

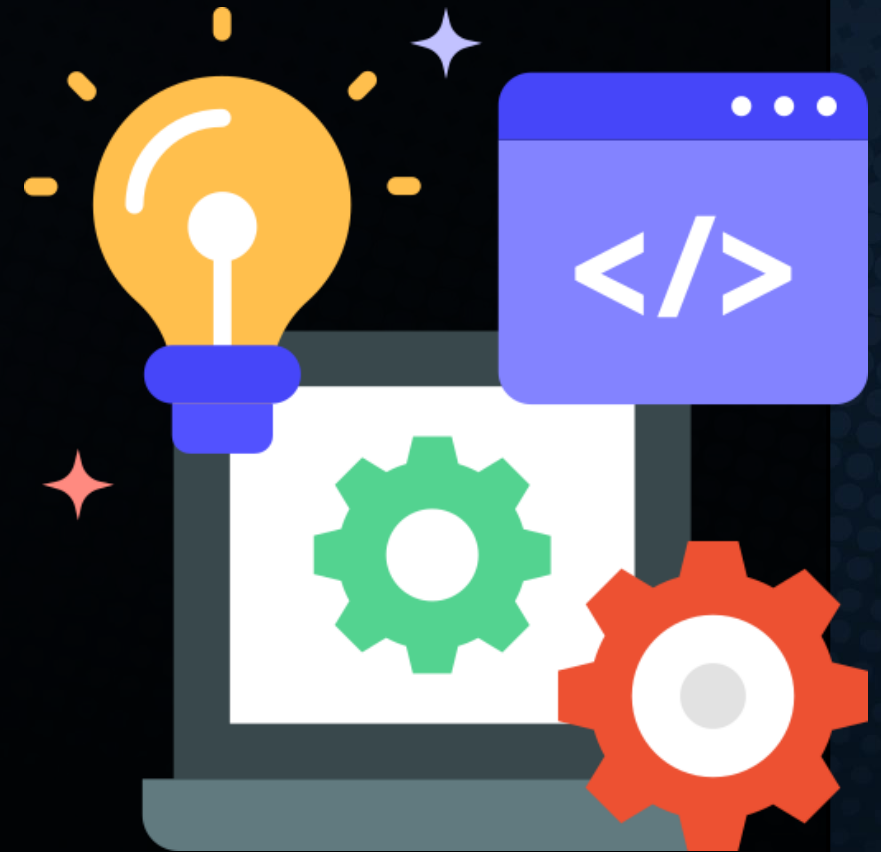


PROJETOS DE ARQUITETURA DE SOFTWARE

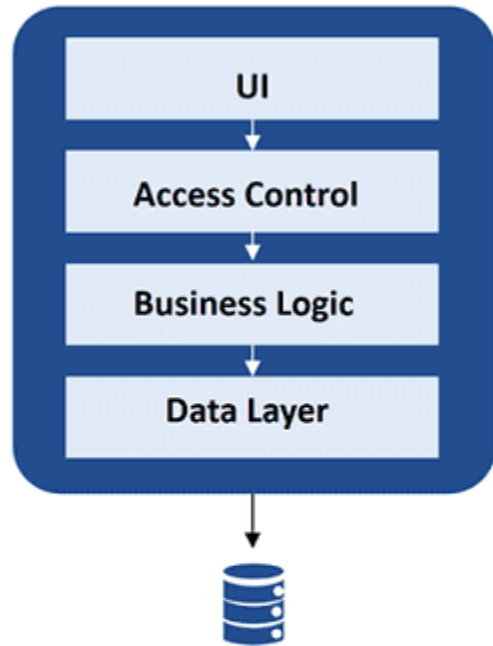
Prof. Luis Fernando dos Santos Pires

ARQUITETURA DE SOFTWARE

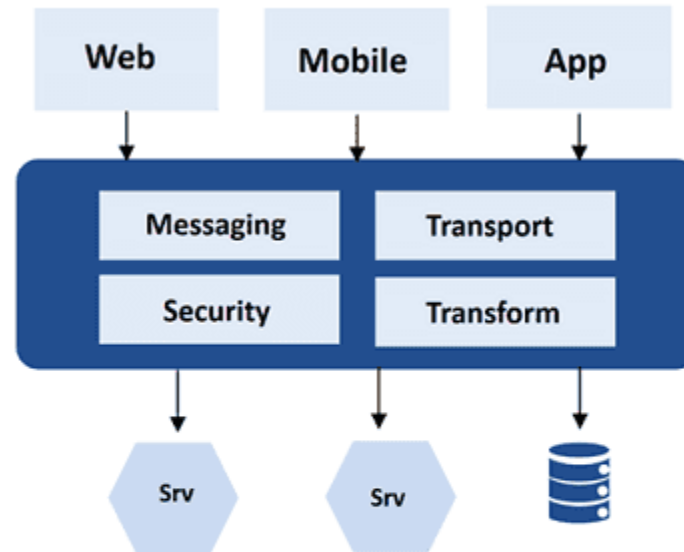
A **arquitetura de software** é o projeto estrutural e organizacional que define como os componentes de um sistema de software interagem entre si. É como a planta baixa de um edifício: estabelece os principais blocos funcionais, as conexões entre eles e as diretrizes para a construção. A arquitetura guia decisões importantes, como a escolha de padrões de design, tecnologias, distribuição de responsabilidades e escalabilidade. Uma boa arquitetura promove a manutenção, a flexibilidade e a eficiência do software, permitindo que ele evolua de maneira sustentável.



