Naming in Distributed Systems

Hong-Linh Truong
Distributed Systems Group,
Vienna University of Technology

truong@dsg.tuwien.ac.at
dsg.tuwien.ac.at/staff/truong

1

DISTRIBUTED SYSTEMS GROUP

What is this lecture about?

- Understand how to create names/identifiers for entities in distributed systems

- Understand how to manage and resolve names to provide further detailed information about entities

- Examine main techniques/frameworks/services for the creation and management of names in distributed systems

- Try to see the "fundamentals" in state-of-the art distributed systems

DISTRIBUTED SYSTEMS GROUP

Learning Materials

- Main reading:
    - Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall
        - Chapter 5
    - George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair„Distributed Systems – Concepts and Design", 5nd Edition
        - Chapters 10 & 13

- Examples and questions in the lecture

DISTRIBUTED SYSTEMS GROUP

Outline

- Basic concepts and design principles
- Flat naming
- Structured naming
- Attribute-based naming
- Some naming systems in the Web
- Summary

DISTRIBUTED SYSTEMS GROUP

BASIC CONCEPTS AND DESIGN PRINCIPLES

DISTRIBUTED SYSTEMS GROUP

If your distributed system has to manage information about a marathon

DISTRIBUTED SYSTEMS GROUP

People
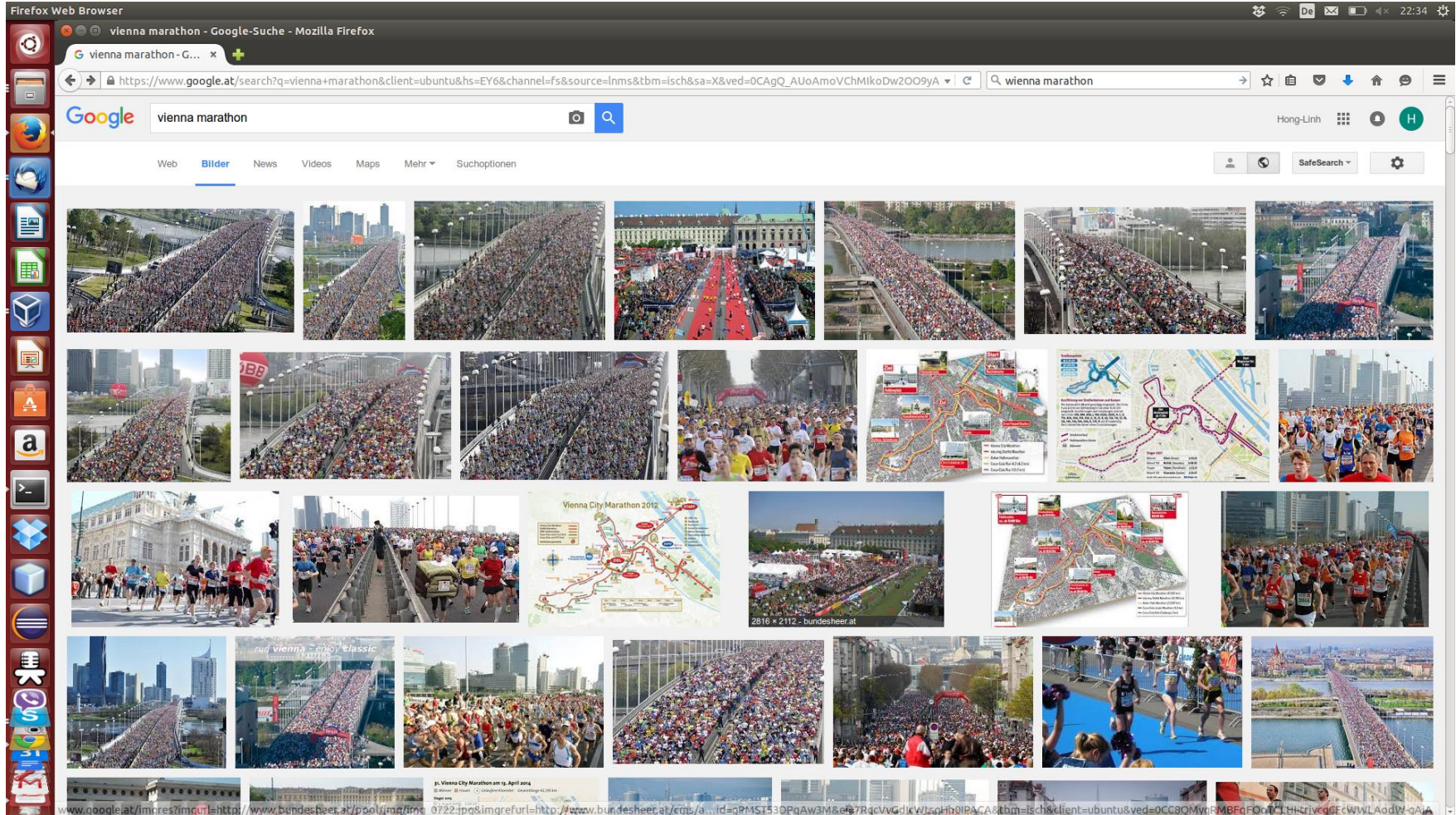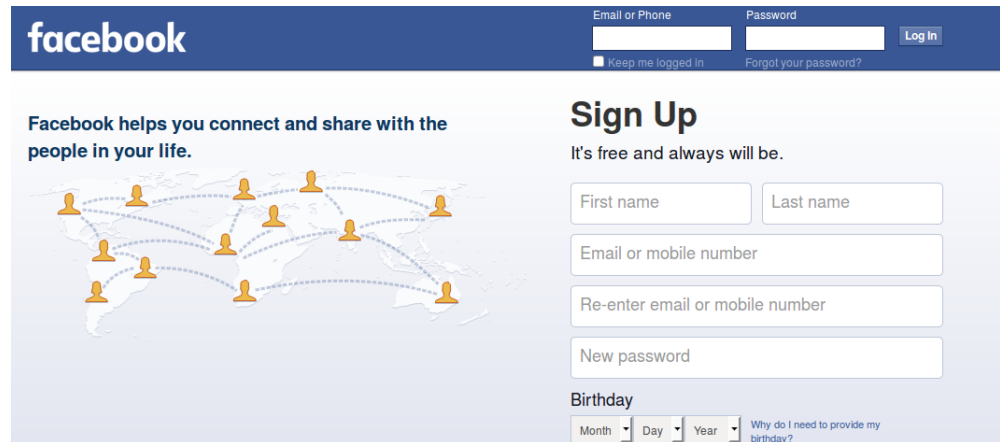
Front lobby entrance of building 17, one of the largest buildings on Microsoft's main campus in Redmond

Type	Public
Traded as	NASDAQ: MSFT Dow Jones Industrial Average Component NASDAQ-100 Component S&P 500 Component
Industry	Computer software Consumer electronics Computer hardware
Founded	April 4, 1975; 40 years ago Albuquerque, New Mexico, U.S.
Founders	Bill Gates, Paul Allen
Headquarters	Microsoft Redmond Campus, Redmond, Washington, U.S.
Area served	Worldwide
Key people	John W. Thompson (Chairman) Satya Nadella (CEO) Bill Gates (founder, technology advisor) Brad Smith (President)[1]
Products	Windows · Office · Servers · Skype · Visual Studio · Dynamics · Azure · Xbox · Surface · Mobile · more...
Services	MSN · Bing · OneDrive · MSDN · Outlook.com · TechNet · Xbox Live
Revenue	▲ US$ 93.58 billion (2015)[2]
Operating income	▼ US$ 18.16 billion (2015)[2]
Net income	▼ US$ 12.19 billion (2015)[2]
Total assets	▲ US$ 176.22 billion (2015)[2]
Total equity	▼ US$ 80.08 billion (2015)[2]
Number of employees	118,584 (March 2015)[3]
Subsidiaries	List of Microsoft subsidiaries
Website	Microsoft.com

"968 million daily active users on average for June 2015"

Sources:
http://newsroom.fb.com/company-info/
https://en.wikipedia.org/wiki/Microsoft

DISTRIBUTED SYSTEMS GROUP

Songs or things (Internet of Things)

What is Spotify?

Spotify brings you the right music for every moment – on computers, mobiles, tablets, home entertainment systems, cars, gaming consoles and more.

Just search for music you love, or let Spotify play you something great. Create and listen to your playlists for free or subscribe to Premium for on-demand access at the highest audio quality – with zero ads.

Fast facts

- Subscribers: Over 20 million
- Active users: Over 75 million
- Revenue paid to rightsholders: $3bn
- Number of songs: Over 30 million
- Number of playlists: Over 1.5 billion
- Available in 58 markets – Andorra, Argentina, Austria, Australia, Belgium, Bolivia, Brazil, Bulgaria, Canada, Chile, Colombia, Costa Rica, Cyprus, Czech Republic, Denmark, Dominican Republic, Ecuador, El Salvador, Estonia, Finland, France, Germany, Greece, Guatemala, Honduras, Hong Kong, Hungary, Iceland, Ireland, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malaysia, Malta, Mexico, Monaco, New Zealand, Netherlands, Nicaragua, Norway, Panama, Paraguay, Peru, Philippines, Poland, Portugal, Singapore, Slovakia, Spain, Sweden, Switzerland, Taiwan, Turkey, UK, Uruguay and USA.

Source: https://press.spotify.com/au/information/

8
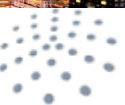
Q: Can you list some entities that are relevant to the previous mentioned distributed systems?

So how do we identify such a large number of entities?

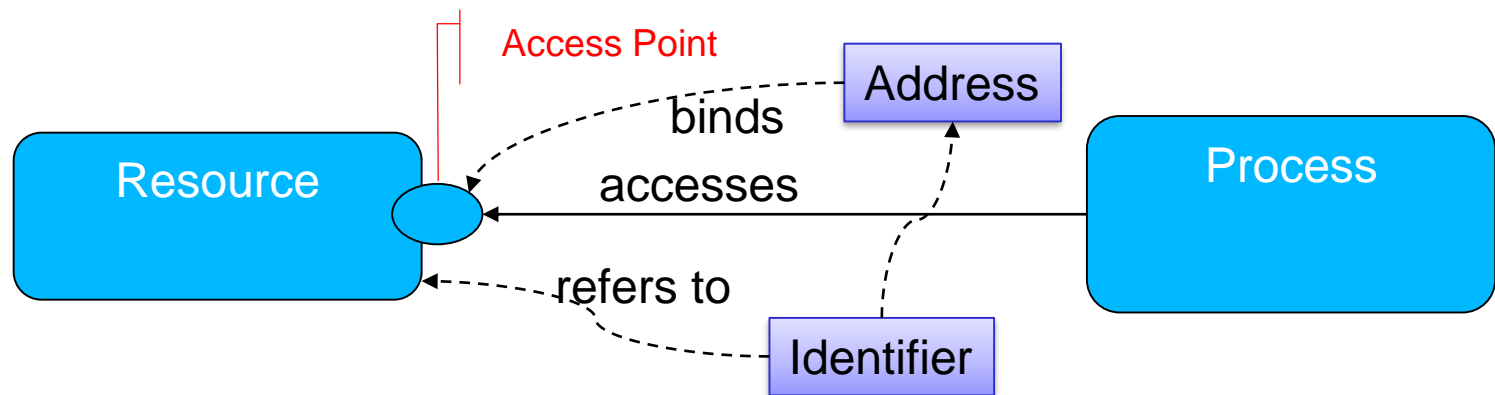And why?

