

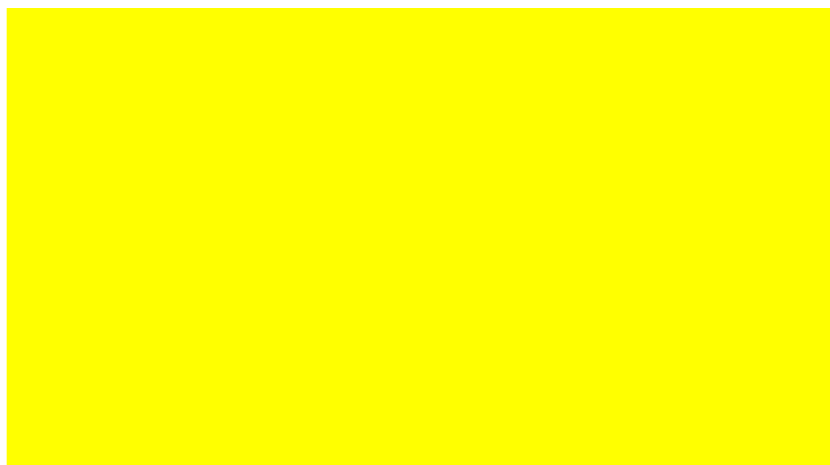
Introdução à Arquitetura de Software

Fernando Castor e Paulo Borba



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO

Programação Modular

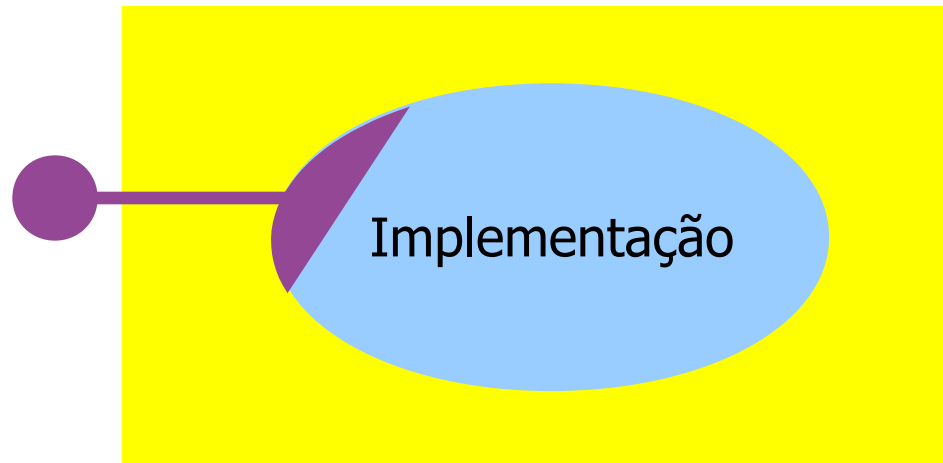


Programação Modular



Implementação

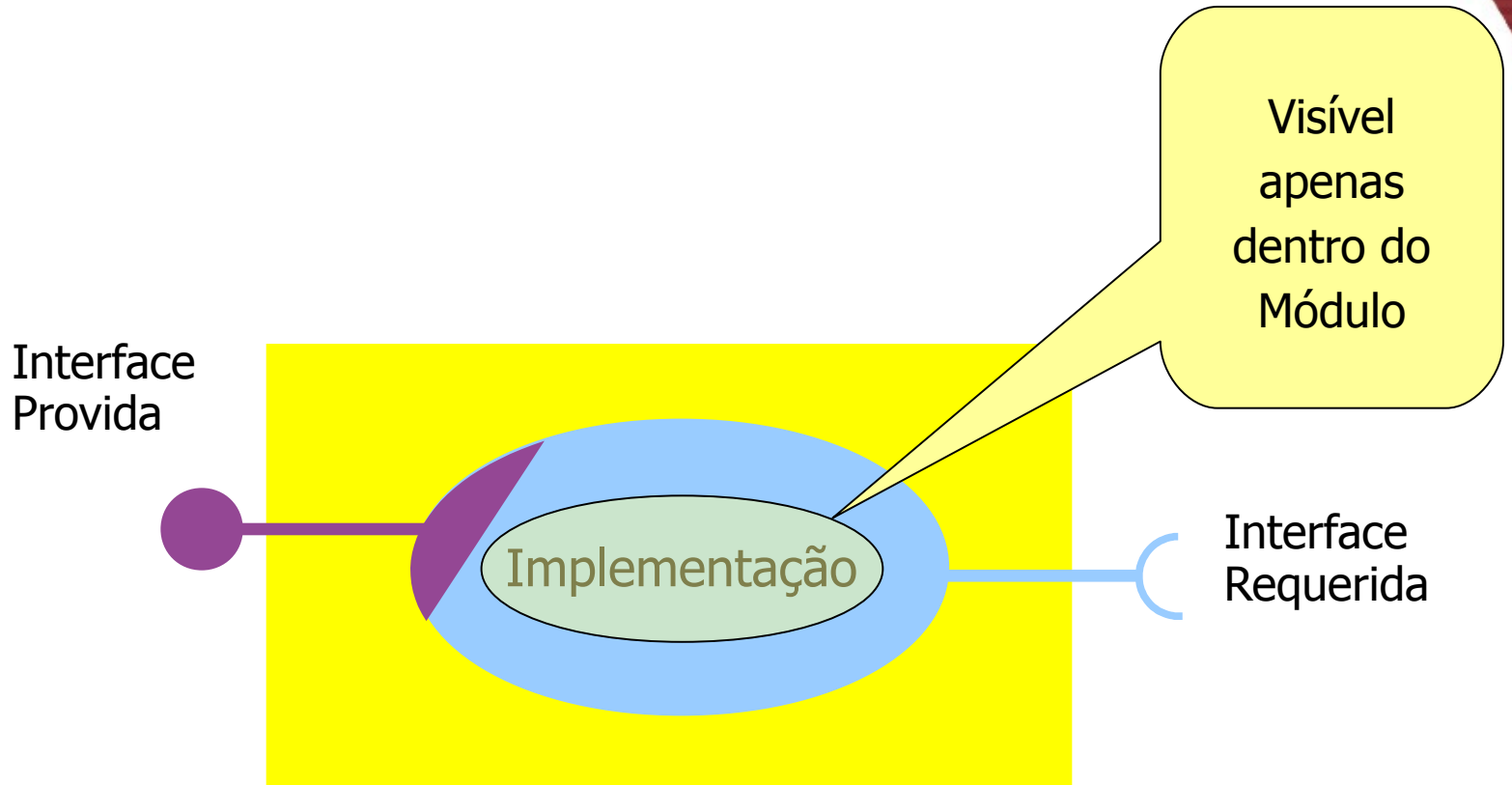
Programação Modular



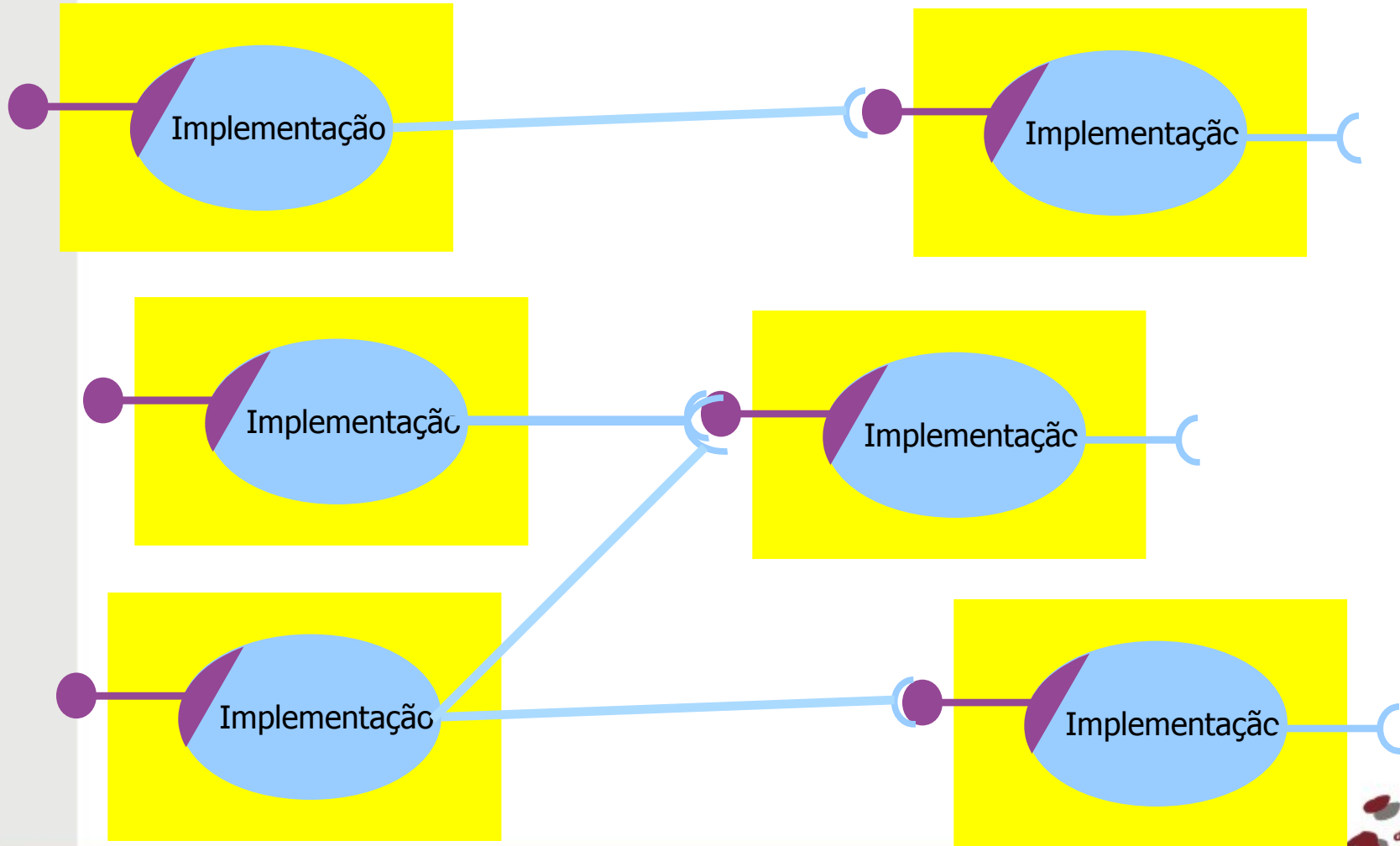
Programação Modular



Programação Modular