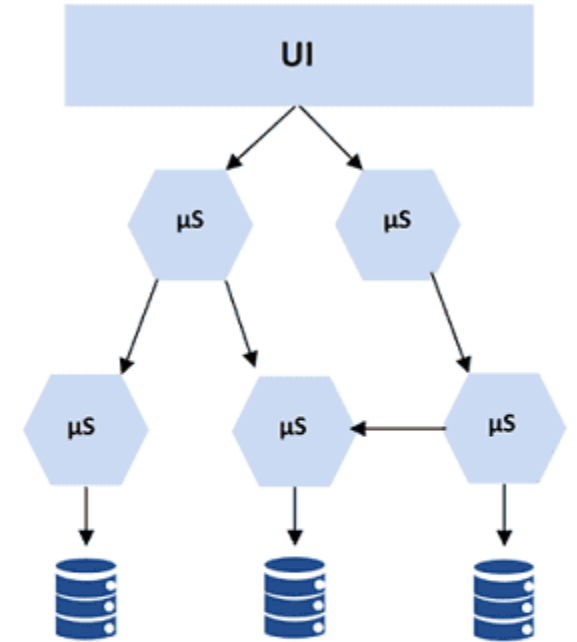
Evolução da Arquitetura de Software



Monolithic



Service-oriented



Microservices

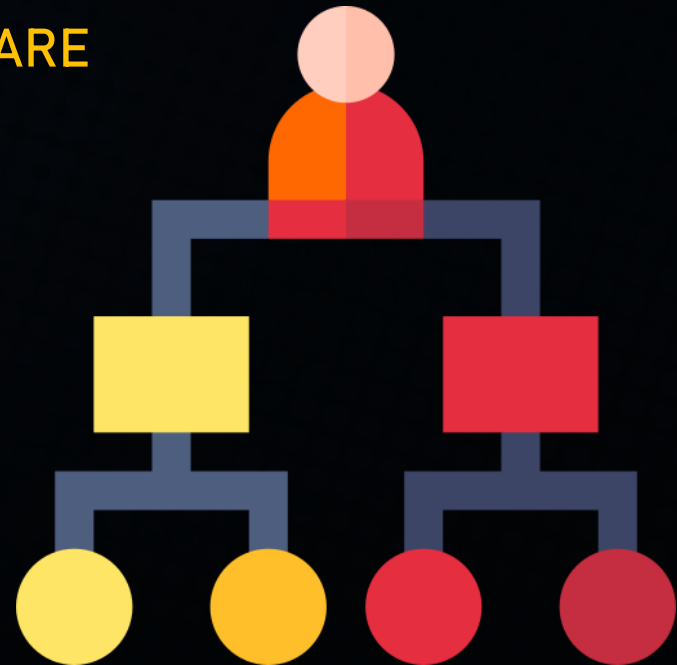
DIAGRAMA DE ARQUITETURA DE SOFTWARE

Um diagrama de arquitetura de software é uma representação visual que descreve a estrutura e organização de um sistema de software. Ele mapeia os principais componentes do sistema, suas interações e fluxos de dados. Imagine-o como um mapa detalhado que guia os desenvolvedores na construção de um edifício complexo. Esses diagramas ajudam a entender a separação de responsabilidades, a modularização e as interfaces do sistema, além de facilitar a comunicação entre equipes.

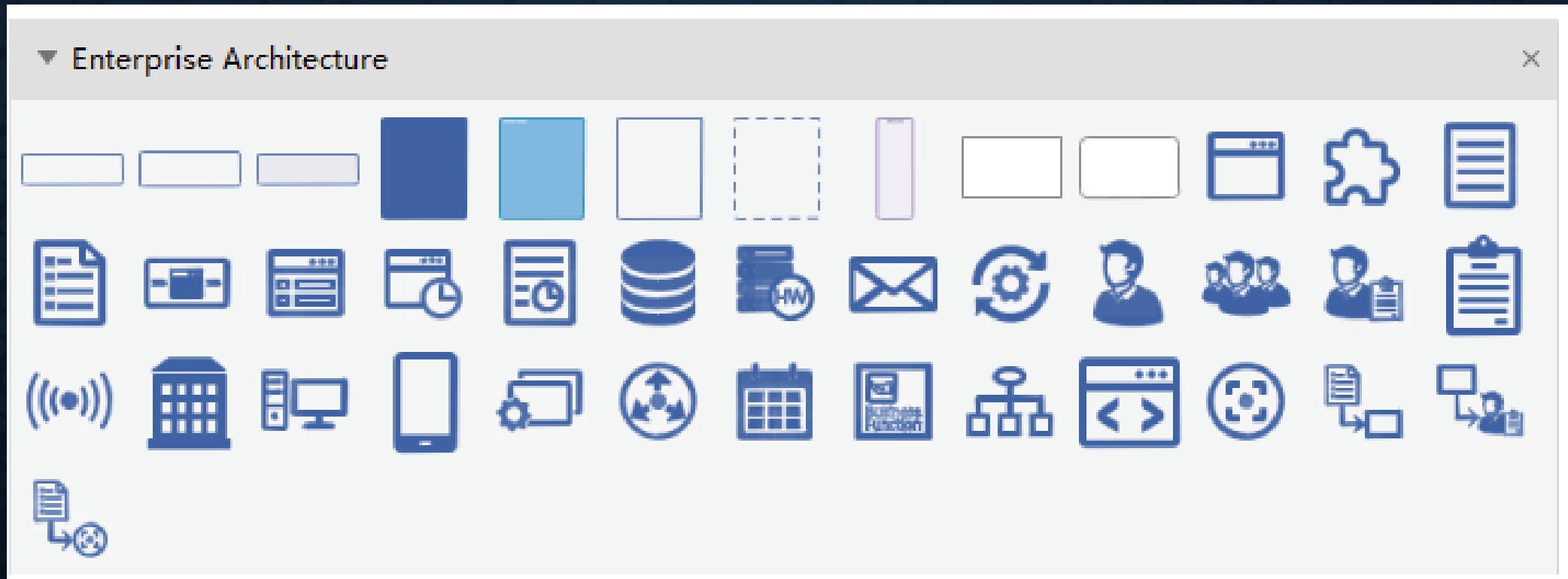


BENEFÍCIOS DO DIAGRAMA DE ARQUITETURA DE SOFTWARE

A criação de diagramas de arquitetura é essencial para projetar sistemas robustos e escaláveis. Eles permitem que arquitetos e desenvolvedores visualizem o panorama geral, escolham tecnologias apropriadas e definam padrões de design. Além disso, esses diagramas ajudam a manter a manutenção, a flexibilidade e a eficiência do software, permitindo que ele evolua de maneira sustentável.

