IDS Projects

Vania Marangozova-Martin

2017-2018

Administrativia

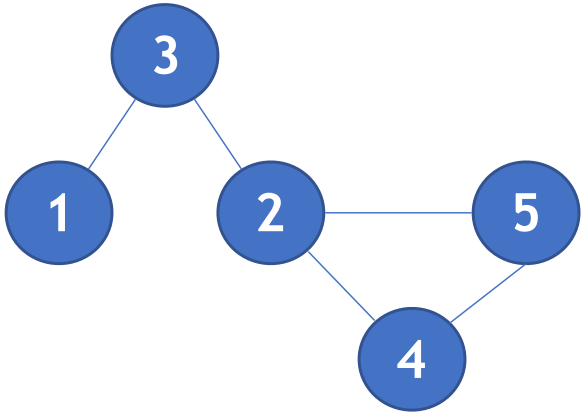
- Organisation
  - 2 members per group
  - ~10 different projects / group
  - Choice of subject is based on FIFO strategy
- Two types of projects
  - Algo-oriented:
    the focus is on the distributed algorithm you will need to put in place and you use Java RMI or RabbitMQ (technologies we use)
  - Techno-oriented:
    the focus is on the new technology you will need to learn
- Project advancement: the allocated time is not much, be sure to plan your work accordingly
- Presentation of your project
  - Slide presentation + demo
  - 15 minutes

Project 1 [ALGO]: Overlay (1)

	1	2	3	4	5
1			1		
2			1	1	1
3	1	1			
4		1			
5		1			

1. Input : matrix describing a graph

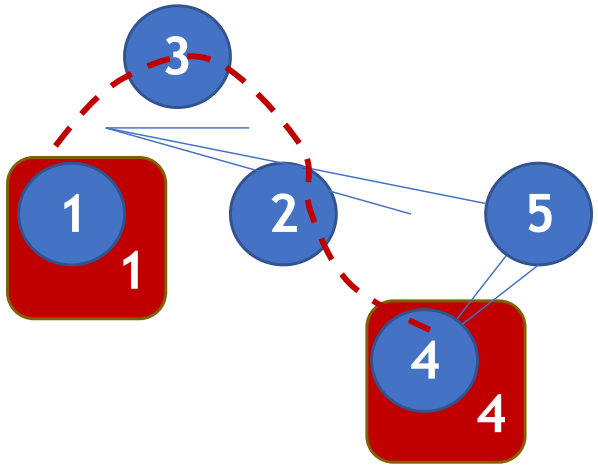
2. The graph gives the physical topology of your system

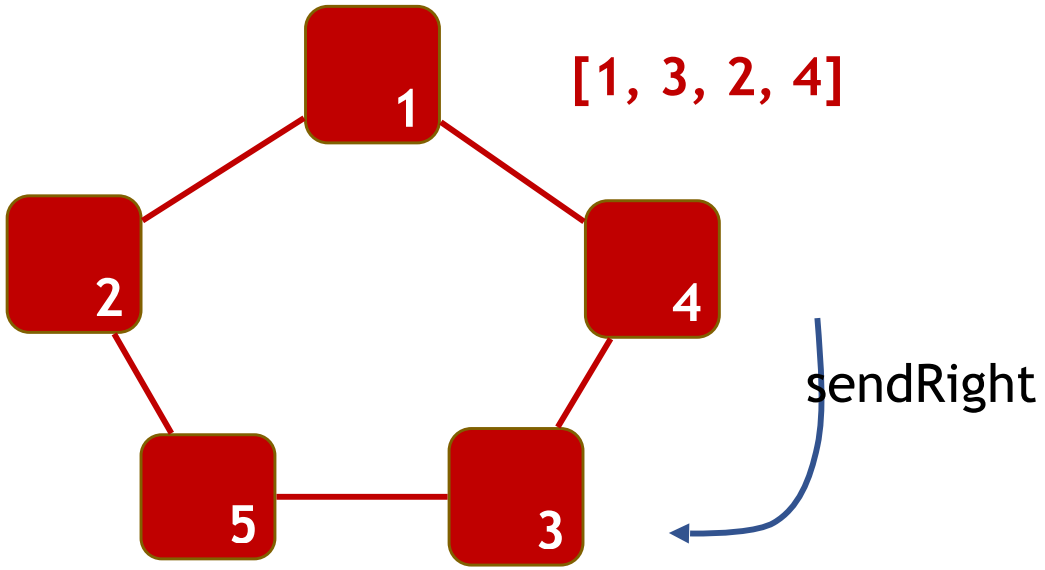


3. Instantiate it with RabbitMQ or Java RMI to have a running distributed system

Project 1 [ALGO]: Overlay (1)

4. Implement a virtual ring
   - Decide of the nodes identifiers
   - Decide who is neighbor of who
   - Compute the routing between a ring node and its two neighbors
   - `sendLeft(Message m)` and `sendRight(Message m)`



