

Computação Distribuída

Cap VII – Sistemas de Nomeação

Licenciatura em Engenharia Informática

Universidade Lusófona

Prof. Paulo Guedes (paulo.guedes@ulusofona.pt)



Sistemas de Nomeação

- ▶ A identificação de recursos num ambiente distribuído é essencial para permitir a sua localização
 - Ficheiros
 - Serviços
 - Objectos
- ▶ É necessário um modo de designação
 - Nome: independente do sistema (geralmente em formato textual)
 - Identificador: referência interna do sistema (ex: formato binário)
- ▶ A relação entre um nome e um recurso é designada por associação - *name binding*
- ▶ A tradução do seu nome para um identificador é designada por resolução - *name resolution*
- ▶ Um nome pode ser constituído por uma série de campos com domínios de resolução distintos

Visão conceptual de um sistema de nomeação

Nome	Identificador
Nome1	Identificador1
Nome2	Identificador2
Nome3	Identificador3
...	...
NomeN	IdentificadorN
...	...
...	...
...	...

Operações Bind e Lookup

Bind (nomeX, IdentificadorX)

Insere o par (nomeX, IdentificadorX) na tabela

Nome	Identificador
Nome1	Identificador1
Nome2	Identificador2
Nome3	Identificador3
...	...
NomeN	IdentificadorN
...	...
NomeX	IdentificadorX
...	...

Id = lookup (nomeX)

Pesquisa o nomeX e devolve o IdentificadorX que lhe está associado

Tipos de Nomes

- ▶ Um nome está associado a um espaço de designação
- ▶ Nomes textuais
 - São nomes simbólicos orientados para o utilizador, facilmente memorizáveis
 - Exemplo: <http://netlab.ulusofona.pt/cd/teoricas/index.php>
- ▶ Nomes puros: o nome não contém qualquer indicação da localização do objeto
 - Não ajudam na localização do objeto
 - Não têm que ser alterados se o objeto mudar de localização
- ▶ Nomes impuros: o nome contém informação sobre a localização do objeto
 - Facilitam a localização do objeto dado o seu nome
 - Têm que ser alterados se o objeto mudar de localização

Espaços de Nomeação

- ▶ Um *espaço de nomeação* - *name space* é o conjunto de todos os nomes válidos, reconhecidos e resolúveis por um determinado serviço
 - Num espaço de nomeação existe uma *única* autoridade administrativa (possivelmente replicada) que atribui e resolve nomes
 - 6.8.1.6.6.9.8.1.2.1.5.3.e164.arpa corresponde ao nº telefone +351 21 8916186 no ENUM - Electronic Number Mapping System
 - /home/alunos/a1234567/trabalho4 é um nome no espaço de nomeação Unix
 - Um espaço de nomeação tem uma sintaxe específica
- ▶ Um espaço de nomeação pode ser
 - Organizado hierarquicamente (DNS, UFS, XMLNS, X-500)
 - Plano ou linear (ex: sistema de nomeação humano)
- ▶ Um serviço de nomeação implementa a funcionalidade de resolução de nomes dentro de um determinado espaço de nomeação

Existem diferentes domínios de validade dos nomes, com regras diferentes

Nome	Identificador
João A.	53301024
Maria B.	63120203
António C.	72123409
...	...
Marta X.	53321234
...	...

Nome	Identificador
NomeA	IdentificadorA
NomeB	IdentificadorB
NomeC	IdentificadorC
...	...
NomeX	IdentificadorX
...	...

Nome	Identificador
www.lusofona.pt	193.137.75.244
www.google.com	216.58.211.228
www.Facebook.com	69.171.250.35
...	...
<u>www.myserv.pt</u>	70.71.72.73
...	...

Identificadores

► Identificadores únicos

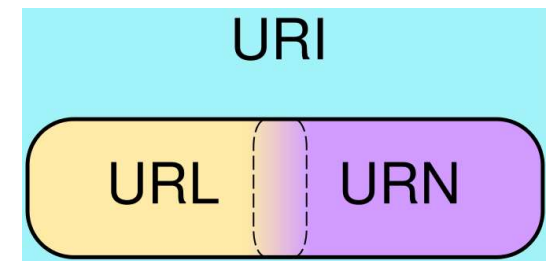
- **UIDs: Unique Identifiers** usados internamente pelos sistemas
 - FF.33.2B.45.8A.5F.2C.35.28.5E.29 designa um serviço no contexto de um servidor Java a correr numa JVM
 - Não faz sentido no contexto de um outro sistema

► Endereço

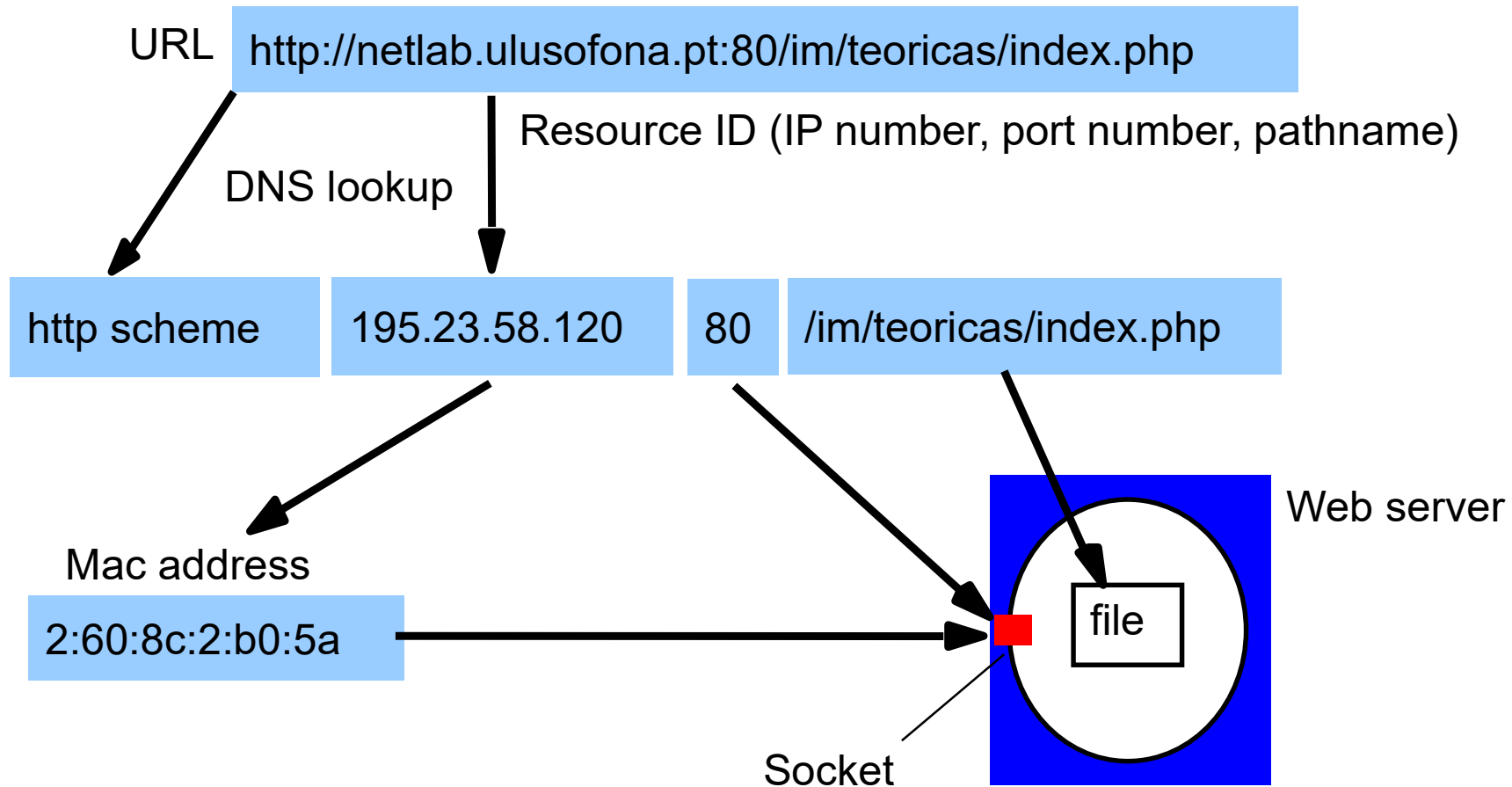
- **Forma particular de nomeação** permitindo acesso directo ao recurso
 - 192.168.15.10 no contexto da rede interna do Laboratório
 - 21 757 70 06 designa um cliente da rede PSTN de Lisboa no contexto de numeração nacional
 - 0x0a005c8e endereço memória no contexto de um processo

