“Web and Real Time Communication Systems” project

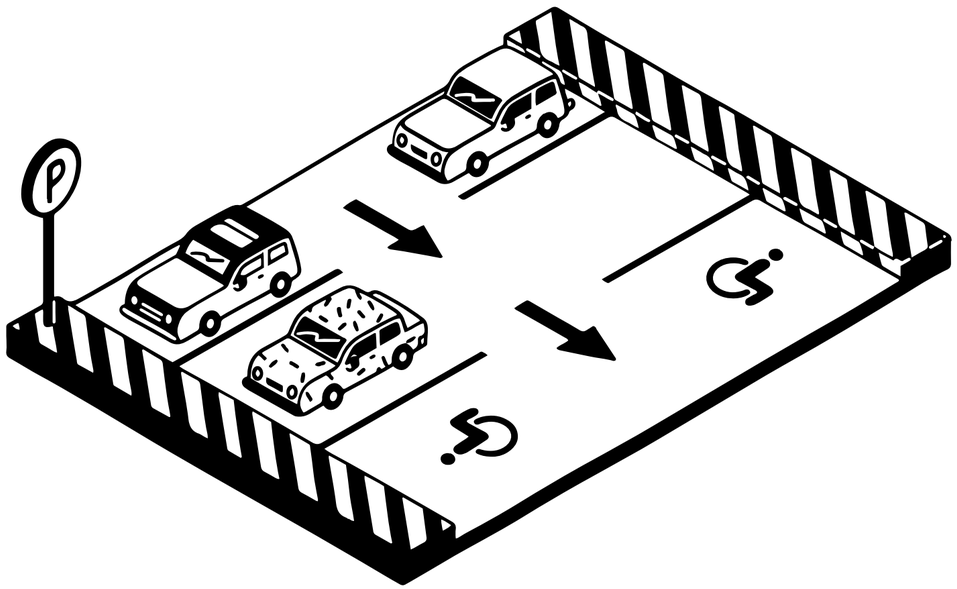
“Distributed Systems and IoT” project

DI.P.S. 

Antonio Boccarossa M63001643

Francesco Brunello M63001655

Table of CONTENTS



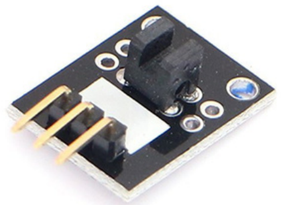
ABSTRACT

This project aims to manage a parking lot by communicating the entries and exits with a central server, where the number of available parking spaces is stored. The goal is to demonstrate the concurrent interaction between multiple system components and the parking lot "manager" component.

In future versions of the project, we plan to implement a distributed election algorithm (for example, the Bully algorithm) in case the server fails (assuming crash-type failures), as well as redundancy for the server to ensure additional robustness for the system.

