

# Sistemas Operativos II

---

## Serviços de Nomes e de Diretoria

# Introdução

---

em SD, os nomes para referir uma variedade de recursos

- computadores, serviços, objetos remotos, utilizadores
- Exemplos
  - url: necessário para aceder a um documento web
  - hostname: necessário para identificar uma máquina

## Serviços de Nomes (*name services*)

- fornecem informação sobre objetos a partir do seu nome, num SD
- relacionados com os serviços de diretorias e *discovery services*
  - Mas estes são mais genéricos e fornecem informação sobre objetos que satisfazem uma determinada descrição

# Conceitos

---

## Nomes legíveis por humanos

- `/etc/passwd` ou `http://www.uevora.pt`

## Identificador

- nome interpretado por programas, escolhido a pensar na eficiência de procura e armazenamento

## Nomes puros e não-puros [Needham]

- os nomes não-puros contêm informação sobre o objeto que designam, por exemplo sobre a localização do objeto

## Endereço

- valor que identifica a localização de um objeto (não o objeto)
- não é apropriado para identificação

# Conceitos

---

## *Resolver um nome*

- obter a informação necessária para efetuar uma ação sobre o objeto com esse nome

## *Binding*

- associação entre nome e objeto
- em geral, existem associações (*binds*) entre nome e atributos do objeto
  - atributo: uma propriedade do objeto, que tem um valor
- exemplos de bindings:
  - DNS: mapeia nomes em atributos de computadores (IP, ...)
  - X500: mapeia nome da pessoa num conjunto de atributos (e-mail, telefone...)
  - CORBA Naming Service, RMIRegistry
    - mapeiam o nome do objeto na sua referência remota

