

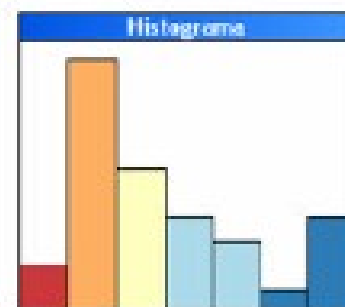
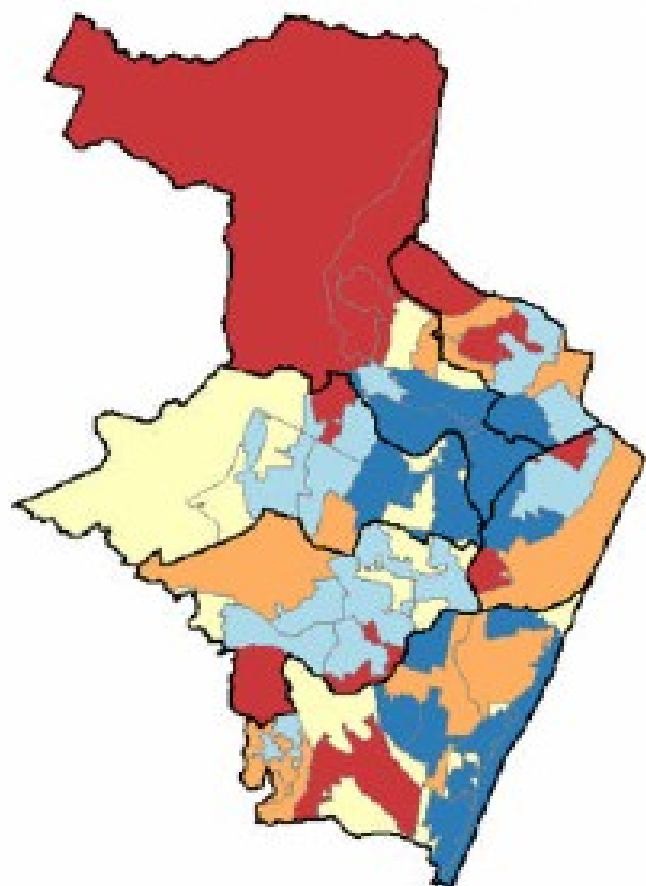
Documentação de Arquiteturas de Software

Fernando Castor e Paulo Borba



Fonte: <http://www.recife.pe.gov.br/cidade/projetos/mapas/mapa0.jpg>

FIG. 1 - RECIFE - Índice de Desenvolvimento Humano Municipal, por Unidades de Desenvolvimento Humano - 2000



Legenda

0.692 a 0.709	(12)
0.700 a 0.725	(12)
0.726 a 0.765	(12)
0.766 a 0.804	(13)
0.805 a 0.864	(13)

Fonte: RECIFE.Prefeitura; PNUD. Atlas do Desenvolvimento Humano no Recife. 2005.



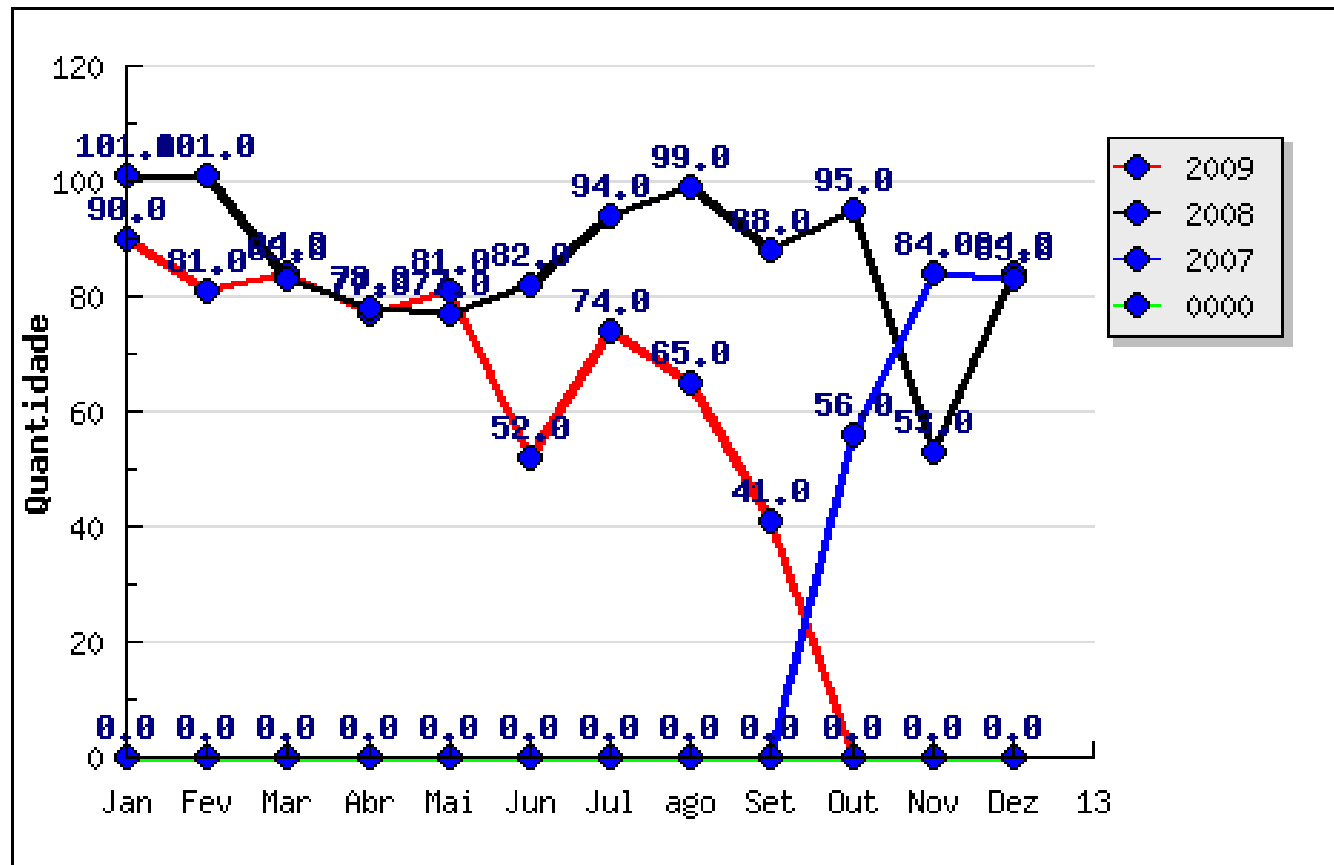
Fontes: <http://banco.agenciaoglobo.com.br/Imagens/Preview/200803/d0a553a0-ce18-4dc4-97e7-79c058db89c9.jpg> e
http://imagens.portaisdamoda.com.br/gal83_13146lis.jpg



