

Arquitetura de software

Professor: Eduardo Cizeski Meneghel

PADRÕES E ESTILOS DE ARQUITETURAS

PADRÕES E ESTILOS

- Padrões e estilos de arquitetura de software são abordagens e diretrizes para projetar a estrutura de um sistema de software.

ESTILOS ABORDADOS ANTERIORMENTE

- Monolítica vs distribuída;
- Cliente-servidor;
- Arquitetura em camadas;
- Arquitetura orientada a serviços (SOA);
- Arquitetura baseada em microsserviços.

ESTILOS ABORDADOS ANTERIORMENTE

- Arquitetura em pipes e filtros;
- Arquitetura orientada a eventos
- Arquitetura em estilo publicador-subscritor;
- Arquitetura estilo MVC.

ESTILO DE ARQUITETURA MVVM

PADRÕES E ESTILOS

- O objetivo do MVVM é prover uma separação de responsabilidades entre a view e sua lógica, contudo, isto agrega benefícios como aumento da testabilidade da aplicação.

PADRÕES E ESTILOS

Model

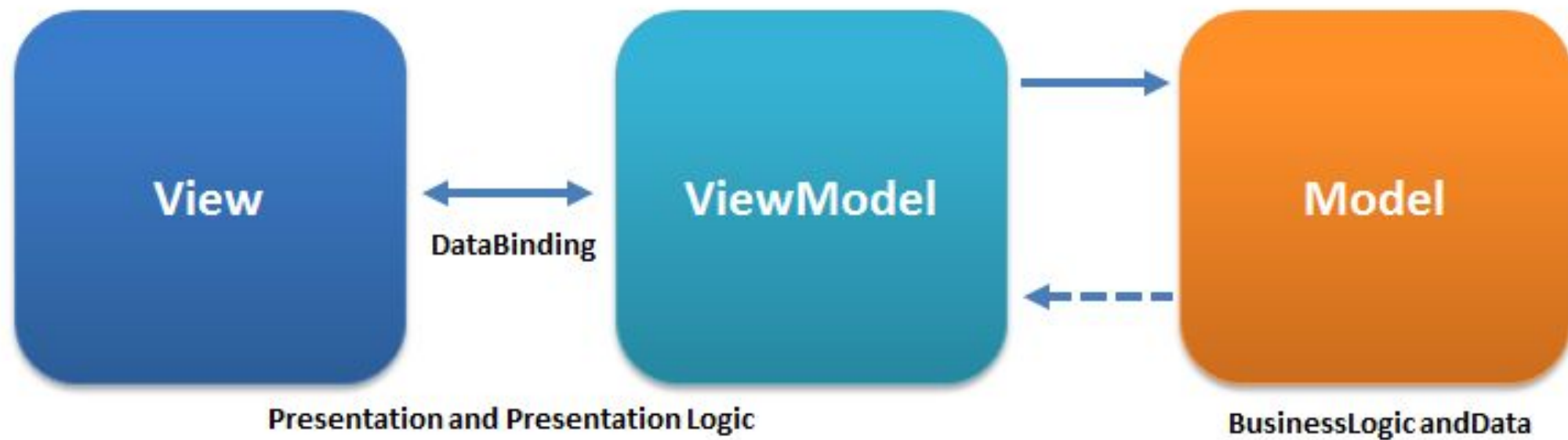
- O Model tem função similar ao do MVC, contendo implementação do modelo de domínio da aplicação que inclui o modelo de dados, regras de negócio e validações de lógica.

PADRÕES E ESTILOS

View

- Entidade responsável por definir a estrutura, layout e aparência do que será exibido na tela.

PADRÕES E ESTILOS



PADRÕES E ESTILOS

ViewModel

- Ela disponibiliza para a View uma lógica de apresentação e coordena as interações da View com a Model, além de poder implementar a lógica de validação para garantir a consistência dos dados.

PADRÕES E ESTILOS

Vantagens

- Permite simplificar testes de unidades;
- O desenvolvedor pode optar por reaproveitar um ViewModel em várias Views;
- O código para atualizar a View é pequeno, uma vez que as alterações podem ser propagadas automaticamente.

PADRÕES E ESTILOS

Desvantagens

- É uma estrutura com uma complexidade maior e para aplicações pequenas poderá adicionar camadas e regras desnecessárias.
- Aplicações que possuem vários pontos de alteração de um mesmo Model, podem acabar exibindo dados desatualizados.

