Intelligent Boat Patrolling System

Project presentation – RSA 2023

Nuno Fahla - 97631

Miguel Tavares – 98448

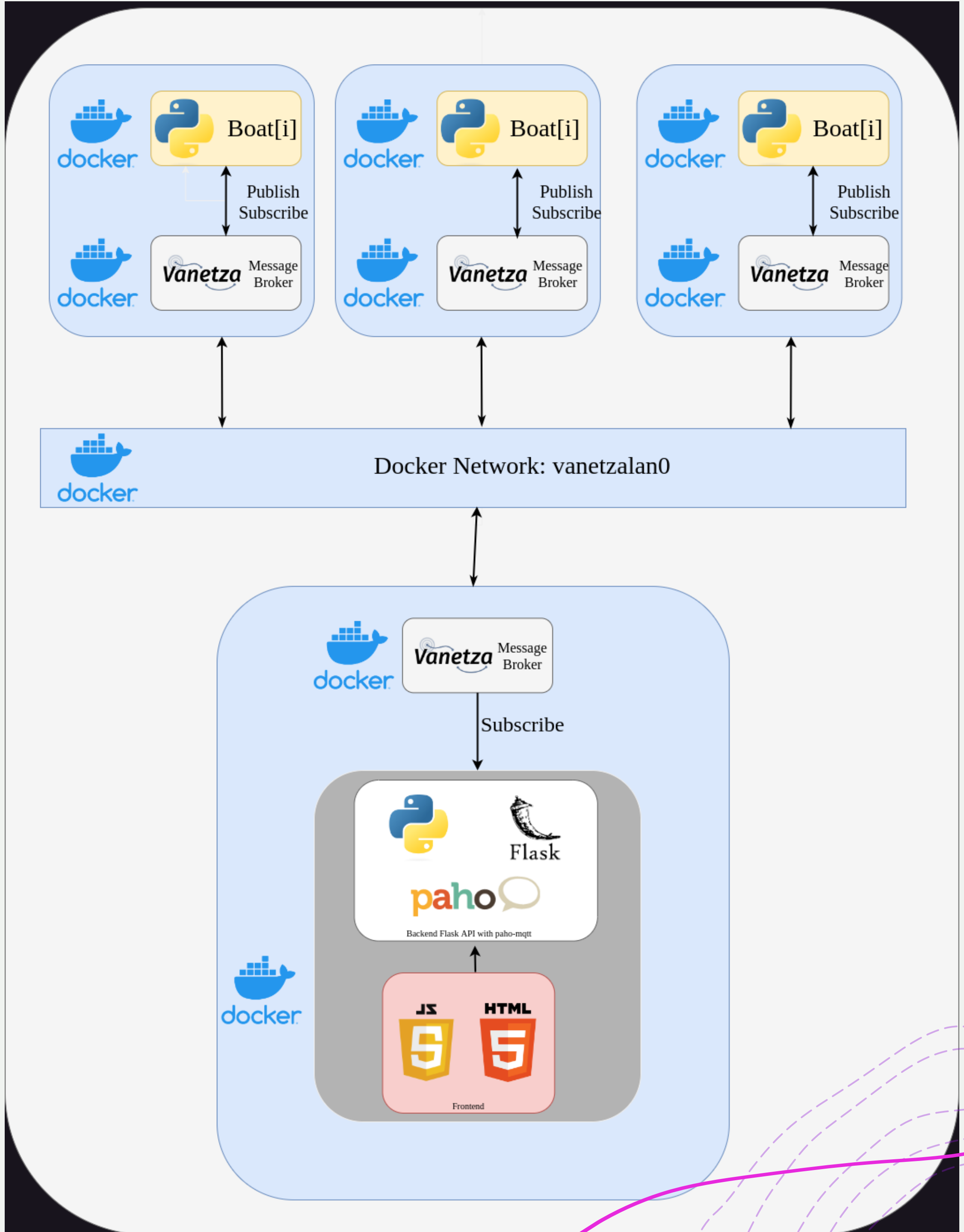
Objectives

- Develop a patrol system using boats in a maritime area to gather relevant information and scout the whole area as fast and as efficiently as possible
- The boats should patrol the whole area independently of one another, passing through previously patrolled areas as little as possible and avoiding collisions
- The area is divided into a grid with reference points, to ensure that the boats easily coordinate their movement
- Boats should warn one another of the patrolled reference points and share their current location in real-time with tracking platform
- They should form an ad-hoc network, to ensure that any number of boats can perform the task
- The task must be carried out successfully if at least only one boat survives

