

INSTITUTO MAUÁ DE TECNOLOGIA



Linguagens 2

Back-end

Prof. Tiago Sanches da Silva

CRUD

CRUD – (Create, Read, Update e Delete)

O CRUD vem do inglês, das palavras (Create, Read, Update, Delete) que quer dizer basicamente as 4 (quatro) principais operações com um banco de dados (inserir, ler, atualizar, excluir).

Quando pensamos em qualquer sistema dinâmico, a primeira coisa que vem na nossa cabeça é o CRUD, sem ele nada funciona, pois é com ele que realizamos as interações com o banco de dados.


Introdução a web service

Introdução a web service

É necessário uma padronização que garanta **a comunicação** entre o provedor de serviço e o sistema que realizou a requisição.

Essa comunicação, por sua vez, segue alguns protocolos, principalmente relacionados **ao formato da transmissão de dados**, o que permite que, uma vez que um sistema implemente essas regras, qualquer outro sistema que siga o mesmo protocolo seja capaz de se comunicar com ele.

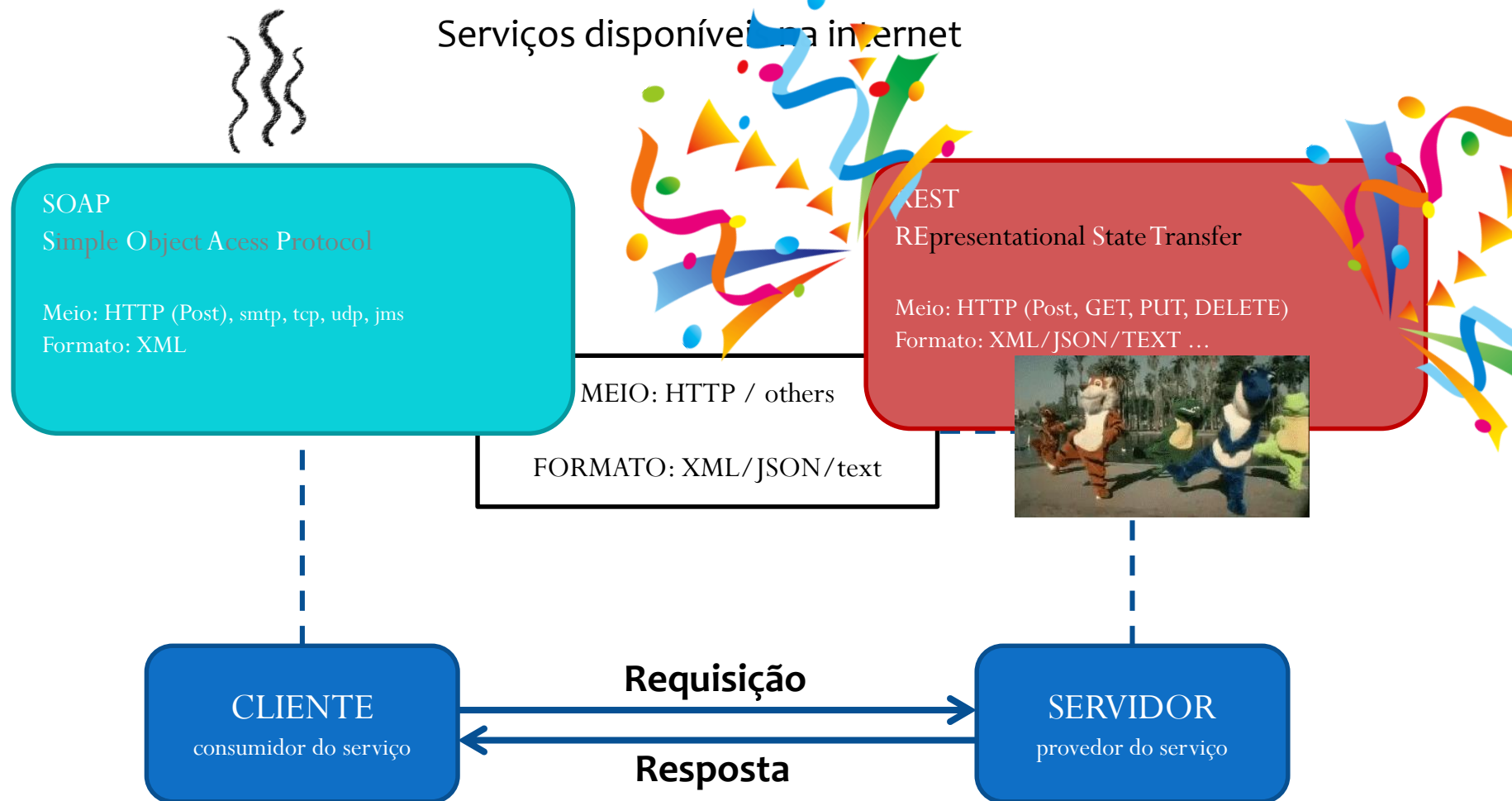
Entre essas padronizações estipuladas, as duas de maior destaque são o protocolo **SOAP** e o modelo de arquitetura **REST**, as quais iremos discutir nos próximos tópicos.



