Software Architecture Document

Better Than Autopilot

Team: SuperCoolGuys

Product Owner: Richard Kha

Scrum Master: Long Nguyen

Team Members:

Michael Gresham,

Richard Gresham,

Luke Higgott,

Louis Zuckerman,

Nathaniel Marquez,

Thomas Del Rosario

May 09, 2022

Revision History

Date	Version	Description	Author
02/18/22	1.0	Added purpose and scope	Michael Gresham
03/02/22	1.1	Added Use cases and Use case diagram	Michael & Richard Gresham
03/10/22	1.2	Added Logical Diagram	Long Nguyen
03/10/22	1.3	Added Architectural Representation	Richard Gresham
03/13/22	1.4	Added Architectural Goals and Constraints	Thomas Del Rosario & Richard Kha
03/14/22	1.5	Implemented Quality Section of SAD	Richard Gresham
03/15/22	1.6	Finished size and performance section	Michael Gresham
03/16/22	1.7	Added Table of Contents and Table of Figures. Also included the overview	Michael Gresham
03/16/22	1.7	Updated references and Table of contents	Richard Gresham
3/19/22	1.8	Updated Table of contents and organized numbering of contents	Thomas Del Rosario
03/22/22	1.9	Cleaned up and removed unnecessary items. As well as updating the table of figures.	Michael Grehsam
03/22/22	1.9	Updated console module packages	Long Nguyen

		decomposition and class design	
4/28/22	2.0	Added hyperlinked table of contents. Added section breaks between sections. Formatted for uniformity, updated table of figures.	Louis Zuckerman
4/28/22	2.1	Added appendix reviewed and changed several commented areas, and added comments to the documents.	Richard Gresham
05/02/22	2.2	Added use cases to the Use Case Section, and updated table of contents as well.	Michael Gresham

Table of Contents

Table of Figures

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of Sonny Motors Company's Navigation and Automatic Steering System, using a number of architectural views from the 4+1 model to depict different aspects of the system. It is intended to capture and display the significant architectural decisions that have been made on the system.

1.2 Scope

This Software Architecture Document provides an architectural overview of the Navigation and Automatic Steering System. This system is being developed by Sonny Motors Company to guide their automobiles on city streets and highways from their current locations to their supplied destinations. It will perform this task using a computer that controls two steering actuators, receives GPS coordinates, handles speed change request, detects obstacles, delivers on board messages to drivers, and provides an interface so that the user can interact with the system. Note the system is not meant to be autonomous and still requires a driver to be focused on the road in case the NASS system needs to be suspended.

1.3 Definitions, Acronyms, and Abbreviations

NASS - Navigation and Automatic Steering System

Steering Actuator - Is a device meant to aide in the Steering of a vehicle

GPS - Global Positioning System: a navigational system using satellite signals to fix the location of a radio receiver on or above the earth's surface.

ECU - Electronic Control Unit is a small device in a vehicle's body that is responsible for controlling a specific function.

CAN - Controller Area Network or CAN bus is a robust vehicle bus standard designed to Allow microcontrollers and devices to communicate with each other's applications without a host computer.

VANET - Vehicular ad hoc Network which is the wireless multi-hop network and inter-vehicular communications, which is used when gaining information from groups of moving or stationary cars.

IDS - Intrusion detection system is a device or software application that monitors a network for malicious activity or policy violations.

BTA - references the name BetterThanAutopilot.

ROS - Robot Operating System which is an open source robotics middleware suite. Not an OS but a set of software frameworks for robot software development or AI development..

1.4 References

Applicable references are:

1. Kyounggon Kim, Jun Seok Kim, Seonghoon Jeong, Jo-Hee Park, Huy Kang Kim, Cybersecurity for autonomous vehicles: Review of attacks and defense, Computers & Security,Volume 103, 2021, 102150, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2020.102150.

2. P. B. Kruchten, "The 4+1 View Model of architecture," in *IEEE Software*, vol. 12, no. 6, pp. 42-50, Nov. 1995, doi: 10.1109/52.469759.

1.5 Overview

This document serves as the primary documentation of the NASS software architecture. This document will first go over the 4+1 model and the goals and restraints of the documentation. It will then explore the various viewpoints covered in the 4+1 model such as the use-case view, the logical view, development view, process view, and finally the physical view. After the viewpoint this document will cover the size and performance constraints, and finally the quality requirements.

2. Architectural Representation

This document details the architecture of the NASS using the views defined in the "4+1" model. The views used to document the Sonny Motor's Company's Navigation and Automatic Steering System is shown below [2]:

2.1 Use Case View/ Scenarios

Audience: Stakeholders, end-users, analysts/designers

Area: Describes a series of scenarios or use cases that represent some significant functionality of the NASS.

Related Artifacts: Use-Case Model, Use Case Diagram, UML.

2.2 Logical View

Audience: Stakeholders, end-users, analysts/designers

Area: affects Functional Requirements: it describes the design's object model. Also describes the most important use-case realizations via diagrams.

Related Artifacts: Activity diagrams, class diagram, package diagrams

2.3 Development View

Audience: Programmers

Area: Software components: the static organization of the software in its development environment, or its layers and subsystems of the application.

Related Artifacts: Component diagram, Package diagram.

2.4 Process View

Audience: System designers, integrators.

Area: Non-Functional Requirements: captures the concurrency and synchronization aspects of the design, system processes and how they communicate together.

Related Artifacts: Process diagram, activity diagram.

2.5 Physical View

Audience: System engineers, designers

Area: System topology, delivery, installation, and communication. It maps Process View artifacts, processes onto the system hardware.

Related Artifacts: deployment diagram.

3. Architectural Goals and Constraints

In this architecture, there are goals and constraints that are important to the overall design. This section describes the software requirements and how they can significantly impact the architecture.

3.1 Security

The NASS system will prevent any unauthorized activities using precautionary measures such as encryption of data, secure boot, and an embedded firewall.

- Data that is inputted into the system will be encrypted to prevent possible interference.

3.2 Implementation Strategy and Design

- Upon the start of setting the destination, the system will refresh its location every second.
- The integrated system will have its own storage that allows for offline use of maps which allows the customer to reach the designated location 95% of the time without network connectivity.
- Cameras will be integrated throughout the car and communicate with the system to keep track of the lanes or any other nearby objects.
- The architecture will meet the requirements specified by the product owner.
- The system will take speed limits of the current road into account when managing the speed of the vehicle.

3.3 Safety

In case of an unmoving obstacle in the way of the vehicle, a speed change request of 0 will be sent to the control system to apply brakes to the car. The driver will also be notified with a message and sound to take over with manual steering. Any unexpected events will also notify the driver to take manual driving control.

3.4 Portability

The software needed will need to be compatible with different car models and systems.

3.5 Performance

The software will process requests such as speed change requests and system controls within 100 milliseconds.

3.6 Distribution

The system will have multiple language options for the user to fit their needs.

