# Capítulo II – Modelos de Programação Distribuída (parte 2)

**From: Coulouris, Dollimore and Kindberg**
**Distributed Systems: Concepts and Design**
Edition 3, © Addison-Wesley 2001

Paula Prata,
Departamento de Informática da UBI
http://www.di.ubi.pt/~pprata

1 – Modelos de comunicação por mensagens

2 – Exemplo: Comunicação por mensagens através de Sockets (em Java)

3 – Modelos de avarias

4 - Modelos Arquitecturais

Cliente / Servidor

Múltiplos Servidores

Proxies

Peer processes

5 - Modelos Fundamentais

Interacção

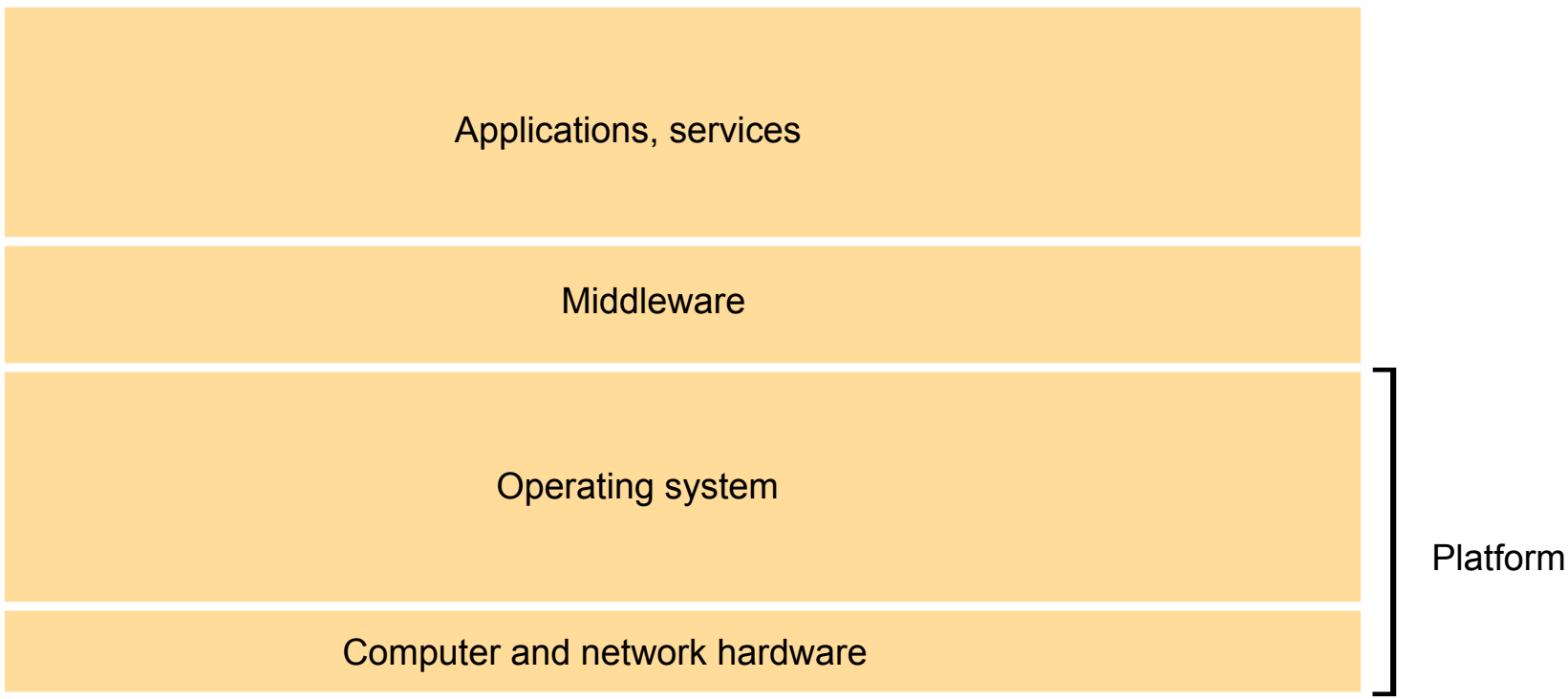
Falhas

Segurança

# 4 - Modelos arquitecturais

Um modelo arquitectural de um sistema distribuído é a estrutura do sistema em termos de localização das suas diferentes partes, e do papel que cada parte desempenha e como se interrelacionam.

Camadas de um sistema distribuído:



## 4 - Modelos arquitecturais

Ex.los de plataformas:

- Intel PII/Windows
- Intel x86/Linux
- Power PC/Solaris
- ...

Middleware:

Camada de software que tem o objectivo de mascarar a heterogeneidade de um sistema distribuído fornecendo um modelo de programação uniforme.

Ex.los

- Sun RPC
- Java RMI
- Corba
- Microsoft DCOM ...

# 4 - Modelos arquitecturais

## Modelo Cliente – Servidor

*(Modelo independente do middleware utilizado)*

Modelo assimétrico:

**Servidor:** processo passivo que quando contactado por um cliente envia a resposta

**Cliente:** contacta o servidor com o objectivo de utilizar um serviço; envia um pedido (request/invocation) e fica à espera da resposta (reply/result)

Cliente e Servidor são papéis que podem ser desempenhados.