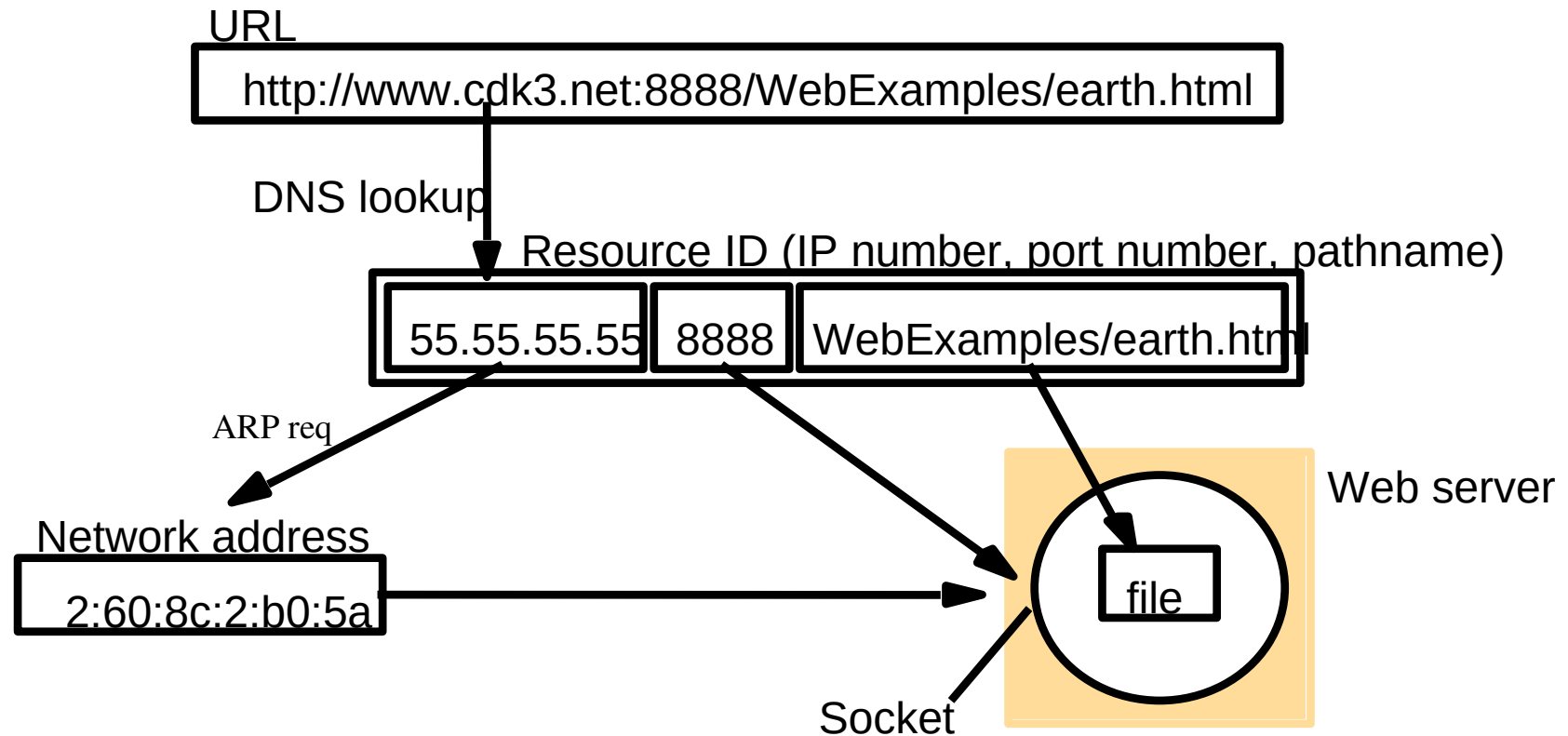
## Endereços

- nomes que também são resolvidos (ARP requests) para *MAC address*

# Nomes Compostos

nomes compostos para aceder a um recurso

- exemplo: URL



# Nomes Compostos

---

## URI – Uniform Resource Identifier

- URL – UR Locator
  - esquema (http, mail, ftp), hostname e pathname até ao documento
  - problema *dangling links*: se o objeto é movido podem ficar apontadores inválidos
- URN – UR Name
  - identificador persistente, mesmo que o recurso mude de localização
  - regista-se o nome (URN), incluindo atributos do documento: o seu URL...
  - se o URL mudar basta atualizar o URL registado para aquele URN
  - a pesquisa faz-se pelo URN e resulta sempre no URL correto
  - formato: `urn:nameSpace:nameSpace-specificName`
  - exemplo: `urn:ISBN:0-201-62433-8`

# Tradução de nomes urn e doi

---

- Serviço de resolução de nomes (serviço de nomes)
- Gestão de identificadores únicos e persistentes para documentos digitais
  - Internet Assigned Numbers Authority
    - [www.iana.org](http://www.iana.org)
  - [www.doi.org](http://www.doi.org)
  - [www.handle.net](http://www.handle.net)

# Serviços de Nomes

- armazenam um ou mais contextos de nomes
- **contexto**: conjunto de associações (*bindings*) entre nomes textuais e atributos dos objetos
  - objetos: utilizadores, computadores, serviços, objetos remotos...
- Operações suportadas por um serviço de nomes:
  - **resolver** um nome – pesquisar atributos para um nome
  - **adicionar, listar e apagar** *bindings* ou contextos
- Motivações para separar a gestão de nomes de outros serviços
  - uniformização
    - usar o mesmo esquema de nomes em serviços diferentes
  - integração
    - facilidade de integração de serviços
  - abertura



# Serviços de Nomes

---

Inicialmente bastava mapear nomes em endereços, para uma única rede LAN ou WAN. A ligação entre as redes aumentou a escala dos SDs e agravou o problema da gestão de nomes.

## Requisitos gerais para um Serviço de Nomes

- nº arbitrário de nomes e servir um nº arbitrário de organizações
- tempo de vida largo
- alta disponibilidade
- isolamento de falhas, tolerância a falhas
  - uma falha localizada não devem levar a uma falha geral
- tolerância de suspeita
  - num SD, aberto, é aceitável que nem todos os clientes confiem num determinado componente do sistema (tomarão medidas para confirmar resultados...)

