RT\_SLAM\_AVOC

Comprehensive Project Report

Author: Abhishek Chauhan

# Table of Contents

Use Word's 'References -> Table of Contents' to generate TOC after opening the document.

# 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[rHÍqò

## .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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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

Top/Module 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\_

Top/Module 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

Top/Module comments:

!/bin/bash

## docs/rest\_api.md

REST API Endpoints  
# Get System Health

Top/Module comments:

REST API Endpoints  
# Get System Health

## docs/websocket\_api.md

Message for Live Pose Updates

Top/Module comments:

Message for Live Pose Updates

## frontend/Dockerfile

Stage 1: Build the React application

Top/Module comments:

Stage 1: Build the React application

## frontend/README.md

Getting Started with Create React App

Top/Module comments:

Getting Started with Create React App

## frontend/public/robots.txt

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

Top/Module 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

Top/Module 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

Top/Module comments:

REST API Endpoints  
# Get System Health

## websocket\_api.md

Message for Live Pose Updates

Top/Module comments:

Message for Live Pose Updates

# List of Figures

|  |  |  |
| --- | --- | --- |
| Figure | Caption | Page (PDF) |
| Architecture.png | 1. Architecture.png | 13 |
| CSS.png | 2. CSS.png | 13 |
| Component architecture.png | 3. Component architecture.png | 14 |
| Components.drawio.png | 4. Components.drawio.png | 14 |
| Data Streaming.png | 5. Data Streaming.png | 15 |
| Mock api request flow.png | 6. Mock api request flow.png | 16 |
| Project Orchestration.png | 7. Project Orchestration.png | 16 |
| React App Entry Point - index.js.png | 8. React App Entry Point - index.js.png | 17 |
| Request flow .drawio.png | 9. Request flow .drawio.png | 18 |
| SLAM System Architecture.png | 10. SLAM System Architecture.png | 19 |
| build process flow.png | 11. build process flow.png | 20 |
| container setup.png | 12. container setup.png | 21 |
| disconnect websocket.png | 13. disconnect websocket.png | 22 |
| global css structure.png | 14. global css structure.png | 22 |
| multi stage build.png | 15. multi stage build.png | 23 |
| package.json.png | 16. package.json.png | 23 |
| polling & rendering.png | 17. polling & rendering.png | 23 |
| rendering flow.png | 18. rendering flow.png | 24 |
| reportWebVitals.js - Sequence Flow.png | 19. reportWebVitals.js - Sequence Flow.png | 25 |
| sequence flow.png | 20. sequence flow.png | 26 |
| structure app.js.png | 21. structure app.js.png | 26 |
| structure(1).png | 22. structure(1).png | 26 |
| structure(2).png | 23. structure(2).png | 27 |
| structure(3).png | 24. structure(3).png | 27 |
| structure.png | 25. structure.png | 28 |
| vehicle\_pose.png | 26. vehicle\_pose.png | 28 |