Fonte: <http://www.recife.pe.gov.br/cidade/projetos/mapas/index.html>

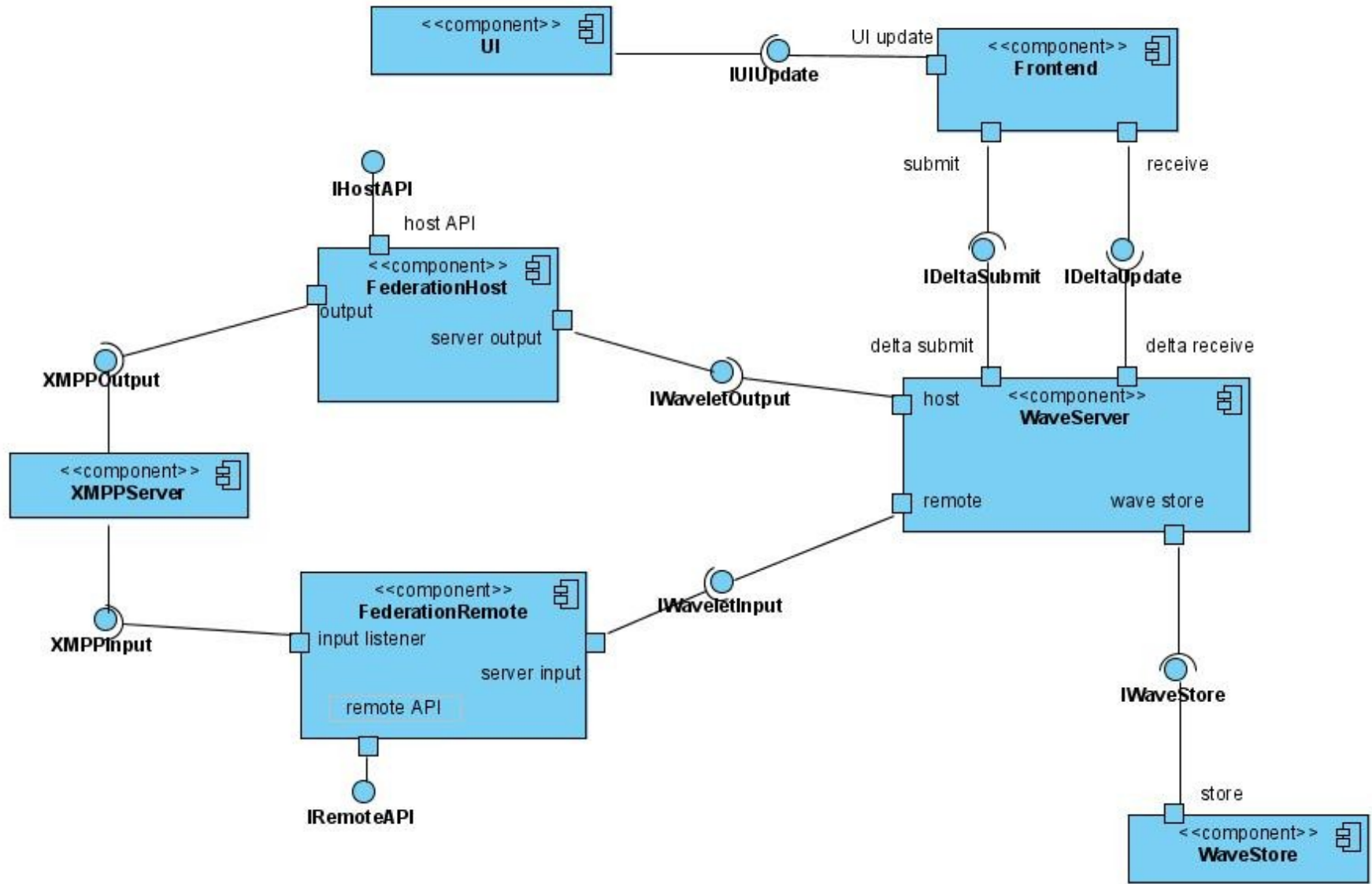


Fonte: <http://www.transportes.gov.br/bit/estados/port/pe.htm>

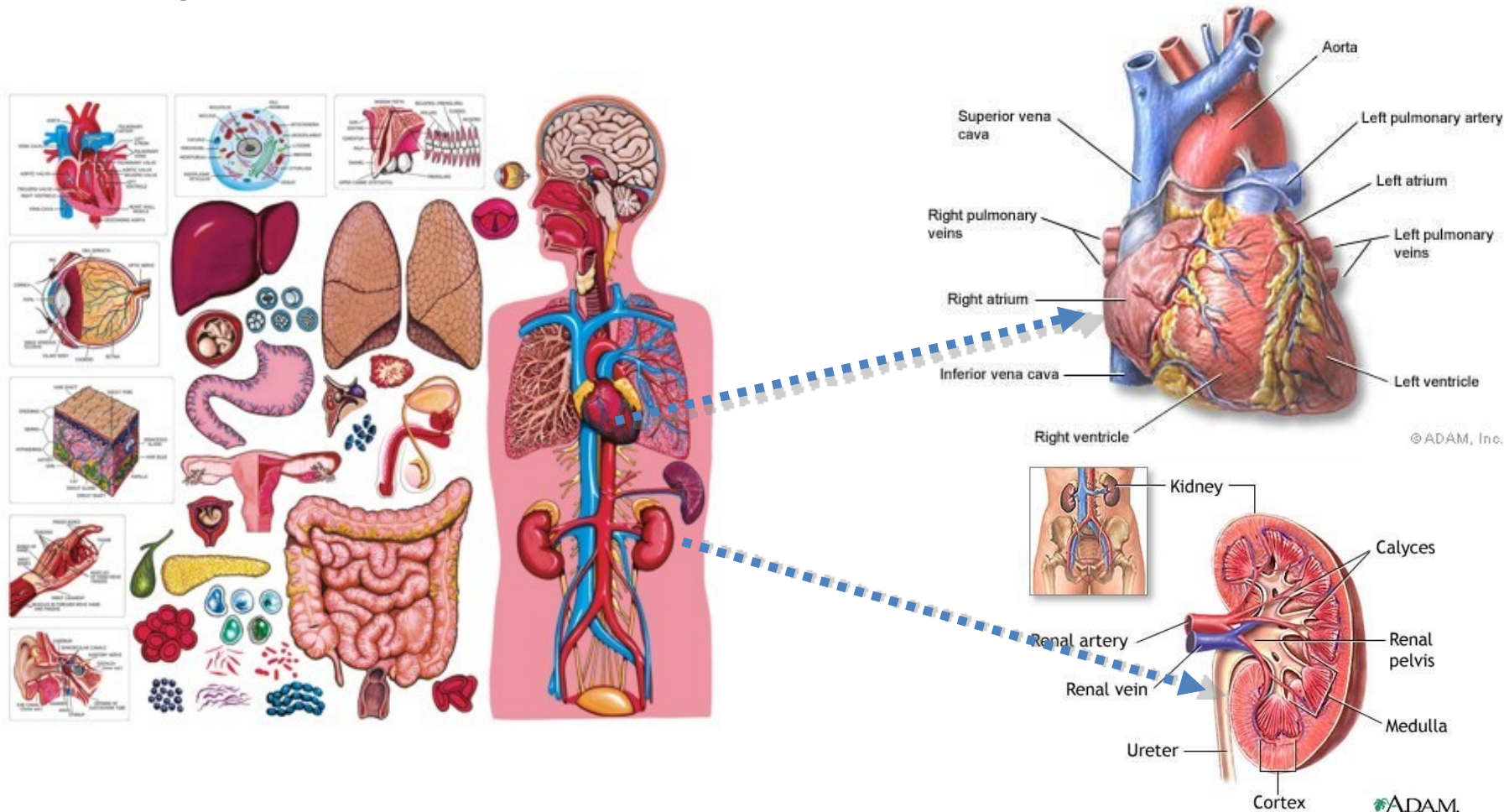


Fonte:

<http://www.pebodycount.com.br/ocorrencia/ocorrenciaGrafico.php>



É difícil entender um sistema complexo em sua totalidade!



Fonte: <http://www.learningwonders.com/cart/images/T/humanbody04.jpg>

Sistemas de software modernos
são extremamente complexos

As arquiteturas de tais sistemas
refletem essa complexidade

Como documentá-las?

Documentação de Arquiteturas

- Documenta-se **visões**
 - Não se documenta a arquitetura “inteira”
- A escolha das visões a documentar depende
 - Da informação que se deseja documentar/comunicar
 - Das análises que serão feitas
 - Do público alvo

End users

- functionality

Logical view

Programmers

- software management

Development view

Scenarios

Process view

Physical view

System integrators

- performance
- scalability
- throughput

System engineers

- system topology
- delivery
- installation
- telecommunication

Por que Documentar a Arquitetura?

- Engenheiros de Requisitos
- Projetistas dos Módulos
- Testadores
- Gerentes
- Clientes
- Novos arquitetos
- Responsáveis por manutenção
- Desenvolvedores de outros sistemas



O que Documentar?

- Visões
 - Forças/Requisitos/Restrições
 - Decisões de projeto
 - Informação estrutural
 - Comportamento (interações em tempo de execução)
 - Interfaces
- Relacionamentos entre visões

Views

Section 1. Primary Presentation of the View



OR

Textual version
of the primary
presentation

Section 2. Element Catalog

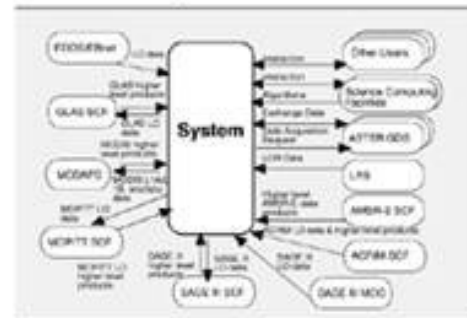
Section 2.A Elements and their properties

