CSE 601: Distributed Systems

Toukir Ahammed

Introduction to Distributed System

What is a distributed system?

> Hardware or software **components** located at **networked** computers that communicate or **coordinate** their actions only by **passing messages**.
>
> *-Coulouris, Dollimore, Kindberg, Blair*

What is a distributed system?

A collection of **autonomous computing elements**, connected by a **network**, which appear to its users as a **single coherent system**.

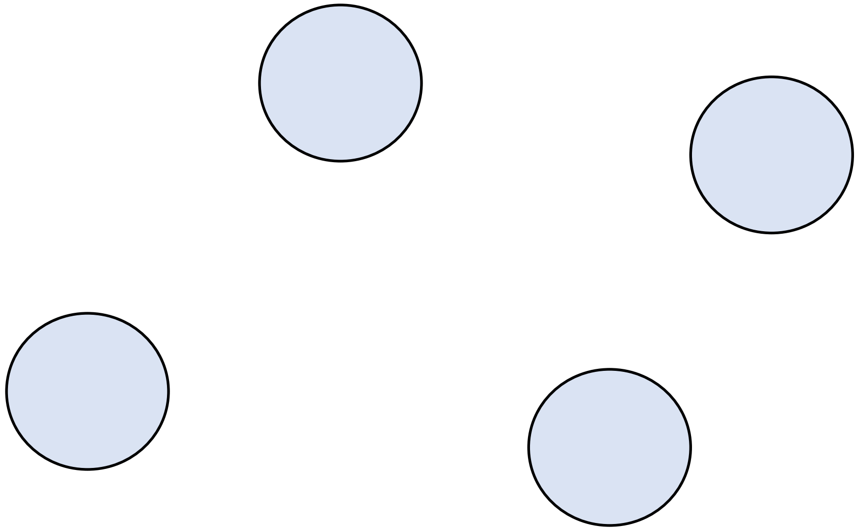
-Steen and Tanenbaum

What is a distributed system?

A system in which **components** located on **networked** computers communicate and **coordinate** their actions by **passing messages**. The components interact with each other in order to achieve a **common goal**.

- Wikipedia

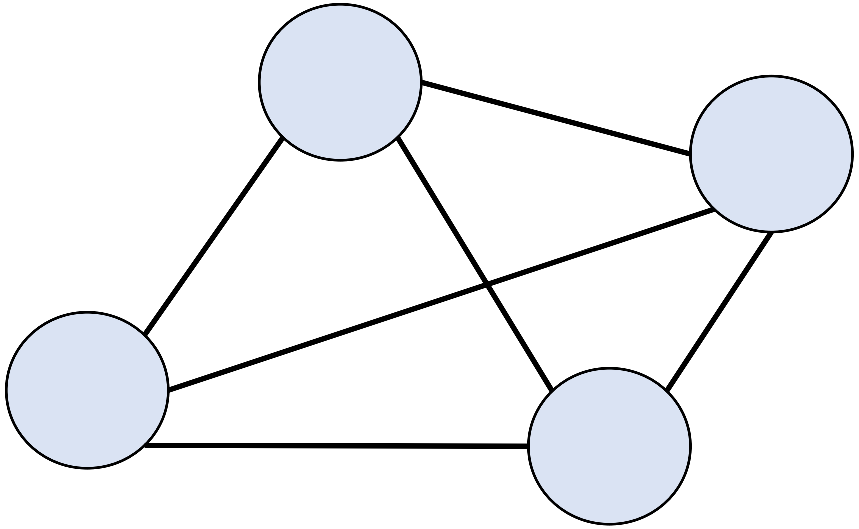
What is a distributed system?



