

SEMANA  
**DECODER**

# MODELAGEM DO SISTEMA EAD

**DIA 1**

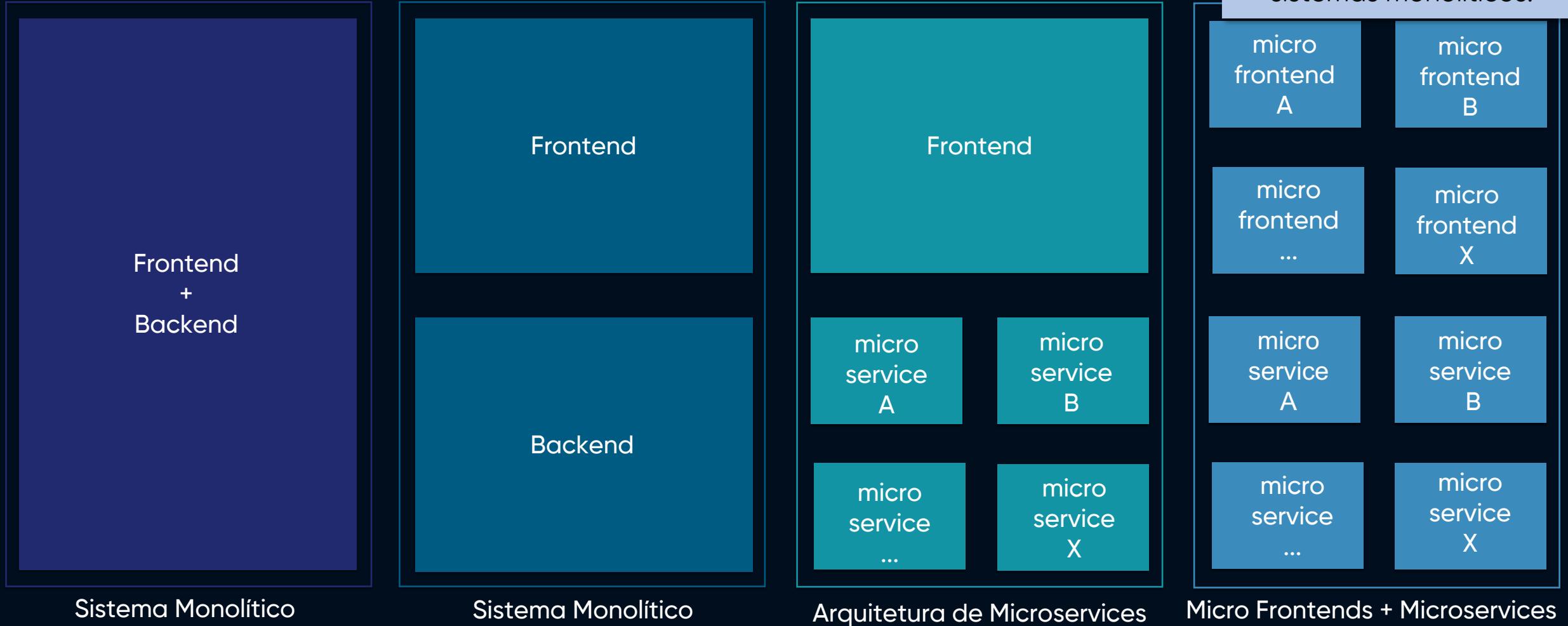


SEMANA  
**DECODER**  
com Michelli Brito

# BOAS PRÁTICAS E PADRÕES NA MODELAGEM DE UMA ARQUITETURA DE MICROSERVICES COM SPRING

Dia 01

# Evolução Arquitetural



evolução arquitetural

## COMPLEMENTANDO

# Principais Benefícios dos Microservices

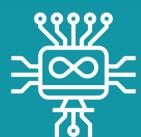
Manutenibilidade



Alta Disponibilidade



Flexibilidade  
Tecnológica



Independência  
das Equipes



Melhor  
Performance



Divisão da Complexidade do Negócio



Isolamento a Falhas



Alta Escalabilidade



Baixo Acoplamento



Melhor  
Testabilidade



Agilidade nas  
Mudanças



Isolamento  
Modelagem Dados



Aumento  
Resiliência



# Arquitetura de Microservices é um modelo flexível e sustentável...

**Modelo que mais facilmente é capaz de acompanhar as frenéticas mudanças dos negócios e as modernizações tecnológicas.**

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

Modelagem na Prática

## SISTEMA EAD

Educação a Distância



User



Course



Notification



Payment

# Modelando os Microservices de Negócios

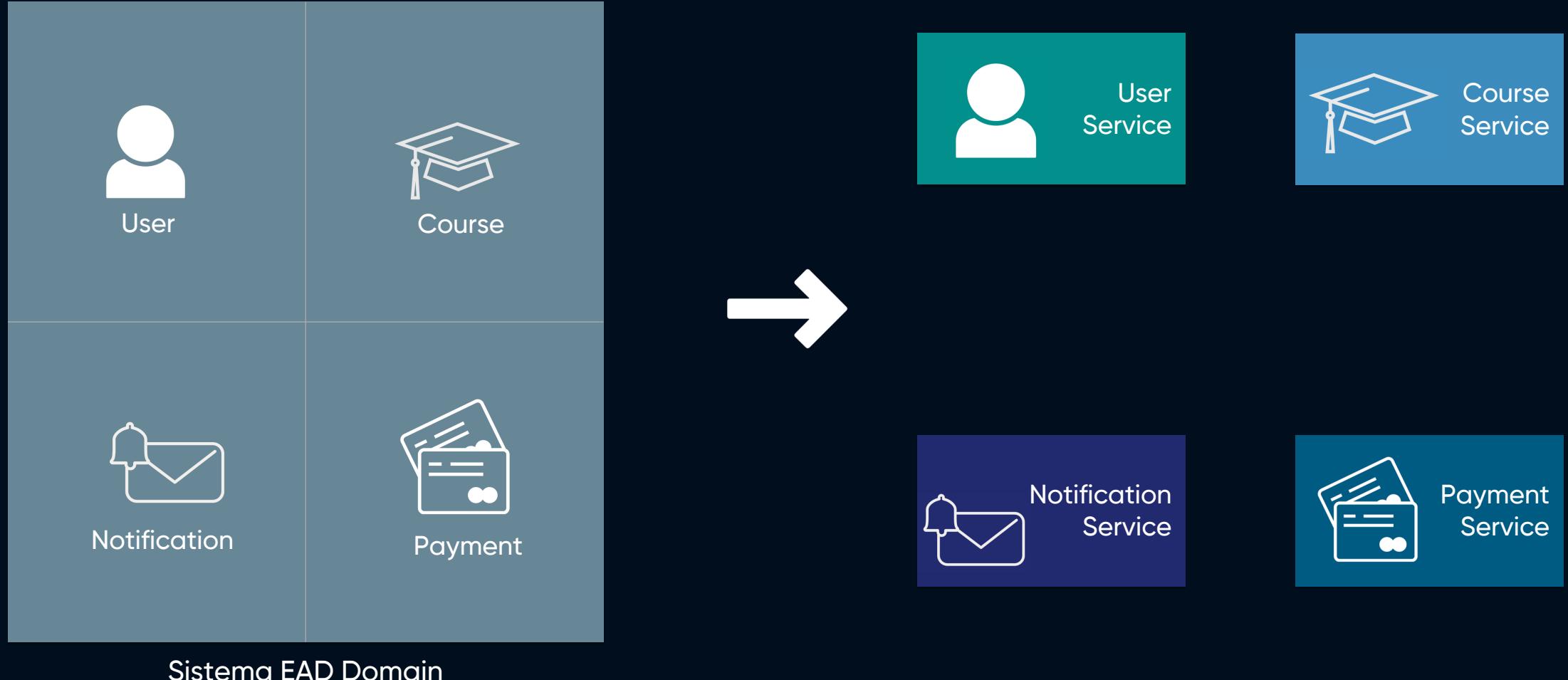
User  
Service

Course  
Service

Payment  
Service

Notification  
Service

# Divisão do Negócio em Microservices



# Granularidade dos Microservices

User Service

Notification Service

Payment Service

Course Service

SMTP Gmail Service

PagSeguro Service

SMTP AWS Service

Paypal Service

SMS Service

É essencial o conhecimento profundo do negócio.

# Arquitetura de Microservices do **SISTEMA EAD**

Arquitetura EAD com seus Microservices de Negócios

User  
Service

Course  
Service

Payment  
Service

Notification  
Service

# Microservices com Spring Boot

Arquitetura EAD utilizando Spring Boot

Spring Boot

Spring Boot

Spring Boot

Spring Boot

Spring Projects

Spring Boot



Spring  
Framework



Servidor embutido  
(Tomcat ou Netty)

XML <bean>  
ou @Configuration

---

# Spring Boot

# Maturidade do ecossistema Spring

Microservices



Cloud



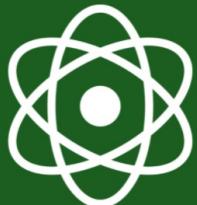
Event Driven



Batch



Reactive



Database



# Spring Projects

Spring Boot



Spring Security



Spring Cloud



Spring Data



Spring Web



Spring AMQP



Spring Hateoas



Spring Batch

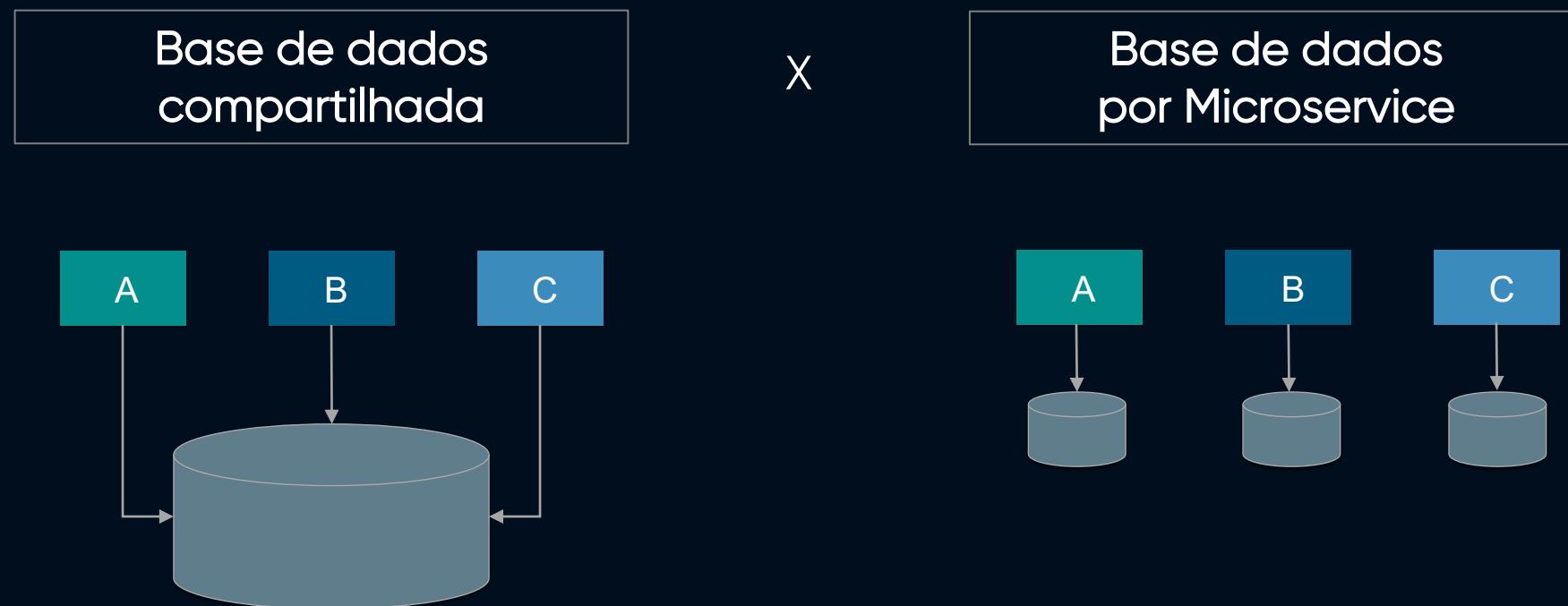


Spring Framework

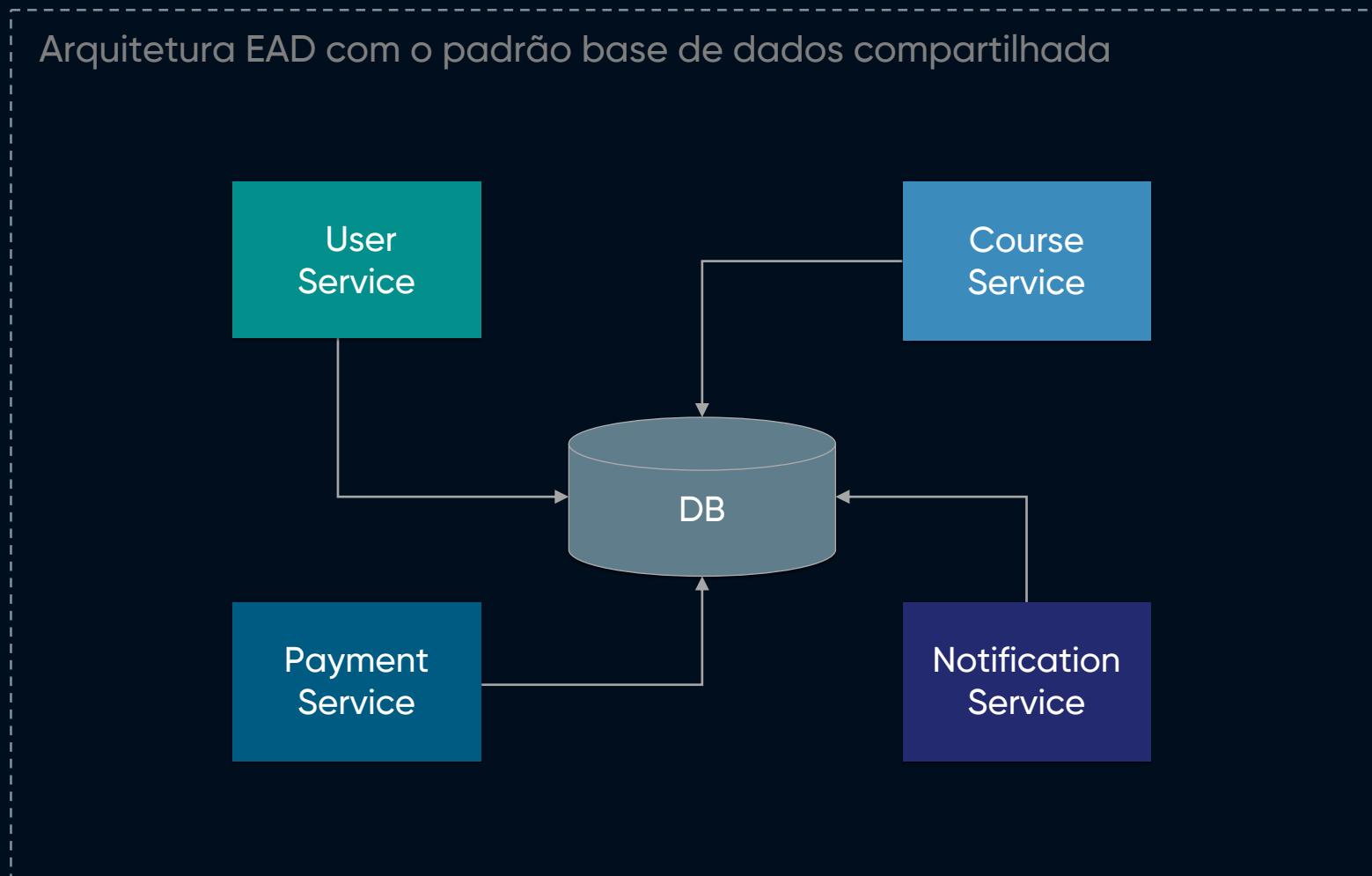


entre outros...