4. Use-Case View

The NASS use cases are:

- Selecting Route
- Selecting Destination
- Updating Navigation Display
- Suspending automatic Steering
- Driver Assumes Manual Control
- Resuming automatic Steering
- Turning OFF/ON Automatic Steering (Driver's initiated)
- Managing Speed.
- Changing Route
- Send Obstacle Warning Message
- Lane Changing/Turning
- Emergency Brakes
- Pull Over
- Warning Message
- Send Velocity Position Message
- Send Available Routes
- Send Selected Route

4.1 Use-Case Diagrams

Figure 4.1. Use-case Diagram: UCs: 1-14

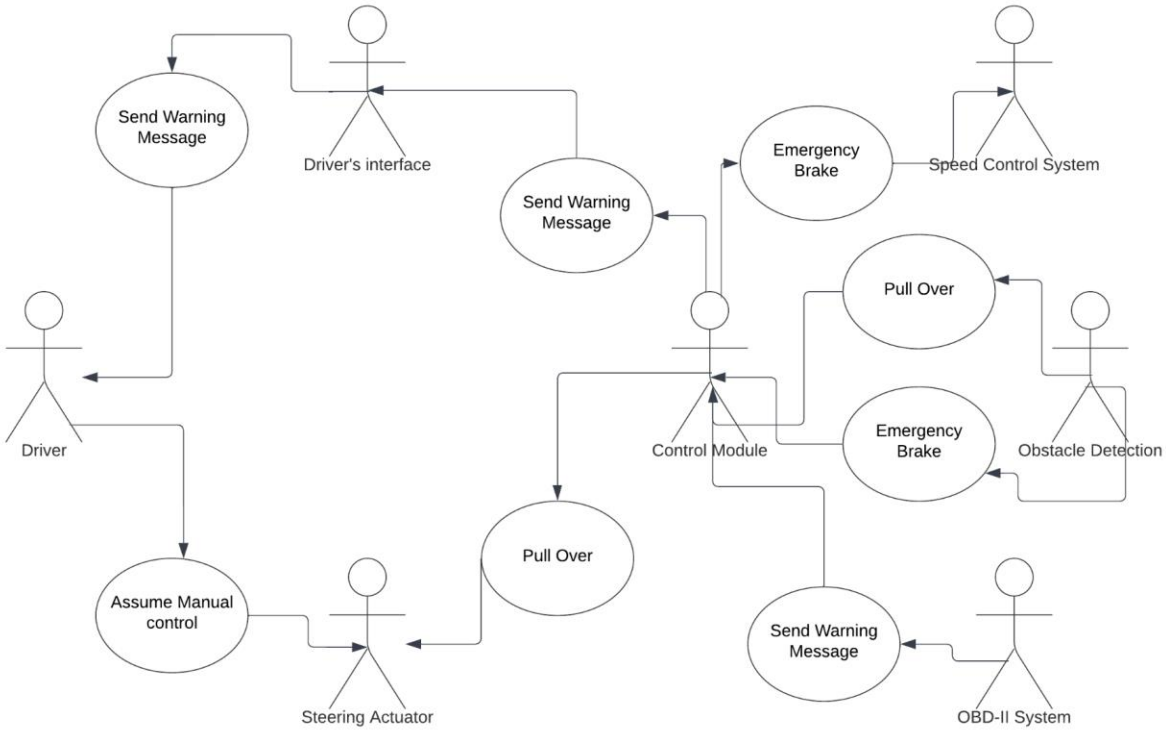


Figure 4.2 Use-case Diagram Part 2, UCs:15-17

4.1.1 Select Route

Brief Description: This use case allows the driver to select a destination and to select an available route to the destination for the NASS system to follow. The actors in this use case is the driver, the gps system, the command module, and finally the drivers interface.

4.1.2 Turn on/off Automatic Steering:

Brief Description: This allows the driver to initiate the deactivation of the NASS system and the reactivation of it. The main actor is the Driver in this use case, with the Drivers interface actor being affected by it.

4.1.3 Updating Navigation Display:

Brief Description: This use case allows the NASS system to update navigation display in the drivers interface. The GPS system updates the cars current velocity and position to the command module, and every 4th successful update the command module will send a message to drivers interface to update Navigation Display. The main actor is the GPS Receiver, the command module, and the driver's interface

4.1.4 Send Velocity Position Message:

Brief Description: This use case allows the GPS receiver to send updates on the vehicle's current position and velocity to the control module. The actors are the control module and the GPS receiver.

4.1.5 Driver Assumes Manual Control:

Brief Description: This use case allows for the ceasing of the NASS upon the detection from the Steering Actuator of the Driver manually steering the vehicle. The main actors are the driver and the Steering Actuator.

4.1.6 Select Destination:

Brief Description: This use case allows the Driver to select a destination and send it to the GPS Receiver. The main actor is the driver and the GPS receiver.

4.1.7 Suspend Automatic Steering:

Brief Description: This use allows for the suspension of the NASS upon a state change message of Suspend from the Driver's interface. The main actors are the Driver's interface and the control module actor.

4.1.8 Resume Automatic Steering:

Brief Description: This use case allows for the resumption of the NASS upon a State Change message of Resume from the Driver's interface. It will then check for two things, one is if the driver has stayed on the same route, nothing will change, if the driver deviated from the route a Select Route Message is sent to the driver. The main actors is the Driver's interface and the control module.

4.1.9 Managing Speed:

Brief Description:This use case allows the speed control system to send messages to the control modules on the current vehicle speed, and receive messages from the control module on speed change requests. The main actor is the Speed Control system and the Control module.

4.1.10 Turn/Change lane request:

Brief Description: This use case allows the control module to send lane change and turn request to the steering actuators. The GPS receiver affects this use case as the selected route determines when the vehicle needs to

turn or change lanes. In addition the lack of obstacle warning from the obstacle detection system means it's safe to turn or change lanes.

4.1.11 Send Obstacle Warning Message

Brief Description: This use case allows the obstacle detection to send an obstacle warning to the control module. This may result in a manage speed request to slow or stop the car, or a change lane/turn request to avoid the Obstacle. The actors for this is the Obstacle Detection system and the control module.

4.1.12 Send Available Routes:

Brief Description: This use case allows for the Control module to send a message of available Routes to the Driver's interface for the driver to choose from. The actors in this use case is the Control Module, Driver and the Driver's interface.

4.1.13 Send Selected Route:

Brief Description: This use case allows the driver's interface to send the selected route to the command module. The actors are the drivers interface and the control module.

4.1.14 Send cease automatic Steering message:

Brief Description: This use allows for the Steering actuator to inform the Control Module that the Driver has assumed manual control of the Vehicle. The actors in this use case are the Driver, Steering Actuator, and the Control Module.

4.1.15 Emergency Brake

Brief Description: This use case allows the NSS system to apply brakes on the Vehicle if it detects an imminent collision with an object ahead of it. The actors for this use case is the collision detection system, the command module, and the speed control system.

4.1.16 Pull Over

Brief Description: This use case allows the vehicles to pull up to the side of the street if it detects the presence of an emergency vehicle with its siren on. If the system can't find a path to move out on it will send a warning message for the user to take over driving. The main actors for this use case is the collision detection system, the command modules, and the steering actuator.

4.1.17 Warning Message

Brief Description: This use case allows the NSS system to send a warning message to the driver interface if an error occurs in the NSS system, and it will make a distinct sound telling the driver to take over the vehicle. The main actors involved are the OBD II system, the command module, and the drivers interface.

