Distributed Systems

INF 3190, Vår 2010

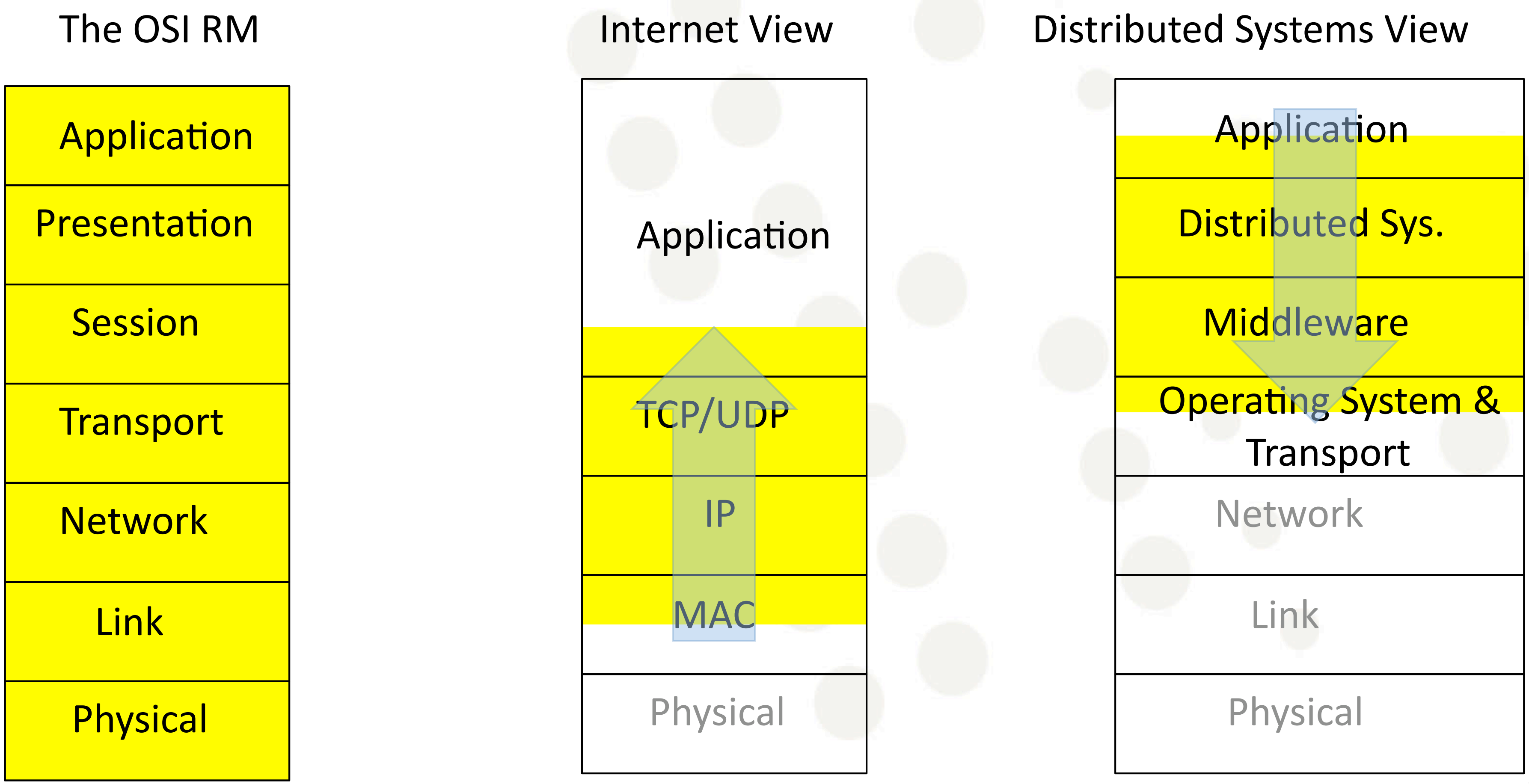
Michael Welzl

UNIVERSITY OF OSLO

What is a distributed system (DS)? Many definitions

- [Coulouris & Emmerich]
  - A distributed system consists of hardware and software components located in a network of computers that communicate and coordinate their actions only by passing messages.
- [Tanenbaum & van Steen]
  - A distributed system is a collection of independent computers that appears to its users as a single coherent system.
- [Lamport]
  - A distributed system is a system that prevents you from doing any work when a computer you have never heard about, fails.

