Distributed Systems Development

Paulo Gandra de Sousa

psousa@dei.isep.ipp.pt

MSc in Computer Engineering

DEI/ISEP

Programação de Sistemas Distribuidos

Paulo Gandra de Sousa

psousa@dei.isep.ipp.pt

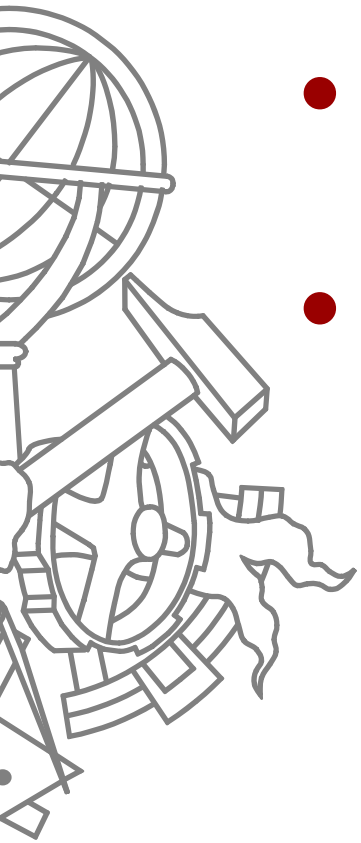
Mestrado em Engenharia Informática

DEI/ISEP

Disclaimer

- Parts of this presentation are from:
  - Tannembaum
  - Coulouris
  - Marta Kwiatkowska (06-06798)
  - Ciarán O'Leary (DT249-4)
  - Scott Shenker and Doug Terry (CS 294)
  - Daniel Ortiz-Arroyo (DE7)

Today's lesson



- Introduction to DS
  - Definition
- Characterization
  - Motivation
  - Pros and cons
  - Issues

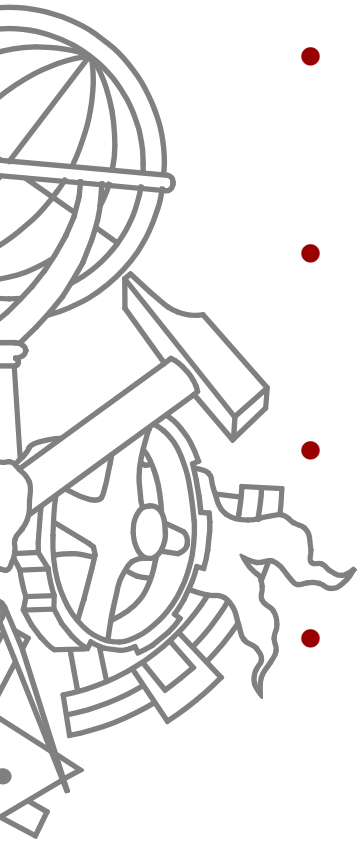
INTRODUCTION

What is a Distributed system?

An example DS

source: http://www.uweb.ucsb.edu/~fkart/research.htm

What is a Distributed System (DS)?

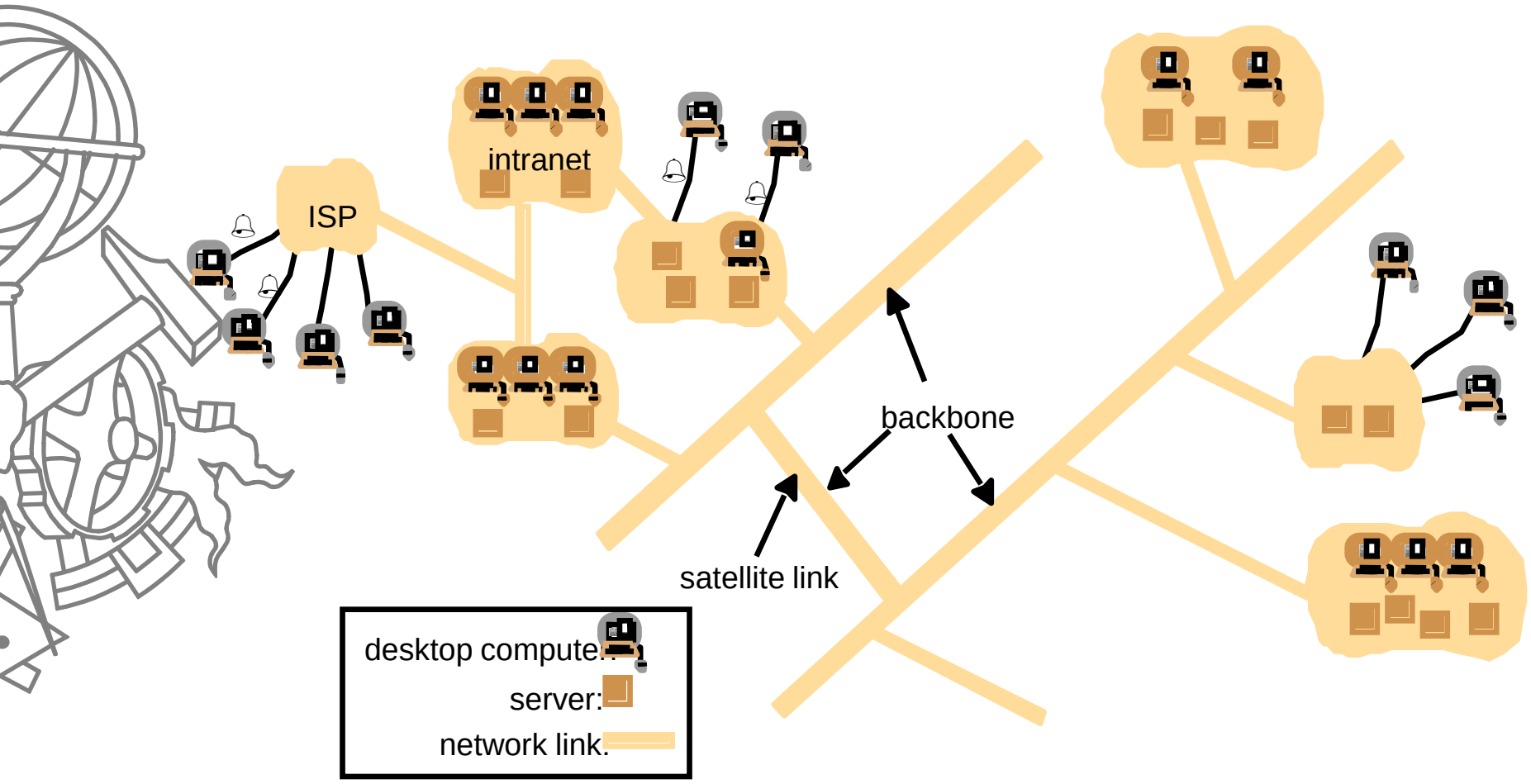


- A collection of independent computers that appears to its users as a single coherent system
  - A. Tanenbaum
- One in which components located at networked computers communicate and coordinate their actions by message passing
  - G. Coulouris
- You know you have one when the crash of a computer you have never heard of stops you from getting any work done
  - Leslie Lamport
- A distributed system is a system designed to support the development of applications and services which can exploit a physical architecture consisting of multiple, autonomous processing elements that do not share primary memory but cooperate by sending asynchronous messages over a communication network
  - Blair & Stefani