# Serviços de Nomes

---

## Exemplos:

- Global Name Service
- Internet Domain Name System (DNS)
  - menos ambicioso, não foi pensado especialmente para um grande nº de objetos
  - continua a ser amplamente usado
  - tira partido da replicação e cache (ver a seguir)

# Serviços de Nomes – ainda conceitos

## espaço de nomes (*name space*)

- conjunto de todos os nomes válidos para determinado serviço e que obedecem a regras sintáticas específicas
- **espaço de nomes hierárquico**: os nomes têm uma estrutura que reflete a sua posição na hierarquia do espaço
  - ex: /etc/passwd

## Hierarquia (vantagem)

- permite o mesmo “nome relativo” em zonas diferentes
- diferentes contextos geridos por diferentes entidades
- o espaço de nomes hierárquico pode ser potencialmente infinito

## Espaço de nomes DNS: estrutura hierárquica

### Nomes DNS (*domain name*)

- os nomes são compostos por labels separadas por “.”
- exemplo: `alunos.di.uevora.pt`
- os servidores DNS não trabalham com prefixos mas sim com o nome completo

# Serviços de Nomes – ainda conceitos

---

## Aliases

- nomes alternativos, em geral mais convenientes. Permitem transparência.
- ex: mail.di.uevora.pt -> host.di.uevora.pt

## Domínio (naming domain) – não confundir com domain name

- espaço de nomes para o qual existe uma autoridade administrativa autónoma que gere os nomes lá existentes
- o nome do domínio é usualmente o sufixo comum aos nomes que contem
- ex: uevora.pt
- a administração de domínios pode ser delegada em parte para sub-domínios
  - servidores de nomes próprios
  - autoridade administrativa própria

# Serviços de Nomes

um serviço de nomes, como o DNS, oferece um espaço de nomes global, onde um nome específico refere o mesmo objeto, independentemente do cliente que faz a pesquisa

## Resolução de Nomes

- processo em que se apresenta um nome a um contexto, que tem um *binding* direto desse nome num conjunto de atributos, ou efetua a procura de um *binding* num outro contexto
- com os *aliases* pode haver ciclos
  - detetar e abandonar a resolução

## Alta Disponibilidade

- partição e replicação dos dados
- Vantagem do uso de partição de dados ou uso de replicação:
  - não é necessário ter toda a informação num só servidor
  - evitar afunilamentos (*bottlenecks*)
- a partição implica que um servidor pode não ter os dados suficientes para resolver um nome sem a ajuda de outros servidores

# Serviços de Nomes

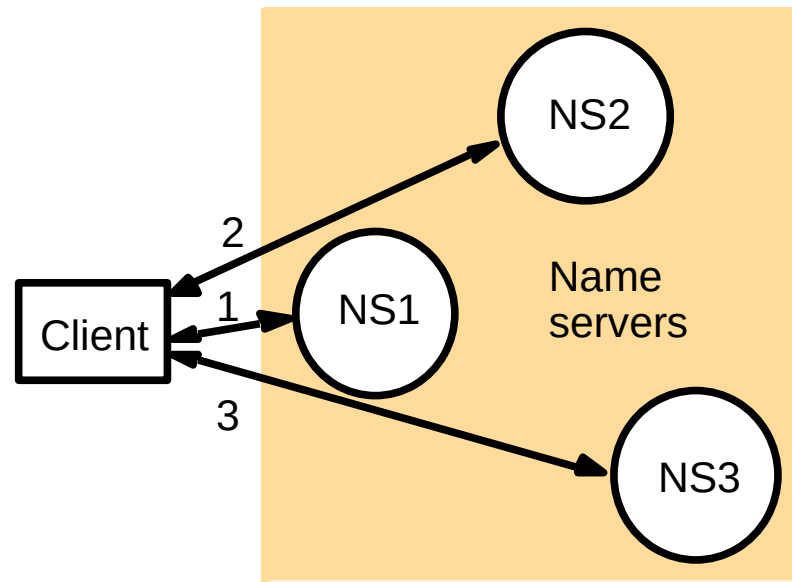
**Navegação:** pesquisa por vários servidores para resolver um nome

- pode ser
  - 1) controlada pelo cliente
    - a) iterativa
      - se um servidor não tem a resposta, o cliente liga-se ao seguinte
    - b) multicast
      - o cliente faz um multicast com o nome a resolver para um grupo de servidores. O servidor que tem a informação responde-lhe em unicast.
  - 2) controlada pelo servidor (o cliente liga-se a um só servidor)
    - a) não recursiva
      - de modo iterativo ou multicast (entre os servidores)
    - b) recursiva

