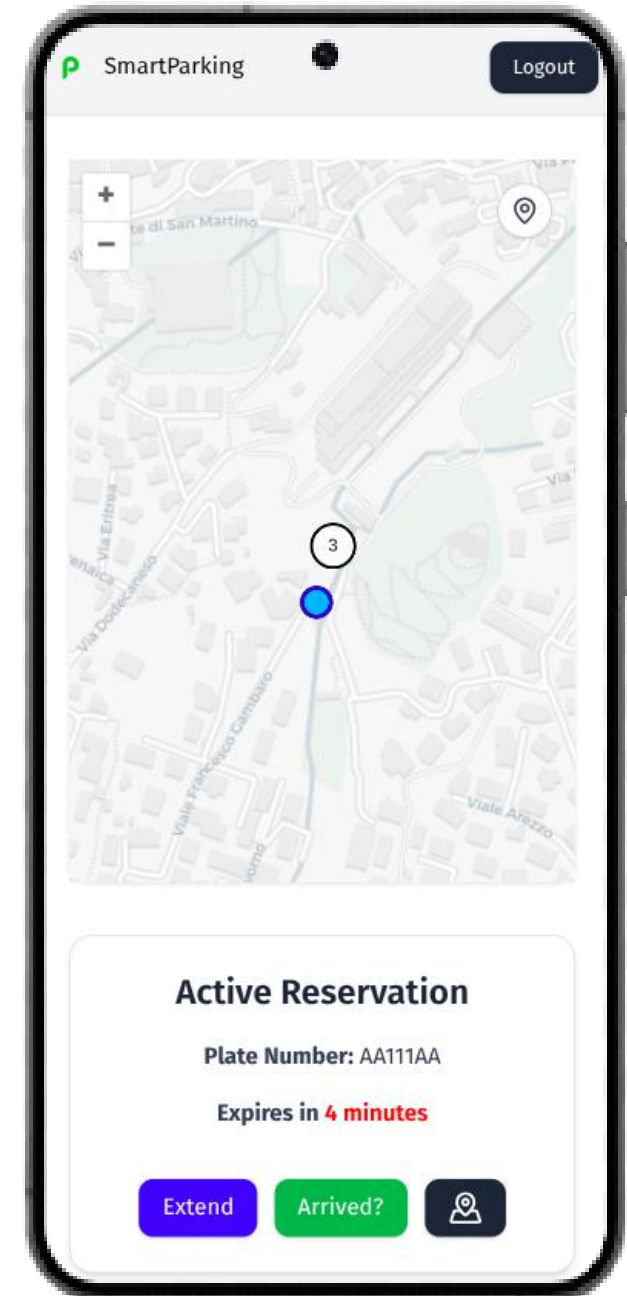# *Smart Parking*

Frattini Michele
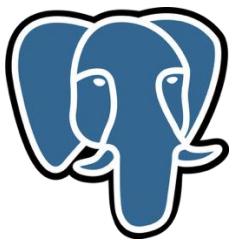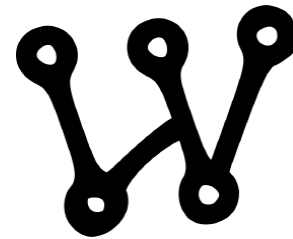Scarzella Anna
Timossi Luigi

Introduction

System for booking a parking space through a dedicated web app

Using an infrastructure composed of various devices that communicate with each other, using the MQTT protocol
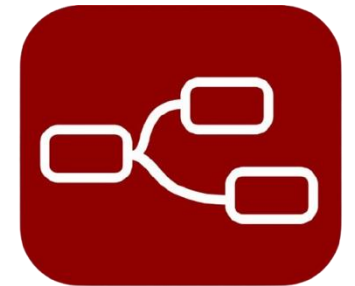
Used technologies

