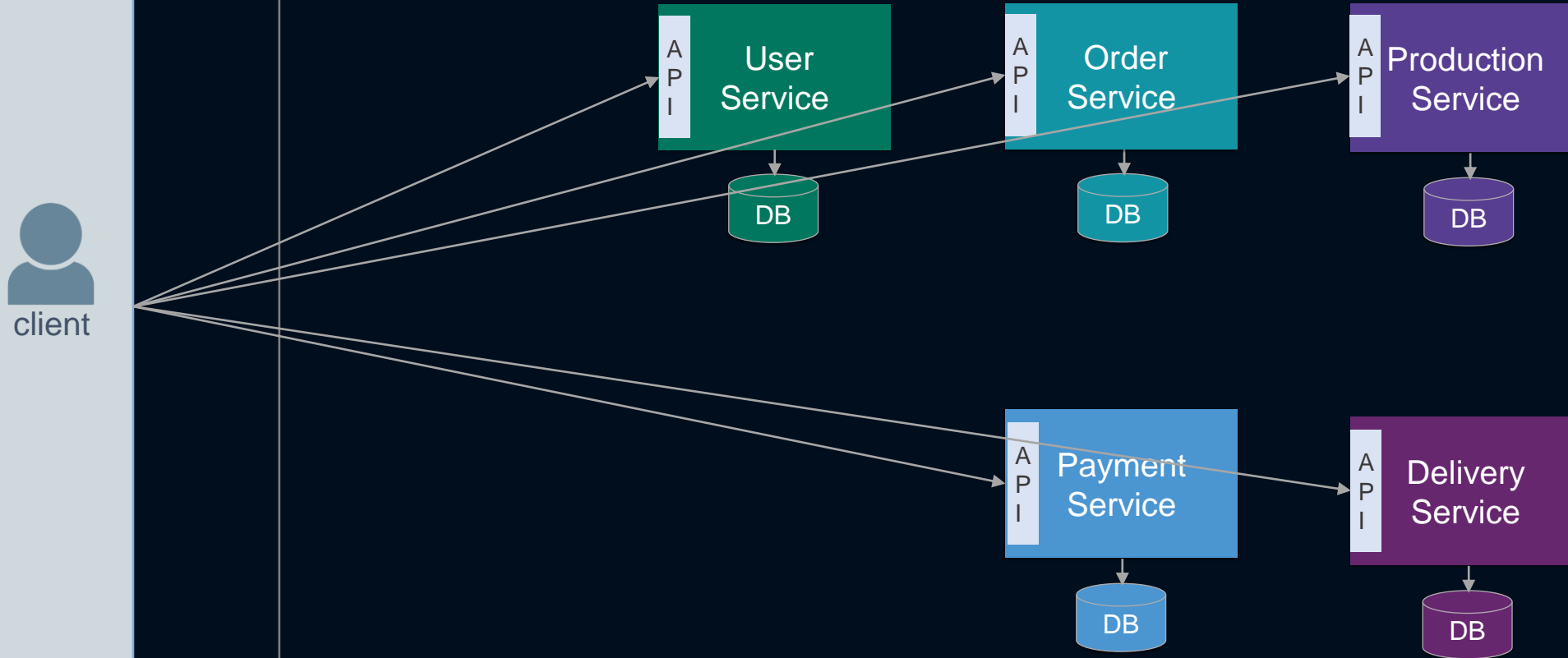
Service
Registry

Config
Server



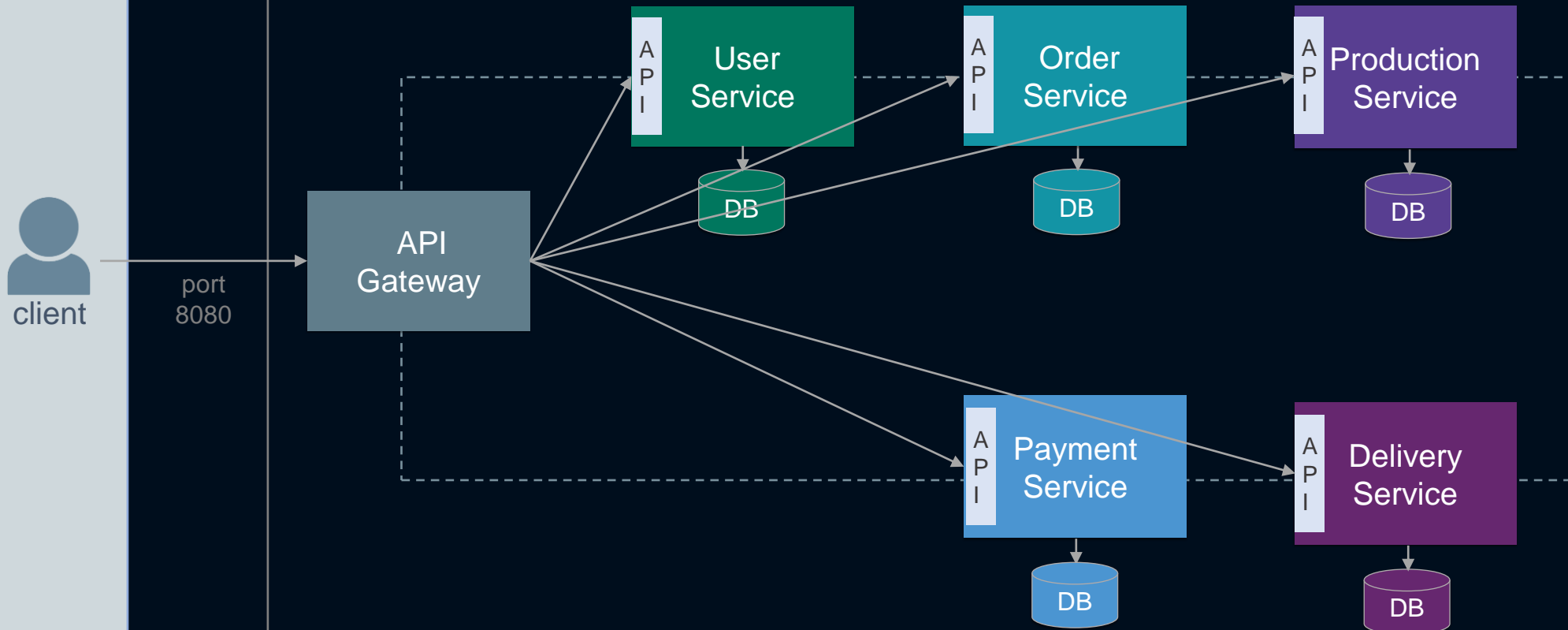
Modelando API Gateway

Arquitetura do Food4You sem API Gateway



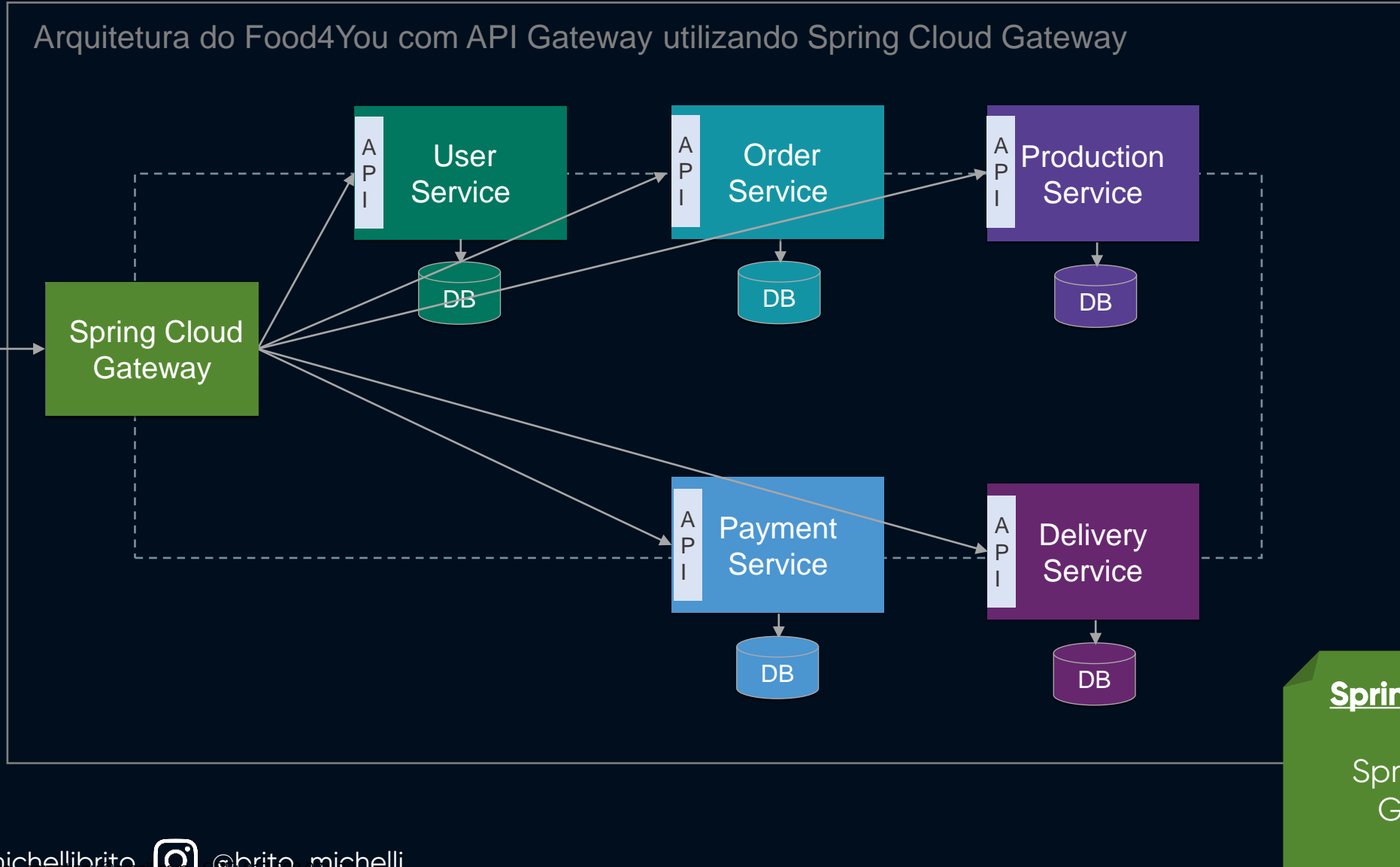
Modelando API Gateway

Arquitetura do Food4You com API Gateway



API Gateway também é um Microservice.

