ECUST / FH Lübeck

Study Program „Information Technology“

Distributed Systems

Chapter 1: Introduction

Prof. Dr.-Ing. Menno Heeren

Slides partly from Prof. Dr.-Ing. H. Hellbrück

Copyright notice: Slides for this chapter of this course have been taken from the set of course materials “Distributed Systems” provided by Andrew S. Tanenbaum, Maarten van Steen and Coulouris, Dollimore and Kindberg. The layout has been adapted to the presentation needs of FH Lübeck.

Learning Goals for Students

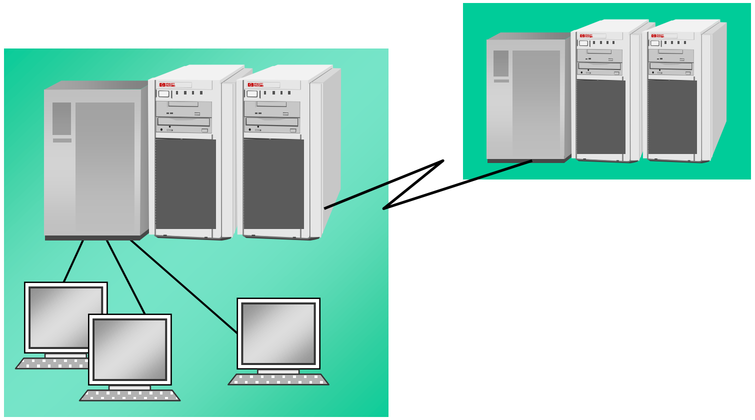
- Acquire a basic understanding of Distributed Systems
  - Explain important terms in Distributed Systems.
  - Describe the goals of Distributed Systems.
  - Explain the difference between Distributed Systems and Parallel Systems.
  - Explain the desired characteristics of Distributed Systems
  - Explain the different notions of transparency in Distributed Systems.
  - Explain the challenges when designing and implementing Distributed Systems/Distributed Applications

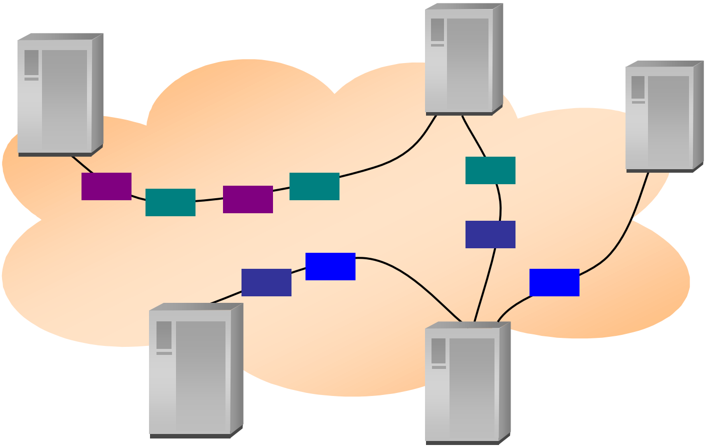
Overview

- A brief history of Networking & Distributed systems
- A „formal“ Definition of Distributed systems
- Some Examples of large Distributed Systems
- (Desired) Characteristics of a DS
- Some selected Challenges

Computer Networks: 1960s-1970s

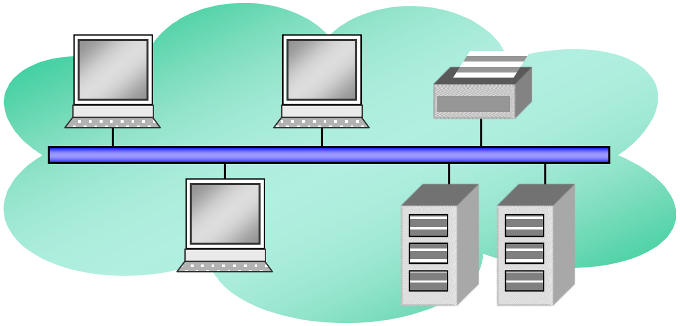
- Terminal to host
  - Network of terminals to access a Mainframe or Minicomputer.
- Host to Host
  - Towards Data Networks based on Packet-Switching.
  - Domination of proprietary networking technologies (IBM SNA, DEC DECnet, ...).
  - Emergence of TCP/IP based networks (ARPANET).

