Aplicações WEB

- Em sua concepção a WEB é um meio para a troca de documentos HTML entre diferentes pontos, utilizando a infra-estrutura oferecida pela Internet.
- A medida que a WEB se popularizou novas aplicações começaram a surgir
- Em muitos sites da WEB o conteúdo não poderia ser constituído de código HTML **estático, mas precisaria ser alterado, muitas das vezes a** cada minuto.
- A partir desta necessidade começaram a surgir meios para se conseguir a produção de conteúdos **dinâmicos**
- A primeira proposta para a criação de conteúdos dinâmicos foi através do CGI (Common Gateway Interface)
- Este mecanismo permite a execução de um código, escrito em C ou Perl, através do navegador

Aplicações WEB Usando CGI

- Uma aplicação CGI pode ser invocada da seguinte forma:

http://www.server.com/cgi-bin/MyExecutable?name1=value1&name2=value2

- Aplicação CGI, porém apresenta algumas desvantagens:
  - Criada a partir de linguagens procedimentais
  - Instabilidade em uma aplicação CGI pode impedir até a operação do servidor
  - Problemas de Escalabilidade. A cada chamada uma nova instância da aplicação é criada, criando um novo thread e consumindo recursos do servidor.
  - Outro aspecto é que através de uma aplicação CGI não é possível agregar recursos na aplicação como: estabelecimento de sessão; autenticação e autorização.

Outras tecnologias para aplicação Web Dinâmicas

- 1995
  - Personal Home page (PHP) 2.0
- 1996
  - Active Server Pages (ASP) 1.0
- 1997
  - Active Server Pages (ASP) 2.0
  - Java Servlet 1.0
- 1998
  - PHP3; Java Servlet 2.1
  - LAMP (Linux, Apache, MYSQL, PHP)
- 1999
  - Java Servlet 2.2
  - JavaServer Pages (JSP)
- 2000
  - PHP4
- 2002
  - ASP.net
- 2003
  - ASP.net 1.1; Java Servlet 2.4
- 2004
  - PHP5
- 2005
  - ASP.net 2; Django (Python); Ruby