Section 2.B Relations and their properties

Section 2.C Element interfaces

Section 2.D Element behavior

Section 3. Context Diagram



Section 4. Variability Guide

Section 5. Architecture Background

Section 5.A Design rationale

Section 5.B Analysis of results

Section 5.C Assumptions

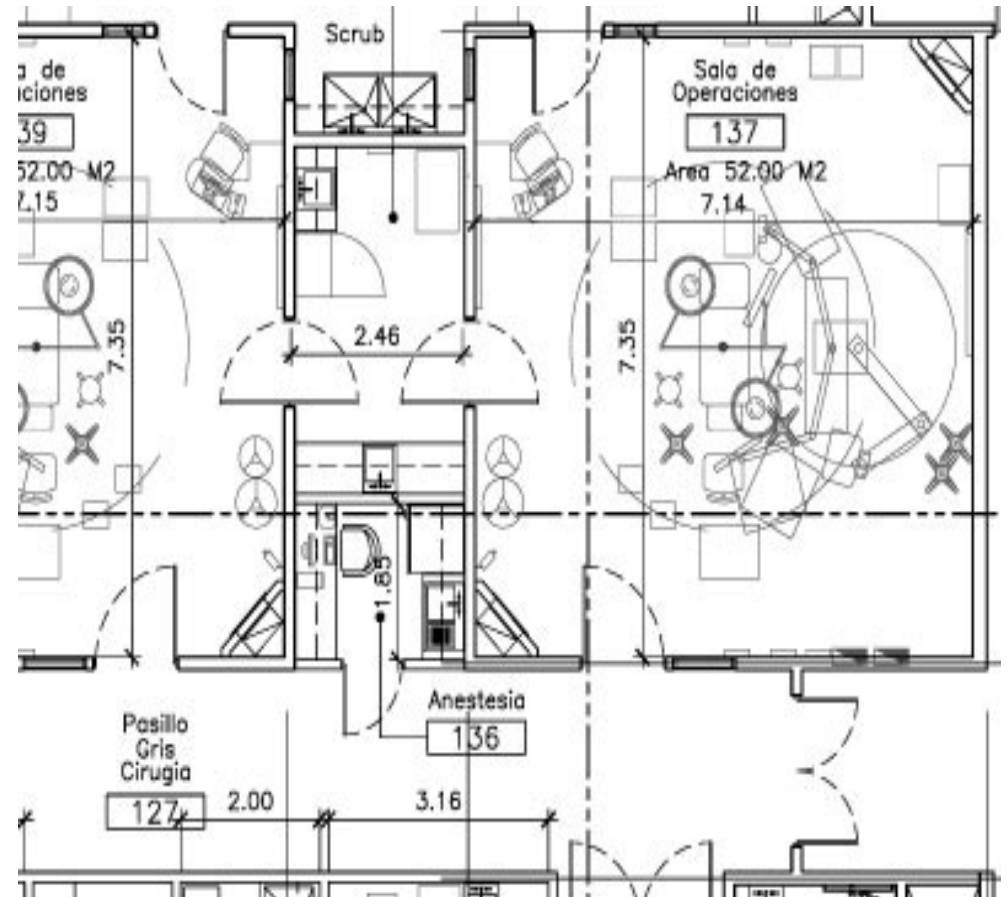
Section 6. Glossary of Terms

Section 7. Other Information

Existem vários tipos de visões!

Quais usar? E por quê?

Visões de Módulos



Fonte: http://www.heydaycreative.com/_images/portfolio/port-cad.jpg

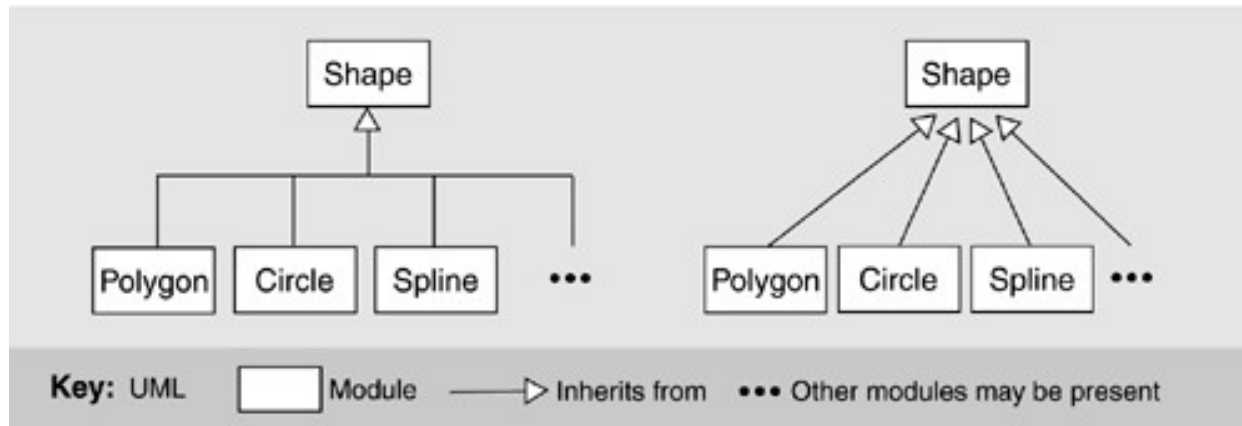
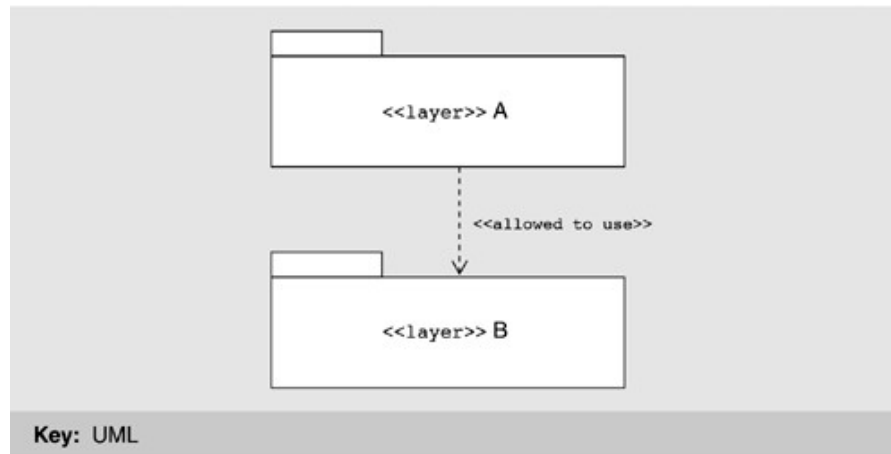
Visões de Módulos

- Mostram o sistema em termos de unidades de desenvolvimento
 - E dependências entre elas
- Camadas, Uso, Decomposição, Generalização
- Evidenciam diversas **características importantes**

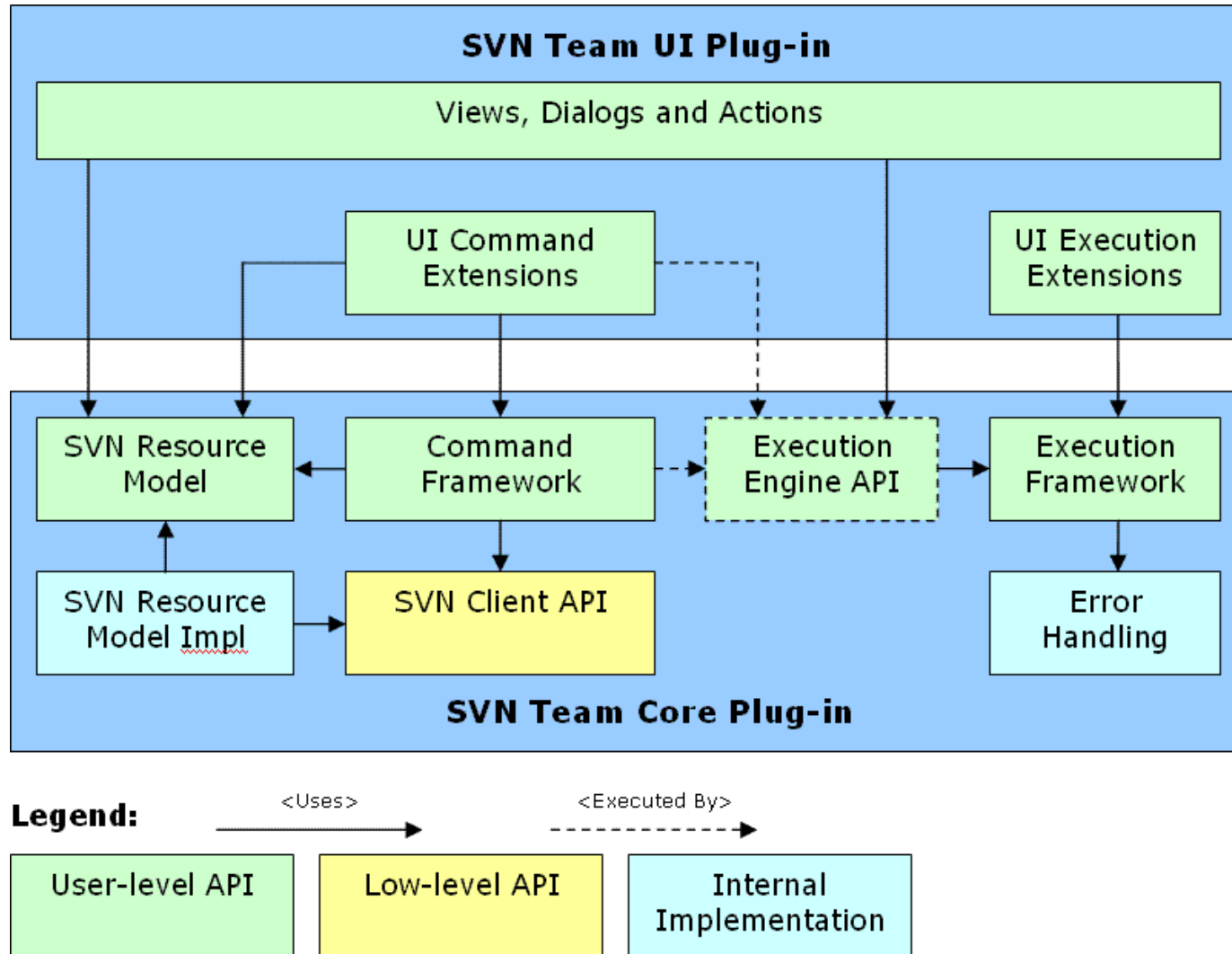
Visões de módulos evidenciam...

