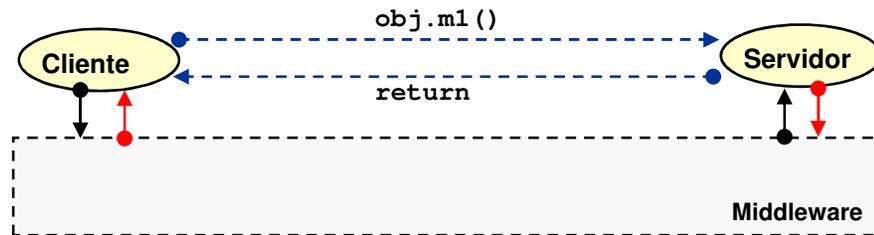


# Modelos de Middleware

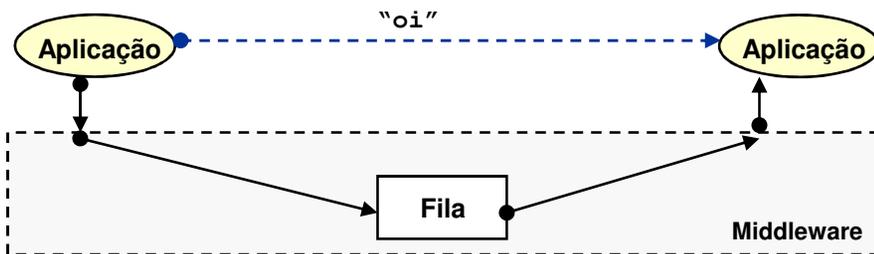
Universidade Federal de Pernambuco  
Centro de Informática  
Recife - Pernambuco



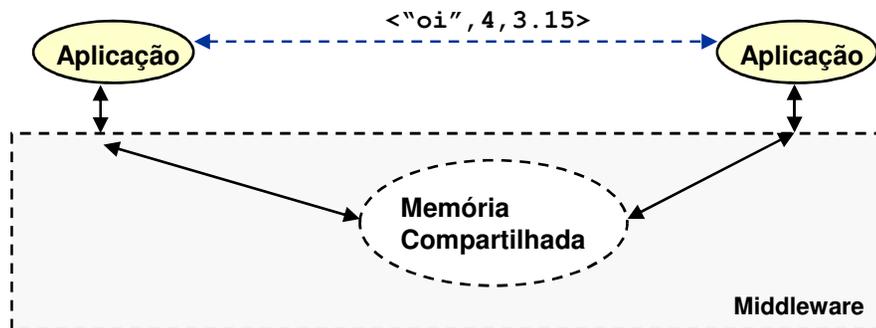
# Relembrando...



- Baseado em RPC



- Passagem de mensagem



- Memória compartilhada

# Modelos de Middleware

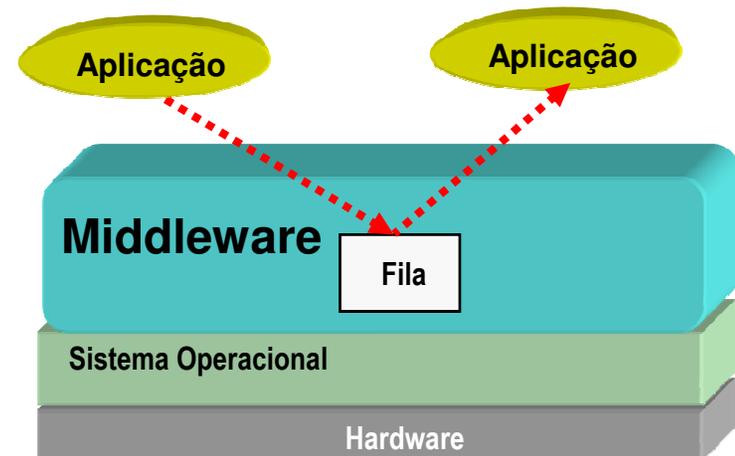
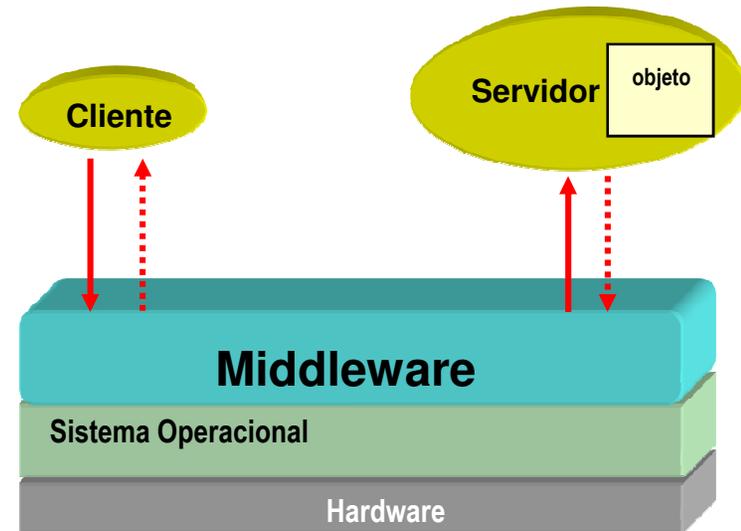
- há várias classificações
  - Vinoski & Talarian
  - Bakken & Britton & Tanenbaum & Abdul-Fatah
  - modelos de middleware da “próxima geração”
  - Emmerich [2000]
- baseadas em diversas características do middleware
- Não há uma mais usada, nem amplamente adotada

# Classificação 1

- parâmetros para classificação
  - tipo de comunicação
  - linguagens p/ construção da aplicação
  - forma de disponibilização
  - ambiente de execução

# Classificação 1 – Tipo de Comunicação

- comunicação síncrona
  - baseada em chamadas de procedimentos
  - centrada “objeto/função”
  - e.g., CORBA, RMI
- comunicação assíncrona
  - baseado em filas
  - “centrada na informação”
  - e.g., JMS



# Classificação 1 – Linguagem da Aplicação

## ■ diversas linguagens

- ◆ IDL para descrição da interface do serviço
- ◆ linguagem de programação para implementação do serviço
- ◆ e.g., CORBA (IDL CORBA) + Java

## ■ única linguagem

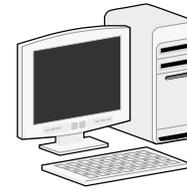
- ◆ interface e implementação do serviço usando uma única linguagem
- ◆ e.g., RMI

# Classificação 1 - Forma de Disponibilização

- proprietário
  - “no futuro o meu middleware será padrão de facto”
  - e.g., MQSeries (IBM)
- baseado em padrões
  - nem sempre é possível padronizar tudo, burocracia, político
  - e.g., CORBA
- código aberto
  - implementação aberta
  - pode ser baseado em padrão
  - e.g., JBOSS

# Classificação 1 - Ambiente de Execução

- computadores pessoais
  - automação do negócio
  - suporte à aplicações genéricas
  - altamente **reconfiguráveis**
  - facilidade de **integração** é fundamental
- embarcados
  - suporte à aplicações **específicas**
  - **restrições** de capacidade de processamento
  - desempenho e **tempo-real** são fundamentais



# Classificação 2

- parâmetro de classificação:  
baseada nos produtos comerciais
- monitores de processamento de transação
  - e.g., monitores de TP
- chamadas remotas de procedimento
  - e.g, RPC
- *baseadas em brokers*
  - e.g., CORBA
- “caseiras”
  - e.g., este middleware que mostrei em aula
- orientados a mensagem
  - passagem de mensagem (*publish/subscribe*)
  - enfileiramento de mensagens (PTP)