URIs: *Uniform Resource Identifiers*

- ▶ Sistema de designação que permite identificar recursos na Internet
 - Uniforme porque incorpora uma série de sistemas de designação
 - URI *scheme* - scheme:scheme-specific-name
 - Para cada *scheme* existem procedimentos específicos para a resolução do seu espaço de nomes
 - <http://www.service.cc>
 - <ftp://server.school.edu>
 - <rmi://somehost/rmi-service>
- ▶ URL: *Uniform Resource Locator*
 - É um URI específico para localização de recursos
 - scheme:scheme-specific-location
 - Ex: <http://dns-name/filesystem-name>
 - Está associado à localização do recurso
 - Se este se move ou é removido o URL perde o significado
- ▶ URN: *Uniform Resource Name*
 - É um URI de nomeação “puro” no qual o nome não tem associada qualquer localização
 - urn:nameSpace:nameSpace-specificName
 - Ex: urn:ISBN:0-201-64233-8 sistema de nomeação de livros
 - Para aceder a um URN é necessário convertê-lo num URL

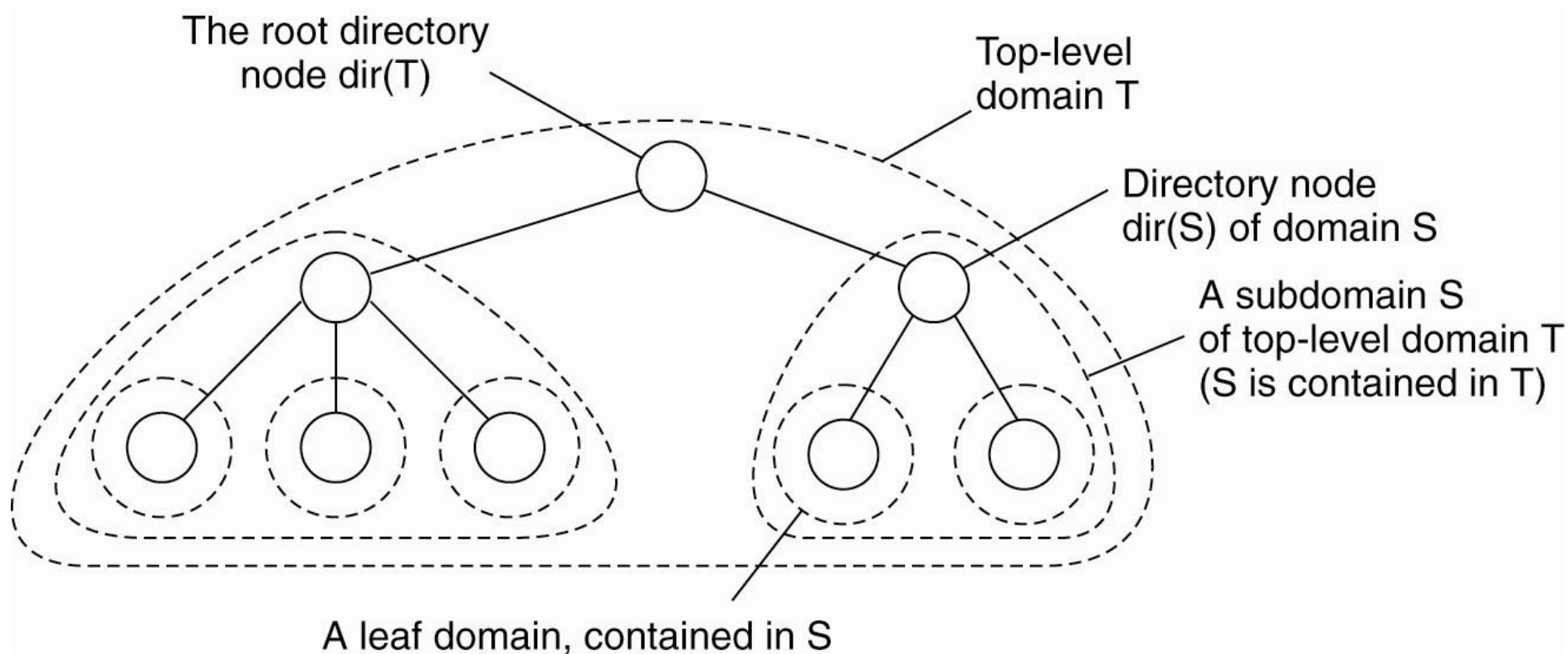


Exemplo de Resolução de URL

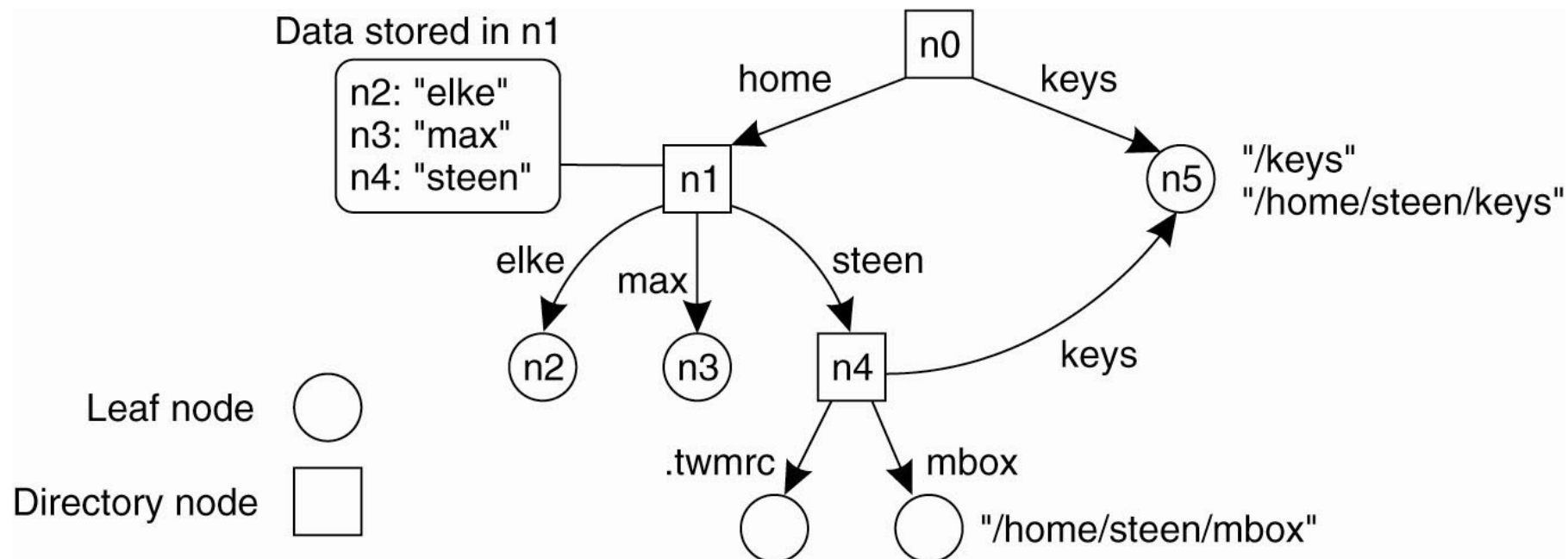


- ▶ Um URL agrega vários domínios de resolução distintos
- ▶ Para resolver um URL, é necessário invocar vários sistemas de resolução de nomes