# Design Diagrams

Architecture.png

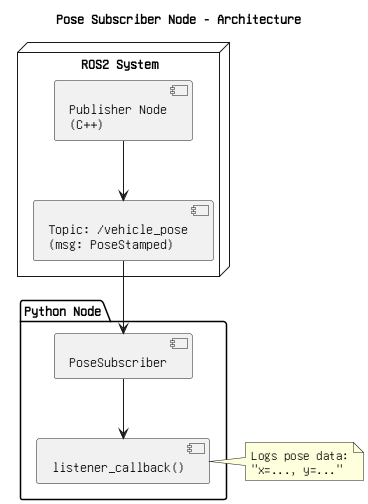


Figure: 1. Architecture.png

CSS.png

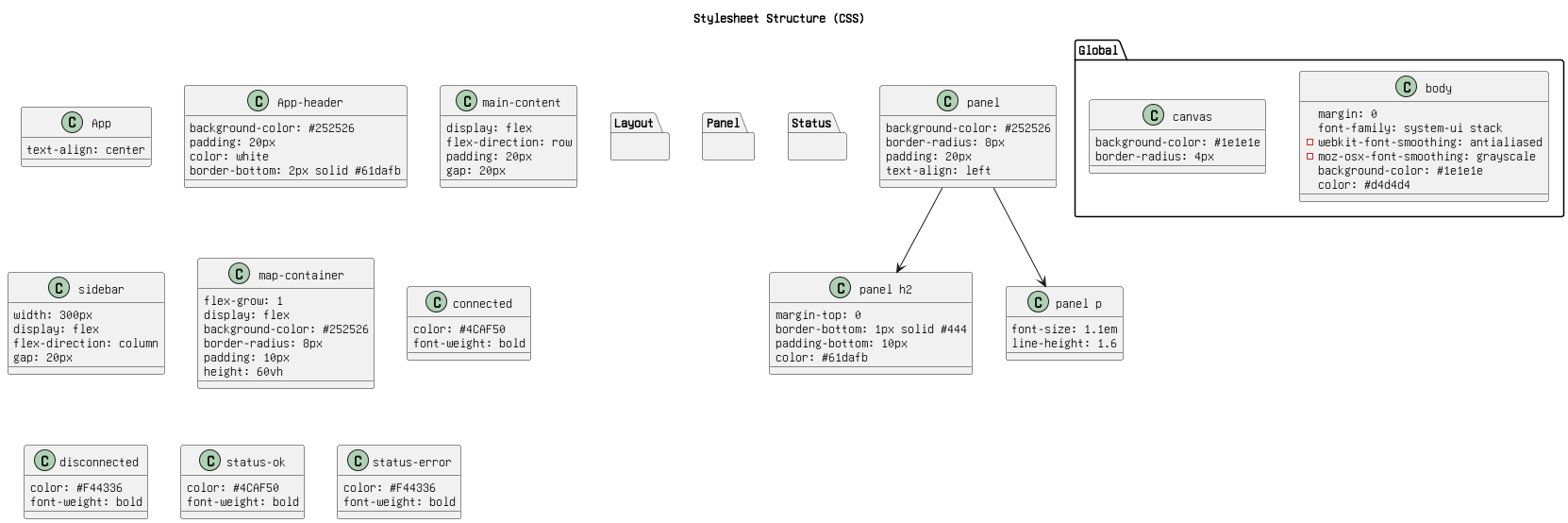


Figure: 2. CSS.png

Component architecture.png

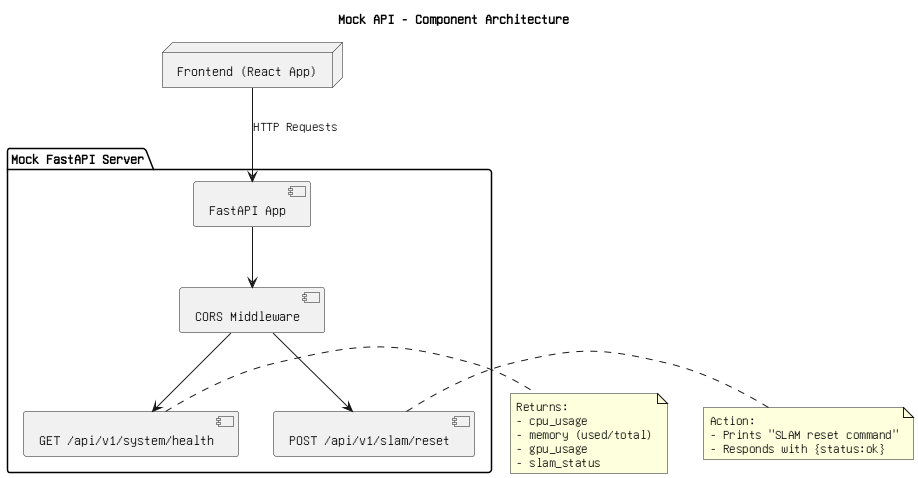


Figure: 3. Component architecture.png

Components.drawio.png

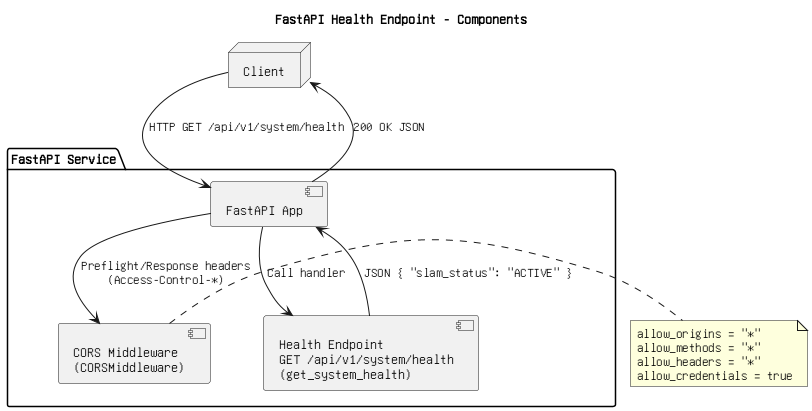


Figure: 4. Components.drawio.png

