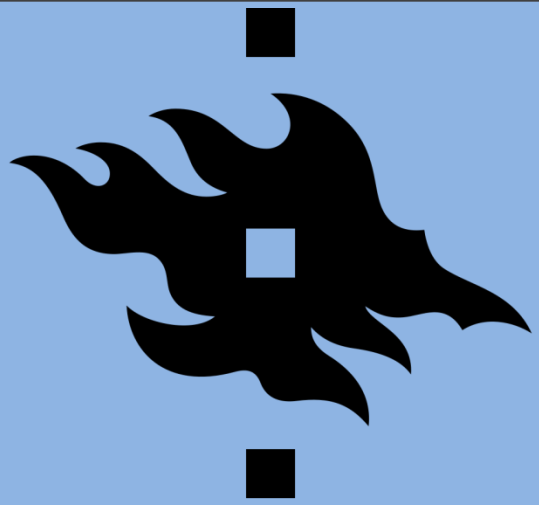


DISTRIBUTED SYSTEMS

Lecture 1 – Introduction to distributed systems

Huber Flores, Xiang Su, Pan Hui
{firstname.lastname}@helsinki.fi

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Helsinki, Finland, 2019.

Agenda

- Concept and definition
  - What is a distributed system
- Modelling
  - How to reason about correctness

Concept and definition

WHAT IS A DISTRIBUTED SYSTEM?

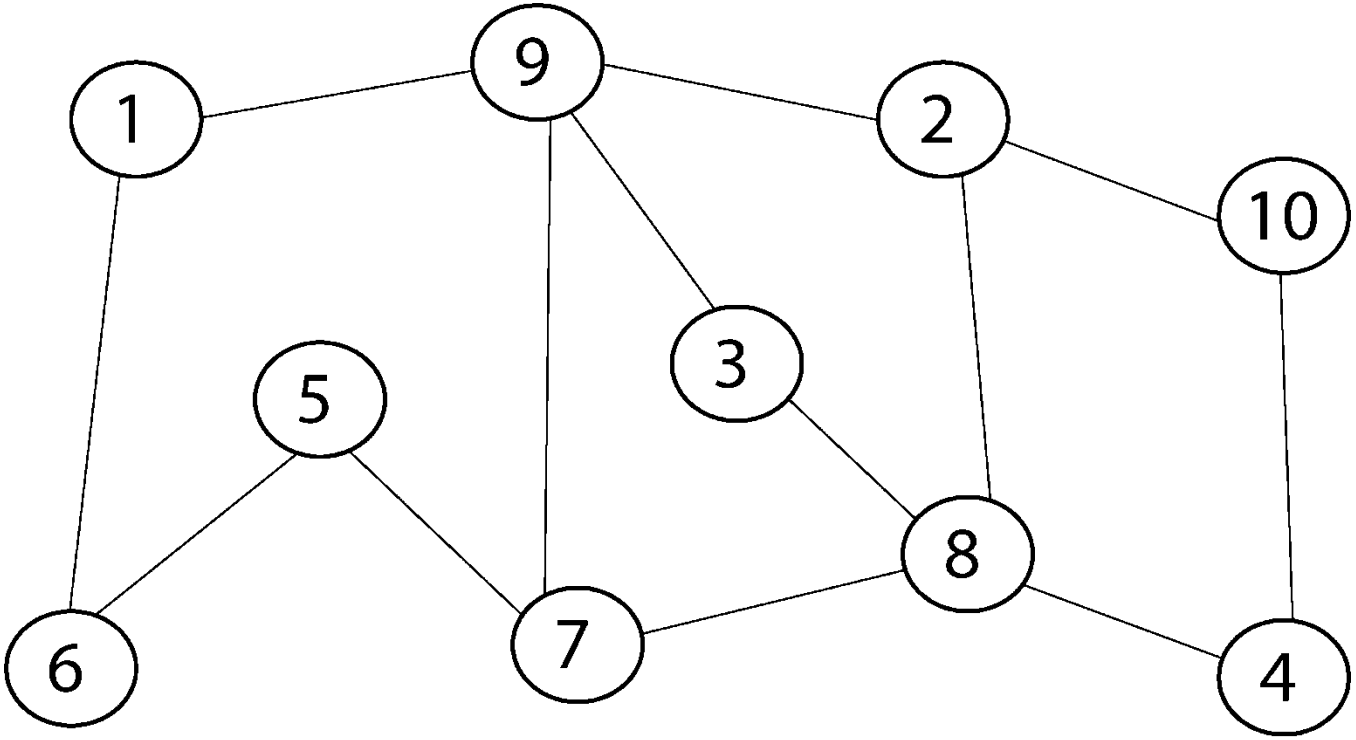
Examples of distributed systems

A few examples of distributed systems are:

- Google – *for big data processing*
- Facebook – *for content sharing*
- Skype – *for voice and video*
- BitTorrent (P2P Network) – *for sharing any file*
- Sensor Networks – *for monitoring of artifacts*
- Vehicular networking – *for traffic analysis*
- Surveillance systems – *for security*

What is a distributed system?

