

# ACH2147 - Desenvolvimento de Sistemas de Informação Distribuídos

## Aula 12 – Abordagens de Nomeação

Norton Trevisan Roman

2 de junho de 2022

# Abordagens de Nomeação

- Nomes Estruturados
  - Espaço de Nomes
  - Resolução de Nomes
  - Implementação de Espaços de Nomes
- Nomeação Baseada em Atributos (Hierárquica)

# Abordagens de Nomeação

- **Nomes Estruturados**
  - **Espaço de Nomes**
  - Resolução de Nomes
  - Implementação de Espaços de Nomes
- Nomeação Baseada em Atributos (Hierárquica)

# Nomes Estruturados

## Espaço de Nomes

- Nomes planos são bons para máquinas
  - Mas pouco convenientes para humanos

# Nomes Estruturados

## Espaço de Nomes

- Nomes planos são bons para máquinas
  - Mas pouco convenientes para humanos
- Uma alternativa é o uso de nomes estruturados
  - Compostos por nomes simples, fáceis às pessoas

# Nomes Estruturados

## Espaço de Nomes

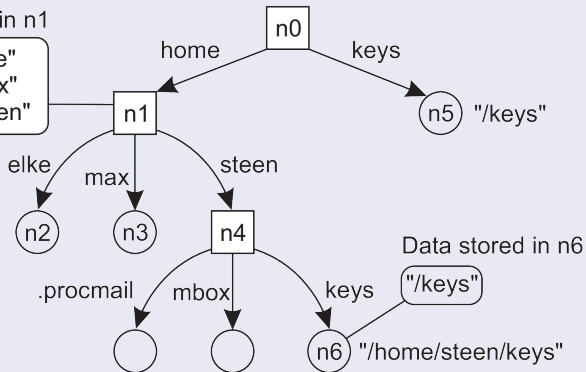
- Nomes planos são bons para máquinas
  - Mas pouco convenientes para humanos
- Uma alternativa é o uso de nomes estruturados
  - Compostos por nomes simples, fáceis às pessoas
- Nomes são comumente organizados em **espaços de nomes**
  - No caso de nomes estruturados, podem ser representados por grafos dirigidos rotulados

# Nomes Estruturados

## Espaço de Nomes

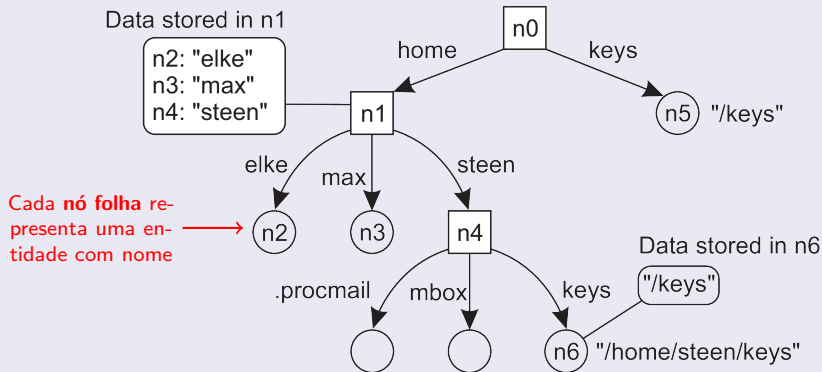
Data stored in n1

n2: "elke"  
n3: "max"  
n4: "steen"



