

# ***WEB SERVICE E SPRING***

# ***ATIVIDADES DE HOJE***

## Horário das atividades

- 13h30 Faísca
- 13h45 Retrospectiva
- 15h00 Intervalo
- 15h15 Introdução ao Web Service e Spring
- 16h40 Checkout
- 17h00 Chamada



# ***RETROSPECTIVA***

**Objetivo proposto:** Melhorar a colaboração entre os grupos, identificar boas práticas ou desafios comuns no desenvolvimento Java, entre outros.

# Retrospectiva

## Etapas da Retrospectiva

- **Coleta de Dados** - Coletar dados sobre a Sprint. Isso pode incluir métricas de desempenho, como a velocidade da equipe, itens do Product Backlog concluídos e não concluídos, incidentes relatados, entre outros.
- **Geração de insights** - A equipe se reúne para analisar e discutir o que aconteceu durante a Sprint. Os membros compartilham suas percepções sobre o que funcionou bem e o que não funcionou. Isso leva à identificação de pontos fortes e áreas que precisam de melhoria.
- **Definição de planos de ação** – A equipe Scrum colaborativamente identifica ações concretas para melhorar o processo na próxima Sprint. Isso pode envolver a criação de tarefas específicas, ajustes em práticas ou até mesmo mudanças na forma como a equipe trabalha.
- **Encerramento**



# Estruturas de retrospectiva

## Começar, Parar, Continuar

- **Começar:** Nesta parte, os participantes sugerem atividades, práticas ou processos que a equipe deve iniciar para melhorar o trabalho.
- **Parar:** Aqui, a equipe identifica atividades, práticas ou processos que não estão sendo eficazes e devem ser interrompidos ou abandonados.
- **Continuar:** Nesta etapa, a equipe destaca o que está funcionando bem e deve ser mantido ou aprimorado.



# Estruturas de retrospectiva

## Três porquinhos

- **Casa de Palha :** O que está literalmente pendurado por um fio e pode cair facilmente a qualquer minuto? O objetivo é identificar qualquer coisa que esteja em risco de quebrar. São tarefas ou questões de alta prioridade.
- **Casa de madeira:** O que já é bastante sólido, mas ainda pode ser melhorado? Estas questões ainda requerem atenção, mas têm uma prioridade inferior às tarefas de alta prioridade mencionadas acima.
- **Casa de Tijolos:** O que fizemos que é 100% sólido como uma rocha? Essas tarefas são resultados dos quais uma equipe pode se orgulhar e, portanto, não requerem mais trabalho ou atenção



# Estruturas de retrospectiva



**Soluções/Ajustes**

# ***WEB SERVICE***





# O que é Web Service?

- Os Web Services são componentes que permite que as aplicações enviem e recebam dados geralmente em formato XML ou JSON.
- Cada aplicação pode ter a sua própria linguagem que ao se comunicar com outra aplicação diferente, mesmo sendo em outro servidor e desenvolvida usando outra linguagem de programação, a comunicação é traduzida para os formatos XML ou JSON.
- O Mesmo Web Service pode ser acessado por diferentes aplicações em diferentes plataformas.

