



# Top. Esp em Banco de Dados

Profa. Marta Mattoso

[marta@cos.ufrj.br](mailto:marta@cos.ufrj.br)

<http://www.cos.ufrj.br/~marta>

COPPE- Sistemas / UFRJ

Arquiteturas em Banco de Dados

---

# Roteiro

- Introdução a arquiteturas de SGBD
- SGBD (ANSI/SPARC)
- SGBD Distribuído
- Cliente / Servidor
- Sistemas Heterogêneos
- Bibliografia

# Arquitetura

- Define a estrutura do sistema
  - Identificação dos componentes
  - Definição das funções de cada componente
  - Definição de relacionamentos e interações entre os componentes

# Padronização

- Modelo de Referência
  - Uma referência conceitual cujo propósito é dividir o trabalho de padronização em partes gerenciáveis e mostrar num nível mais alto como essas partes se relacionam entre si.

# Padronização

## ○ Abordagens

### ➤ Baseada em componentes

- Os componentes do sistema são definidos junto com as relações entre os componentes.
- Bom para projeto e implementação de sistemas.

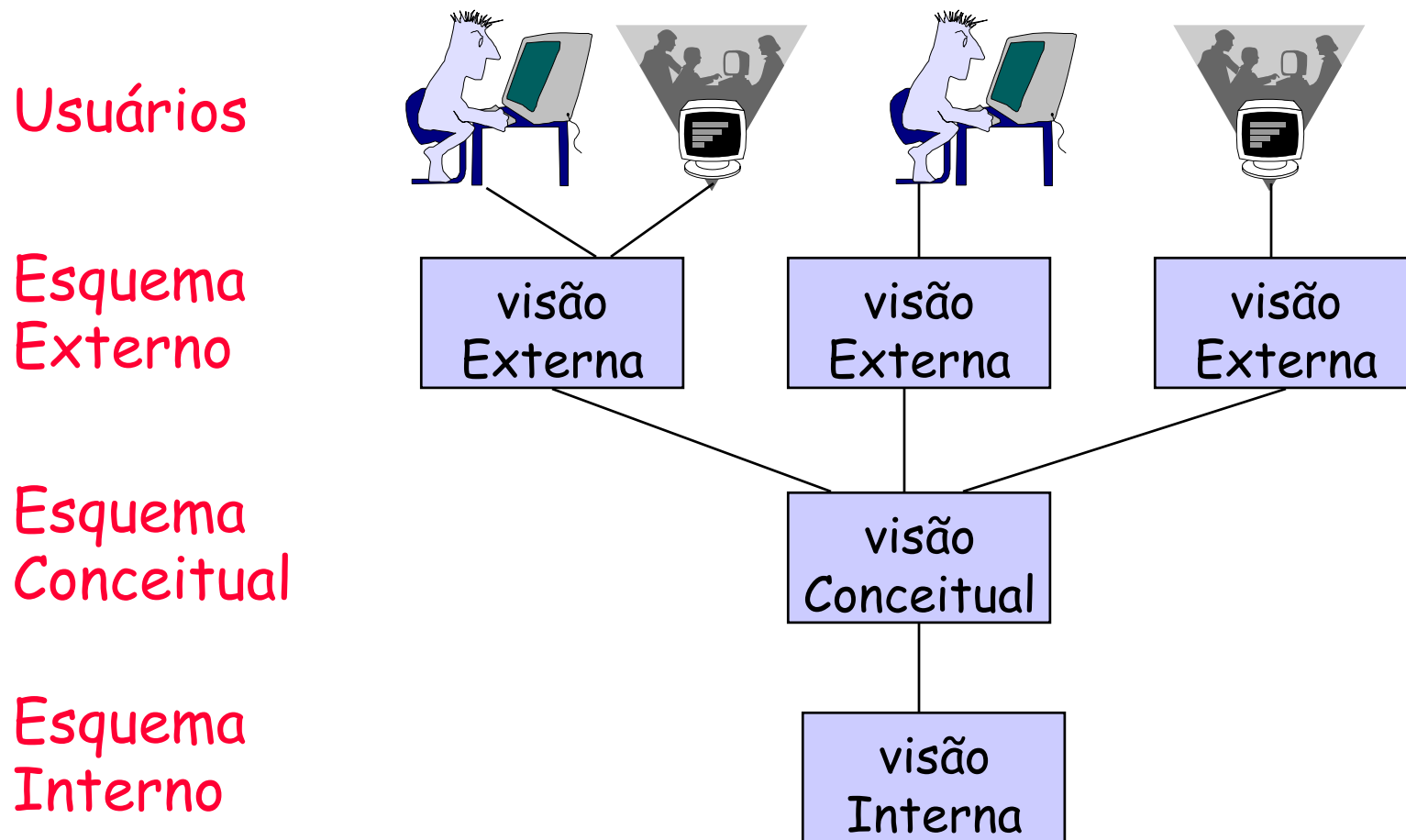
### ➤ Baseada em funções

- Classes de usuários são identificados junto com a funcionalidade que o sistema fornecerá a cada classe.
- Os objetivos são identificados. Mas como alcançar ?

### ➤ Baseada em dados

- Identifica as diferentes formas de descrever dados e especifica as unidades funcionais que irão definir e usar os dados de acordo com as formas.

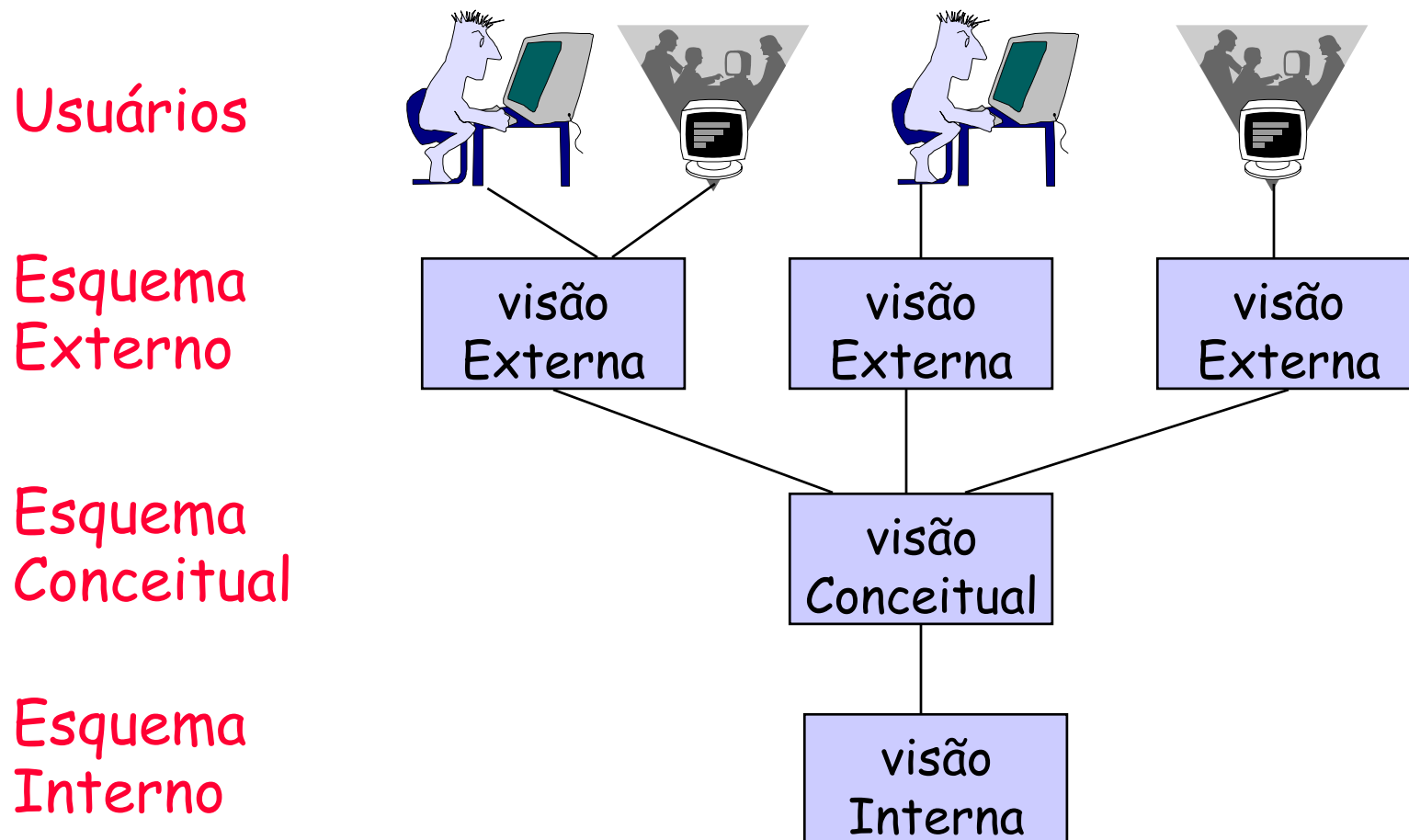
# Arquitetura ANSI/SPARC (dados)



# Roteiro

- ✓ Introdução a arquiteturas de SGBD
- Arquitetura baseada em dados
  - SGBD (ANSI/SPARC)
    - nível Externo
    - nível Conceitual
    - nível Interno
  - SGBD Distribuído
- Cliente / Servidor
- Sistemas Heterogêneos

# Arquitetura ANSI/SPARC (dados)





# Definição do esquema conceitual

```
RELATION CIA_AEREA [
  KEY = {NOME}
  ATTRIBUTES = {
    NOME:          CHAR (20)
    FATURAM:       NUMERIC (6)
    PRESID:        CHAR (20)
    SEDE:          CHAR (50)
  }
]

RELATION PASSAGEIROS [
  KEY = {NOME}
  ATTRIBUTES = {
    NOME:          CHAR (40)
    TELEFONE:      CHAR (12)
  }
]
```

# Definição do esquema conceitual

```
RELATION VOOS [
  KEY = {VOO}
  ATTRIBUTES = {
    VOO:          CHAR (5)
    TARIFA:       NUMERIC (8)
    ASS:          NUMERIC (6)
    CIA:          CHAR (20)
  }
]

RELATION RESERVAS [
  KEY = {VOO, PASSAGEIRO}
  ATTRIBUTES = {
    VOO:          CHAR (5)
    PASSAGEIRO:  CHAR (40)
  }
]
```

# Definição do esquema interno

```
RELATION VOOS [
  KEY = {VOO}
  ATTRIBUTES = {
    VOO:          CHAR (5)
    TARIFA:       NUMERIC (8)
    ASS:          NUMERIC (6)
    CIA:          CHAR (20)
  }
]

INTERNAL_REL VOOS [
  INDEX ON V# CALL VOINX
  FIELD = {
    HEADER:       BYTE (1)
    V#:           BYTE (5)
    TARIFA:       NUMERIC (8)
    ASS:          NUMERIC (6)
    CIA:          CHAR (20)
  }
]
```