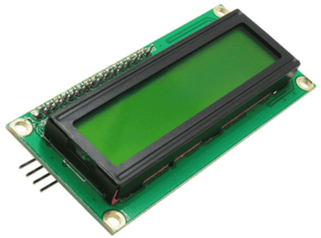
Components

5x ESP-WROOM-32



4x Photo-interrupter Modules

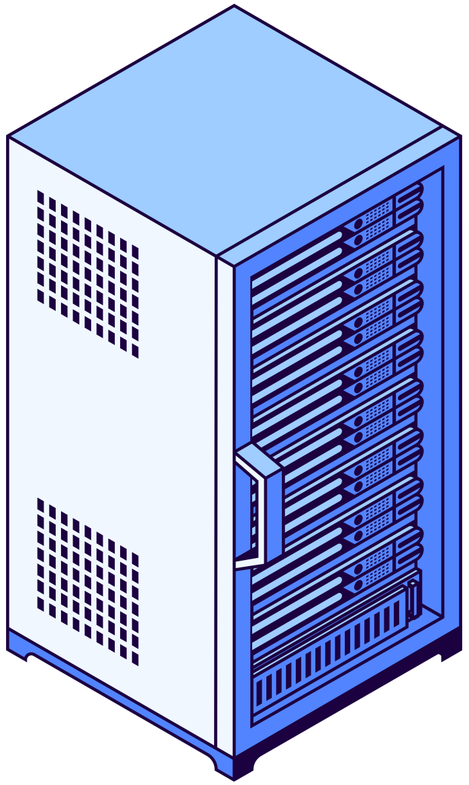
4x Servomotors SG90



1x “16x2” LCD I2C

“““Light””” Assumptions

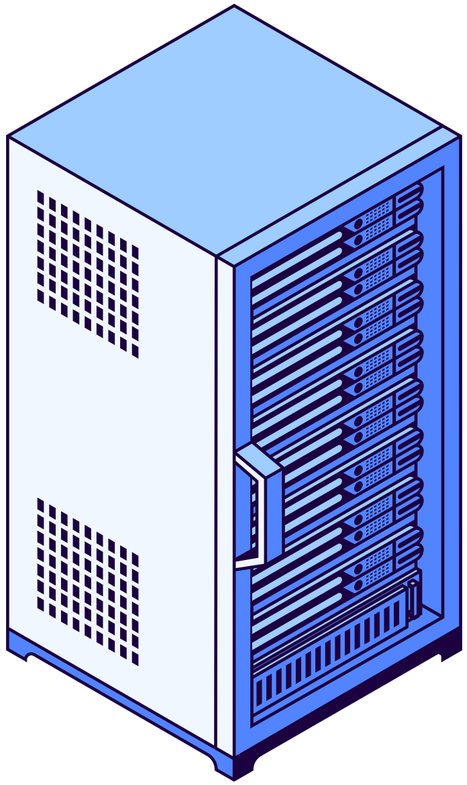
- **SERVER NEVER FAILS**
- **CLIENTS NEVER FAIL**
- **CONNECTIONS NEVER FAIL**
- **THE SYSTEM IS PARTIALLY SYNCHRONOUS**



“Lock Server” Algorithm

REQUIREMENTS:

- **SAFETY:** THIS IS ENSURED BY THE SERVER, WHICH IS THE ONLY ENTITY THAT CAN GRANT ACCESS TO THE SECTION.
- **LIVENESS:** THIS IS GUARANTEED BY THE RELIABILITY OF THE COMMUNICATION CHANNELS AND THE SERVER.
- **ORDERING:** THIS IS **NOT** SATISFIED BECAUSE THE SERVER PROCESSES REQUESTS USING A “FIFO” APPROACH.