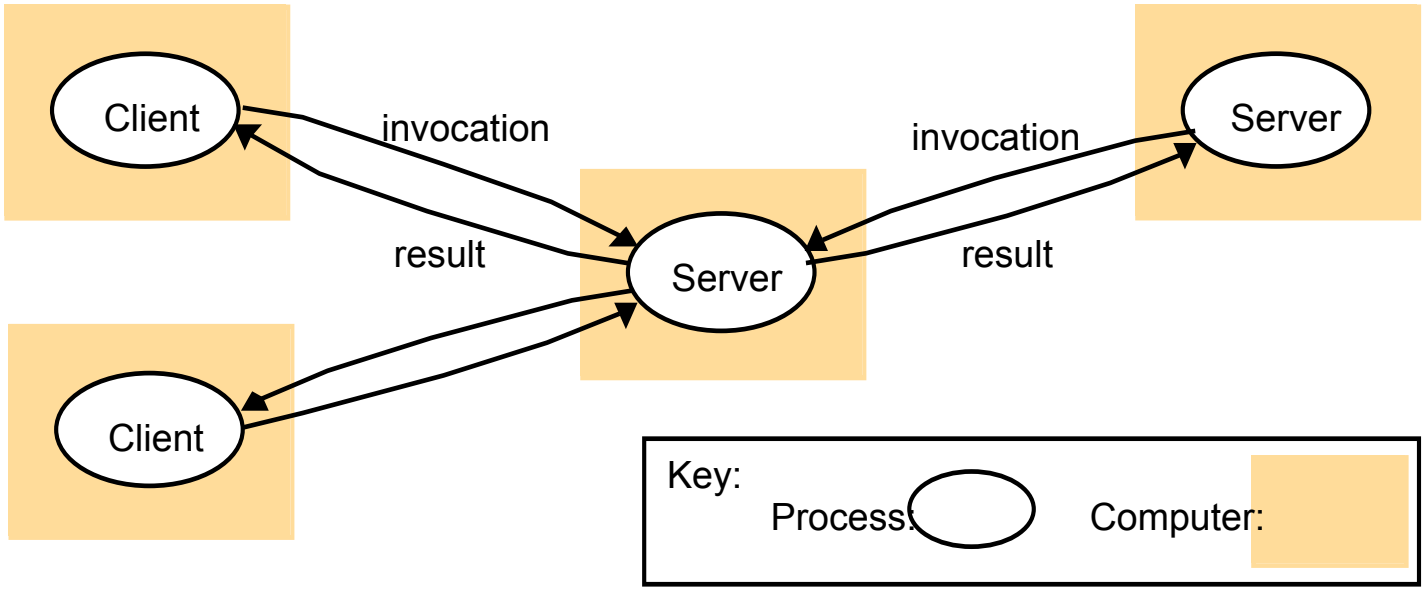
Uma entidade pode simultaneamente ser cliente e servidor. Um processo para responder a um pedido, pode ter que recorrer a outro serviço, sendo cliente deste.

*Ex.lo: um motor de pesquisa que usa "web crawlers" para pesquisar servidores web*

...

# 4 - Modelos arquitecturais

## Modelo Cliente – Servidor

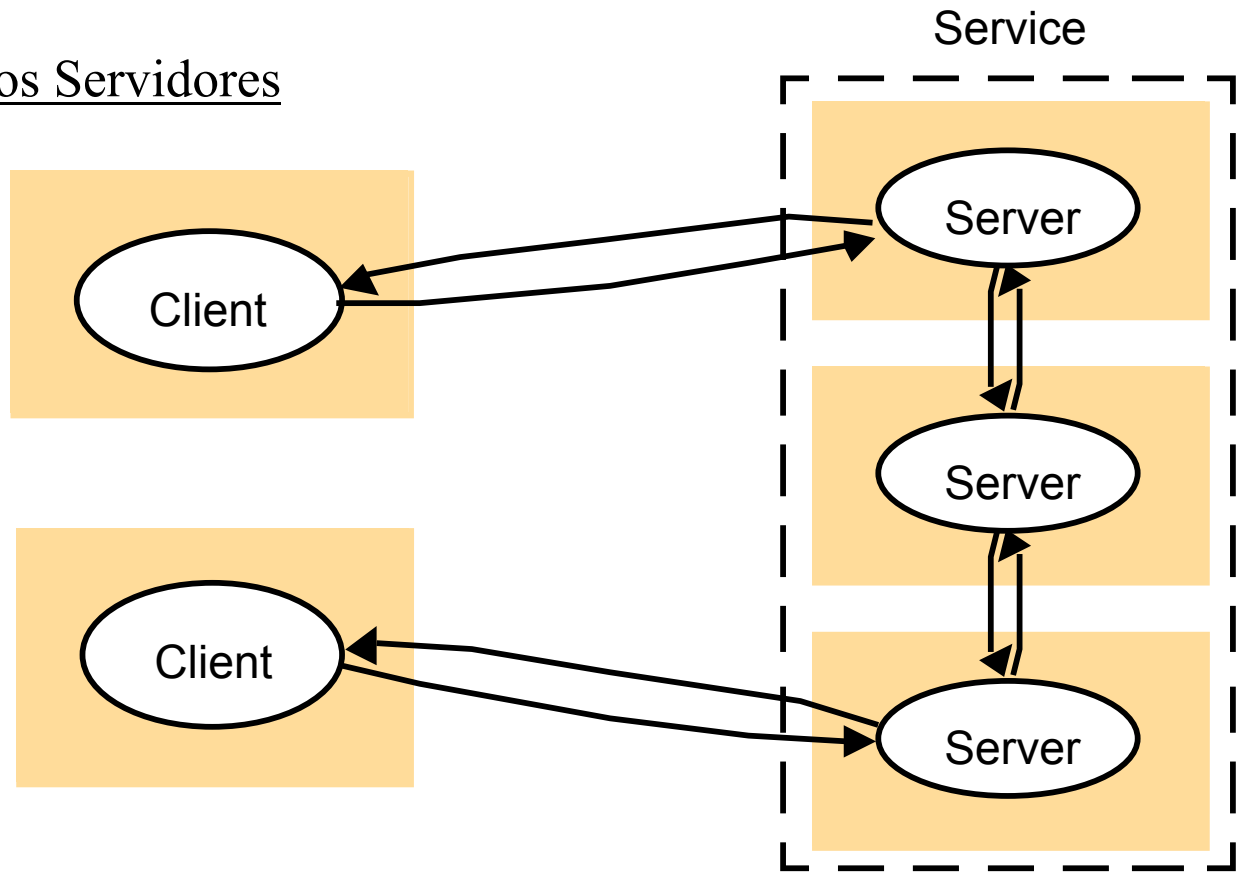



## Múltiplos Servidores

Um serviço pode ser implementado por vários processos servidores localizados em diferentes computadores.