# Classificação 3

- parâmetros para classificação

- abstrações fornecidas para a programação

- ◆ tuplas distribuídas, procedimentos, fila de mensagem, objetos

distribuídos

- tipo de heterogeneidade suportada

- ◆ sistema operacional, linguagem de programação, rede

(cabeadas, sem-fio), hardware

# Classificação 4

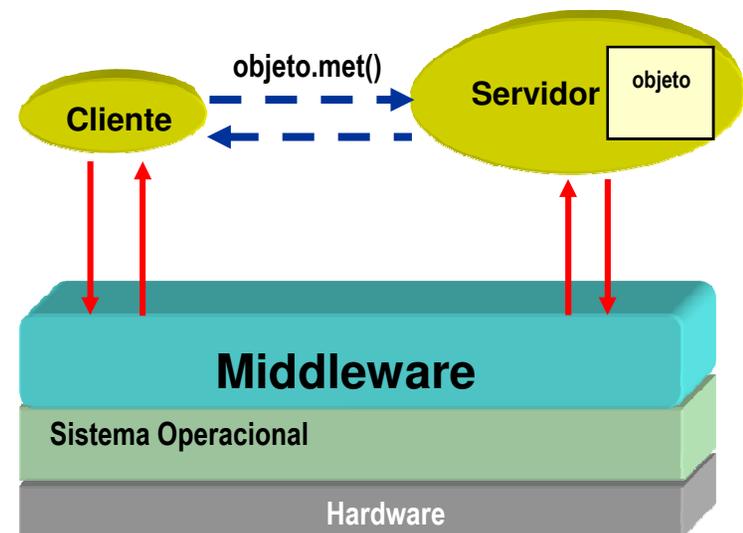
- parâmetros para classificação
  - tipos de entidades estão se comunicando
    - ◆ e.g., processos, objetos, programas, componentes
  - como as entidades de comunicam
    - ◆ request/reply (cliente-servidor)
    - ◆ peer-to-peer
    - ◆ push (publish/subscribe)
  - interface de programação
    - ◆ APIs
    - ◆ bibliotecas
    - ◆ frameworks

# Classificação 5

- parâmetro de classificação: tipo de mediação
- mediação implícita
- mediação explícita

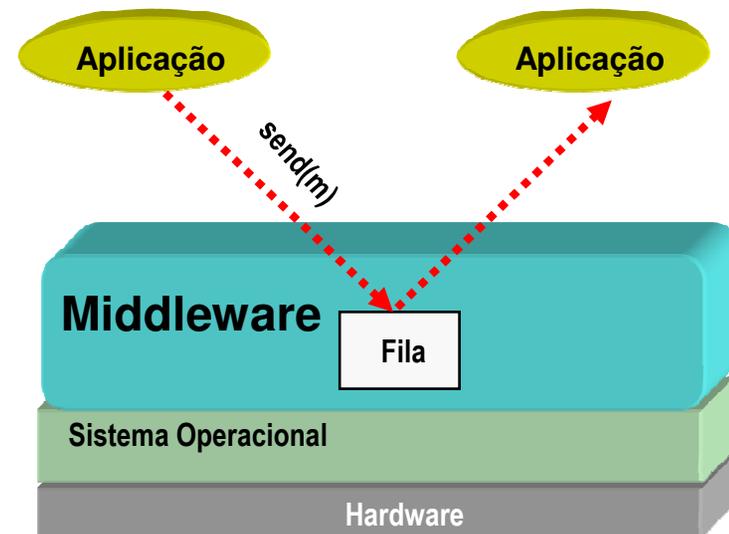
# Classificação 5 – Mediação Implícita

- os componentes não “sabem” da intermediação do middleware
- os componentes fornecem interfaces



# Classificação 5 - Mediação Explícita

- os componentes “sabem” da intermediação do middleware
- os componentes não fornecem interfaces para acesso



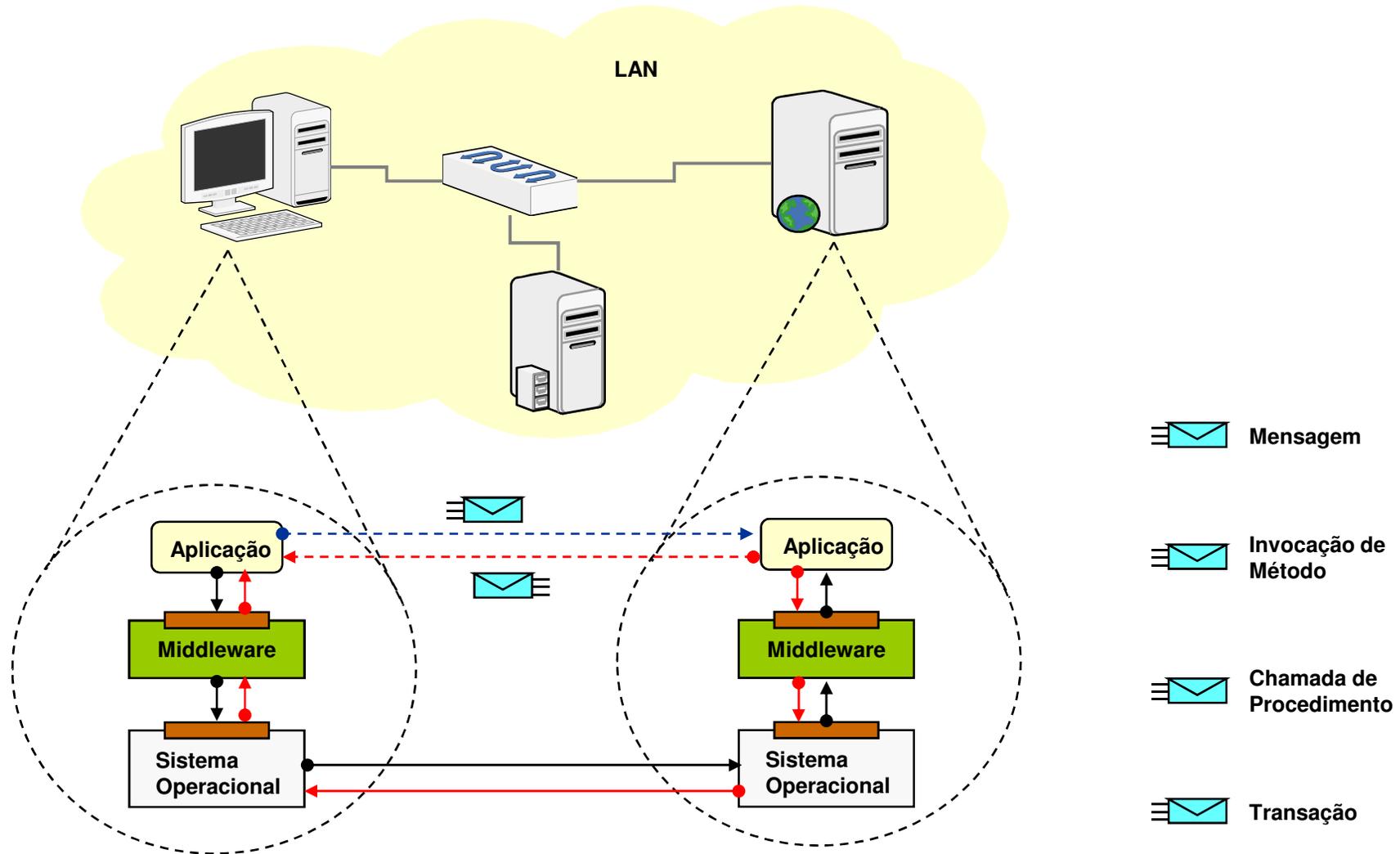
# Classificação 6 (Middleware da Próxima Geração)

