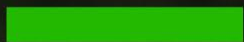




# LAND OF THE CURIOUS



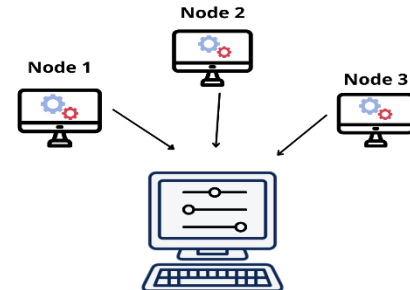
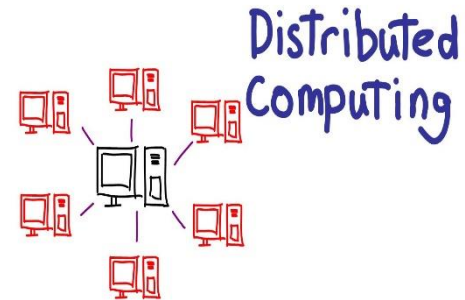
# CT30A3401

## Distributed Systems

### Lecture 1

Bilal Naqvi, PhD.

[syed.naqvi@lut.fi](mailto:syed.naqvi@lut.fi)



# Definitions

- What is a **system**?
  - early 17<sup>th</sup> century
  - origin from French 'système' or late Latin 'systema', from Greek 'sustēma'
  - **Definition:**
    - "A complex whole; a set of connected parts; an organized assembly of resources and procedures (collection of ...) united and regulated by interaction or interdependence to accomplish a set of specific functions"
- Can be classified in different categories
  - Distributed systems are one

# Distributed Systems: Definitions

- A collection of **independent computers** that appears to its users as a single coherent system. [Tanenbaum]
- A system in which hardware and software components of networked computers **communicate and coordinate** their activity **only by passing messages**. [Coulouris]

# Distributed Systems: Definitions

- A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable. [Leslie Lamport, Microsoft Research]

# Distributed Systems: Definitions

- A computing platform built with many computers that:
  - Operate **concurrently**;
  - Are **physically distributed**;
  - Are **linked by a network**; and
  - Have **independent clocks**

# Distributed system as per student responses

## What is a distributed system?



Many computers working together but presented as one

System that is broken into smaller parts

Connected computers that appears as one to outsiders

Processes are done by many different devices/components and compiled into a single result

modular system that allows multiple parts to work for one objective

A programme that is distributed to many servers

Pieces of system, subsystems, that are separated but working as one

physically separated systems that work as one

multiple agents are connected/communicate with each other

# Characteristics

- No global clock
  - Nodes are not synchronized
- No shared memory
  - Data should be synchronized using messages
- Geographical distribution
  - Nodes may be situated in different countries
- Heterogeneity
  - Nodes can have different performance, software and operating system



# Why do we need Distributed Systems?

- World a global village
  - people in different parts of the world need to work together
  - each person needs a hardware to be physically close to them
- Information is all over the place (distributed) and needs to be shared
- For sharing the hardware (provides more resources by doing work in parallel; more efficient resource utilisation)

# Examples

- Intranets, Internet, www, email...
- Net banking
- Airline reservation systems
- Taxi Services (for example UBER)
- Others?