Independent components or elements

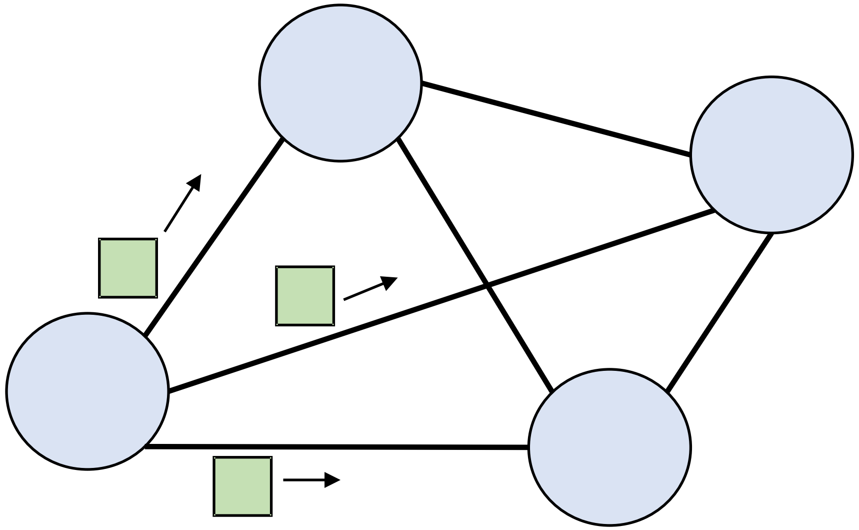
(software processes or any piece of hardware used to run a process, store data, etc.)

What is a distributed system?



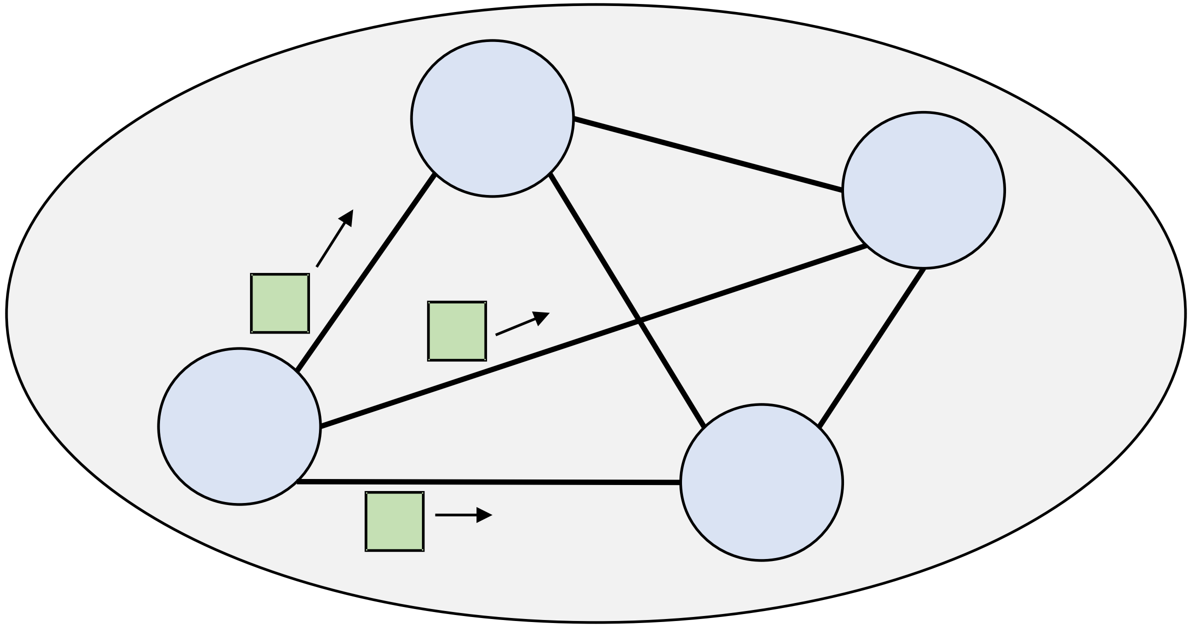
Independent components or elements that are **connected by a network**.

What is a distributed system?



Independent components or elements that are **connected by a network** and **communicate by passing message.**

What is a distributed system?



Independent components or elements that are **connected by a network** and **communicate by passing message** to achieve **common goal**, appearing as a single coherent system.

What is a distributed system?

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*

What is a distributed system?

More definitions:

“A collection of independent computers that appears to its users as a single coherent system.”

“A computing environment in which various components are spread across multiple nodes (*computer, phone, car, robot or other computing devices*) on a network trying to achieve some task together.”