5. Logical View

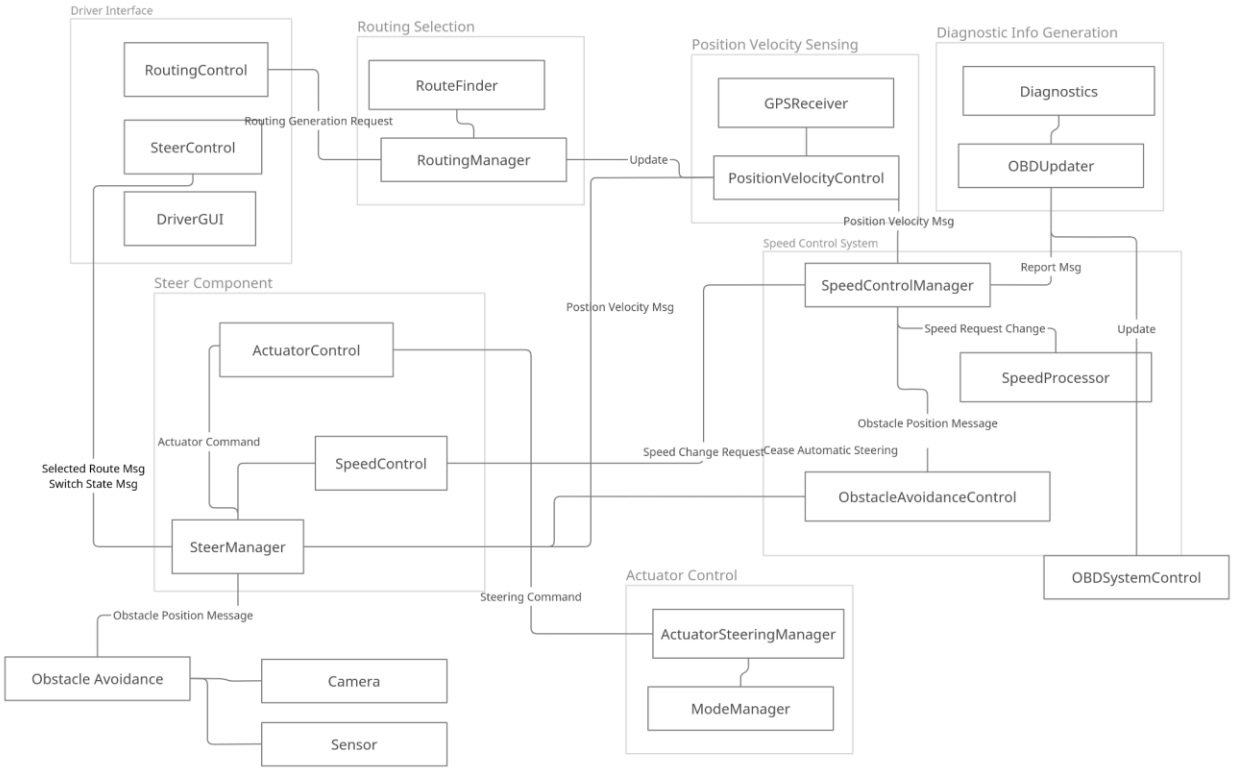


Figure 5.1 Logical View Component Diagram

The NASS Control Module is the primary component that establishes relationships between components, and also starts the mapping process among software and hardware such as sensors, cameras, and ECUs. There is currently no standard infrastructure for the Autonomous Vehicle System, BTA is designed following high-level functional components of standard ROS architecture: perception, planning, and control.

Perception utilizes detection hardware to create environment models in real-time. Multiple technologies are used to support the mapping process, and BTA perception is implemented using monolithic cameras and ultrasonic radars. The details of the hardware will be displayed during physical view.

Planning happens when the environment model and route selection information are both available. This phase can be divided into 3 tasks which can be available route selection (get the 3 best routes from current location to entered destination), behavior planning (evaluate the known obstacles such as high school zone, highway ramp, intersection), and trajectory generation for the navigation dashboard.

Control refers to the process of executing computational operations involved in both Perception and Planning. Control is end-goal focused, since it dictates the safety and performance of autonomous steering and acceleration. The output will decide the timing and magnitude of steering actuators and speed change accordingly to the given route selection and environment.

5.1 Architecturally Significant Design Packages

5.1.1 Vehicle Optic

The development of IoT technology in vehicles has become very popular which desires a more advanced adaptive system overtime. There are two directions that will affect the entirety of the system infrastructure: distributed vs. centralized. BTA is designed following the centralized architecture, which simplifies the coherence of the system and enhances processing time. Distributed systems can offer better parallelism in terms of performance, but that can be compensated with higher computing units.

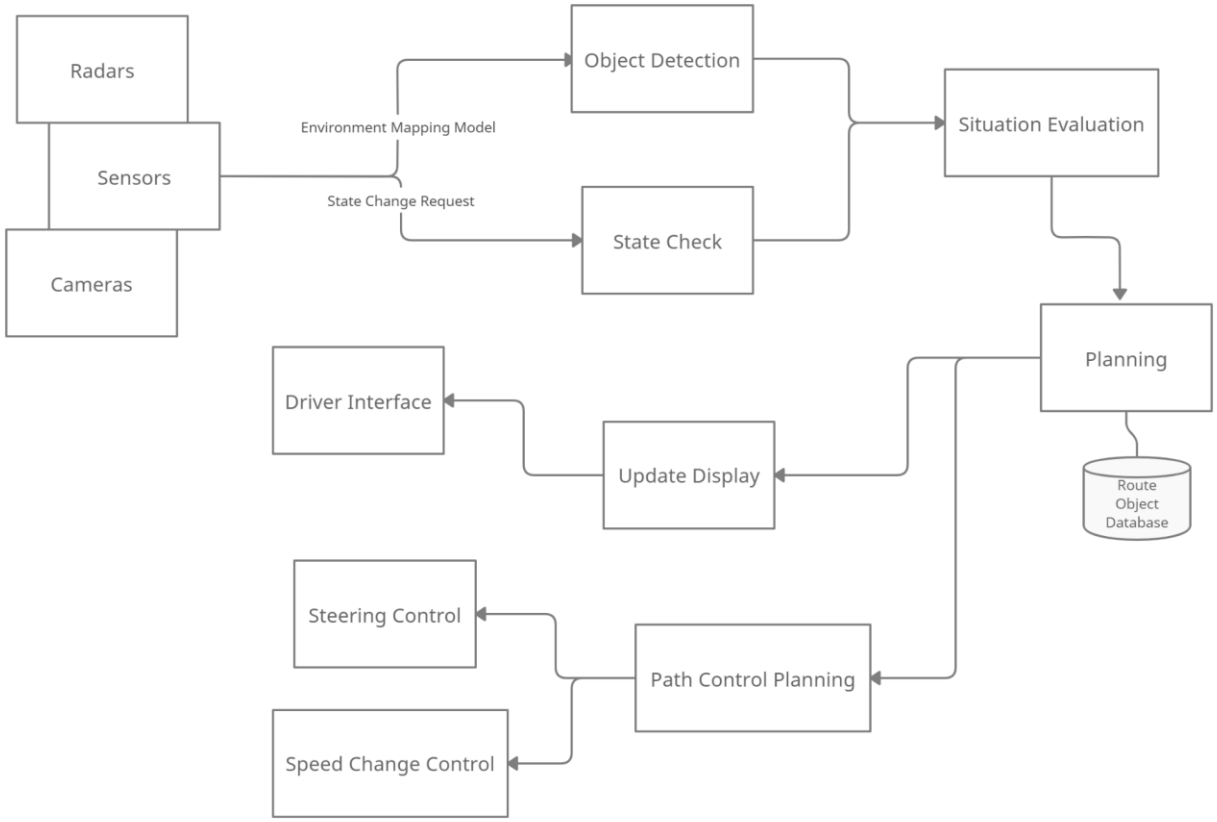


Figure 5.2 Logical View Component Diagram

In order to ensure the most safety for drivers, the cycle of environment mapping and evaluation of mapping scenarios will be happening in real-time. The obstacle

sensing process can be decomposed into 2 subtasks: Localization and Perception. The combination of their outputs will be the vision for the software optic.