9

Why naming systems are important?

- **Entity:** any kind of objects we see in distributed systems: process, file, printer, host, communication endpoint, etc

- The **usefulness** of naming services

  - Identification

  - Providing detailed description

  - Foundations for communication, security, auditing, etc.

DISTRIBUTED SYSTEMS GROUP

Why naming systems are complex?

- Diverse types of and complex dependencies among entities at different levels
  - E.g, printing service → the network level communication endpoints → the data link level communication endpoints
- There are just so many entities, how do we create and manage names and identify entities?

DISTRIBUTED SYSTEMS GROUP

Names, identifiers, and addresses

- **Name**: set of bits/characters used to identify/refer to an entity, a collective of entities, etc. in a context
  - Simply comparing two names, we might not be able to know if they refer to the same entity
- **Identifier**: a name that uniquely identifies an entity
  - the identifier is unique and refers to only one entity
- **Address**: the name of an access point, the location of an entity

Access Point

Resource — binds — Address — accesses — Process

refers to — Identifier

DISTRIBUTED SYSTEMS GROUP

Naming design principles

- Data models/structures for naming services
  - information about names
  - Strongly related to database topics

- Processes in naming services
  - E.g., Creation, management, update, query, and resolution activities
  - Interaction protocols
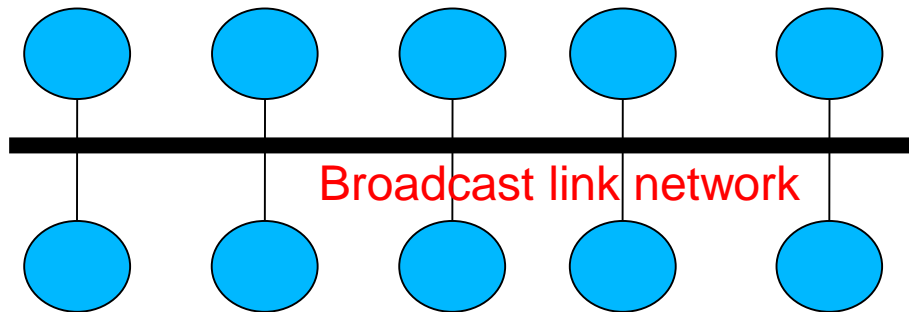
DISTRIBUTED SYSTEMS GROUP

Naming design principles

- **Name space**
  - Contains all valid names recognized and managed by a service
    - A valid name might not be bound to any entity
    - Alias: a name refers to another name
  - Naming domain
    - Name space with a single administrative authority which manages names for the name space

- **Name resolution**
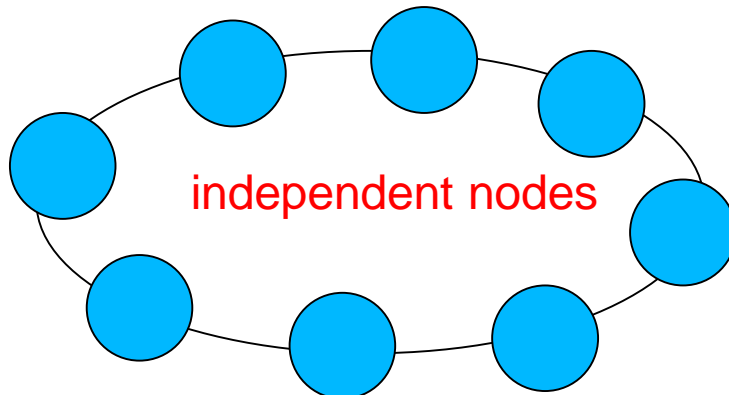 - A process to look up information/attributes from a name

DISTRIBUTED SYSTEMS GROUP

Naming design principles

- Naming design is based on specific system organizations and characteristics

Examples

- Network ←→Ethernet
- Identifier: IP and MAC address
- Name resolution: the network address to the data link address

Broadcast link network

independent nodes

- P2P systems
- Identifier: m-bit key
- Name resolution: distributed hash tables
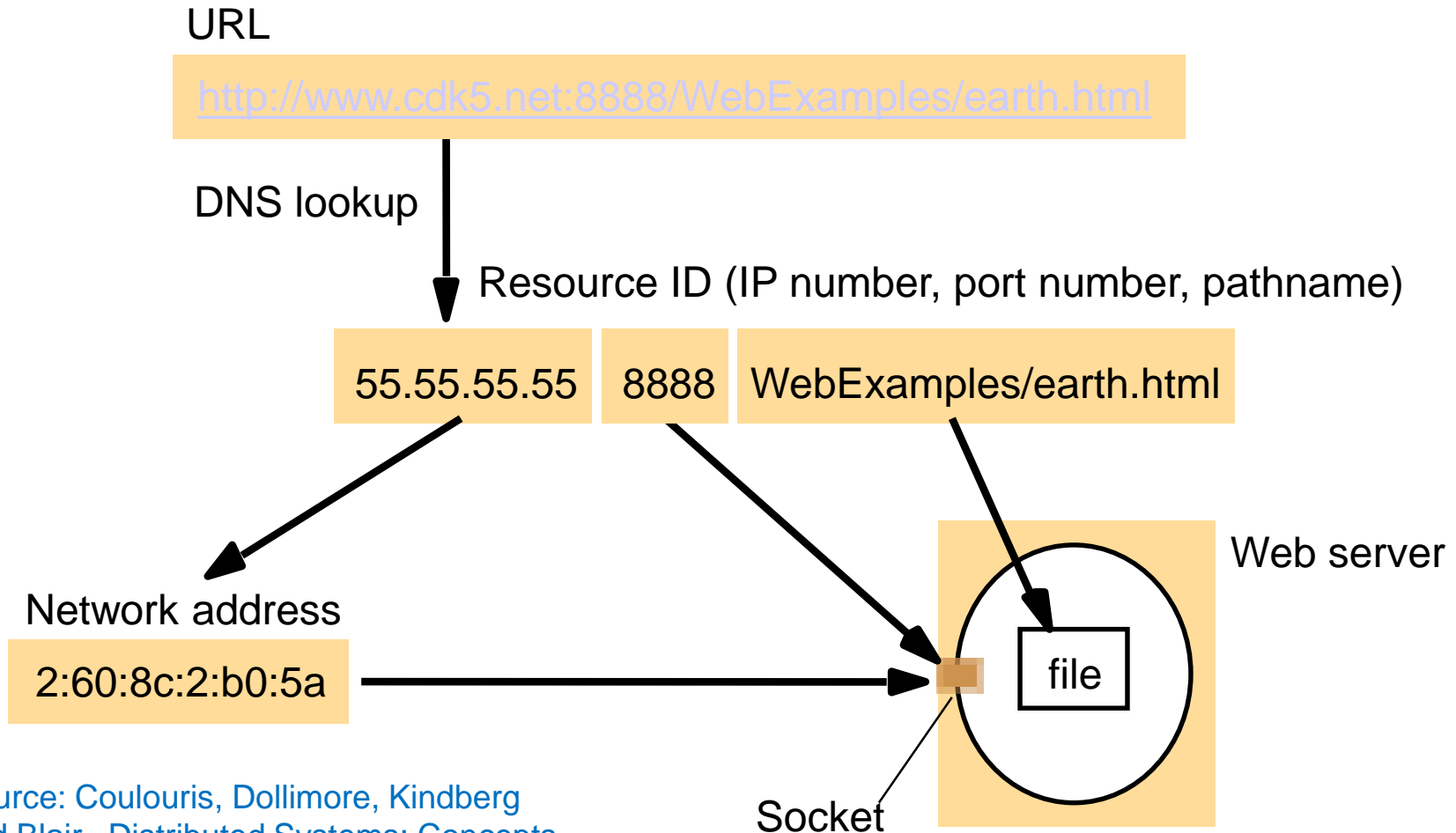
DISTRIBUTED SYSTEMS GROUP

Naming design principles

- Structures and characteristics of names are based on different purposes

- Data structure:

  - Can be simple, no structure at all, e.g., a set of bits:
    $ uuid
    bcff7102-3632-11e3-8d4a-0050b6590a3a

  - Can be complex

    - Include several data items to reflect different aspects on a single entity

  - Readability:

    - Human-readable or machine-processable formats

DISTRIBUTED SYSTEMS GROUP

Naming design principles

- **Diverse name-to-address binding mechanisms**
  - How a name is associated with an address or how an identifier is associated with an entity
  - Names can be changed over the time and names are valid in specific contexts
    - Dynamic or static binding?
- **Distributed or centralized management**
  - Naming data is distributed over many places or not
- **Discovery/Resolution protocol**
  - Names are managed by distributed services
  - Noone/single system can have a complete view of all names

DISTRIBUTED SYSTEMS GROUP

Examples of relationships among different names/identifiers

URL

http://www.cdk5.net:8888/WebExamples/earth.html

DNS lookup

Resource ID (IP number, port number, pathname)

| 55.55.55.55 | 8888 | WebExamples/earth.html |

Network address

2:60:8c:2:b0:5a

Web server

file

Socket

Source: Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design   Edn. 5

DISTRIBUTED SYSTEMS GROUP

FLAT NAMING

Flat naming

Unstructured/flat names: identifiers have no structured description, e.g., just a set of bits

- Simple way to represent identifiers
- Do not contain additional information for understanding the entity
- Examples
    - Internet Address at the Network layer
    - m-bit numbers in Distributed Hash Tables

Q: For which types of systems flat naming is suitable

DISTRIBUTED SYSTEMS GROUP

Broadcast based Name Resolution

- Principles
  - Assume that we want find the access point of the entity en
  - Broadcast the identifier of en, e.g., broadcast(ID(en))
  - Only en will return the access point, when the broadcast message reaches nodes
- Examples
  - ARP: from IP address to MAC address (the datalink access point)

```
mail.infosys.tuwien.ac.at (128.131.172.240) at 00:19:b9:f2:07:55 [ether] on eth0
sw-ea-1.kom.tuwien.ac.at (128.131.172.1) at 00:08:e3:ff:fc:c8 [ether] on eth0
```

DISTRIBUTED SYSTEMS GROUP

Dynamic systems

- A set of nodes is used to manage entities

- Nodes form a system which has no centralized coordination

  - In an overlay network

- Nodes can join/leave/fail anytime

- A large number of nodes but a node knows only a subset of nodes

- Examples

  - Large-scale p2p systems, e.g., Chord, CAN (Content Addressable Network), and Pastry

How do we define identifiers for such a system?

