

Sistemas Distribuídos

Modelos de Sistemas

Altamira de Souza Queiroz

Introdução

- ❖ Sistemas em ambiente do mundo real são projetados:
 - ❖ Maior variedade possível de circunstâncias
 - ❖ Perante ameaças e dificuldades
- ❖ Os desafios sugerem que SD:
 - ❖ Compartilham importantes propriedades
 - ❖ Problemas comuns de projeto.



Introdução

- ❖ Modelos físicos

- ❖ Composição de Hardware
 - ❖ Redes de interconexão

- ❖ Modelos de arquitetura

- ❖ Cliente servidor e peer-to-peer

- ❖ Modelos fundamentais

- ❖ Modelos de interação
 - ❖ Modelos de falhas
 - ❖ Modelos de segurança



Modelos físicos

- ❖ Representação dos elementos de hardware de um SD.
- ❖ Modelo básico
- ❖ Gerações de SD
 - ❖ Primitivos: surgimento das redes locais e suportam uma pequena variedade de serviços.
 - ❖ Adaptados para a Internet: nós interconectados por uma rede de redes.
 - ❖ Contemporâneos: surgimento da computação móvel, ubíqua e computação em nuvem.



Modelo de arquitetura

- ❖ Estrutura em termos de componentes especificados separadamente e suas inter-relações.
- ❖ Objetivo: garantir a demanda atual e futura imposta
 - ❖ Confiável, gerenciável, adaptável e rentável.
- ❖ Elementos arquitetônicos
- ❖ Padrões arquitetônicos
- ❖ Plataformas de middleware



Elementos arquitetônicos

❖ Entidades de Comunicação

- ❖ Ponto de vista do sistema: Processos, nós e as threads
- ❖ Ponto de vista do problema: Objetos , componentes e serviços Web

❖ Paradigmas de comunicação

- ❖ Comunicação entre processos: primitivas de passagem de mensagens
- ❖ Invocação remota: paradigmas mais comuns
- ❖ Comunicação indireta: intermediário.

