Sistemas Distribuídos

Aleardo Manacero Jr.

Sistema de arquivos distribuídos

- Na ótica de sistemas distribuídos o tratamento de sistemas de arquivos deve:
  - Considerar a possibilidade de distribuição das informações, e
  - Garantir transparência no acesso às informações

Sistema de arquivos distribuídos

- Isso implica em que as funcionalidades existentes em sistemas convencionais devem ser modificadas para garantir essas características

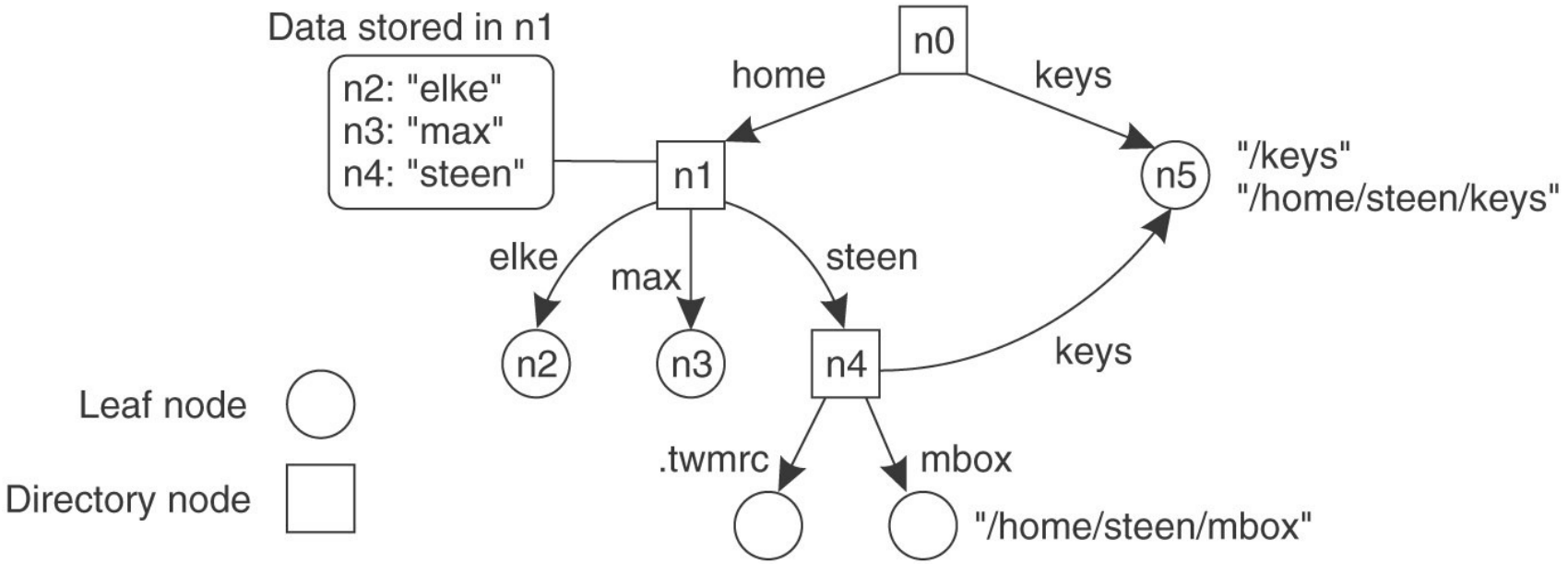
Sistema de arquivos distribuídos

- As principais funcionalidades são:
  - Nomeação
  - Consistência

Nomeação

- É a tarefa de associar nomes a entidades presentes no sistema
- As entidades podem ser arquivos, diretórios, máquinas, etc.
- Considerando a distribuição do sistema, cabe ao serviço de nomeação a localização dessas entidades

Grafo de nomes



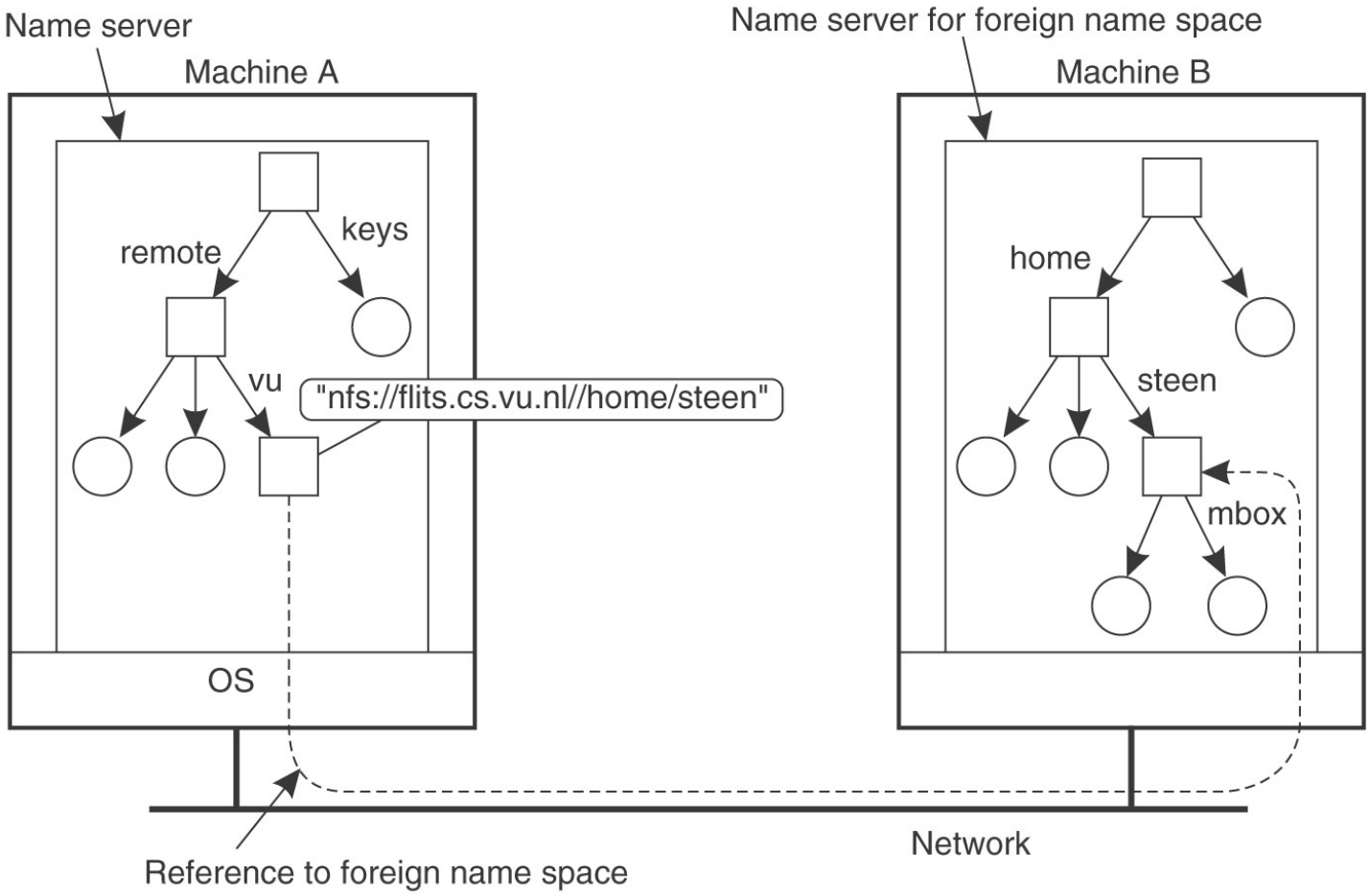
Nomeação e ligação

- A definição de nomes carrega, também, a possibilidade do uso de apelidos e, conseqüentemente, de links entre arquivos
- Isso, por sua vez, habilita a criação de pontos de montagem para o sistema de arquivos

Montagem de sistemas de arquivos

- Pode ser gerada criando-se:
  - Um ponto para montagem, num diretório local
  - Um ponto montado, num diretório remoto, devidamente identificado
  - Um protocolo de acesso ao diretório remoto
- Esse é o mecanismo do NFS da Sun

Montagem de sistemas de arquivos

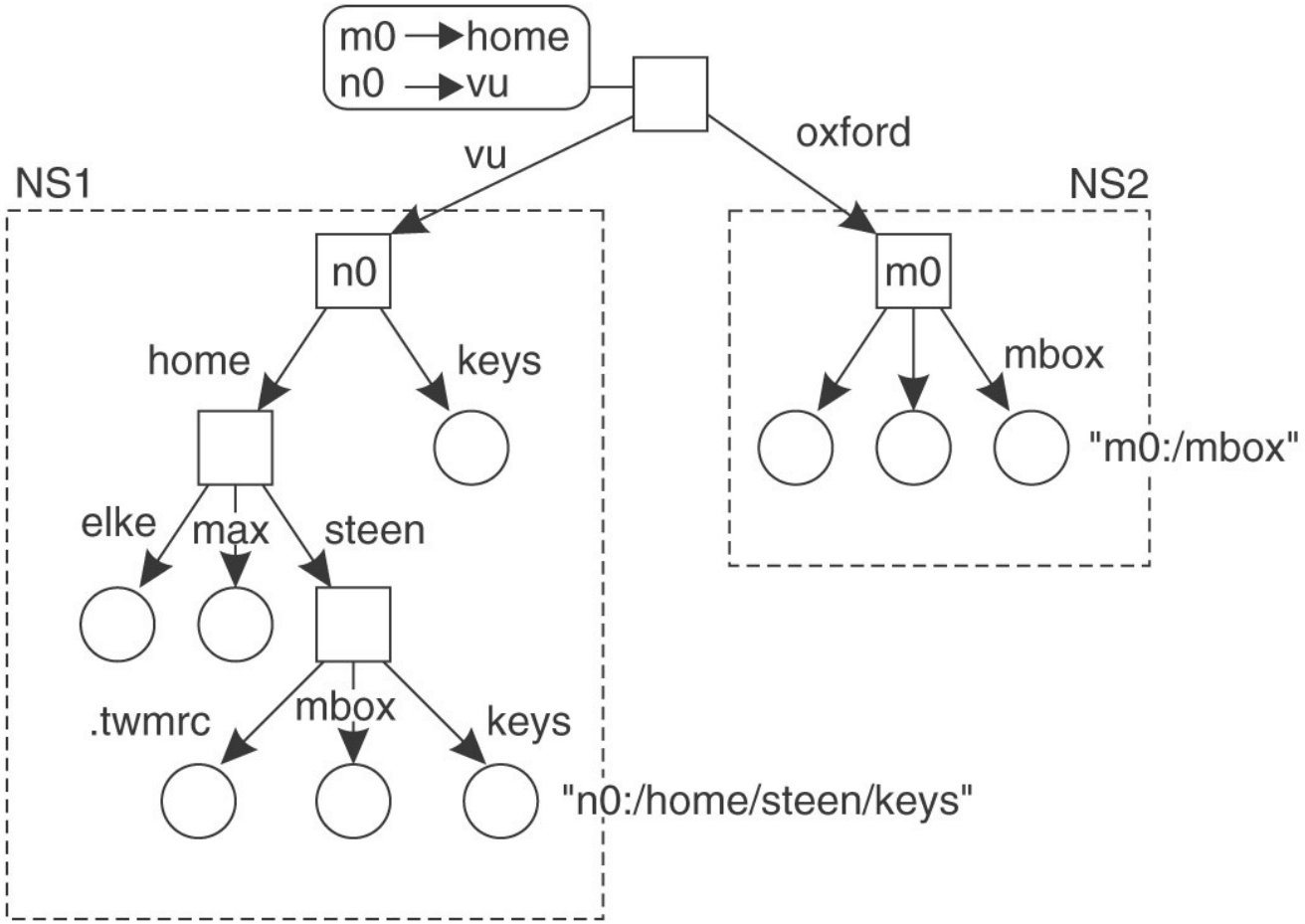


Montagem de sistema de arquivos

- Outra possibilidade é transformar os sistemas locais em sub-sistemas de um novo sistema