Data Streaming.png

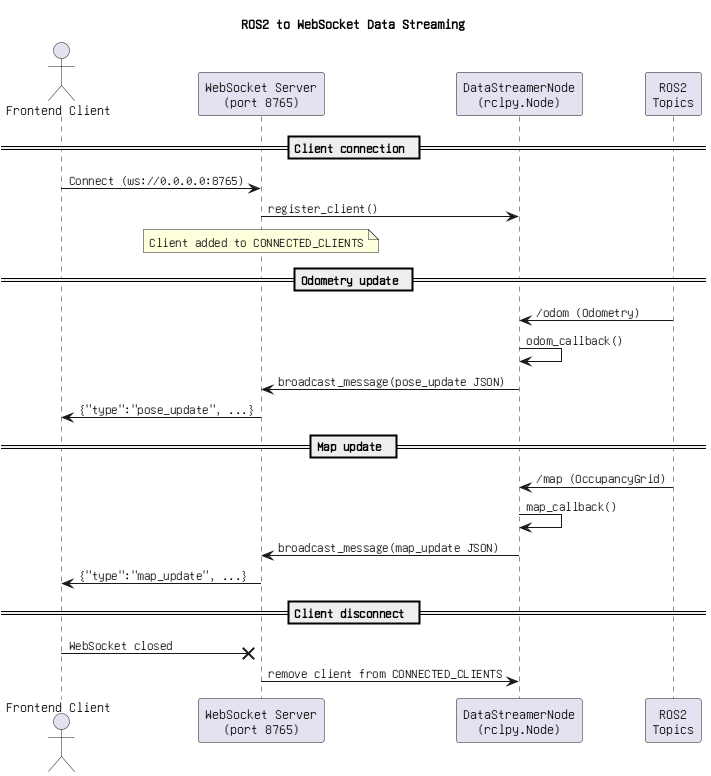


Figure: 5. Data Streaming.png

Mock api request flow.png

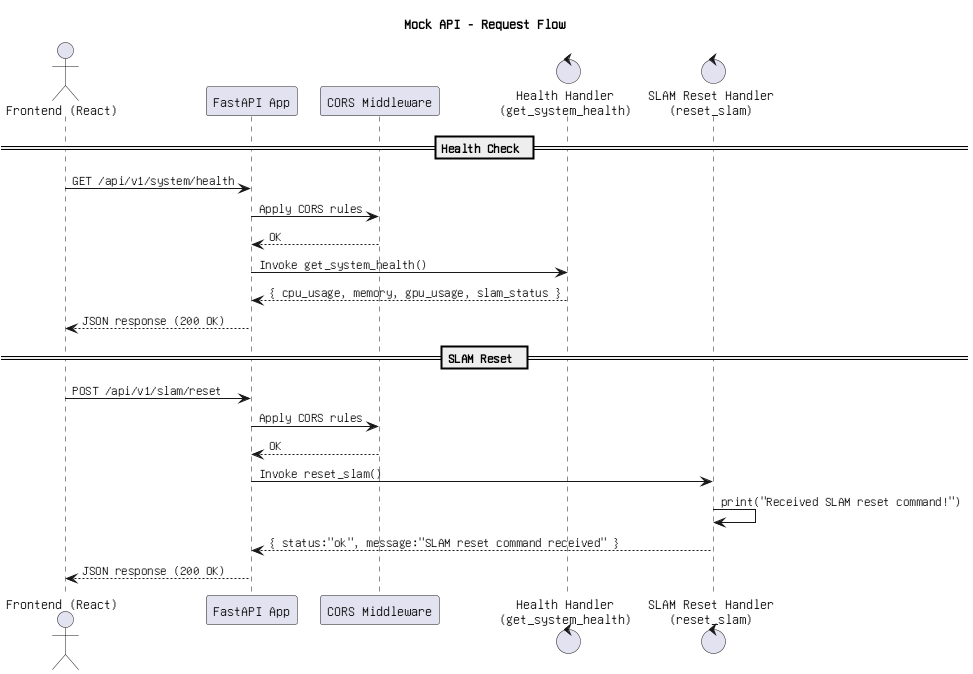


Figure: 6. Mock api request flow.png

Project Orchestration.png

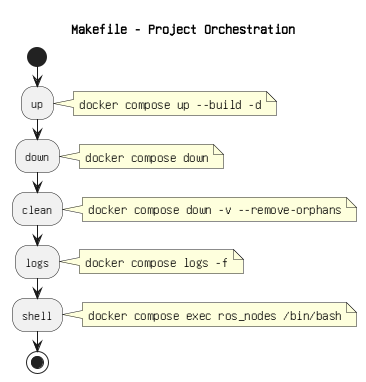


Figure: 7. Project Orchestration.png

React App Entry Point - index.js.png