API Gateway com Spring Cloud Gateway

**Spring Projects**Spring Cloud
Gateway

Spring Cloud Gateway

Inserido na versão 5 do Spring Framework

Baseado no projeto Reactor

Suporte para programação Reativa

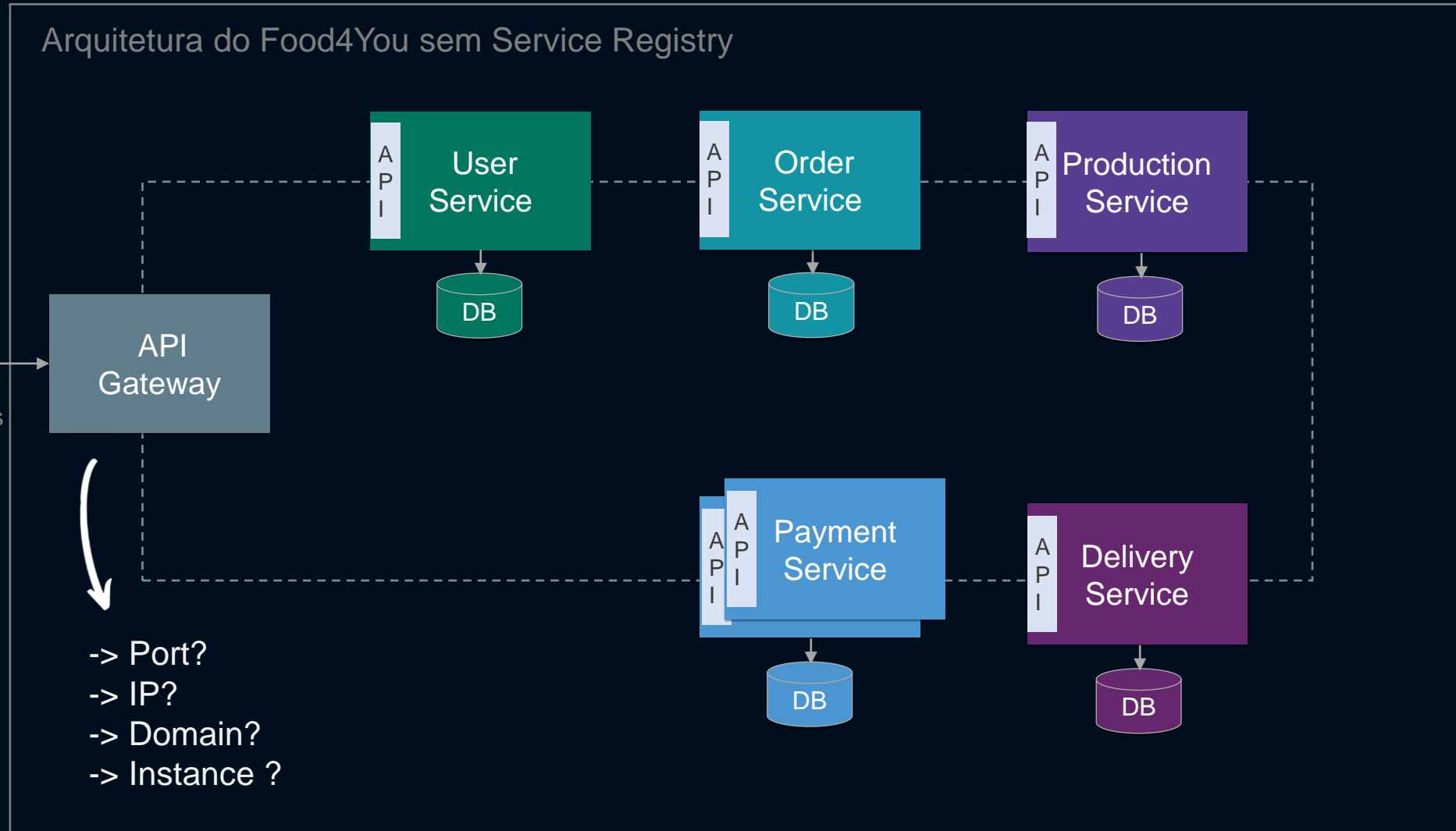
Servidor Netty

Roteamento para APIs,
definição de filtros e predicados

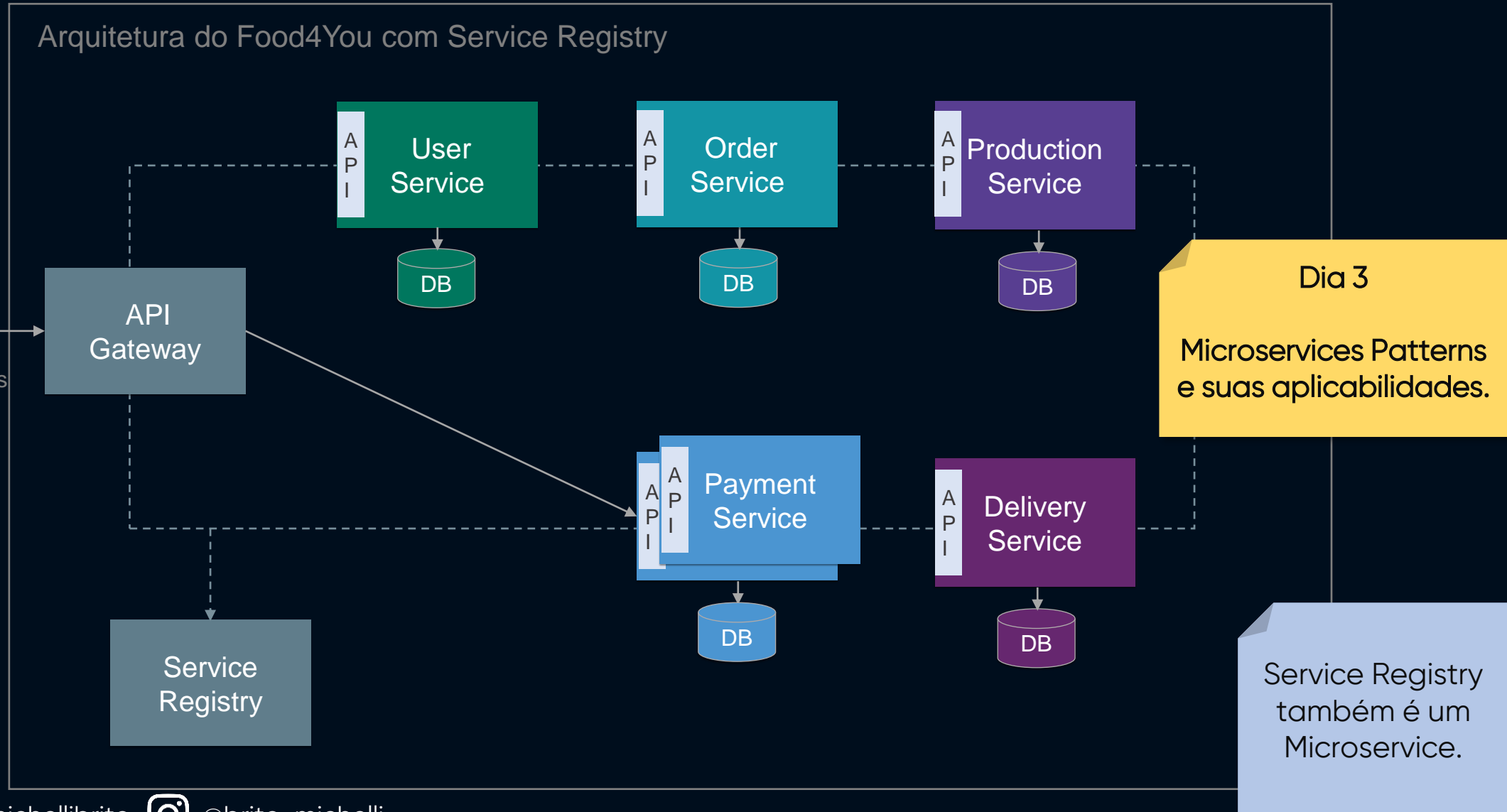
Integração com Service Registry
para registro e descoberta de
serviços

```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-starter-gateway</artifactId>  
</dependency>
```

Modelando Service Registry / Discovery

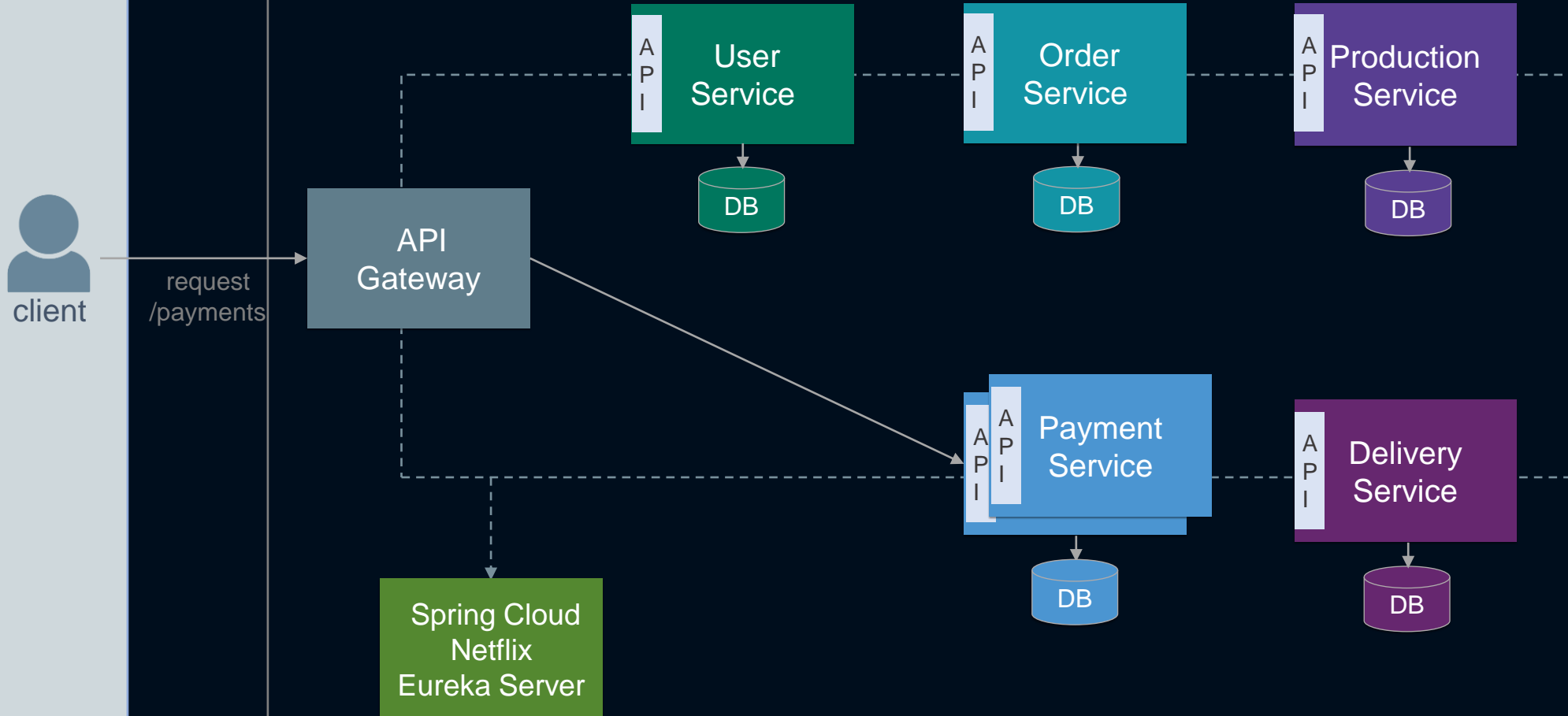


Modelando Service Registry / Discovery



Registry Discovery com Spring Cloud Netflix Eureka

Arquitetura do Food4You com Service Registry utilizando Spring Cloud Netflix Eureka



Spring Projects

Spring Cloud
Netflix Eureka

COMPLEMENTANDO

Spring Cloud Netflix Eureka



Instancias podem ser registradas em um servidor Eureka: Eureka Server

Os Eureka Clients podem descobrir essas instancias quando necessário

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
</dependency>
```

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
```

The screenshot shows the Spring Eureka web interface. At the top, there's a navigation bar with the 'spring Eureka' logo and links for 'HOME' and 'LAST 1000 SINCE STARTUP'. Below this, the 'System Status' section displays a table with system metrics. The 'DS Replicas' section shows 'localhost' as the only replica. The 'Instances currently registered with Eureka' section shows a table with columns for Application, AMIs, Availability Zones, and Status, but it indicates 'No instances available'. The 'General Info' section at the bottom provides a table of system information.