PADRÕES E ESTILOS

- Tal modelo foi criado pelo arquiteto de software do WPF e Silverlight da Microsoft, **John Gossman** em 2005. Desde então, ele vem sendo usado principalmente no desenvolvimento mobile.
- O MVVM pode ser usado em Swift, Java, Dart (através do framework Flutter) e ainda os frameworks de JavaScript.

ESTILO DE ARQUITETURA MDA



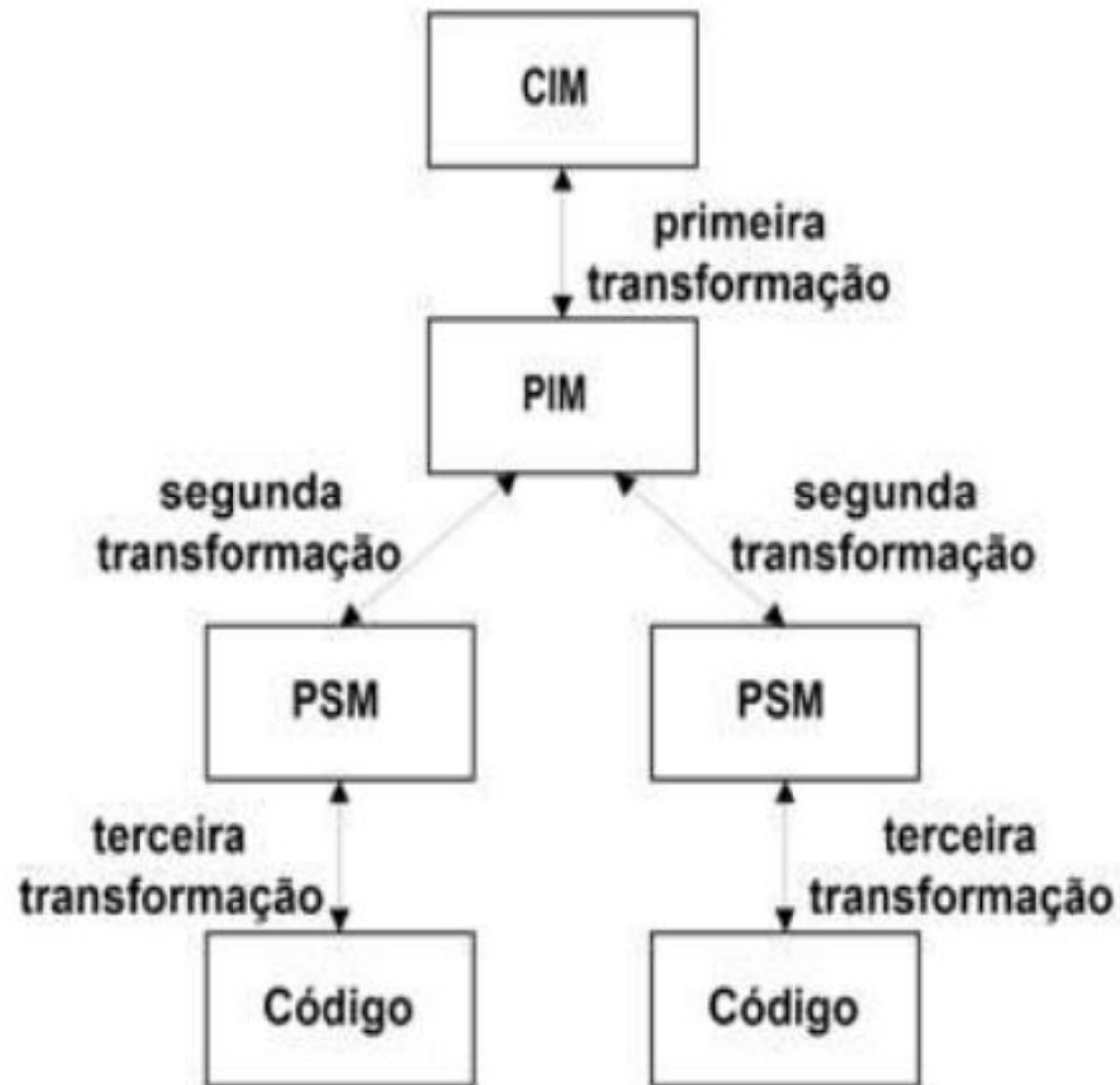
PADRÕES E ESTILOS

- MDA - Model Driven Architecture ou arquitetura orientada por modelos;
- Criado em 2001 pela OMG (Object Management Group).