Programação Modular



Benefícios Esperados da Programação Modular

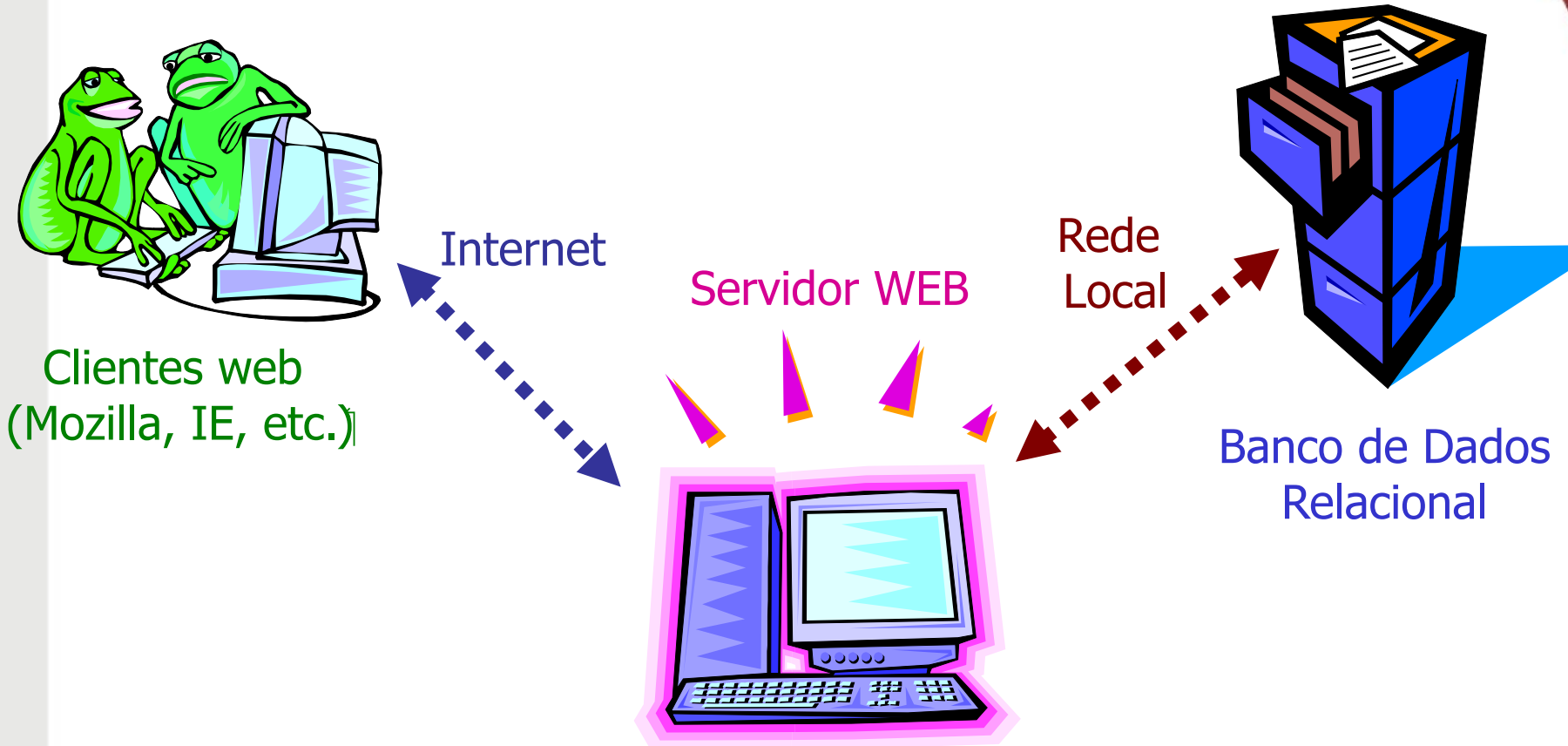
[Parnas, 1972]

- (1) Tempo de desenvolvimento encurtado
 - (2) Possibilidade de aplicar mudanças drásticas a um módulo sem a necessidade de mudar outros
 - (3) Organização das tarefas e da equipe de desenvolvimento
 - (4) Possibilidade de estudar o sistema olhando para um módulo de cada vez
- ✓ Interações entre módulos

