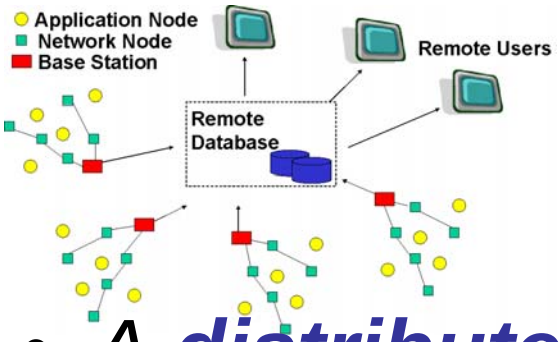
Distributed Systems
Conclusions & Exam

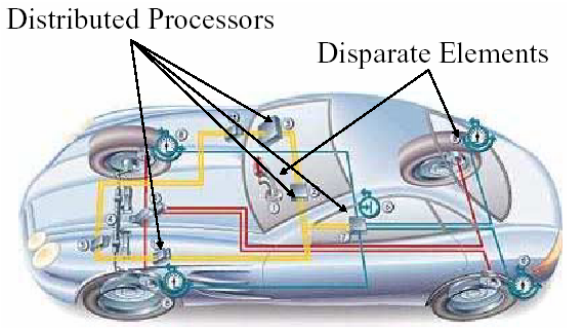
Brian Nielsen

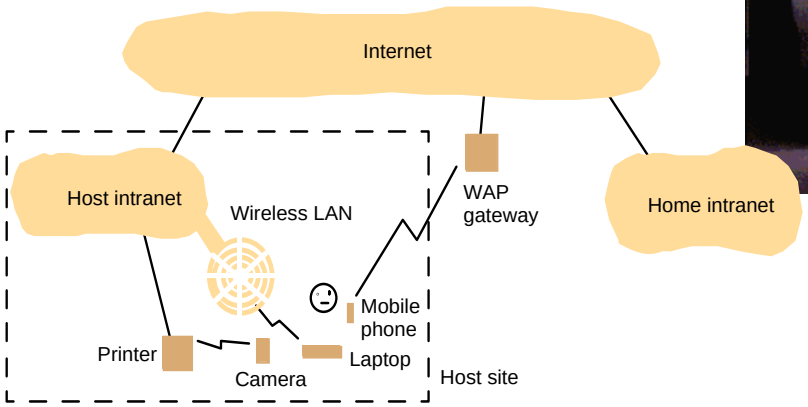
bnielsen@cs.aau.dk



Definition

- *A **distributed system** is the one in which hardware and software components at **networked computers** communicate and coordinate their activity only by **passing messages.***





Consequences

- Concurrent execution of processes
  - Users work independently & share resources
  - non-determinism, race-conditions, synchronization, mutex, deadlock, liveness, …
- No global clock
  - Each computer has its own clock
  - There are limits to the accuracy with which computers in a network can synchronize their clocks
- No global state
  - Coordination is done by message exchange
  - Generally, there is no single process in the distributed system that would have a knowledge of the current global state of the system
- **Units may fail independently.**
  - Network faults can result in the isolation of computers that continue executing
  - A system failure or crash might not be immediately known to other systems

Why a Distributed System?

- Resource Sharing
- Functional distribution
  - computers have different functional capabilities yet may need to share resources
    - Client / server
    - Data gathering / data processing
- Inherent distribution in application domain
    - cash register and inventory systems for supermarket chains
    - computer supported collaborative work
- Economics
  - collections of microprocessors offer a better price/ performance ratio than large mainframes

Why a Distributed System?