PADRÕES E ESTILOS

- A MDA utiliza para construir seu ciclo de vida, os chamados modelos, que são todos os artefatos como especificação de requisitos, descrição da arquitetura, descrição do modelo e código.
- A MDA define frameworks que separam as especificações da funcionalidade do sistema de sua implementação em uma plataforma específica.

PADRÕES E ESTILOS



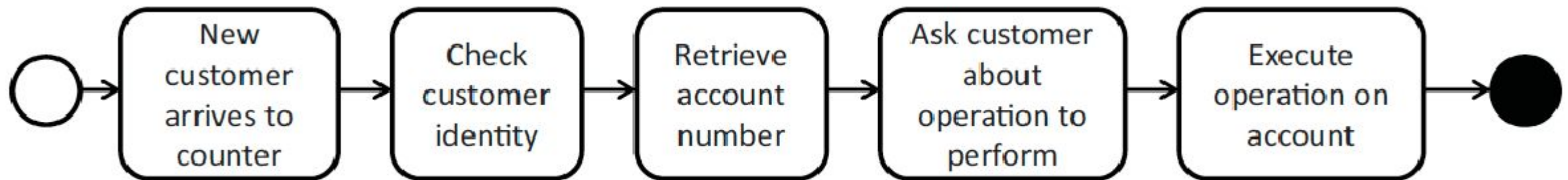
PADRÕES E ESTILOS

Modelo CIM (Modelo Independente de Computação)

- É onde se inicia a primeira fase, ou seja, o entendimento adequado do negócio, especializando-se nos processos e determinando o que necessariamente o sistema que irá automatizar o determinado processo deve fazer.

PADRÕES E ESTILOS

Modelo CIM (Modelo Independente de Computação)



PADRÕES E ESTILOS

Modelo PIM (Modelo independente de plataforma)

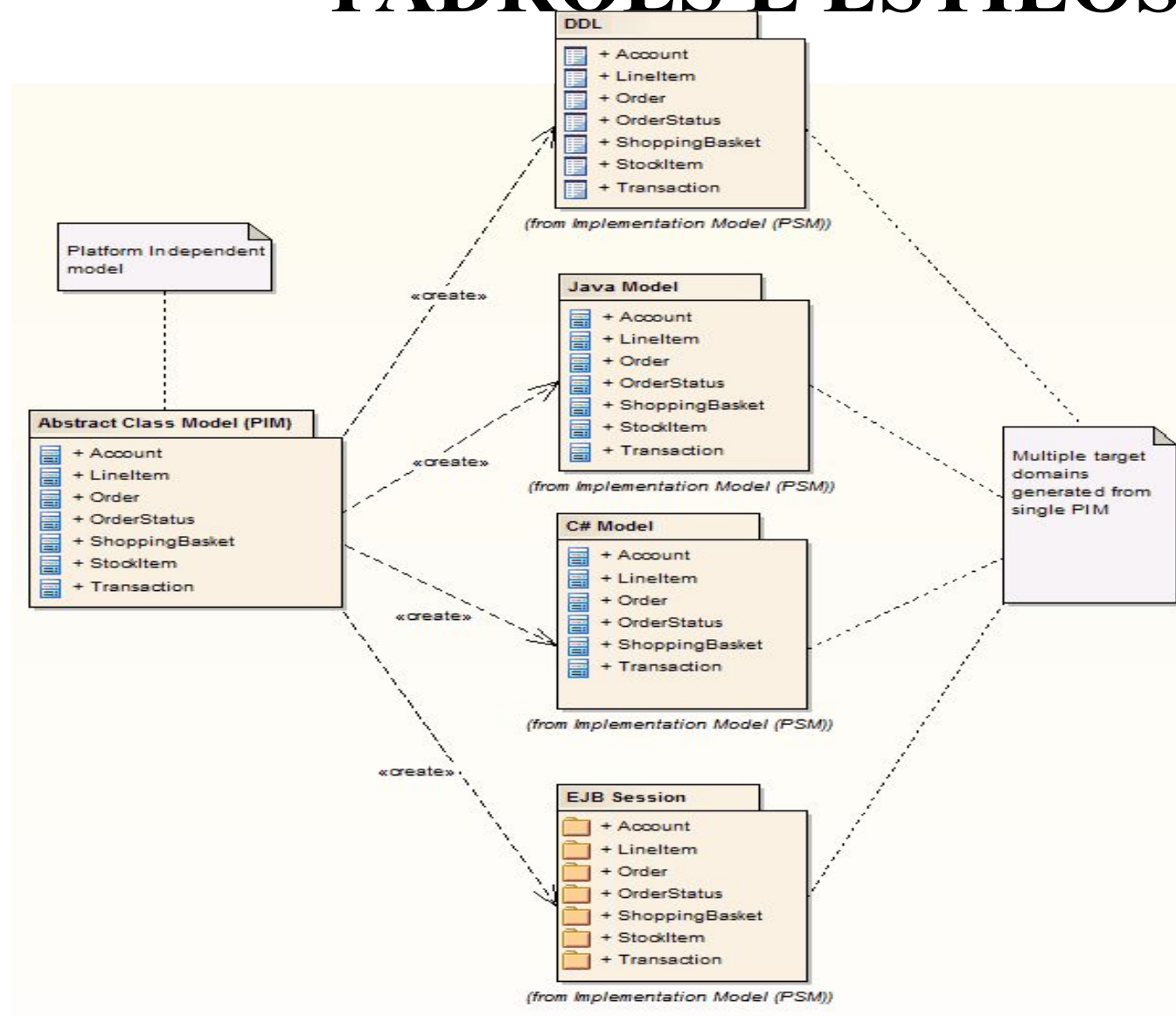
- O modelo PIM foca no ponto de visão de operações independente de plataforma.
- O PIM define um conjunto de componentes e funcionalidades, que são definidas independentemente de qualquer plataforma, ou seja, sem a necessidade de adicionar no modelo tecnologias, que serão utilizadas no projeto. Ele pode ser visto como uma abstração do sistema que pode ser realizado por diferentes plataformas específicas.