# Modelando Base de Dados em Microservices



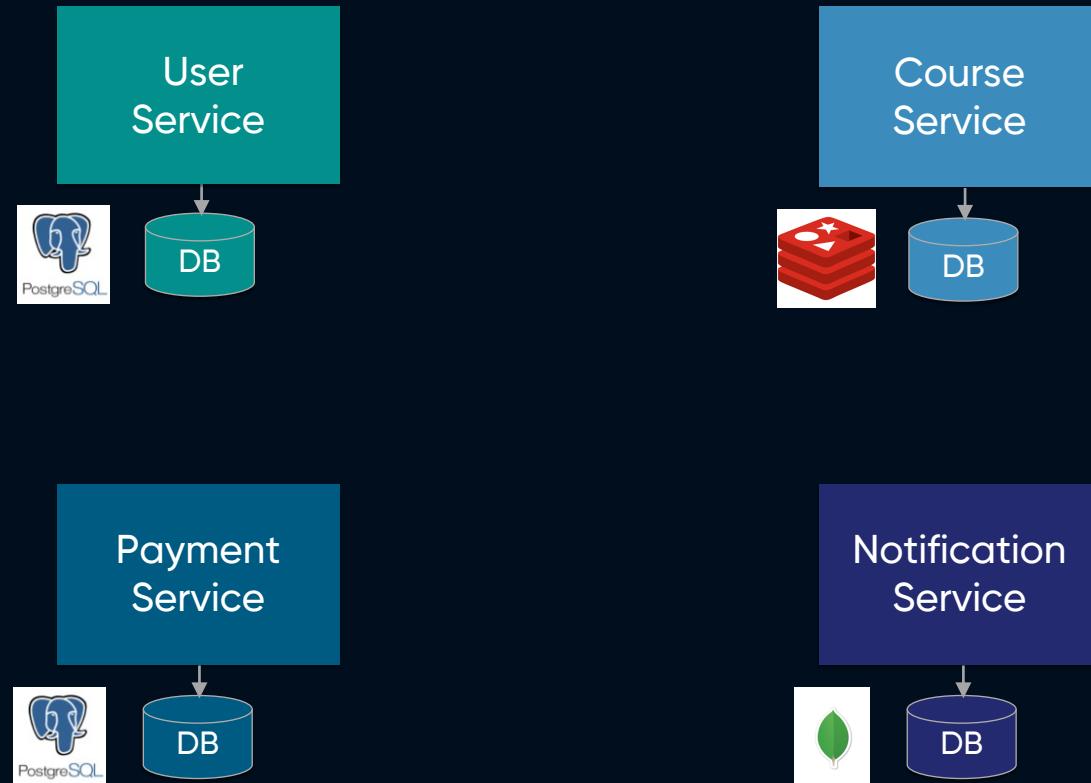
# Base de Dados Compartilhada



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

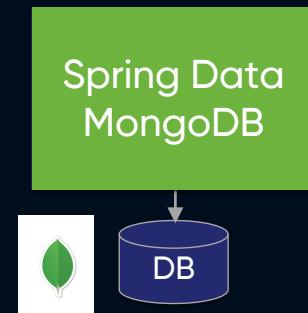
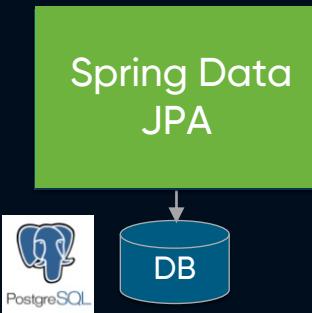
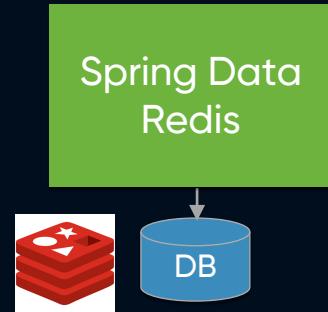
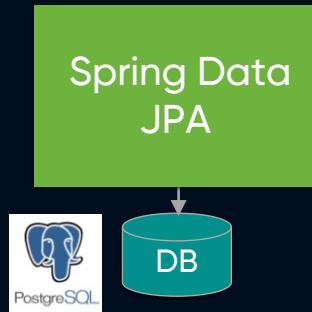
# Base de Dados por Microservices

Arquitetura EAD com o padrão base de dados por microservices



# Base de Dados por Microservices com Spring Data

Arquitetura EAD utilizando Spring Data



Spring Projects

Spring Data

# Spring Data



Spring Data JPA

Spring Data Redis

Spring Data MongoDB

Spring Data Cassandra

Spring Data JDBC

Spring Data Elasticsearch

# Quando temos bases 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 3

Gestão de dados  
distribuídos, teorema  
CAP...

DESCOMPLICANDO

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

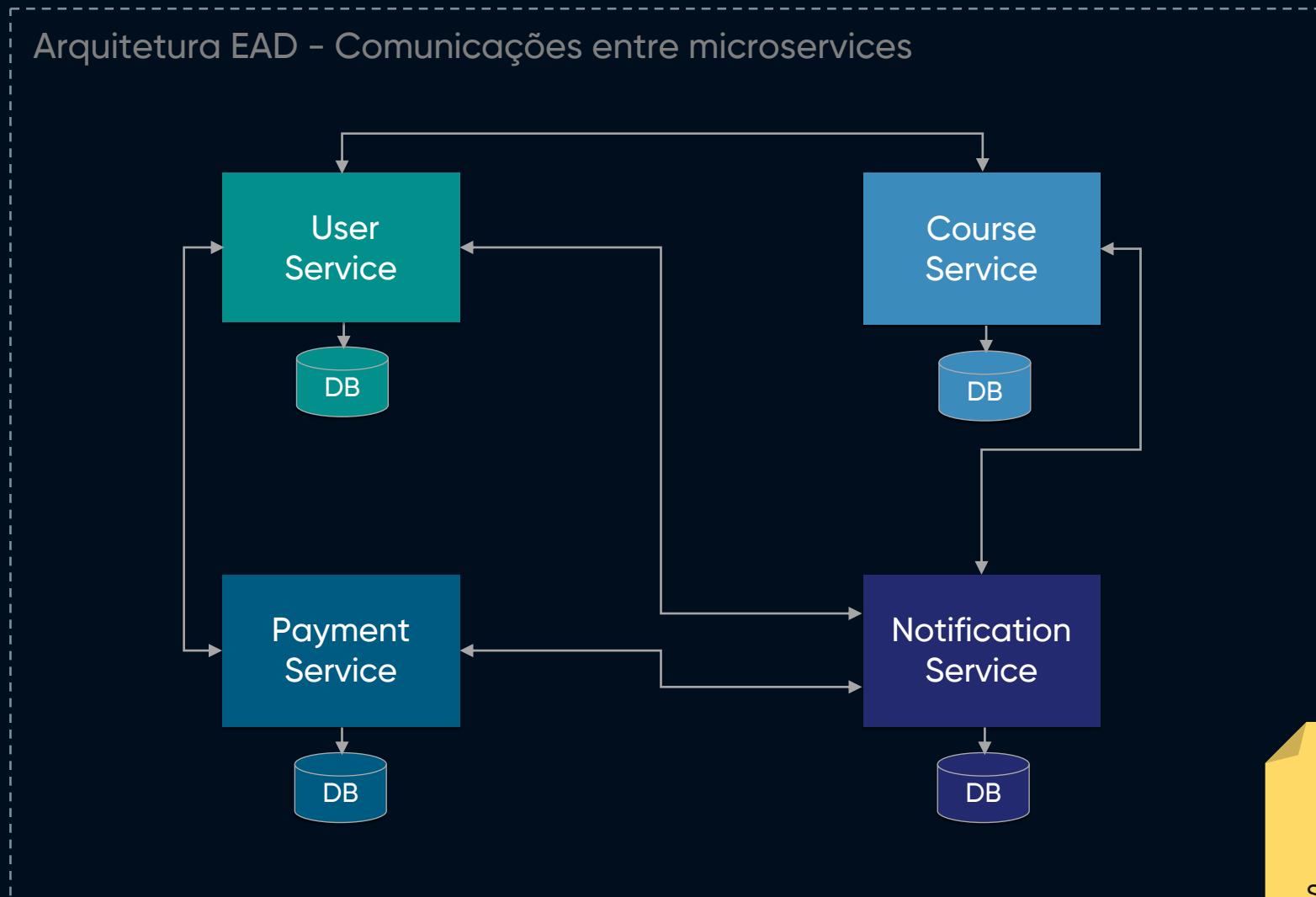
Base de Dados por  
Microservices



Sincronia e Replicação  
dos Dados Distribuídos



Comunicação entre  
Microservices



Dia 3  
Padrões de  
comunicações,  
sincronia e replicação  
de dados.

# Não existe desacoplamento absoluto entre Microservices...

O objetivo é sempre buscar o menor acoplamento possível...

mas em alguns casos, pode ser que o alto acoplamento em microservices seja consequência da própria regra de negócio em questão.

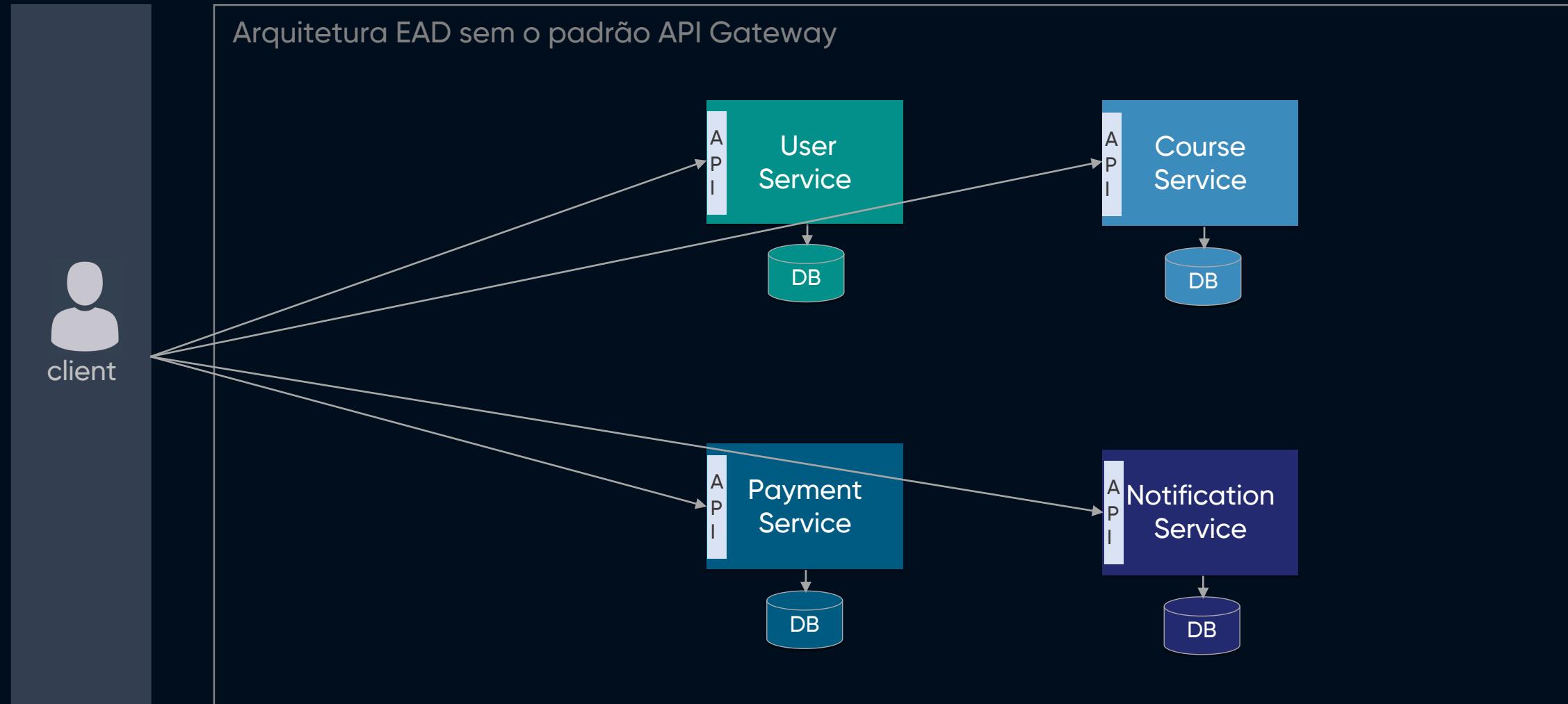
# Modelando os Microservices de Configurações