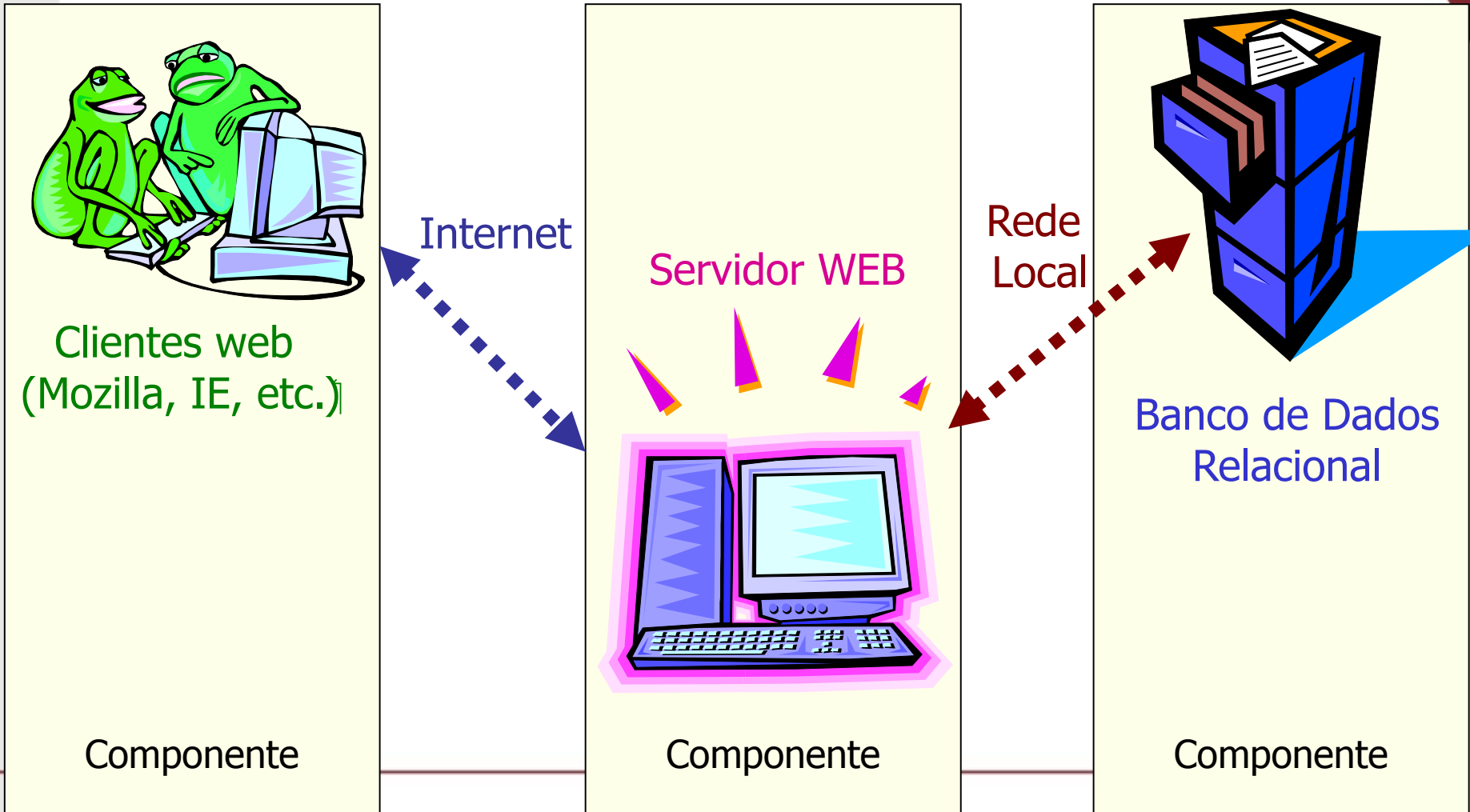
Arquitetura de Software

- A **estrutura** de um sistema de software, que engloba
 - elementos de software;
 - suas propriedades visíveis externamente;
 - e os relacionamentos e interações entre eles
- As primeiras **decisões** tomadas no **projeto** de um sistema
 - **As mais importantes!**
- Uma arquitetura de *software* é composta por **componentes** e **conectores**

