



# Fleet Monitoring System

Team Name: sddec20-12

Team Members: Joe Herrera, Lorenzo Chavarria, Nicolas De La Cruz, and Marco Yopez-Gomez

Faculty Adviser: Dr. Lotfi ben Othmane

Client: Dr. Lotfi ben Othmane

## Introduction

### Problem:

- Companies have fleets of mobile assets that need to be monitored and managed

### Solution:

- A PiCan connected to each vehicle sends data to server and then to the client
- Mobile application that includes features like client chat, and GPS location, vehicle data analyzation

## Users & Uses

### Fleet Management Users:

- Project Managers
- Project Supervisors
- Vehicle Driver

### Application Uses:

- View vehicle OBD data
- View every vehicles GPS position
- Communicate with fleet team

## Technical Details

### Client:

- Android OS developed in Java

### Server:

- NodeJS
- Microservices
- Socket.io for websocket real time communication

### Database:

- MongoDB

### Raspberry Pi:

- PiCan Model

## Standards

- Agile Development Standards
- JavaScript Coding Standards
- Google Java Style Standards
- MongoDB Schema Design Standards

## Design Requirements

### Functional Requirements

- Communicate data from a vehicle to server
- Record data into database
- Display vehicle data on a map for users
- Allow users to communicate within a messaging system

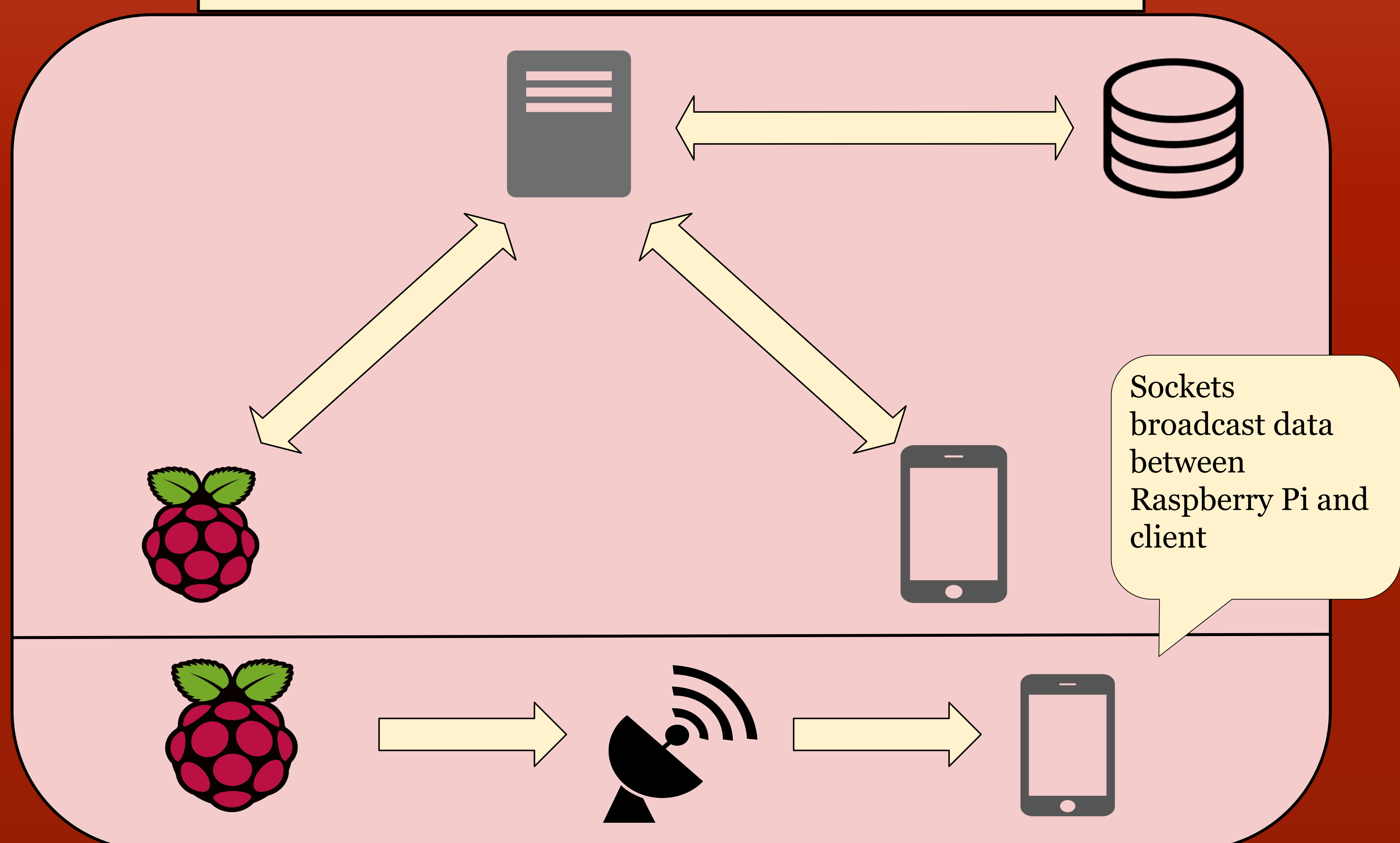
### Non-Functional Requirements

- Server should support 50 clients with response time of less than 5 sec.
- Features should be 5 sec. away from being navigated
- 99% of time, app should be running for users
- Server must be running and updated 100% of time

### Operating Environment

- Windows OS
- Android OS
- Raspberry Pi OS

## Design Approach



## Testing

### Integration Testing

- All API functionality is tested using Postman tests

### Manual Testing

- Used a vehicle simulator to communicate with Pi to generate and send vehicle data to be sent to server