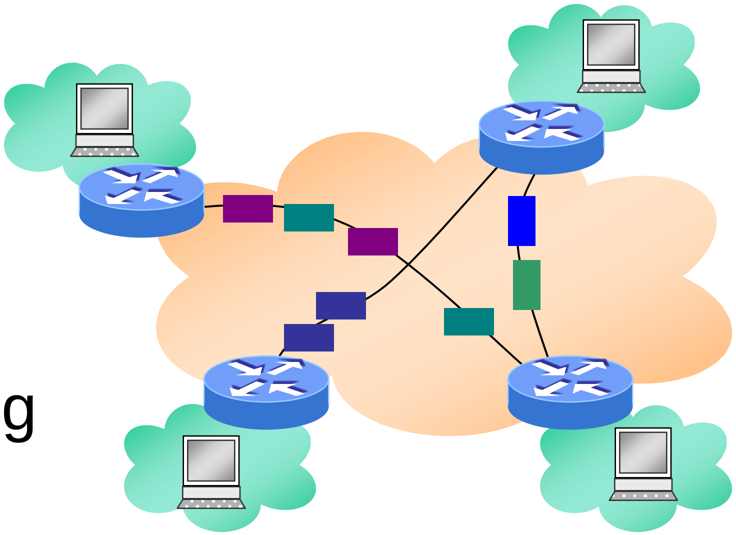




Computer Networks: 1980s

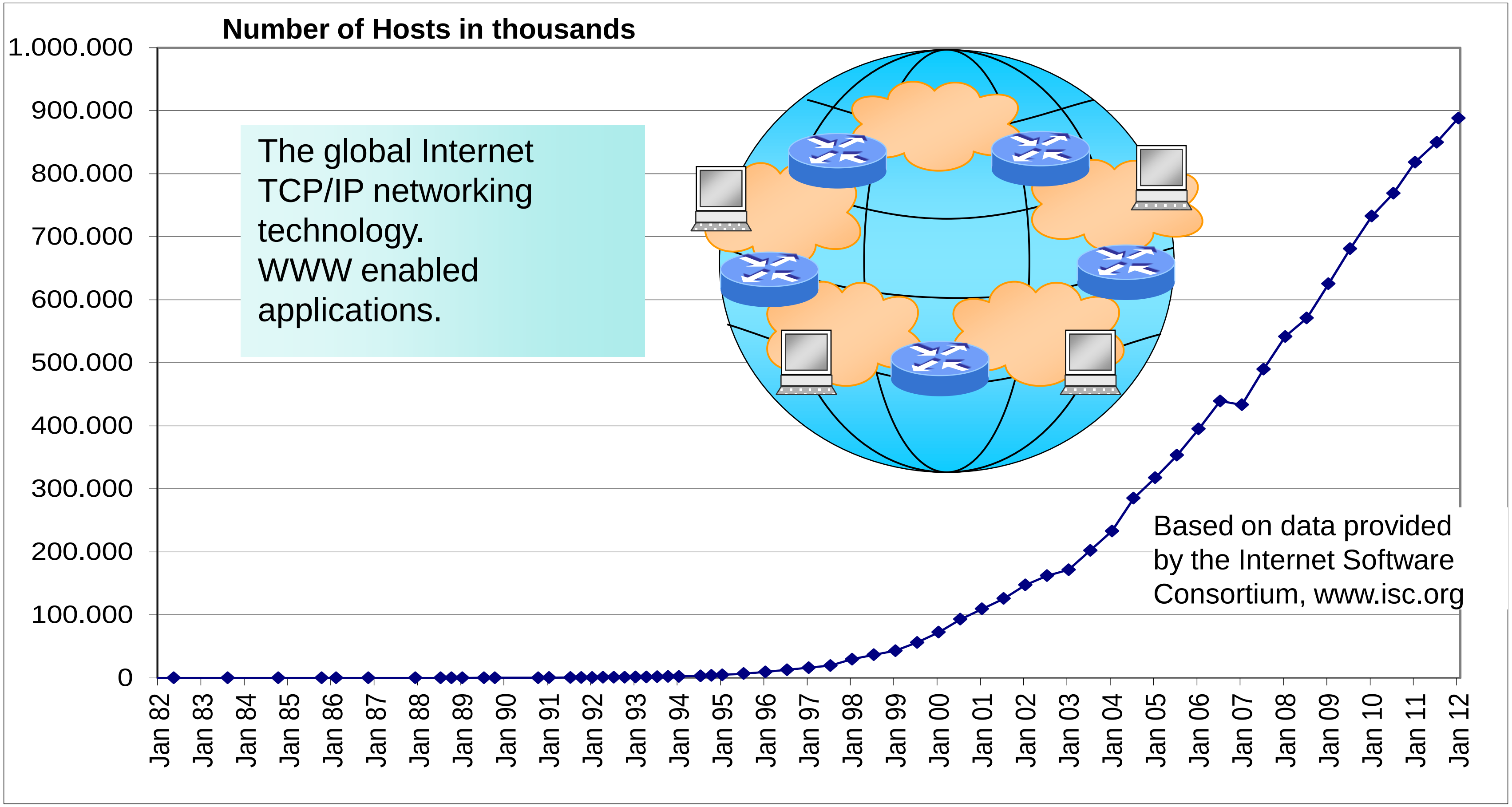
- High performance, low cost local network technologies.
  - Networks of workstations (e.g., PCs) and servers.
  - A new paradigm: distributed client-server computing.
- Public and private wide area data networks
  - Mature packet-switching technology: TCP/IP, X.25.
  - Driving application: e-mail. File transfer, telnet, etc.
- Emergence of open networking technologies (TCP/IP, OSI).