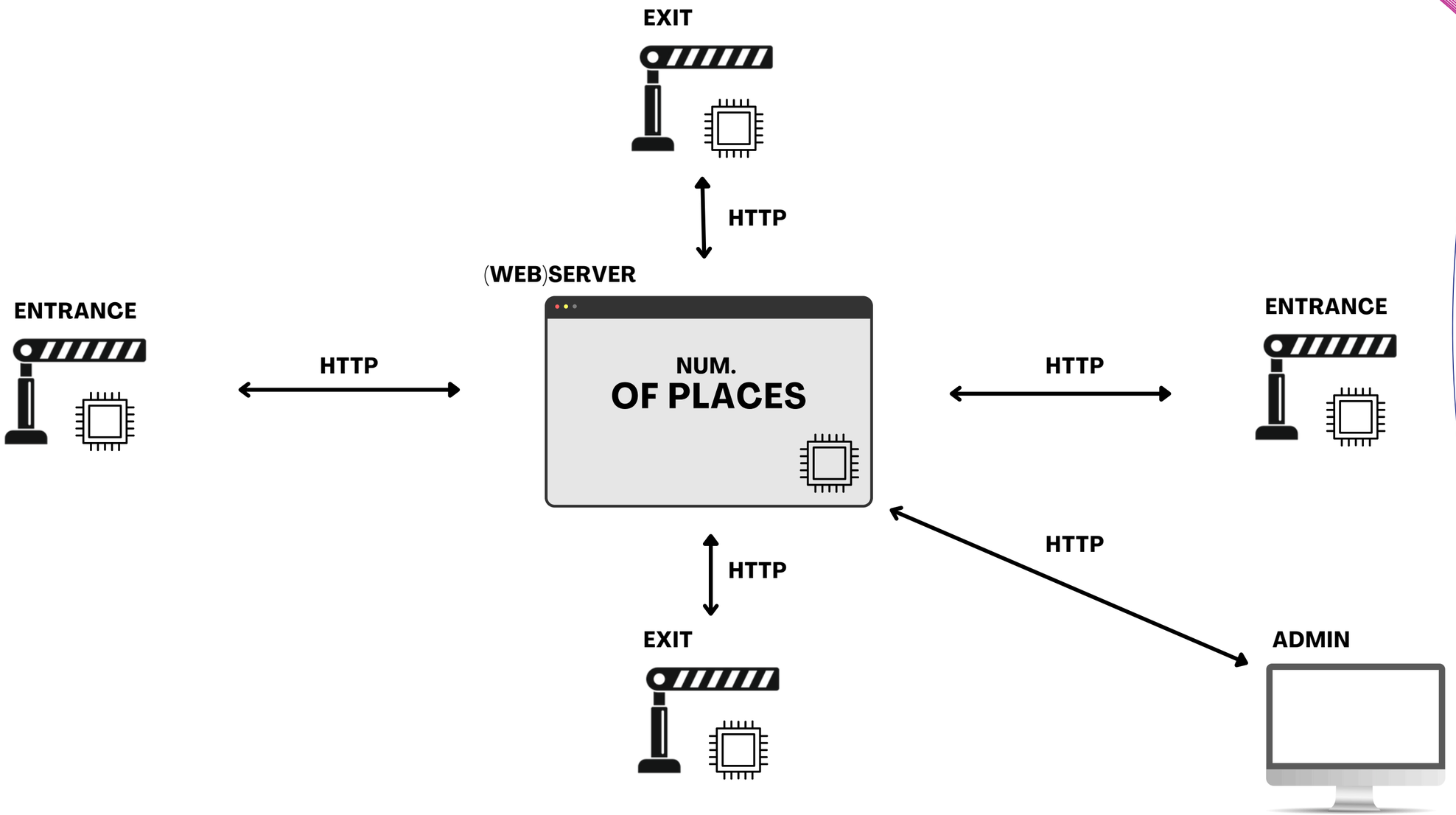


“Lock Server” Algorithm

EVALUATION:

- **NETWORK BANDWIDTH USED:** THIS IS PROPORTIONAL TO THE NUMBER OF REQUESTS TO ACCESS THE SECTION.
- **CLIENT DELAY:** THE DELAY EXPERIENCED BY A SINGLE CLIENT IS 2 MESSAGES (*REQUEST LOCK* AND *GRANT LOCK (ACK IN THIS CASE)*).
- **SYNCHRONIZATION DELAY:** THE DELAY BETWEEN ONE CLIENT EXITING AND ANOTHER CLIENT ENTERING THE SECTION IS 2 MESSAGES (*RELEASE LOCK* AND *GRANT LOCK* TO A PENDING PROCESS).