PADRÕES E ESTILOS

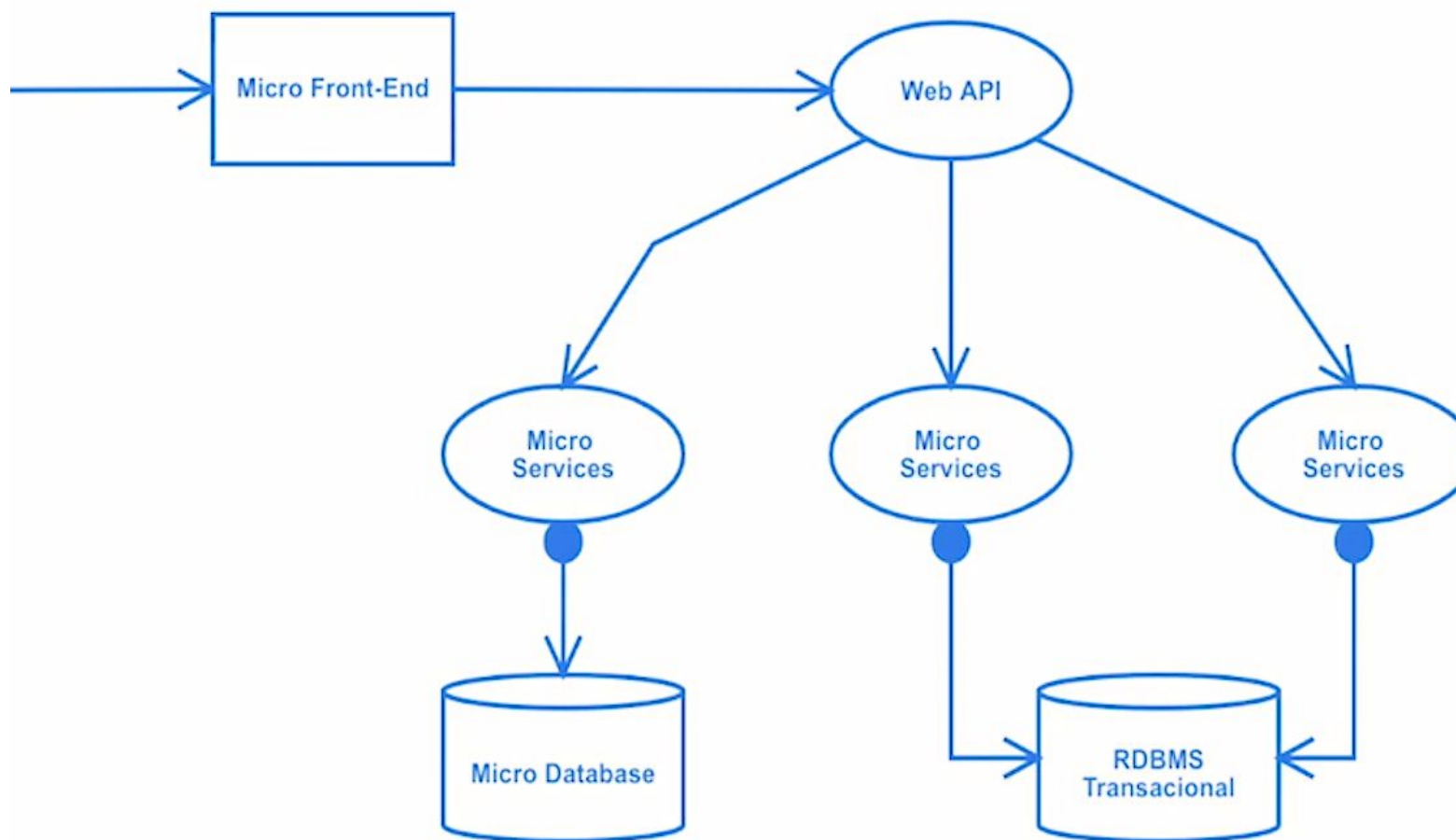
Modelo PSM (Modelos Específicos de Plataforma)