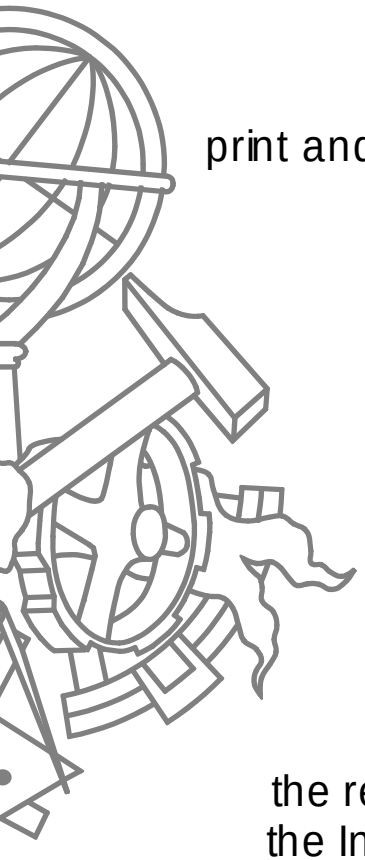
Caracterization of DS

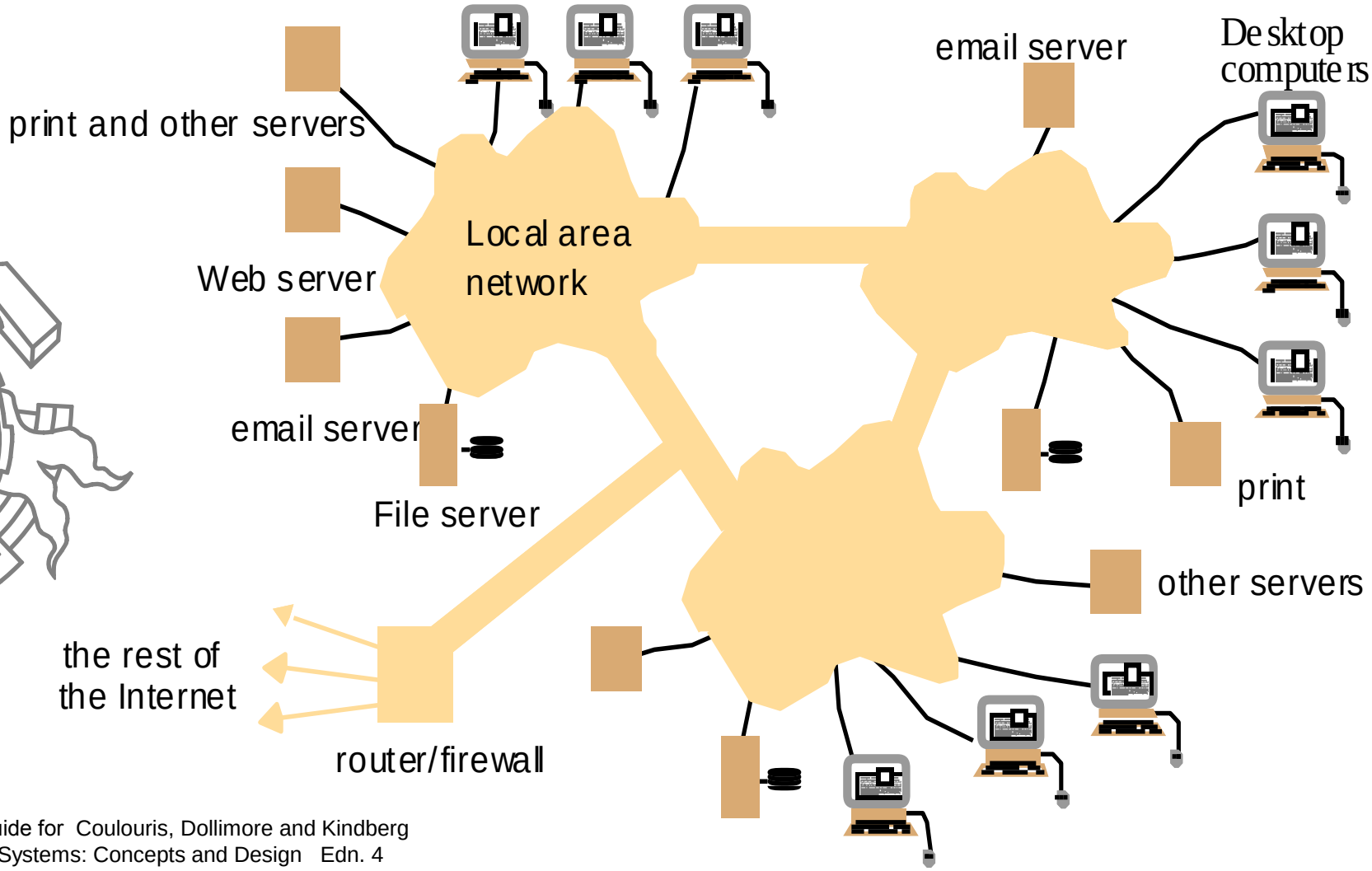
- According to P. Enslow [Computer 1978] a distributed processing system must have:
  - **multiple** general-purpose **resource** components that can be dynamically reassigned.
  - **physical distribution** of components using a two-party network protocol for **communication**.
  - high-level operating system, built on top of local operating systems, for unified control.
  - system **transparency** permitting services to be requested by name only.
  - cooperative **autonomy** characterizing the interactions between resources.