- Isso é feito no Serviço de Nomes Global (GNS) criado pela DEC

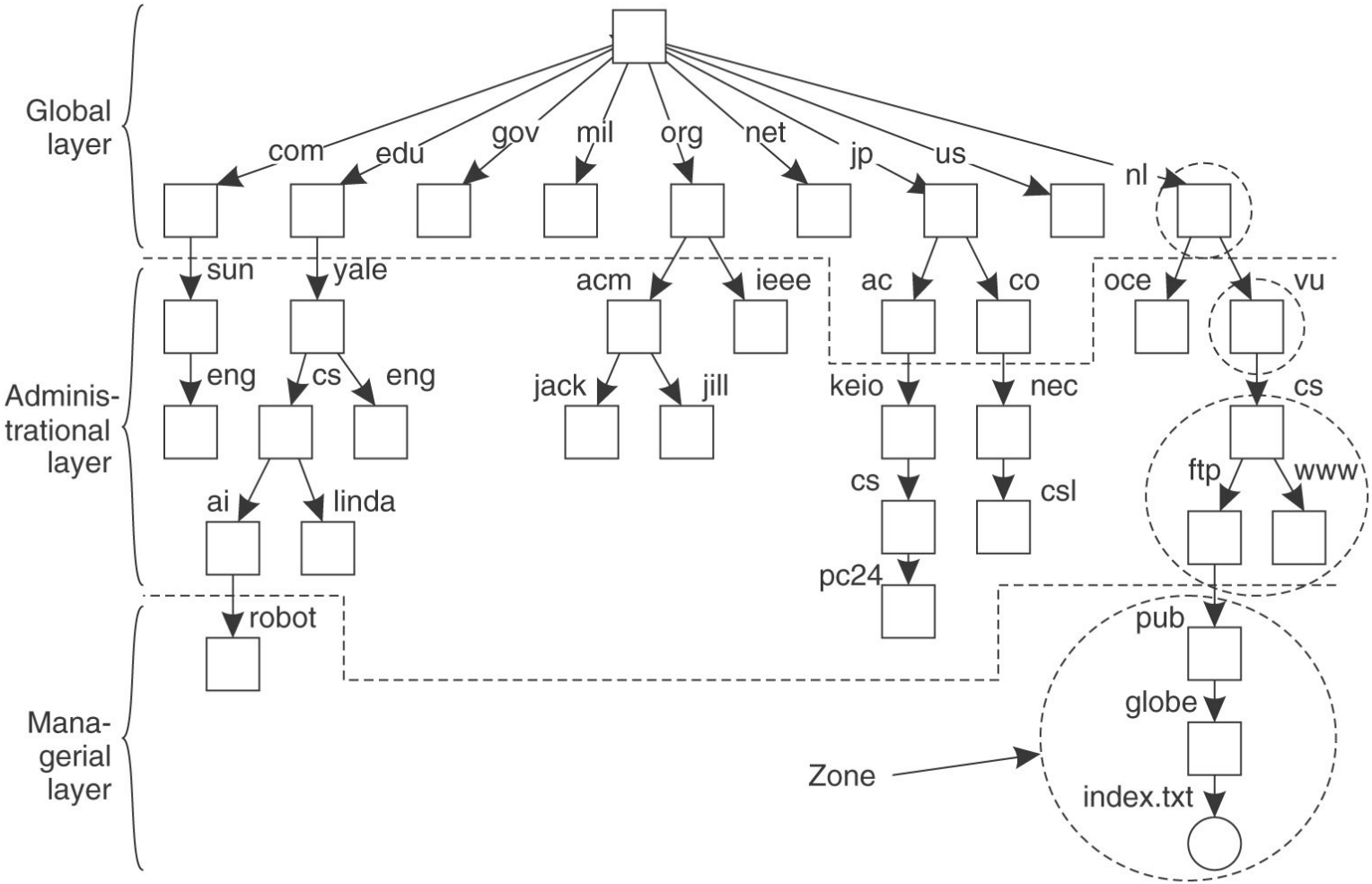
Montagem de sistema de arquivos



Espaço de nomes

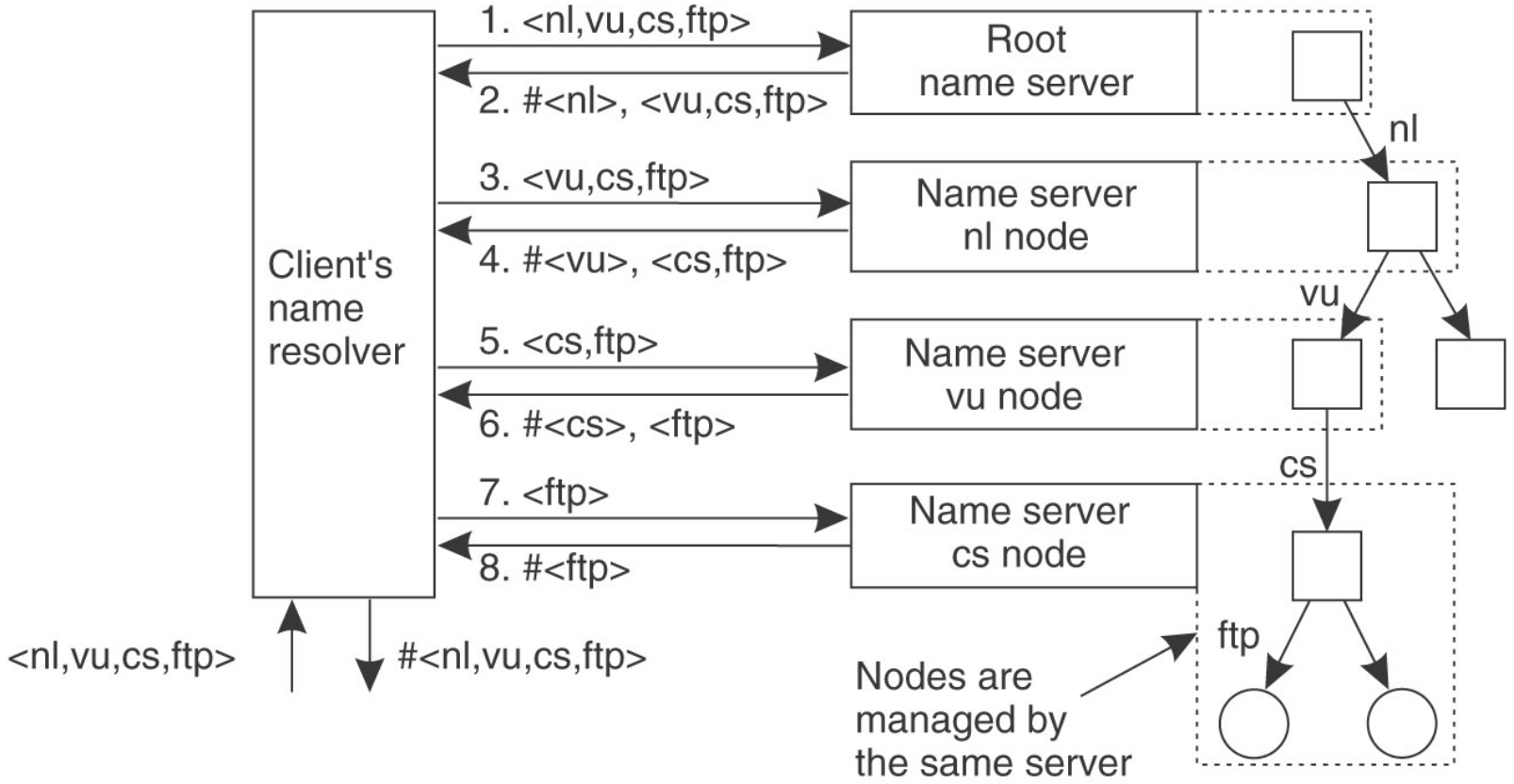
- A definição de espaços de nomes é essencial para as operações de localização e mapeamento de arquivos
- Os espaços de nomes podem ser classificados segundo sua distribuição geográfica (mundo, organizações e departamentos)

Espaço de nomes



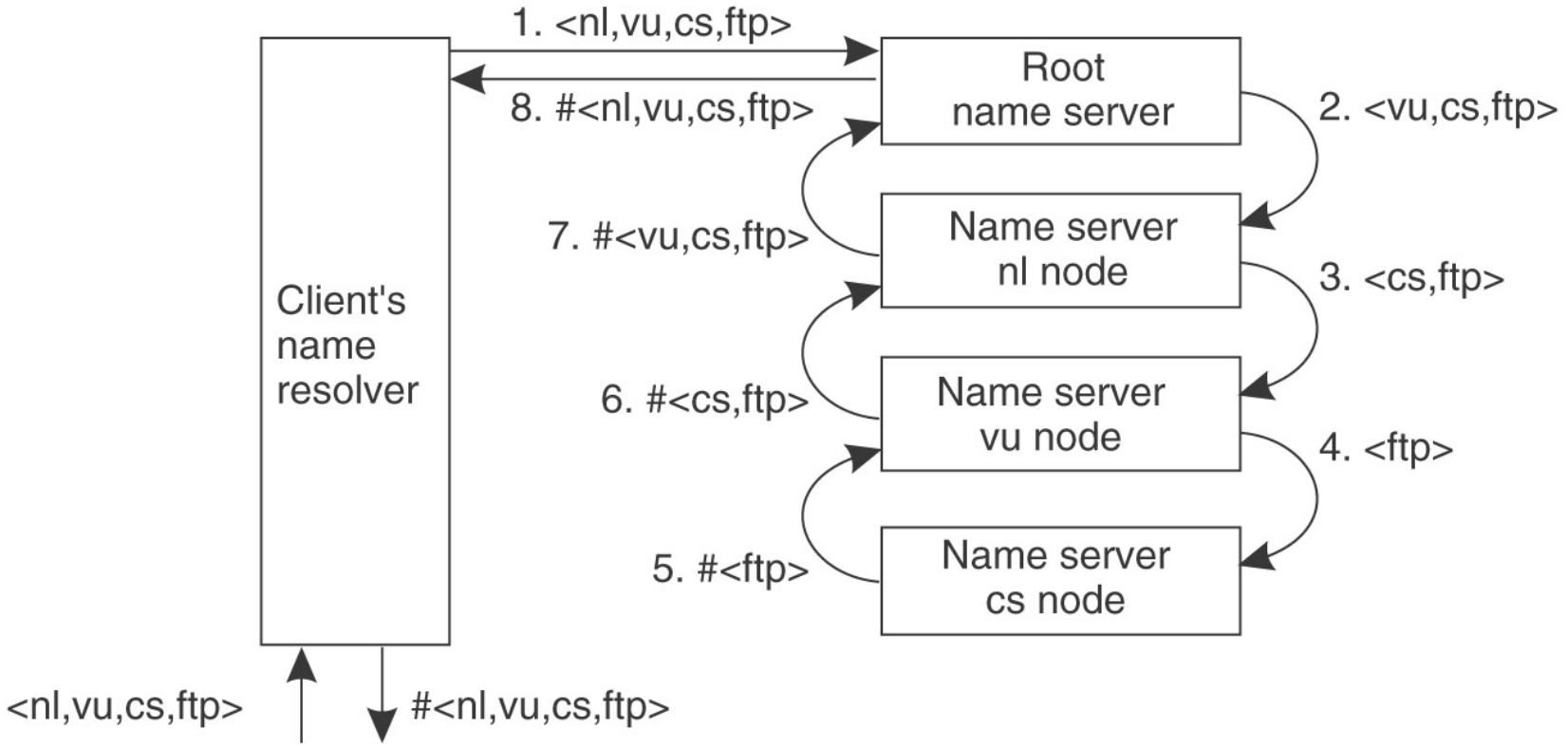
Resolução de nomes

- Modelo iterativo



Resolução de nomes

- Modelo recursivo



Resolução de nomes

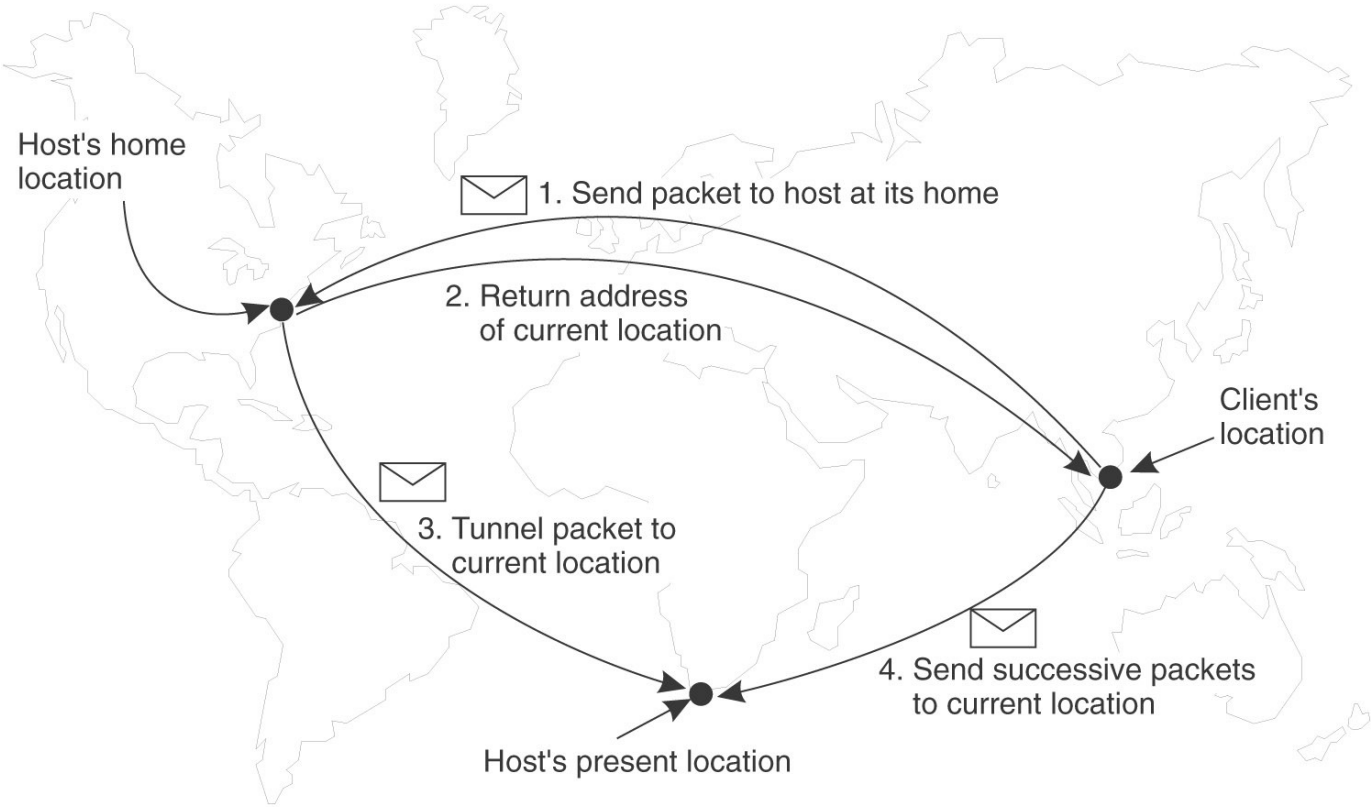
- O serviço de resolução de nomes é definido em protocolos como o DNS e o X.500 (ISO)

Localização de entidades

- Os serviços de nomeação e resolução de nomes são aplicáveis em sistemas estáticos
- No caso de entidades móveis a solução passa pela criação de serviços de localização

