RT_SLAM_AVOC

Comprehensive Project Report

Author: Abhishek Chauhan

Introduction

Auto-generated report for project located at: /home/l-100791/RT_SLAM_qorix_incubDeployment

Project Files & Analysis

.git/HEAD

ref: refs/heads/main

.git/config

```
[core]
 repositoryformatversion = 0
 filemode = true
 bare = false
 logallrefupdates = true
[remote "origin"]
 url = https://github.com/deploymentqorix/RT_SLAM_qorix_incubDeployment.git
 fetch = +refs
```

.git/description

Unnamed repository; edit this file 'description' to name the repository.

.git/index

DIRC<h¢ñ"<qzh¢ñ"<qzN—‘•¤èè×øa‹2È42êû»ÜüØg>”pÁÇ.github/workflows/npm-gulp.ymlh¢ñ"Gnh¢ñ"GnN—’•¤èè0ÅÀjY€iH•ÜjÑ±}ë&tEâ'LICENSEh¢ñ"Gnh¢ñ"GnN—“•¤èèë°"']5)ÅôW‡õÅ¤¤ëý3aõMakefileh¢ñ"Gnh¢ñ"GnN—••¤èè.G!f•[“rHÍqòž

.git/info/exclude

git ls-files --others --exclude-from=.git/info/exclude
Lines that start with '#' are comments. For a project mostly in C, the following would be a good set of exclude patterns (uncomment them if you w

Module/Top comments:

git ls-files --others --exclude-from=.git/info/exclude
Lines that start with '#' are comments.
For a project mostly in C, the following would be a good set of
exclude patterns (uncomment them if you want to use them):
*.[oa]
*~

.git/logs/HEAD

0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from

https://github.com/deploymentqorix/RT_SLAM_qorix_incu

.git/logs/refs/heads/main

0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from https://github.com/deploymentqorix/RT_SLAM_qorix_incu

.git/logs/refs/remotes/origin/HEAD

0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from https://github.com/deploymentqorix/RT_SLAM_qorix_incu

.git/packed-refs

pack-refs with: peeled fully-peeled sorted

Module/Top comments:

pack-refs with: peeled fully-peeled sorted

.git/refs/heads/main

533a0d96e0beff3ef5f5aa66a021ef808b9d93ab

.git/refs/remotes/origin/HEAD

ref: refs/remotes/origin/main

LICENSE

MIT License

Copyright (c) 2025 deploymentqorix

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to d

Makefile

The main Makefile for the entire project

Module/Top comments:

The main Makefile for the entire project

backend/api_server/main.py

```
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
```

allow_credentials=True,
allow_

Functions / Methods:

- get_system_health()

backend/data_streamer/main.py

```
import rclpy
from rclpy.node import Node
from nav_msgs.msg import Odometry
from nav_msgs.msg import OccupancyGrid
import asyncio
import websockets
import json

CONNECTED_CLIENTS = set()

async def reg
```

Classes:

- DataStreamerNode:

Functions / Methods:

- __init__(self)
- odom_callback(self, msg)
- map_callback(self, msg)

backend/data_streamer/vehicle_pose.py

```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import PoseStamped

class PoseSubscriber(Node):
    def __init__(self):
        super().__init__('pose_subscriber_for_streamer')
```

Classes:

- PoseSubscriber:

Functions / Methods:

- __init__(self)
- listener_callback(self, msg)

This function is called every time a message is received.

- main(args=None)

backend/dev_scripts/mock_backend.py

File: backend/dev_scripts/mock_api.py

Description: A standalone FastAPI server to provide mock data for frontend development. To run: python3 mock_api.py

Module/Top comments:

File: backend/dev_scripts/mock_api.py
Description: A standalone FastAPI server to provide mock data for frontend development.
To run: python3 mock_api.py

Functions / Methods:

- get_system_health()

This endpoint simulates the real /system/health endpoint.
    It returns a hardcoded, fake system health status.

- reset_slam()

This endpoint simulates receiving a reset command from the UI.

backend/install/_local_setup_util_ps1.py

Copyright 2016-2019 Dirk Thomas
Licensed under the Apache License, Version 2.0

Module/Top comments:

Copyright 2016-2019 Dirk Thomas
Licensed under the Apache License, Version 2.0

Functions / Methods:

- main(argv=sys.argv[1:])

- get_packages(prefix_path, merged_install)

- add_package_runtime_dependencies(path, packages)

Check the path and if it exists extract the packages runtime dependencies.

    :param Path path: The resource file containing the runtime dependencies
    :param dict packages: A mapping from package names to the sets of runtime
      dependencies to add to

- order_packages(packages)

Order packages topologically.

    :param dict packages: A mapping from package name to the set of runtime
      dependencies
    :returns: The package names
    :rtype: list

- reduce_cycle_set(packages)

Reduce the set of packages to the ones part of the circular dependency.

    :param dict packages: A mapping from package name to the set of runtime
      dependencies which is modified in place

- _include_comments()

- get_commands(pkg_name, prefix, primary_extension, additional_extension)
- handle_dsv_types_except_source(type_, remainder, prefix)
- _append_unique_value(name, value)
- _prepend_unique_value(name, value)
- _remove_ending_separators()
- _set(name, value)
- _set_if_unset(name, value)

backend/install/_local_setup_util_sh.py

Copyright 2016-2019 Dirk Thomas
Licensed under the Apache License, Version 2.0

Module/Top comments:

Copyright 2016-2019 Dirk Thomas
Licensed under the Apache License, Version 2.0

Functions / Methods:

- main(argv=sys.argv[1:])
- get_packages(prefix_path, merged_install)
- add_package_runtime_dependencies(path, packages)

Check the path and if it exists extract the packages runtime dependencies.

:param Path path: The resource file containing the runtime dependencies
:param dict packages: A mapping from package names to the sets of runtime dependencies to add to

- order_packages(packages)

Order packages topologically.

:param dict packages: A mapping from package name to the set of runtime dependencies
:returns: The package names
:rtype: list

- reduce_cycle_set(packages)

Reduce the set of packages to the ones part of the circular dependency.

:param dict packages: A mapping from package name to the set of runtime dependencies which is modified in place

- _include_comments()
- get_commands(pkg_name, prefix, primary_extension, additional_extension)
- handle_dsv_types_except_source(type_, remainder, prefix)
- _append_unique_value(name, value)
- _prepend_unique_value(name, value)

- _remove_ending_separators()
- _set(name, value)
- _set_if_unset(name, value)

backend/install/local_setup.ps1

generated from colcon_powershell/shell/template/prefix.ps1.em
This script extends the environment with all packages contained in this
prefix path. check environment variable for custom Python executab

Module/Top comments:

generated from colcon_powershell/shell/template/prefix.ps1.em
This script extends the environment with all packages contained in this
prefix path.
check environment variable for custom Python executable

Functions / Methods:

- if ($env:COLCON_PYTHON_EXECUTABLE) {
- if (!(Test-Path "$env:COLCON_PYTHON_EXECUTABLE" -PathType Leaf)) {
- if (!(Test-Path "$_colcon_python_executable" -PathType Leaf)) {
- if (!(Get-Command "python3" -ErrorAction SilentlyContinue)) {
- if (Test-Path $_colcon_prefix_powershell_source_script_param) {
- if ($env:COLCON_TRACE) {
- if ($env:COLCON_TRACE) {
- if ($_colcon_ordered_commands) {

backend/install/local_setup.sh

generated from colcon_core/shell/template/prefix.sh.em
This script extends the environment with all packages contained in this
prefix path. since a plain shell script can't determine its own path when

Module/Top comments:

generated from colcon_core/shell/template/prefix.sh.em
This script extends the environment with all packages contained in this
prefix path.
since a plain shell script can't determine its own path when being sourced
either use the provided COLCON_CURRENT_PREFIX
or fall back to the build time prefix (if it exists)

Functions / Methods:

- _colcon_prefix_sh_prepend_unique_value() {
- _colcon_prefix_sh_source_script() {

backend/install/ros2_nodes/share/ros2_nodes/package.ps1

generated from colcon_powershell/shell/template/package.ps1.em
function to append a value to a variable

which uses colons as separators
duplicates as well as leading separators are avoided
first argum

Module/Top comments:

generated from colcon_powershell/shell/template/package.ps1.em
function to append a value to a variable
which uses colons as separators
duplicates as well as leading separators are avoided
first argument: the name of the result variable
second argument: the value to be prepended

Functions / Methods:

- if (Test-Path Env:$_listname) {
- if ($_values) {
- if ($_) {
- if ($_ -eq $_value) {
- if ($_all_values) {
- if (!$_duplicate) {
- if ($_all_values) {
- if (Test-Path Env:$_listname) {
- if ($_values) {
- if ($_) {
- if ($_ -ne $_value) {
- if (Test-Path $_colcon_package_source_powershell_script) {
- if ($env:COLCON_TRACE) {

backend/install/ros2_nodes/share/ros2_nodes/package.sh

generated from colcon_core/shell/template/package.sh.em
This script extends the environment for this package. function to prepend a value to a variable
which uses colons as separators
duplicates as we

Module/Top comments:

generated from colcon_core/shell/template/package.sh.em
This script extends the environment for this package.
function to prepend a value to a variable
which uses colons as separators
duplicates as well as trailing separators are avoided
first argument: the name of the result variable
second argument: the value to be prepended

Functions / Methods:

- _colcon_prepend_unique_value() {

backend/install/setup.ps1

generated from colcon_powershell/shell/template/prefix_chain.ps1.em
This script extends the environment with the environment of other prefix
paths which were sourced when this file was generated as we

Module/Top comments:

generated from colcon_powershell/shell/template/prefix_chain.ps1.em
This script extends the environment with the environment of other prefix
paths which were sourced when this file was generated as well as all packages
contained in this prefix path.
function to source another script with conditional trace output
first argument: the path of the script

Functions / Methods:

- if (Test-Path $_colcon_prefix_chain_powershell_source_script_param) {
- if ($env:COLCON_TRACE) {

backend/install/setup.sh

generated from colcon_core/shell/template/prefix_chain.sh.em
This script extends the environment with the environment of other prefix
paths which were sourced when this file was generated as well as a

Module/Top comments:

generated from colcon_core/shell/template/prefix_chain.sh.em
This script extends the environment with the environment of other prefix
paths which were sourced when this file was generated as well as all packages
contained in this prefix path.
since a plain shell script can't determine its own path when being sourced
either use the provided COLCON_CURRENT_PREFIX
or fall back to the build time prefix (if it exists)

Functions / Methods:

- _colcon_prefix_chain_sh_source_script() {

backend/requirements.txt

Python packages for the backend services

Module/Top comments:

Python packages for the backend services

backend/ros2_nodes/ros2_nodes/slam_node/CMakeLists.txt

cmake_minimum_required(VERSION 3.8)
project(slam_node)

# Find all dependencies
find_package(ament_cmake REQUIRED)
find_package(rclcpp REQUIRED)

find_package(geometry_msgs REQUIRED)
find_package(senso

backend/ros2_nodes/ros2_nodes/slam_node/src/slam_logic.cpp

include <Eigen/Dense>
include <vector>
include "rclcpp/rclcpp.hpp"
include "geometry_msgs/msg/pose_stamped.hpp"
include "sensor_msgs/msg/imu.hpp"
include "sensor_msgs/msg/laser_scan.hpp"
include "nav_