A typical portion of the Internet

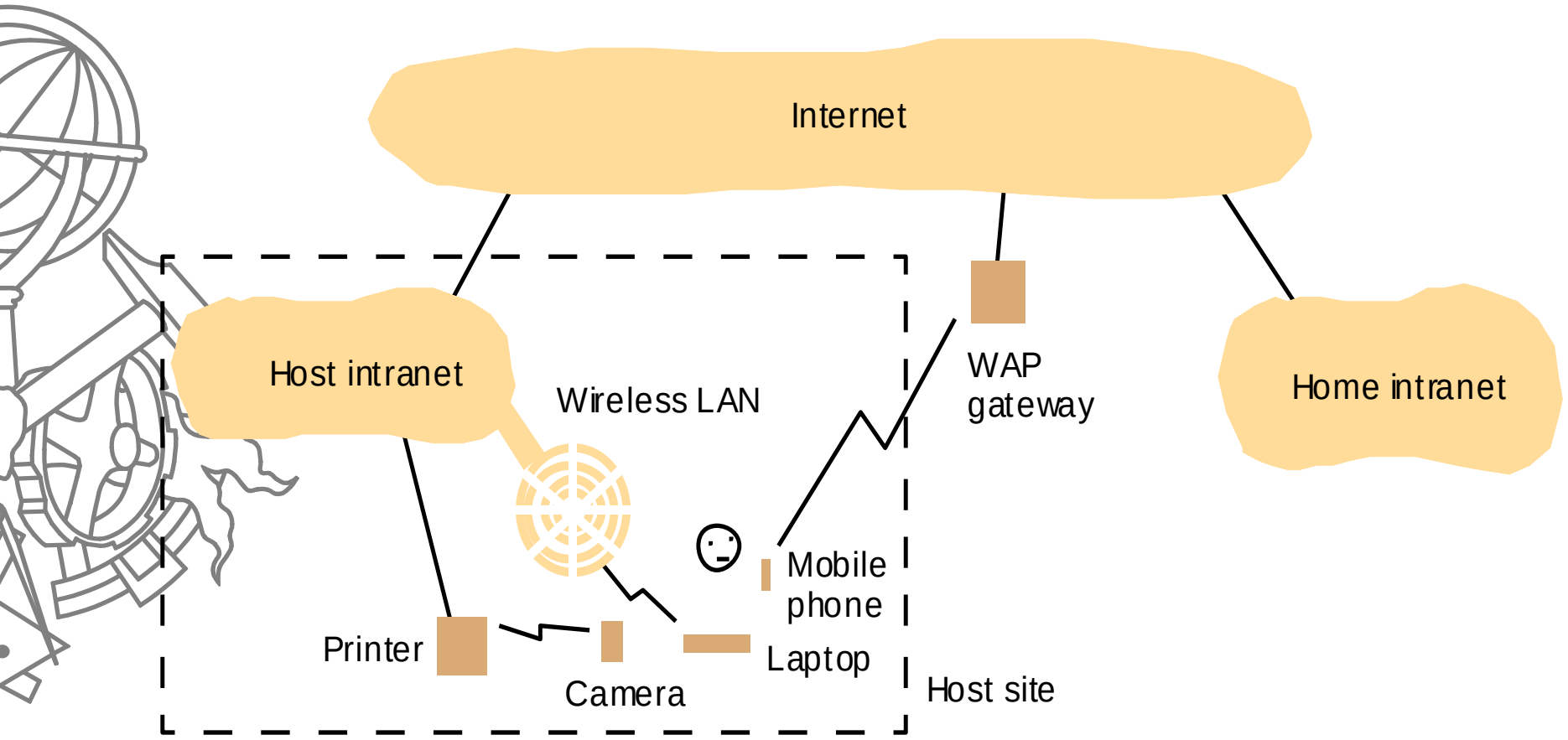


A typical intranet

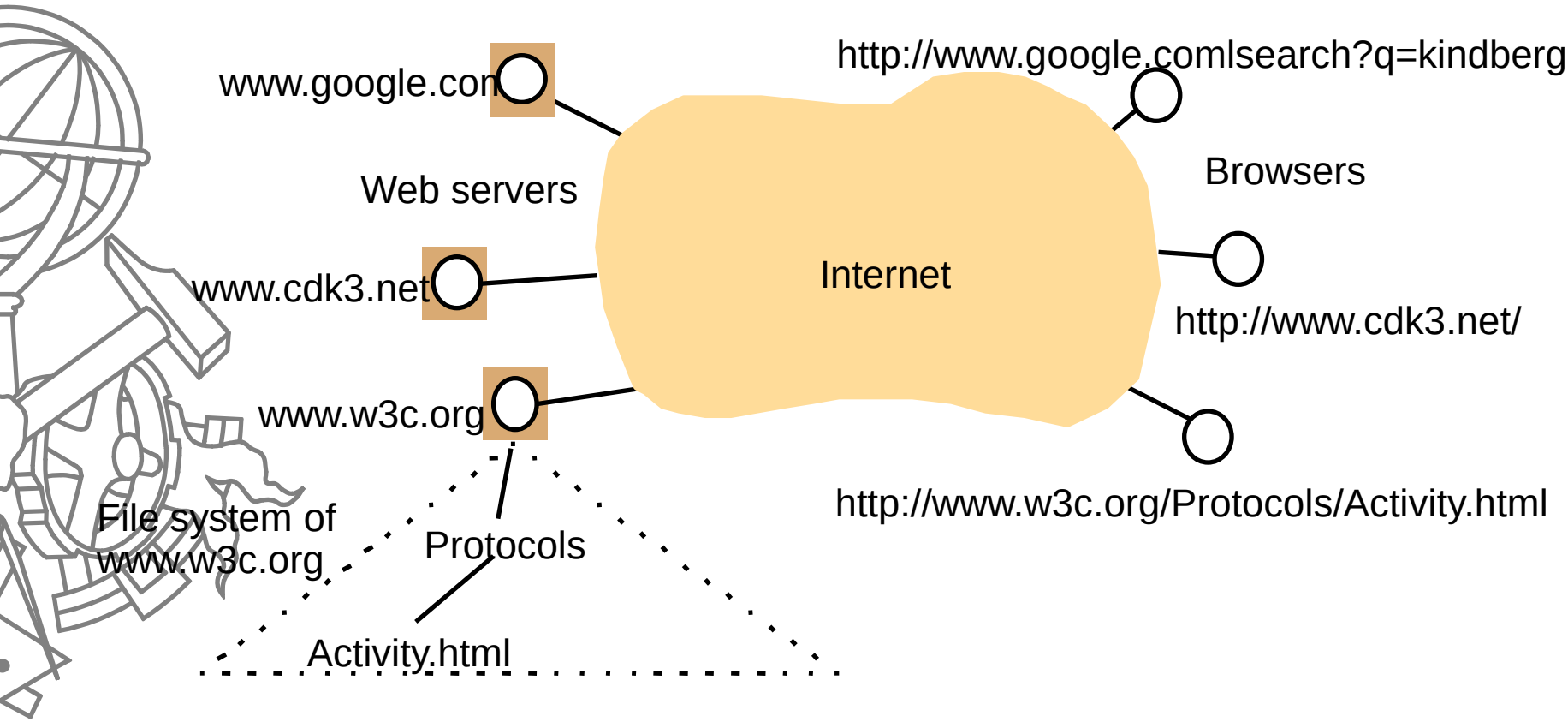




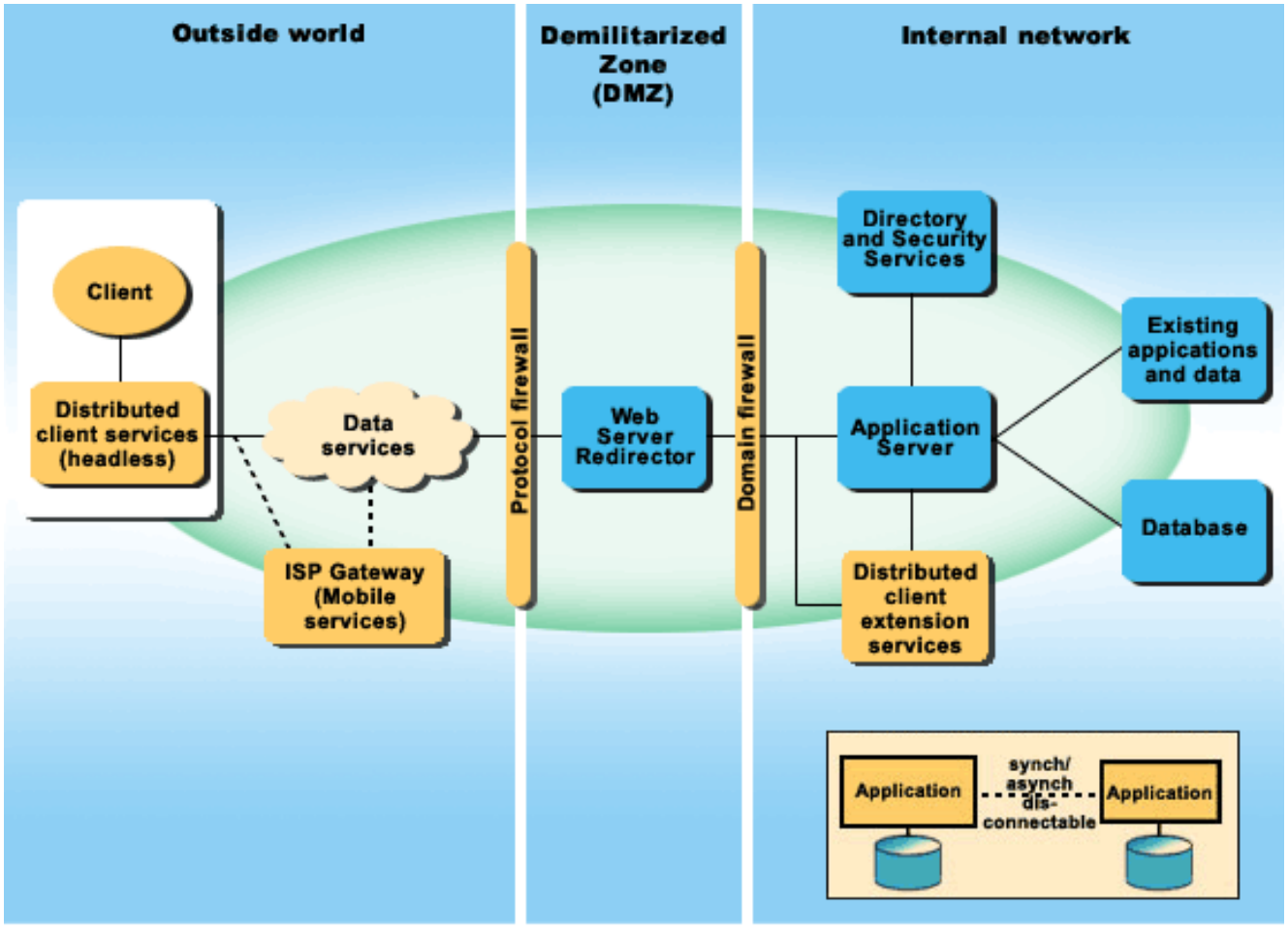
Portable and handheld devices in a DS



Web servers and web browsers

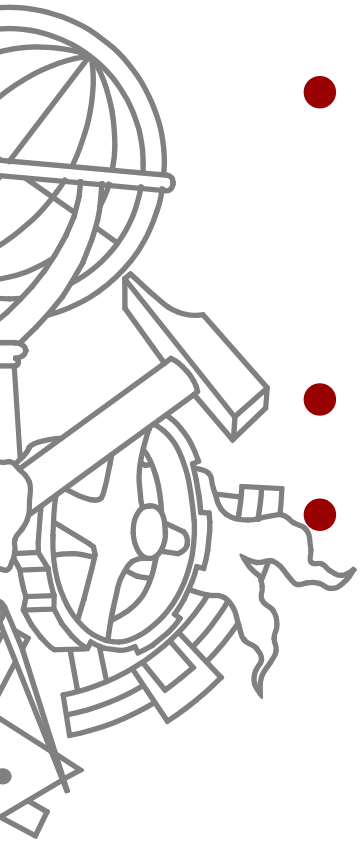


A typical DS: business web application



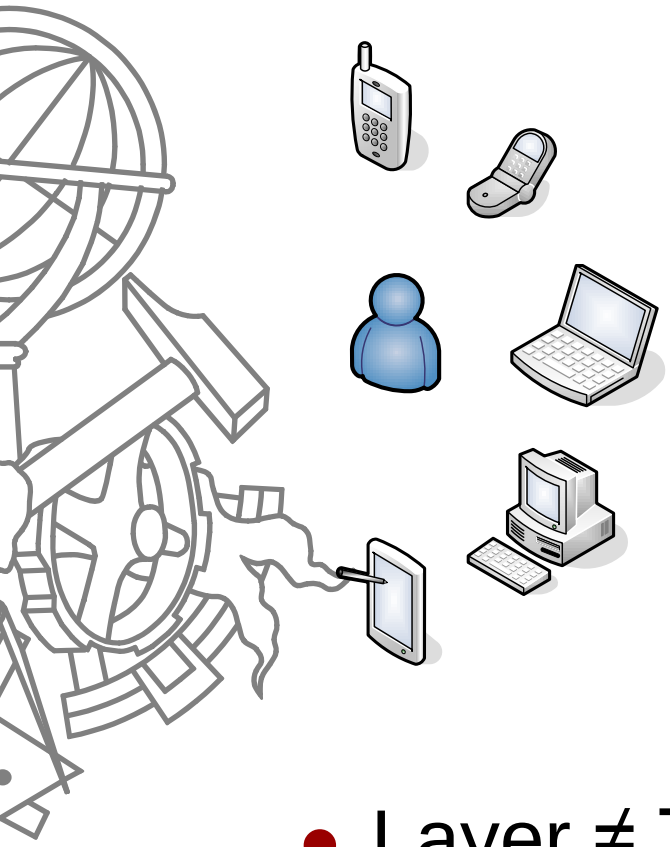
source: http://www.ibm.com/developerworks/patterns/access/at7-runtime.html

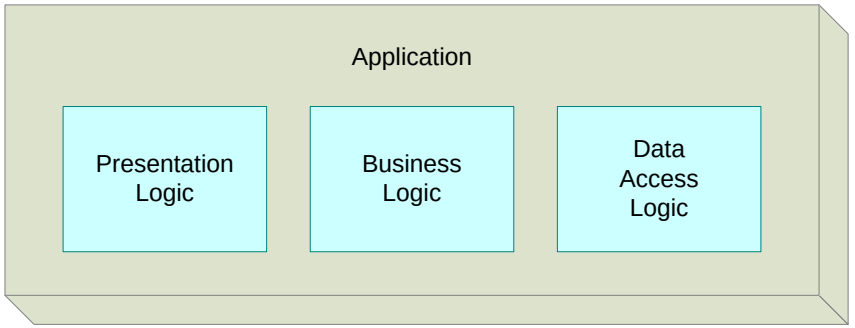
Layered applications vs. Distributed systems



- Layered (e.g., 3 layers) applications divide the application logically not necessarilly physically
- Distributed systems divide physically
- Typically a layered (distributed) application needs that the whole application (layers) is available