- Nuxt
- Shadcn
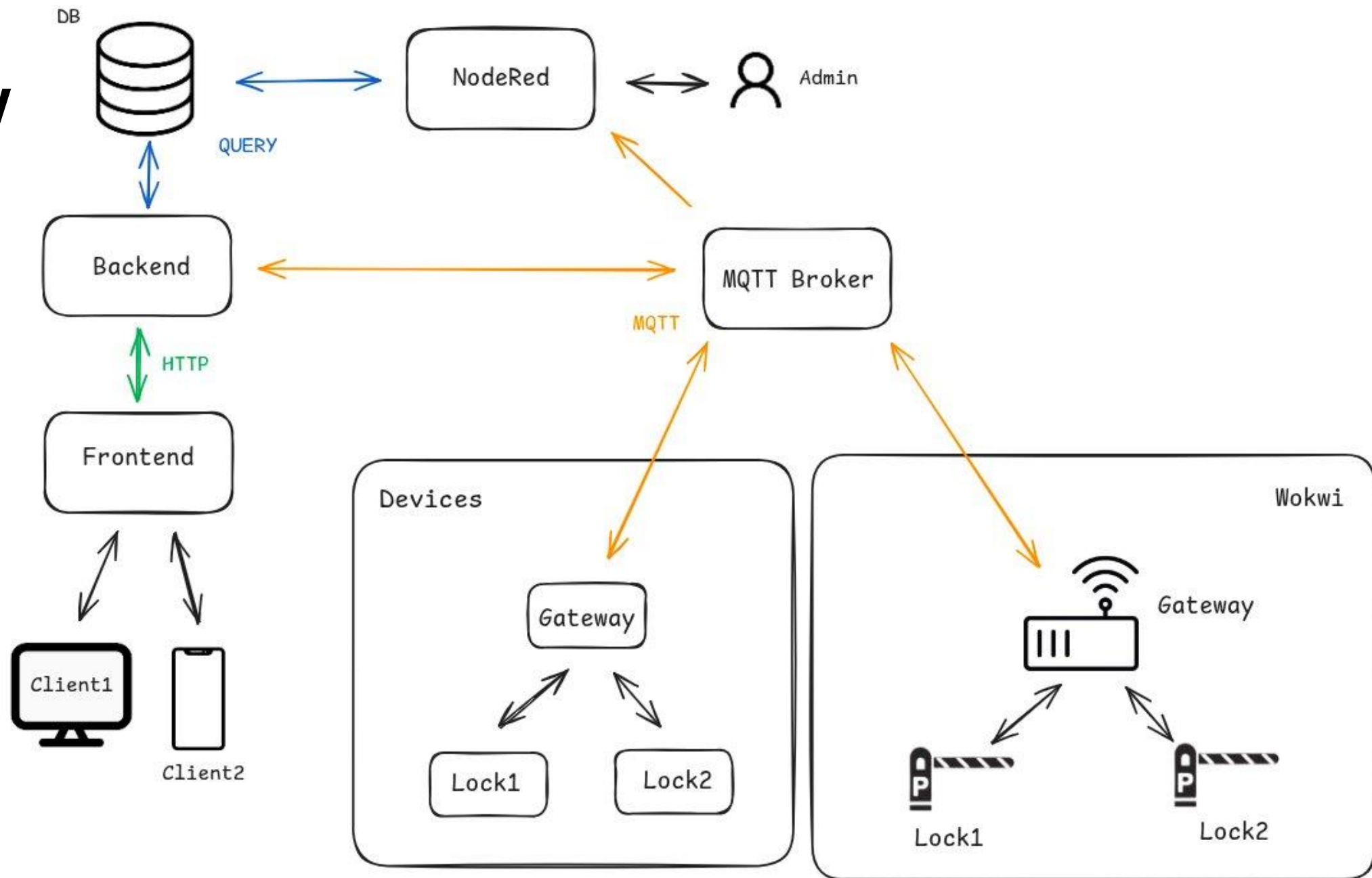- Postgres
- Docker
- NodeJS
- Wokwi
- NodeRed
- ExpressJS