Networks and distributed systems



Necessary considerations

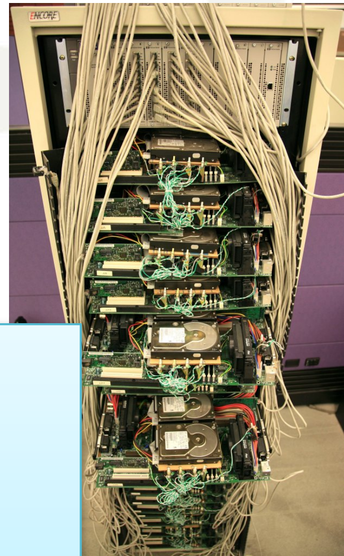
- Independent failure of components
  - “partial failure” & incomplete information
- Unreliable communication
  - Loss of connection and messages. Message bit errors
- Unsecure communication
  - Possibility of unauthorised recording and modification of messages
- Expensive communication
  - Communication between computers usually has less bandwidth, longer latency, and costs more, than between independent processes on the same computer
- Concurrency
  - components execute in concurrent processes that read and update shared resources. Requires coordination
- No global clock
  - makes coordination difficult (ordering of events)

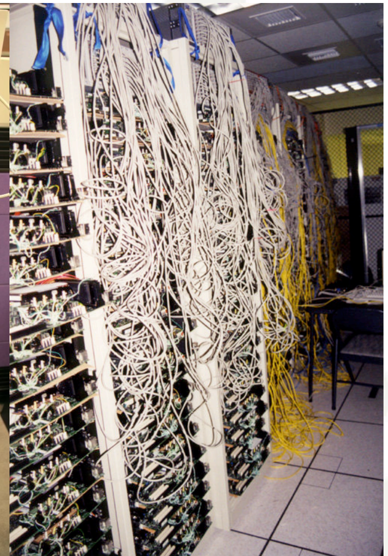
Common requirements

- Resource sharing
  - the possibility of using available resources any where
  - servers provide resources to clients
- Openness
  - an open distributed system can be extended and improved incrementally
  - requires publication of component interfaces and standards protocols and for accessing interfaces
- Scalability
  - the ability to serve more users, provide acceptable response times with increased amount of data
- Fault tolerance
  - maintain availability even when individual components fail
- Allow heterogeneity
  - network and hardware, operating system, programming languages, implementations by different developers

Example: Google File-System

early days





Challenges:
- Scalability
- Fault-tolerance
- Auto recovery

Distribution transparency

- An important goal of a distributed system is to hide the fact that its processes and resources are physically distributed across multiple computers

- A distributed system that is able to present itself to its users and applications as if it were only a single computer system is said to be **transparent**

