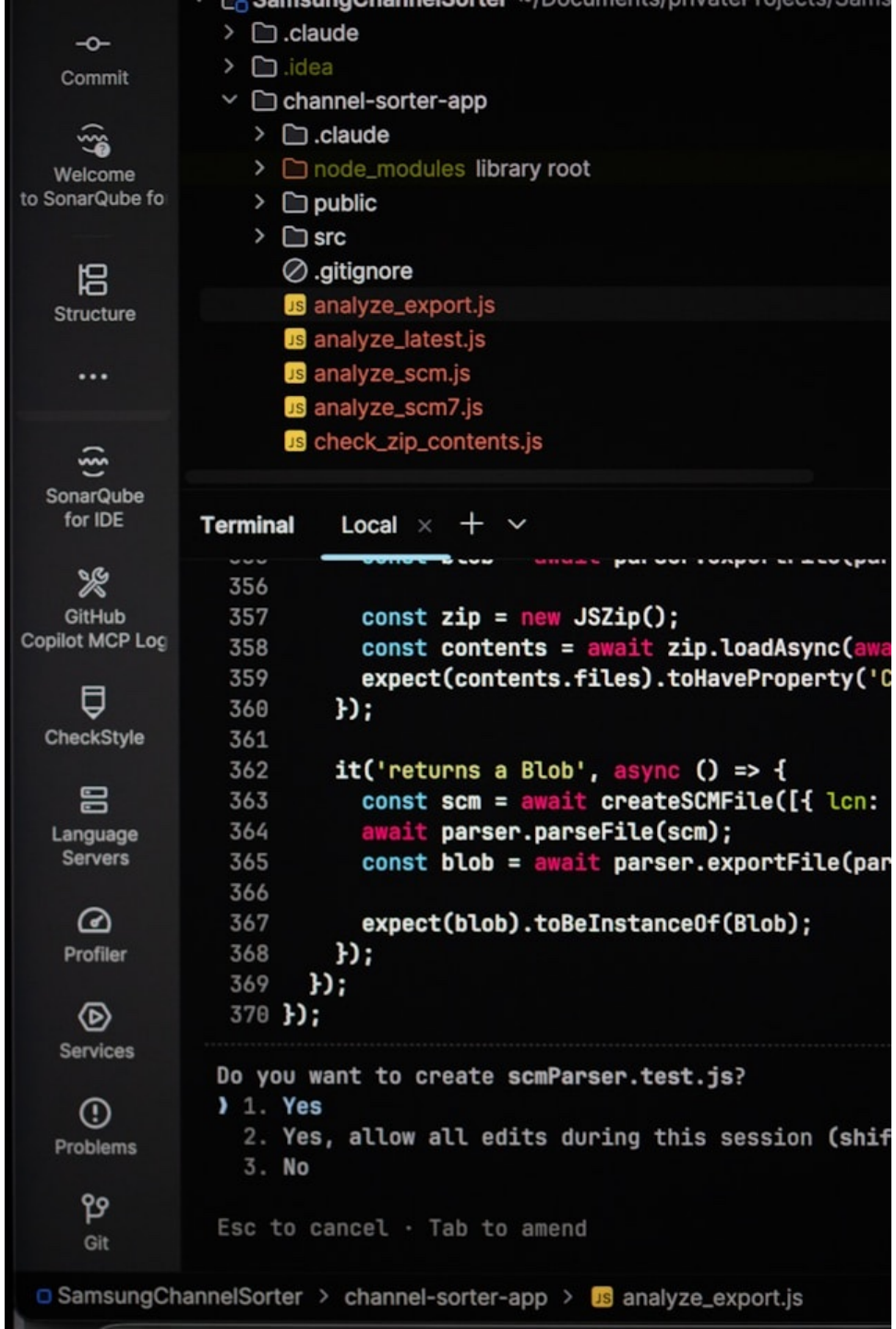


5954025 - Sistemas Distribuídos

Aula 09a - Web Services: SOAP

Prof. Dr. Denis M. L. Martins
DCM | FFCLRP | USP

Objetivos de Aprendizagem

- Compreender o conceito de **Arquitetura Orientada a Serviços (SOA)**.
- Explicar o papel de **serviços** na integração entre sistemas.
- Diferenciar **SOAP Web Services** de outras abordagens de comunicação distribuída.
- Entender a estrutura de uma mensagem **SOAP**.
- Identificar o papel de **WSDL** e **XML** em serviços SOAP.
- Reconhecer vantagens, limitações e casos de uso de SOAP em ambientes corporativos.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Arquitetura Orientada a Serviços (SOA)