# Definição de visão externa

## Exemplo 1

Criação da visão **TARIFA** a partir de **VOOS**

```
CREATE VIEW TARIFA (VOO, VALOR)
AS      SELECT      VOO, TARIFA
          FROM        VOO
```

# Definição de visão externa

## Exemplo 2

Criação da visão Contato a partir de Passageiros e Reservas

```
CREATE VIEW CONTATO (VOO, NOME, TELEFONE)
AS      SELECT      VOO, NOME, TELEFONE
          FROM      RESERVAS, PASSAGEIROS
          WHERE     PASSAGEIRO = NOME
```

# Esquema conceitual - ex. livro

```
RELATION EMP [  
  KEY = {ENO}  
  ATTRIBUTES = {  
    ENO : CHARACTER(9)  
    ENOME : CHARACTER(15)  
    CARGO : CHARACTER(10)  
  }  
]
```

```
RELATION PAY [  
  KEY = {TITLE}  
  ATTRIBUTES = {  
    CARGO : CHARACTER(10)  
    SAL : NUMERIC(6)  
  }  
]
```

# Esquema conceitual - ex. livro

RELATION PROJ [

KEY = {PNO}

ATTRIBUTES = {

PNO : CHARACTER(7)

PNAME : CHARACTER(20)

BUDGET : NUMERIC(7)

}

]

RELATION ASG [

KEY = {ENO,PNO}

ATTRIBUTES = {

ENO : CHARACTER(9)

PNO : CHARACTER(7)

RESP : CHARACTER(10)

DUR : NUMERIC(3)

}

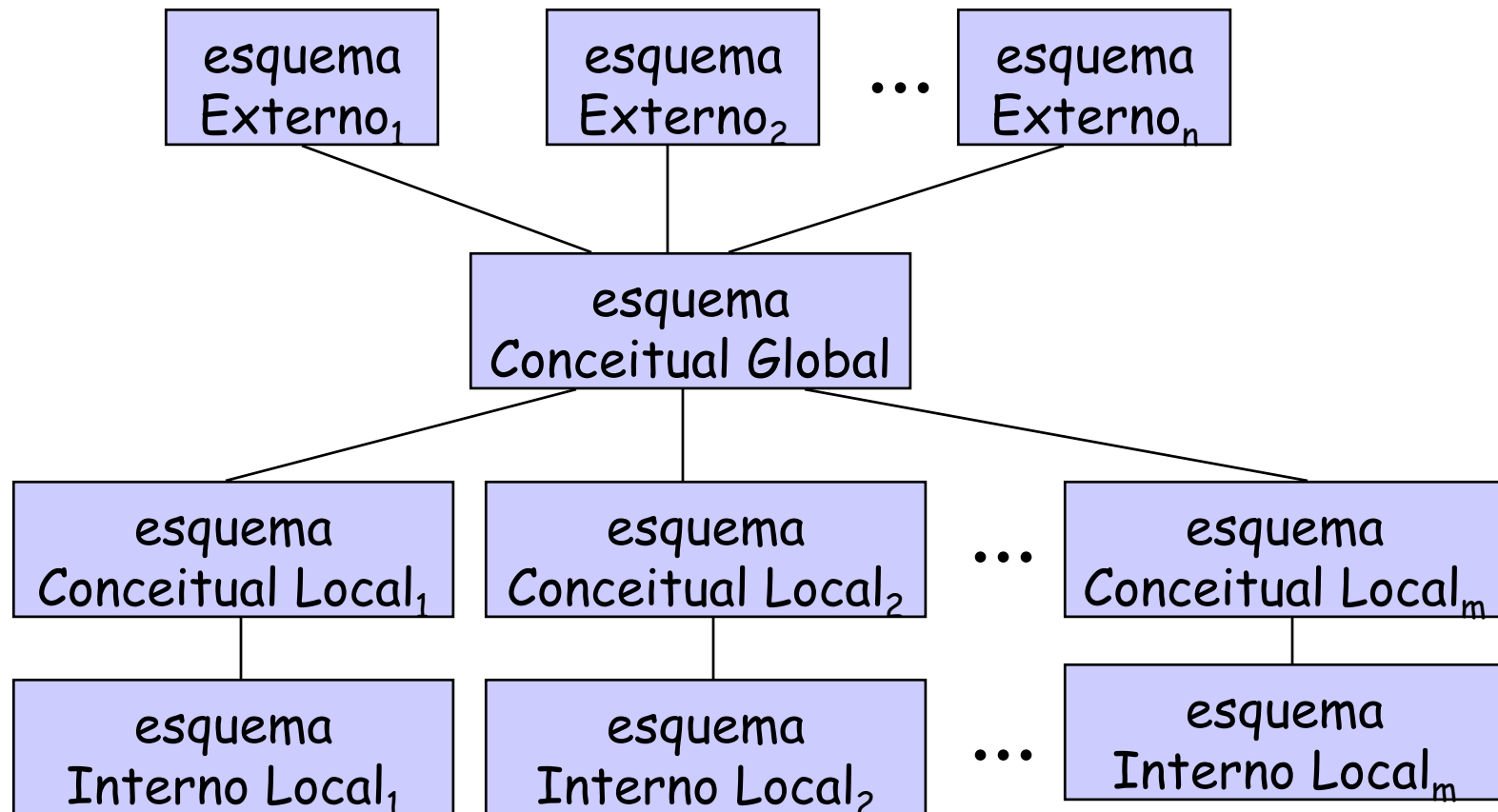
]

# Roteiro

- ✓ Introdução a arquiteturas de SGBD
- ✓ Arquitetura baseada em dados
  - ✓ SGBD (ANSI/SPARC)
    - SGBD Distribuído
  - Cliente / Servidor
  - Sistemas Heterogêneos



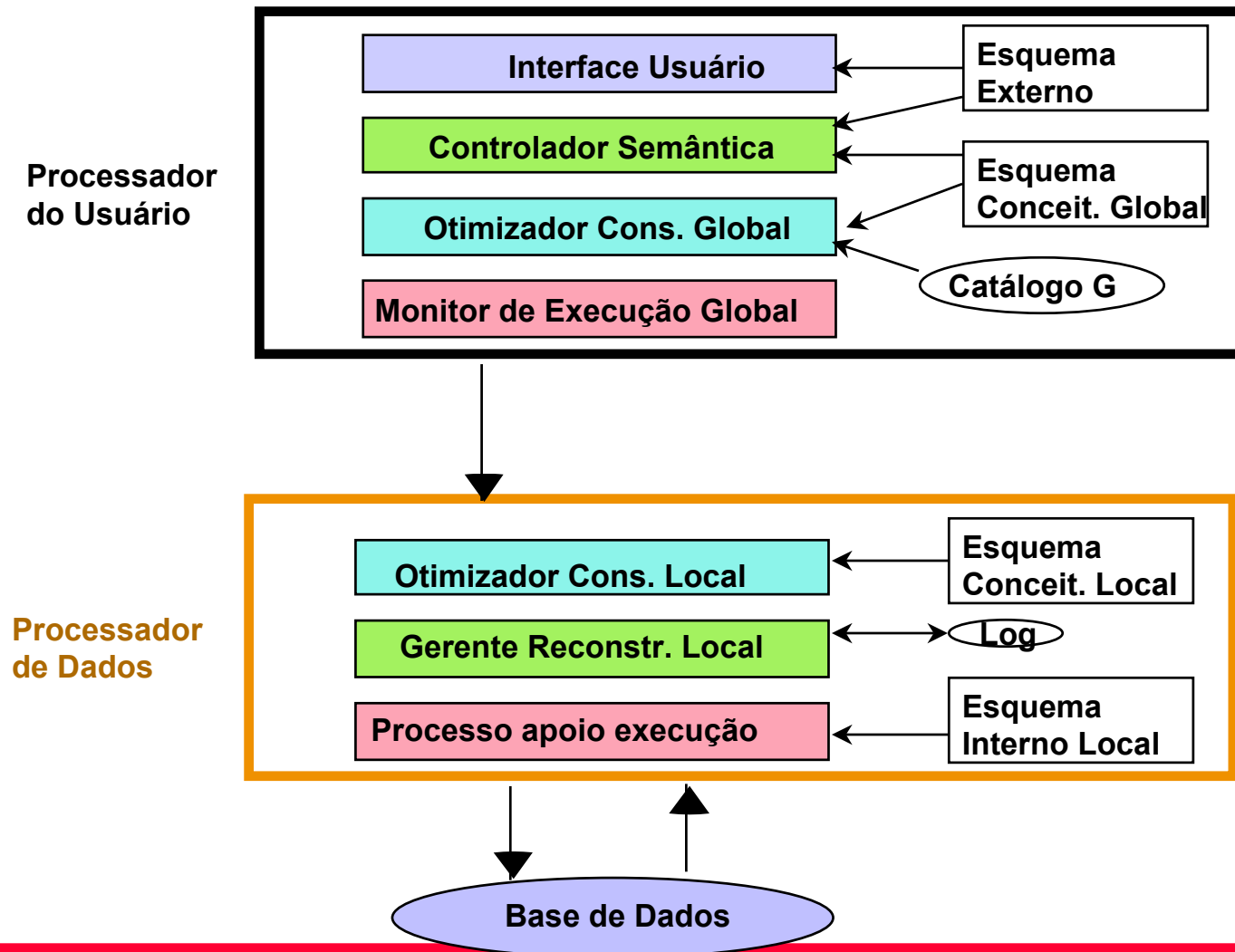
# Arquitetura de SGBD Distribuído



# Roteiro

- ✓ Introdução a arquiteturas de SGBD
- ✓ Arquitetura baseada em dados
  - ✓ SGBD (ANSI/SPARC)
  - ✓ SGBD Distribuído
- **Arquitetura baseada em componentes**
  - SGBD Distribuído
  - Cliente / Servidor
  - Sistemas Heterogêneos