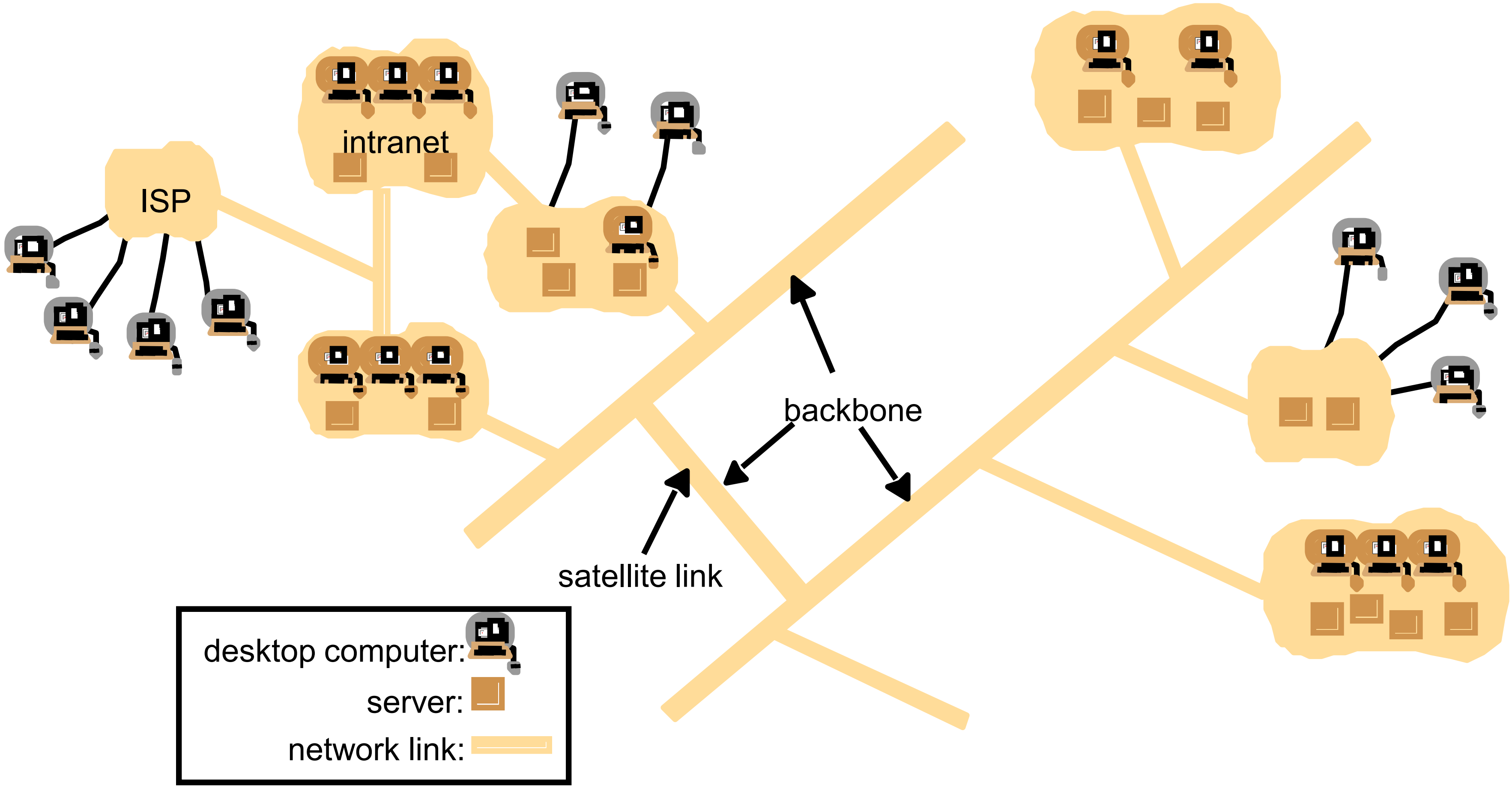
Examples of distributed systems

- Machine learning (for compute)
- P2P file sharing (high availability, share large files, piracy)
- Google search engine (for storage and bandwidth)
- Facebook (for storage and bandwidth)
- Black hole image (distributed observation)
- IOT (Sensors on a network)
- Blockchain (decentralized record of transactions) etc.