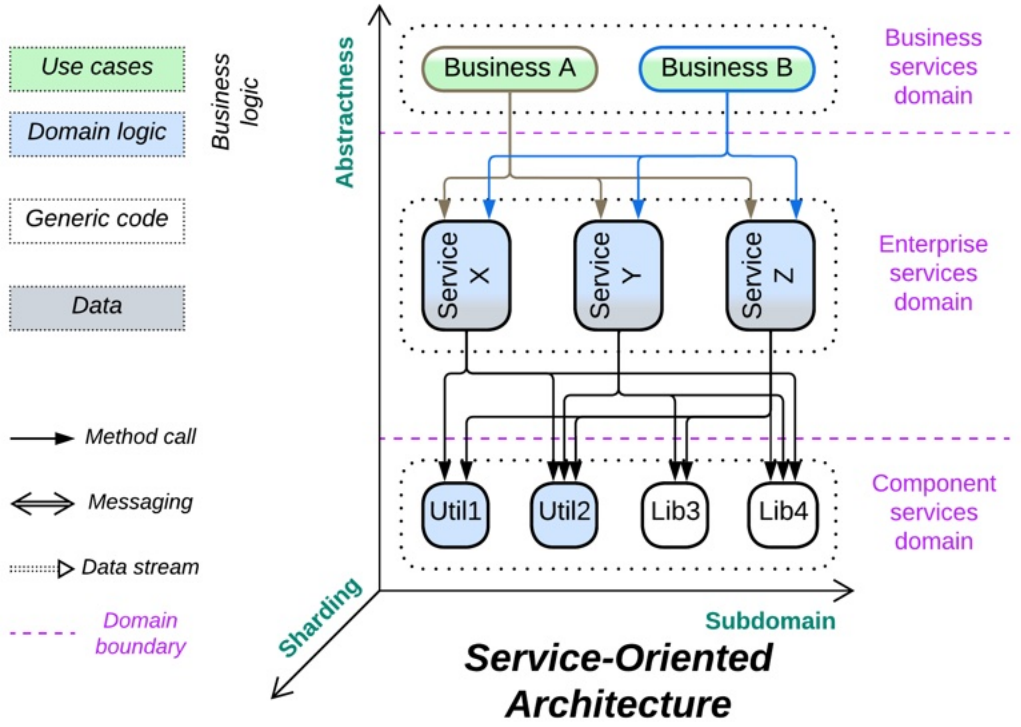
SOA (Service-Oriented Architecture) é um estilo arquitetural em que funcionalidades de negócio são disponibilizadas como **serviços**.

- Um serviço executa uma função específica.
- Esse serviço pode ser reutilizado por diferentes aplicações.
- A comunicação ocorre por interfaces bem definidas.
- O foco está em **interoperabilidade**, **reuso** e **integração**.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Ideia Central da SOA

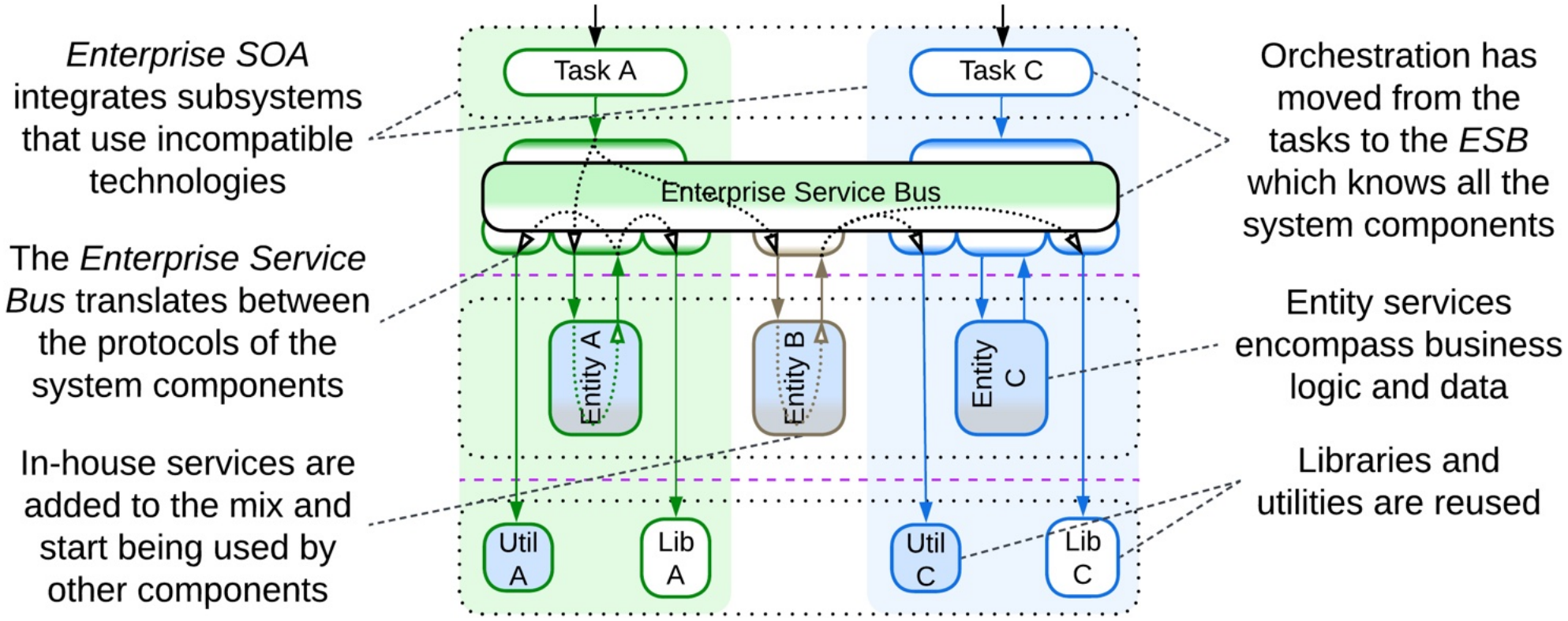
- Em vez de cada sistema implementar tudo sozinho, funcionalidades podem ser expostas como serviços reutilizáveis.
- Isso reduz retrabalho e facilita integração entre aplicações.
- A lógica de negócio pode ficar centralizada em serviços especializados.
- Diferentes sistemas podem consumir o mesmo serviço.