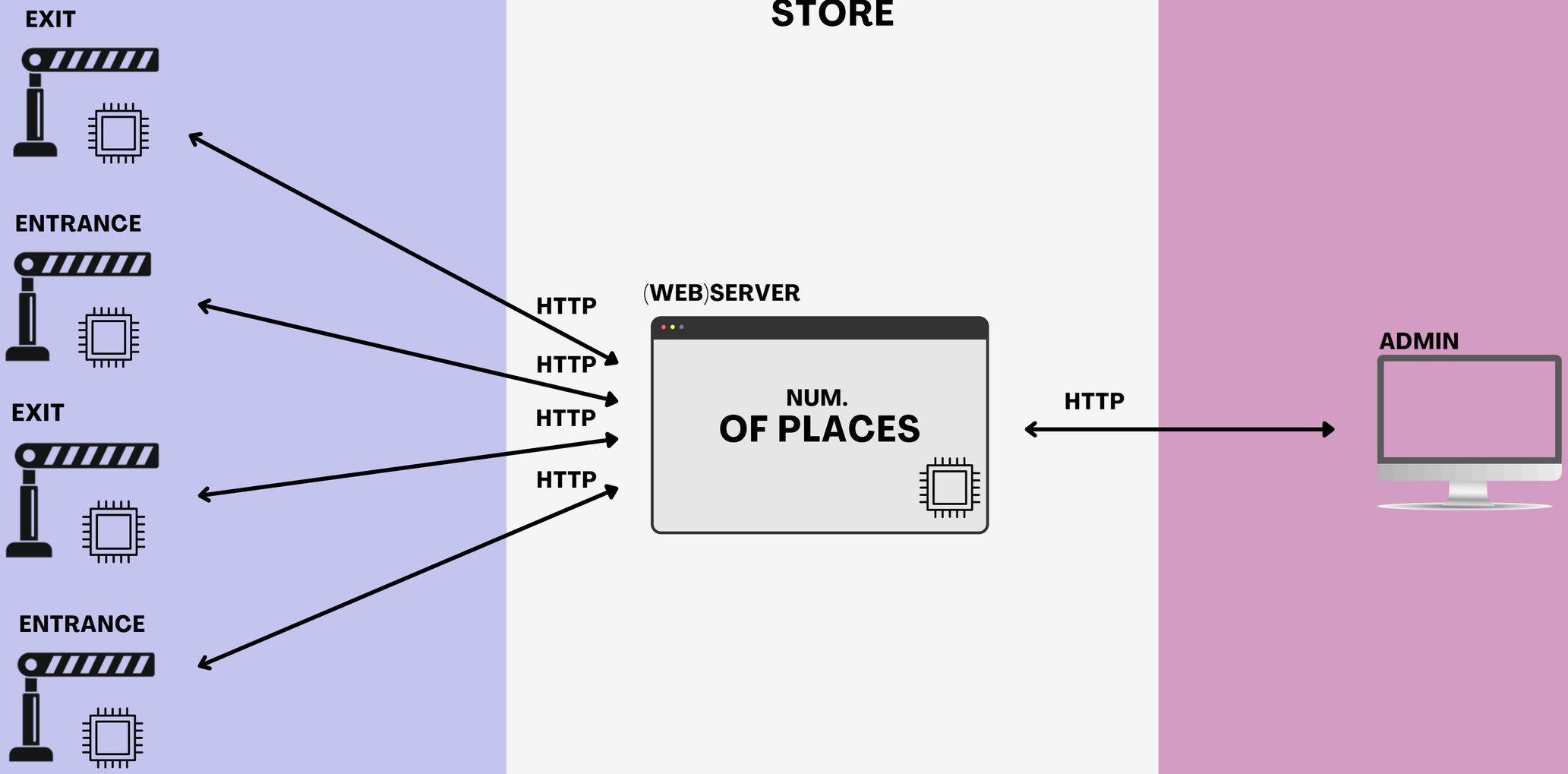
How Does it Work?