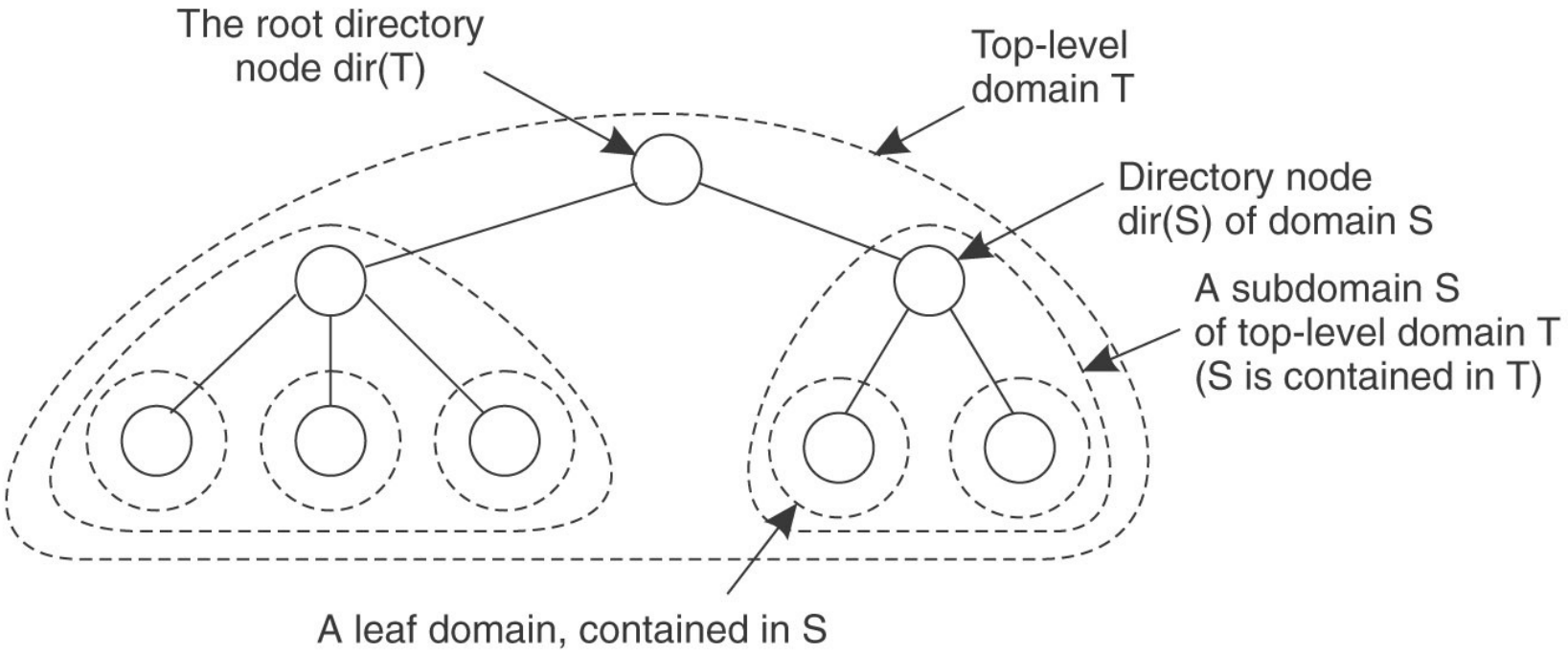
Localização de entidades

- Modelo local de IP móvel



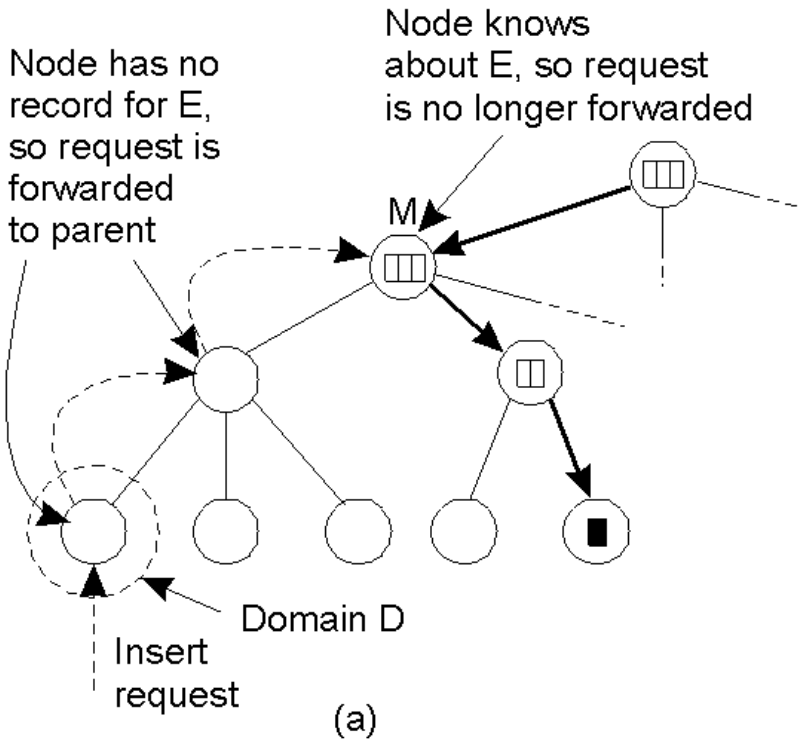
Localização de entidades

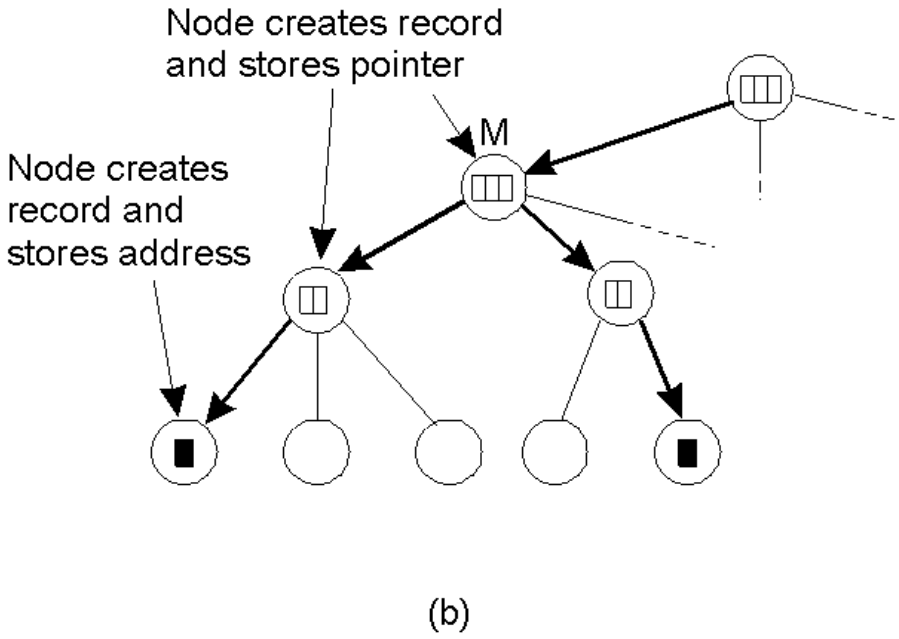
- Modelo hierárquico



Localização de arquivos

- Inserção e cadeia de ponteiros

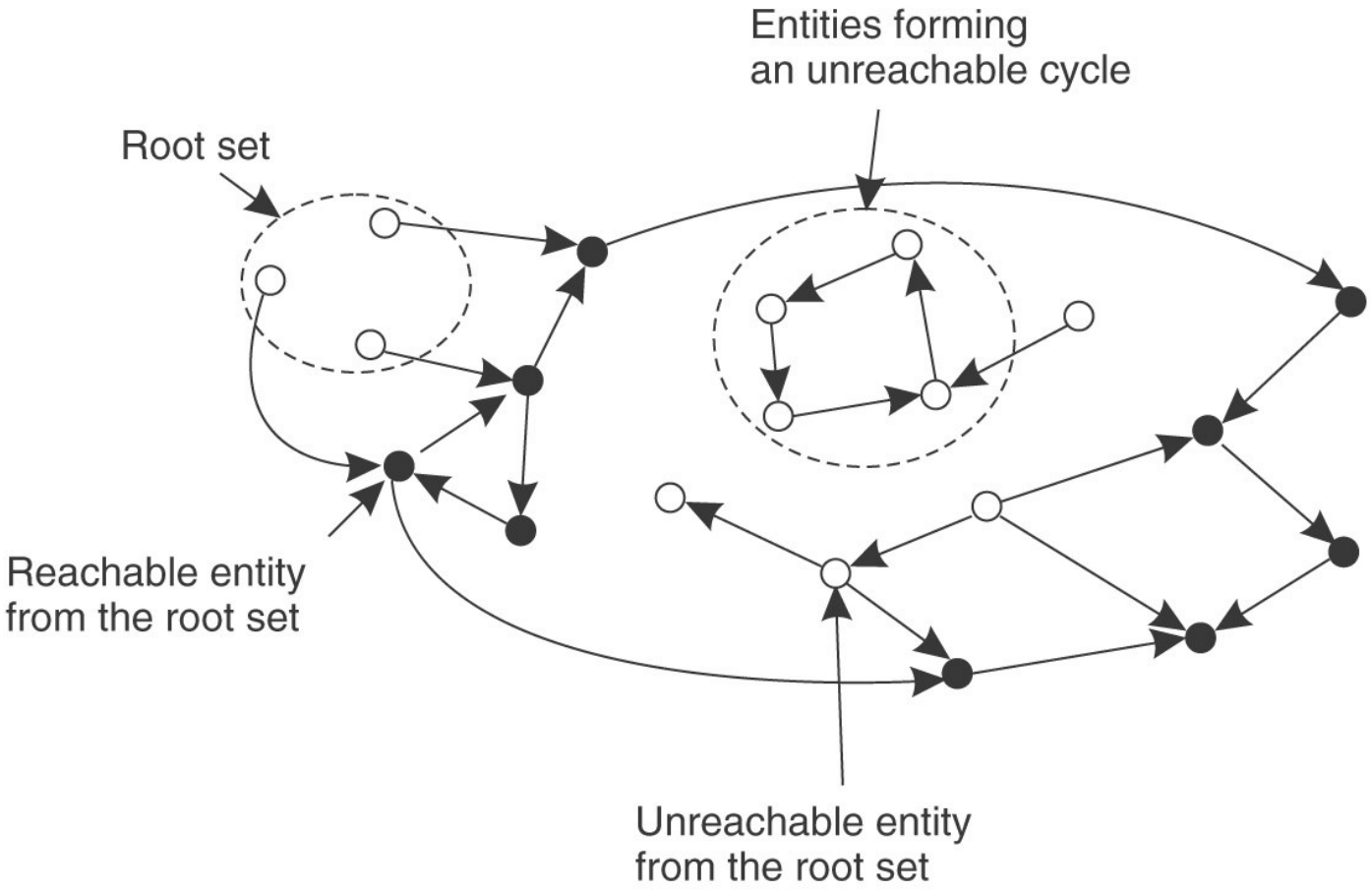




Entidades não referenciadas

- A multiplicação de referências para entidades num SD cria problemas no momento da remoção de entidades não referenciadas
- Isso cria a demanda por coletores de lixo distribuídos, baseados em modelos de alcançabilidade

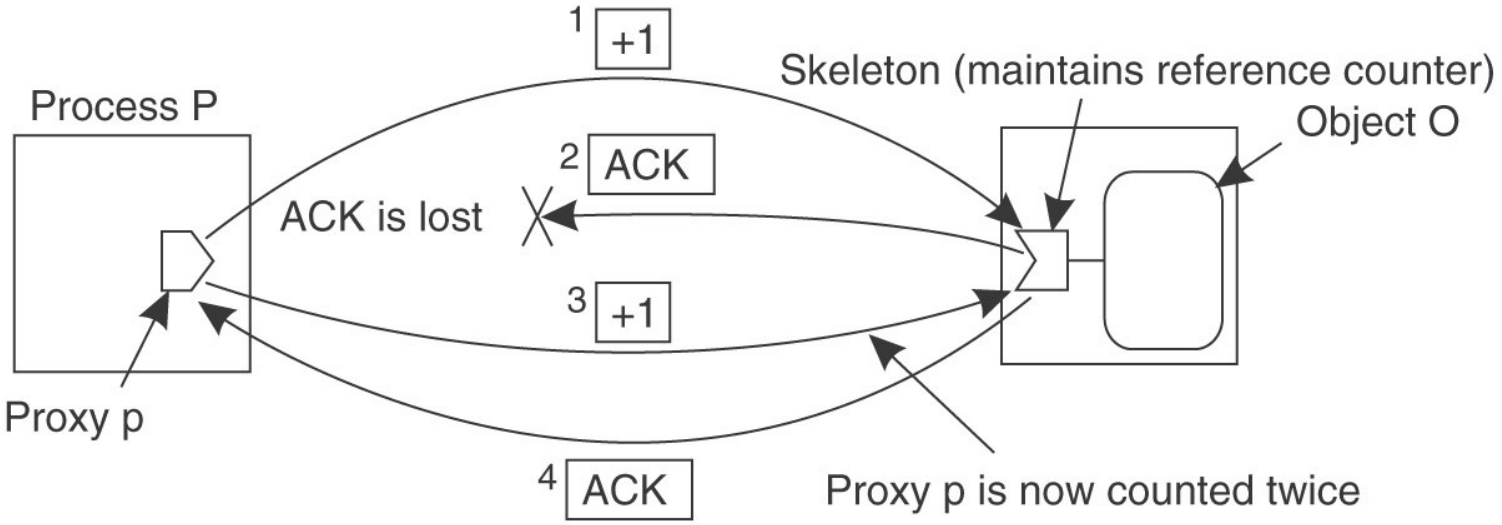
Modelo de alcançabilidade



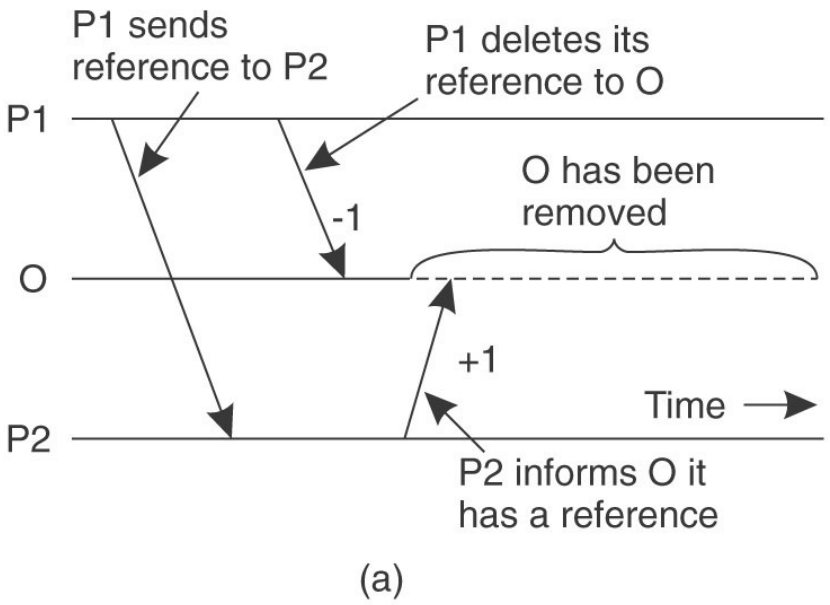
Entidades não referenciadas

- Outra forma de ação inclui modelos de contagem de referências, que podem ser:
  - Simples
  - Ponderado
  - Lista de referências