DISTRIBUTED SYSTEMS GROUP

Distributed Hash Tables

- Main concepts
    - m-bit is used for the keyspace for identifiers (of nodes and entities)
    - (Processing) Node identifier nodeID is one key in the keyspace
    - An entity en is identified by a hash function k=hash(en)
    - A node with ID p is responsible for managing entities whose keys are within a range of keys
        - If (k=hash(en) ∈ range(p)), then put (k, en) will store en in p
    - Nodes will relay messages (including entities/name resolution requests) till the messages reach the right destination

Example - Chord

- A ring network with $[0…2^m − 1]$ positions for nodes in clockwise

- nodeID = hash(IP)

- the successor of k, successor(k), is the smallest node identifier that $\geq$k (in mod 2^m)

- A key k of entity en will be managed by the first node p where p =successor(k)$\geq$ k=hash(en)

  - the first node clockwise from k
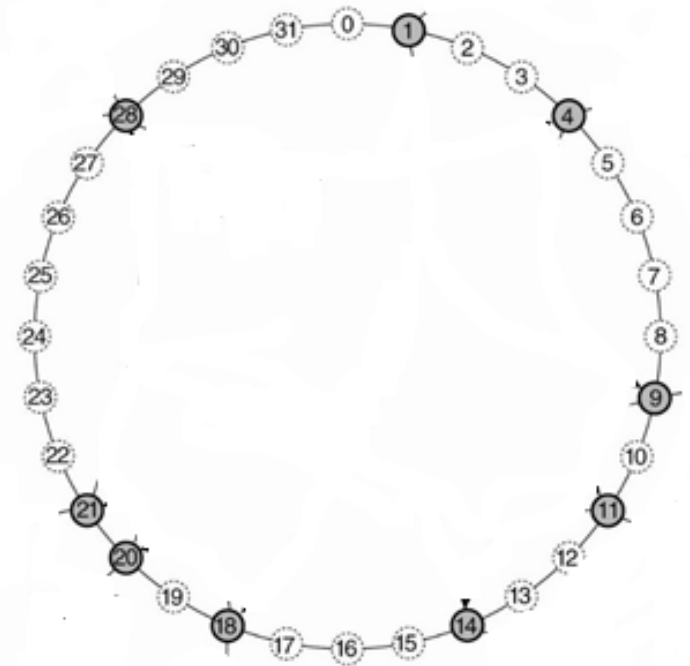
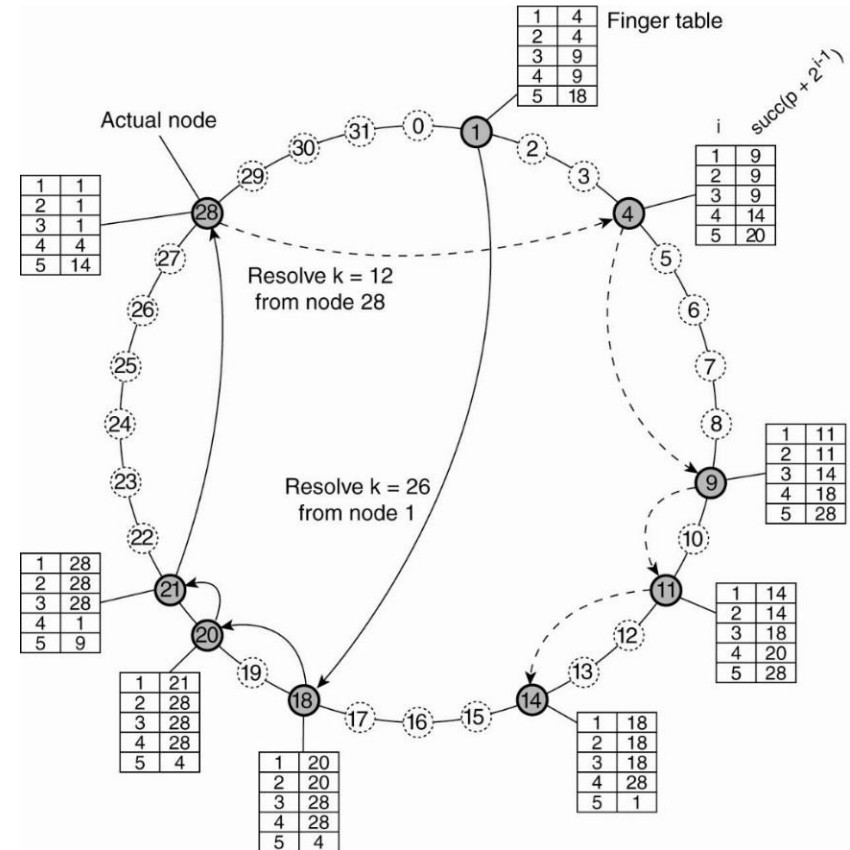Q: if you want to manage files in 8 computers, how many bits would you use for the keyspace? ☺

http://pdos.csail.mit.edu/papers/chord:sigcomm01/

DISTRIBUTED SYSTEMS GROUP

Homework – understanding Chord

- Resolving at p

  - Keep m entries in a finger table FT

 $FT_p[i]$
    $= (successor(p$

Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall

STRUCTURED NAMING

DISTRIBUTED SYSTEMS GROUP

Name spaces

- Names are organized into a name space which can be modeled as a graph:
    - Leaf node versus directory node
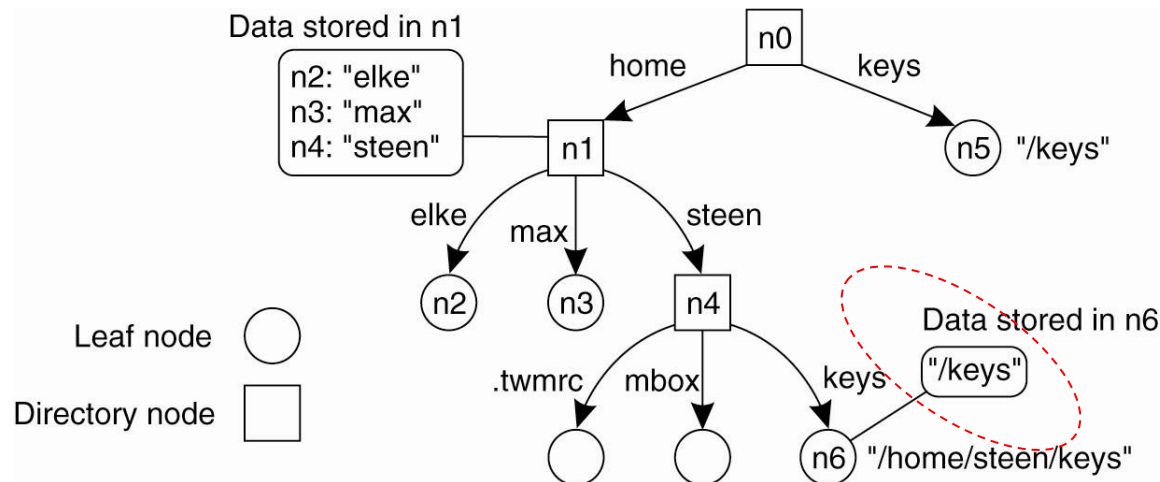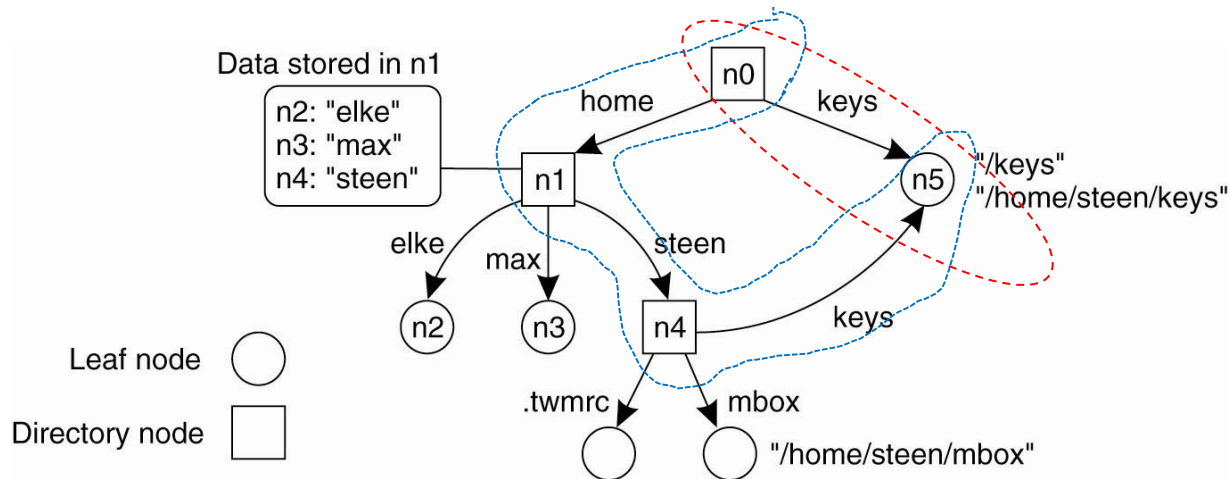- Each leaf node represents an entity; nodes are also entities

Directory table (label,identifier)

Data stored in n1

An absolute path name

An relative path name

Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall
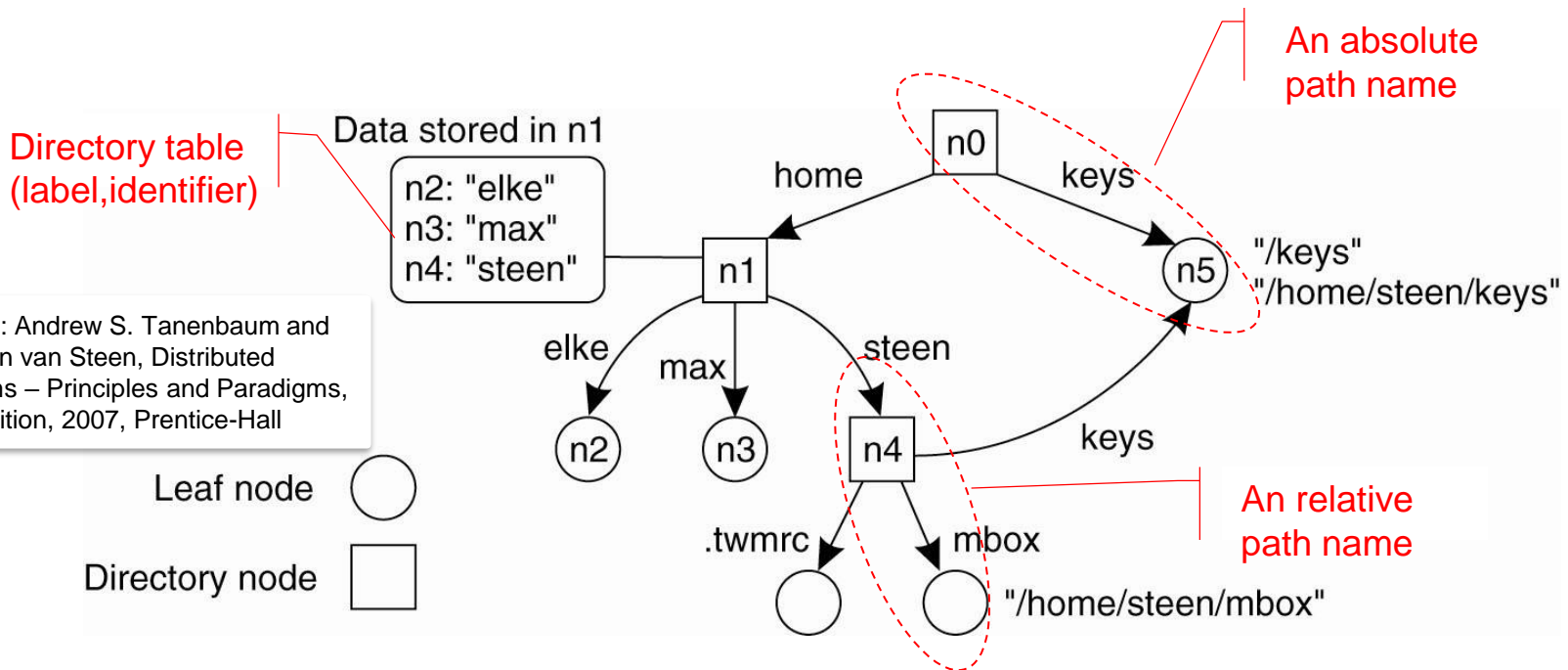
Name resolution – Closure Mechanism

- Name resolution:
  N:<label1,label2,label3,…labeln>
    - Start from node N
    - Lookup (label1,identifier1) in N's directory table
    - Lookup (label2, identifier2) in identifier1's directory table
    - and so on

Closure Mechanism: determine where and how name resolution would be started

- E.g., name resolution for /home/truong/ds.txt ?