# 4 - Modelos arquitecturais

## Múltiplos Servidores



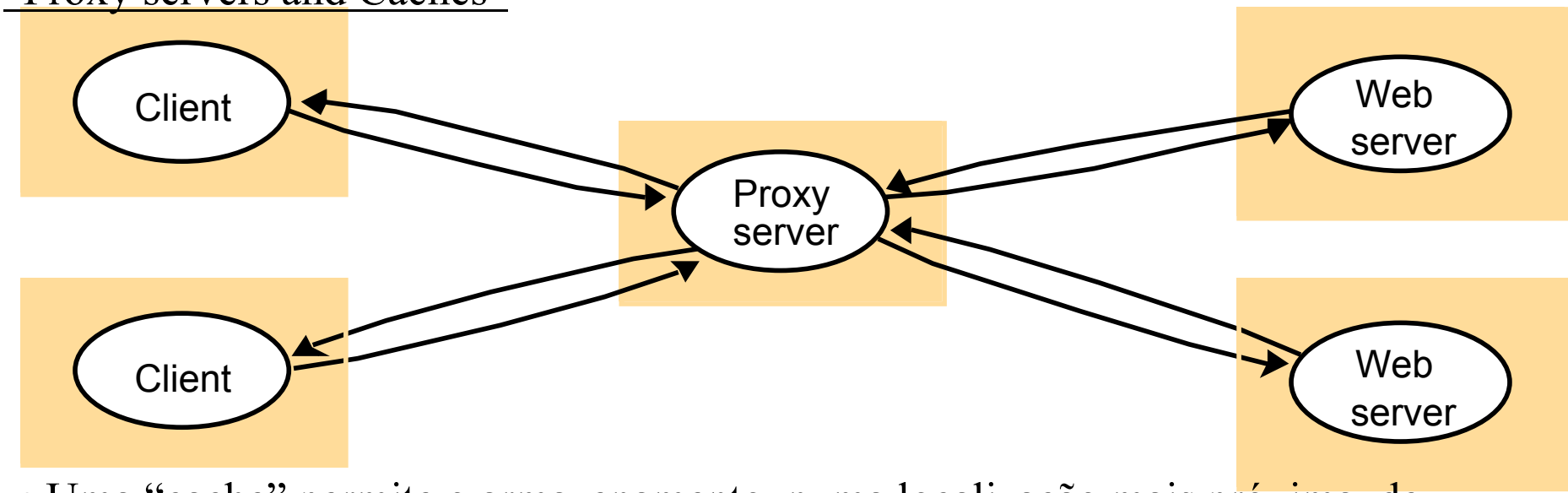

• Entidades que fornecem o serviço estão distribuídas por diferentes máquinas

ou

• Cópias replicadas pelas diferentes máquinas - > disponibilidade > tolerância a falhas

# 4 - Modelos arquitecturais

"Proxy servers and Caches"




• Uma "cache" permite o armazenamento, numa localização mais próxima, de dados/objectos recentemente usados

• Quando um cliente necessita de um objecto, o serviço de "caching" verifica se possui uma cópia actualizada do objecto, em caso afirmativo fornece essa cópia.

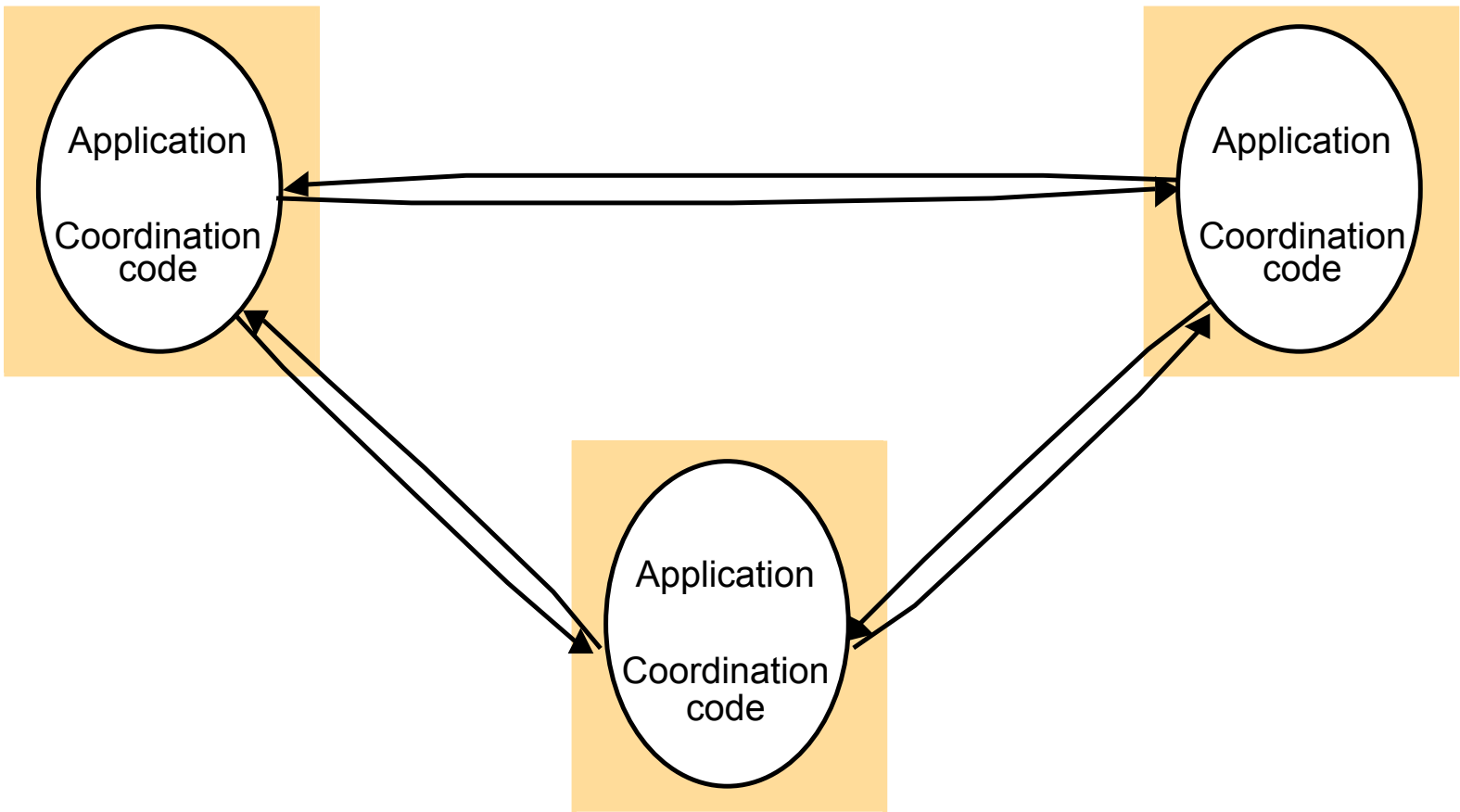
• Uma "cache" pode estar localizada no cliente ou em servidores "proxy" que são partilhados por vários clientes.

• Objectivo: aumentar a disponibilidade e a performance do serviço

# 4 - Modelos arquitecturais

"Processos pares" (peer processes)

Todos os processos desempenham papeis similares. Cada processo é responsável pela consistência dos seus dados (recursos) e pela sincronização das várias operações.