> Fonte da Imagem: Metapatterns.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

SOA em detalhes



> Na imagem: Orchestração é implementada no Enterprise Service Bus (ESB), o que pode se tornar um gargalo.
>
> Fonte: Metapatterns.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

O que é um serviço?

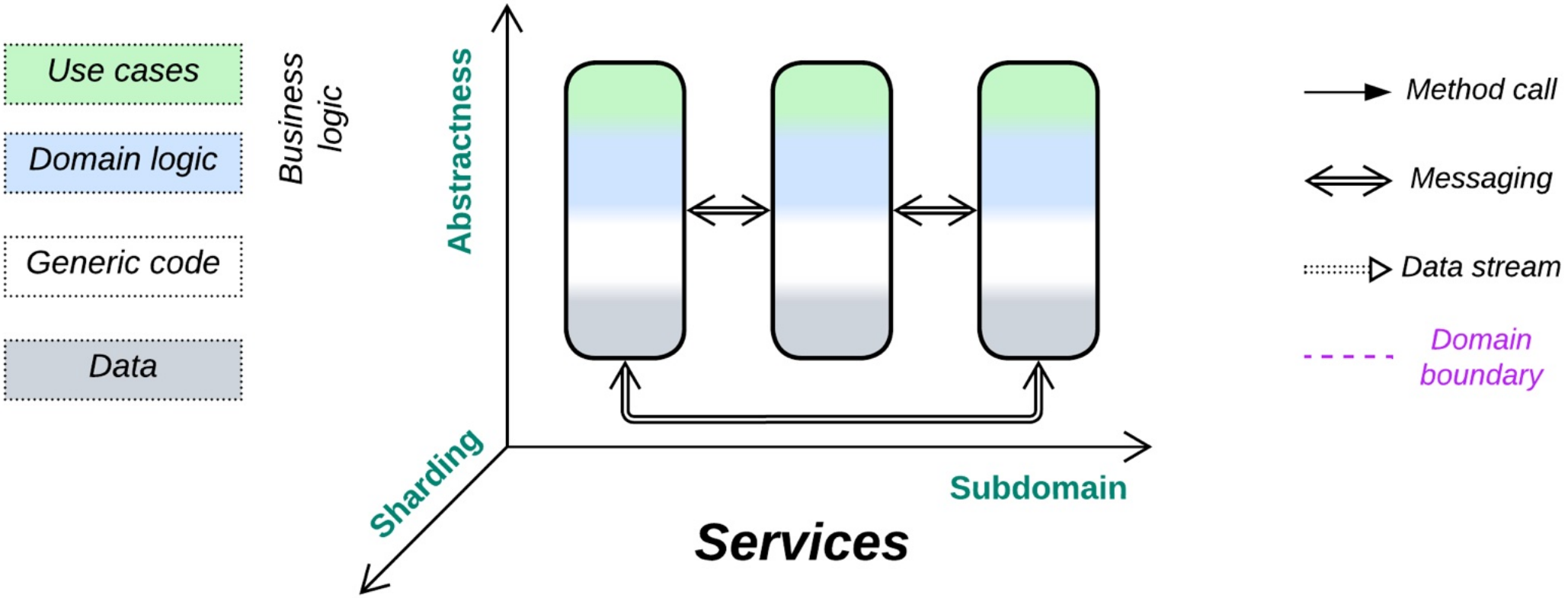
Um **serviço** é uma funcionalidade disponibilizada para uso por outras aplicações.

Características:

- Possui responsabilidade bem definida.
- Pode ser acessado remotamente.
- Expõe uma interface padronizada.
- Esconde detalhes internos de implementação.
- Pode ser reutilizado em diferentes contextos.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

O que é um serviço?



Na imagem: Estrutura geral de serviços. Fonte: Metapatterns.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Exemplos de Serviços

- Consultar saldo bancário.
- Emitir nota fiscal.
- Calcular frete.
- Validar CPF.
- Registrar pedido.
- Consultar dados de um aluno.
- Atualizar status de uma entrega.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

