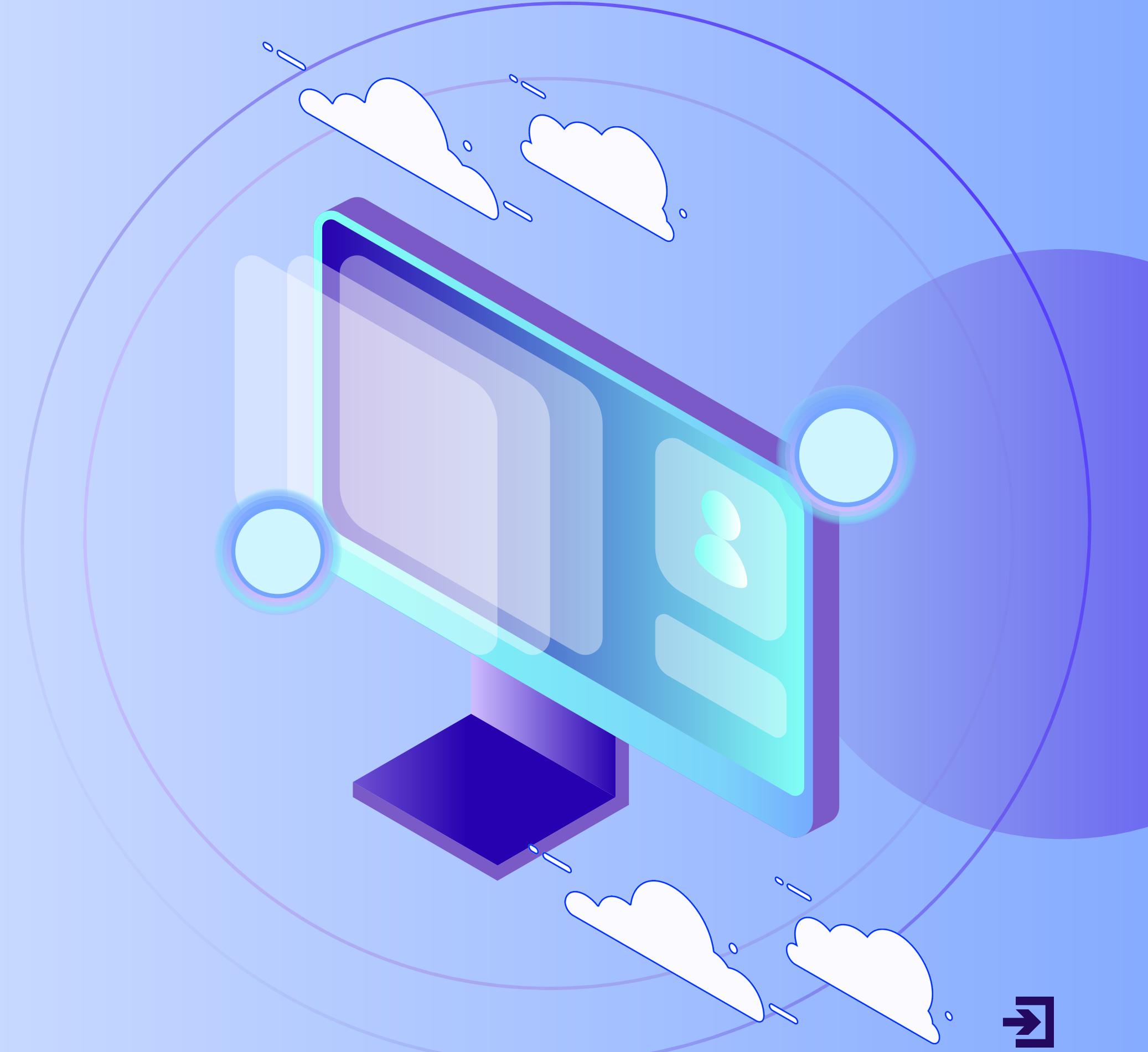




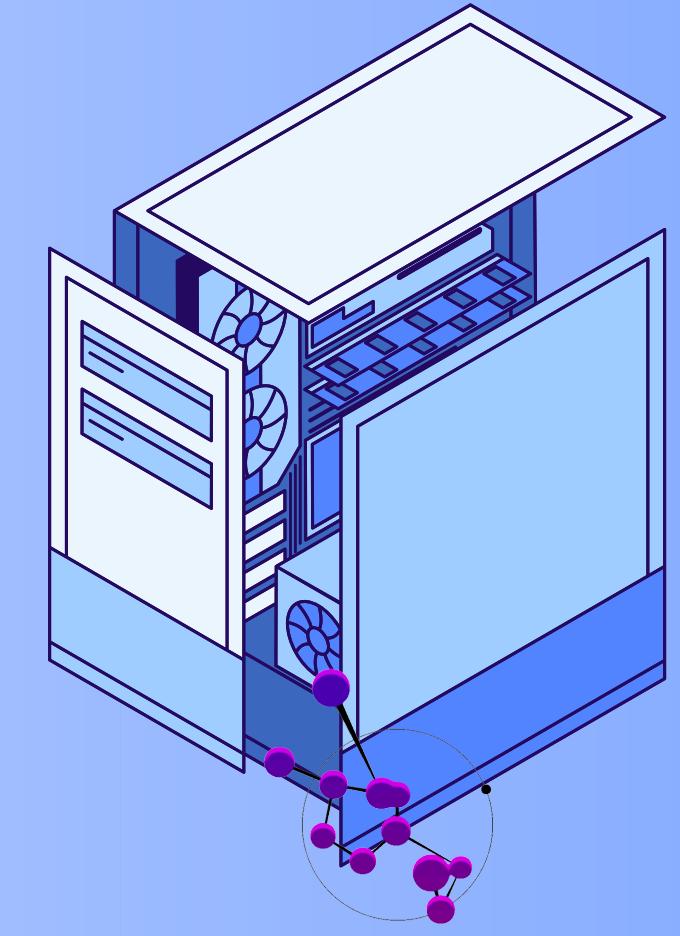
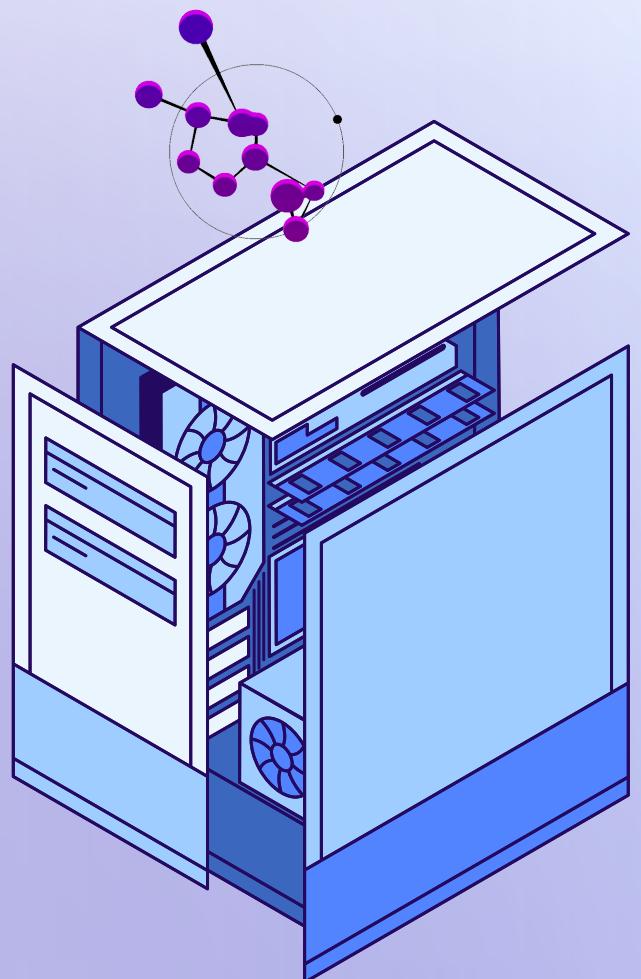
# MAEKAWA'S ALGORITHM





# MAEKAWA'S ALGORITHM FOR DISTRIBUTED MUTUAL EXCLUSION

- Subject: Distributed Systems
- Topic: Mutual Exclusion
- Prepared by: group 5





# INTRODUCTION

---

Distributed systems consist of multiple independent processes that share resources. To avoid conflicts, only one process must be allowed to enter the critical section at a time. Mutual exclusion ensures safe and consistent access to shared resources. Maekawa's algorithm provides a decentralized solution without using a central coordinator.



## NEED FOR MAEKAWA'S ALGORITHM

### Why Maekawa's Algorithm?

Traditional algorithms like Lamport and Ricart–Agrawala require communication with all processes.

This increases message overhead as the system size grows.

Maekawa's algorithm reduces communication by using a quorum-based permission system, making it more scalable for large distributed systems.