Tipos de Web Service

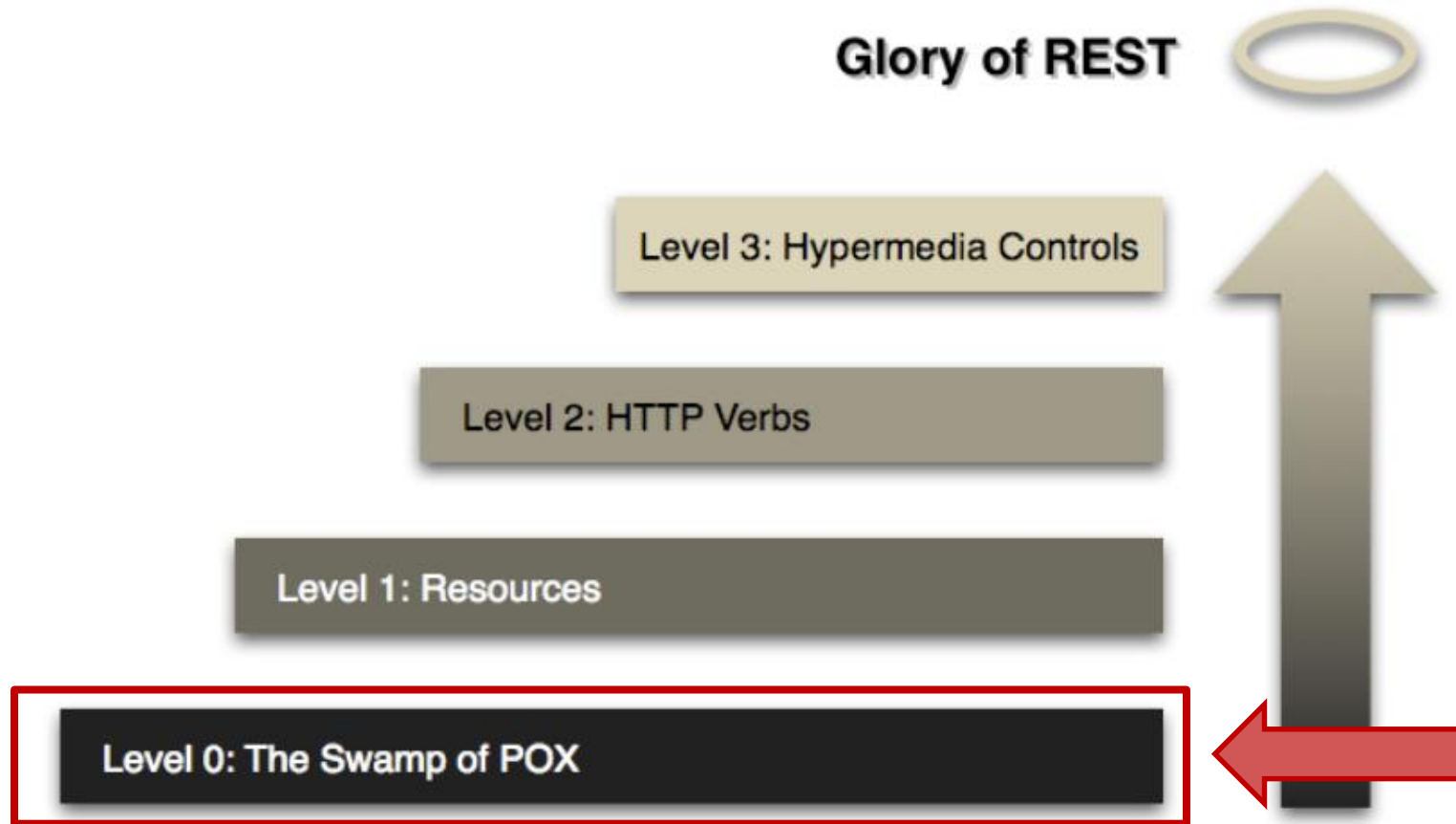
WEB SERVICES

Serviços disponíveis na internet



Maturidade de Richardson

Modelo de maturidade de Richardson



Evoluiremos durante as aulas.

Voltando ao CRUD