Messages

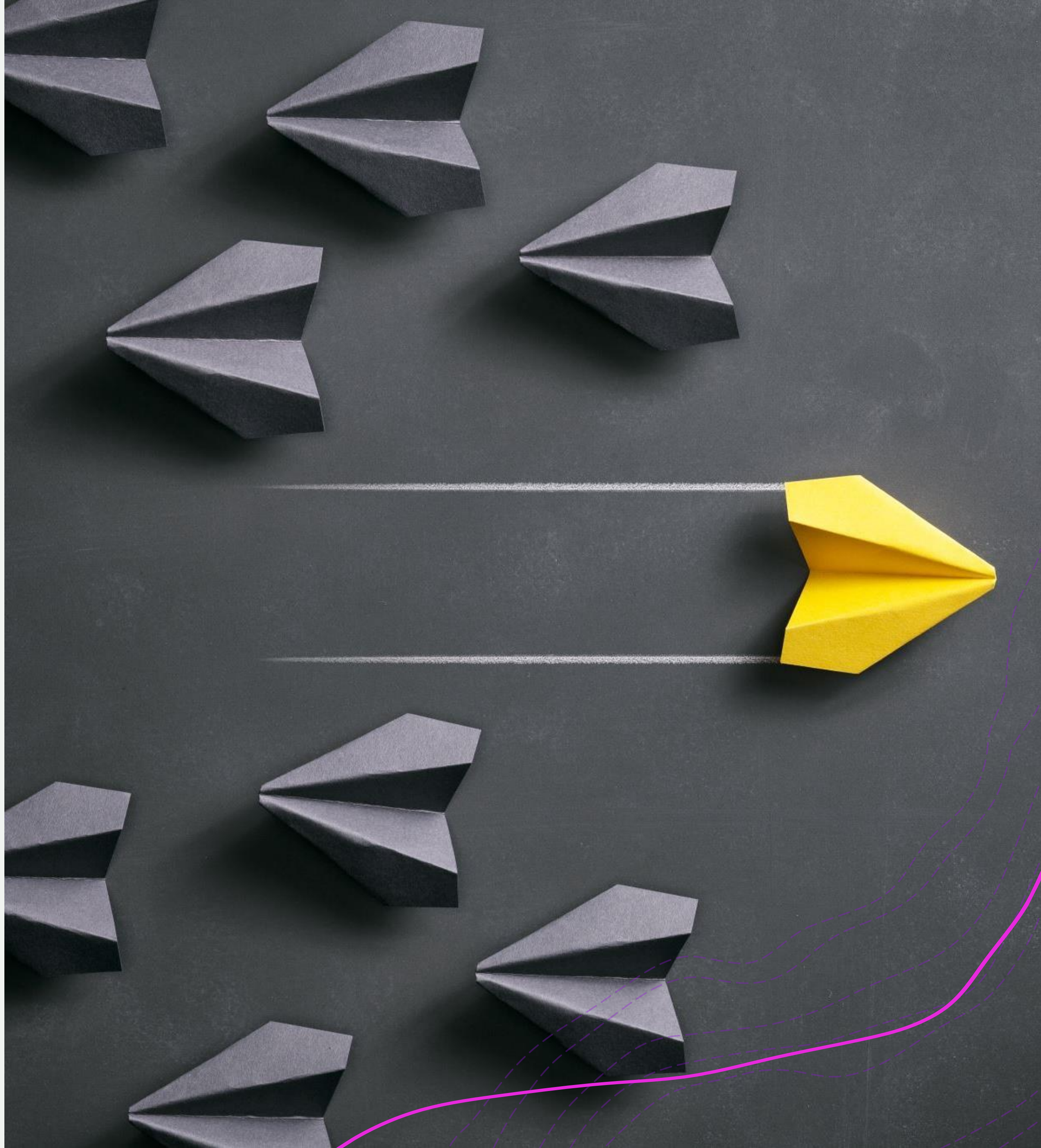
- CAMs (Cooperative Awareness Messages):
  - The visualization platform subscribes these messages to show in real-time the location of the boat sending them and the course it has plotted.
  - On simulation of each movement, the boat's python script publishes a CAM that is received by the Flask API's broker to represent the new coordinate through JS
- DENMs (Decentralized Environmental Notification Messages):
  - After a boat discovers a central point of the reference grid of the area, these messages are shared between boats to notify the discovery.
  - After arriving at that central point and sending the CAM, the boat's python script publishes a CAM that is received by all containers in the docker network that are also emulating a boat