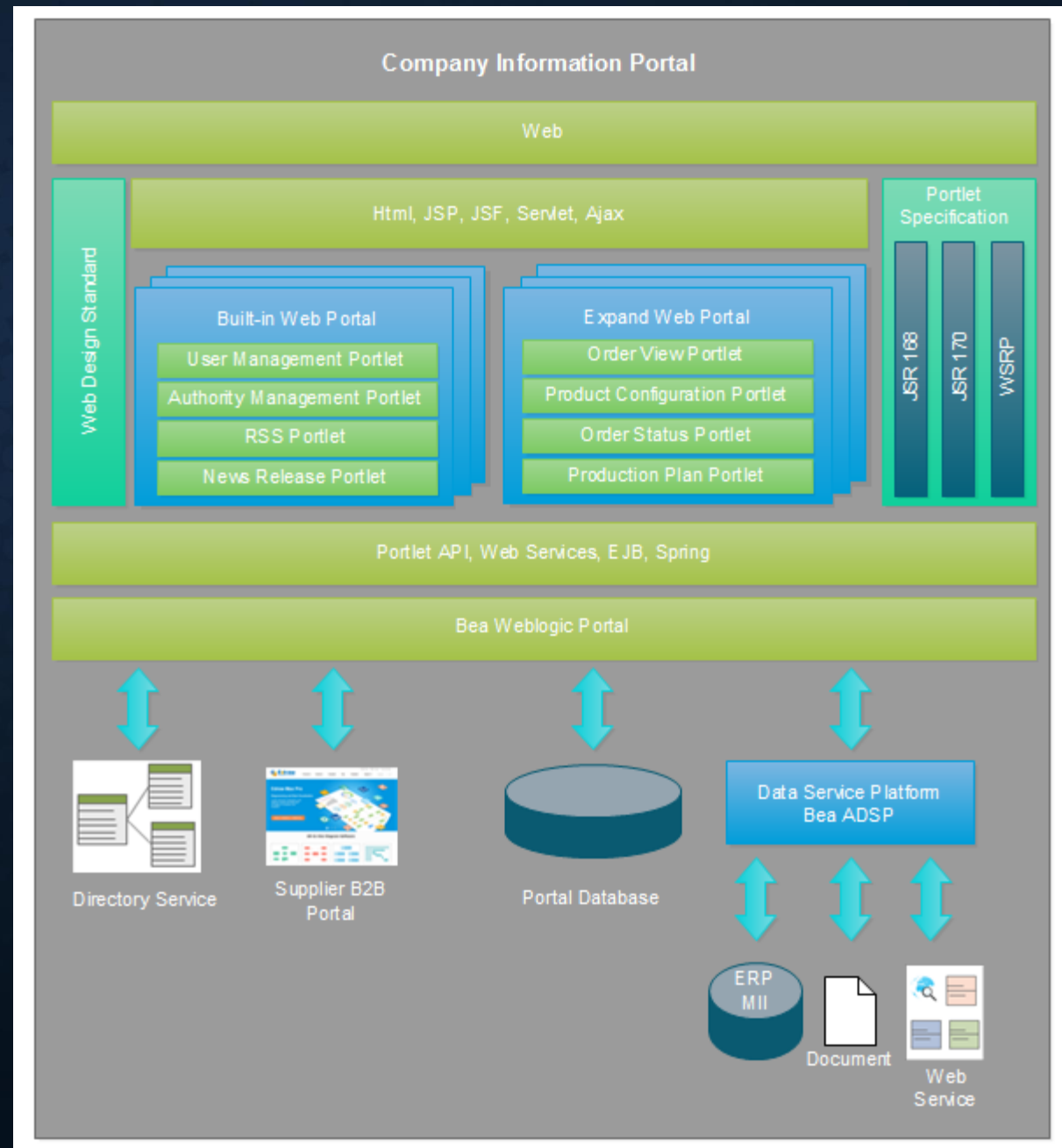


Elementos Comuns em Diagramas

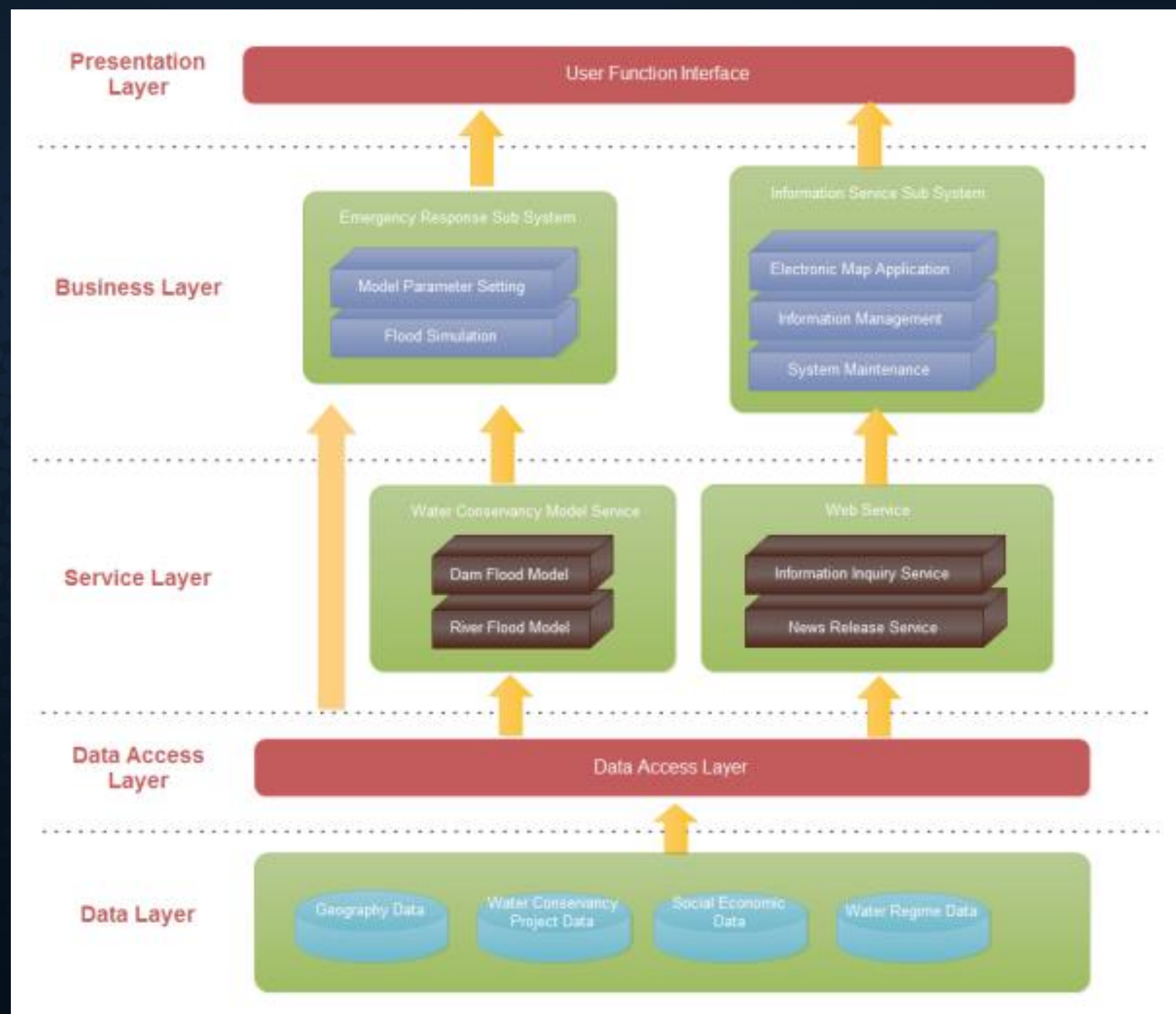


Arquitetura em Camadas

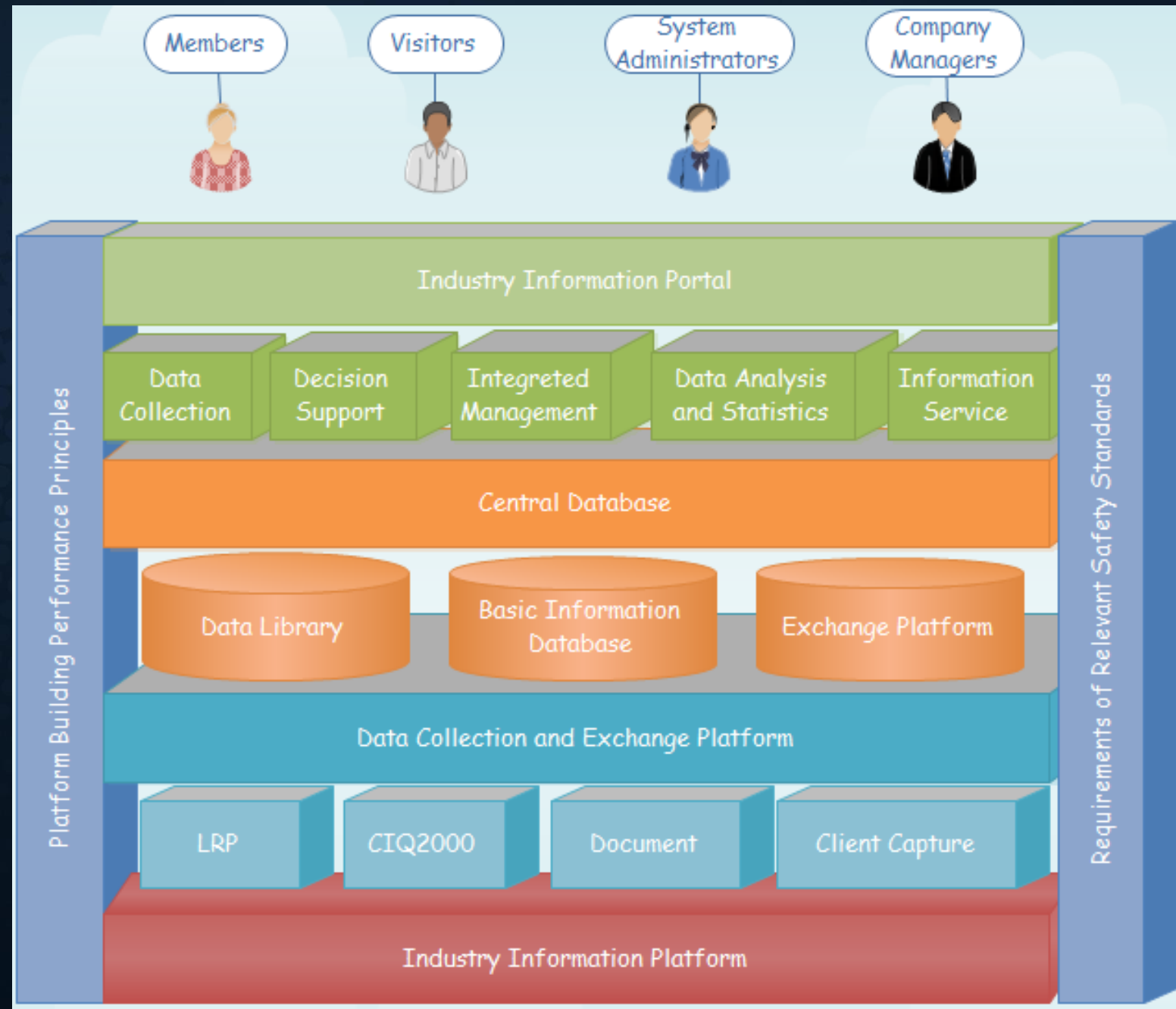
A arquitetura em camadas é uma abordagem crucial para o desenvolvimento de sistemas de software robustos e escaláveis. Nesse modelo, a aplicação é dividida em camadas distintas, cada uma com responsabilidades bem definidas. Cada camada possui um propósito específico, promovendo a separação de interesses e a modularidade do sistema. Por exemplo, podemos ter camadas como a de apresentação (interface com o usuário), lógica de negócios e acesso a dados. Essa organização facilita a manutenção, a evolução e a compreensão do software, além de permitir que componentes sejam reutilizáveis e substituíveis.