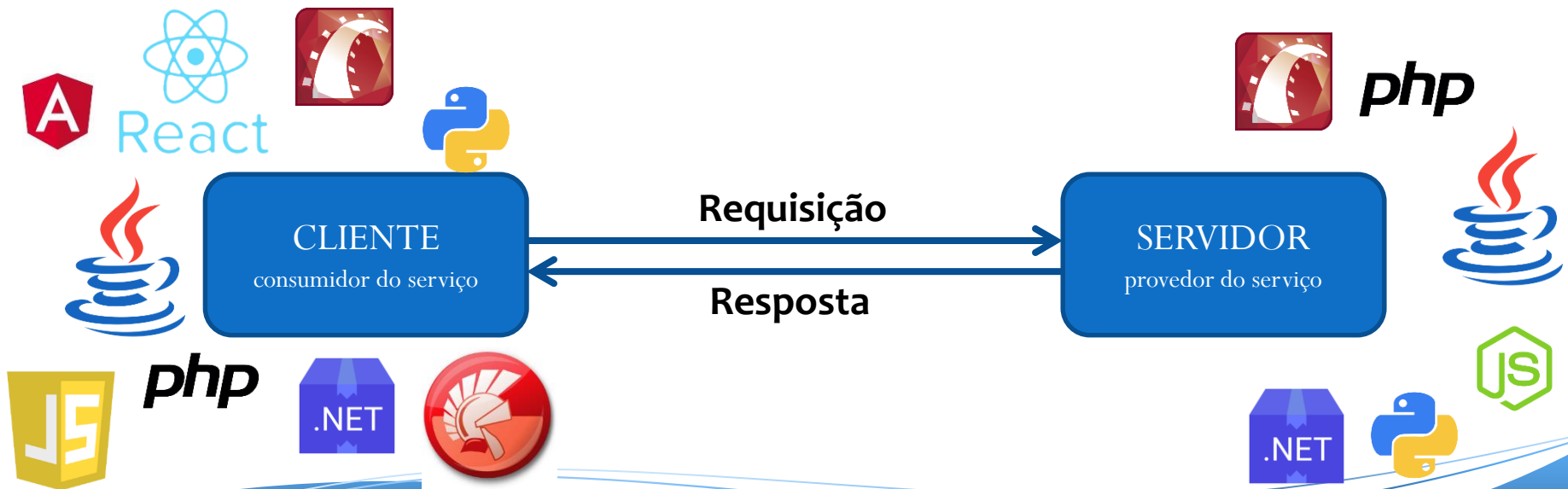
Voltando ao CRUD

Vamos construir e documentar nosso CRUD para que outras aplicações possam utilizar. E aos poucos vamos chegando mais próximos de um Web Service Restful, que será apresentado em aulas futuras.