Module/Top comments:

include <Eigen/Dense>
include <vector>
include "rclcpp/rclcpp.hpp"
include "geometry_msgs/msg/pose_stamped.hpp"
include "sensor_msgs/msg/imu.hpp"
include "sensor_msgs/msg/laser_scan.hpp"
include "nav_msgs/msg/occupancy_grid.hpp"
include "nav_msgs/msg/odometry.hpp"
include "tf2_ros/transform_broadcaster.h"
include "tf2/LinearMath/Quaternion.h"

Functions / Methods:

- SlamNode() : Node("slam_node") {
- for (size_t i = 0; i < msg->ranges.size(); ++i) {
- if (map_x >= 0 && map_x < map_width_ && map_y >= 0 && map_y < map_height_) {
- void imu_callback(const sensor_msgs::msg::Imu::SharedPtr msg) {
- void scan_callback(const sensor_msgs::msg::LaserScan::SharedPtr msg) {
- void publish_odometry() {
- int main(int argc, char * argv[]) {

backend/startup.sh

!/bin/bash

Module/Top comments:

!/bin/bash

docs/rest_api.md

REST API Endpoints
Get System Health

Module/Top comments:

REST API Endpoints
Get System Health

docs/websocket_api.md

Message for Live Pose Updates

Module/Top comments:

Message for Live Pose Updates

frontend/Dockerfile

Stage 1: Build the React application

Module/Top comments:

Stage 1: Build the React application

frontend/README.md

Getting Started with Create React App

Module/Top comments:

Getting Started with Create React App

frontend/public/robots.txt

https://www.robotstxt.org/robotstxt.html

Module/Top comments:

https://www.robotstxt.org/robotstxt.html

frontend/src/App.js

import React, { useState, useEffect } from 'react';
import { connectWebSocket, disconnectWebSocket } from './services/websocketService';
import MapView from './components/MapView';
import SystemHealth

Functions / Methods:

- function App() {

- if (data.type === 'pose_update') {

- function App() {

frontend/src/App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
```

frontend/src/components/MapView.js

```
import React, { useRef, useEffect } from 'react';

const MapView = ({ pose, trail, mapData }) => {
  const canvasRef = useRef(null);
```

```
  useEffect(() => {
    const canvas = canvasRef.current;
    cons
```

Functions / Methods:

- if (mapData) {
- for (let i = 0; i < grid_data.length; i++) {
- if (value === 100) {
- if (trail.length > 1) {
- for (let i = 1; i < trail.length; i++) {

frontend/src/components/SystemHealthPanel.js

```
import React, { useState, useEffect } from 'react';
import { fetchSystemHealth } from '../services/apiService';

const SystemHealthPanel = () => {
  const [health, setHealth] = useState({
    cpu_usag
```

frontend/src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(do
```

frontend/src/reportWebVitals.js

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
      getCLS(onPerfEnt
```

Functions / Methods:

- if (onPerfEntry && onPerfEntry instanceof Function) {

frontend/src/services/apiService.js

```
const API_URL = "http://localhost:8000";
const MOCK_MODE = true; // Uses mock data from websocketService

export const fetchSystemHealth = async () => {
  if (MOCK_MODE) {
    // --- MOCK MODE: Return
```

Functions / Methods:

- if (MOCK_MODE) {

- if (!response.ok) {

frontend/src/services/websocketService.js

```
const MOCK_MODE = false;
const WEBSOCKET_URL = "ws://localhost:3000/ws";

let socket = null;
let mockInterval = null;