Exemplo Arquitetura

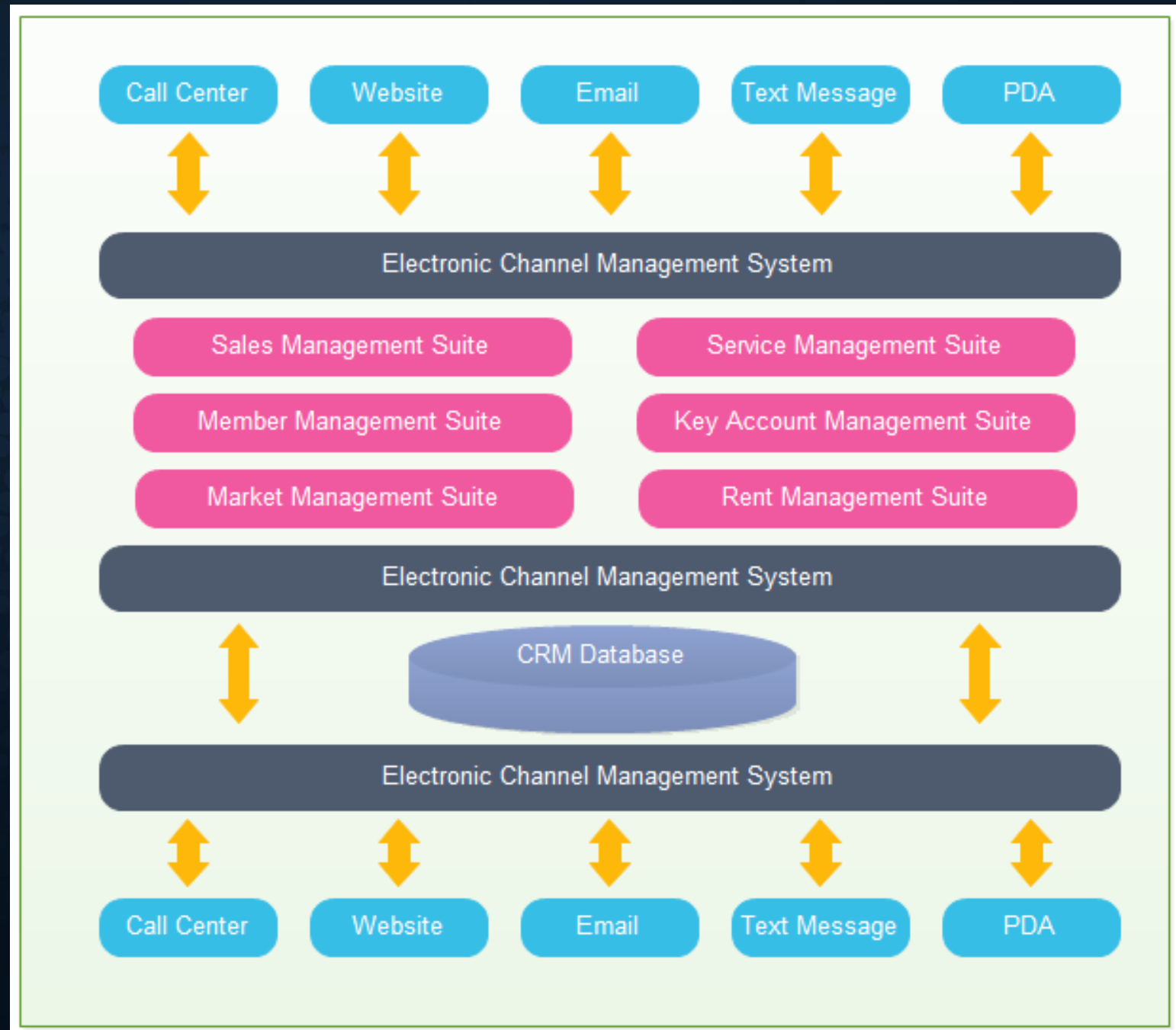


Exemplo Arquitetura

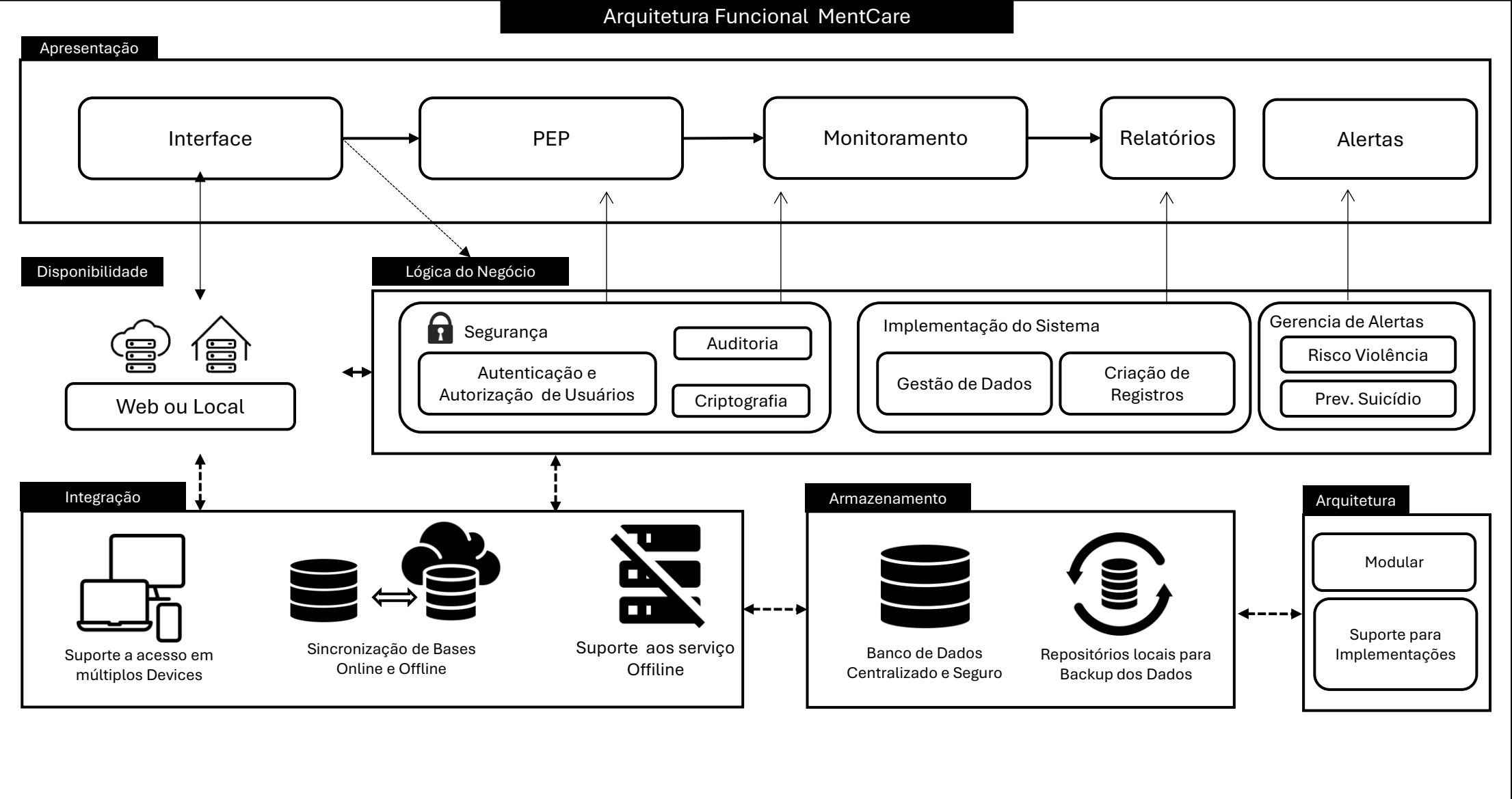


CRM Arquitetura

O diagrama representa um Sistema de Gerenciamento de Canais Eletrônicos. Nele, destacam-se duas suítes: Market Management Suite e Rent Management Suite à esquerda, e Sales Management Suite e Key Account Management Suite à direita. Essas suítes estão conectadas a cinco canais: Call Center, Website, Email, Text Message e PDA. No centro, temos o CRM Database. O fluxo de dados é indicado por setas, e o diagrama organiza informações complexas sobre a estrutura e fluxo desse sistema em um contexto empresarial.