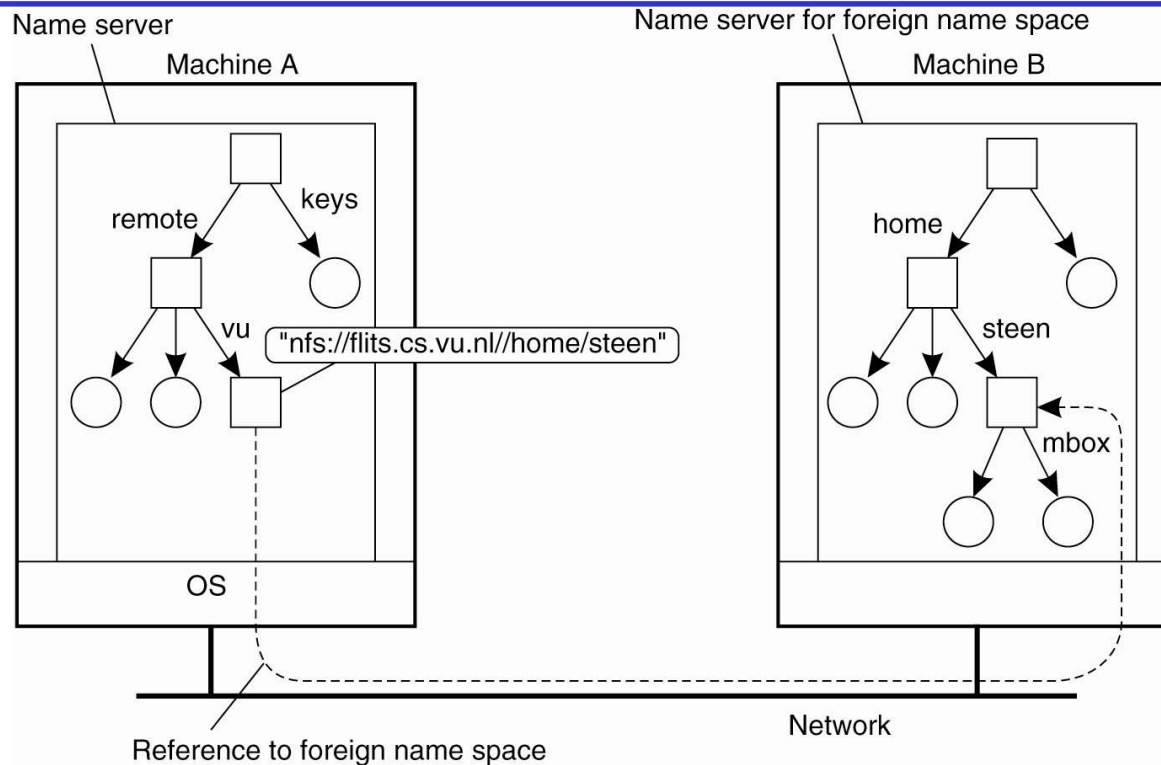- Or for https://me.yahoo.com/a/.....

DISTRIBUTED SYSTEMS GROUP

Enabling Alias Using Links

Hard links: multiple absolute paths names referring to the same node

Symbolic links: leaf node storing an absolute path name

Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall

Name resolution - Mounting

- A directory node (mounting point) in a remote server can be mounted into a local node (mount point)

Name server

Machine A

Name server for foreign name space

Machine B

remote   keys

vu   "nfs://flits.cs.vu.nl//home/steen"

OS

Reference to foreign name space

home

steen

mbox

Network

e.g., NFS and Samba

Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall

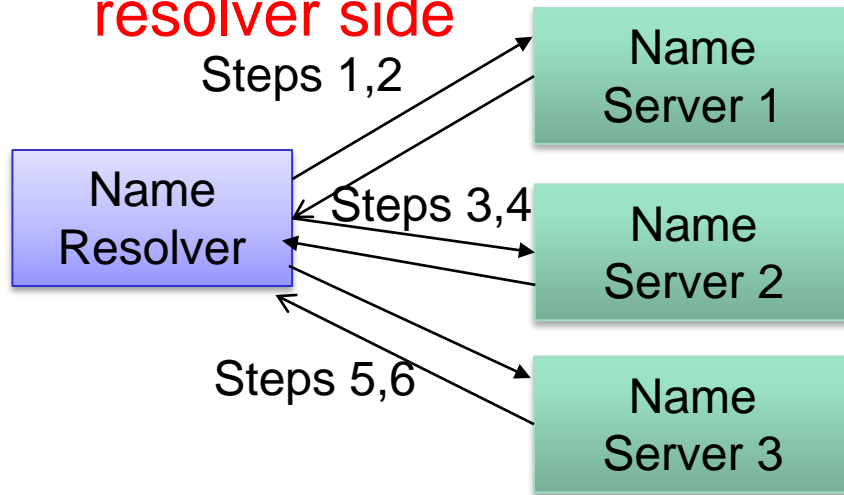DISTRIBUTED SYSTEMS GROUP

Name space implementation

- **Distributed name management**
  - Several servers are used for managing names
- **Many distribution layers**
  - **Global layer:** the root node and its close nodes
  - **Administrational layer:** directory nodes managed within a single organization
  - **Managerial layer:** nodes typically change regularly.

Example in Domain Name System
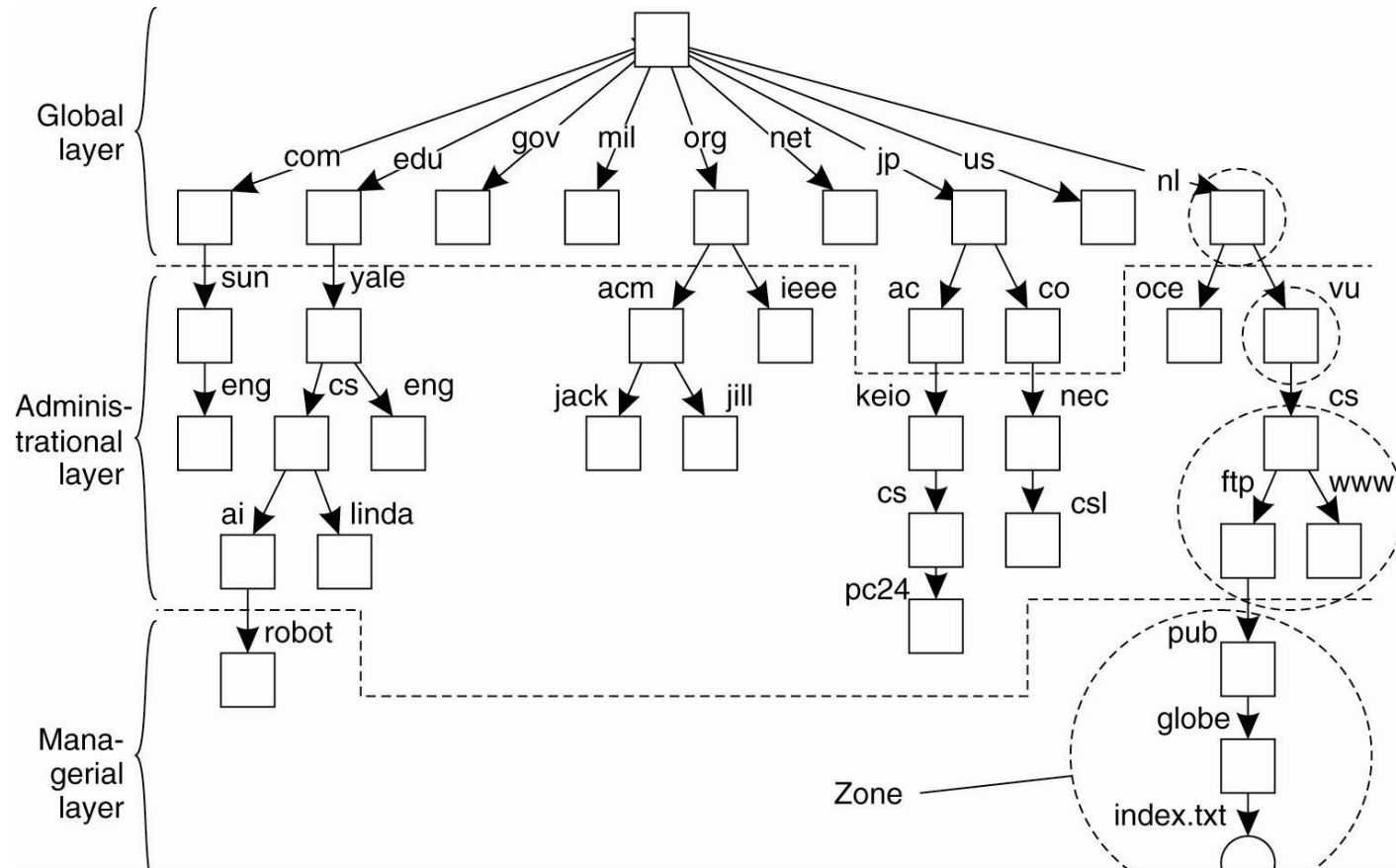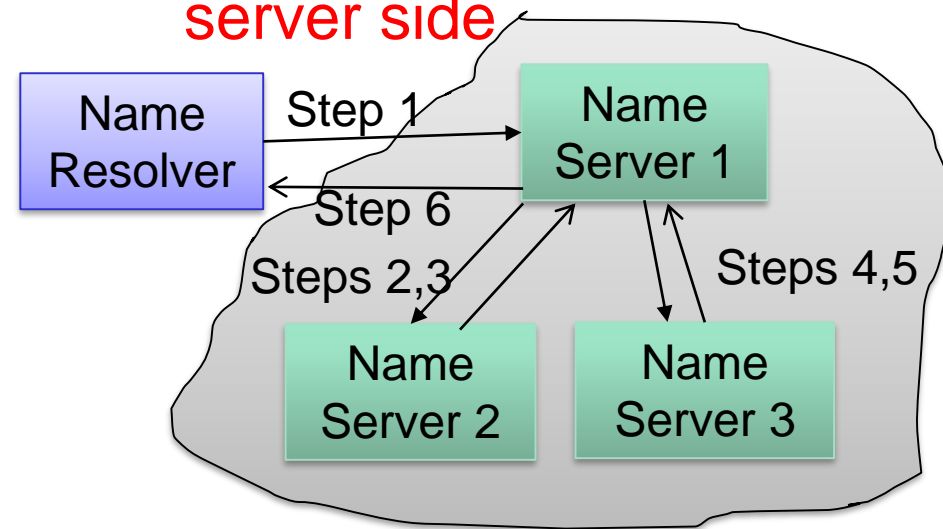

Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall
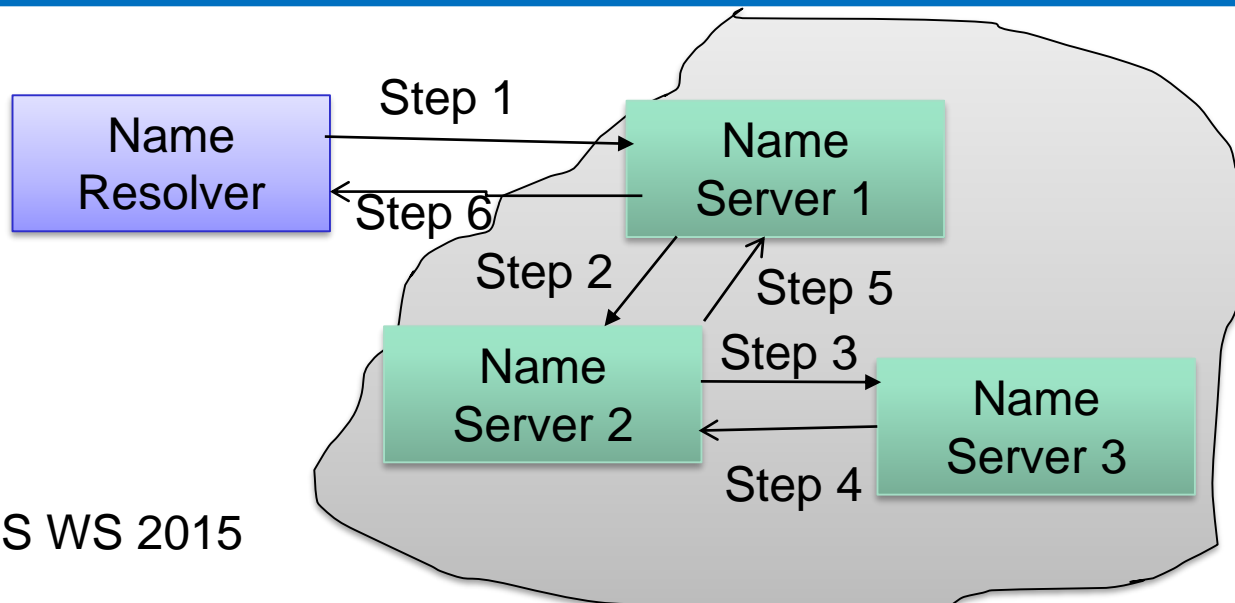
Characteristics of distribution layers

| Item	Global	Administrational	Managerial
Geographical scale of network	Worldwide	Organization	Department
Total number of nodes	Few	Many	Vast numbers
Responsiveness to lookups	Seconds	Milliseconds	Immediate
Update propagation	Lazy	Immediate	Immediate
Number of replicas	Many	None or few	None
Is client-side caching applied?	Yes	Yes	Sometimes

Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall

DISTRIBUTED SYSTEMS GROUP

Name Resolution

Iterative name resolution at resolver side

Steps 1,2

Steps 3,4

Steps 5,6

Name Resolver

Name Server 1

Name Server 2

Name Server 3

Iterative name resolution at server side

Name Resolver

Step 1

Step 6

Steps 2,3

Steps 4,5

Name Server 1

Name Server 2

Name Server 3

Name Resolver

Step 1

Step 6

Step 2

Step 5

Step 3

Step 4

Name Server 1

Name Server 2

Name Server 3

Recursive name resolution

DS WS 2015

DISTRIBUTED SYSTEMS GROUP

Example -- Iterative name resolution
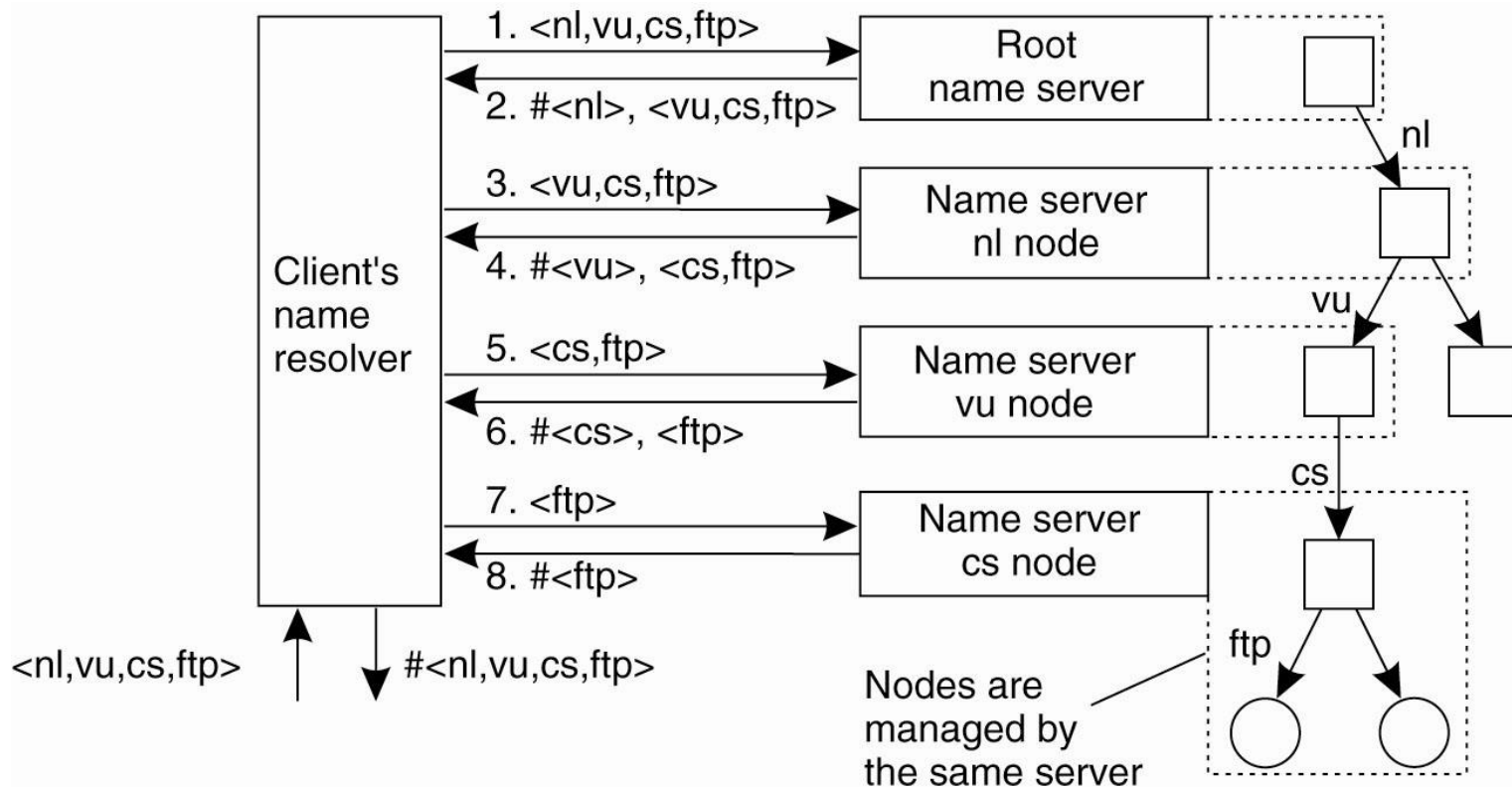
Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall

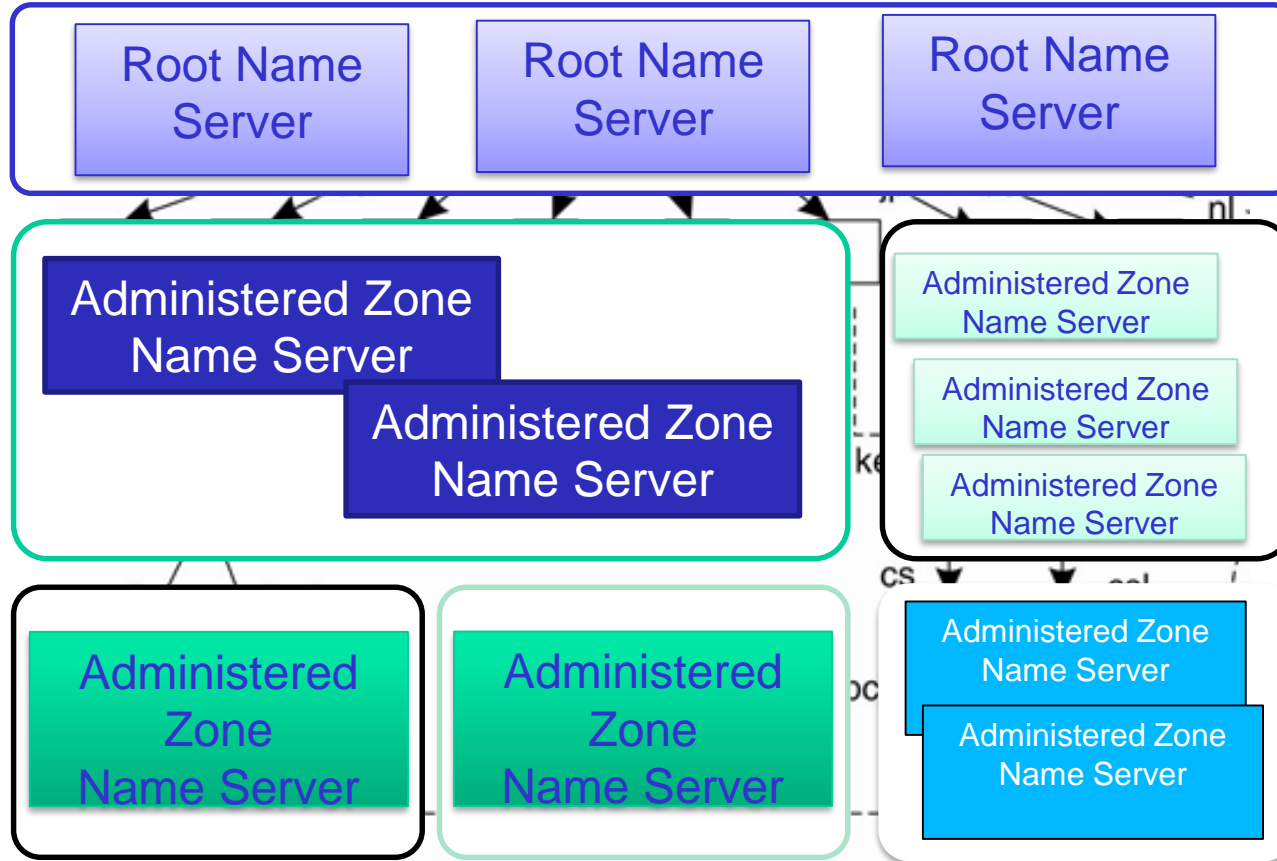# Example -- Recursive name resolution
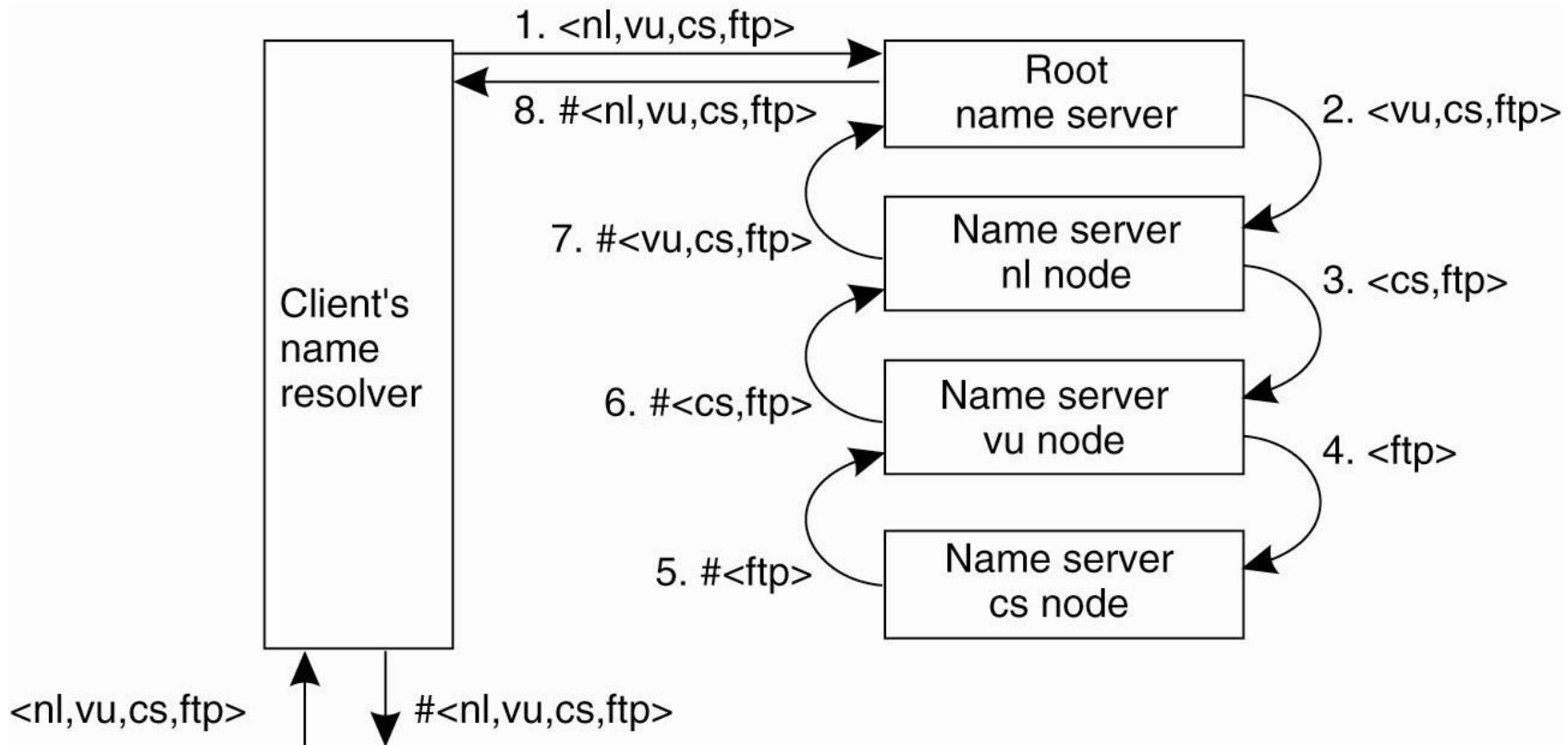
Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall

Q: What are pros and cons of the recursive name resolution?

EXAMPLE OF INTERNET DOMAIN NAM SYSTEM (DNS)

DISTRIBUTED SYSTEMS GROUP

Domain Name System (DNS) in Internet

- We use to remember „human-readable" machine name → we have the name hierarchy

    - E.g., www.facebook.com

- But machines in Internet use IP address

    - E.g., 31.13.84.33

    - Application communication use IP addresses and ports

- DNS

    - Mapping from the domain name hierarchy to IP addresses

> www.facebook.com   canonical name = star.c10r.facebook.com.
> Name: star.c10r.facebook.com
> Address: 31.13.84.33

DISTRIBUTED SYSTEMS GROUP

DNS Information records

Type of record	Associated entity	Description
SOA	Zone	Holds information on the represented zone
A	Host	Contains an IP address of the host this node represents
MX	Domain	Refers to a mail server to handle mail addressed to this node
SRV	Domain	Refers to a server handling a specific service
NS	Zone	Refers to a name server that implements the represented zone
CNAME	Node	Symbolic link with the primary name of the represented node
PTR	Host	Contains the canonical name of a host
HINFO	Host	Holds information on the host this node represents
TXT	Any kind	Contains any entity-specific information considered useful

Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall
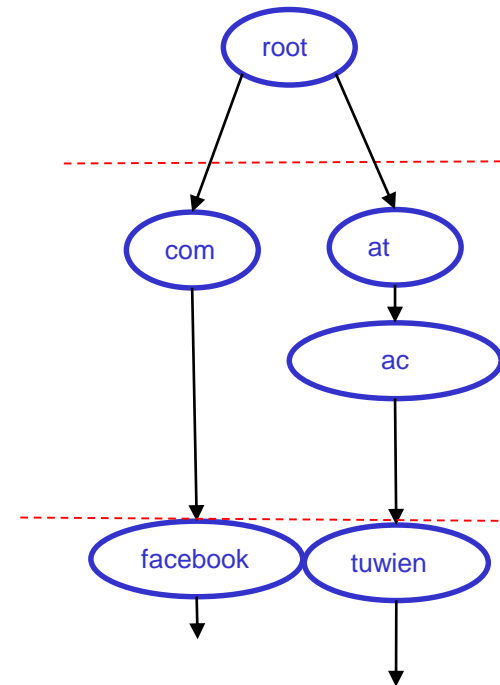
DISTRIBUTED SYSTEMS GROUP

Domain Name Space

NS RR ("resource record") names the nameserver authoritative for delegated subzone

"zone delegation"

"delegated subzone"

When a system administrator wants to let another administrator manage a part of a zone, the first administrator's nameserver **delegates** part of the zone to another nameserver.

= **resource records** associated with name

= **zone** of authority, managed by a **name server**

see also: RFC 1034 4.2: How the database is divided into zones.
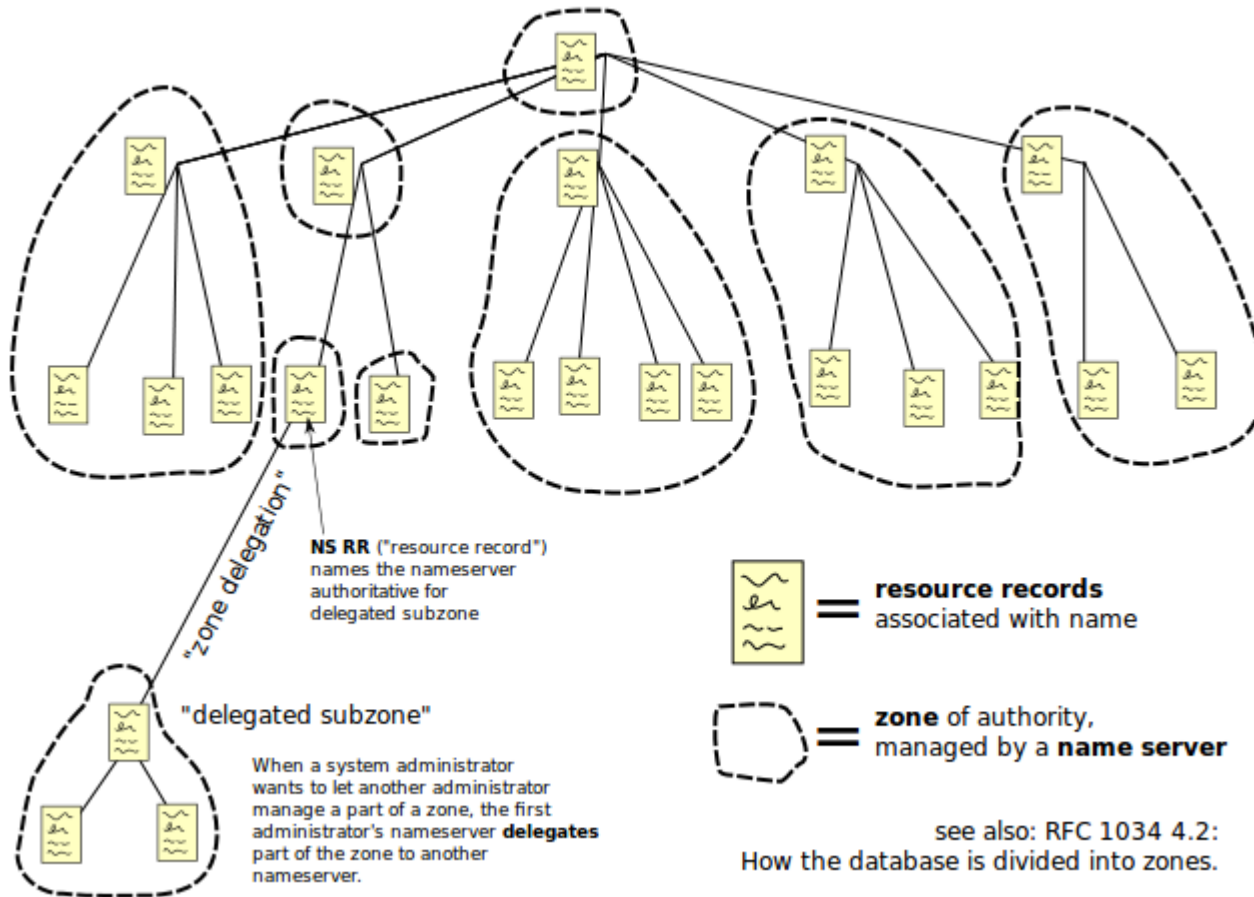
Source: https://upload.wikimedia.org/wikipedia/commons/b/b1/Domain_name_space.svg

DISTRIBUTED SYSTEMS GROUP

DNS Name Servers

Root Name Server	Root Name Server	Root Name Server

Administered Zone Name Server

Administered Zone Name Server

Administered Zone Name Server

Administered Zone Name Server

Administered Zone Name Server

Administered Zone Name Server

Administered Zone Name Server

Administered Zone Name Server

Administered Zone Name Server

- **Authoritative name server:** answer requests for a zone
- **Primary and secondary servers:** the main server and the replicated server (maintained copied data from the main server)
- **Caching server**

DS WS 2015   41

DISTRIBUTED SYSTEMS GROUP

DNS Queries

- **Simple host name resolution**
    - Which is the IP of www.tuwien.ac.at?

- **Email server name resolution**
    - Which is the email server for truong@dsg.tuwien.ac.at ?