# O que é Web Service?

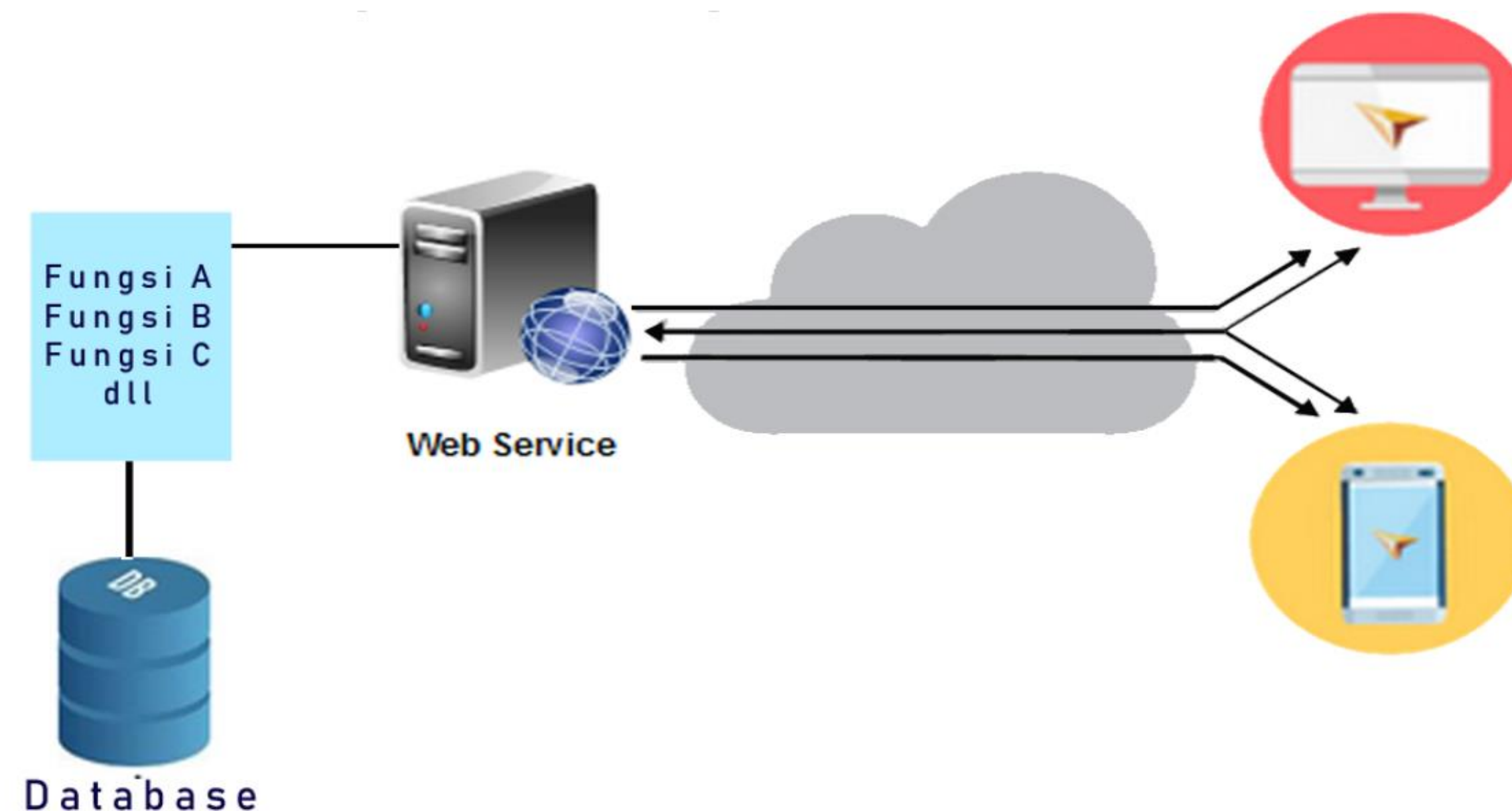
- Recebe e faz chamadas (Requisições e Respostas) com qualquer linguagem.
- Permite a comunicação entre ambientes
- Comunicação independente de sistema operacional e de linguagem de programação;



Uma empresa pode ter um software para desktop em .net, Um para iOS nativo em Objective C, Um para Android nativo em Java e uma aplicação Web com front-end desenvolvido em Javascript utilizando React e todos podem estar conectados ao mesmo Web Service e ele que “conversa” com o Banco de Dados e serve como back-end para todas as funcionalidades.

