INTRODUCTION

DISTRIBUTED SYSTEMS

Master of Science in Cyber Security

A.Y. 2024/2025

SAPIENZA
UNIVERSITÀ DI ROMA

CIS SAPIENZA
CYBER INTELLIGENCE AND INFORMATION SECURITY

Welcome

Welcome

Sixth Edition of the Course

Many thanks to the students of the previous editions
2019, 2020, 2021, 2022, 2023

INTRODUCTION OUTLINE

- General Information
- Introduction

GENERAL INFORMATION

GENERAL INFORMATION

- 6 CFUs
- First semester
  - 23 September – December
- Professor: Giuseppe Antonio Di Luna
  - diluna@diag.uniroma1.it
  - Office hours meet: In presence (Via Ariosto 25, Office B215) - book it with an email.
  - Distributed Systems Interests:
    - Algorithms and Real Systems
    - One Line CV: Phd Sapienza 2011-2015, University of Ottawa 2015-2017, Aix-Marseille and CNRS 2017-2018, Sapienza 2019-Now.

GENERAL INFORMATION

- Class Schedule
  - **Monday 14:00-17:00 Aula Alfa Via Salaria 113/117**
  - **Wednesday 11:00-13:00 [Aula A - Pietro Benedetti] Building: RM027**

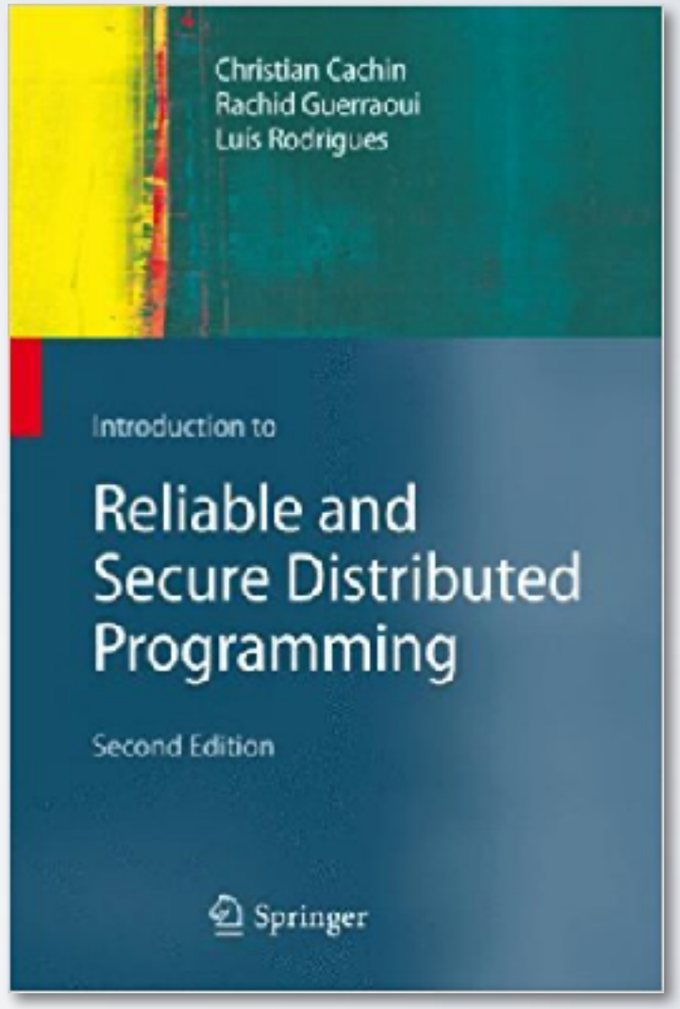
?

GENERAL INFORMATION

- Check the course website periodically!
- Web site: https://sites.google.com/view/distributed-systems-2024/