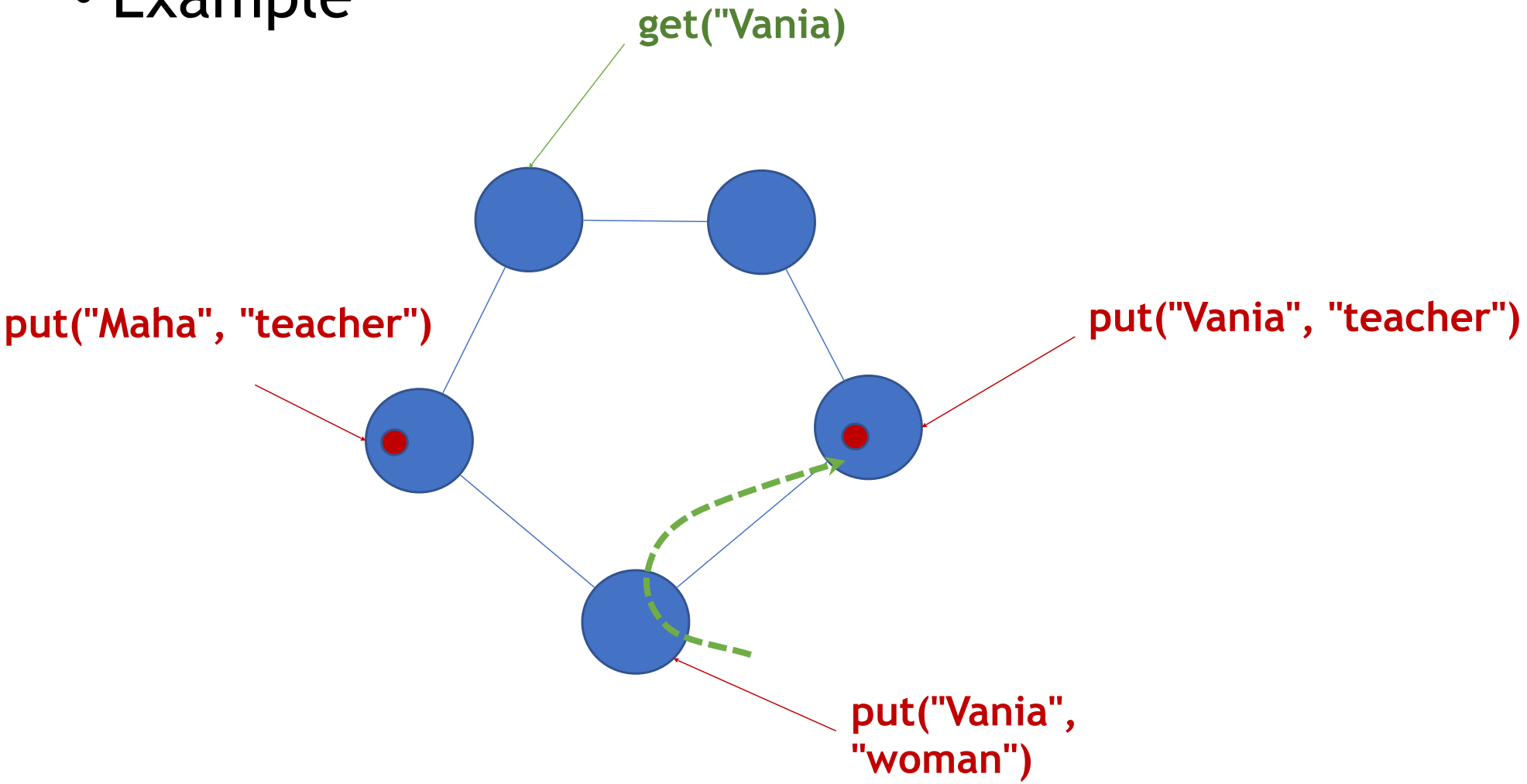


Project 2 [ALGO]: DHT (Distributed Hash Table) (1)

- Hashtable key -> value
- Implement a distributed hashtable
  - primitives `put(key,value)`, `get(key,value)`
  - there should not be two values with the same key
- Choose a distributed topology
  - ring is the simplest ☺
- Instantiate it with RabbitMQ or Java RMI

Project 2 [ALGO]: DHT (Distributed Hash Table) (2)

- Example

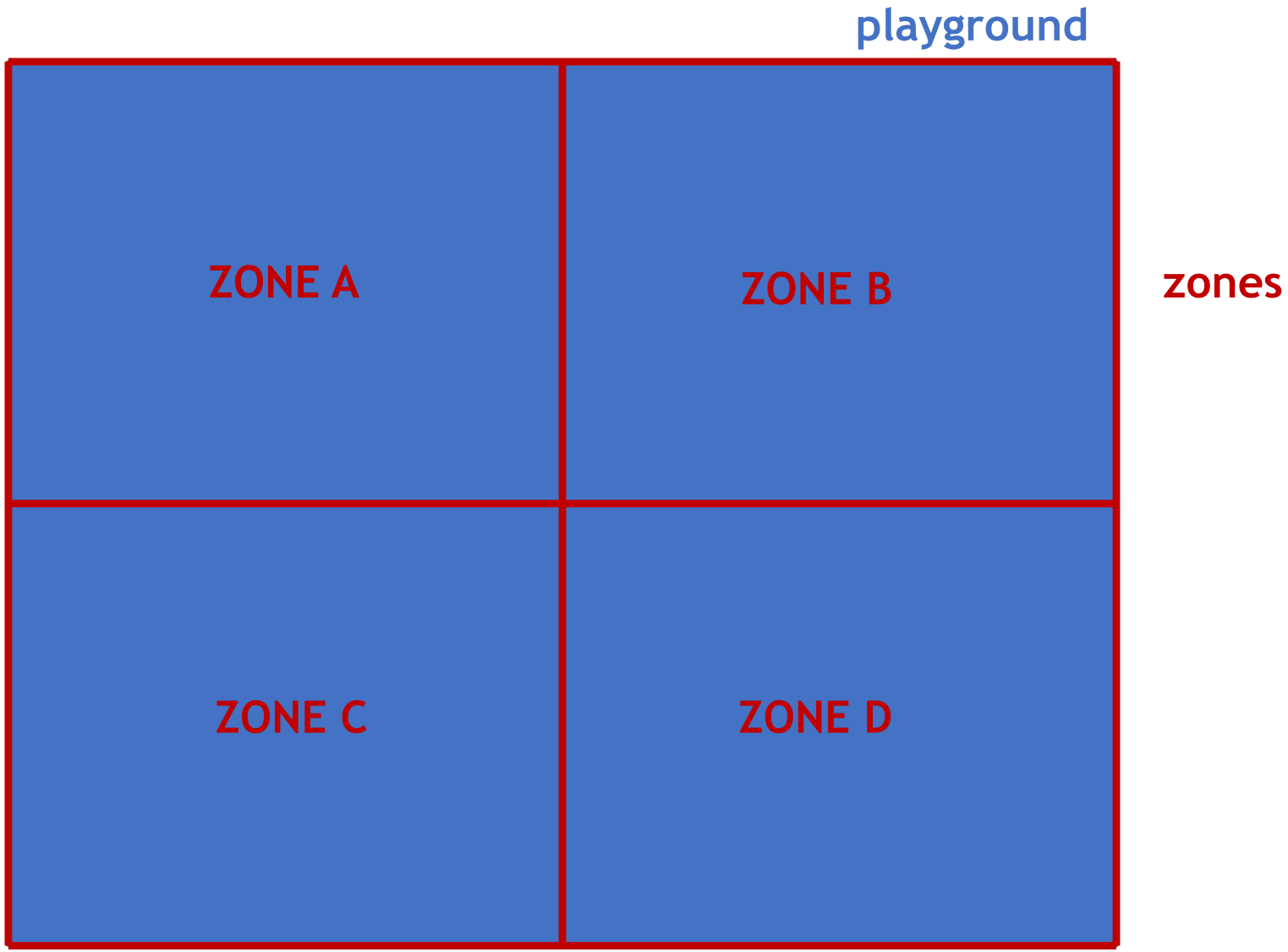


Project 3 [ALGO]: Multi-player Game (1)

