



Laboratório de Engenharia de Software I

Profa. Lucia Vilela Leite Filgueiras

Turma 01: quartas-feiras

Turma 02: quintas-feiras

Aula 03

Arquitetura de software

Roteiro da aula

Hora	Atividade
08:20-08:40	Apresentação do roteiro da aula. Apresentação sobre arquitetura de software.
08:40-10:00	Apresentação das tecnologias e seu uso.
10:00-11:00	Planejamento do estudo e dos sprints
11:00-12:00	Apresentação do planejamento

Onde estamos?

Figura extraída do livro Learning UML 2.0, [Russ Miles](#); [Kim Hamilton](#)

Figure 1-7. Philippe Kruchten's 4+1 view model

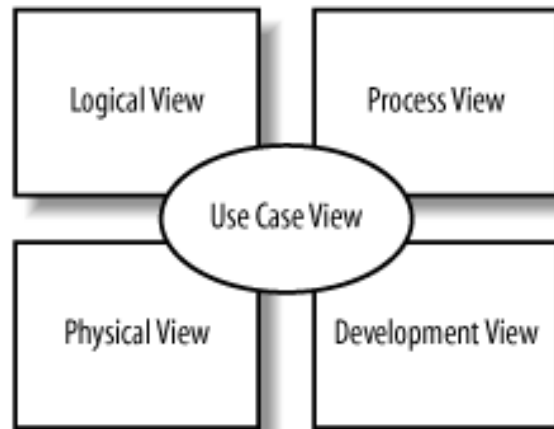


Figure 13-1. The Development View describes how your system's parts are organized into modules, which are represented as packages in UML

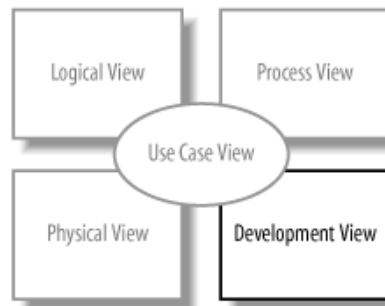
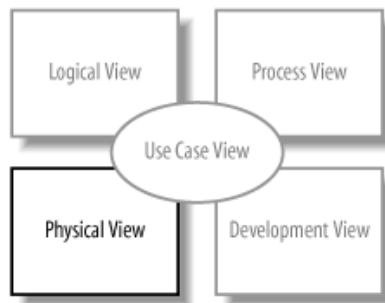


Figure 15-1. Deployment diagrams focus on the Physical View of your system



Agenda

- Conceitos de arquitetura
- Estilos de arquitetura

The background features a large, solid green circle on the right side. To its left, a black, crescent-shaped graphic element curves around the green circle. The entire composition is set against a white background with several thin, light gray lines that form concentric and overlapping circular patterns, some of which are dashed.

Conceitos de arquitetura

Casa de cachorro

- Pode ser construída por uma pessoa só, com um conjunto básico de ferramentas, com pouco conhecimento do processo de construção, com nenhuma necessidade de modelos.

Foto de <http://www.rgoarchitects.com>



Palafita

- Em geral envolve mais de uma pessoa no processo de construção, requer conhecimento do processo de construção e do domínio, necessidade baixa de modelos.





Casa urbana

- Construída de forma mais eficiente por um time, com processos bem definidos, ferramentas adequadas, modelos ao longo do processo de concepção e construção.



Cidade planejada

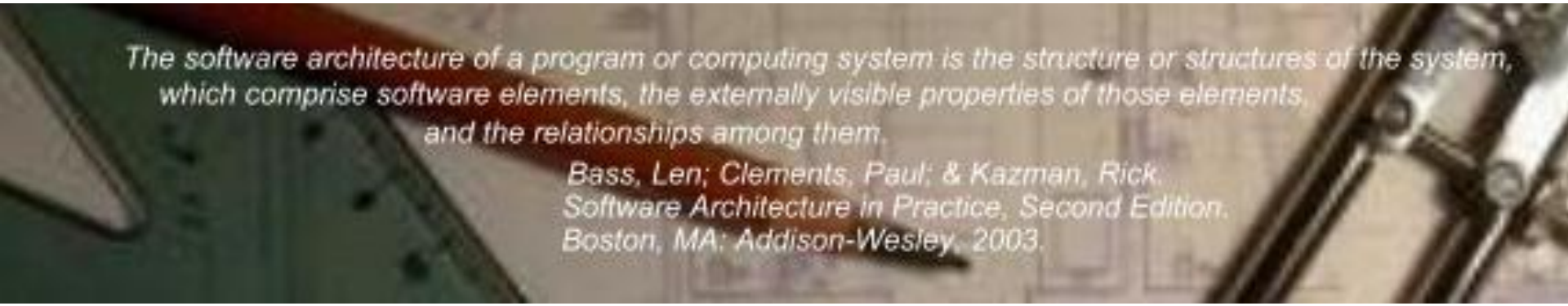
Equipe multidisciplinar, vários níveis de planejamento, execução integrada

O que é diferente, em termos do projeto de arquitetura?