# 4 - Modelos arquitecturais

Variantes do modelo cliente servidor resultantes de

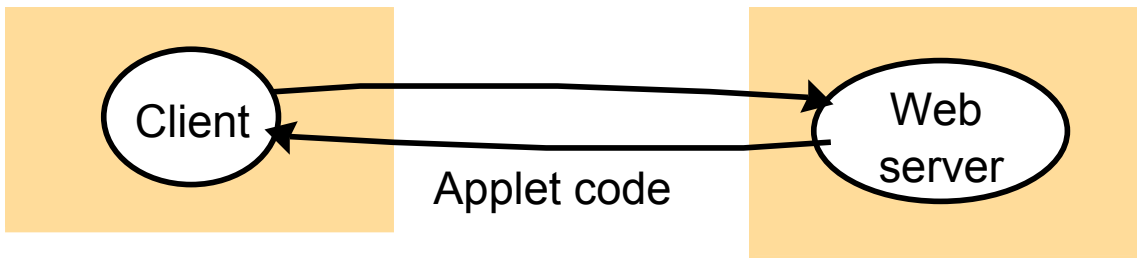
. Uso de código móvel

. Uso de sistemas com hardware limitado

. Requisitos de adicionar/remover ao/do sistema periféricos móveis

Web Applets

a) client request results in the downloading of applet code




b) client interacts with the applet

# 4 - Modelos arquitecturais

## Agentes móveis

Um Agente é um programa executável que pode "mover-se" de uma máquina para outra. Age em nome de um utilizador específico, e num dado computador para o qual se transfere realiza algum serviço para o seu proprietário, podendo obter informações que mais tarde transmitirá ao local de origem.

*Entidade capaz de interagir autonomamente com o ambiente que o rodeia, podendo apresentar características de adaptabilidade, mobilidade, cooperação ou competição.*

. Problemas de segurança

. Dificuldades em realizar o trabalho devido a problemas impossíveis de prever

## Network Computers

Computadores sem disco nem periféricos, contando com o apoio da rede para fornecer os serviços ao utilizador. Aplicações executam localmente, os ficheiros são geridos por um servidor remoto.

*Solução económica para centros de computação com poucos recursos*

# 4 - Modelos arquitecturais

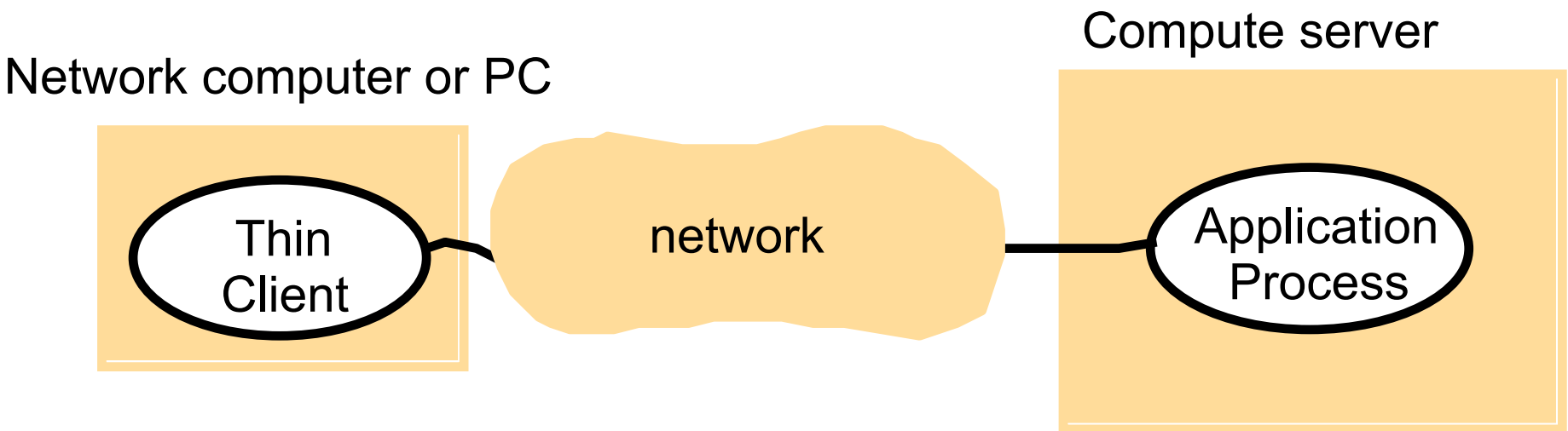
Thin Clients

- interface gráfica, baseada em windows, na máquina local ao utilizador

- as aplicações executam no servidor

*problemas para aplicações gráficas interactivas*

Ex. Citrix WinFrame



# 4 - Modelos arquitecturais

Equipamentos móveis e redes espontâneas

- Laptops, PDA (Personal Digital Assistant), Telemóveis, Câmeras Digitais, Máquinas de Lavar Roupa, relógios, etc

- Protocolos Wireless: BlueTooth, Infrared, HomeRF

Principais características das redes sem fios:

- Configuração é feita automaticamente sem intervenção humana

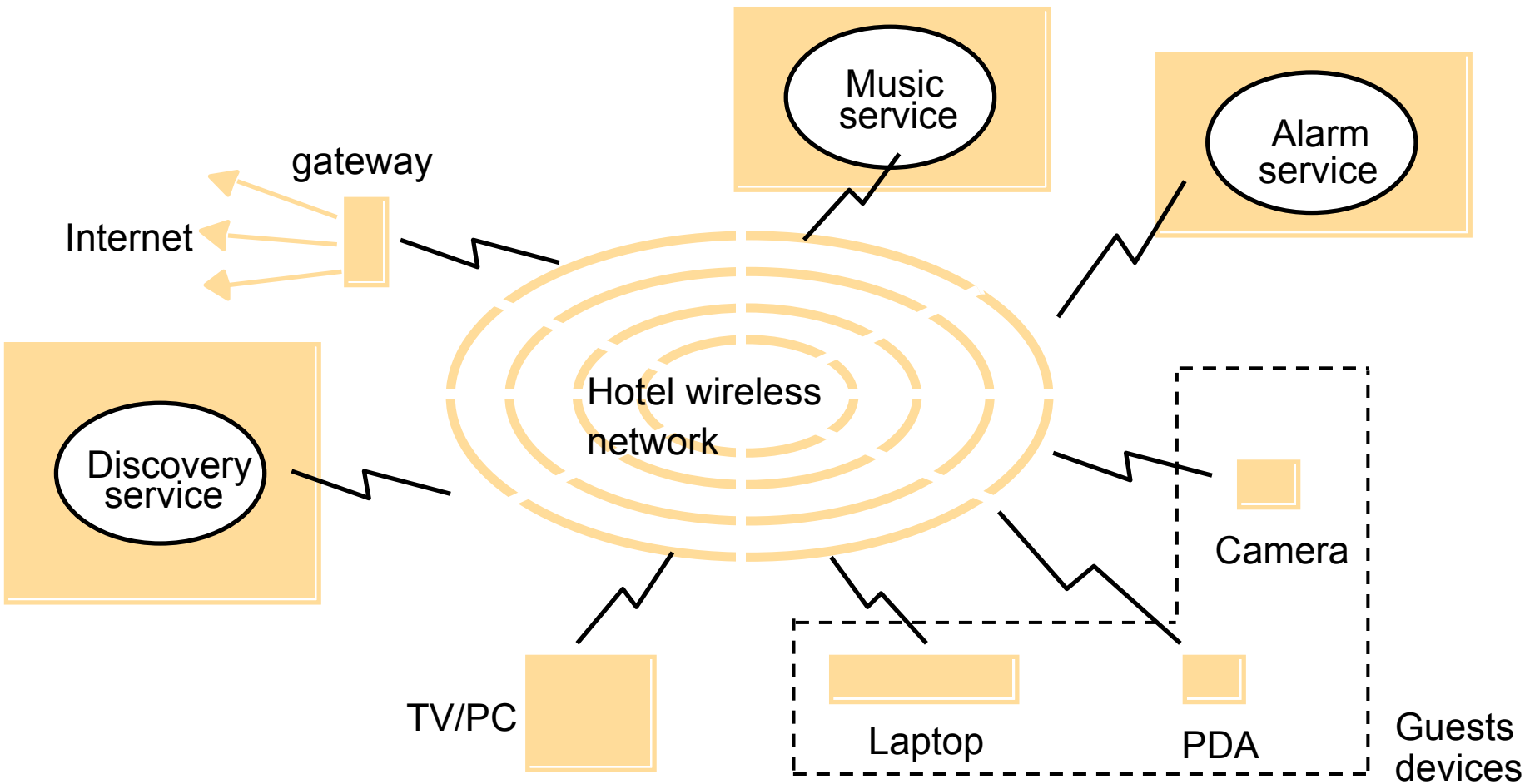
- Os equipamentos digitais móveis descobrem por si os serviços disponíveis

Problemas:

- Conexão limitada (se os dispositivos se afastam demasiado do local de transmissão?)

- Segurança e privacidade

# 4 - Modelos arquitecturais



## 4 - Modelos arquitecturais

Redes espontâneas exigem:

Um meio de os clientes (equipamentos móveis) descobrirem que serviços estão disponíveis na rede a que se ligaram.

Um "discovery service" é um servidor (um ou mais processos) que mantém uma lista dos tipos e características dos serviços disponíveis dentro da rede local sem fios.

Oferecem dois tipos de serviços:

- Registo de serviços – aceita pedidos para registar numa base de dados os detalhes de cada serviço disponível

- Lookup de serviços - aceita "queries" aos serviços disponíveis, fornecendo detalhes suficientes para que o cliente se possa ligar ao serviço que escolher

# 5 - Modelos Fundamentais

Modelos fundamentais

Os sistemas distribuídos podem ainda ser analisados segundo 3 aspectos transversais a todos os sistemas:

. Modelo de Interacção

. Modelo de Avarias

. Modelo de Segurança

a) Modelo de interacção

Interacção é a acção (comunicação e sincronização) entre as partes para realizar um qualquer trabalho

É afectada por dois aspectos:

1 . Performance dos canais de comunicação

2 . Inexistência de um tempo global

## 5 - Modelos Fundamentais

1 . Performance dos canais de comunicação

. Latência

*Intervalo de tempo que medeia entre o início da transmissão de uma mensagem por um processo e o início da sua recepção pelo outro processo.*

Depende de:

- Demora (“delay”) de transmissão pela rede
- Tempo requerido pelo sistema operativo em ambos os lados da comunicação
- Demora no acesso aos recursos da rede

. Largura de banda (“bandwidth”)

*Total de informação que pode ser transmitida pela rede num dado tempo*

## 5 - Modelos Fundamentais

1 . Performance dos canais de comunicação

. Jitter

*Variação no tempo necessário para enviar grupos de mensagens consecutivos constituintes de uma informação transmitida de um ponto para outro na rede.*

*(importante na transmissão de som e imagem)*

2. Inexistência de um tempo global

Cada computador tem um relógio "clock" interno.

. Cada relógio tem um "**drift"** (um desvio) do tempo de referência

. Os "drifts" de dois relógios distintos são também distintos

*(o que significa que entre eles o tempo será sempre divergente)*

- Uma solução passa por obter o tempo fornecido pelo GPS – Global Positioning System, e enviar aos participantes do sistema distribuído. Delays no envio dessa mensagem!!!

## 5 - Modelos Fundamentais

Duas variantes no modelo de interacção

1 – Sistemas distribuídos síncronos

Sistemas onde podem existir limites máximos de tempo conhecidos para:

tempos de execução dos processos, atrasos na comunicação, variações no tempo de referência

Se:

- o tempo necessário para executar cada passo de um processo tem um limite inferior e um limite superior conhecidos

- cada mensagem transmitida por um canal é recebida dentro de um limite de tempo conhecido

- cada processo tem um relógio cujo desvio máximo para o tempo de referência é conhecido

podem definir-se definir “timeouts” para detectar falhas.

*Dificuldade em encontrar os limites para os tempos, mais difícil ainda provar a sua correcção*

# 5 - Modelos Fundamentais

## 2 – Sistemas distribuídos assíncronos

Não possui limites para:

. tempo de execução dos processos – cada passo de execução pode levar um tempo arbitrariamente longo

. tempo de transmissão de mensagens - uma mensagem pode chegar rapidamente ou demorar dias

. o desvio do tempo de referência pode ser qualquer

(exemplo de um sistema assíncrono: Internet)

Como lidar com longos tempos de espera:

O sistema podem avisar o utilizador que o tempo de espera pode ser longo e solicitar uma alternativa

Dar oportunidade ao utilizados para fazer outras coisas

...