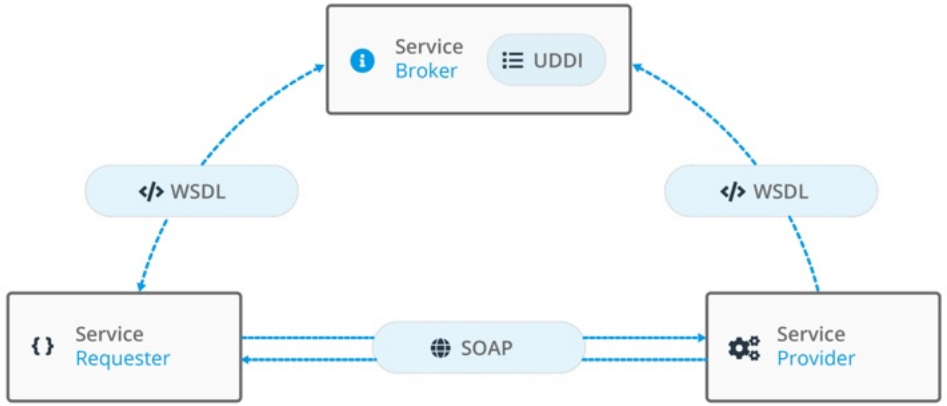
Estrutura conceitual da SOA

- **Service Provider**: quem oferece o serviço.
- **Service Consumer**: quem utiliza o serviço.
- **Service Contract**: definição da interface e das regras.
- **Service Registry**: local onde serviços podem ser publicados e encontrados.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Fluxo básico em SOA

1. Um sistema publica um serviço.
2. Outro sistema descobre ou já conhece esse serviço.
3. O consumidor envia uma requisição.
4. O provedor processa a operação.
5. O provedor devolve a resposta.



> Fonte da imagem: Codeless Platforms

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

O que são Web Services

Definição: Mecanismos que permitem comunicação entre diferentes aplicações em redes, independentemente de linguagem ou plataforma.

- **Objetivo Principal:** Garantir a **interoperabilidade**. Sistemas de diferentes linguagens/plataformas conversam.
- **Como Funciona:** Expondo funcionalidades como serviços (APIs).
- **Exemplo Simples:** Um aplicativo móvel (iOS) precisa consultar o saldo bancário em um servidor legado (Java).
- Usam padrões como XML, HTTP, SOAP e WSDL.

> Note que nem todo serviço em SOA precisa ser um Web Service, mas Web Services foram amplamente utilizados para implementar SOA.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

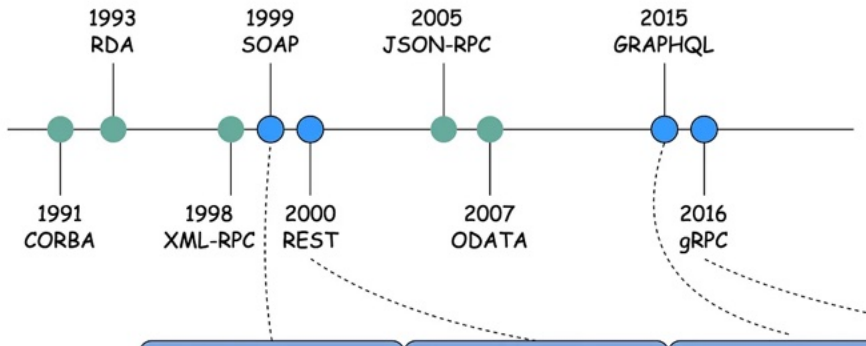
Evolução Tecnológica

SOAP foi muito utilizado em sistemas corporativos e integrações empresariais.

> SOAP não morreu! Ele ainda é usado sistemas legados (empresariais, bancários, etc.). Fonte da imagem: ByteByteGo.

API Architectural Styles Comparison

Source: altexsoft



	SOAP (Simple Object Access Protocol)	REST (REpresentational State Transfer)	GraphQL	RPC (Remote Procedure Call)
Organized in terms of	enveloped message structure	compliance with six architectural constraints	schema & type system	local procedure call
Format	XML only	XML, JSON, HTML, plain text	JSON	JSON, XML, Protobuf, Thrift, FlatBuffers
Learning curve	Difficult	Easy	Medium	Easy
Community	Small	Large	Growing	Large
Use cases	- payment gateways - identity management - CRM solutions - financial and telecommunication services - legacy system support	- public APIs - simple resource-driven apps	- mobile APIs - complex systems - micro-services	- command and action-oriented APIs - high performance communication in massive micro-services systems