- Porte
- Riscos do projeto
- Stakeholders
- Custo
- Perfil das equipes envolvidas
- Ferramentas
- Processo
- Planejamento
- Necessidade de documentação
- Aderência a normas e padrões

O que é arquitetura de software

- É o conjunto formado pelas partes integrantes de um software e pelas relações entre essas partes e delas com outros softwares.
- A arquitetura expressa também as decisões de projeto de alto nível:
 - Escolha dos elementos estruturais e comportamentais que compõem o software;
 - Decisões pelo estilo a ser adotado para uma solução
 - Compatibilidade com outras soluções de uma corporação ou família de produtos.



The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.

*Bass, Len; Clements, Paul; & Kazman, Rick.
Software Architecture in Practice, Second Edition.
Boston, MA: Addison-Wesley, 2003.*

<http://www.sei.cmu.edu/architecture/index.html>

Definições

- “Architecture is the organizational structure of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems” (UML 1.3).
- “The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them” (Bass et al. Software Architecture in Practice, Addison-Wesley 1997).

Características da arquitetura

- A arquitetura define os **componentes** principais do software
- A arquitetura evidencia as relações/interfaces entre os componentes
- A arquitetura esconde detalhes dos componentes que não estejam relacionados com as interfaces entre eles
- A arquitetura é pensada desde cedo no projeto
 - Ela representa decisões críticas do projeto
 - Depois de definida, é difícil alterar
 - Tem impacto direto na qualidade do sistema