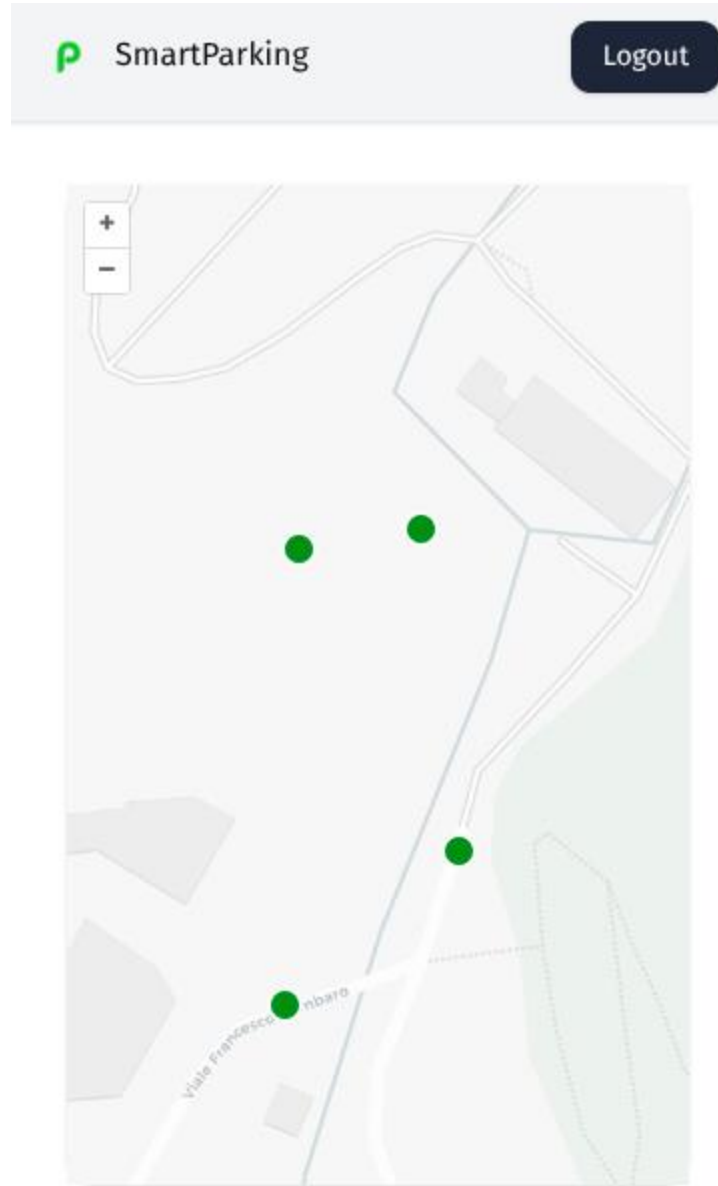# Flow

Frontend

Nuxt, 2 pages:

- index.vue: user login and registration
- map.vue: the map and its logic

Shadcn library for component graphics

Viewable from both smartphones and Desktops

Nuxt

shadcn/ui

Map.vue

- Open Layers
  - free, open source JavaScript library
- Watch lock statuses
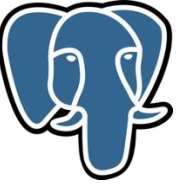- Manage your reservation

Backend

Routes and middleware:

- Used NodeJS with Express framework
- Routes are the possible API calls and are protected with control and authentication middleware
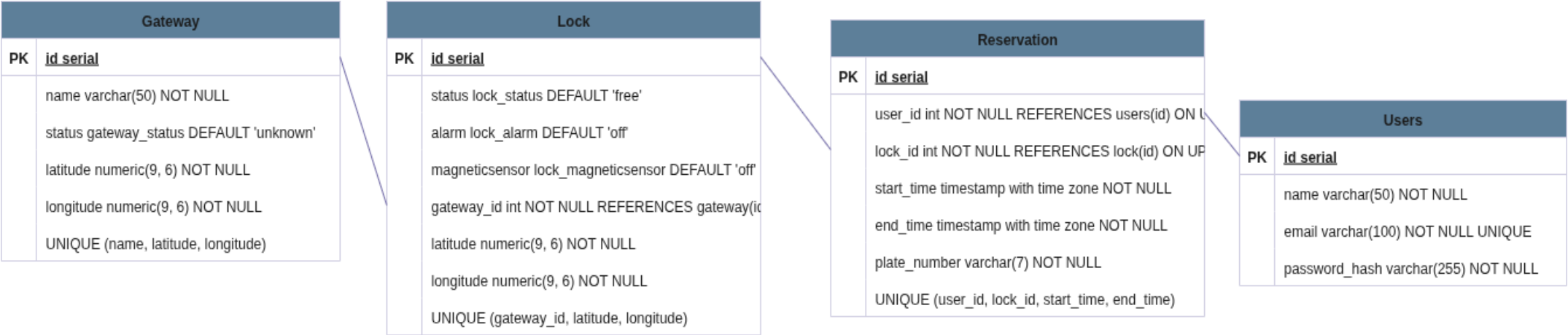
Controllers for the logic

Sequelize with models for DB connection and query

Database

- Postgres with Migration
- Custom types for statuses
- Access with user and password

Gateway	
PK	id serial
	name varchar(50) NOT NULL
	status gateway_status DEFAULT 'unknown'
	latitude numeric(9, 6) NOT NULL
	longitude numeric(9, 6) NOT NULL
	UNIQUE (name, latitude, longitude)

Lock	
PK	id serial
	status lock_status DEFAULT 'free'
	alarm lock_alarm DEFAULT 'off'
	magneticsensor lock_magneticsensor DEFAULT 'off'
	gateway_id int NOT NULL REFERENCES gateway(id
	latitude numeric(9, 6) NOT NULL
	longitude numeric(9, 6) NOT NULL
	UNIQUE (gateway_id, latitude, longitude)

Reservation	
PK	id serial
	user_id int NOT NULL REFERENCES users(id) ON U
	lock_id int NOT NULL REFERENCES lock(id) ON UP
	start_time timestamp with time zone NOT NULL
	end_time timestamp with time zone NOT NULL
	plate_number varchar(7) NOT NULL
	UNIQUE (user_id, lock_id, start_time, end_time)

Users	
PK	id serial
	name varchar(50) NOT NULL
	email varchar(100) NOT NULL UNIQUE
	password_hash varchar(255) NOT NULL

MQTT broker

Used nodeJS with aedes library

Accepts both TCP and webSocket connections

We chose to use it locally:

- To avoid frequent disconnections and excessive delays

MQTT - topics

- **/[ID_gateway]/up_link:** Used for publishing information on the changed status of the locks associated with a gateway