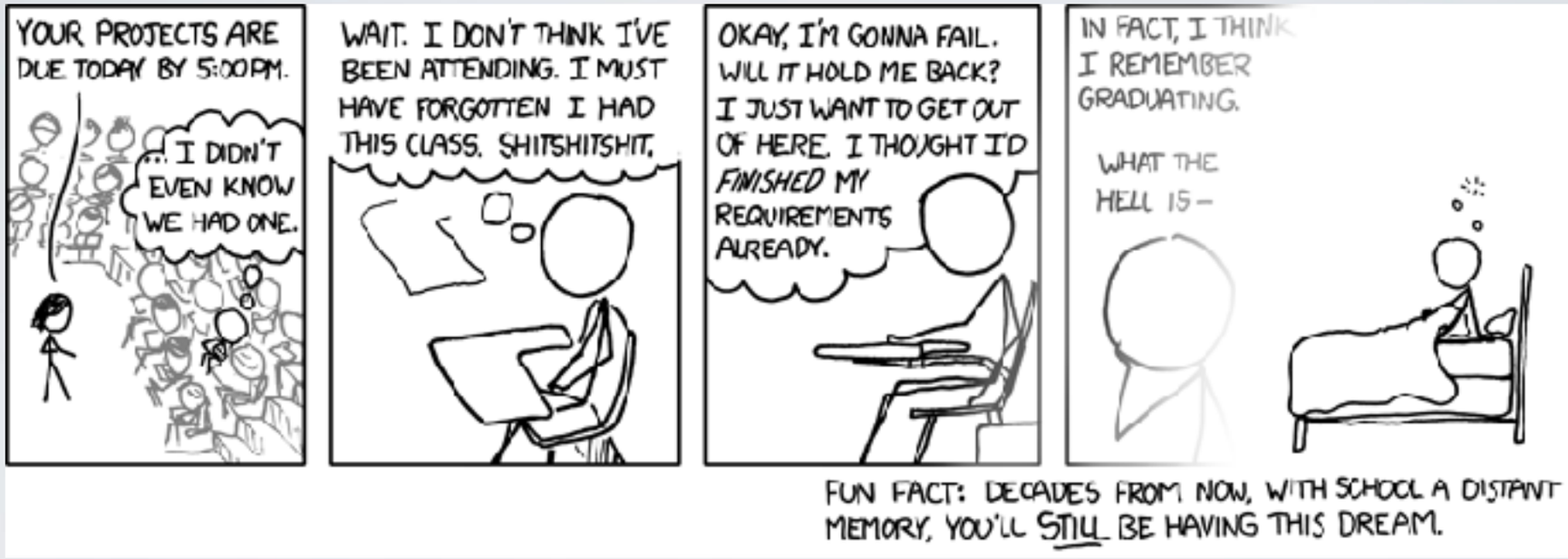
GENERAL INFORMATION

- Material
  - Main text book: C. Cachin, R. Guerraoui and L. Rodrigues. Introduction to Reliable and Secure Distributed Programming, Springer, 2011
  - Scientific papers (*)
  - Slides (*)
  - Exam examples (*)



(*) available on course website.

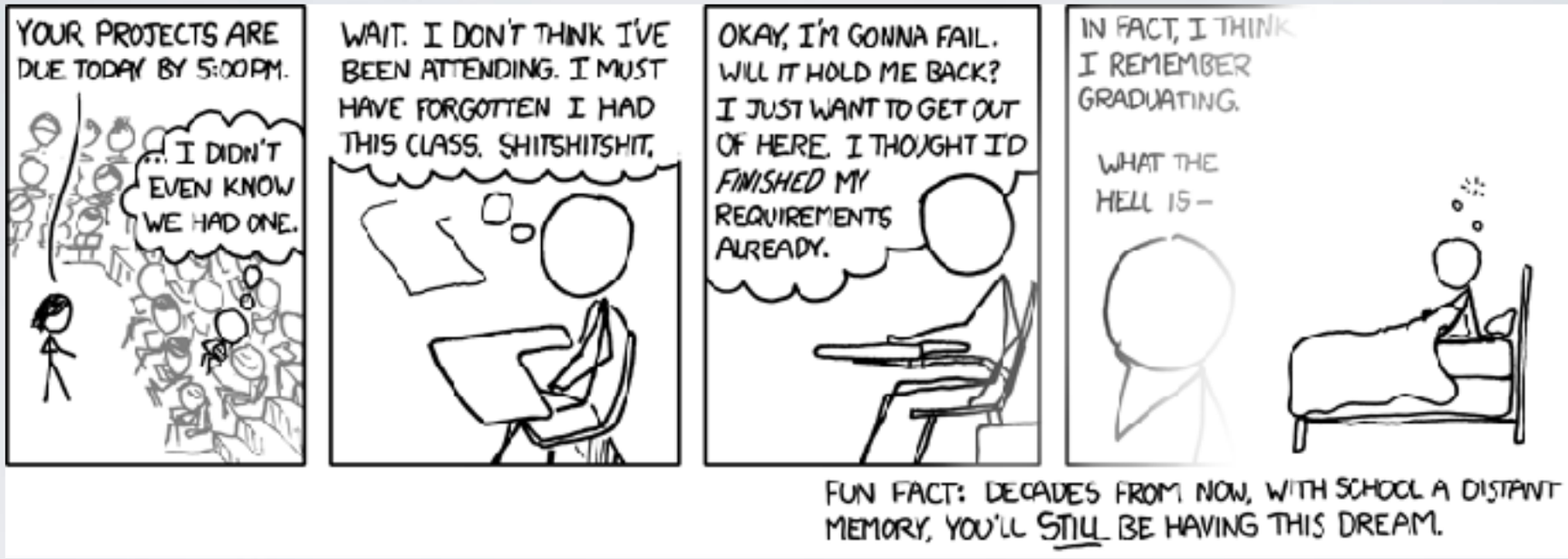
GENERAL INFORMATION



- Examination
  - Written test
    - Theory questions
    - Exercises
    - Design of distributed algorithms
  - Oral test mandatory for laude:
    - Discussion of the written test outcomes
    - Open questions

Closed Books Exam!