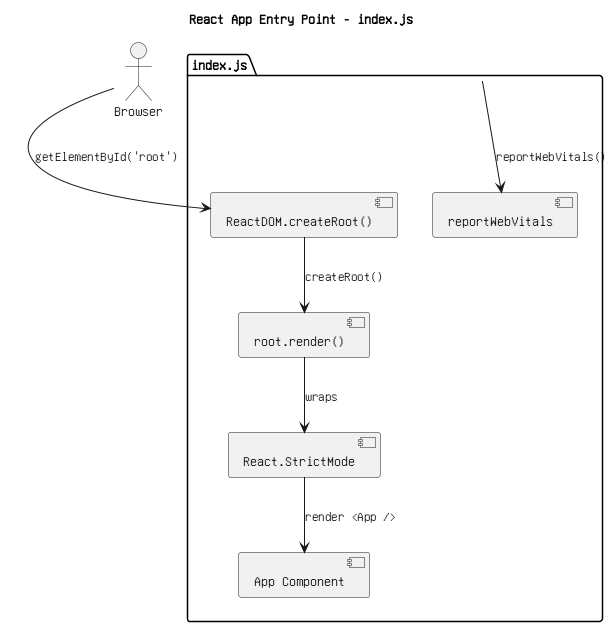


Figure: 8. React App Entry Point - index.js.png

Request flow .drawio.png

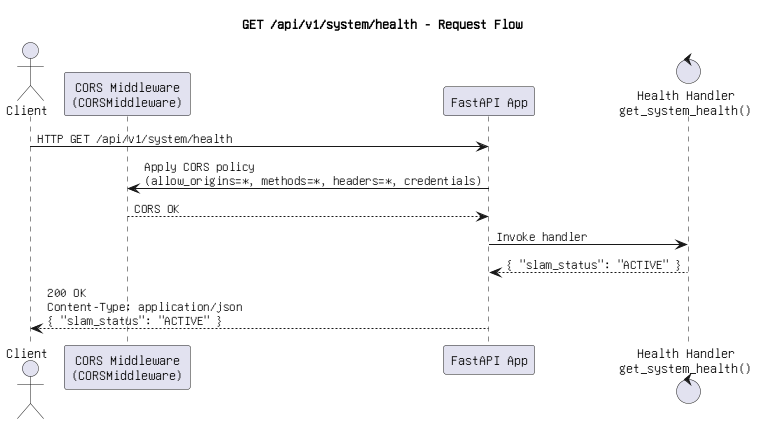


Figure: 9. Request flow .drawio.png

SLAM System Architecture.png

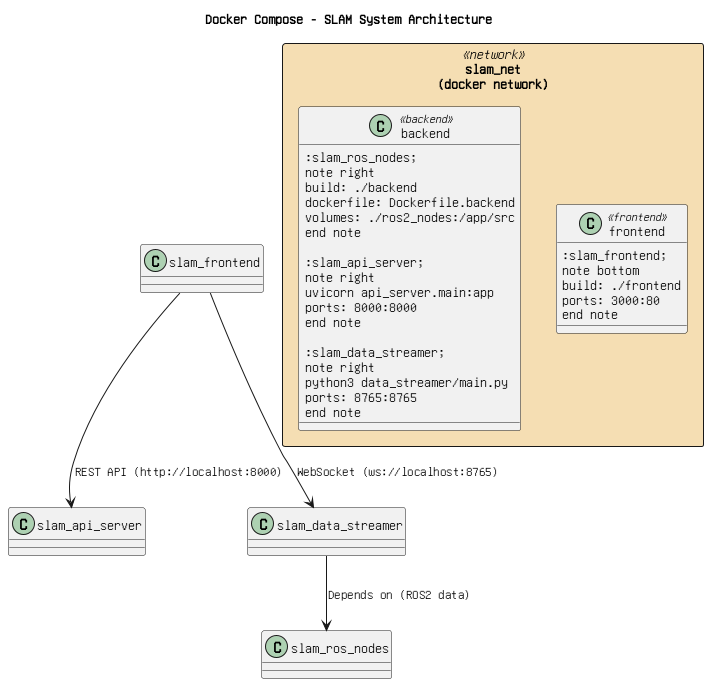


Figure: 10. SLAM System Architecture.png

build process flow.png

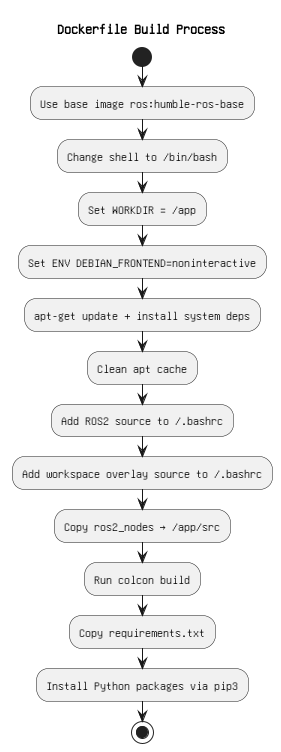