# 5 - Modelos Fundamentais

## O modelo de interacção e o problema do ordenação de eventos

*Por vezes é importante conhecer a ordem pela qual ocorreu um conjunto de eventos.*

Ex.lo

Sejam os utilizadores X,Y,Z e A que trocam mails para marcar uma reunião:

. X envia uma mensagem, com o assunto: Meeting, para Y,Z e A

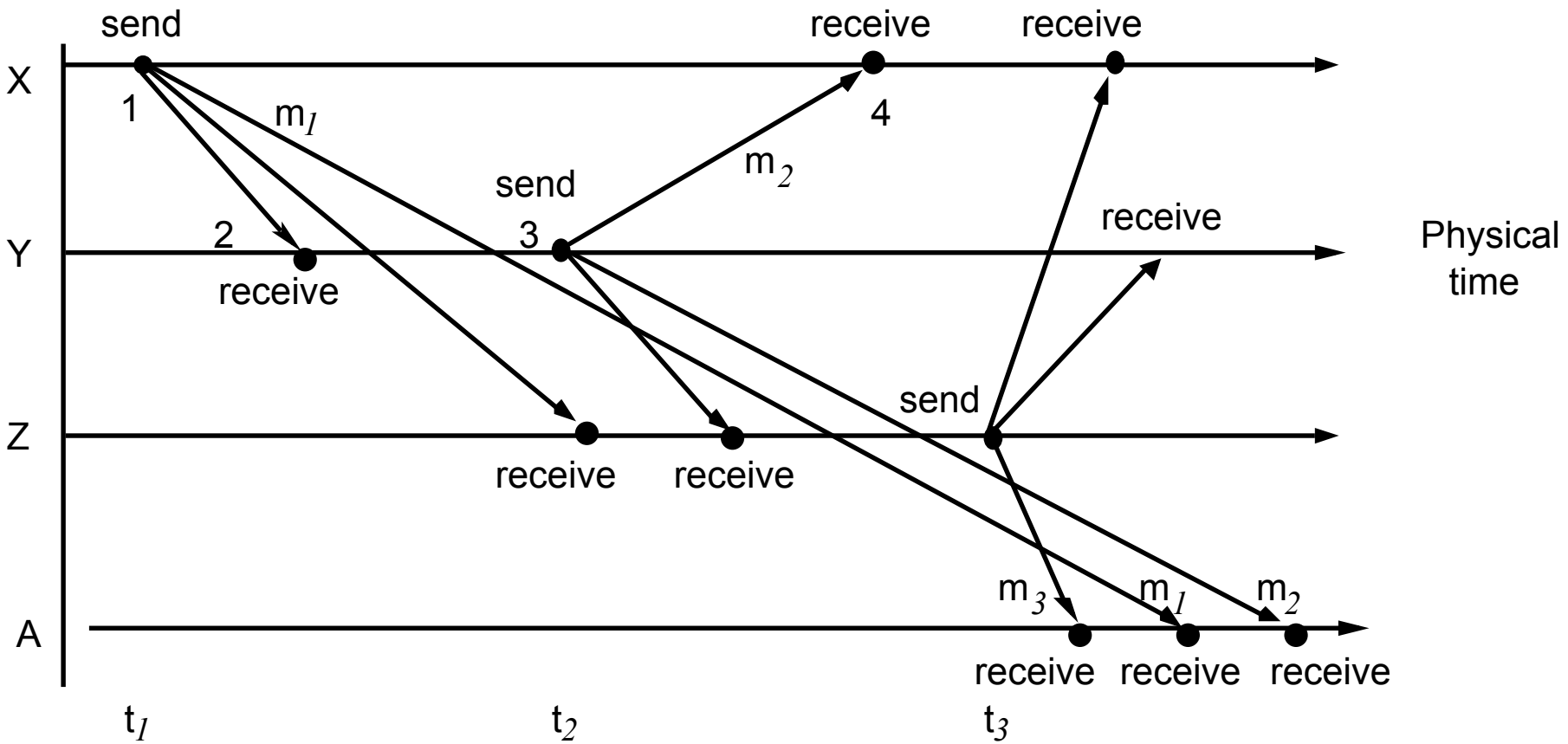
. Y e Z respondem para os outros com o assunto: Re: Meeting

Uma vez que não há limites no tempo de comunicação as mensagens podem ser entregues de tal forma que o utilizador A receba as mensagens pela ordem:

| **De:** | **Subject:** |
|---|---|
| Z | Re:Meeting |
| X | Meeting |
| Y | Re:Meeting |

# 5 - Modelos Fundamentais

## O modelo de interacção e o problema do ordenação de eventos



# 5 - Modelos Fundamentais

## O modelo de interacção e o problema do ordenação de eventos

Solução proposta por Lamport [1978]

Criar um tempo lógico para marcar a sequência de eventos e determinar a ordem correcta em que eles aparecem no tempo.

Ver capitulo 10 (Coulouris) ...

# 5 - Modelos Fundamentais

## b) O modelo de Avarias

*Uma avaria é qualquer alteração do comportamento do sistema em relação ao esperado (i.é, em relação à sua especificação)*

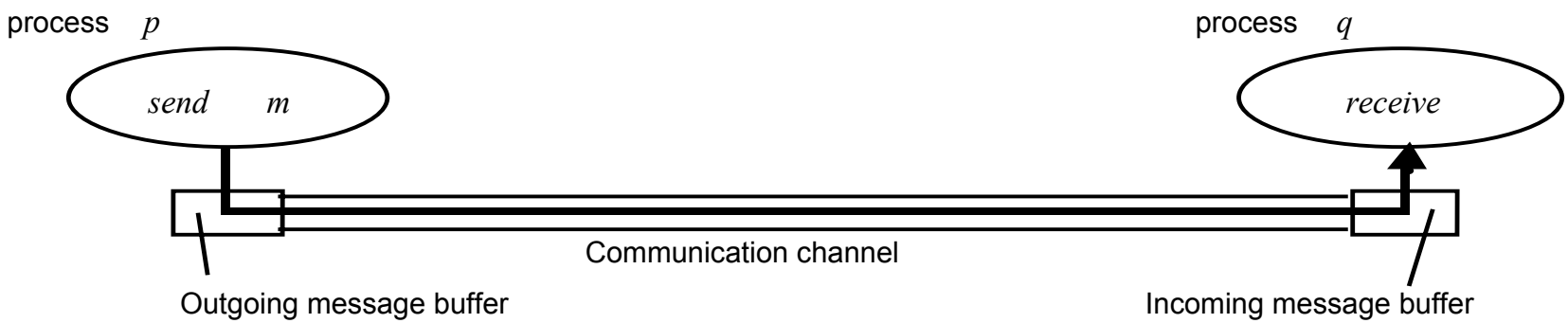
Define de que maneira as avarias podem ocorrer

. Podem atingir os processos ou os canais de comunicação

Tipos de Avarias:

. Avarias por omissão

- quando um processo "deixa de funcionar" em algum ponto do sistema distribuído
- quando o canal de comunicação falha



# 5 - Modelos Fundamentais

. Avarias por omissão

Fail-stop – o processo bloqueou (crashed) e esse facto pôde ser detectado por outros processos.