- facilidade de manutenção/extensão
- facilidade de compreensão
- facilidade de substituição
- reuso
- portabilidade
- divisão de esforço de desenvolvimento

Módulos na UML



Diagramas Esquemáticos



Matrizes de Estrutura de Projeto

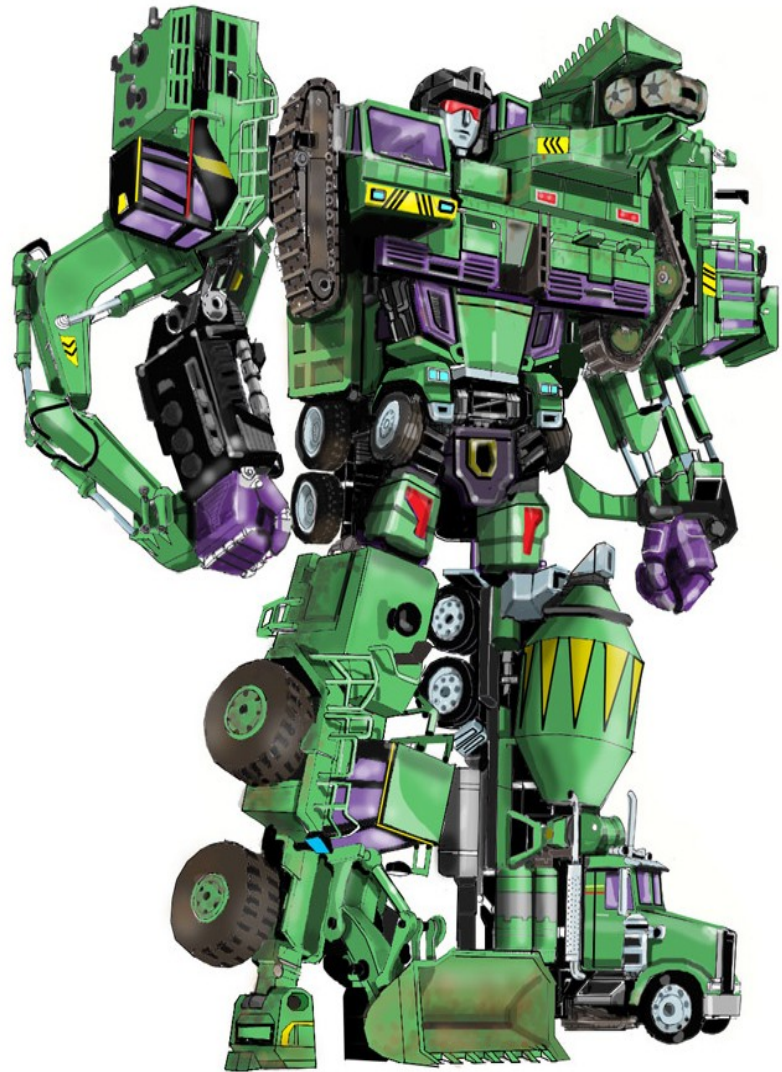
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
util_xmldocuments	1	.																				
util_tracer	2		.																			
components_servicelocator	3			.																		
components_address	4	X			.																	
components_creditcard	5	X				.																
components_lineitem	6	X					.															
components_asyncsender	7			X				.														
components_contactinfo	8	X		X	X				.													
components_catalog	9		X	X						.												
components_purchaseorder	10	X		X		X	X		X		.											
components_cart	11									X		.										
components_customer	12					X			X				.									
components_supplierpo	13	X		X			X		X					.								
components_processmanager	14														.							
components_signon	15															.						
components_uidgen	16																.					
components_mailer	17	X																.				
components_encodingfilter	18																		.			
waf_controller	19		X	X																.		
waf_view	20																				.	
apps_opc	21	X		X	X		X		X		X				X			X				.

Fonte: "Trading Obliviousness for Modularity with Cooperative Aspect-oriented Programming", de Kevin Hoffman e Patrick Eugster. Submetido para publicação.

Verificando Habilidades

- Para representar sua arquitetura do Google Wave, foram usadas visões de módulos?
 - O que foi representado (ou poderia ser)?
 - Qual notação foi empregada?

Visões de Componentes e Conectores



Visões de Componentes e Conectores

- Estrutura do sistema em tempo de execução
- **Conectores** (interações) representados de forma explícita
 - Comunicação (adaptadores distribuídos)
 - Coordenação (mediadores)
 - Conversão (adaptadores)
 - Facilitação (balanceamento de carga)

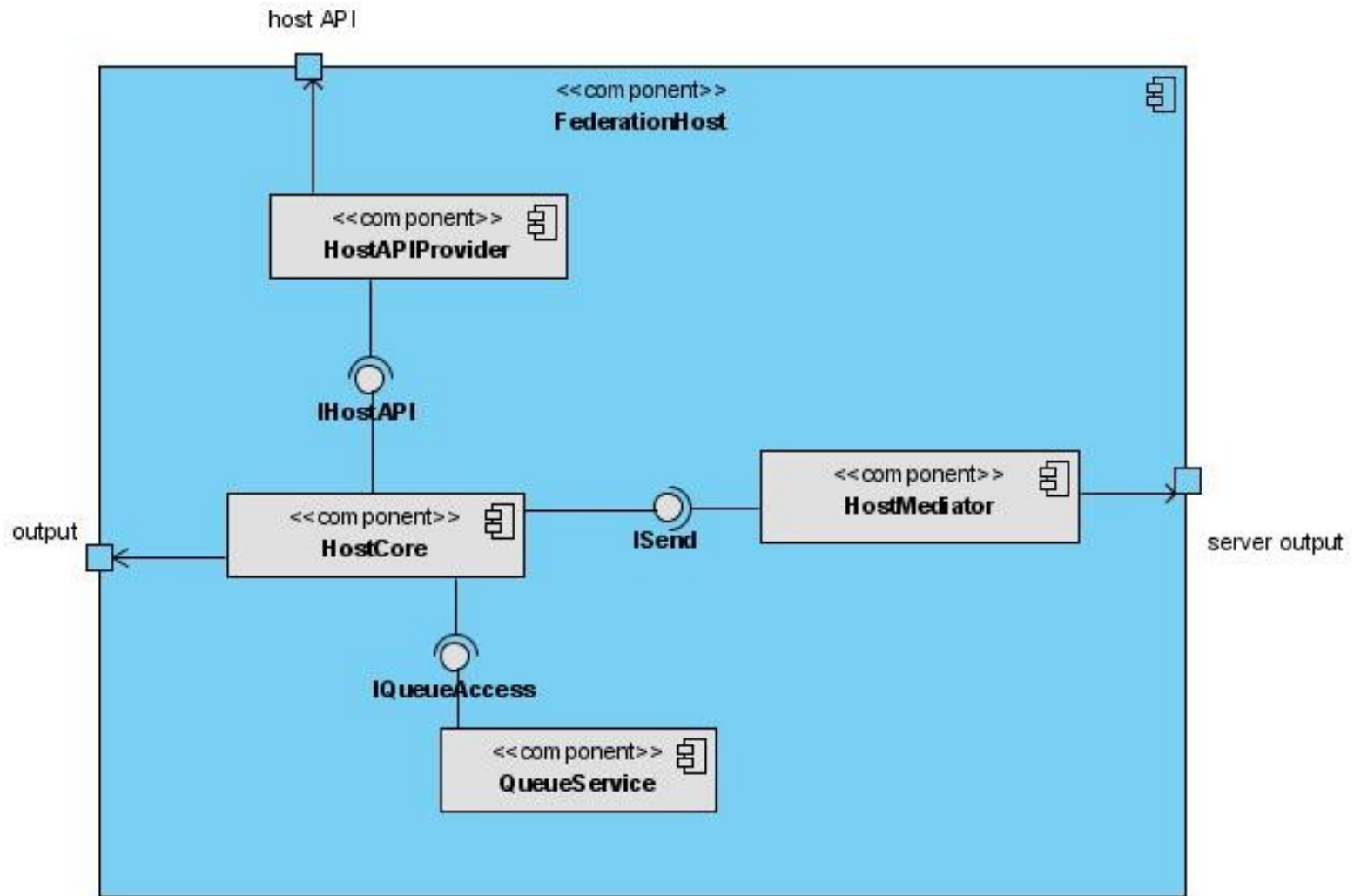
Úteis para Analisar...