- **/[ID_gateway]/down_link:** Used to send commands to the locks associated with a gateway

- **/[ID_gateway]/down_link_ack:** Used to acknowledge a command sent to a gateway

- **/[ID_gateway]/heartbeat:** Used for periodically publishing heartbeats of the gateways and associated locks to monitor their status
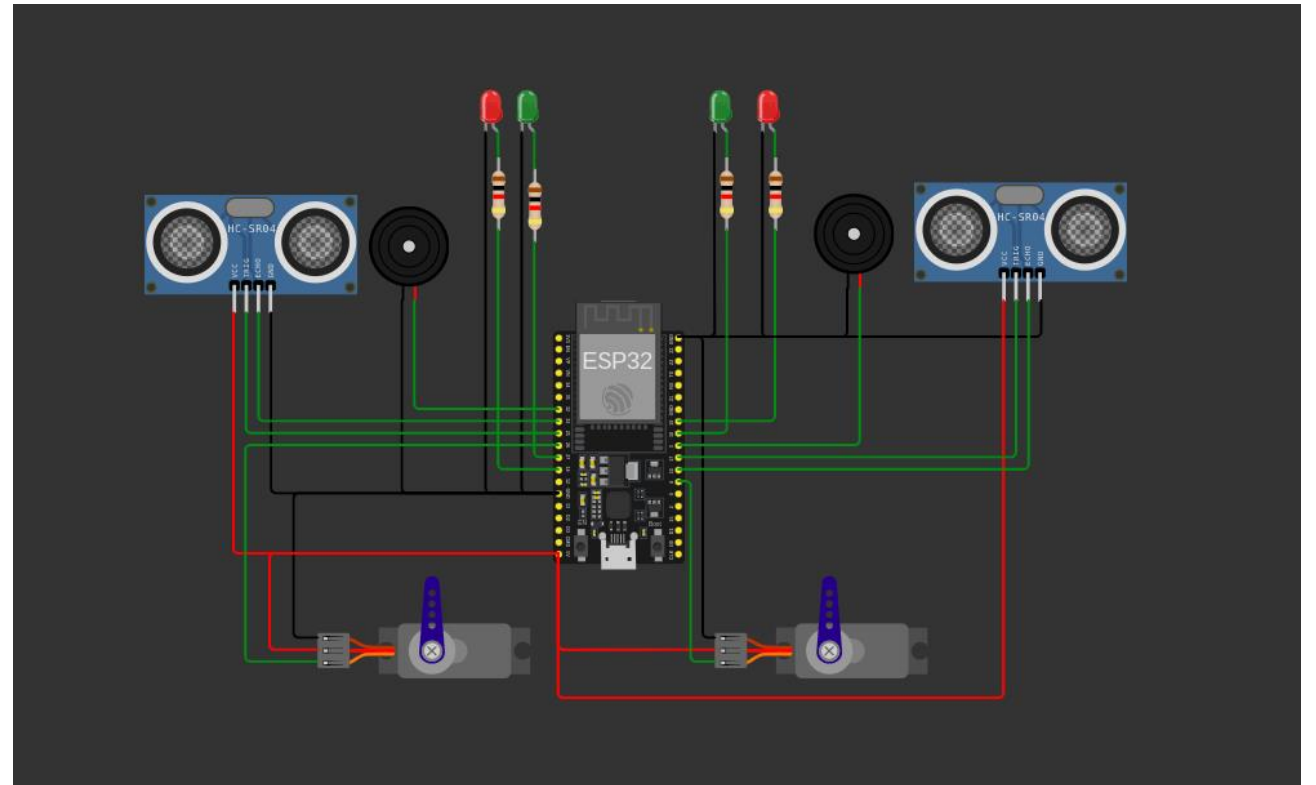
Devices

Two simulation strategies:

- NodeJS
- Wokwi

Simulations both interchangeable with each other (as behavior and functionality)

In our case we have:

- 2 locks and 2 gateways with NodeJS
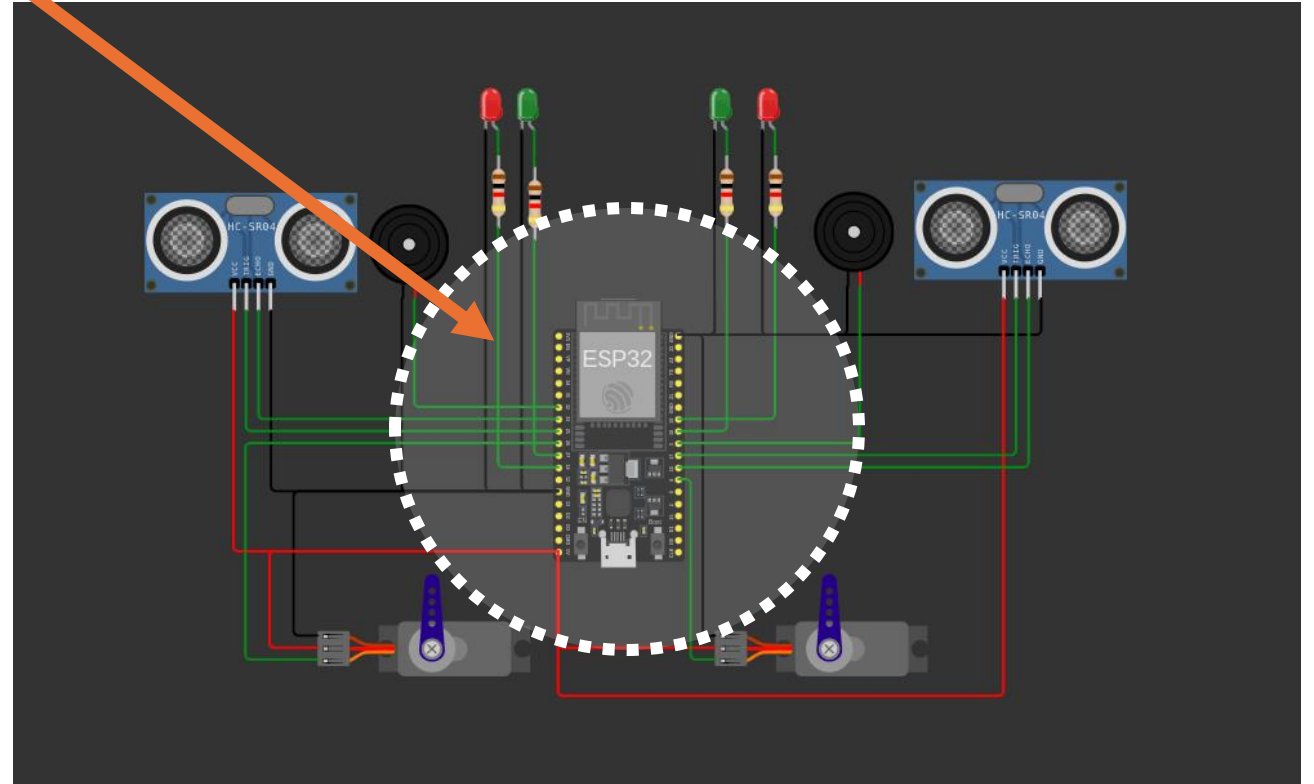- 2 locks and 1 gateway with Wokwi

Wokwi

ESP32 allows you to connect different sensors and manage them, it also has integrated Wi-Fi connection to use the MQTT protocol
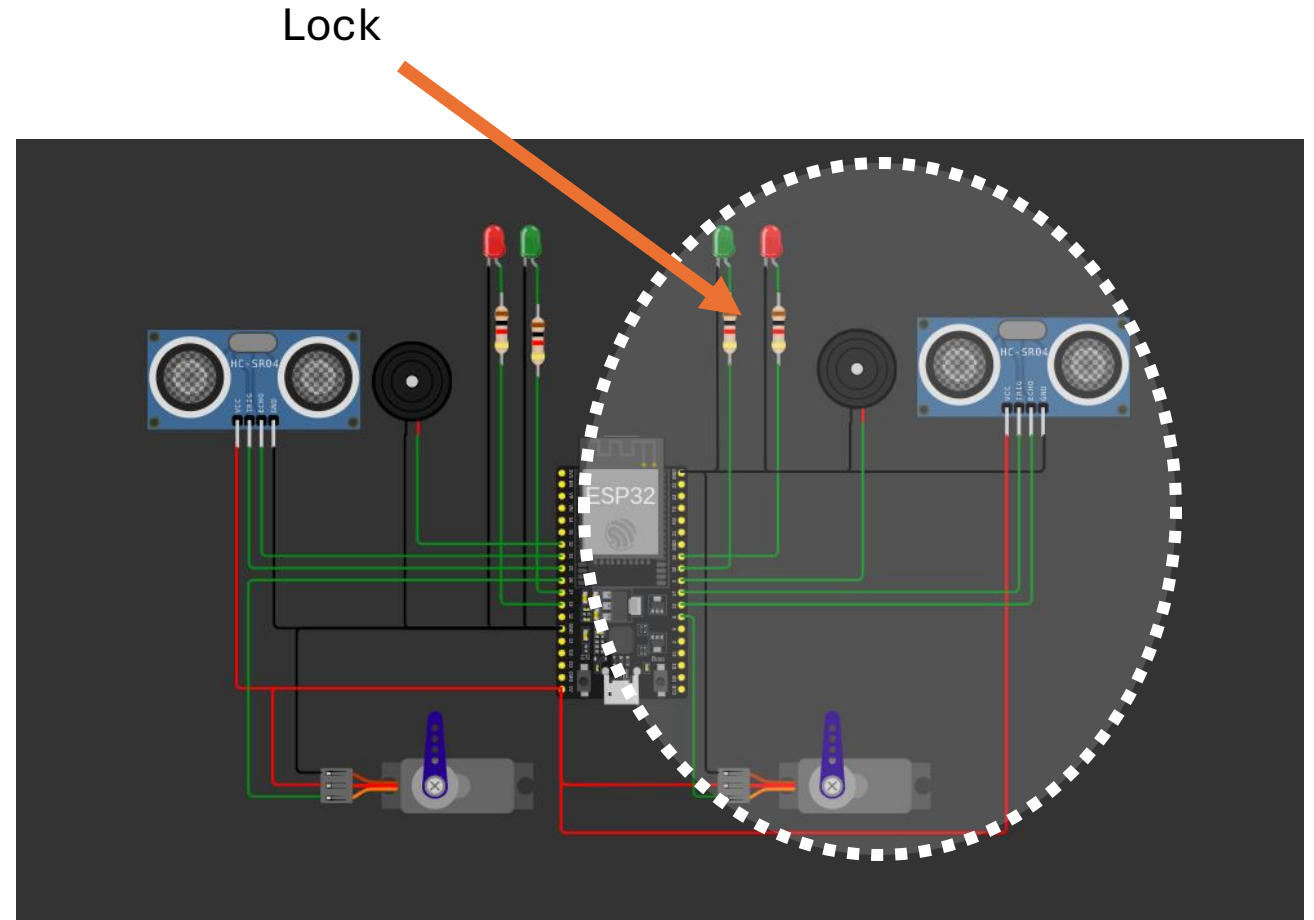
Wokwi - Gateway