export const connectWebSocket = (onMessageCallback) => {
  if (MOCK_MODE) {
    c
```

Functions / Methods:

- if (MOCK_MODE) {

- if (MOCK_MODE) {

frontend/src/setupTests.js

jest-dom adds custom jest matchers for asserting on DOM nodes. allows you to do things like:
expect(element).toHaveTextContent(/react/i)
learn more: https://github.com/testing-library/jest-dom

Module/Top comments:

jest-dom adds custom jest matchers for asserting on DOM nodes.
allows you to do things like:
expect(element).toHaveTextContent(/react/i)
learn more: https://github.com/testing-library/jest-dom

rest_api.md

REST API Endpoints
Get System Health

Module/Top comments:

REST API Endpoints
Get System Health

websocket_api.md

Message for Live Pose Updates

Module/Top comments:

Message for Live Pose Updates

Design Diagrams

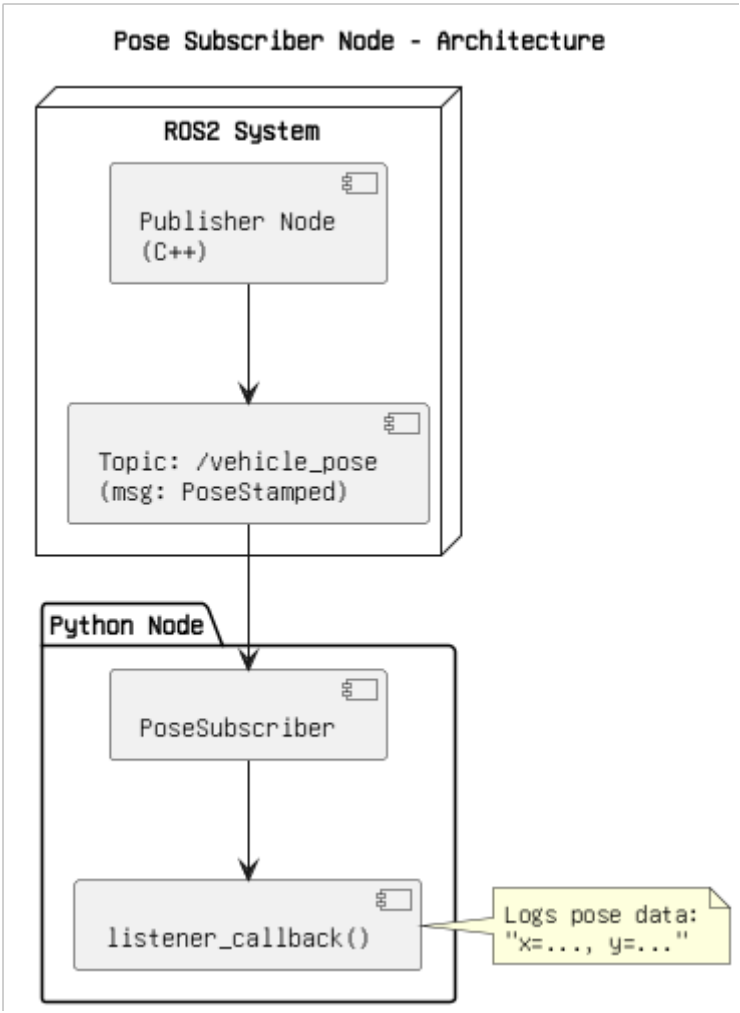

Figure: 1. Architecture.png

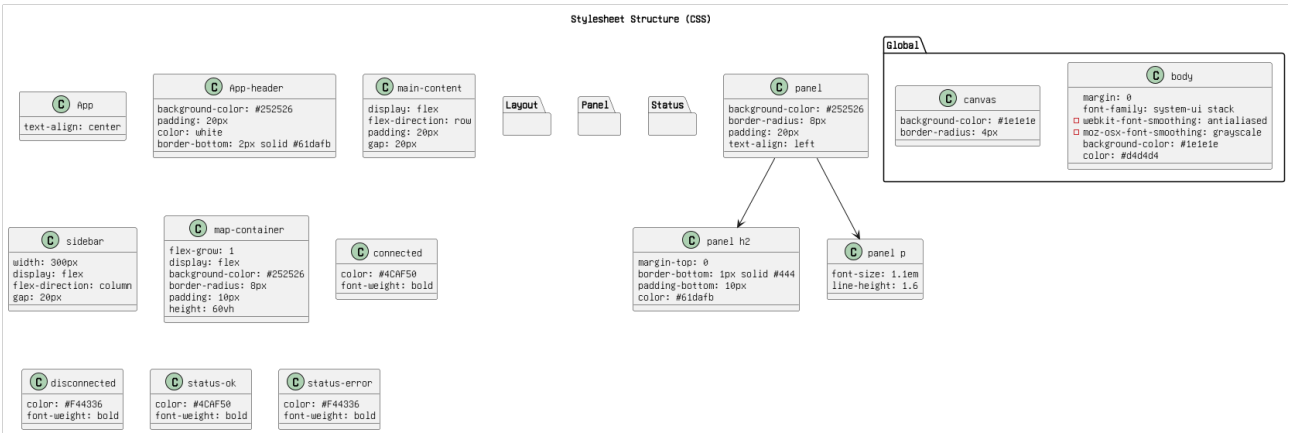