Figure: 11. build process flow.png

container setup.png

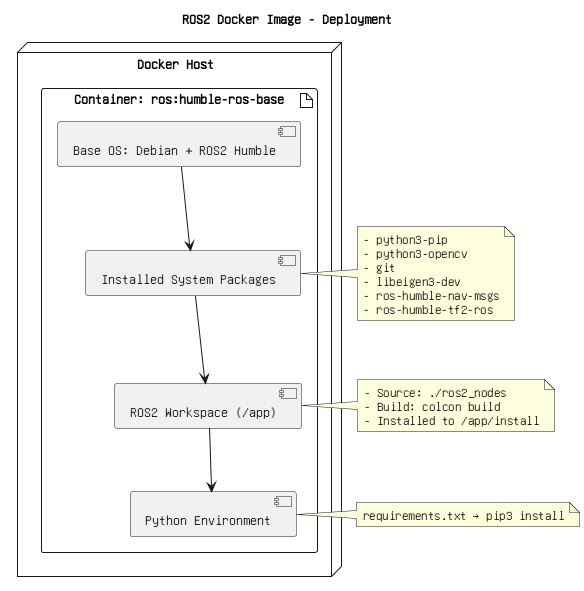


Figure: 12. container setup.png

disconnect websocket.png

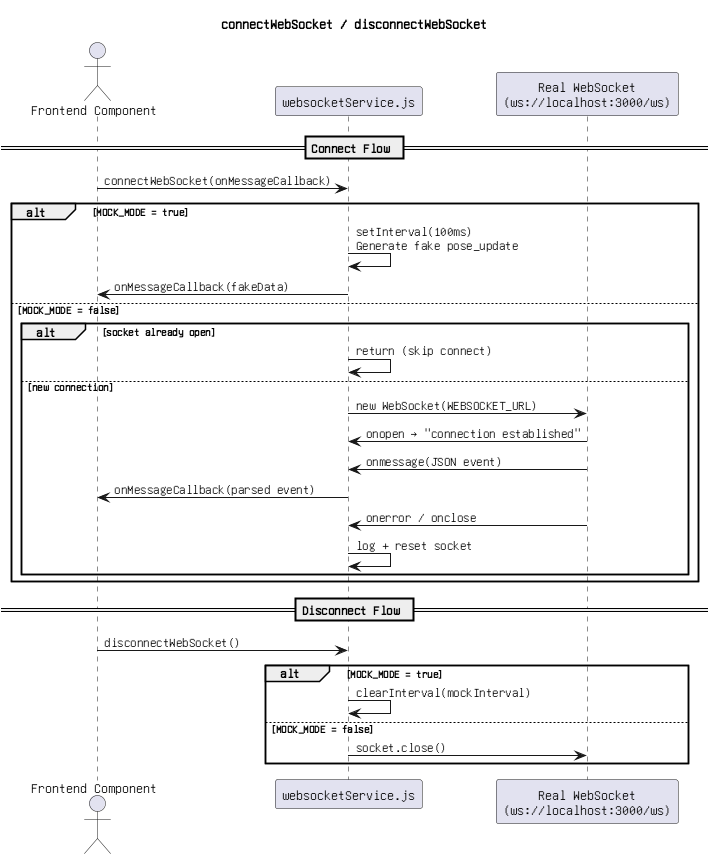


Figure: 13. disconnect websocket.png

global css structure.png

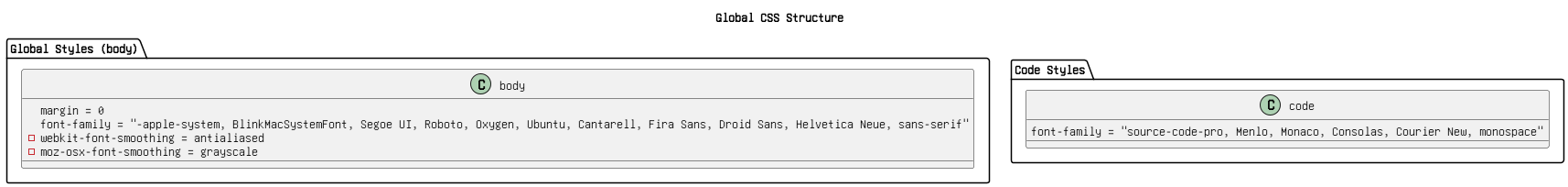


Figure: 14. global css structure.png

multi stage build.png