Computer Networks Today

- Host : A host is a domain name that has an IP address (A) record associated with it. This would be any computer System connected to the Internet (via full or part-time, direct or dialup connections). *i.e.. nw.com, www.nw.com*

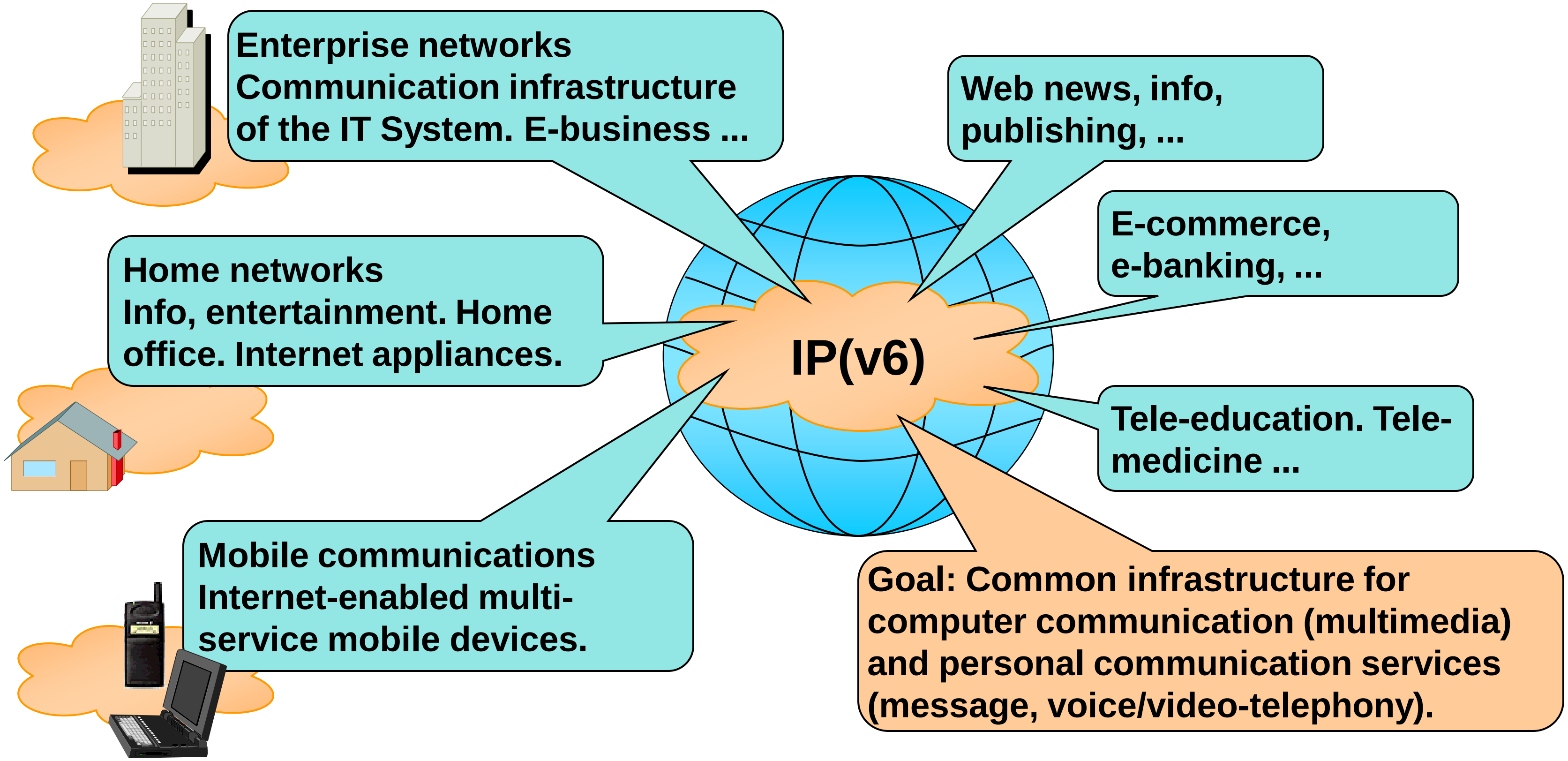


The global Internet TCP/IP networking technology.
WWW enabled applications.

Based on data provided by the Internet Software Consortium, www.isc.org

Computer Networks: 2010s

- IP on everything" (Vint Cerf, Internet Patriarch)
  - Towards a global multi-services network, using IP(v6) as the core transport technology.



Enterprise networks
Communication infrastructure of the IT System. E-business ...

Web news, info, publishing, ...

E-commerce, e-banking, ...

Home networks
Info, entertainment. Home office. Internet appliances.

IP(v6)

Tele-education. Tele-medicine ...

Mobile communications
Internet-enabled multi-service mobile devices.

Goal: Common infrastructure for computer communication (multimedia) and personal communication services (message, voice/video-telephony).

Networks and Distributed Systems

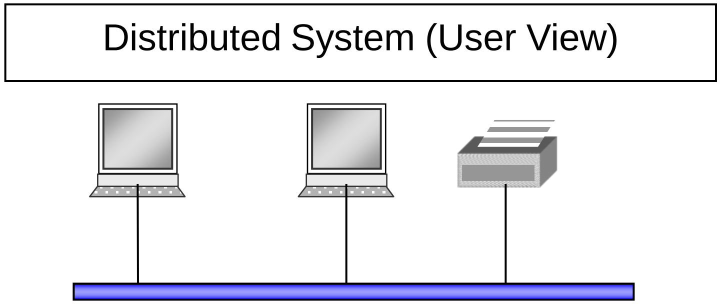
- We talked about the development of networks.
- We will not further focus on them here, they are covered in the Course “Computer Networks”.
- **BUT**
- Computer Networks are an ***indispensable*** tool to implement Distributed Systems!
- So what, is a Distributed System (DS), or a Distributed Application (DA)?

