

ICS 362 Distributed Systems

Distributed Systems: Part 2

Lecturer: Toby Daniel



Transparency

What is transparency?

Transparency

Access Transparency

Location Transparency

Migration Transparency

Relocation Transparency

Replication Transparency

Concurrency Transparency

Failure Transparency

Persistency Transparency

Access Transparency

- Different Resources may represent data in different formats, but this shouldn't be an issue for the user.
 - A user on an Intel workstation sending data to a Sun SPARC machine, shouldn't be concerned that Intel orders its bytes by little endian format (high order bytes first) while SPARC uses big endian format (low order bytes first).
- Different file naming formats should also not be of concern to the user. '/' or '\'.

Location Transparency

- Location Transparency refers to the physical position of a resource, which should be hidden from the user.
- This is normally achieved through naming, where normally only logical names are used:

www.payap.ac.th/english/index.html



What is DNS?

Migration / Relocation Transparency

- In the previous web address, you have no idea whether index.html has always been on the payap server ...
or when it might have moved there.
- If resources can be moved without affecting the way the resource is accessed then **migration transparency** is provided. If that movement occurs while the resource is being accessed, then **relocation transparency** is provided. Consider moving around using a wireless laptop.

Replication Transparency

- The efficiency of distributed systems can be improved greatly by locating replicas (copies) of a resources physically closer to a user.
- Replication transparency enables the system to do this, without the user knowing they are using a replica.



What examples of replication transparency can you think of?

Concurrency Transparency

- A goal of distributed systems is often sharing of resources between users.
- These users may wish to access or even update the same data at the same time (concurrently).
- An important challenge when designing distributed systems is how to deal with concurrent accesses.



What problems can occur with concurrent access?

Failure Transparency

- Failure Transparency tries to mask failures so that they are not seen or noticed by the users
- It is difficult to identify between a resource that has failed and a resource which is performing badly (slowly).
 - Consider opening a webpage - is it dead or painfully slow, how long should the browser wait?



What can be done to achieve failure transparency?

Persistence Transparency

- Persistence transparency refers to masking whether a resource (data resource) is in primary or secondary memory.
 - It shouldn't matter to the user whether they are performing operations on volatile memory or on a disk.

Transparency Summarised

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk



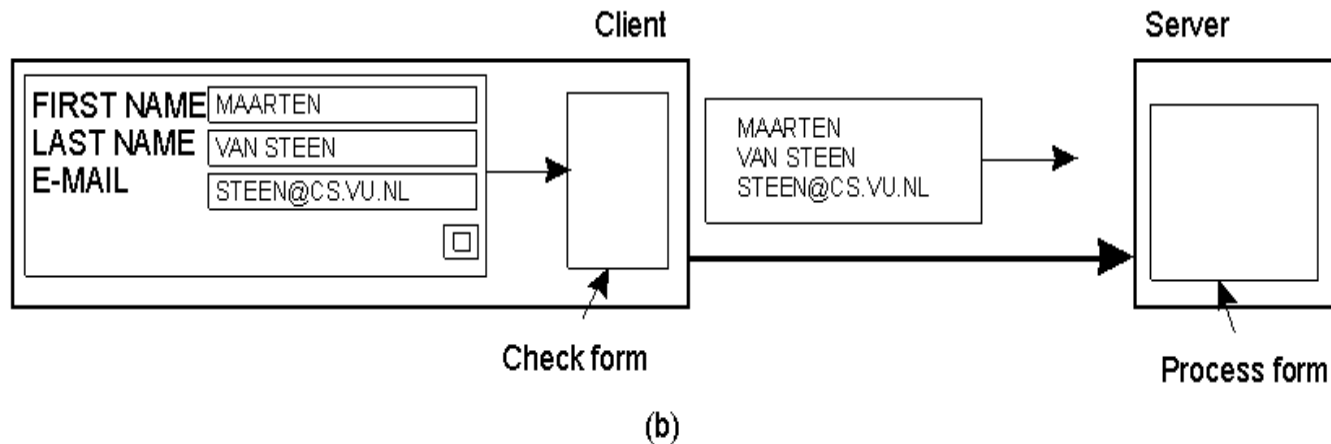
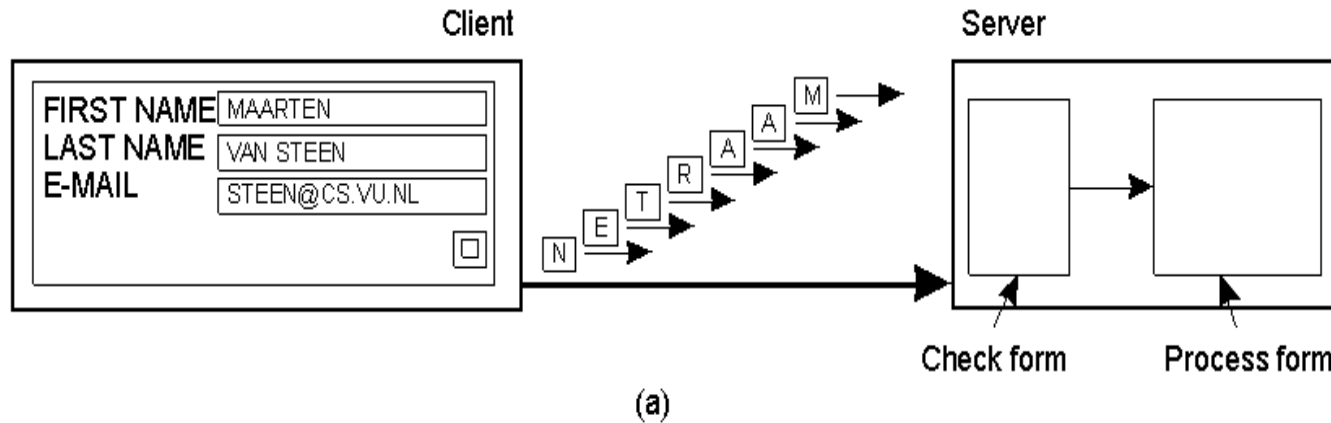
Scalability