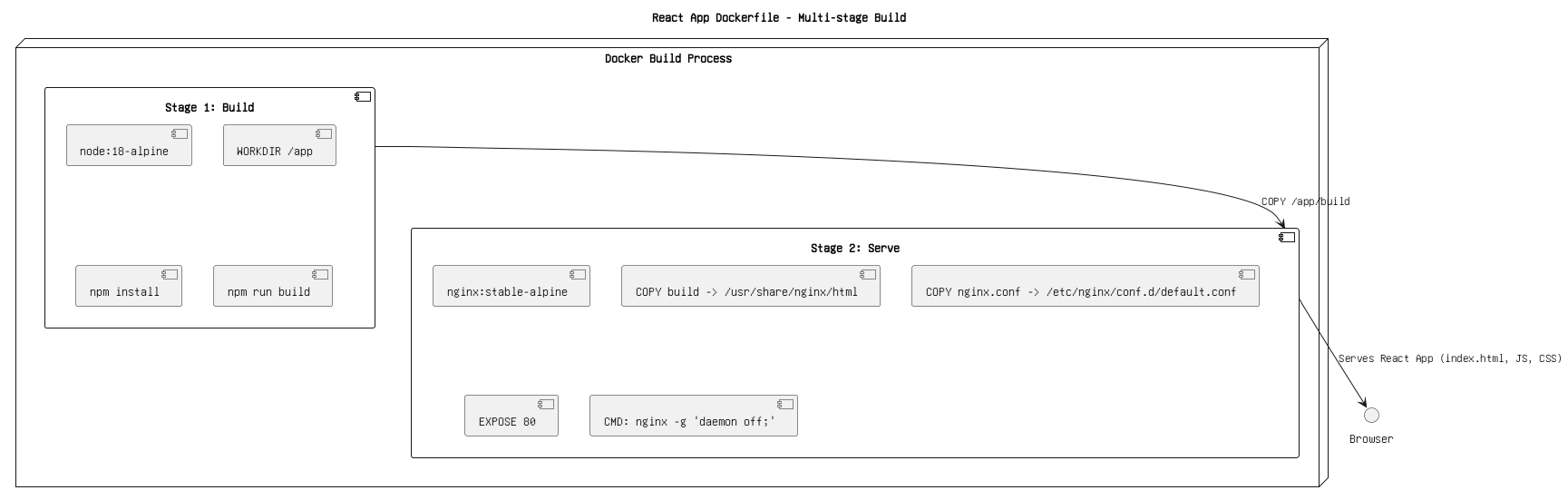


Figure: 15. multi stage build.png

package.json.png

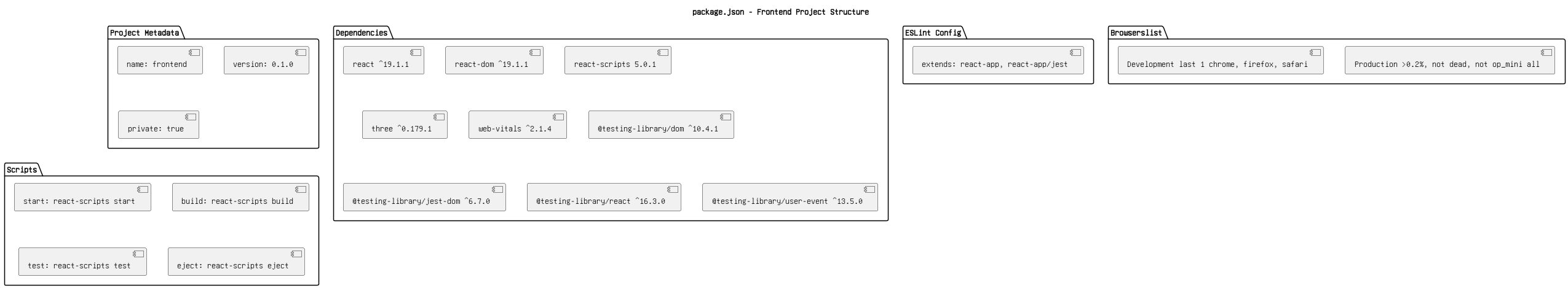


Figure: 16. package.json.png

polling & rendering.png

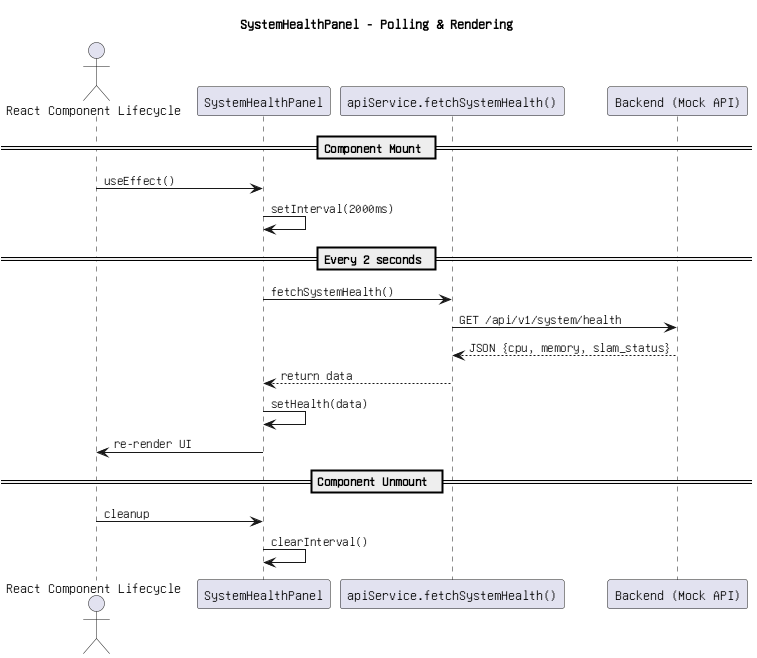


Figure: 17. polling & rendering.png

rendering flow.png

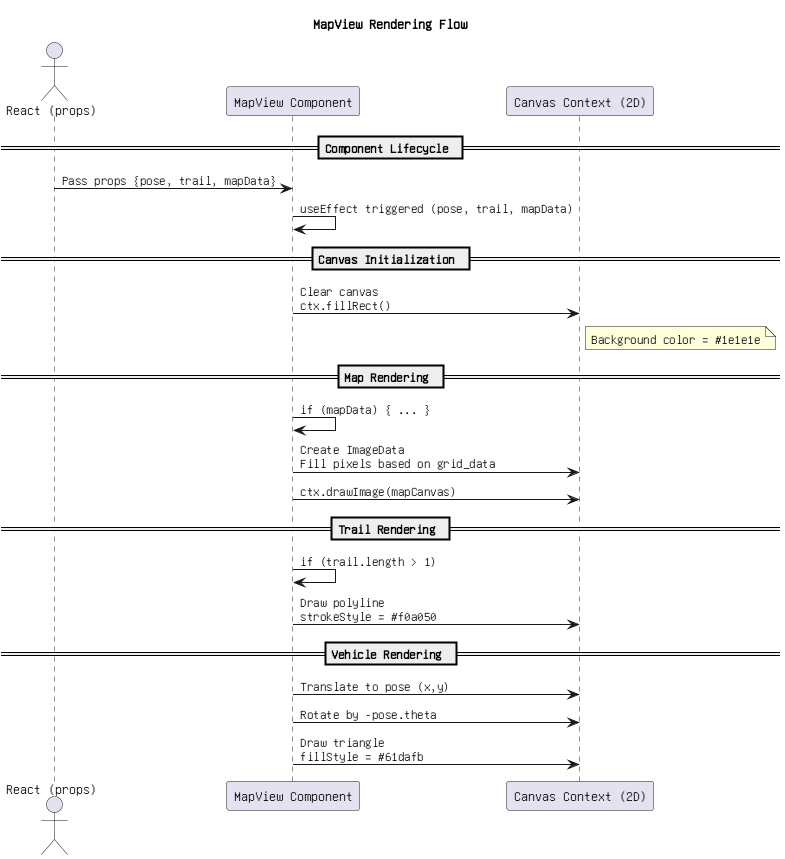


Figure: 18. rendering flow.png

reportWebVitals.js - Sequence Flow.png