What is a Distributed System?

A practically-oriented definition:

A Distributed System

- consists of a collection of autonomous computers
- linked by a Computer Network
- equipped with Distributed System Software



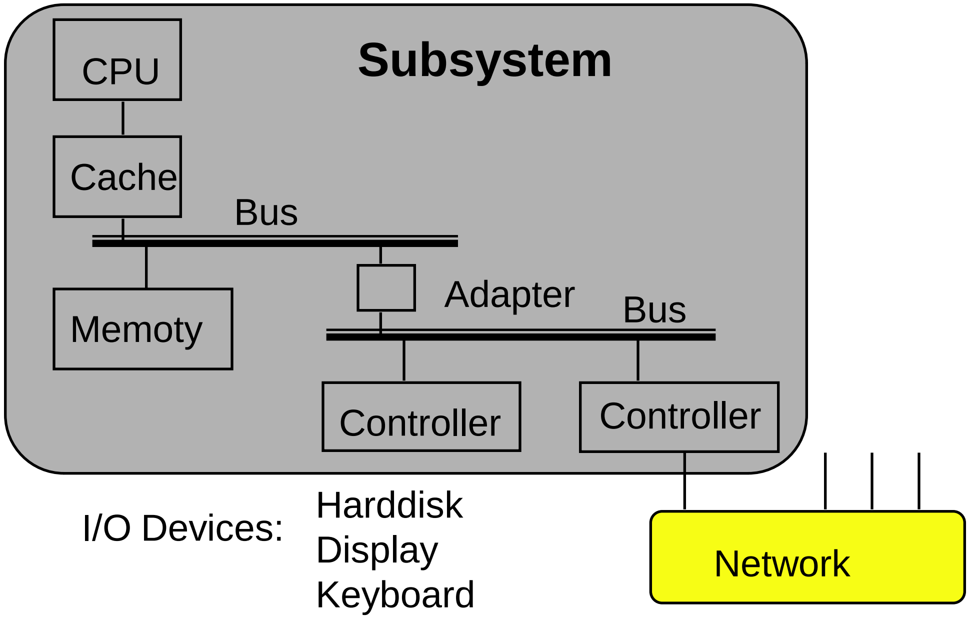
What is a Distributed System?

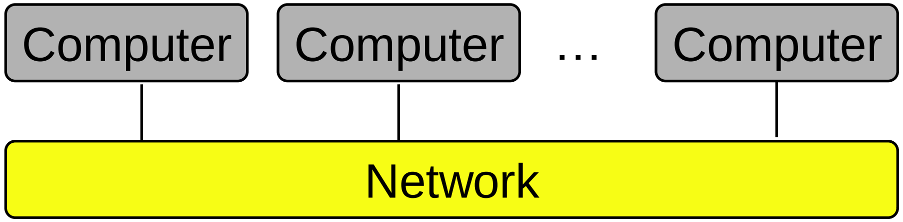
- A more general definition:
- A DS is a system in which
  - Hardware and software components
  - located at networked computers
  - communicate and coordinate their actions
  - by passing messages.

- A distributed Application is an application that runs on top of a Network/Distributed System.
  It consists of different components that interact in order to solve a problem for the application’s user.

Term: Distributed System (DS)

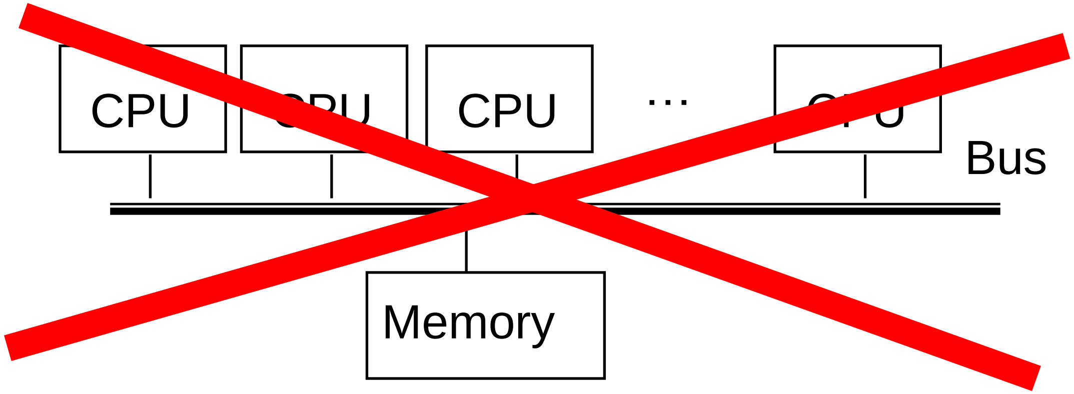
- **System** is composed of several components (**Subsystems**)
- Each subsystem fulfills a subtask
- Together they solve a **common task**
- Components are connected via a **Communication Network**