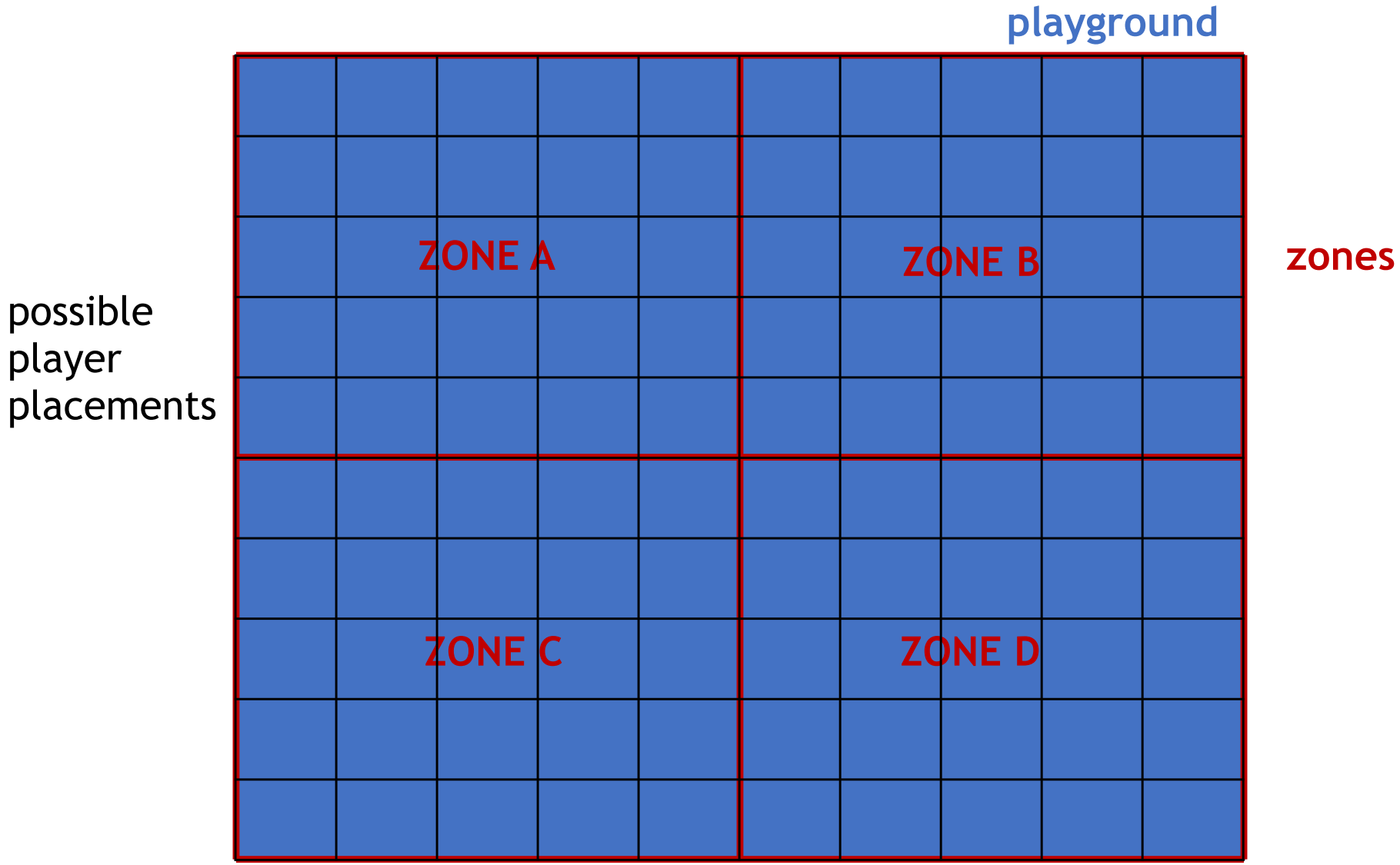
- Goal : Follow the movements of players
- Setting
  - The playground is separated in zones
    - you need to decide how
  - Players move around and pass from one zone to another
    - how do they move?
    - should avoid collisions
    - should say "Hello" to each other
  - Each zone is managed by a node
  - The nodes that manage the set of zones are connected in a distributed topology
    - You need to choose what topology

Project 3 [ALGO]: Multi-player Game (2)
EXAMPLE of a setting

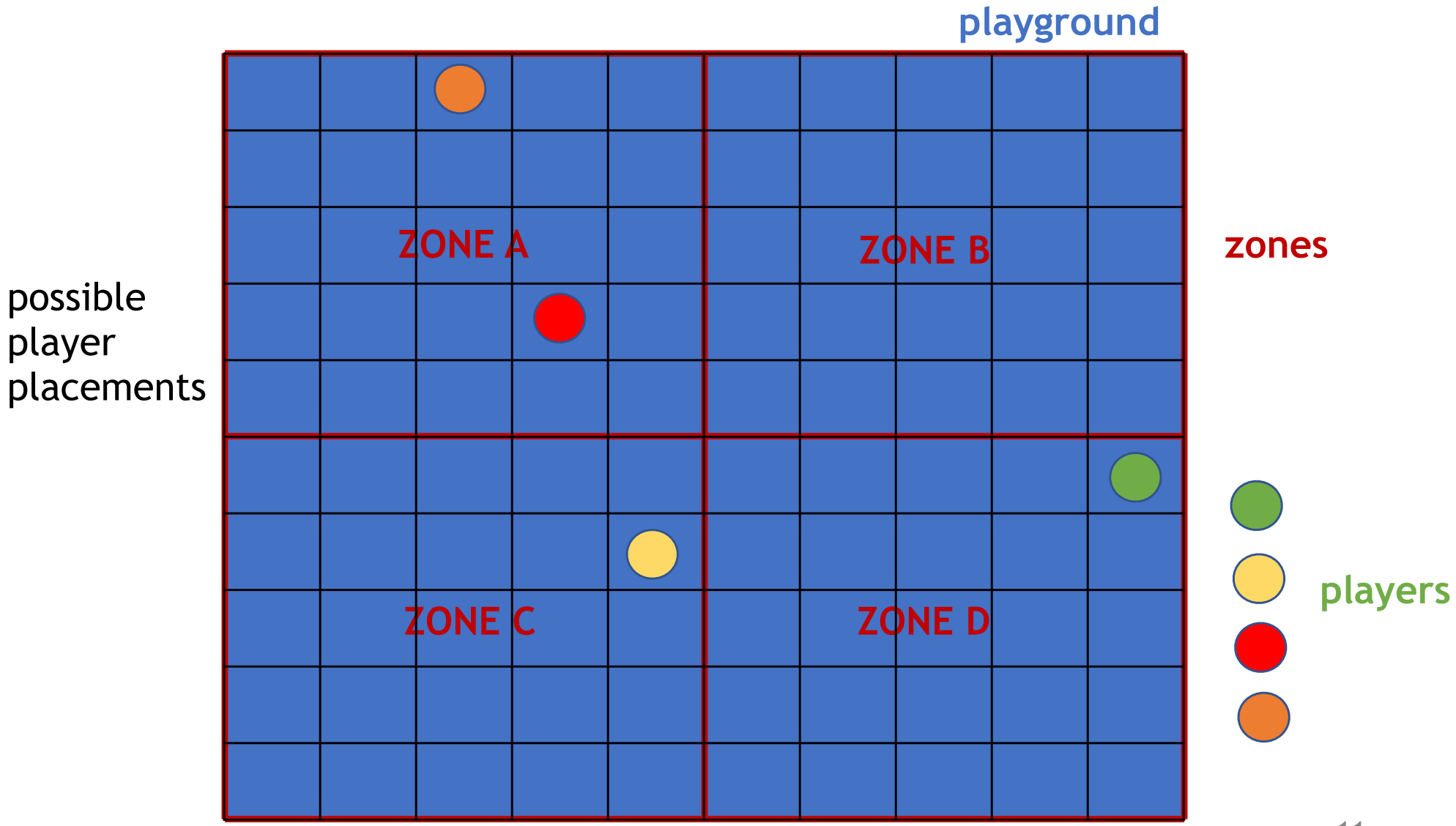
Project 3 [ALGO]: Multi-player Game (2)
EXAMPLE of a setting