GENERAL INFORMATION



- Examination - Grading - My internal, and personal, definition of high grades:
  - 26,27 - Good knowledge of the course content.
  - 28,29 - Mastery of the course content.
  - 30 - Complete Mastery of the course content. Your knowledge of the course material is the same as mine.
  - 30 e Lode (Laude). **Extraordinary** - You deserve to be acknowledged for your merits- You know more than me.

GENERAL INFORMATION

What you will lean?

- Model a **distributed system**
- Dissect a problem in basic building blocks
- Specify distributed **abstractions**
- Design distributed **fault-tolerant protocols**

GENERAL INFORMATION

What you will lean?

- Coordinating a set of **computational entities** to solve a **common task** despite **failures** and **unpredictable communication** is **hard!**

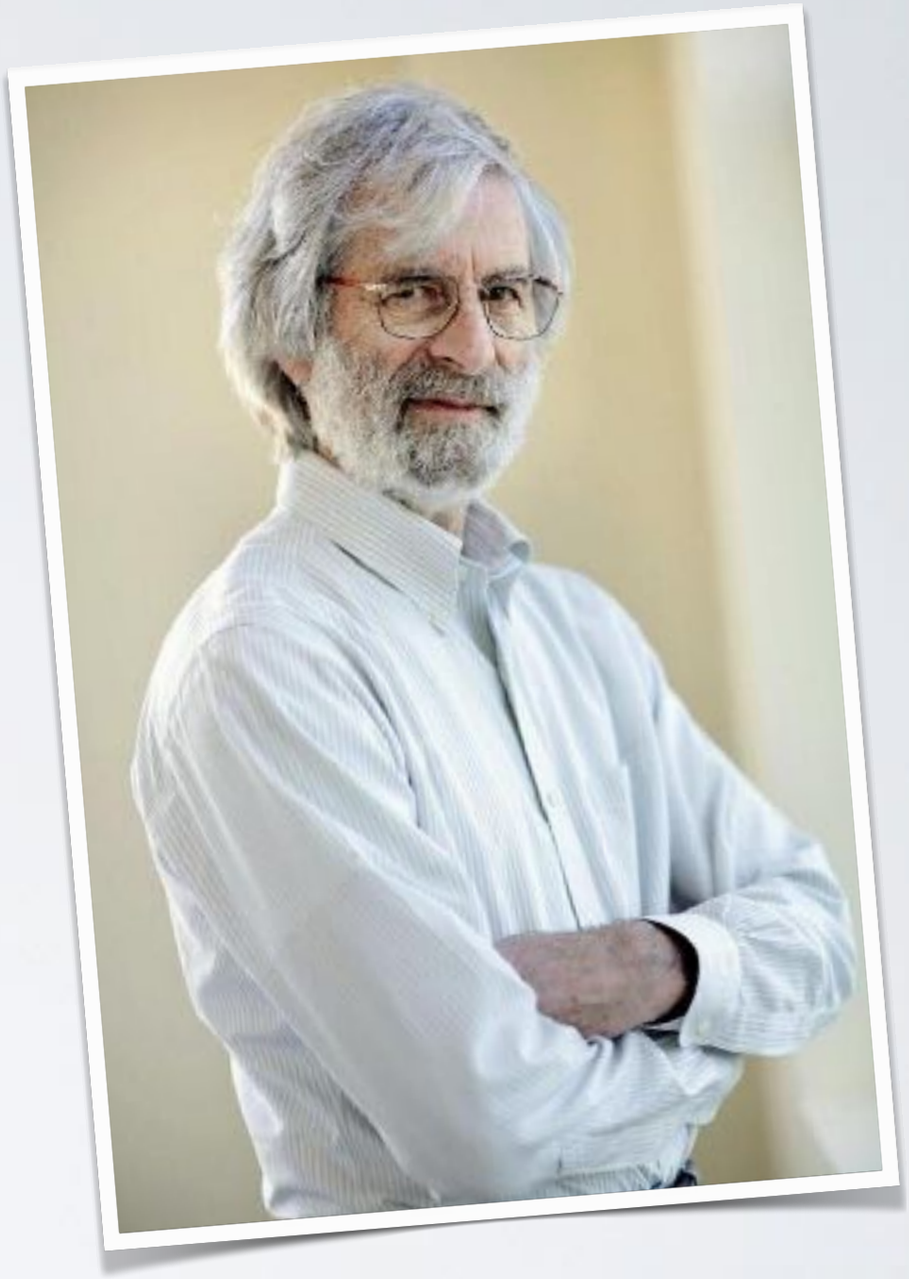
DEFINITION

WHAT IS A DISTRIBUTED SYSTEM?