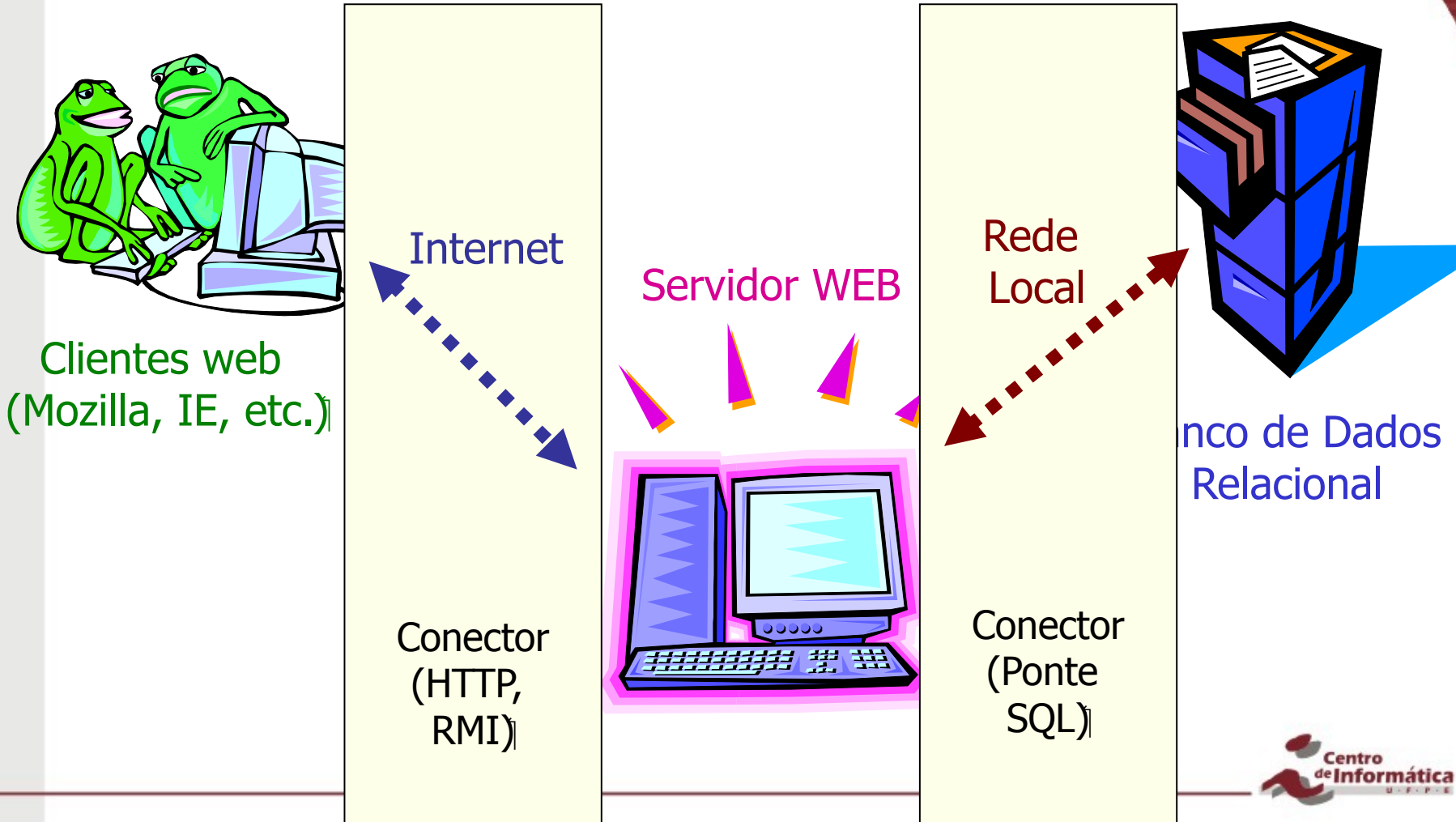
Uma Arquitetura 3-tier



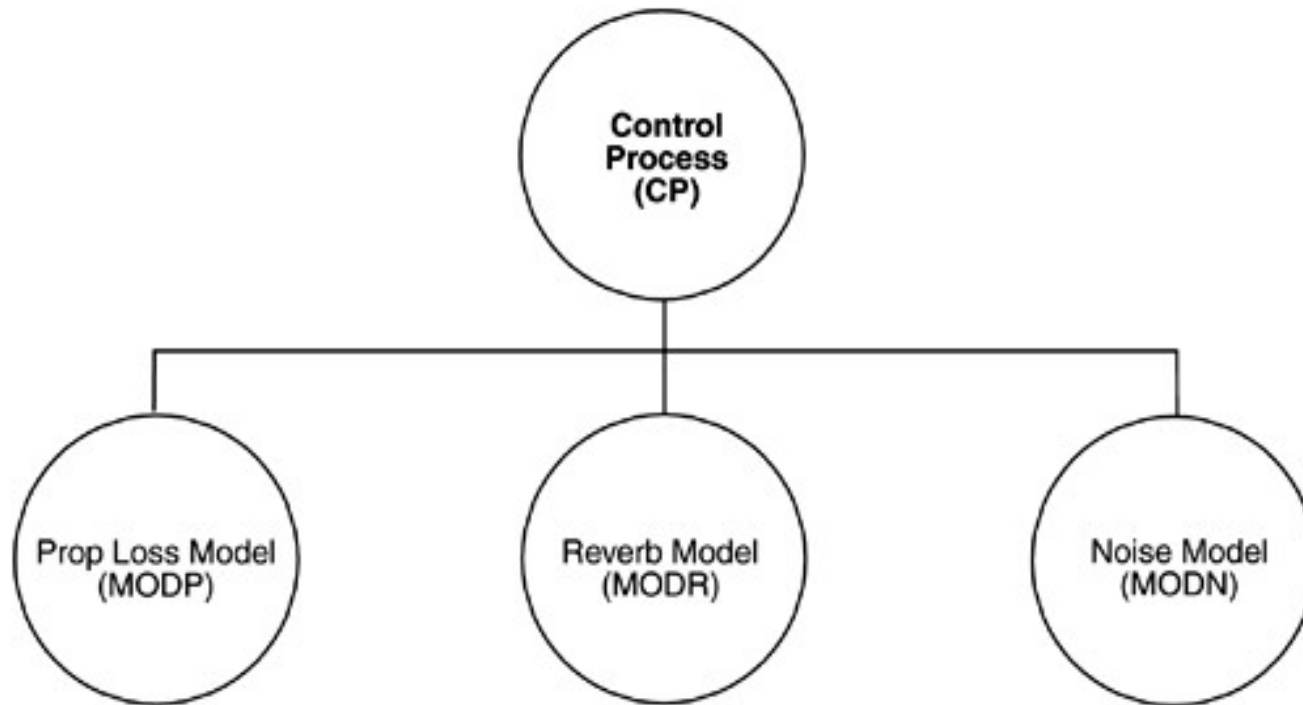
Uma Arquitetura 3-tier



Uma Arquitetura 3-tier



O que é Arquitetura de Softw. e o que não é?



- O que é possível inferir a partir da figura?
- O que **não é**?

O que é Arquitetura de Softw. e o que não é?

- Arquitetura define elementos de software
 - Foco no **comportamento externamente observável**
 - E interações
 - Responsabilidades** das partes
 - Abstrações** de elementos de software
- **Todo** sistema tem uma arquitetura
- Indica como requisitos são satisfeitos

- **Detalhes** de implementação não entram
- Algoritmos **normalmente** não afetam a arquitetura
- Tudo que está lá deve estar lá por um **motivo**

Projeto Arquitetural

- O processo de projeto que estabelece
 - Os **subsistemas** que constituem um sistema
 - A maneira como esses subsistemas **interagem**
- Incluindo algumas decisões tecnológicas
 - Ex. Plataforma de componentes, SGBD
- A saída desse processo de projeto é uma descrição da **arquitetura de software**.
- A arquitetura de software lida com os **requisitos** do sistema
 - Grande ênfase nos **não-funcionais**

Projeto Arquitetural

É o primeiro estágio do projeto do sistema

A ligação entre os processos de especificação e de projeto detalhado

Freqüentemente conduzido **em paralelo** às atividades de especificação

Envolve a identificação dos componentes principais do sistema e sua interação

Componentes => unidades de **modularidade**

Vantagens de uma Arquitetura Explícita

Análise de requisitos não-funcionais

Comunicação com os *stakeholders*

Organização de equipes e tarefas

Pode ser usada para fazer estimativas

Diversas vantagens da programação modular

Reuso em larga escala

Suas **partes** também

Algumas soluções são **universalmente aceitas**

Conflitos de Arquitetura

Componentes de alta granularidade aprimoram o desempenho mas diminuem a **facilidade de manutenção**