Figure: 2. CSS.png

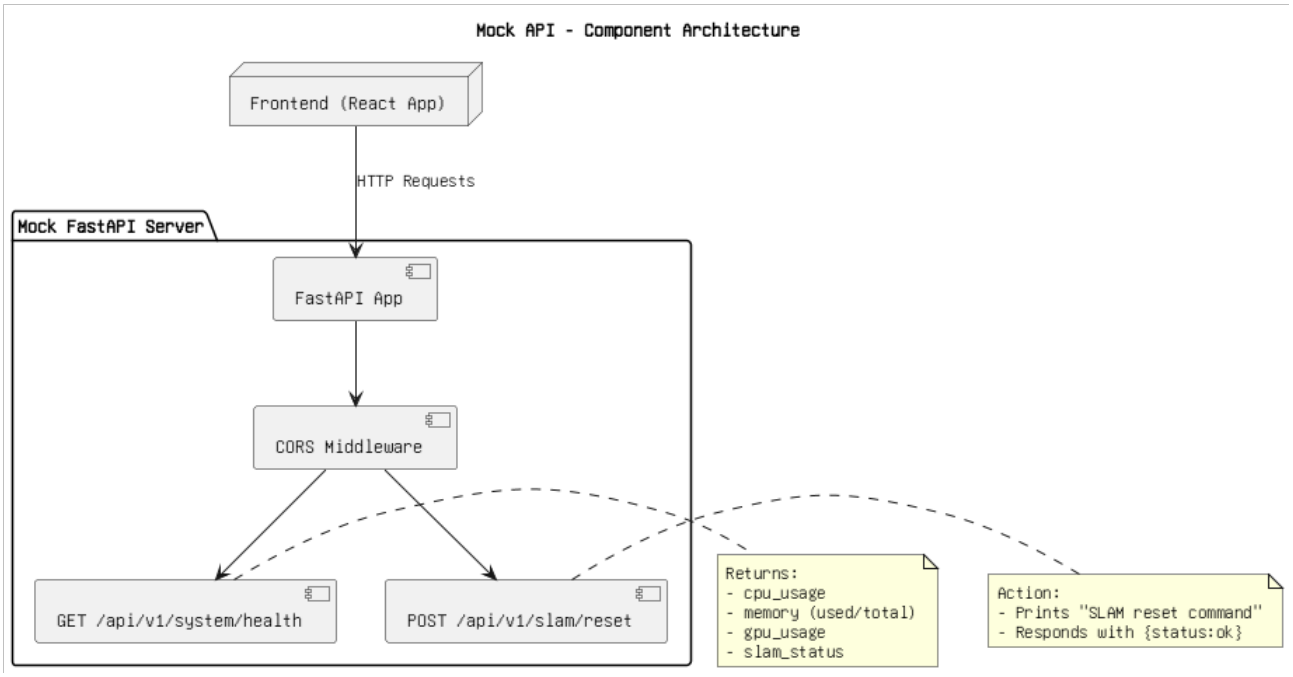

Figure: 3. Component architecture.png

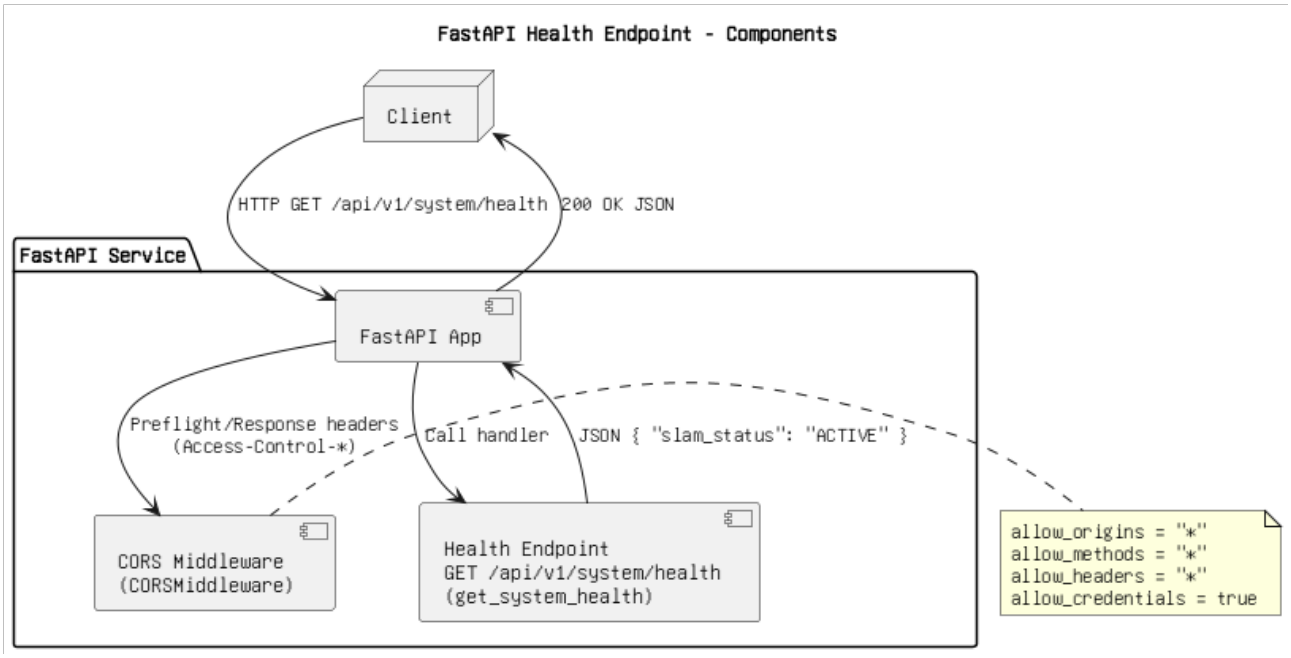

Figure: 4. Components.drawio.png

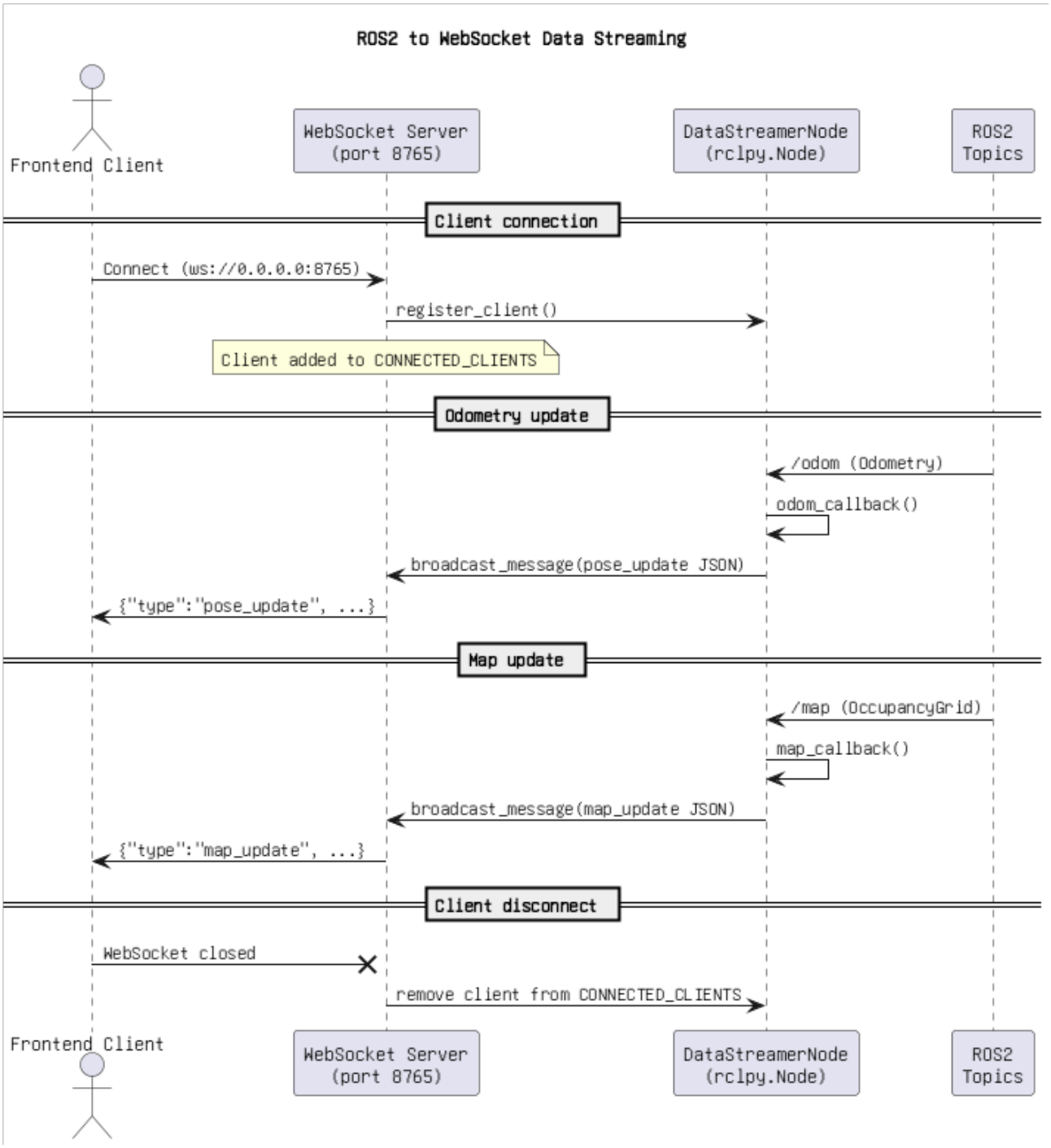

Figure: 5. Data Streaming.png

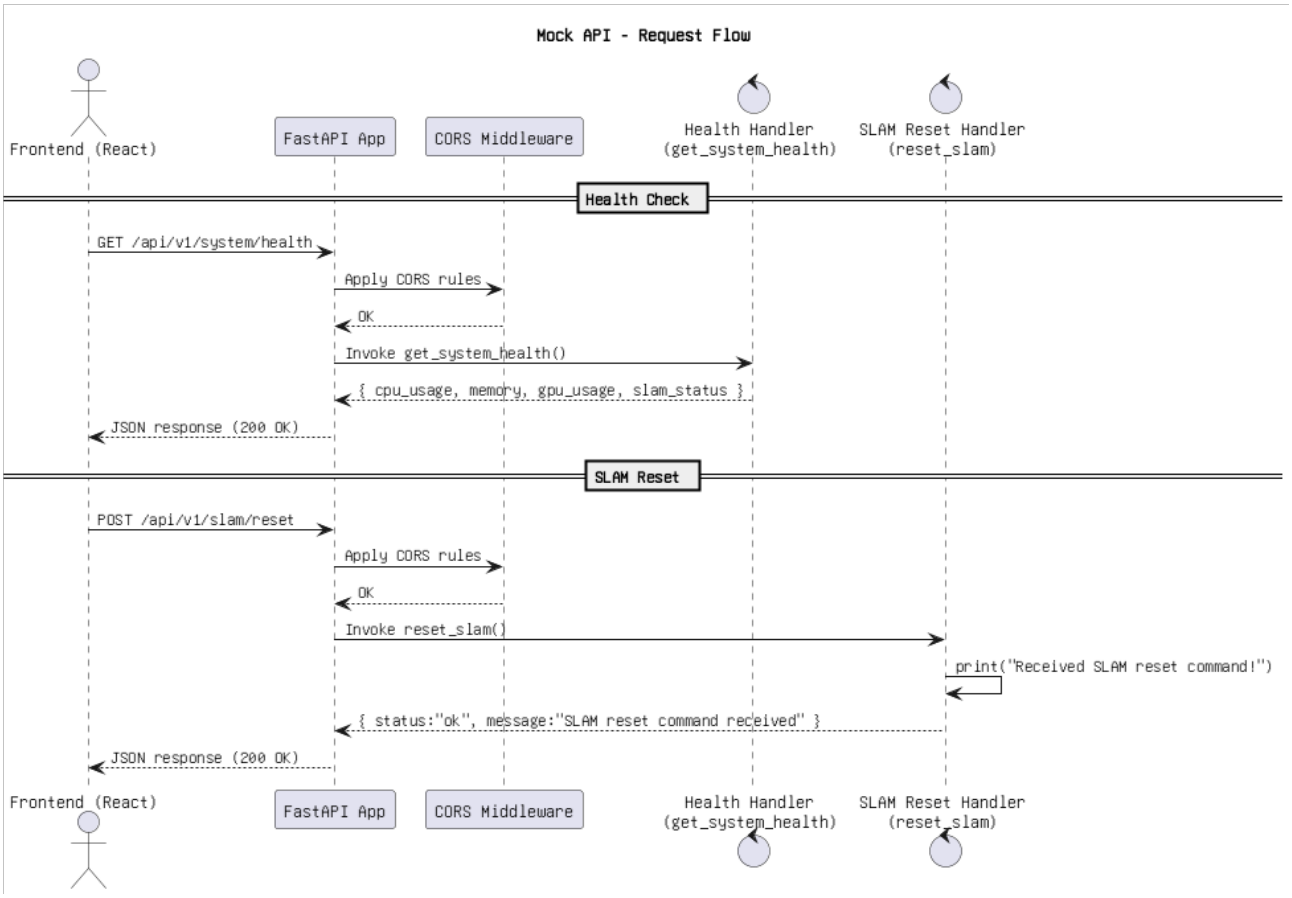

Figure: 6. Mock api request flow.png

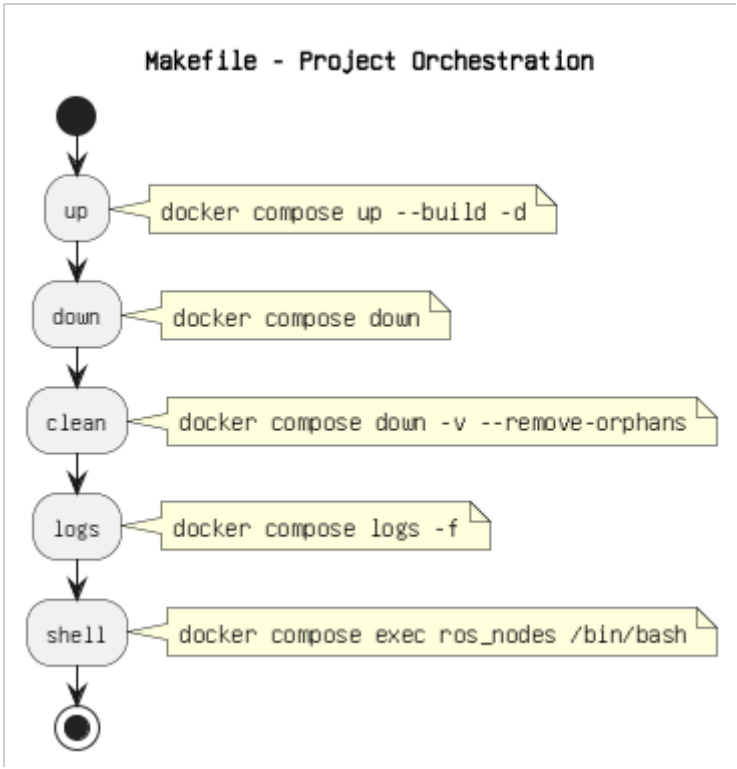

Figure: 7. Project Orchestration.png

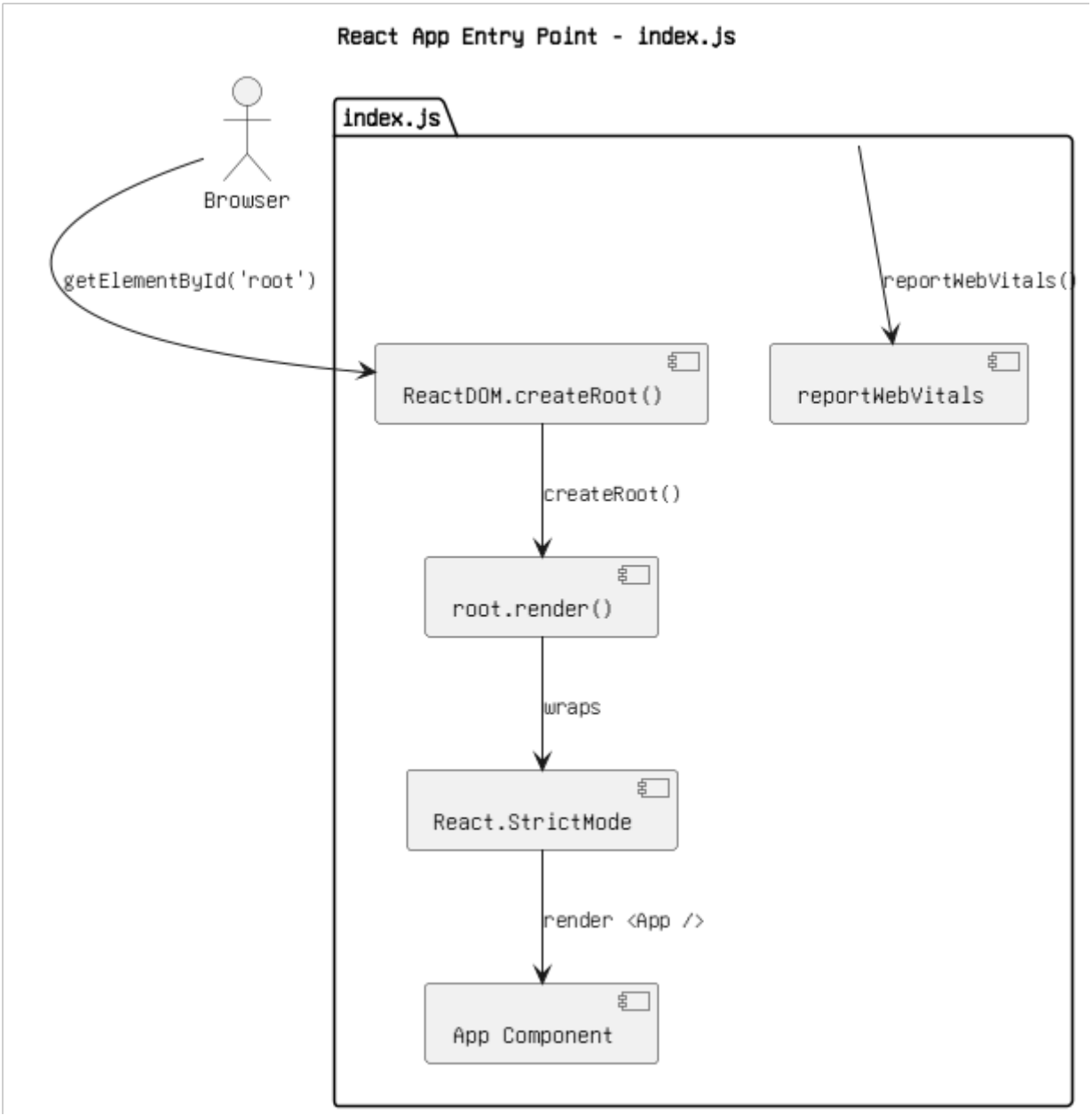