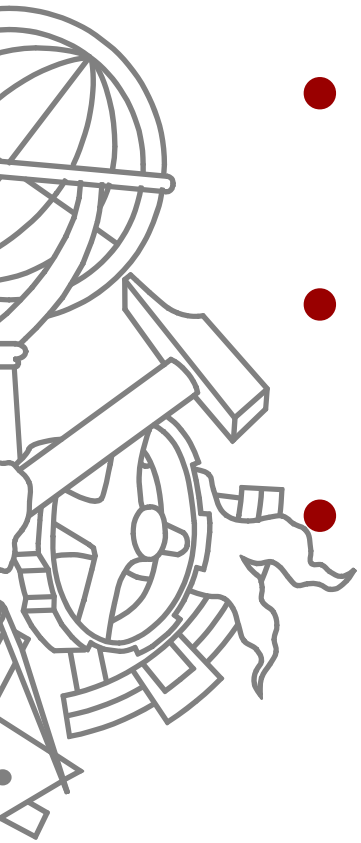
3-Layered application





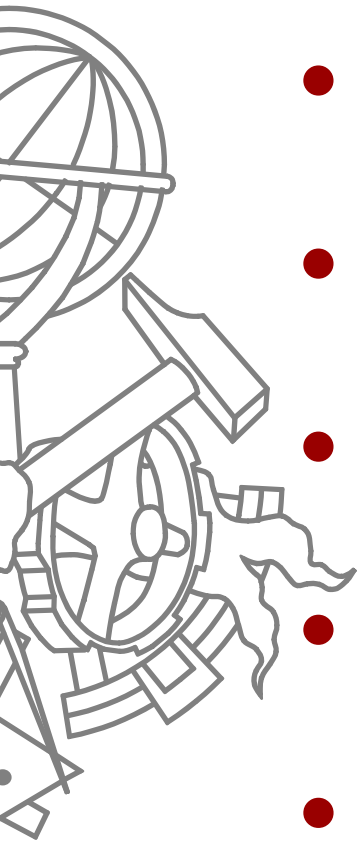
- Layer ≠ Tier

Distributed applications vs. Distributed systems



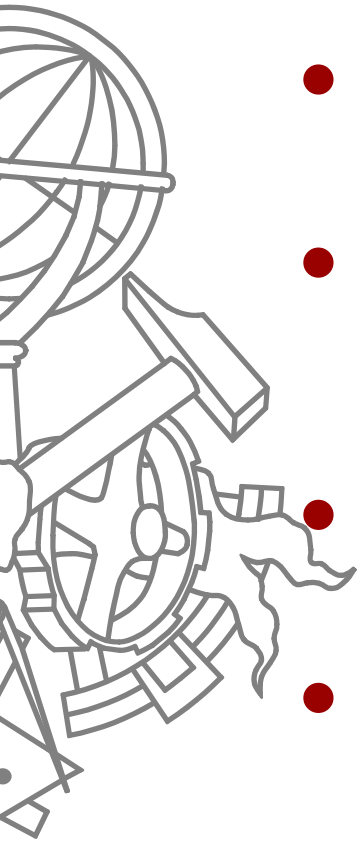
- A distributed application is a distributed system
- A system is also a collection of applications / hardware
- System ≥ application

Hot topics 1985



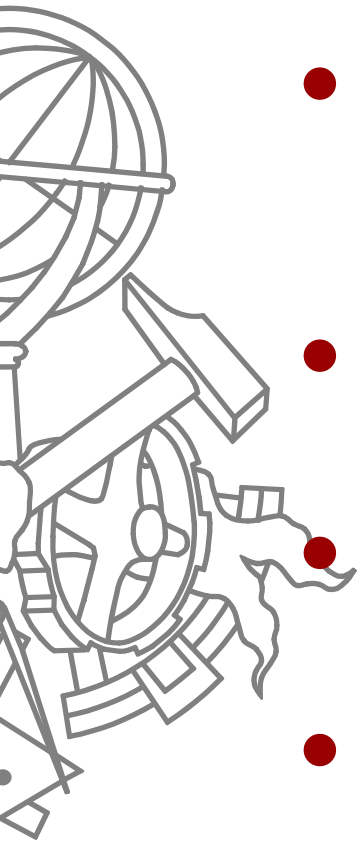
- Electronic mail
  - emphasis on interpersonal communication
- Distributed name servers
  - emphasis on locating objects
- Distributed file servers
  - emphasis on sharing data
- Transaction processing
  - emphasis on reliability and recovery
- Programming languages for distributed computing

Hot topics 1995



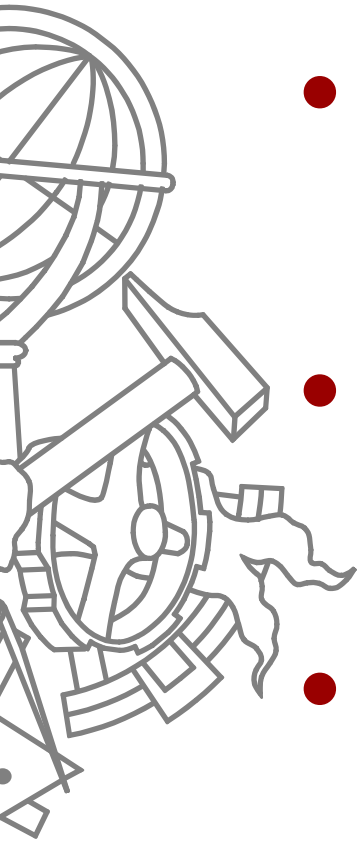
- Information superhighway
  - emphasis on high-speed networks and coverage
- Enterprise middleware
  - emphasis on coherence, interoperability, vendor-independence, standards (e.g. CORBA, DCE)
- Networks of workstations
  - emphasis on resource sharing
- Mobile Computing
  - emphasis on portable computers and wireless networks

Hot topics 2005



- Autonomic computing
  - emphasis on self-configuration, self-monitoring, …
- Peer-to-peer computing
  - emphasis on protocols and resource sharing
- Sensor and ad-hoc wireless networks
  - emphasis on small, self-organizing devices
- Ubiquitous computing
  - emphasis on anytime, anywhere access