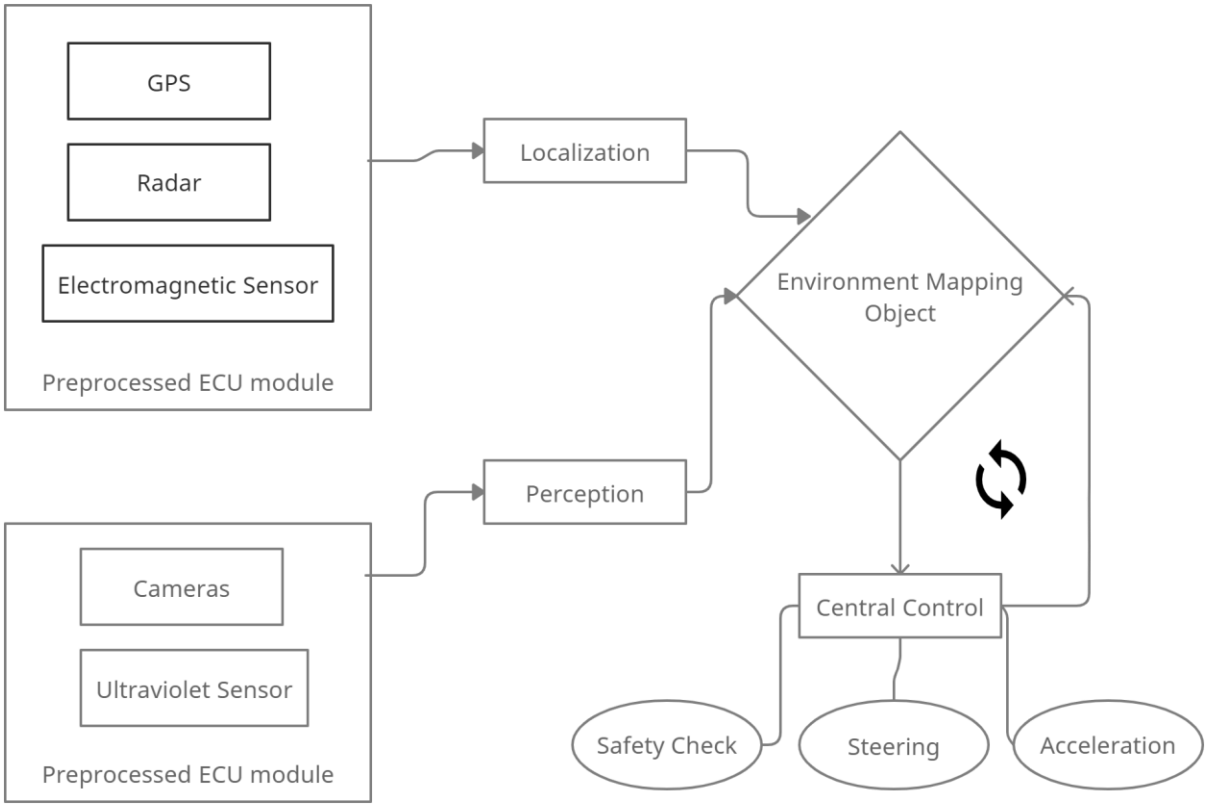


Figure 5.3 Real-time Perception Workflow Diagram

5.1.2 Advanced Navigation System

The on-board navigation system will be embedded with Route Selection which enhances driver experience and performs trajectory modeling for the selected route. The module will allow drivers to enter destination from in-vehicle driver interface and discover the best path from the current GPS location to the selected address. Route Selection encompasses both online and offline route search, OTA map updates, real-time traffic information, optimization guidance (i.e. toll or toll-free).

World Mapping returns trajectory modeling object and turn-by-turn direction for In-vehicle Interface. Following standard mapping software architecture, the module will be decomposed into 4 submodules: area search, navigation tracking, maps rendering, and segments computation. The four modules follow monolithic design and are loosely coupled with Cassandra to manage worldwide geospatial data.

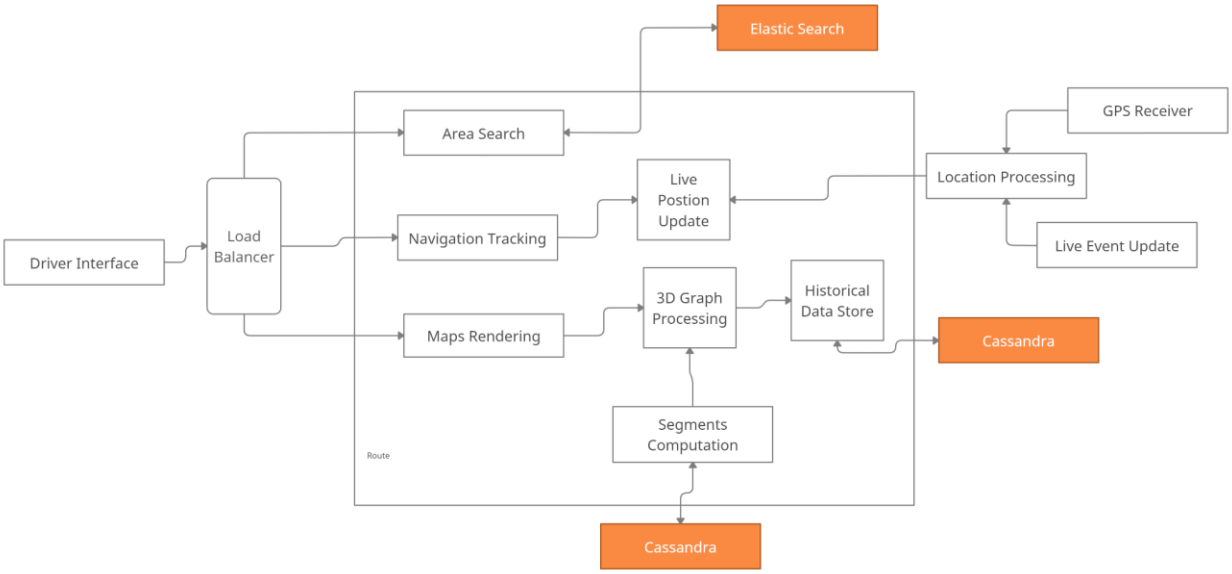


Figure 5.4 Real-time Perception Workflow Diagram

5.1.3 Connectivity Manager

Vehicles that are opted in for BTA Control System will be added a built-in cellular data module that connects to adaptive bandwidths from in-house remote internet service.