- O PSM é mais específico para o sistema em termos de tecnologia de implementação, como um modelo de banco de dados ou um modelo EJB (Enterprise JavaBeans).

PADRÕES E ESTILOS



PADRÕES E ESTILOS



PADRÕES E ESTILOS

- Produtividade: A **transformação** do PIM para o PSM precisa ser definida uma única vez e pode ser aplicada no desenvolvimento de diversos sistemas. Devido a este fato, tem-se uma redução no tempo de desenvolvimento.

PADRÕES E ESTILOS

- **Portabilidade:** Dentro da MDA a portabilidade é alcançada através do foco dado no desenvolvimento do PIM, que é independente de plataforma. Um mesmo PIM pode ser automaticamente transformado em vários PSMs de diferentes plataformas.

PADRÕES E ESTILOS

- Planejamento e detalhamento da aplicação;

PADRÕES E ESTILOS

- Planejamento e detalhamento da aplicação;

PADRÕES E ESTILOS

Opinião de Martin Fowler sobre MDA:

“Como exemplo, considere a lógica comportamental. Não consigo ver que desenhar diagramas de sequência ou diagramas de atividades seja tão bom, e muito menos melhor, do que escrever código em uma linguagem moderna.”

PADRÕES E ESTILOS

Opinião de Martin Fowler sobre MDA:

“Mesmo que a UML forme um ambiente de programação eficaz, ela ainda precisa se tornar popular. Como ex-Smalltalker, sei muito bem que mesmo as melhores línguas nem sempre se tornam populares.”

PADRÕES E ESTILOS

- Listas de exercícios - MVVM e MDA

PRINCÍPIOS DE INTEGRAÇÃO ENTRE SISTEMAS



INTEGRAÇÃO ENTRE SISTEMAS

Difícilmente um software conseguirá atender sozinho todas as demandas de seu contexto de operação.

INTEGRAÇÃO ENTRE SISTEMAS

Para cumprir, de fato, os objetivos do negócio, a arquitetura de um software deverá contemplar, além dele próprio, a relação com os demais “integrantes sistêmicos”, delimitando responsabilidades e relacionamentos.

INTEGRAÇÃO ENTRE SISTEMAS

- Integrações são aspectos críticos da arquitetura;

INTEGRAÇÃO ENTRE SISTEMAS

- As decisões arquiteturais que dizem respeito ao formato de comunicação entre sistemas tem impacto direto em atributos de qualidade como:
 - Segurança;
 - Resiliência;
 - Escalabilidade;
 - Desempenho.

INTEGRAÇÃO ENTRE SISTEMAS

- Forma de comunicação
 - Síncrona;
 - Assíncrona.

INTEGRAÇÃO ENTRE SISTEMAS

- Consistência
 - Atomicidade;
 - Eventual.

INTEGRAÇÃO ENTRE SISTEMAS

- Coordenação
 - Coreografia
 - Orquestração.