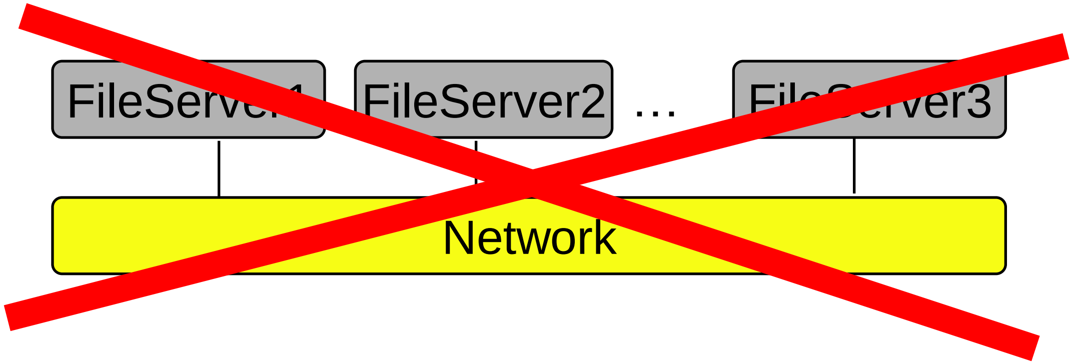


Distributed System ?

Parallel **tightly coupled** system



Loosely coupled system
without common task



Last Definition (by Negation)

- A distributed system where there are no:
  - shared, global clock
  - shared memory
  - accurate failure detection

Funny Definition

You know you have a distributed system when the crash of a computer you’ve never heard of stops you from getting any work done.

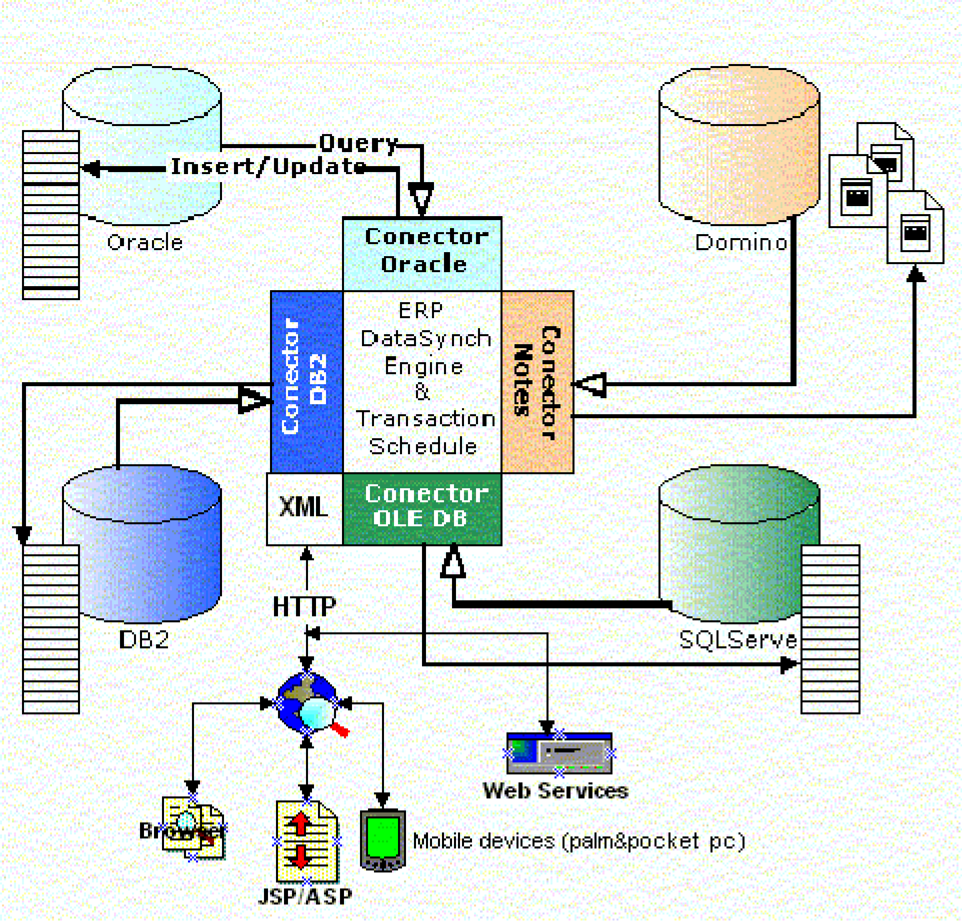
from Leslie Lamport

Important Goals of DS

- Enable computers to coordinate their activities
- Share resources: Hardware, Software, and Data
- Users should perceive a single, integrated computing facility, even though it may be implemented by many computers in different locations.