Figure: 8. React App Entry Point - index.js.png

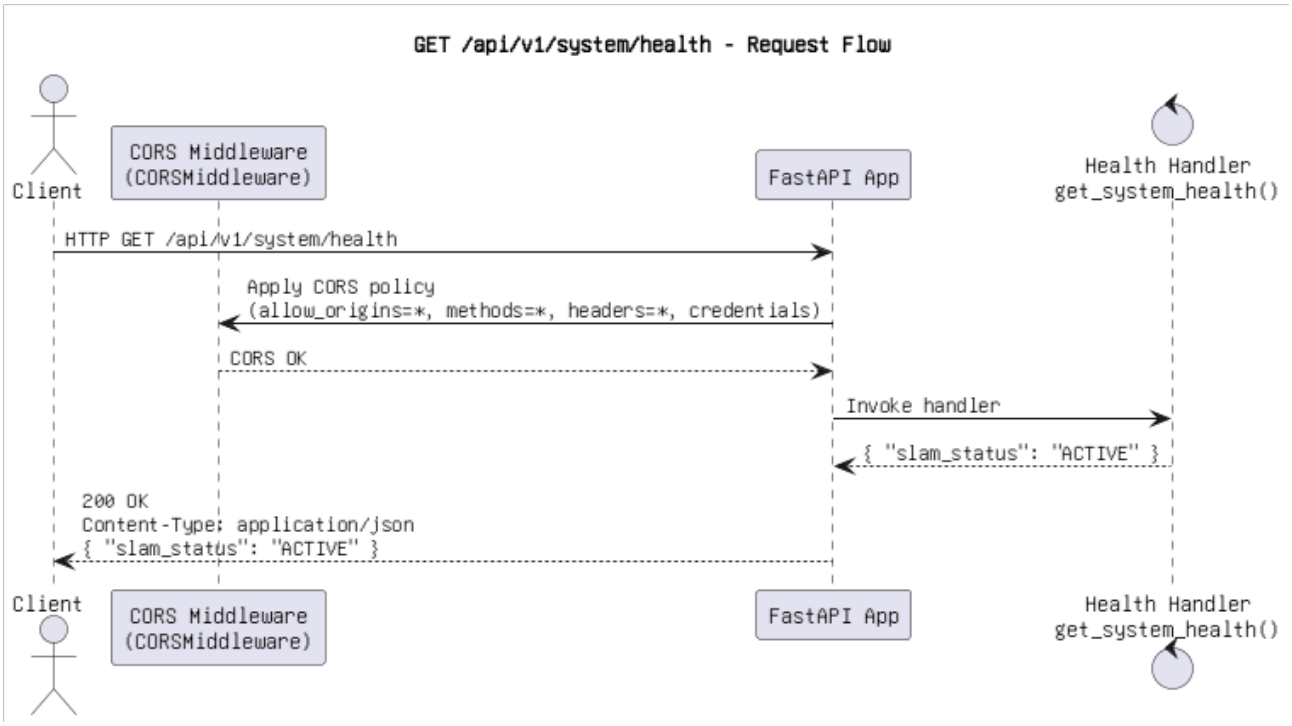

Figure: 9. Request flow .drawio.png

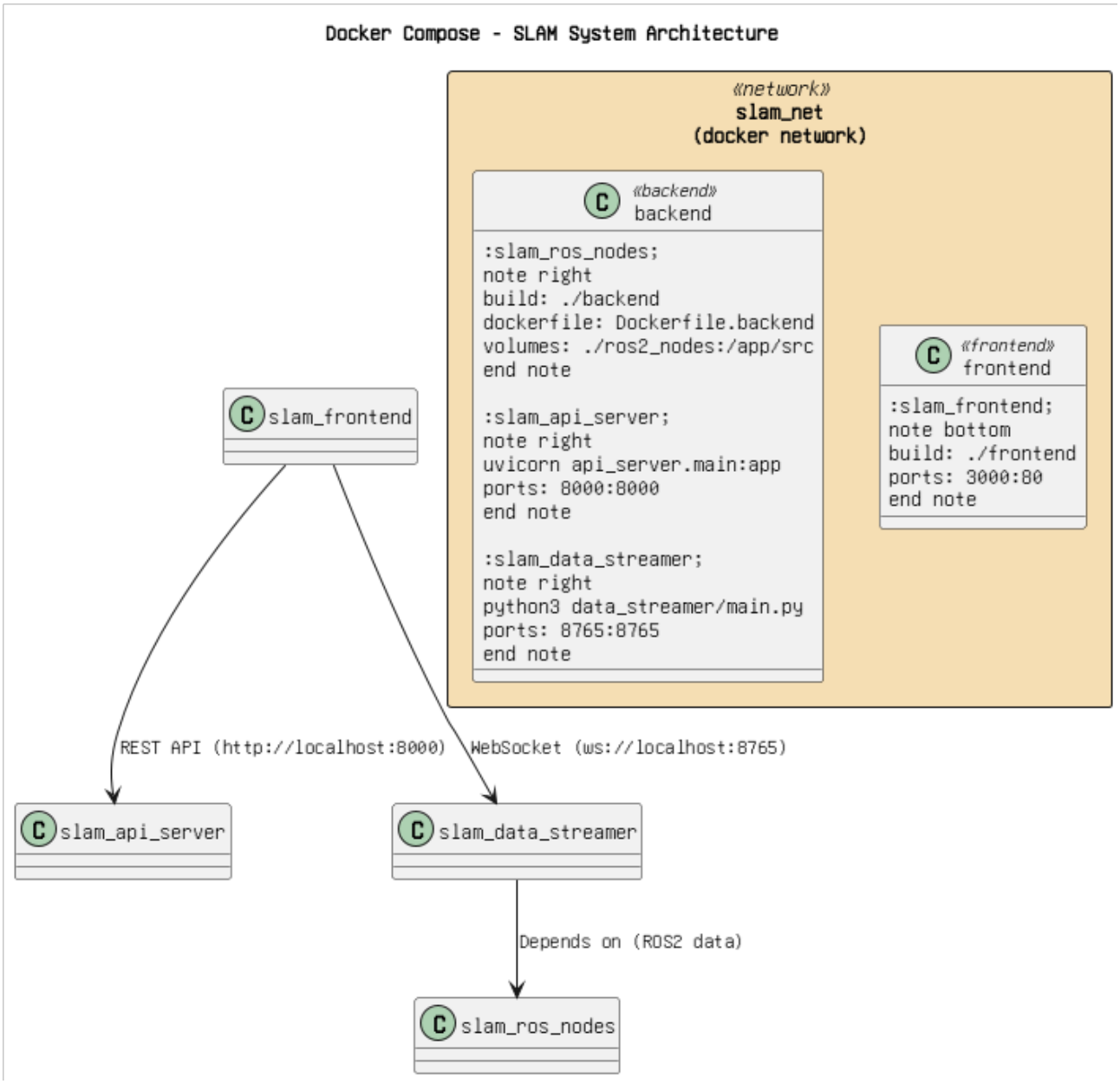

Figure: 10. SLAM System Architecture.png

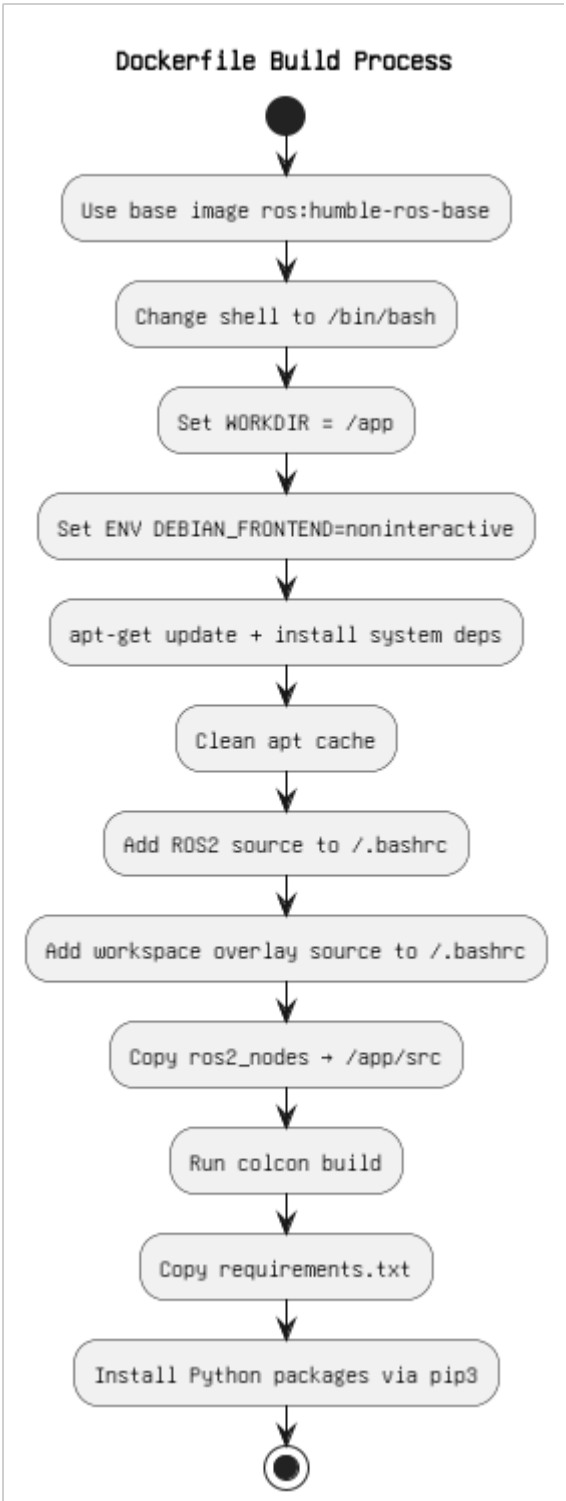

Figure: 11. build process flow.png

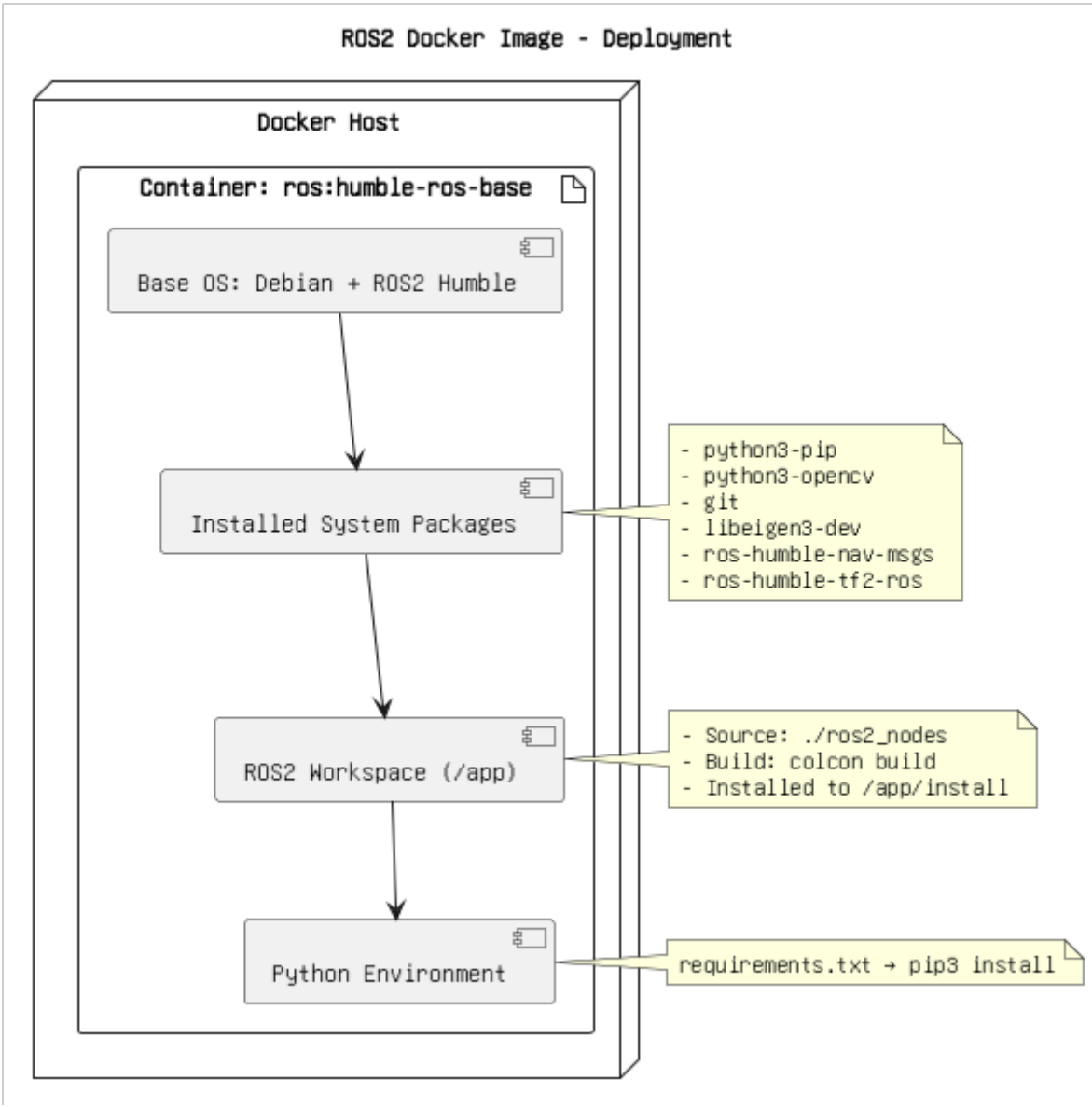

Figure: 12. container setup.png

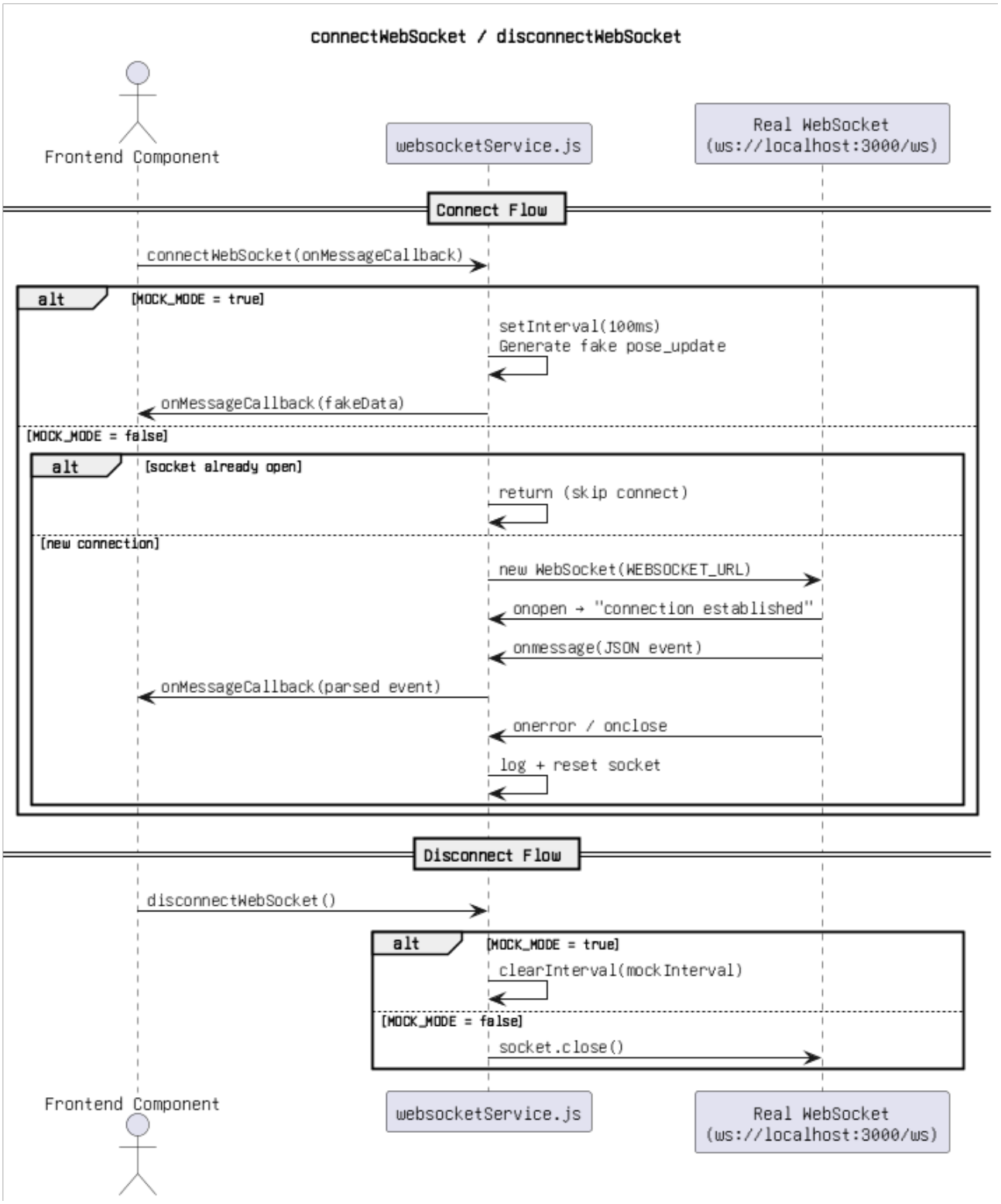

Figure: 13. disconnect websocket.png

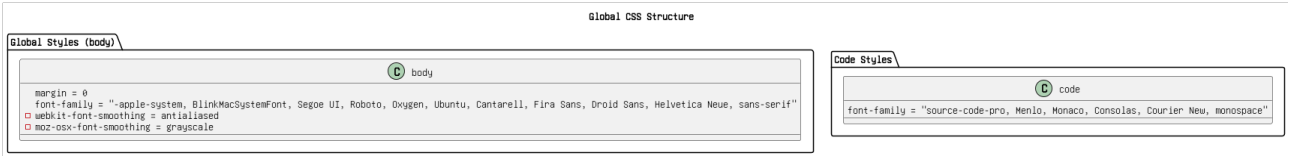