- A distributed system is a set of **spatially separate entities**, each of these with a **certain computational power** that are able to **communicate** and to **coordinate** among themselves for reaching a **common goal**.

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*)

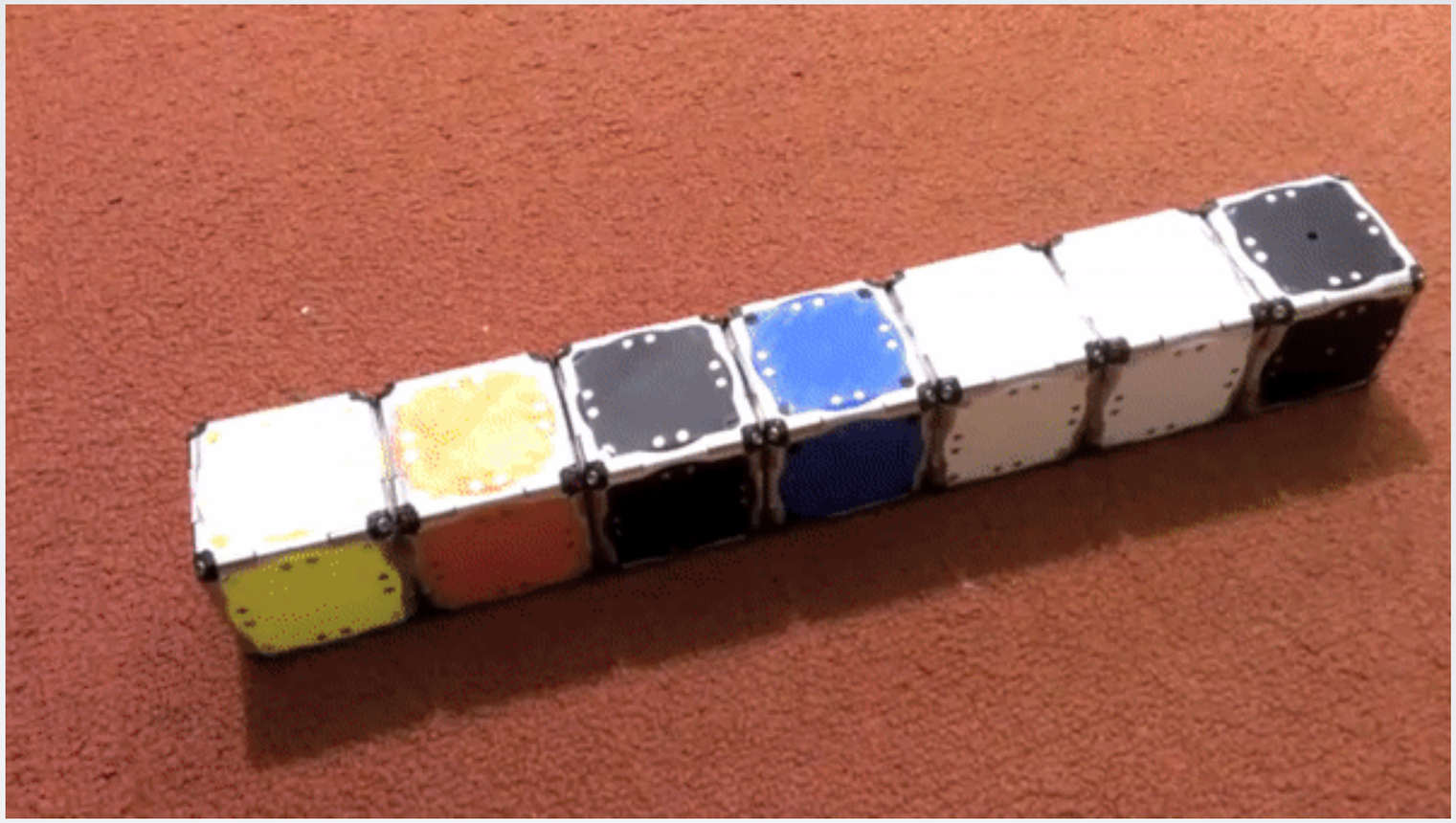


WHAT IS A DISTRIBUTED SYSTEM?