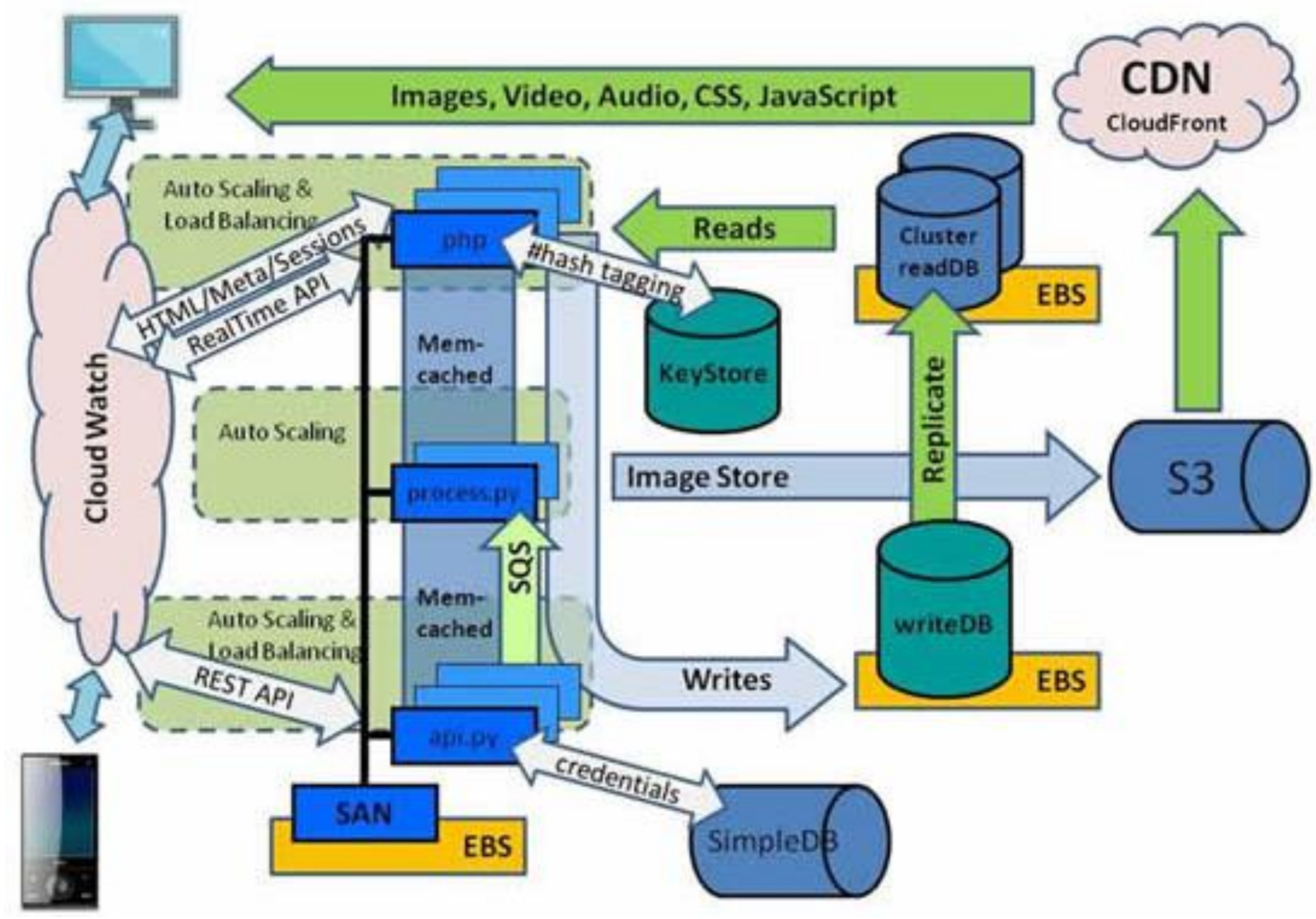
Examples ERP - Systems

- Systems like SAP are large Distributed Systems.