# Nomes Estruturados

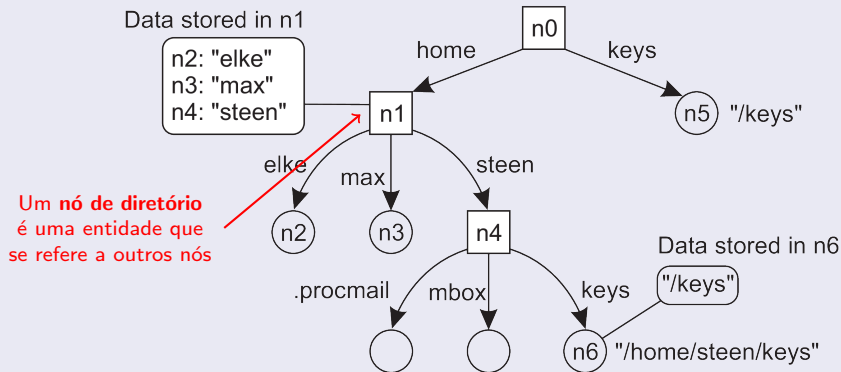
## Espaço de Nomes





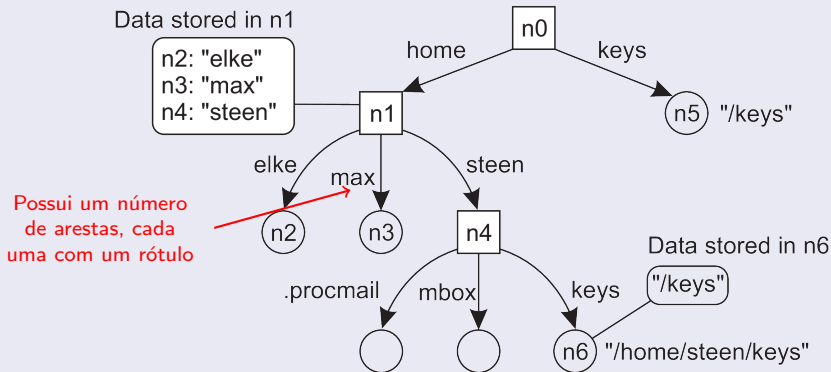
# Nomes Estruturados

## Espaço de Nomes



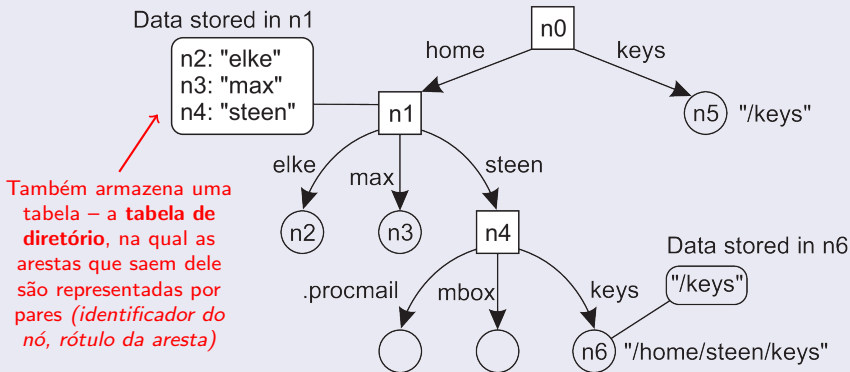
# Nomes Estruturados

## Espaço de Nomes



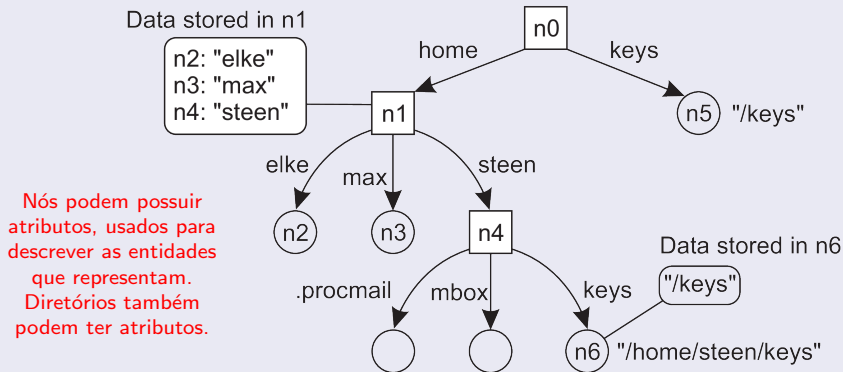
# Nomes Estruturados

## Espaço de Nomes



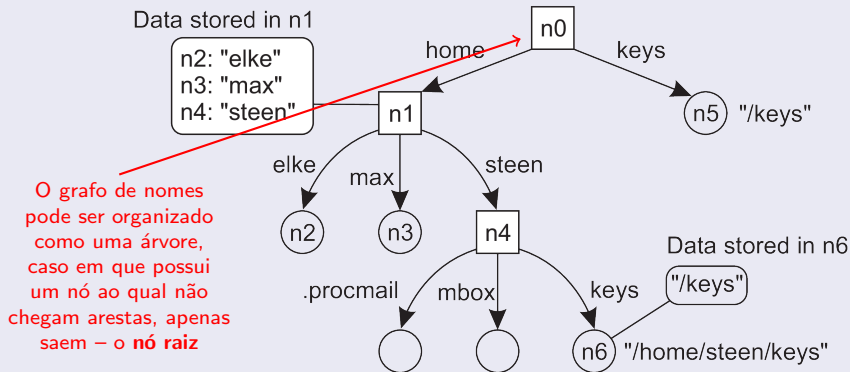
# Nomes Estruturados

## Espaço de Nomes



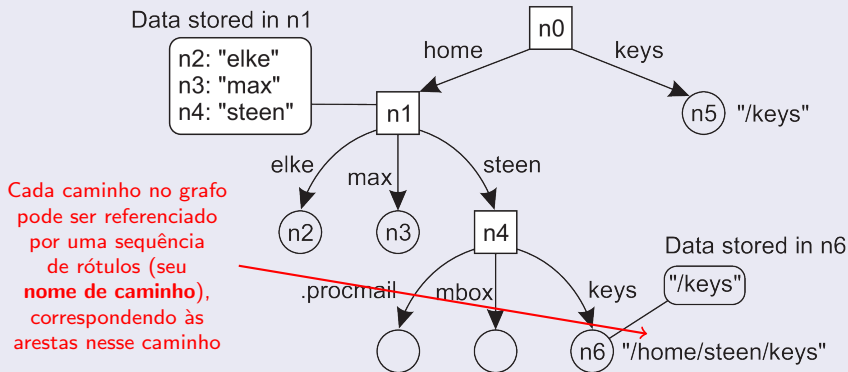
# Nomes Estruturados

## Espaço de Nomes



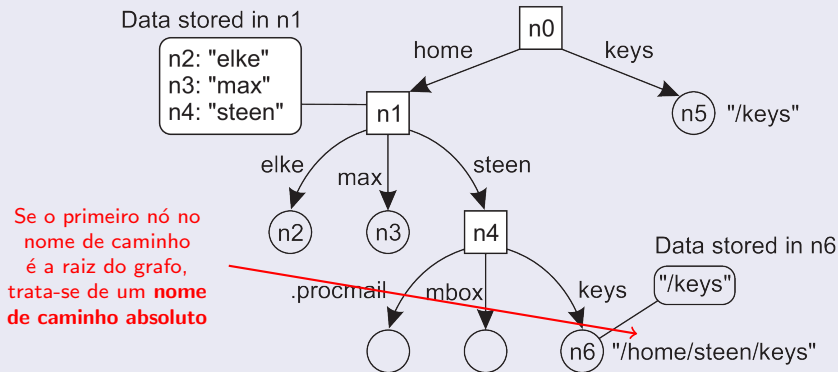
# Nomes Estruturados

## Espaço de Nomes



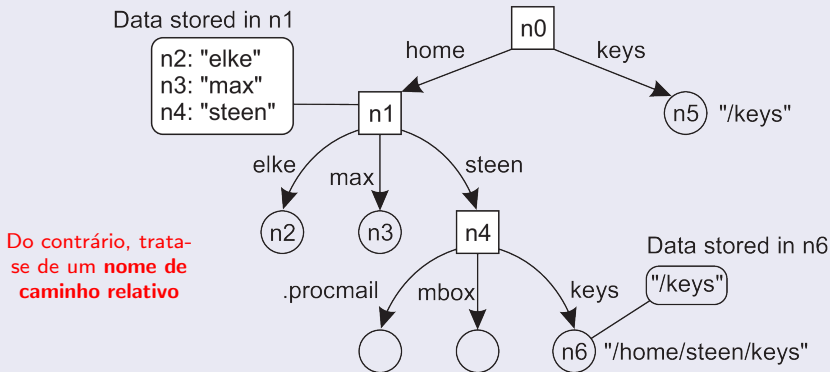
# Nomes Estruturados

## Espaço de Nomes



# Nomes Estruturados

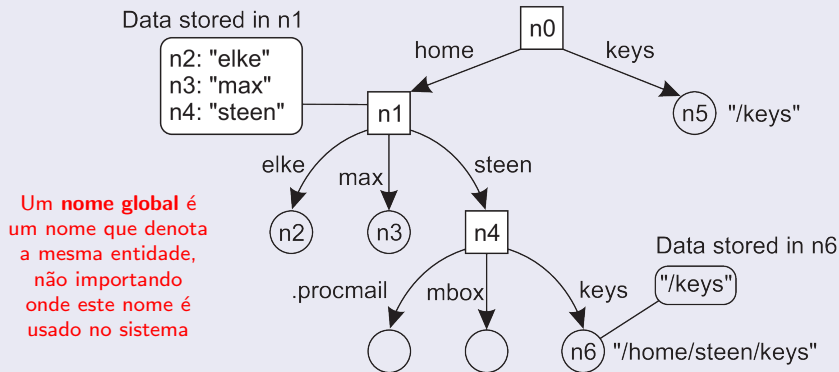
## Espaço de Nomes





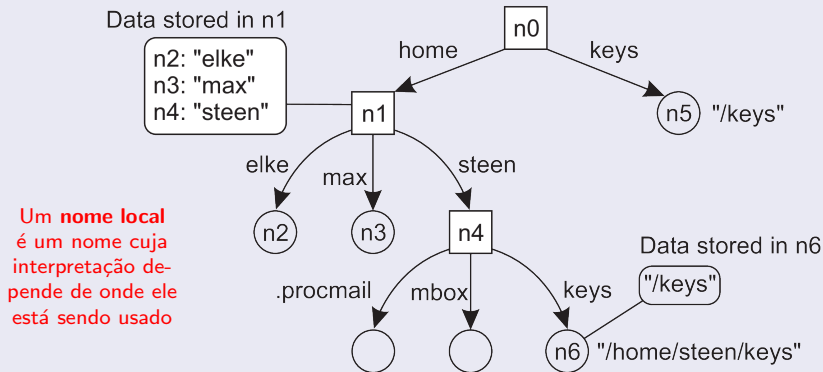
# Nomes Estruturados

## Espaço de Nomes



# Nomes Estruturados

## Espaço de Nomes

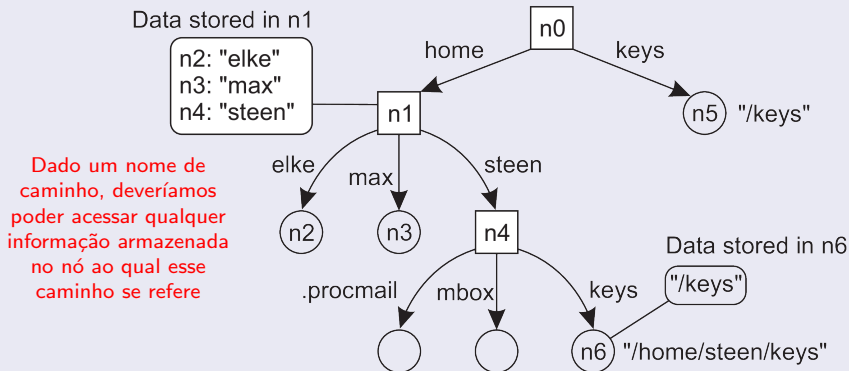


# Abordagens de Nomeação

- **Nomes Estruturados**
  - Espaço de Nomes
  - **Resolução de Nomes**
  - Implementação de Espaços de Nomes
- Nomeação Baseada em Atributos (Hierárquica)

# Nomes Estruturados

## Resolução de Nomes

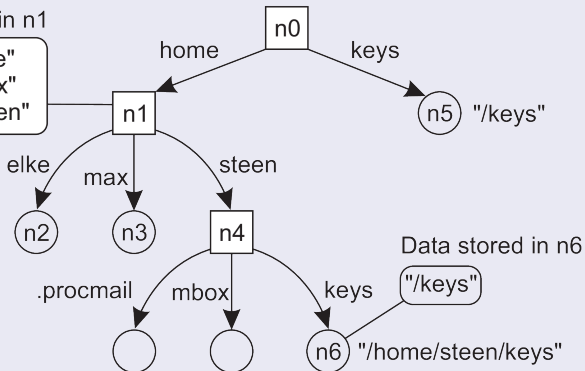


# Nomes Estruturados

## Resolução de Nomes

Data stored in n1

n2: "elke"  
n3: "max"  
n4: "steen"

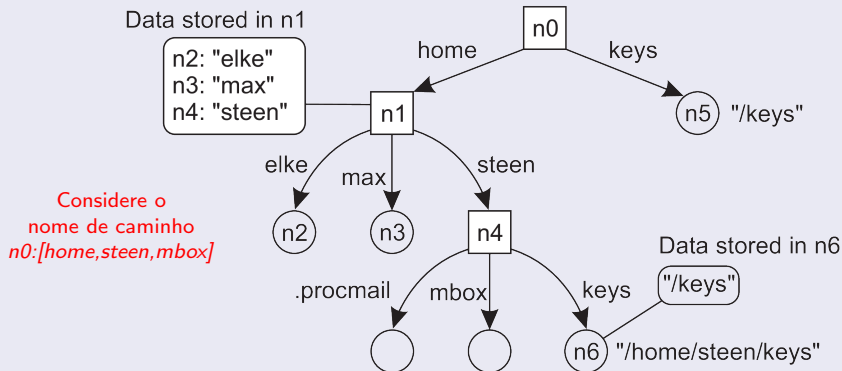


Dado um nome de caminho, deveríamos poder acessar qualquer informação armazenada no nó ao qual esse caminho se refere