INTEGRAÇÃO ENTRE SISTEMAS

- Coordenação - Coreografia:

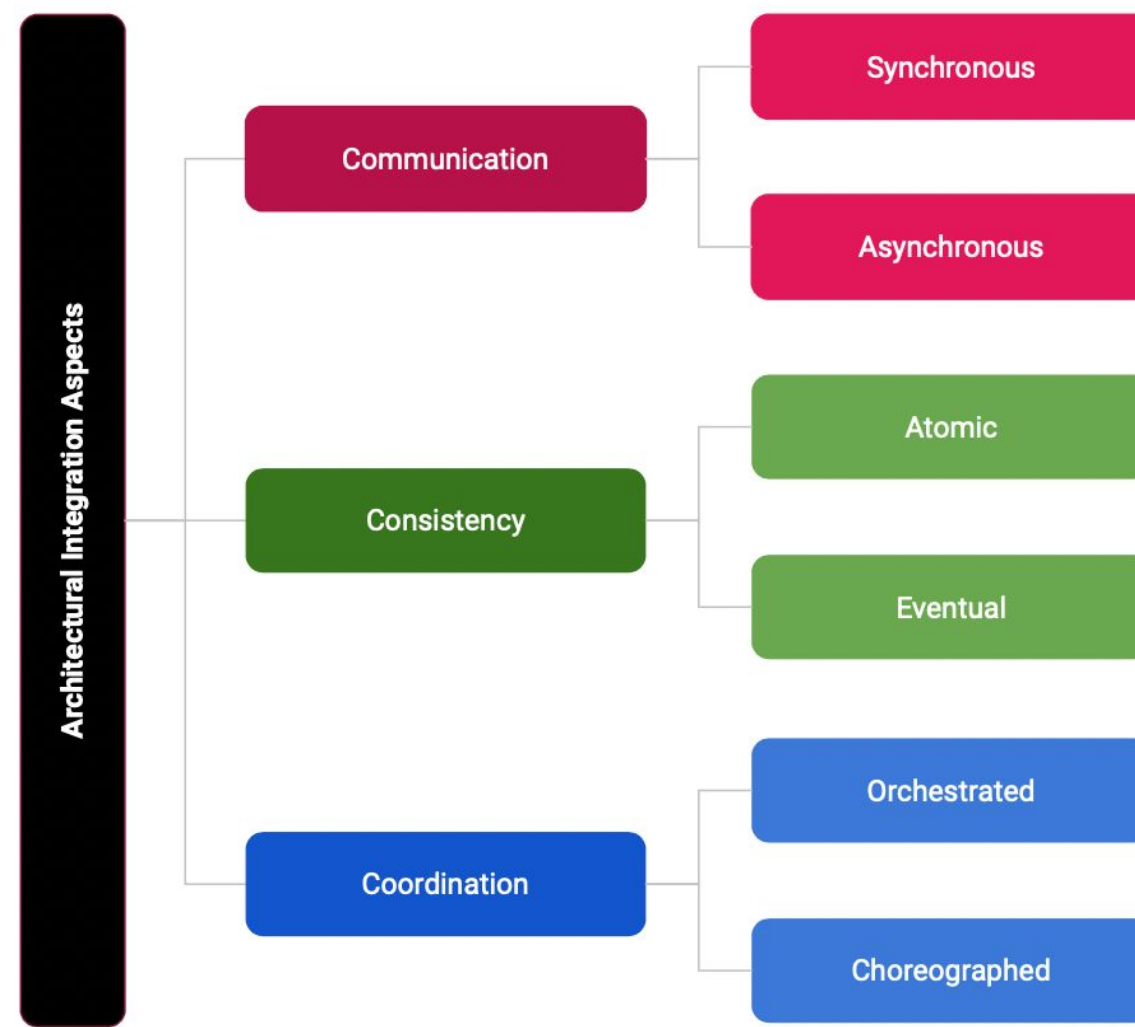
“Imagine um cenário em que há três serviços: A, B e C. Quando o serviço A conclui uma tarefa importante, ele dispara um evento que o serviço B e o serviço C podem ouvir. Cada um desses serviços reage a esse evento, executando ações específicas. Portanto, a lógica de coordenação está distribuída entre os próprios serviços.”

INTEGRAÇÃO ENTRE SISTEMAS

- Coordenação - Orquestração:

“Em um cenário de orquestração, o orquestrador pode ditar que o serviço A deve ser executado, esperar sua conclusão e, em seguida, chamar o serviço B, seguido pelo serviço C. O orquestrador tem uma visão completa do fluxo de trabalho e pode até mesmo lidar com fluxos alternativos ou exceções.”

INTEGRAÇÃO ENTRE SISTEMAS



PADRÕES E ESTILOS

- Listas de exercícios - Comunicação entre sistemas