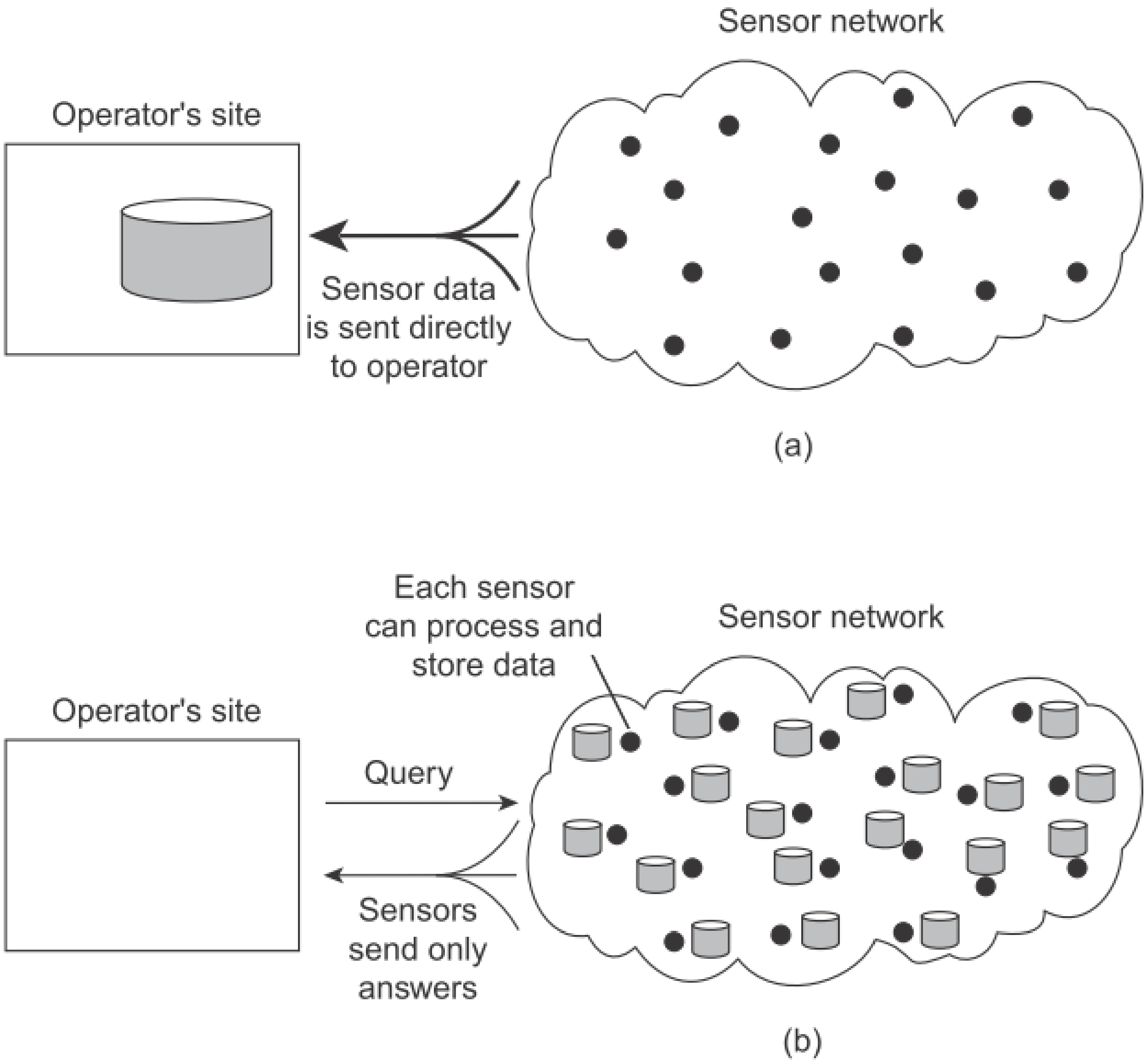


Examples – Amazon

- Offering Cloud Infrastructure



Example Sensor Networks



Operator's site

Sensor network

Sensor data is sent directly to operator

(a)

Operator's site

Each sensor can process and store data

Sensor network

Query

Sensors send only answers

(b)

(Desired Key) Characteristics of DS

- Resource Sharing
- Openness
- Concurrency
- Scalability
- Security
- Transparency
- Fault Tolerance

Resource Sharing

- Hardware Devices: Printers, Disk Drives and other peripherals
- Data Sharing: Software tools, shared objects in a single database, CSCW, Project Work, Team Management
- Management:
  - Client-Server Model: Server manages resources which are used by clients
  - Object Model: shared resources are viewed as objects and can be accessed via their ID

Openness

- Determines to what extent the system can be extended in various ways
- Without disrupting existing services
- Achieved by publishing the Software Interfaces
- Example: UNIX includes Programming Language C. Features of the Computers and Operating System are made available through System Calls.

- Openness will be an issue when we talk about the realization of DS by Distributed Object Systems.

Concurrency

- Several coexisting processes in one system
- if only one processor: Execution by interleaving
- if n processors: Parallel execution
- Concurrency on clients (Application Program) and servers (concurrent resource access)
- important: Synchronization

Scalability

- Algorithms, protocols and procedures, that work with just a few systems, should also work when more computers are added in DS.
- This is easier to achieve than in centralized systems, since resources can be added.

Security

- Data Security in DS has many aspects:
  - Confidentiality: Data can only be read by the intended recipient
  - Integrity: Data has not been altered during transmission
  - Authenticity: Data has really been sent by the person who claims to be the sender
- Security is one of the most important aspects of today’s DS, especially on the Internet because
  - Money is often involved (e-commerce)
  - Personal data is involved
- We will talk in more detail about security at the end of the course (if time is left).

Transparency

- Users of the system are unaware of the fact that the system consists of separated components.
- System is perceived as a whole.

Types of Transparency

- **Access Transparency**: enables local and remote resources to be accessed using identical operations.
- **Location Transparency**: enables resources to be accessed without knowledge of their location.
- **Concurrency Transparency**: enables several processes to operate concurrently using shared resources without interference between them.
- **Replication Transparency**: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or Application Programmers.
- **Failure Transparency**: enables the concealment of faults, allowing users and Application Programs to complete their tasks despite failure of Hardware or Software components.
- **Mobility Transparency**: allows movement of resources and clients within a system without affecting the operation of users or programs.
- **Performance Transparency**: allows the system to be reconfigured to improve performance as loads vary.
- **Scaling Transparency**: allows the system and applications to expand in scale without change to the system structure or the Application Algorithms.