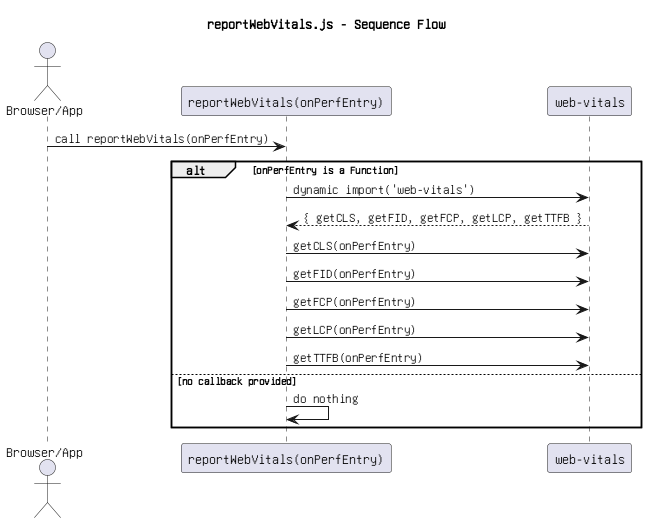


Figure: 19. reportWebVitals.js - Sequence Flow.png

sequence flow.png

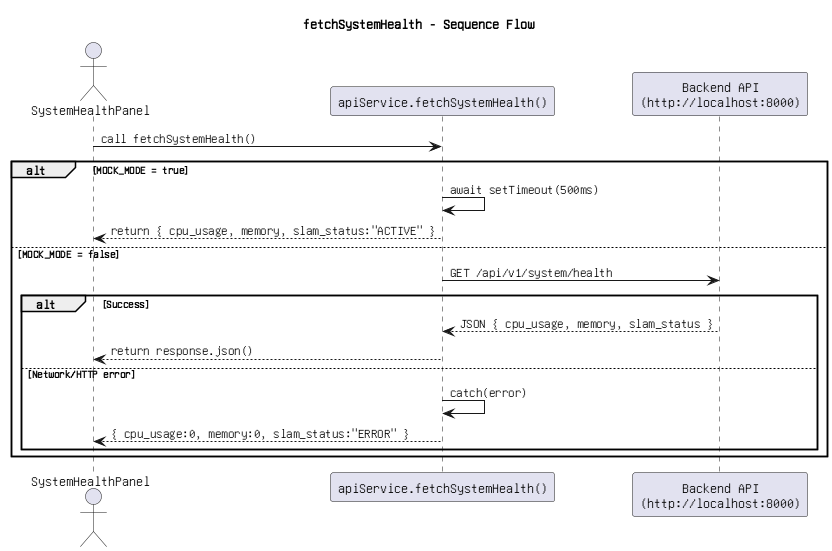


Figure: 20. sequence flow.png

structure app.js.png

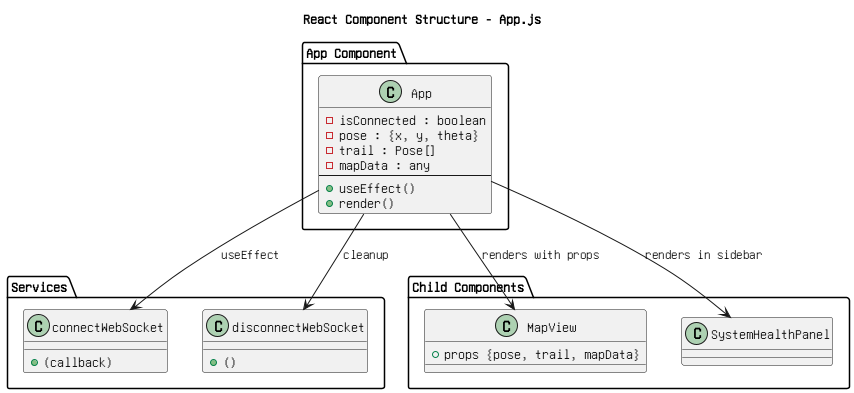


Figure: 21. structure app.js.png

structure(1).png

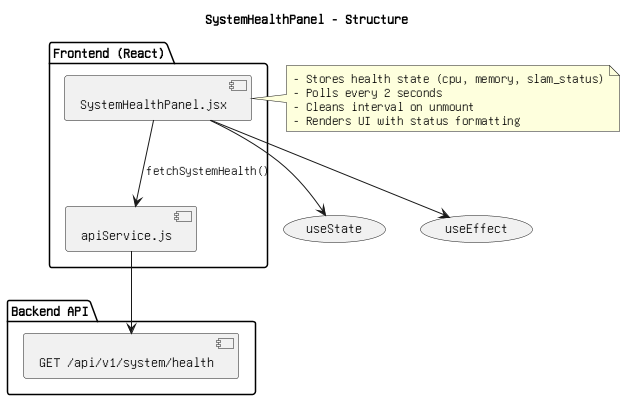