Espaço de Nomeação Hierárquico



Aliasing



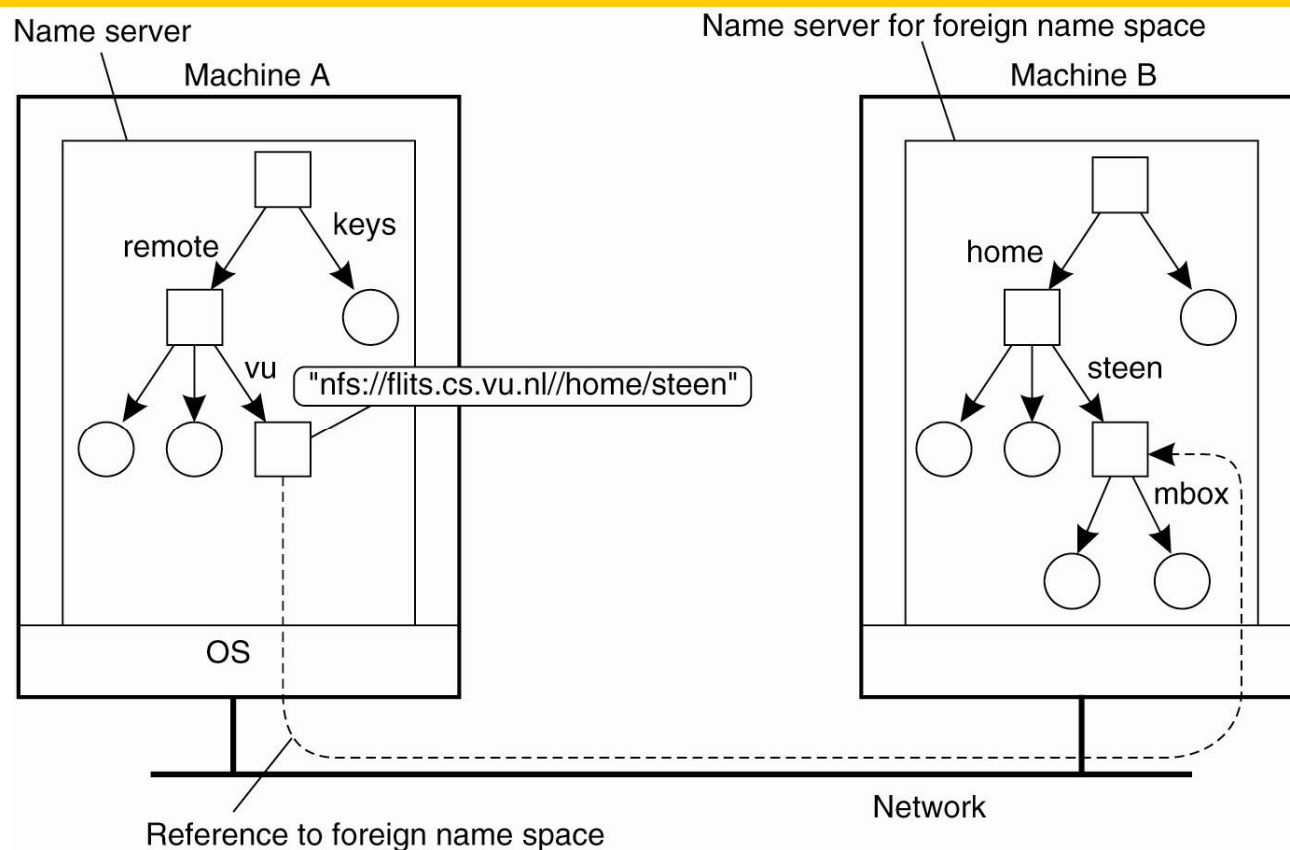
► O nó n5 pode ser designado por:

- /keys
- /home/steen/keys

Combinação de Espaços de Nomeação

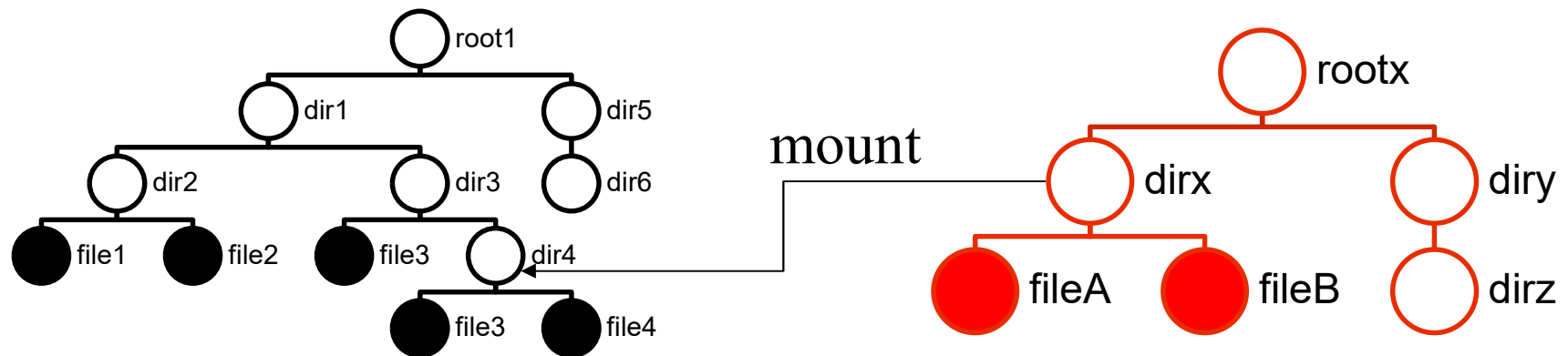
- ▶ Para a resolução de um nome global torna-se necessário combinar vários espaços de nomeação
 - Cada serviço de nomeação resolve a respectiva parte do nome
- ▶ Concatenação
 - Permite concatenar espaços de nomeação locais e remotos
 - Ex: *montagem* de sistemas de ficheiros NFS em Unix
 - `mount srv1:/users /home/users`
 - `/home/users/aluno/documentos/aulas`
 - A resolução do nome *atravessa* o ponto de montagem
- ▶ Customização
 - Permite adaptar o espaço de nomeação às exigências das aplicações ou utilizadores
 - Associações entre elementos de espaços de nomeação distintos
 - *aliases* de nomes DNS
 - *Symbolic links* Unix

Concatenação NFS

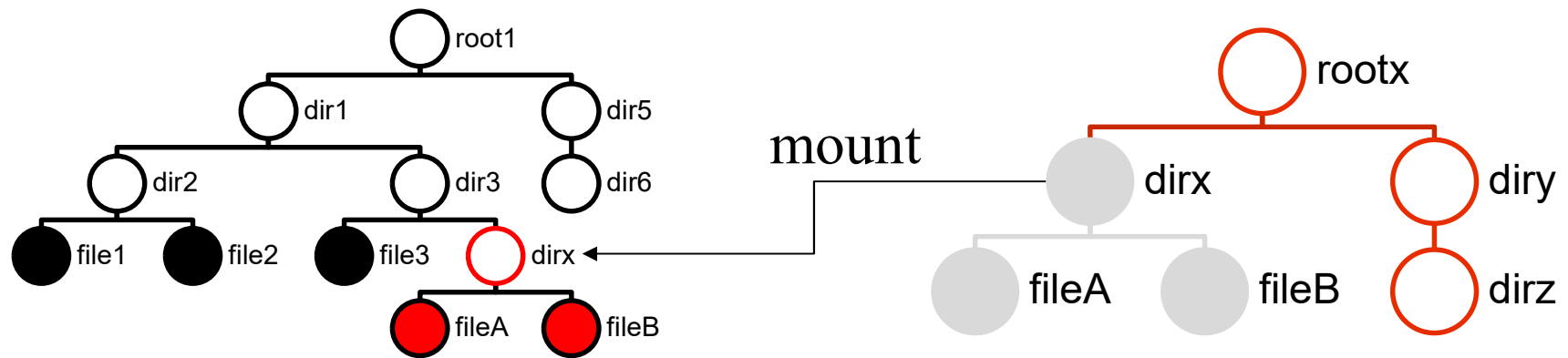


- Informação necessária para *montar* um espaço de nomeação remoto
 - O nome de um protocolo de acesso
 - O nome do servidor
 - O ponto de concatenação: nome do recurso local e do recurso de substituição