O processo de analisar o nome é chamado de **resolução de nome**

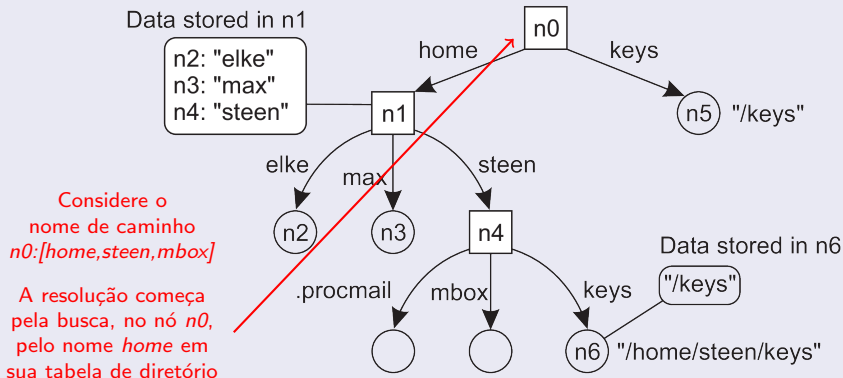
# Nomes Estruturados

## Resolução de Nomes



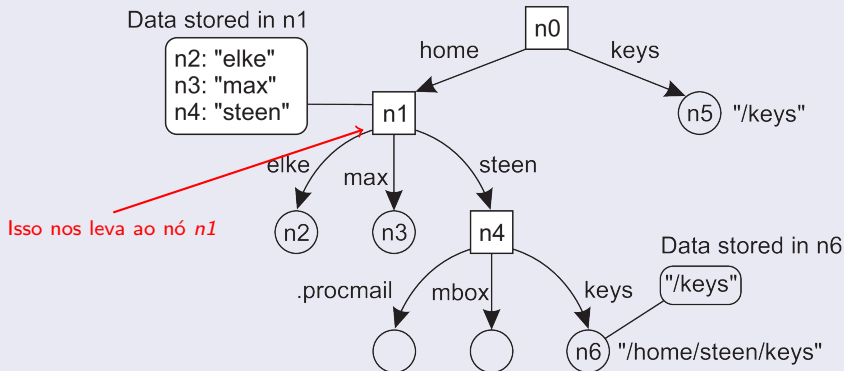
# Nomes Estruturados

## Resolução de Nomes



# Nomes Estruturados

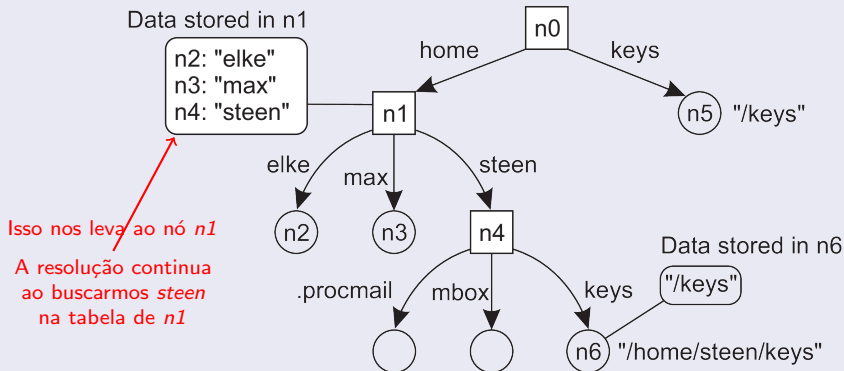
## Resolução de Nomes





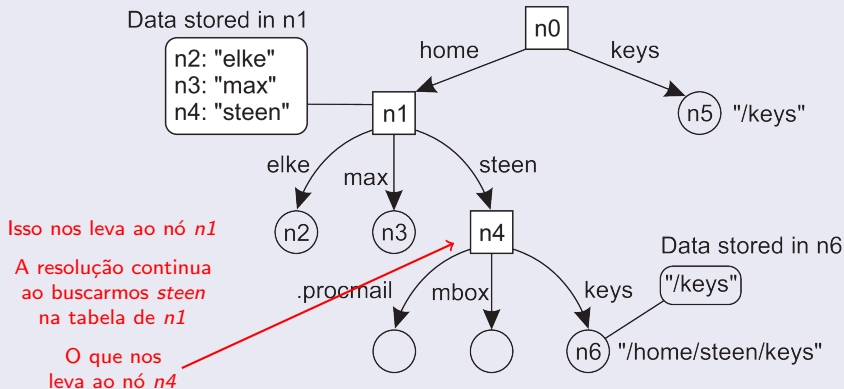
# Nomes Estruturados

## Resolução de Nomes



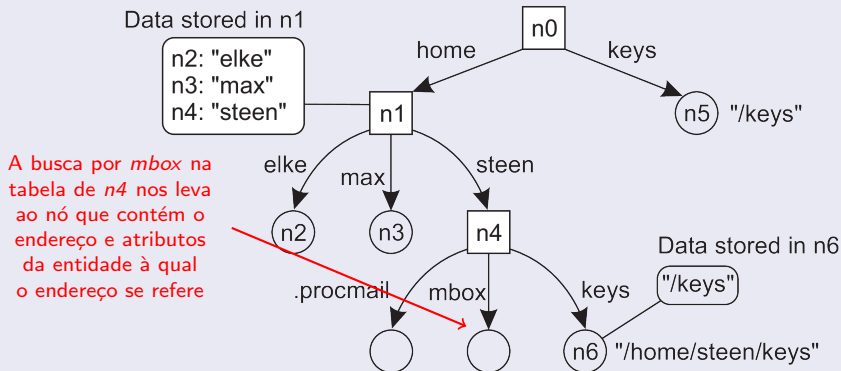
# Nomes Estruturados

## Resolução de Nomes



# Nomes Estruturados

## Resolução de Nomes



# Nomes Estruturados

## Resolução de Nomes: Mecanismo de *closure*

- Para resolver um nome, contudo, precisamos de um diretório inicial
- Saber **como e onde** iniciar a resolução de um nome é conhecido como **mecanismo de closure**
- Essencialmente, trata-se da escolha do nó inicial em um grafo de espaço de nomes

## Resolução de Nomes: Mecanismo de *closure*

- Para resolver um nome, contudo, precisamos de um diretório inicial
  - Saber **como e onde** iniciar a resolução de um nome é conhecido como **mecanismo de closure**
  - Essencialmente, trata-se da escolha do nó inicial em um grafo de espaço de nomes
- Ex:
  - `each.usp.br`: inicia em um servidor DNS
  - `00551126480149`: disca um número de telefone
  - `200.144.248.41`: rota para um servidor da USP

# Nomes Estruturados

## Resolução de Nomes: *Links* para nomes

- Em grafos de nomes podemos implementar também *alias*
  - Outro nome para a mesma entidade

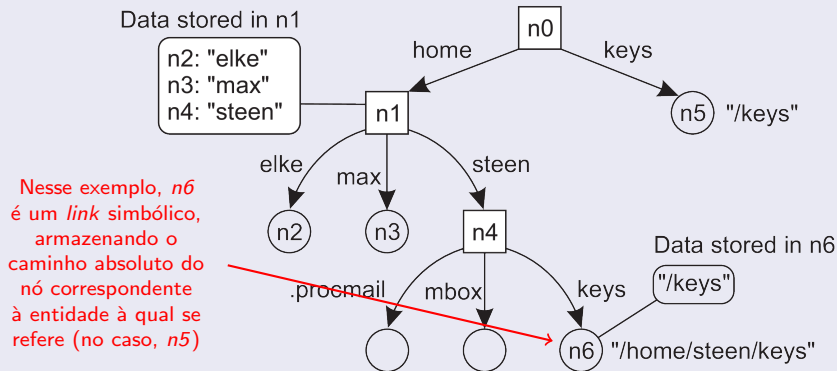
# Nomes Estruturados

## Resolução de Nomes: *Links* para nomes

- Em grafos de nomes podemos implementar também *aliases*
  - Outro nome para a mesma entidade
- 2 modos de implementação:
  - **Hard link**: permitir que caminhos absolutos se refiram ao mesmo nó no grafo
  - **Soft link** (ou **link simbólico**):
    - Representar uma entidade por um nó folha
    - Em vez de armazenar o endereço ou estado da entidade nele, o nó armazena um caminho absoluto

# Nomes Estruturados

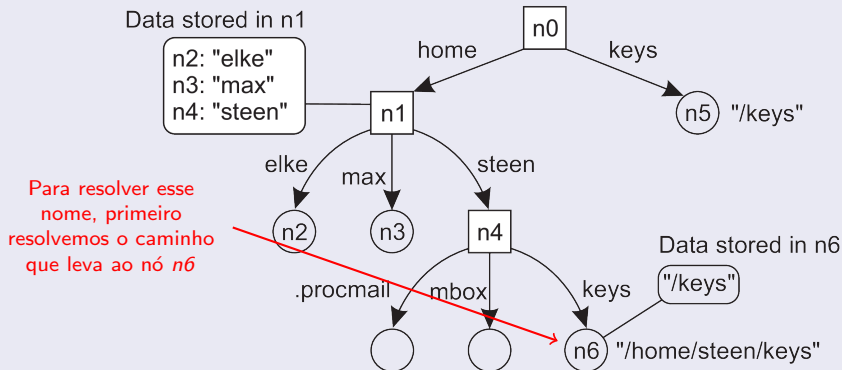
## Resolução de Nomes: *Links* simbólicos





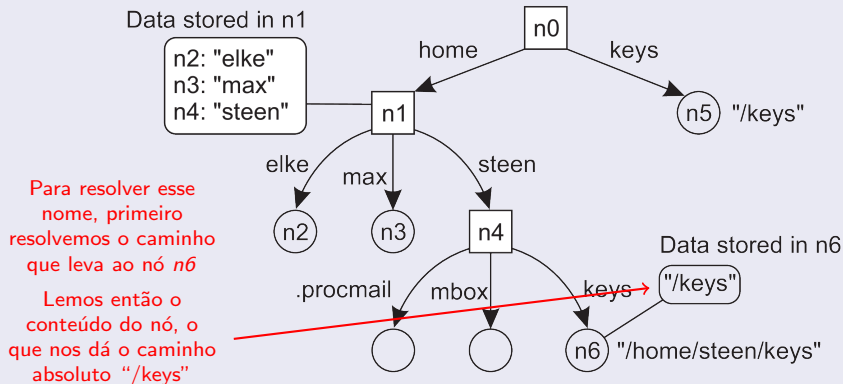
# Nomes Estruturados

## Resolução de Nomes: *Links* simbólicos



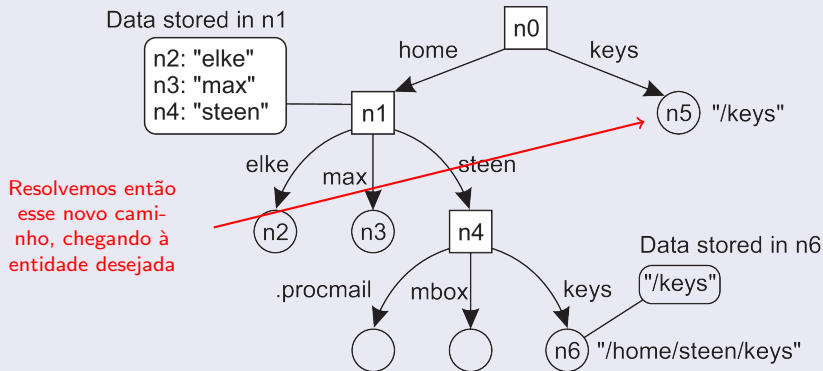
# Nomes Estruturados

## Resolução de Nomes: *Links* simbólicos



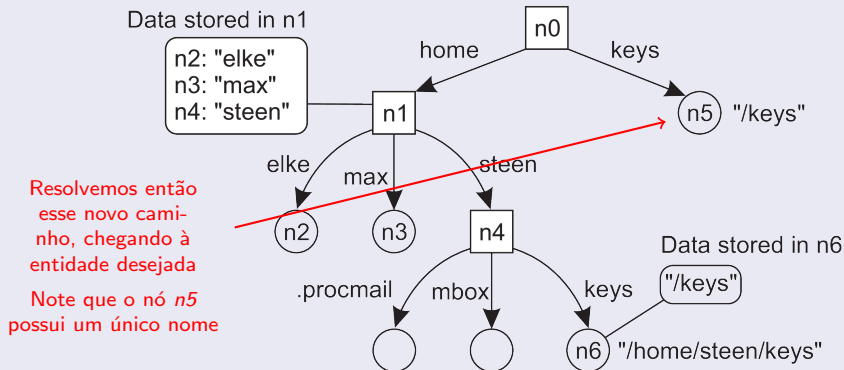
# Nomes Estruturados

## Resolução de Nomes: *Links* simbólicos



# Nomes Estruturados

## Resolução de Nomes: *Links* simbólicos



# Nomes Estruturados

## Resolução de Nomes: Montagem

- Podemos usar a resolução de nomes para mesclar diferentes espaços de nomes de modo transparente

## Resolução de Nomes: Montagem

- Podemos usar a resolução de nomes para mesclar diferentes espaços de nomes de modo transparente
- Fazemos isso através da **montagem**
  - Permite que um nó diretório armazene o identificador de um nó diretório de Espaços de Nomes diferente
  - Esse espaço diferente é o **espaço de nomes estrangeiro**
  - O nó de diretório que armazena o identificador é um **mount point**
  - O nó no espaço de nomes estrangeiro, onde continuaremos a resolução, é um **mounting point**

## Resolução de Nomes: Montagem

- Durante a resolução de um nome, busca-se pelo *mounting point*
  - E a resolução continua ao acessarmos sua tabela de diretório

## Resolução de Nomes: Montagem

- Durante a resolução de um nome, busca-se pelo *mounting point*
  - E a resolução continua ao acessarmos sua tabela de diretório
- Precisamos então de um nó diretório que sirva de *mount point*
  - Armazenando a informação necessária para identificação e acesso do *mounting point*