**Crash –** o processo aparentemente bloqueou mas não é possível garantir que apenas deixou de responder por estar muito lento, ou porque as mensagens que enviou não chegaram

**Omission –** uma mensagem colocada no buffer de emissão nunca chega ao buffer de recepção (pode ocorrer por falta de espaço no buffer)

Send-omission – uma mensagem perde-se entre o emissor e o buffer de emissão

Receive-omission – uma mensagem perde-se entre o buffer de recepção e o receptor

## 5 - Modelos Fundamentais

. Avarias arbitrárias

Qualquer tipo de erro pode acontecer

- Nos processos – o processo não responde, o estado do processo é corrompido, responde de forma errada, responde fora de tempo.

- Nos canais de comunicação – mensagens corrompidas, mensagens não entregues, mensagens duplicadas, mensagens inexistentes são entregues

*( São raras de ocorrer nos canais de comunicação porque o software de comunicação protege as mensagens com somas de verificação ("checksums"), números de sequenciamento, etc)*

. Avarias em tempo

- Ocorrem quando o tempo limite para um evento ocorrer é ultrapassado

- Em sistemas eminentemente síncronos é um indicativo seguro de falha

*Importantes em sistemas de tempo real.*
*(por ex.lo: sistemas de controlo, sistemas multimedia*

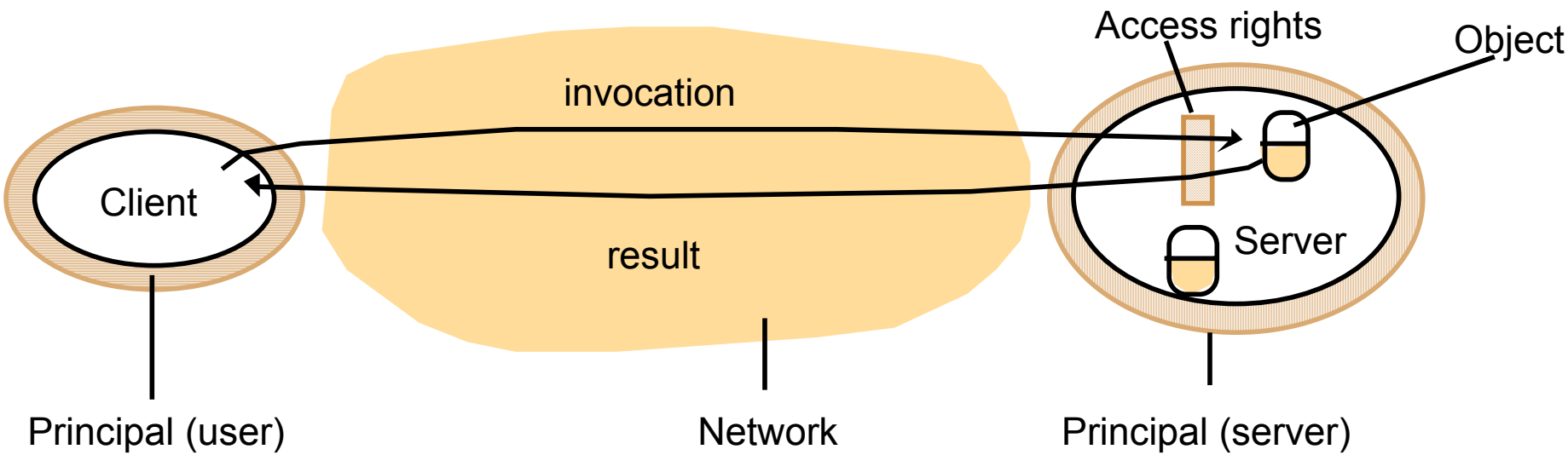
| *Class of failure* | *Affects* | *Description* |
|---|---|---|
| Fail-stop | Process | Process halts and remains halted. Other processes may detect this state. |
| Crash | Process | Process halts and remains halted. Other processes may not be able to detect this state. |
| Omission | Channel | A message inserted in an outgoing message buffer never arrives at the other end's incoming message buffer. |
| Send-omission | Process | A process completes a *send,* but the message is not put in its outgoing message buffer. |
| Receive-omission | Process | A message is put in a process's incoming message buffer, but that process does not receive it. |
| Arbitrary (Byzantine) | Process or channel | Process/channel exhibits arbitrary behaviour: it may send/transmit arbitrary messages at arbitrary times, commit omissions; a process may stop or take an incorrect step. |

| *Class of Failure* | *Affects* | *Description* |
|---|---|---|
| Clock | Process | Process's local clock exceeds the bounds on its rate of drift from real time. |
| Performance | Process | Process exceeds the bounds on the interval between two steps. |
| Performance | Channel | A message's transmission takes longer than the stated bound. |

## c) O modelo de Segurança

Protecção das entidades do sistema, processo/utilizador ("principal")

Direitos de acesso especificam que entidades podem aceder, e de que forma, a que recursos.




*Ex. Que entidade pode executar que operações num dados objecto*

## 5 - Modelos Fundamentais

- O servidor é responsável por verificar a identidade de quem (entidade) fez o pedido e verificar se essa entidade tem direitos de acesso para realizar a operação pretendida.

- O cliente deverá verificar a identidade de quem lhe enviou a resposta, para ver se a resposta veio da entidade esperada.