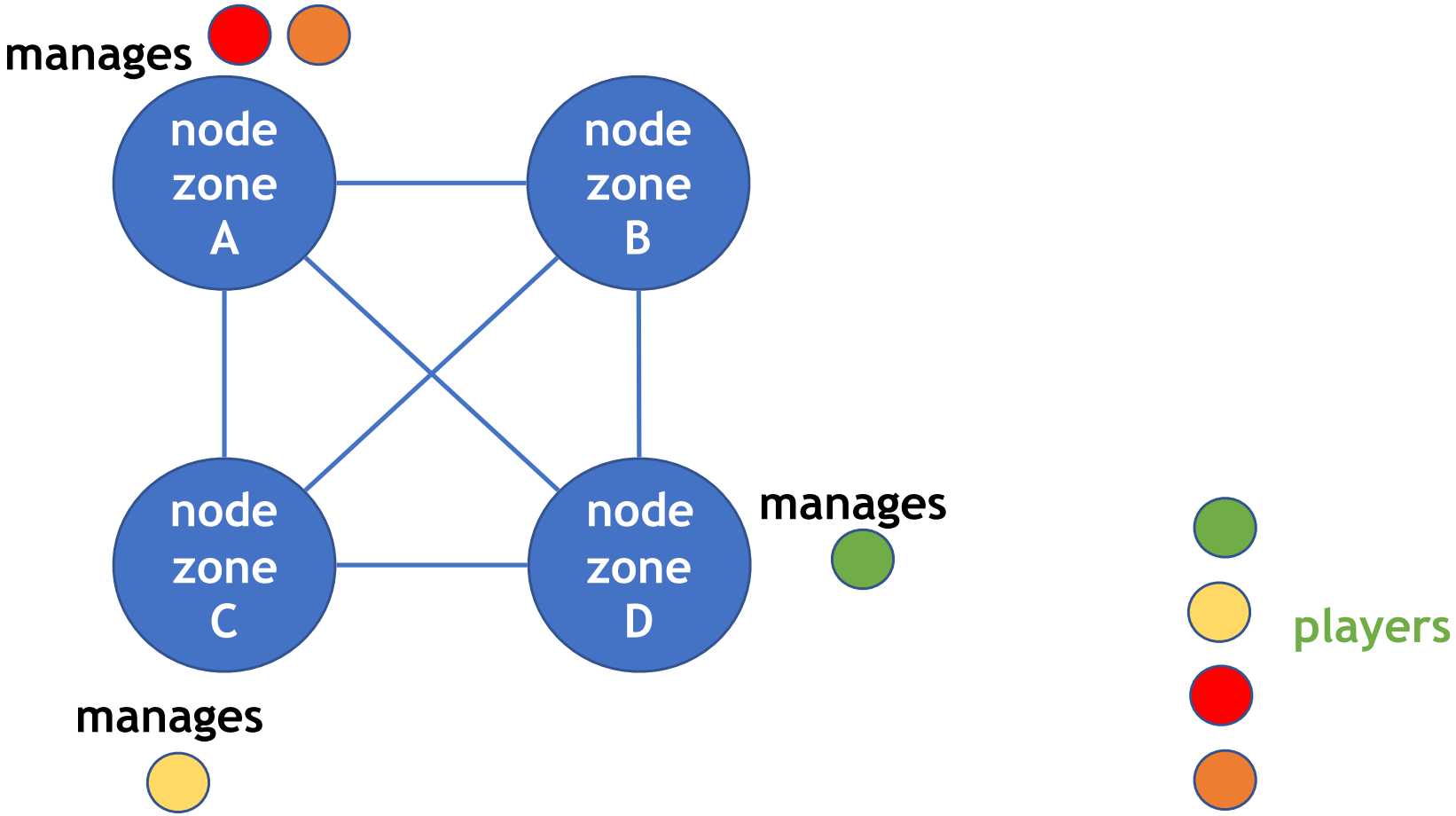
Project 3 [ALGO]: Multi-player Game (2) EXAMPLE of a setting



Project 3 [ALGO]: Multi-player Game (2) EXAMPLE of a setting

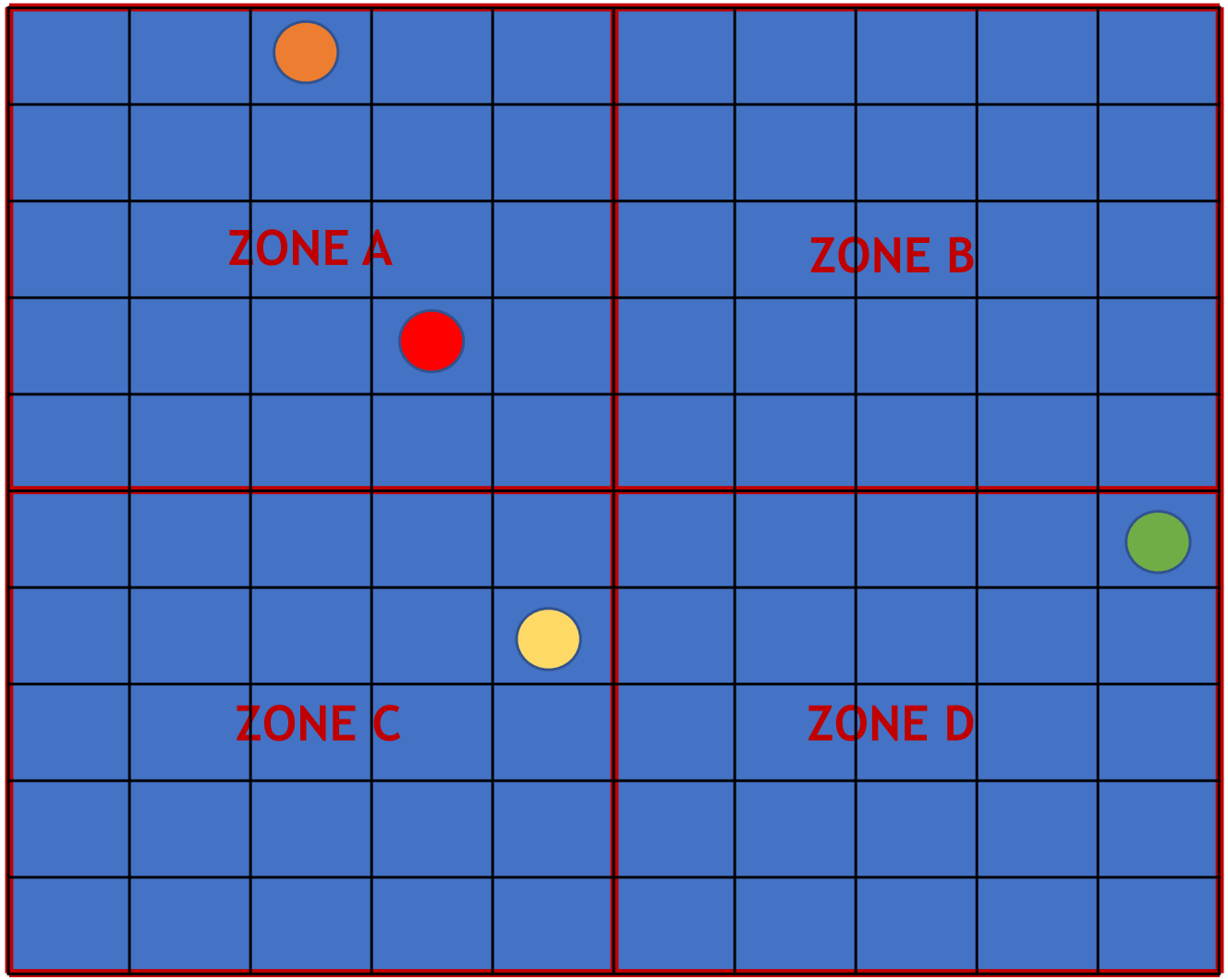


Project 3 [ALGO]: Multi-player Game (2)
EXAMPLE: The corresponding dist. architecture