- “classificados” de acordo com características do domínio da aplicação
  - adaptativos / reconfiguráveis / reflexivos
  - embarcados
  - \*-aware
    - ◆ deixam de ser “*black-box*”

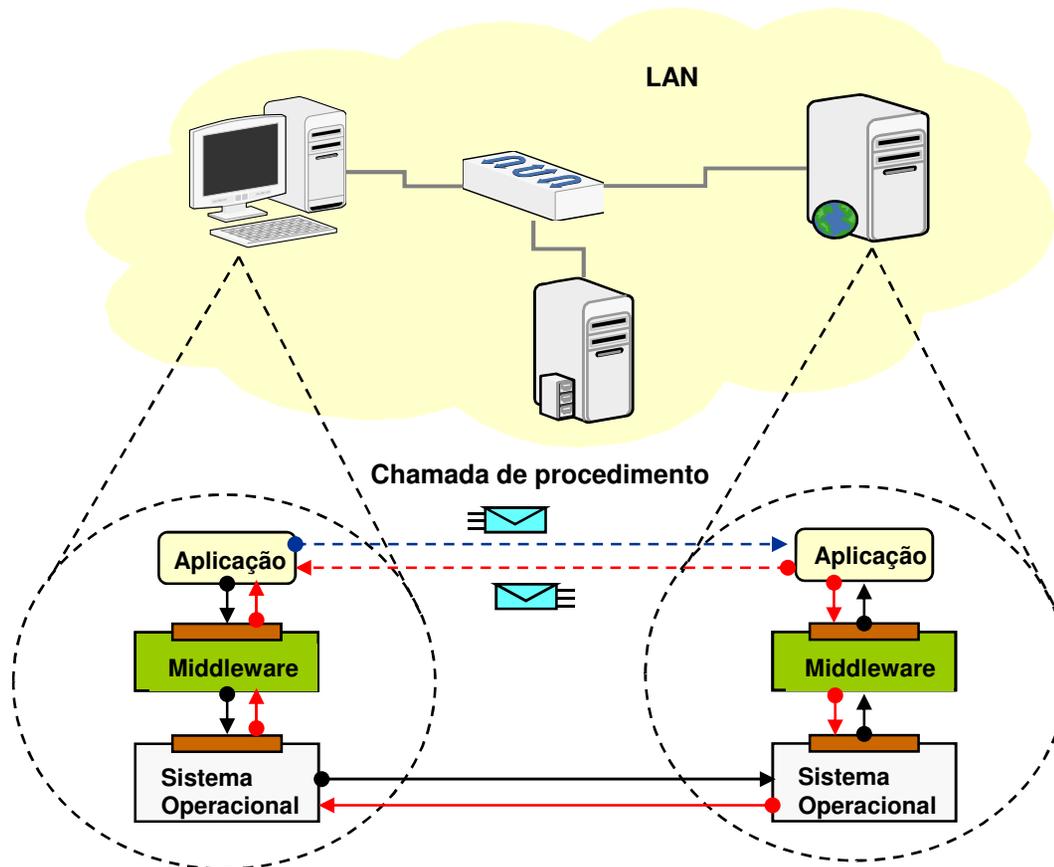
# Classificação 7

- parâmetro: baseada no tipo de primitiva que o middleware fornece para interação
  - transacional (transações)
  - orientado a mensagem (passagem de mensagem)
  - procedural (chamada remota de procedimento)
  - objeto (invocação de objeto remoto)