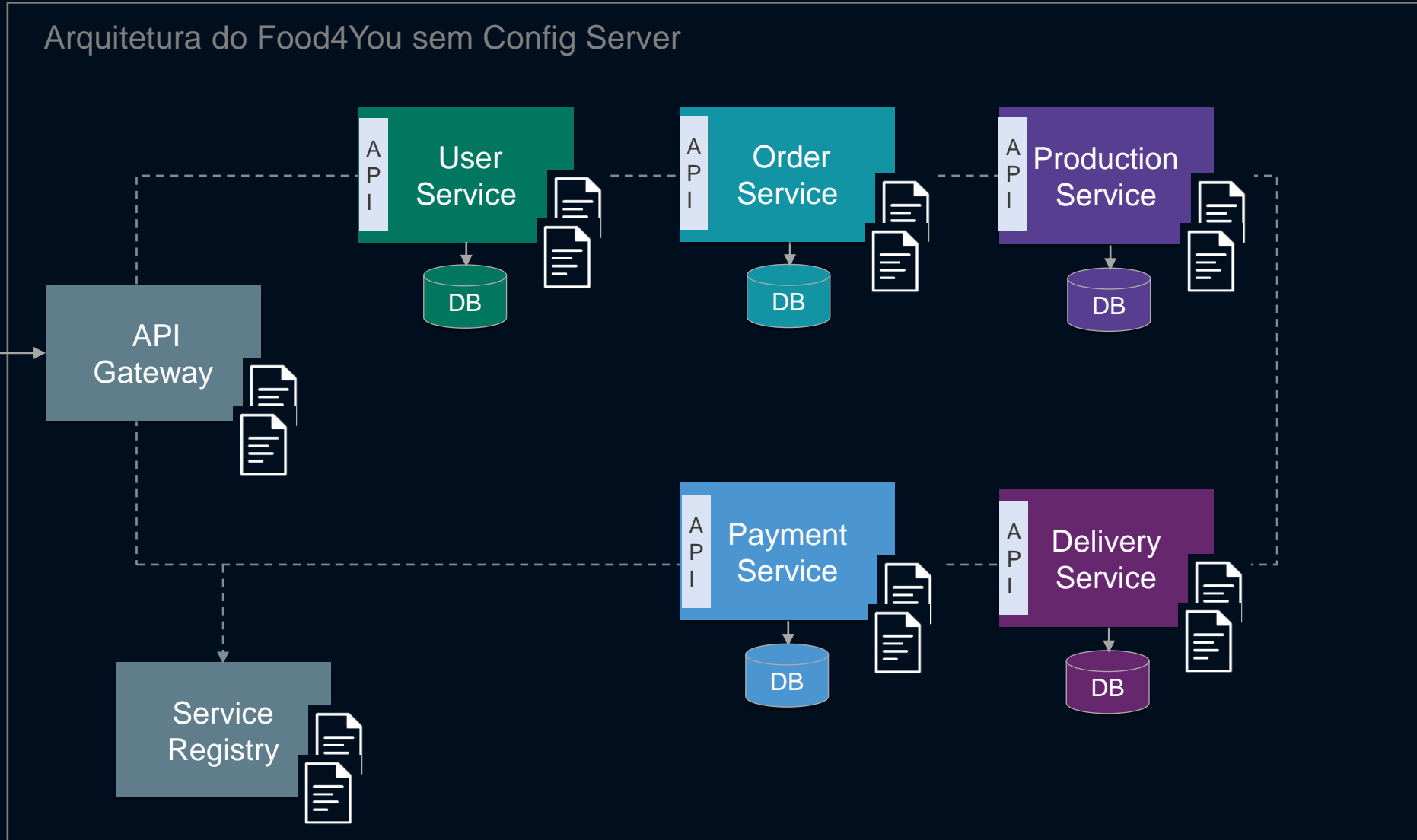
System Status	
Environment	N/A
Data center	N/A
Current time	2022-02-20T09:24:42 -0300
Uptime	00:02
Lease expiration enabled	false
Renews threshold	1
Renews (last min)	0

DS Replicas	
localhost	

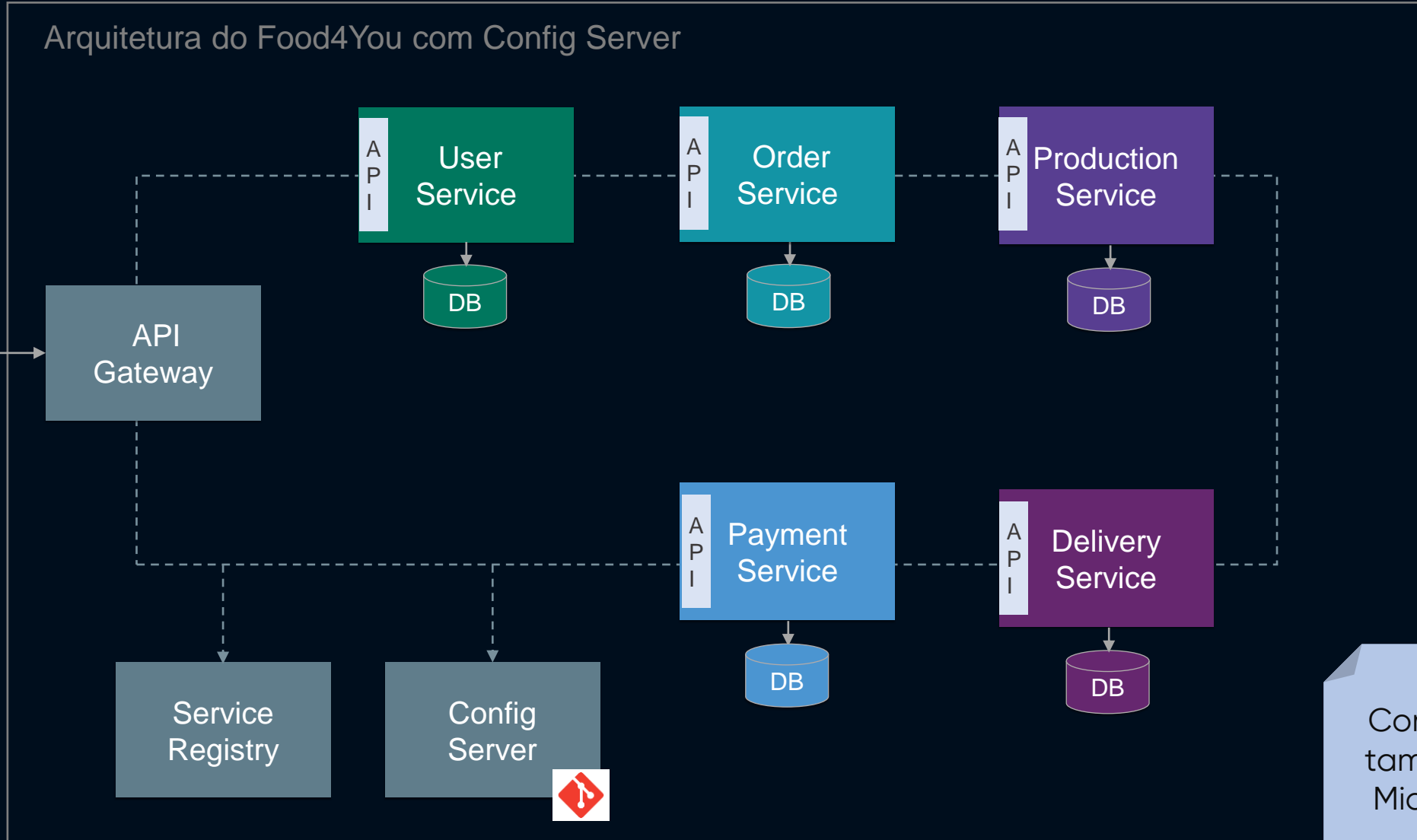
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
No instances available			

General Info	
Name	Value
total-avail-memory	154mb
num-of-cpus	8
current-memory-usage	50mb (32%)
server-up-time	00:02
registered-replicas	http://localhost:8761/eureka/
unavailable-replicas	http://localhost:8761/eureka/
available-replicas	

Modelando Config Server

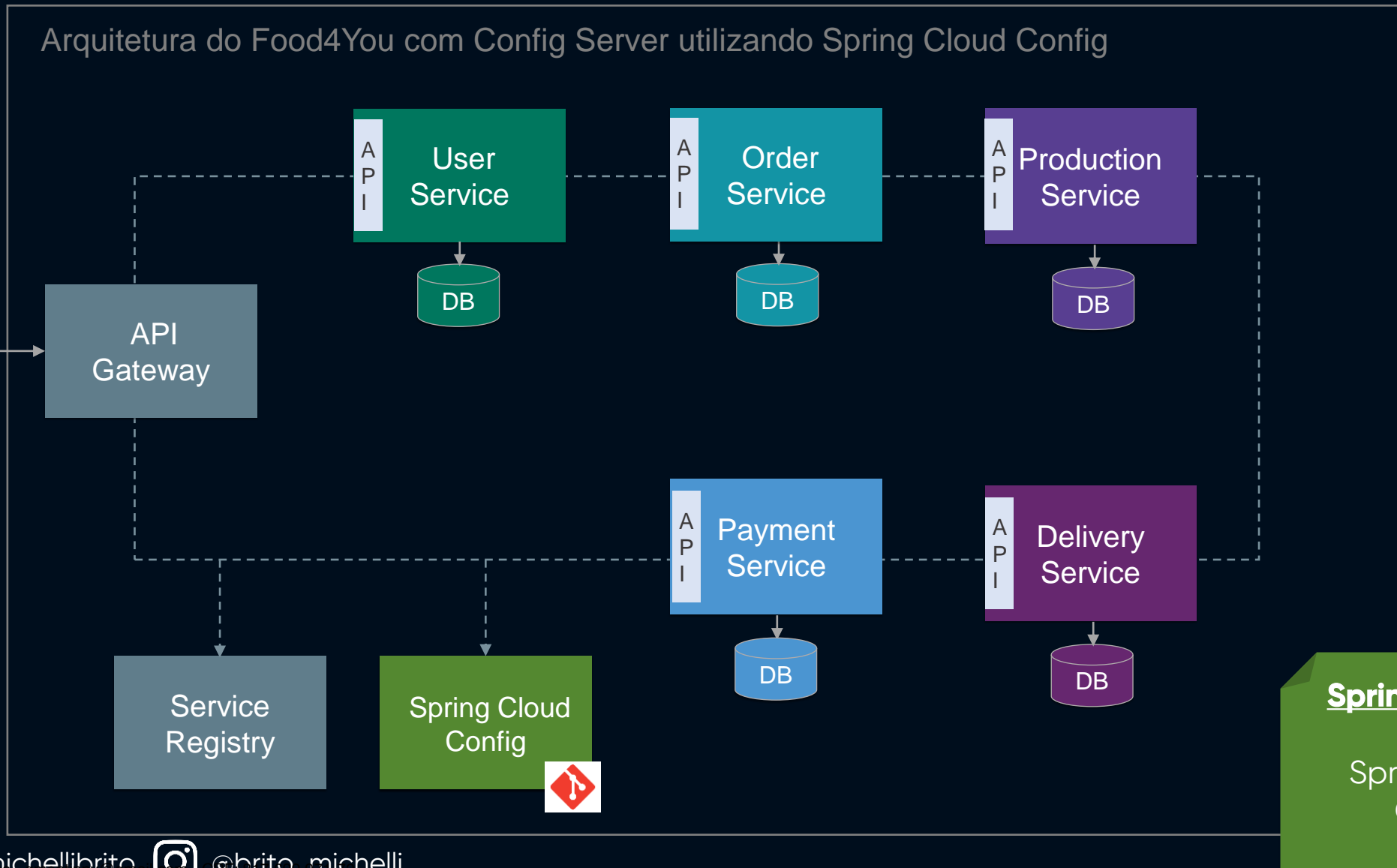


Modelando Config Server



Config Server também é um Microservice.