# O que é Web Service?

- Programa que você cria para que uma ponta se comunique com outra ponta;
- Banco de dados, dentro do application service roda o seu programa de web service, softwares podem utilizar desse web service para chegar ate o banco de dados e falar **“Hey eu quero consultar os dados aqui porque quero jogar para o meu sistema”**





# O que é API?

## Application Program Interface

- Interface entre aplicativo e programação;
- É criada para que um sistema possa utilizar as funcionalidades de outro sistema, ou seja, que eles se comuniquem;
- Elas não precisam trabalhar via web. Já que o conceito pode estar presente em qualquer tipo de sistema...

.... até o sistema operacional.

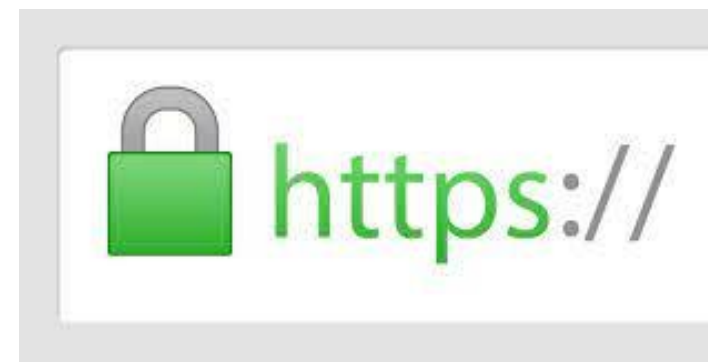
Permitindo controlar sistemas de arquivos, permissões, redes, dispositivos, etc...





# Segurança na API

- Uso de SSL nas conexões, isso quer dizer que toda comunicação será transportada Criptografada, através do protocolo HTTPS



- Autenticação para proteção de dados sensíveis. Sendo a maneira preferida a geração de TOKENS (chaves guardadas e validadas como se fosse uma senha) que são enviados junto às chamadas.

# Retorno API

Que tipo de info ela retorna, na **WEB**

- Formato utilizado para retornar os dados das API's
- JSON – Ocupa pouco espaço em memória e muito fácil de transportar via rede. Amplamente utilizado em aplicativos da web e APIs REST.
- XML - Usado em várias tecnologias, incluindo configurações
- de documentos e intercâmbio de dados em muitos contextos.

```
{  
  "nome": "João Silva",  
  "idade": 30,  
  "cidade": "São Paulo",  
  "interesses": ["Programação", "Viagens", "Esportes"],  
  "contato": {  
    "email": "joao@example.com",  
    "telefone": "+55 (11) 1234-5678"  
  }  
}
```

```
< Pessoa >  
  < nome >João Silva</ nome >  
  < idade >30</ idade >  
  < cidade >São Paulo</ cidade >  
  < interesses >  
    < interesse >Programação</ interesse >  
    < interesse >Viagens</ interesse >  
    < interesse >Esportes</ interesse >  
  </ interesses >  
  < contato >  
    < email >joao@example.com</ email >  
    < telefone >+55 (11) 1234-5678</ telefone >  
  </ contato >  
</ Pessoa >
```