- Abstract view: A network of nodes/processes

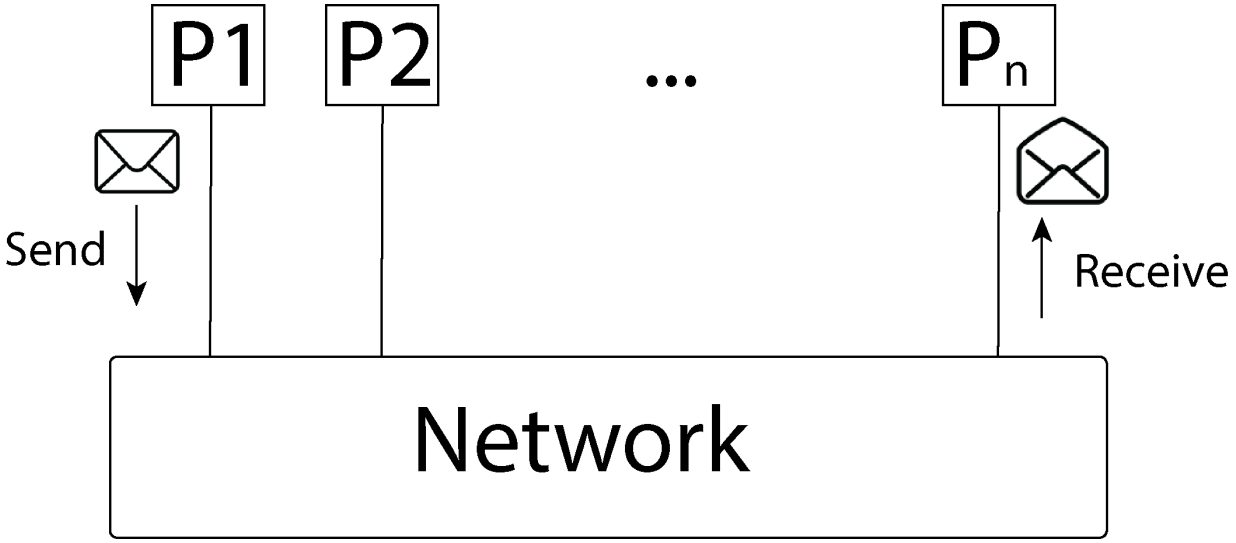


What is a distributed system?

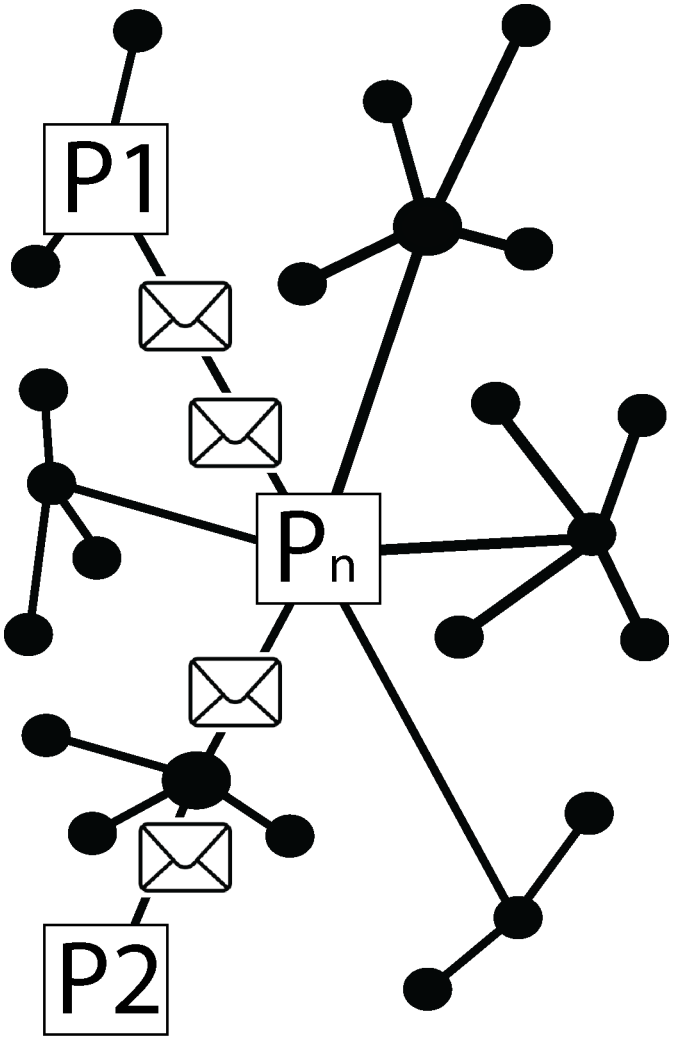
- Definition 1 (from Van Steen)
  - A distributed system is a collection of **autonomous computing elements** that appears to its users as a **single coherent system.**
- Definition 2 (from Gosh)
  - It is a **network of processes**.
  - The **nodes** are processes, and the **edges** are communication channels.
  - The processes (or computers) coordinate their activities, so that users perceive it as a **single, integrated computing service with a well-defined goal**.

What is a distributed system?

- A set of nodes, connected by a network, which appear to its users as a single coherent system (from Haridi)



What is a distributed system?