# Classificação 7

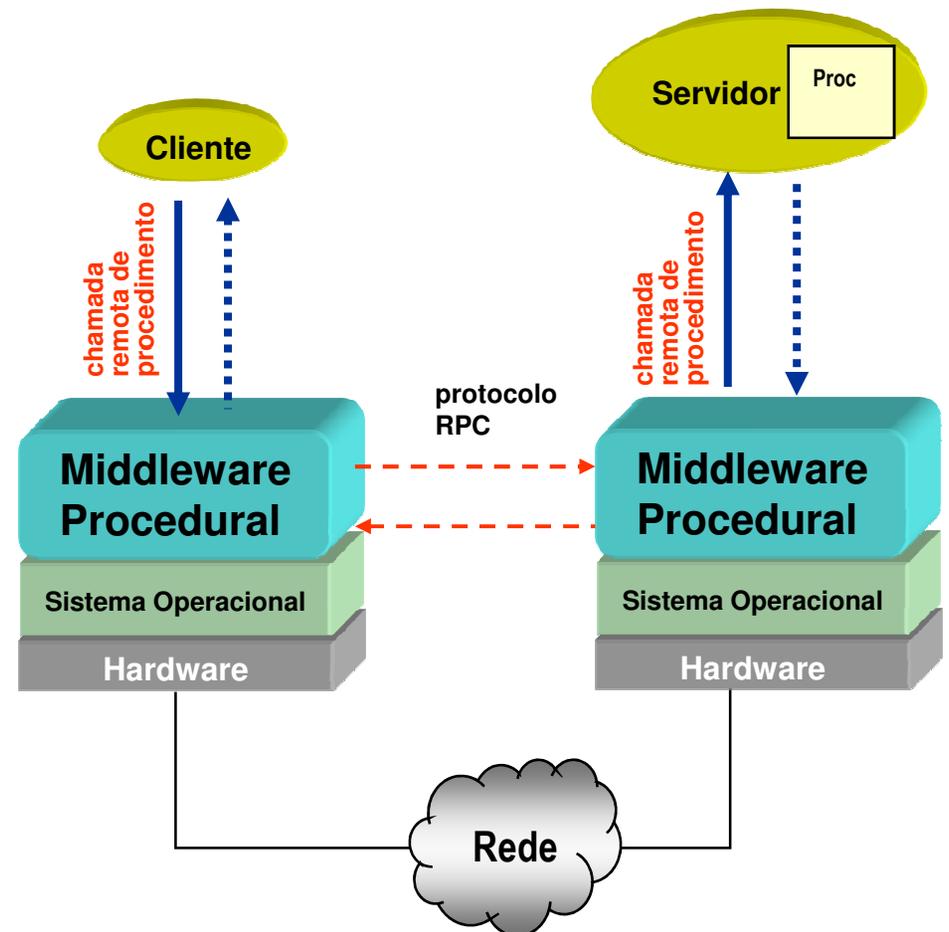


# Classificação 7- Middleware Procedural



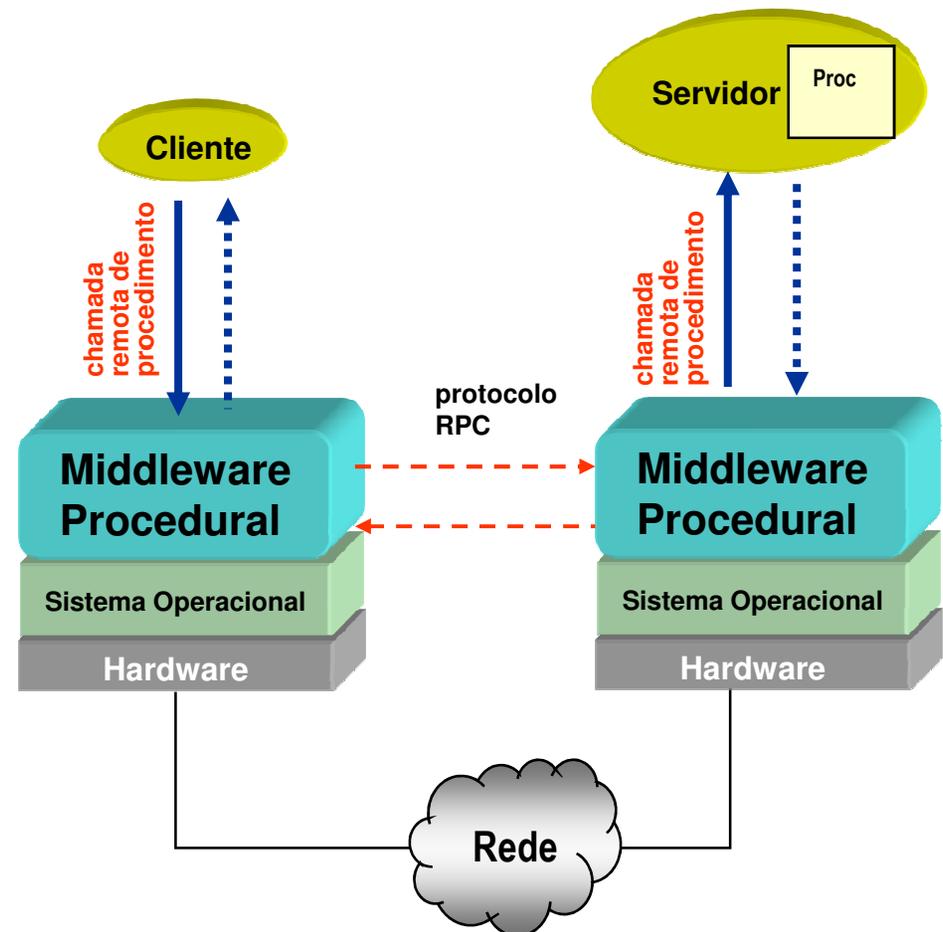
# Classificação 7- Middleware Procedural

- Primitiva de interação: chamada remota de procedimento (1-1)
- comunicação síncrona suportada naturalmente
- protocolo request/wait-for-reply
- e.g., DCE (Distributed Computing Environment) e RPC
- ativação do procedimento: permanente ou por demanda (*inetd* no unix)

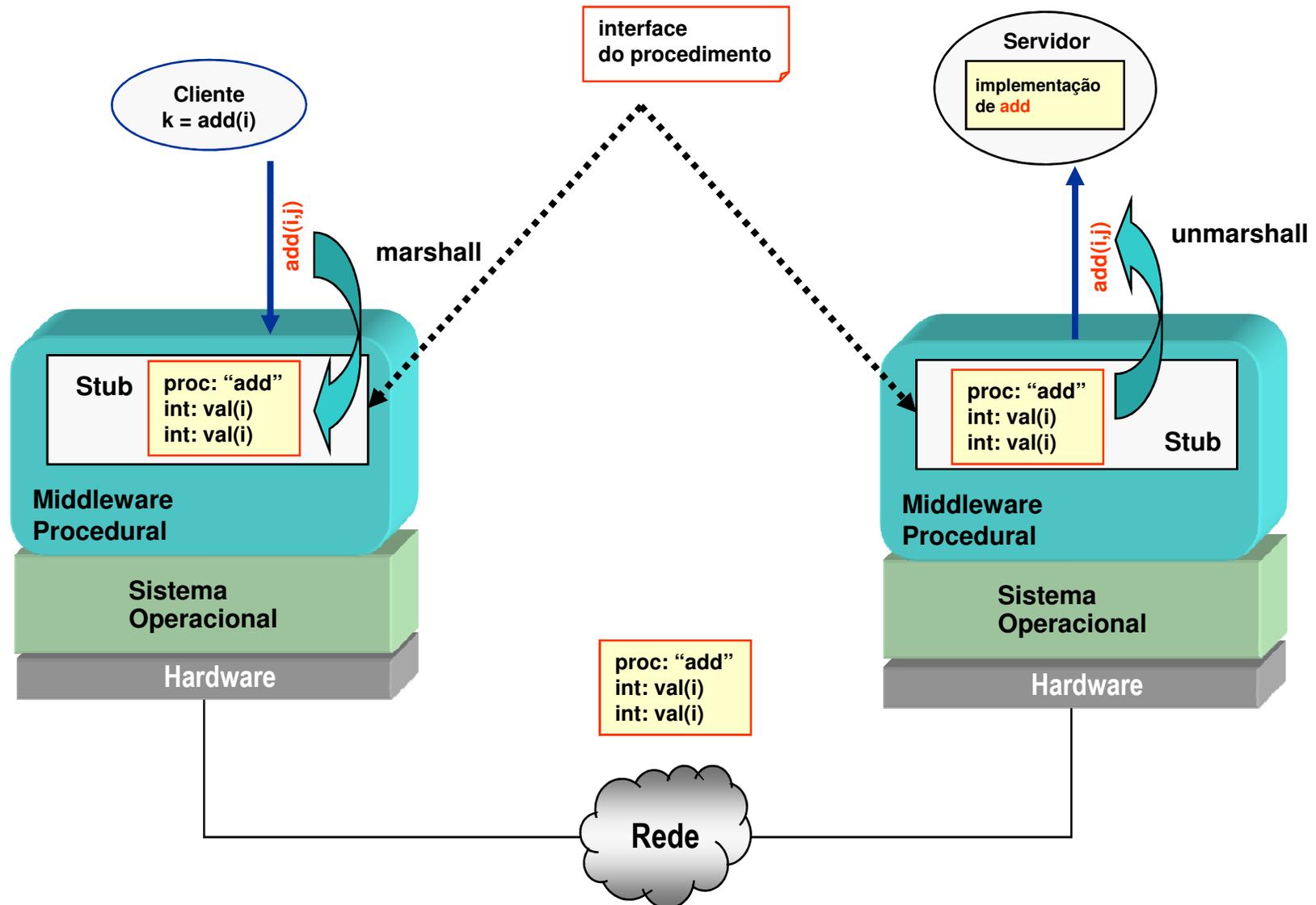


# Classificação 7- Middleware Procedural

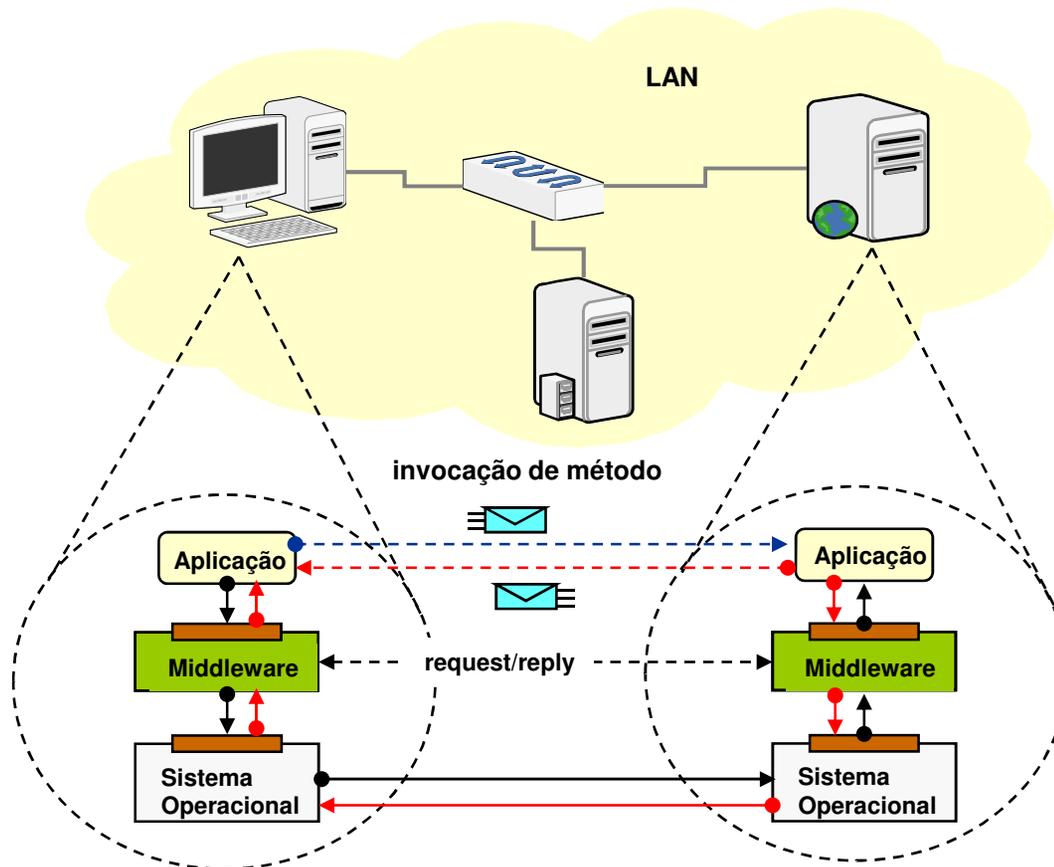
- semântica da chamada
  - *at-most once*
  - *exactly-once* e transação não suportadas
- os tipos usados como parâmetros são padronizados
- definição de interfaces independentes da linguagem de implementação