## Resolução de Nomes: Montagem em SDs

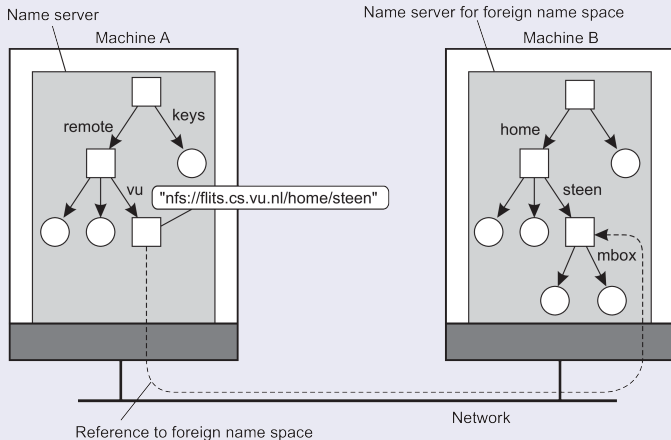
- Para montar um espaço estrangeiro em um sistema distribuído, precisamos ter
  - O nome do protocolo de acesso
  - O nome do servidor
  - O nome do *mounting point* no espaço de nomes estrangeiro

## Resolução de Nomes: Montagem em SDs

- Para montar um espaço estrangeiro em um sistema distribuído, precisamos ter
  - O nome do protocolo de acesso
  - O nome do servidor
  - O nome do *mounting point* no espaço de nomes estrangeiro
- Uma possibilidade é representarmos esses nomes como uma URL
  - Ex: NFS

# Nomes Estruturados

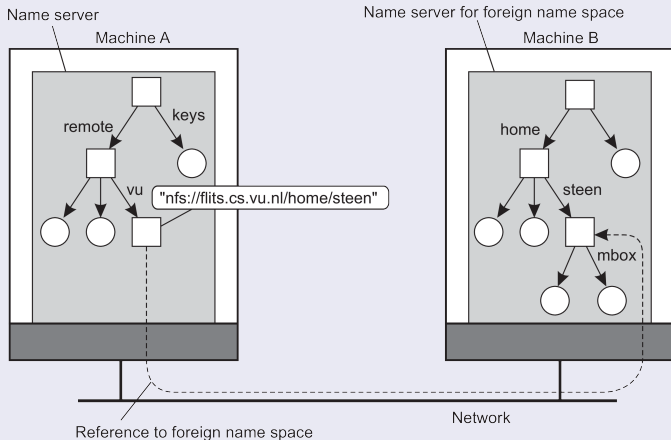
## Resolução de Nomes: NFS



# Nomes Estruturados

## Resolução de Nomes: NFS

Suponha que que-  
remos acessar `"/re-  
mote/vu/mbox"`

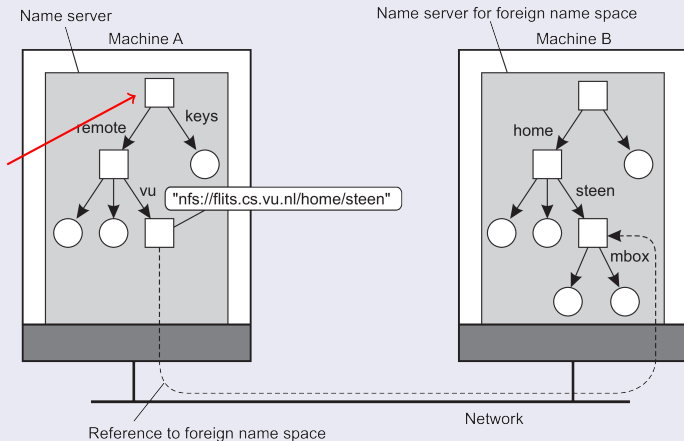


# Nomes Estruturados

## Resolução de Nomes: NFS

Suponha que que-  
remos acessar `"/re-  
mote/vu/mbox"`

Sua resolução  
começa no diretório  
raiz do cliente



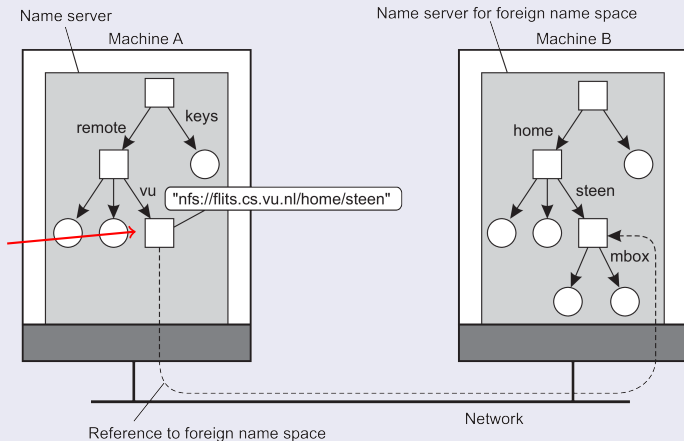
# Nomes Estruturados

## Resolução de Nomes: NFS

Suponha que que-  
remos acessar “/re-  
mote/vu/mbox”

Sua resolução  
começa no diretório  
raiz do cliente

E continua até o  
nó “/remote/vu”



# Nomes Estruturados

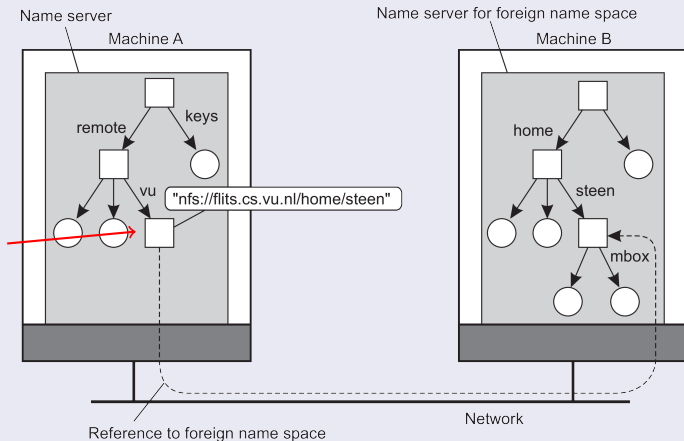
## Resolução de Nomes: NFS

Suponha que que-  
remos acessar “/re-  
mote/vu/mbox”

Sua resolução  
começa no diretório  
raiz do cliente

E continua até o  
nó “/remote/vu”

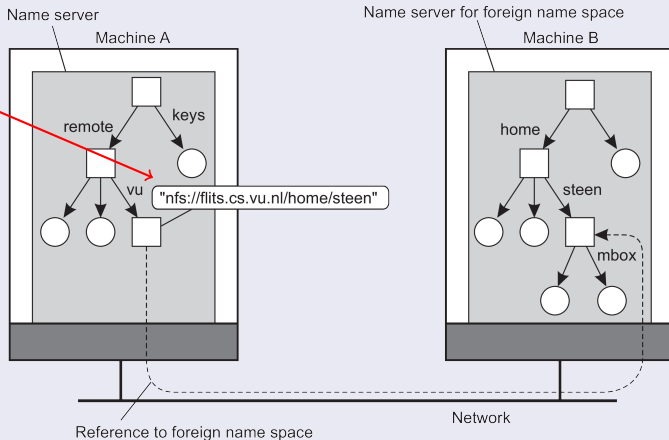
Nó diretório usado  
para armazenar  
a URL para o  
espaço remoto



# Nomes Estruturados

## Resolução de Nomes: NFS

A URL é retornada, fazendo com que o cliente contacte o servidor "flits.cs.vu.nl" usando o protocolo NFS

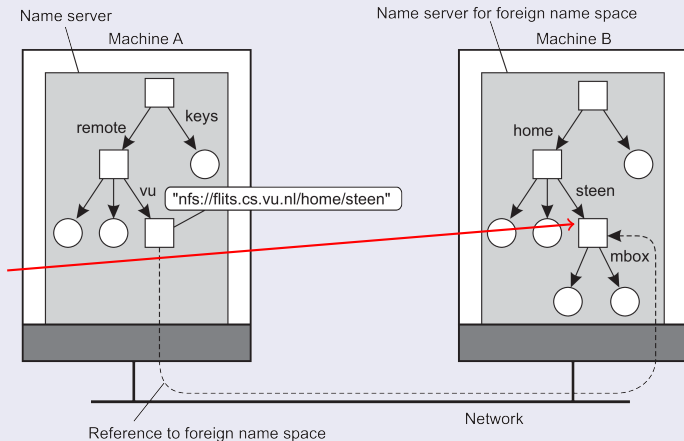




# Nomes Estruturados

## Resolução de Nomes: NFS

A URL é retornada, fazendo com que o cliente contacte o servidor "flits.cs.vu.nl" usando o protocolo NFS. Em seguida, o diretório "/home/steen" é acessado (conforme consta da URL).



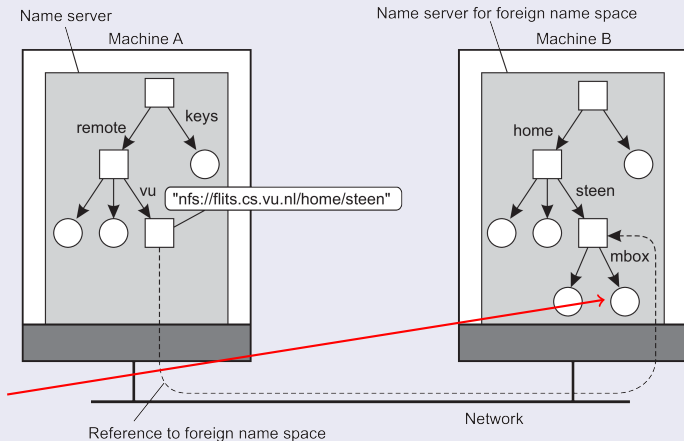
# Nomes Estruturados

## Resolução de Nomes: NFS

A URL é retornada, fazendo com que o cliente contacte o servidor "flits.cs.vu.nl" usando o protocolo NFS

Em seguida, o diretório "/home/steen" é acessado (conforme consta da URL)

E a resolução continua a partir desse nó diretório, chegando a "mbox"