# Navegação controlada pelo cliente

## pesquisa iterativa

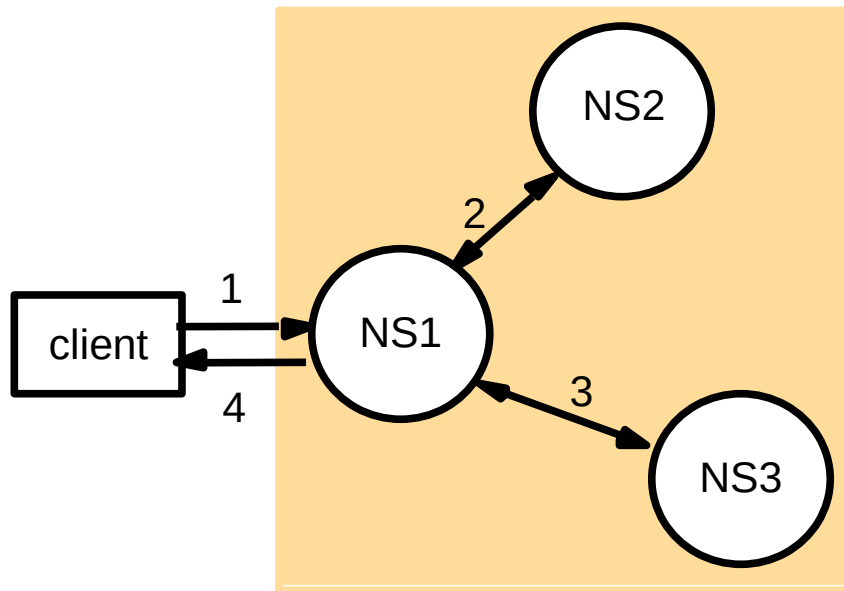
- o cliente contacta sucessivamente cada servidor até resolver o nome
- um servidor pode sugerir o nome de outro servidor



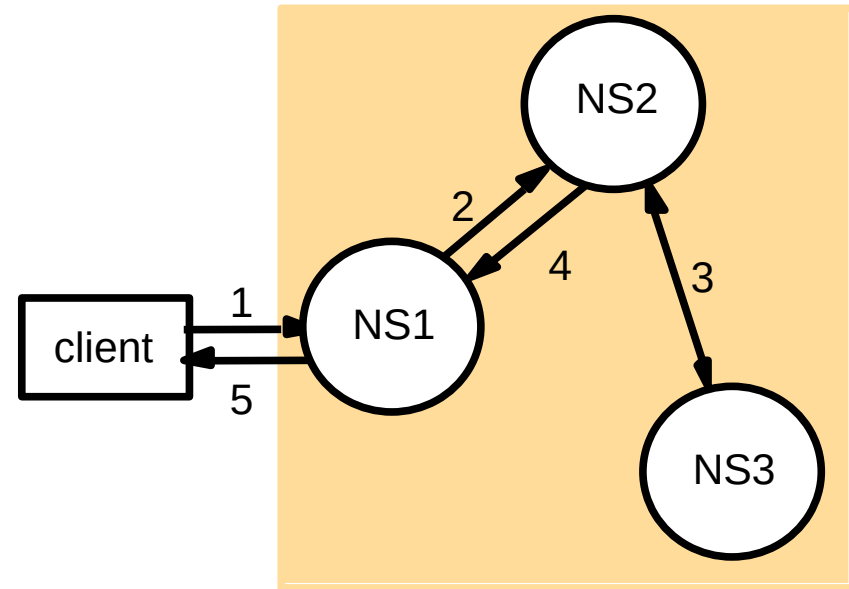
A client iteratively contacts name servers NS1–NS3 in order to resolve a name

# Navegação controlada pelo servidor

## Pesquisa Não Recursiva e Recursiva



Non-recursive  
server-controlled



Recursive  
server-controlled

A name server NS1 communicates with other name servers on behalf of a client



# Serviços de Nomes e Cache

---

DNS: clientes e servidores costumam ter uma cache

ajuda a manter uma boa performance, de ambos os lados

Constata-se:

- é necessário resolver o mesmo nome várias vezes (ex: web)
- os *bindings* (nome, atributos) não mudam muito depois de criados
- a maior parte das operações junto aos servidores são de leitura

# Internet Domain Name System (DNS)

serviço de nomes amplamente usado na internet e intranets

O esquema de nomes inicial para a Internet tinha os problemas:

- não era escalável
- organizações queriam administrar diretamente os seus nomes
- não era um serviço de nomes genérico (destinava-se apenas a endereços)

DNS supera esses problemas

- permite registrar atributos de objetos em geral (mais usado para computadores e endereços)
- as organizações podem gerir os seus nomes (há autoridades)
- qualquer nome pode ser resolvido por qualquer cliente em qualquer localização
- Boa performance, resolve os problemas de escala com:
  - partição hierárquica da base de nomes, replicação e cache

# Internet Domain Name System (DNS)

## divisão/partição do espaço de nomes da internet em domínios:

- por tipo de organização (nos US)
  - com: organizações comerciais; edu: universidades; gov: entidades governamentais; mil: militares; net: relacionadas com redes; org: outras organizações; int: “internacionais”
- geograficamente: um domínio para cada país e para o qual existe uma autoridade administrativa própria
  - us
  - uk
  - fr
  - es
  - pt
  - br
  - ...
  - União Europeia: .eu