Coherent system, integrated service

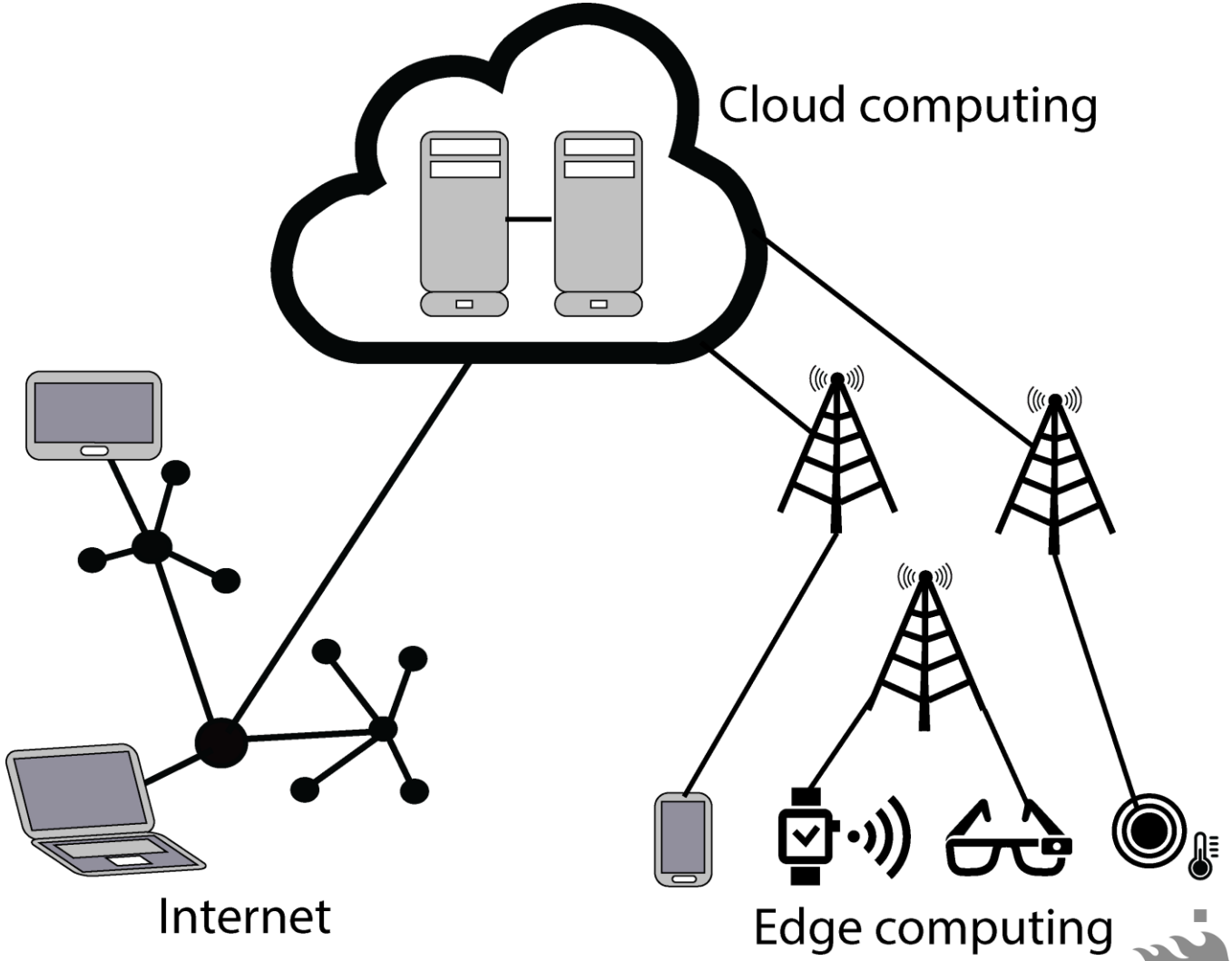
- End user cannot tell where the computation is performed
- End user (or the application) does not care where data is stored
- Replication is hidden
- Geographic distribution

- Goals
  - Sharing resources
  - Openness
  - Scalability
  - Availability

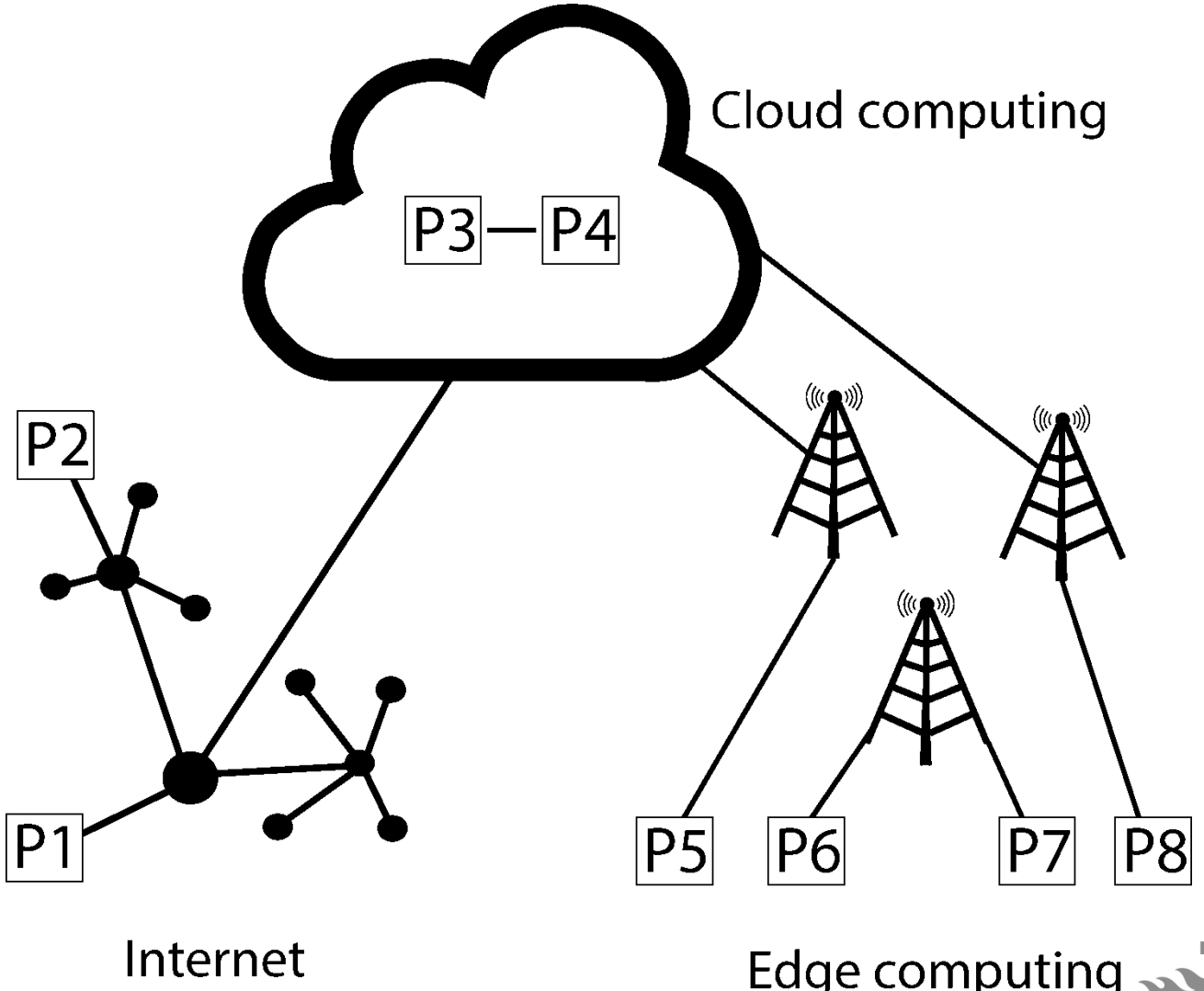
Why to study distributed systems?