Figure: 14. global css structure.png

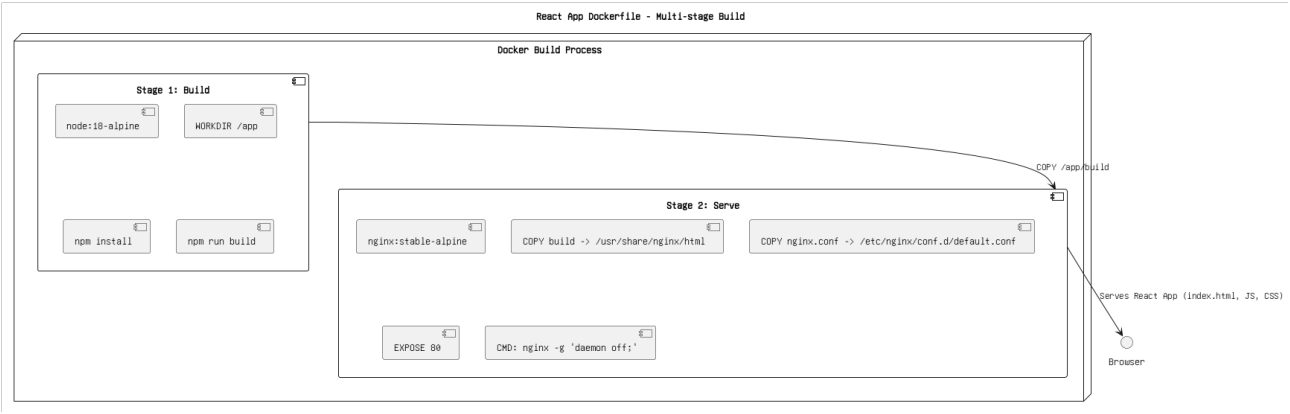

Figure: 15. multi stage build.png

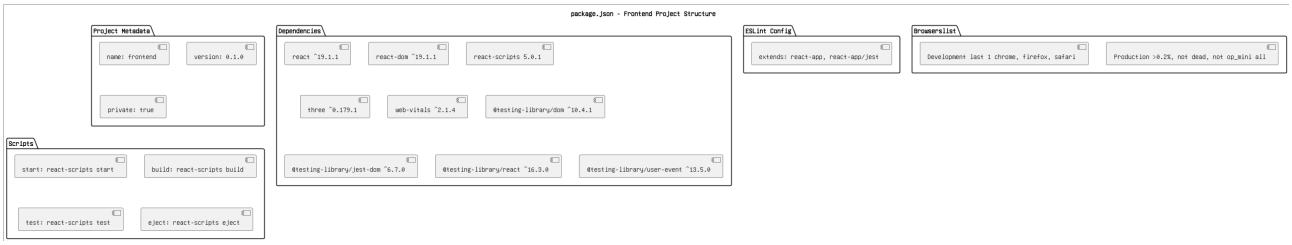

Figure: 16. package.json.png

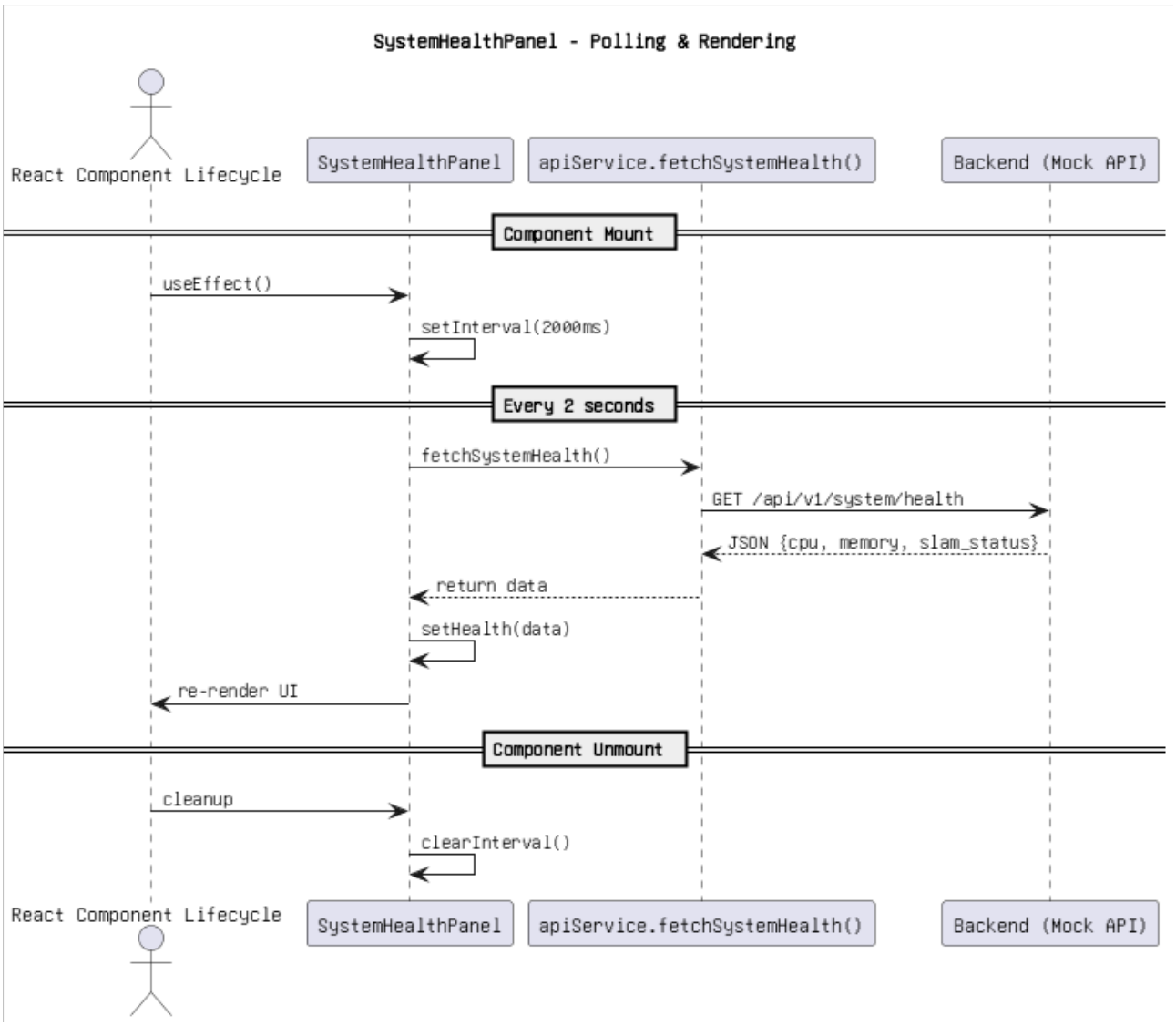

Figure: 17. polling & rendering.png

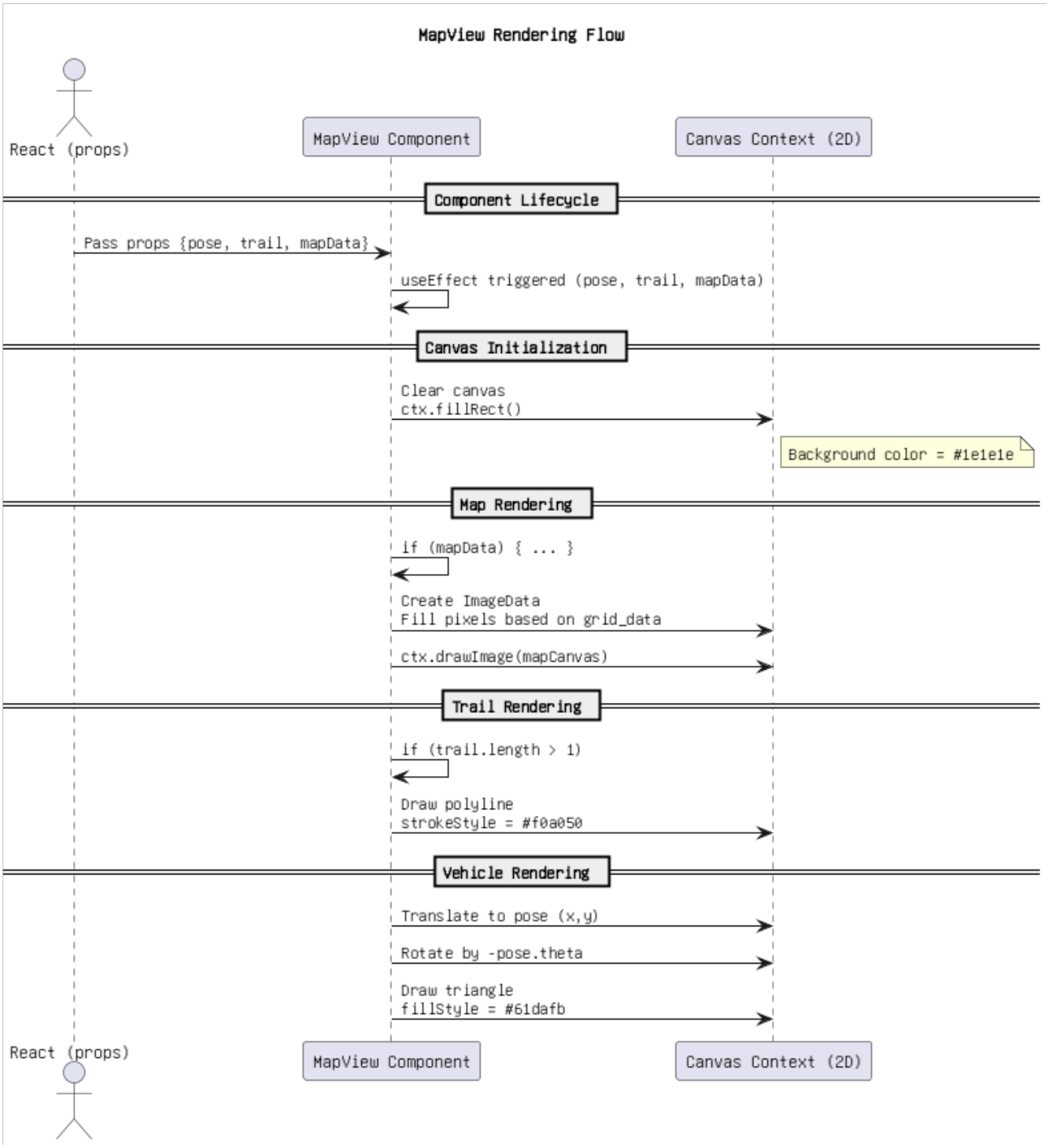

Figure: 18. rendering flow.png

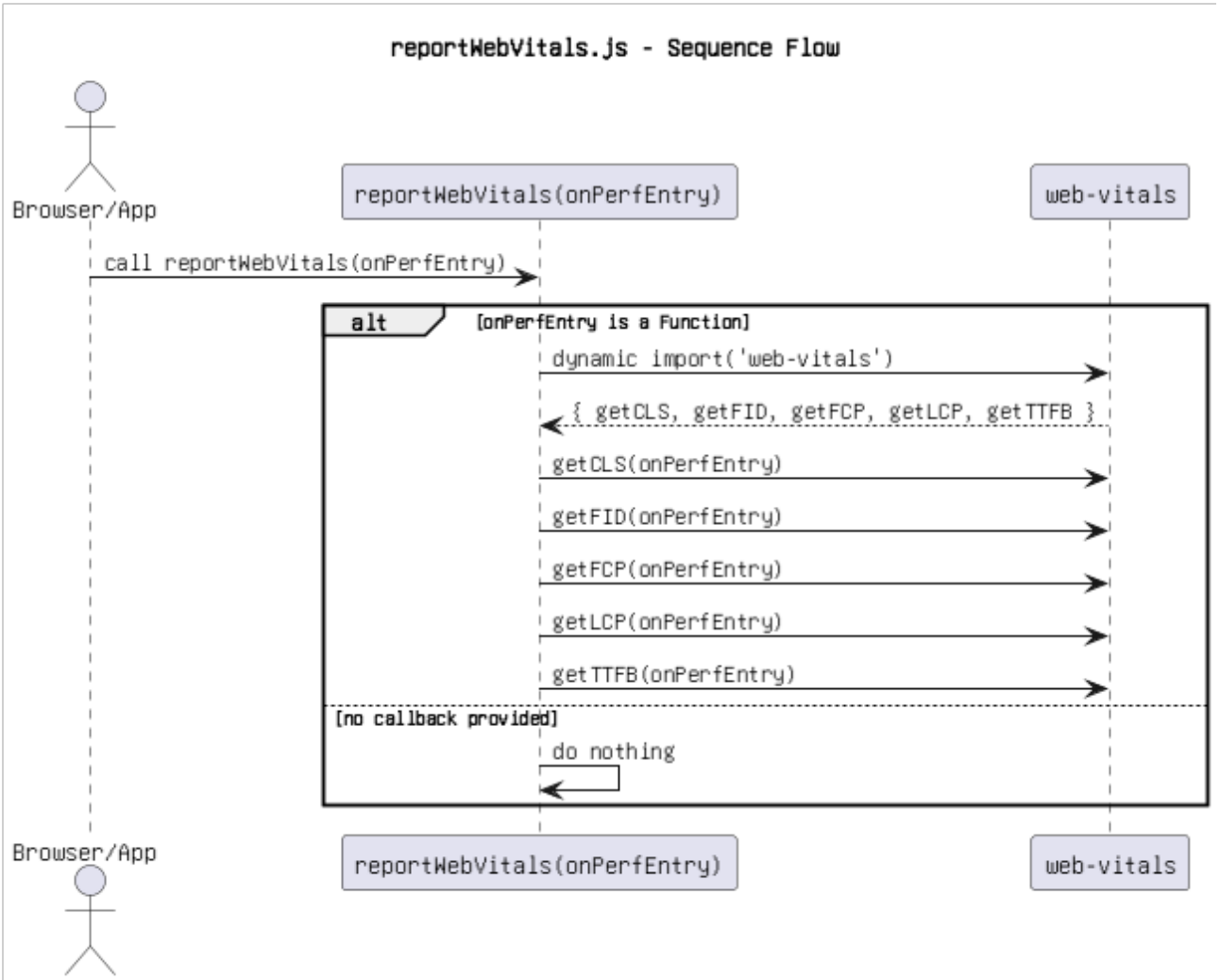

Figure: 19. reportWebVitals.js - Sequence Flow.png

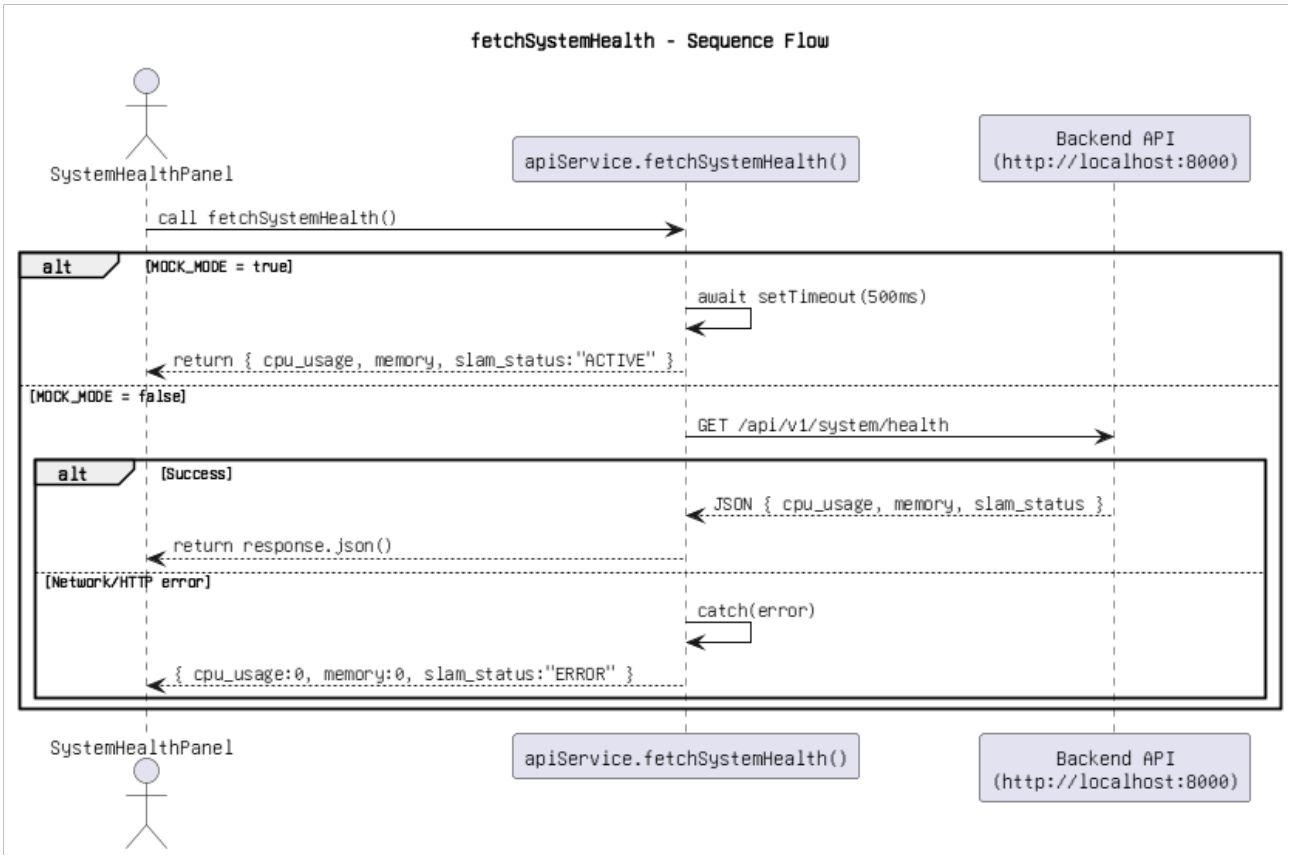