API  
Gateway

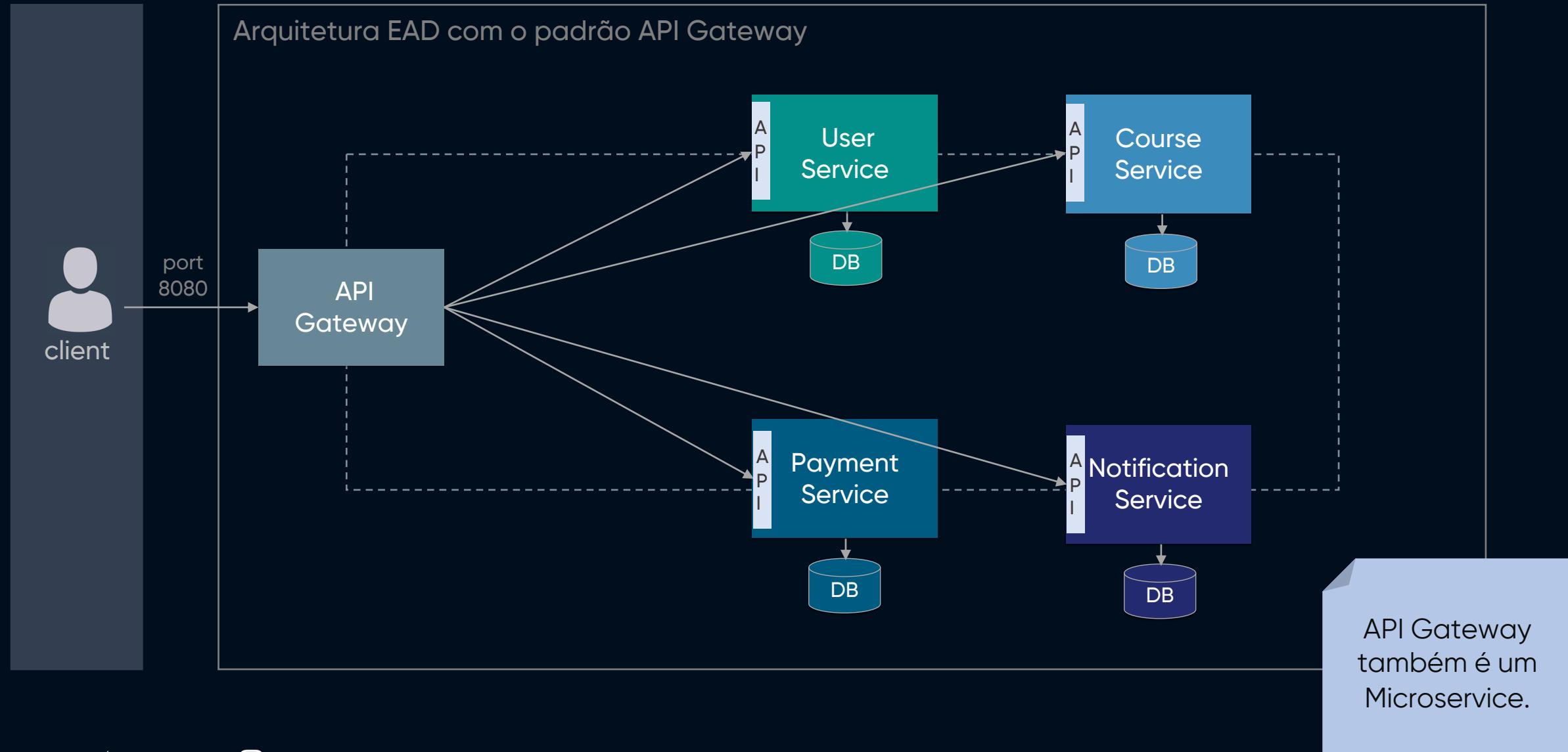
Service  
Registry

Config  
Server

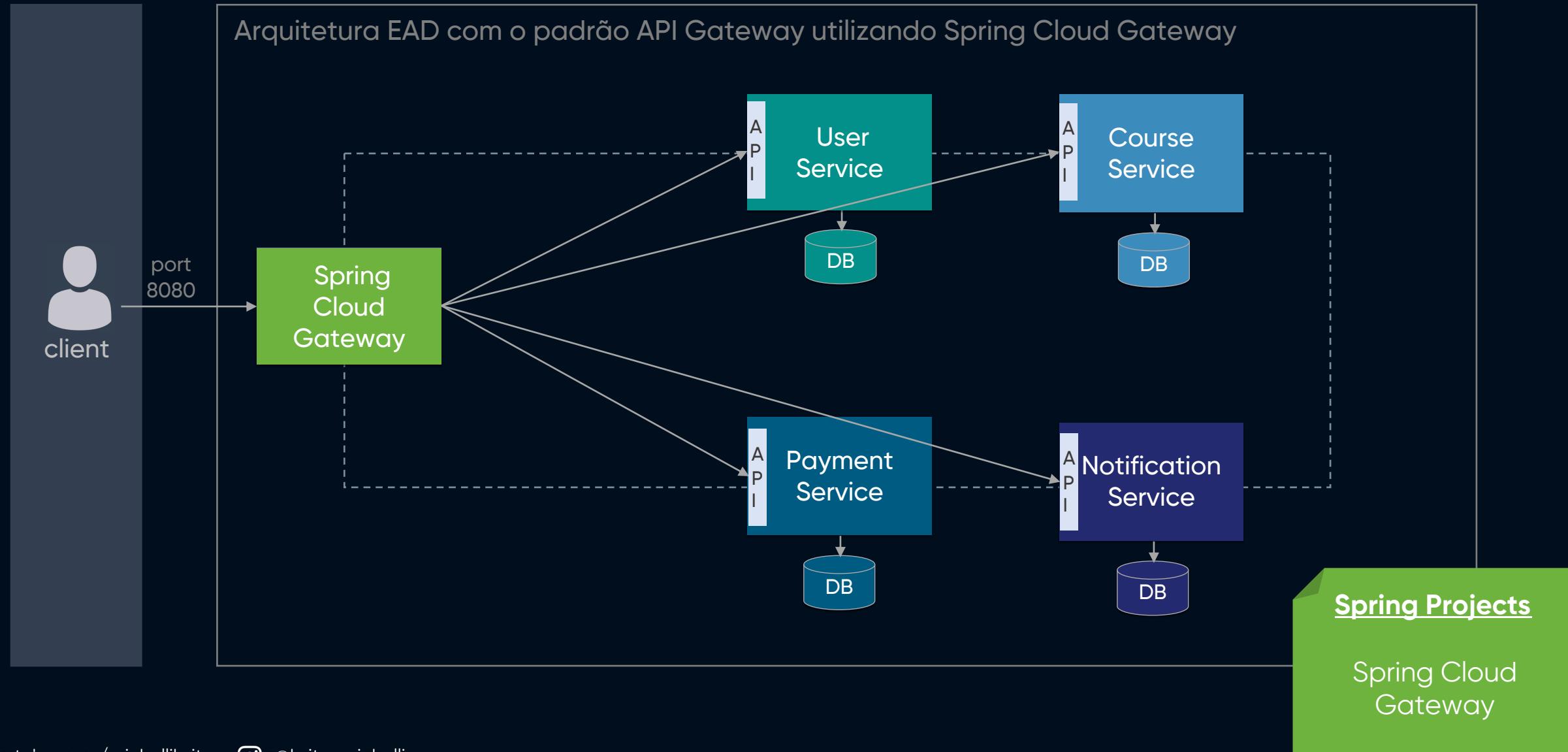
# Modelando o padrão API Gateway



# Modelando o padrão API Gateway



# Padrão API Gateway com Spring Cloud Gateway





# Spring Cloud Gateway

Inserido na versão 5 do Spring Framework

Roteamento para APIs,  
definição de filtros e predicados

Baseado no projeto Reactor

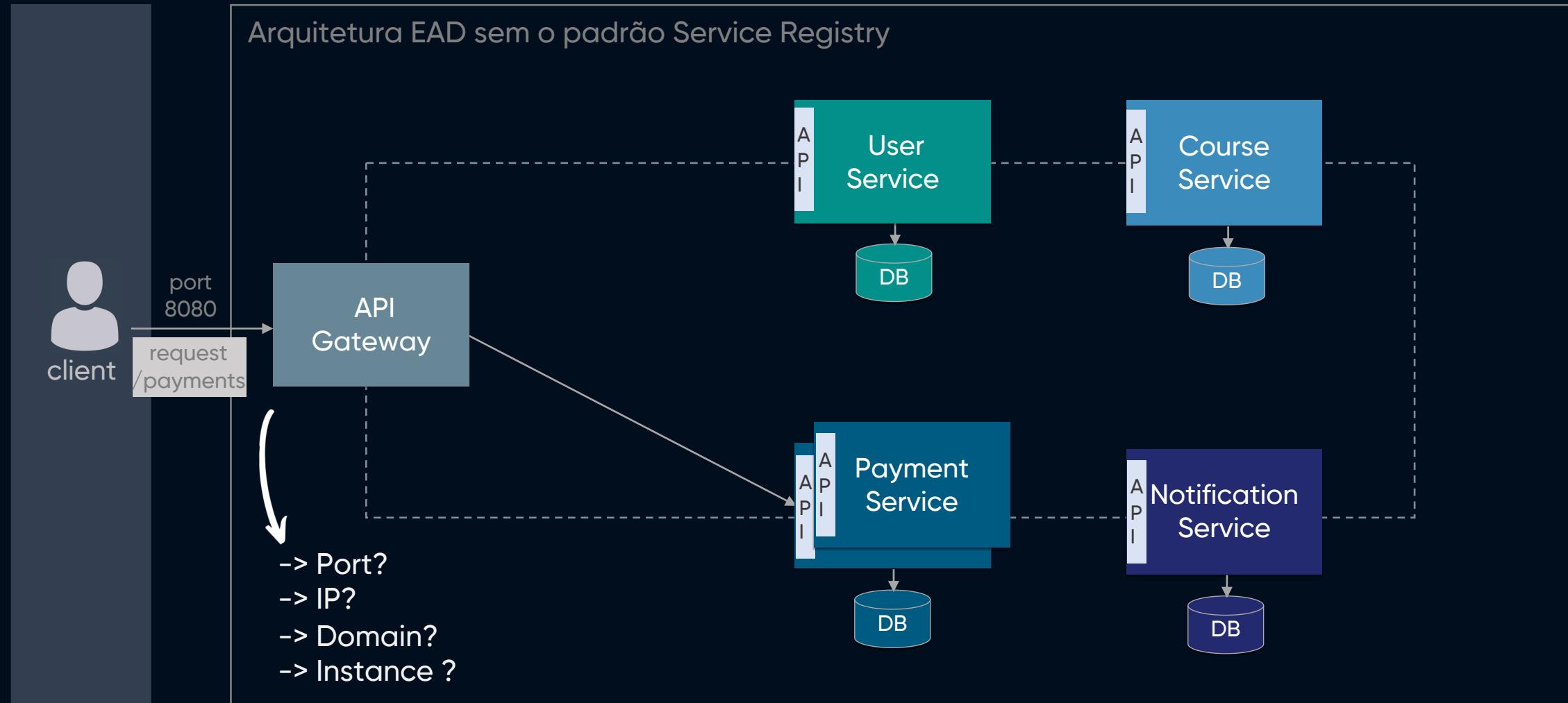
Integração com Service Registry para  
registro e descoberta de serviços