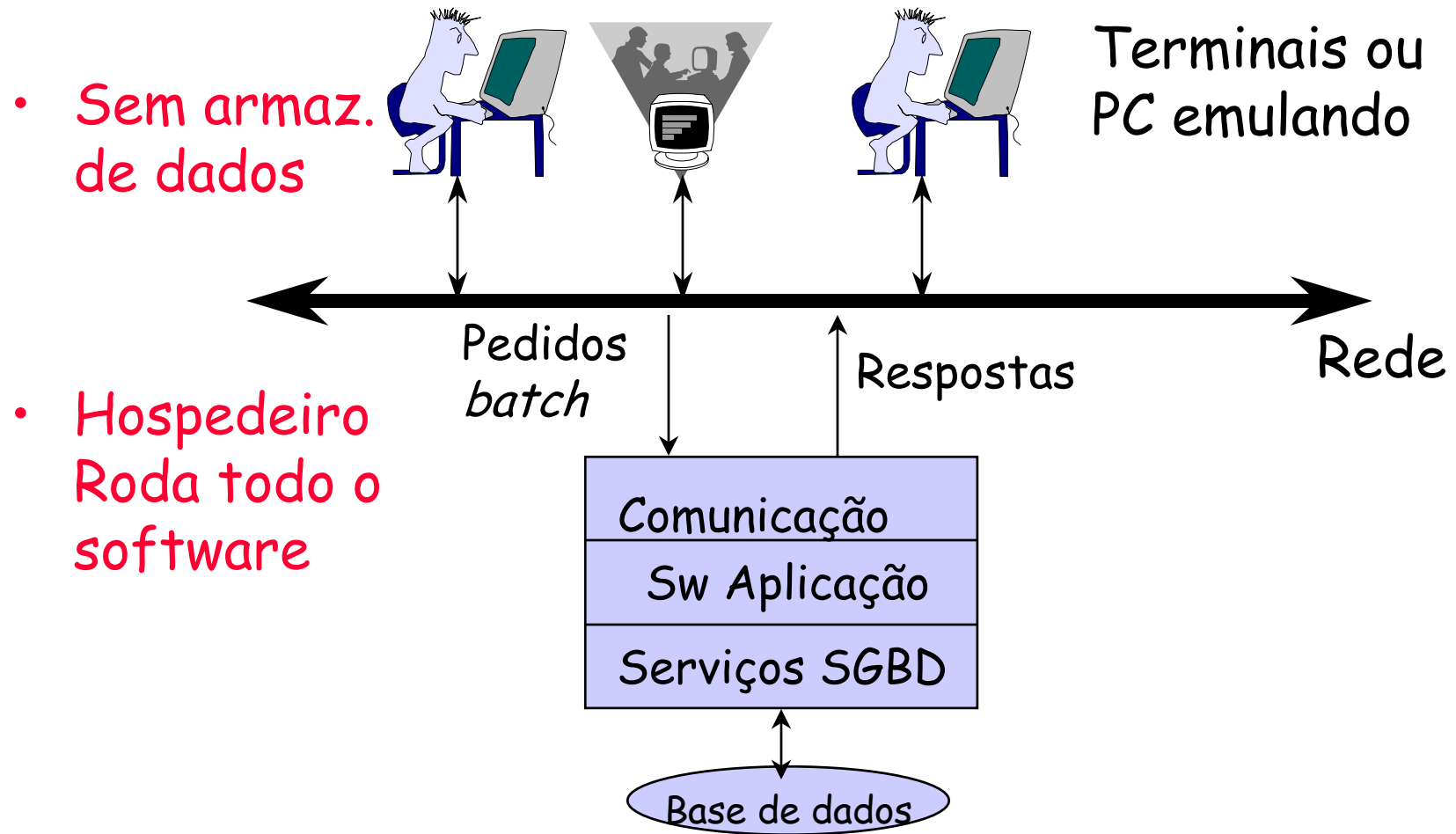
# Arquit. Componentes do SGBDD



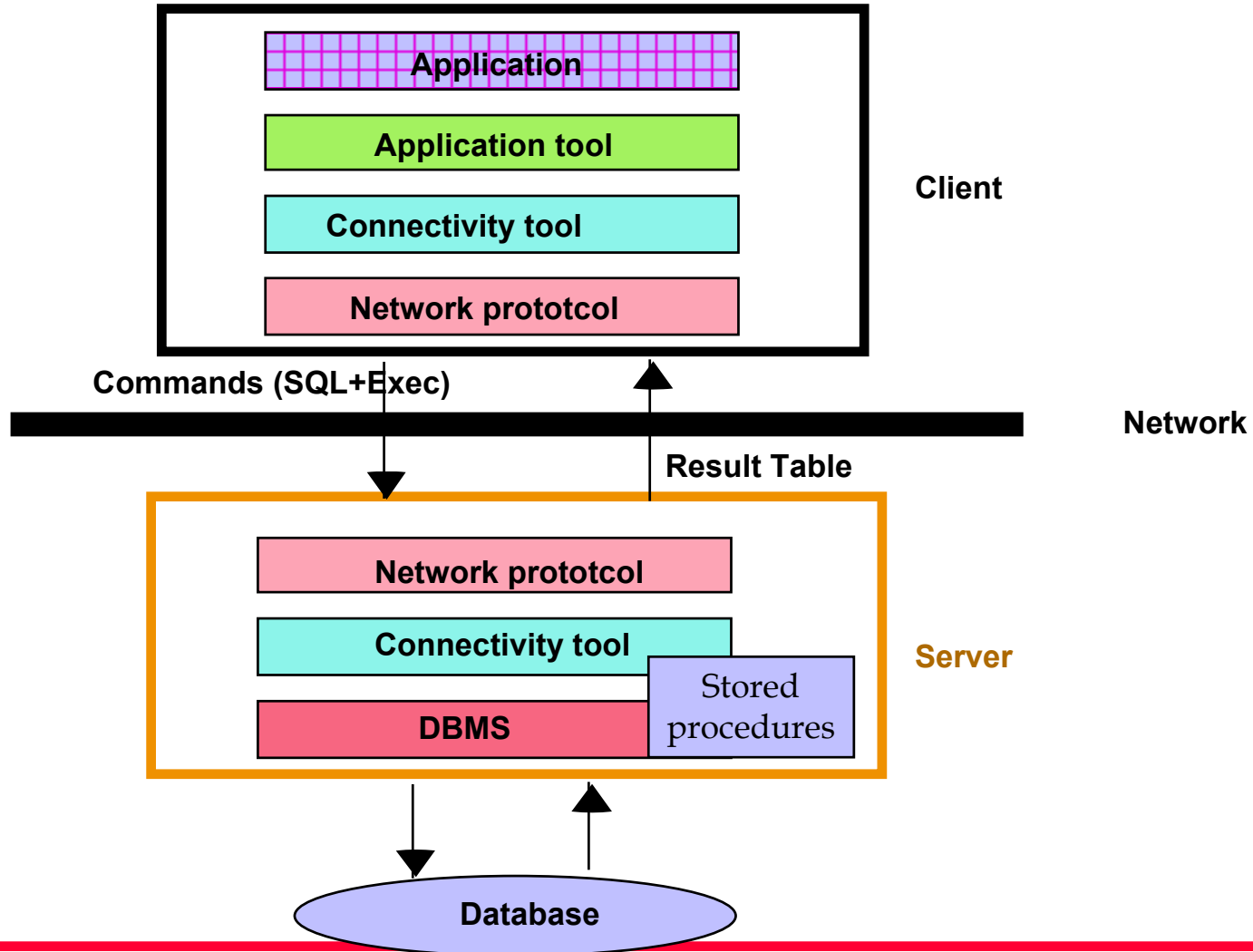
# Roteiro

- ✓ Introdução a arquiteturas de SGBD
- ✓ SGBD (ANSI/SPARC)
- ✓ SGBD Distribuído
- Cliente / Servidor
  - Muitos Clientes / Um Servidor
  - Muitos Clientes / Muitos Servidores
  - Três camadas (servidor de aplicação)
- Sistemas Heterogêneos

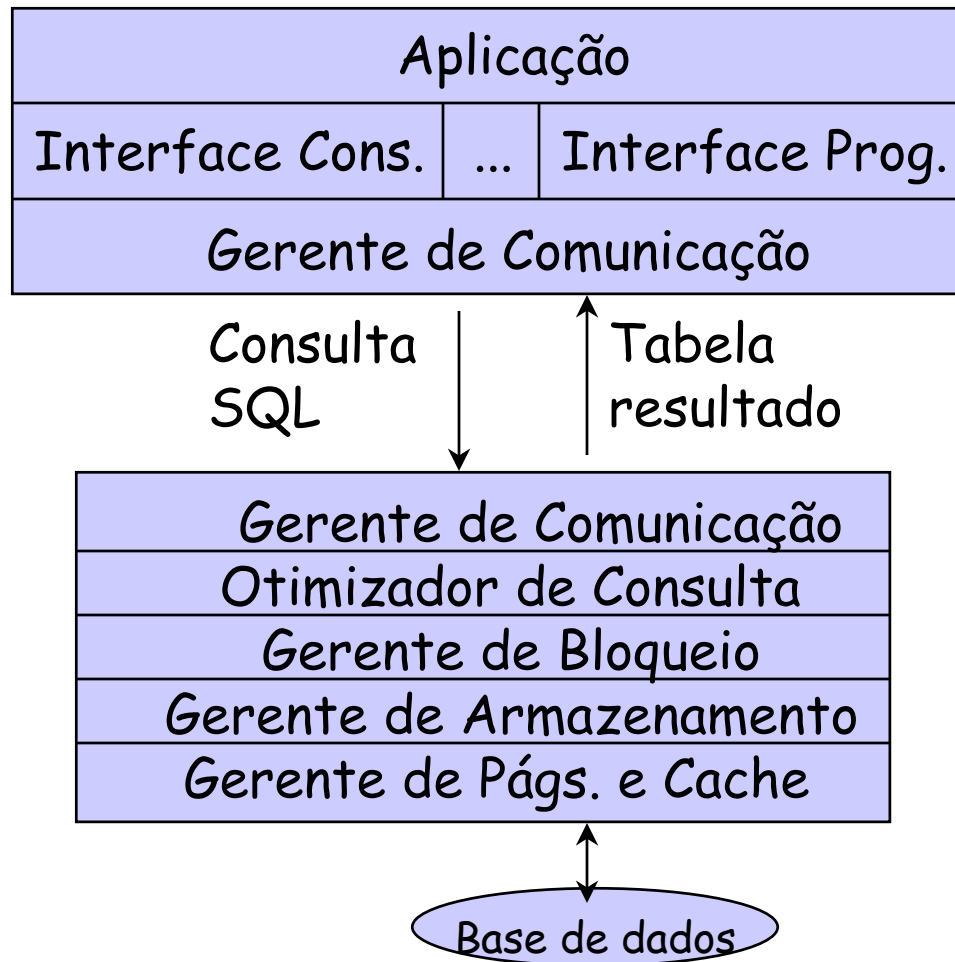
# Acesso Compartilhado ao SGBD Centralizado