# Internet Domain Name System (DNS)

## Queries DNS

- procurar atributos dado um nome

## Tipos de Query DNS

**mais usadas**

### 1. hostname resolution

- hostname -> IP

### 2. mail host location

- nome -> endereço IP do servidor de mail respectivo

### 3. reverse resolution

- endereço IP -> nome

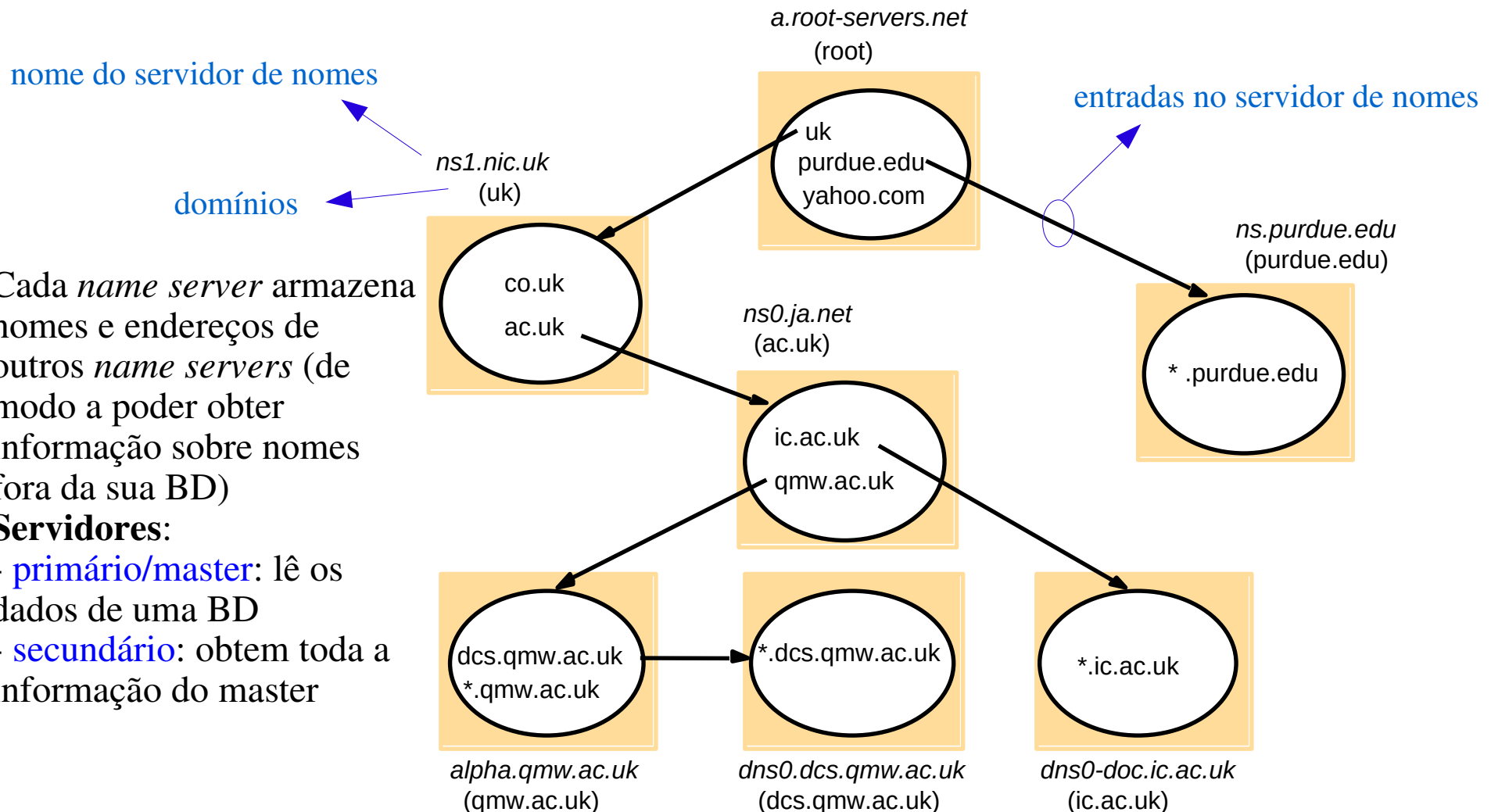
### 4. host information

- informações sobre arquitetura e SO de uma máquina

### 5. Well-known services

- lista de serviços disponibilizados por uma máquina: ftp, telnet, ssh, http

# Servidores de Nomes DNS



# Servidores de Nomes DNS

## Registo de um Recurso num servidor DNS

campos		
<i>Record type</i>	<i>Meaning</i>	<i>Main contents</i>
A	A computer address	IP number
NS	An authoritative name server	Domain name for server
CNAME	The canonical name for an alias	Domain name for alias
SOA	Marks the start of data for a zone	Parameters governing the zone
WKS	A well-known service description	List of service names and protocols
PTR	Domain name pointer (reverse lookups)	Domain name
HINFO	Host information	Machine architecture and operating system
MX	Mail exchange	List of <preference, host> pairs
TXT	Text string	Arbitrary text – <b>permite qualquer atributo</b>

# Internet Domain Name System (DNS)

---

## Outra Utilização DNS: Repartir a carga por vários servidores

- ter um nome partilhado por vários computadores
- um registo para cada computador
- o DNS server resolve o nome de acordo com o esquema *round robin*

## Protocolo DNS

- protocolo RR, UDP
- navegação iterativa ou recursiva

# Serviços de Diretoria e *Discovery Services*

nos serviços de nomes existem pares (nome, atributos) e a pesquisa efetua-se pelo nome... mas em determinados cenários pode interessar a pesquisa pelos atributos!

## Serviço de Diretoria

- guarda associações (*bindings*) entre nomes e atributos e permite pesquisa de entradas baseada nos atributos
  - o nome é tratado como mais um atributo
- Exemplos: *MS Active Directory Services*, X.500, LDAP

## Analogia

- serviço de nomes: “páginas brancas” (lista telefónica)
- serviço de diretorias: páginas amarelas



# Serviços de Diretoria e *Discovery Services*

---

## *Discovery Services*

- serviços de diretoria que guardam informação sobre os serviços disponibilizados num ambiente de *spontaneous networking*
- Para Quê
  - fornecem uma interface para o registo automático de serviços e uma interface para clientes procurarem serviços disponíveis (impressora...)