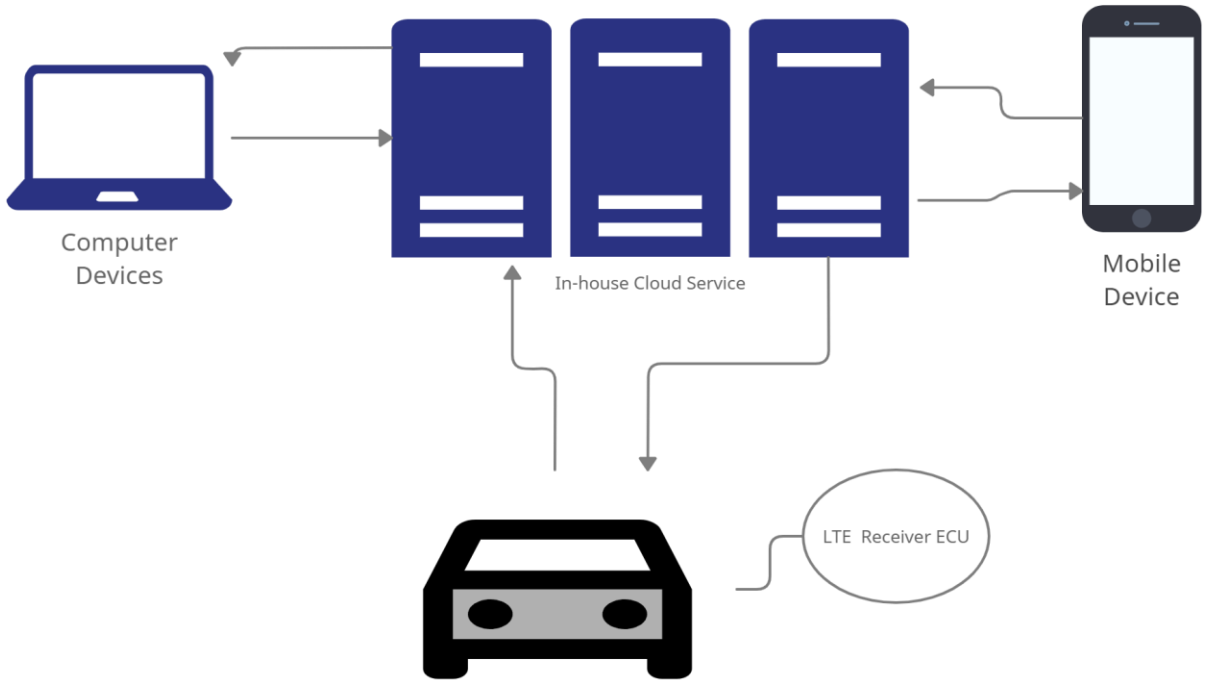


Figure 5.5 Connectivity Diagram

5.1.4 In-Vehicle Experience

BTA comes with Driver Dashboard Interface that allows access to services that it provides. The rendering process uses on-module data to produce changes and updates for the driver.

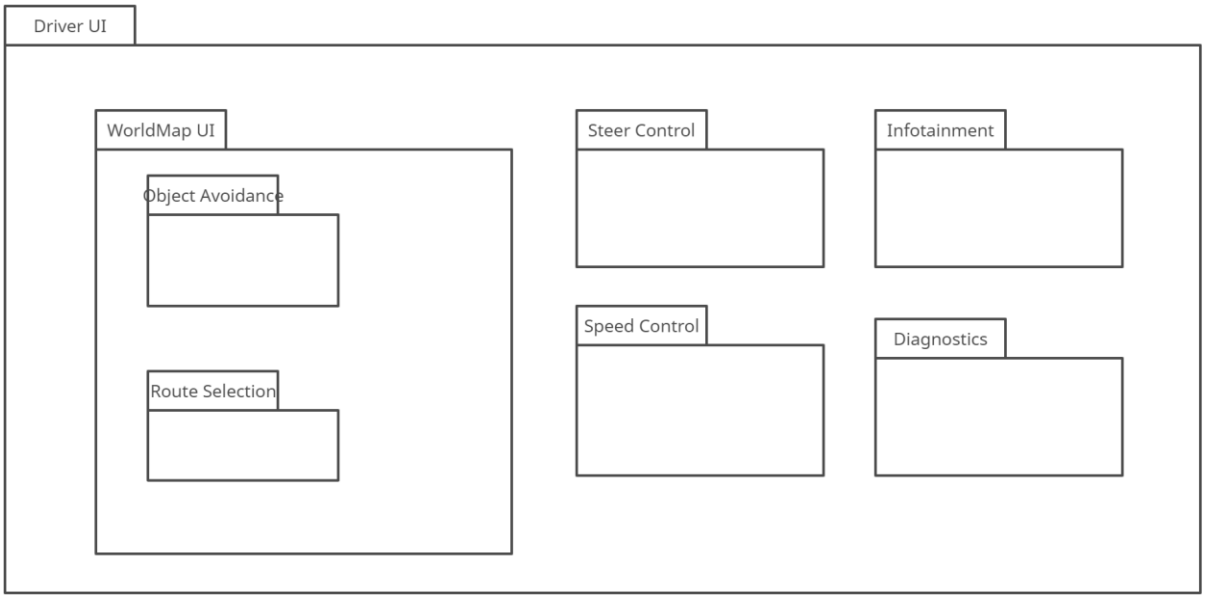


Figure 5.6 DriverUI Package Diagram

5.1.5 Central Control Module

1. *Overview*

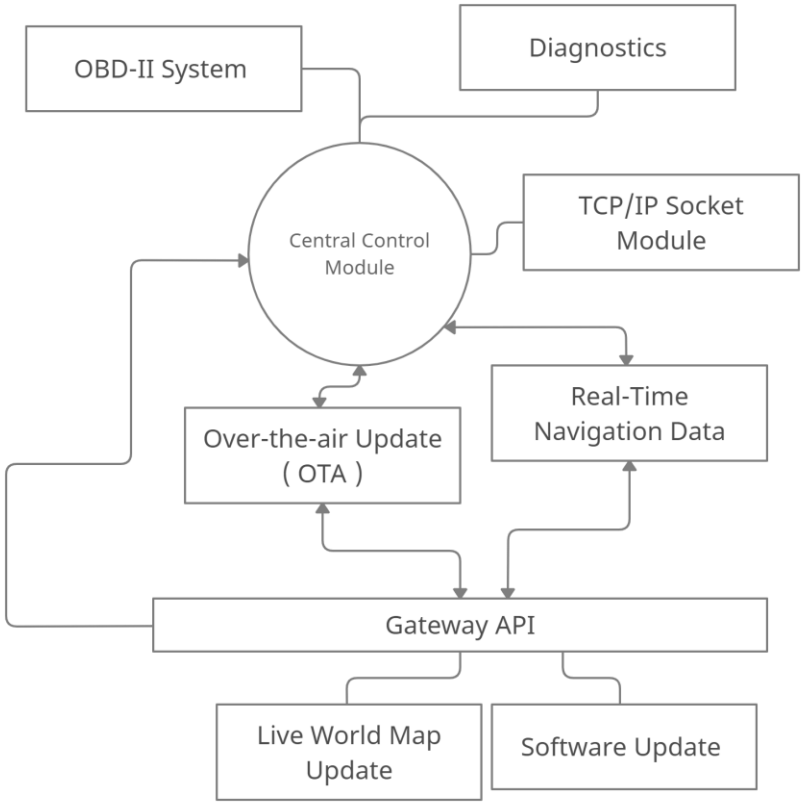


Figure 5.7 Central Control Module Diagram

Central Control will encompass diagnostics, OBD-2 report, communication with application server, steering and speeding control. The status of the module will be managed by states.

2. *State Definition*

0 - OFF : Car is OFF.

1 - ON : Car is ON and NO ROUTE is selected.

2 - ACTIVATE : Car is ON and A ROUTE is selected.

3 - SUSPEND : Car is ON and Suspend Msg received.

4 - RESUME : Car is ON and Resume Msg received.

5 - CEASE : Car is ON and Cease Msg received.

a. Definition

0 - OFF

+ Not connected to Application Server.
+ Steer and Speed Control not engaged.

1 - ON:

+ Connect to Application Server.
+ Wait for Available Route Selection Response from Application Server.
+ Update Navigation Display.