Hot topics 2007

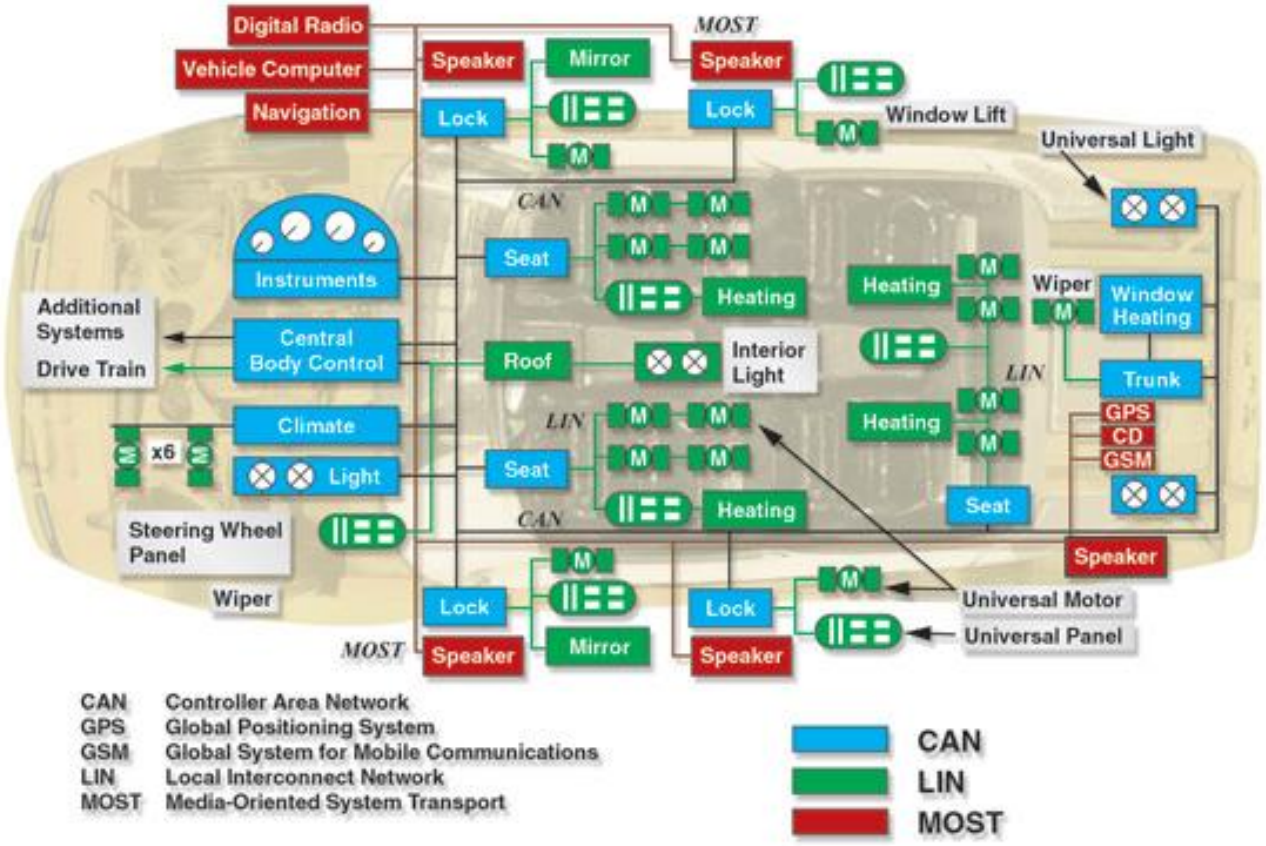


- Data center computing
  - emphasis on scalability, fault-tolerance, manageability
- Autonomic computing
  - emphasis on self-configuration, self-monitoring, …
- Social networks
  - emphasis on casual information sharing

Some Recent Applications

- Drive by wire, TTP, CAN
- Sensor Networks:
  - A computer wireless network of *spatially distributed devices using sensors to monitor temperature, sound,* vibration, pressure, motion or pollutants
  - Intelligent Dust
- Smart Labels:
  - Identification of packages
  - Get power from radio waves

Distributed, embeded, real-time system

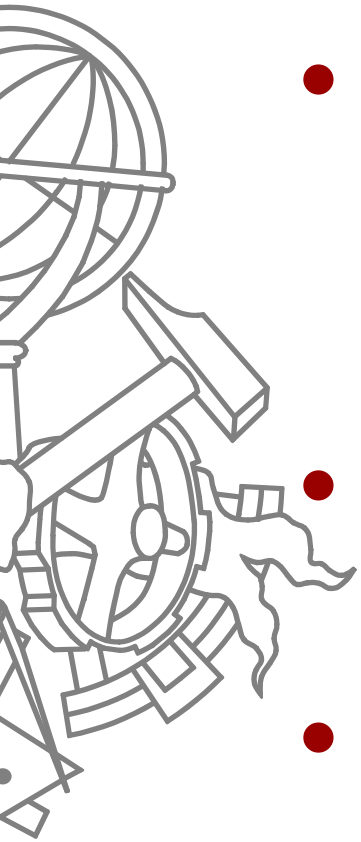


source: http://www.future-mag.com/0611/061113.asp

Give examples of a Distributed system you use?

CARACTERIZATION OF DISTRIBUTED SYSTEMS

Caracterization of DS



- Main features
  - geographical distribution of autonomous computers
  - communication through cable/fibre/wireless/... connections
- Advantages
  - interaction, co-operation and sharing of resources
- Benefits
  - reduced costs, improved availability and performance

Caracterization of DS (2)

- Advantages over centralized systems:
  - naturally distributed information
  - Autonomy
  - availability/reliability
  - modular growth
  - integration of existing systems
  - Capacity
  - cost/performance
  - resource sharing
  - guaranteed response

- Note: these are only potential advantages!

Should I build a DS?

- Why distribute?
  - Sharing
    - Resources
    - Devices
  - Reliability/Availability
  - Performance
  - Load balancing
  - Scalability
  - Openness

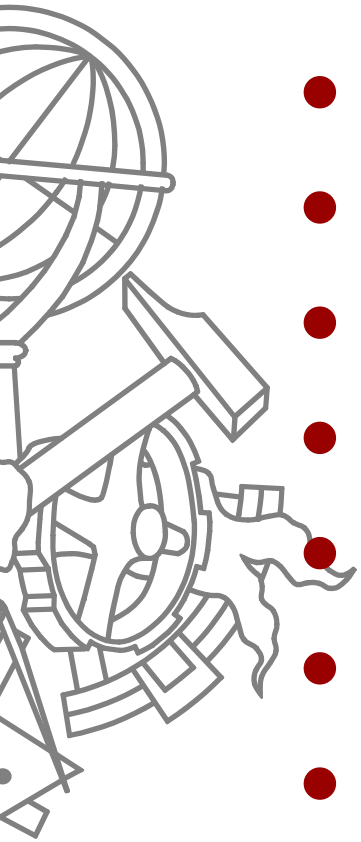
Should I build a DS? (2)

- Why not distribute?
  - System management
  - Overall complexity
  - Communication overhead
  - Security

Distributed vs. Centralized Systems

- What makes distributed systems harder:
  - variations in communication bandwidth/delay
  - absence of global knowledge
  - no global clock
  - partial failures
  - heterogeneous components
  - insecure communication
  - multiple administrative domains
  - size and complexity

Distributed Systems' Issues

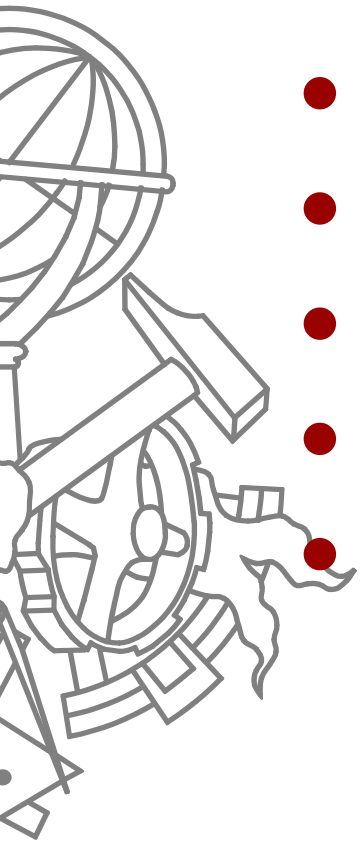


- Heterogeneity
- Security
- Scalability
- Failure handling
- Concurrency
- Message passing
- Naming
- Transparency