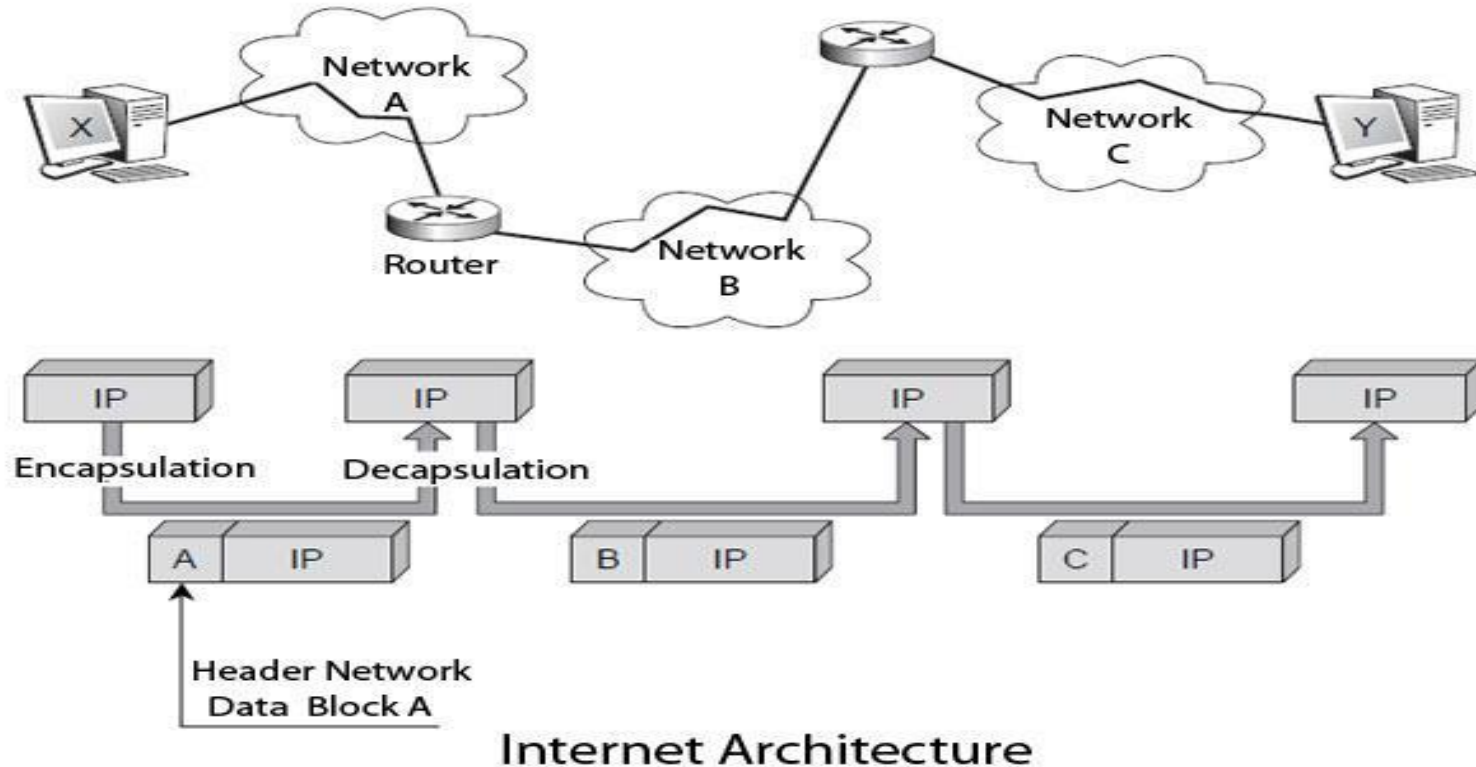
# Examples (by students of the course)

Examples for distributed systems you know of  
and/or have used?