Figure: 22. structure(1).png

structure(2).png

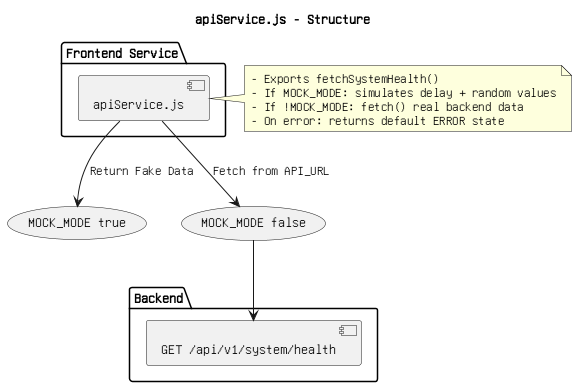


Figure: 23. structure(2).png

structure(3).png

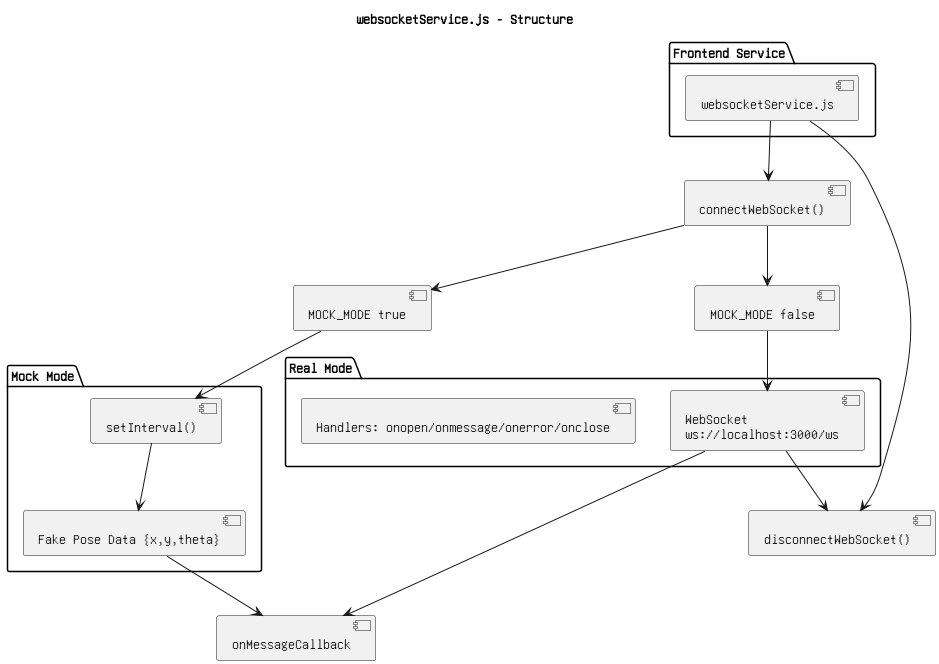


Figure: 24. structure(3).png

structure.png

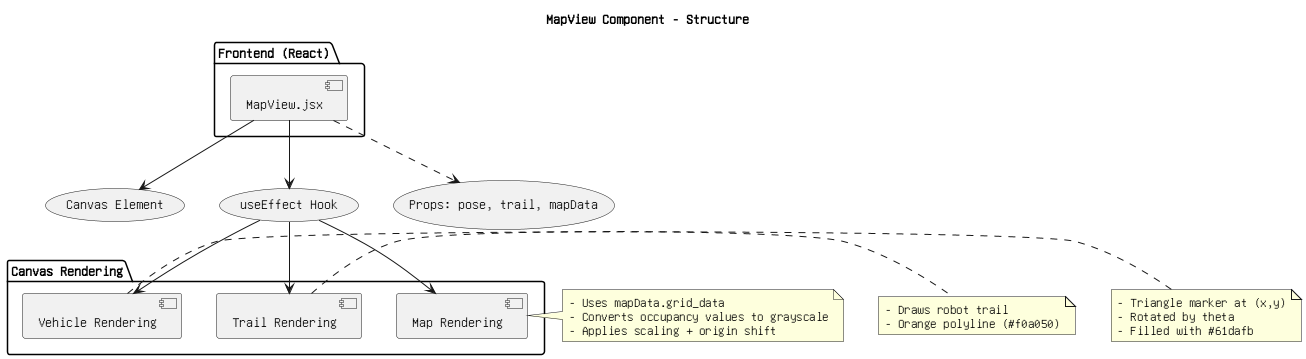


Figure: 25. structure.png

vehicle\_pose.png

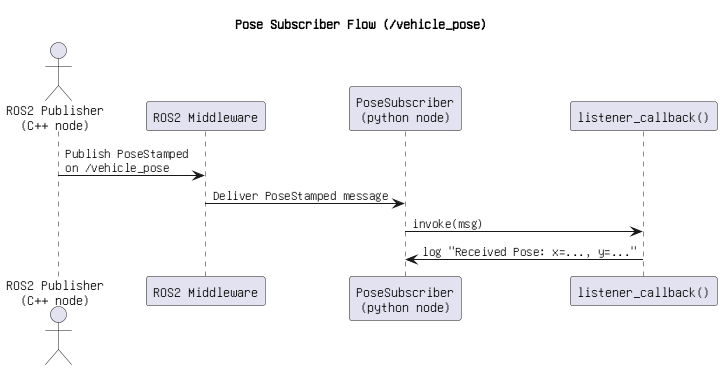


Figure: 26. vehicle\_pose.png