Aplicações Distribuídas Camadas

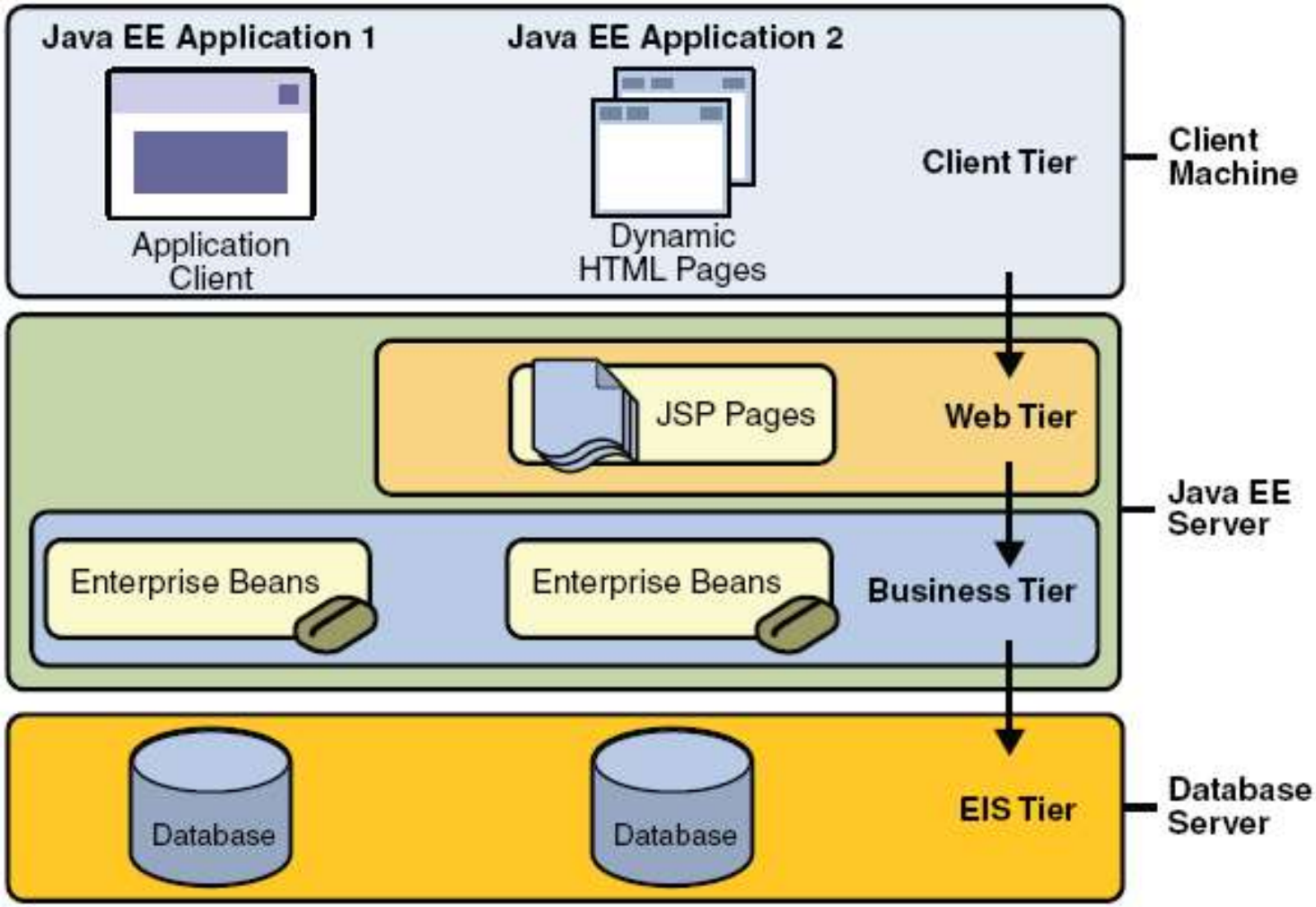
- O modelo para desenvolvimento de aplicações web é baseado em uma arquitetura multicamadas
- Cada camada contém componentes de acordo com a sua função.
- Basicamente existem as seguintes camadas:
  - Cliente (Client Tier)
  - Camada Web (Web Tier)
  - Camada de Negócios (Bussiness Tier)
  - Camada Enterprise Information System (EIS)
- A camada Web Tier estará presente quando a aplicação for baseada na WEB ou seja, o cliente é um navegador (browser)
- A arquitetura acima é conhecida como três camadas (3-tier) visto que existem 3 máquinas envolvidas:
  - O cliente;
  - O servidor de aplicações
  - O servidor de banco de dados

Aplicações Distribuídas Camadas

- Cliente (Client Tier)
  - Componentes que são executados na máquina do cliente.
  - Podem ser executados em um Browser ou uma aplicação desktop ou ainda mobile
- Camada Web (Web Tier)
  - Componentes que são executados em um servidor
  - Basicamente tratam da apresentação do conteúdo na WEB
  - Esta camada é responsável por gerar e enviar para o cliente o conteúdo gerado de forma dinâmica
- Camada de Negócios (Bussiness Tier)
  - Contém os objetos relacionados ao negócio e suas regras
  - Existem servidores de aplicações que oferecem recursos como controle de transações, de sessões para os objetos desta camada.
- Camada Enterprise Information System (EIS)
  - Consiste dos recursos que serão utilizados pela aplicação.
  - Inclui gerenciadores de banco de dados (DBMS) e aplicações legadas, baseadas em mainframes, por exemplo.
  - Geralmente há conectores para ligar a camada Business Tier com esta camada