# Classificação 7 – Middleware Procedural

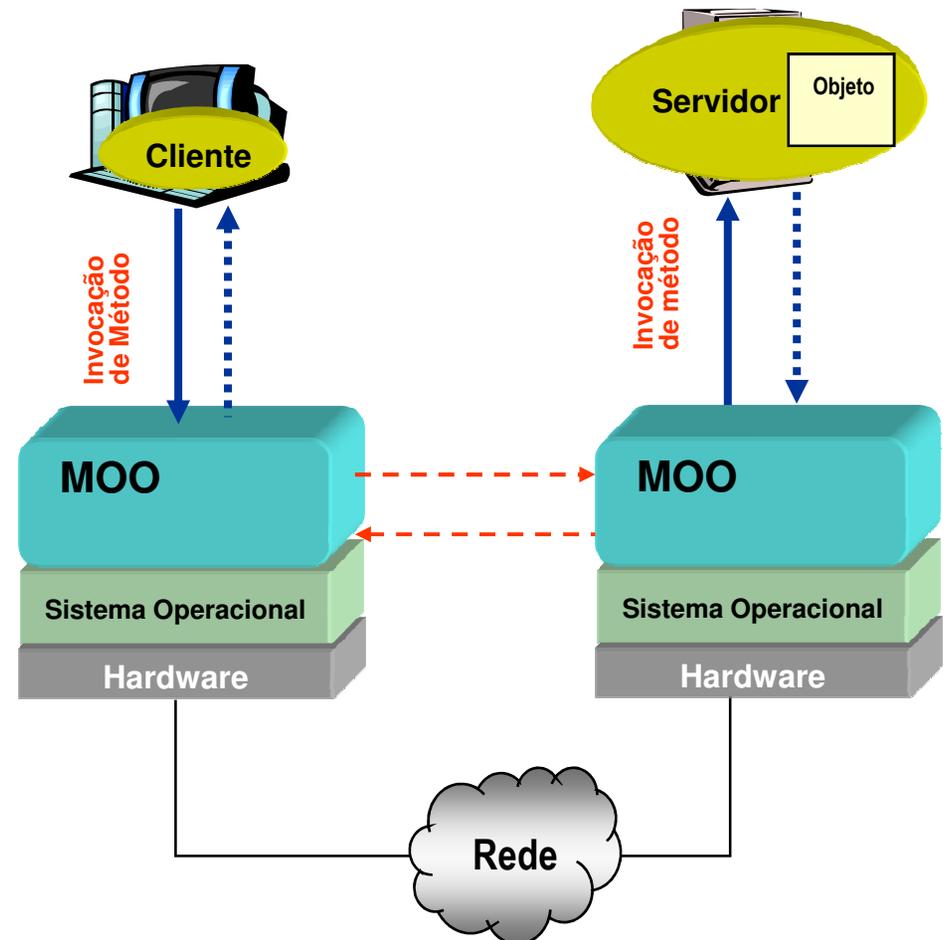


# Classificação 7 - Middleware Orientado a Objetos

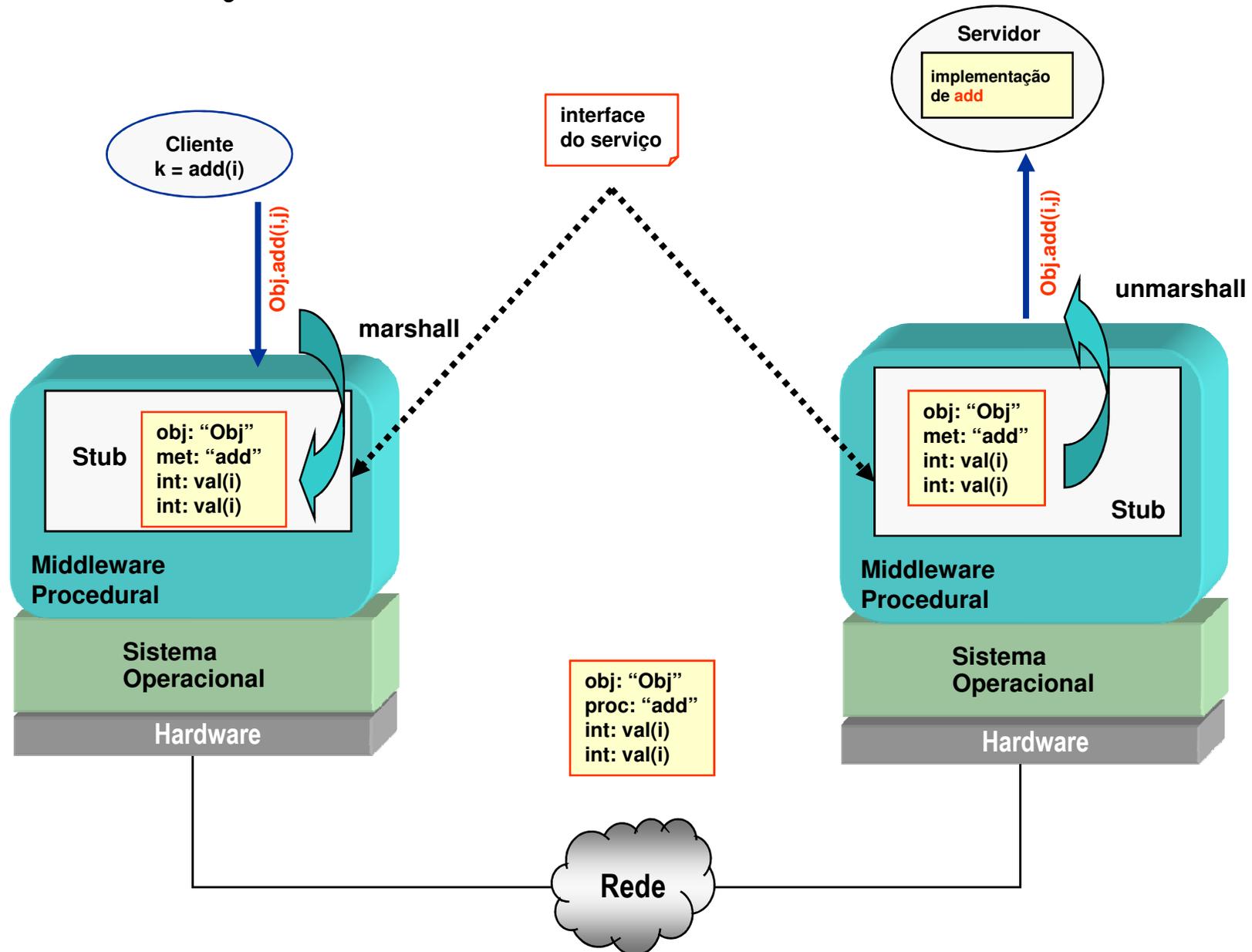


# Classificação 7- Middleware Orientado a Objetos

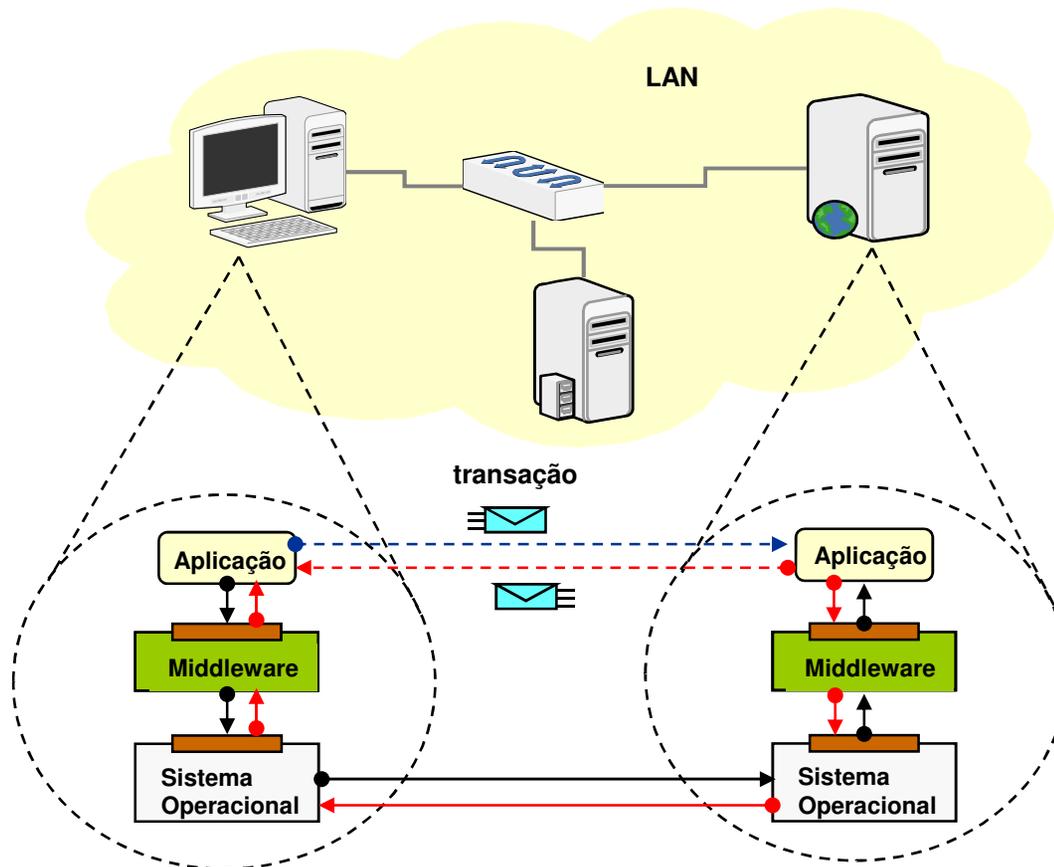
- primitiva de interação: invocação de método (1-1, at-most-once)
- evolução do middleware procedural
- comunicação síncrona entre objetos distribuídos suportada naturalmente