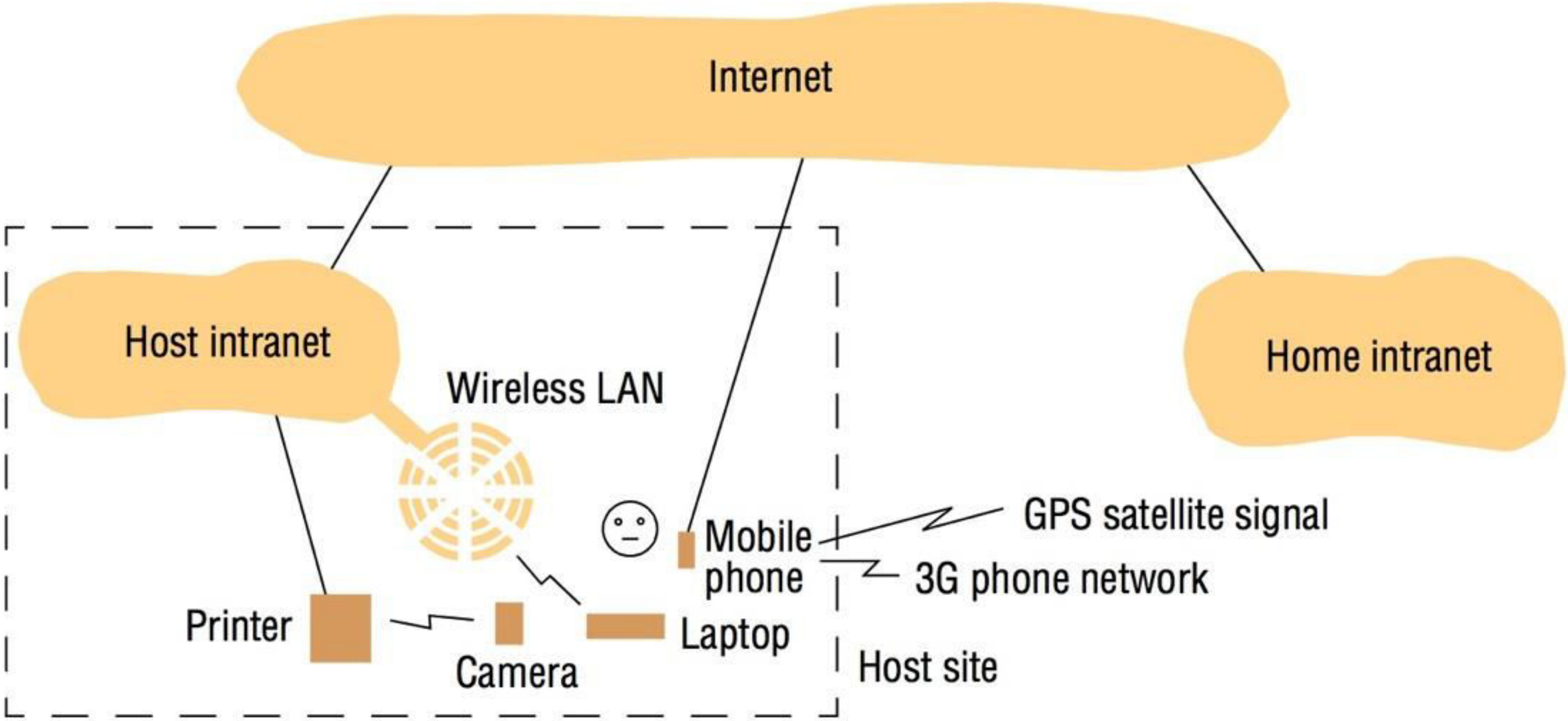
Examples of distributed systems

- World Wide Web
- A cluster of nodes on the cloud (AWS, Azure, GCP)
- Multi-player games
- BitTorrent
- Online banking
- …………