Requisitos não funcionais condicionam a arquitetura

Segurança de acesso

Privacidade

Desempenho

Usabilidade

Disponibilidade

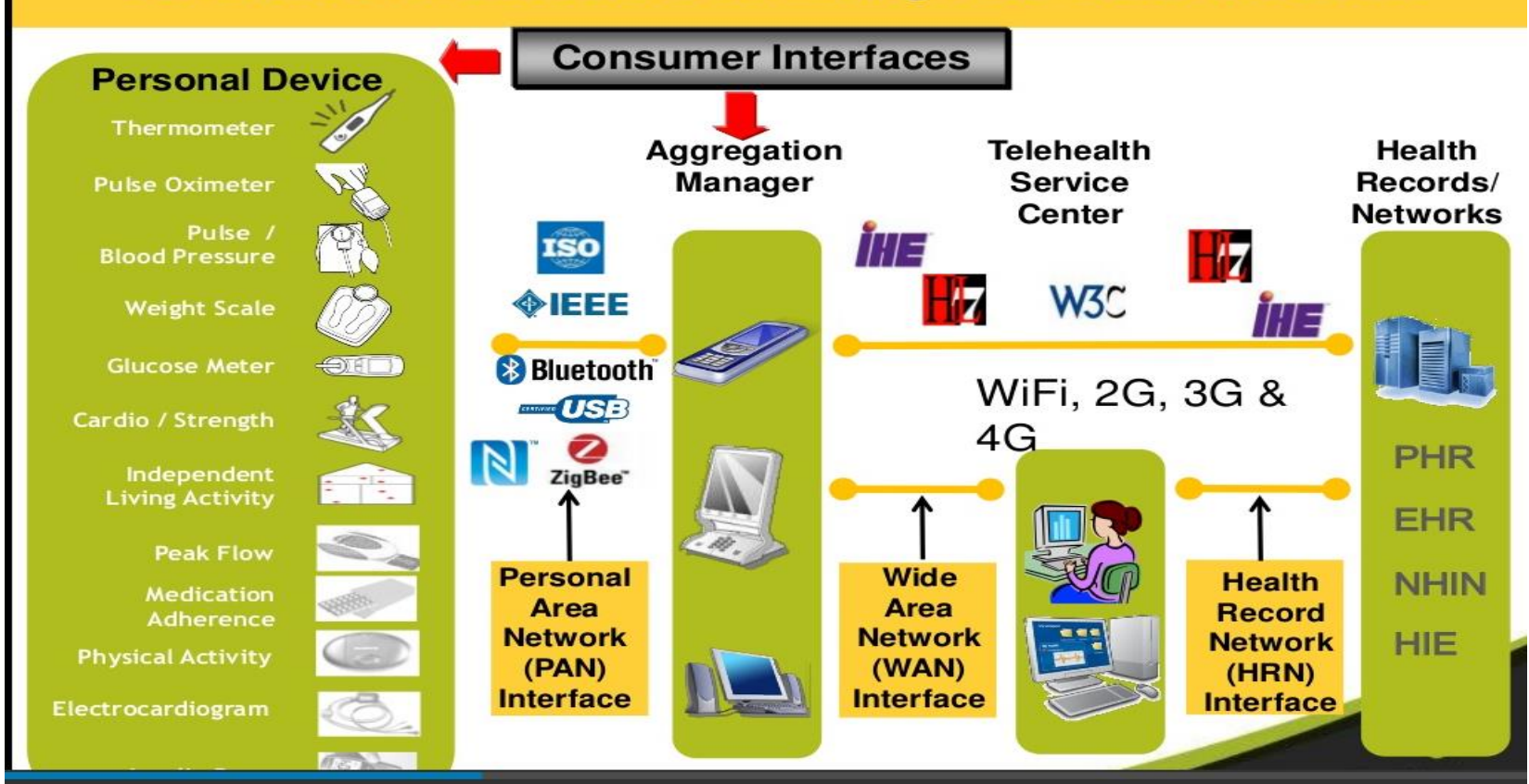
Arquiteturas de referência

- Uma arquitetura que abrange o conhecimento sobre como projetar arquiteturas concretas de sistemas de um domínio de aplicação específico. Deve ser apoiada por uma terminologia de domínio que seja amplamente entendida, que não seja ambígua e unificada [Nakagawa, 2011]
- Provê regras claras para construção de componentes, reuso e integração.

Exemplo

ected Health

Continua's Architecture: Enabling PCH at the Interface



The background features a large, solid green circle on the right side. To its left, there is a black, crescent-shaped graphic element. The entire composition is set against a white background with several thin, light gray curved lines and dashed lines that create a sense of motion and depth.

Estilos de arquitetura

Categories of Architectural Styles

- Hierarchical Software Architecture
 - Layered
- Distributed Software Architecture
 - Client Server
 - SOA
- Data Flow Software Architecture
 - Pipe n Filter
 - Batch Sequential
- Event Based Software Architecture
- Data Centered Software Architecture
 - Black board
 - Shared Repository
- Interaction-Oriented Software Architectures
 - Model View Controller
- Component-Based Software Architecture

Arquitetura em camadas (n-tier)