Mount: exemplo



Mount: exemplo



Perguntas a que devo ser capaz de responder

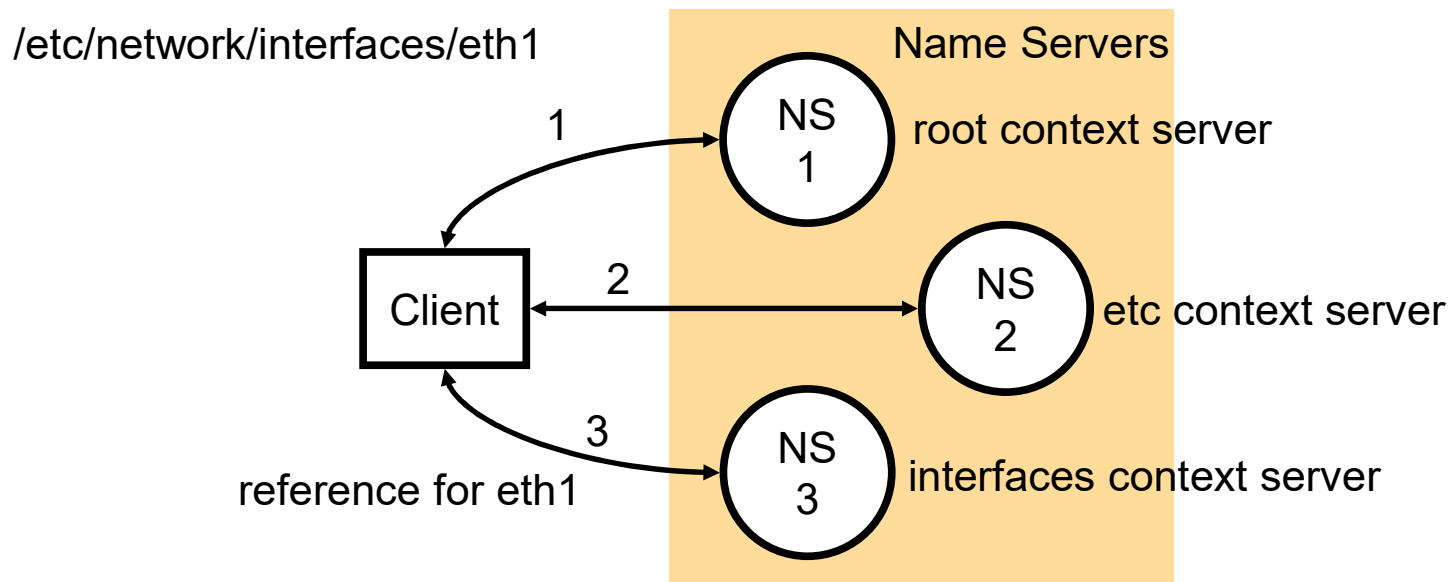
- ▶ O que são nomes e identificadores num serviço de nomes ?
- ▶ O que são as operações Bind e Lookup num serviço de nomes ?
- ▶ O que são nomes puros e impuros ? Os nomes hierárquicos são puros ou impuros ?
- ▶ Quais são os serviços de nomes necessários para resolver um URL ?
- ▶ O que é aliasing ?
- ▶ O que faz a operação de mount em espaços de nomes ?

Serviço de Nomeação

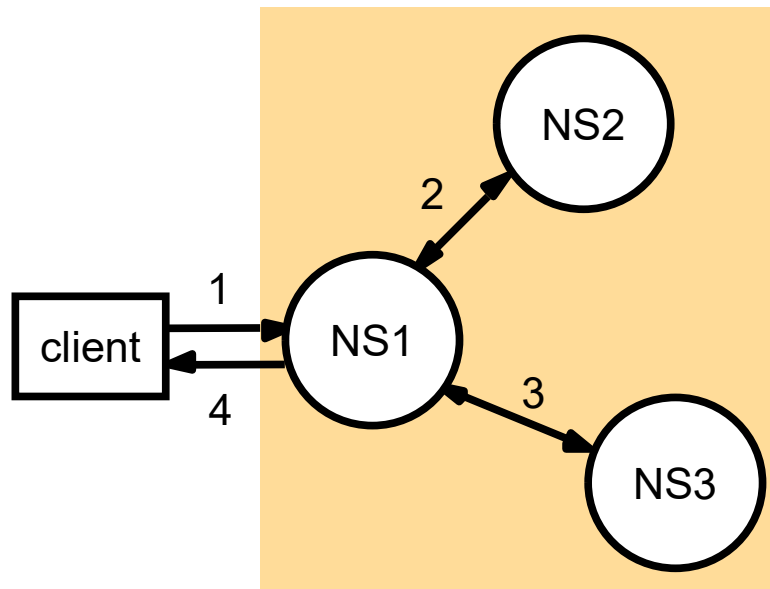
- ▶ Um Serviço de Nomeação permite realizar a resolução de um nome dentro do espaço de nomeação
- ▶ A resolução é sempre realizada num contexto ou directório associado
- ▶ O resultado de uma resolução é geralmente uma referência interna para um recurso, que permite aceder aos seus atributos ou funcionalidades
 - Características de um ficheiro (tamanho, tipo, permissões)
 - Métodos de um objecto
- ▶ As operações associadas à nomeação:
 - **Bind**: cria uma associação entre um nome e uma referência para os atributos do recurso
 - **Lookup**: permite realizar a resolução do nome

Resolução Iterativa

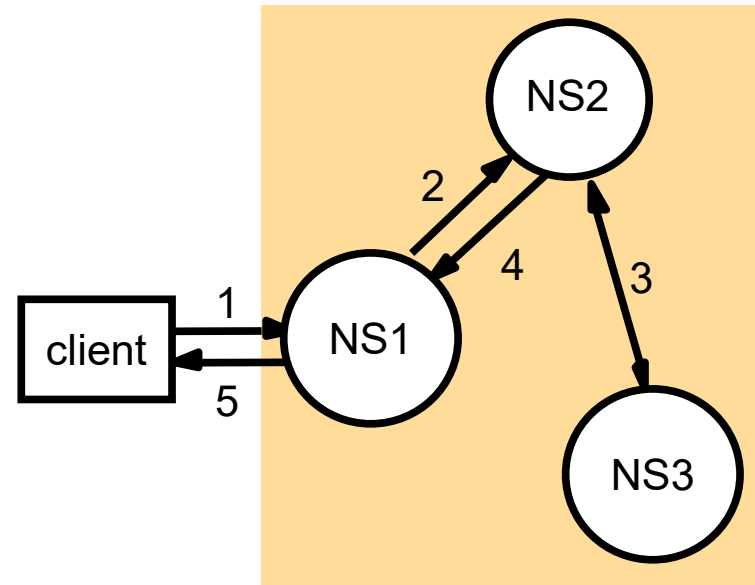
- ▶ A resolução é geralmente um processo iterativo em que cada componente do nome é apresentado ao servidor do contexto de topo (TLD)
 - Se o servidor conhece o nome devolve uma referência para os atributos
 - Se o nome é uma folha ou nó terminal o processo para
 - Se o nome é um nó, o processo continua no contexto do componente seguinte



Resolução Recursiva



Non-recursive server-controlled



Recursive server-controlled

- Se o servidor conhece o nome devolve uma referência para os atributos
 - Se o nome é uma folha ou nó terminal o processo para
- Se o nome é um nó, o processo continua o servidor de topo interage com os outros servidores para resolver o resto do nome
 - Pode interagir de forma iterativa ou recursiva

Caching e Réplica de Serviços de Nomes

- ▶ O serviços de nomeação são essenciais para o acesso a serviços distribuídos
- ▶ Por isso torna-se necessário mascarar as falhas dos servidores de nomes utilizando
 - *Caching* da resolução de nomes nos clientes
 - O *cache* pode ser organizado por contextos ou espaço de nomeação
 - Replicação dos vários servidores de contextos
 - Os servidores podem também ter *caches* de outros contextos
- ▶ O *caching* e a replicação permitem também melhorar as performances da resolução de nomes
 - Diminui latência e tráfego
- ▶ Técnicas de *caching* e replicação implicam a existência de manter a coerência entre as várias cópias da informação
 - Réplicas de tipo primário/secundário com propagação de actualizações
 - Gestão de *cache* baseado em *aging* ou com *call-backs* do servidor para o cliente (ex. AFS)

Serviços de Directório e de Descoberta

- ▶ Um **Serviço de Directório** permite obter uma referência para objectos através da indicação de um subconjunto dos seus atributos
 - O nome pode ser um dos atributos mas pode não ser necessário conhecê-lo
 - Ex. qual é o serviço de objectos que implementa cálculo matricial ?
 - O resultado de uma resolução pode ser um conjunto (vasto) de referências
 - Ex.: X-500, LDAP, UDDI são serviços de directório
- ▶ Um **Serviço de Descoberta** é um serviço de directório que permite o registo e descoberta dinâmica de funcionalidades de dispositivos voláteis num ambiente distribuído
 - A associação a dispositivos é feita baseada na funcionalidade e não no nome
 - A descoberta pode ser feita por *multicast* iniciado pelos clientes ou pelos servidores
 - Ex.: Jini, Bluetooth
- ▶ Serviço de Directório = *yellow pages*
 - Serviço de Nomeação = *white pages*