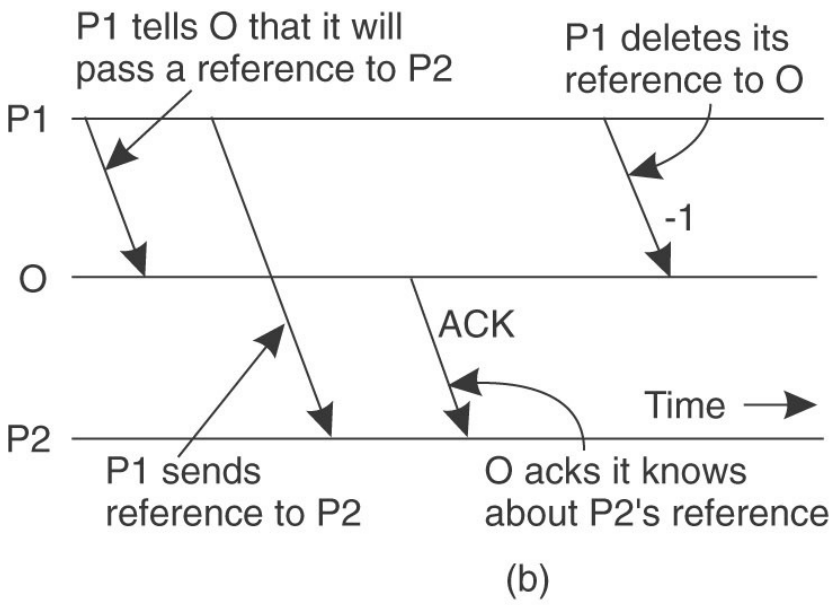
Modelos de contagem: Simples



Modelos de contagem: Simples

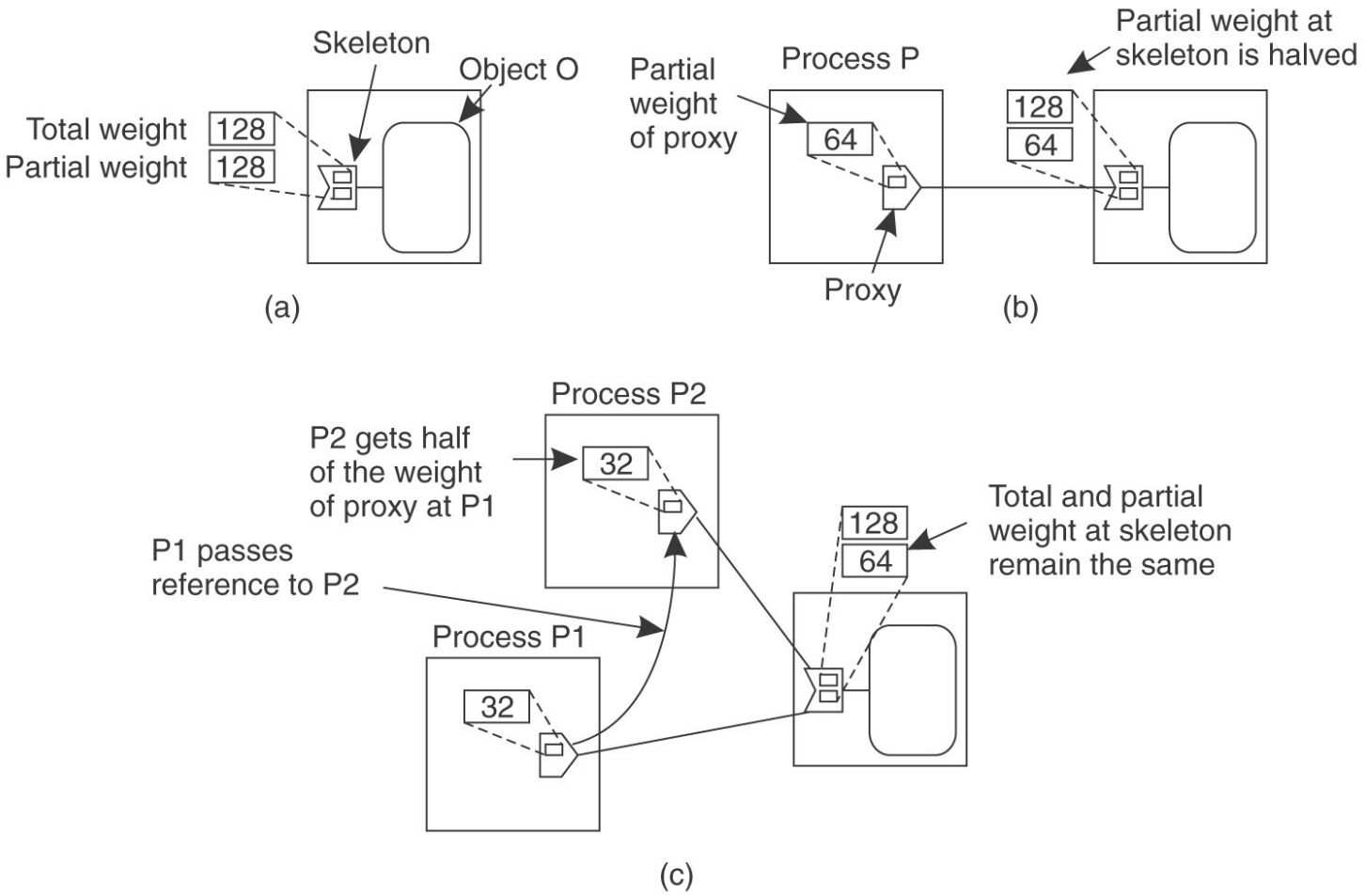


(a)



(b)

Modelos de contagem: Ponderada



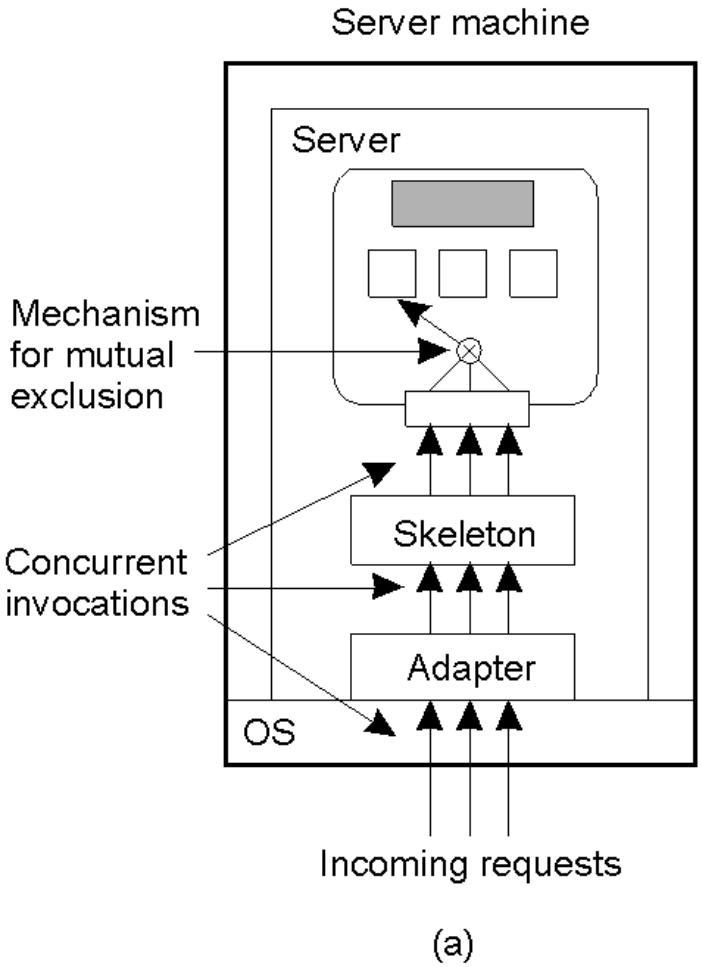
Consistência em SD

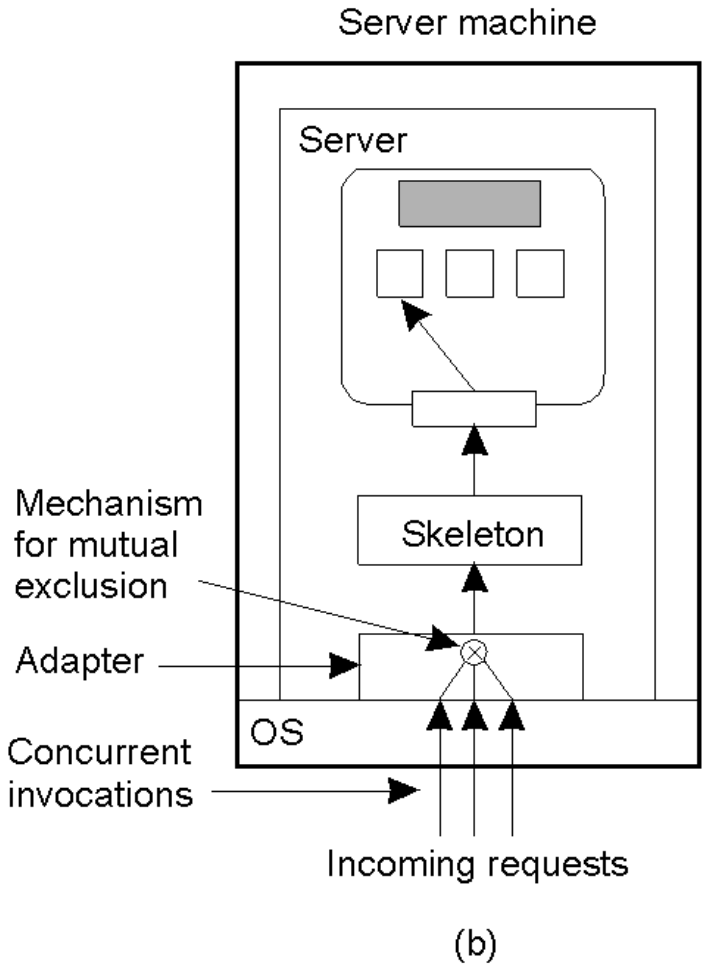
- Em SD existe também a multiplicação de cópias dos objetos (dados) nele armazenados
- Isso ocorre por necessidade de desempenho (paralelização de tarefas) ou de confiabilidade (disponibilidade do sistema)
- É o que chamamos replicação de dados

Replicação de objetos

- A existência de objetos replicados demanda a criação de mecanismos para tratamento de concorrência no seu acesso
- Isso pode ser feito pelo próprio objeto ou por agentes externos a ele

Replicação de objetos



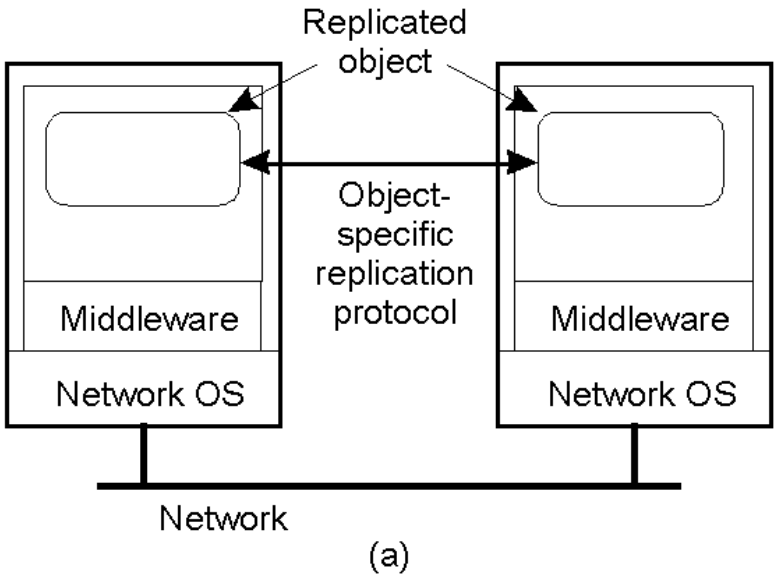


Replicação de objetos

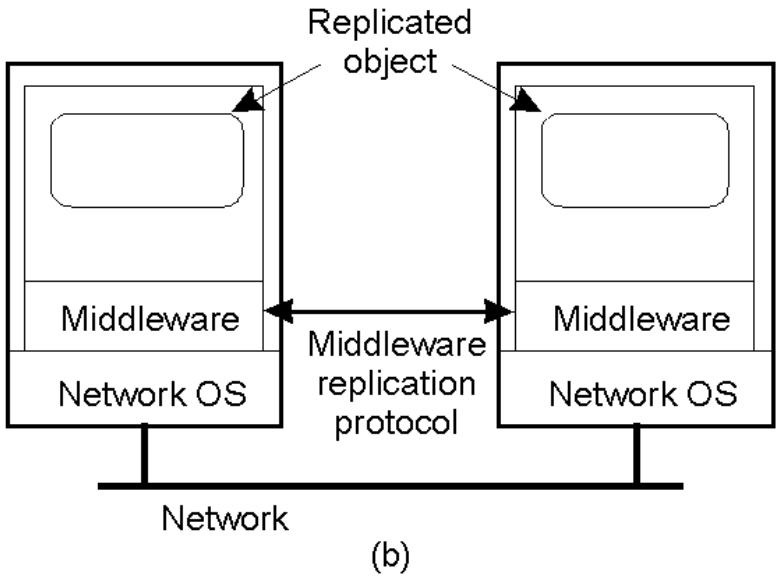
- Além disso, o tratamento dado aos objetos replicados pode ocorrer em diferentes níveis de sistema, como:
  - Nível da própria aplicação
  - Nível de middleware

Replicação de objetos

- Nível da aplicação



- Nível do Middleware



Modelos de replicação

- A consistência da replicação pode ser implementada através de dois modelos
- Um deles é centrado nos dados replicados
- O outro modelo é centrado nos clientes desses dados

Consistência centrada em dados

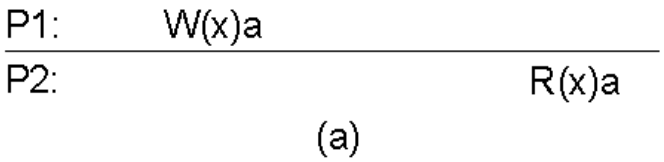
- A consistência de um objeto é dependente de operações de leitura e escrita sobre suas múltiplas cópias
- Um modelo de consistência define regras a serem seguidas nessas operações de modo a que o sistema possa garantir a integridade dos dados

Consistência centrada em dados

- No caso de consistência centrada em dados as regras definem como os dados são vistos pelos processos
- Os níveis de consistência variam entre estrito, seqüencial, casual, FIFO, fraca, entrada e liberação

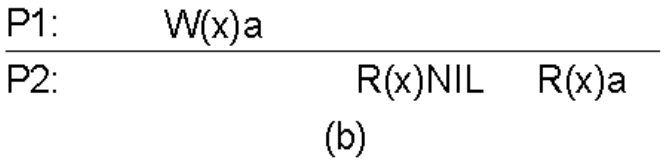
Consistência estrita

- Atendida



(a)