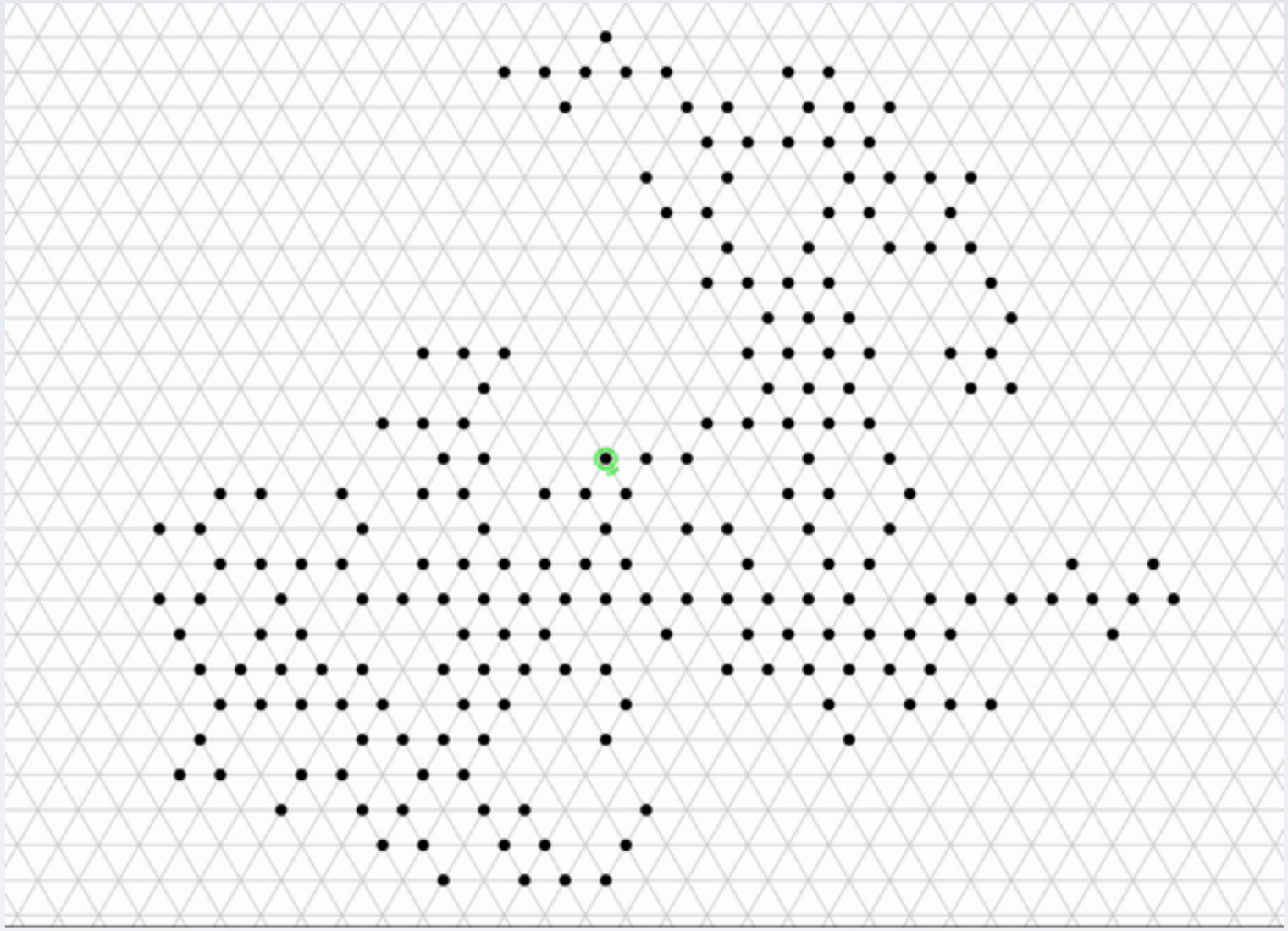
By example:

- The world wide web – information, resource sharing
- Clusters, Network of workstations
- Distributed manufacturing systems (e.g.,automated assembly line)
- Network of branch office computers - Information system to handle automatic processing of orders
- Network of embedded systems (e.g., redundant systems on airplanes)
- P2P networks (BitTorrent)
- Blockchain and Distributed Ledgers

WHAT IS A DISTRIBUTED SYSTEM?



WHAT IS A DISTRIBUTED SYSTEM?



COMMON POINTS ACROSS DEFINITIONS

- Set of entities: computers, machines, software;
- Distributed: spatially separated, independent.
- Common Goal: communication, coordination, resource sharing.