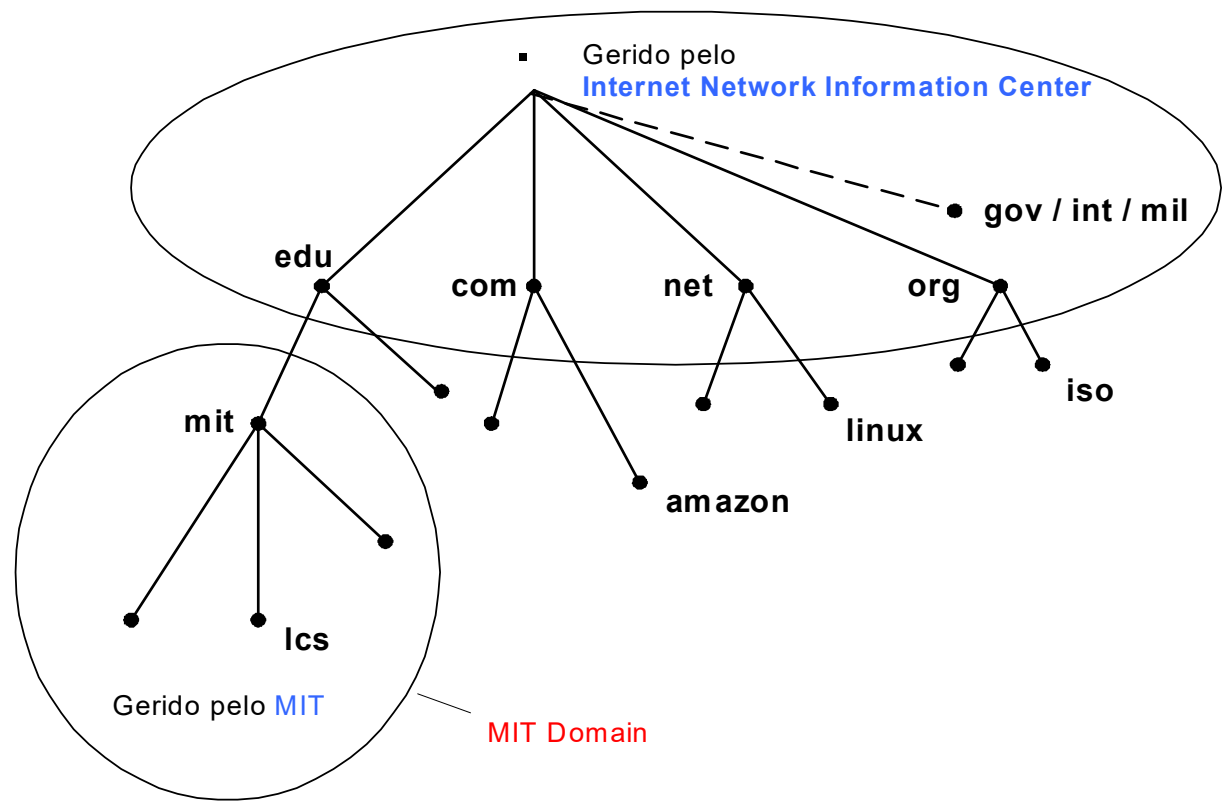
DNS (*Domain Name Service*)

- ▶ Arquitectura para registo e resolução de nomes de máquinas da Internet
 - Inicialmente proposta em 1983
- ▶ Serviço de nomes mais utilizado na Internet

DNS: Características

- ▶ Espaço de nomes global, hierárquico, e homogéneo
 - Nomes impuros
- ▶ Cada contexto designa-se por domínio
- ▶ Cada domínio é uma entidade administrativa:
 - Pode criar e remover nomes
 - Resolve nomes
 - Pode delegar responsabilidades em sub-domínios
- ▶ Tipos de nomes:
 - Global (*Fully Qualified Domain Name*)
Ex. `maqx.ulusofona.pt`
 - Local (resolvido no domínio corrente)
Ex. `maqx`

DNS: Estrutura

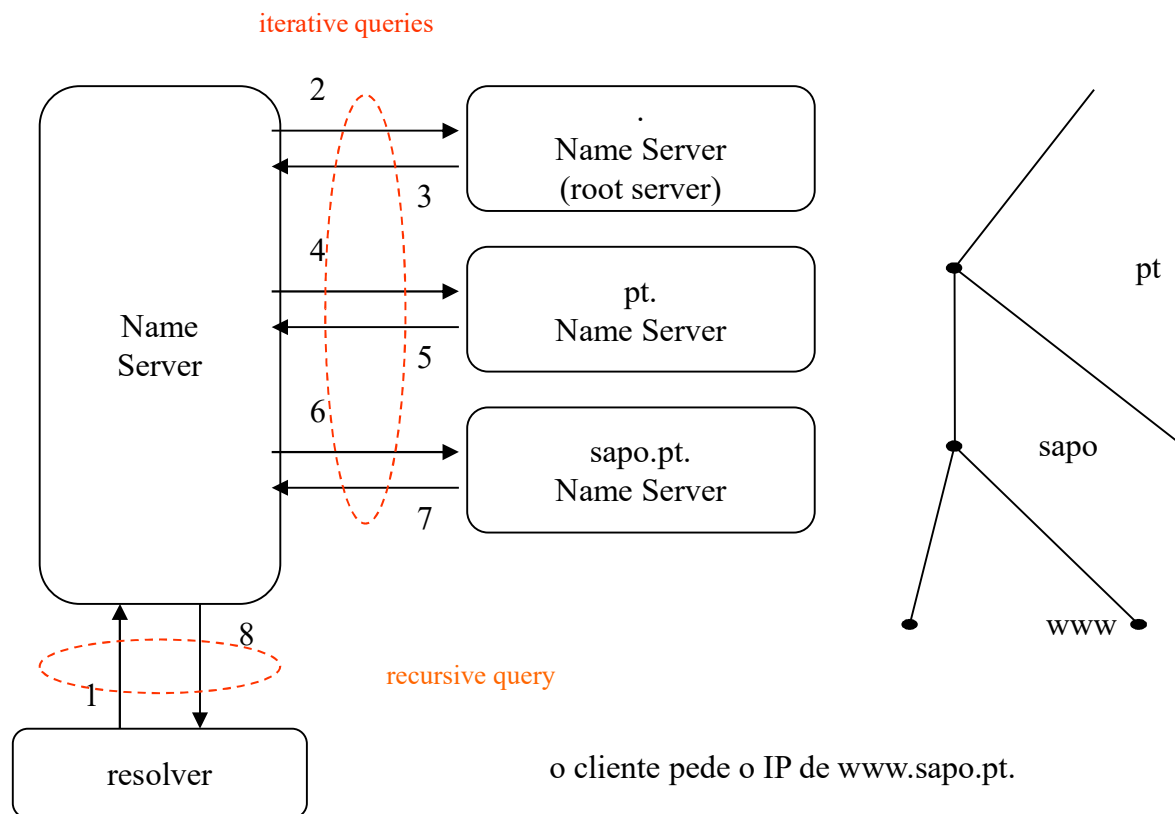


Nome de domínio	Tipo de organização
com	Comercial
edu	Educação
org	Sem fins lucrativos
net	Redes
gov	Governamental (não militar)
mil	Governamental e militar
num	Números de telefone
arpa	Reverse DNS
xx	Código de país (2 letras) ISO 3166

DNS: Zonas

- ▶ Unidades de fraccionamento da hierarquia
- ▶ Uma zona é uma unidade de administração:
 - Cada domínio pertence a uma zona
 - Cada zona pode gerir um ou mais domínios
 - A Zona é que constitui a autoridade.
- ▶ Por cada zona existe um conjunto de servidores:
 - Primário
 - Secundários (com réplicas da BD do primário)
- ▶ Cada servidor indica a sua autoridade sobre os dados que fornece
 - Primário: autoridade total sobre os dados do domínio
 - Secundários: não possuem autoridade alguma

DNS: Resoluções recursivas ou iterativas



DNS: Informação em cada domínio

▶ Registos RR (*Resource Register*)

- Pares nome → valor tipificados
- A tipificação exprime:
 - Classe: família de nomes (ex. IP para endereços IP)
 - Tipo: semântica de utilização do nome
- Cada RR possui um TTL (*time to live*)
 - Serve para invalidar periodicamente RR em *cache*

▶ Informação estrutural (RRs do tipo NS)