Forms of transparency

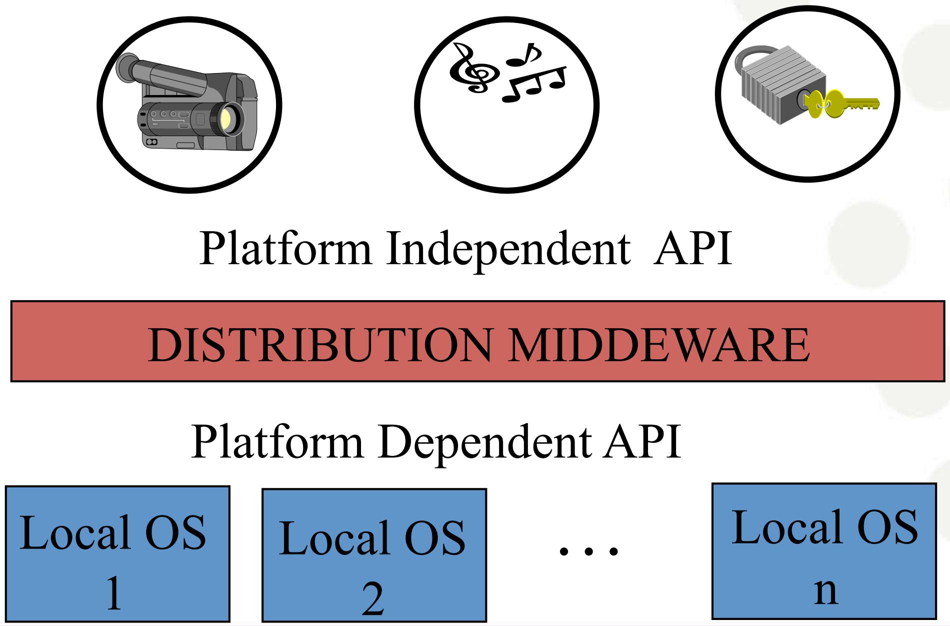
Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that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

Different forms of transparency in a distributed system (ISO, 1995).

- Trade-off between degree of transparency and performance of a system
  - Do we really always want transparency? e.g. what about context-awareness...

Middleware

- Layer of software offering a single-system view
- Offers portability and interoperability
- Simplifies development of distributed applications and services

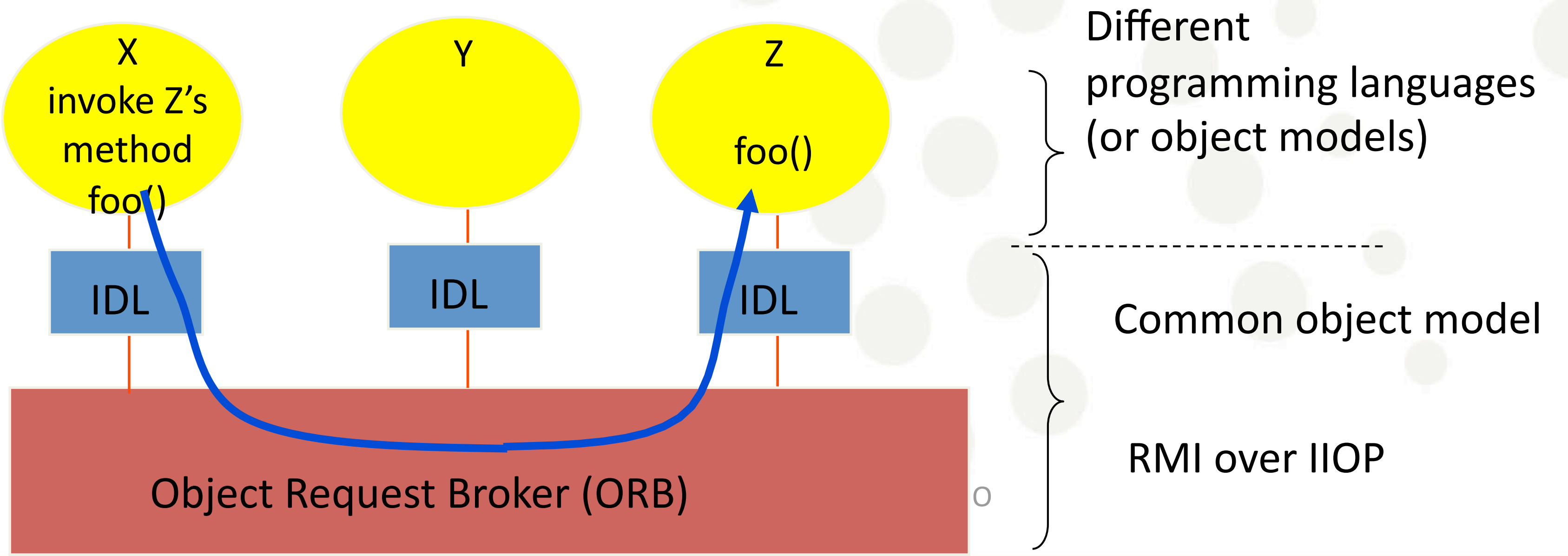


Distributed applications and services

- transaction oriented (ODTP XA)
- message oriented(IBM MQSeries)
- remote procedure call (X/Open DCE)
- object-based (CORBA, COM, Java)