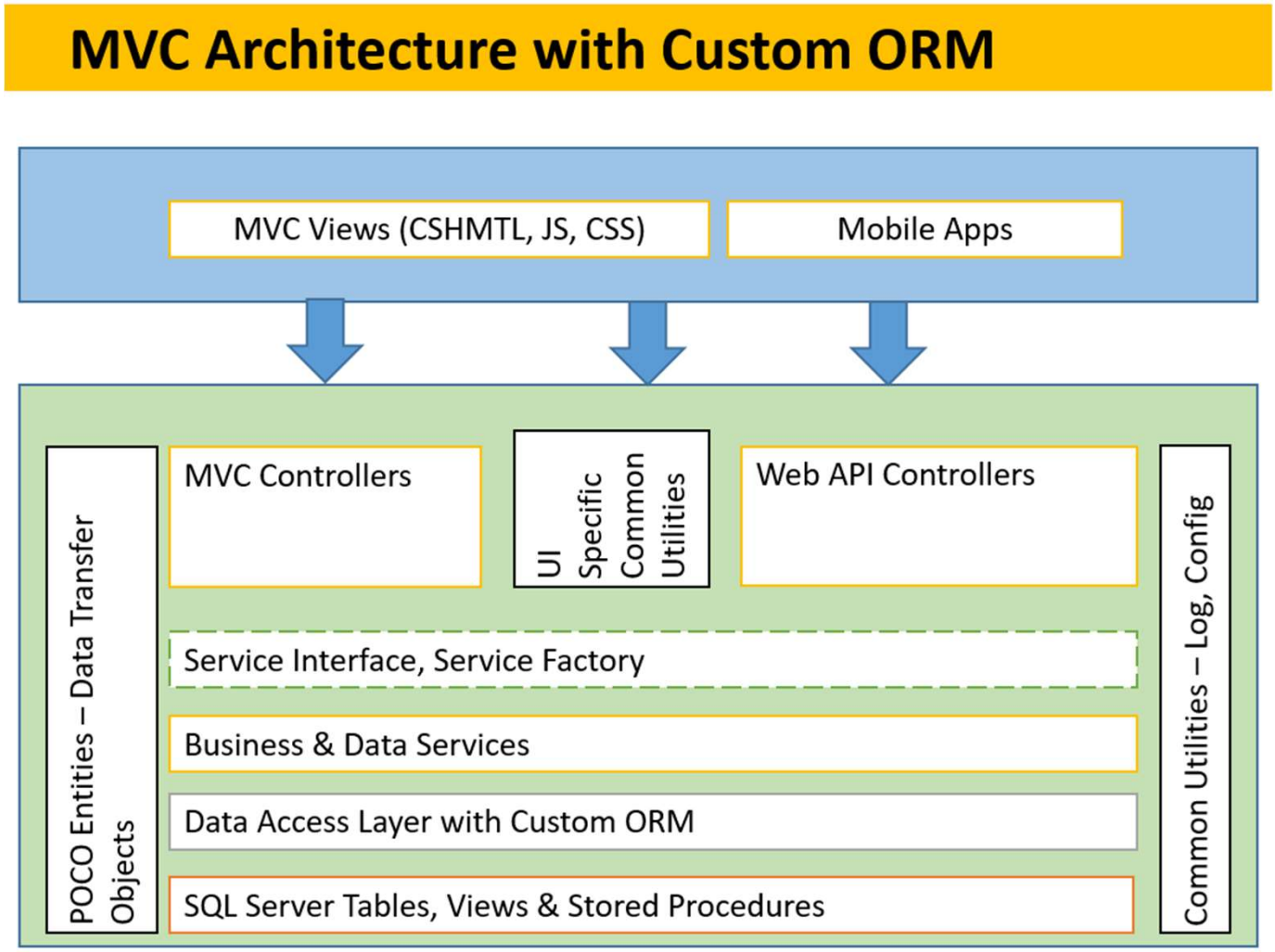
Aplicações Distribuídas Camadas – Arquitetura Java EE