2 - ACTIVATE:

+ Speed and Steer Control engaged.
+ Update Navigation Display: route selection map, gps, speed.

3 - SUSPEND:

+ Update Navigation Display: ‘take control’ alert, gps, speed.
+ No Speed and Steer Control.

4 - RESUME:

+ Enable Speed and Steer Control.
+ Update Navigation Display: route selection map, gps, speed.
+ IF A Route Currently Selected:
    + Change state to ACTIVATE
+ ELSE:
    + Change state to IDLE

5 - CEASE:

+ Update Navigation Display: status.
+ Disengage Steer and Speed Control.
+ Alert driver to restart the system.
+ IF Car is turned OFF or Restart key is triggered:
    + Change state to OFF
+ ELSE:
    + Wait for turning OFF

b. Diagram

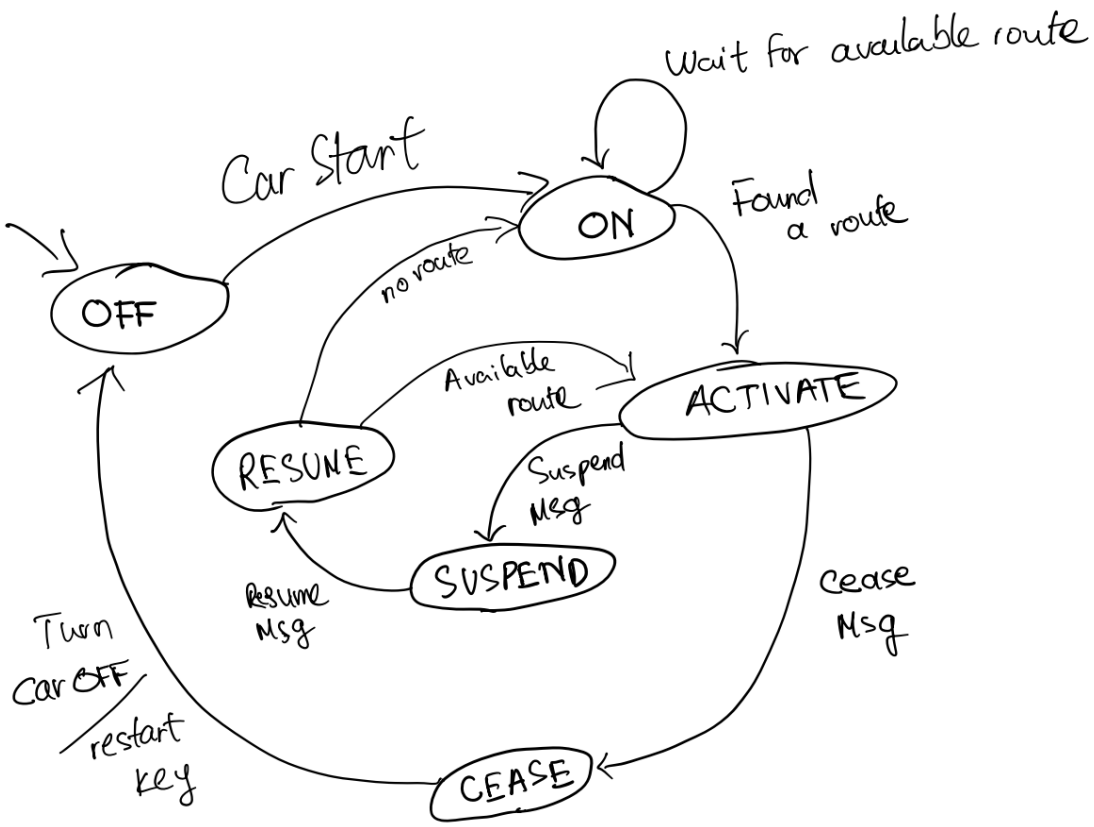


Figure 5.8 Control Module State Diagram

5.1.6 Steering Control

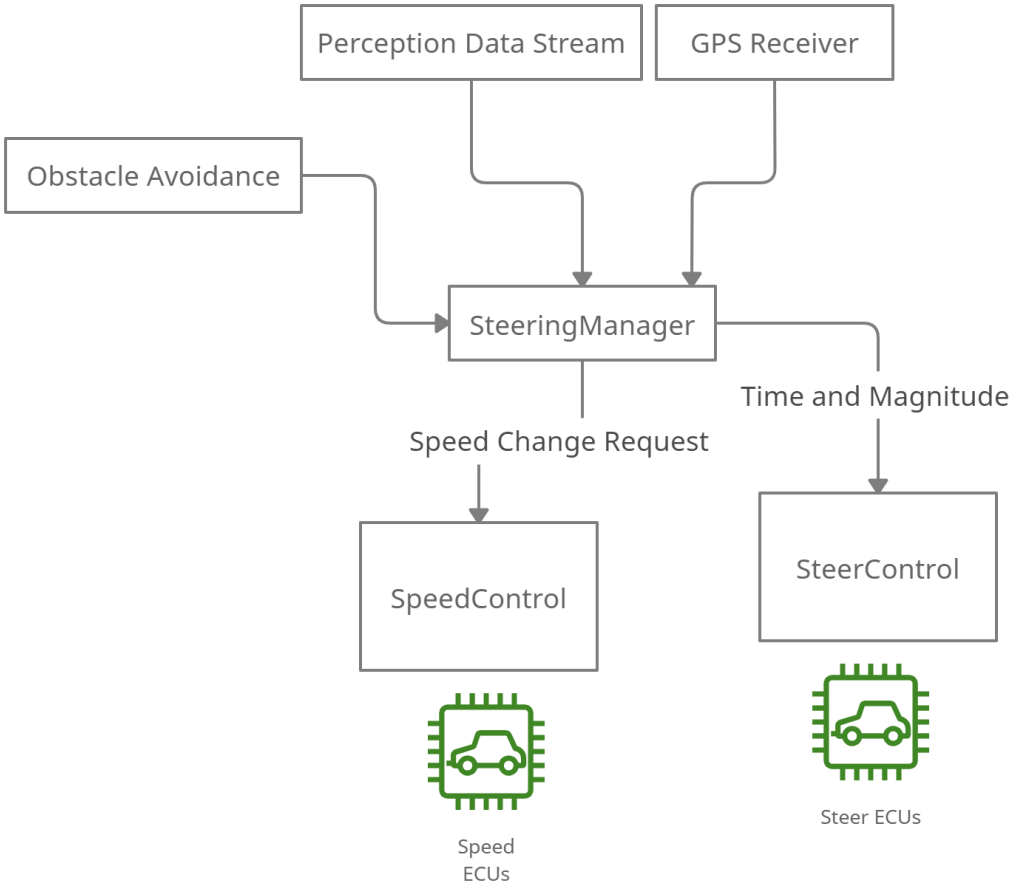


Figure 5.9 Control Module State Diagram

GuidingControl will dictate the behavior of two physical functions: steering and speeding. The module encompasses 3 main submodules that will perform complete functionality of steering automation:

- SteerControl
- SpeedControl
- SteeringManager

SteerControl will manage the acceleration force and rotation angle signal on the steering actuator which directly affect on-road steering and car wheel turns. When being held by the driver, SteerControl will send a message to switch NASS state to Suspend.

SpeedControl will adjust the speed in real-time based on TCP/IP requests from Vehicle Optics Module. Suspend State will disable SpeedControl until Resume.

SteeringManager is the primary interface that connects with Position and Velocity Sensing, SpeedControl, SteerControl, and Route Selection.