- **Reverse resolution**
    - From IP to hostname

- **Host information**

- **Other services**

DISTRIBUTED SYSTEMS GROUP

Examples

- Iterative hostname resolution:
  http://www.simpledns.com/lookup-dg.aspx

- Mail server resolution:
  https://www.mailive.com/mxlookup/

DISTRIBUTED SYSTEMS GROUP

ATTRIBUTE-BASED NAMING

DISTRIBUTED SYSTEMS GROUP

Attributes/Values

- A tuple (attribute,value) can be used to describe a property
  - E.g., („country","Austria"), („language", „German"),
- A set of tuples (attribute, value) can be used to describe an entity

AustriaInfo

Attribute	Value
CountryName	Austria
Language	German
MemberofEU	Yes
Capital	Vienna

Attribute-based naming systems

- Employ (attribute,value) tuples for describing entities
  - Why are flat and structured naming not enough?
- Also called directory services
- Naming resolution
  - Usually based on querying mechanism
  - Querying usually deal with the whole space
- Implementations
  - LDAP
  - RDF (Resource Description Framework)

DISTRIBUTED SYSTEMS GROUP

LDAP data model

- **Object class**: describe information about objects/entities using **tuple(attribute,value)**
    - Hierarchical object class

- **Directory entry**: object entry for a particular object, alias entry for alternative naming and subentry for other information

- **Directory Information Base** (DIB): collection of all directory entries
    - Each entry is identified by a **distinguished name** (DN)

- **Directory Information Tree** (DIT): the tree structure for entries in DIB

DISTRIBUTED SYSTEMS GROUP

LDAP – Lightweight Directory Access Protocol

- http://tools.ietf.org/html/rfc4510

- Example of attributes/values

Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Comp. Sc.
CommonName	CN	Main server
Mail_Servers	—	137.37.20.3, 130.37.24.6, 137.37.20.10
FTP_Server	—	130.37.20.20
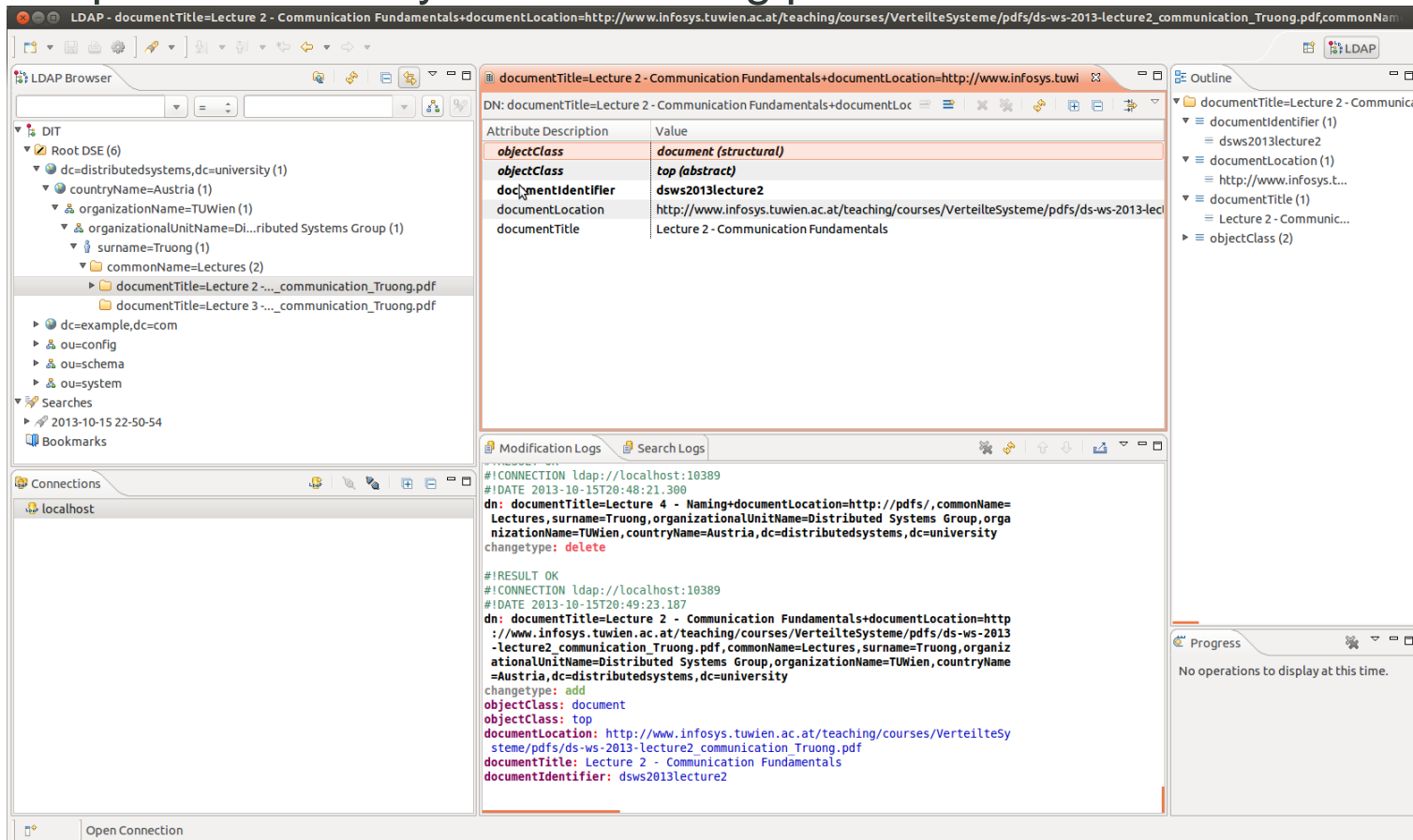
Source: Andrew S. Tanenbaum and Maarten van Steen, Distributed Systems – Principles and Paradigms, 2nd Edition, 2007, Prentice-Hall

DISTRIBUTED SYSTEMS GROUP

LDAP-- Interaction

Client-server protocol

DS WS 2015
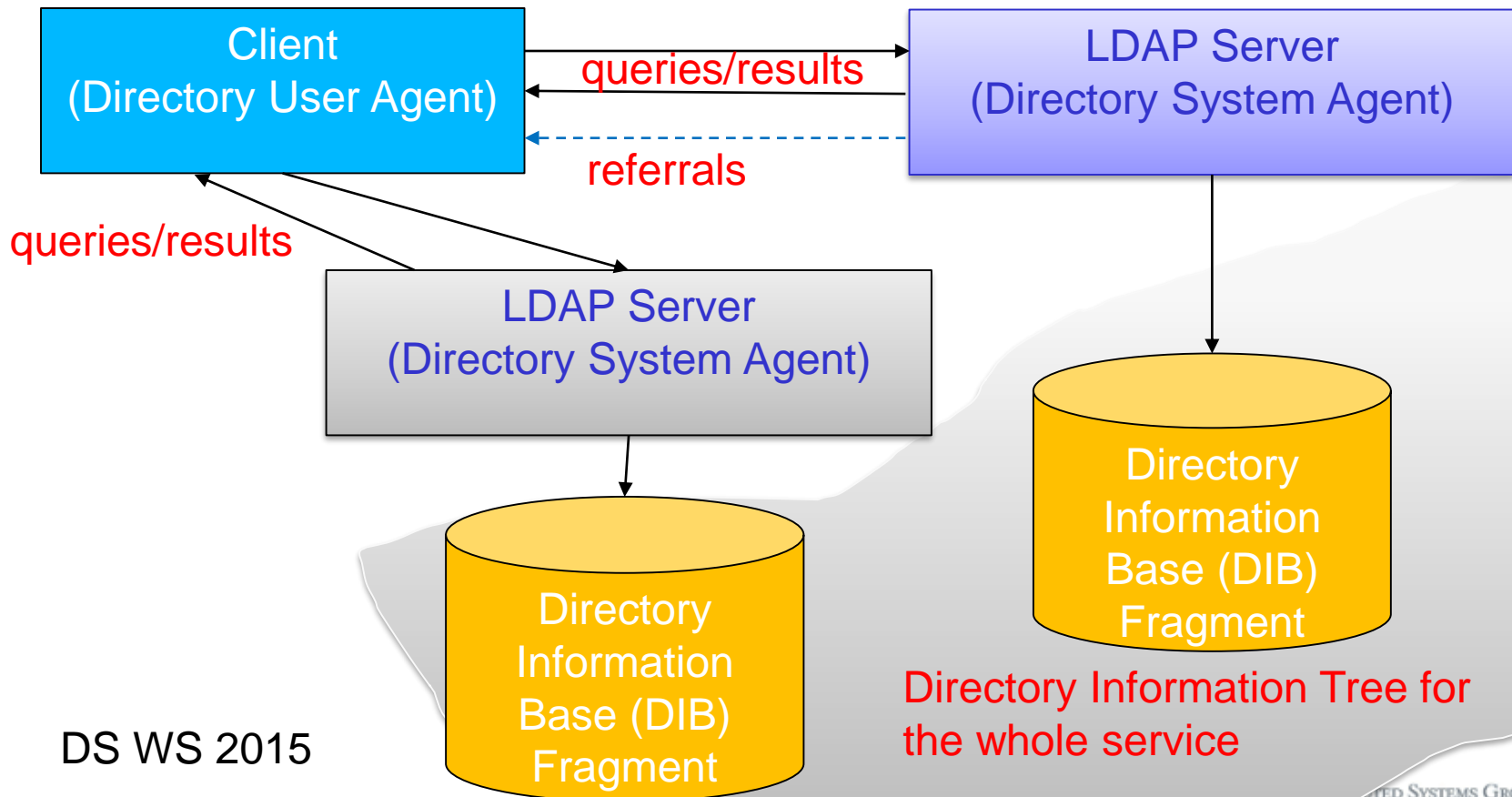
Example with Apache DS/DS Studio