Separação de responsabilidades -> mudanças são isoladas

Cada camada se comunica apenas com as adjacentes

Útil para sistemas interativos

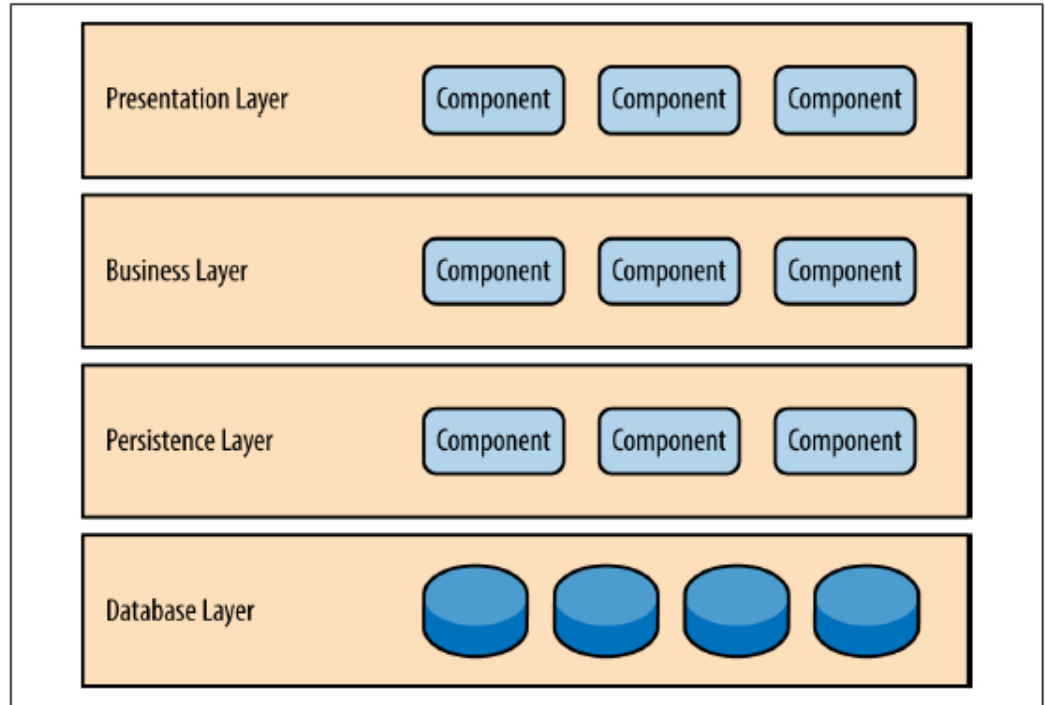


Figure 1-1. Layered architecture pattern

Exemplo

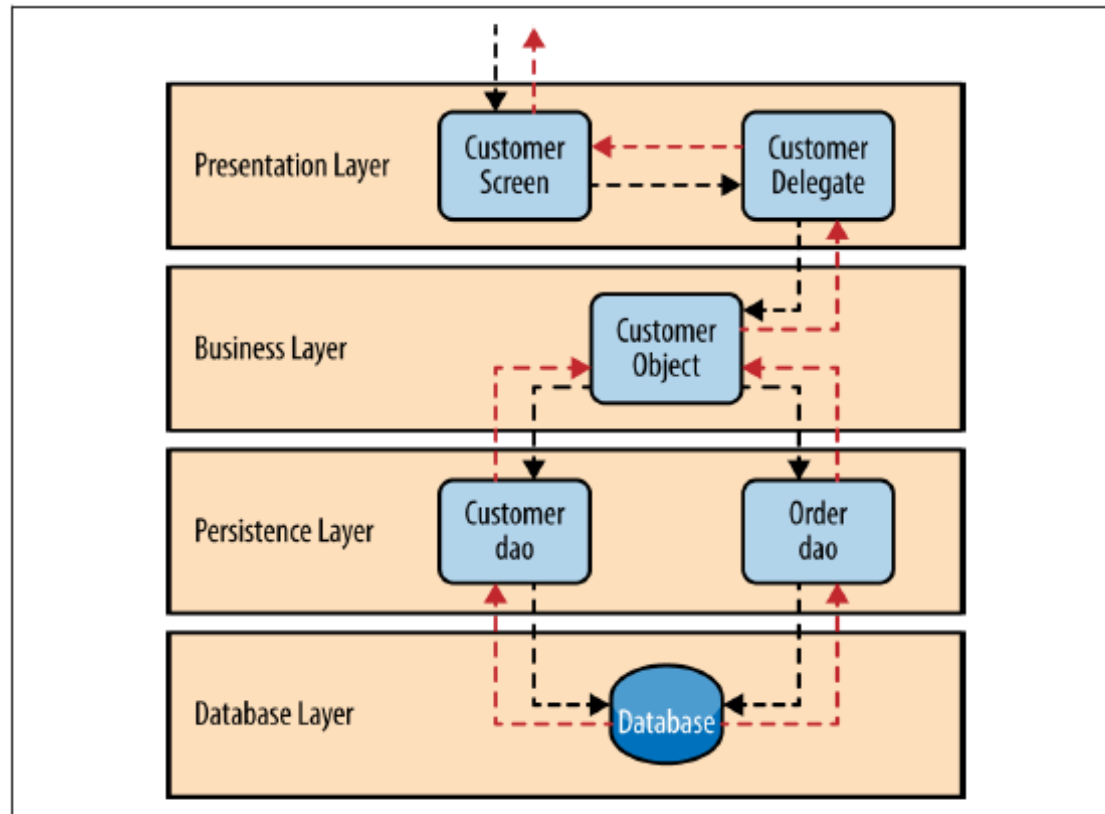
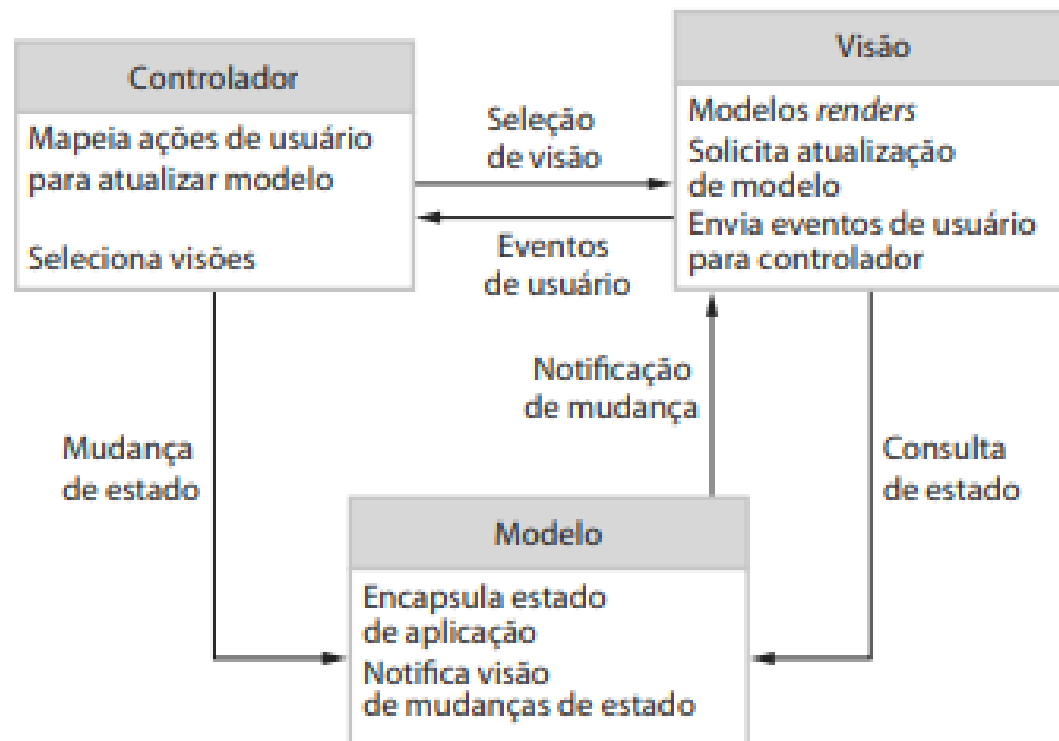