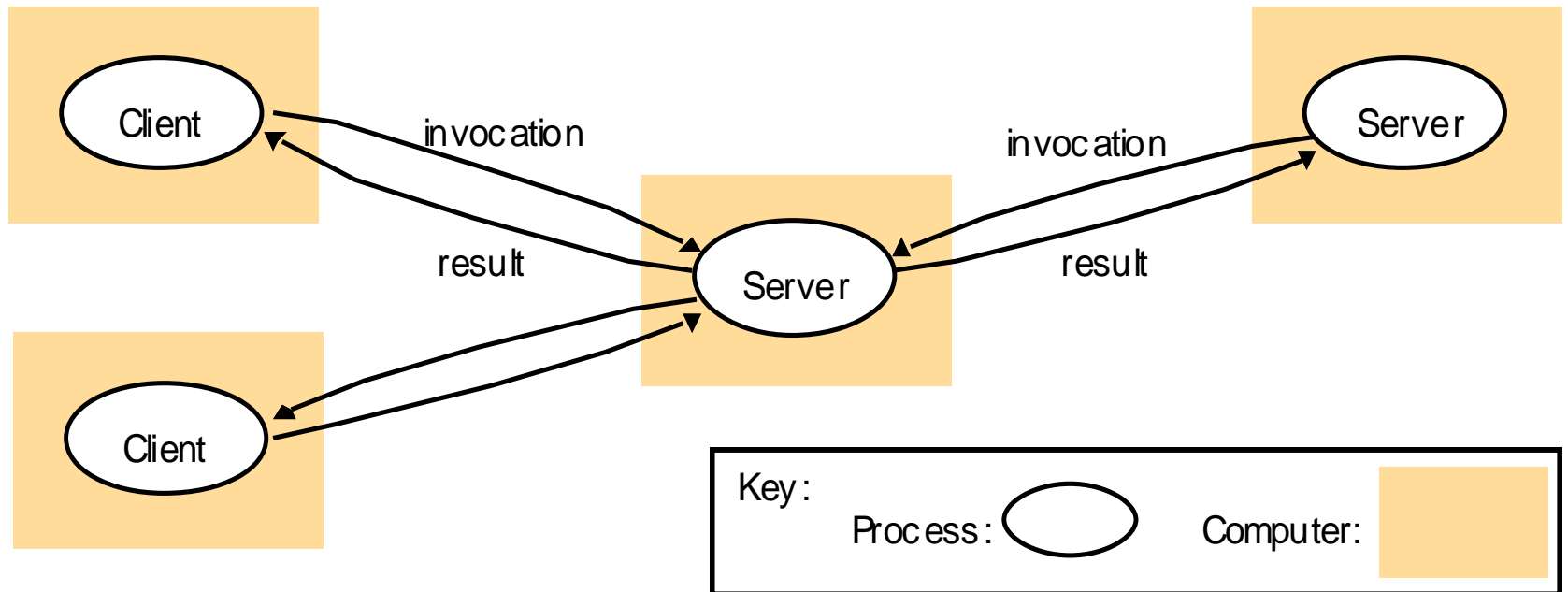


Elementos arquitetônicos

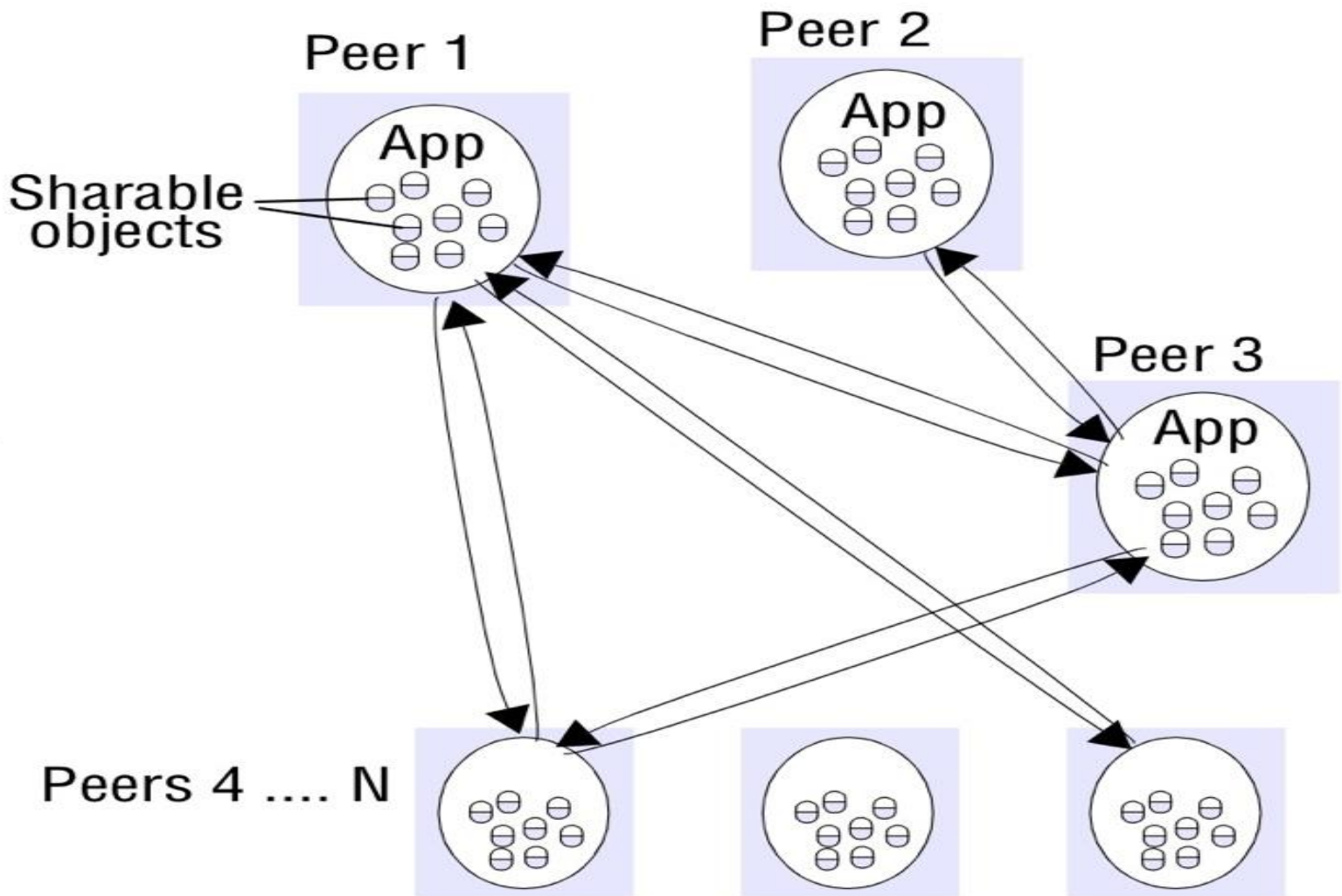
<i>Communicating entities (what is communicating)</i>		<i>Communication paradigms (how they communicate)</i>		
<i>System-oriented entities</i>	<i>Problem- oriented entities</i>	<i>Interprocess communication</i>	<i>Remote invocation</i>	<i>Indirect communication</i>
Nodes	Objects	Message passing	Request- reply	Group communication
Processes	Components	Sockets	RPC	Publish-subscribe
	Web services	Multicast	RMI	Message queues
				Tuple spaces
				DSM