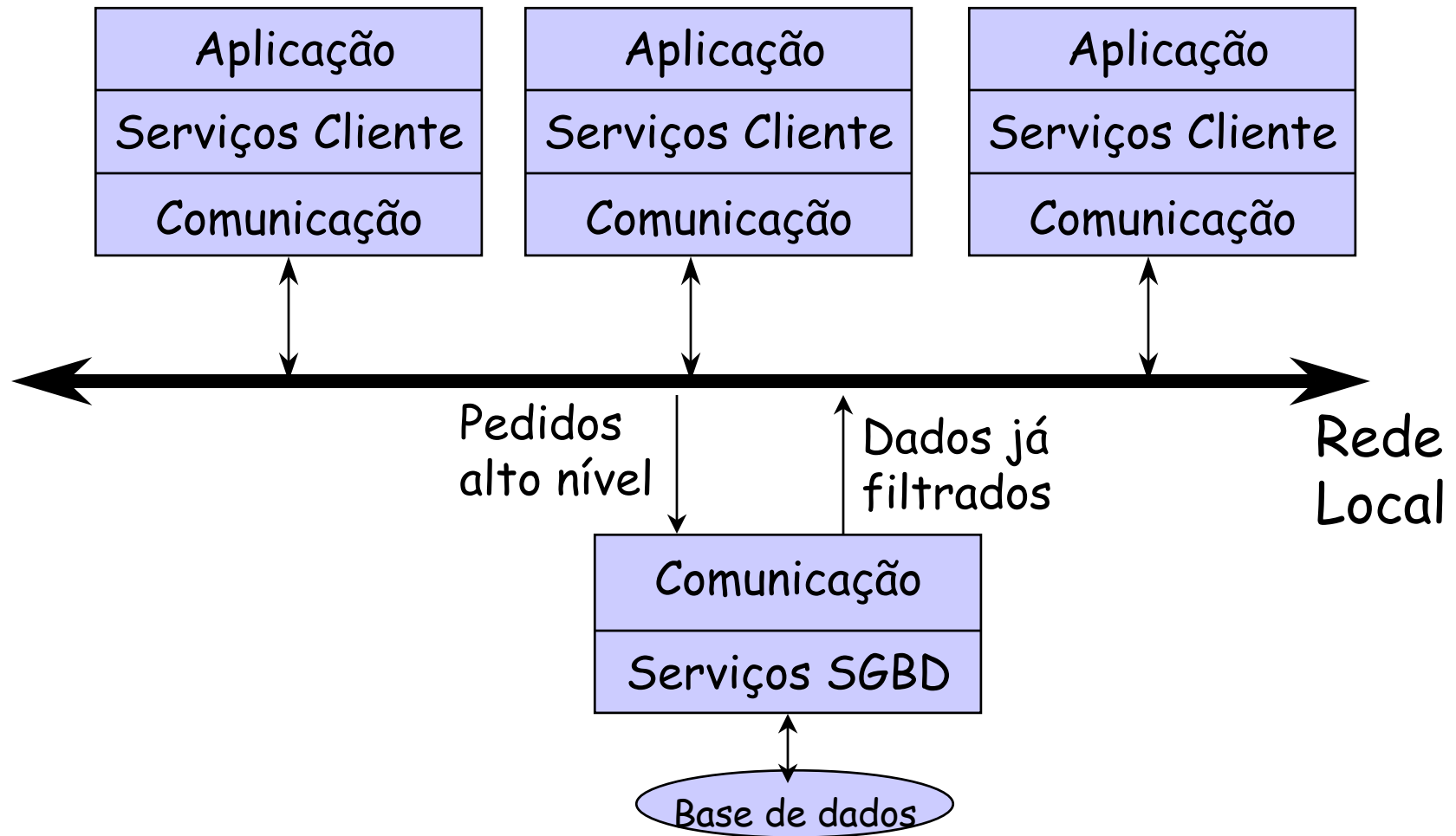
# Distribuição de tarefas [Valduriez99]



# Distribuição de Tarefas



# Muitos Clientes / Um Servidor





# Vantagens Arquit. Cliente/Servidor

- Divisão de trabalho mais adequada
- Escalonamento de recursos horizontal e vertical
- Máquinas cliente com bom preço/desemp.
- Pode-se usar ferramentas conhecidas (cliente)
- Cliente acessa dados remotos (via padrões)
- Todos serviços do SGBD disponíveis p/ cliente
- Melhor preço/desemp. do sistema global

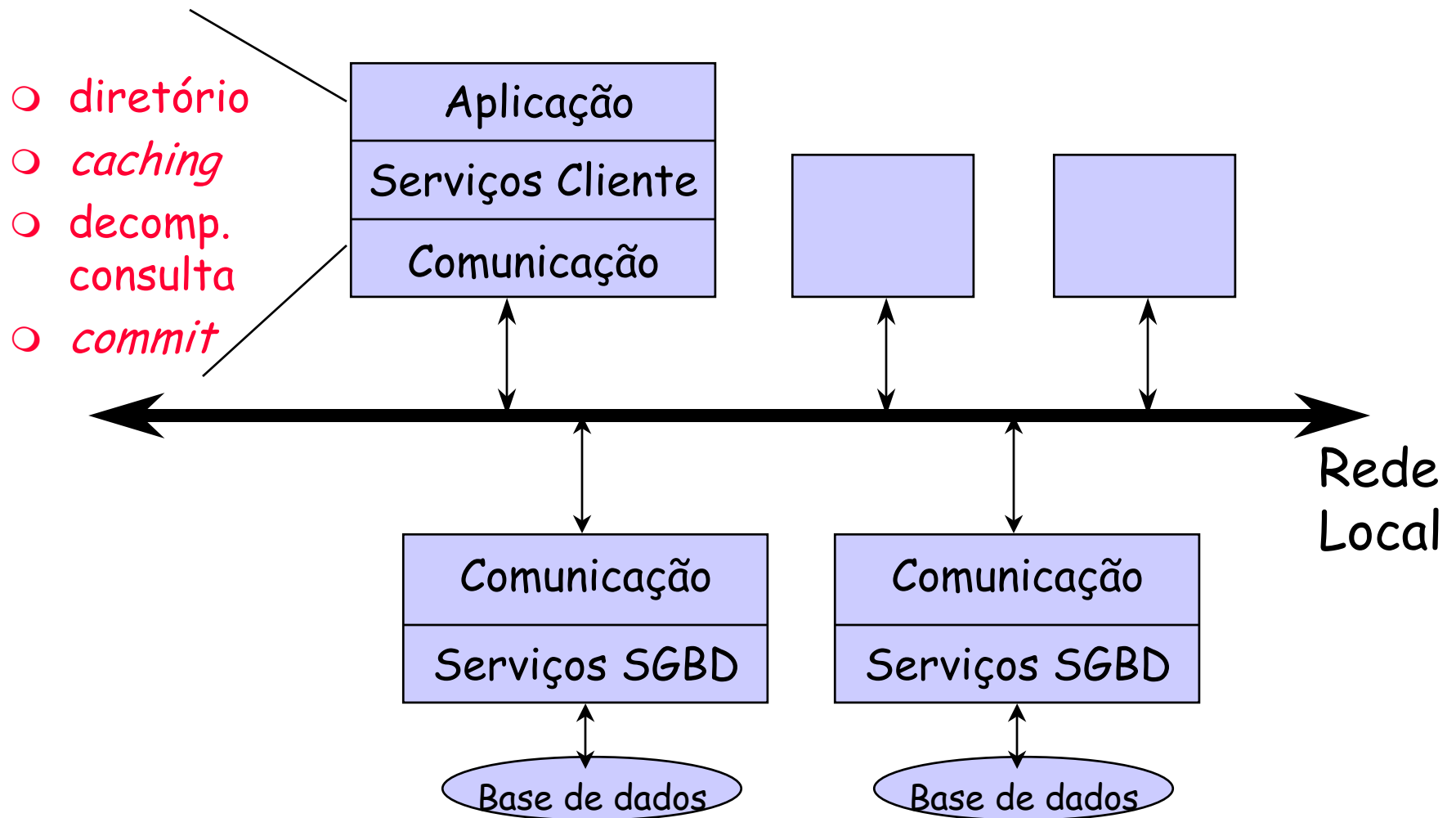
# Problemas com Muitos-Clientes/Um Servidor

- O servidor é um ponto de estrangulamento
- O servidor é um ponto de falha único (não há alternativas)
- O crescimento da base de dados é difícil

# Roteiro

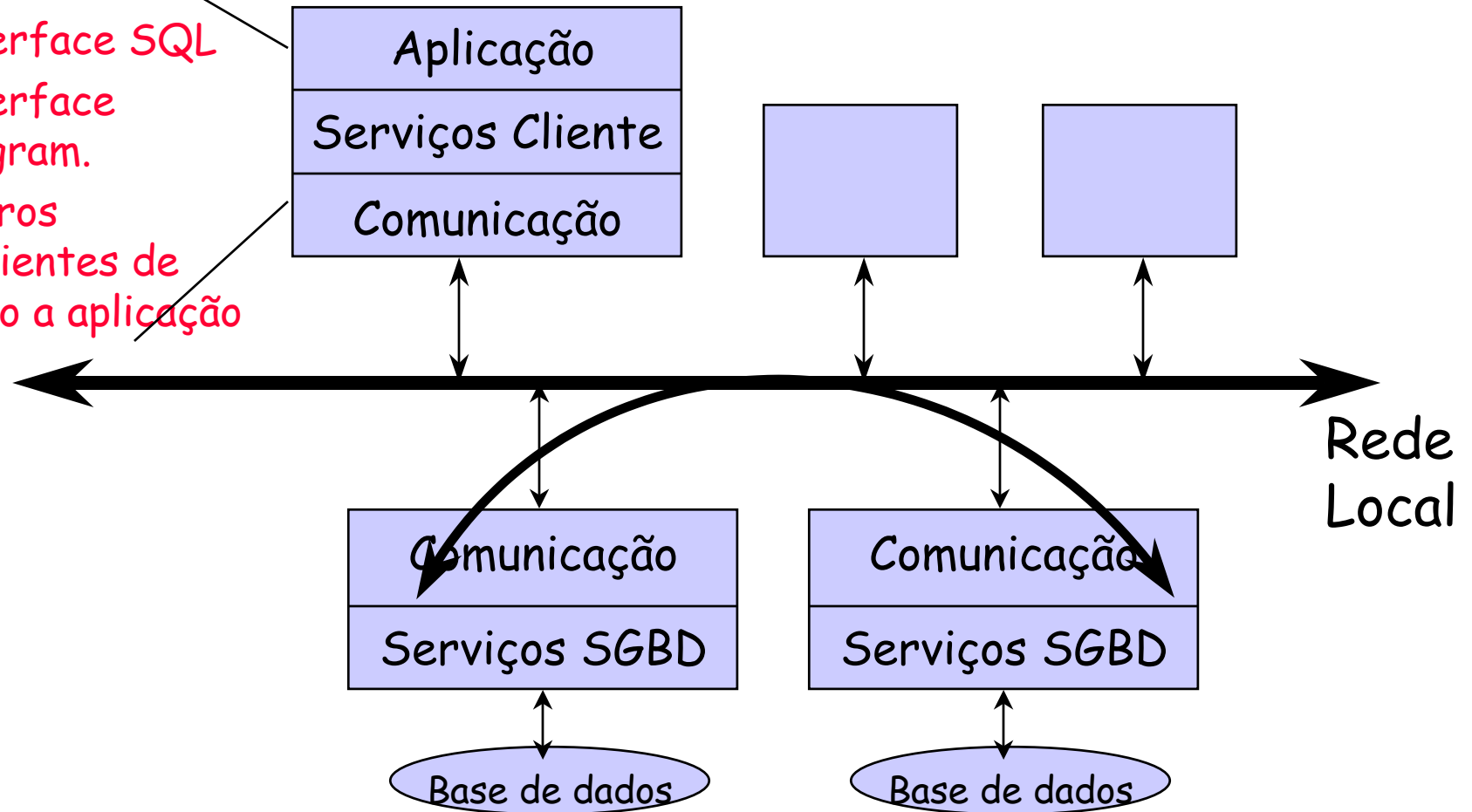
- ✓ Introdução a arquiteturas de SGBD
- ✓ SGBD (ANSI/SPARC)
- ✓ SGBD Distribuído
- Cliente / Servidor
  - ✓ Muitos Clientes / Um Servidor
  - Muitos Clientes / Muitos Servidores
  - Três camadas (servidor de aplicação)
- Sistemas Heterogêneos