- Desempenho
 - vazão, latência e uso de recursos
- Confiabilidade
- Disponibilidade
- Protocolos de interação
- Escalabilidade

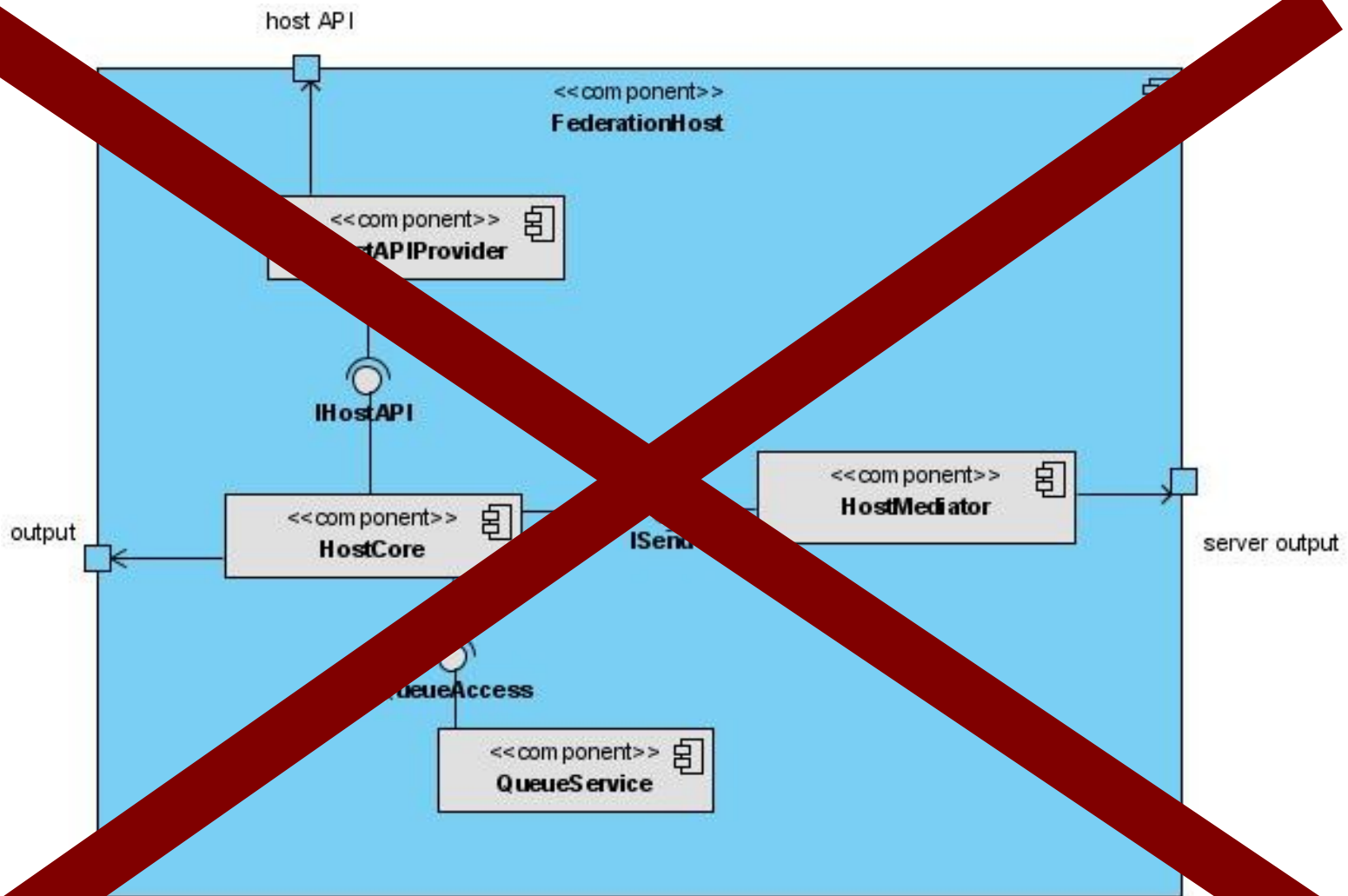
Documentando Componentes e Conectores

- UML
 - Apenas a partir da versão 2.0
- Diagramas de blocos
- Notações formais

Um exemplo usando UML



Um exemplo usando UML



Arquitetura Filtros e Canos na UML

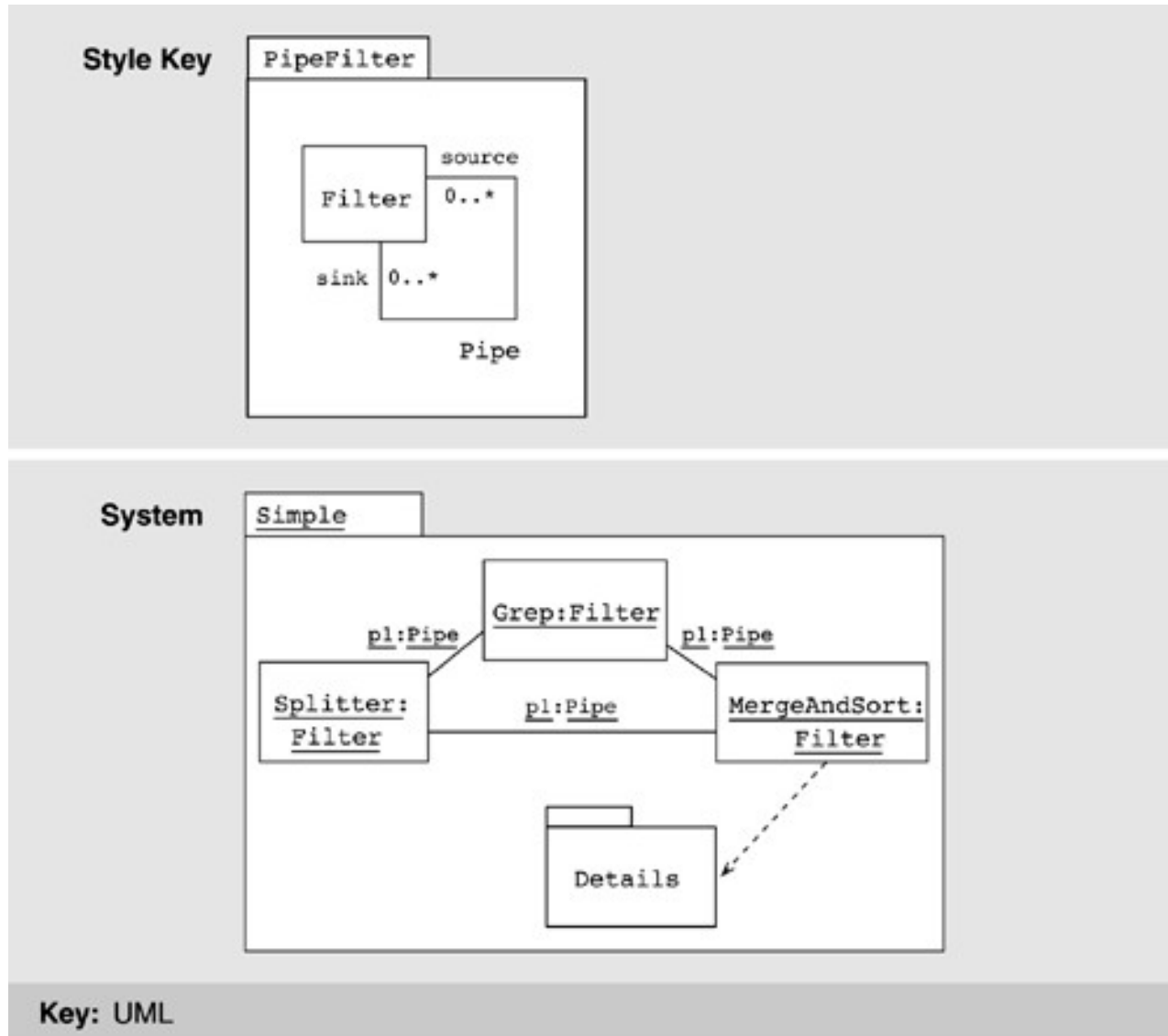
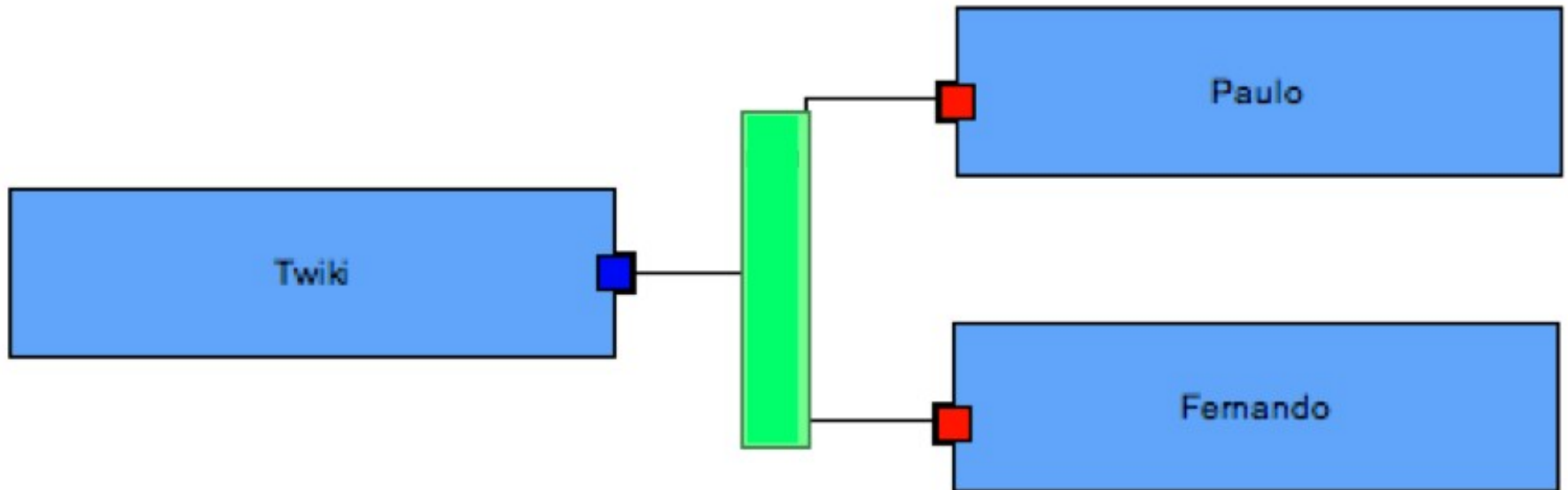
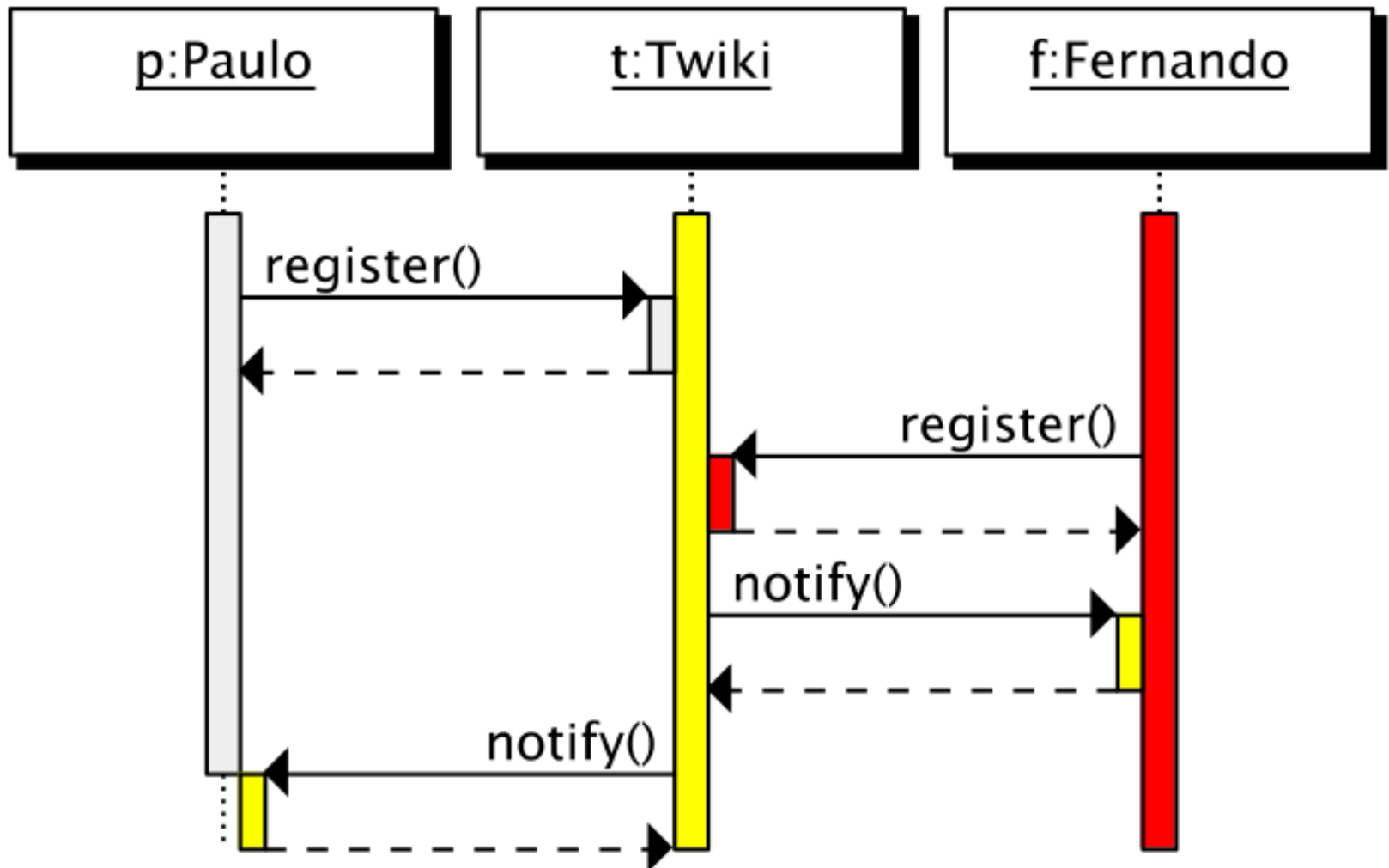