# API e Web Service

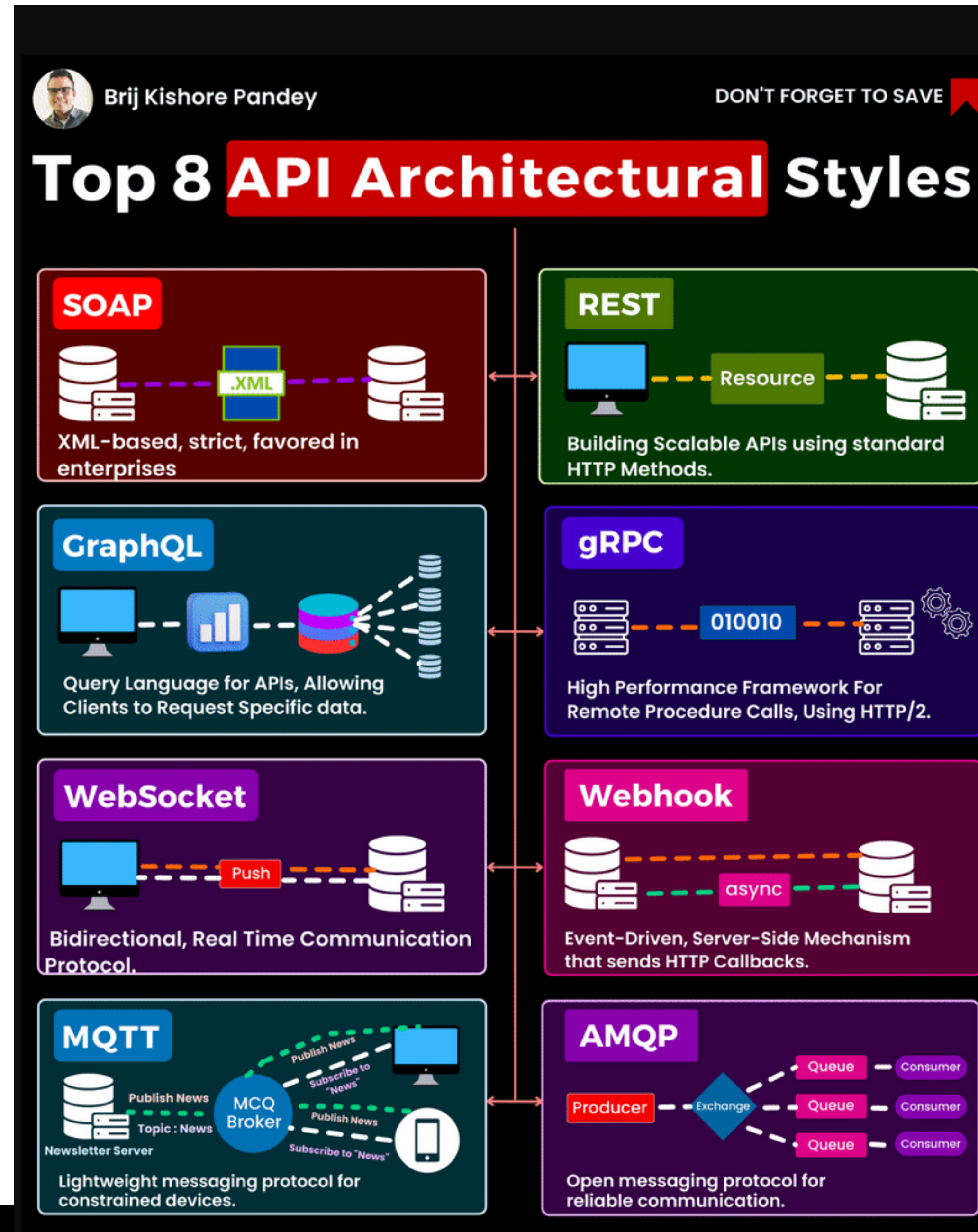
## API X Web Service

- Compilado de funcionalidade mais direcionado;
- API pode usar qualquer estilo de comunicação;
- Não necessitam de rede;
- Facilita a interface direta com um aplicativo;
- Nem todas as APIs são Web services.
- Compilado de funcionalidades atra~ões da rede;
- Um serviço Web utiliza apenas três estilos de comunicação: SOAP, REST e XML-RPC;
- Um Web Service sempre precisa de uma rede para o seu funcionamento;
- É uma aplicação;
- Todos os Web Services são APIs.



# Tipos de API

- SOAP (Simple Object Access Protocol)
- REST (Representational State Transfer)
- Embora SOAP e REST sejam os mais conhecidos, há outros tipos de web services, como **\*\*JSON-RPC\*\***, **\*\*XML-RPC\*\*** e **\*\*gRPC\*\***, que são usados em casos específicos ou em ambientes que exigem requisitos particulares.