- Facts:
  - It is hard to find systems that do not work in a distributed manner
    - Fault tolerance, reliability and better performance
  - Demand for resources has increased dramatically, a single computer cannot cope with it
    - Internet services
  - Our society works in a networked fashion

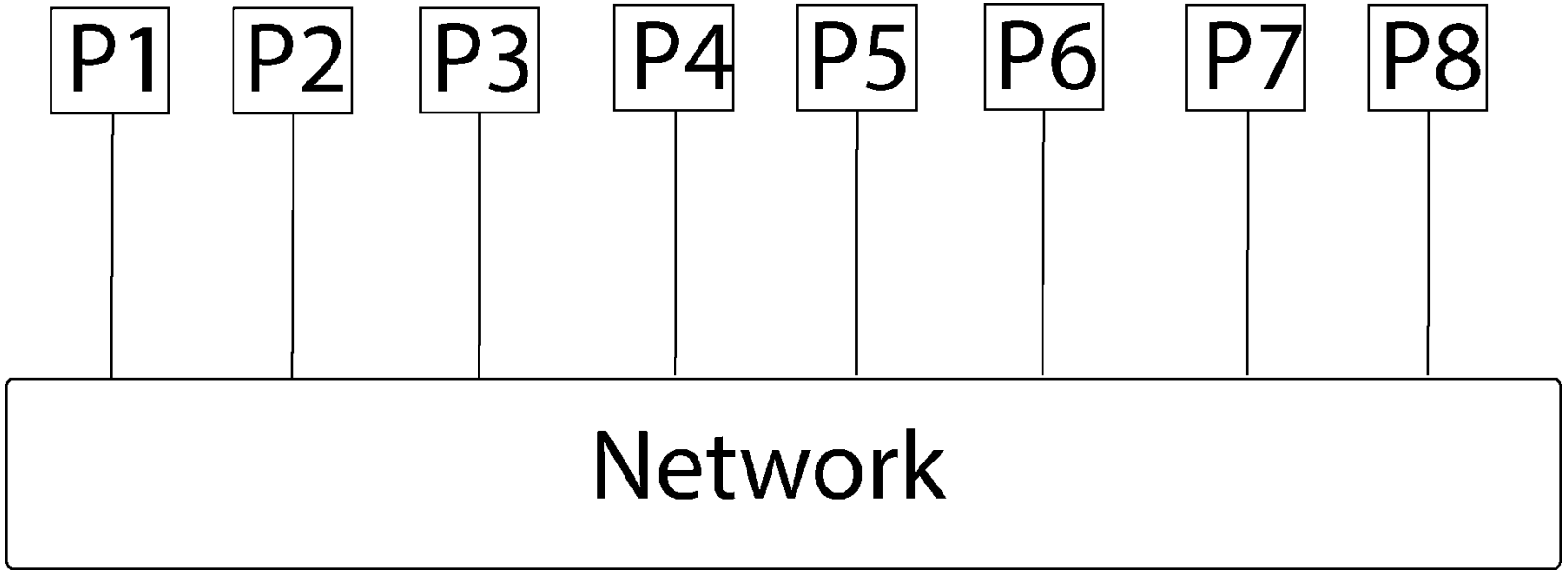
Why to study distributed systems?



Why to study distributed systems?



Why to study distributed systems?