Diagrama de Blocos do AcmeStudio



Devem ter pelo menos uma sintaxe bem definida
(idealmente uma semântica também)

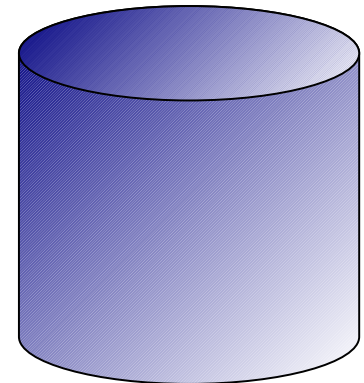
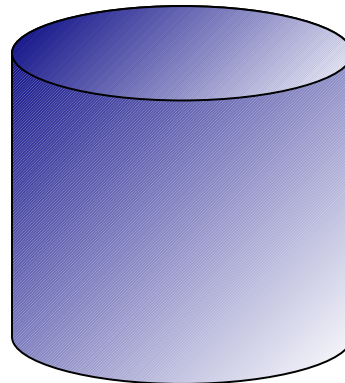
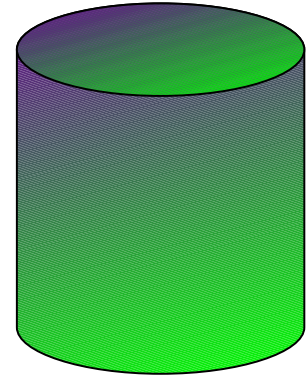
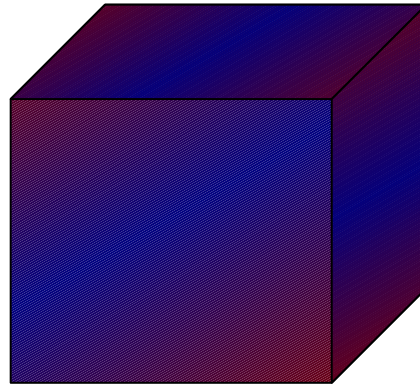
Comportamento dos Elementos



Especificação Parcial em ACME

```
import families\AlteredLayeredFam.acme;
System Normal-Bank : AlteredLayeredFam =
  new AlteredLayeredFam extended with {
    Component MySql : layerT = new layerT extended with {
      Property handlesAsynchRequests : boolean = false;
      Property layerLevel : int = 10;
      Property multiThreaded : boolean = false;
      Port lowerPort : sendRequestT = new sendRequestT extended with {
        Property protocol : requestProtocolT =
          PC << default : requestProtocolT = PC; >>;
        ...
      }; ...
    };
  };
Connector connMySqlPersistence = {
  Role r = { ... }; ...
}; ...
}
```