- interface dos serviços descritas por linguagens específicas (IDLs)
- e.g., CORBA, RMI



# Classificação 7- Middleware Orientado a Objetos



# Classificação 7 – Middleware Transacional

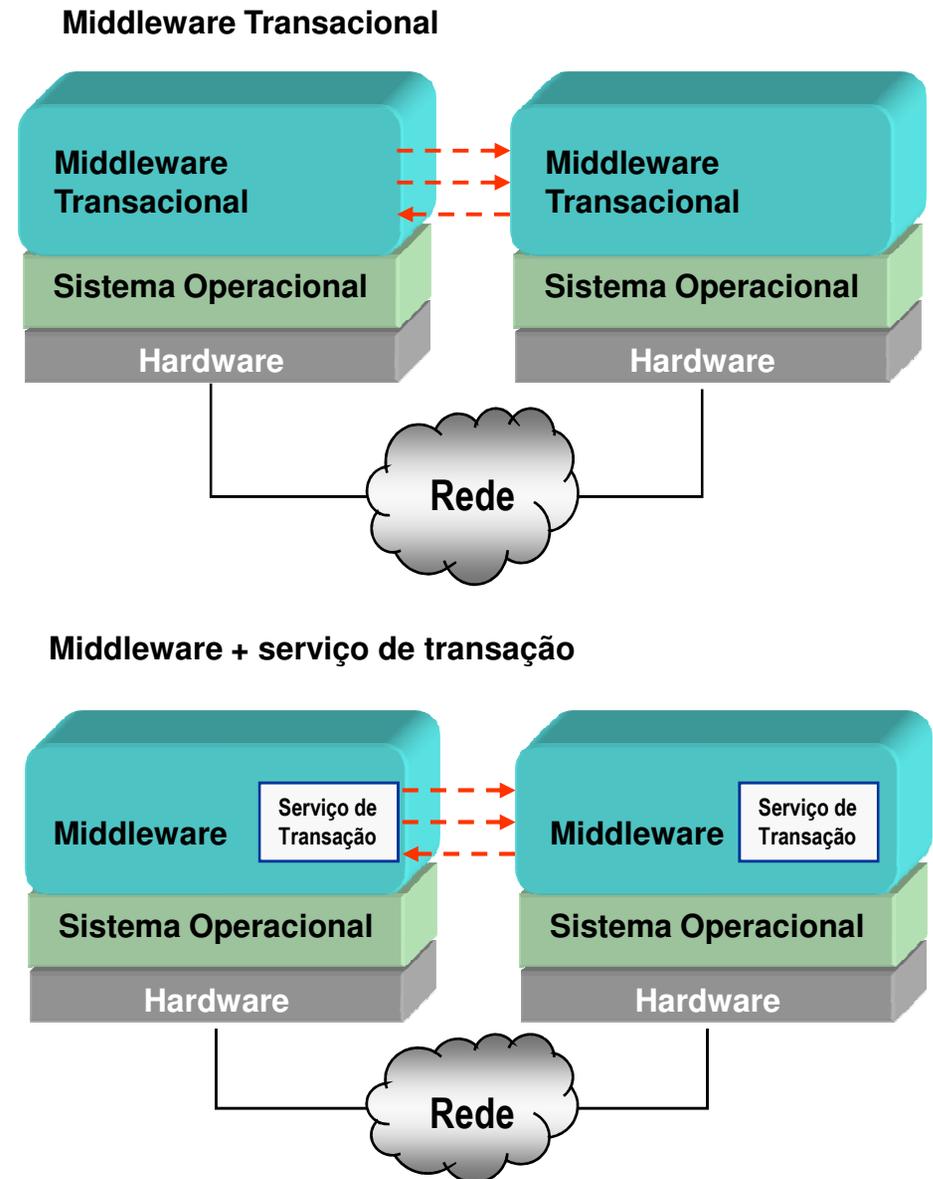


# Classificação 7 – Middleware Transacional

- Primitiva de interação: **transação**
  - chamada de procedimento remoto integrada + controle de transações
- protocolo comumente usado: protocolo *two-phase commit*
- comunicação síncrona suportada naturalmente
- comunicação **1-1**