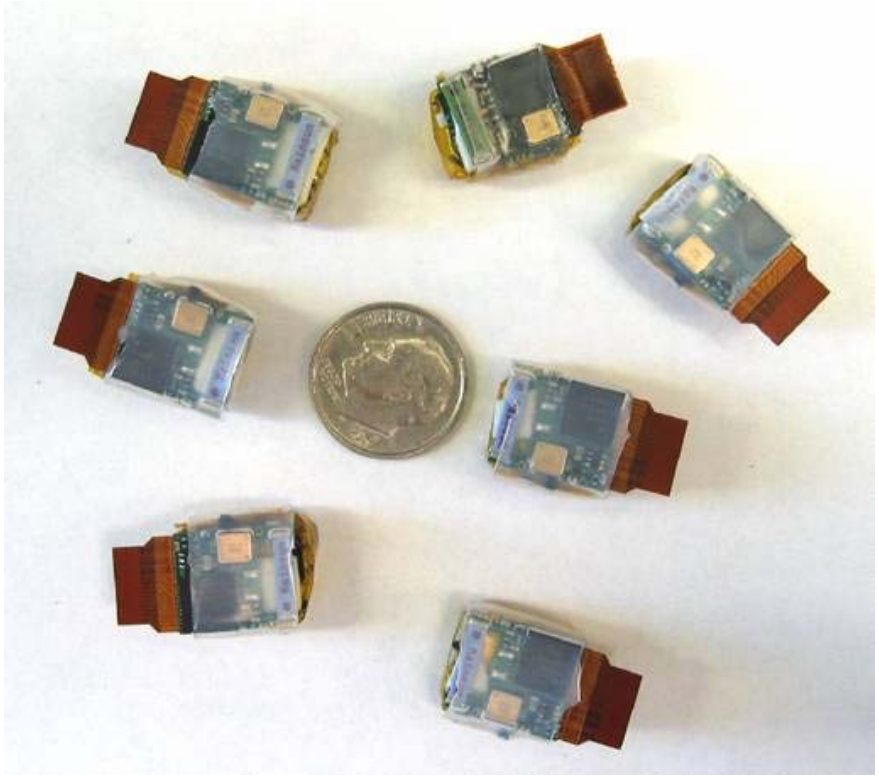
- Load balancing
  - assign tasks to processors such that the overall system performance is optimized
- Replication of processing power
  - independent processors working on the same task
- Increased Reliability
  - Exploit independent failures property and
  - Redundancy

Why Not?

- Multiple Points of Failures
  - Leslie Lamport: *”a distributed system is is one in which the failure of a computer you didn’t even know existed can render your own program unusable”*
- Complexity
  - Advanced solutions to
    - Concurrency, asynchrony, non-determinism,
    - paritial-failures,
  - message passing, performance bottlenecks
- Security
- Administration (multiple adm. organizations)