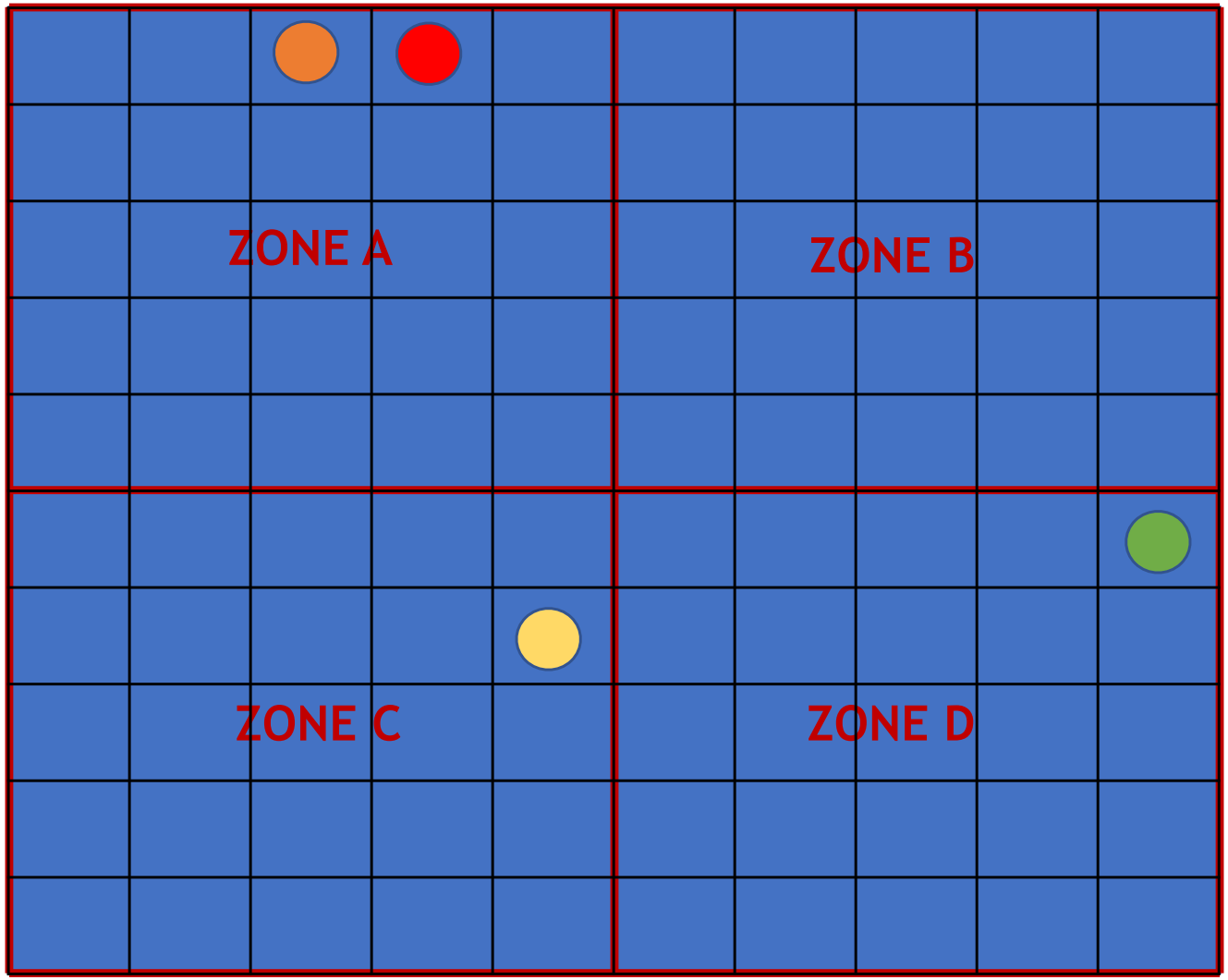
Project 3 : Multi-player Game (2)

Red says Hello to Orange



Project 3 : Multi-player Game (2)

Red says Hello to Orange



Green enters zone C

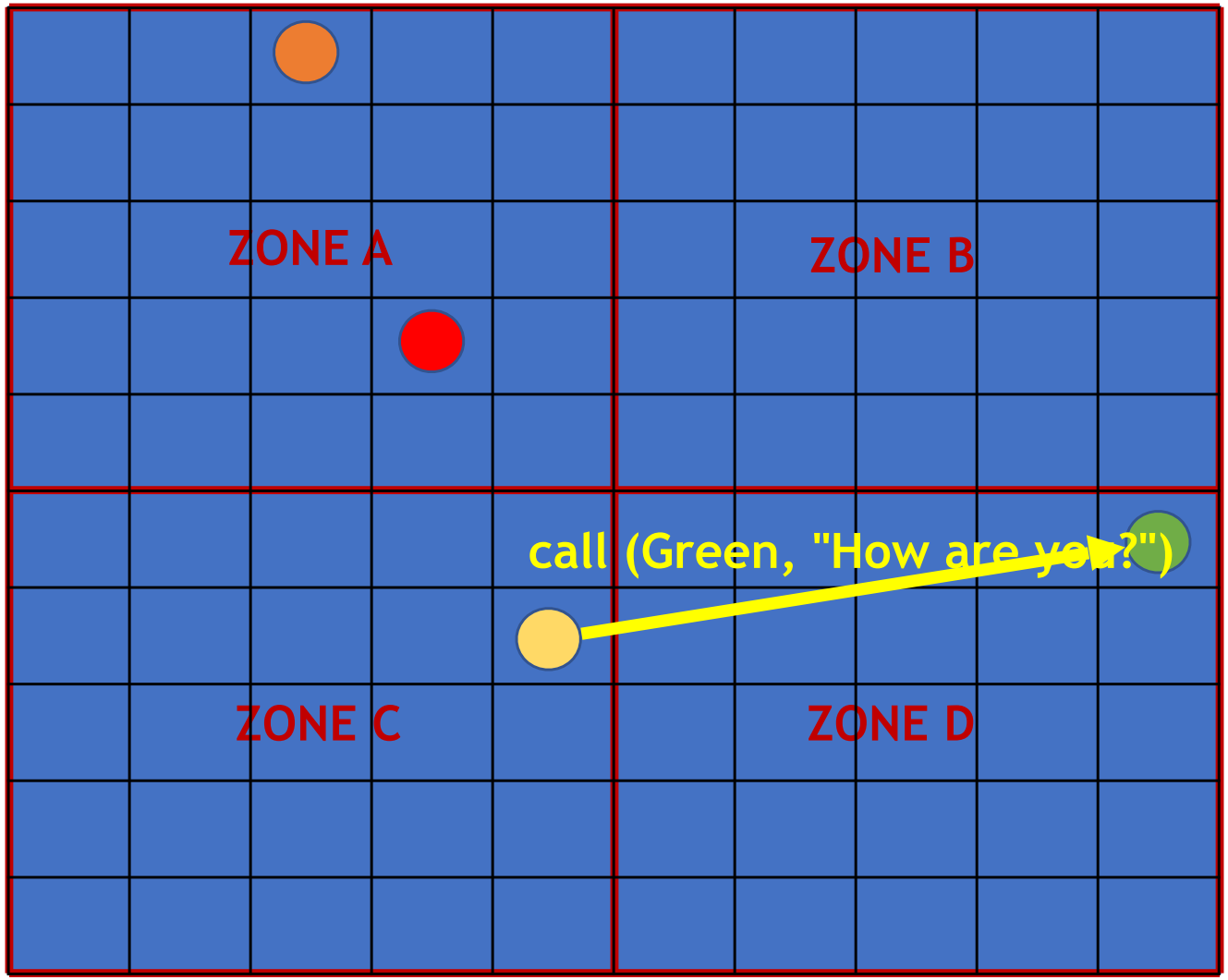
Project 4 [ALGO]: Phone calls routing (1)