What is scalability?

Scalability Problems

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

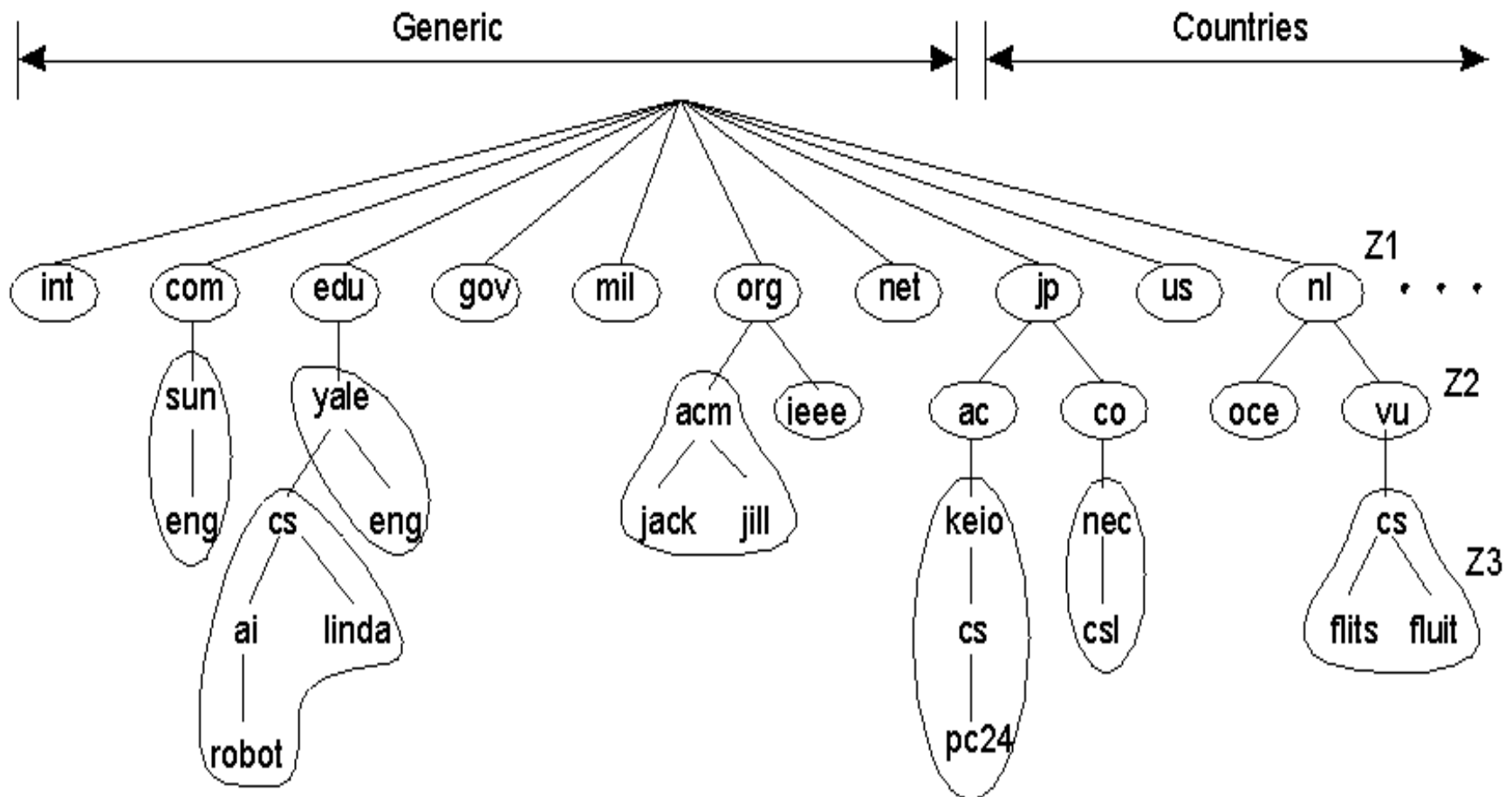
Scaling Techniques



The difference between letting:

- b) a server or
- c) a client check forms as they are being filled

Scaling Techniques



An example of dividing the DNS name space into zones.