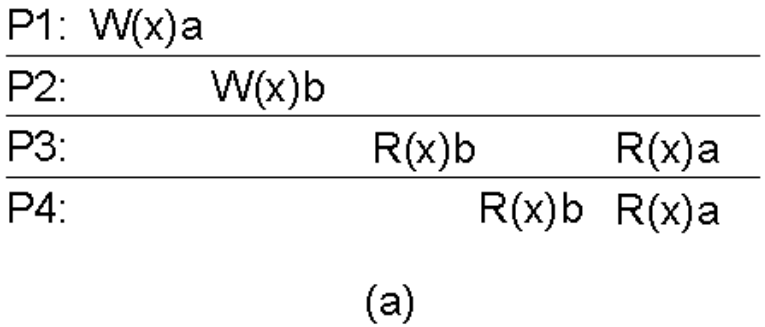
- Não atendida



(b)

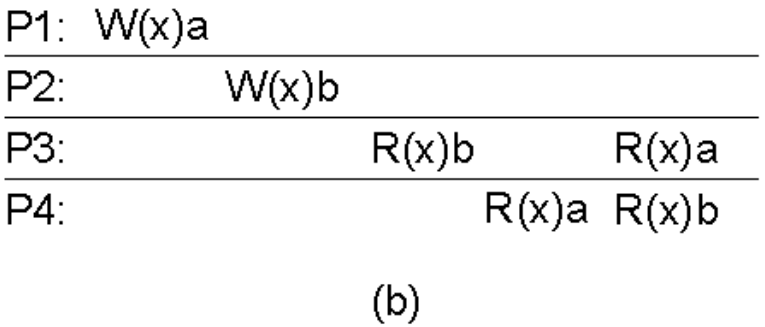
Consistência seqüencial

- Atendida



(a)

- Não atendida



(b)

Consistência FIFO

P1: W(x)a						
P2:	R(x)a	W(x)b	W(x)c			
P3:				R(x)b	R(x)a	R(x)c
P4:				R(x)a	R(x)b	R(x)c

Consistência fraca

- Válida

P1: W(x)a	W(x)b	S			
P2:			R(x)a	R(x)b	S
P3:			R(x)b	R(x)a	S

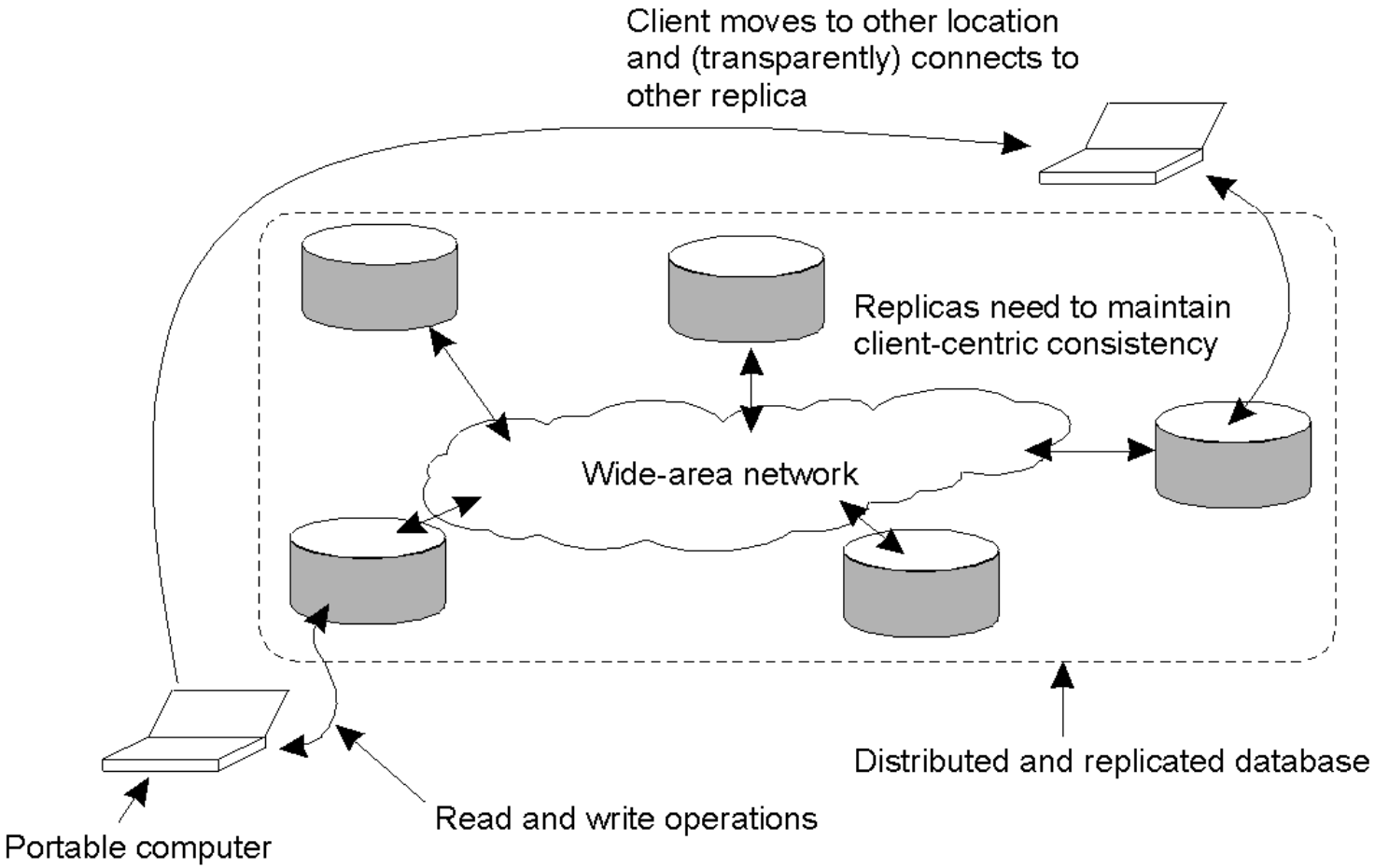
- Inválida

P1: W(x)a	W(x)b	S		
P2:			S	R(x)a

Consistência centrada em clientes

- Aqui o modelo de consistência considera que o que importa é a visão que clientes têm dos dados e que a maioria das operações é de leitura
- Isso leva a um modelo simples e fraco, chamado de consistência eventual

Consistência centrada em clientes



Consistência monotônica

- Leitura monotônica – garante que após ler um objeto um dado processo nunca lerá uma versão anterior do mesmo
- Escrita monotônica – garante que toda escrita sobre um objeto será completada em todas as cópias antes que o mesmo processo faça outra operação de escrita sobre ele

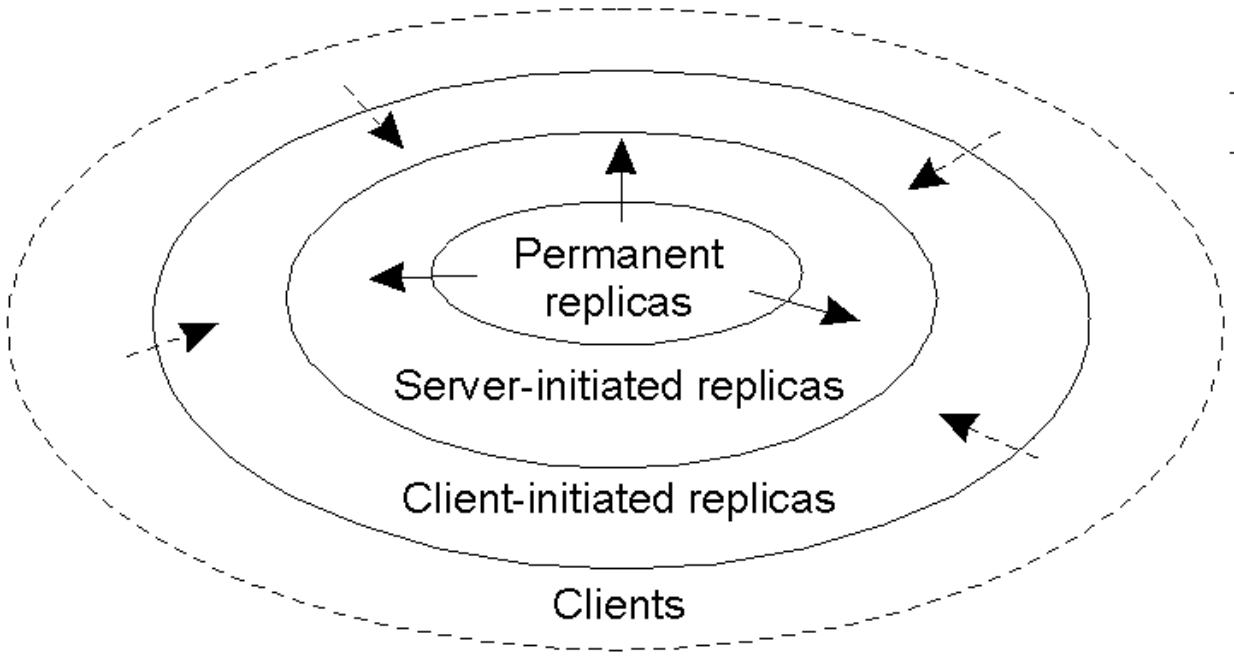
Consistência monotônica

- Leia-seus-escritos – que garante que a leitura de um objeto escrito pelo mesmo processo trará sempre seu valor atualizado
- Escrita-após-leitura – em que se garante que a operação de escrita de objeto recém lido ocorrerá sobre o mesmo valor ou sua atualização

Modelos de distribuição

- As réplicas de objetos podem ser criadas (e distribuídas) a partir de três níveis de iniciativa
  - Réplicas permanentes
  - Réplicas criadas pelo servidor
  - Réplicas criadas pela aplicação (cliente)

Modelos de distribuição



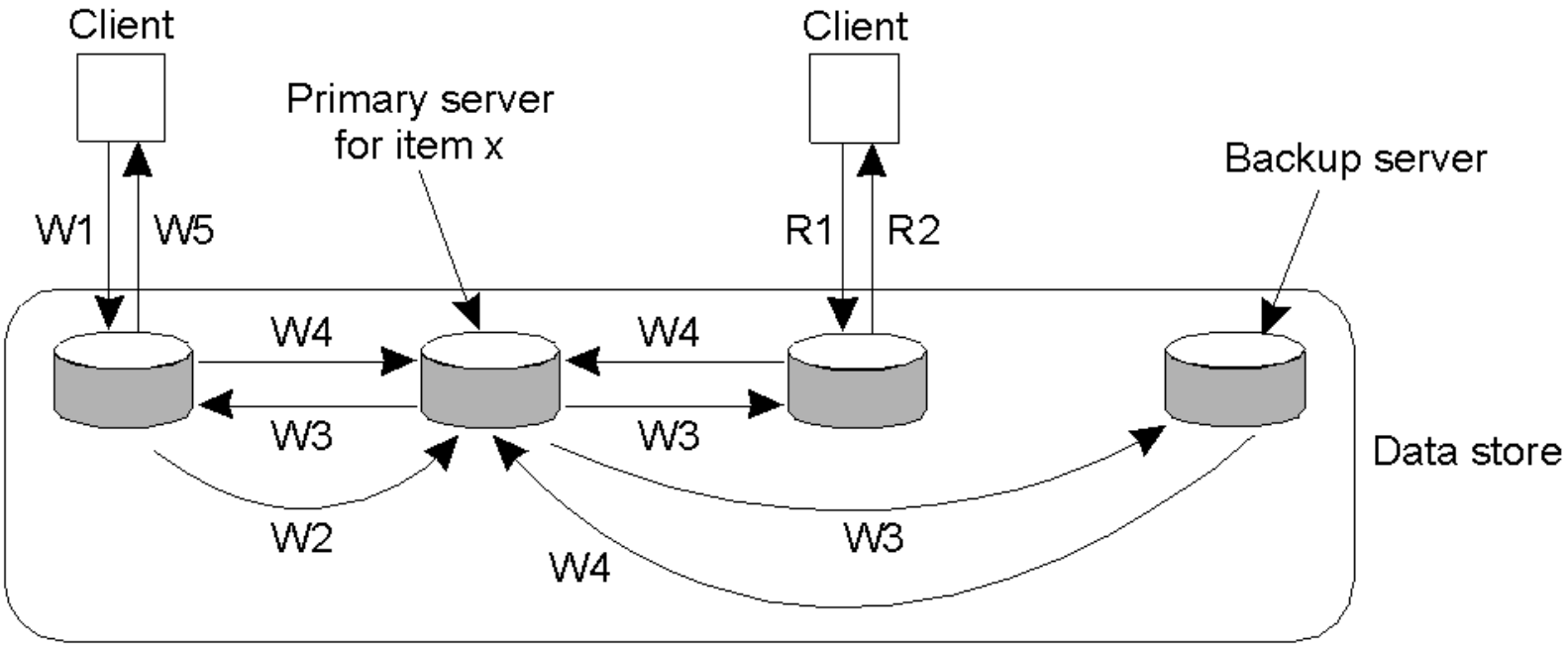
Server-initiated replication

Client-initiated replication

Modelos de atualização

- Finalmente, temos que definir modelos para a forma que os dados replicados serão atualizados no SD
- Nesse caso temos os modelos push (“empurrar as alterações”) e pull (“puxar as alterações”)