## Registo de um serviço: impressora

- conjunto de atributos:
  - resourceClass=printer, type=laser, colour=yes, ..., url=http://printer.di.uevora.pt/services/printer87

# Jini

---

*discovery Service* para *spontaneous networking*, baseado em Java  
(comunicações por RMI)

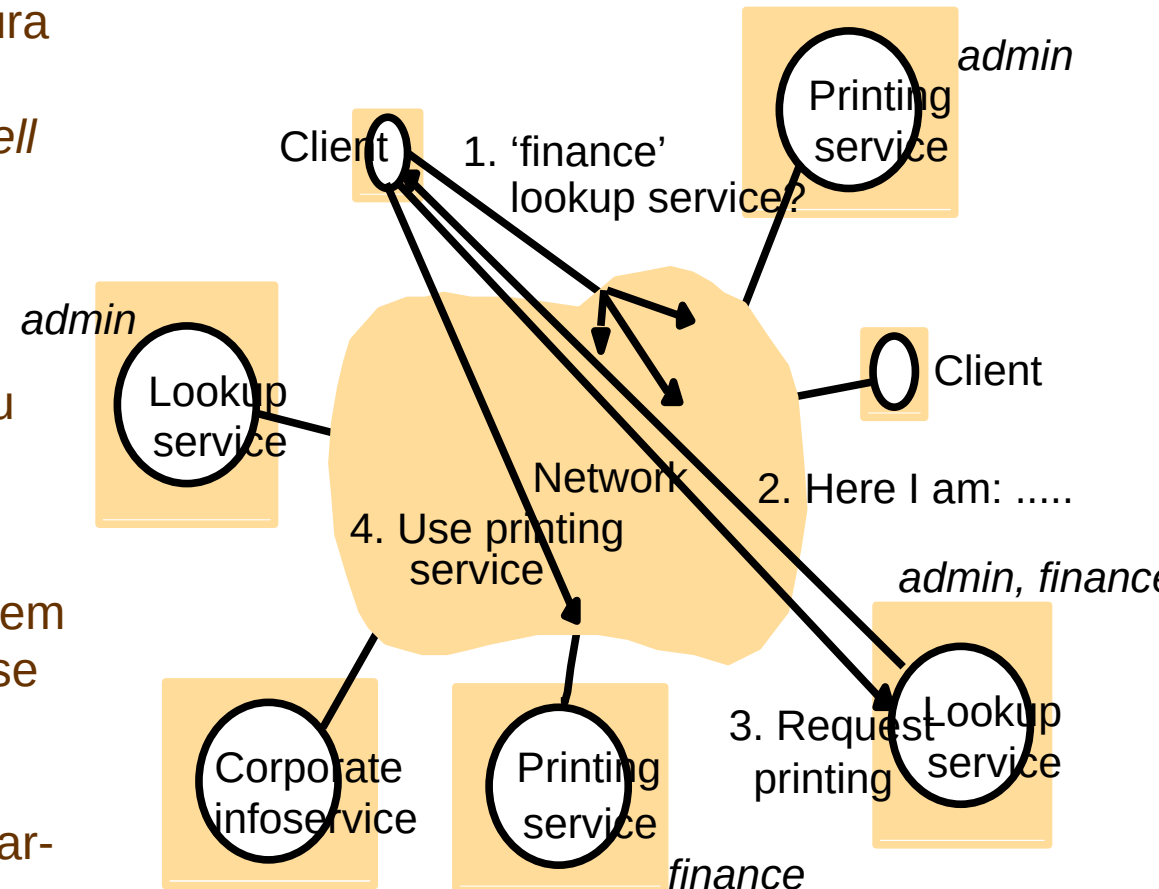
## Funcionalidades

- **service discovery** (sem configuração manual ou negociação prévia)
- transações entre componentes do ambiente *spontaneous networking*
- espaços de dados comuns

# Jini Service Discovery

1. ao entrar no ambiente, um cliente procura o *Lookup Service*. Para isso faz um multicast (com time-to-live) para um *well known* endereço IP.
2. Um *Lookup Service* deteta o pedido e anuncia-se respondendo em unicast
3. o cliente pode agora fazer um lookup ou registrar um serviço no servidor

Os *Lookup Services* para além de escutarem por pedidos podem também anunciar-se automaticamente efetuando multicast para o mesmo endereço. Os clientes escutam o anúncio e podem depois ligar-se diretamente por unicast.