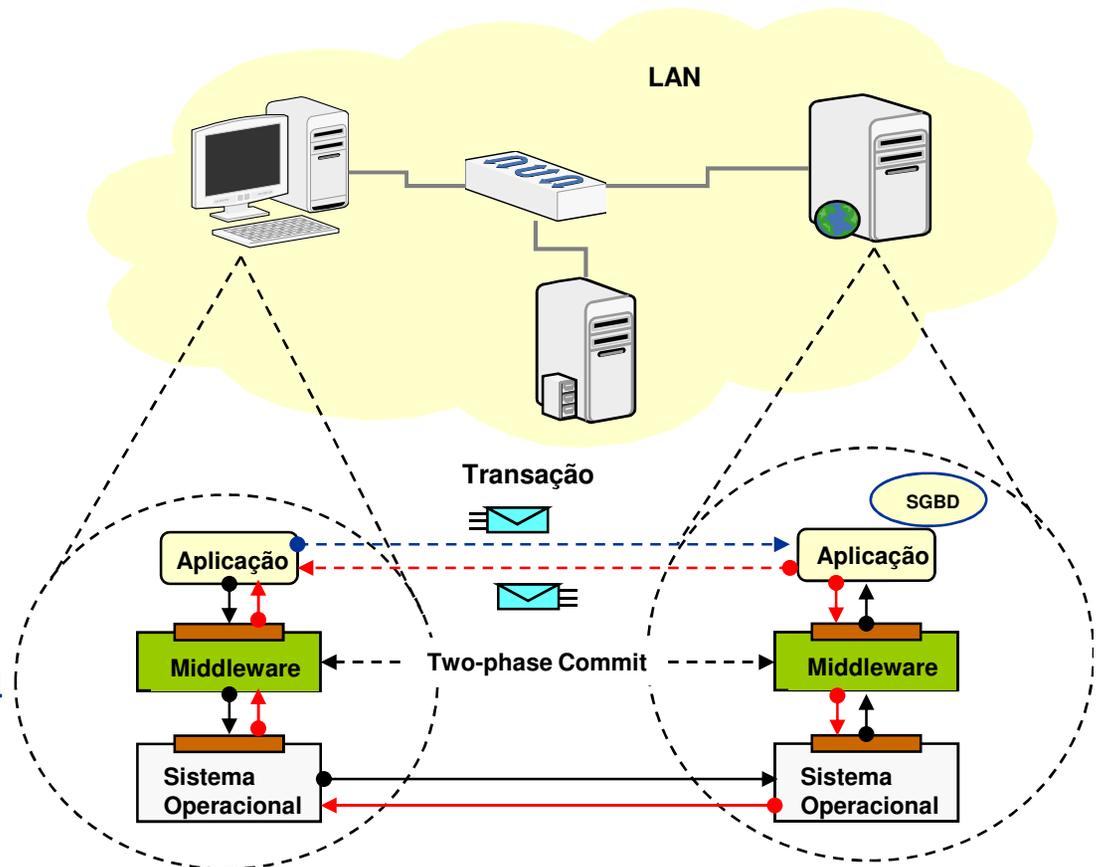
# Classificação 7 - **Middleware Transacional**

- fornecem suporte (coordenação e sincronização) à execução de transações de acesso à bases de dados
- esconde a complexidade da implementação de transações em redes
- tipicamente construído sobre outro middleware
- usado em sistemas de processamento de transações de alta escala (banco)



# Classificação 7 - Middleware Transacional

- o cliente pode fazer vários *requests* em uma única transação
- se há SGBDs nos servidores: a implementação do TPC fica a cargo dos SGBDs



# Classificação 7 - **Middleware Transacional**

- escalabilidade

- suporta balanceamento de carga e replicação (escalabilidade)

- heterogeneidade

- SO e LP

- dados (geralmente não suportada)

- ◆ estruturas de dados complexas usadas como parâmetros nas interações

# Classificação 7 - Middleware Transacional

## ■ problemas

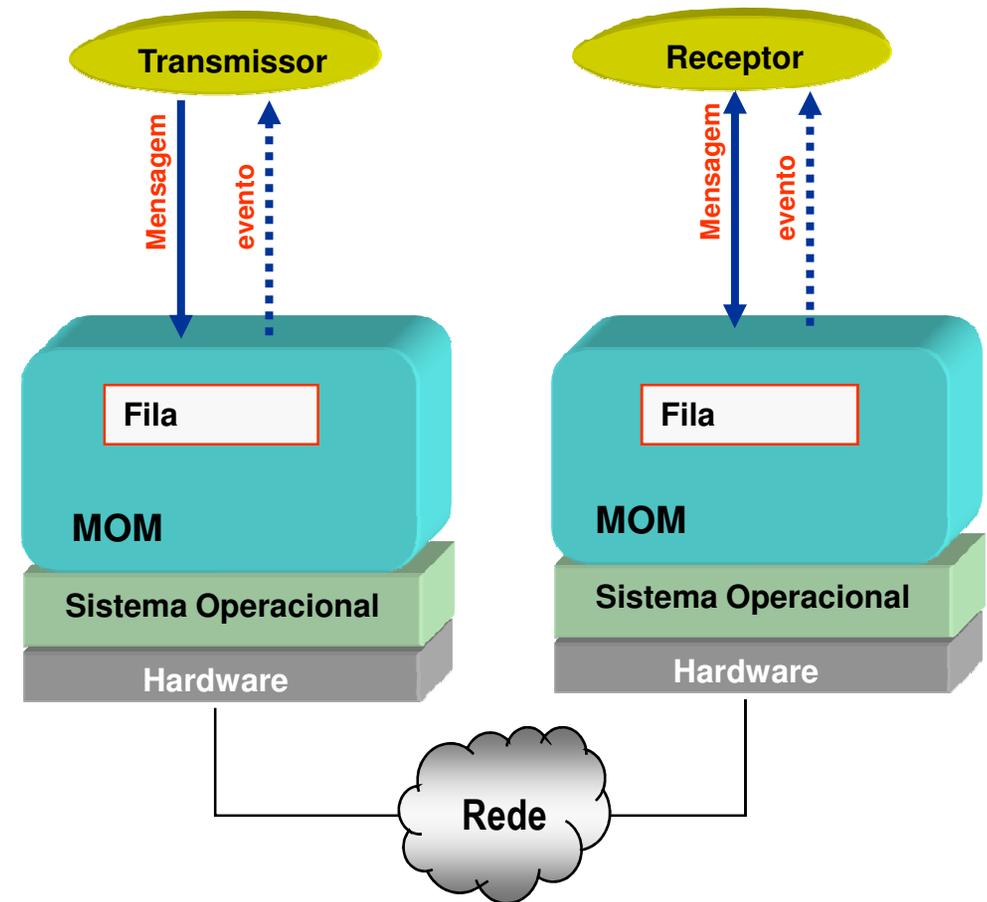
- overhead da implementação das propriedades ACID é alto
- (un)marshall geralmente implementado manualmente

## ■ produtos

- CICS (IBM)
- Encina (DCE-based)
- OMG Object Transaction Service (CORBA-based)
- Microsoft Transaction Server (DCOM-based)
- Java Transaction Service (Java RMI-based)
- BEA Tuxedo (BEA Tuxedo)
- JOTM (Java Open Transaction Manager)