Figure 1-4. Layered architecture example

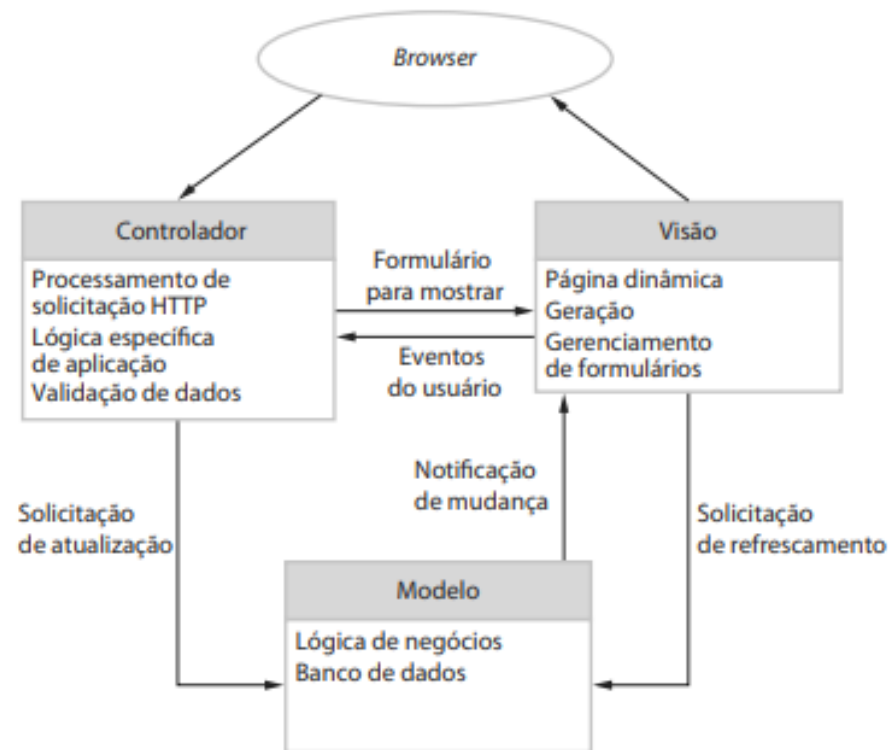
MVC (Sommerville)