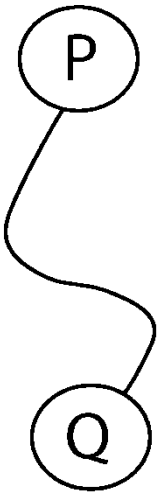
Distributed transparency

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

[source]: van Steen & Tannenbaum

Distributed computing structure

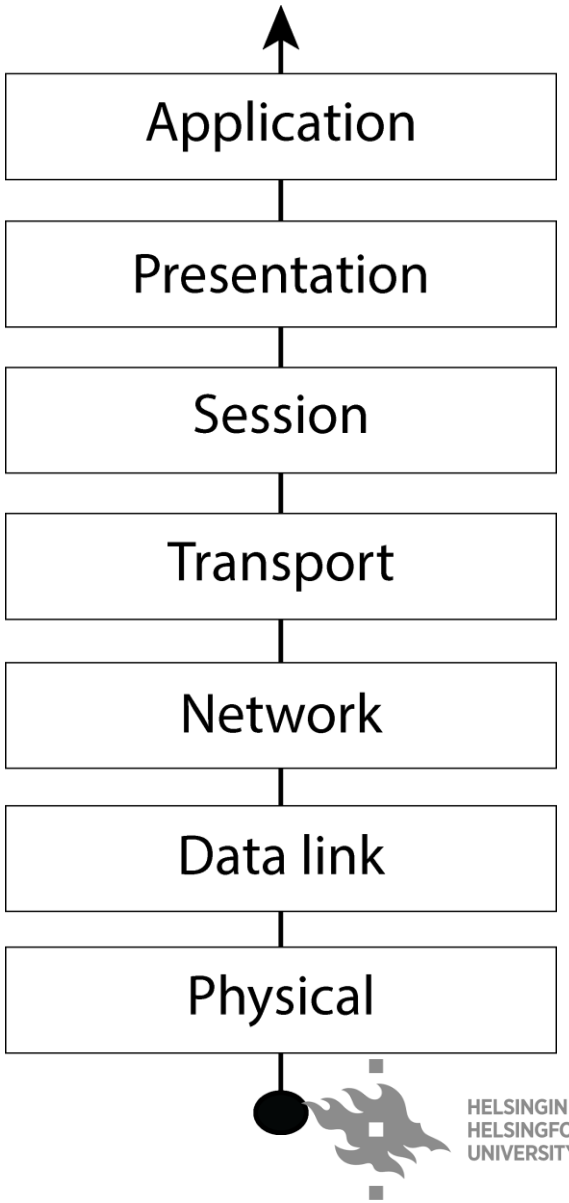
- Distributed computing relies on **inter-process communication**, which involves the various **layers of networking**.
  - Message passing
- Distributed computing helps create ***simple abstractions*** for these layers to facilitate program writing.
  - E.g. message, packet, datagram, frame



Create a reliable channel between P and Q that are 10,000 miles away

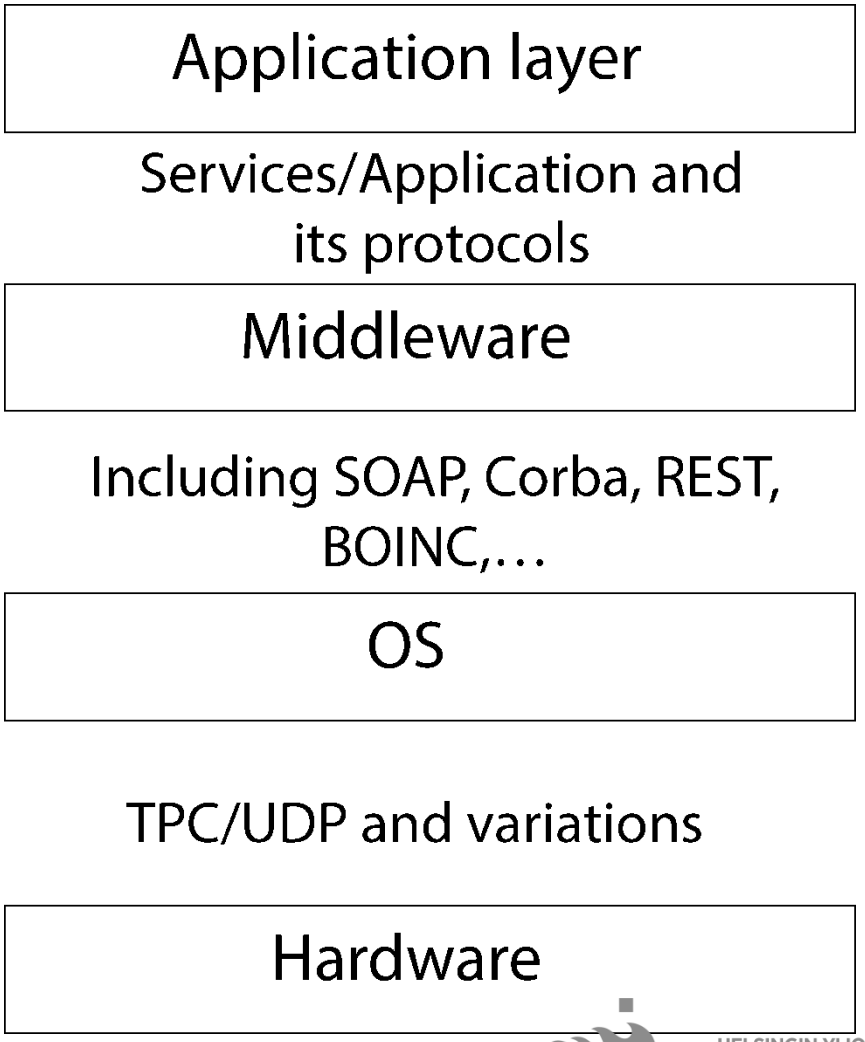
Network layers

- Message from sender goes through all the layers on its way
- Protocols/ functionality described on each layer separately
- Abstraction!!!