A introdução de dados redundantes aprimora a disponibilidade, mas torna a **proteção** mais **difícil**

Localizar as funcionalidades críticas de segurança em poucos locais pode criar **gargalos de desempenho**

Escolhas/Decisões de projeto

Projeto Arquitetural é um Processo Criativo!

- Cada sistema envolve diferentes decisões/requisitos/conflitos/restrições
- Envolve solucionar os problemas representados pelos requisitos
- Novas soluções vs. Soluções pré-existentes
- **Rationale** => justificativas para as escolhas

Escolhas de Projeto

- Escolhas feitas durante o projeto de um sistema
Precisam ser **justificadas**
- Afetam sua capacidade de fornecer seu serviço
- Normalmente resultam em compromissos
- É importante avaliar as opções existentes
- Não estão restritas ao projeto arquitetural!
- É importante documentar as opções rejeitadas

Alguns Exemplos de Perguntas Interessantes

Como representar o mapa em um sistema que traça rotas percorridas por ônibus **de modo a minimizar o trabalho da equipe?**

Como garantir a confiabilidade de um servidor **a um baixo custo?**

Qual a maneira mais eficiente de se construir uma grade de horários levando-se em conta as várias restrições impostas por professores, diretores e regras departamentais?

Como tornar o sistema capaz de lidar com 1.000.000 de usuários simultâneos **sem sobrecarregar a rede?**

Características de Um Sistema Atreladas à sua Arquitetura

- Desempenho
- Proteção (*security*)
- Segurança (*safety*)
- Disponibilidade
- Confiabilidade
- Escalabilidade
- Facilidade de manutenção

E muitas outras...

Estilos Arquiteturais

- Capturam conjuntos de escolhas de projeto comuns a vários sistemas
- Podem representar interações como elementos de primeira ordem
 - **Conectores**
- Exemplos comuns:
 - Decomposição Modular
 - Implantação
 - Cliente/Servidor
 - Camadas
- Associados à representação da arquitetura