Suporte para programação Reativa

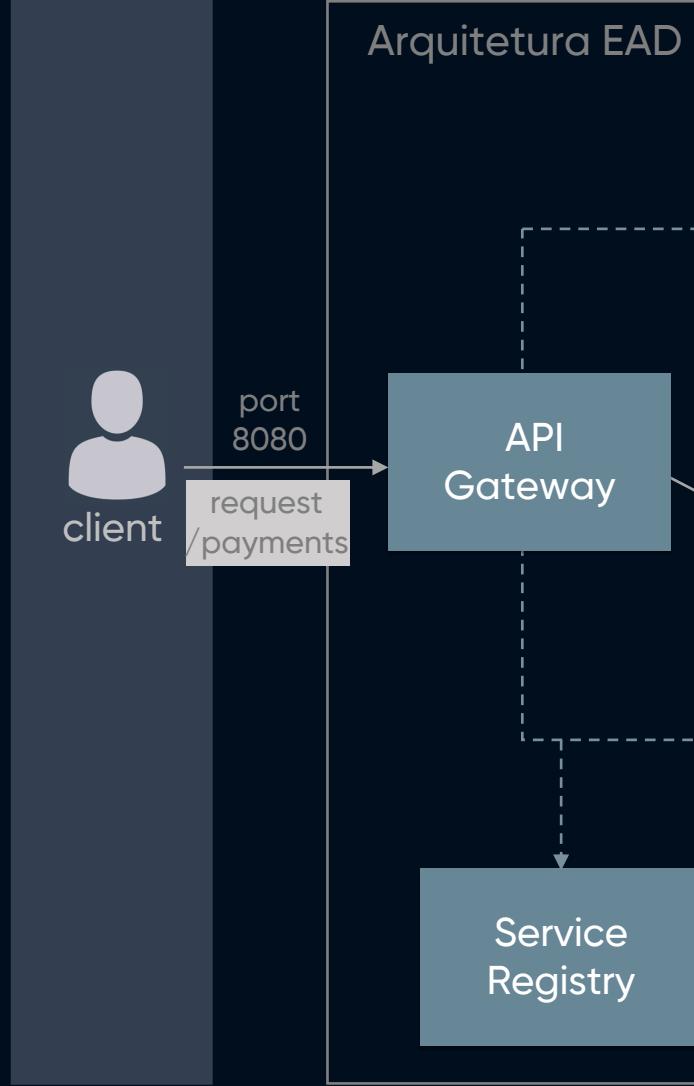
Limitação de taxa de solicitação

Servlet Container: Servidor Netty

# Modelando o padrão Service Registry / Discovery



# Modelando o padrão Service Registry / Discovery

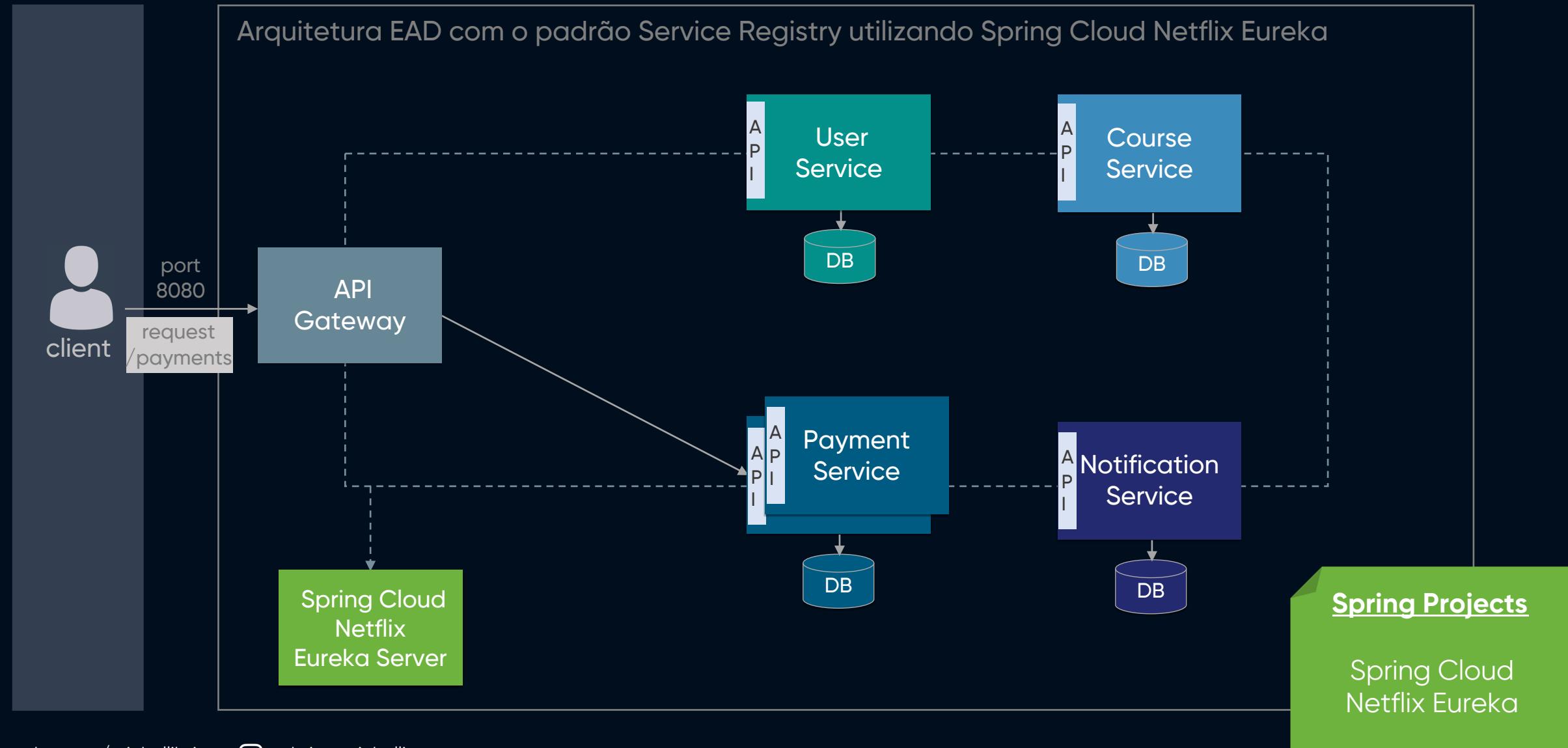


Dia 2

Microservices Patterns  
e suas aplicabilidades.

Service Registry  
também é um  
Microservice.

# Padrão Registry Discovery com Spring Cloud Netflix Eureka





# Spring Cloud Netflix Eureka

Instancias podem ser registradas em um servidor Eureka: Eureka Server

Os Eureka Clients podem descobrir essas instancias quando necessário

Dashboard Eureka para controle e visualização dos serviços registrados

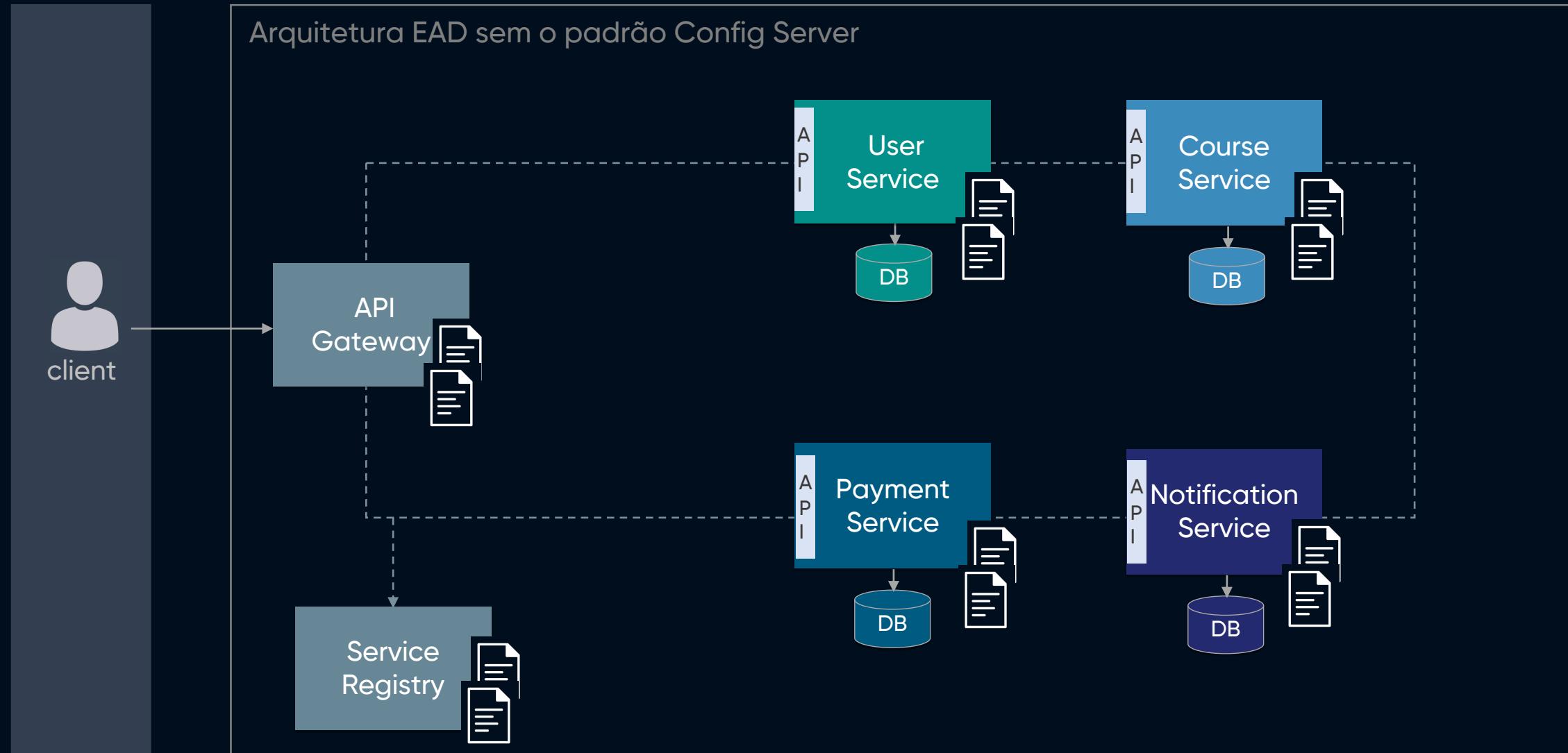
The screenshot shows the Spring Cloud Netflix Eureka dashboard. At the top, it displays 'spring Eureka' and 'HOME LAST 1000 SINCE STARTUP'. Below this, the 'System Status' section shows 'Environment' and 'Data center' both listed as 'N/A'. To the right, detailed status metrics are provided:

	Value
Current time	2022-02-20T09:24:42-0300
Uptime	00:02
Lease expiration enabled	false
Renews threshold	1
Renews (last min)	0

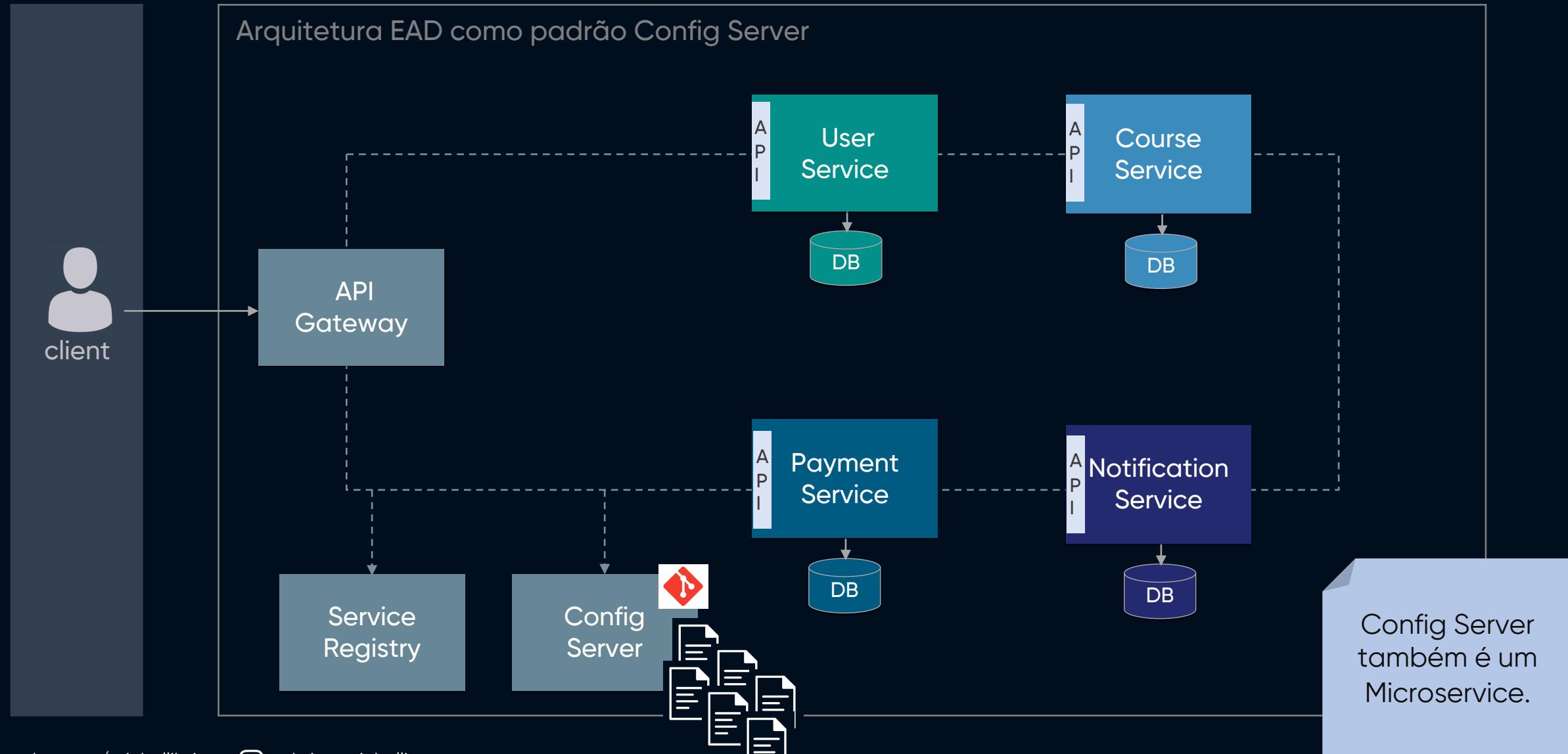
The 'DS Replicas' section shows a single entry for 'localhost'. The 'Instances currently registered with Eureka' section is empty, stating 'No instances available'. The 'General Info' section provides various service statistics:

Name	Value
total-available-memory	154mb
num-of-cpus	8
current-memory-usage	50mb (32%)
server-upptime	00:02
registered-replicas	<a href="http://localhost:8761/eureka/">http://localhost:8761/eureka/</a>
unavailable-replicas	<a href="http://localhost:8761/eureka/">http://localhost:8761/eureka/</a>
available-replicas	