General Architecture



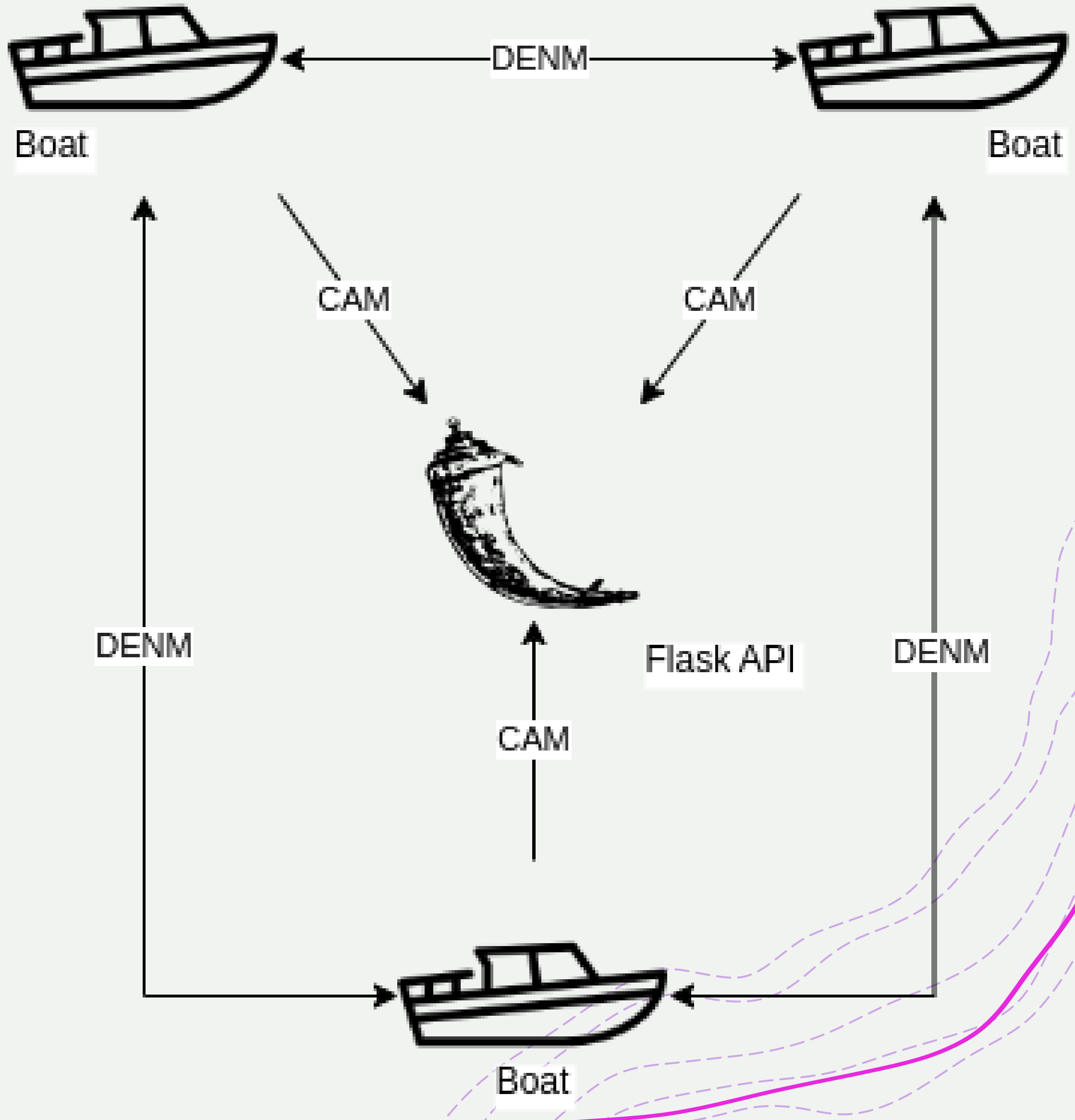
Timeline (setup)

- The simulation needs at least one boat present. The maximum number of boats is equal to the number of central points in the grid
- The boats are initialized with environment variables defining the patrolling area's limits and the precision of the grid's central points (simulates range of boat's vision), in the *Dockerfile*
- In *docker-compose.yml* we can define a specific boat's starting point and amount it moves(simulates speed)
- The boats start by calculating the center points of the grid with the area's limits, connect to their respective Vanetza container acting as the embedded broker and publish their starting location to the Flask API
- They are now ready to start their life-cycle



Timeline (boat life-cycle)

- In the begging of the life-cycle, the nearest point is calculated
- Then, given the current position and the current destination(closest point), a list of intermediate coordinates are generated
- For each point of the list, the boat publishes a CAM with the information, and sleep for some milliseconds, simulating the time it took to move from the previous point
- After that, if the current point is also the predicted destination, the boat publishes a DENM with that information(center point of the grid)
- If the boat isn't at the predicted destination, it means the life-cycle was interrupted by an incoming DENM saying that this boat's current destination was already discovered
- The cycle stops when all center points of the grid have been visited, shutting down the boat
- If a boat fails, it stays in the current position, waiting for the maintenance team


Boat
DENM
Boat
CAM
CAM
DENM
Flask API
DENM
CAM
Boat