Config Server com Spring Cloud Config



COMPLEMENTANDO

Spring Cloud Config



Suporte para configuração externa em uma arquitetura de Microservices

Config Server para gerenciar configurações de múltiplos ambientes

Utiliza do Git para armazenamento e versionamento das propriedades

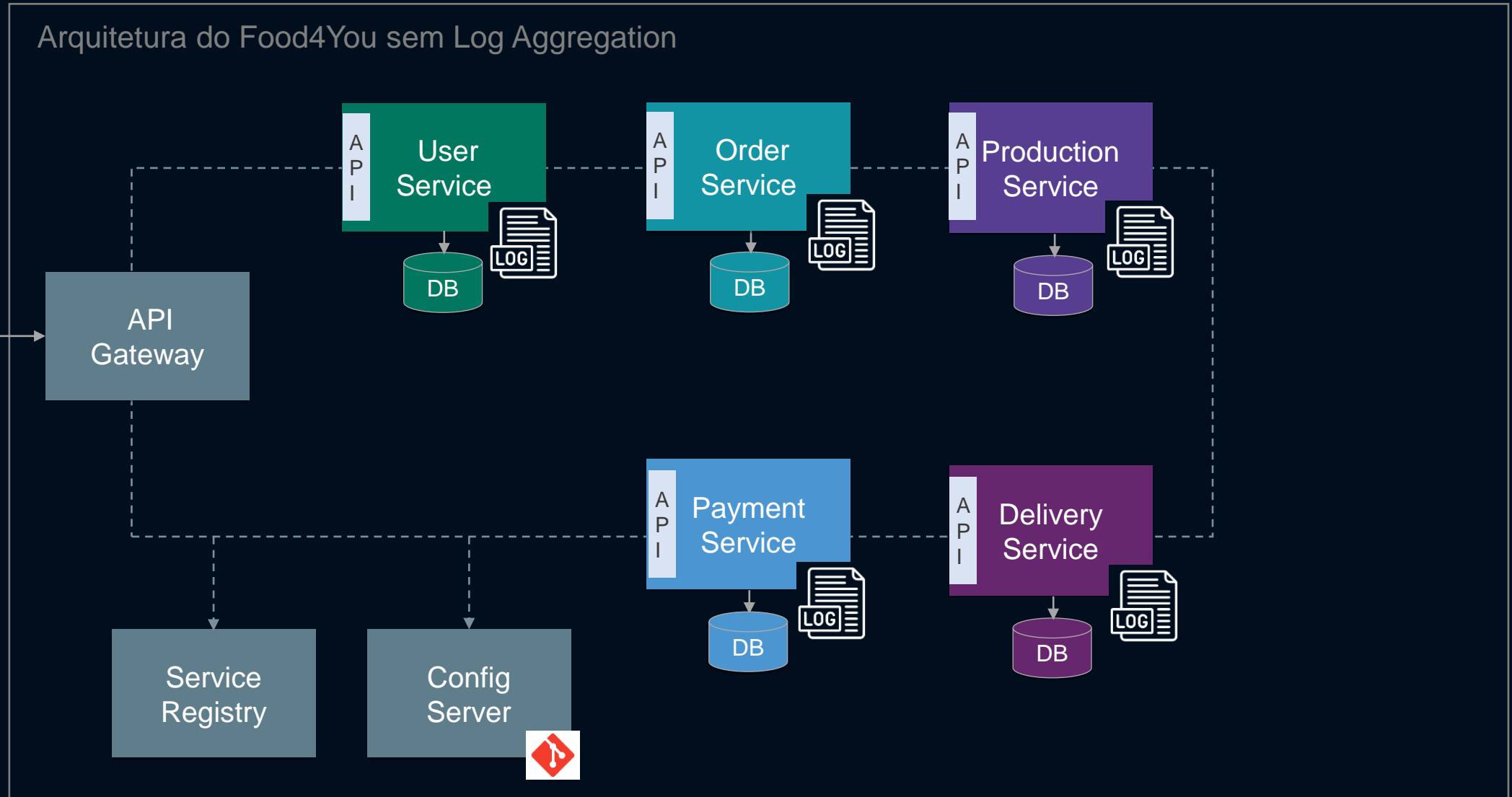
APIs para consultas de propriedades

Criptografia e Descriptografia de valores de propriedades

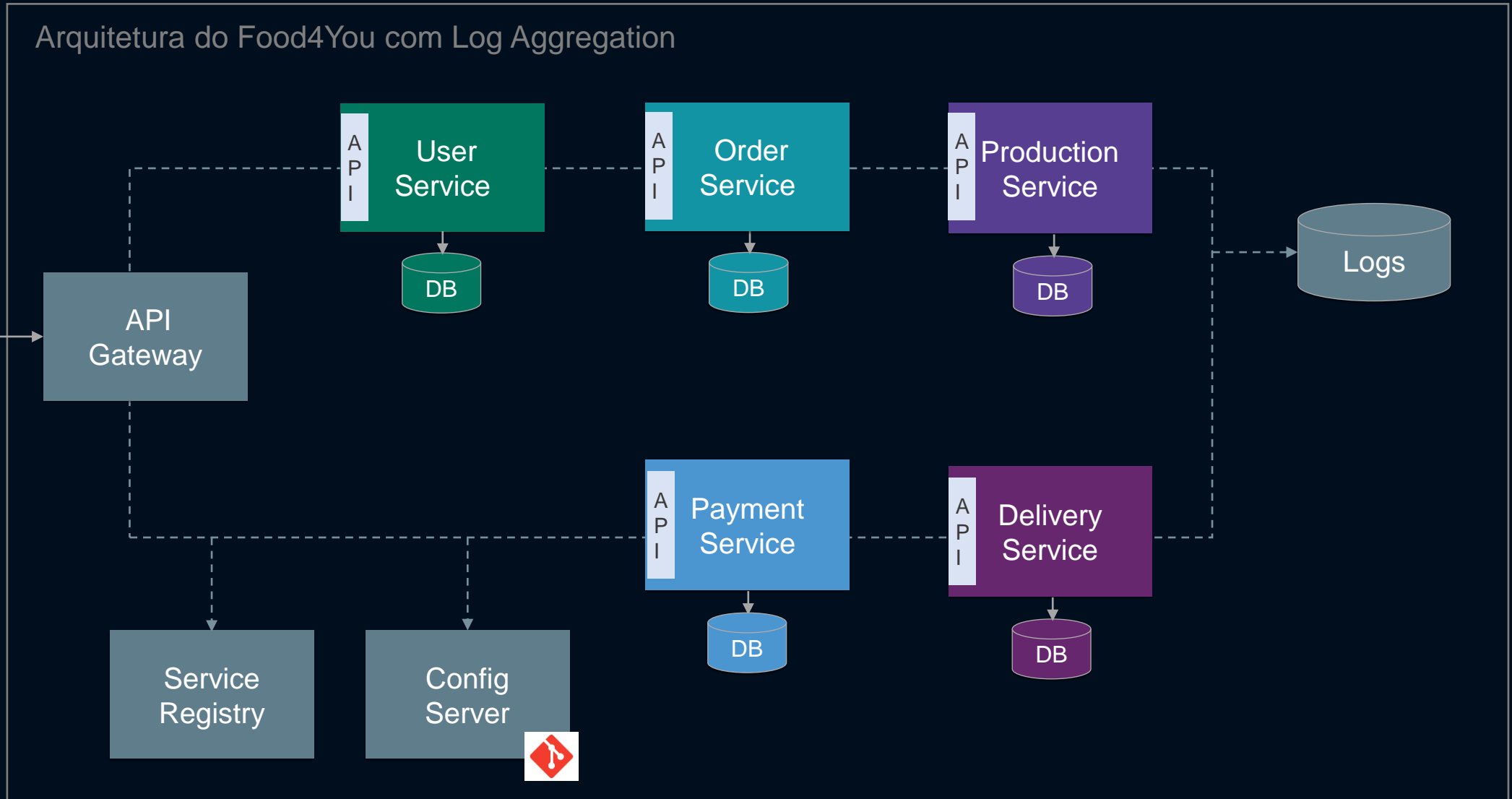
```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-config-server</artifactId>  
</dependency>
```

```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-starter-config</artifactId>  
</dependency>
```

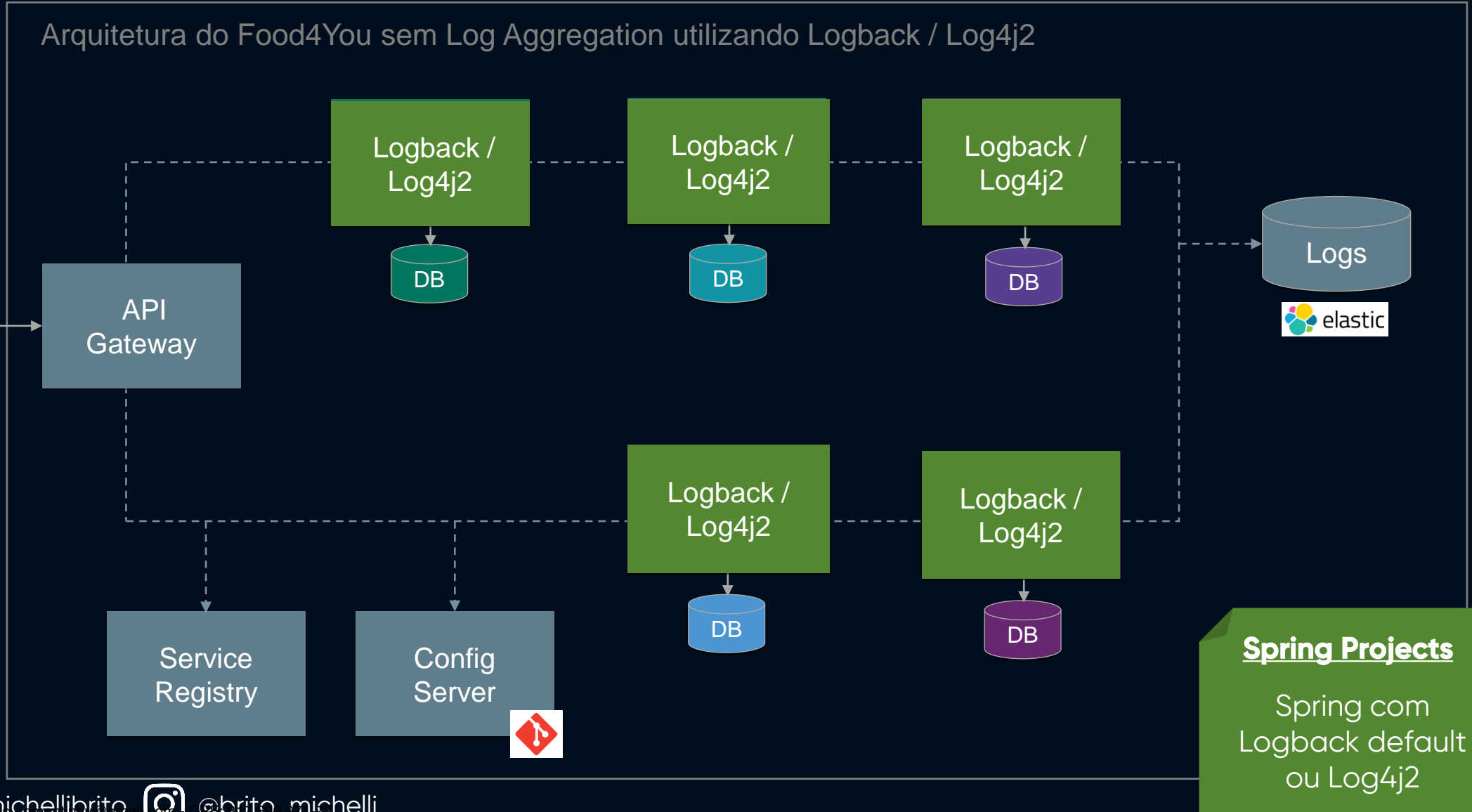
Modelando Observabilidade com Log Aggregation



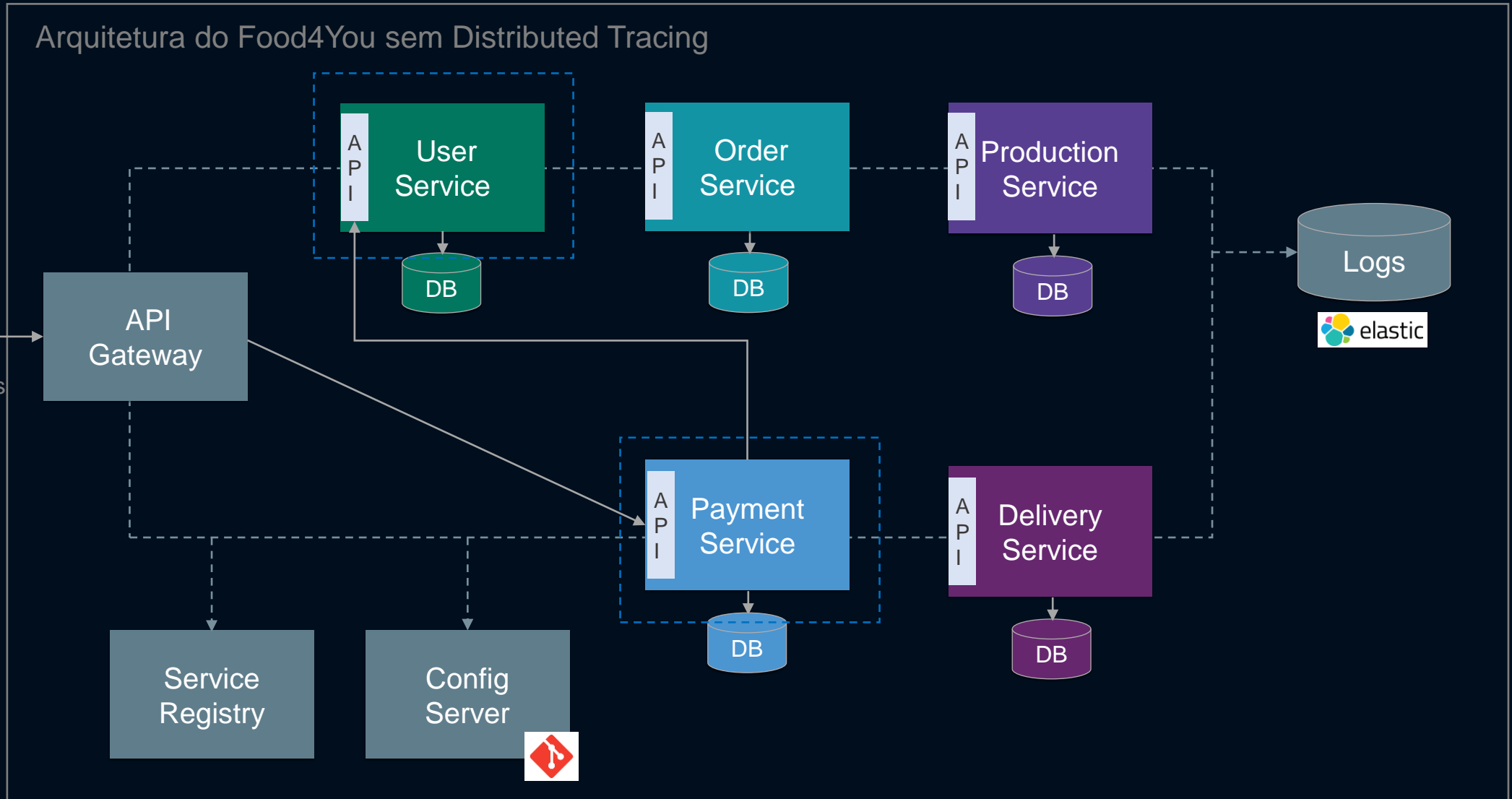
Modelando Observabilidade com Log Aggregation