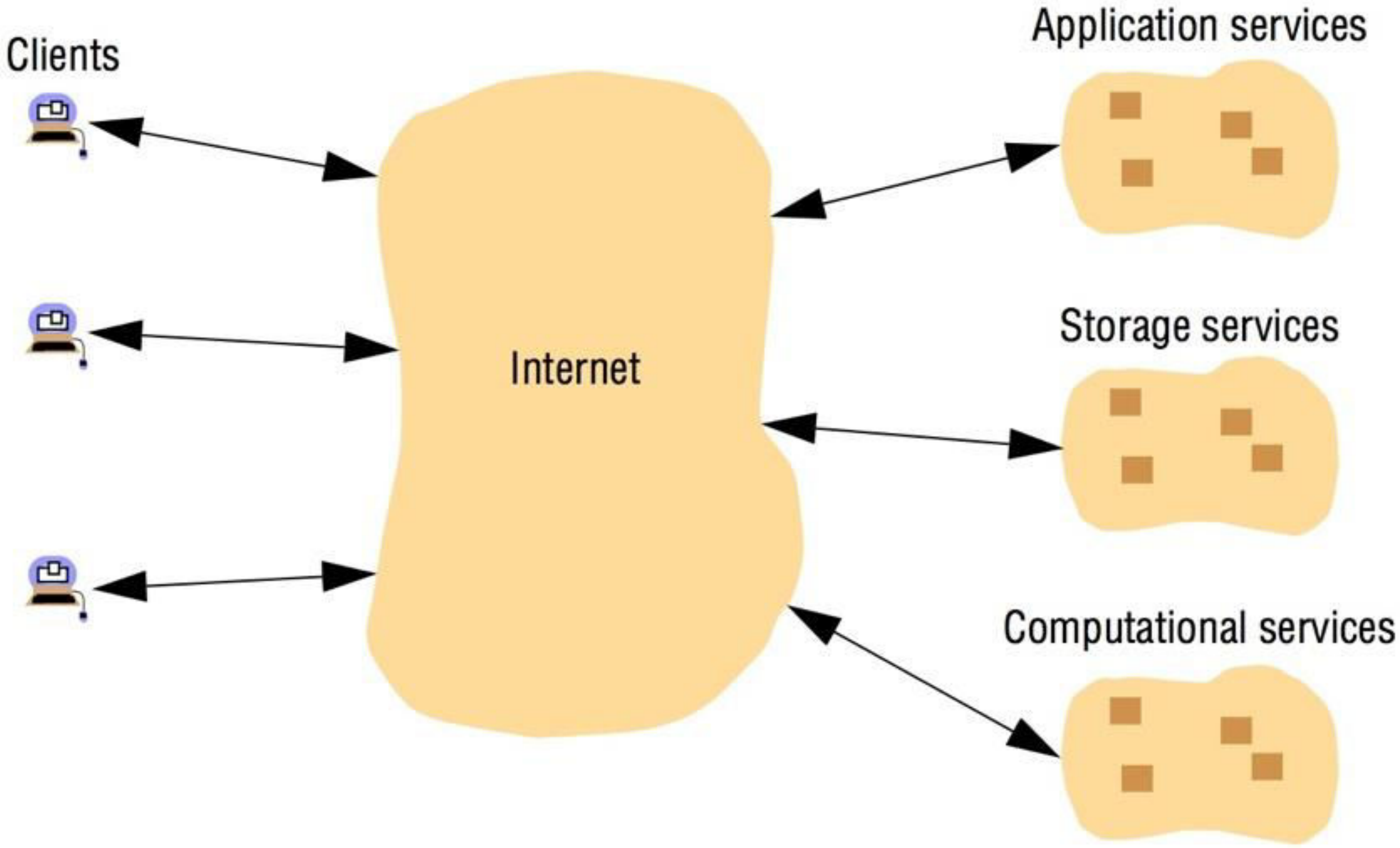
A typical portion of the Internet



Portable and handheld devices in a distributed system



Cloud computing



Why distributed systems?

- Nature of the application
  - Multiplayer games, P2P file sharing, client requesting a service.
- Availability despite unreliable components
  - A service shouldn’t fail when one computer does.
- Conquer geographic separation
  - A web request in India is faster served by a server in India than by a server in US.
- Scale up capacity
  - More CPU cycles, more memory, more storage, etc.
- Customize computers for specific tasks
  - E.g. for storage, email, backup.

Why make a system distributed?

- It’s inherently distributed:

  e.g. sending a message from your mobile phone to your friend’s phone

- For better reliability:

  even if one node fails, the system as a whole keeps functioning

- For better performance:
  - get data from a nearby node rather than one halfway round the world
- To solve bigger problems:
  - e.g. huge amounts of data, can’t fit on one machine

Why NOT make a system distributed?