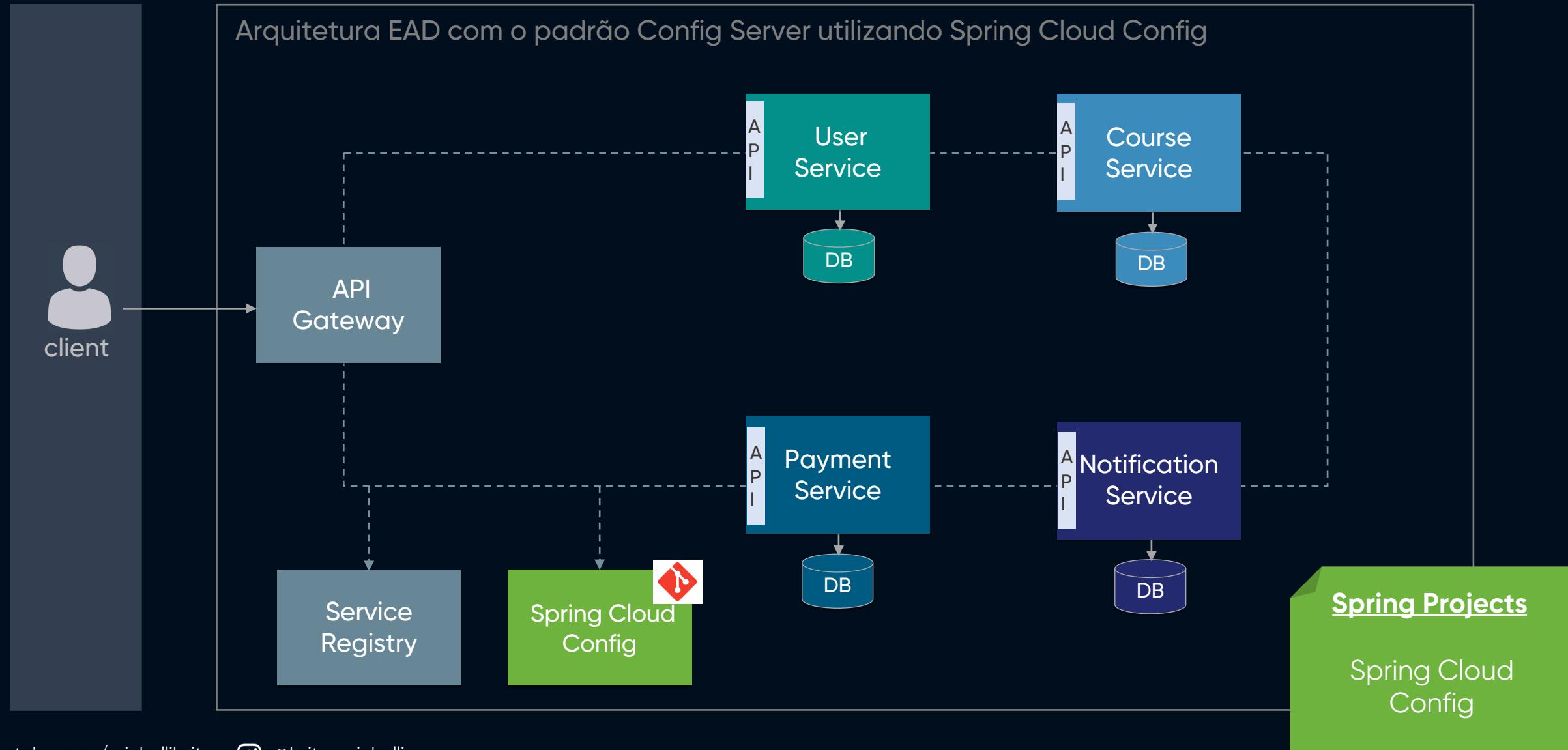
# Modelando o padrão Config Server



# Modelando o padrão Config Server



# Padrão Config Server com Spring Cloud Config





# Spring Cloud Config

Suporte para configuração externa  
em uma arquitetura de Microservices

Utiliza do Git para armazenamento e  
versionamento das propriedades

Config Server para gerenciar  
configurações de múltiplos ambientes

Criptografia e Descriptografia de  
valores de propriedades

APIs para consultas de propriedades

Suporta uso de @Value

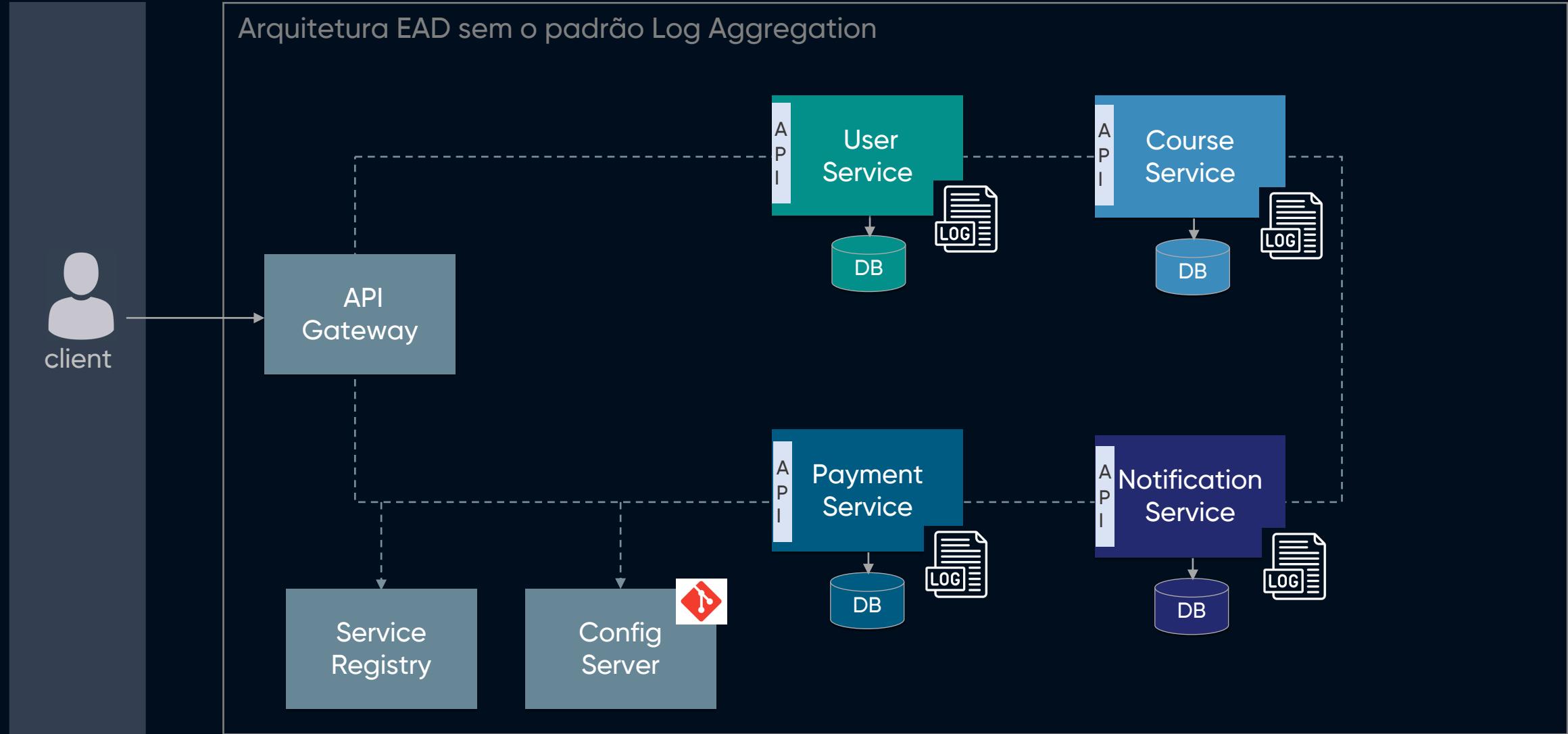
# Modelando padrões de Observabilidade em Microservices

Log Aggregation

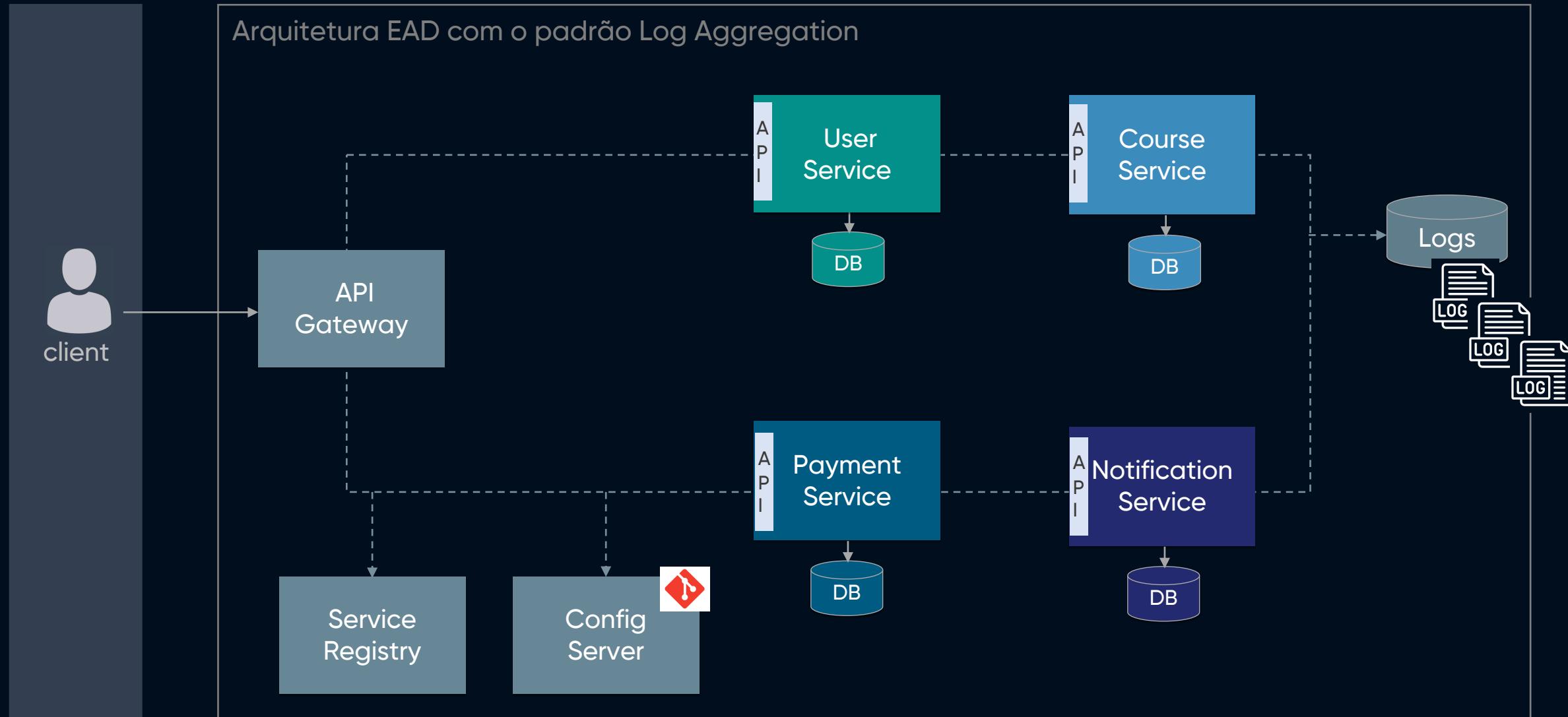
Distributed Tracing

Metrics

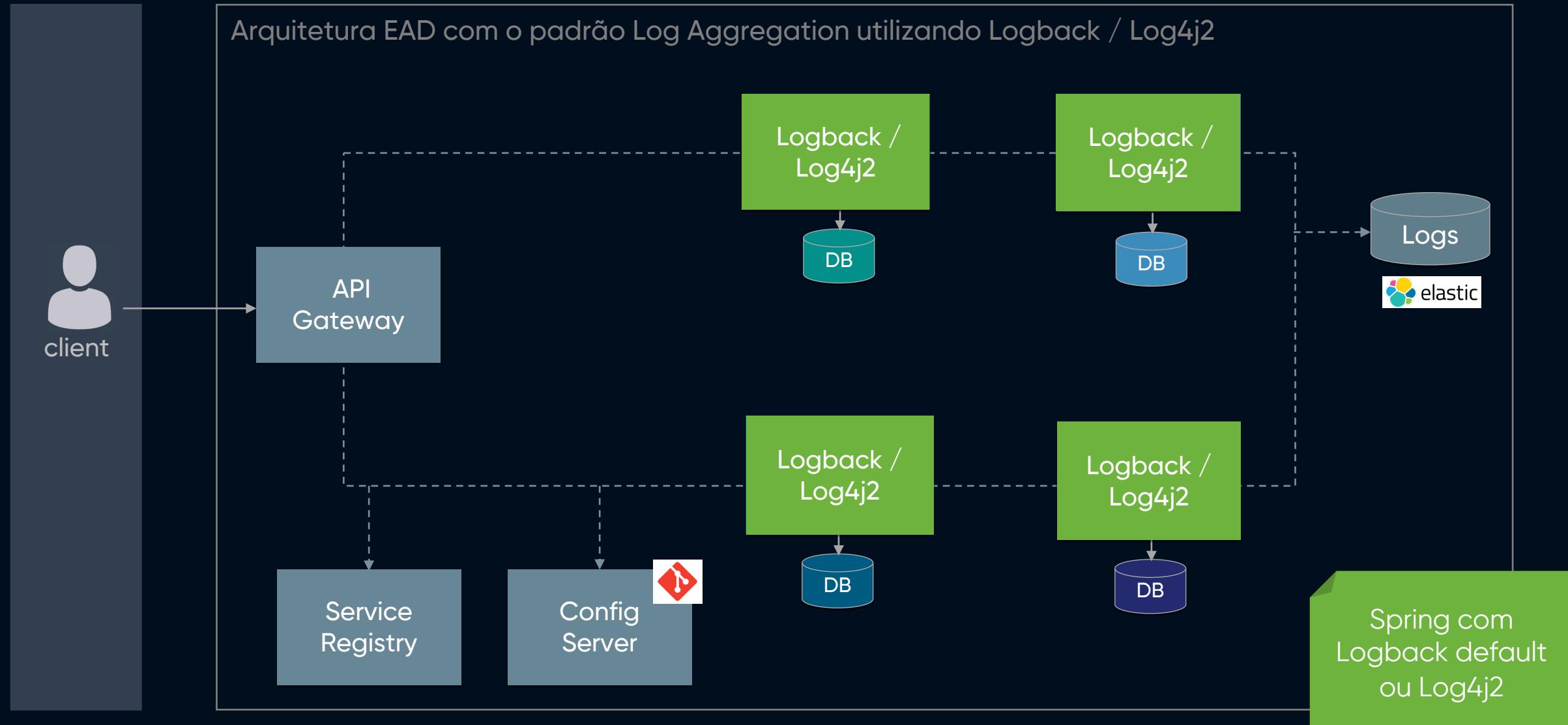
# Modelando Observabilidade com Log Aggregation



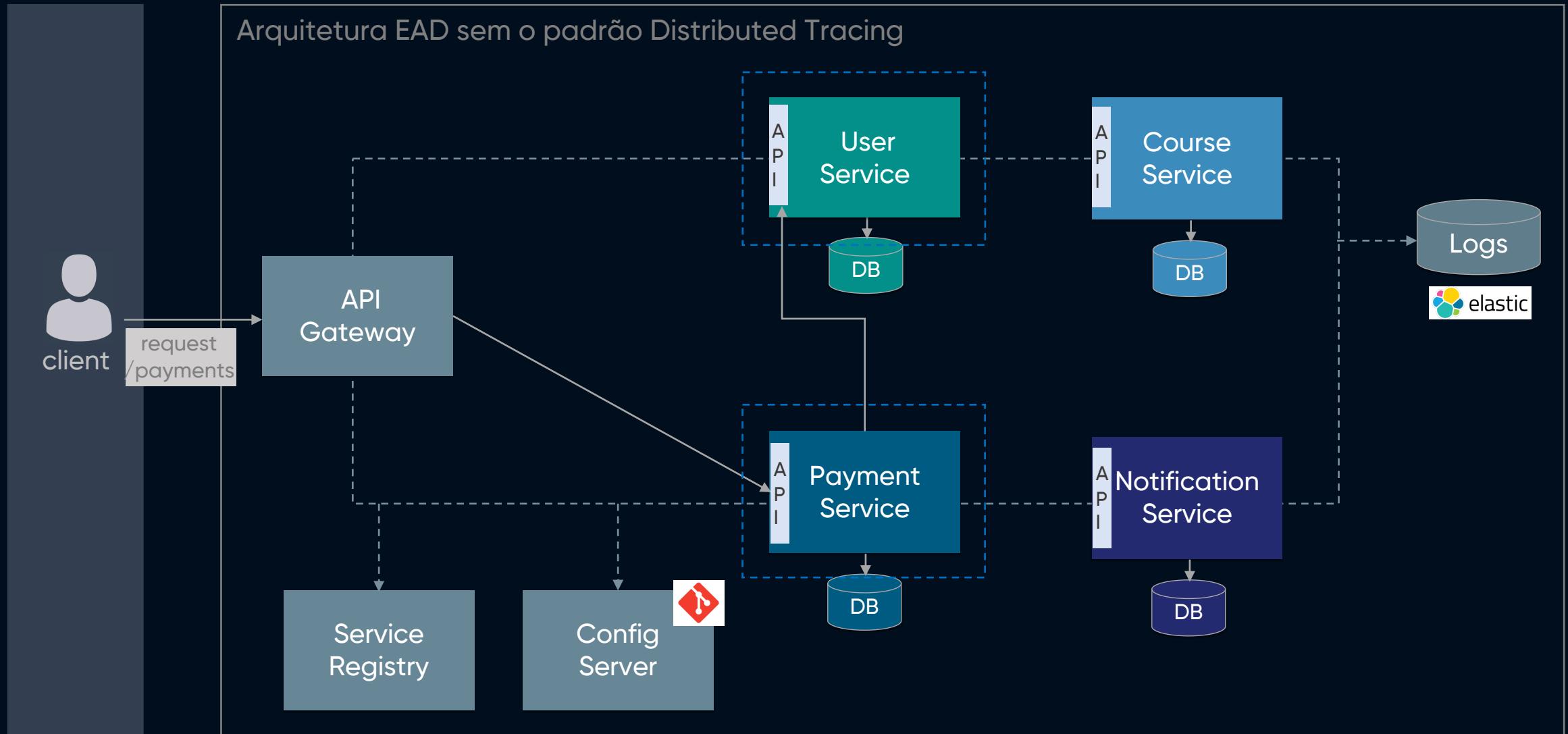
# Modelando Observabilidade com Log Aggregation