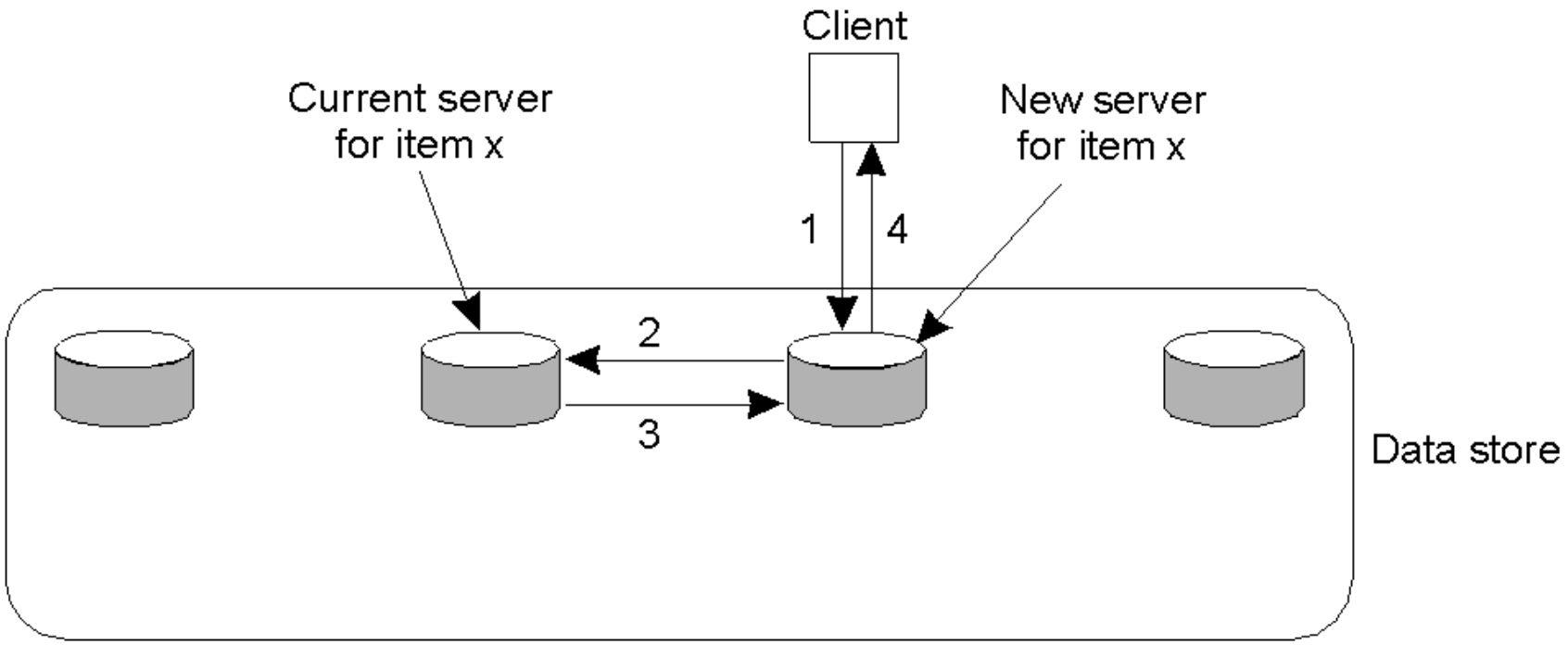
“Pushing” atualizações



W1. Write request
W2. Forward request to primary
W3. Tell backups to update
W4. Acknowledge update
W5. Acknowledge write completed

R1. Read request
R2. Response to read

“Pulling” atualizações

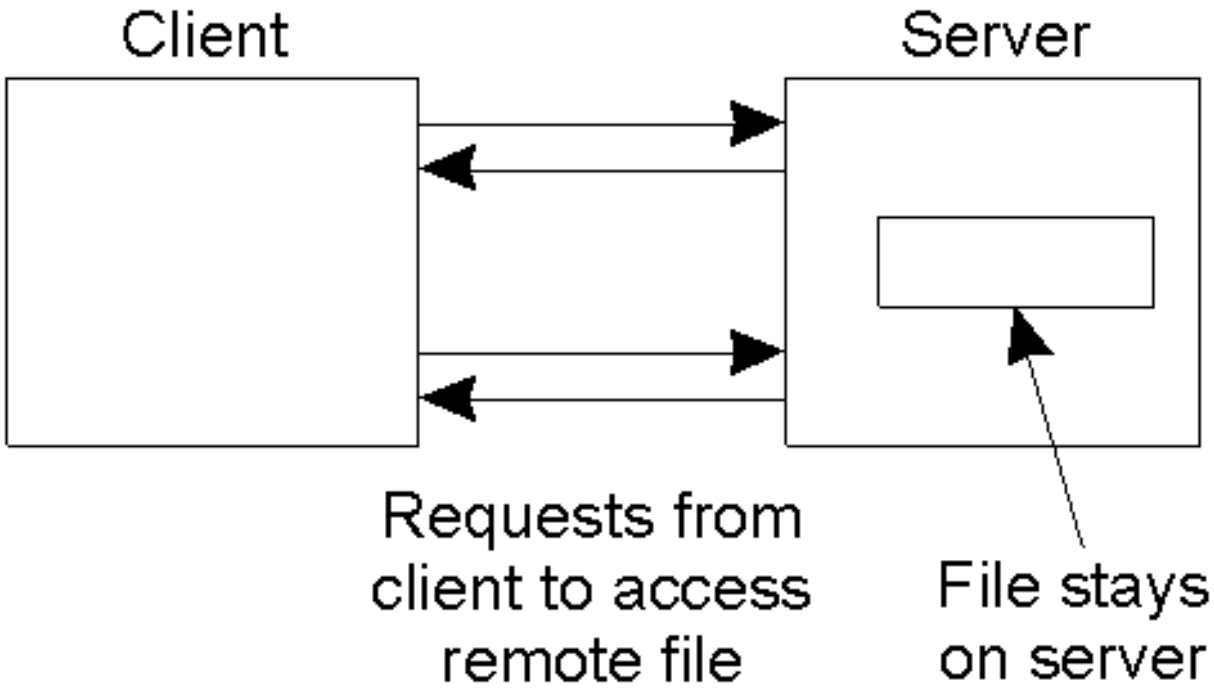


1. Read or write request
2. Forward request to current server for x
3. Move item x to client's server
4. Return result of operation on client's server

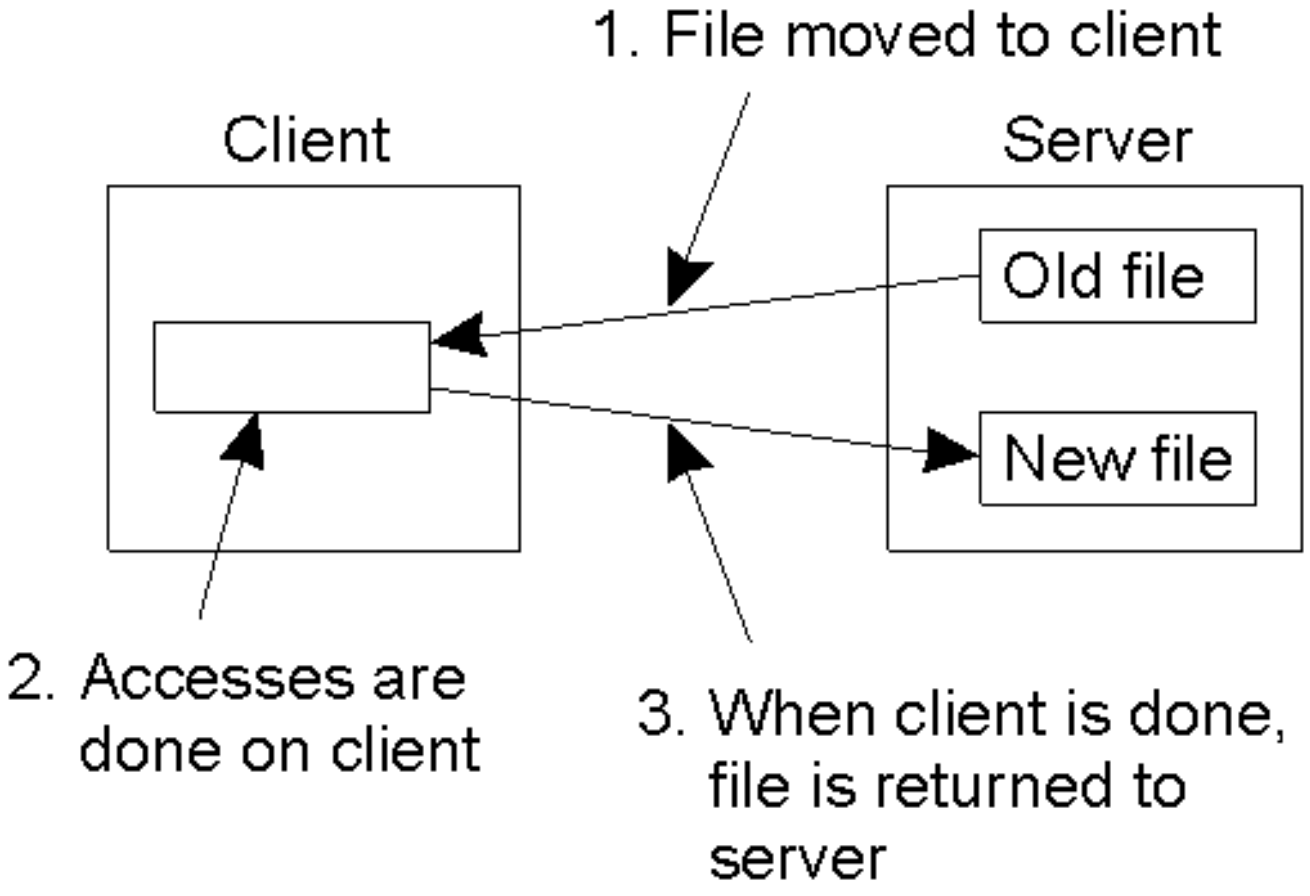
Sistema de arquivos distribuídos

- NFS – originalmente desenvolvido para as estações de trabalho Sun, mas com uso generalizado hoje em dia
- Usa um modelo de serviço de arquivos remotos