5.2 Architecturally Significant Design Classes

Following monolithic design, the central control module will handle all requests and distribute them accordingly to other classes that are responsible for the task.

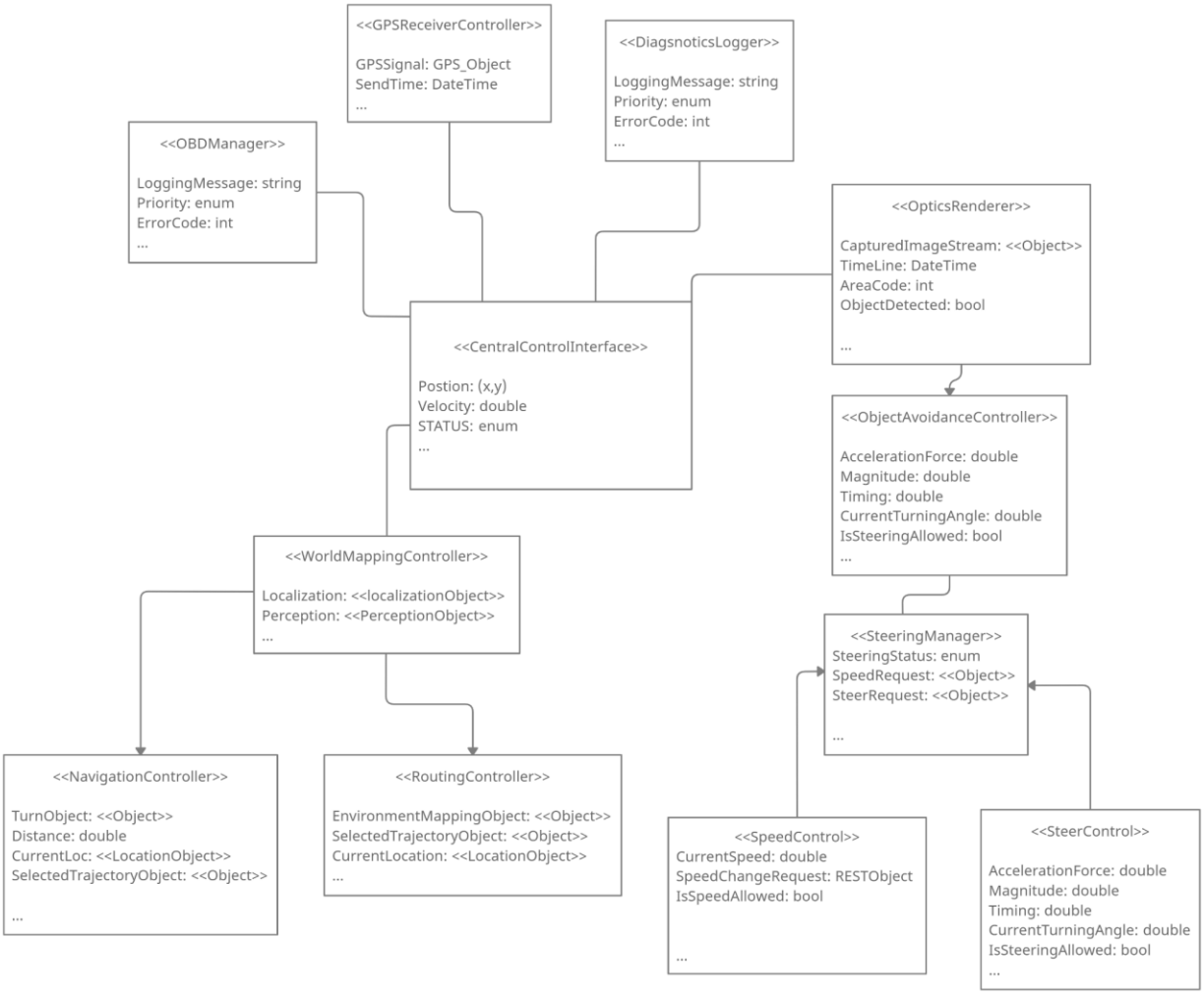


Figure 5.10 Control Module State Diagram

6. Implementation/Development View

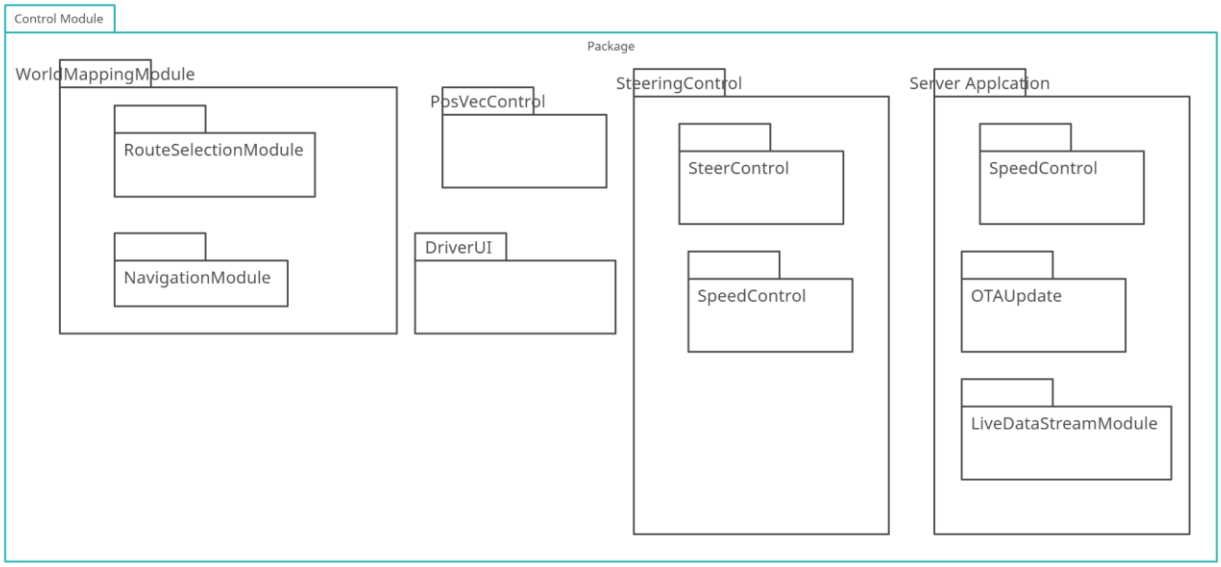
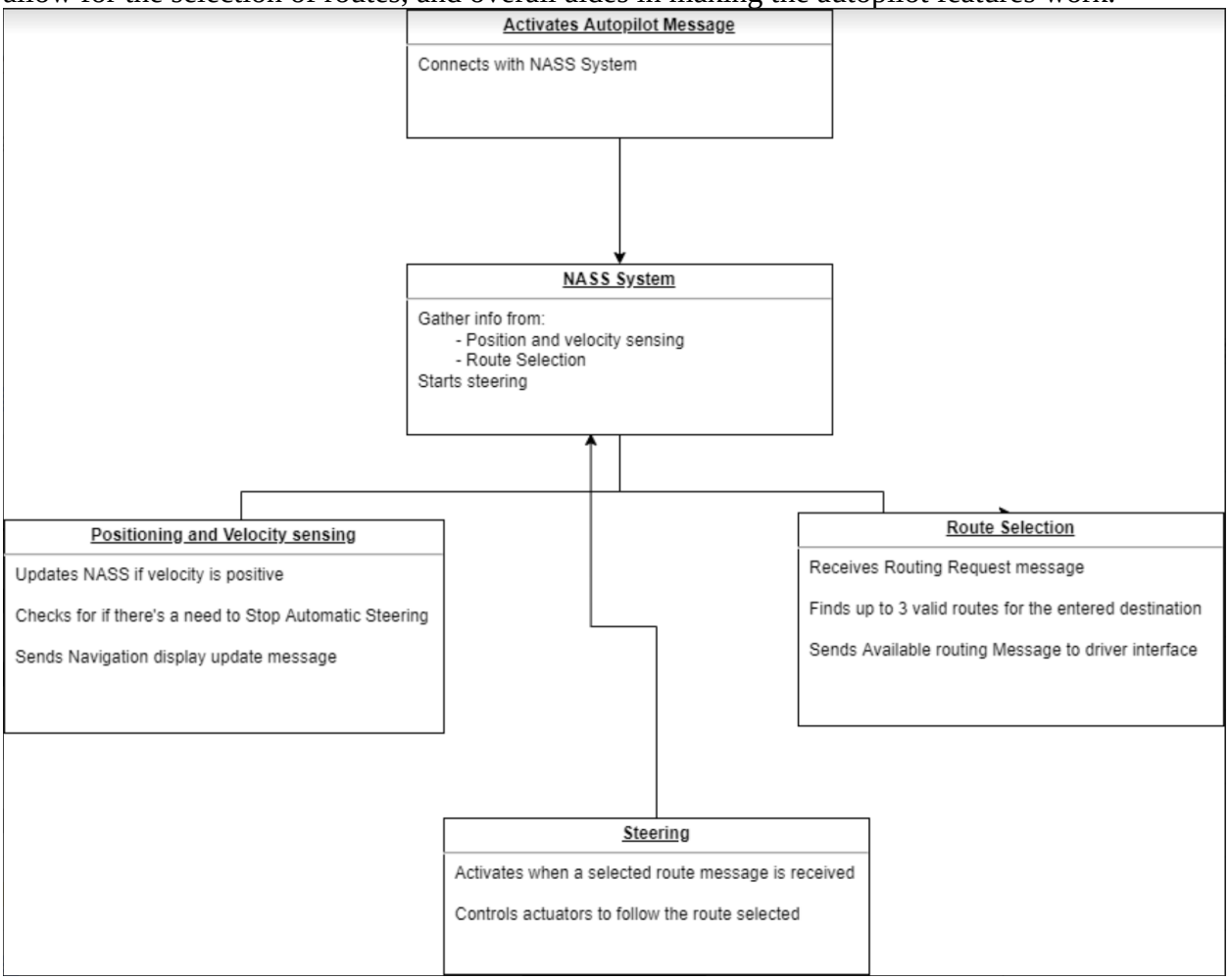