COMMON POINTS ACROSS DEFINITIONS

We will focus on **static networks** of **processes**

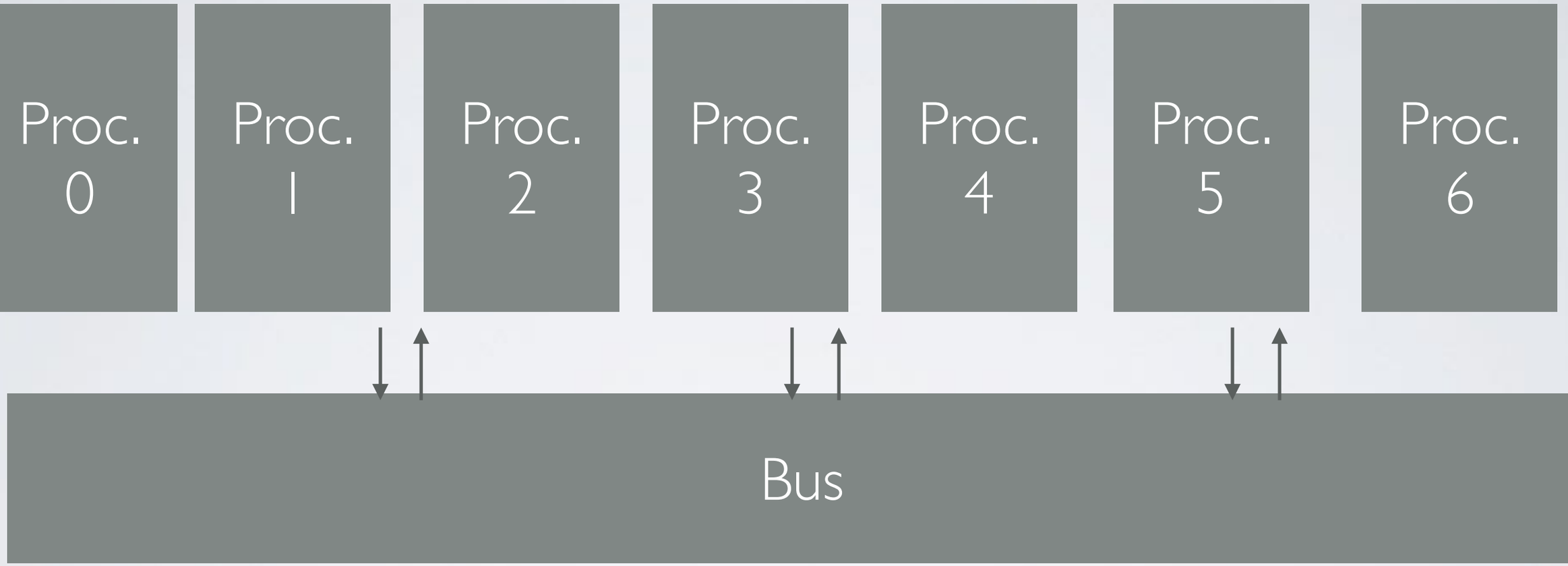
COMMON POINTS ACROSS DEFINITIONS

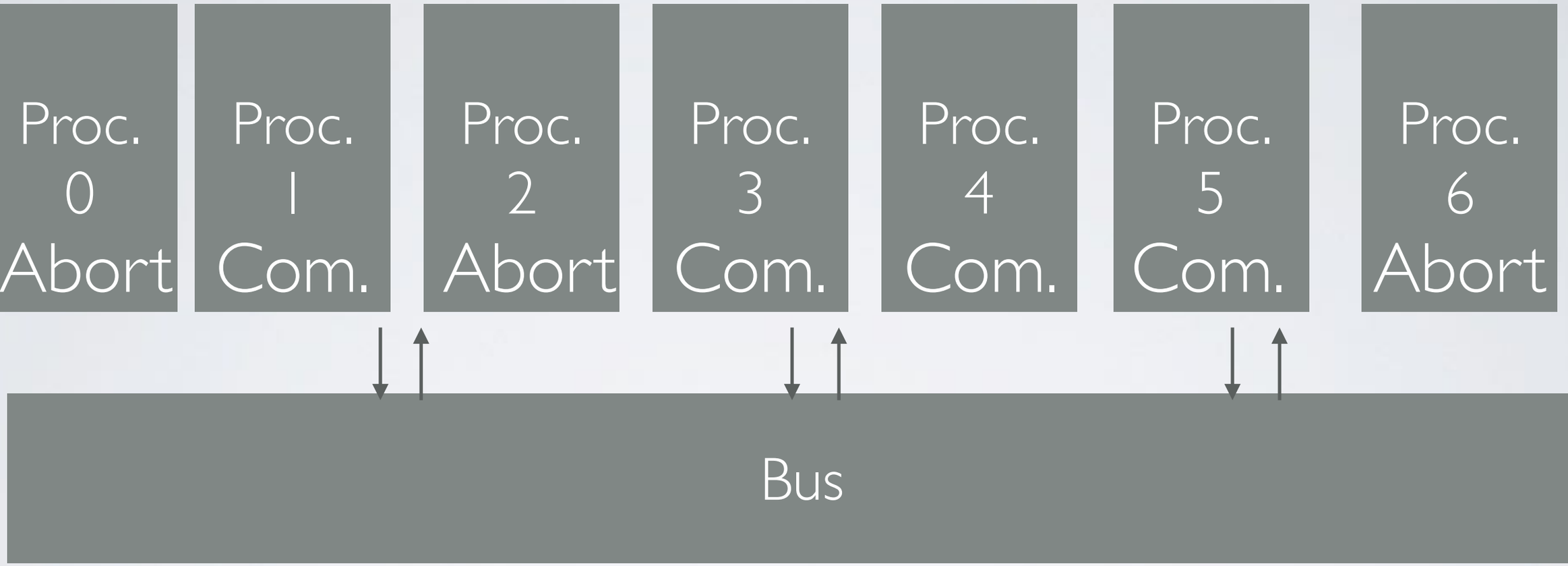
We will focus on **static networks** of **processes**

A PROBLEM

- 1978: NASA SIFT project

- You have several computers performing the same task.
- Goal: **Fault-tolerant agreement**
- Problems:
  - Failures


Proc. 0
Proc. 1
Proc. 2
Proc. 3
Proc. 4
Proc. 5
Proc. 6
Bus


Proc. 0 Abort
Proc. 1 Com.
Proc. 2 Abort
Proc. 3 Com.
Proc. 4 Com.
Proc. 5 Com.
Proc. 6 Abort
Bus

FAILURES

- All **computer systems fail** (sooner or later). It is a good design practice to **build robust systems**.
- **Failure detection**
- **Failure masking**
- **Failure tolerance**
- **Failure Recovery**

FAILURES

- All **computer systems fail** (sooner or later). It is a good design practice to **build robust systems**.
- **Failure detection**
  - Example: Checksum detects a corrupted packet
- **Failure masking**
  - Example: message retransmission
  - Example: redundancy
- **Failure tolerance**
  - Example: intrusion tolerant systems
- **Failure recovery**