Architecture layers

- Different layers – different abstractions
- Assumptions about the lower layer's behaviour
- A process in each layer



Critical challenges

- Knowledge is local – no single central control point
- Clocks are not synchronized
- No globally shared address space
- Topology and routing – Everything is dynamic
- Scalability
- Processes and links fails
  - Fault tolerance, e.g., CODA File System

Common issues

- Leader election
- Mutual exclusion
- Time synchronization
- Distributed snapshot
- Reliable multicast
- Replica management
- Consensus

Implementation

- Real network
  - Cellular network, 3G, LTE, 5G
- Simulation
  - Programming languages, e.g., Python, Java, Spark, Erlang, and so on
  - Multi-core, multi-thread, shared-memory
- Cloud
  - Virtualization (on the fly and on demand)

Hands-on session

- Execute python routines to communicate two processes
  - https://gist.github.com/huberflores/6a5ecee3ef4920d16b4c0cb1c737bb6f

MODELLING

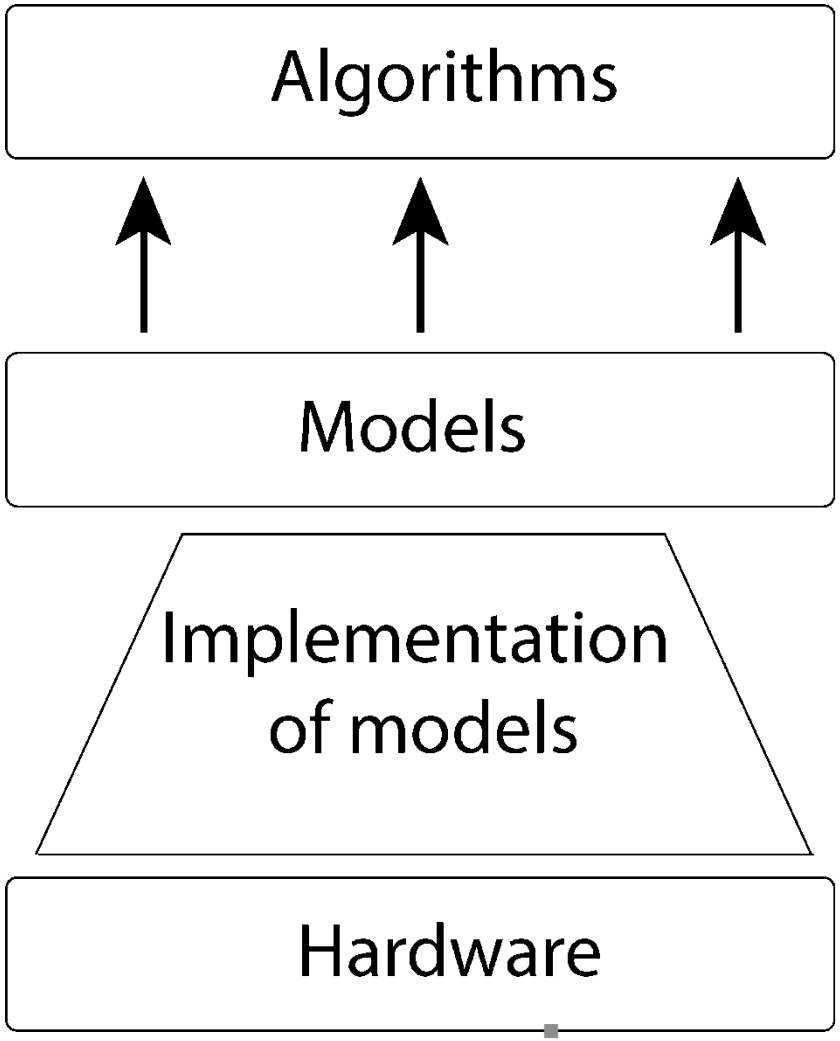
Models

- We will reason about distributed systems by relying on **models.** There are many dimensions of variability in distributed systems. Examples

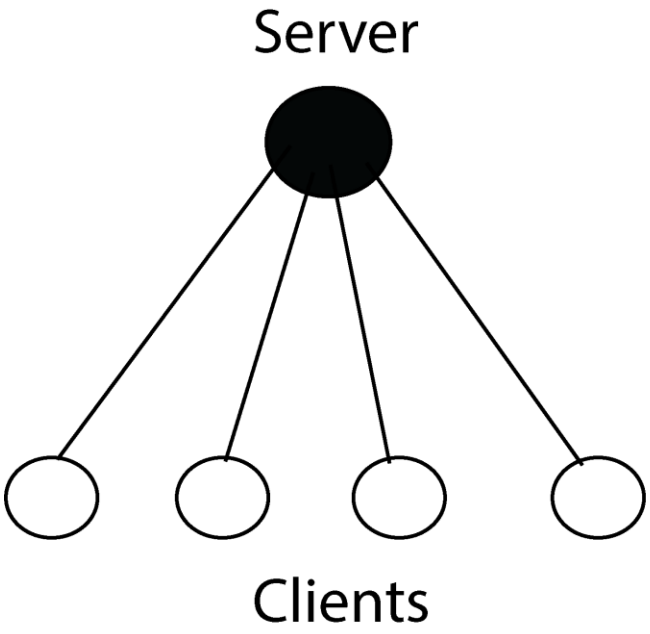
  - Type of processors
  - inter-process communication mechanisms
  - Timing assumptions
  - Failure classes
  - Security features, etc.