Gateway

The gateway is simulated along with the LoRa links with the esp32 and its locks
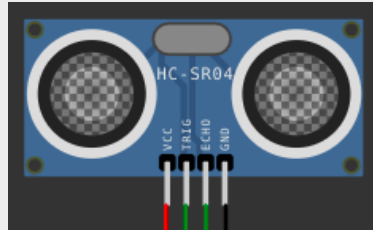
Wokwi - Lock

Lock

the locks consist of their components: two LEDs, a buzzer, an arm servo, and an ultrasonic proximity sensor

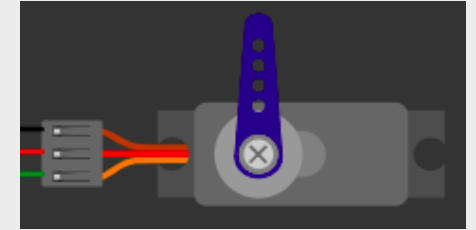# In details – Actors and sensors

Ultrasonic distance sensor:

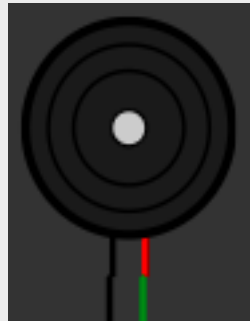The only sensor we have, useful for measuring the distance between the car and the ground

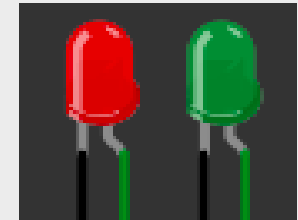## Servo:

Represents the arm of the lock
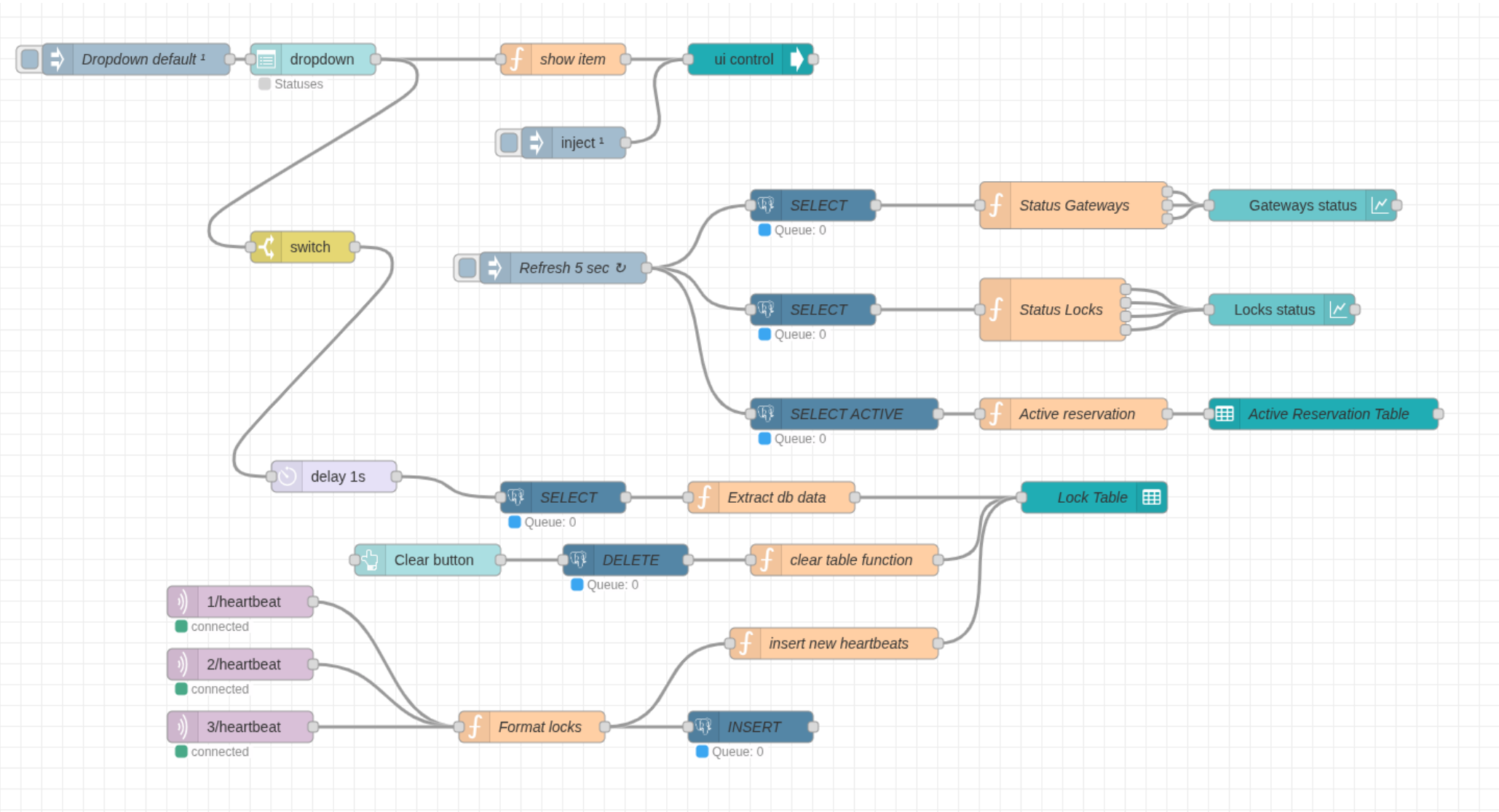

Buzzer:

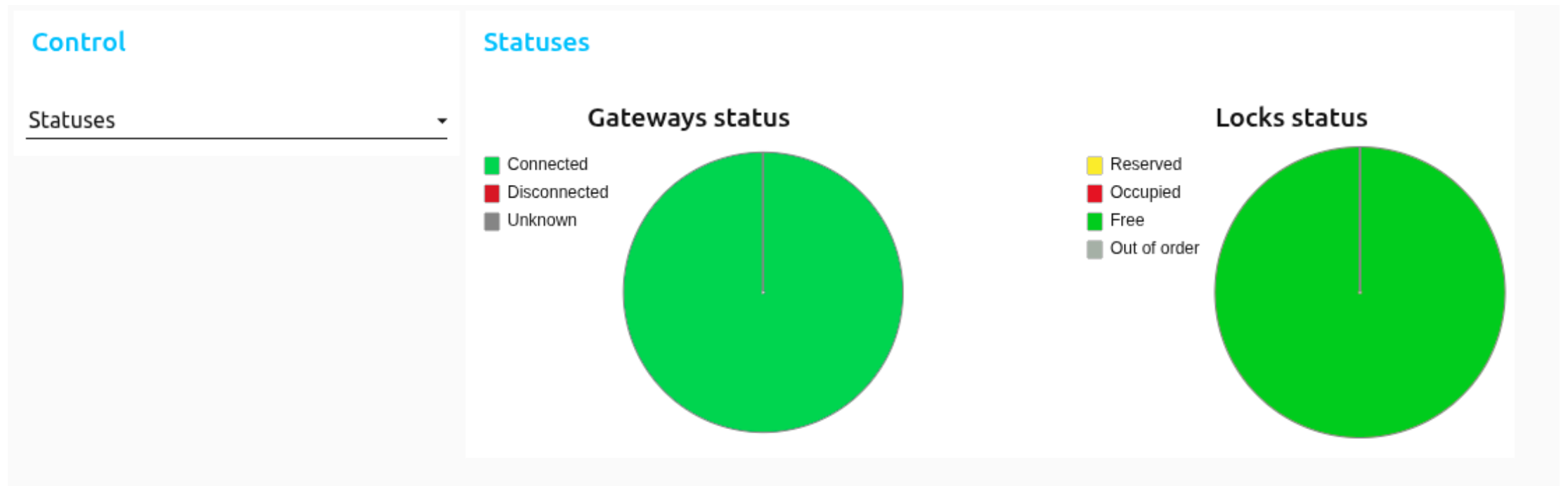Emit a sound when the lock raise up and down