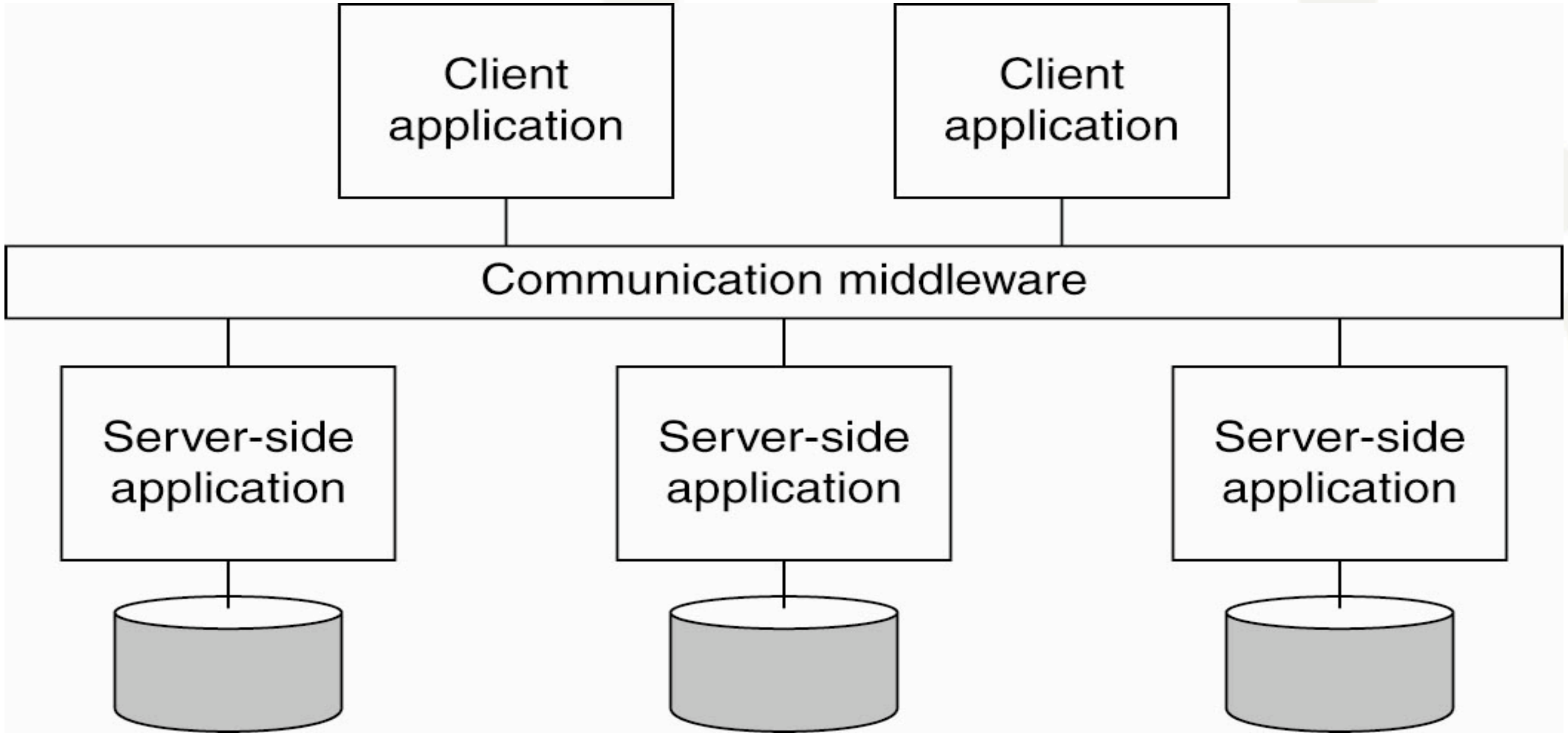
Example communication middleware: CORBA

Clients may invoke methods of remote objects without worrying about:
object location, programming language,
operating system platform, communication
protocols or hardware.