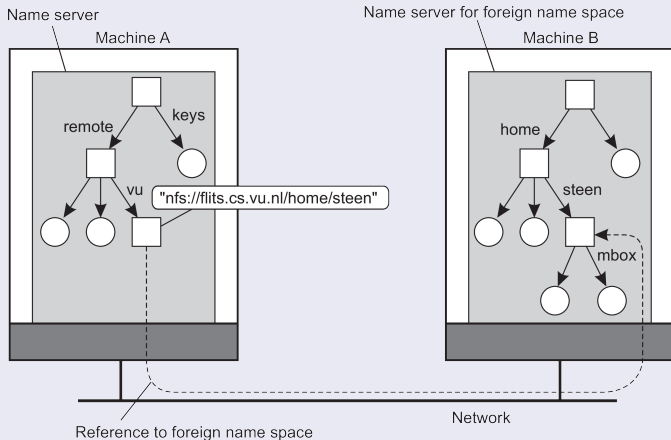
# Nomes Estruturados

## Resolução de Nomes: NFS

Montar um sistema de arquivos remoto permite comandos do tipo

```
cliente$ ls -l /remote/vu
```

que listará os arquivos no diretório “/home/steen” do servidor



# Abordagens de Nomeação

- **Nomes Estruturados**
  - Espaço de Nomes
  - Resolução de Nomes
  - **Implementação de Espaços de Nomes**
- Nomeação Baseada em Atributos (Hierárquica)

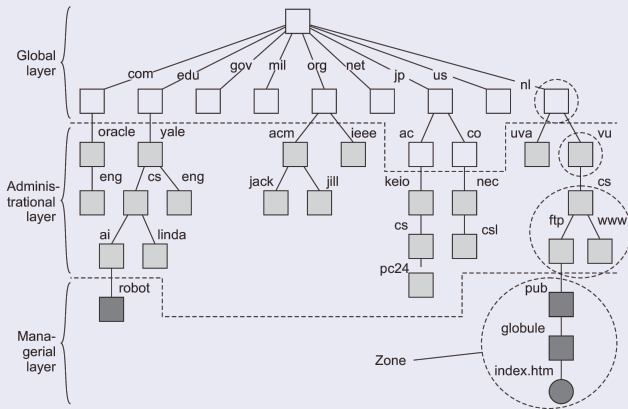
## Implementação de Espaços de Nomes

- Principal questão para Sistemas Distribuídos
  - Como distribuir o processo de resolução de nomes, bem como o gerenciamento do espaço de nomes, entre múltiplas máquinas, pela distribuição de nós do grafo de nomes?

# Nomes Estruturados

## Implementação de Espaços de Nomes

- Distinguímos 3 níveis

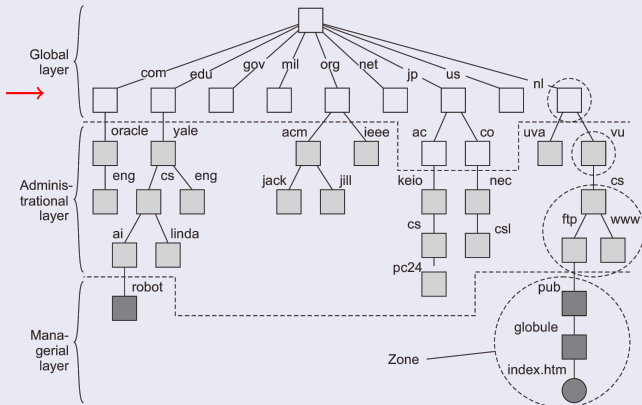


# Nomes Estruturados

## Implementação de Espaços de Nomes

- Distinguímos 3 níveis

**Nível global:**  
consiste dos nós  
diretórios de alto  
nível (o nó raiz  
e seus filhos).



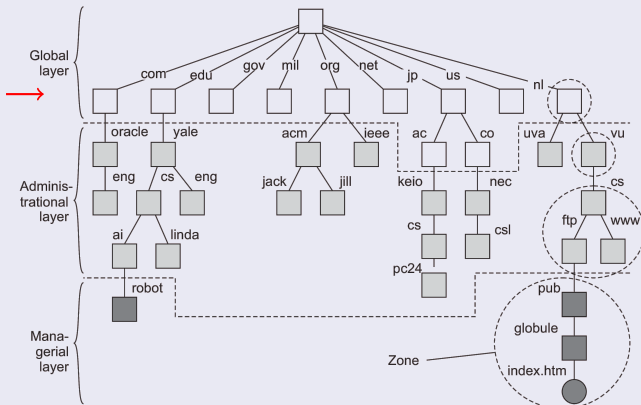
# Nomes Estruturados

## Implementação de Espaços de Nomes

- Distinguímos 3 níveis

**Nível global:**  
consiste dos nós diretórios de alto nível (o nó raiz e seus filhos).

São gerenciados em conjunto por diferentes administradores.



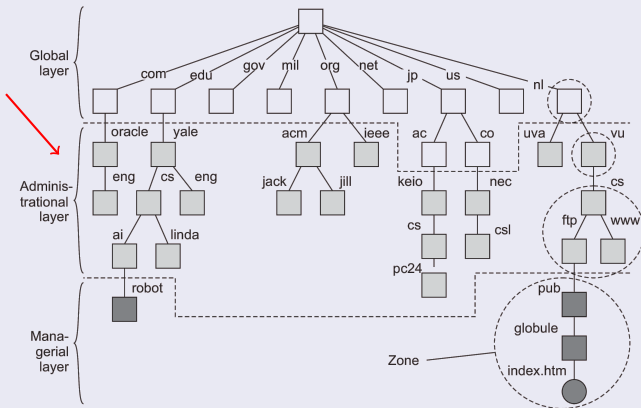


# Nomes Estruturados

## Implementação de Espaços de Nomes

- Distinguímos 3 níveis

**Nível administrativo:** nós de diretório de nível intermediário.



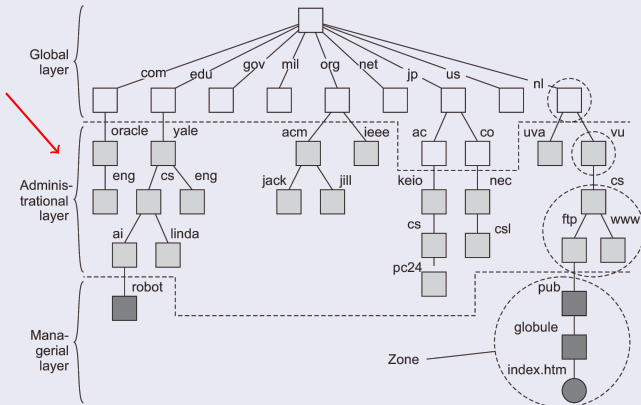
# Nomes Estruturados

## Implementação de Espaços de Nomes

- Distinguímos 3 níveis

**Nível administrativo:** nós de diretório de nível intermediário.

Podem ser agrupados de modo que cada grupo seja responsabilidade de um administrador diferente.

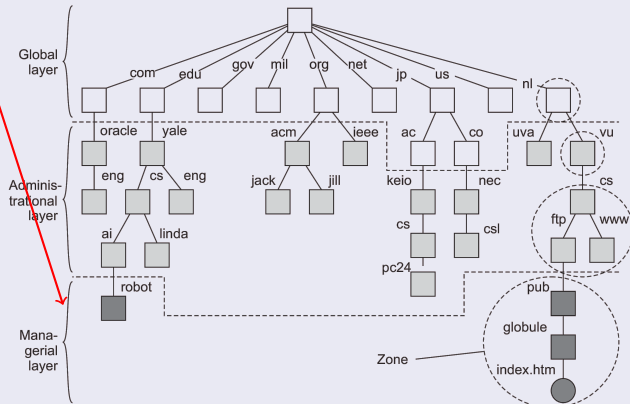


# Nomes Estruturados

## Implementação de Espaços de Nomes

- Distinguímos 3 níveis

**Nível gerencial:**  
nós diretórios de  
nível inferior dentro  
de uma única  
administração.



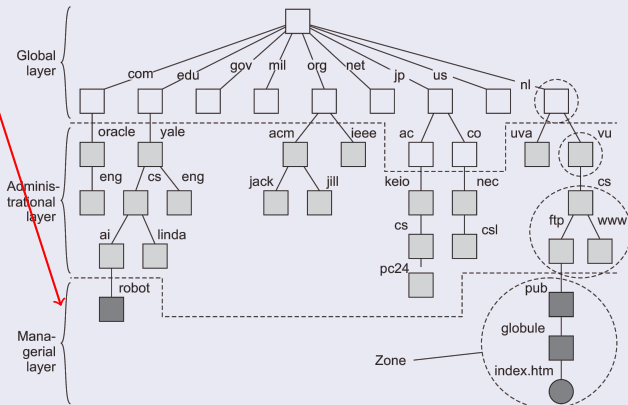
# Nomes Estruturados

## Implementação de Espaços de Nomes

- Distinguímos 3 níveis

**Nível gerencial:**  
nós diretórios de  
nível inferior dentro  
de uma única  
administração.

O problema  
principal é mapear  
os nós de diretório  
aos servidores  
de nomes locais.



# Nomes Estruturados

## Implementação de Espaços de Nomes

- Comparação entre os níveis

	<b>Global</b>	<b>Administrativo</b>	<b>Gerencial</b>
Escala geográfica	Mundial	Organização	Departamento
Nº de Nós	Poucos	Muitos	Qdes. enormes
Responsividade a buscas	Segundos	Milissegundos	Imediato
Propagação de atualizações	Tardio	Imediato	Imediato
Nº de Réplicas	Muitos	Nenhum ou poucos	Nenhum
Cache no lado do cliente	Sim	Sim	Às vezes

## Implementação de EN: Resolução de Nomes

- Modelo
  - Cada cliente tem acesso a um *name resolver* local, responsável por garantir que o processo de resolução seja executado

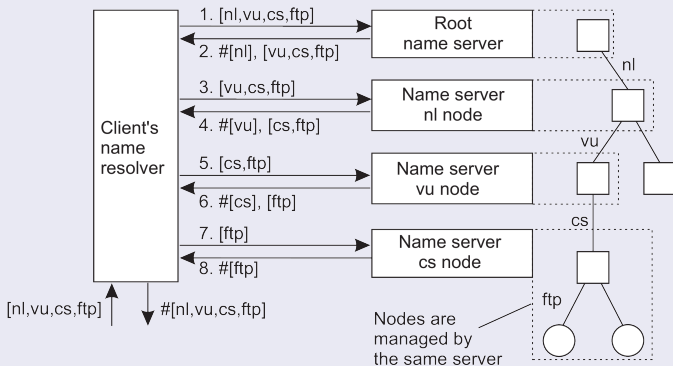
## Implementação de EN: Resolução de Nomes

- Modelo
  - Cada cliente tem acesso a um *name resolver* local, responsável por garantir que o processo de resolução seja executado
- Há 2 modos de implementar a resolução de nomes
  - Resolução de Nomes Iterativa
  - Resolução de Nomes Recursiva

# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: `ftp://ftp.cs.vu.nl/pub/globe/index.html`