- Localização de servidores de zonas

DNS: Tipos de registos

Tipo de registo	Conteúdo
A	Endereço IP
CNAME	Nome simbólico para outro nome DNS
HINFO	Arquitetura e sistema operativo do nó
NS	Servidor de uma zona
MX	Máquina ou domínio do servidor preferencial de e-mail
SOA	Parâmetros que definem a zona
PTR	Nome DNS para resolução inversa de um endereço IP
TXT	Texto arbitrário
WKS	Descrição de um serviço com os respectivos nomes e protocolos

DNS: exemplo de registos

<i>domain name</i>	<i>time to live</i>	<i>class</i>	<i>type</i>	<i>value</i>
<i>dcs.qmul.ac.uk</i>	<i>1D</i>	<i>IN</i>	<i>NS</i>	<i>dns0</i>
<i>dcs.qmul.ac.uk</i>	<i>1D</i>	<i>IN</i>	<i>NS</i>	<i>dns1</i>
<i>dcs.qmul.ac.uk</i>	<i>1D</i>	<i>IN</i>	<i>NS</i>	<i>cancer.ucs.ed.ac.uk</i>
<i>dcs.qmul.ac.uk</i>	<i>1D</i>	<i>IN</i>	<i>MX</i>	<i>1 mail1.qmul.ac.uk</i>
<i>dcs.qmul.ac.uk</i>	<i>1D</i>	<i>IN</i>	<i>MX</i>	<i>2 mail2.qmul.ac.uk</i>

<i>domain name</i>	<i>time to live</i>	<i>class</i>	<i>type</i>	<i>value</i>
<i>www</i>	<i>1D</i>	<i>IN</i>	<i>CNAME</i>	<i>apricot</i>
<i>apricot</i>	<i>1D</i>	<i>IN</i>	<i>A</i>	<i>138.37.88.248</i>

<i>dcs</i>	<i>1D</i>	<i>IN</i>	<i>NS</i>	<i>dns0.dcs</i>
<i>dns0.dcs</i>	<i>1D</i>	<i>IN</i>	<i>A</i>	<i>138.37.88.249</i>
<i>dcs</i>	<i>1D</i>	<i>IN</i>	<i>NS</i>	<i>dns1.dcs</i>
<i>dns1.dcs</i>	<i>1D</i>	<i>IN</i>	<i>A</i>	<i>138.37.94.248</i>

Servidores DNS

- ▶ Associado a uma zona existe sempre um servidor
 - Contém a base de dados com os nomes desse conjunto de domínios
- ▶ Servidor sempre replicado
 - **Primário**: mantém a base de dados, onde se efectuam as actualizações
 - **Secundário**: contém uma cópia da informação do primário, actualizada periodicamente com um protocolo dedicado
- ▶ Todos os servidores mantêm caches
 - Validade indicada pelo parâmetro TTL

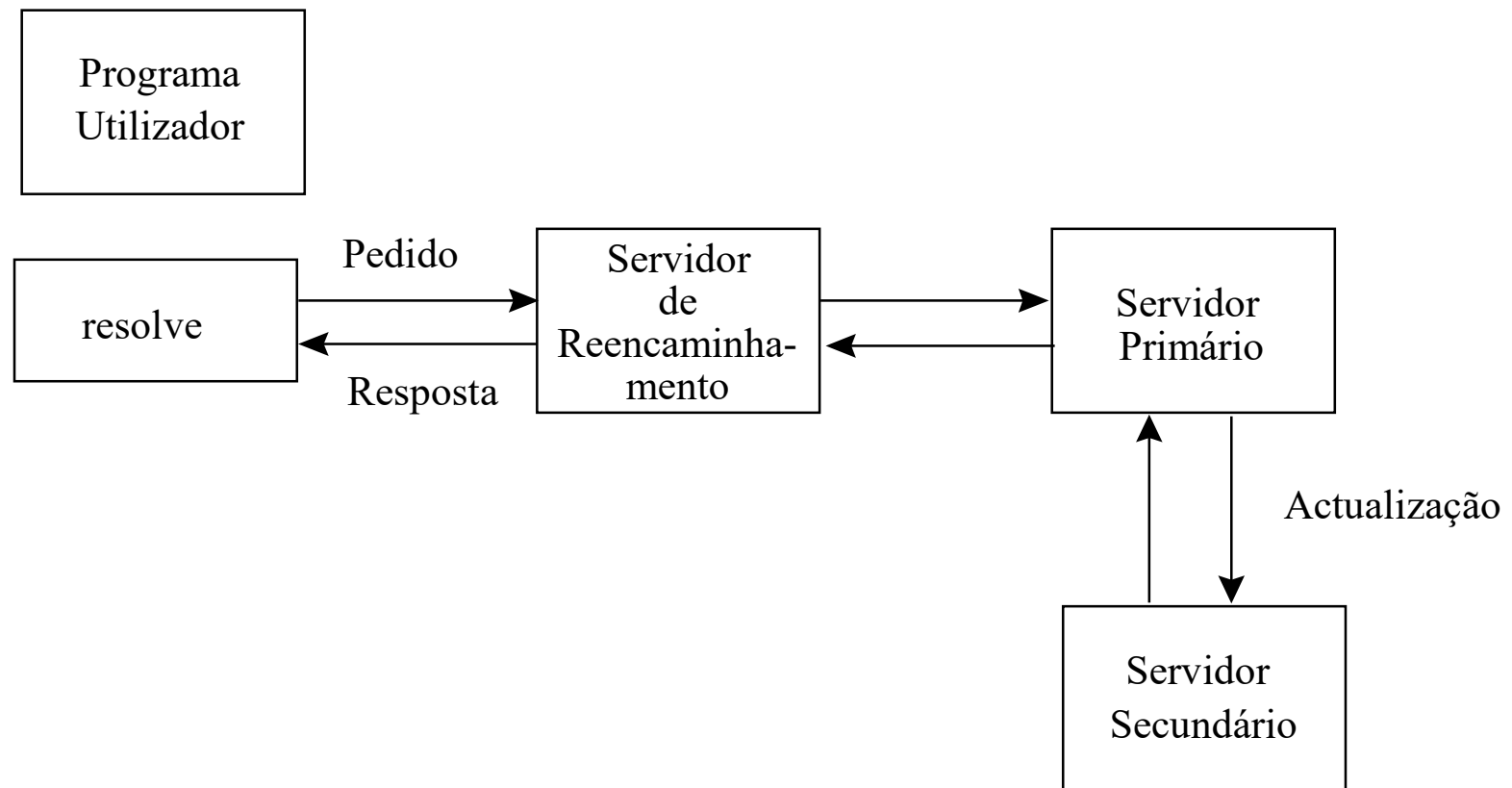
BIND - Berkeley Internet Name Domain

- ▶ Implementação do DNS para Unix
- ▶ Contém 2 componentes:
 - resolver: conjunto de rotinas cliente
 - Integradas na biblioteca de C (`/lib/libc.a`)
 - Usadas pelas rotinas de resolução de nomes (`gethostbyname`, `gethostbyaddr`)
 - named: servidor de nomes