Cliente-Servidor



Peer-to-Peer



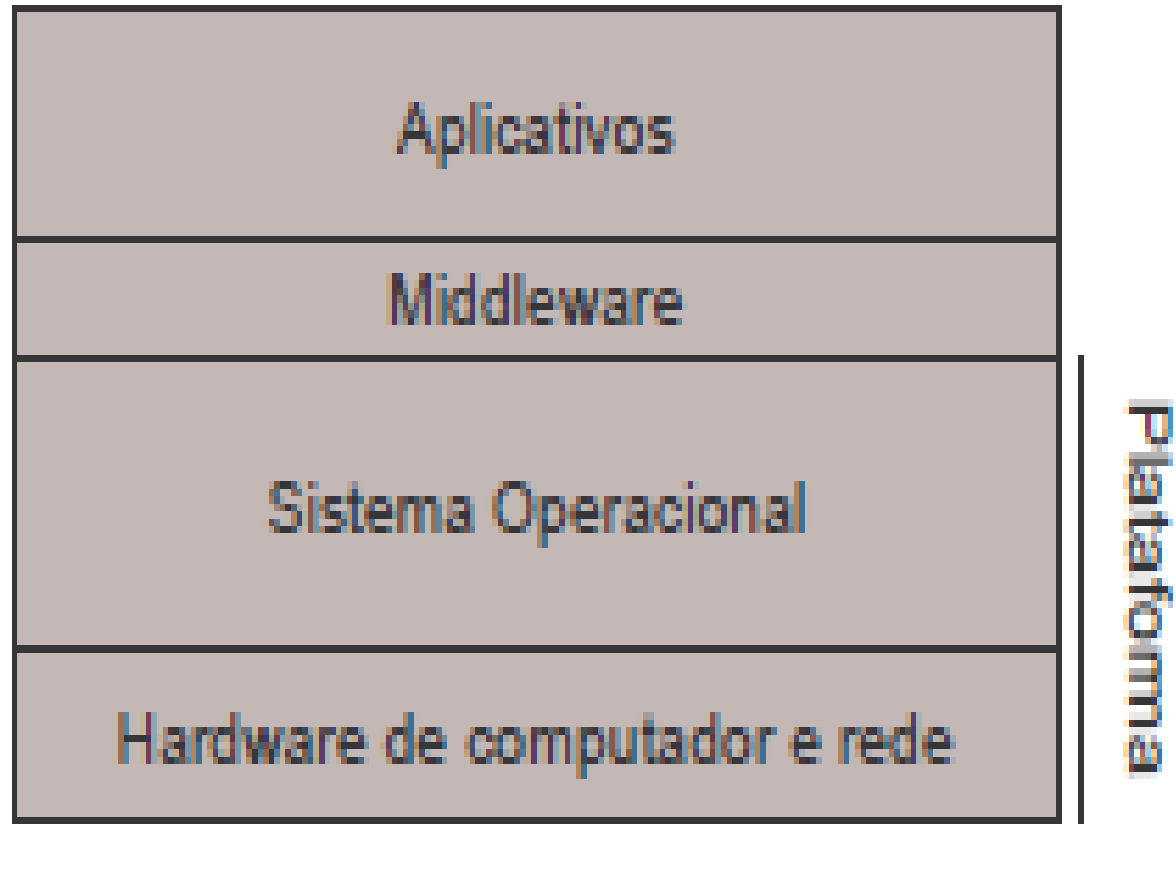
Padrão arquitetônico

❖ Arquitetura de SW

- ❖ antes: estruturação de SW em único computador
- ❖ Recente: serviços oferecidos e solicitados entre processos localizados em computadores diferentes
- ❖ Serviço distribuído pode ser fornecido por um ou mais servidores interagindo com clientes



Camadas de SW



Camadas de SW

- ❖ **Plataforma:** camadas de HW e SW de mais baixo nível
 - ❖ Interface de programação para as camadas superiores para facilitar a comunicação e coordenação de processos.
- ❖ **Middleware:** camada com objetivo de mascarar a heterogeneidade
 - ❖ Fornecer um modelo de programação conveniente



Middleware

- ❖ Simplifica as atividades de comunicação através de abstrações
 - ❖ Invocação de métodos remotos
 - ❖ Comunicação entre um grupo de processos
 - ❖ Notificação de eventos
 - ❖ Particionamento
 - ❖ Posicionamento e recuperação de objetos compartilhados
 - ❖ Replicação de objetos compartilhados
 - ❖ Transmissão de dados multimídia em tempo real



Middleware

- ❖ Primeiros: Sun RPC e ISIS
- ❖ Atualmente há vários produtos e padrões orientados a objetos
 - ❖ CORBA
 - ❖ RMI Java
 - ❖ Serviços web
 - ❖ O RM-ODP do ISSO/ITU-T
- ❖ Fornece serviços aos programas aplicativos



Middleware - Limitações

- ❖ Muitos aplicativos baseiam-se completamente nos serviços de middleware
 - ❖ Ex. BD de nomes e end. em que o MW forneça apenas invocação a método remoto em modelo cliente-servidor
- ❖ Alguns aspectos exigem suporte em nível de aplicativos.
 - ❖ Ex. transferência de grandes mens. de correio eletrônico



Middleware - Limitações

- ❖ A primeira vista é simples aplicação de TCP
- ❖ Problema é a transferência de arquivo muito grande em uma rede não confiável
 - ❖ TCP capacidade de detecção e correção de erros, mas não se recupera de problemas mais sérios de rede
 - ❖ O middleware usa uma versão mais simples de comunicação para melhoria de desempenho



Middleware - Limitações

- ❖ Verificações, mecanismos de correção de erro e medidas de segurança em muitos níveis
- ❖ O sistema de comunicação garantirá parte da correção exigida.
- ❖ desnecessária devido a complexidade e redundância de operações