# Muitos Clientes/ Muitos Servidores



# Servidor para Servidor

- Interface SQL
- Interface Program.
- Outros ambientes de apoio a aplicação



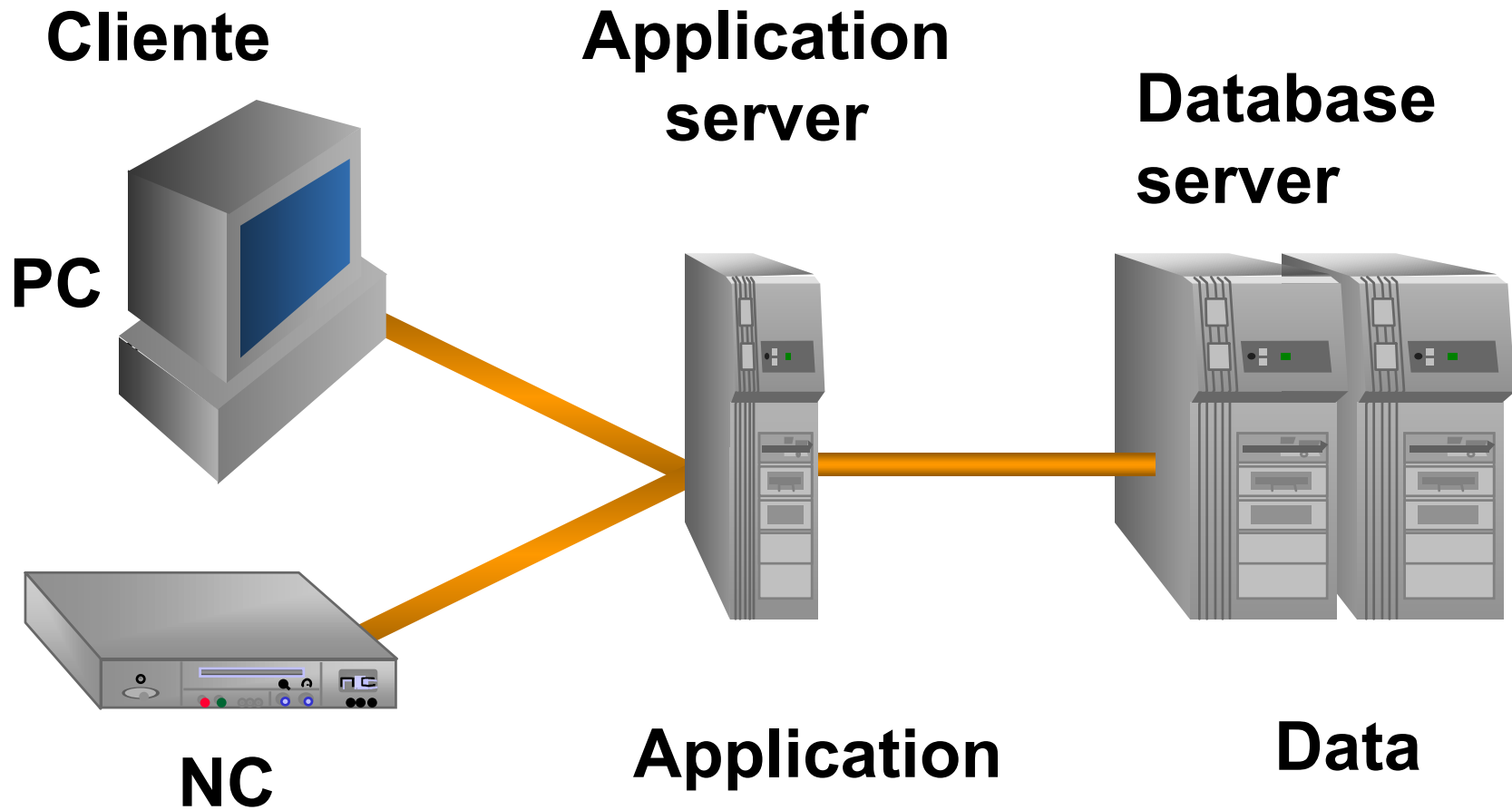
# Limitações de Cliente/ Servidor

- Operar é difícil
  - necessita de vários especialistas: SGBD, rede
  - ferramentas heterogêneas
  - dificuldade de evolução
    - aplicação muda => propaga em todos os clientes
    - cria nova base => replica *storage procedures*
- Custo de manutenção é alto

# Roteiro

- ✓ Introdução a arquiteturas de SGBD
- ✓ SGBD (ANSI/SPARC)
- ✓ SGBD Distribuído
- Cliente / Servidor
  - ✓ Muitos Clientes / Um Servidor
  - ✓ Muitos Clientes / Muitos Servidores
  - Três camadas (servidor de aplicação)
- Sistemas Heterogêneos

# Cliente/Servidor 3 camadas [Valduriez99]



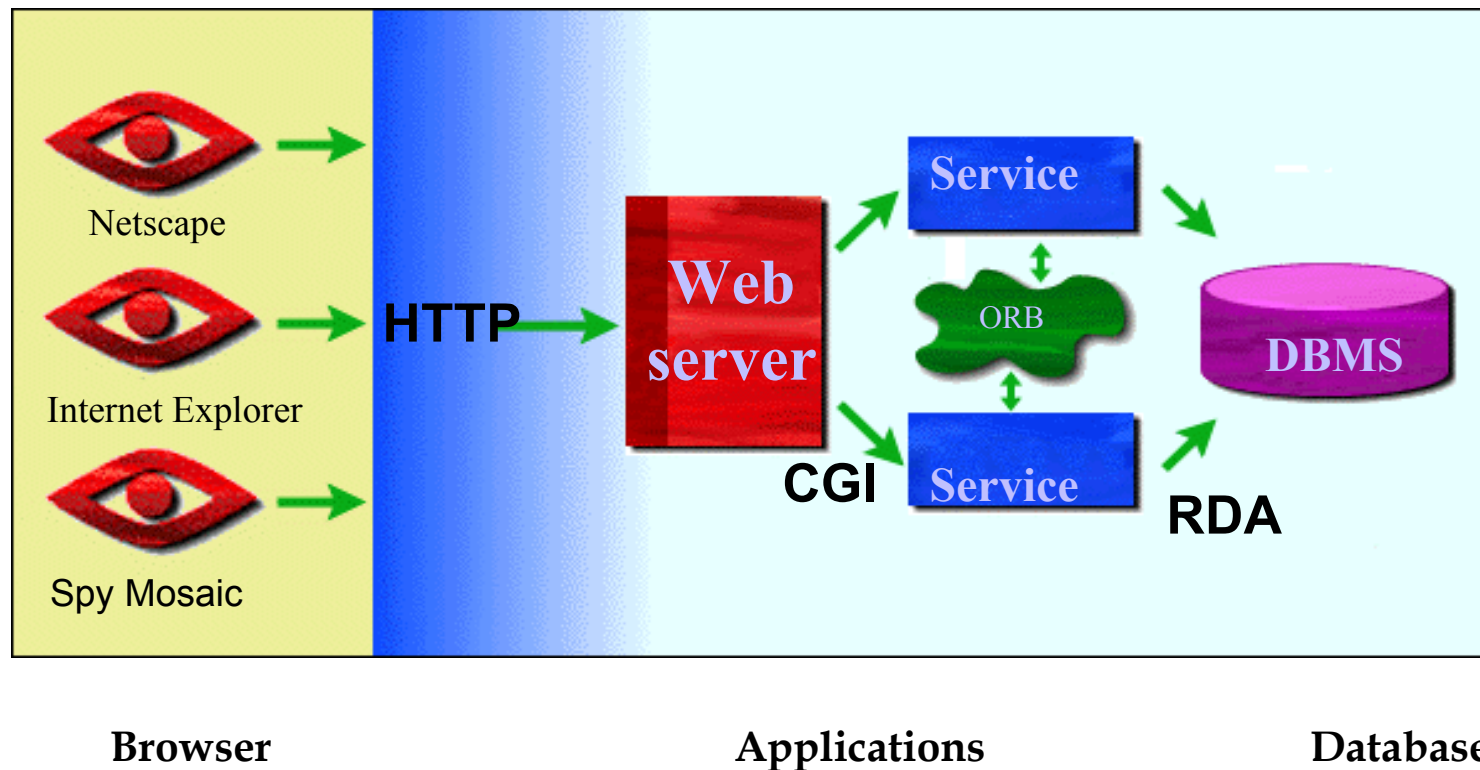


# Vantagens - três camadas (3-tier)

- Cliente Universal
  - padrão, *browser* multimedia
  - applets
  - portabilidade
- Servidor de aplicação
  - facilita manutenção e desenvolvimento
  - facilita acesso a múltiplos servidores de BD
- Naturalmente estende p/ *n camadas*

# Acesso ao SGBD por *Web Browser*

## 4 camadas [Valduriez99]



# Acesso ao SGBD por *Web Browser*

- Web browser se comunica com servidor Web usando HTTP
- Através do CGI (Common Gateway Interface) é possível invocar um programa executável (acesso SQL)
- Quando o SGBD executa a consulta o servidor da aplicação transforma as tuplas do resultado em páginas HTML

# Servidor de Aplicação

- **Simplex = servidor web**
  - aplicações pequenas, sem transações
- **Servidor Empresarial**
  - aplicações grandes e distribuídas
  - transações
    - apoio a padrões: CORBA
  - componentes
    - apoio a padrões: CORBA, COM, EJB
  - ferramentas de desenvolvimento

# Roteiro

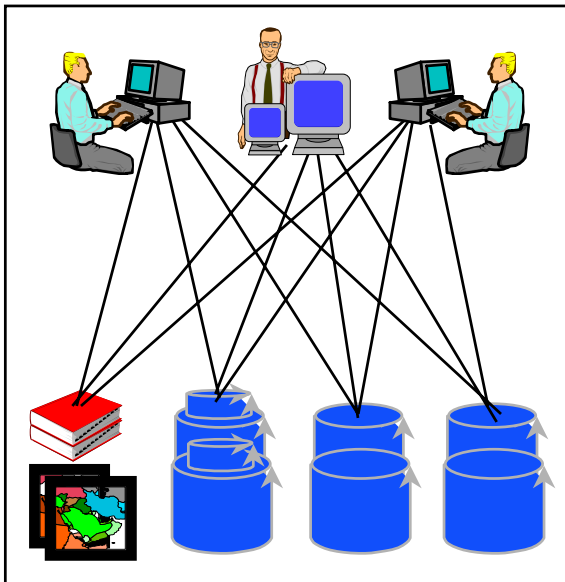
- ✓ Introdução a arquiteturas de SGBD
- ✓ SGBD (ANSI/SPARC)
- ✓ SGBD Distribuído
- ✓ Cliente / Servidor
- Sistemas Heterogêneos