- The same setting as for the multi-player game
- Users are in the zones
- A user should be able to call another user
  - locate the user
  - send him/her a message
  - primitive `call(userID, msg)`
- Start with stationary users
- Continue with mobile users

Project 4 [ALGO]: Phone calls routing (2)

EXAMPLE

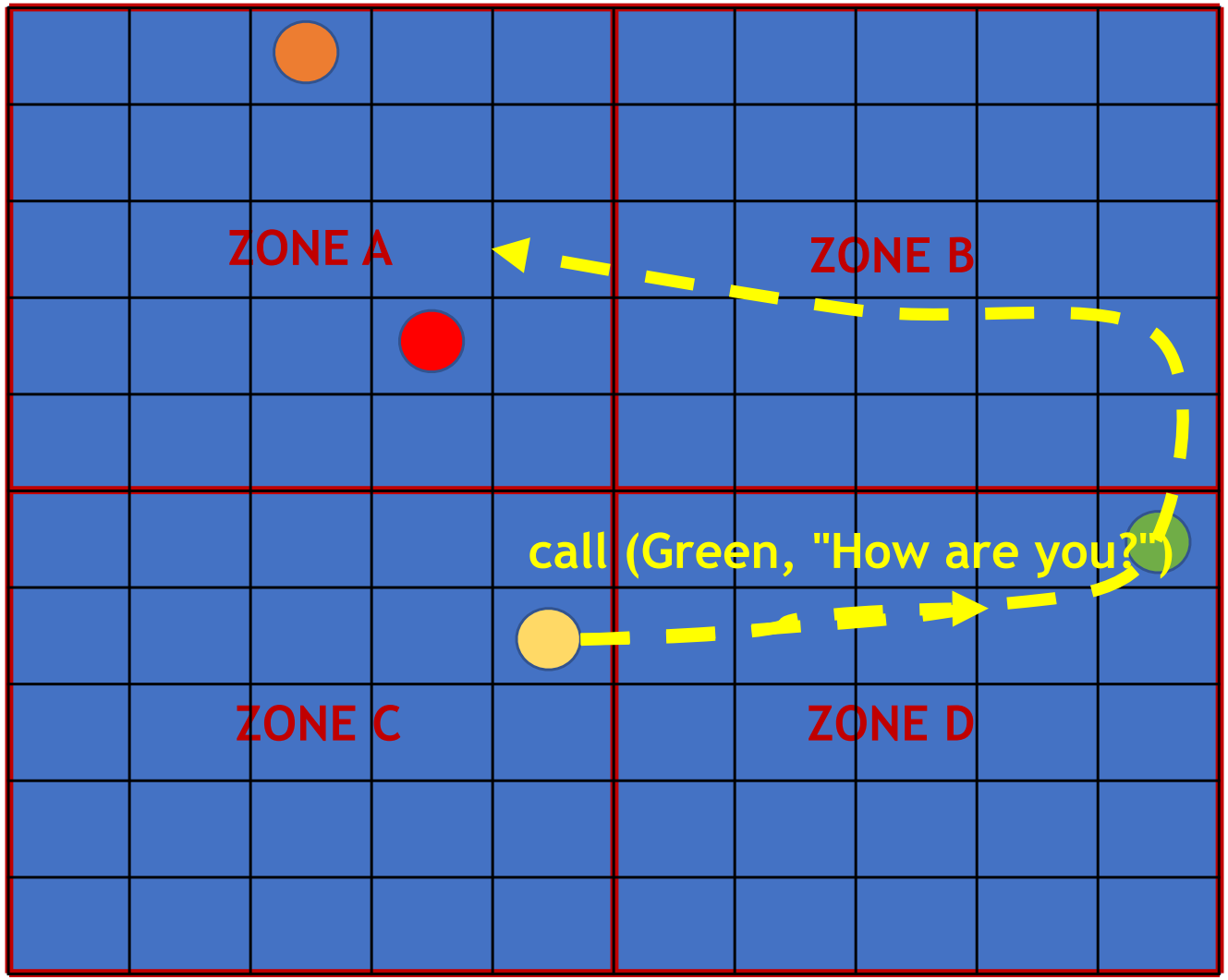
stationary users



Project 4 [ALGO]: Phone calls routing (3)

EXAMPLE

mobile users

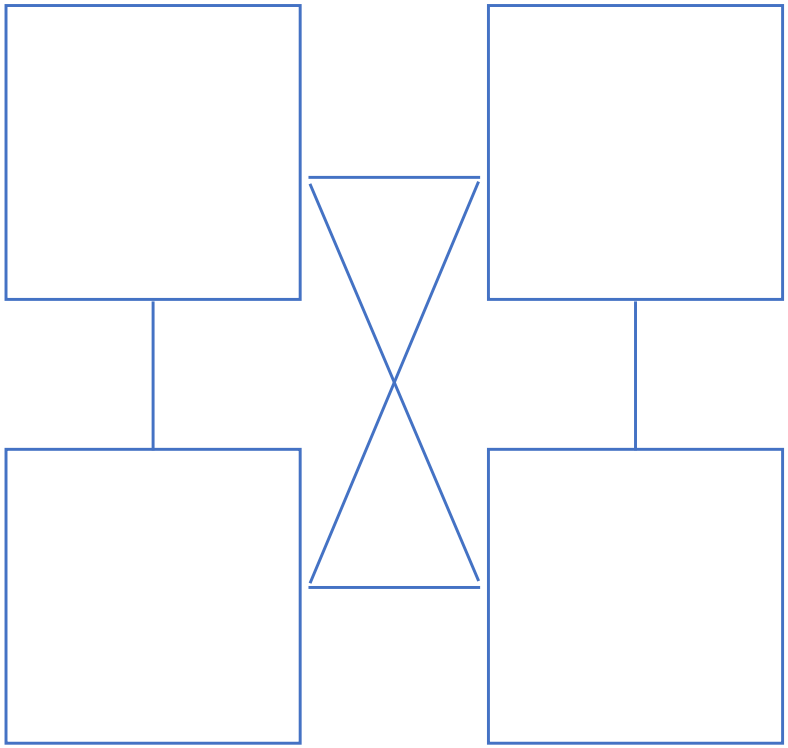


Project 5 [ALGO]: DSM (Distributed Shared Memory)

- Distribute a memory over several physical machines
- The machines are interconnected into a distributed topology
  - You need to choose the topology
- Implement the primitives
  - `value read(address)`
  - `write(address, value)`
- You need to decide what is an address and how to manage addresses
- You need to decide what kind of values you will manipulate