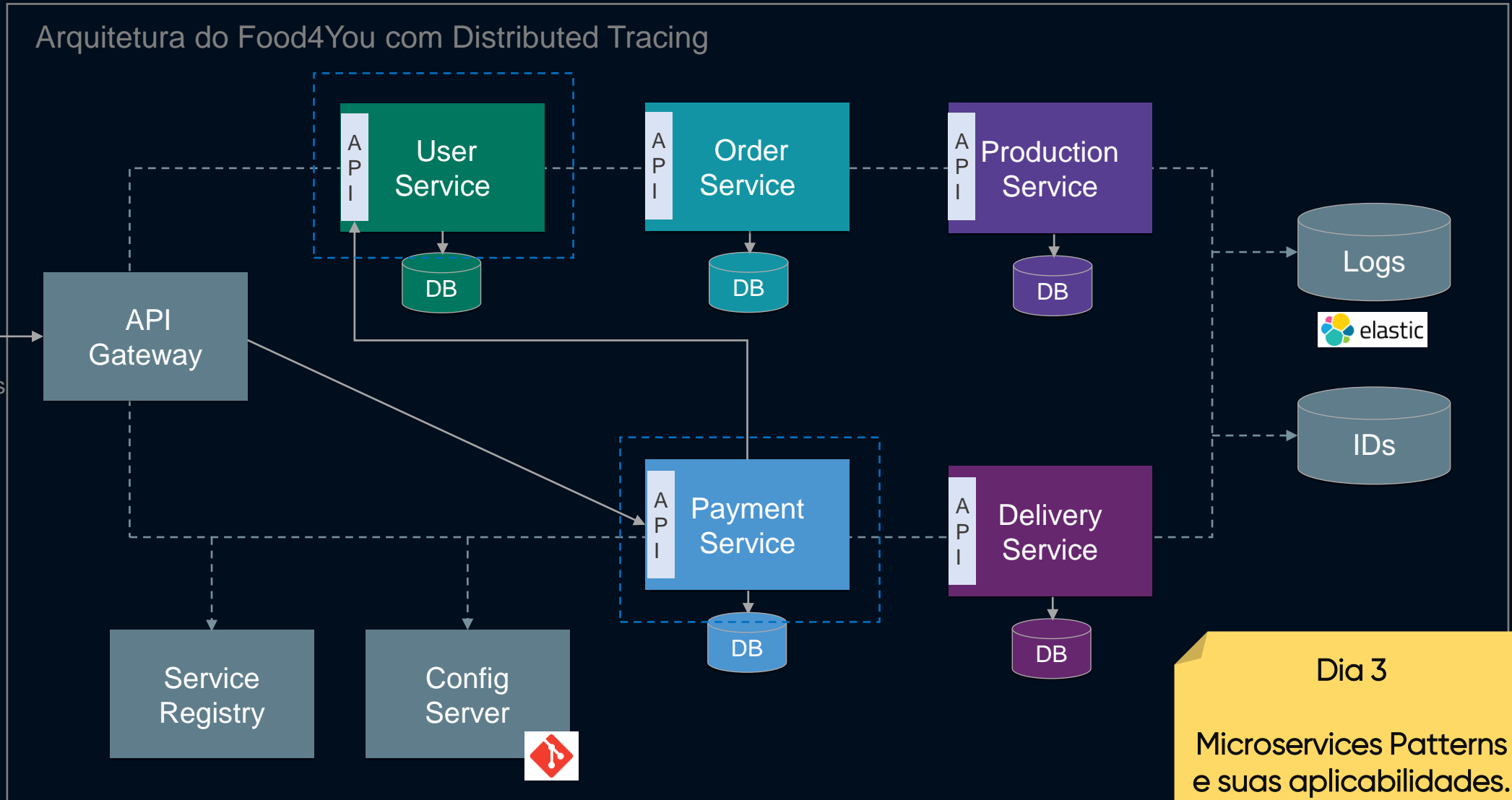
Log Aggregation com ELK e Logback/Log4j2



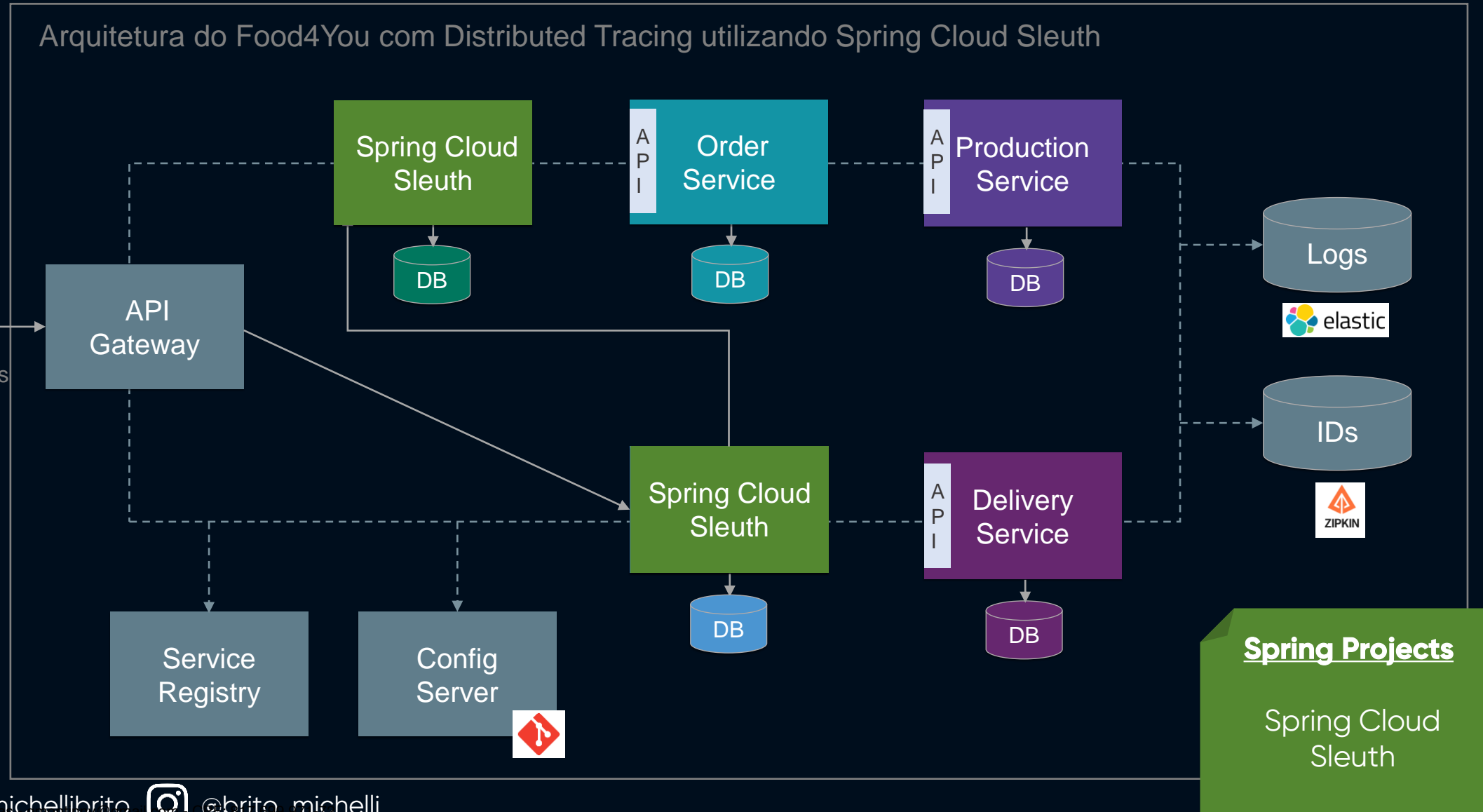
Modelando Observabilidade com Distributed Tracing



Modelando Observabilidade com Distributed Tracing



Distributed Tracing com Spring Cloud Sleuth



COMPLEMENTANDO

Spring Cloud Sleuth

Suporte para rastreamento distribuído

Geração de identificadores Trace ID
(único entre todas as chamadas)

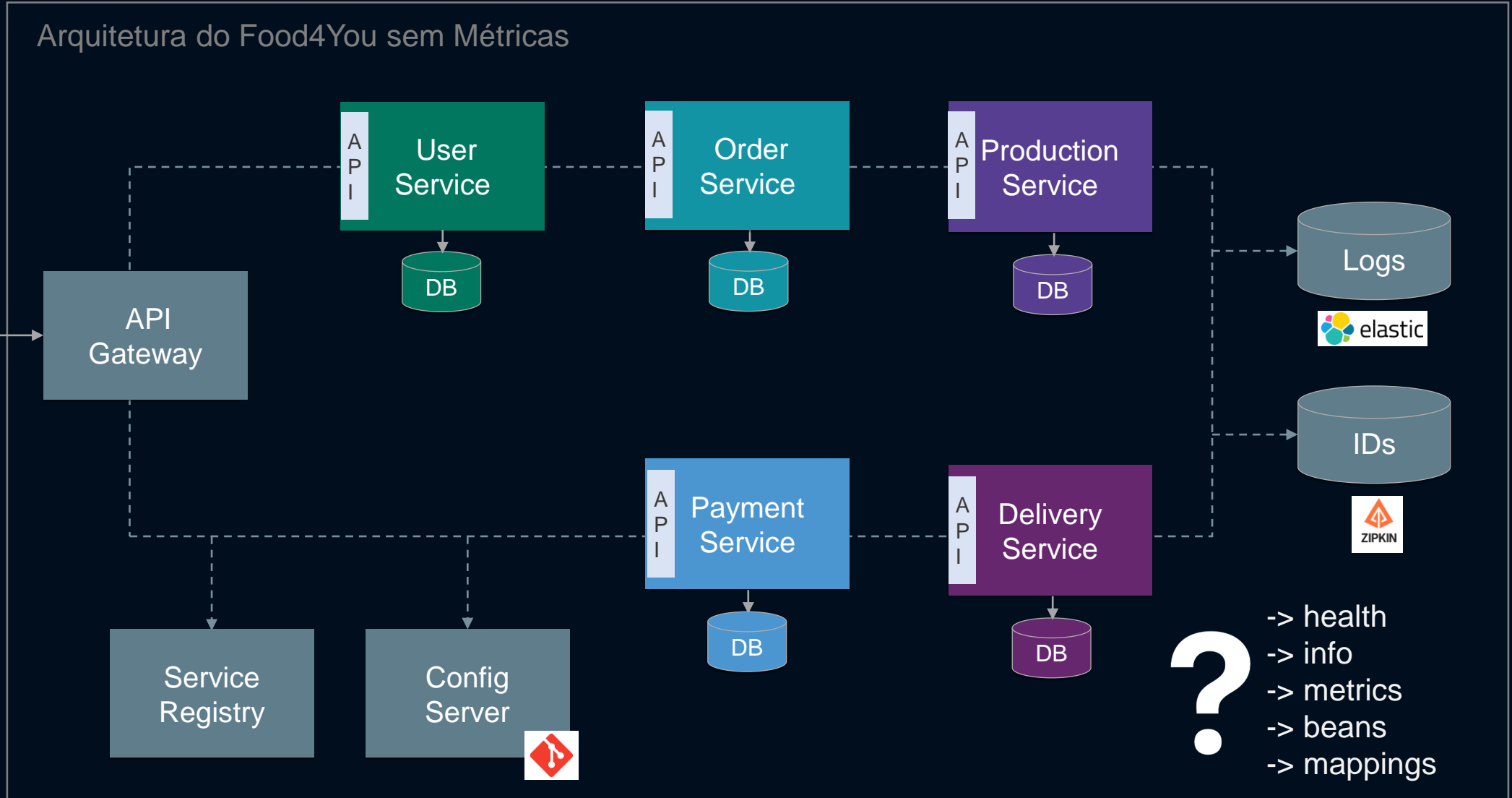
Geração de identificadores Span ID
(individuais para cada chamada)