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

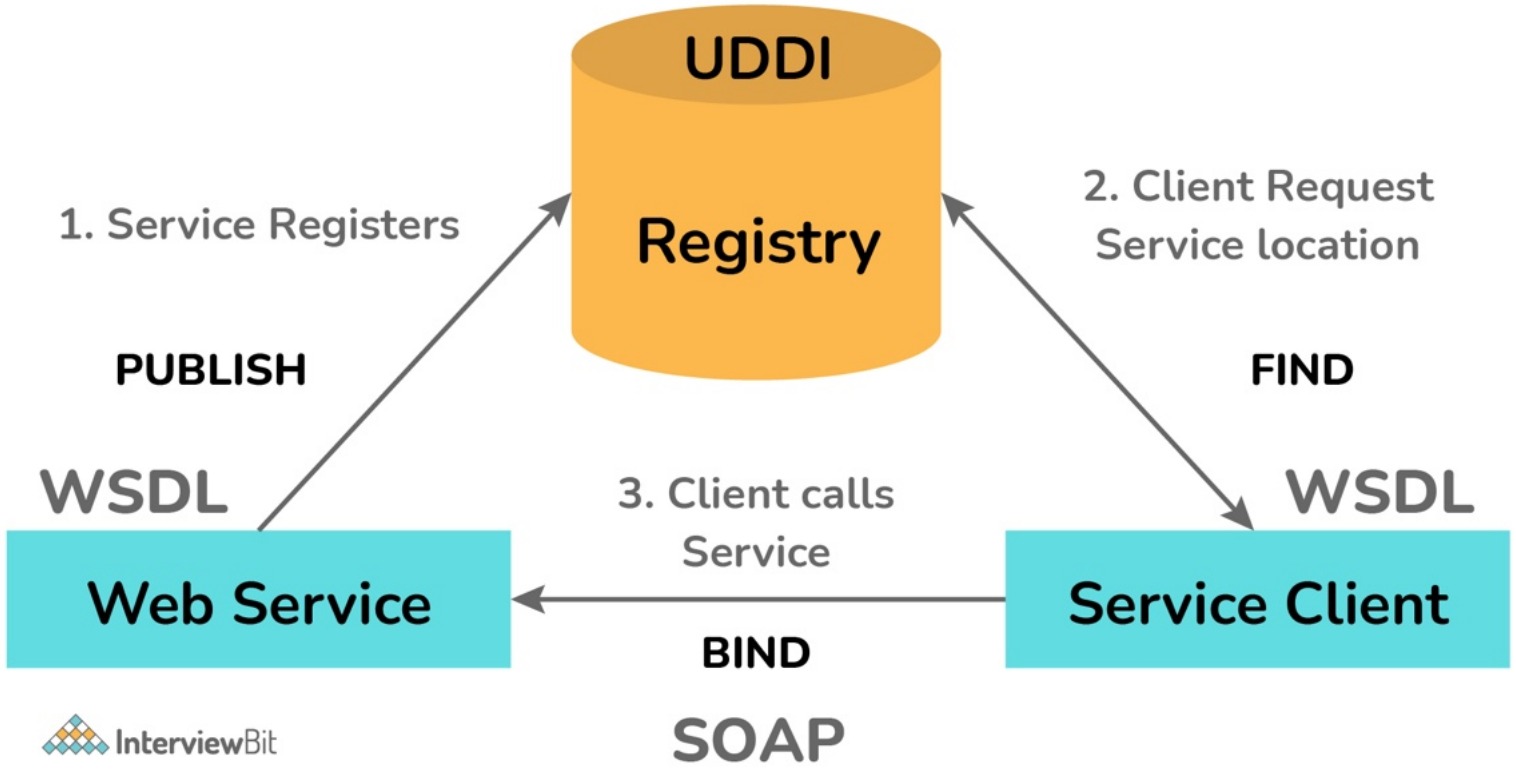
Protocolo SOAP (Simple Object Access Protocol)

- **O que é:** Um protocolo baseado em XML, usado para formatar mensagens trocadas entre serviços.
- **Base Tecnológica:** Utiliza o **XML** como formato de dados principal.
- **Estrutura Rígida:** Define um *envelope* (pacote) obrigatório para todas as mensagens.
- **Vantagem Chave:** Suporte nativo a padrões complexos (segurança, transações).
- **Exemplo Simples:** SOAP é como enviar uma carta registrada com formulário preenchido: o envelope garante que nada se perca ou seja mal interpretado.

> SOAP pode operar sobre HTTP, SMTP e outros protocolos. É fortemente orientado a contrato.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

SOAP: Visão Geral



> Na imagem: Visão geral de comunicação via SOAP. Fonte: Interviewbit.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Por que SOAP foi importante?

SOAP ganhou relevância porque oferecia:

- Padronização formal para integração.
- Compatibilidade entre plataformas distintas.
- Forte definição contratual.
- Recursos avançados para segurança, confiabilidade e transações.

Cenários típicos de uso do SOAP

- Sistemas bancários.
- Integração com ERPs.
- Serviços corporativos críticos.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Componentes do SOAP

- **1. XML Envelope:** O wrapper obrigatório da mensagem (Header, Body).
  - **Header**: Metadados (quem enviou, segurança, etc.).
  - **Body**: A informação real que está sendo transmitida.
  - **Envelope**: O container principal de tudo.
- **2. WSDL (Web Services Description Language):** É o "contrato" do serviço.
  - Descreve *o quê* o serviço faz, *como* chamá-lo e *quais* tipos de dados espera.
  - É a documentação formal que o cliente usa para gerar código.
- **3. Transporte:** Geralmente via HTTP (ou JMS).