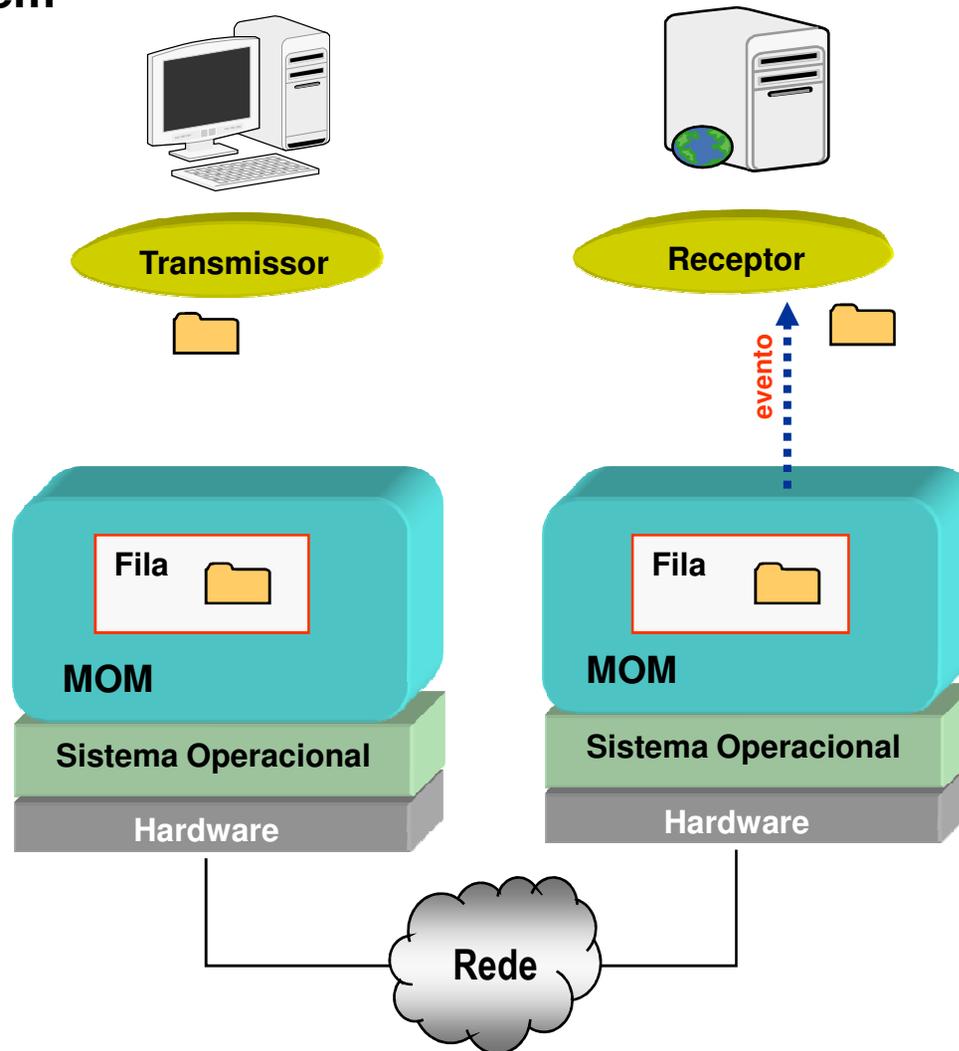
# Classificação 7 - **Middleware Orientado a Mensagem (MOM)**

- primitiva de interação: **passagem de mensagem**
- comunicação assíncrona/em grupo naturalmente implementadas
- uso de fila (temporárias, persistentes) de mensagens
- sub-tipos:
  - fila de mensagem
  - publish/subscribe
- e.g., MQSeries, Sun ONE Message Queue (free), JORAM (free)
- comunicação: **1-1, 1-N**

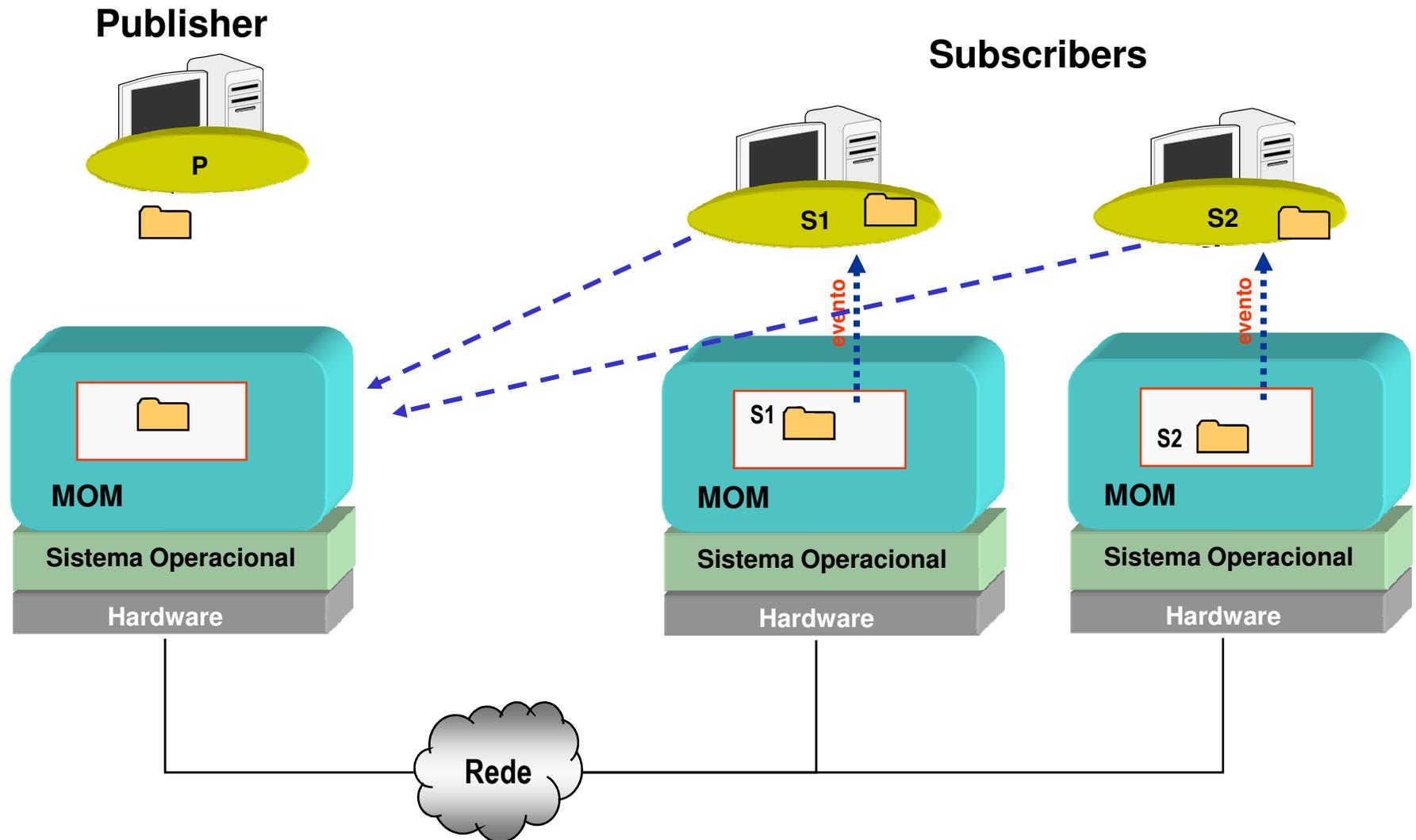


# Classificação 7 - Middleware Orientado a Mensagem (MOM)

## Fila de Mensagem



# Classificação 7 - **Middleware Orientado a Mensagem (MOM)**



# Classificação 7- Middleware Orientado a Mensagem (MOM)

## Microsoft Messaging Queue