Integração com Zipkin para análise e
interpretação dos ids de forma
centralizada

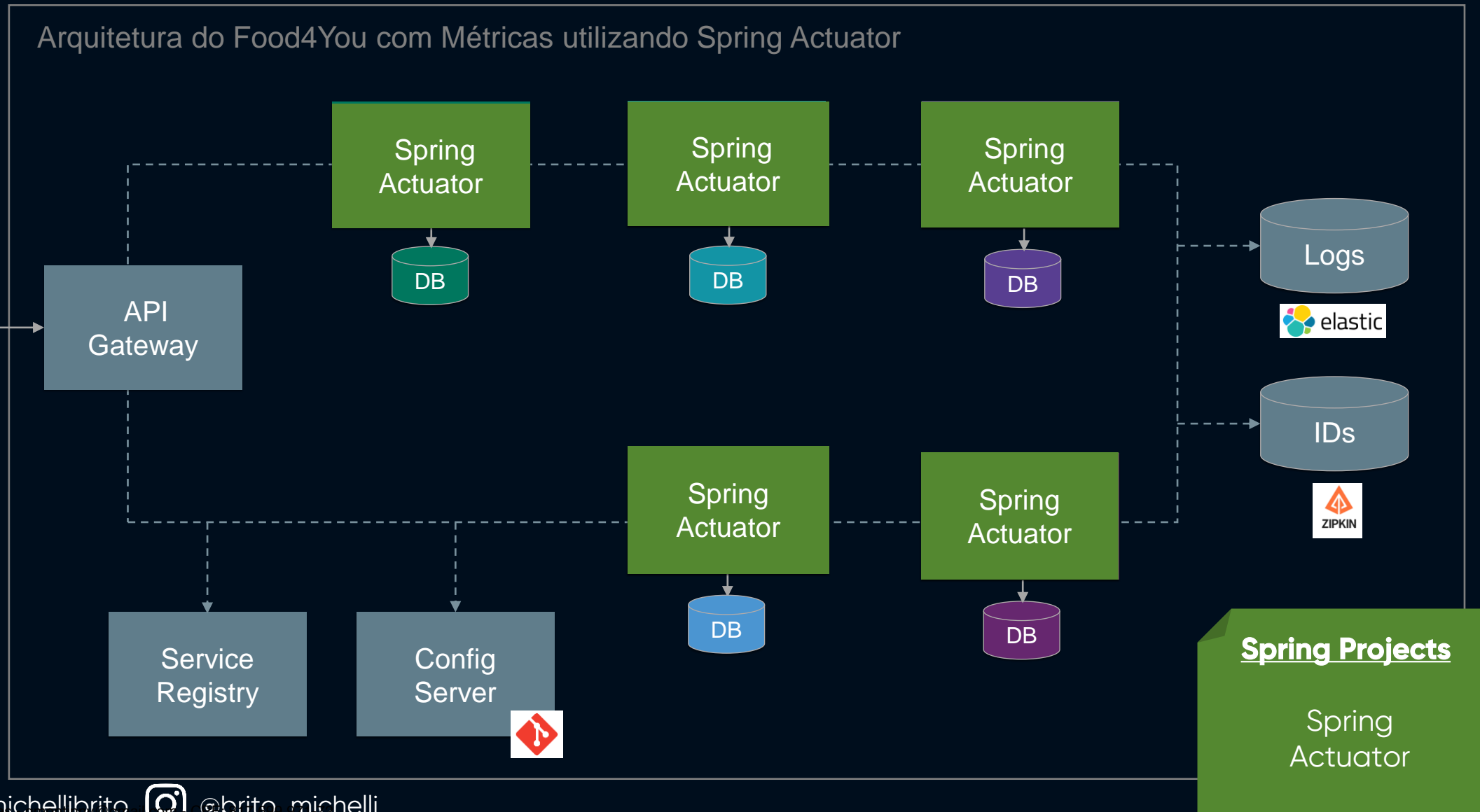
```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-starter-sleuth</artifactId>  
</dependency>
```

```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-sleuth-zipkin</artifactId>  
</dependency>
```