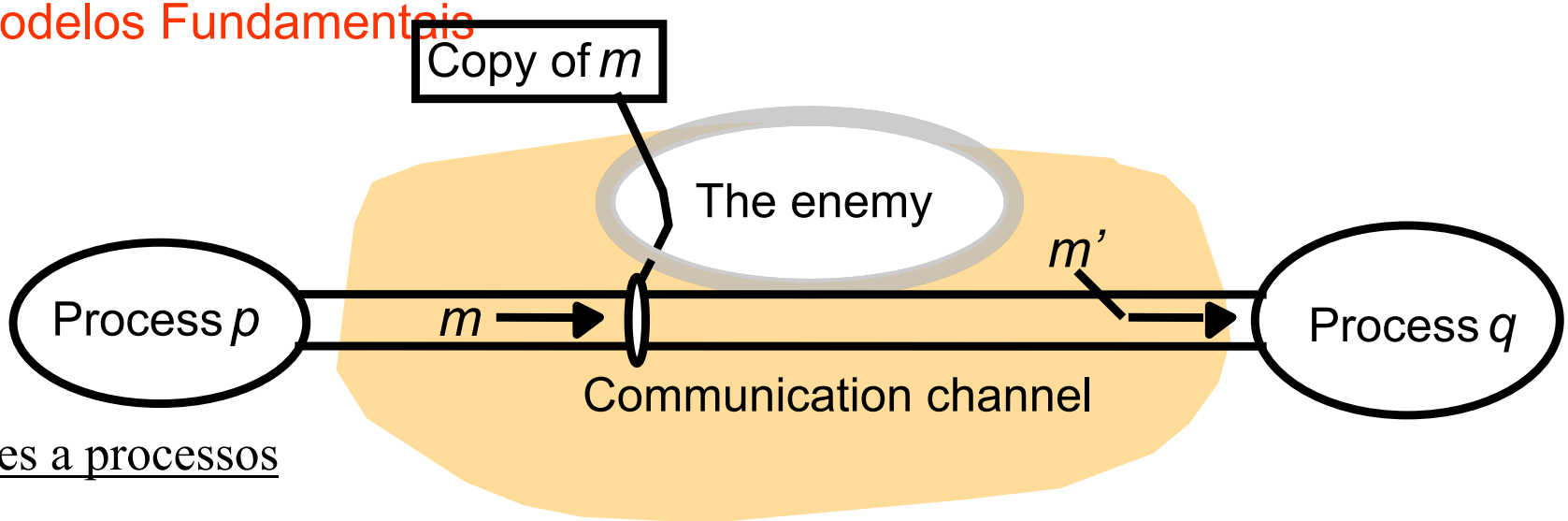
Que ameaças?

Supondo que existe um processo inimigo (adversário) capaz de:

- enviar qualquer mensagem para qualquer processo
- interceptar (ler/copiar) qualquer mensagem trocada entre 2 processos

Classificação das ameaças:

- aos processos
- à comunicação
- negação de serviço




Ataques a processos

- Ao projectar um servidor, ter consciência de que

  Os protocolos de rede não oferecem protecção para que o servidor saiba a identidade do emissor

  (IP inclui o endereço do computador origem da mensagem mas um processo inimigo pode forjar esse endereço)

  - Um cliente também não dispõe de métodos para validar as respostas de um servidor

*Em ambos os casos um processo inimigo pode fazer-se passar pela entidade (cliente /servidor) e enviar a mensagem solicitada*

# 5 - Modelos Fundamentais

Ataques a canais de comunicação

- Um processo inimigo pode copiar, alterar ou injectar mensagens na rede

A comunicação pode ser violada por processos que observam a rede à procura de mensagens significativas

*(essas mensagens podem posteriormente ser reveladas a terceiros)*

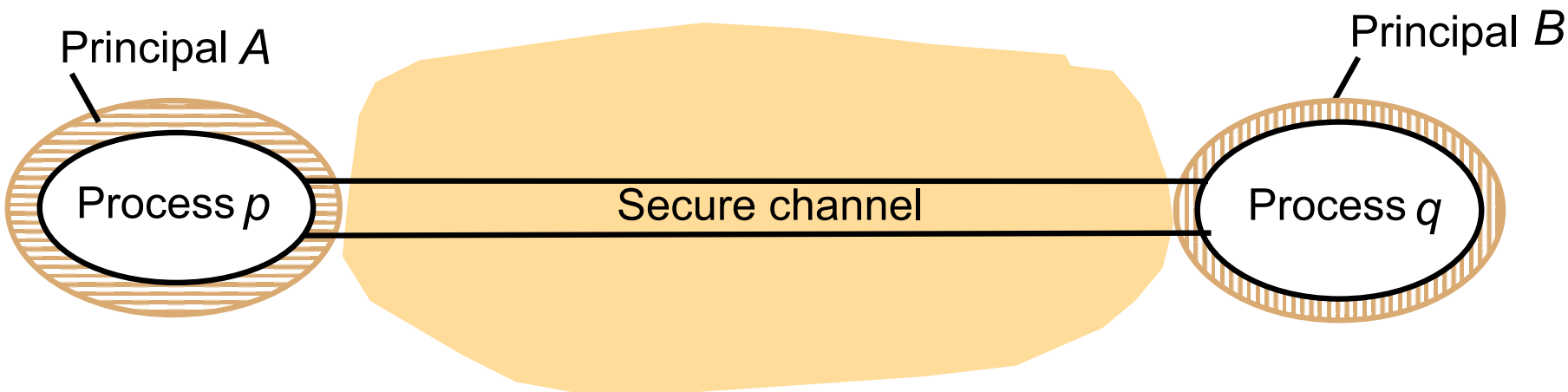
Negação de serviço

Um processo intruso captura uma mensagem de solicitação de serviço e retransmite-a

inúmeras vezes ao destinatário, fazendo-o executar sistematicamente o mesmo serviço e ultrapassando a sua capacidade de resposta

Como lidar com estas ameaças?

- utilização de canais seguros




Definição de canal seguro:

Canal utilizado para comunicação entre dois processos com as seguintes características:

- Cada processo pode identificar com 100% de confiança a entidade responsável pela execução do outro processo

- As mensagens que são transferidas de um processo para outro são garantidas do ponto de vista da integridade e da privacidade

- As mensagens têm garantia de não repetibilidade ou reenvio por ordem distinta

(cada mensagem inclui um tempo físico ou lógico)

## 5 - Modelos Fundamentais

Criptografia:

´Técnica de codificar o conteúdo de uma mensagem de forma a “esconder” o seu conteúdo.

É necessário que ambos os processos possuam a chave de codificação /descodificação

*jtup f tfhsfep*

Autenticação:

Incluir na mensagem uma porção (encriptada) que contenha informação suficiente para identificar a entidade e verificar os seus direitos de acesso

Criar uma modelo de segurança

- analisar as principais ameaças – riscos envolvidos /possíveis consequências

- fazer o balanço entre o custo de proteger o sistema e o perigo que de facto as ameaças representam