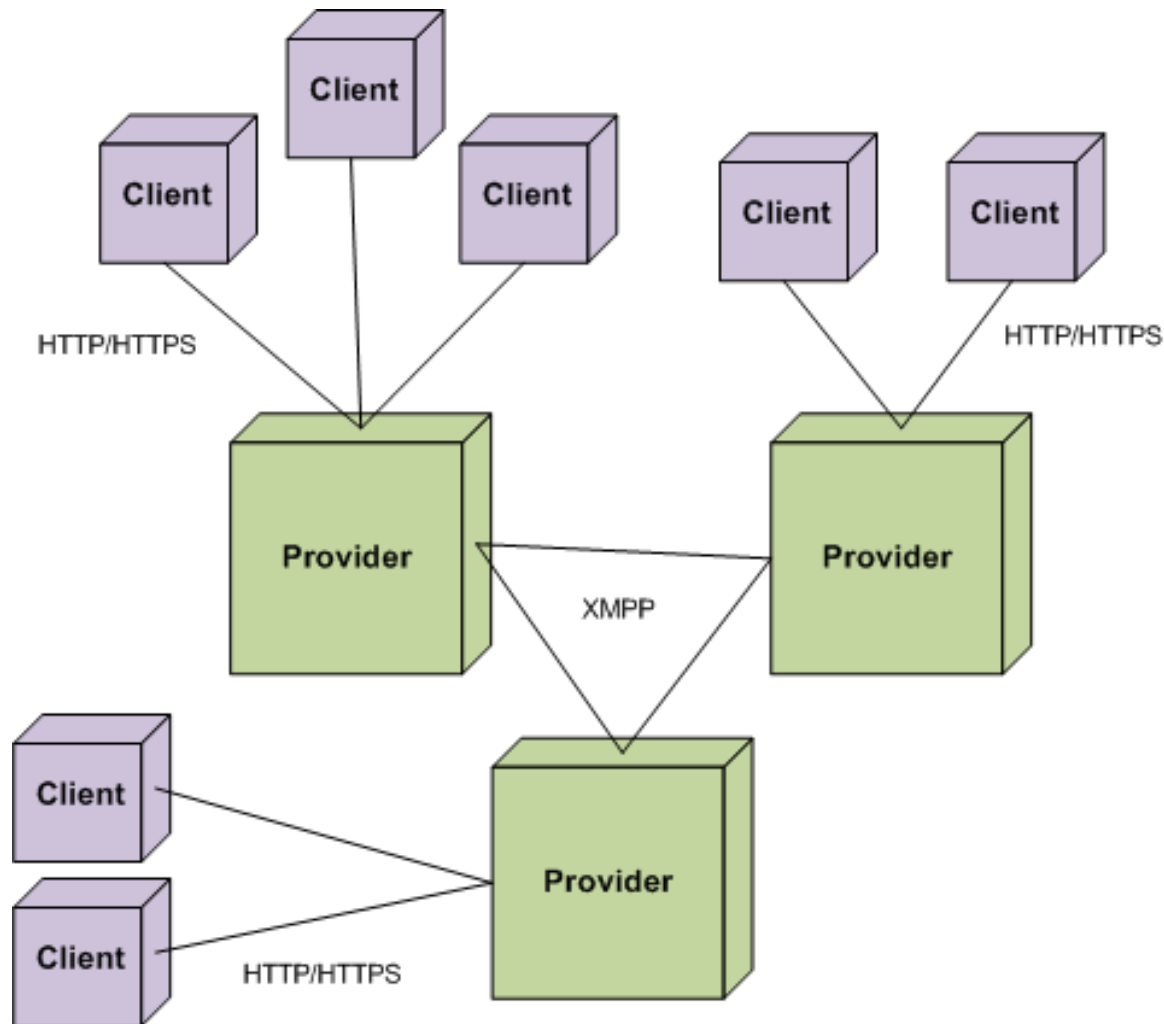
Visões de Alocação



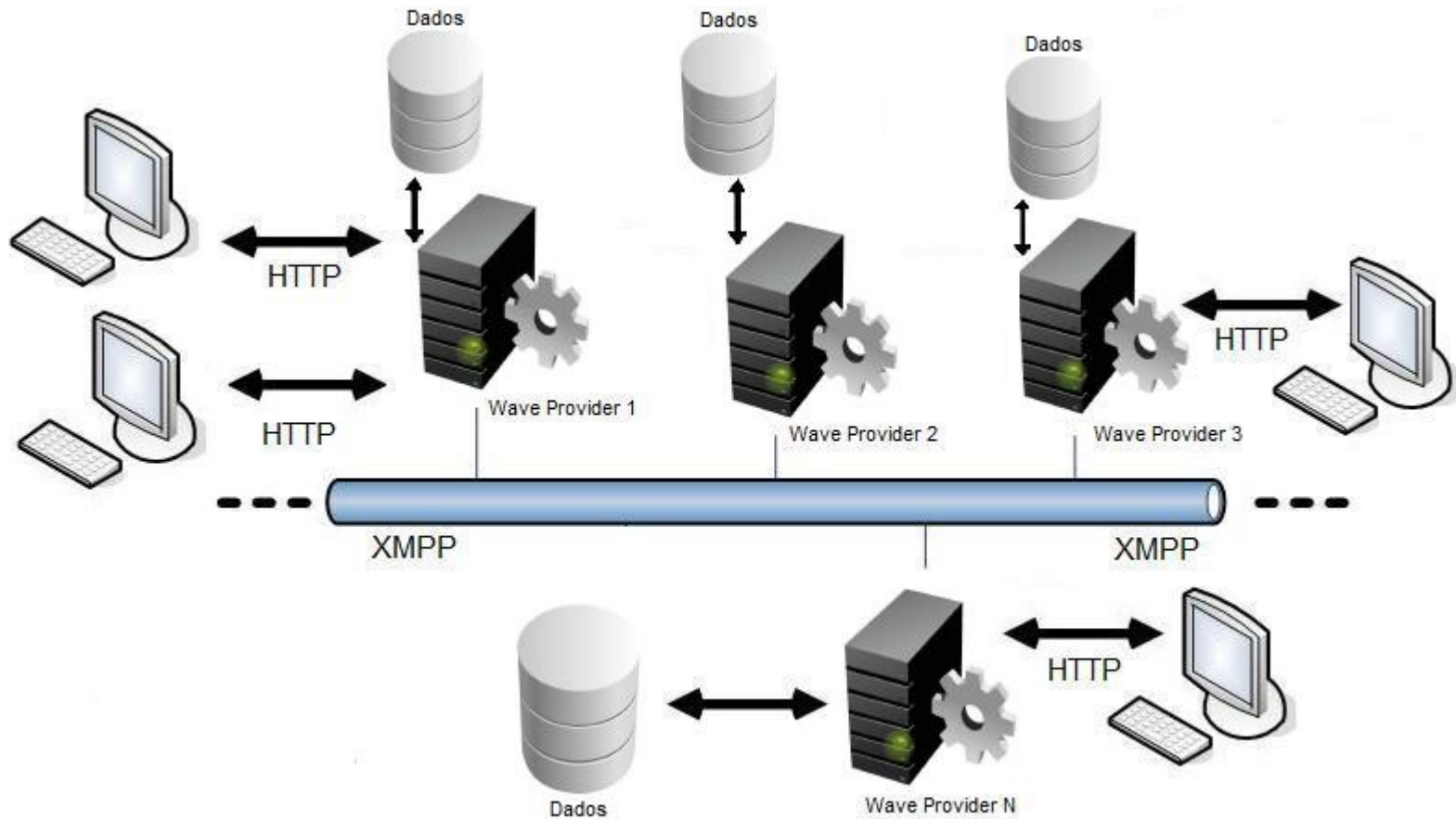
Visões de Alocação

- Relacionam elementos do sistema a elementos externos (não-software)
 - Muito diferentes entre si
- Clarificam questões como:
 - Alocação dos componentes de software aos de hardware
 - Alocação das tarefas de desenvolvimento aos membros da equipe
 - Alocação de módulos ao ambiente de desenvolvimento
 - Relacionado com gerenciamento de configuração
- Influenciam vários atributos de qualidade

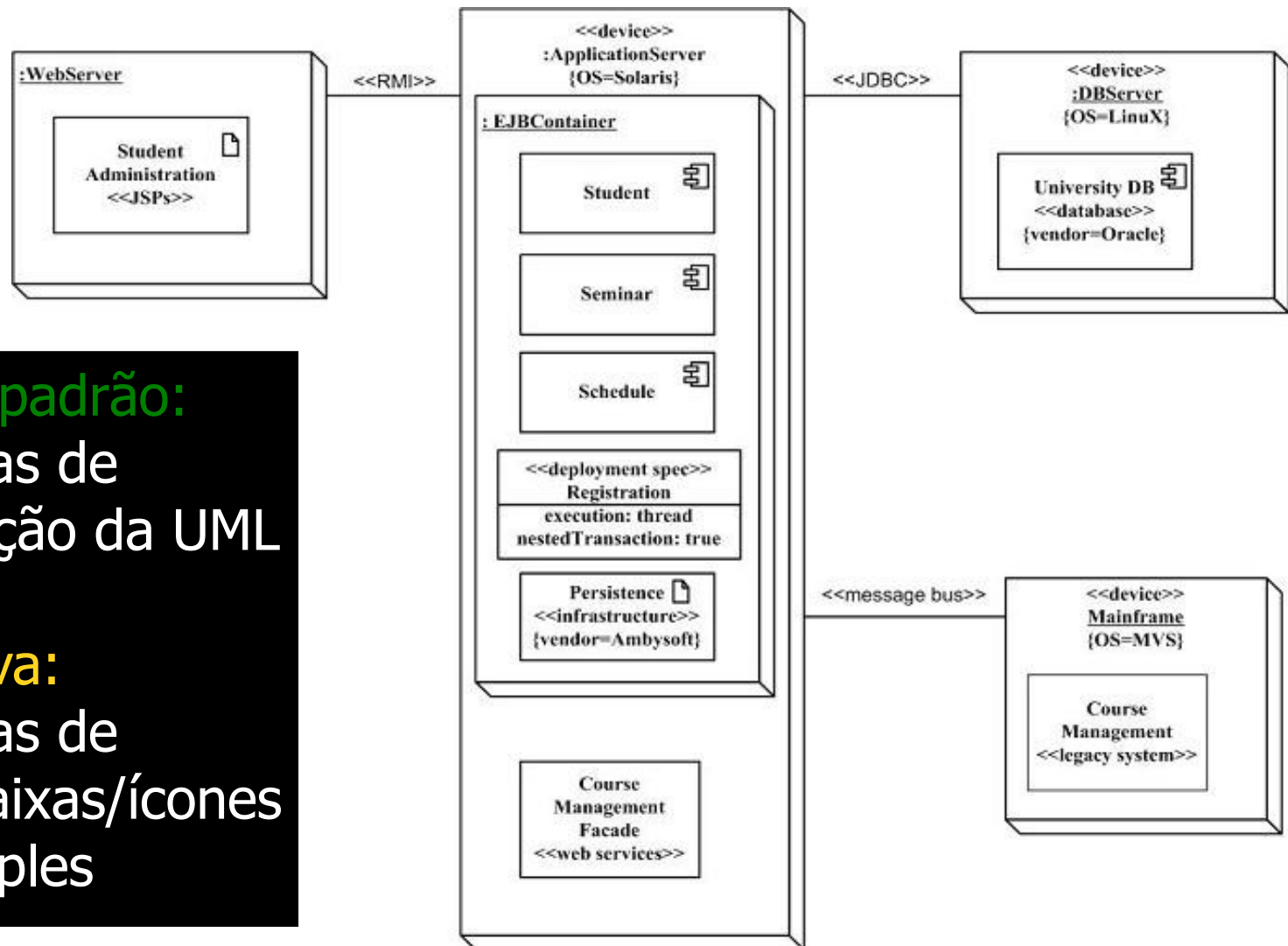
Há problemas nesta visão de implantação?
(ou é algum outro tipo de visão?)