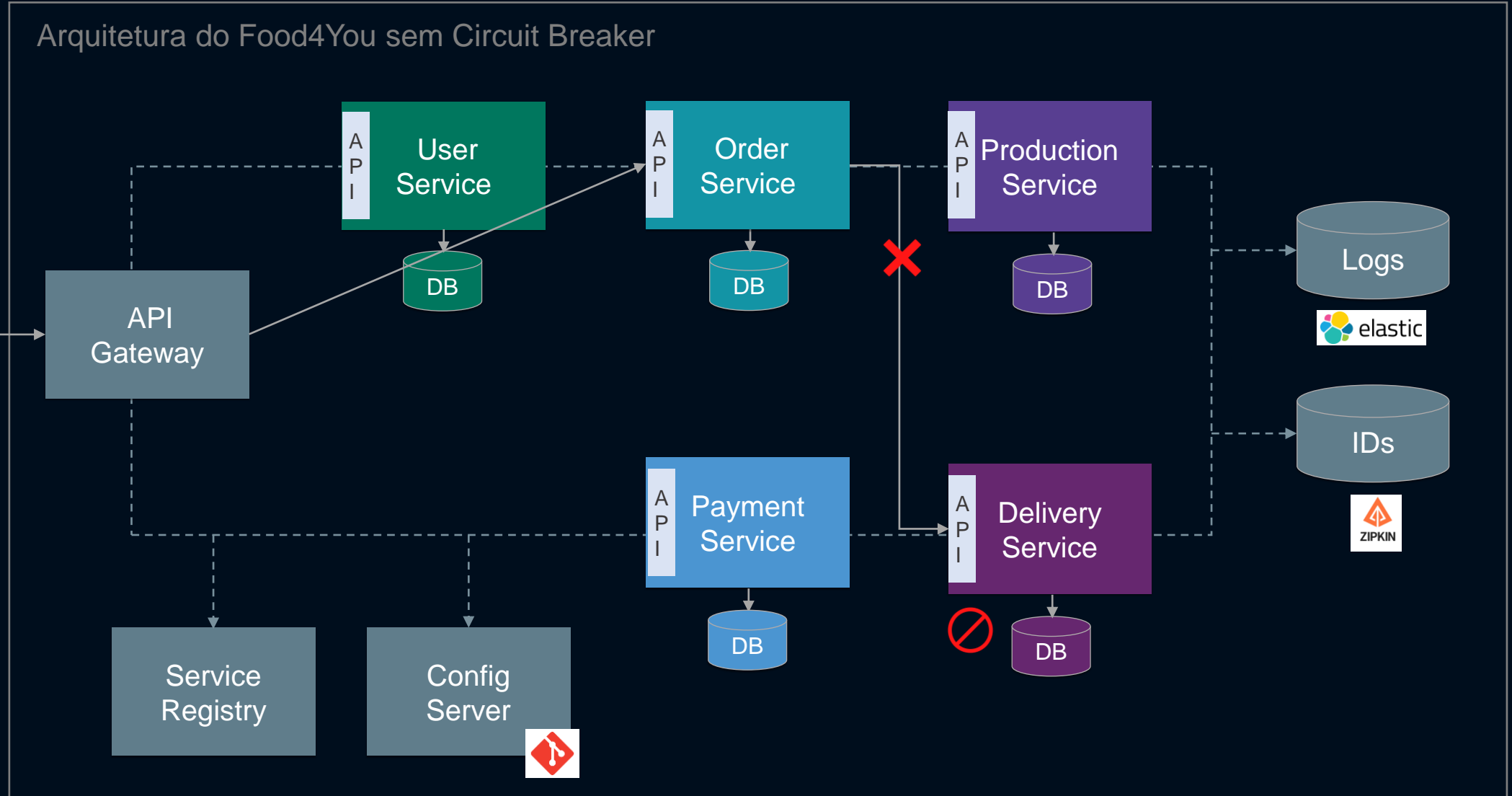
Modelando Observabilidade com Métricas



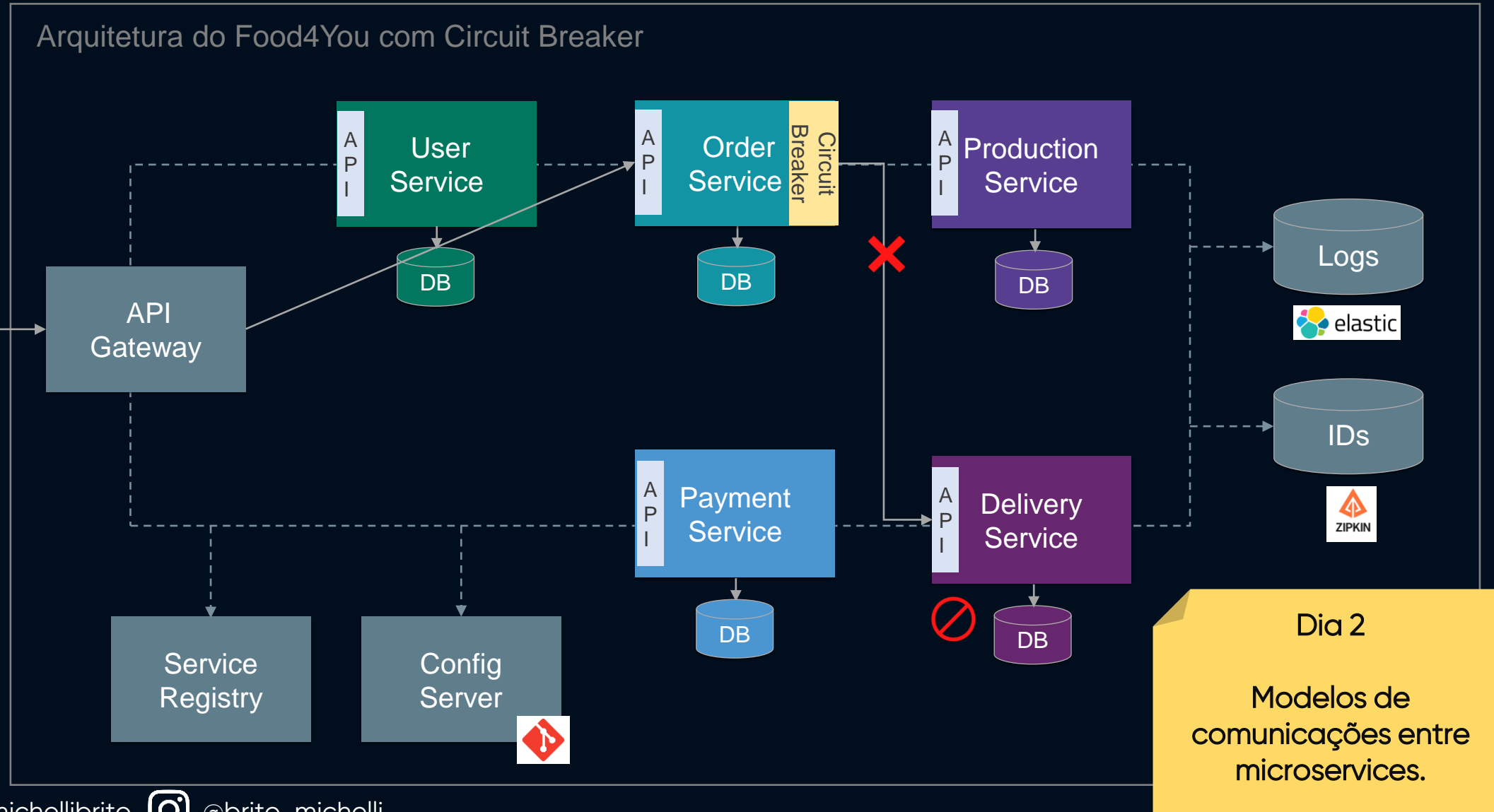
Modelando Métricas com Spring Actuator



Modelando Resiliência com Circuit Breaker

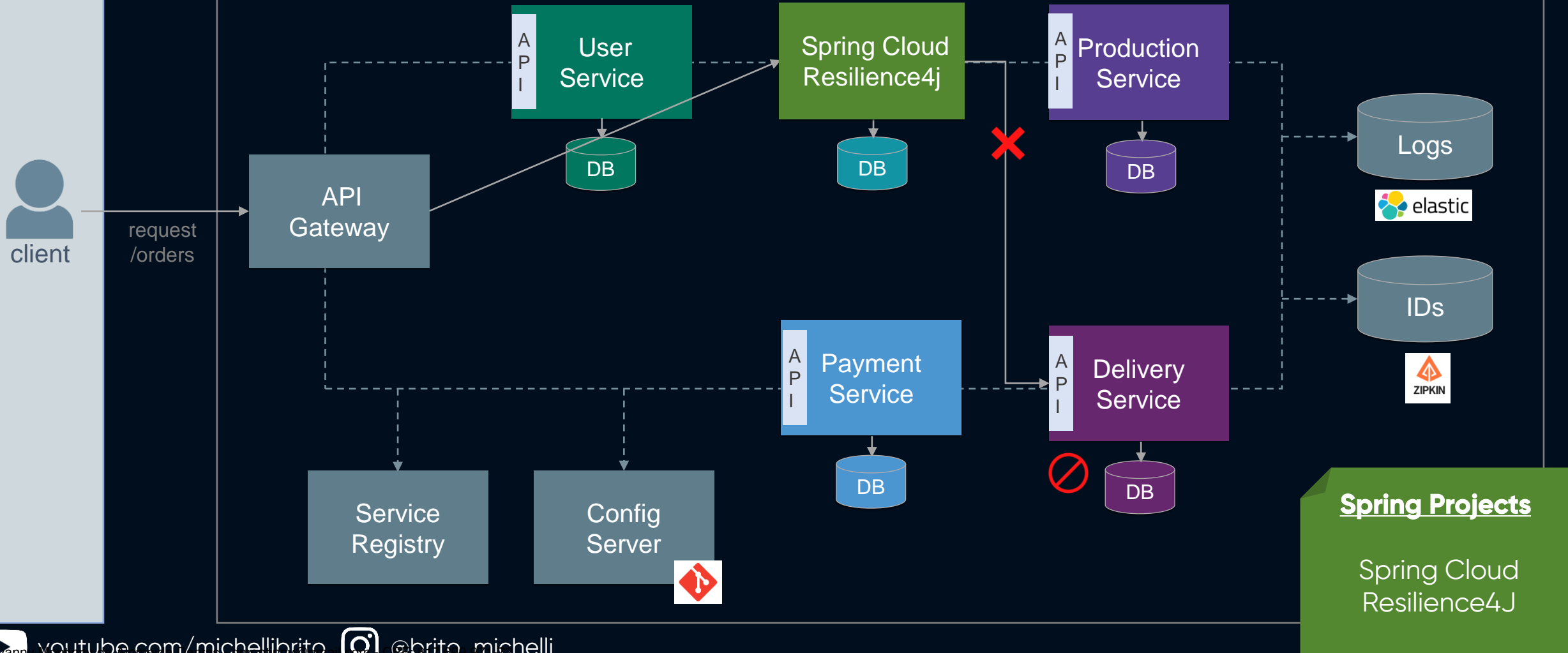


Modelando Resiliência com Circuit Breaker

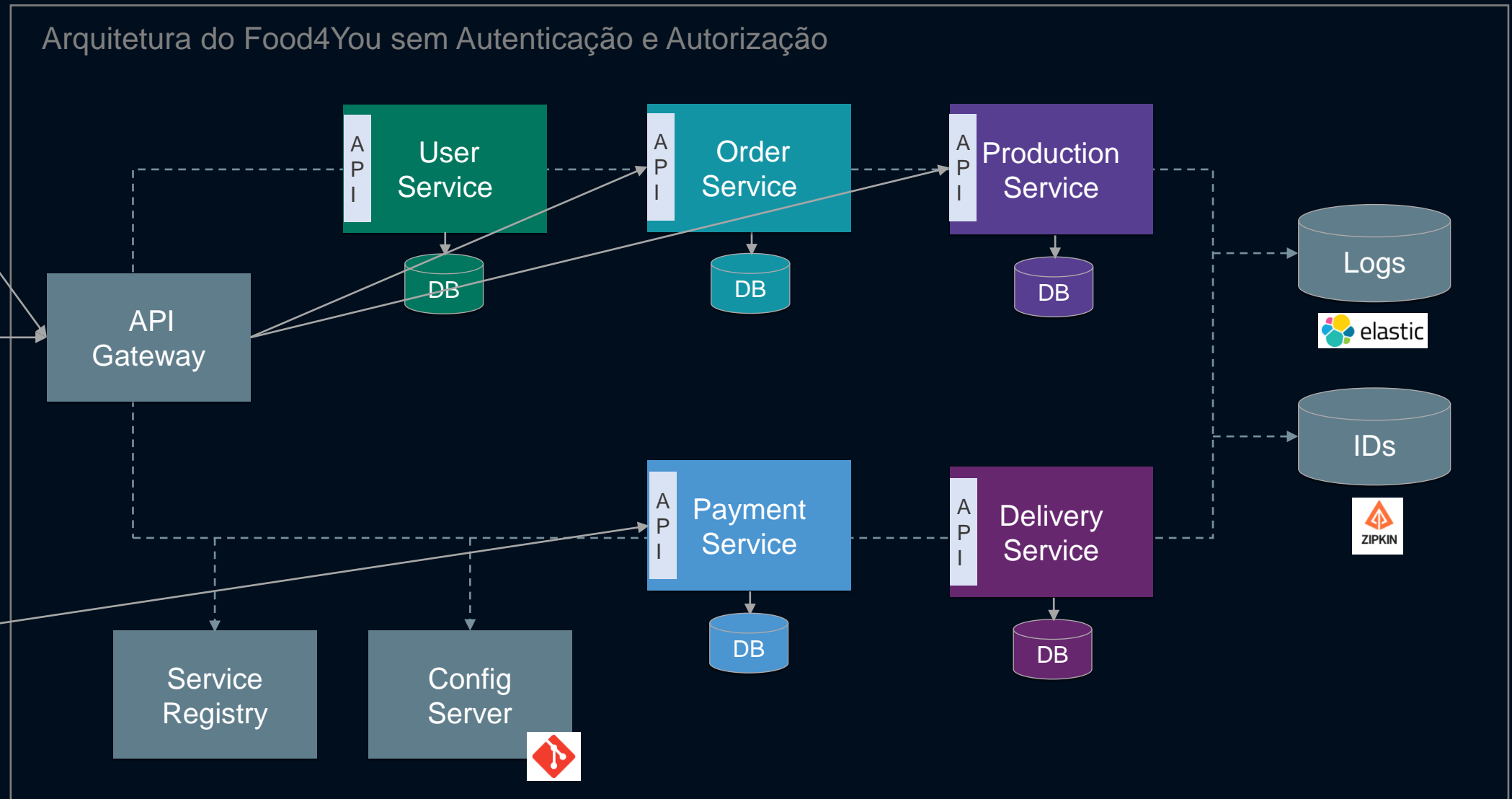


Circuit Breaker com Spring Cloud Resilience4J

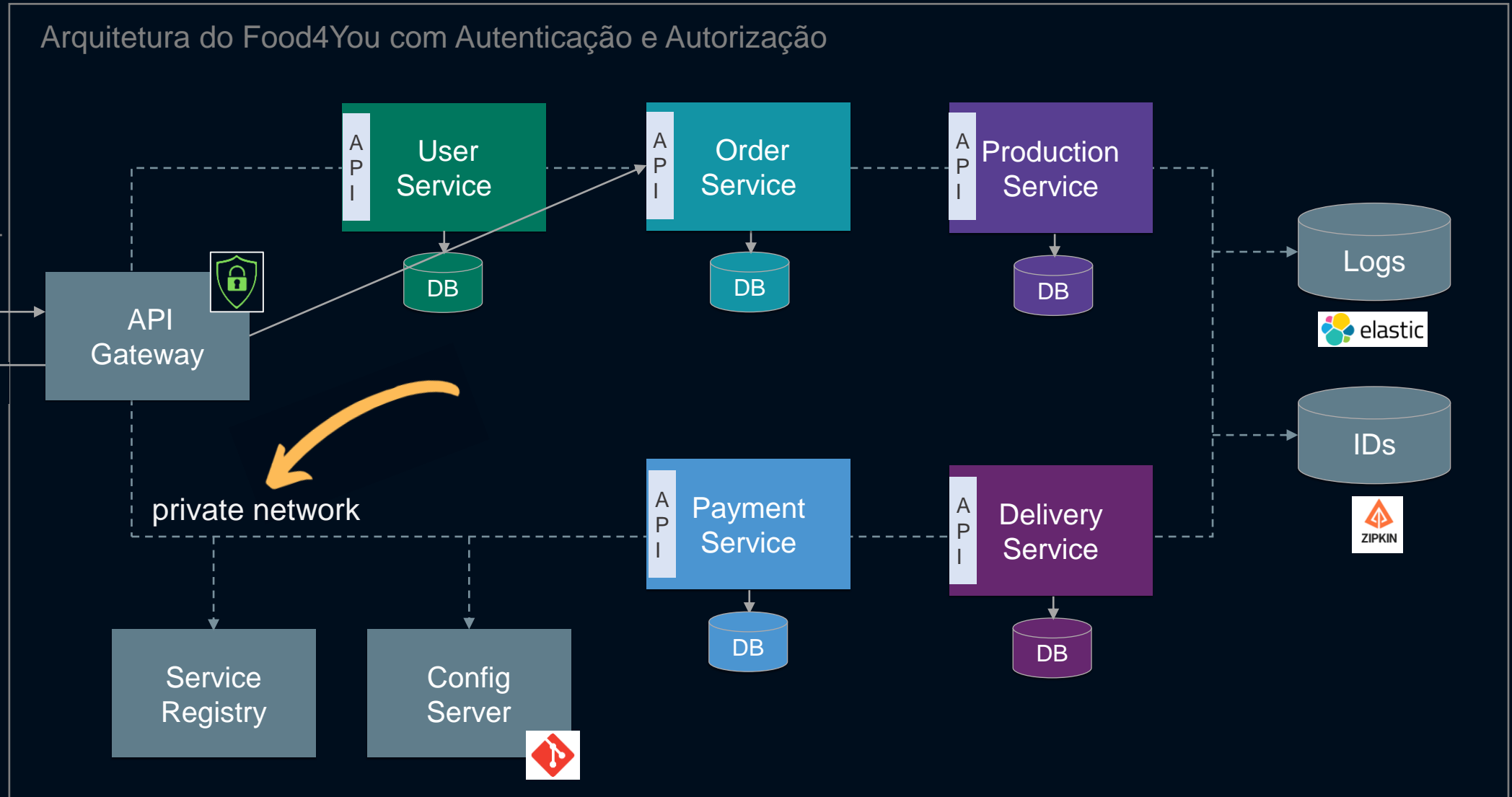
Arquitetura do Food4You com Circuit Breaker utilizando Spring Cloud Circuit Breaker Resilience4J