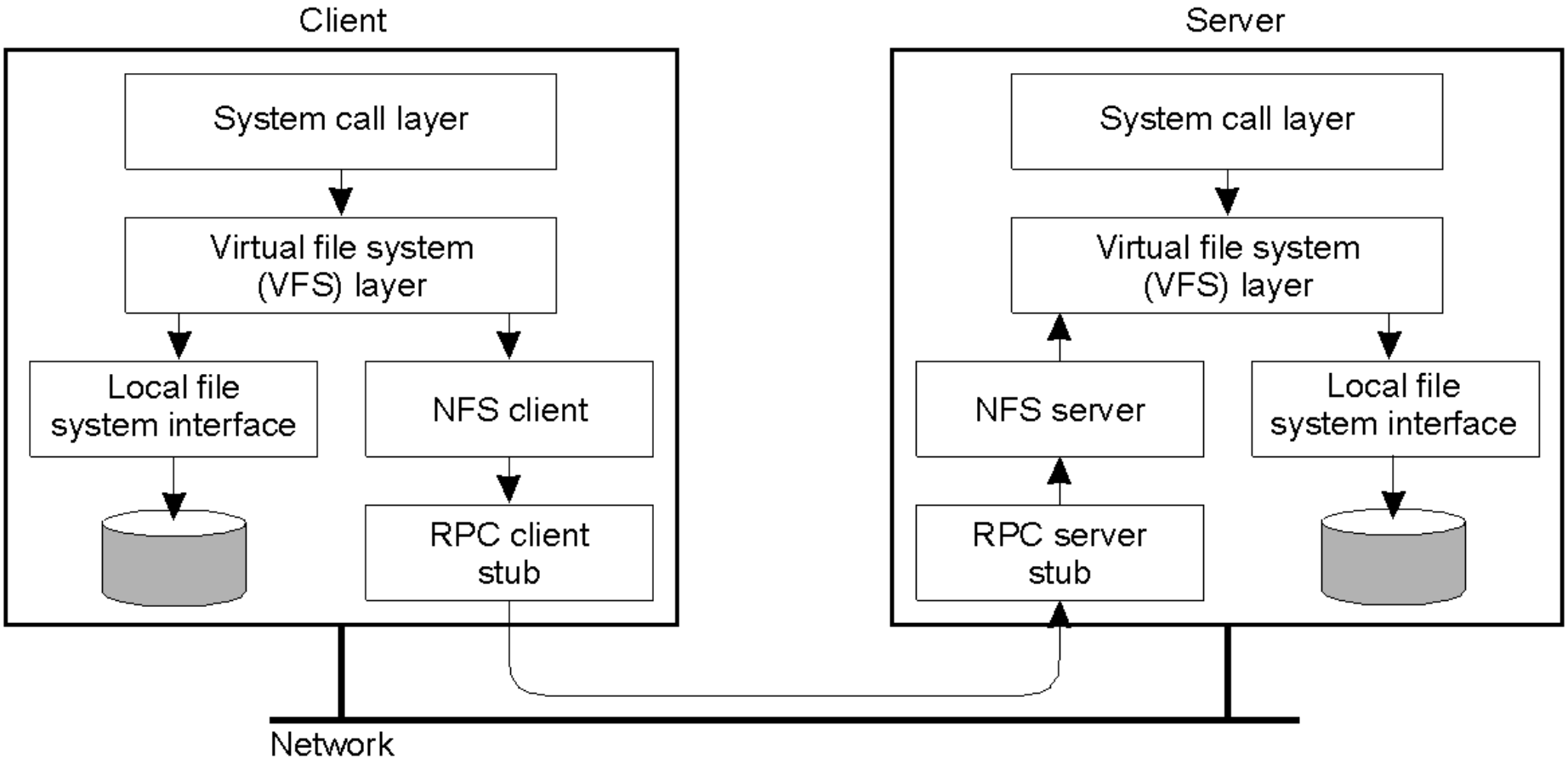
Modelo de acesso remoto



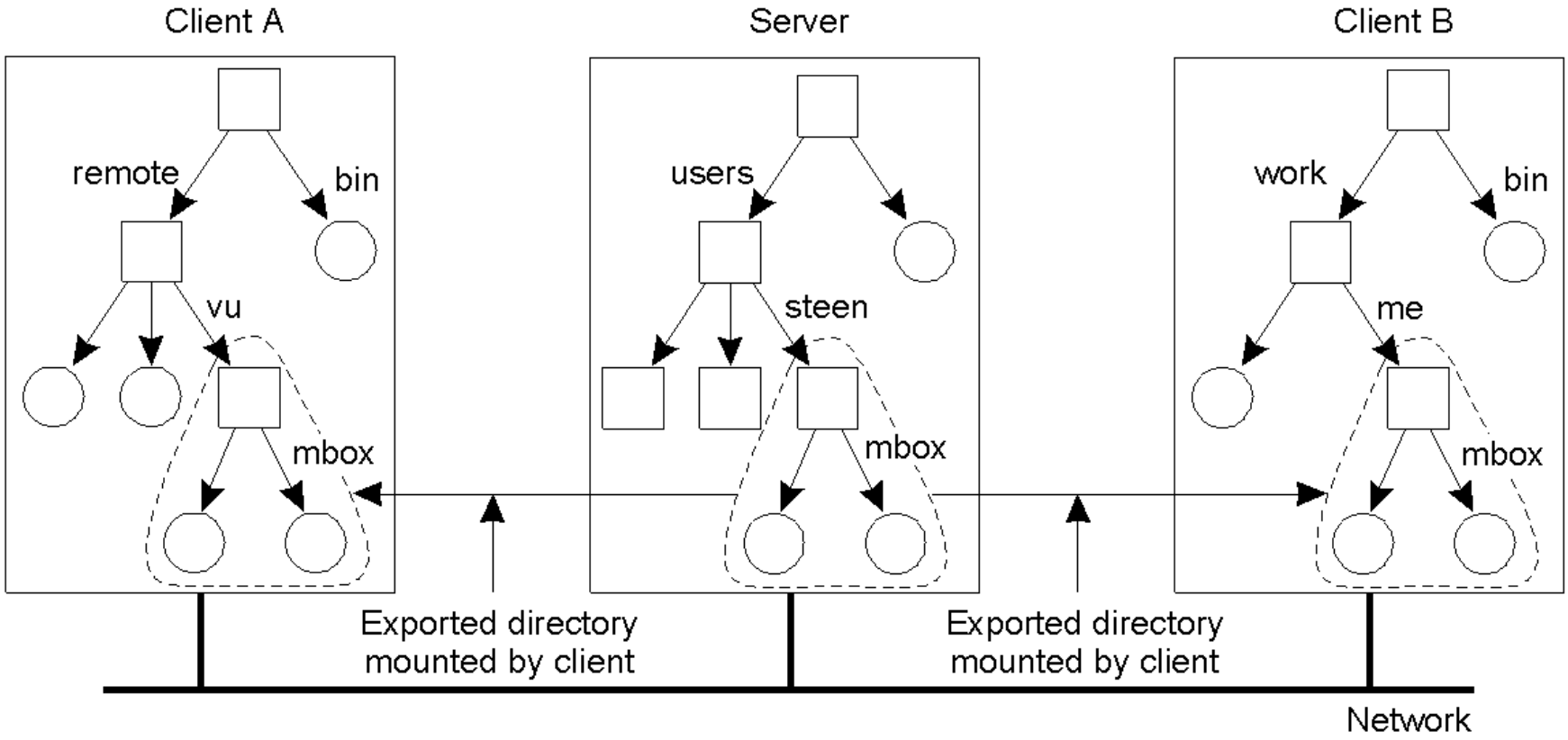
Modelo de upload/download



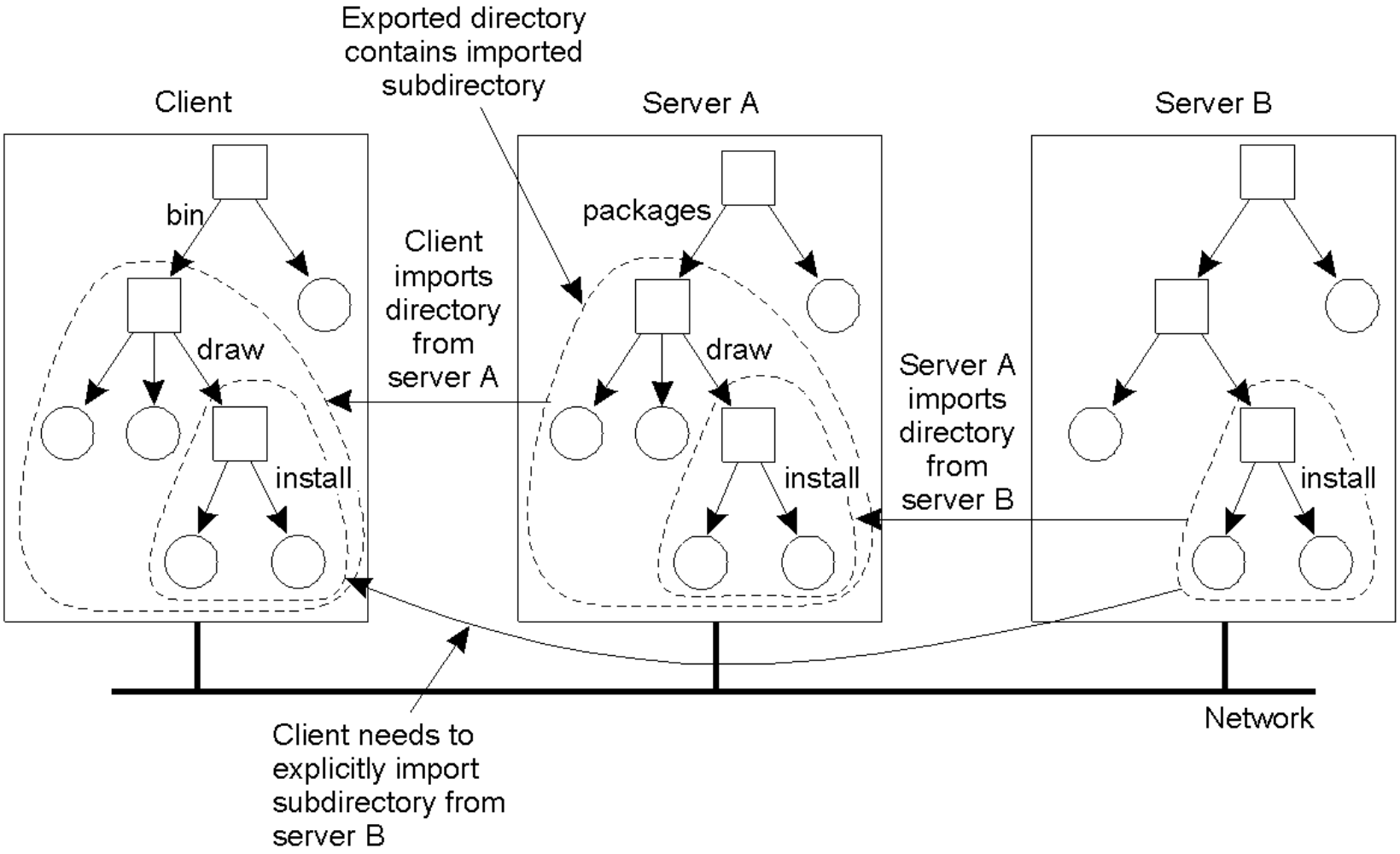
Arquitetura básica do NFS



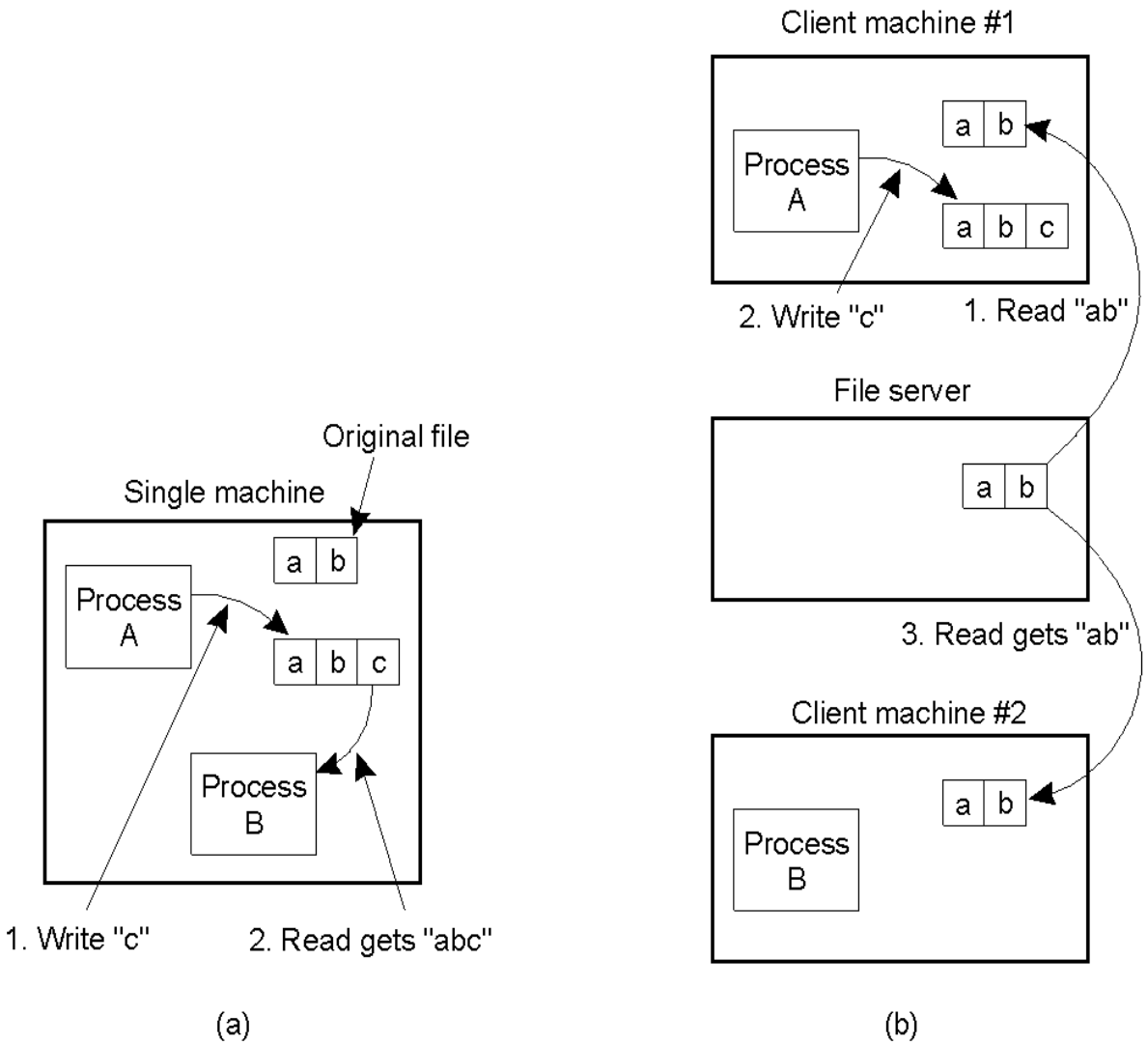
Nomeação


Client A
Server
Client B
remote
bin
vu
mbox
users
steen
mbox
work
bin
me
mbox
Exported directory
mounted by client
Exported directory
mounted by client
Network

Nomeação


Exported directory contains imported subdirectory
Client
Server A
Server B
bin
draw
install
packages
draw
install
install
Client imports directory from server A
Server A imports directory from server B
Network
Client needs to explicitly import subdirectory from server B