Example DS: Enterprise Application Integration (EAI)

Distributed Information System with full integration of business data and business processes



Distributed Computing Systems

- History: parallel processing at a growing scale
  - Parallel CPU architectures
  - Multiprocessor machines
  - Clusters
  - (“Massively Distributed“) computers on the Internet: Grid
    - transparency again: “power Grid“ metaphor
    - Most common underlying middleware: Globus Toolkit (now entirely based on Web Services)

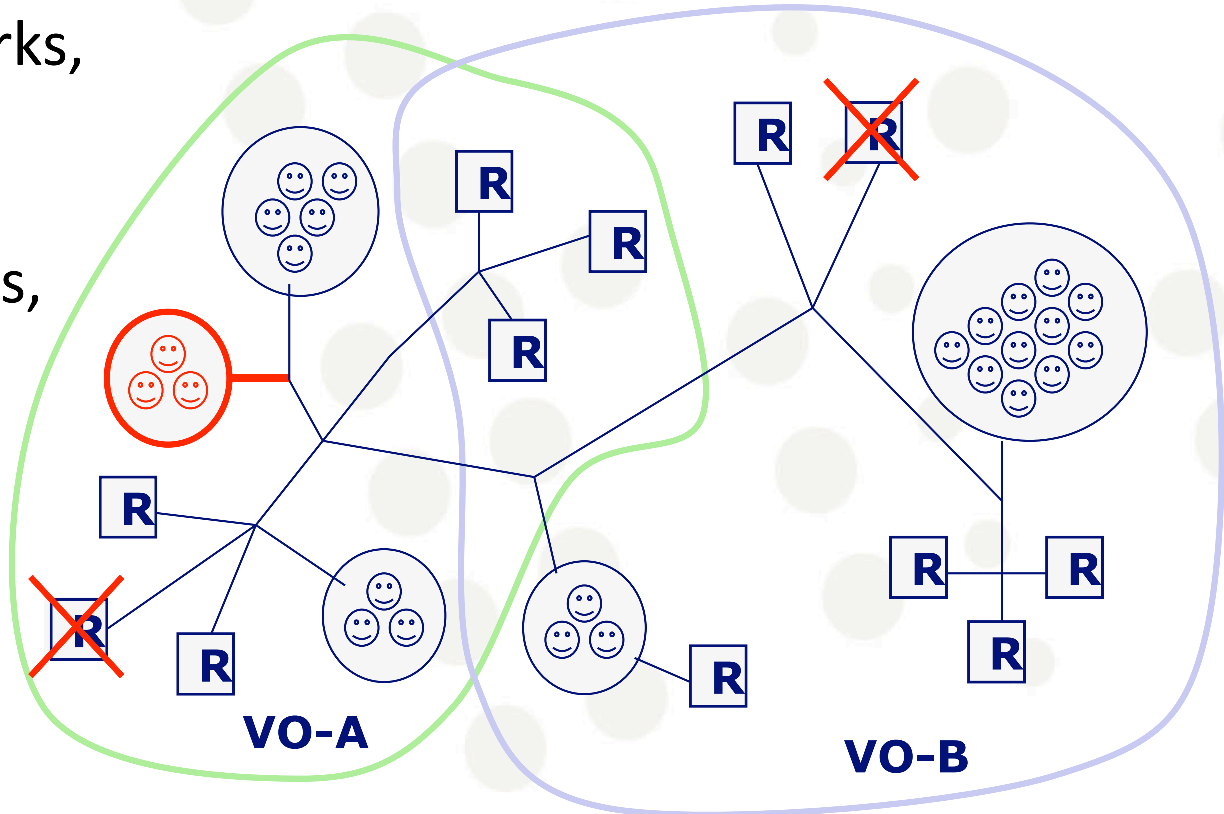
Grid example: e-science - CERN LHC

- Largest machine built by humans: particle accelerator and collider with a circumference of 27 kilometers
  - Said to generate 10 Petabytes (10^7 Gigabytes) of information per year
  - This information must be processed and stored somewhere
- Beyond the scope of a single institution to manage this problem
  - Projects: LCG (LHC Computing Grid), EGEE (Enabling Grids for E-sciencE)