- http://directory.apache.org/

- Apache DS: a directory service supporting LDAP and others

- Apache Directory Studio: tooling platform for LDAP

NAMING SERVICES IN THE WEB

DISTRIBUTED SYSTEMS GROUP
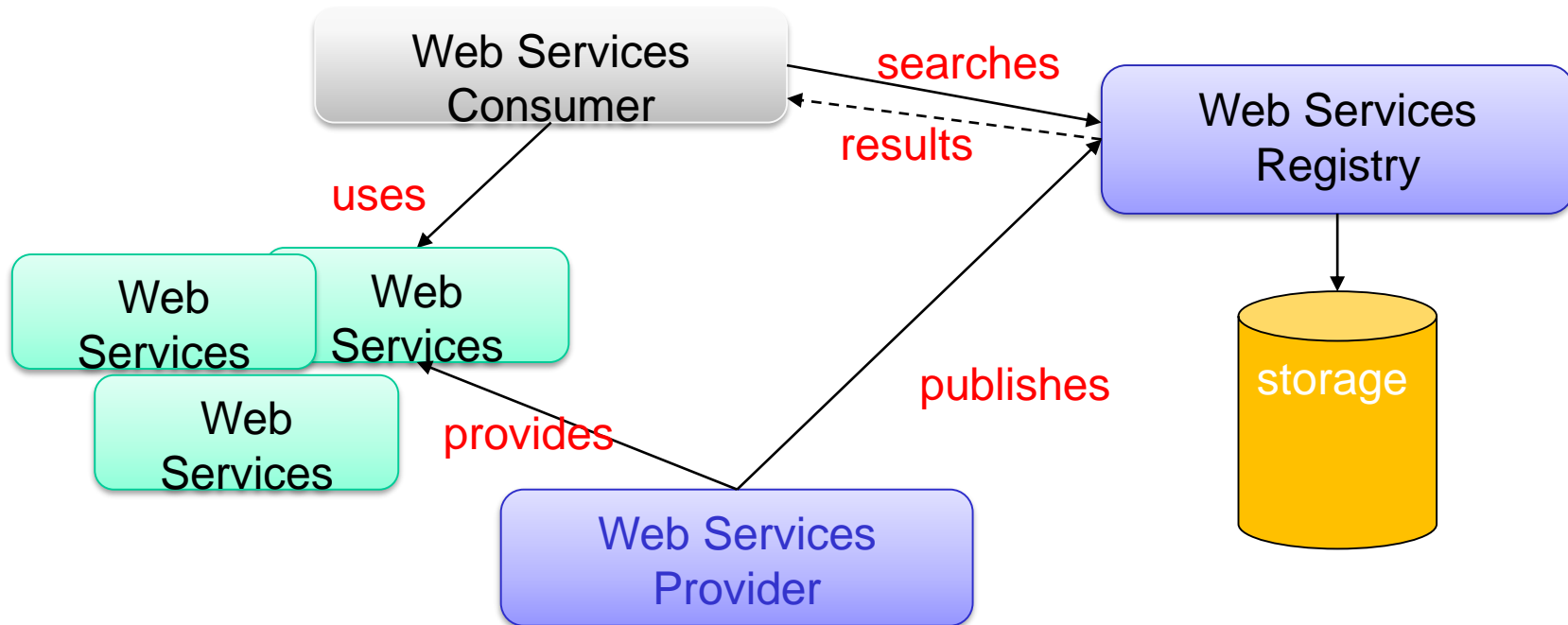
Web services – service identifier

- **Web service:** basically an entity which offers software function via well-defined, interoperable interfaces that can be accessed through the network
  - E.g., http://www.webservicex.net/globalweather.asmx

- **Web services identifier:**
  - WSDL
    - A web service can be described via WSDL. Inside WSDL, there are several „addresses" that identify where and how to call the service access points
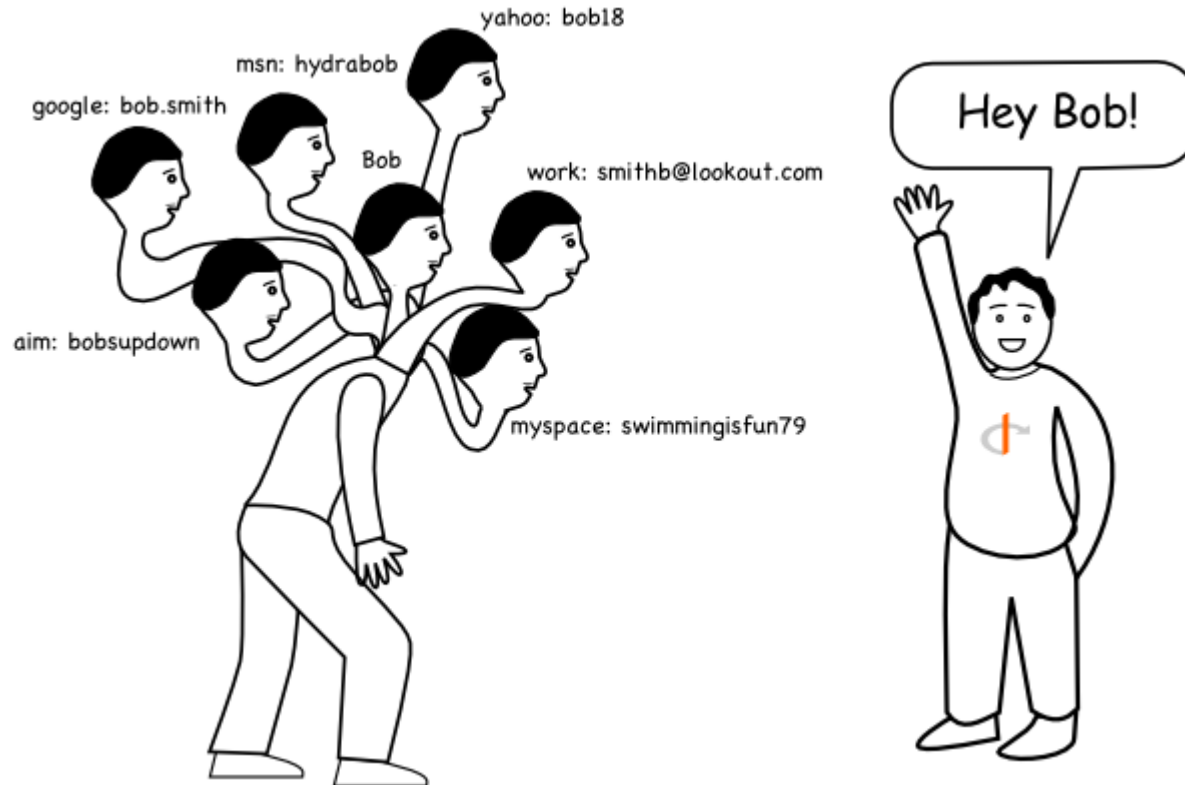
DISTRIBUTED SYSTEMS GROUP

Web services -- discovery

- Registry implementations
  - WSO2 Governance Registry - http://wso2.com/products/governance-registry/
  - java UDDI (jUDDI) - http://juddi.apache.org/

Identifiers in the Web

Source: http://openidexplained.com/

OpenID – people identifier in the Web

- Several services offering individual identifiers
- But there will be no single provider for all people

We need mechanisms to accept identifiers from different providers

- OpenID standard enables identifiers for people that can be accepted by several service providers
- An OpenID identifier is described as a URL
  - E.g., https://me.yahoo.com/a/.....

Q: Why can an OpenID identifier be considered unique?

DISTRIBUTED SYSTEMS GROUP

Example

Using OpenID to login to some services

OpenID interactions
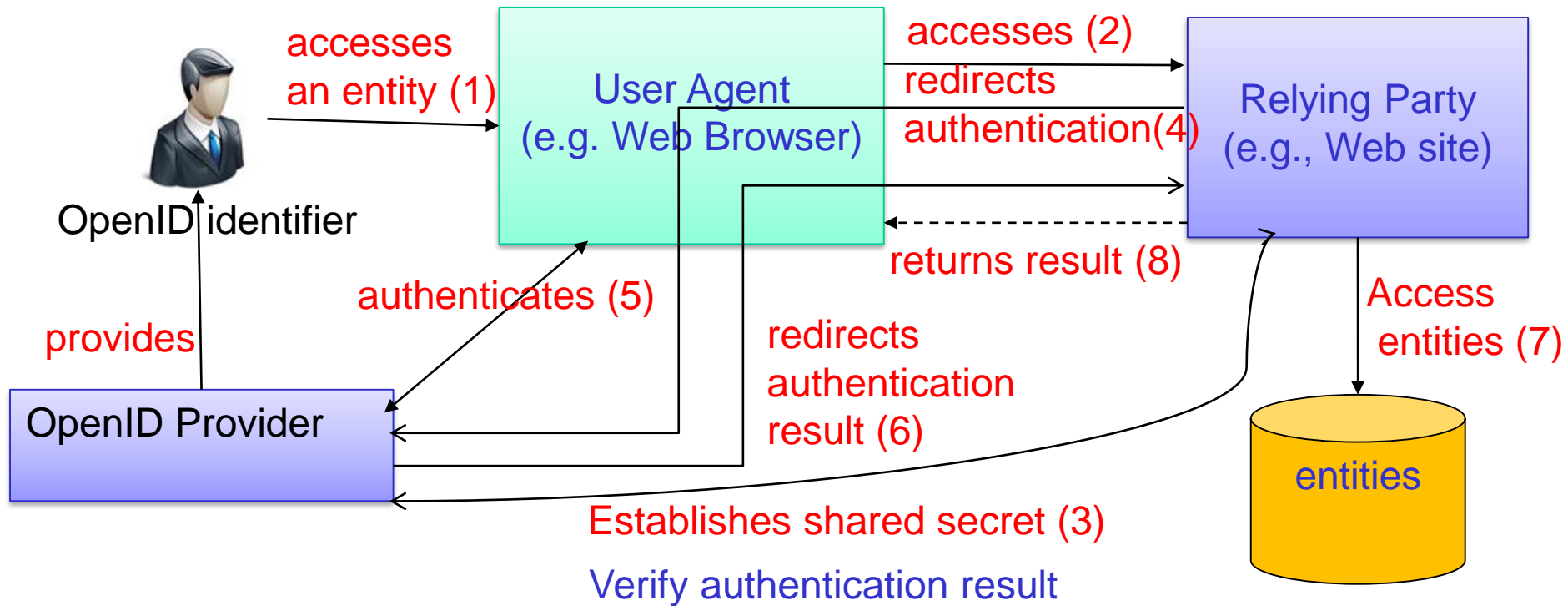
Take away message: Mechanisms for integrating different specific naming services

A REAL-WORLD HOME WORK

Problems

- A very big organization in EU has many services and its own employees from different locations. It uses distributed LDAP servers for managing names/identifiers of its employees and services

- The organization has a lot of external users from different companies and freelancers (external partners)
  - Some companies are big with a lot of people working for the organization in a short term, some have only a few people

- The organization wants to support the collaboration among members of different teams and a team consists of people from the organization and external partners
  - The organization does not want to manage external people but it trusts its external partners

Approach to solution

- The organization asked us possible solutions for managing team members by allowing them to access different services of the organization

- We suggested the organization to develop

  - Develop an OpenID service so that the organization is also an OpenID provider, by using OpenID-to-LDAP software to interface to internal LDAP servers

  - A naming service interfaces to external OpenID servers and the organization's OpenID service

  - Each team consists of a set of members, each member is unified identified by an OpenID

  - Each team is associated with a set of services that it can use, the service information is stored in LDAP server.

- Homework: design your solution based on our suggestion so that given a team you can find out member details and team services

DISTRIBUTED SYSTEMS GROUP

Summary

- Naming is a complex issue
    - Fundamental for other topics, e.g., communication and access control in distributed systems
- Data models/structures versus processes
- Different models
    - Flat, structured and attributed-based naming
- Different techniques to manage names
    - Centralized versus distributed
- Different protocols for naming resolution
- Dont forget to play with some simple examples to understand existing concepts

DISTRIBUTED SYSTEMS GROUP

Summary

When you finish this lecture:

- Understand different naming techniques
  - Mechanisms, and Pros and cons
- Be able to design (simple) naming services for specific systems

In real world: naming in LAN/Smart homes, DHT, Filesystem (names and mounting), DNS, LDAP, Web Services, and OpenID

Advanced systems: Facebook, Spotify, Netflix, IoT, etc.

DISTRIBUTED SYSTEMS GROUP

Thanks for your attention

Hong-Linh Truong
Distributed Systems Group
Vienna University of Technology
truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/staff/truong

DISTRIBUTED SYSTEMS GROUP