Models

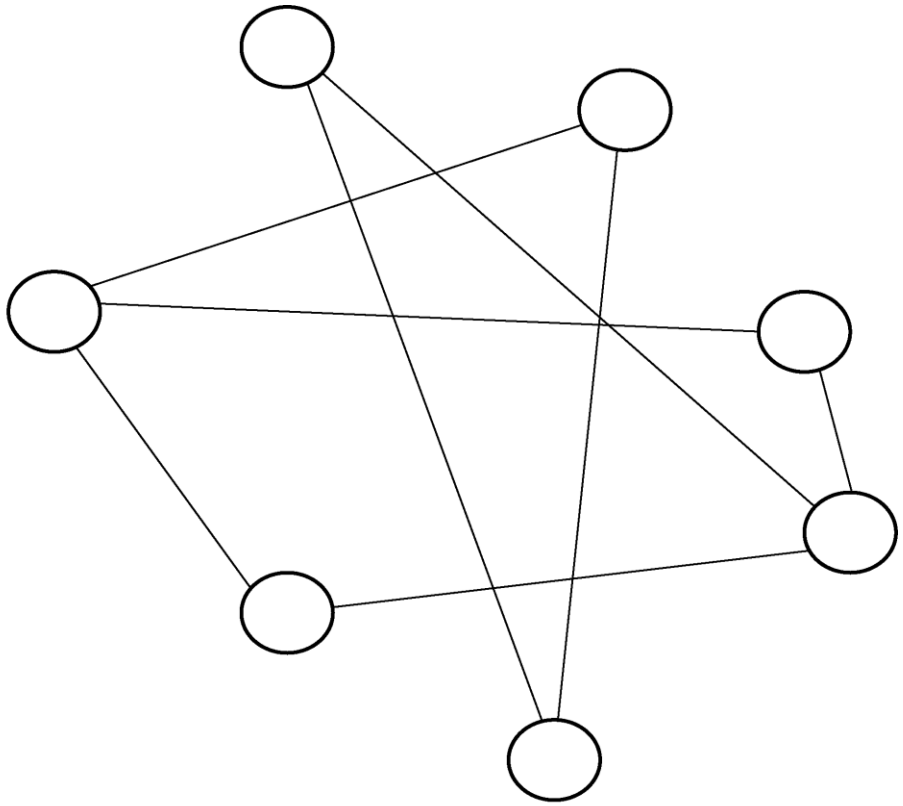
- Models are **simple abstractions** that help to overcome the variability -- abstractions that **preserve the essential features**, but **hide the implementation details** and simplify writing distributed algorithms for problem solving



A classification



Client-server model

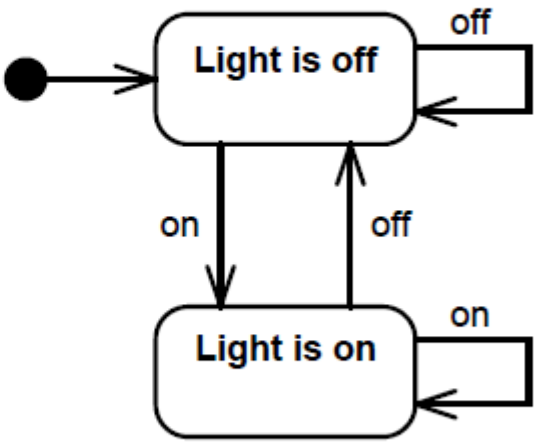


Peer-to-Peer (P2P) model

Models

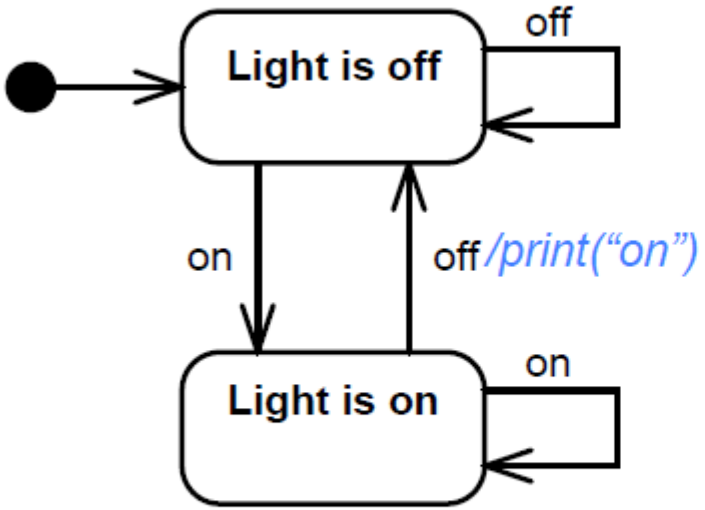
- Components of a system are processes
  - Finite state machine
    - Each process is depicted by a state

State machine (Automaton)