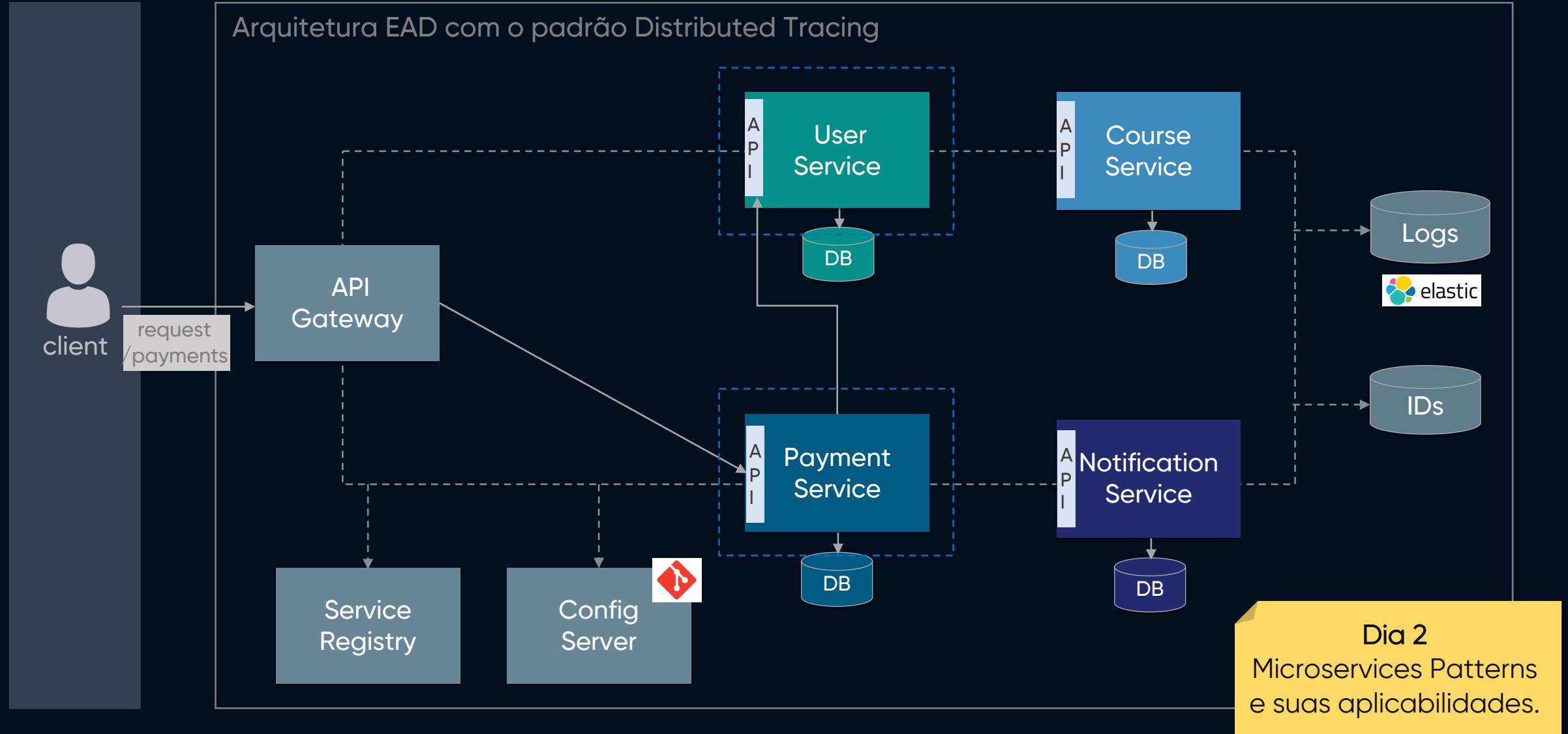
# Padrão Log Aggregation com ELK e Logback/Log4j2



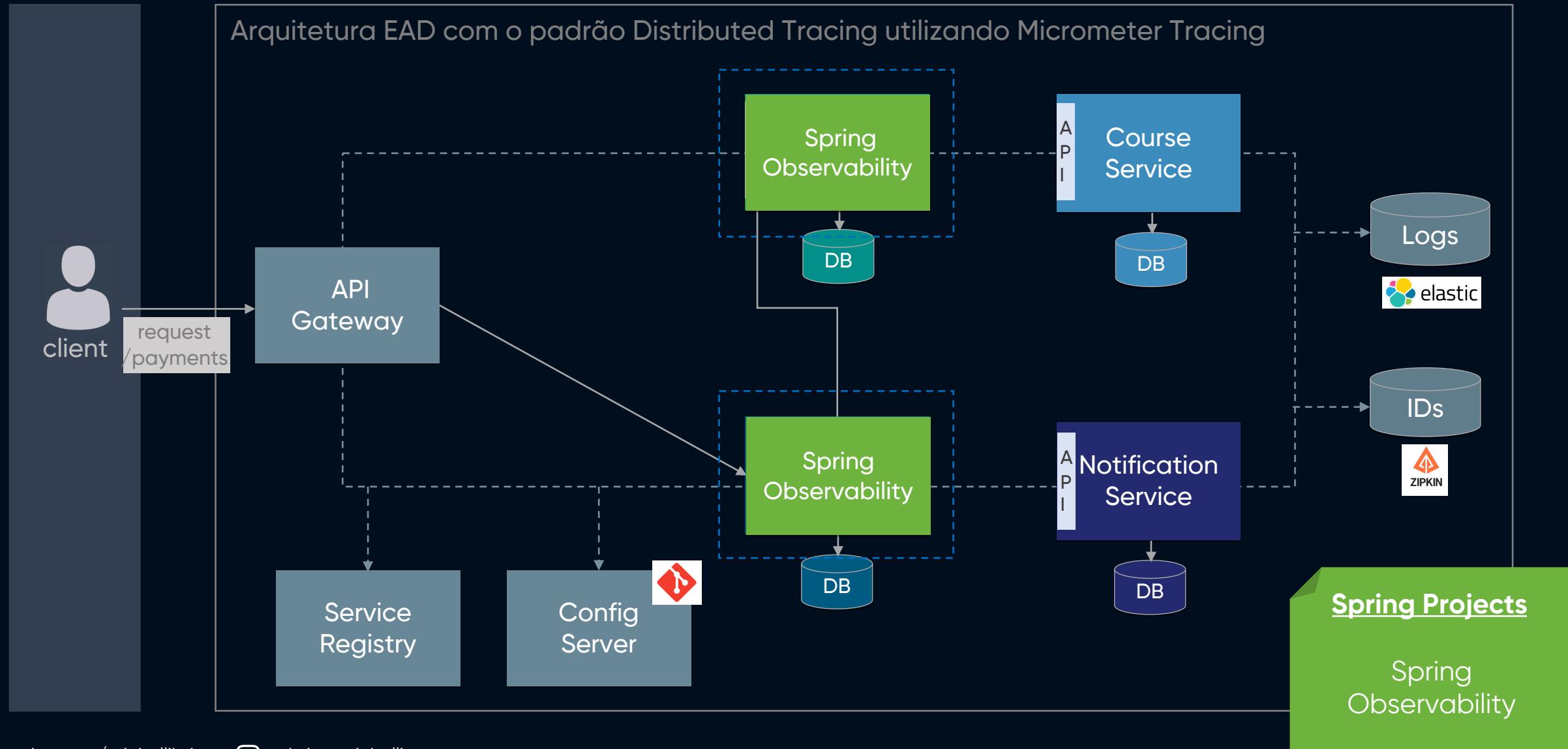
# Modelando Observabilidade com Distributed Tracing



# Modelando Observabilidade com Distributed Tracing



# Padrão Distributed Tracing com Spring Observability





# Spring Observability

Disponível na versão Spring Boot 3 e  
Spring Framework 6

Baseado no projeto Micrometer e  
Micrometer Tracing

Antigo projeto Spring Cloud Sleuth

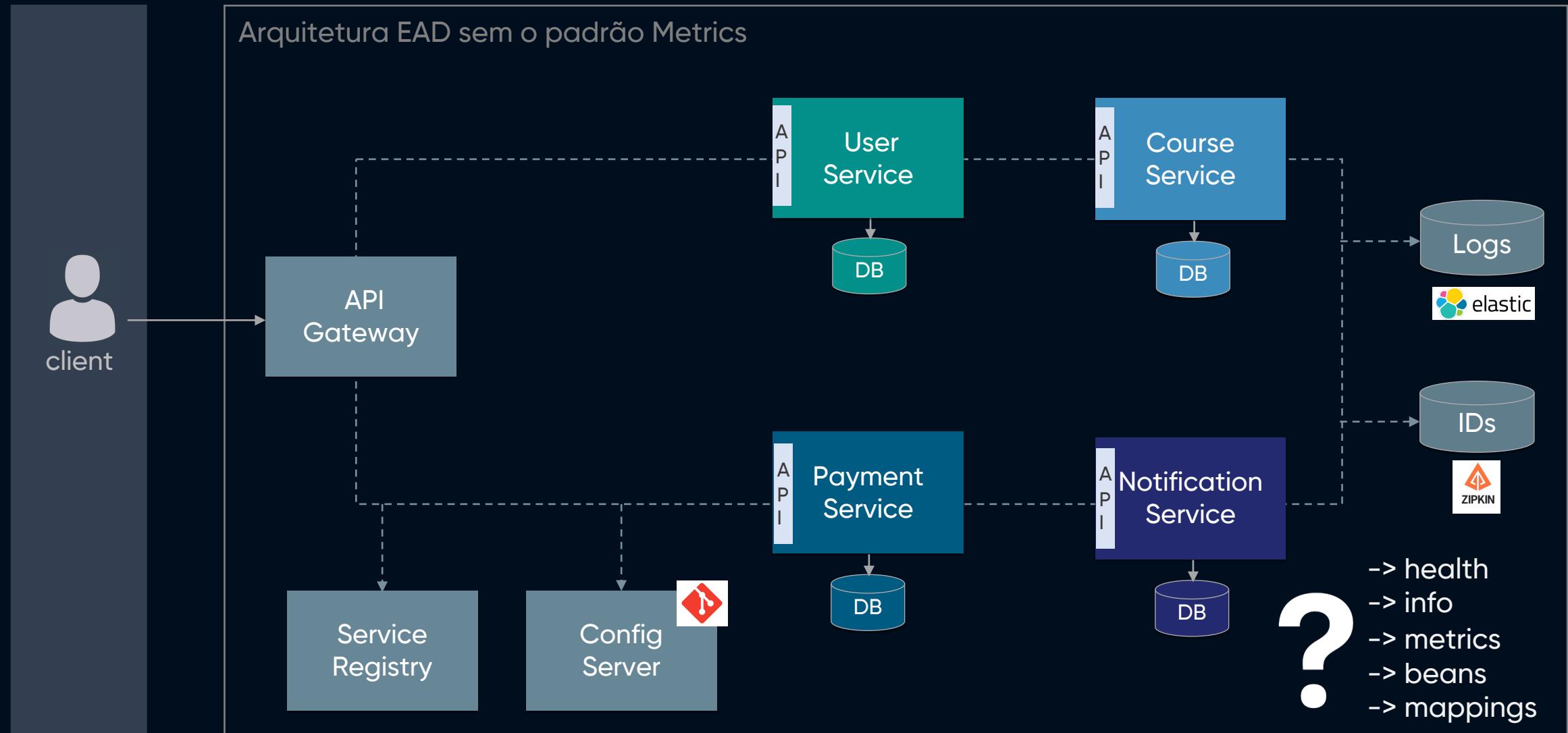
Integração com Spring Actuator

Observabilidade em operações únicas  
ou rastreamento completo de  
solicitações HTTP