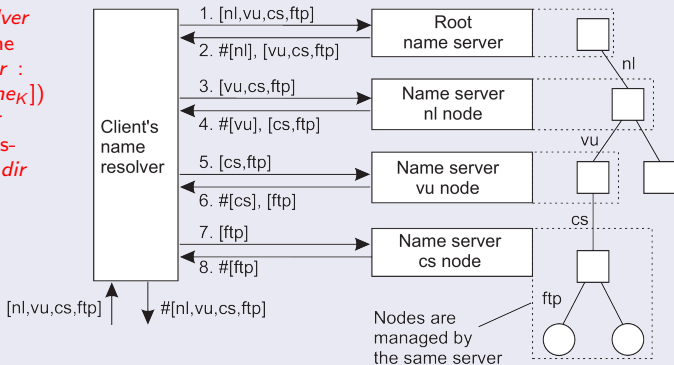


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

O *name resolver* envia o nome completo (*dir* :  $[nome_1, \dots, nome_K]$ ) ao servidor de nomes responsável por *dir*



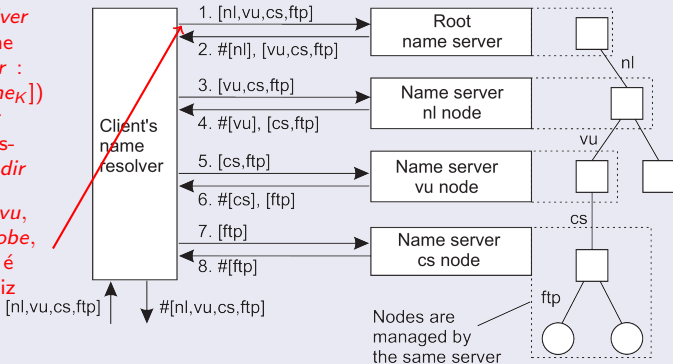
# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

O *name resolver* envia o nome completo (*dir* :  $[nome_1, \dots, nome_K]$ ) ao servidor de nomes responsável por *dir*

No caso,  $[nl, vu, cs, ftp, pub, globe, index.html]$  é enviado à raiz

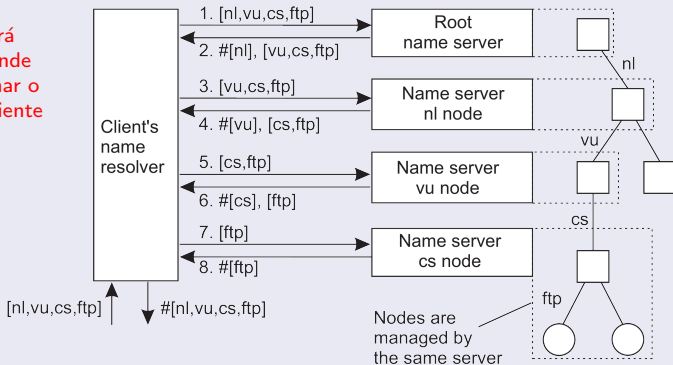


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

O servidor irá resolver até onde puder, e retornar o resultado ao cliente



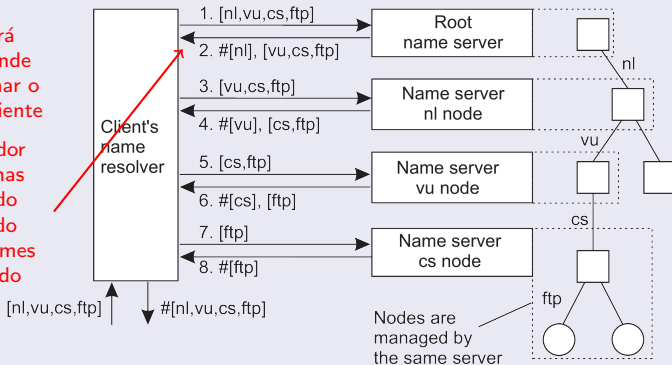
# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

O servidor irá resolver até onde puder, e retornar o resultado ao cliente

Aqui, o servidor conhece apenas *nl*, retornando o endereço do servidor de nomes a ele associado

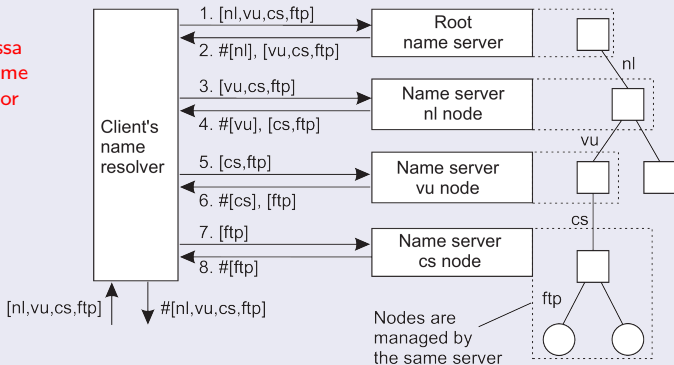


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

O cliente passa  
o resto do nome  
a este servidor



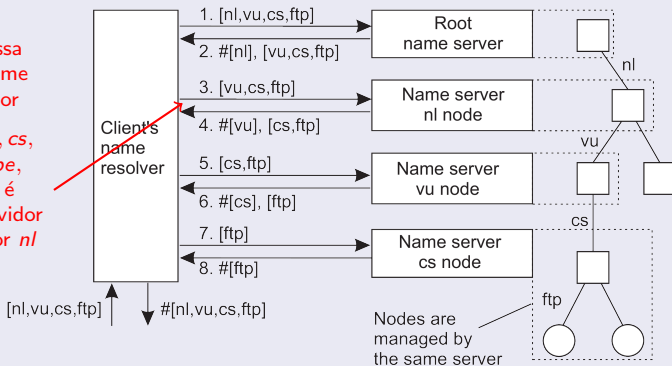
# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

O cliente passa o resto do nome a este servidor

Aqui, *nl* : [*vu*, *cs*, *ftp*, *pub*, *globe*, *index.html*] é passado ao servidor responsável por *nl*

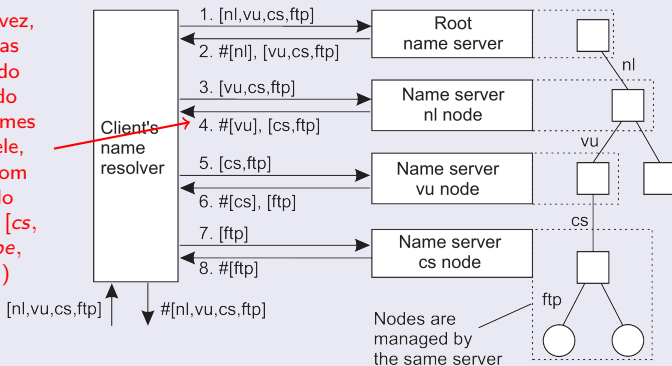


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

Este, por sua vez, resolve apenas *vu*, retornando o endereço do servidor de nomes associado a ele, juntamente com o restante do caminho (*vu* : [*cs*, *ftp*, *pub*, *globe*, *index.html*])

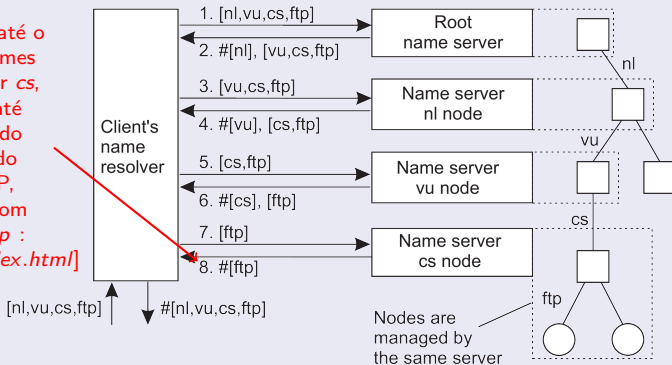


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: `ftp://ftp.cs.vu.nl/pub/globe/index.html`

E assim segue até o servidor de nomes responsável por *cs*, que resolve até *ftp*, retornando o endereço do servidor FTP, juntamente com o caminho *ftp* : *[pub, globe, index.html]*



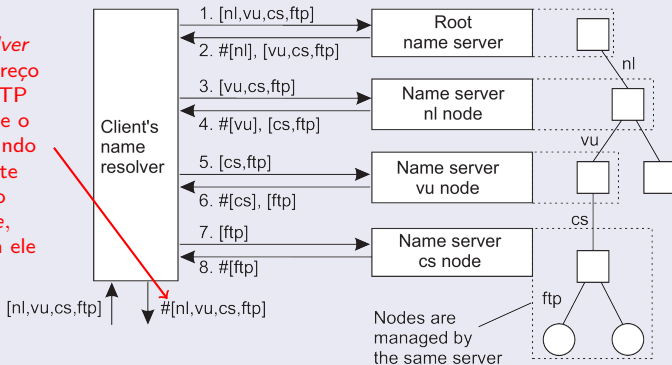


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa
  - Ex: `ftp://ftp.cs.vu.nl/pub/globe/index.html`

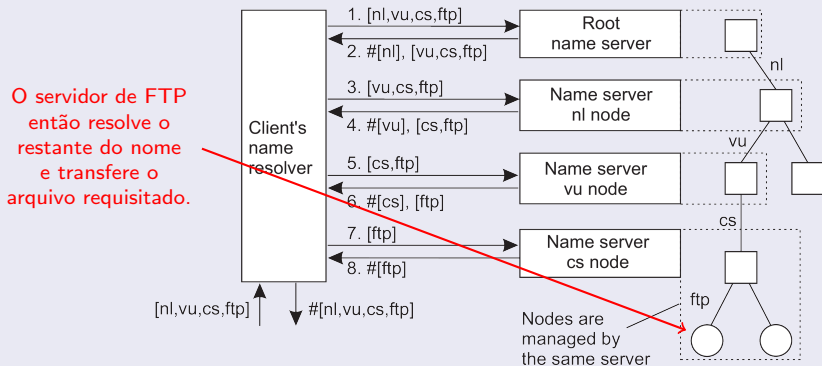
*O name resolver retorna o endereço do servidor FTP ao cliente, que o contata, passando a última parte do caminho ([pub, globe, index.html]) a ele*



# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

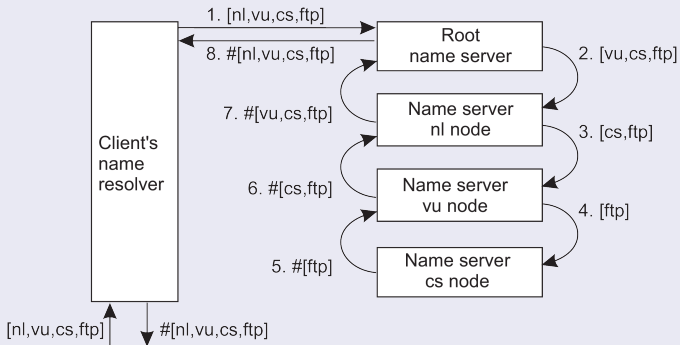
- Resolução de Nomes Iterativa
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html



# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

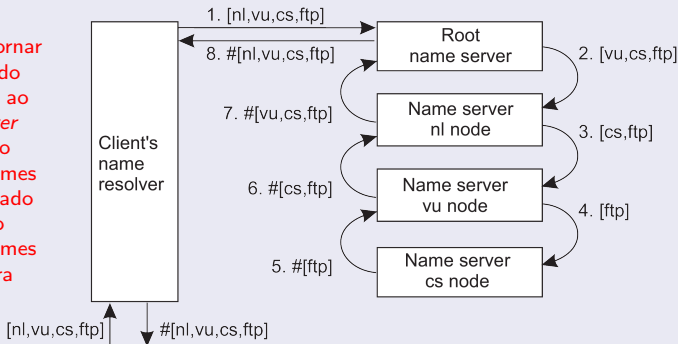


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

Em vez de retornar cada resultado intermediário ao *name resolver* do cliente, o servidor de nomes passa o resultado ao próximo servidor de nomes que encontra

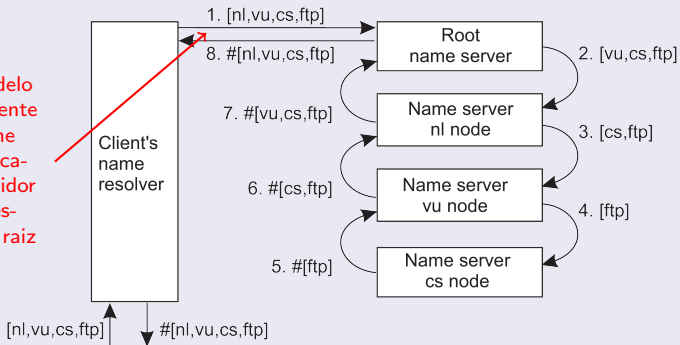


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

Como no modelo iterativo, o cliente envia o nome completo do caminho ao servidor de nomes responsável pela raiz

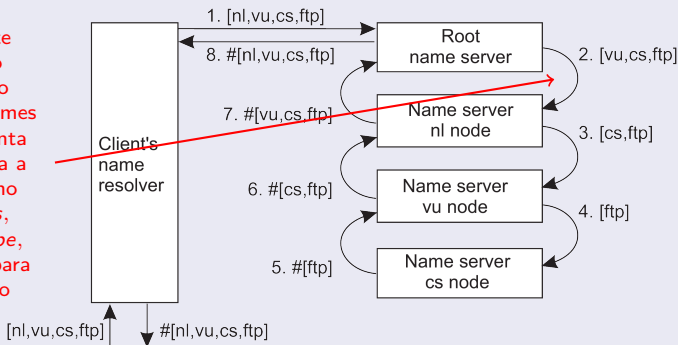


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

Quando este encontra o endereço do servidor de nomes que implementa o nó *nl*, passa a ele o caminho *nl* : [vu, cs, ftp, pub, globe, index.html] para ser resolvido

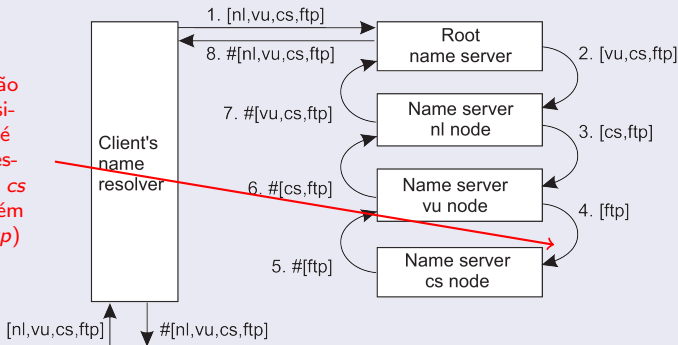


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

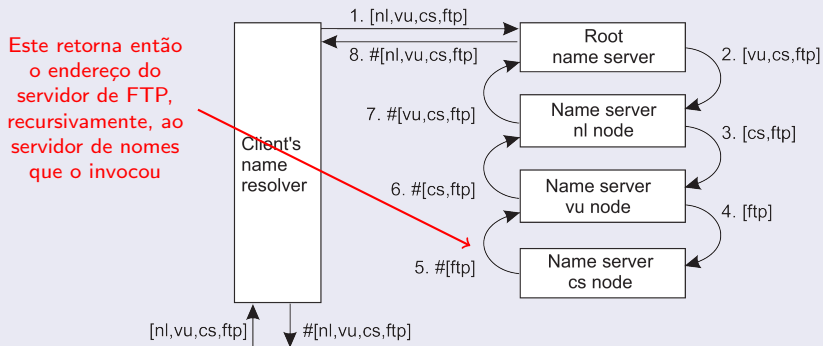
E a requisição desce recursivamente até o servidor responsável por *cs* (e que também reconhece *ftp*)



# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html





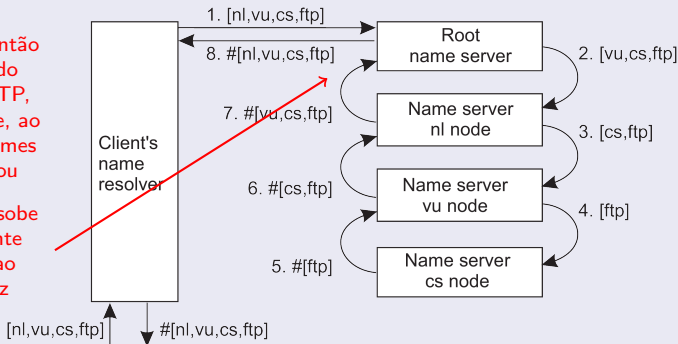
# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

Este retorna então o endereço do servidor de FTP, recursivamente, ao servidor de nomes que o invocou

E o endereço sobe recursivamente até chegar ao servidor raiz

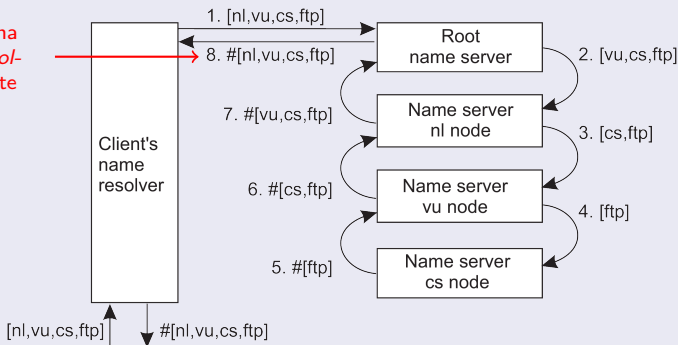


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

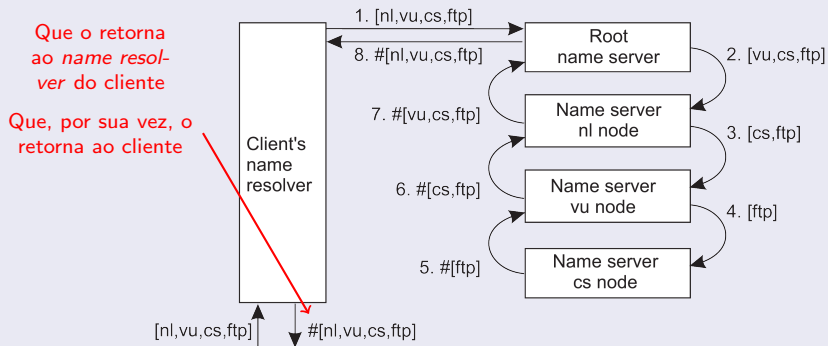
Que o retorna  
ao *name resolver*  
do cliente



# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html



# Nomes Estruturados

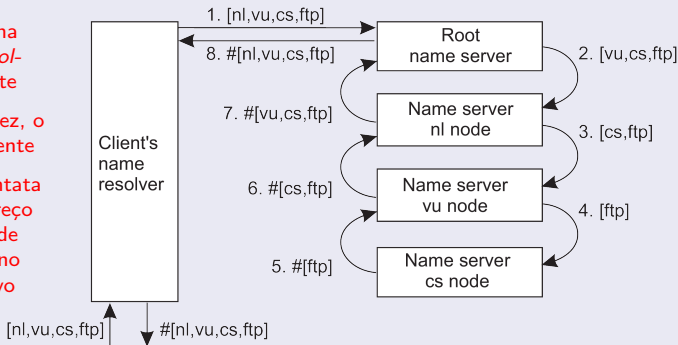
## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Recursiva
  - Ex: ftp://ftp.cs.vu.nl/pub/globe/index.html

Que o retorna  
ao *name resolver*  
do cliente

Que, por sua vez, o  
retorna ao cliente

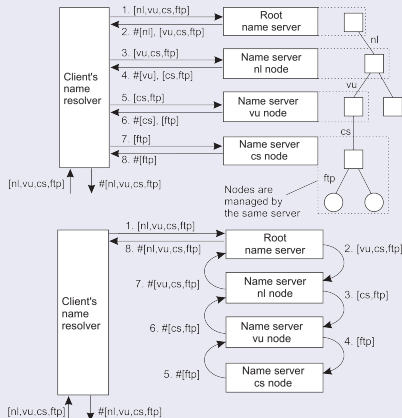
E o cliente contata  
então o endereço  
do servidor de  
FTP, como no  
caso iterativo



# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa × Recursiva

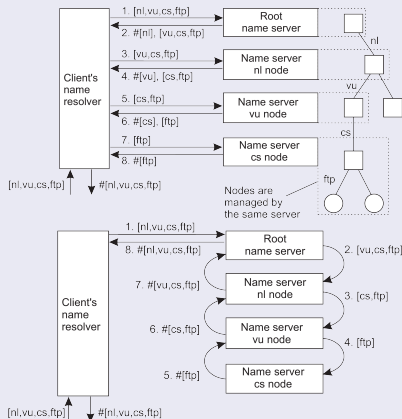


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa × Recursiva

- Resolução recursiva exige mais de cada servidor de nomes
- Eles têm que tratar de toda a resolução, ainda que com a ajuda de outros
- Exigência tão pesada que servidores na camada global suportam apenas resolução iterativa