Source: Globus presentation by Ian Foster

Grid structure: Virtual Organizations, Virtual Teams

- Distributed resources and people
- Linked by networks, crossing admin domains
- Sharing resources, common goals
- Dynamic



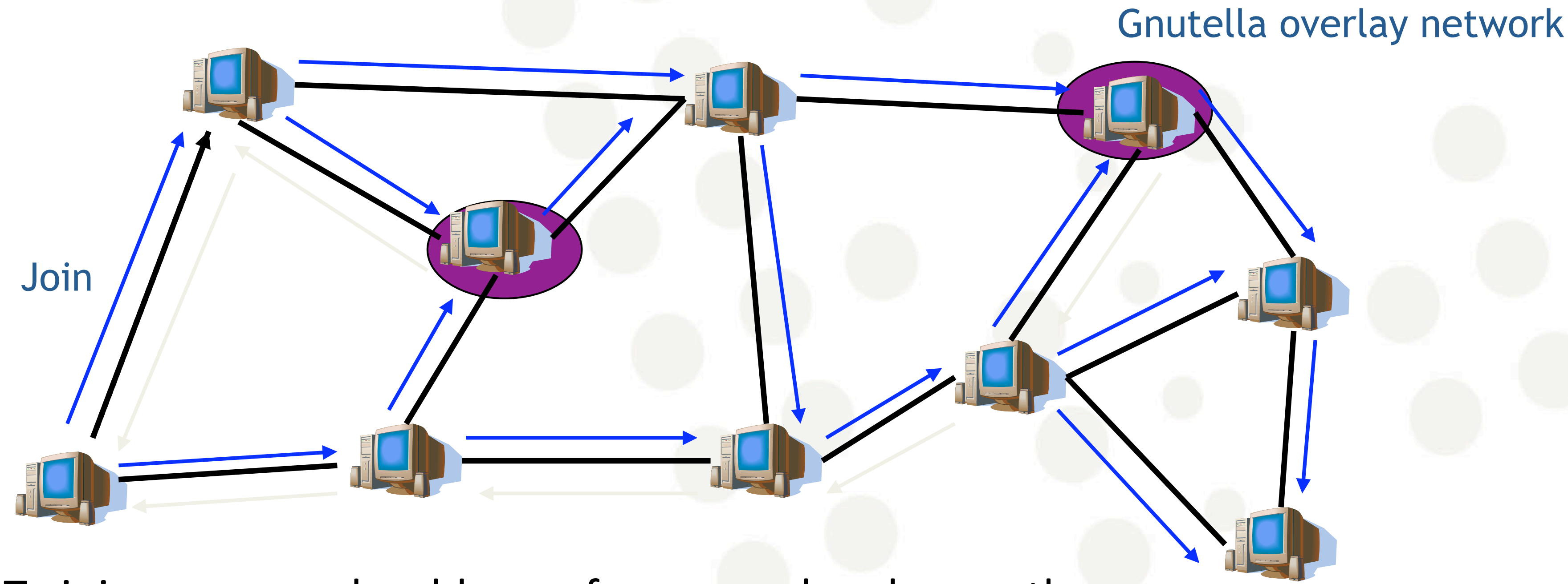
Source: Globus presentation by Ian Foster

Cloud Computing

- New “paradigm”
  - Often regarded as Grid computing with a business case: Grid computing minus VOs, VTs
  - Cloud providers sell cloud access as a service e.g. Amazon