# O que é SOAP?

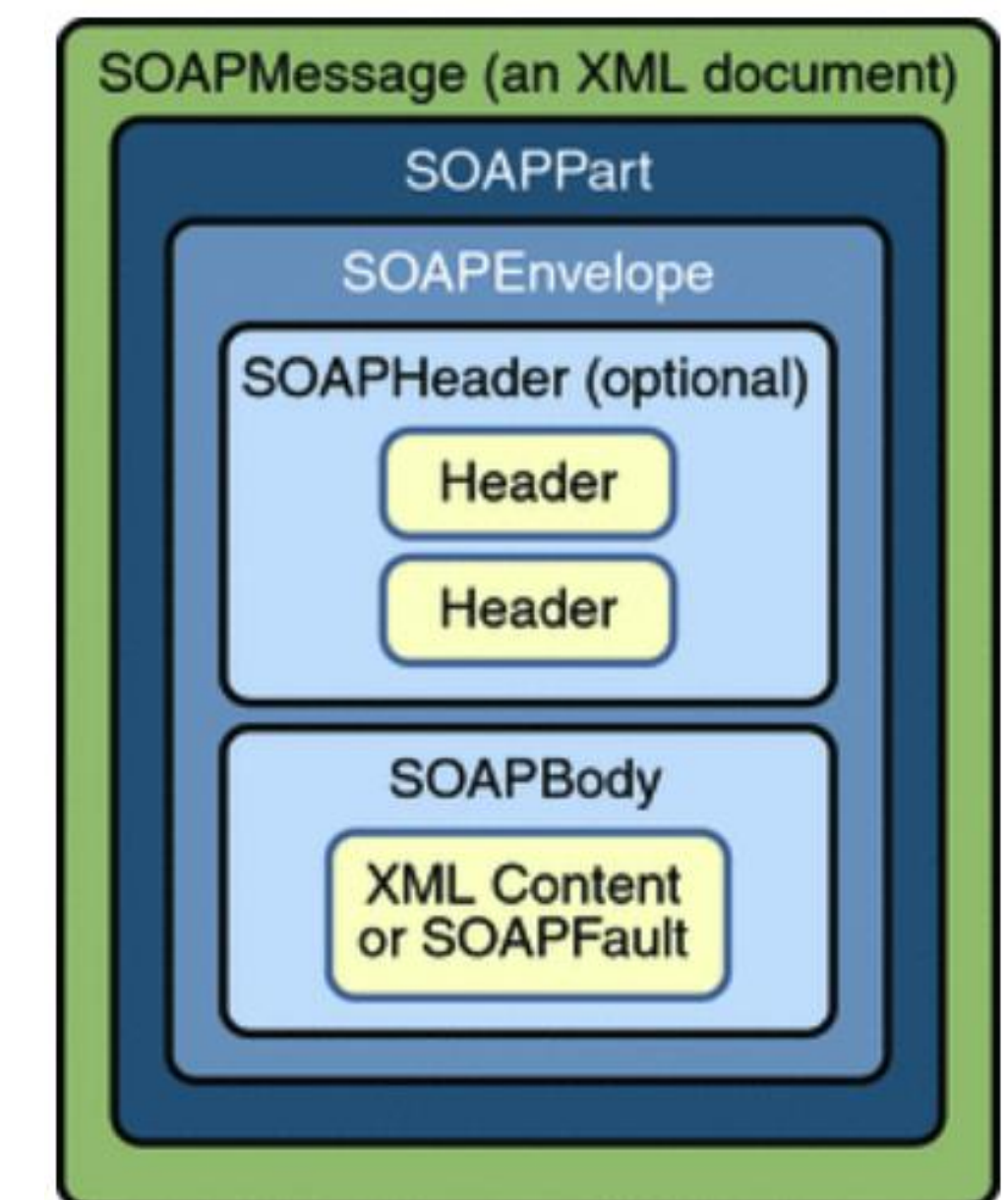