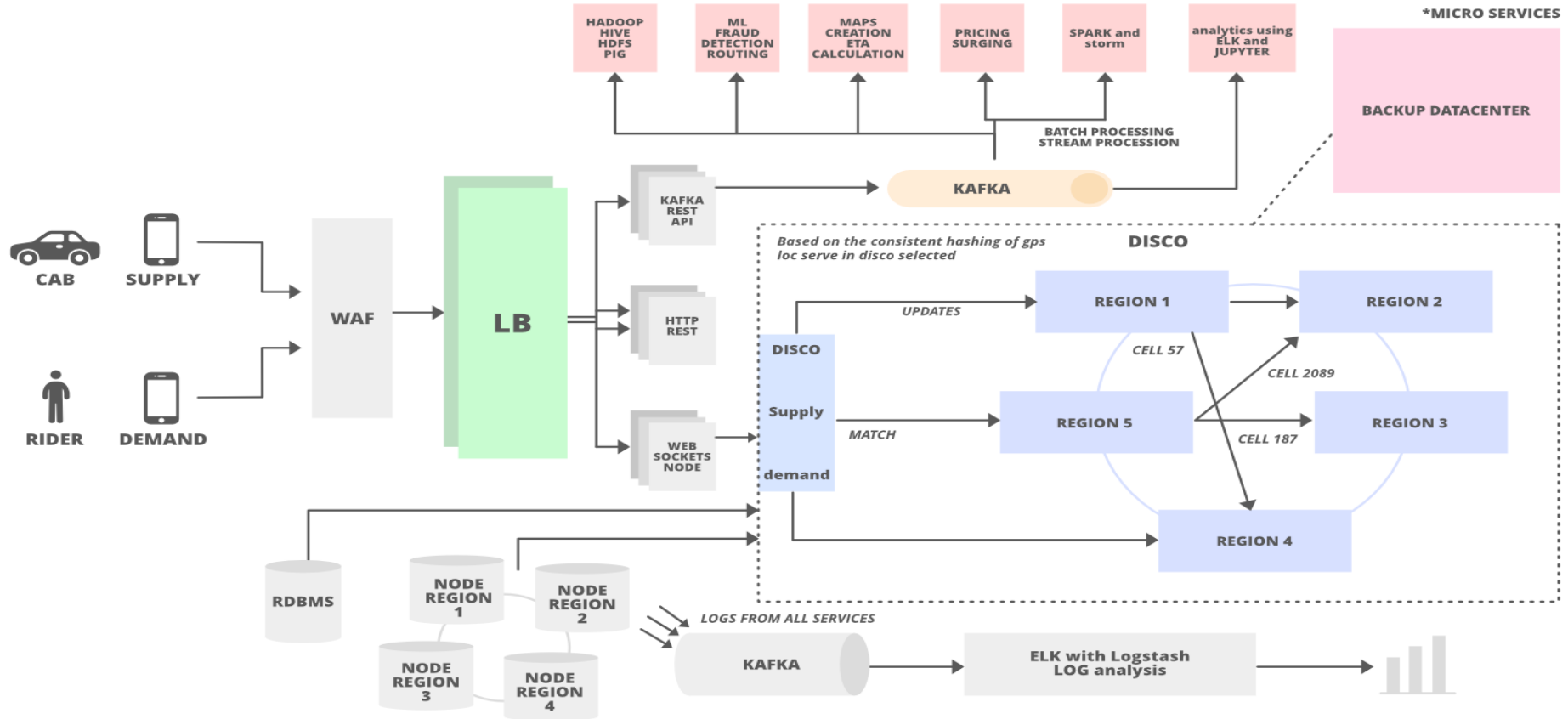
 Mentimeter



# Examples: Internet



# Examples: UBER



# Implications

- Concurrent execution of processes
  - synchronization, deadlocks
- No global clock
  - coordination is done by message exchange
  - no single global notion of the correct time
- No global state
  - no process has a knowledge of the current global state of the system
- Units may fail independently
  - network faults may isolate computers that are still running
  - system failures may not be immediately known

# Basic Terms and Concepts

- **Resource** is an any hardware or software entity, that is available in the distributed network
- **Node** is an any device in the distributed network
- **Server** is a supplier of the information
- **Client** is a consumer of the information
- **Peer** is a server and a client at the same time
- **Service** is a network entity, that provides access to some resources

# Design Challenges

- **Heterogeneity**
  - Heterogeneous components must be able to interoperate
- **Openness**
  - Interfaces should allow components to be added or replaced
- **Security**
  - The system should only be used in the way intended



# Design Challenges

- **Scalability**
  - System should handle increasing number of clients
- **Failure handling**
  - Failure in one components should not result the whole system failure
- **Transparency**
  - Distributed nature should be hidden from the clients