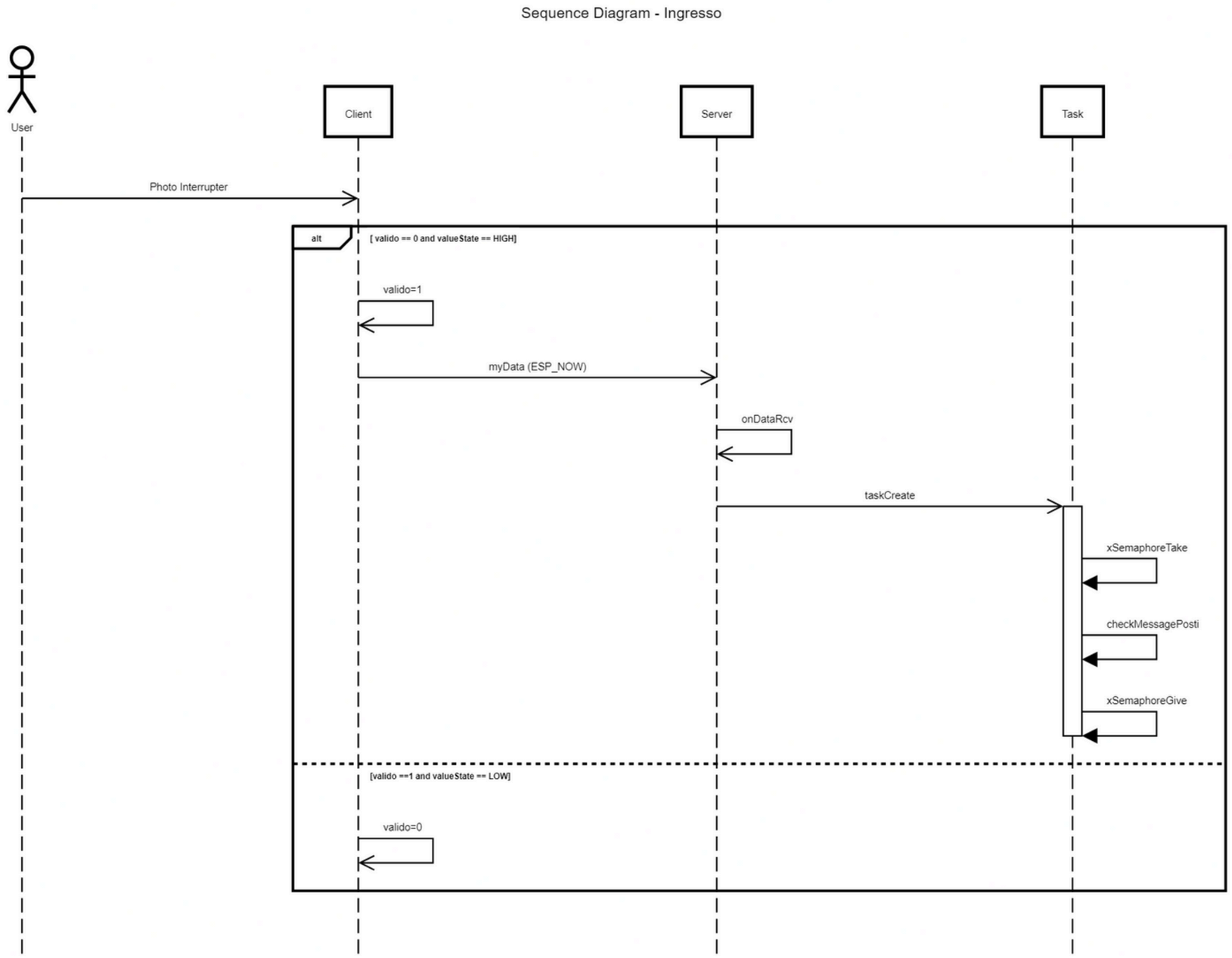
LOCAL ENVIRONMENT

TRANSPORT & STORE

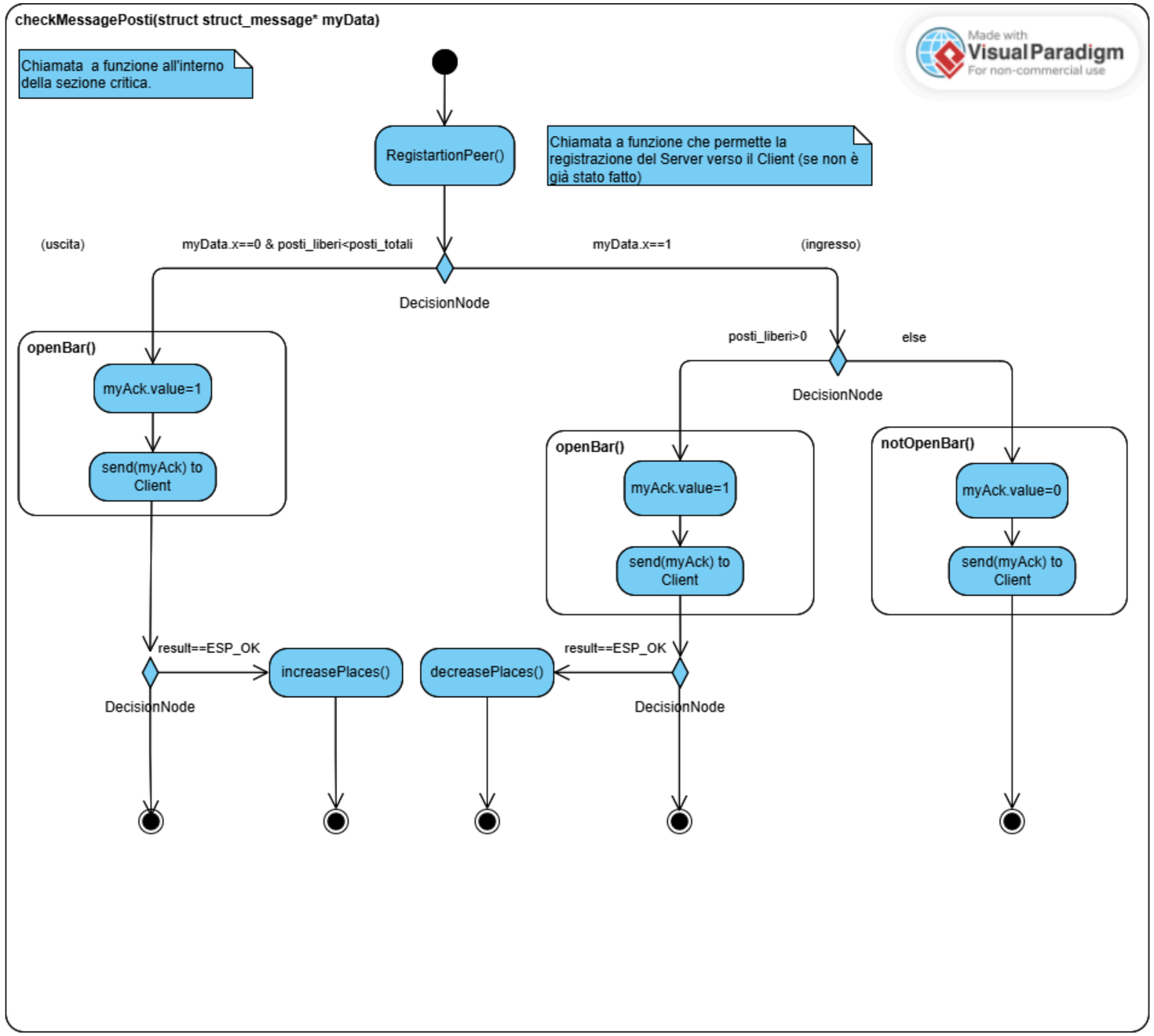
AVAILABILITY



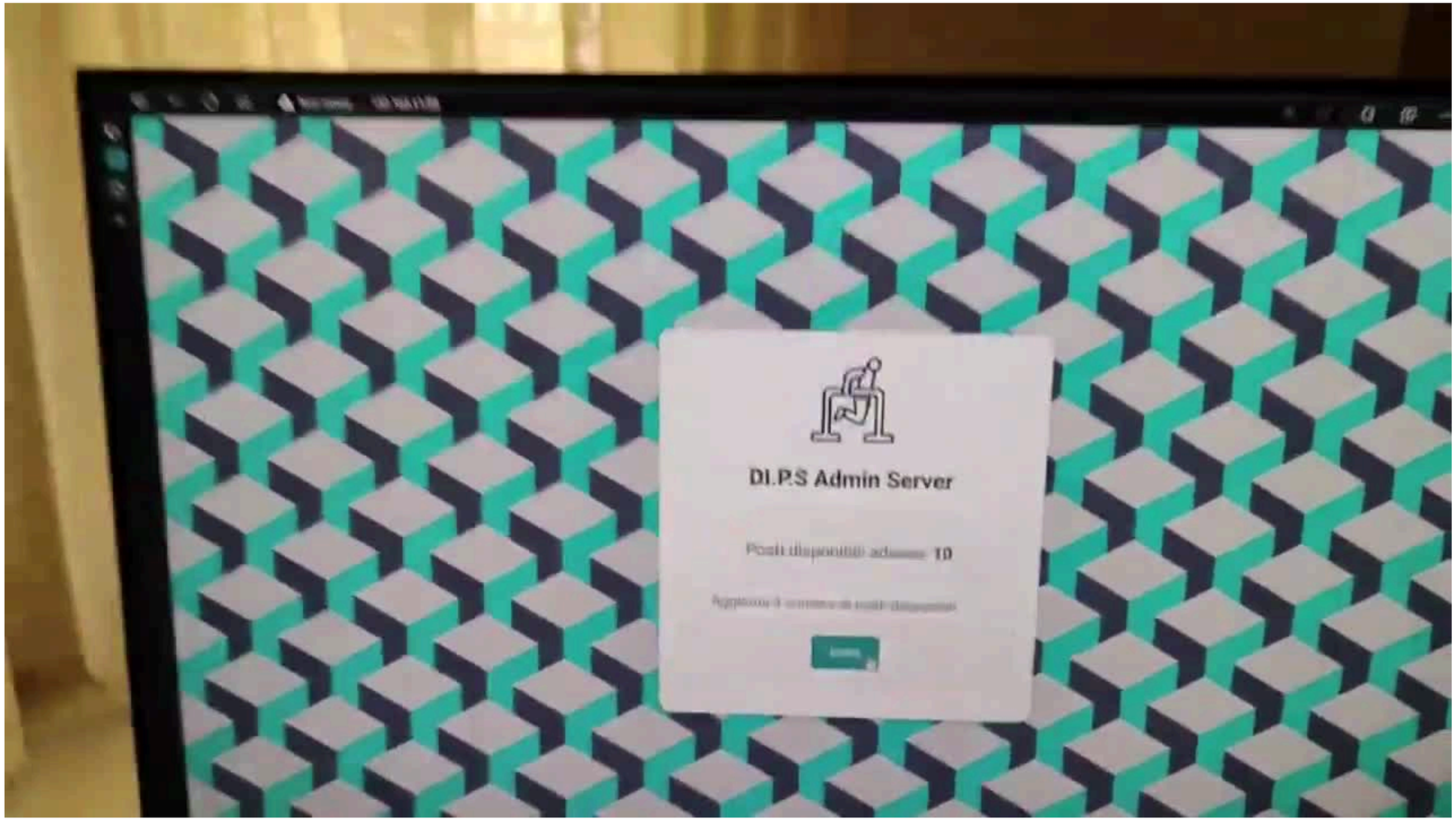
Sequence Diagram



Activity Diagram



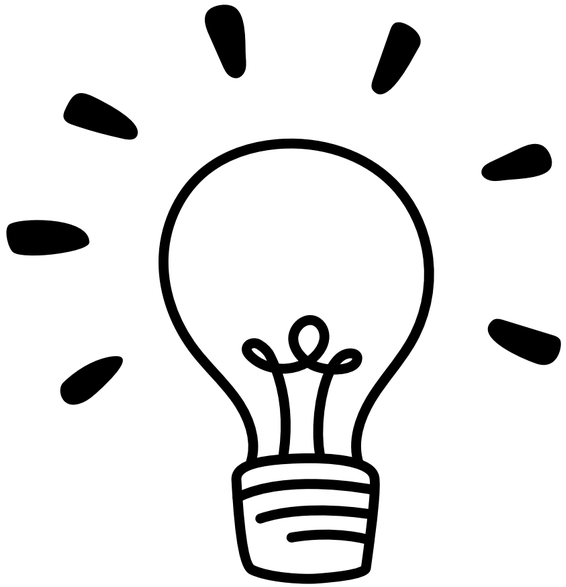
Video Demo