# BASIC CONCEPT



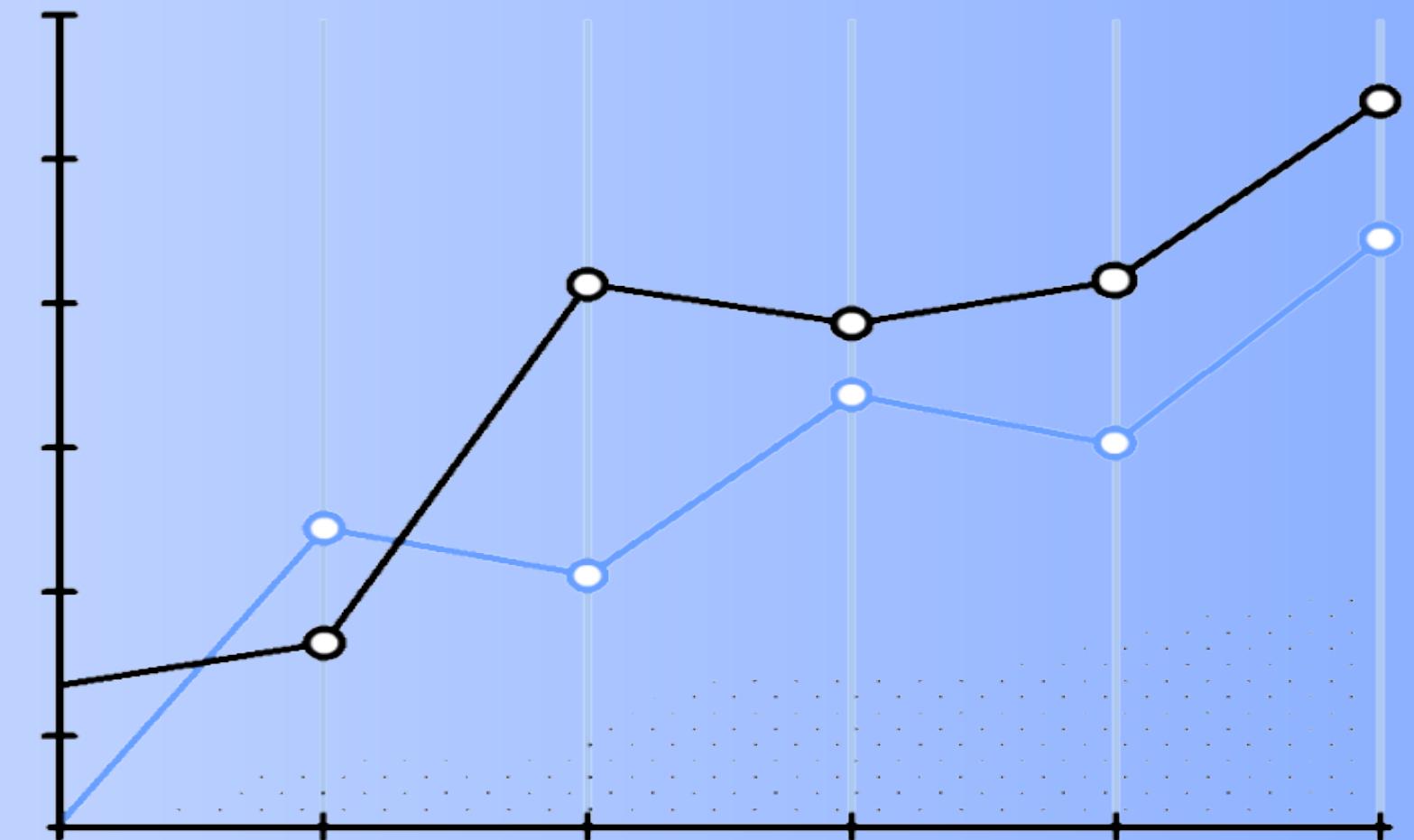
- Each process is assigned a small group of processes called a voting set or quorum.
- To enter the critical section, a process must obtain permission from all members of its quorum.
- Any two voting sets must intersect at least at one common process.
- This intersection guarantees mutual exclusion.

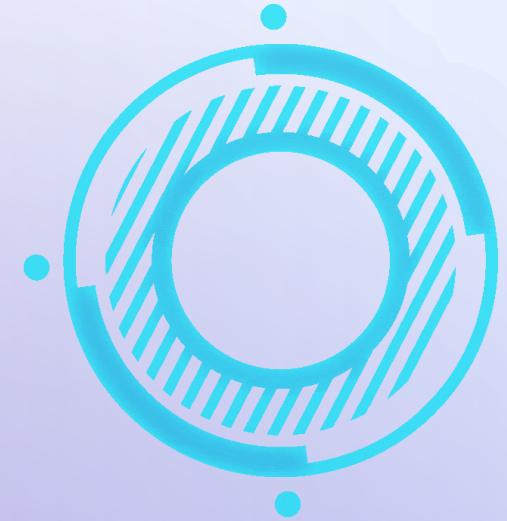




# QUORUM PROPERTIES

- For a system with  $N$  processes, each quorum contains approximately  $\sqrt{N}$  processes.
- Every voting set intersects with other voting sets.
- Each process participates in multiple voting sets,
- ensuring that no two processes can enter the critical section simultaneously.





# MESSAGE TYPES

**Request** is sent to ask permission to enter the critical section.

- GRANT is sent when permission is given.
- FAILED indicates the vote is currently unavailable.
- RELEASE is sent after exiting the critical section.
- INQUIRE and YIELD are used to resolve deadlocks.





Liceria Tech



# REQUEST PHASE

- When a process wants to enter the critical section,
- it sends REQUEST messages to all members of its voting set.
- Each request carries a timestamp to determine priority.
- A process can grant permission to only one request at a time.





# GRANTING RULES

- If a quorum member is free, it sends a GRANT message.
- If it has already granted permission to another process,
- It sends a FAILED message and queues the request
- Priority among requests is decided using timestamps.





**THANK YOU**