Fault Tolerance

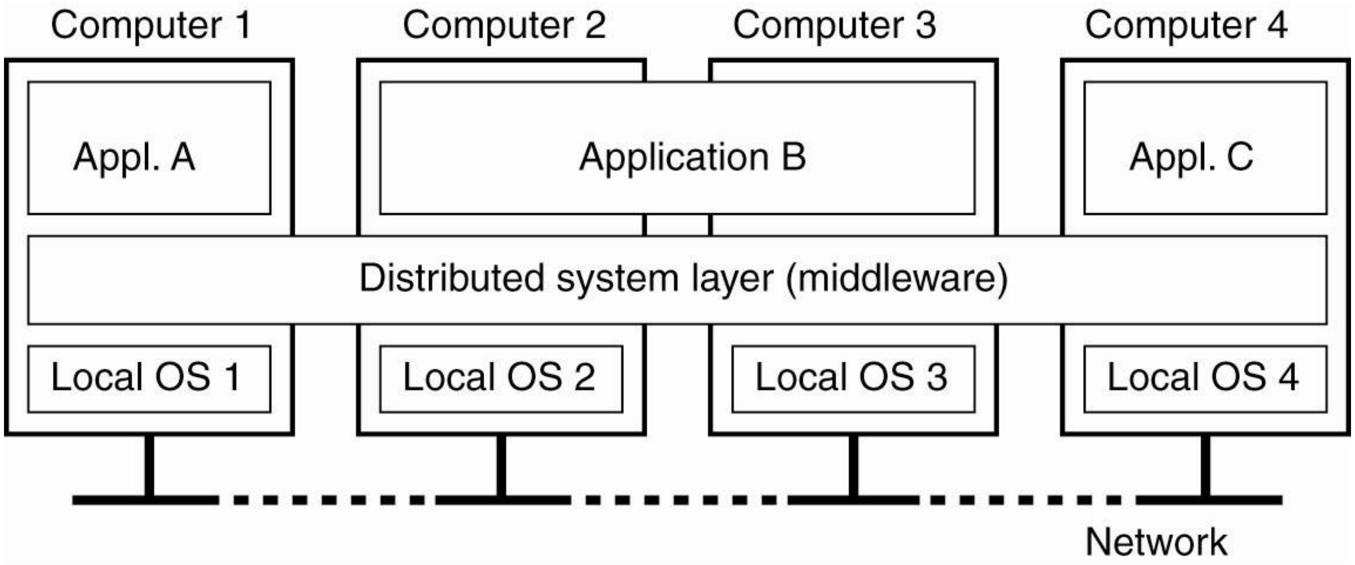
- How can failures be controlled in single Computer Systems?
  - Hardware Redundancy: Standby Hardware
  - Software Recovery: Rollback to a safe state when fault is detected
- Handling of Failures in Distributed Systems
  - Detection (e.g., message checksum)
  - Masking (e.g., email retransmission)
  - Tolerance (e.g., replicated servers)
  - Recovery (e.g., log files)
- Distributed systems use a mix of these techniques
- If there is time left, we will talk about some of the major issues.

Design of DS Challenges: Wrong Assumptions for Network

- Many distributed systems are needlessly complex caused by mistakes that required patching later on. There are many **false assumptions**:
  - The network is reliable
  - The network is secure
  - The network is homogeneous
  - The topology does not change
  - Latency is zero
  - Bandwidth is infinite
  - Transport cost is zero
  - There is a single administrator

Design of DS Challenges: Heterogenity

- Levels:
  - Network
  - Computing hardware
  - Operating systems
  - Programming languages
  - Multiple implementations
- Solution Middleware : Software layer that abstracts from the above providing a uniform computational model



Extreme Distributed Computing

- „Classical“ vs extreme distributed computing
  - Classical Distributed System problems include agreement, total order broadcast, atomic commit, replication, etc.
  - Extreme Distributed System problems include Self-* properties, Scalability, full decentralization, etc.

- Flocks of birds
  - Flying in a flock is good: probability of being killed by a predatoris reduced
  - Flying in a flock is bad: probability of finding food is reduced
  - Birds self-organize themselves in a flock
  - No central authority

Summary

- Distributed Systems are everywhere.
- You use Distributed Systems in daily life without noticing.
- Therefore, an important property of Distributed Systems is transparency.
- For the secure use of Distributed Systems security measures are needed.
- Designing and Implementing Distributed Systems is challenging.
- Self-Healing Properties are a must for very large Distributed Systems (inspired from nature).

Outlook

- System Models
  - Architecture of DS
  - Client-Server Principle
  - Requirements of Distributed System
- Protocols
- Middleware

Literature

- **Andrew S. Tanenbaum et al.: “Distributed Systems : Principles and Paradigms” 2nd ed., Pearson/Prentice Hall 2007** **VK 2150 2007 A 2555**
- **Coulouris et al.: „Distributed Systems“, 4th ed., Addison-Wesley, 2005., Signatur:** VK 1690 2005 A 1471**.**
- The coordinated attack and the jealous amazons. http://www.dsi.uniroma1.it/~asd3/dispense/attack+amazons.pdf .
- Links
  - Two Generals problem http://en.wikipedia.org/wiki/Two_Generals'_Problem
  - Byzantine Generals problem http://research.microsoft.com/en-us/um/people/lamport/pubs/byz.pdf
  - Bitcoin http://en.wikipedia.org/wiki/Bitcoin
  - The S3 incident http://status.aws.amazon.com/s3-20080720.html
  - Solving the unsolvable http://cacm.acm.org/magazines/2011/7/109895-solving-the-unsolvable/fulltext
  - Rise & fall of Corba http://queue.acm.org/detail.cfm?id=1142044