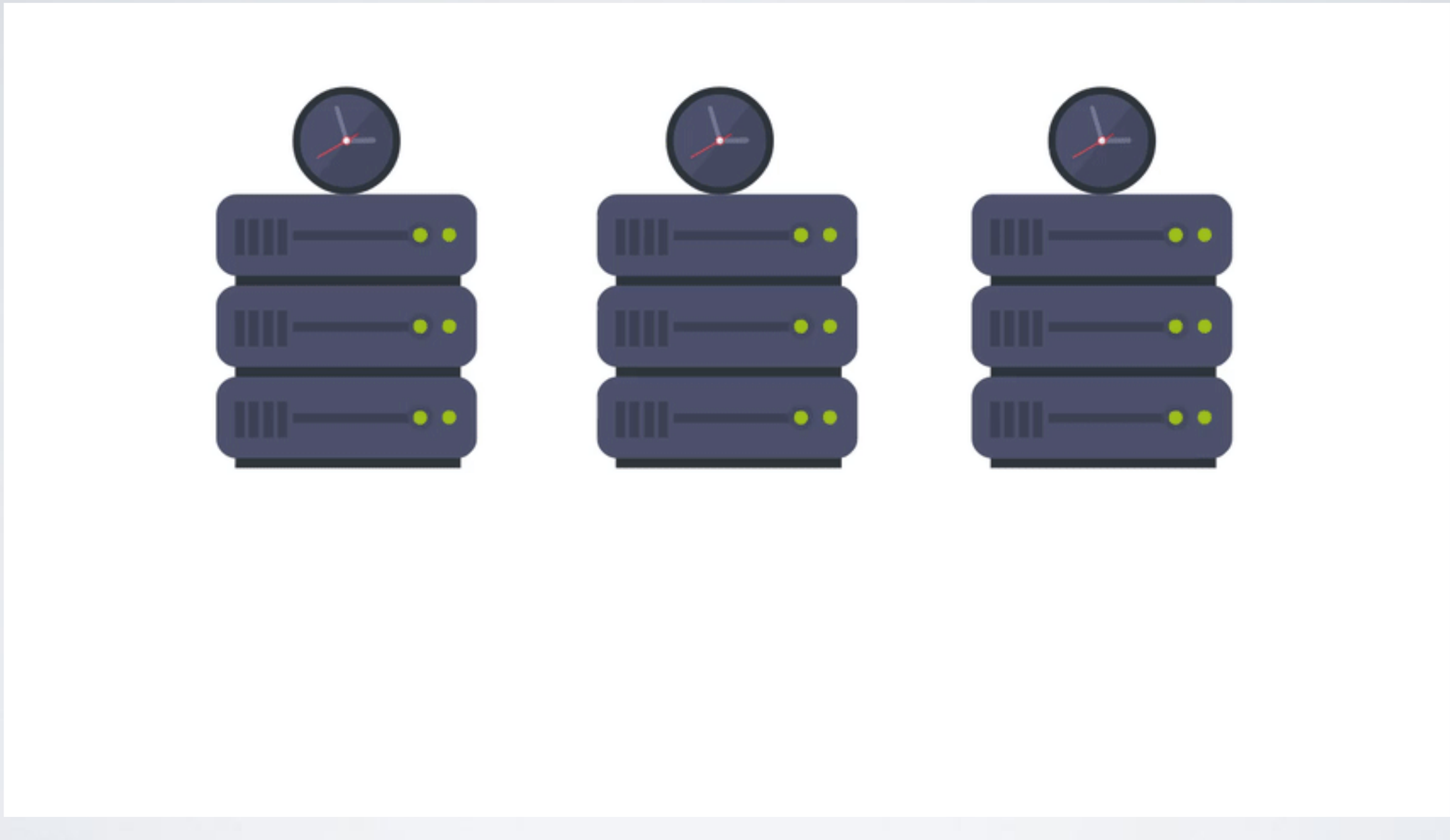
FAILURES ARE UNCERTAIN

A **SIMPLE** AGREEMENT PROBLEM

Bob

Alice

WHEN WE MOVE TO LARGE SCALE



WHEN WE MOVE TO LARGE SCALE

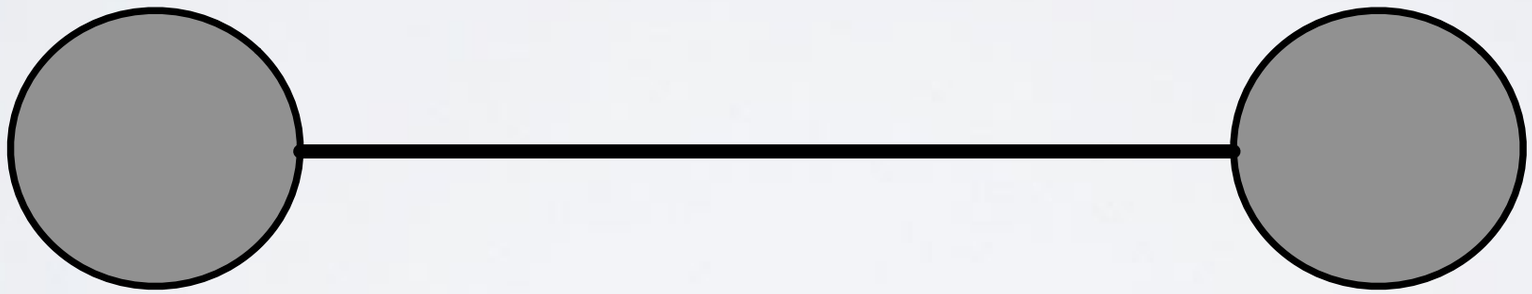
WHEN WE MOVE TO LARGE SCALE

NO GLOBAL CLOCK

- When programs need to cooperate they coordinate their actions by **exchanging messages**.
- Close coordination often depends on a **shared idea of the time** at which events occur.
- There are limits to the accuracy with which computers on a network can **synchronize their clocks**:
  - Network delays are not constant (or even bounded).
  - Clock drift cannot be known in a precise way.

A FIGHT AGAINST UNCERTAINTY

- ASYCHRONY
- FAILURES



Local view

OUTCOMES

- Chubby - google: distributed lock manager. Used in MapReduce, BigTable, Google Filesystem.
- Zookeeper - yahoo (now apache): Hadoop, Kafka.
- Blockchains.

METHODOLOGY

Models, Abstractions, Specifications, Algorithms

OUTLINE

- M1: Models, Abstractions, and Basic Concepts
- M2: Time in Distributed Systems
- M3: Basic Broadcast Primitives
- M4: Shared Memories
- M5: Consensus
- M6: Total Ordering and Replicated State Machine
- M7: BFT
- M8: Blockchains