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

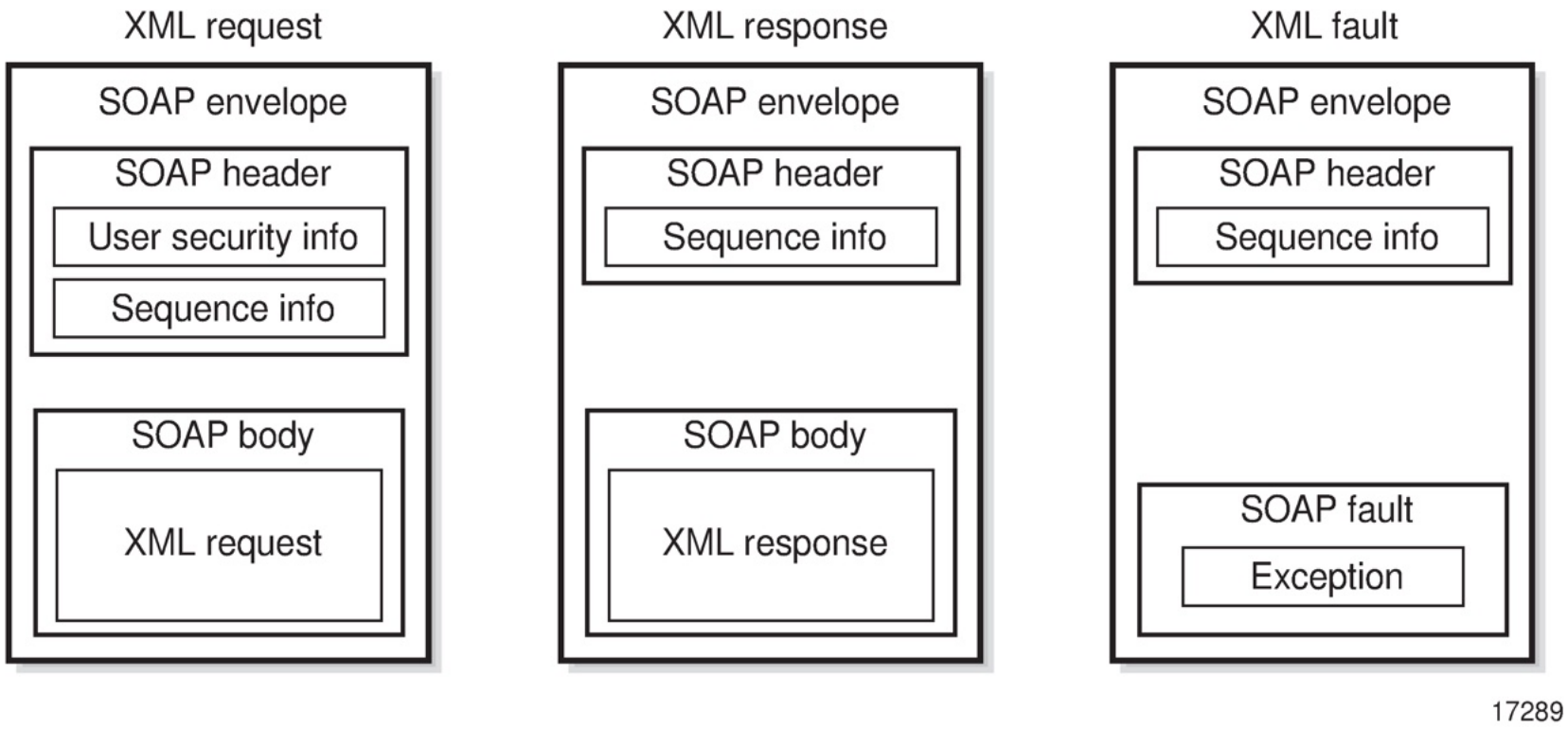
Estrutura de uma Mensagem SOAP

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
        <!-- Metadados (ex: Autenticação, Logs) -->
    </soap:Header>
    <soap:Body>
        <!-- Dados da operação -->
    </soap:Body>
</soap:Envelope>
```

- **Envelope:** A "carta" que envolve a mensagem.
- **Header:** Informações opcionais (segurança, transação).
- **Body:** Onde os dados reais da chamada residem.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Estrutura de uma Mensagem SOAP



> Na imagem: Diferentes tipos de mensagem SOAP. Fonte: Nokia

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Requisição SOAP

Cenário: Transferência bancária.

```
<soap:Body>
    <TransferirDinheiro xmlns="http://servicos.banco.com">
        <contaOrigem>1234-5</contaOrigem>
        <contaDestino>6789-0</contaDestino>
        <valor>100.00</valor>
    </TransferirDinheiro>
</soap:Body>
```

> O namespace (**xmlns**) define o contrato da operação.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Resposta SOAP (Sucesso)

```
<soap:Envelope>
    <soap:Header/>
    <soap:Body>
        <TransferirDinheiroResponse xmlns="...">
            <status>Sucesso</status>
            <transacaoId>TX-998877</transacaoId>
        </TransferirDinheiroResponse>
    </soap:Body>
</soap:Envelope>
```


Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Resposta SOAP (Erro)