Servidores de Nomes

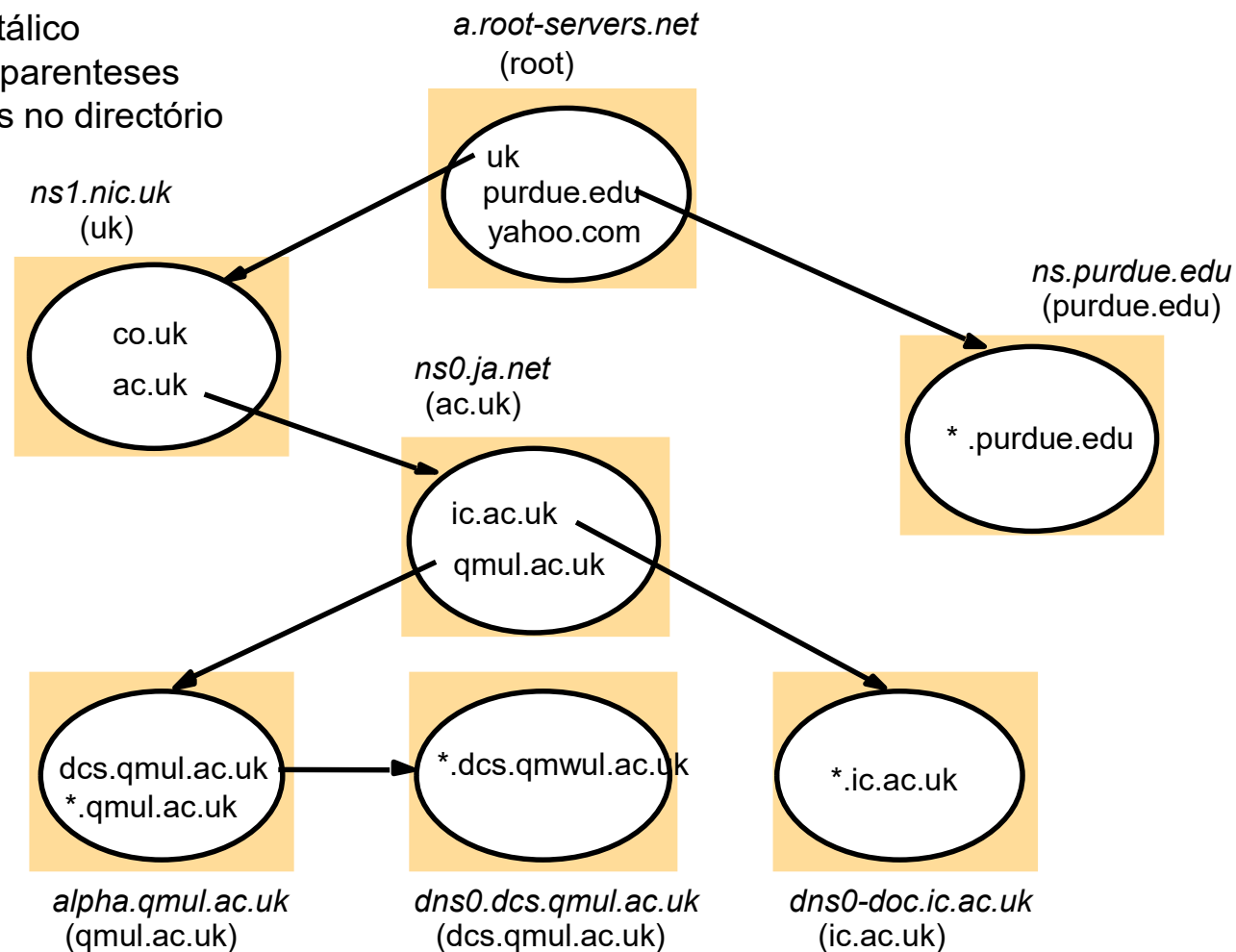
- ▶ **Master: autoridade no domínio**
 - Mantém todos os dados do domínio
 - Primary master: carrega a base de dados de disco
 - Secondary master: na inicialização recebe a base de dados do primary server. Periodicamente contacta o primary master para a actualizar
 - Um servidor pode ser master para mais que um domínio, sendo primary para um e secondary para outros
- ▶ **Caching: apenas mantém dados em cache**
 - Contacta os outros servidores para obter a informação
 - Não é autoridade para nenhum domínio
- ▶ **Remote: servidor remoto**
- ▶ **Slave: redirige os pedidos que não consegue servir para uma lista de servidores, e não para os master**

Exemplo de Arquitectura do BIND



Servidores DNS

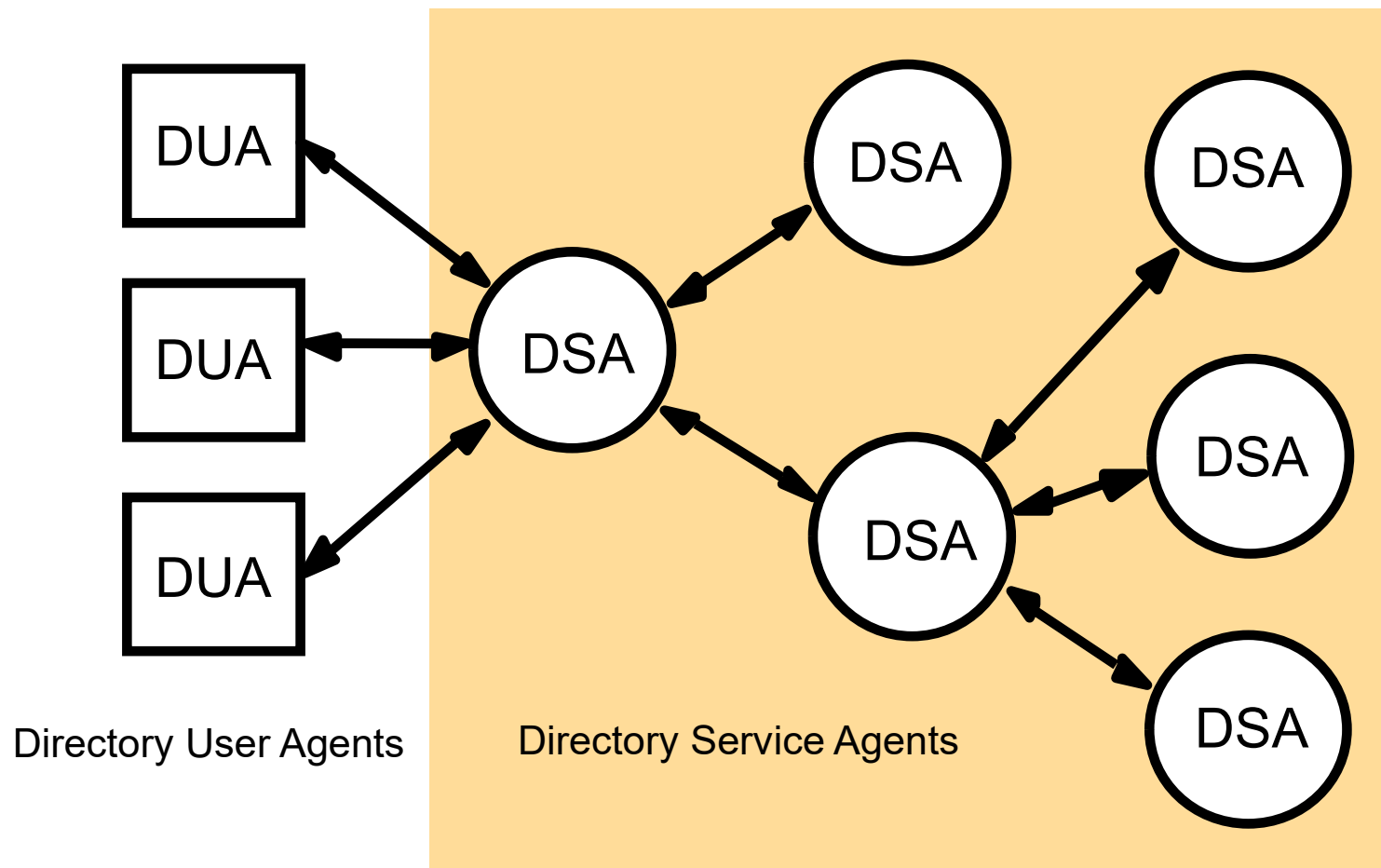
- Nomes dos servidores em *itálico*
- Nomes dos domínios entre parenteses
- Setas representam entradas no directório