# Serviço de Diretorias X.500

pode ser usado como um serviço de nomes, mas é pensado para satisfazer *queries* descritivas (relativas a vários atributos)

serve de base para o LDAP

árvore de nomes: *Directory Information Tree* (DIT)

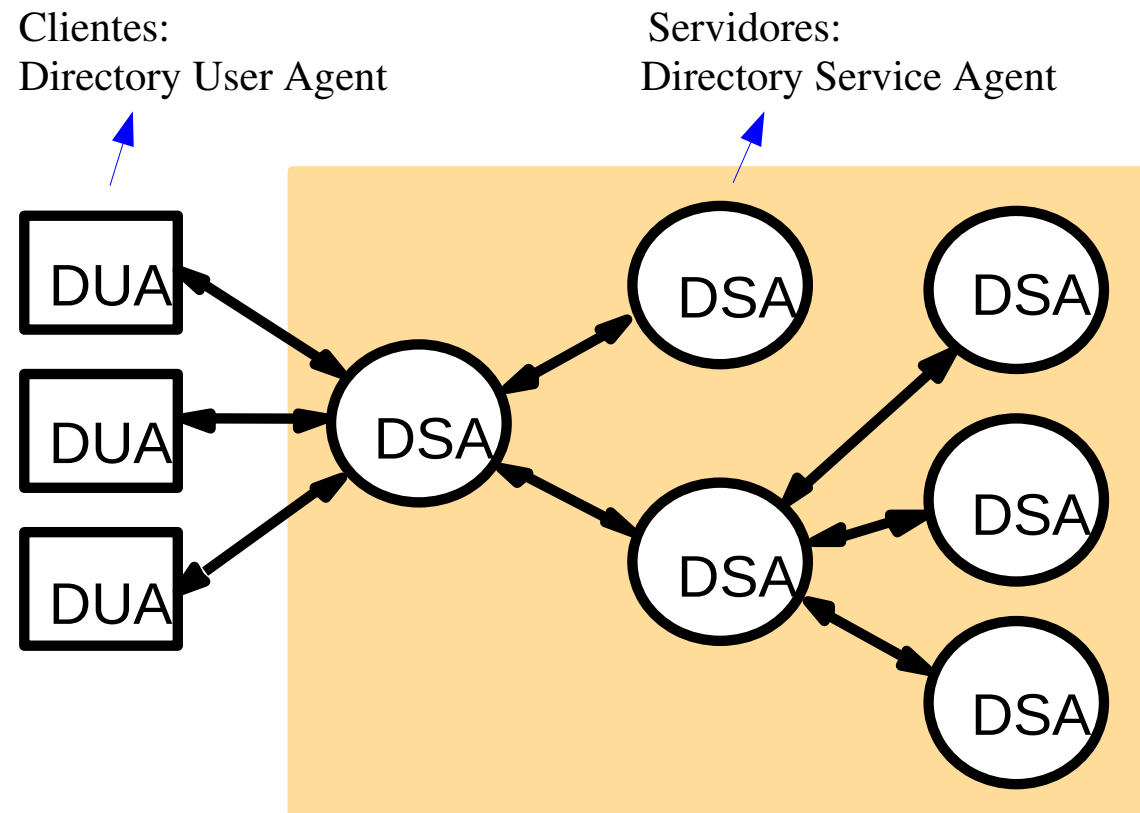
estrutura de diretorias, incluindo os dados associados a cada nó:

- *Directory Information Base* (DIB)

Pretende-se uma única DIB para integrar a informação de todas as organizações no mundo, dividida por muitos servidores X.500 (em geral um por organização)

# Arquitetura X.500

se a informação necessária não está no segmento de DIB mantido pelo servidor, este invoca outros servidores ou redireciona o cliente para outro servidor



# LDAP

---

a interface para o serviço X.500 utiliza as camadas superiores do modelo OSI.  
Existe uma abordagem mais leve que permite aos clientes aceder ao serviço diretamente por TCP: **LDAP**

## Lightweight Directory Access Protocol (LDAP)

- baseado no X.500
- pode funcionar também com um servidor não X.500, desde que obedeça à interface para o LDAP
- amplamente usado, permite controlo de acessos mediante autenticação
- Exemplo de serviços LDAP:
  - MS Active Directory Service

# Java e Serviços de nomes e de diretoria

## API para Serviços de Nomes e de Diretorias Java

- JNDI (*Java Naming and Directory Interface*)

<https://docs.oracle.com/javase/tutorial/jndi/index.html>