Informações relevantes

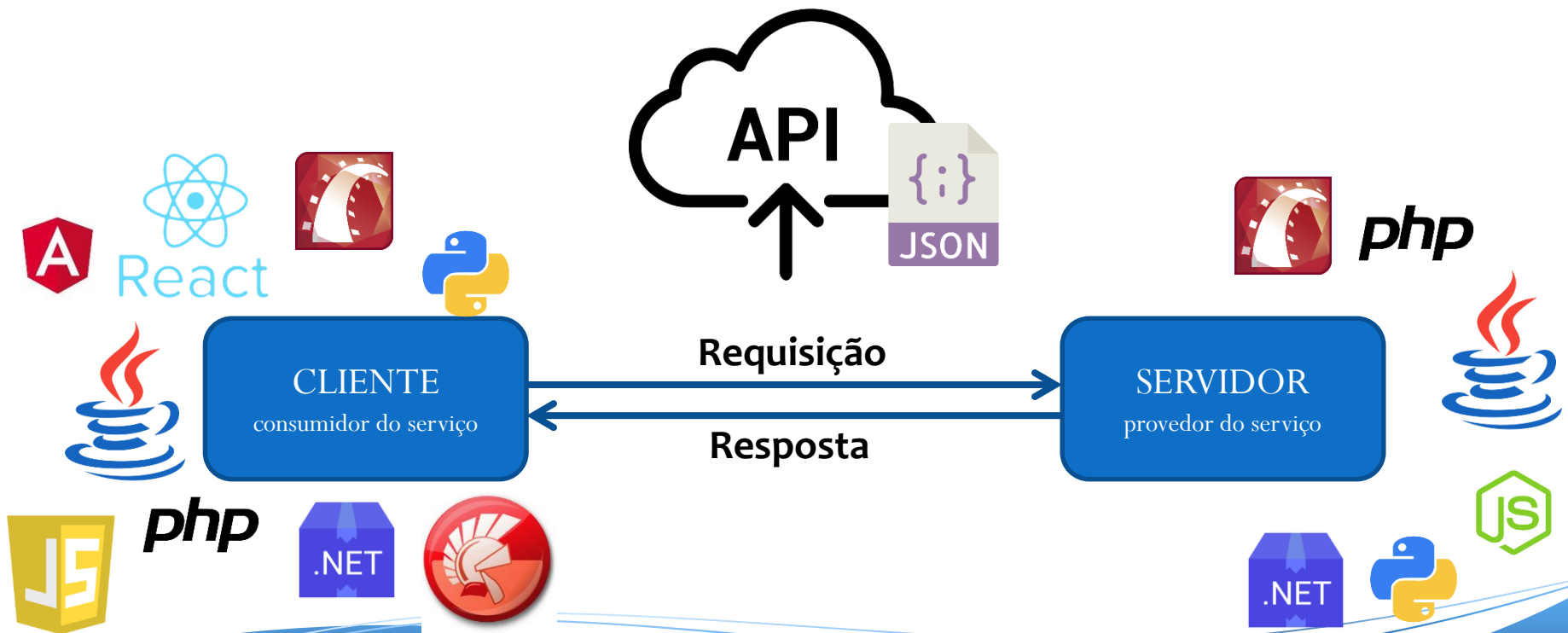
Em que linguagem o serviço web deve ser implementado?
Qualquer uma!

E quais linguagens podem implementar um SW que irá consumir o serviço?
Qualquer uma também, obvio!



Informações relevantes

Como eles irão conversar entre si?



Informações relevantes

Usaremos JSON para comunicação entre cliente servidor.



Para todo serviço devemos especificar:

- Verbo HTTP (GET, POST, UPDATE, DELETE)
- A URL () do serviço
- Informações de entrada (via parâmetros ou JSON no corpo)
- Qual as respostas esperadas

Exemplo – Create - Insert

Inserir um cliente no sistema

POST

http://domínio/cliente

Entrada:

```
{  
    "nome": "Ramon Candido",  
    "idade": 52,  
    "cpf": "123.456.789-0"  
}
```

Resposta:

Sucesso:

Status Code: 200 OK (Trocaremos para 201 created depois)
Sem corpo!

Falha:

Status Code: 40X
Corpo opcional:
{
 "erro": "Json incorreto"
}

Exemplo – Read - Select

Recuperar um cliente no sistema

GET

`http://domínio/cliente?id=12`

Resposta:

Sucesso:

Status Code: 200

```
{  
    "nome": "Ronalzinho",  
    "idade": 22,  
    "cpf": "123.456.789-0"  
}
```

Falha:

Status Code: 40X

Corpo opcional:

```
{  
    "erro": "Json incorreto"  
}
```

Pratica!

Vamos a parte prática.

Perguntas?