# Integração de Sistemas Heterogêneos (sistemas legados)

- Ponto único de acesso a fontes de informações distribuídas e heterogêneas
- Visão homogênea sobre dados heterogêneos
- Acesso eficiente a dados relevantes
- Acesso uniforme via Internet - Intranet

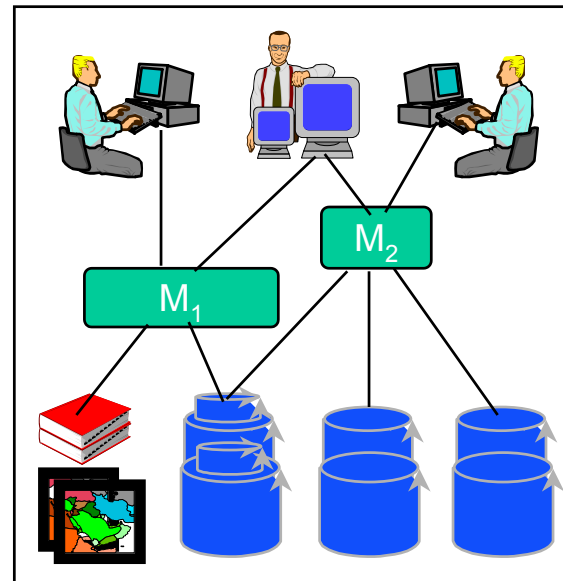
# Arquiteturas de integração

## Acoplam. fraco



Um módulo de acesso por aplicação

## Federado



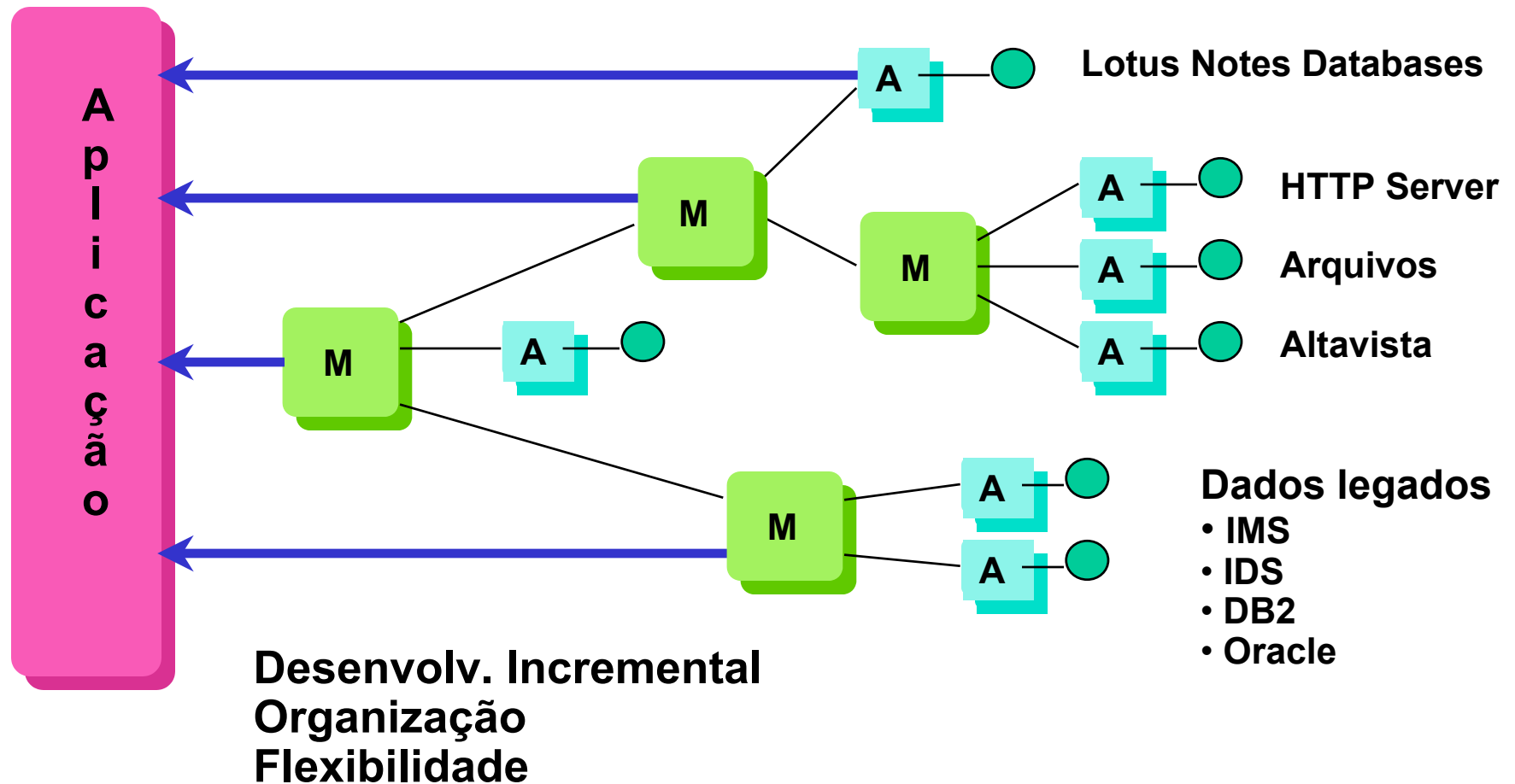
Um mediador por domínio

## Integrado



Um módulo global de integração

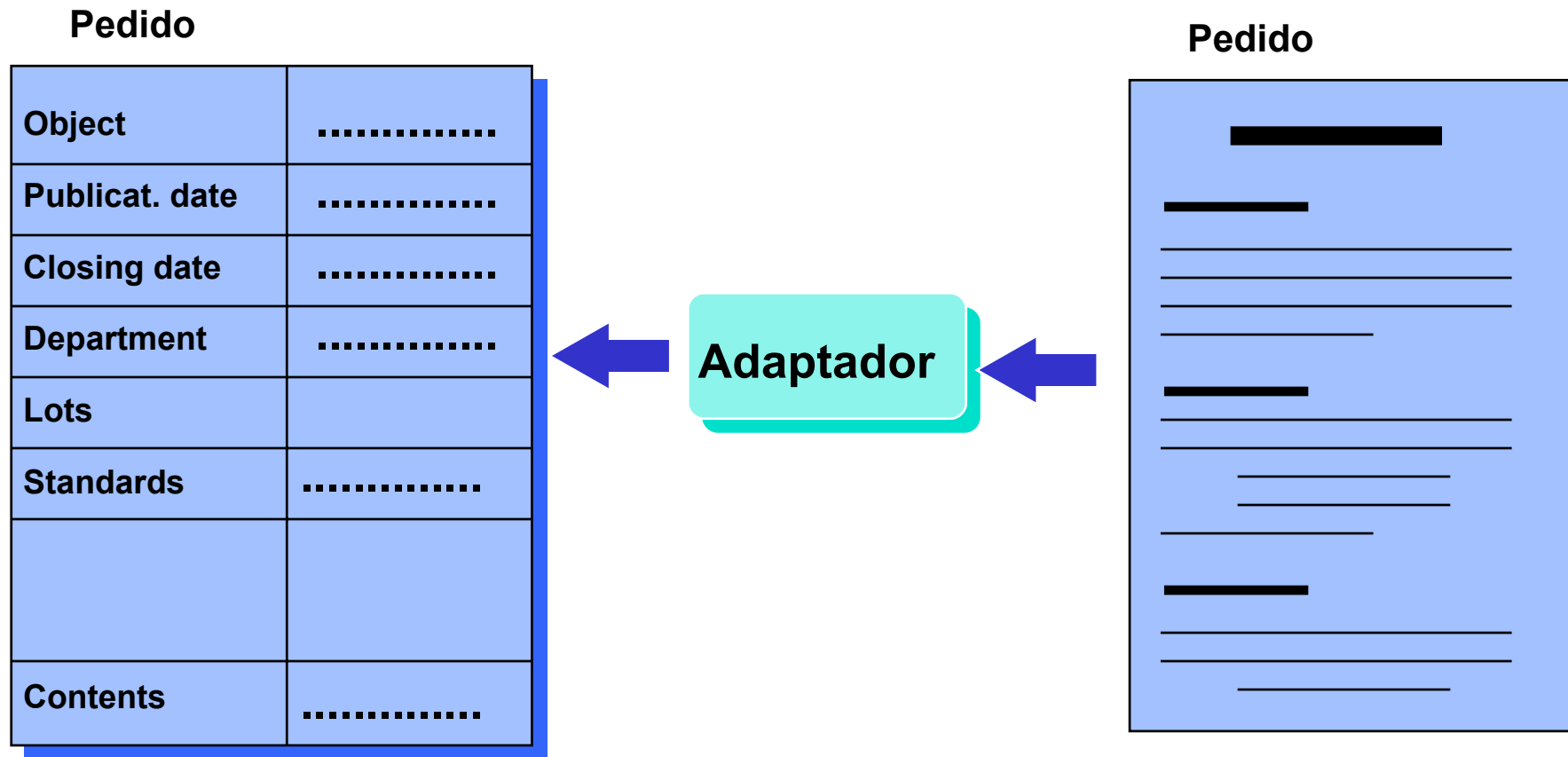
# Uso de mediadores com integração de domínios