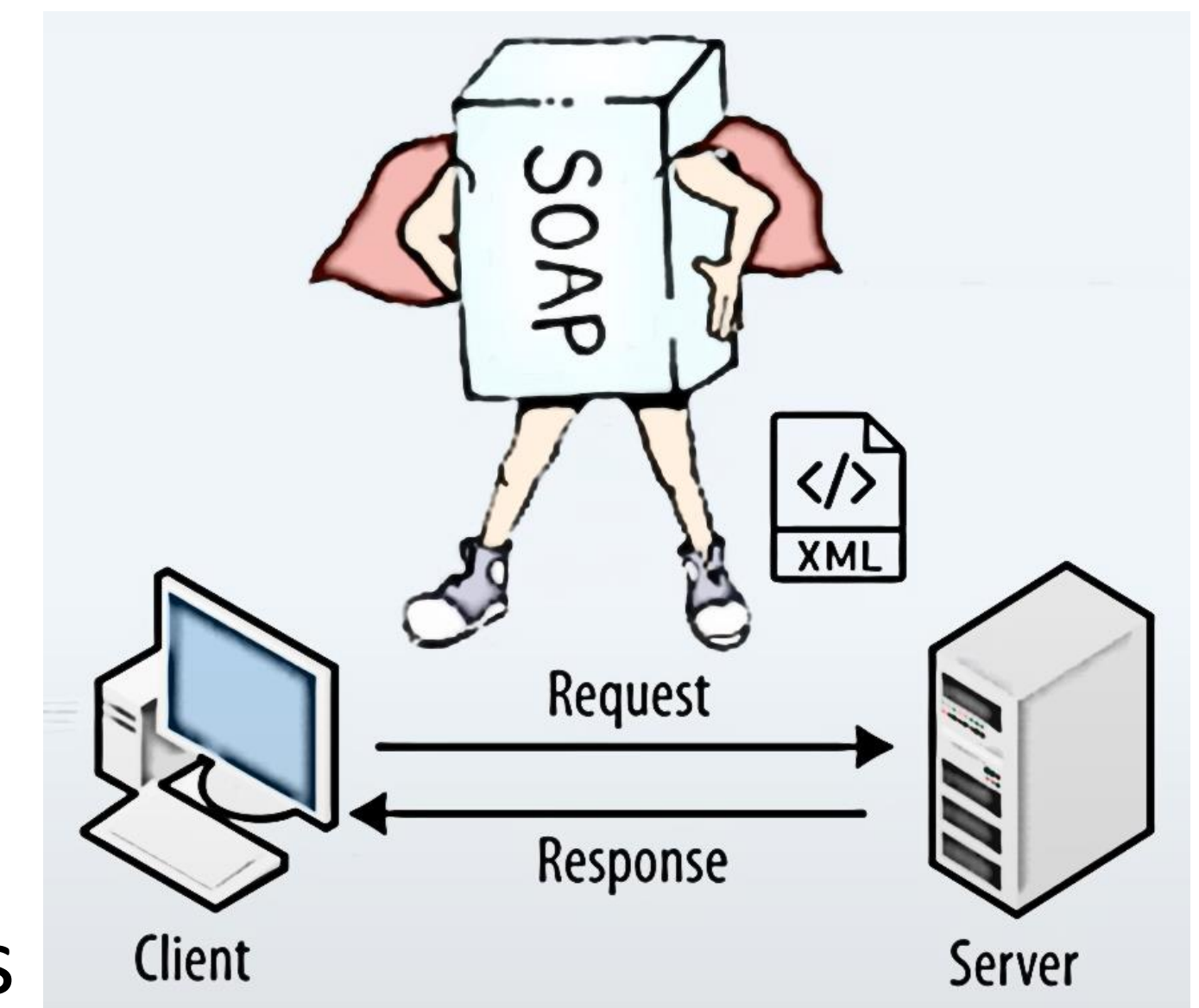
## Simple Object Access Protocol