Issues : Heterogeneity

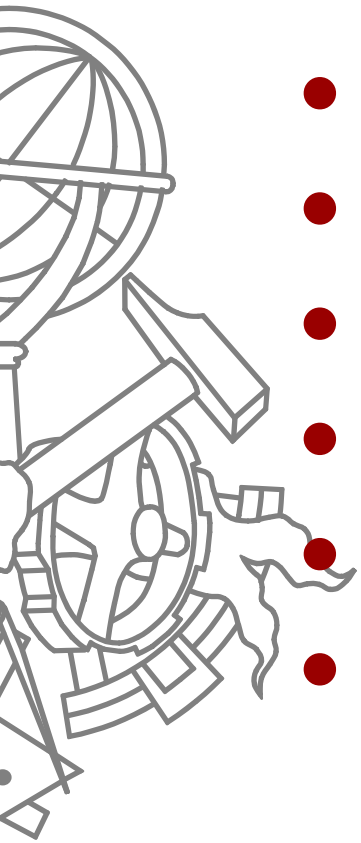
- Diversity of
  - Networks / network protocols
  - Computer hardware
  - Operating systems
  - Programming languages
  - Developers
- Middleware

Issues : Security



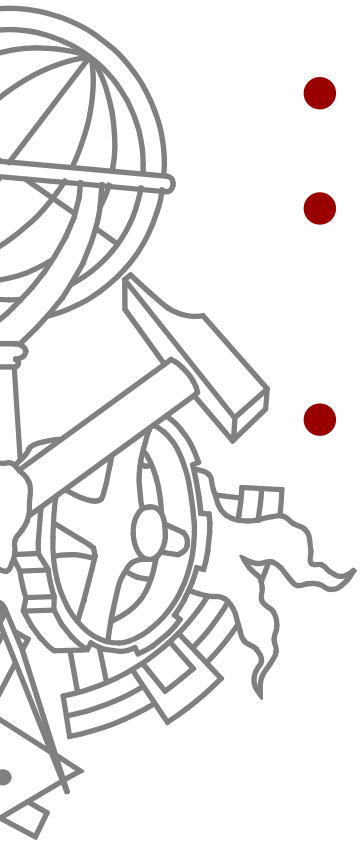
- Encryption
- Authentication
- Access rights
- Denial of service
- Mobility

Issues : Failure handling



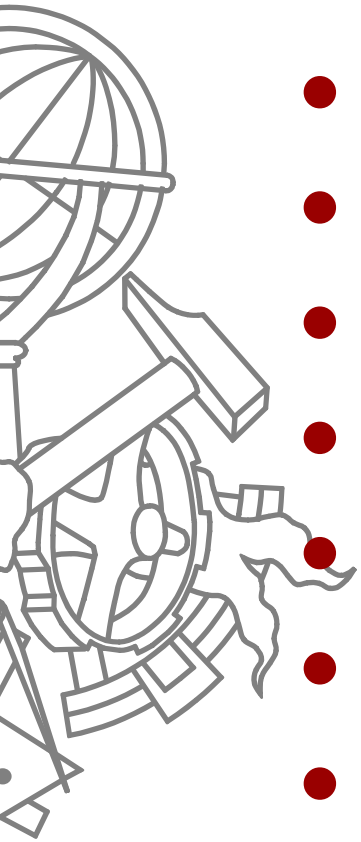
- Failures in distributed systems are partial
- Detecting failures
- Masking failures
- Tolerating failures
- Failure recovery
- Redundency

Issues : Concurrency



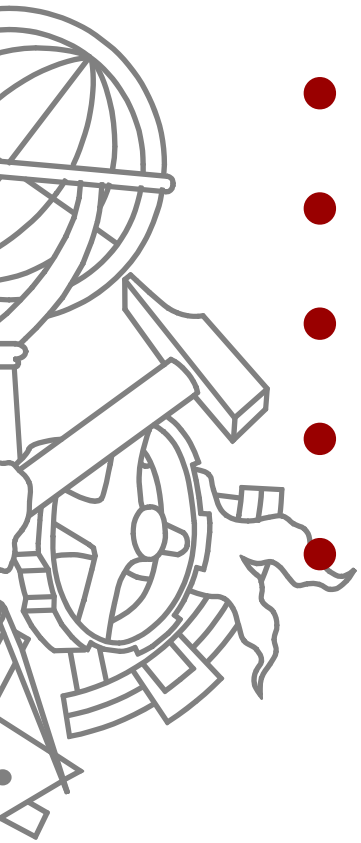
- Services are shared by clients
- Multiple, concurrent access to the same service/data
- ACID – Transactions
  - Atomicity
  - Consistency
  - Isolation
  - Durability

Issues : Message passing



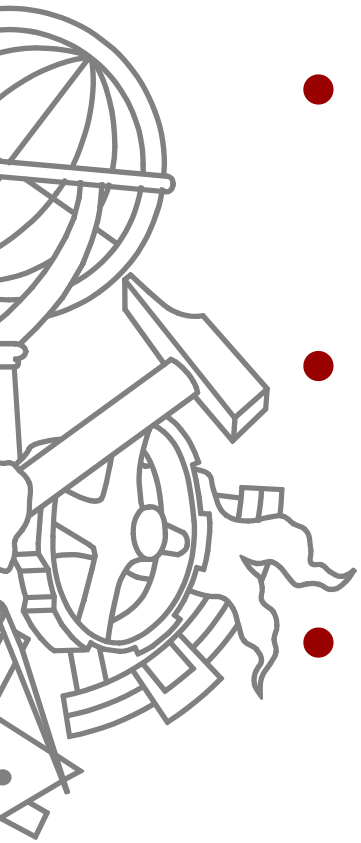
- Shared memory
- Shared database
- Blackboard
- Communication API (e.g., sockets)
- RPC
- Remote Objects
- Unicast/multicast

Issues : Naming



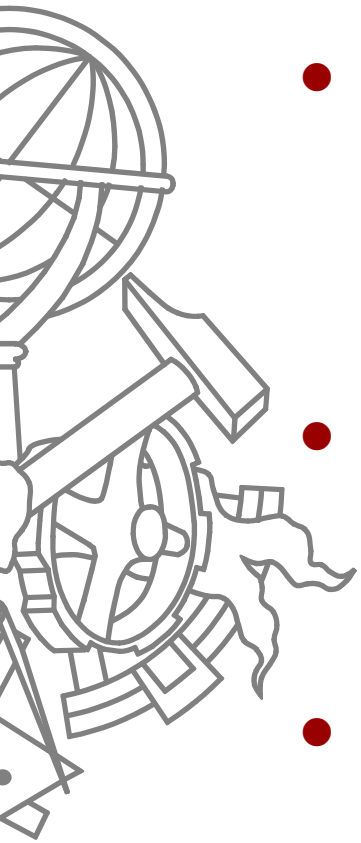
- Identify resources
- URLs
- IP addresses
- Naming services
- Lookup