A organização do MVC



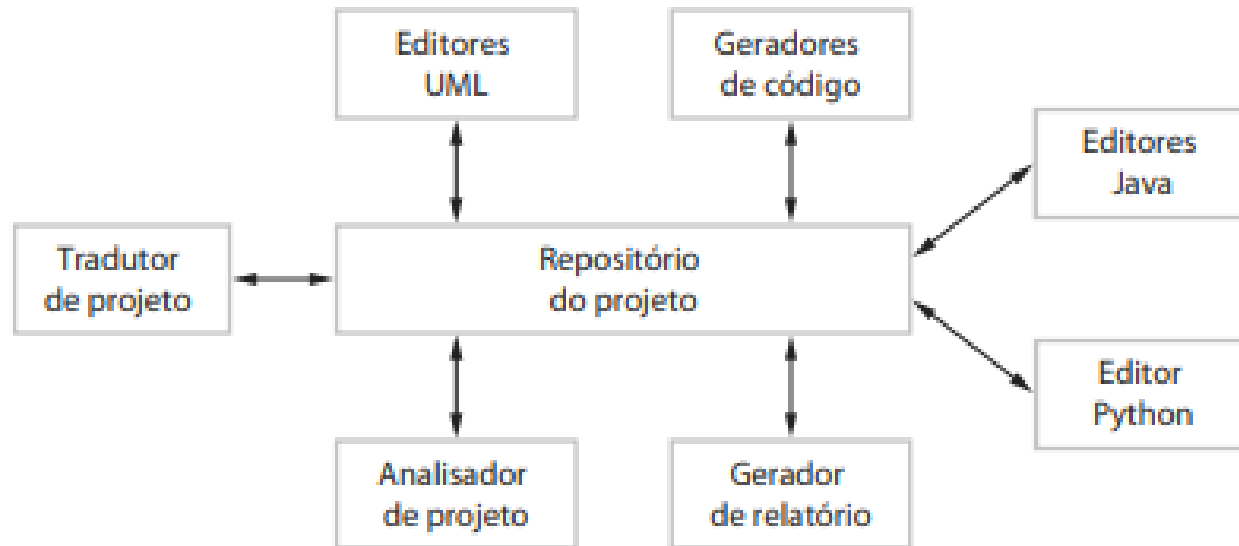
MVC para Web (Sommerville)