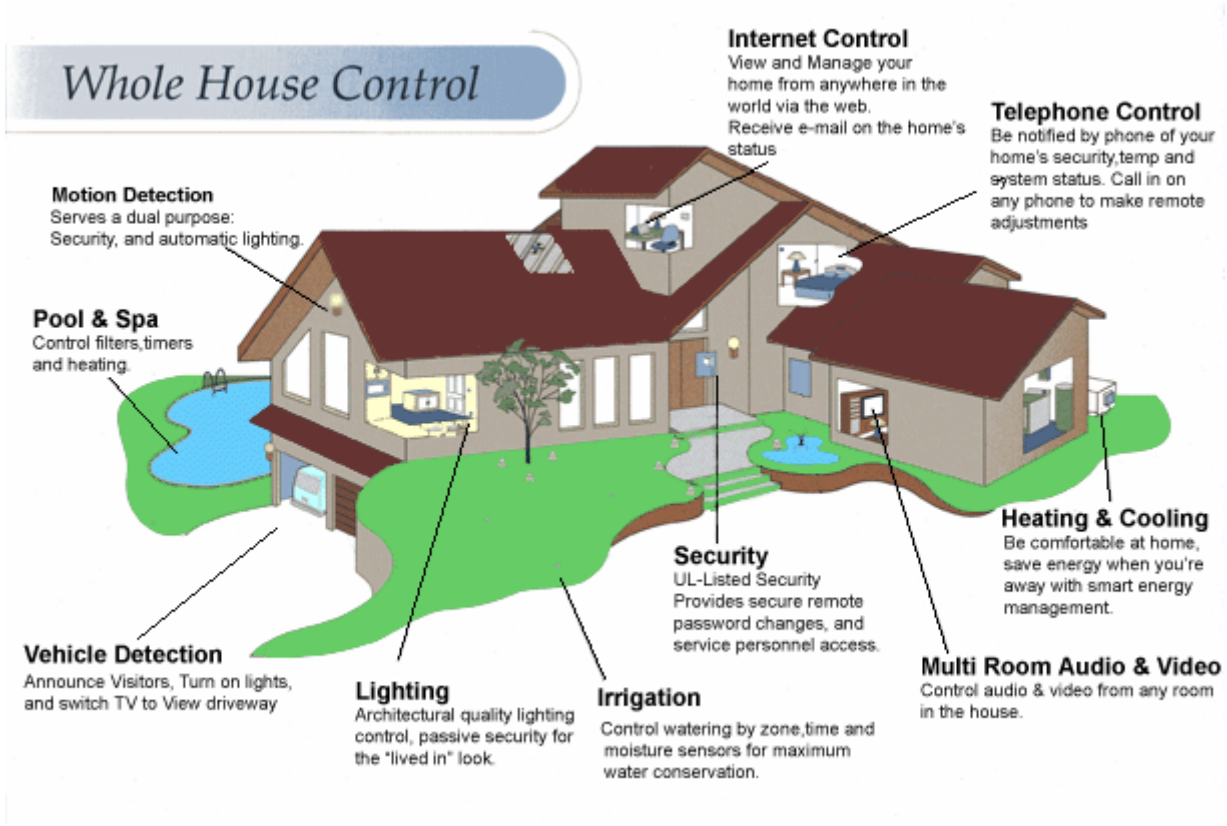
Trends

- Increasing Integration and Convergence from the very small to the very big



Ubiquitous Computing

- “existing or being everywhere at the same time”
- Embedding computation into the environment and everyday objects would enable people to interact with information-processing devices more naturally and casually than they currently do, and in ways that suit whatever location or context they find themselves in.