- Visão Geral das Camadas



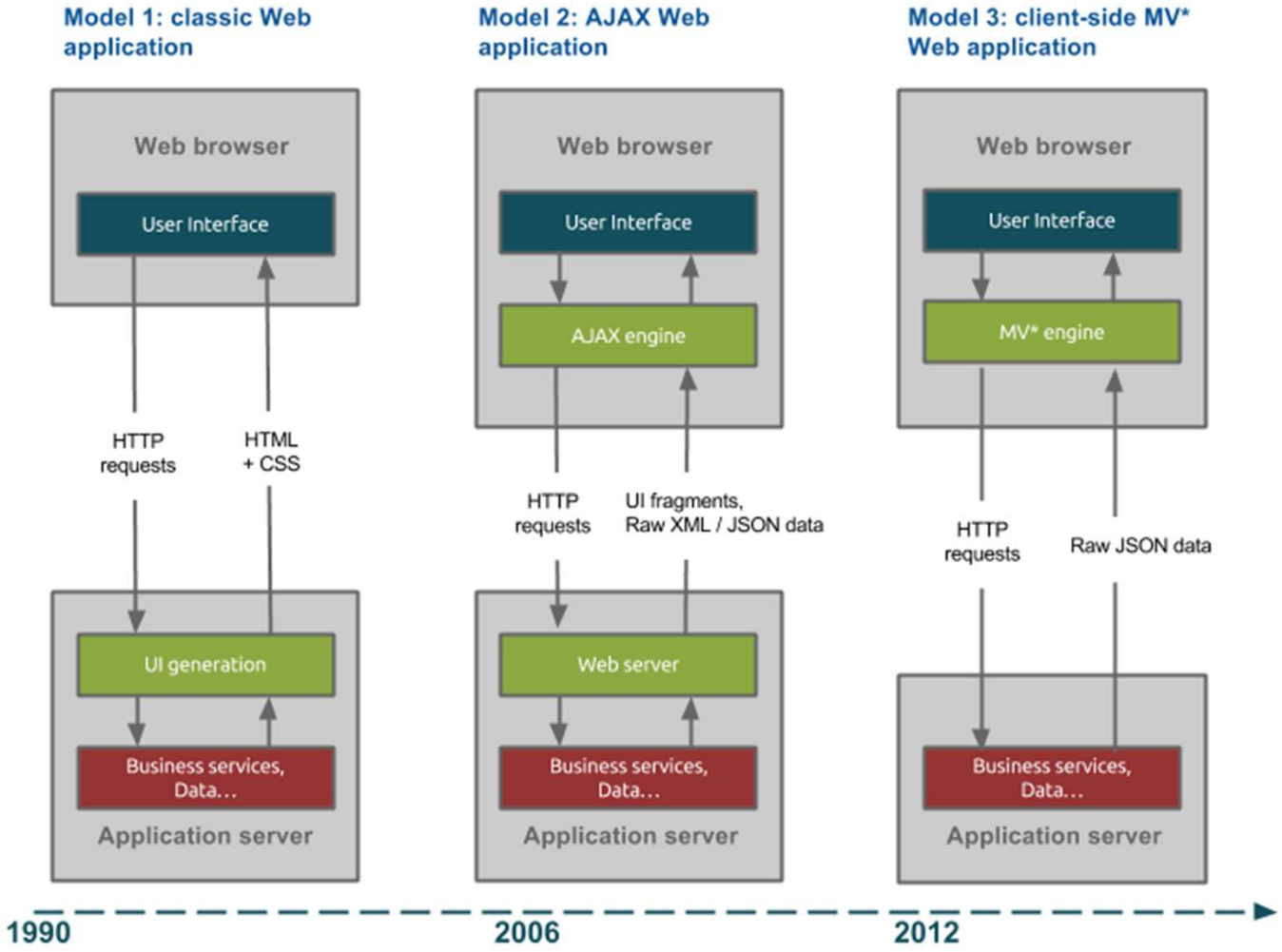
Aplicações Distribuídas
Camadas – Arquitetura .NET