- Protocolo de comunicação baseado em XML (Extensible Markup Language) que consiste em quatro elementos com funções específicas para cada um deles.
- **Envelope:** Define o que está na msg e como processá-la.
- **Cabeçalho(header):** Determina os detalhes e requisitos adicionais para a mensagem, como autenticação.
- **Corpo(body):** Inclui a solicitação ou resposta.
- **Erros (Fault):** mostra todos os dados sobre quaisquer erros que possam surgir durante a solicitação e resposta da API.

<https://www.guru99.com/soap-simple-object-access-protocol.html>

<https://medium.com/@derya.cortuk/soap-web-services-2fb7e921eb1e>

- Embora o SOAP tenha muitas vantagens em termos de segurança e estruturação de dados, ele também pode ser mais pesado em comparação com alternativas mais leves, como REST.



# O que é API REST?

## Representational State Transfer

- É uma API como outra qualquer, mas utilizando o protocolo HTTP (versão 1.1)
- O protocolo HTTP (Hypertext Transfer Protocol) aceita alguns métodos com (GET, POST, HEAD), o que permite entrar nas páginas, enviar formulários com os dados, etc...
- O REST, estamos falando do HTTP 1.1 que aceita novos métodos como o PUT, DELETE, TRACE e o CONNECT, além dos outros.



# O que é API REST?

Por que ficou tão famoso?

- É utilizado como um padrão para fazer a troca de informações entre as aplicações.
- Conceitos que visam utilizar de forma mais eficiente o protocolo de comunicação HTTP (de forma mais amigável).
- Utiliza de forma eficiente a semântica do HTTP:
- GET – Método de consulta
- POST – Envia as informações
- PUT – Atualizar as informações
- DELETE – Remove informação