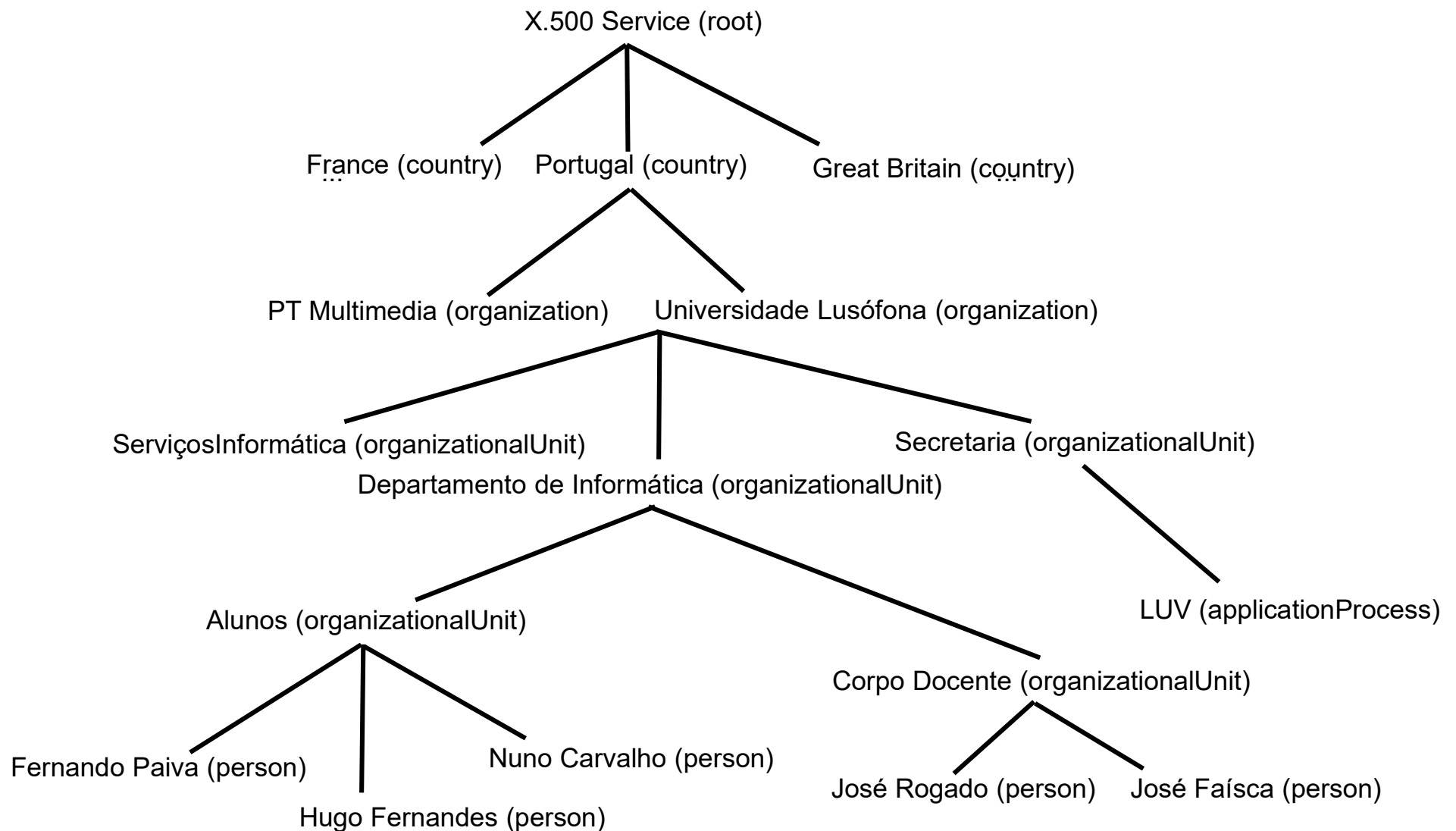
Serviço de Directório X.500

- ▶ Norma conjunta ISO/CCITT com objectivos ambiciosos:
 - Serviço de Directório distribuído de grande dimensão com servidores espalhados por todo o mundo
 - Permitir o armazenamento de grande variedade de informação
 - Desde informação respeitante a indivíduos até catálogos de componentes
 - Poder ser utilizado como um serviço de nomeação clássico
- ▶ Estrutura
 - Estrutura Hierárquica organizada em árvore
 - **DIT** - Directory Information Tree
 - A informação em cada nó é composta por um nome e um conjunto de atributos
 - **DIB** – Directory Information Base
 - O objectivo da norma era permitir a existência de um DIB a nível mundial com sub-directórios localizados em diferentes países
- ▶ Terminologia
 - Servidor: **DSA** - *Directory Service Agent*
 - Cliente: **DUA** - *Directory User Agent*
 - Nome local: **RDN** – *Relative Distinguished Name*
 - Nome Global: **DN** - *Distinguished Name*
 - Concatenação de todos os nomes locais desde a raiz

Arquitetura de Serviços X.500



Exemplo de Árvore X.500



Estrutura do DIB

- ▶ A estrutura de um elemento do DIB é flexível e consiste num conjunto variável de atributos
- ▶ Um atributo é constituído por um tipo e um ou mais valores
 - Existem tipos predefinidos e podem ser definidos novos tipos
 - *countryName, organizationName, commonName, telephoneNumber, mailbox, ...*
 - Cada elemento do DIB é definido por uma **objectClass** que determina as suas características (tipo, sintaxe, etc...)
 - *Organization, organizationalUnit, person, ...*
 - Novos objectClasses podem ser definidos
 - As características das *objectClasses* são herdadas
- ▶ Um elemento do DIB é determinado pelo seu *distinguished attribute*
 - É designado por *Relative Distinguished Name* e identifica um elemento de forma única no nível da DIT a que pertence
 - Ex: Portugal, Corpo Docente, Alunos, José Rogado, etc..
 - A concatenação de todos os RDNs desde a raiz forma o *Distinguished Name*
 - Ex.: *cn=José Rogado, ou=Corpo Docente, ou=Dpto de Informática, o=ULHT, c=Portugal*

Operações no Directório

► Procura

- **A realização de procuras eficazes é uma das propriedades essenciais de um directório, pelo que a sua implementação deve fazer apelo a réplicas e *caching*.**
- Tem como argumento o nome de um nó para o início da procura e um conjunto de valores de atributos para os quais se pretende realizar a filtragem
- São devolvidos todos os elementos cujos atributos satisfaçam o critério de procura
- A procura pode ser uma operação extremamente pesada se a DIT for vasta e os critérios de procura pouco selectivos

► Leitura

- Tem como argumento um DN ou RDN e um conjunto de atributos dos quais se pretendem os valores
- O nome é resolvido através da procura na DIT possivelmente percorrendo vários níveis e depois de localizado o DSA que gere o nome, este devolve a informação ao DUA

Operações no Directório

► Administração e actualização

- A interface de acesso do DSA inclui primitivas para adicionar, apagar, mover e modificar elementos do directório
- A operações de modificação não têm um desempenho elevado, pois o directório é sobretudo um repositório de consulta
- Os acessos ao directório podem ser protegidos através de vários tipos de controlo
 - Simple password ou Public Key Encryption baseado em certificados X.509

► Implementação

- A norma X.500 não define a forma como o directório é implementado
 - A comunicação entre DUA e DSA é realizado através do Directory Access Protocol (DAP) baseado nos protocolos ISO, pouco utilizado hoje em dia
- A implementação e o protocolo de acesso constituem as principais razões pelas quais o X.500 não teve o sucesso previsto

O LDAP

- ▶ O X.500 não teve a propagação prevista a nível mundial
 - Sistema complexo e com sintaxe de nomeação pouco intuitiva
 - O DNS foi adoptado como standard de facto na Internet
 - É contudo muito utilizado em organizações como directório de serviços, utilizadores ou credenciais de autenticação
- ▶ *Lightweight Directory Access Protocol - LDAP*
 - Um protocolo mais leve baseado em TCP/IP foi desenvolvido na Univ. de Michigan e utilizado para aceder a directórios que implementam o standard X.500
 - Existe uma versão freeware: OpenLDAP (www.openldap.org)
- ▶ Inúmeras versões de LDAP de vários fabricantes são utilizados como repositórios de informação em plataformas IMS (*Identity Management Systems*)
 - Sun Directory Server
 - Microsoft Active Directory
 - Novell eDirectory
 - Isode M-Vault

Actualização de Conteúdo LDAP

- ▶ Para criar um directório o modificar o seu conteúdo existe um formato específico que permite criar scripts de gestão
 - **LDAP Data Interchange Format (LDIF)**
- ▶ Permite a importação ou exportação de conteúdos inteiros de um directório
- ▶ Utiliza uma sintaxe simples baseada na utilização dos identificadores do DIB
 - **dn: distinguished name**
 - **dc: domain component**
 - **ou: organizational unit**
 - **cn: common name**
- ▶ As operações são efectuadas sobre os campos indicados
 - **add, replace, delete**

Perguntas a que devo ser capaz de responder

- ▶ O que é a resolução de nomes iterativa e recursiva ?
- ▶ Porque é que o caching funciona bem nos serviços de nomes ?
- ▶ Que tipos de servidores de nomes existem no DNS e qual a função de cada um ?
- ▶ O que é o TTL (time to live) mantido pelos servidores nas suas caches ?

Referências

▶ Referências X.500

- <http://sec.cs.kent.ac.uk/x500book>

▶ Referências LDAP

- http://docstore.mik.ua/orelly/perl/sysadmin/appb_01.htm
- <http://www.openldap.org>
- <http://www.novell.com/products/edirectory>
- <http://www.isode.com/products/m-vault.html>

▶ Trabalho Complementar

- Ler o tutorial sobre o JNDI (Java Naming and Directory Interface) particularmente a secção sobre *Naming and Directory Concepts*
- <http://java.sun.com/docs/books/tutorial/jndi/concepts/index.html>