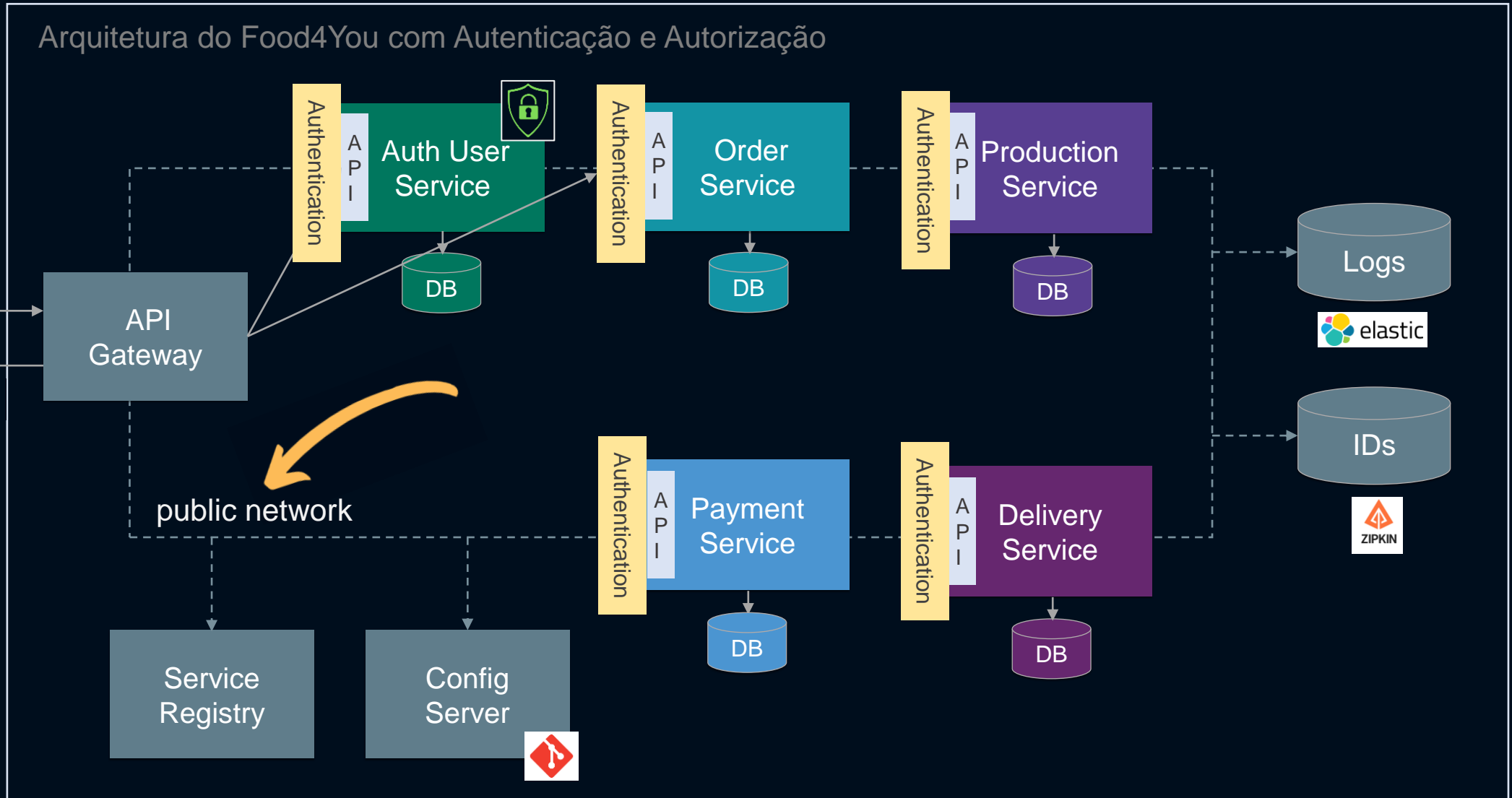
Modelando Autenticação e Autorização



Modelo 1: Autenticação e Autorização no Gateway



Modelo 2: Autenticação e Autorização Distribuída





COMPLEMENTANDO

Spring Security

Projeto que fornece autenticação e autorização para aplicações Java

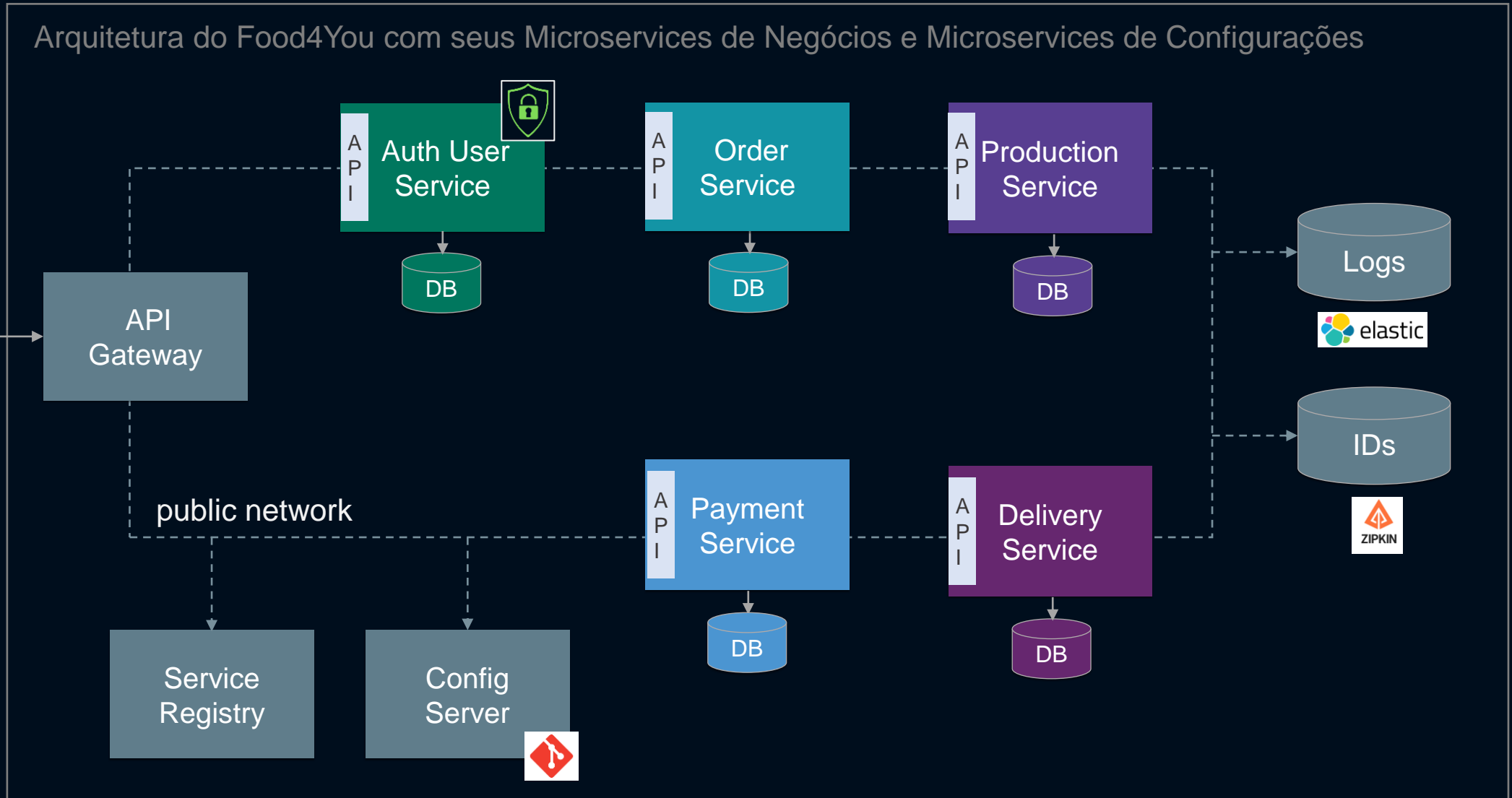
Suporte para implementação de Basic Authentication em memória e em database, JWT e OAuth

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-security</artifactId>  
</dependency>
```

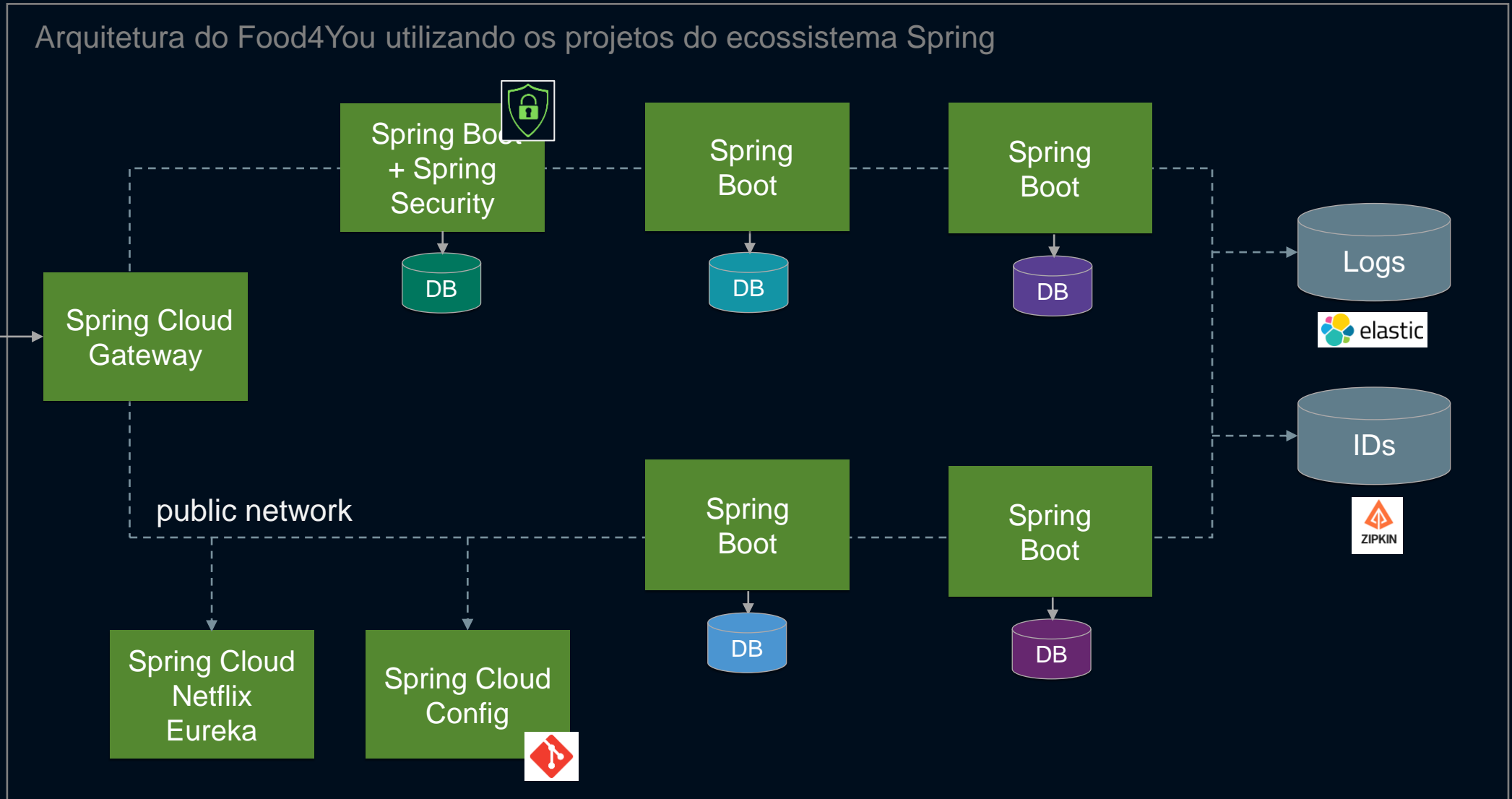
Arquitetura Completa



Arquitetura completa do **Food4You**



Arquitetura completa do **Food4You**



O entendimento do negócio juntamente com as boas práticas e padrões são essenciais para você fazer as melhores escolhas e definir as melhores soluções e assim ser um profissional mais qualificado no mercado.

A discussão de Microservices não deve ser sobre o tamanho ou complexidade do negócio, mas principalmente sobre projetar sistemas sustentáveis.