# Características do REST

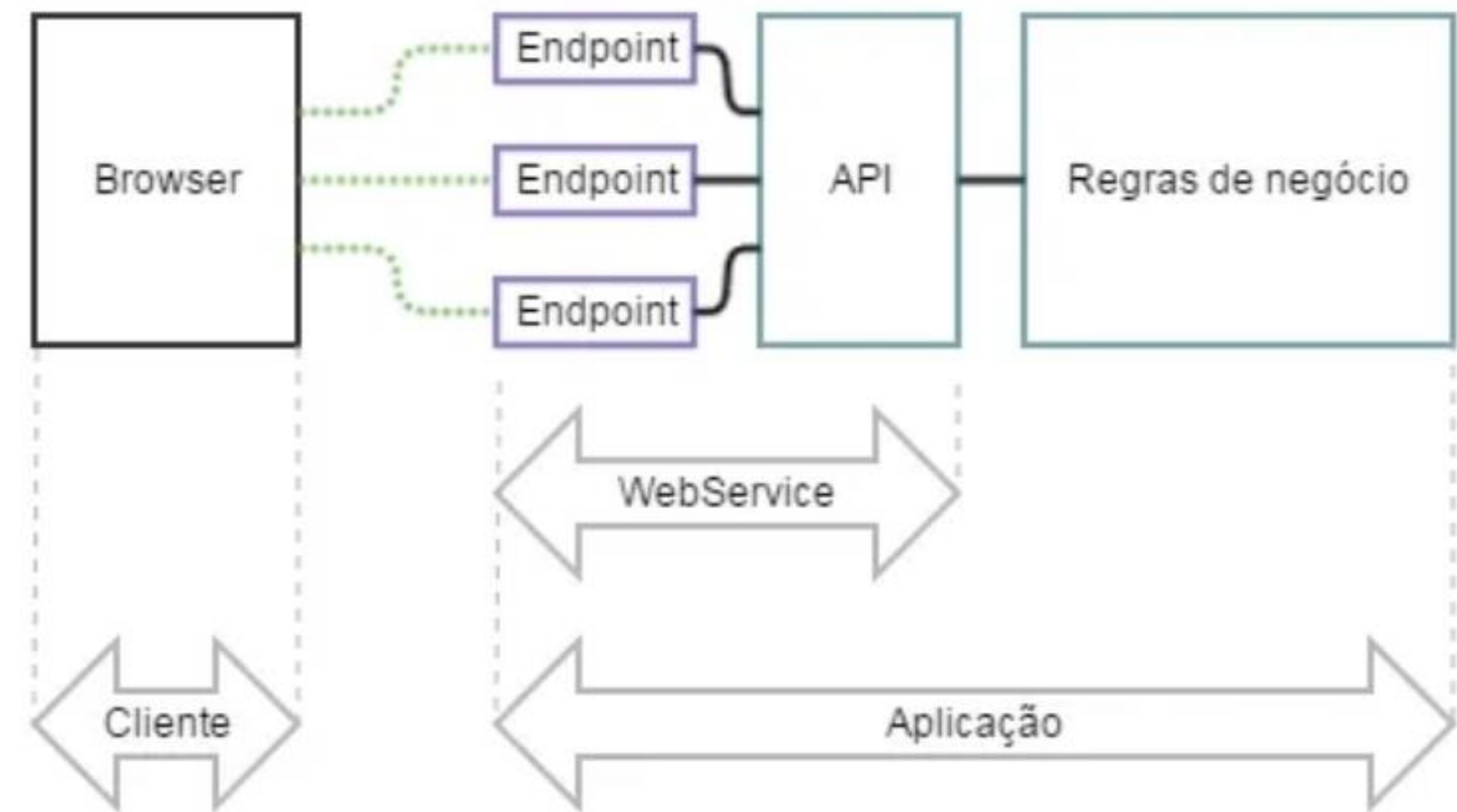
- **Stateless** – Cada requisição é completamente independente, não vai ser armazenado nenhuma informação dela.
- Utilização dos métodos **GET, POST, PUT, DELETE**
- **Sintaxe Universal** para identificar os recursos, as **URLs (Uniform Resource Identifier)**.
- Deve ter os **mime-type** (content-types - tipos de conteúdo), para solicitar e retornar o conteúdo, através dele o cliente escolhe o formato que a aplicação deverá retornar o conteúdo. Como o JSON, XML, HTML, PDF, TXT, etc..
- Código de retorno do protocolo HTTP, que é retornado sempre que uma requisição HTTP é feita.
- <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status>
- <https://http.cat/>
- <https://http.dog/>

**404**  
Page not found



# O que é Endpoints?

- Como próprio nome diz é a extremidade de um canal de comunicação.
- URL de um servidor ou de um serviço. Ele é o que o serviço expõe.
- Características ABC:
  - ADDRESS - onde o serviço está hospedado.
  - BINDING - como ele pode ser acessado.
  - CONTRACT - o que pode ser visto no serviço.



# O que é Endpoints?

## Boas práticas

- Exemplo de EndPoint: <https://abccompany.com/products/>

### 1º - De preferencia ao uso do plural quando disponibilizar uma rota;

- Ex:meuapp/api/produto **por:** meuapp/api/produtos

### 2º Utilize o padrão kebab-case;

- Ex:meuapp/api/globalConfiguration/ **por:** meuapp/api/global-configuration

### 3º Utilize as URL's em minúsculo;

- Ex:meuapp/api/Produtos **por:** meuapp/api/produtos

# O que é Endpoints?

## Boas práticas

### 4º Evite expor o id do Banco;

- Ex:meuapp/api/usuarios/2 **por:** meuapp/api/usuarios/felipe

### 5º Não sobrescrever o sentido original do HTTP:

- Ex: Get meuapp/usuarios/buscar-novos-clientes **por:** Get meuapp/usuarios/novos-clientes
- POSTmeuapp/usuarios/felipe/transacao/pagar **ou** PUTmeuapp/usuarios/felipe/status

# ***SPRING***





# ***SPRING***



<https://spring.io/>

A large, stylized pink star with five points, positioned on the left side of the slide. The star is partially cut off by the left edge.

***Agradecemos a sua atenção!***