```
<soap:Body>
    <soap:Fault>
        <faultstring>Saldo insuficiente</faultstring>
        <faultcode>INVALID_AMOUNT</faultcode>
    </soap:Fault>
</soap:Body>
```

- **Fault Element:** Se algo der errado, o SOAP retorna um elemento **<soap:Fault>**.
- **Conteúdo do Erro:** Código de erro, mensagem humana e detalhes técnicos.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

WSDL

Web Services Description Language (WSDL) é uma linguagem baseada em XML usada para descrever um serviço web. Ela informa:

- quais operações o serviço oferece;
- quais parâmetros cada operação recebe;
- quais tipos de dados são usados;
- quais respostas podem ser retornadas;
- em qual endereço o serviço está disponível;
- qual protocolo deve ser usado.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

WSDL (cont.)

O WSDL funciona como um contrato técnico entre o provedor e o consumidor do serviço.

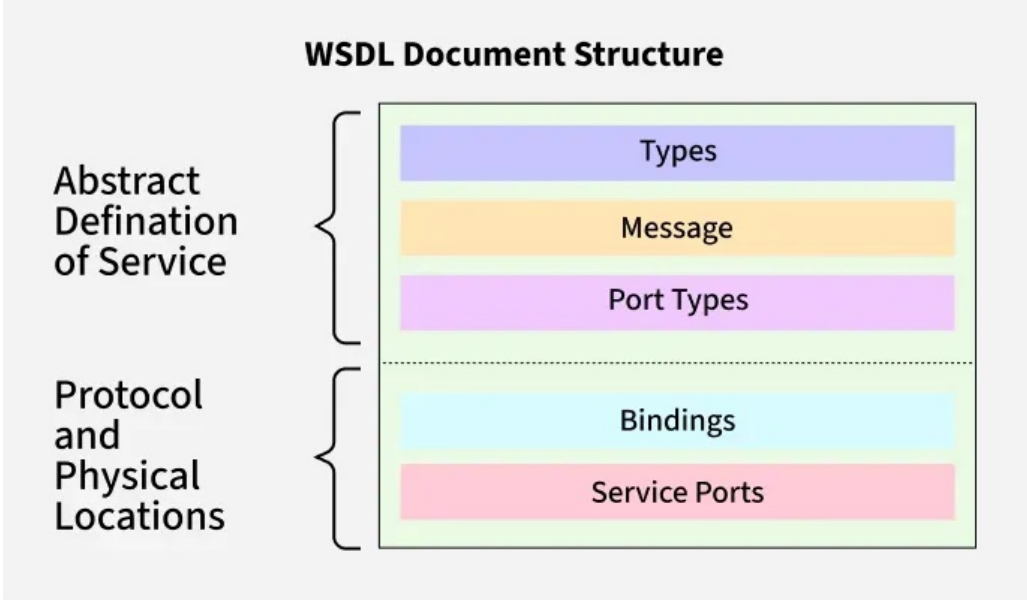
- Cliente não precisa adivinhar como usar o serviço;
- Regras de comunicação ficam documentadas;
- Diferentes linguagens e plataformas conseguem interoperar;
- Ferramentas podem gerar clientes automaticamente.

> WSDL formaliza o contrato do serviço e reduz ambiguidades na integração.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

WSDL (cont.)

- **Types**: quais dados existem.
- **Message**: quais mensagens serão trocadas.
- **PortType**: quais operações existem.
- **Binding**: como acessar as operações.
- **Service**: onde o serviço está disponível.



> Estrutura geral de um documento em WDSL.
>
> Fonte: GeekForGeeks

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

WSDL: Exemplo

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

> Fonte: https://www.w3schools.com/XML/xml_wsdl.asp

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

UDDI

Universal Description, Discovery and Integration (UDDI) é um padrão para publicação e descoberta de serviços.

Sua função principal é permitir que empresas e sistemas:

- publiquem seus serviços;
- encontrem serviços disponíveis;
- descubram informações sobre provedores;
- reutilizem serviços já existentes.

> Se o WSDL é o manual técnico do serviço, o UDDI funciona como uma espécie de catálogo ou lista telefônica de serviços.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

UDDI (cont.)

Um registro UDDI pode conter:

- dados sobre a empresa provedora;
- descrição do serviço;
- categorias e classificações;
- informações técnicas de acesso;
- referência ao WSDL do serviço.

> O UDDI não substitui o WSDL. Ele ajuda a localizá-lo.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Vantagens e Desvantagens do SOAP

Vantagens:

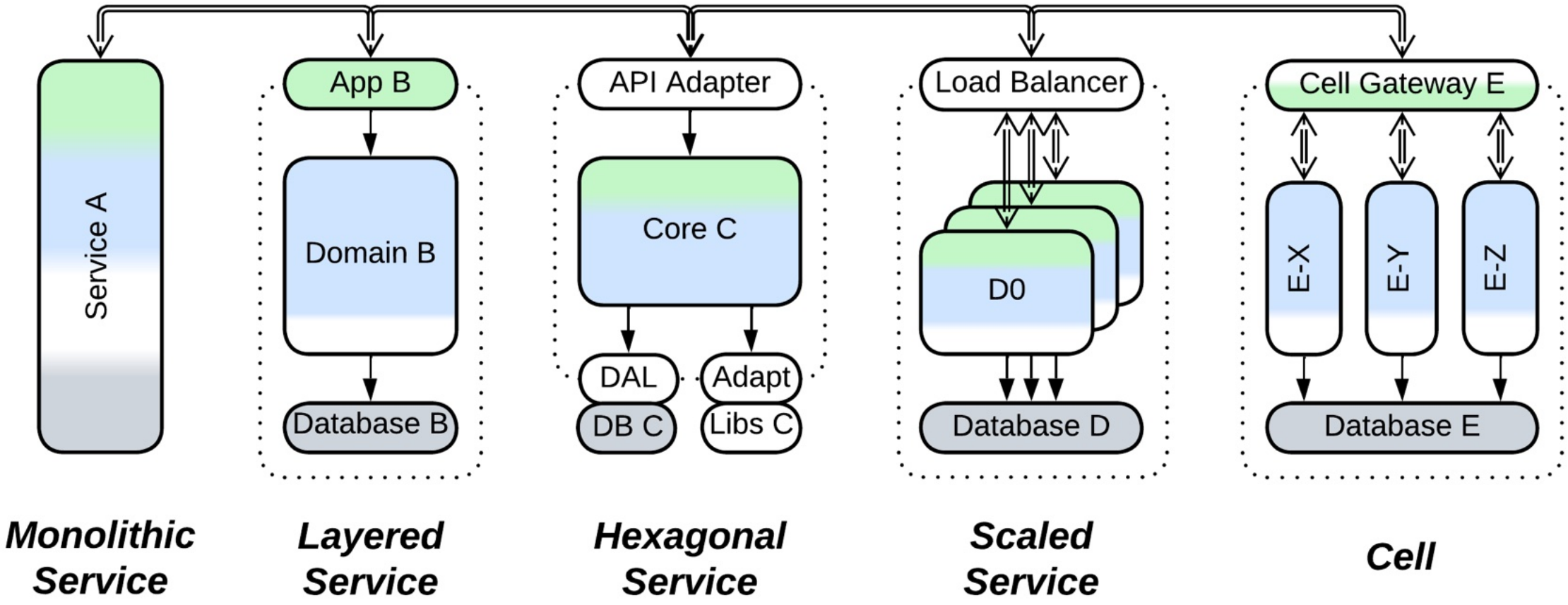
- **Segurança Nativa:** Suporte nativo para WS-Security (assinatura criptográfica).
- **Contrato Rígido:** O WSDL garante que o cliente e servidor falem a mesma "língua".

Desvantagens:

- **Peso:** XML é verboso (muito texto para poucos dados).
- **Performance:** Parsers de XML são mais lentos que JSON.
- **Complexidade:** Configuração inicial e depuração são mais difíceis.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Padrões arquiteturais de Serviço



> Na imagem: Variantes da estrutura interna de serviços. Fonte: Metapatterns.

Conclusão e Próximos Passos

Resumo

- **Web Services:** Permitem comunicação entre sistemas distintos via rede.
- **SOAP:** Protocolo baseado em XML para troca de mensagens estruturadas.
- **Estrutura:** Envelope (cabeça), Header (metadados), Body (dados), Fault (erros).
- **WSDL:** Especificação do contrato do serviço SOAP.
- REST/JSON substituiu o SOAP em muitas APIs públicas devido à simplicidade.

Material Adicional:

- **Documentação Oficial:** W3C SOAP Specification
- SOAP API Guide

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Lab Prático

Vamos simular um serviço bancário simples que calcula o saldo de uma conta.

Siga o passo-a-passo em:
https://denmartins.github.io/labs/web-services-soap

> Créditos da imagem: https://unsplash.com/@safarslife

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Dúvidas e Discussão

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Exercício e Questões

Exercício Prático 1

Crie uma função **criar_requisicao_frete(cep, peso)** que gere uma mensagem SOAP para consultar o valor do frete de uma encomenda.

A operação deve se chamar:

```
<CalculateShipping>
```

E deve receber:

```
<cep>14000000</cep>
<peso>2.5</peso>
```


Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Exercício Prático 2

Implemente um pequeno programa em Python que simule o fluxo completo de uma consulta SOAP:

1. O usuário informa o ID de um aluno.
2. O programa cria uma requisição SOAP.
3. O programa simula o processamento no servidor.
4. O programa retorna uma resposta SOAP.
5. O programa interpreta a resposta e exibe o resultado final.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Questões para estudo

- Defina "Interoperabilidade" no contexto dos sistemas distribuídos. Por que os Web Services foram criados para resolver esse problema?
- Explique a diferença entre **Arquitetura Orientada a Serviços (SOA)** e **SOAP Web Services**. Em sua resposta, deixe claro por que SOA não pode ser considerada a mesma coisa que SOAP.
- Descreva o papel do WSDL em um fluxo de comunicação SOAP. Se você fosse um desenvolvedor cliente, como você usaria o WSDL antes de escrever qualquer linha de código de chamada ao serviço?

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Questões para estudo (cont.)

Um servidor SOAP retornou a seguinte resposta ao cliente:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header/>
    <soap:Body>
        <soap:Fault>
            <faultcode>INVALID_AMOUNT</faultcode>
            <faultstring>Valor da transferência excede limite diário</faultstring>
            <detail>
                <limiteDiario>5000.00</limiteDiario>
                <valorSolicitado>6000.00</valorSolicitado>
            </detail>
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

- (a) Identifique o tipo de erro retornado (ex: erro de validação, erro de sistema, erro de autenticação).
- (b) Qual elemento do SOAP contém a informação técnica detalhada sobre o erro?
- (c) Como um desenvolvedor pode usar essas informações para melhorar sua aplicação (cliente)? Cite dois exemplos.

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br