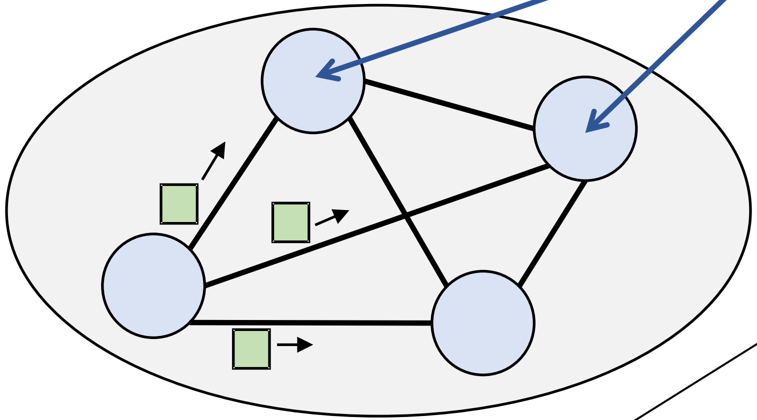
The trouble with distributed systems:

- Communication may fail (and we might not even know it has failed).
- Processes may crash (and we might not know).
- All of this may happen non-deterministically.

Fault tolerance: we want the system as a whole to continue working, even when some parts are faulty.

- This is hard.
- Writing a program to run on a single computer is comparatively easy?!

Challenging properties

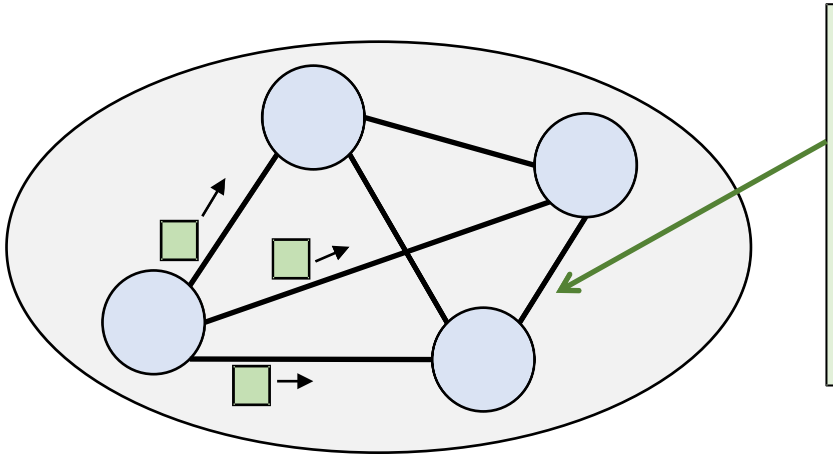


Multiple computers

- Concurrent execution.
- Independent failure.
- Autonomous administration.
- Heterogeneous.
- Large numbers.

Heterogeneity (variety and difference):
the differences that arise in networks, programming languages, hardware, operating systems and differences in software implementation.

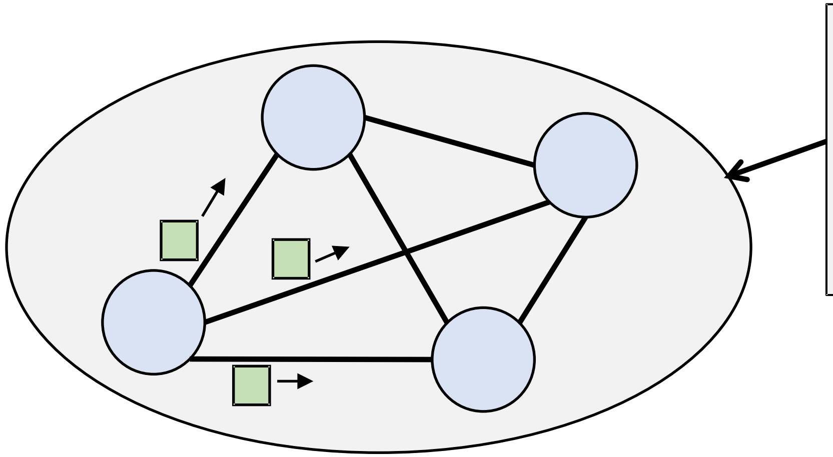
Challenging properties



Networked communication

- Asynchronous
- Unreliable
- Insecure

Challenging properties



Common goal

- Consistency
- Transparency

system that is able to present itself to users and applications as if it were only a single computer system is said to be transparent.

Advantages of Distributed Systems

- **Scalability**:
  - Distributed systems are made on default to be scalable.
  - Whenever there is an increase in workload, users can add more workstations.
  - There is no need to upgrade a single system.
  - Moreover, no any restrictions are placed on the number of machines.