Issues & Possible Solutions

- **IF THE SERVER STOPS WORKING?**
  - **FAILURE RECOVERY MECHANISM**
  - **REDUNDANCY**
  - **DISTRIBUTED ELECTION ALGORITHM** (BULLY ALGORITHM)
  - **DISTRIBUTED MUTUAL EXCLUSION** (RICART-AGRAWALA)

- **IF THE CLIENTS STOP WORKING?**
  - **REDUNDANCY**



Future Extensions

- DEVELOPMENT OF NEW POSSIBLE SOLUTIONS TO MITIGATE FAILURES
- IMPROVEMENT OF ADMIN CAPABILITIES

Here the DSI approach ends...

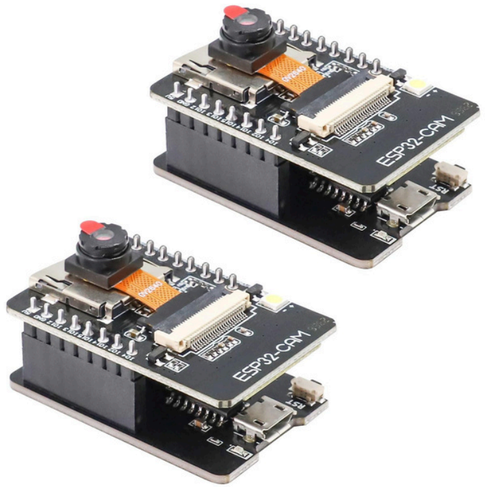
...and the WebRTC approach starts!

What are the updates?