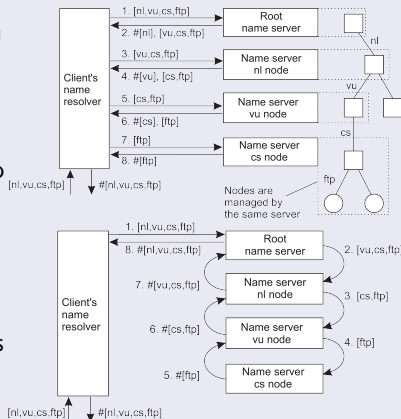


# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

### ● Resolução de Nomes Iterativa × Recursiva

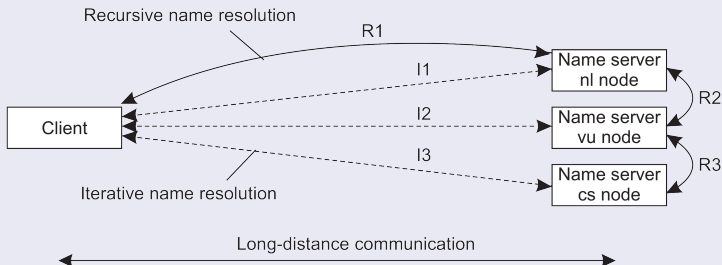
- Resolução recursiva permite a existência de cache em cada servidor, melhorando o desempenho
- Na iterativa o cache está restrito ao cliente
- Se um cliente fizer uma requisição, e outro fizer a mesma logo depois, todos os servidores terão de ser acessados como se fosse a primeira vez



# Nomes Estruturados

## Implementação de EN: Resolução de Nomes

- Resolução de Nomes Iterativa × Recursiva
- Resolução recursiva é frequentemente mais barata com respeito à comunicação
- Caso os servidores estejam próximos e o cliente longe – os custos são ditados pela comunicação entre cliente e servidor





# Abordagens de Nomeação

- **Nomes Estruturados**
  - Espaço de Nomes
  - Resolução de Nomes
  - Implementação de Espaços de Nomes
- **Nomeação Baseada em Atributos (Hierárquica)**

# Nomeação Baseada em Atr. (Hierárquica)

## Serviços de Diretório

- Nomes planos e estruturados fornecem um modo de referenciar entidades único e independente de localização
- Em muitos casos, é mais conveniente nomear e procurar entidades pelos seus **atributos**

# Nomeação Baseada em Atr. (Hierárquica)

## Serviços de Diretório

- Nomes planos e estruturados fornecem um modo de referenciar entidades único e independente de localização
- Em muitos casos, é mais conveniente nomear e procurar entidades pelos seus **atributos**
- Isso requer que o usuário forneça apenas uma descrição do que busca
  - Que descreva uma entidade em termos de pares (*atributo, valor*)
  - Conhecidos como **nomeação baseada em atributos**

# Nomeação Baseada em Atr. (Hierárquica)

## Serviços de Diretório

- Ao especificar os valores de um atributo, o usuário restringe o conjunto de entidades de interesse
- Cabe ao sistema de nomes retornar as entidades que casam com a descrição feita pelo usuário

# Nomeação Baseada em Atr. (Hierárquica)

## Serviços de Diretório

- Ao especificar os valores de um atributo, o usuário restringe o conjunto de entidades de interesse
- Cabe ao sistema de nomes retornar as entidades que casam com a descrição feita pelo usuário
- Sistemas de nomeação baseada em atributos são também chamados de **serviços de diretório**
- Em contraste a sistemas que suportam nomeação estruturada, chamados **sistemas de nomeação**

# Nomeação Baseada em Atr. (Hierárquica)

## Serviços de Diretório

- Ao especificar os valores de um atributo, o usuário restringe o conjunto de entidades de interesse
  - Cabe ao sistema de nomes retornar as entidades que casam com a descrição feita pelo usuário
- Sistemas de nomeação baseada em atributos são também chamados de **serviços de diretório**
  - Em contraste a sistemas que suportam nomeação estruturada, chamados **sistemas de nomeação**
  - Com serviços de diretório, entidades possuem um conjunto de atributos associados que podem ser usados para busca

# Nomeação Baseada em Atr. (Hierárquica)

## Serviços de Diretório – Problema

- Operações de consulta pode ser muito caras
  - Necessitam que os valores dos atributos procurados correspondam aos valores reais das entidades
  - Em princípio, teríamos que inspecionar todas as entidades.

# Nomeação Baseada em Atr. (Hierárquica)

## Serviços de Diretório – Problema

- Operações de consulta pode ser muito caras
  - Necessitam que os valores dos atributos procurados correspondam aos valores reais das entidades
  - Em princípio, teríamos que inspecionar todas as entidades.

## Serviços de Diretório – Solução

Implementar serviços de diretórios básicos na forma de bases de dados, e combiná-los com os sistemas de nomes estruturados tradicionais



# Nomeação Baseada em Atr. (Hierárquica)

## *Lightweight Directory Access Protocol – LDAP*

- Conceitualmente, um serviço de diretório LDAP consiste de uma quantidade de registros
  - Conhecidos como **entradas de diretório**

# Nomeação Baseada em Atr. (Hierárquica)

## *Lightweight Directory Access Protocol – LDAP*

- Conceitualmente, um serviço de diretório LDAP consiste de uma quantidade de registros
  - Conhecidos como **entradas de diretório**
- Cada entrada de diretório consiste de uma coleção de pares (*atributo, valor*)
  - Em que cada atributo possui um tipo associado
  - Cada entrada possui um **nome único**, para facilitar buscas

# Nomeação Baseada em Atr. (Hierárquica)

## *Lightweight Directory Access Protocol – LDAP*

- Exemplo de uma entrada de diretório LDAP:

Attribute	Abbr.	Value
Country	<i>C</i>	NL
Locality	<i>L</i>	Amsterdam
Organization	<i>O</i>	VU University
OrganizationalUnit	<i>OU</i>	Computer Science
CommonName	<i>CN</i>	Main server
Mail_Servers	–	137.37.20.3, 130.37.24.6, 137.37.20.10
FTP_Server	–	130.37.20.20
WWW_Server	–	130.37.20.20

# Nomeação Baseada em Atr. (Hierárquica)

## *Lightweight Directory Access Protocol – LDAP*

- Base de Informação de Diretório
  - *Directory information base* (DIB)
  - Coleção de todas as entradas de diretório em um serviço de diretório LDAP

# Nomeação Baseada em Atr. (Hierárquica)

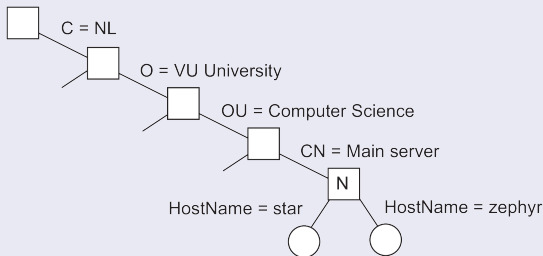
## *Lightweight Directory Access Protocol – LDAP*

- Base de Informação de Diretório
  - *Directory information base* (DIB)
  - Coleção de todas as entradas de diretório em um serviço de diretório LDAP
- Cada registro na DIB é nomeado de forma única, permitindo sua busca
  - Na forma de uma sequência de atributos de nomeação
  - Chamados *Relative Distinguished Name* (RDN)
  - Formam um nome único global

# Nomeação Baseada em Atr. (Hierárquica)

## *Lightweight Directory Access Protocol – LDAP*

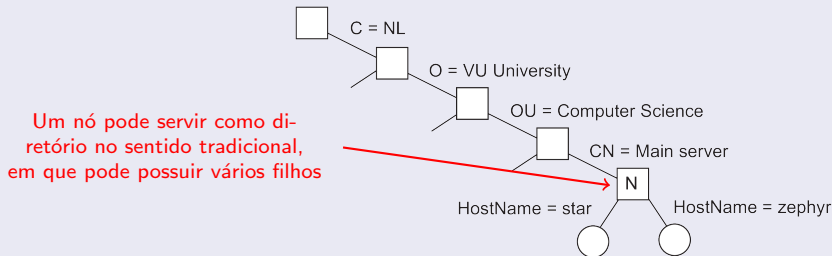
- Isso leva a uma hierarquia de entradas de diretório
  - *Directory Information Tree (DIT)*
  - Grafo de nomes de um serviço de diretório LDAP, no qual cada nó representa uma entrada no diretório



# Nomeação Baseada em Atr. (Hierárquica)

## *Lightweight Directory Access Protocol – LDAP*

- Isso leva a uma hierarquia de entradas de diretório
  - *Directory Information Tree (DIT)*
  - Grafo de nomes de um serviço de diretório LDAP, no qual cada nó representa uma entrada no diretório



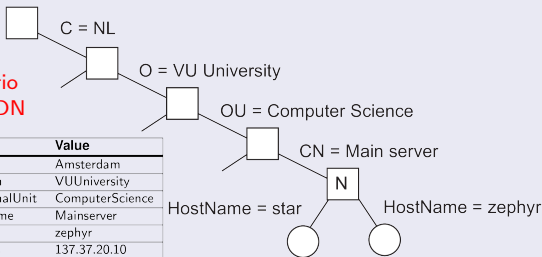
# Nomeação Baseada em Atr. (Hierárquica)

## Lightweight Directory Access Protocol – LDAP

- Isso leva a uma hierarquia de entradas de diretório
  - *Directory Information Tree* (DIT)
  - Grafo de nomes de um serviço de diretório LDAP, no qual cada nó representa uma entrada no diretório

Ex: 2 entradas no diretório  
tendo HostName como RDN

Attribute	Value	Attribute	Value
Locality	Amsterdam	Locality	Amsterdam
Organization	VUUniversity	Organization	VUUniversity
OrganizationalUnit	ComputerScience	OrganizationalUnit	ComputerScience
CommonName	Mainserver	CommonName	Mainserver
HostName	star	HostName	zephyr
HostAddress	192.31.231.42	HostAddress	137.37.20.10





# Nomeação Baseada em Atr. (Hierárquica)

## Lightweight Directory Access Protocol – LDAP

- Isso leva a uma hierarquia de entradas de diretório
  - *Directory Information Tree* (DIT)
  - Grafo de nomes de um serviço de diretório LDAP, no qual cada nó representa uma entrada no diretório

Ex: 2 entradas no diretório tendo HostName como RDN

Attribute	Value	Attribute	Value
Locality	Amsterdam	Locality	Amsterdam
Organization	VUUniversity	Organization	VUUniversity
OrganizationalUnit	ComputerScience	OrganizationalUnit	ComputerScience
CommonName	Mainserver	CommonName	Mainserver
HostName	star	HostName	zephyr
HostAddress	192.31.231.42	HostAddress	137.37.20.10