Arquitetura Funcional



Arquitetura Cliente - Servidor

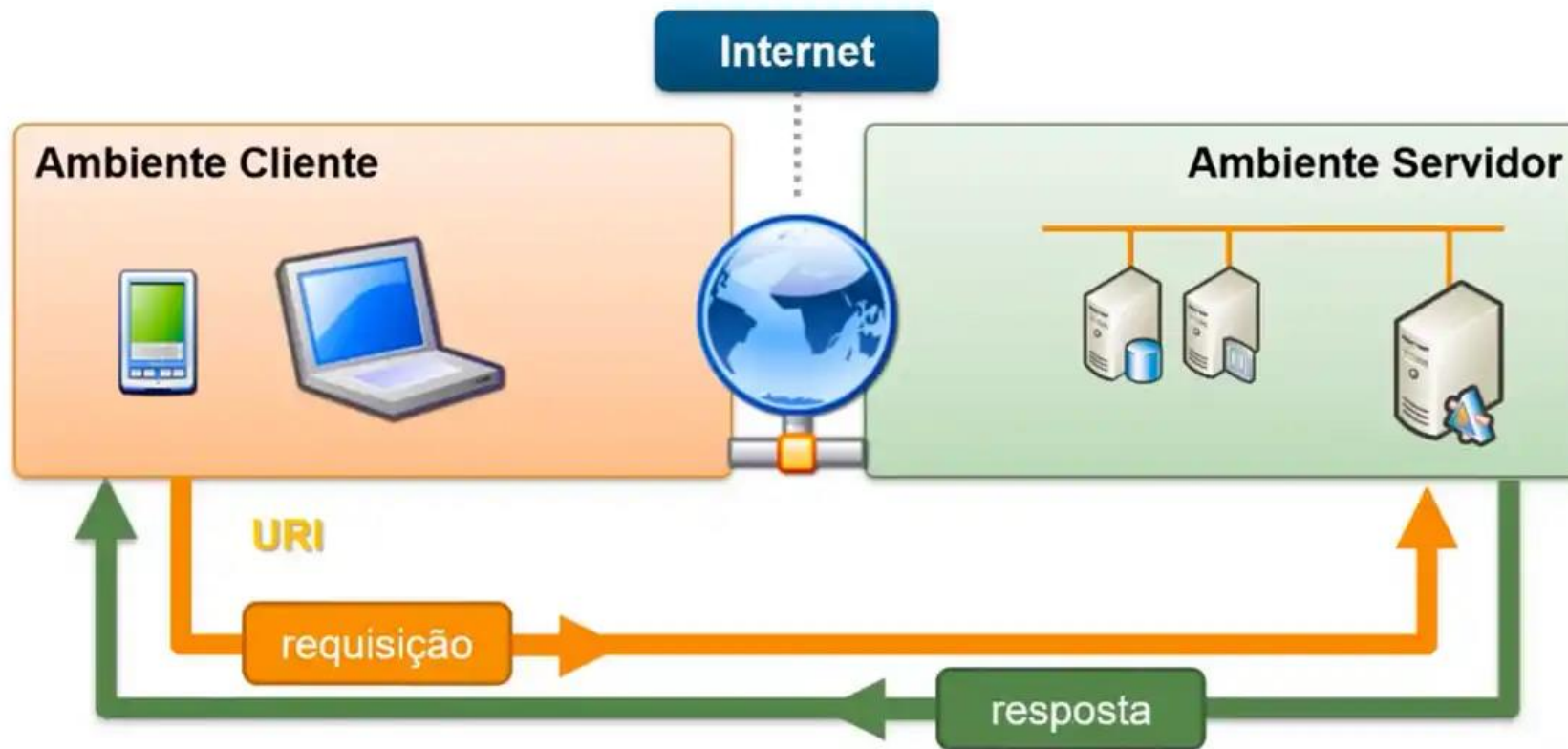
A **arquitetura cliente-servidor** é um modelo amplamente utilizado em ambientes de tecnologia da informação. Nesse paradigma, os computadores são divididos em dois grupos principais:

1.Servidores: Esses são os “provedores” de serviços ou recursos. Os servidores mantêm dados, executam aplicativos e respondem às solicitações dos clientes. Por exemplo, quando você acessa o site do Gmail, os servidores do Google processam suas ações, como enviar e-mails.

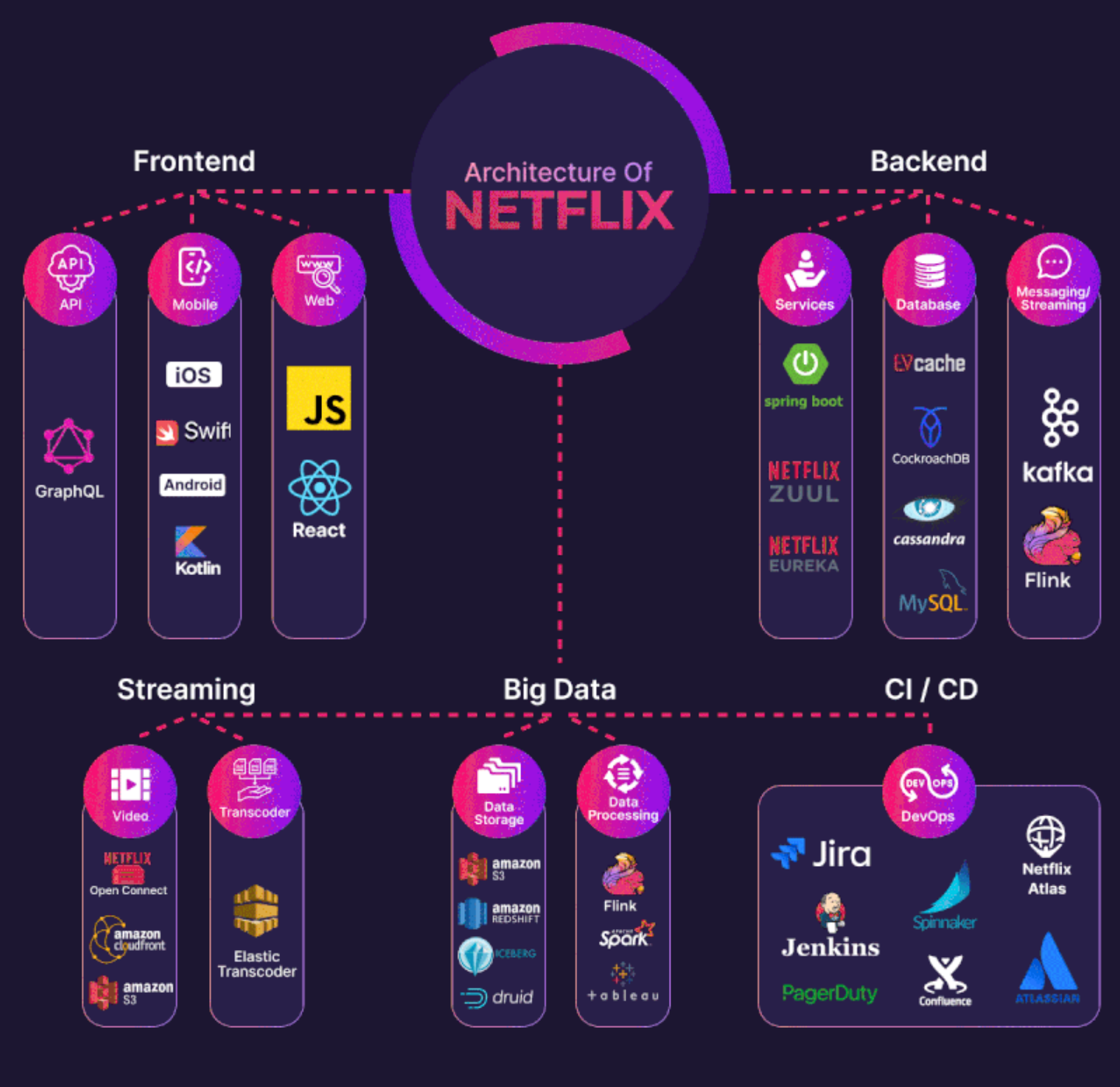
2.Clientes: Os clientes são os dispositivos (como computadores, smartphones ou tablets) que solicitam serviços ou recursos aos servidores. Eles enviam requisições para obter informações ou executar funções específicas. Por exemplo, seu navegador é um cliente que solicita páginas da web a servidores remotos.

A comunicação entre clientes e servidores ocorre por meio de protocolos de rede, como o TCP/IP. Isso permite que empresas distribuam serviços e recursos em vários dispositivos e plataformas, tornando o modelo cliente-servidor altamente escalável e flexível. Imagine os servidores como os “provedores” e os clientes como os “solicitantes”.