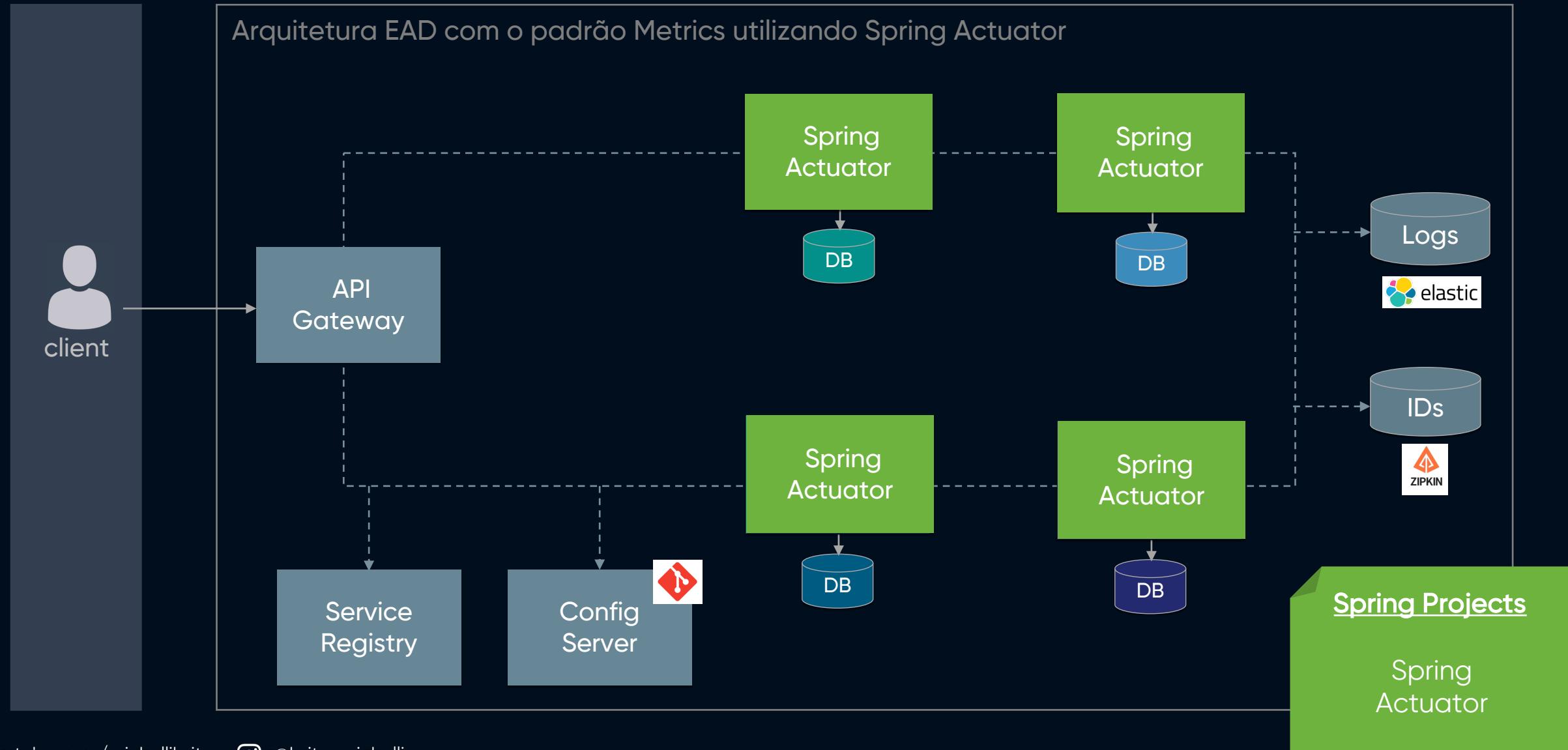
Múltiplas funções como métricas,  
rastreamento e logs

Compatível com Zipkin para  
visualização de IDs de rastreamento

# Modelando Observabilidade com Metrics



# Modelando Observabilidade com Metrics

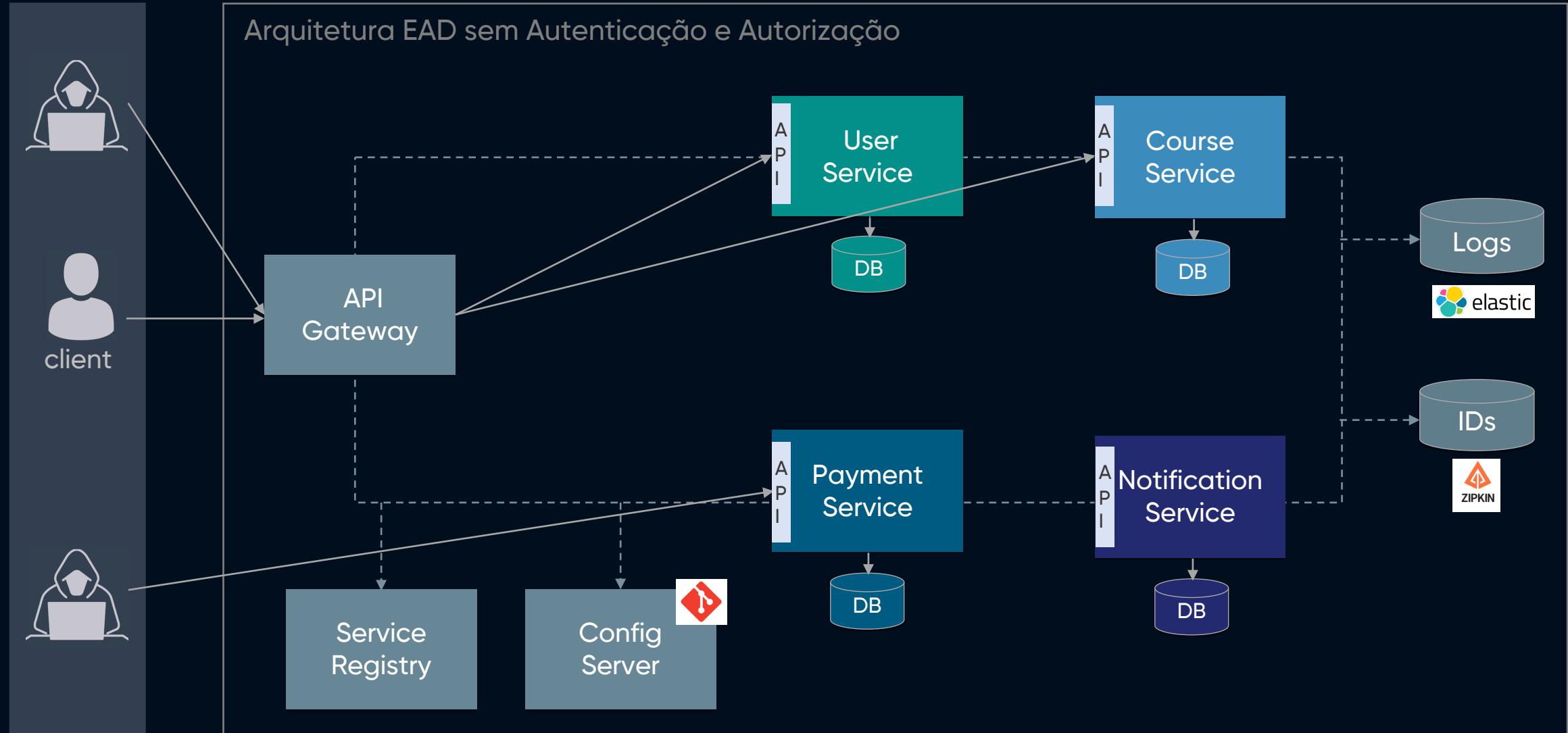


# Modelando padrões de Autenticação em Microservices

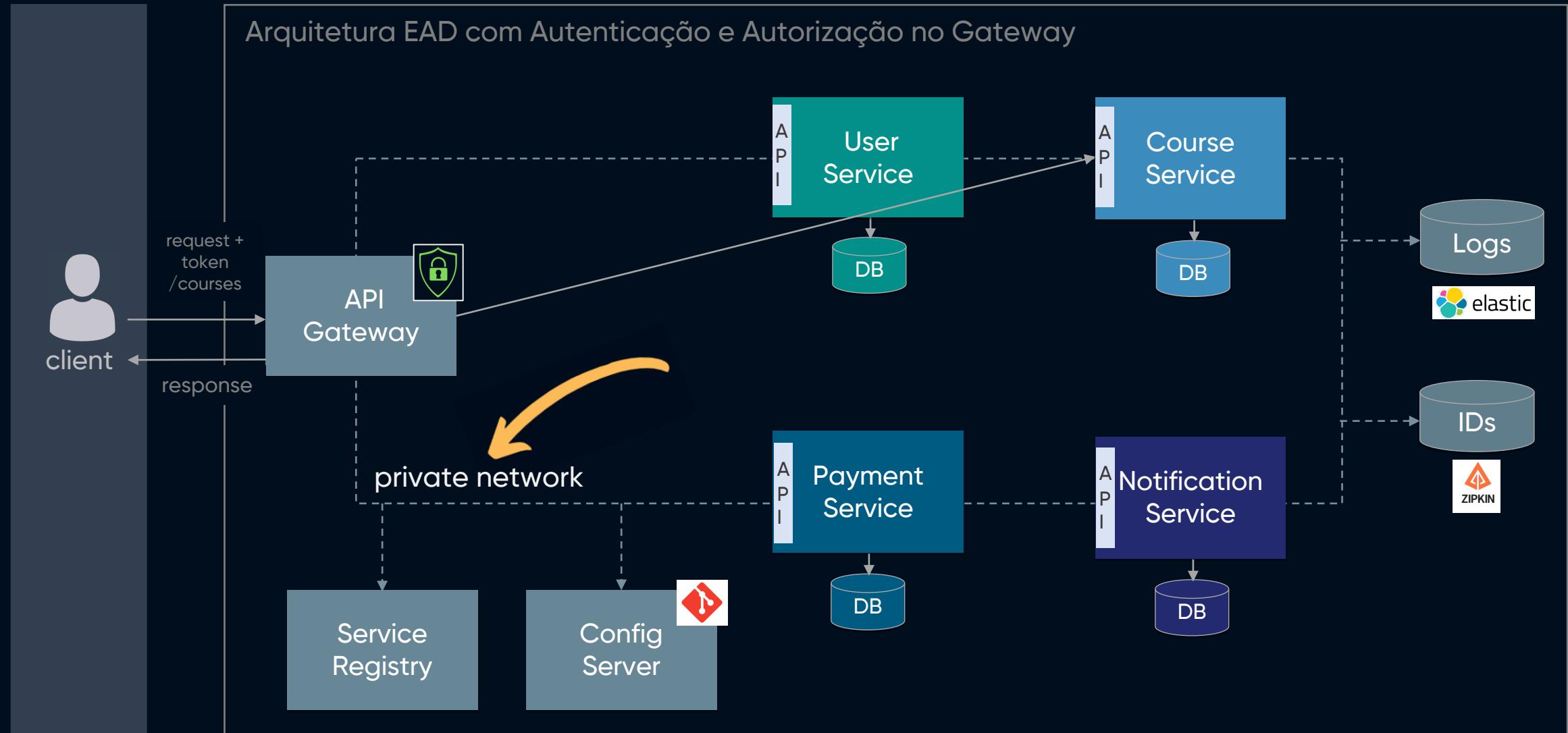
Autenticação e Autorização no Gateway

Autenticação e Autorização Distribuída  
com o padrão Access Token

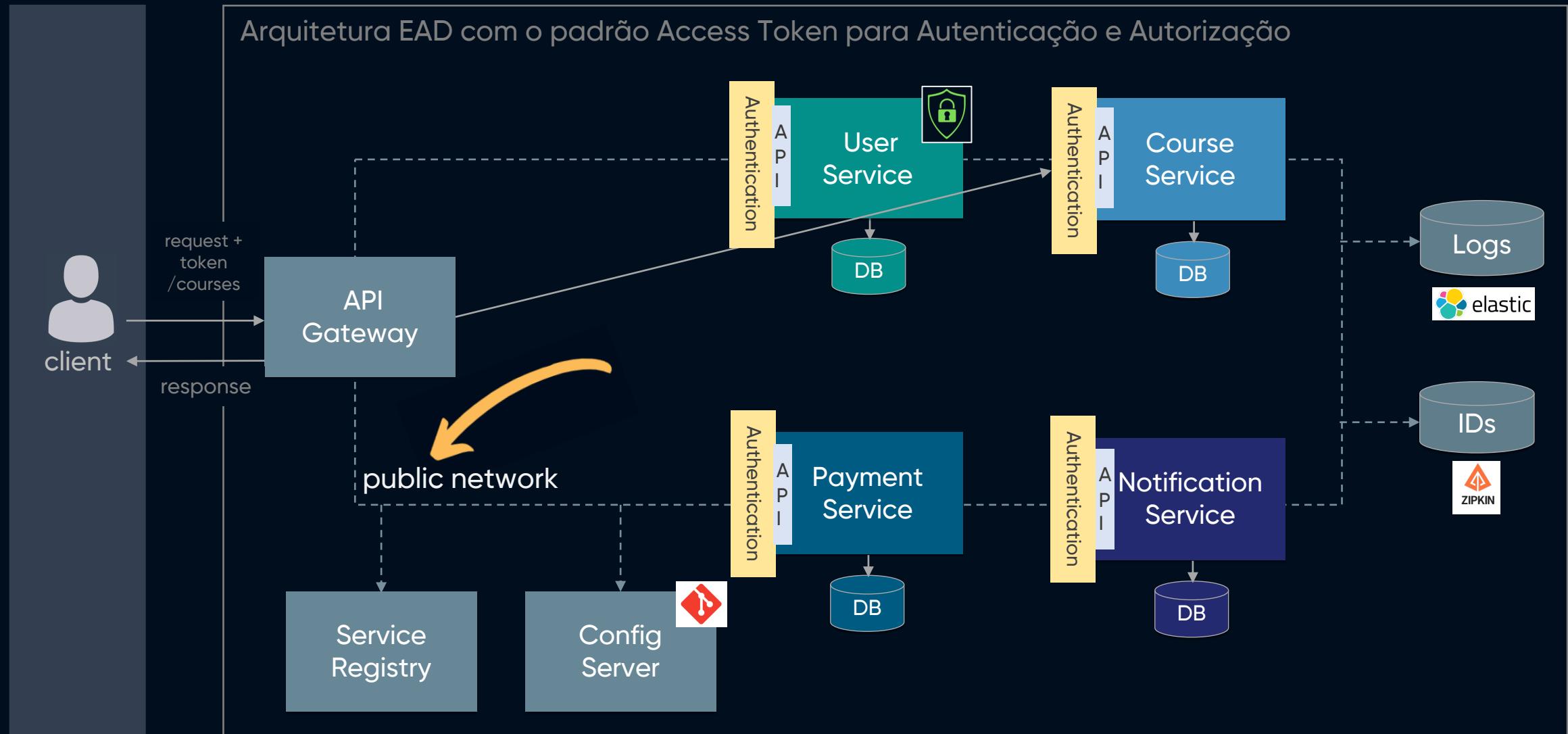
# Modelando Autenticação e Autorização em Microservices