Issues : Transparency



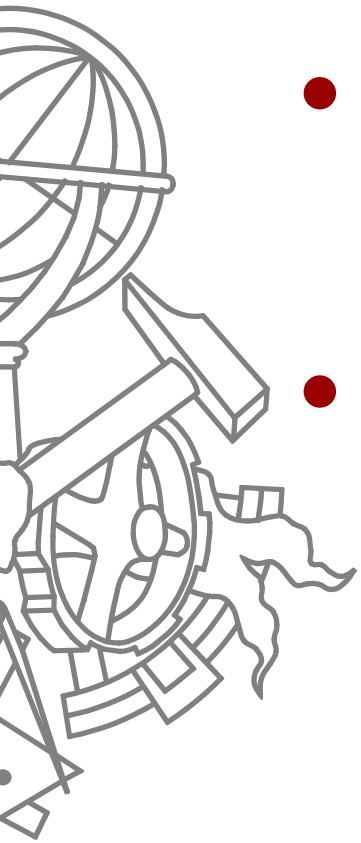
- Access transparency
  - Local and remote resources can be accessed using identical operations
- Location transparency
  - Resources can be accessed without knowledge of their physical or network location
- Concurrency transparency
  - Several processes can operate concurrently using shared resources without interference between them

Issues : Transparency (2)



- Replication transparency
  - Multiple instances of resources can be used to increase reliability and performance without knowledge of the replicas by users or application programmers
- Failure transparency
  - Concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components
- Mobility transparency
  - Allows movement of resources and clients within a system

Issues : Transparency (3)

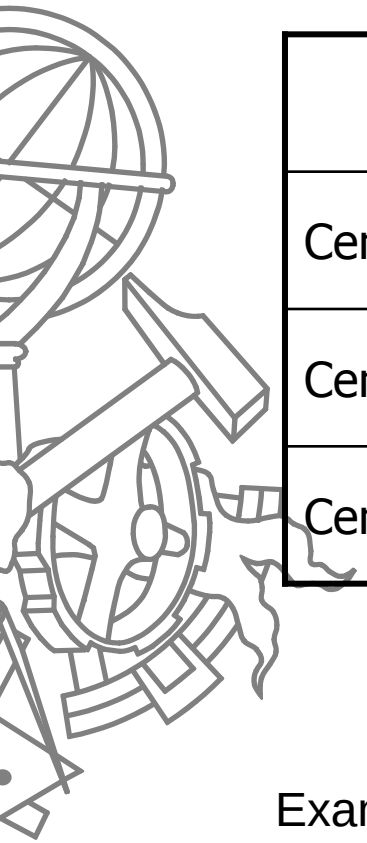


- Performance transparency
  - Allows the system to be reconfigured to improve performance as loads vary
- Scaling transparency
  - Allows the system and applications to expand in scale without change to the system structure or the application algorithms

Issues : Scalability

- System remains operational and effective despite changes in numbers of resources and users
  - Performance loss
  - Availability of hw/sw resources
  - Performance bottlenecks

Scalability Problems

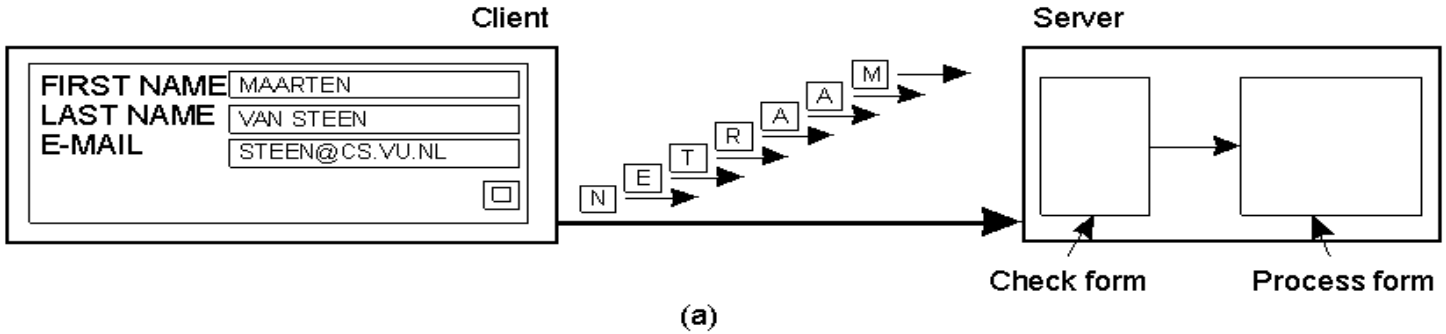


Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

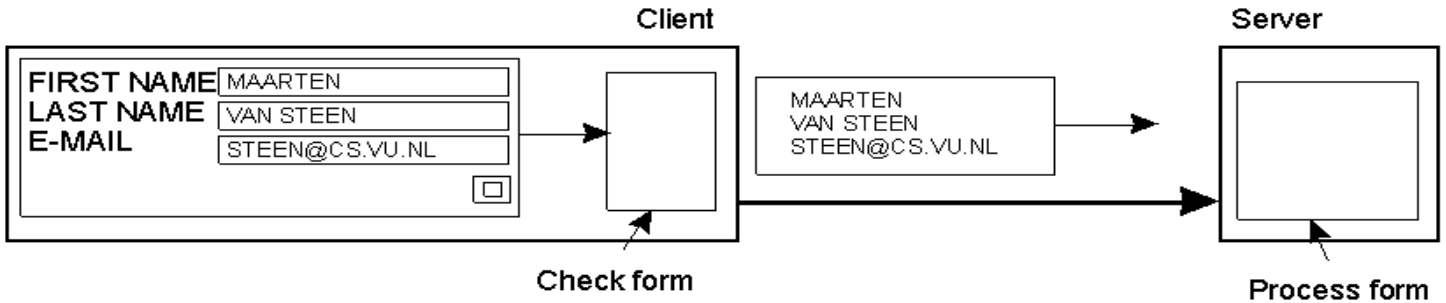
Examples of scalability limitations.

Scaling Techniques

- The difference between letting:
  - a server or
  - a client check forms as they are being filled



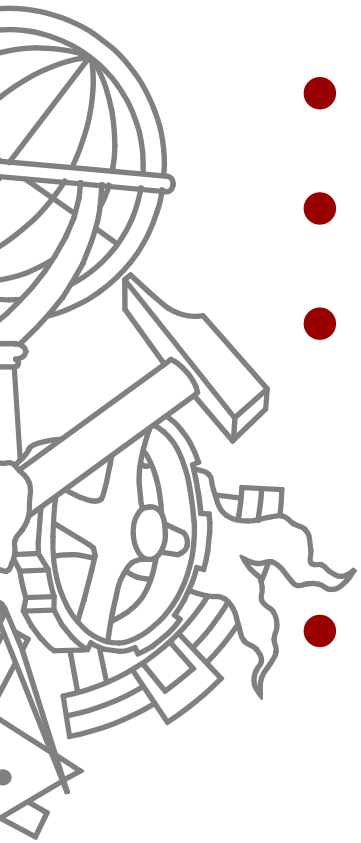
(a)



(b)

Can you identify these issues in na example DS?

Bibliography



- Chapter 1 Tanenbaum
- Chapter 1 & 2 Coulouris
- Wikipedia, “Distributed computing” http://en.wikipedia.org/wiki/Distributed_computing
- Enslow, P. (1978) *What is a "Distributed" Data processing System?*, **Computer**, January 1978.