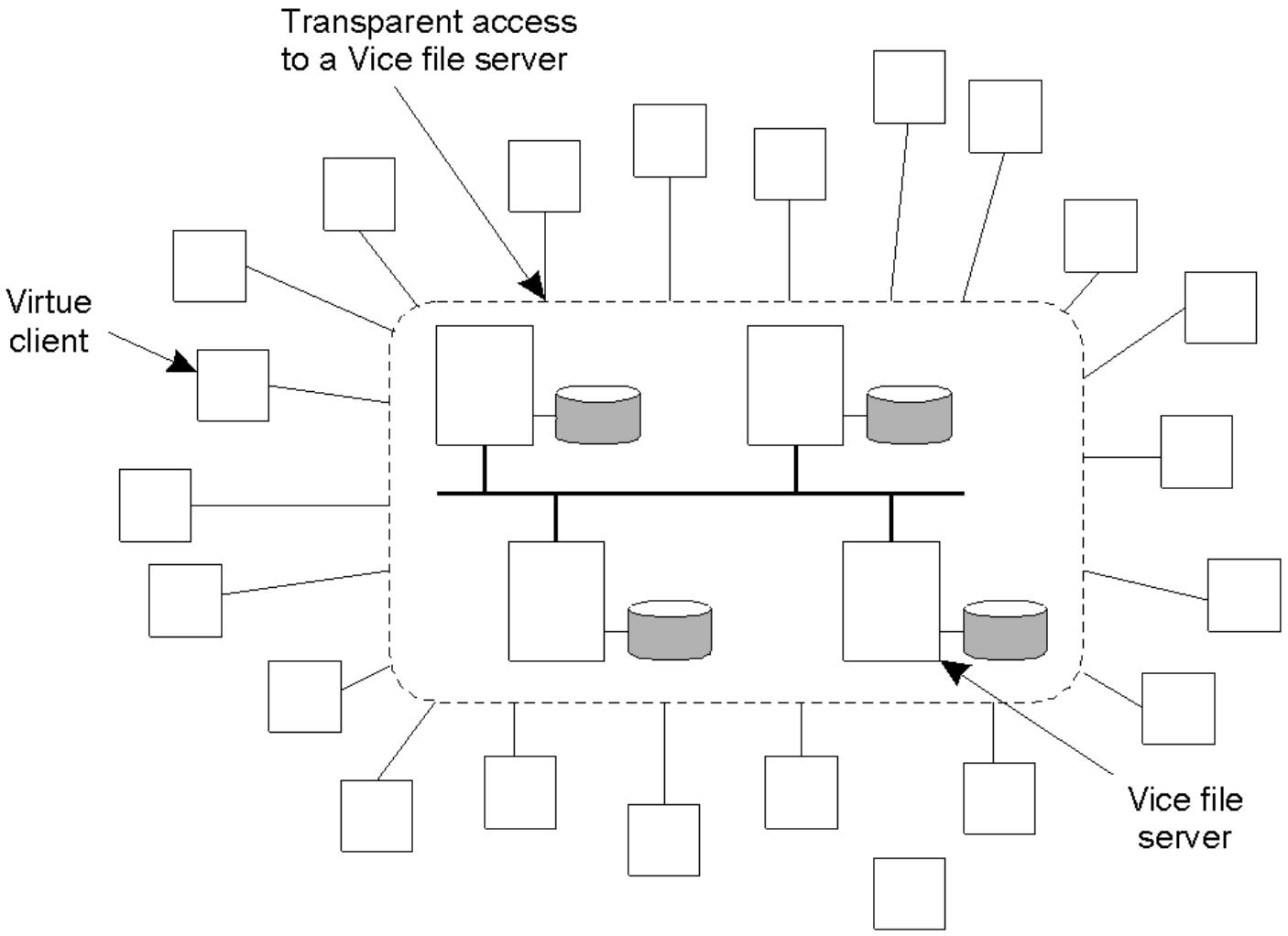
Sincronização



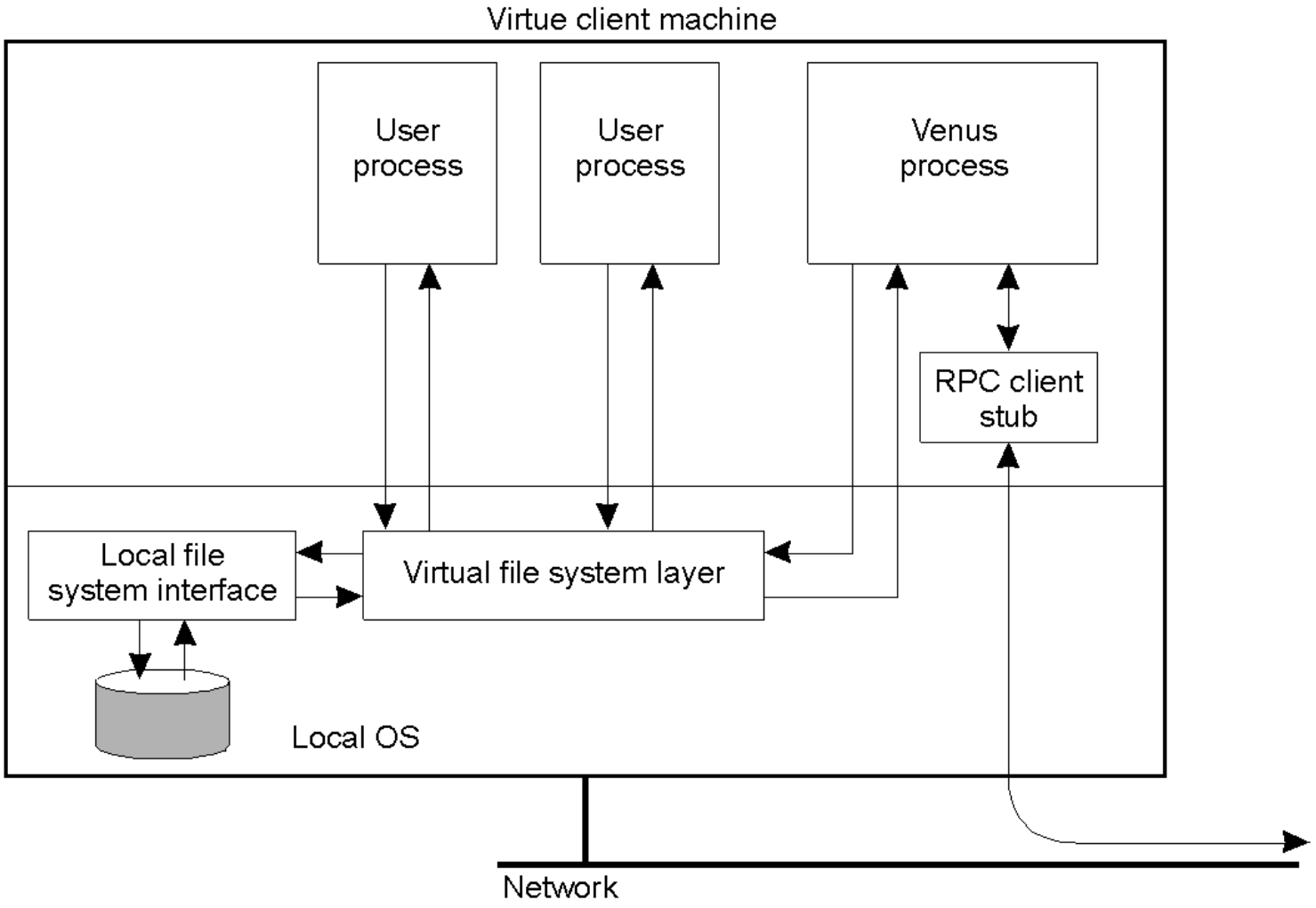
CODA

- É descendente do AFS (Andrew File System)
- Integrado em vários sistemas baseado em Unix, como o linux por exemplo
- Busca garantir uma alta disponibilidade de serviços

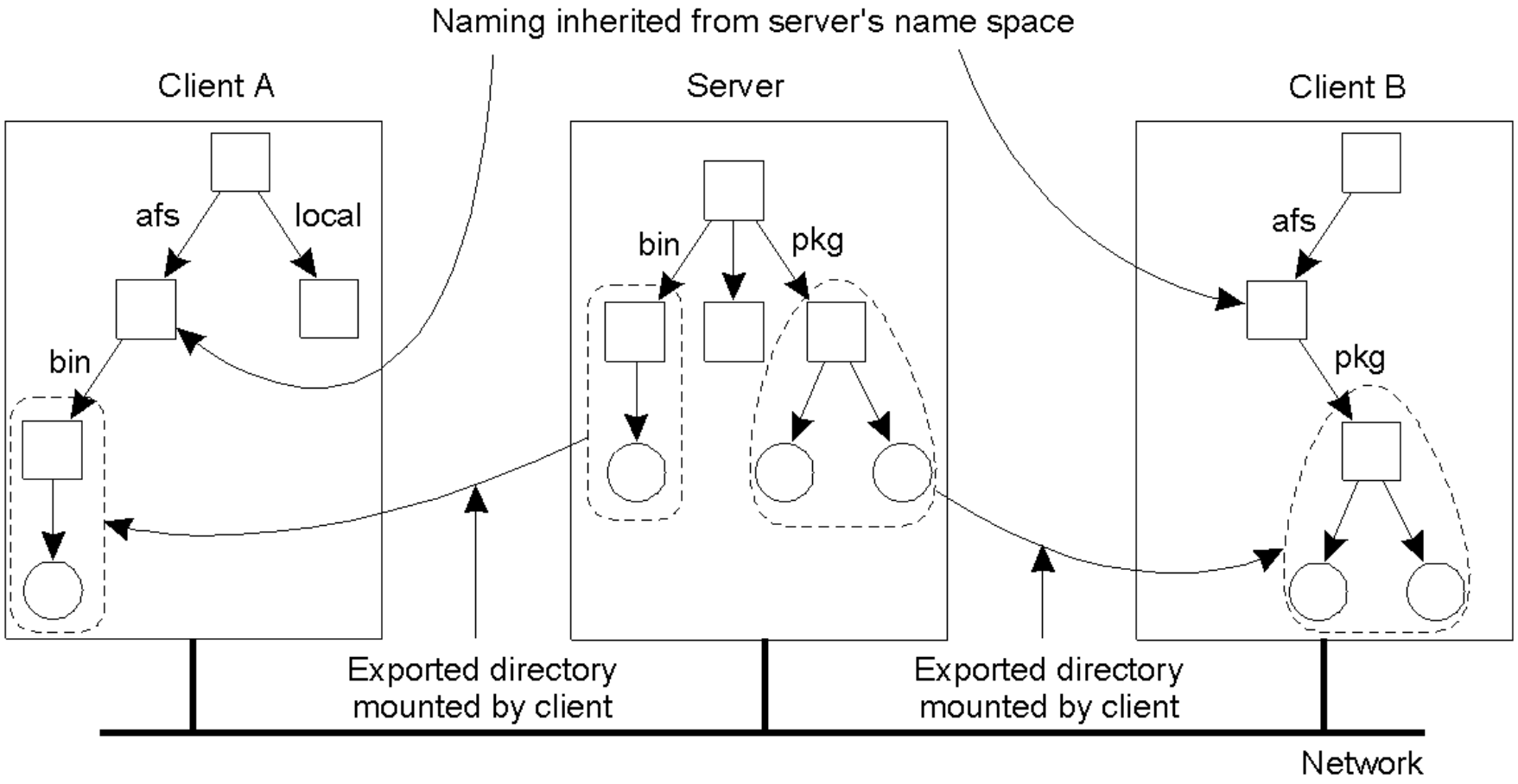
Organização



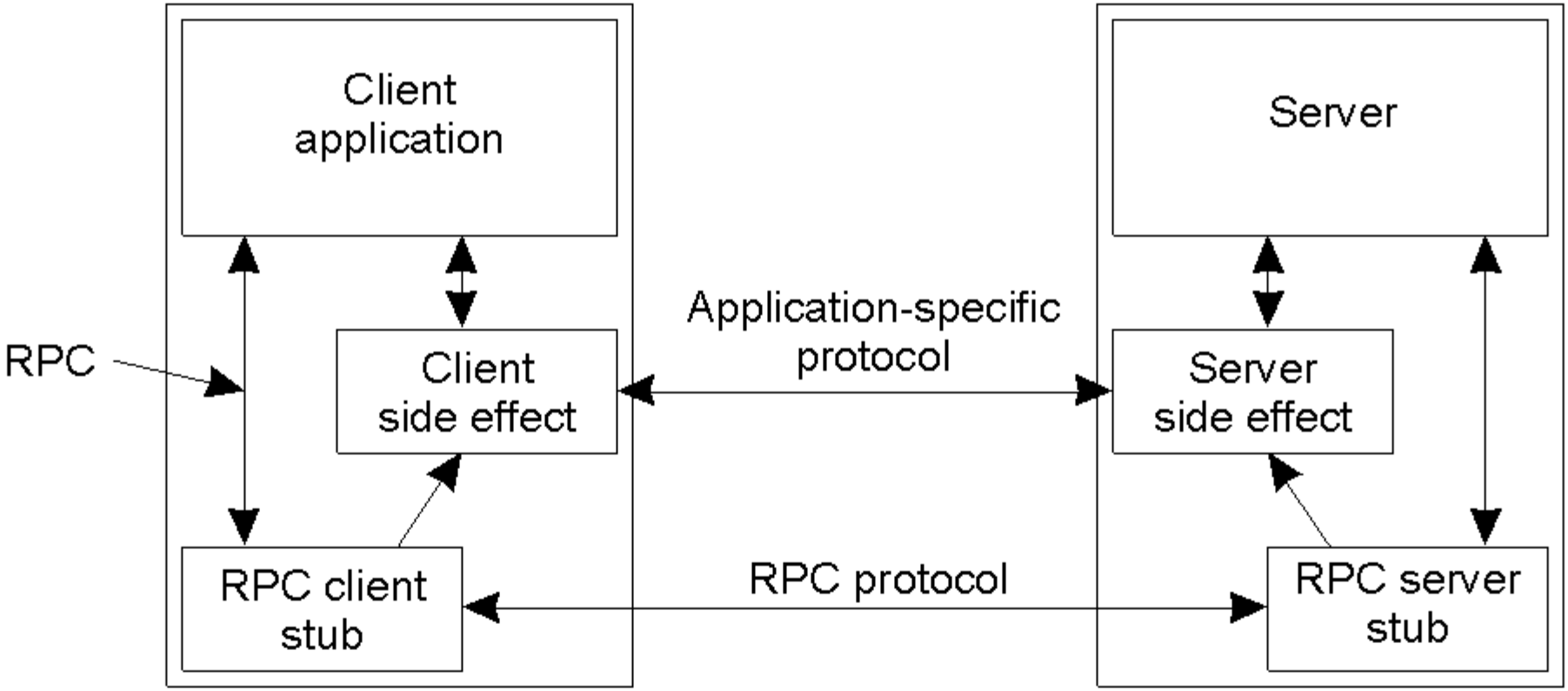
Organização



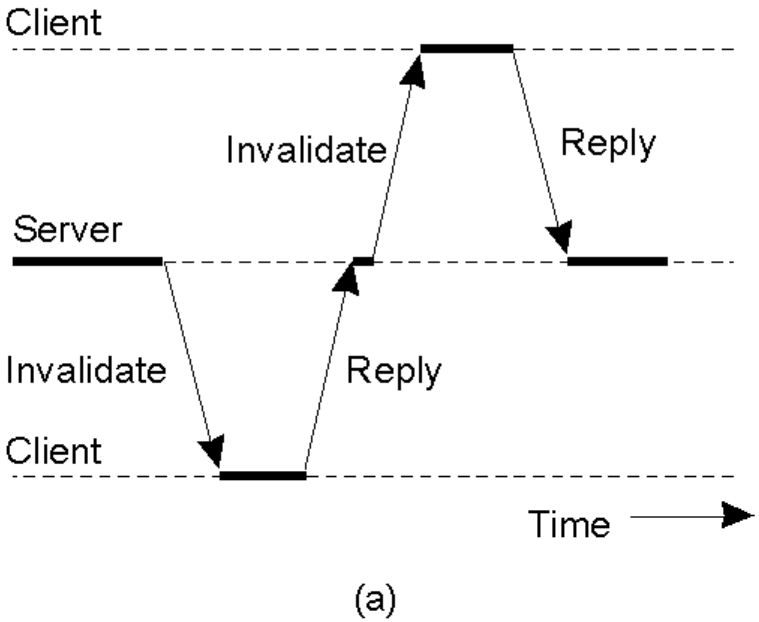
Nomeação


Naming inherited from server's name space
Client A
Server
Client B
afs
local
bin
bin
pkg
afs
pkg
Exported directory
mounted by client
Exported directory
mounted by client
Network

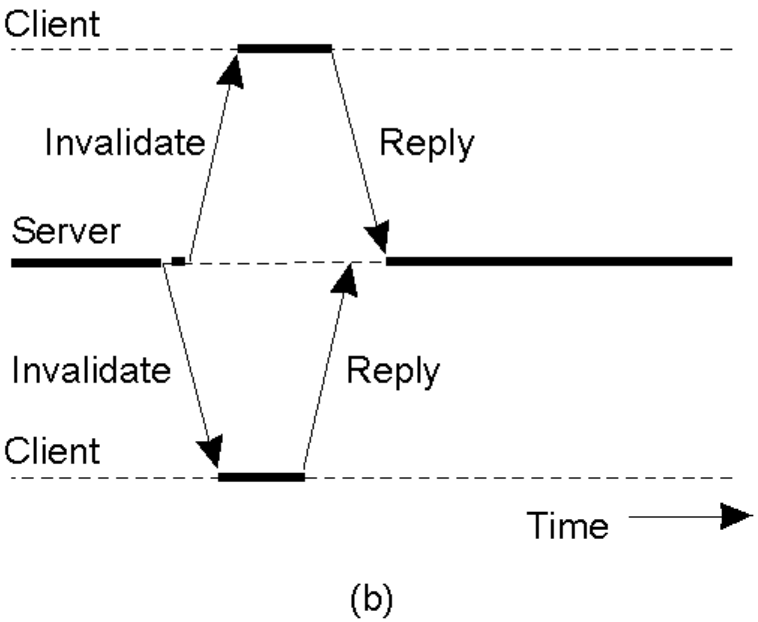
Comunicação



Comunicação



(a)



(b)

Outros sistemas

- Plan9 – busca uniformidade através da centralização de servidores
- XFS – busca a eliminação da figura de servidores com a total distribuição de serviços
- SFS – busca garantir a segurança no acesso aos arquivos do SD