E nesta?



Visão de Implantação



Notação padrão:
Diagramas de
implantação da UML

Alternativa:
Diagramas de
blocos/caixas/ícones
mais simples

Visão de Atribuição de Trabalho

- Arquitetura auxiliando o gerenciamento do projeto
 - Objetivo do desenvolvimento modular
 - Útil para gerenciamento de recursos e realização de estimativas
- Mapeamento de módulos do sistema a humanos
- Útil mesmo quando se adota componentes OTS

Um Exemplo de Visão de Atribuição de Trabalho

Segment	Module	Coordinator
WaveProvider	WaveServerCore	José Dihego
	MessageUpdater	Ademir
WaveClient	Javascript Client	Davi
Communication Subsystem	Host and Remote API Provider	Adauto
WaveletProcessor	Conflict Manager	Rafael
	Transformation Engine	Paulo

Verificando Habilidades

Sua descrição da arquitetura do Google Wave inclui visões de alocação?

- Caso a resposta seja ``**não**'', por que não?
- O que você quis mostrar com elas?
 - Auxiliam na análise de algum atributo de qualidade do sistema?

Interfaces



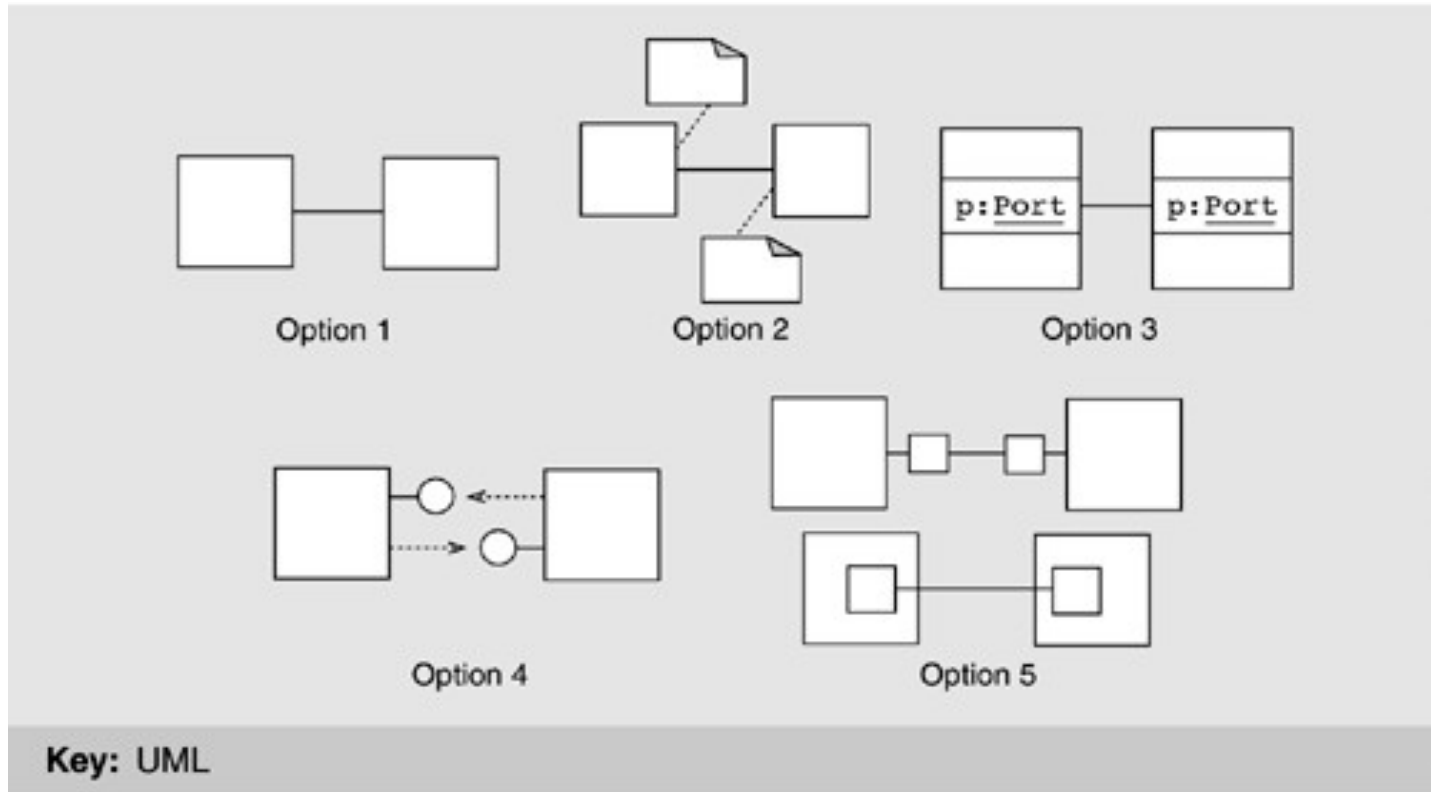
Interfaces na Arquitetura (1)

- Conceito mais amplo que nas LPs
 - Indicam **pontos de interação direta** com um elemento
 - Não necessariamente procedimentos/métodos/funções
 - **Que tipo de coisa uma interface assim poderia especificar?**
- Fenômenos e eventos visíveis externamente

Interfaces na Arquitetura (2)

- Funcionam em duas direções
 - Interfaces providas
 - Interfaces requeridas
- As requeridas especificam
 - Recursos dos quais um elemento depende
 - Ex. A API JDBC
 - Suposições que faz sobre outros elementos
 - Exs. Esquema de banco de dados, sequência de operações, modelo de interação

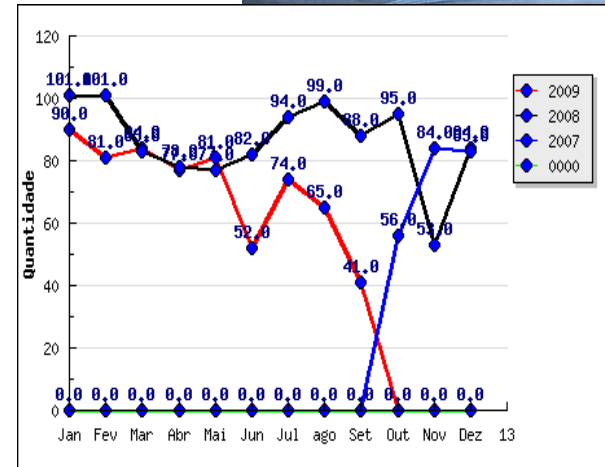
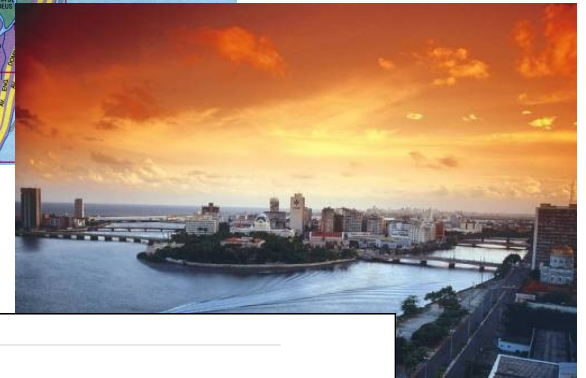
Representando Interfaces Graficamente



- Em geral, dêem preferência às Opções 1 e 2

1. **Identificação** da Interface
2. Recursos providos
 - **Sintaxe**
 - Semântica/restrições (pré- e pós-condições)
3. **Tipos de dados** usados
 - Principalmente os definidos internamente
4. **Erros sinalizados**
5. Recursos requeridos e Suposições
 - Erros tratados
6. Escolhas de projeto e justificativas
7. Guia de uso

Documentando Entre Visões



Três Elementos Cruciais

(da documentação entre visões)

- Como a documentação está organizada
- O que é a arquitetura
- Por que a arquitetura é assim

Modelo para Documentação Entre Visões

Documentation across Views

How the document is organized:

- 1.1 View catalog**
- 1.2 View template**

What the architecture is:

- 2.1 System overview**
- 2.2 Mapping between views**
- 2.3 List of elements and where they appear**
- 2.4 Project glossary**

Why the architecture is the way it is:

- 3.1 Rationale**

Source: Adapted from [Clements 03].

Obrigado!

Dúvidas e comentários:

castor@cin.ufpe.br

phmb@cin.ufpe.br