Uma Arquitetura em Camadas

GUI

<<pode usar>>

Negócios

<<pode usar>>

Acesso a Dados

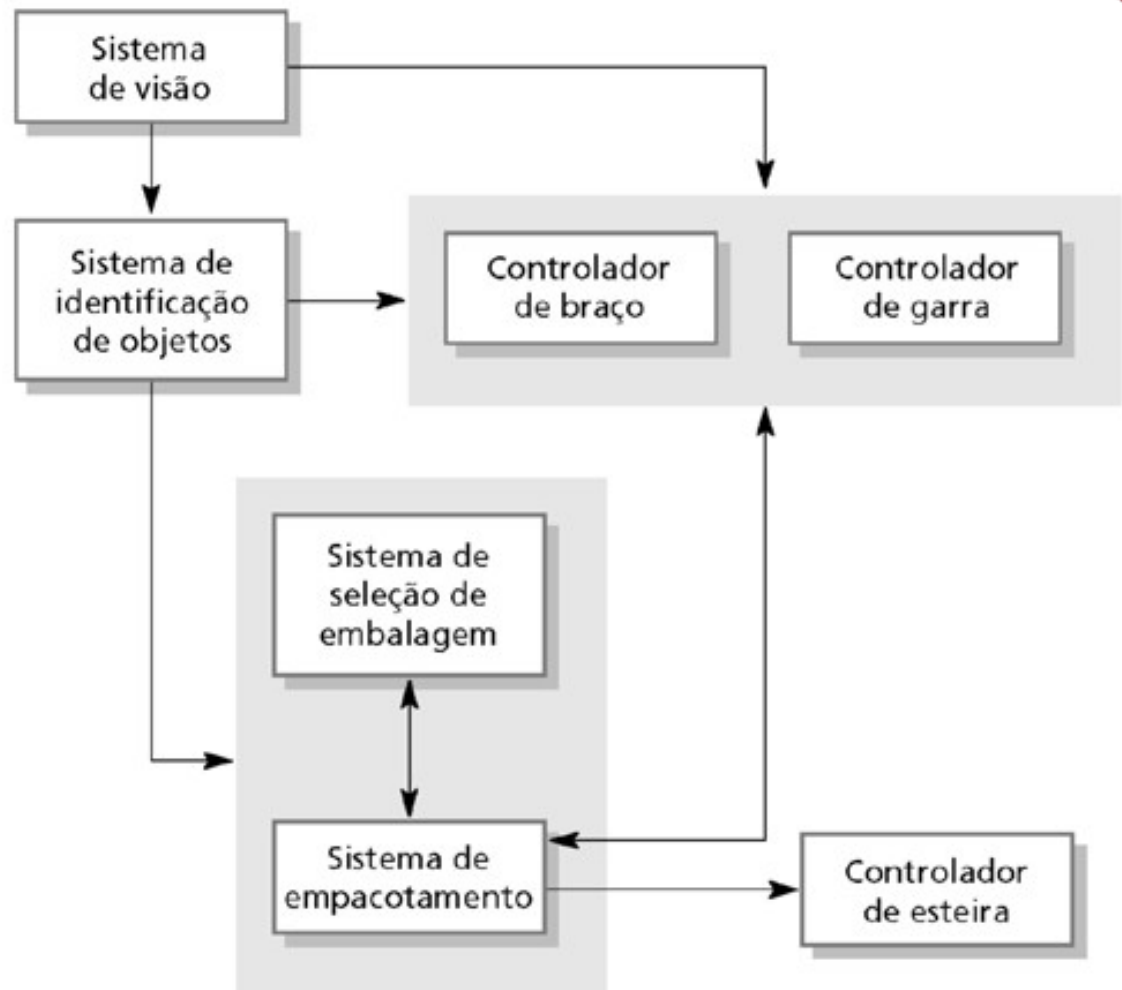
Representação da Arquitetura

- A arquitetura é um ativo importante
 - Pode ser a diferença entre o sucesso e o fracasso
- Representá-la é importante
 - Torna possível "falar" sobre ela
 - O projeto de arquitetura é normalmente expresso como um diagrama de blocos
- Modelos mais específicos também podem ser desenvolvidos.

Sistema de Controle Robotizado de Empacotamento

Figura 11.1

Diagrama de blocos de um sistema de controle robotizado de empacotamento.



Diagramas caixa e linha

Muito abstratos

Não mostram a natureza dos relacionamento de componentes

Nem suas propriedades externamente visíveis

Contudo, são úteis para comunicação com os stakeholders e para planejamento

Alternativas:

Notações formais (Linguagens para Descrição de Arquiteturas)

Notações informais mais organizadas

Visões Arquiteturais

A arquitetura de um sistema software normalmente é representada através de várias **visões**

Visões são maneiras diversas de se enxergar uma mesma arquitetura

Enfocando diferentes **aspectos de interesse**

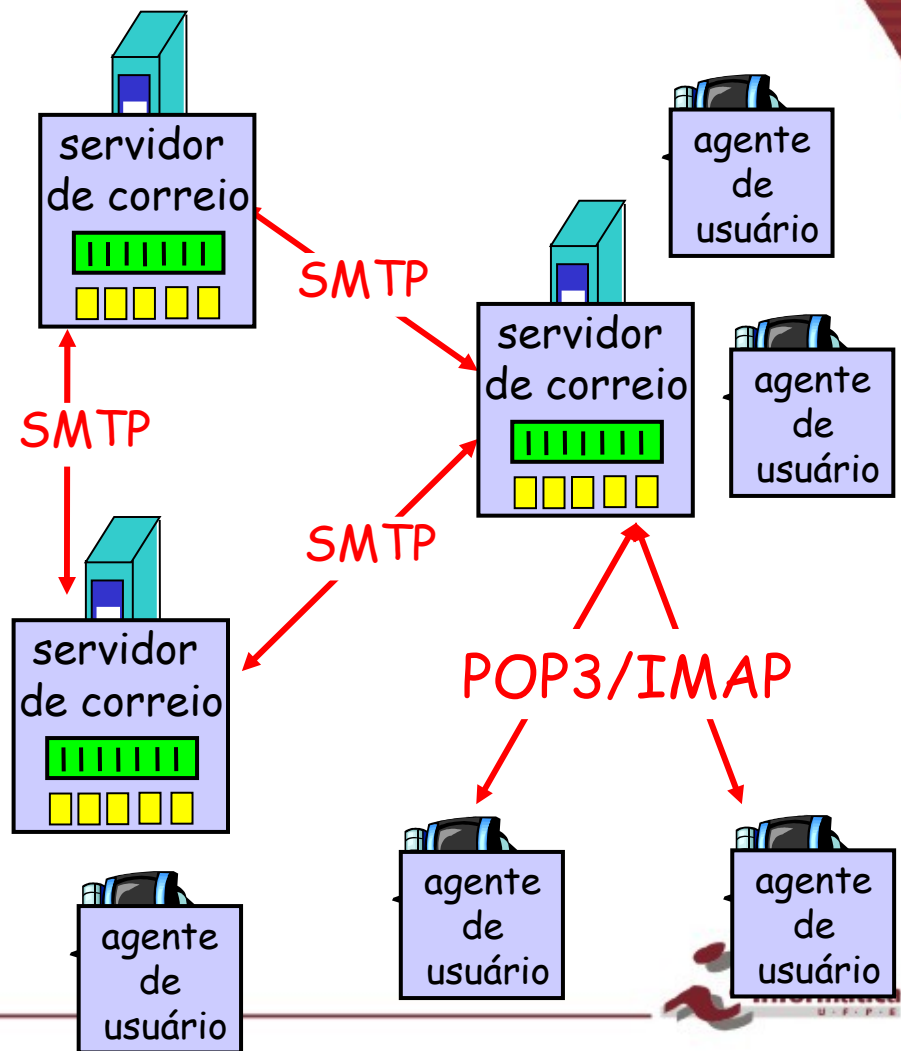
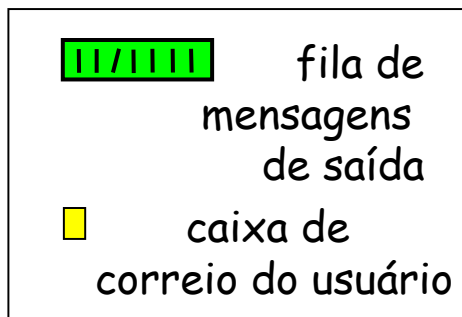
Ex.: as várias plantas de uma casa