Advantages of Distributed Systems

- **Reliability**:
  - Distributed systems are far more reliable than single systems in terms of failures.
  - Even in the case of a single node malfunctioning, it does not pose problems to the remaining servers. Other nodes can continue to function fine.

Advantages of Distributed Systems

- **Low Latency**:
  - Since users can have a node in multiple geographical locations, distributed systems allow the traffic to hit a node that's closest, resulting in low latency and better performance.

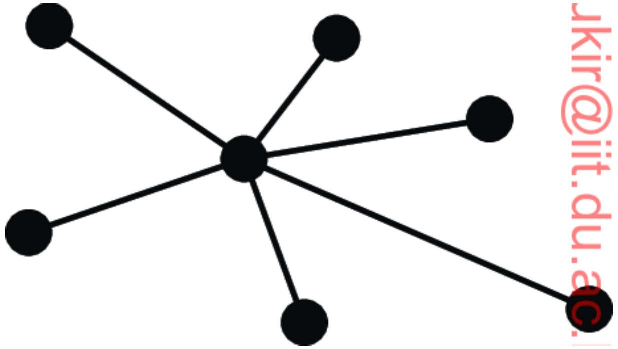
Advantages of Distributed Systems

- **Efficiency**:
  - Distributed systems allow breaking complex problems/data into smaller pieces and have multiple computers work on them in parallel, which can help cut down on the time needed to solve/compute those problems.

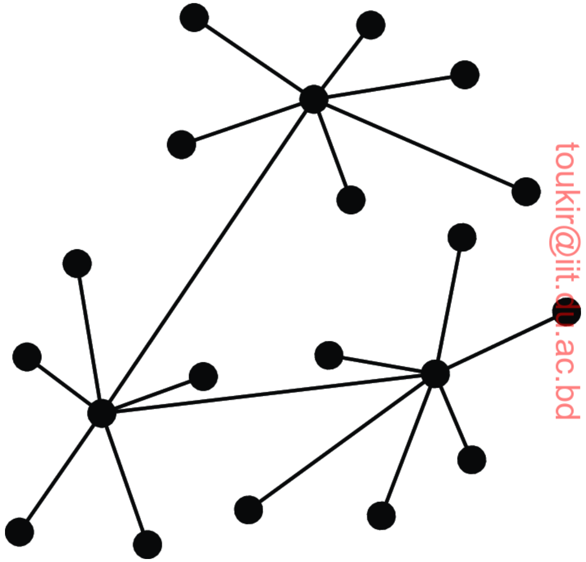
Pitfalls when Developing Distributed Systems

- False assumptions made by first time developer:
- 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 one administrator.

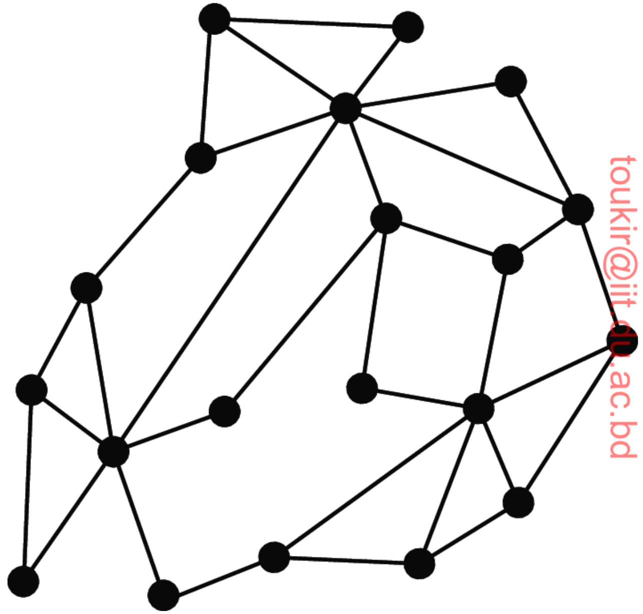
Distributed versus decentralized systems



centralized



decentralized



distributed

Distributed versus decentralized systems

- **Decentralized** is still distributed in the technical sense, but the whole decentralized systems is not owned by one actor. No one company can own a decentralized system, otherwise it wouldn’t be decentralized anymore.

- This means that most systems we will go over today can be thought of as **distributed centralized systems** — and that is what they’re made to be.