- Visão Geral das Camadas



Aplicação Web
Evolução da Arquitetura

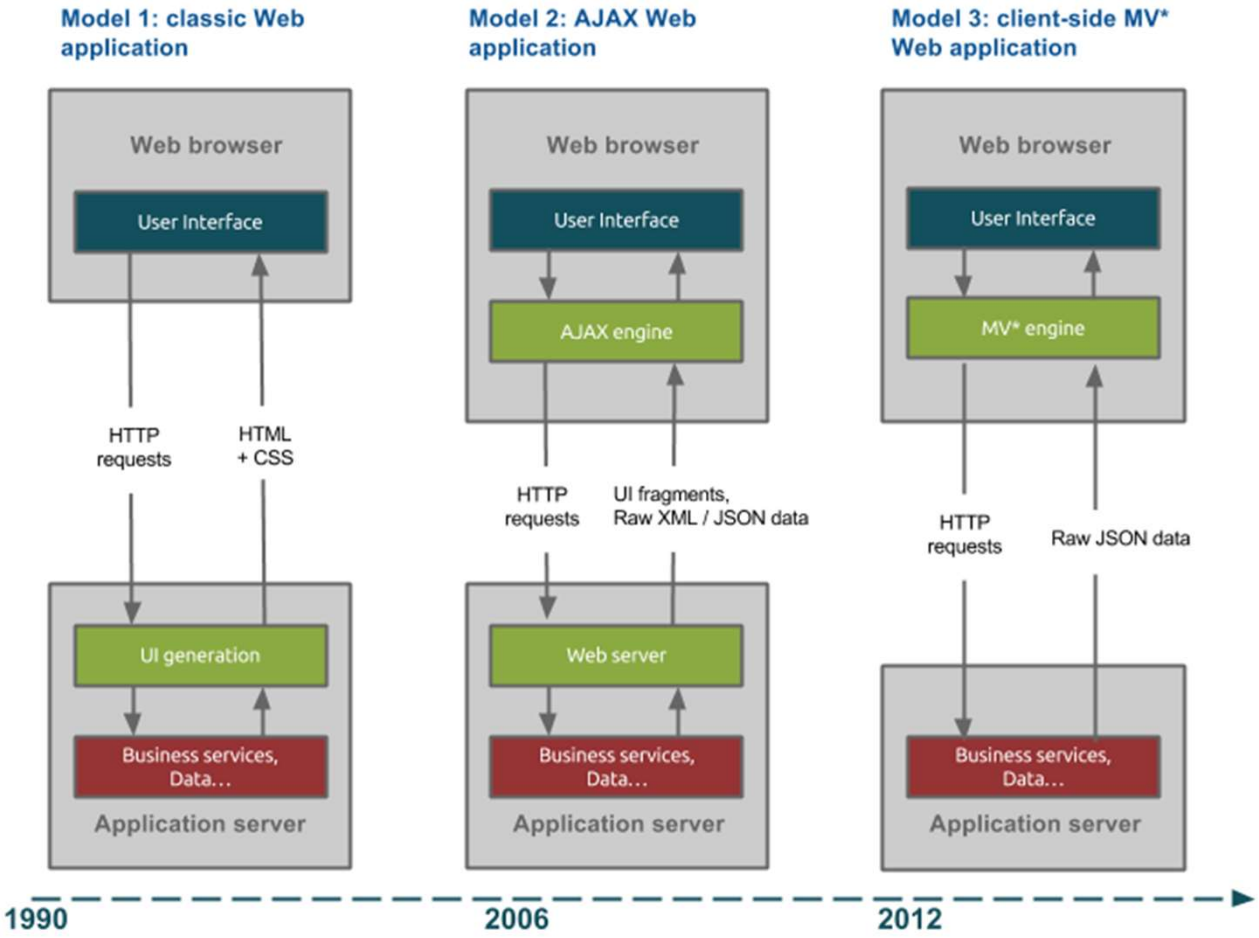
- Evolução



https://blog.octo.com/en/new-web-application-architectures-and-impacts-for-enterprises-1/