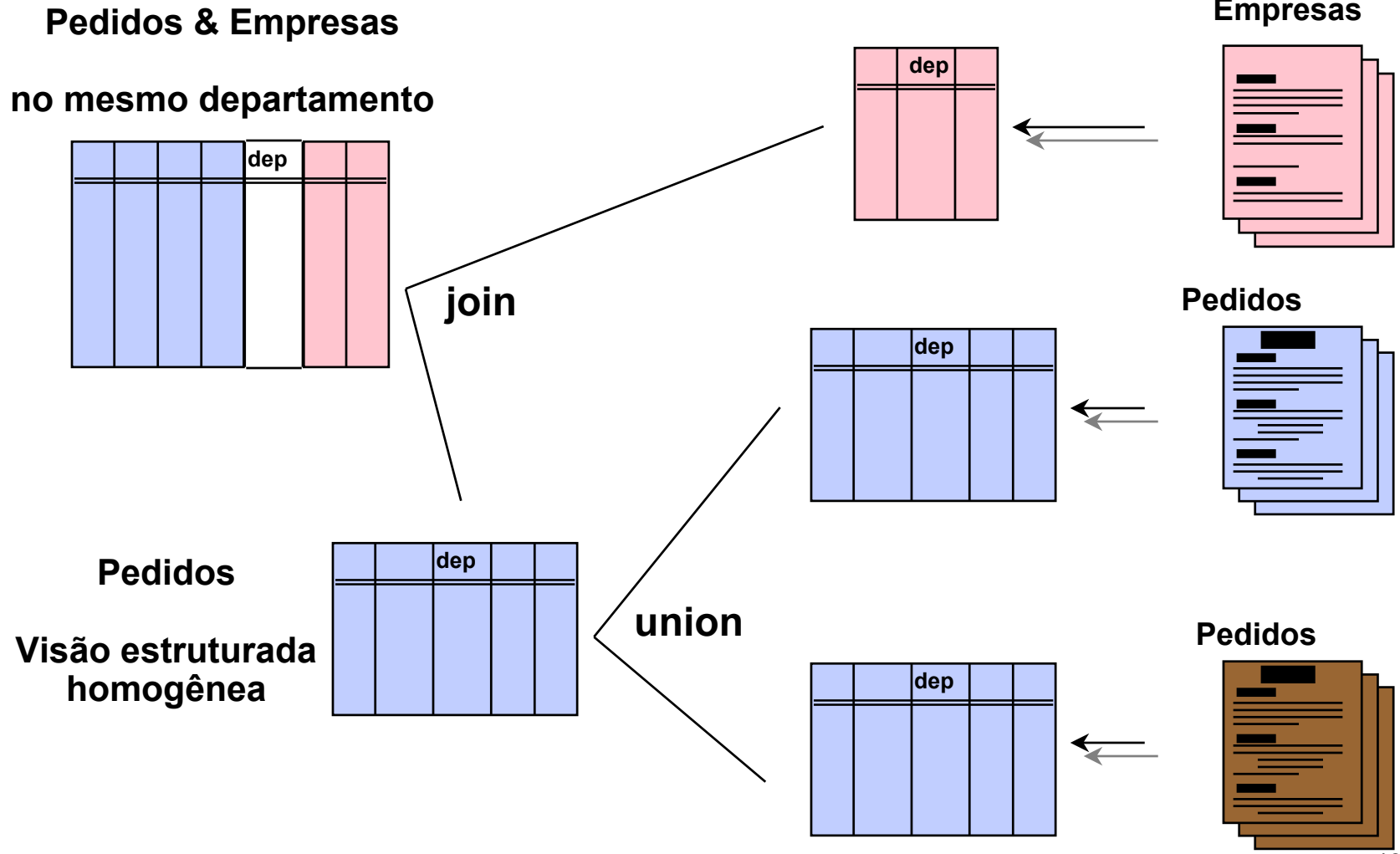


# Funções do Adaptador

## Extração e reestruturação de dados



# Exemplo- integração de informações

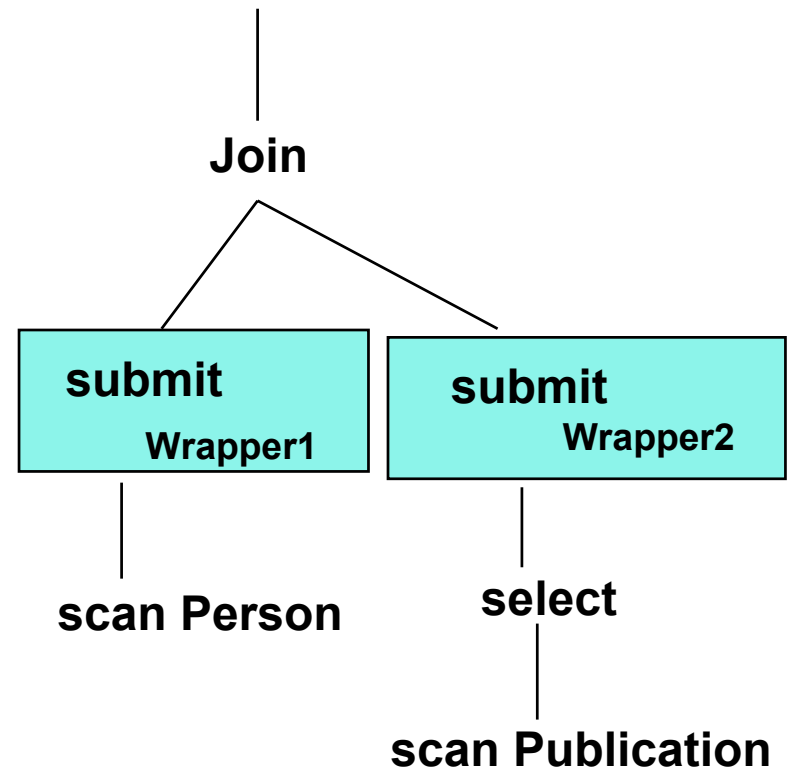
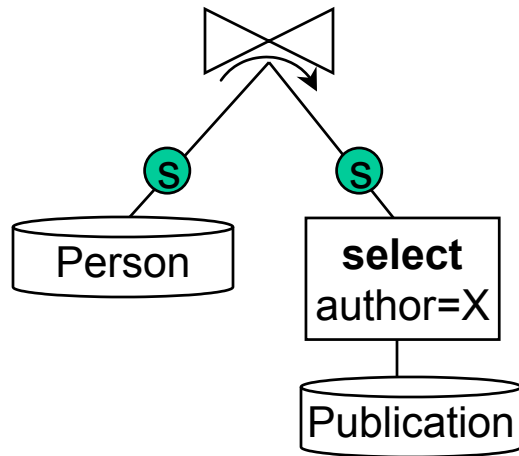


# Exemplo- integração de informações

Person(name, picture), Publication(author, ...)

```
select *  
from Publication pub, Person pers  
where pub.author = pers.name
```

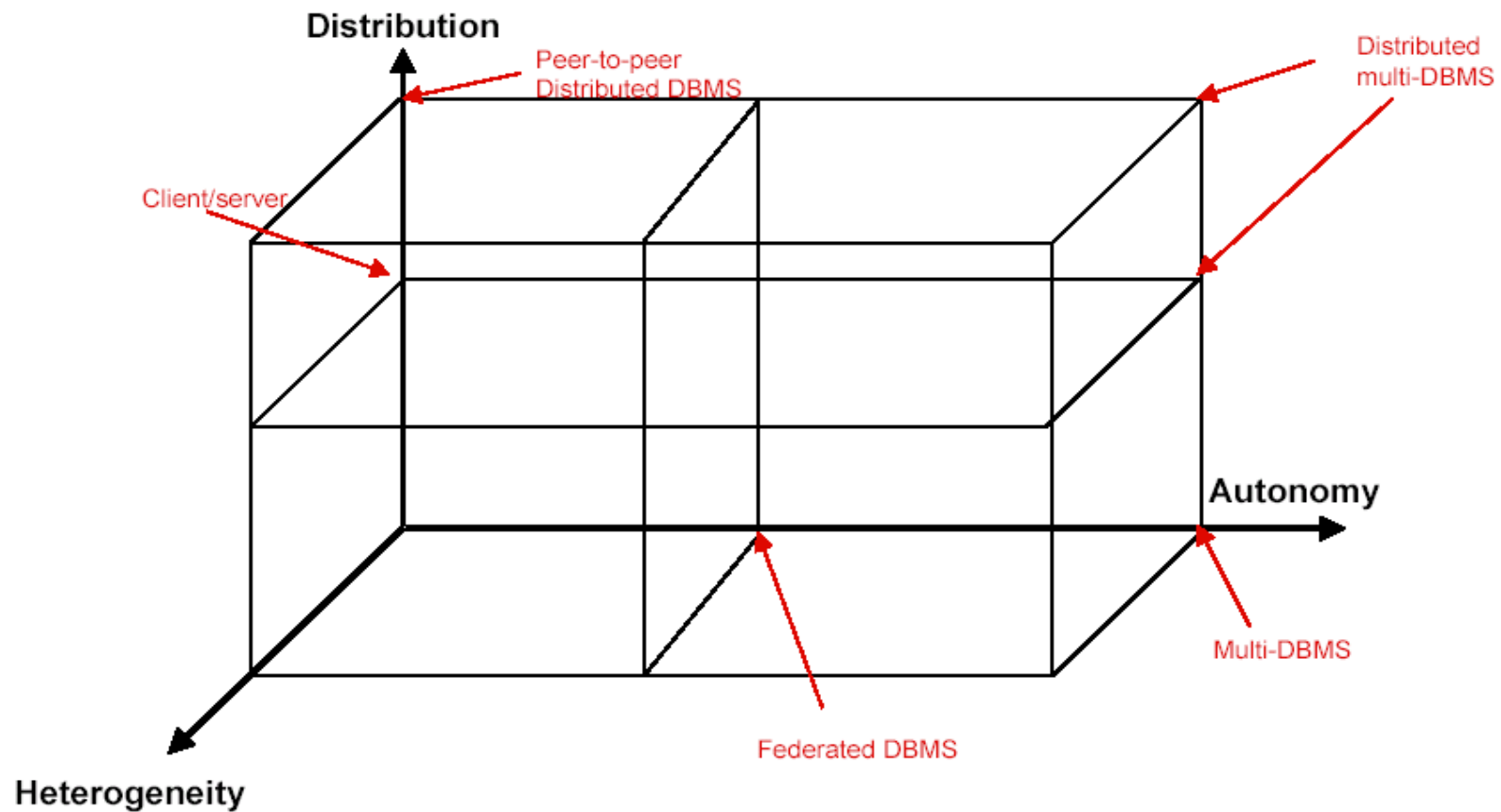
Plano de execução:



# Produtos de Integração de Info.

- Ferramentas Multidatabase com interfaces Web
  - DataJoiner (IBM)
  - Oracle/Star
  - EDA/SQL (Information Builders)
  - Open Database Exchange (B2Systems)
- Ferramentas recuperação info. p/ documentos Web
  - Harvest (U. Colorado)
  - Strudel (ATT)
  - AltaVista (Digital)
- Mediadores na integração BD e documentos
  - Mediation products (Dyade - Bull and Inria)
  - Tsimmis (Stanford U.)