Arquitetura Cliente - Servidor



Arquitetura Técnica



Arquitetura MVC

A arquitetura Model-View-Controller (MVC) é um padrão amplamente utilizado no desenvolvimento de software. Nesse modelo, o sistema é dividido em três componentes principais:

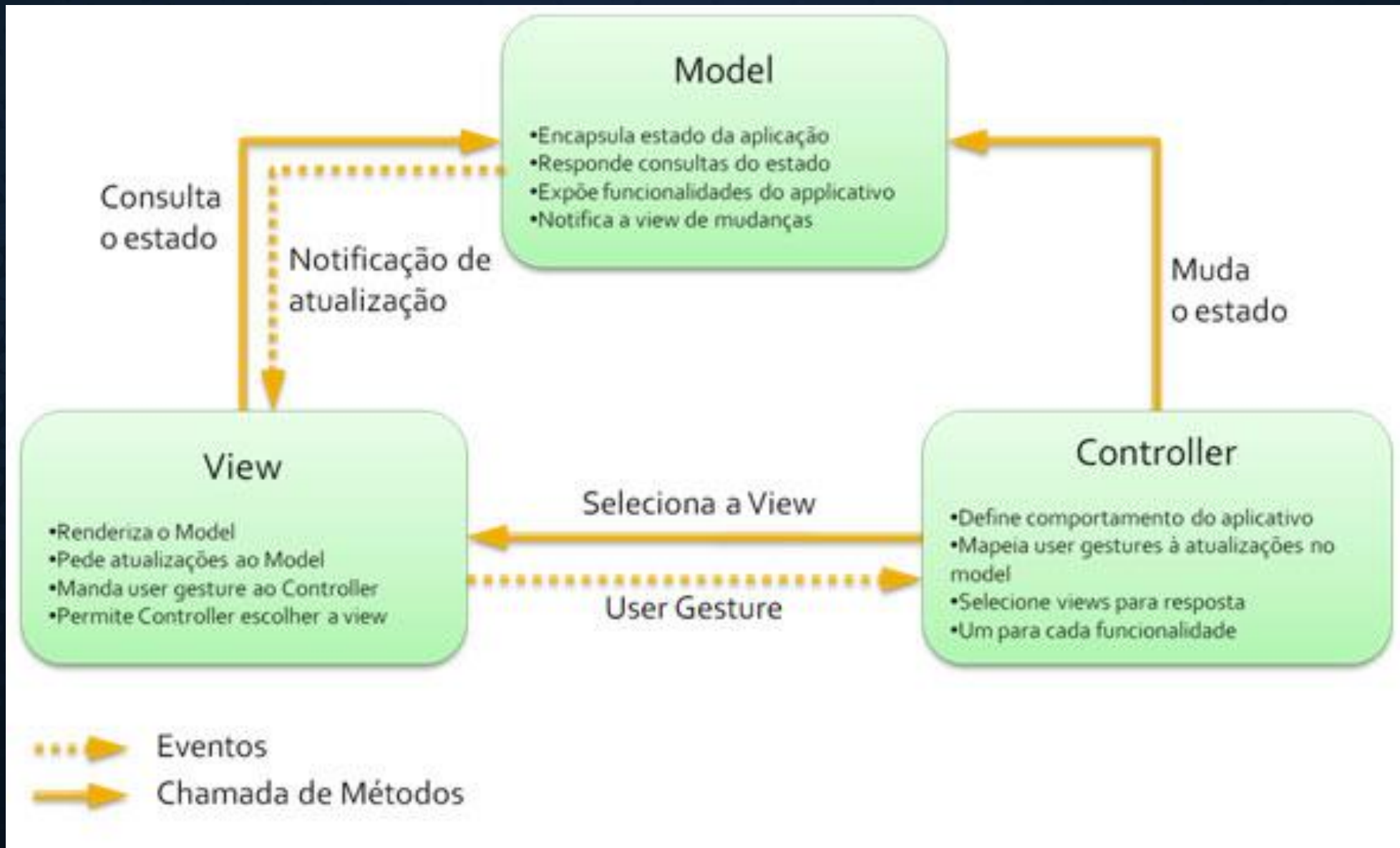
1.Model (Modelo): Representa os dados e a lógica de negócios. Ele gerencia o acesso aos dados, realiza cálculos e processa as regras de negócios. O Modelo é independente da interface do usuário.

2.View (Visualização): É responsável por exibir os dados ao usuário. Ela representa a interface gráfica por meio da qual os usuários interagem com o sistema. As Views são atualizadas sempre que os dados do Modelo mudam.

3.Controller (Controlador): Atua como intermediário entre o Modelo e a View. Ele recebe as entradas do usuário (como cliques em botões) e decide como o Modelo deve ser atualizado. O Controlador também responde a eventos e direciona as ações apropriadas.

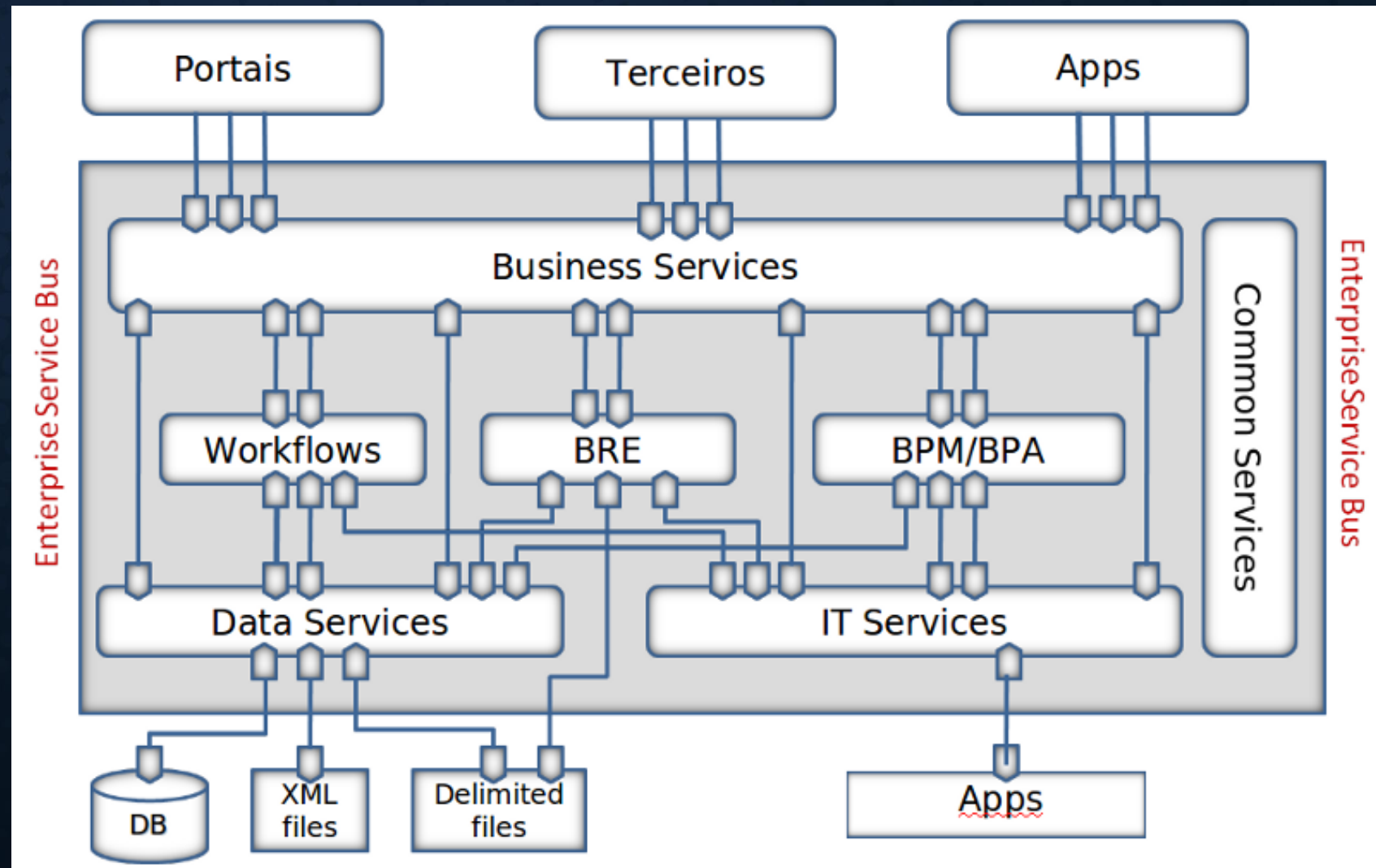
O padrão MVC promove a separação de preocupações, facilitando a manutenção, a escalabilidade e a reutilização do código. Ele é comumente usado em frameworks web, como o Ruby on Rails e o Spring MVC.