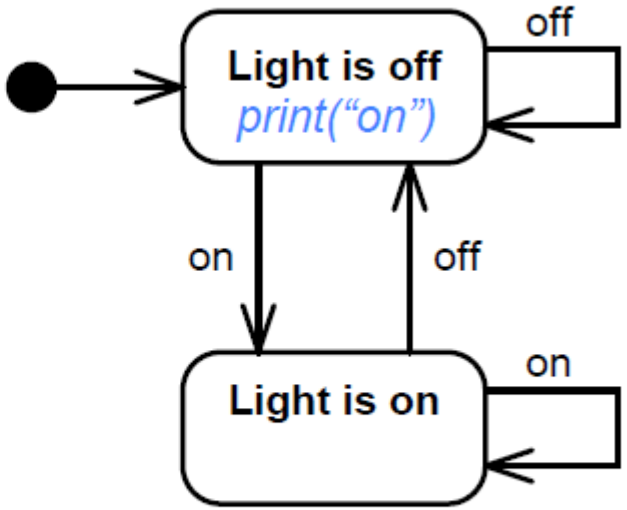


Actions on the automaton

- As the automaton changes state it can generate actions



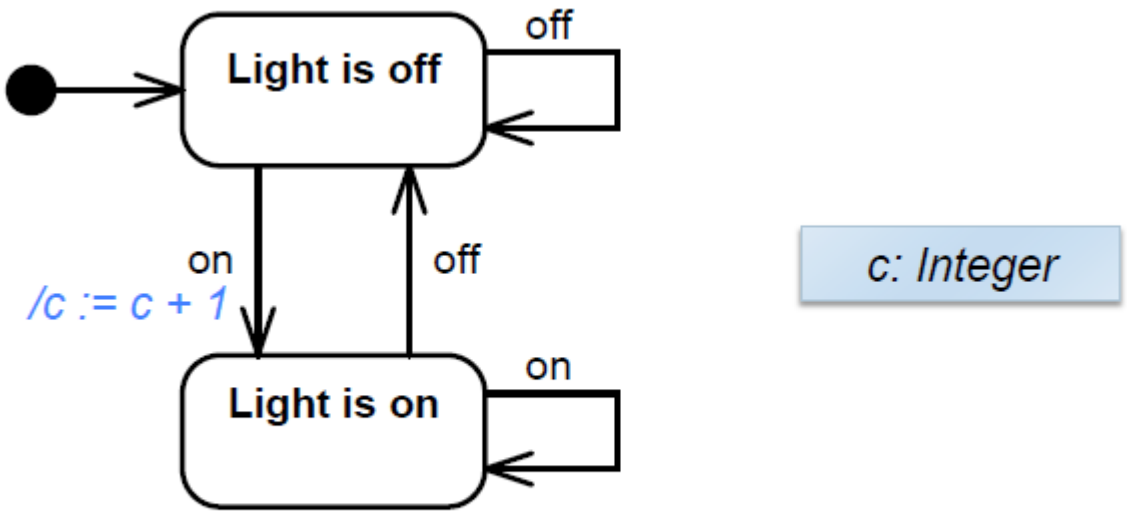
Mealy automaton



Moore automaton

Extended state machines

- Automaton can be extended with variables



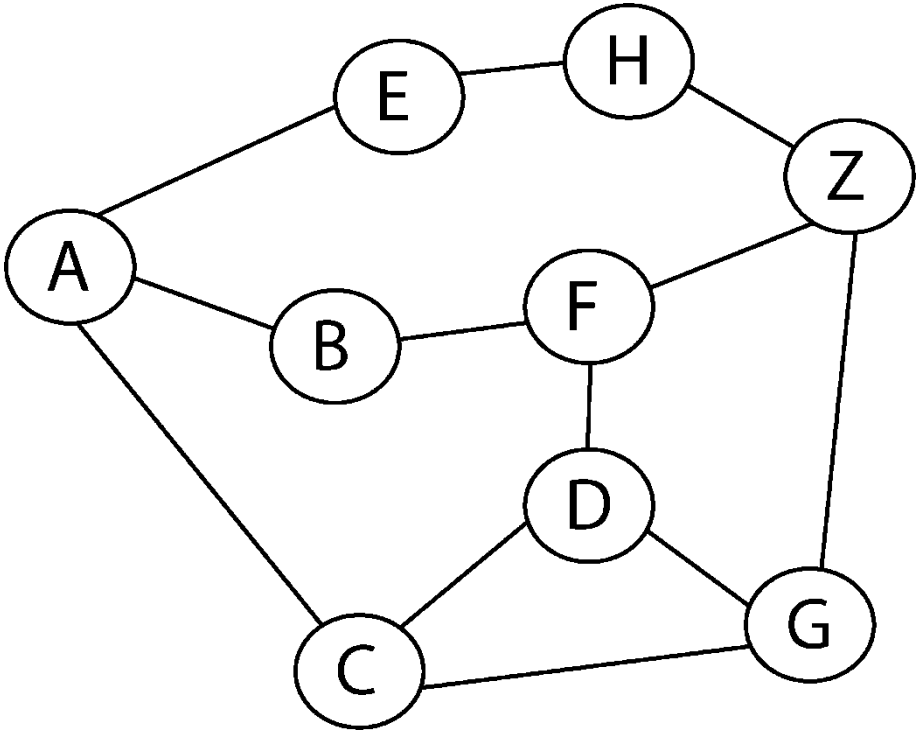
Extended state machines

- States
- Events
  - Call event
  - Signal event
  - Change event
  - Time event
- Transitions and guard conditions
- Transitions effects

Modelling communication

- System topology is a **graph G = (V, E),** where V = set of nodes (sequential processes) E = set of edges (links or channels, bi/unidirectional).

- Four types of actions by a process:
  - **internal action**
  - **input action**
  - **communication action**
  - **output action**



QUESTIONS

Your tasks

- Install Python in your computer
- Register to the course in webOODI, if you have not yet done it