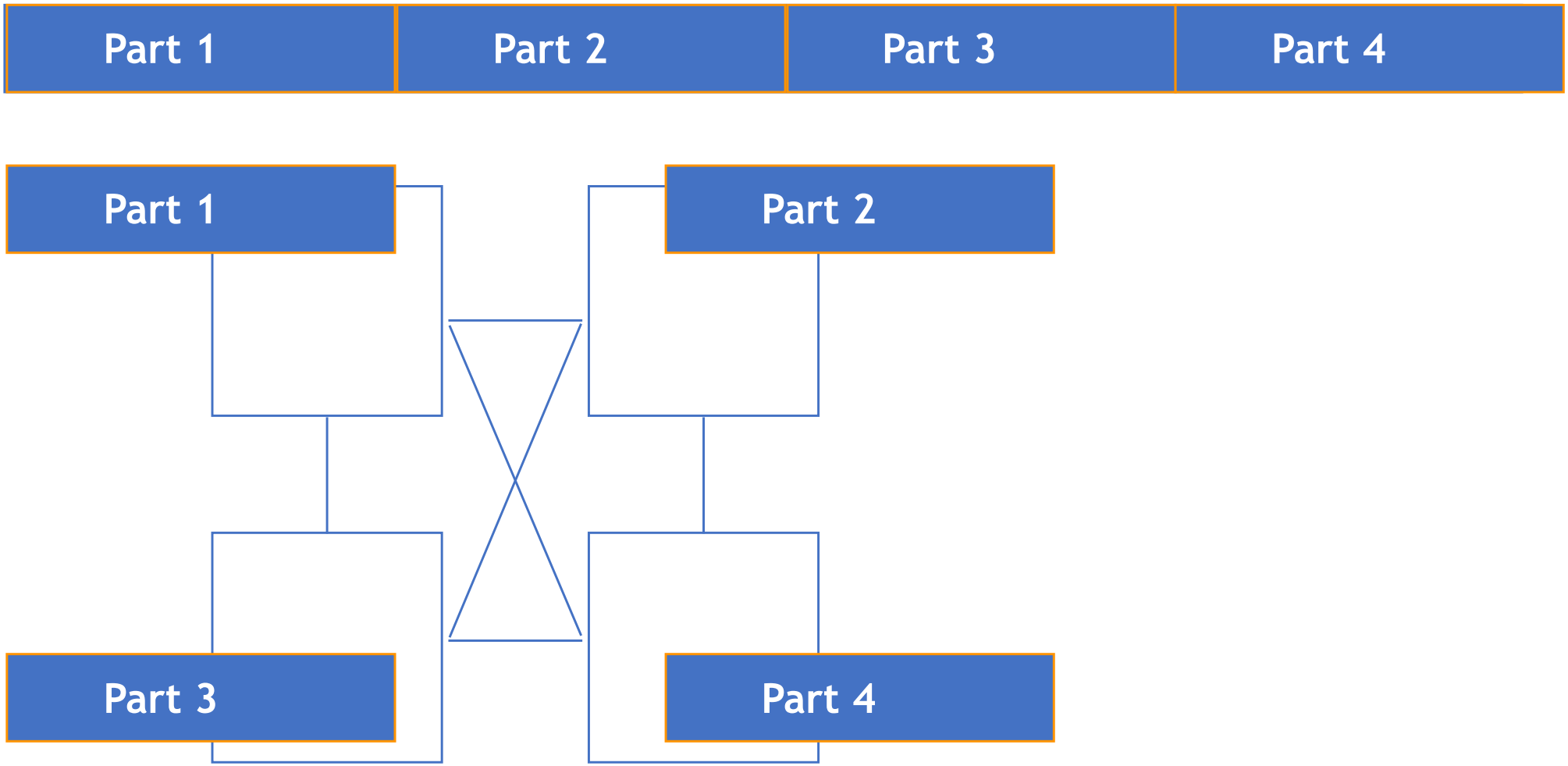
Project 5 [ALGO]: DSM EXAMPE Setting

Distributed Memory Abstraction

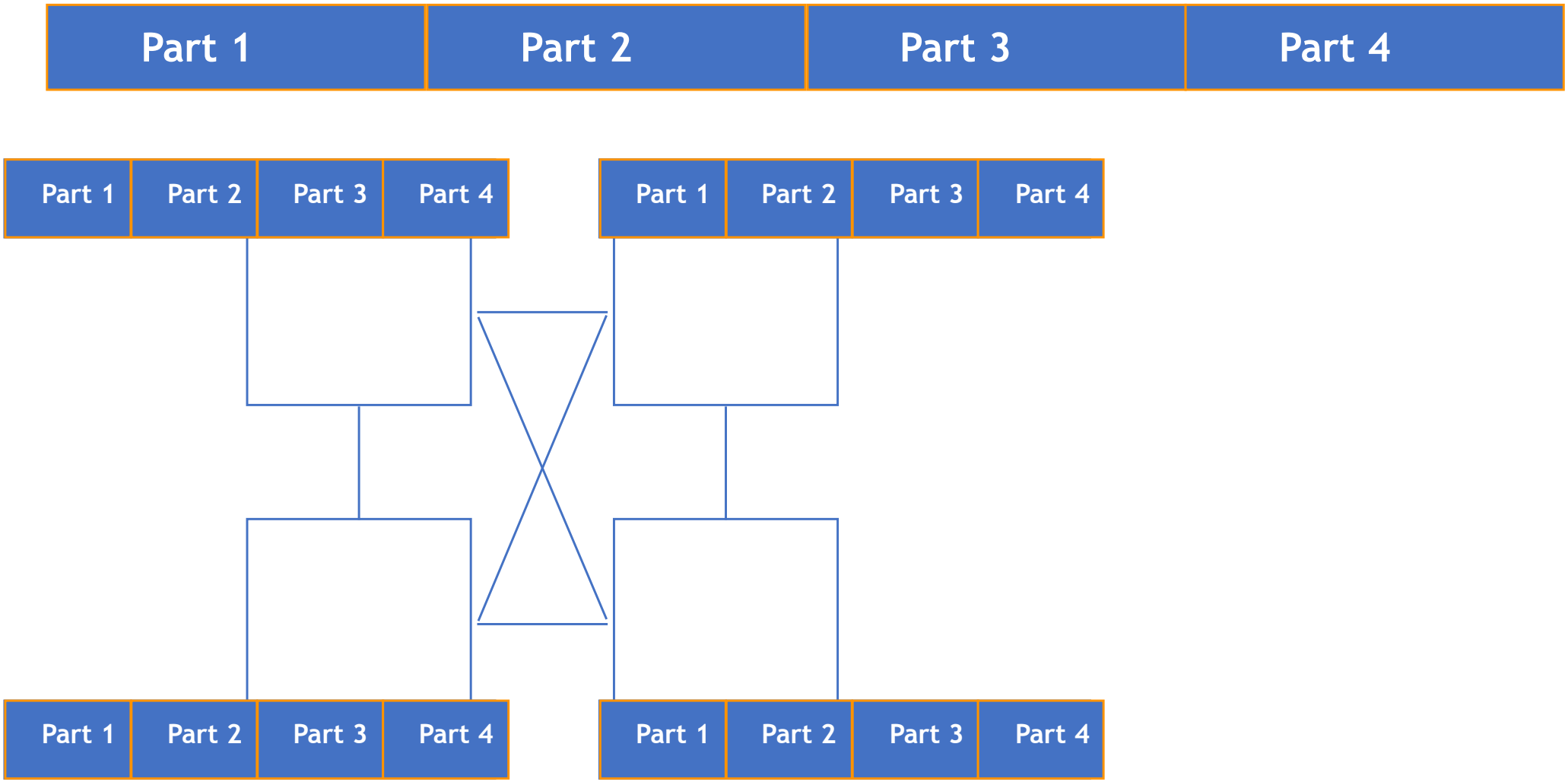


Physical machines in a grid

Project 5 [ALGO]: DSM EXAMPE with Partitioning



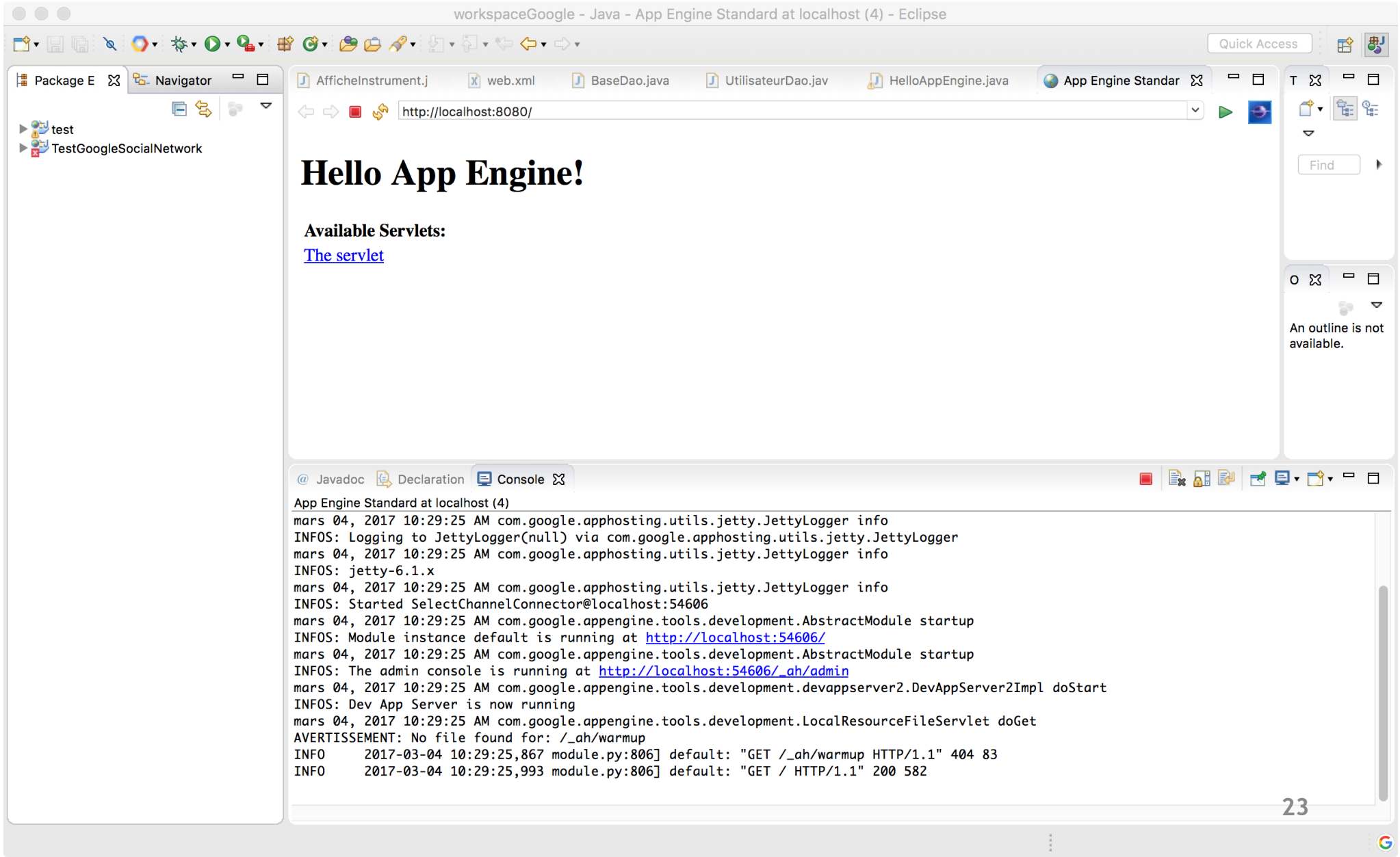
Project 5 [ALGO]: DSM EXAMPE with Replication



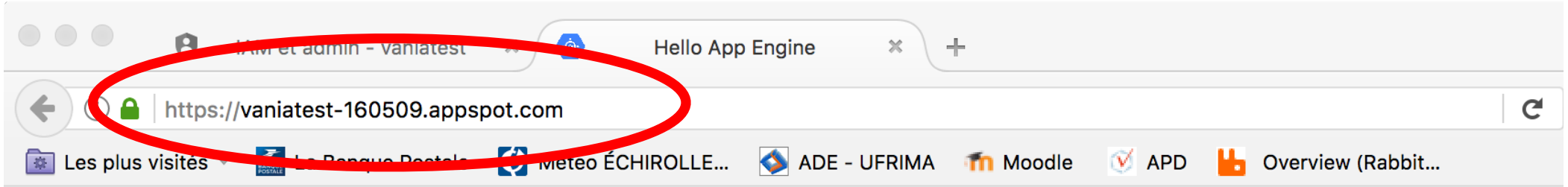
Project 6 [TECHNO] : Social network

- Use the Google App Engine environment and develop a social network
- Upload it to a Google server
  1. Install the Google tools
  2. Connect to Google and Google App Engine
  3. Learn about web servers, HTTP and Java servlets
  4. Possibly use existing tutorials
  5. Develop a social network service

Project 6 [TECHNO] : Social network



Project 6 [TECHNO] : Social network



Hello App Engine!

Available Servlets:

The servlet

Project 7 [TECHNO]: Bookstore

- Use the Google App Engine environment (or another cloud environment) and develop a bookstore online service
- If you use GoogleAppEngine, same steps as the social network project
- else you are on your own ☺

Project 8 [TECHNO]: Web Server (LAMP Architecture)

- Get to know Apache+MySQL based web servers
- Install, deploy and run a simple web server
- The web server may be a small static one or a more advanced dynamic sexy one…
- Put the accent on HTTP and requests
- or/and connect to the database and be able to execute simple data queries

Project 9 [TECHNO]: NoSQL Web Server

- Install, deploy and run a simple web server connected to a NoSQL database

Project 10 [TECHNO]: SQL vs NoSQL (Database performance)

- **Represent the same data with two databases**
- **Evaulate**
  - **the design issues**
  - **the performance**

Project 11 [ALGO] : News feed (Group management)

- Implement the group abstraction
  - `join, leave, publish` functions
- Variations to make it even more complex
  - All members receive messages in the same order
  - All members receive all messages even if network failures
  - Support for node failures & group reconstruction

Project 12 [TECHNO]: You propose :)

- Want to look into something?
- Develop a distributed app with it and then show it/present it/explain it to the class
- (Be sure to check with me beforehand)