Arquitetura MVC



Arquitetura SOA

A arquitetura orientada a serviços (SOA) é um modelo de design de software que enfatiza a modularidade e a reutilização. Nesse paradigma, os sistemas são compostos por serviços independentes, cada um com uma função específica. Esses serviços se comunicam por meio de interfaces padronizadas, como APIs. A SOA permite flexibilidade, escalabilidade e integração entre diferentes aplicativos e plataformas. Ela é amplamente usada em ambientes empresariais para criar sistemas distribuídos e adaptáveis.



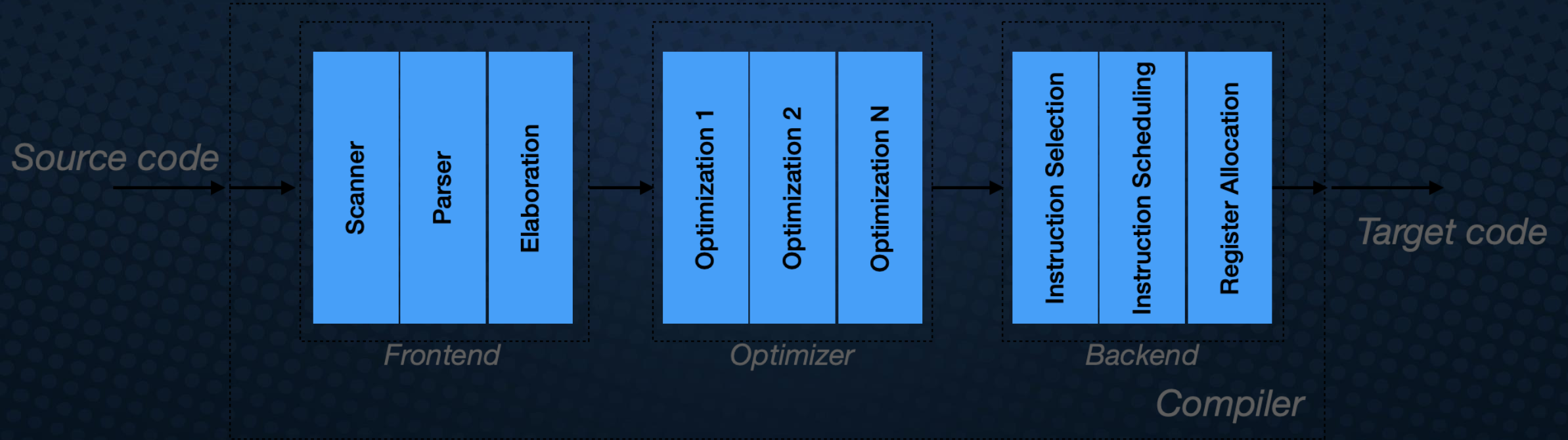
Arquitetura Pipes and filters

Pipe and Filter (ou “Pipes e Filtros”) é um padrão estrutural utilizado em design de sistemas. Nesse modelo, um processo é dividido em uma sequência de etapas discretas chamadas filtros, que são conectadas por canais conhecidos como pipes. Cada filtro é responsável por uma tarefa específica de processamento, como transformar, validar ou agregar dados. Imagine isso como uma linha de montagem, onde cada filtro faz sua parte no processo, e os dados fluem de um para o outro, passando por transformações e refinamentos.

Essa abordagem modular permite que os componentes (os filtros) trabalhem de forma independente, o que melhora a escalabilidade e a reutilização. Um exemplo prático disso é o processamento de vídeo ou áudio, onde cada filtro pode ser responsável por tarefas diferentes, como remover ruídos de fundo ou converter o formato dos dados.



Arquitectura Pipes and filters



Resposta

1. Qual é o papel da arquitetura de software no desenvolvimento de sistemas complexos?
2. Quais são os benefícios de utilizar uma arquitetura em camadas no desenvolvimento de software?
3. Como os diagramas de arquitetura de software facilitam a comunicação entre as equipes de desenvolvimento?
4. Quais são os elementos comuns que devem ser representados em um diagrama de arquitetura de software?
5. Como a arquitetura cliente-servidor promove escalabilidade e flexibilidade em sistemas distribuídos?
6. De que forma o padrão MVC separa as responsabilidades no desenvolvimento de software?
7. Quais são as vantagens da arquitetura orientada a serviços (SOA) em ambientes empresariais?
8. Como o padrão Pipes and Filters ajuda a modularizar e escalar processos de dados complexos?
9. Quais desafios podem surgir ao implementar uma arquitetura modular em sistemas grandes?
10. Como a escolha da arquitetura impacta a manutenção e evolução futura do software?

Desafio

1. Crie a arquitetura funcional ou de seu projeto.
2. Crie a arquitetura Cliente-Servidor.
3. Crie a Arquitetura Técnica.
4. Escolha ou padrão de Arquitetura não mencionado acima e aplique em seu projeto.



PROJETOS DE ARQUITETURA DE SOFTWARE

Prof. Luis Fernando dos Santos Pires