Figure: 20. sequence flow.png

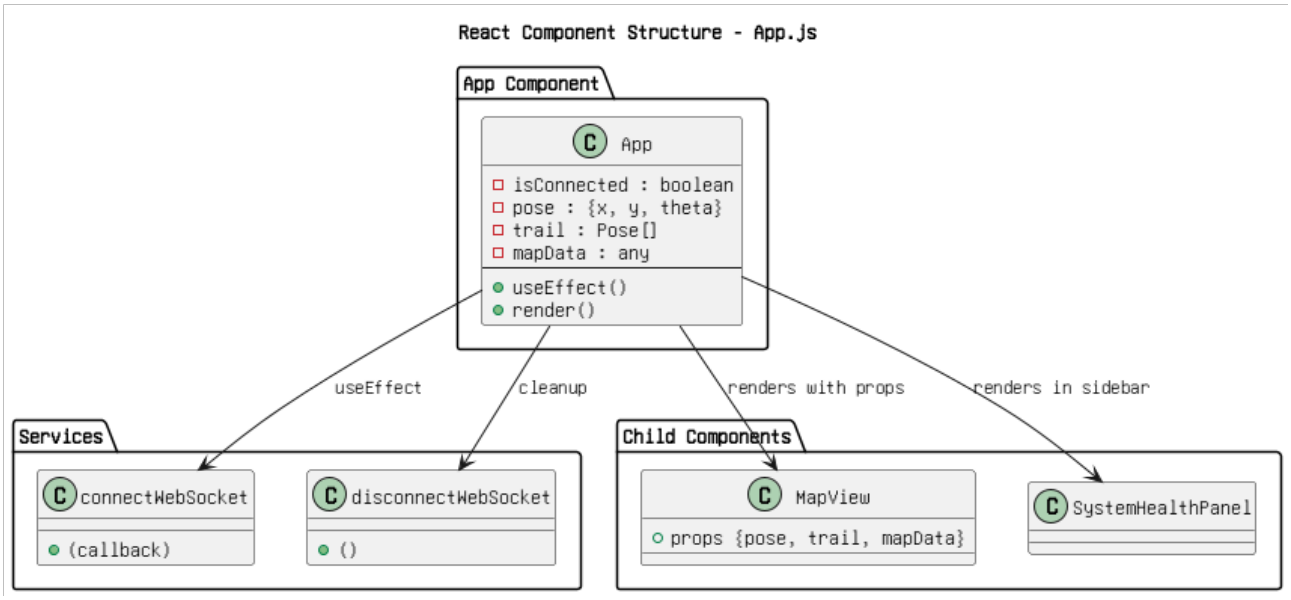

Figure: 21. structure app.js.png

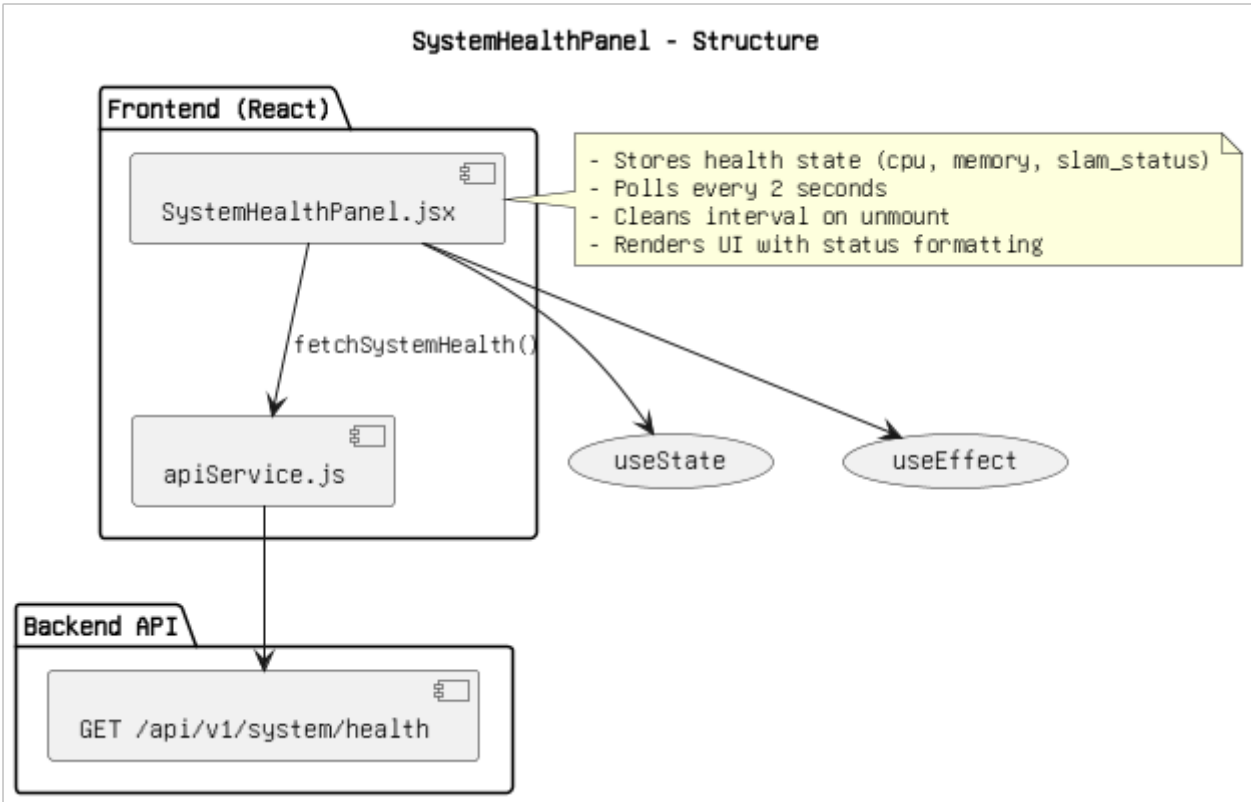

Figure: 22. structure(1).png

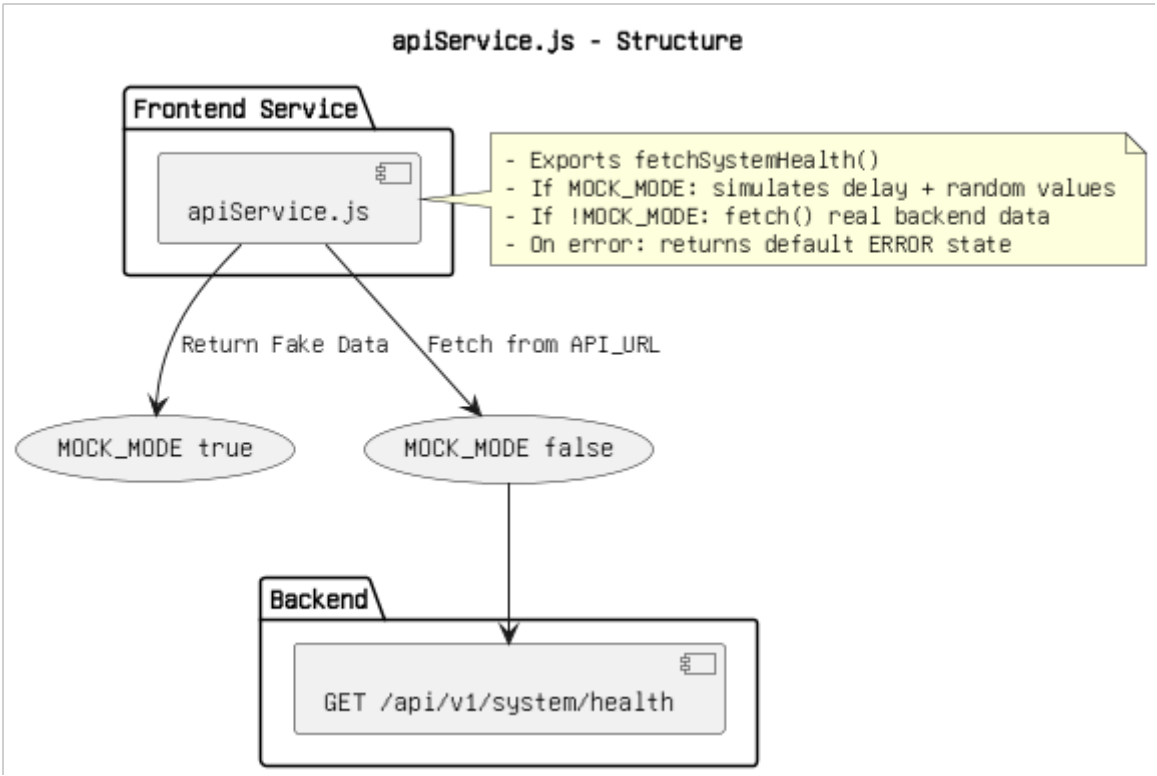

Figure: 23. structure(2).png

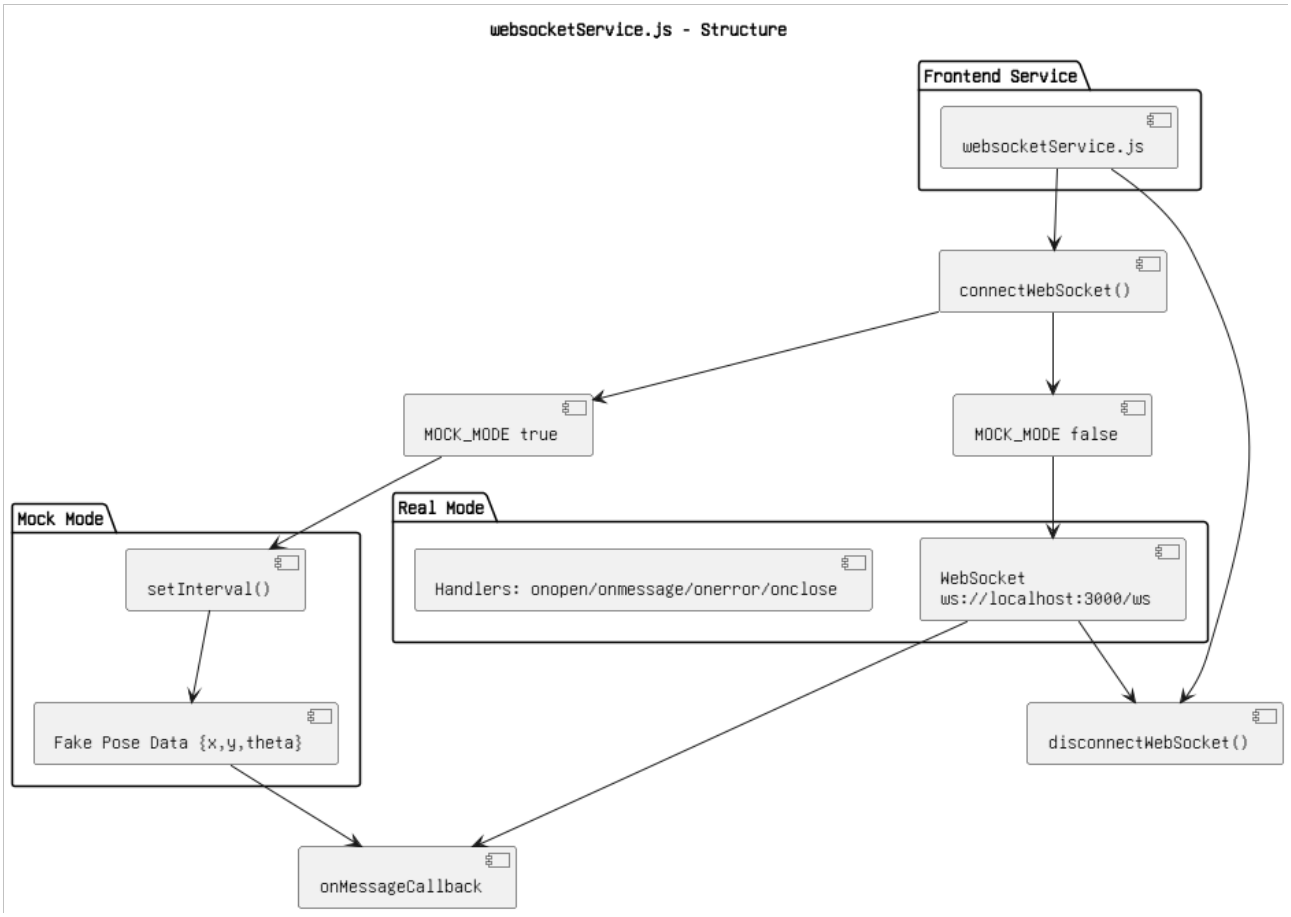

Figure: 24. structure(3).png

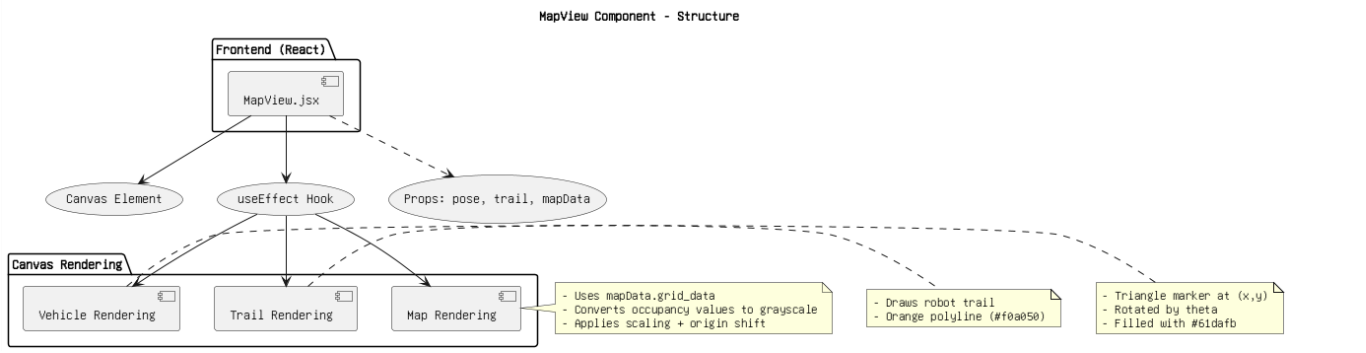

Figure: 25. structure.png

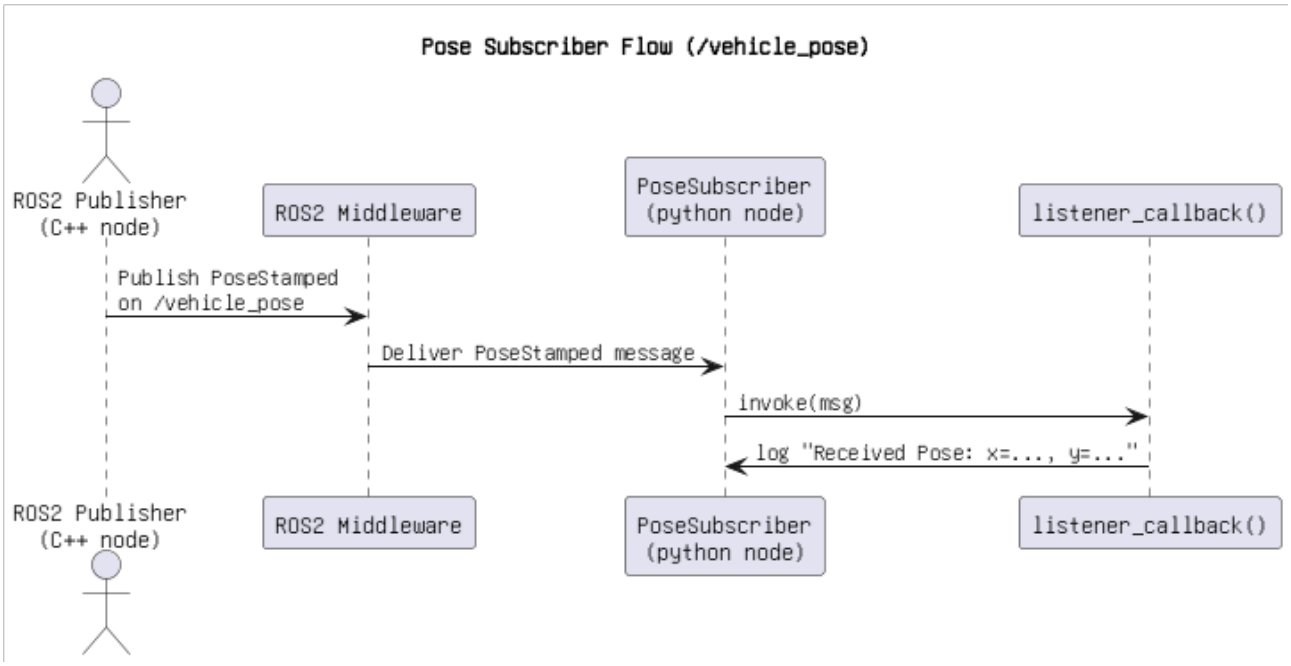

Figure: 26. vehicle_pose.png