Arquiteturas de software são especificadas através de várias visões

Correio Eletrônico – Visão 1

Três principais elementos:

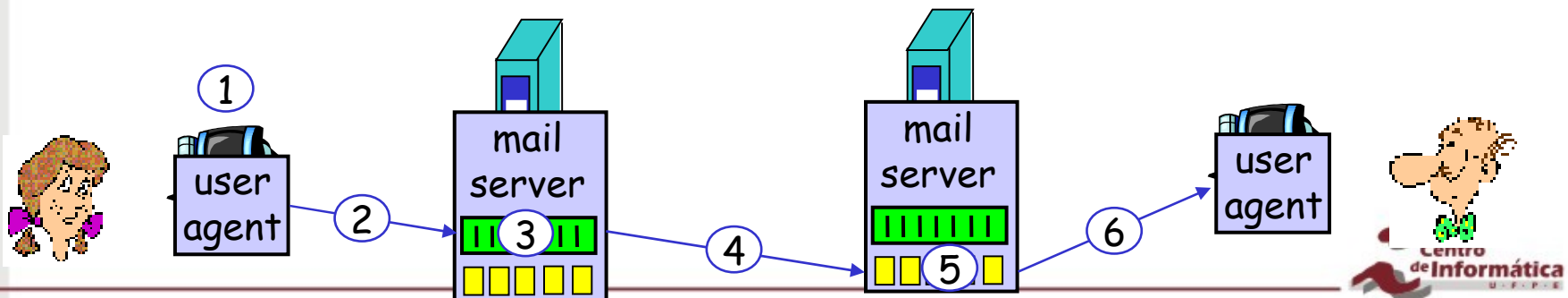
- ❑ agentes de usuário (UA).
- ❑ servidores de correio.
- ❑ simple mail transfer protocol: SMTP.