Aplicação Web
Evolução da Arquitetura

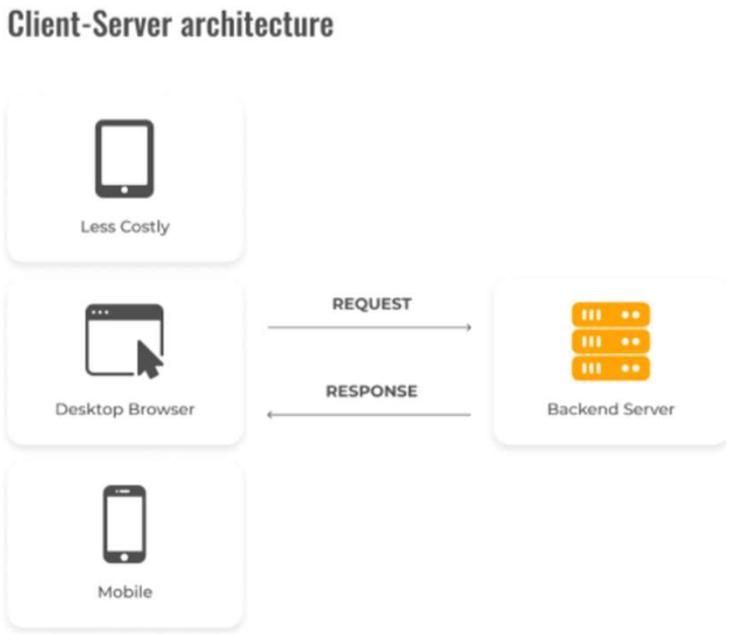
- Evolução
  - Classic Web App
    - Maior execução do lado servidor
  - Ajax Web App
    - Mais responsivo
    - Uso de JS no lado cliente para tratar requisições
  - MV* Web App
    - MV* são implementações do Model-View-Controller com variações como MVVM (Vue)
    - Servidor envia apenas os dados brutos (raw) sem formatação
    - Toda lógica de UI vai para o lado cliente
- Evolução possível por avanço nos navegadores e JavaScript



https://blog.octo.com/en/new-web-application-architectures-and-impacts-for-enterprises-1/

Arquiteturas de Aplicações Web

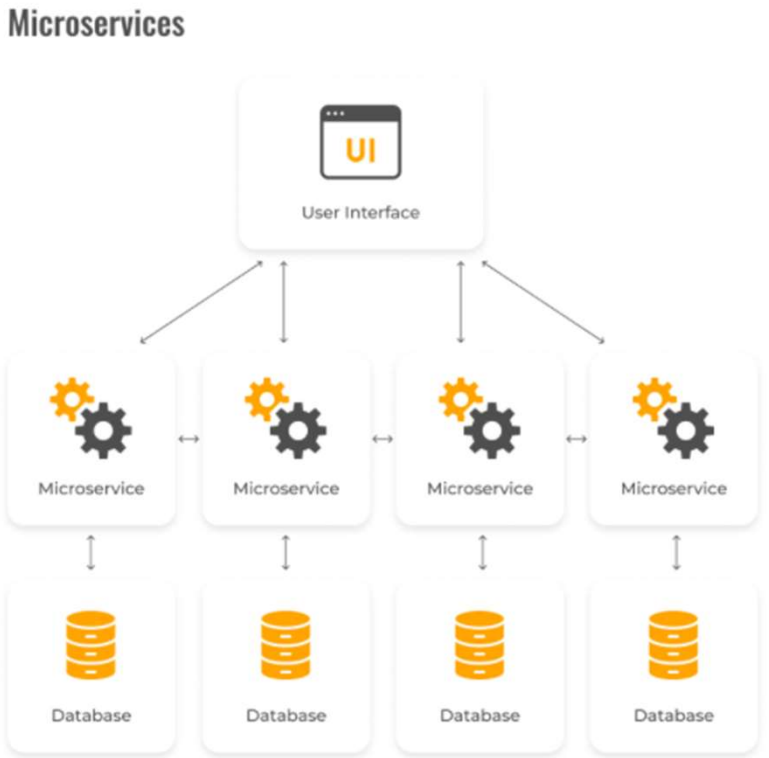
- Client-Server
  - Arquitetura inicial das web apps
  - Chamada arquitetura monolítica



https://lanars.com/blog/web-application-architecture-101

Arquiteturas de Aplicações Web

- Microservices
  - Ideal para projetos maiores
  - Arquitetura mais facilmente escalável
  - Cada microserviço é responsável por uma função ou um grupo de funções da aplicação
  - Simplifica a manutenção do sistema
  - Ambiente de nuvem (cloud)



https://lanars.com/blog/web-application-architecture-101

Arquiteturas de Aplicações Web

- Serverless
  - Baseada em serviços disponíveis em ambientes de cloud (AWS, Azure, etc.)
  - Aplicação fica ligada (coupled) com o fornecedor
  - Parte dos serviços são oferecidas pelo provedor de nuvem