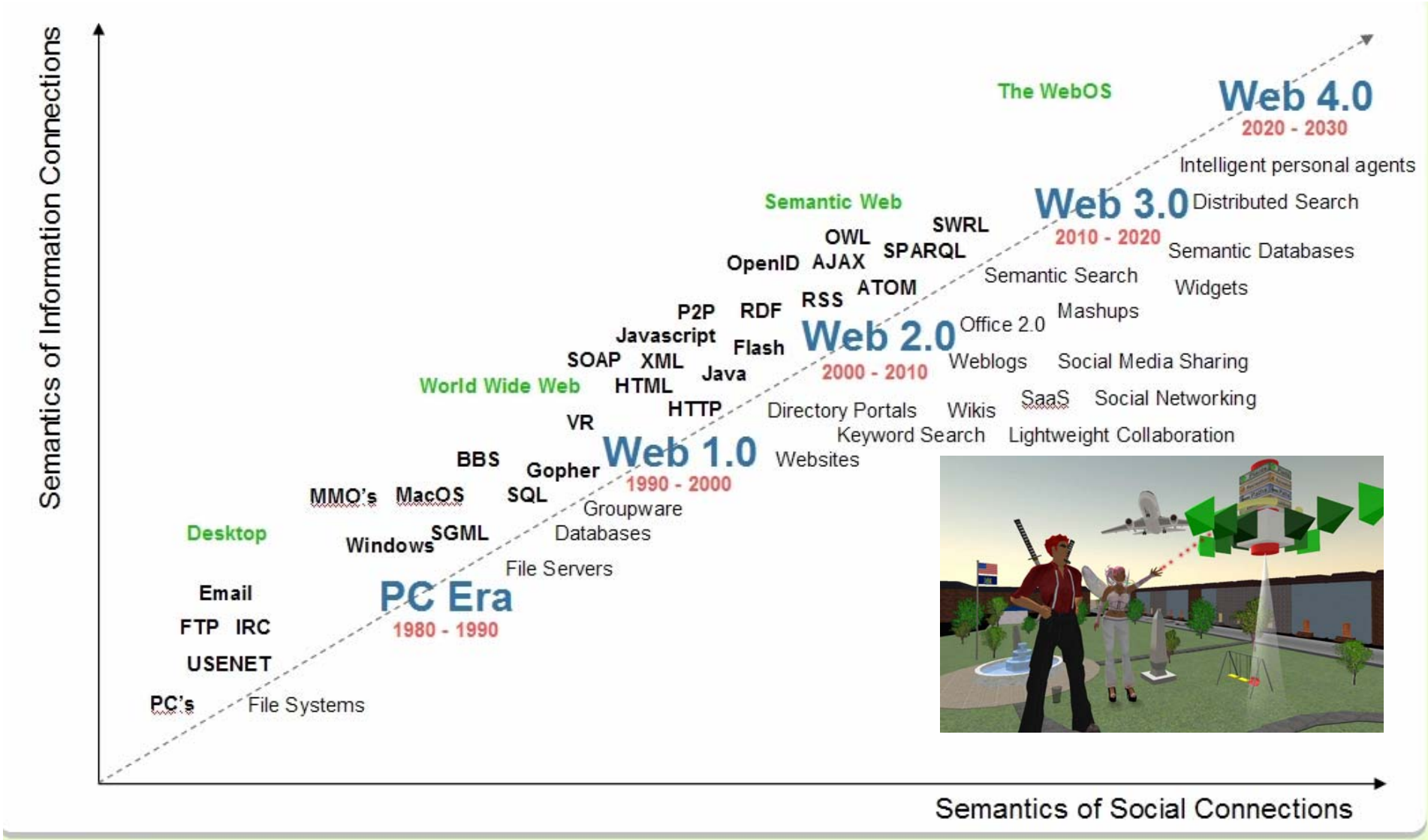
Mobile Internet

- Mobile Agents, Autonomic Computing: autonomous active objects, runtime code migration, service discovery, content distribution and delivery, context-aware computation, intelligence

Large-scale global computing

- Scalable, secure, heterogeneous middleware with QoS provisioning
- GRIDS, cluster computing
- Web-Services
- p2p

Web 3.0



Source: Radar Networks & Nova Spivack, 2007 – www.radarnetworks.com

Study Regulations

Purpose: That the student obtains knowledge about concepts in distributed systems, knowledge about their construction, and an understanding of advantages and disadvantages of their use.

Contents:

- Structure of distributed systems.
- Distributed algorithms.
- Distributed and parallel programming.
- Fault tolerance.
- Examples of one or more distributed systems.

Course Plan

Lecture	Topic
1	**Introduction to Distributed Systems**
2	**Programming Models I**
3	**Programming Models II**
4	**Distributed File Systems**
5	**Peer2peer Systems**
6	**Clock Synchronization**
7	**Distributed Mutual Exclusion & Election**
8	**Multicast communication**
9	**Consensus and study-exercises**
10	**Replication**
13	**Study Exercise**
11	**Web Services**
12	**Introduction to Grid Computing (Guest Lecture by Josva Kleist)**
14	**Study Exercise**
15a	**Conclusions and Exam Information**
15b	**Exam Questioning Hour / Spørgetime**

Learning Goals

- The student must at the concluding examination be able to
  - document knowledge and overview of the involved topics and concepts within distributed systems
  - use correct professional terminology in speech and writing
  - document knowledge about the fundamental properties of distributed systems, their architecture, and explain their consequences on system behavior and design
  - describe/explain basic prototypical distributed problems and distributed algorithms to solve these,
  - compare and evaluate different distributed algorithms and solutions wrt. semantic guarantees/precision, performance and fault-tolerance properties
  - demonstrate skills in realizing/implementing simple distributed systems or algorithms typically in the form of a distributed application.

The Exam

- PE Course
  - Evaluated as part of project exam with your project as starting point
  - Your examiner may include relevant material from the course
  - Know the pensum, consider studying the relevant chapters more intensively
- SE-Course
  - 20 min, Oral, pass-no pass grade with
  - random choice among 10 known topics
  - Read pensum intensively and do selected exercises, and the study exercise

Exam Questions

1. **Time in distributed systems [11.1-11.4].**
   Discuss algorithms to achieve clock synchronization in distributed system, with emphasis on either logical time or physical time.

2. **Mutex and elections [12.1-12.3]**
   Discuss the problems in performing mutual exclusion and leader election in distributed systems, and show mutex or leader elections algorithms.

3. **Multicast [12.4]**
   What are the advantages of multicast communication? Discuss either reliable multicast or ordered multicast algorithms (in both cases remember to discuss semantic models).

4. **Byzantine generals [12.5]**
   Explain what the Byzantine generals problem is. Present impossibility result for 3 Byzantine generals, 1 faulty as well as the solution for 4 Byzantine generals, 1 faulty.

5. **Remote Method Invocation [5.1-5.2, 5.5]**
   Give an introduction to the idea of RMI, and discuss the implementation principles.

6. **Distributed file systems [8.1-8.3]**
   Discuss what is the goal of distributed files systems, and describe SUN NFS.

7. **Replication [15.1-15.4]**
   Discuss the use of replication to achieve either fault tolerance or increased availability.

8. **Peer2peer [10.1-10.5]**
   Discuss the goal of Peer-to-Peer systems, and describe how searches in a Pastry net is performed.

9. **Study-exercise**

10. **Study-exercise**

END