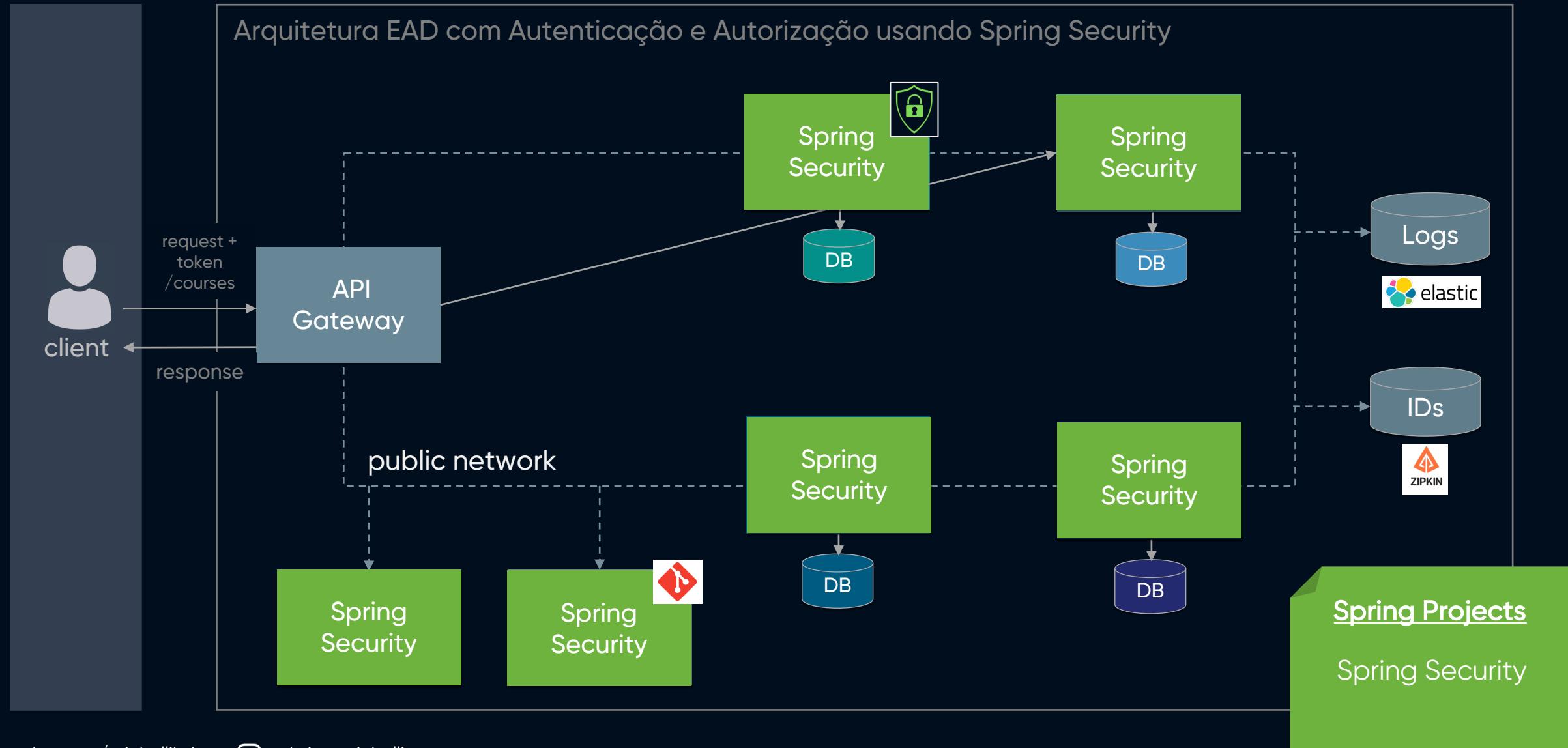
# Modelo 1: Autenticação e Autorização no Gateway



# Modelo 2: Autenticação e Autorização Distribuída com o padrão Access Token



# Autenticação e Autorização com Spring Security





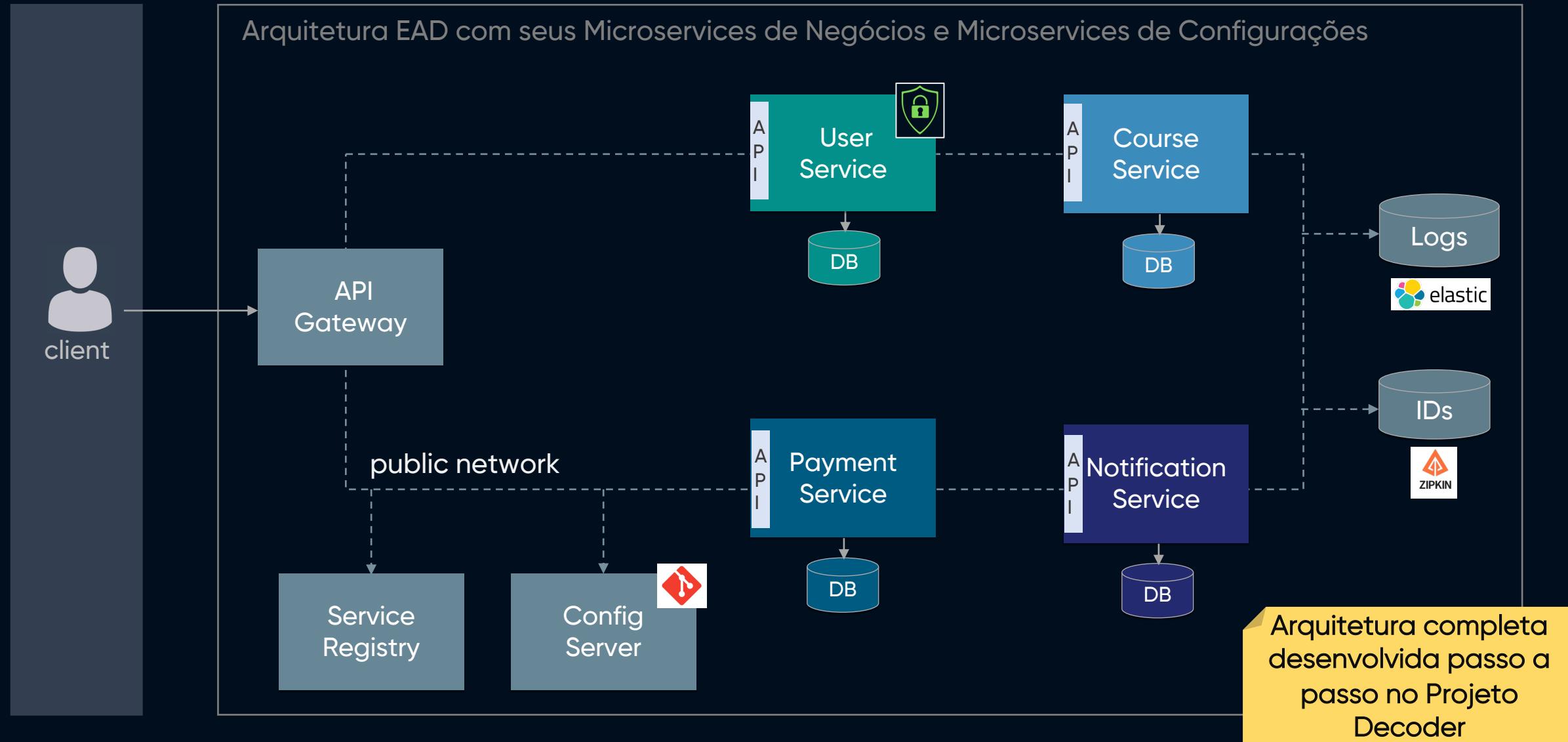
# Spring Security

Projeto que fornece autenticação e autorização para aplicações Java

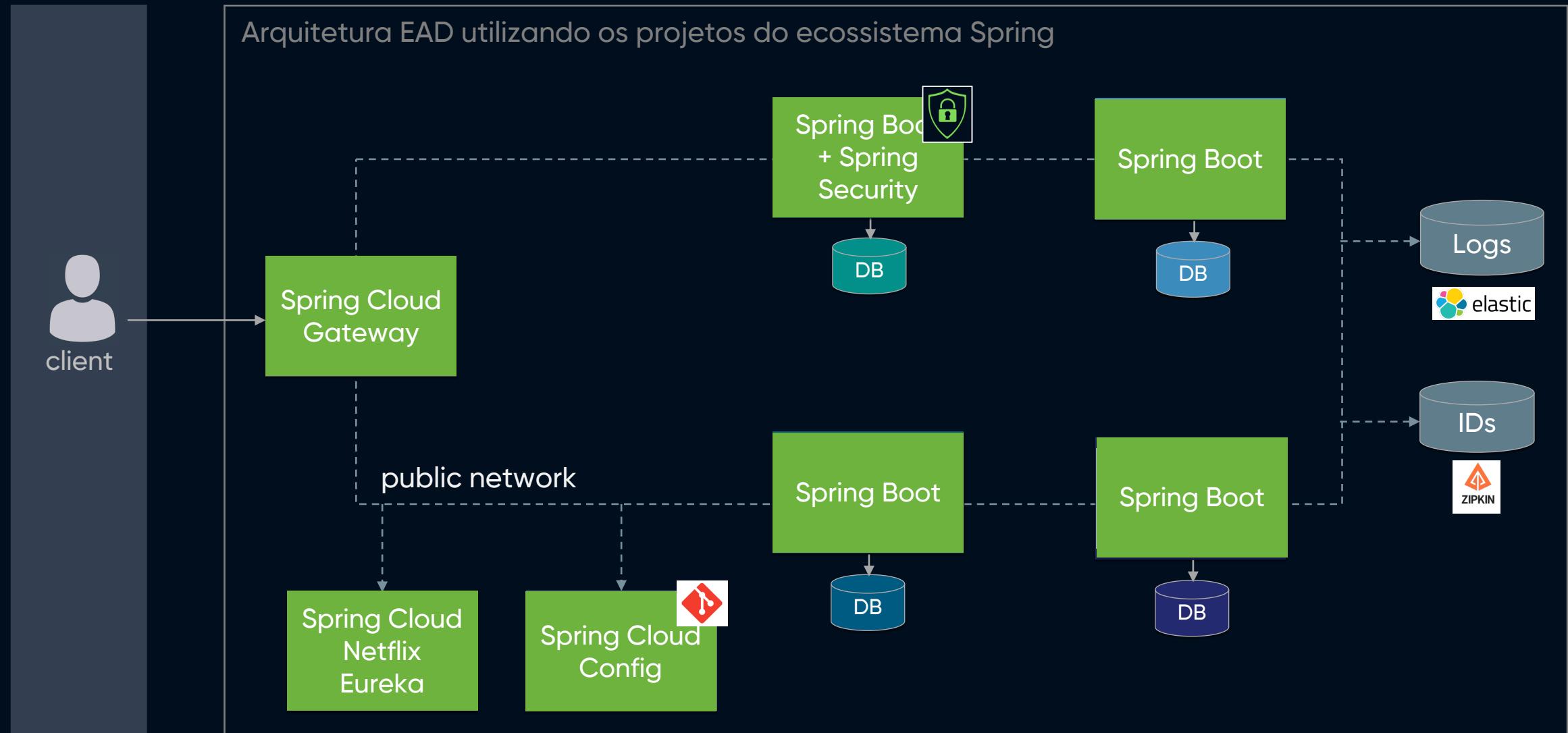
Suporte para implementação de Basic Authentication em memória e em database, JWT e OAuth

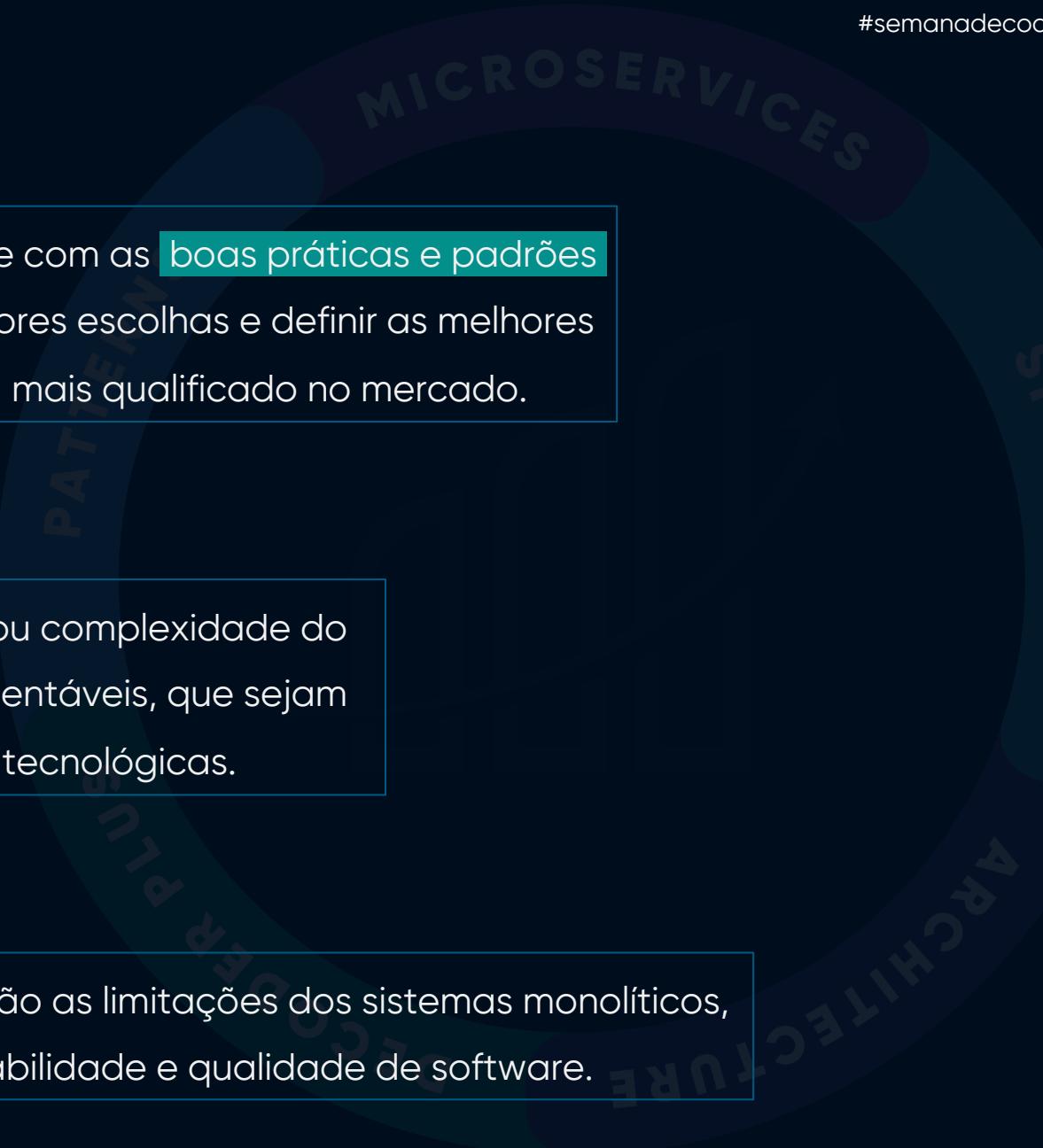
# Arquitetura Completa

# Arquitetura completa do **SISTEMA EAD**



# Arquitetura completa do **SISTEMA EAD**



MICROSERVICES  
PATTERN  
SPRING  
ARCHITECTURE

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 sobre o tamanho ou complexidade do negócio, mas principalmente sobre projetar sistemas sustentáveis, que sejam capazes de acompanhar mudanças de negócios e tecnológicas.

A Arquitetura de Microservices veio como solução as limitações dos sistemas monolíticos, principalmente se tratando de sustentabilidade e qualidade de software.