https://lanars.com/blog/web-application-architecture-101

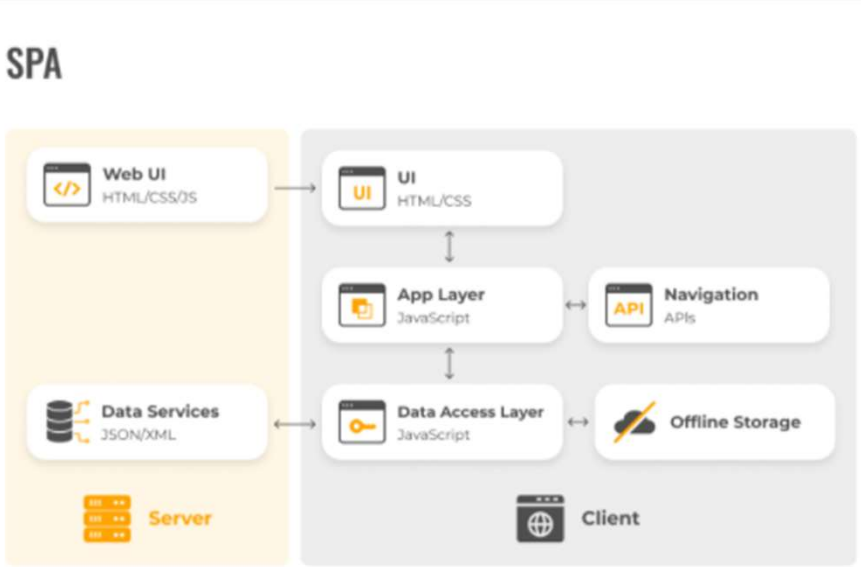
Arquiteturas de Aplicações Web

- Progressive Web App (PWA)
  - Aplicação web funciona como uma aplicação móvel, sem a necessidade de download de aplicativos móveis
  - Podem interagir com as APIs de hardware do equipamento

https://lanars.com/blog/web-application-architecture-101

Arquiteturas de Aplicações Web

- Single Page Applications (SPA)
  - Baseada no conceito que toda interação ocorre em uma única página
  - Exemplos: GMAIL, Google Maps, Netflix, Pinterest, Paypal
  - App responde às entradas do usuário
  - Conceito antigo, porém com novas tecnologias
  - Tecnologias
    - AJAX
    - WebSockets
      - Comunicação full duplex entre navegador e servidor
      - Statefull
    - XML,JSON

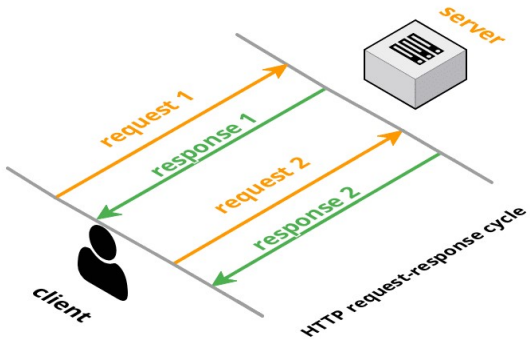


https://lanars.com/blog/web-application-architecture-101

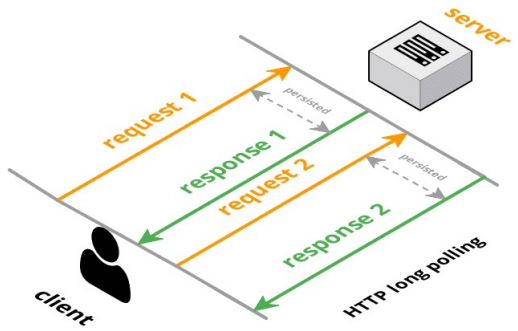
Arquiteturas de Aplicações Web Websockets

- WebSocket
  - Comunicação full duplex entre navegador e servidor
  - HTTP Statefull sobre uma conexão TCP
  - Diferente do conceito inicial do HTTP
    - Uma nova conexão a cada request
  - Evolução do HTTP Long Pooling
    - Long Pooling: Mesma conexão TCP persiste por tempo maior

Http



HTTP Long Pooling



WebSocket