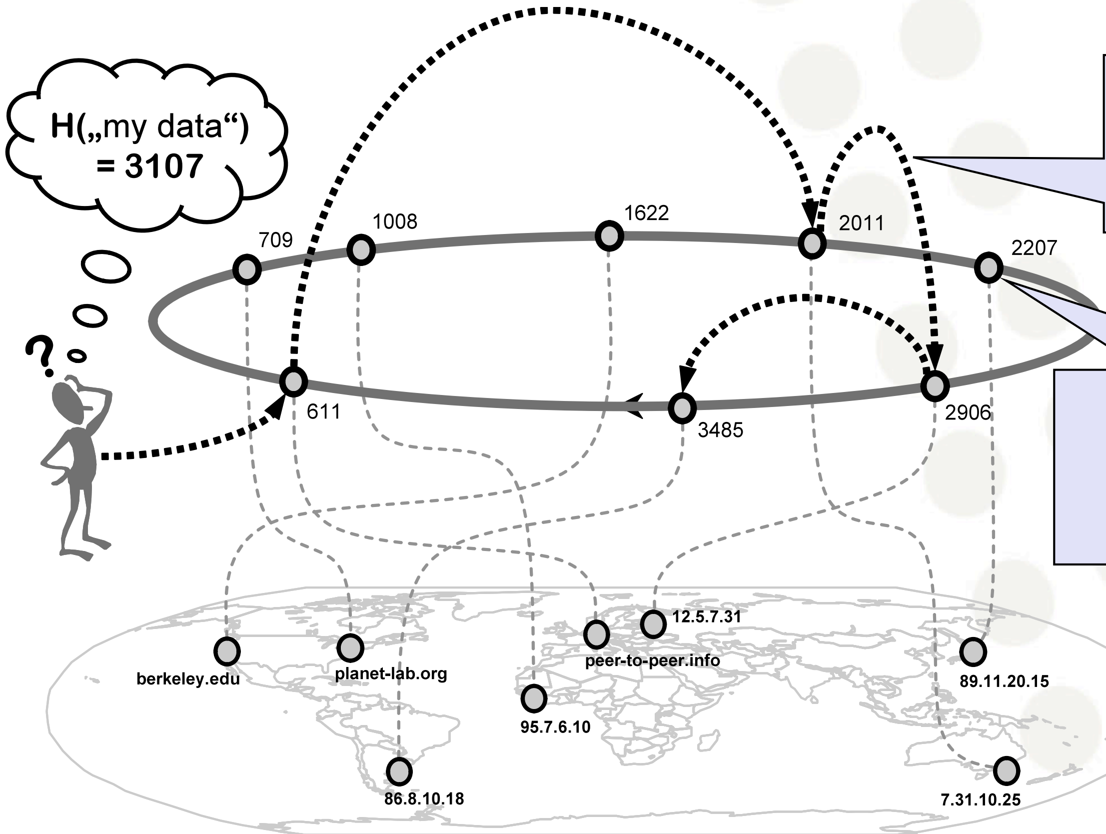
- At least two quite different “visions”
  1. Large server farms: high-speed computing and large & secure data storage on demand
  2. Dumb terminals: all applications become services

P2P Systems: unstructured



- To join, peer needs address of one member, learn others
- Queries are sent to neighbors
- Neighbors forward queries to their neighbors (flooding)
- Replies routed back via query path to querying peer

P2P Systems: structured Distributed Hash Tables (DHT)



Routing to the node storing the data in O(*log(N)*) steps

Nodes store O(*log(N)*) routing information to other nodes

Summary

- Distributed systems
  - components located in a network that communicates and coordinates their actions exclusively by sending messages
- Consequences of distributed systems
  - Independent failure of components
  - Unsecure communication
  - No global clock
- Distribution transparency: providing a single computer system view
- Requirements like resource sharing, openness, scalability, fault tolerance and heterogeneity can be satisfied by distributed systems
  - Many pitfalls when developing distributed systems

References

- A. S. Tanenbaum, M. van Steen, ”Distributed Systems – Principles and Paradigms”, Prentice-Hall 2007
- Slides:
  - INF3190 2009 slides by Frank Eliassen
  - A P2P slide by Jussi Kangasharju
  - A P2P slide by Klaus Wehrle
  - A few figures by Ian Foster