Led:

Red light for occupied /reserved parking slot, green for free ones

NodeRed

NodeRed – Dashboard statuses

NodeRed – Dashboard heartbeats

Control

Heartbeats ▾

Heartbeats

Gateway ID ▲	Lock ID ▲	Status ▲	Timestamp ▲	
3	4	free	2025-07-07T14:05:57.845Z	
2	3	occupied	2025-07-07T14:06:17.943Z	
2	3	occupied	2025-07-07T14:06:37.952Z	
3	4	free	2025-07-07T14:06:57.994Z	
2	3	occupied	2025-07-07T14:07:18.093Z	
3	4	free	2025-07-07T14:07:18.094Z	
2	3	occupied	2025-07-07T14:07:38.142Z	
3	4	free	2025-07-07T14:07:58.143Z	
2	3	occupied	2025-07-07T14:08:18.241Z	
3	4	free	2025-07-07T14:08:38.292Z	
3	4	free	2025-07-07T14:08:58.291Z	
3	4	free	2025-07-07T14:09:18.391Z	
3	4	free	2025-07-07T14:09:38.443Z	
3	4	free	2025-07-07T14:09:58.442Z	
2	3	occupied	2025-07-07T14:10:18.540Z	
3	4	free	2025-07-07T14:10:38.590Z	
3	4	free	2025-07-07T14:10:58.592Z	
2	3	occupied	2025-07-07T14:11:38.740Z	
2	3	occupied	2025-07-07T14:11:58.741Z	
3	4	free	2025-07-07T14:12:18.841Z	
2	3	occupied	2025-07-07T14:12:38.885Z	

CLEAR BUTTON

NodeRed – Dashboard active reservation

User ID	Lock ID	Start Time	End Time	Plate Number
3	1	7/7/2025, 2:33:48 PM	7/7/2025, 2:58:48 PM	AA111AA
1	3	7/7/2025, 2:34:30 PM	7/7/2025, 3:04:30 PM	BB222BB
2	4	7/7/2025, 2:35:12 PM	7/7/2025, 3:40:12 PM	CC333CC

Control

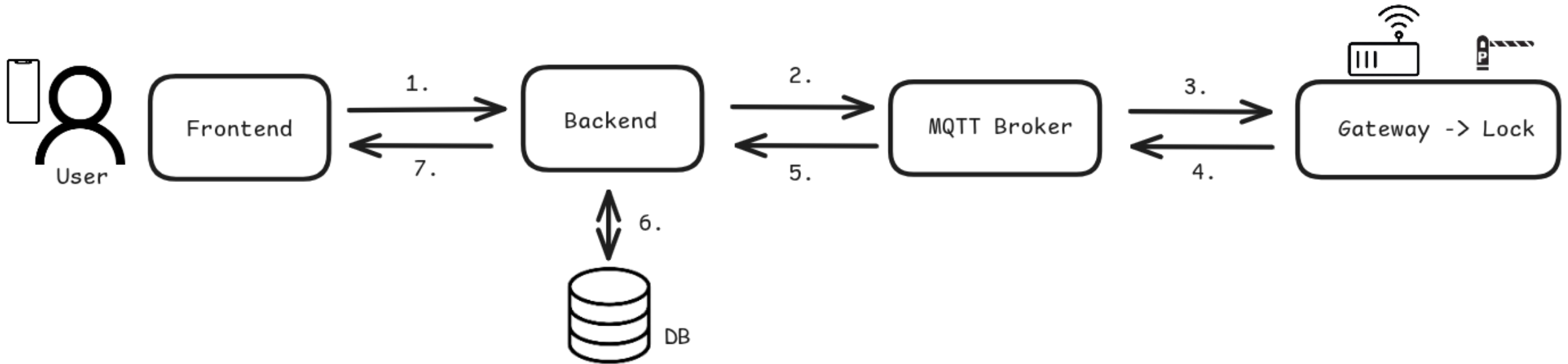ActiveReservation

ActiveReservation

DevOps

- Everything runs locally using Docker Compose for compatibility
- justfile for easy command management
- .env to keep sensitive environment variables safe
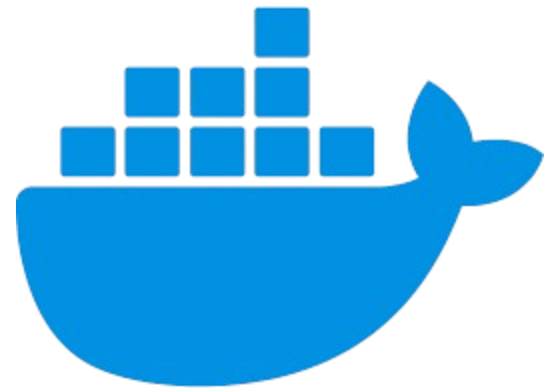
Demo flow

User want to reserve a parking slot via web app

Thanks for your attention!