# Alternativas: Implementação SGBDD



# Dimensões do Problema

## ○ Distribuição

- Se os componentes do sistema estão localizados na mesma máquina ou não

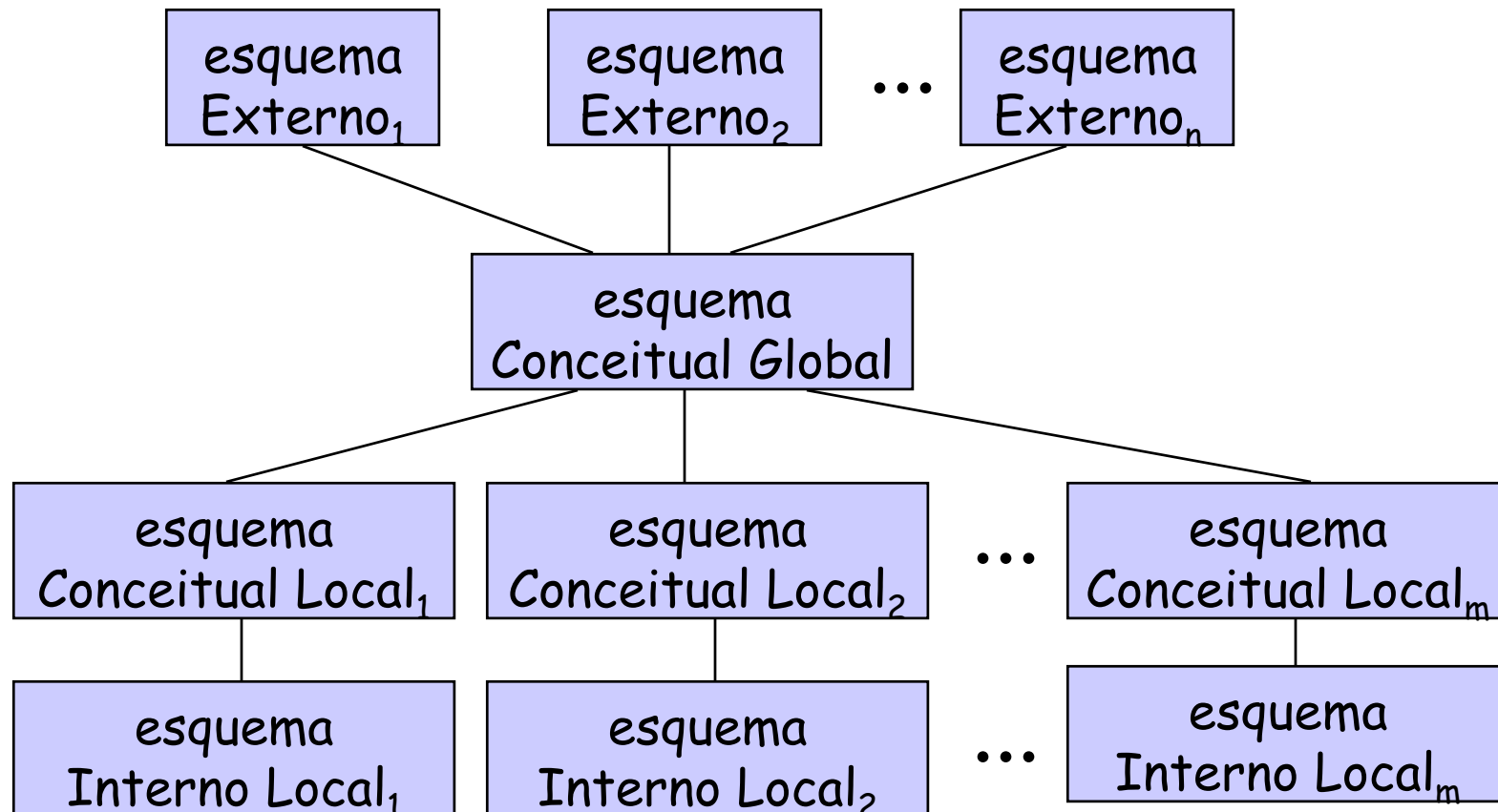
## ○ Heterogeneidade

- Vários níveis (hardware, comunicação, sistema operacional)
- SGBD importantes:
  - modelo de dados, linguagem de consultas, gerência de transações, algoritmos

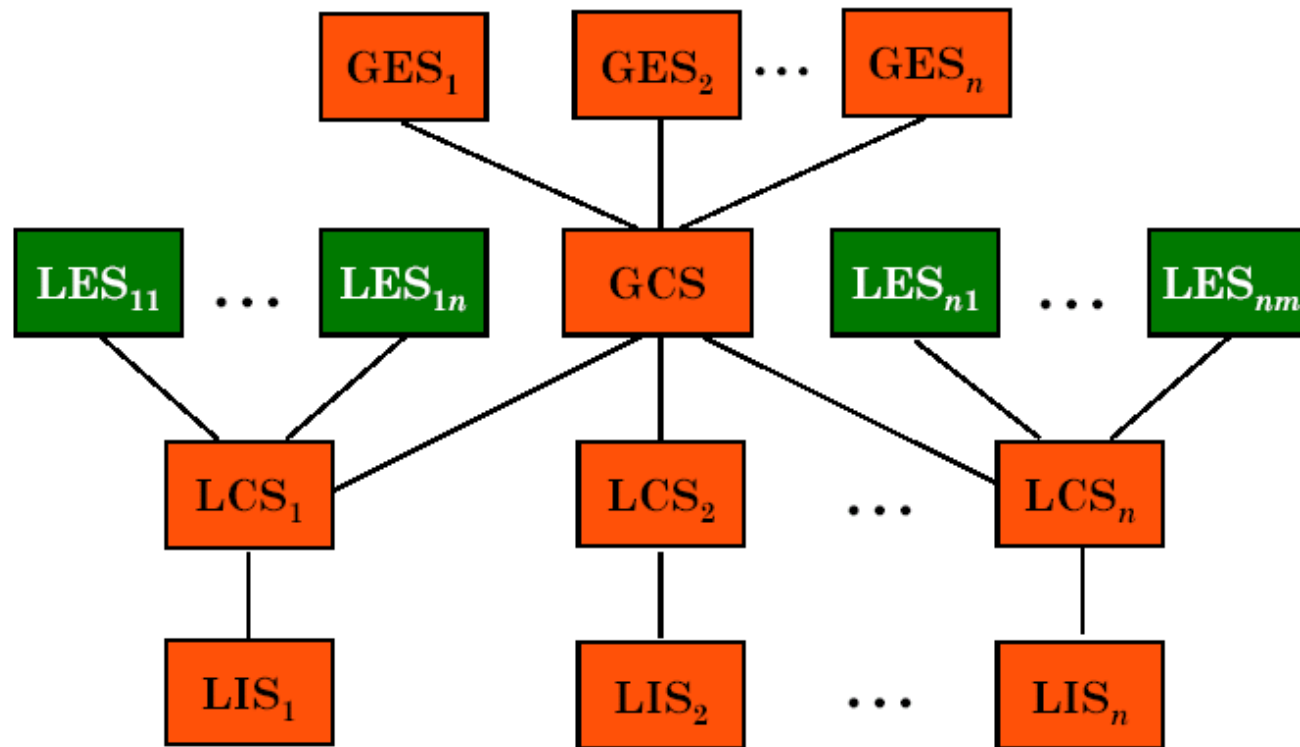
## ○ Autonomia

- Não é bem compreendida
  - Projeto, Comunicação e Execução

# Arquitetura de SGBD-D (dados)

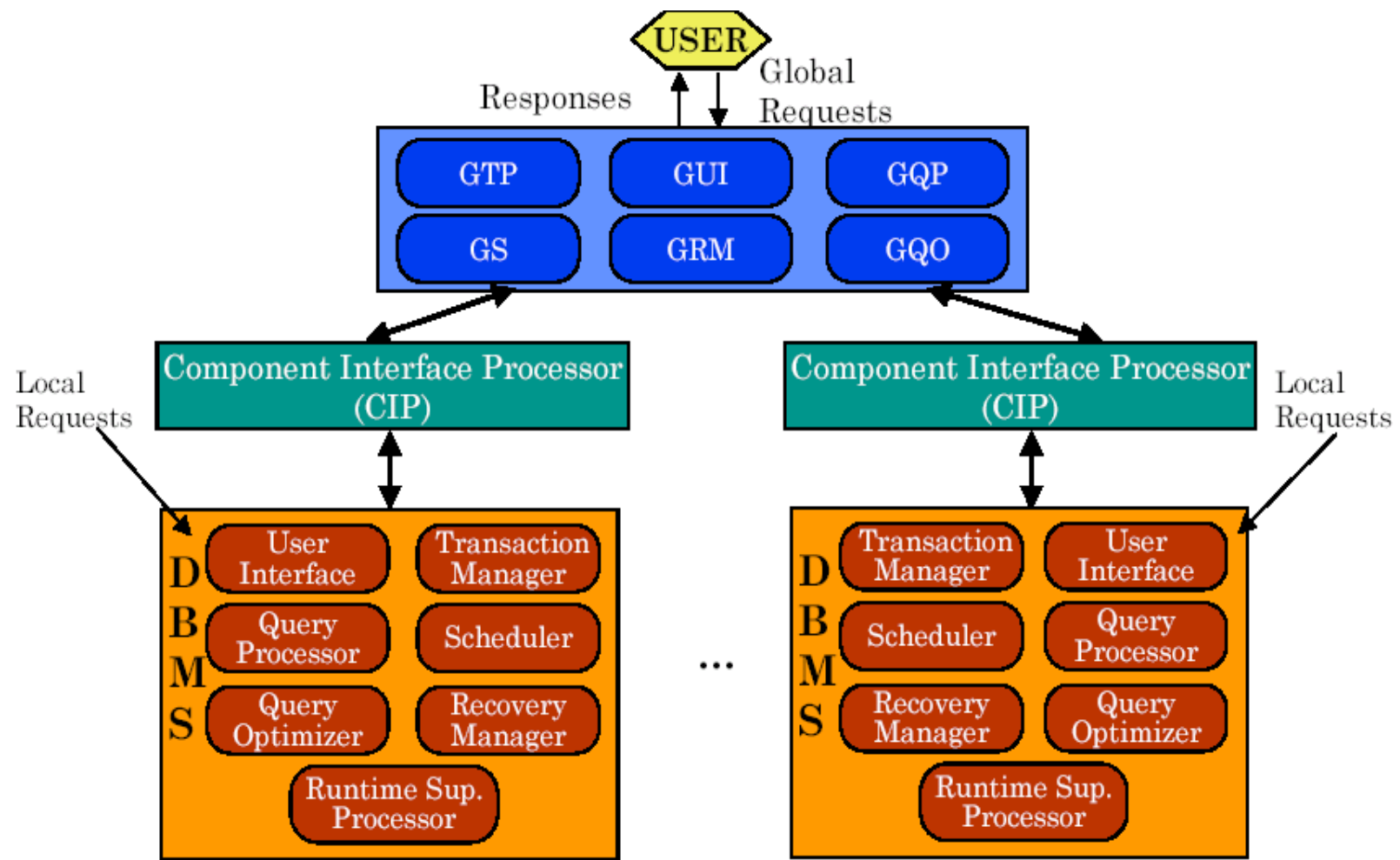


# Arquitetura de Multi-SGBD (dados)





# Arquitetura de Multi-SGBD (comp.)



# Bibliografia

- Introdução a arquiteturas de SGBD
- SGBD (ANSI/SPARC)
- SGBD Distribuído
  - Ozsu,M., Valduriez,P., *Principles of Distributed Database Systems*, Cap 4, Prentice-Hall, 1999.
- Cliente / Servidor
  - Ozsu,M., Valduriez,P., *Principles of Distributed Database Systems*, Cap 4, 16 Prentice-Hall, 1999.
  - Valduriez, P. *Application Servers*, Apresentação PUC, 1999.
  - *Oracle Application Server 4.0 Overview*, White paper,1998.
- Sistemas Heterogêneos
  - Ozsu,M., Valduriez,P., *Principles of Distributed Database Systems*, Cap 4, 16 Prentice-Hall, 1999.