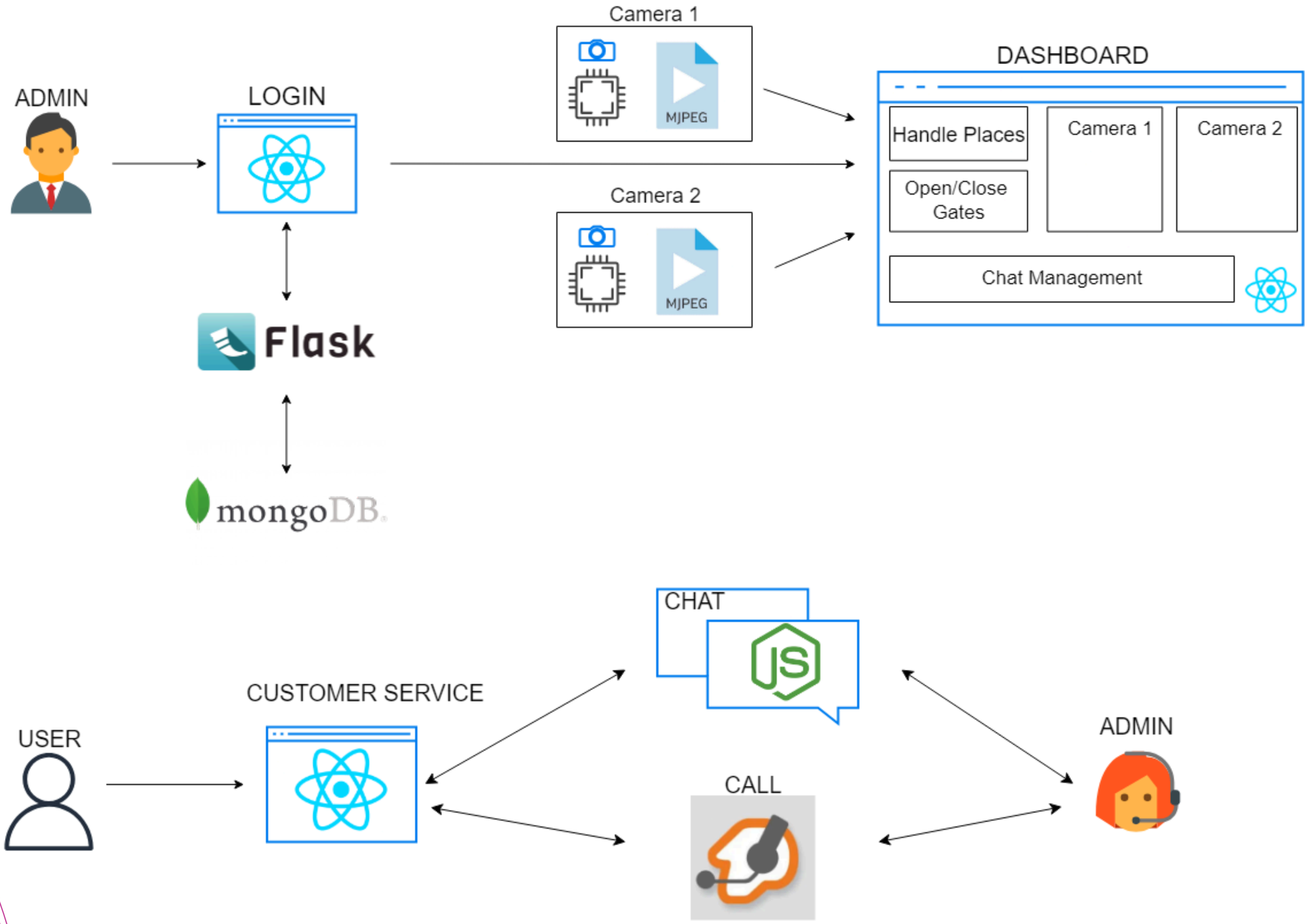
- **Improvement of the Admin interface**, in order to allow him to:
  - **supervise** the parking through two cameras
  - **adjust** the number of available places in the parking
  - **open** (**or close**) a specific gate remotely
  - **handle** the support chats for the users

- **Introduction of a Customer Service**, in which a generic user can contact an admin in order to address some issues.

Technologies



How does it work (2.0) ?



Weakness and Strenght Points

Weakness Points

- The ESP32CAM exposes the live video through MJPEG over HTTP.
  - High bandwidth usage and not scalable
- Many hosting servers.
- Amateur deployment.

Strenght Points

- *Simple* and *Functional* architecture.
- *Fully Fledged* application, easy to improve and extend.
- Good integration between components.

Thanks for Watching...
...and LET'S DIPS!