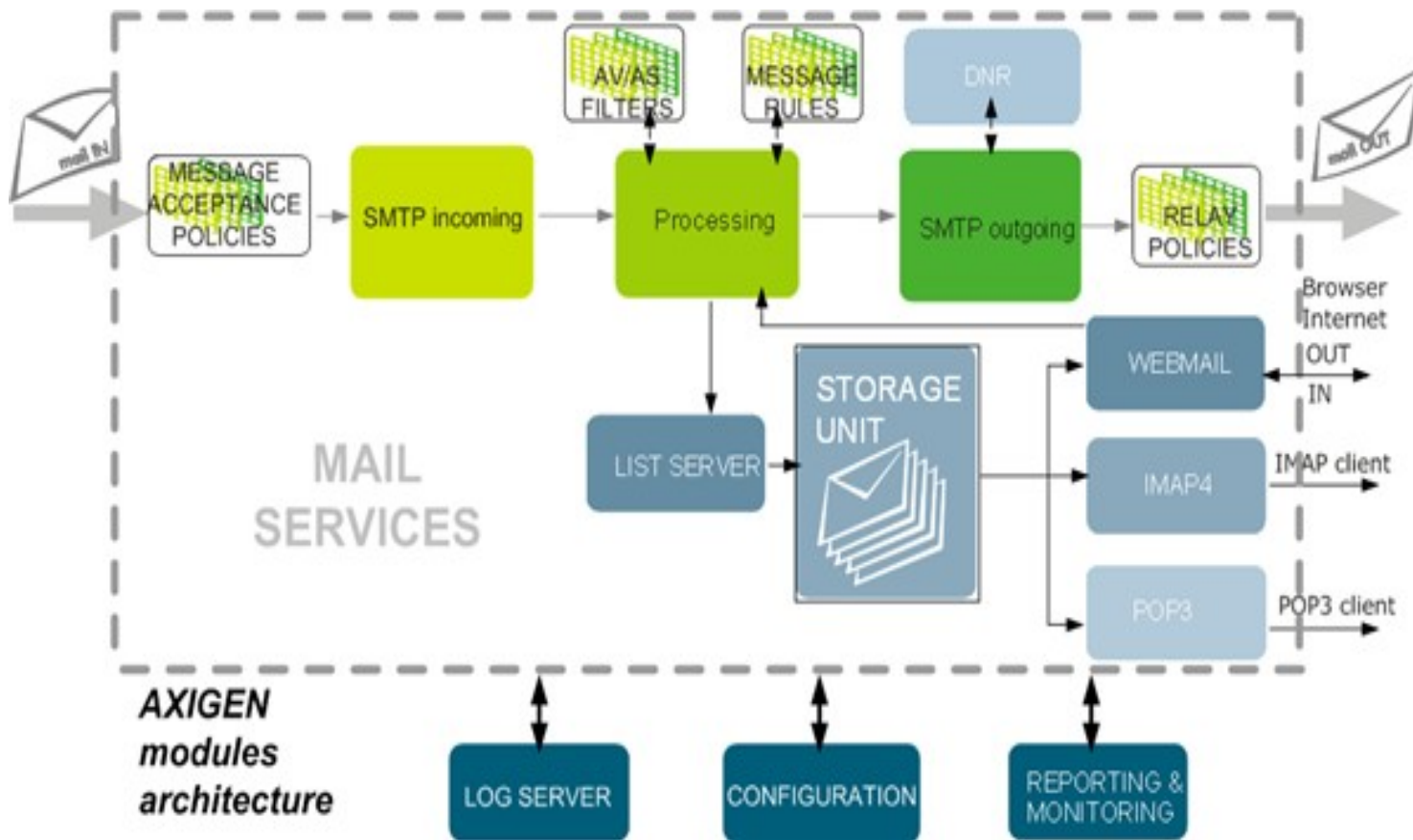
Correio Eletrônico – Visão 2

- 1) Alice usa o UA para compor uma mensagem “para” bob@someschool.edu
- 2) O UA de Alice envia a mensagem para o seu servidor de correio; a mensagem é colocada na fila de mensagens.
- 3) O lado cliente do SMTP abre uma conexão TCP com o servidor de correio de Bob.

- 4) O cliente SMTP envia a mensagem de Alice através da conexão TCP.
- 5) O servidor de correio de Bob coloca a mensagem na caixa de entrada de Bob.
- 6) Bob chama o seu UA para ler a mensagem.

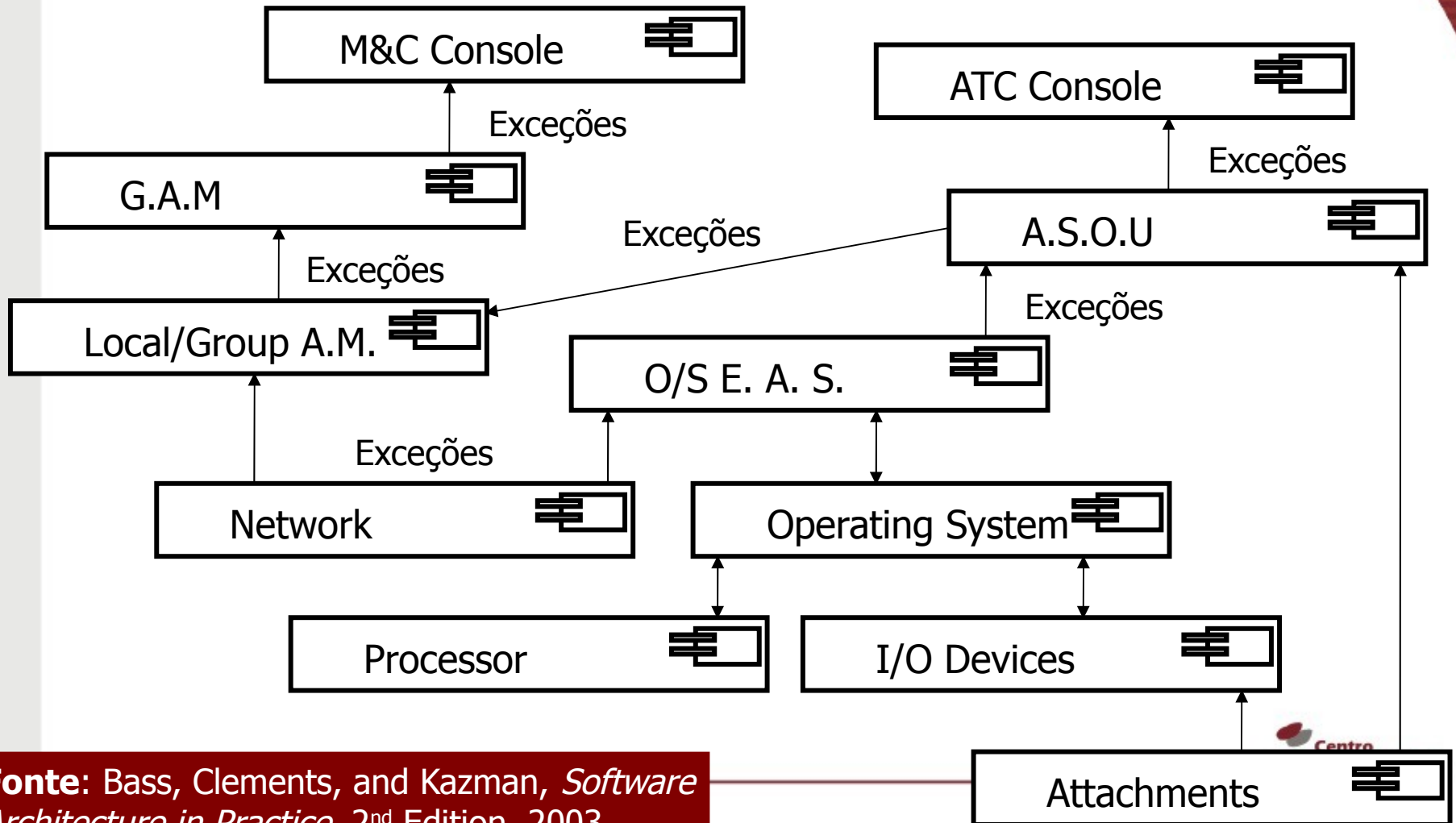


Correio Eletrônico – Visão 3



Fonte: *Axigen Mail Server Documentation - Mail Server Architecture.*
Consultado em 24 de março de 2008
http://www.axigen.com/docs/en/Mail-Server-Architecture_85.html

Um Exemplo de Sistema de Controle de Tráfego Aéreo



Fonte: Bass, Clements, and Kazman, *Software Architecture in Practice*, 2nd Edition, 2003.

Reuso de arquitetura

Sistemas do mesmo domínio freqüentemente têm arquiteturas similares

Resultam em **escolhas de projeto similares**

Reuso de arquiteturas acontece de diversas maneiras:

Conhecimento informal sobre escolhas de projeto

Estilos/Padrões arquiteturais

Frameworks e servidores de aplicação

Linhas de produtos

Obrigado!

Dúvidas e comentários:

castor@cin.ufpe.br

phmb@cin.ufpe.br