Arquitetura de aplicações Web usando o padrão MVC

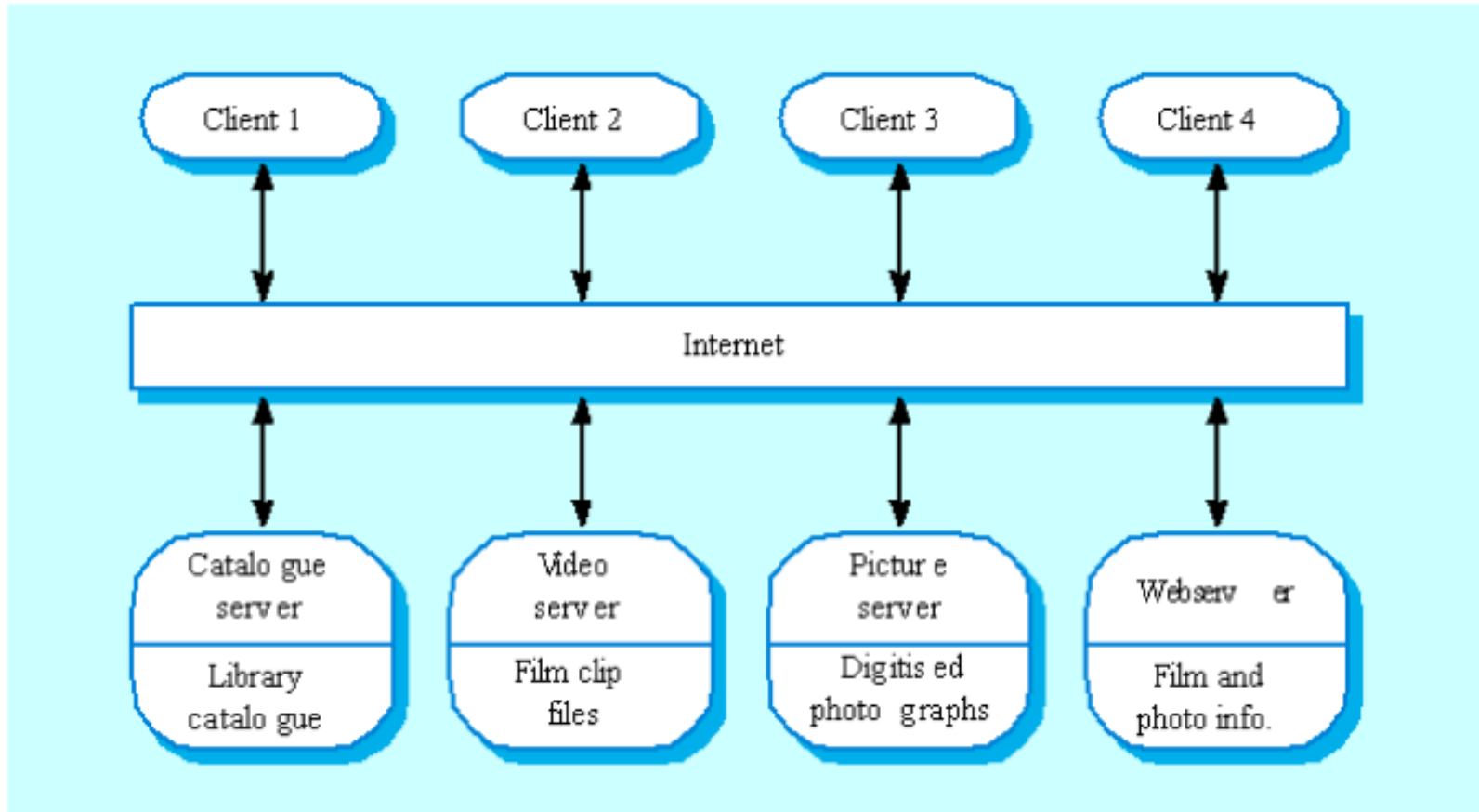


Arquitetura de repositório

Uma arquitetura de repositório para um IDE



Arquitectura cliente-servidor



Arquitetura de microserviços

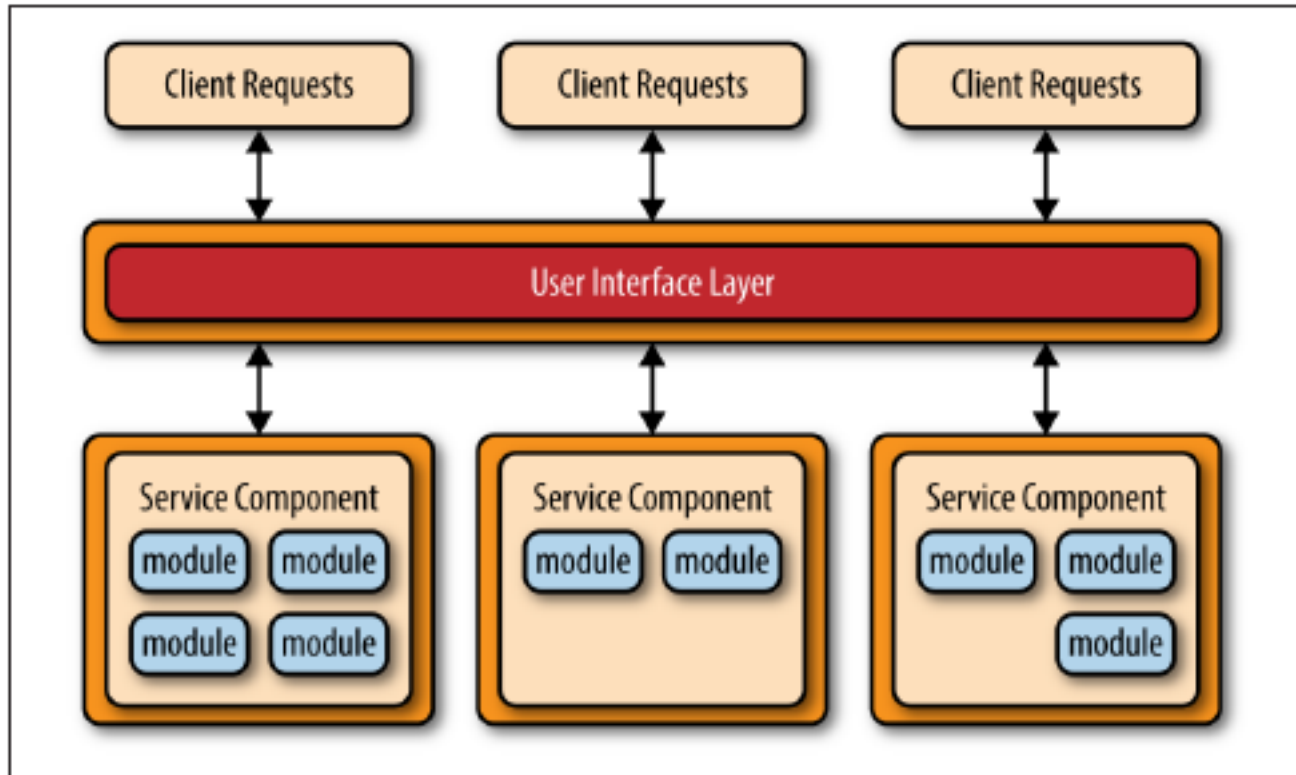


Figure 4-1. Basic Microservices architecture pattern

Arquitectura microkernel

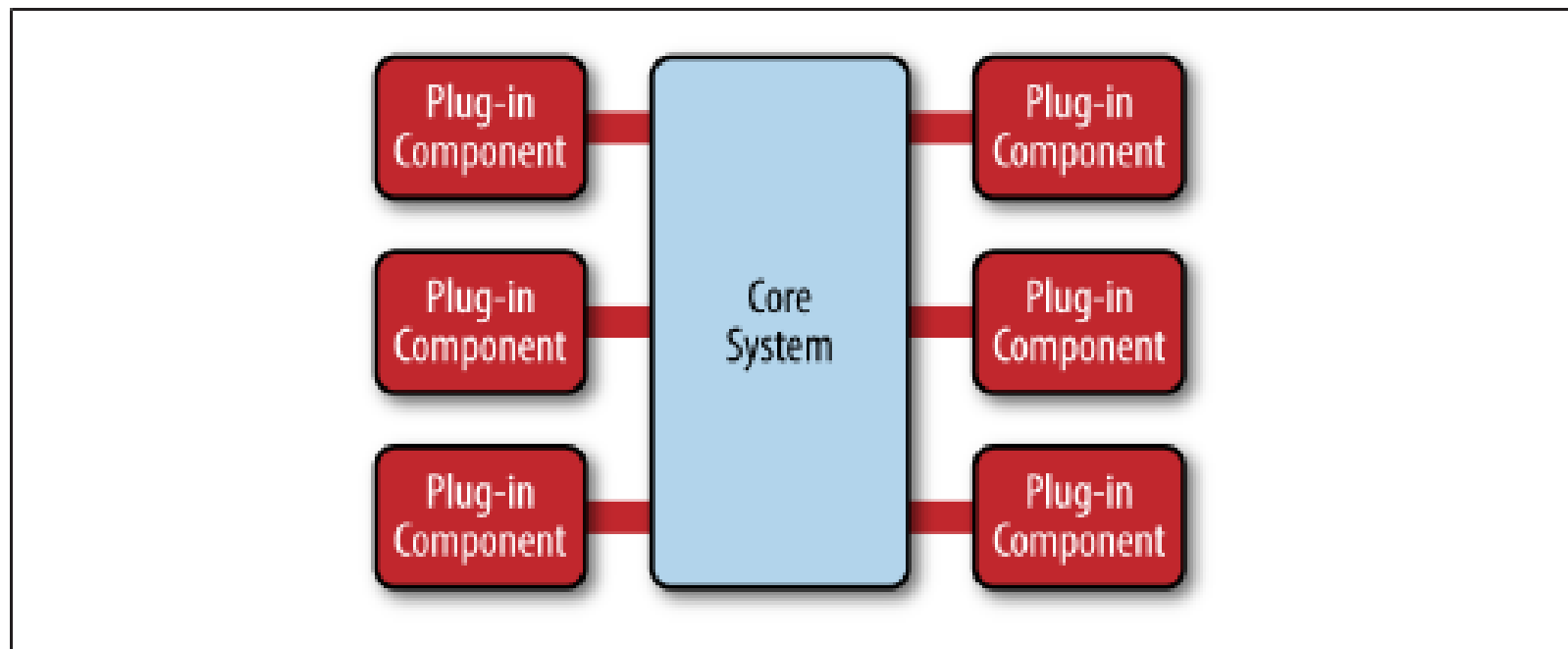


Figure 3-1. Microkernel architecture pattern



Qual arquitetura vamos
usar no laboratório?

Pesquisa

- https://docs.google.com/forms/d/1JKrUiMfJMPg3usRE328BVJ_ooeHfvKd6_f4Brp60nGs/edit#responses

Seleção de tecnologias

Front end

- Linguagens: Javascript ([ES6](#)), [HTML5](#), [CSS](#)
- Bibliotecas: [React 16.4](#), [React Semantic-UI](#)

Back-end

- Linguagens: Python 3.6
- Bibliotecas: [Flask 1.02](#), [FlaskSqlAlchemy](#) (Sql)