Figure 6.1 Control Module State Diagram

The Development Process will initiate with developing generic tasks for 4 main modules: WorldMapping, SteeringControl, Server Application and GPS Control.

7. Process View

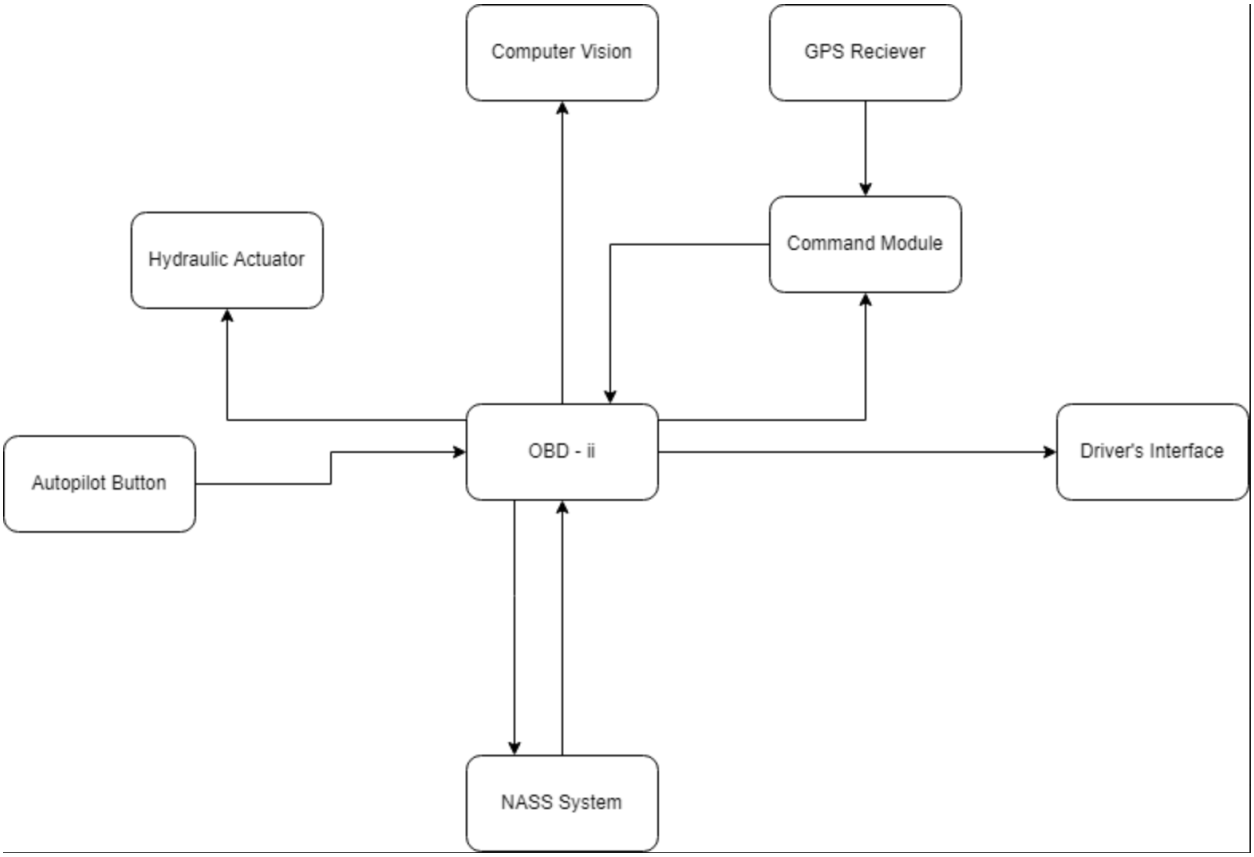
The Process Model illustrates the NASS classes as executable processes. Processes exist to help prevent the vehicle from speeding, avoid obstacles, give the driver the car's current location, allow for the selection of routes, and overall aides in making the autopilot features work.



7.1 Process View Diagram

The Central Control Module is the brain of the system. Most requests are being sent into the Central Control Module. Requests such as Speed Change Requests, Routing Request Messages, and Position update are all sent to the NASS System. Once the NASS System receives the messages, it will send out messages to the required parts of the car in order to make the necessary changes.

8. Deployment/Physical View



8.1 Deployment/Physical View Diagram

GPS Receiver: To handle position and velocity a GPS Receiver will be used in conjunction with a command module. Sends updates on the vehicle's current position and velocity to the control module.

Command Module: The command module will handle creating speed change requests as well as Route Change requests and acts as an intermediary between all components.

Computer Vision: Utilizing Lidar, the Computer Vision (Obstacle Detection System) ODS focuses on crash avoidance. ODS can use Lidar or cameras and build a scene in the system to detect obstacles and their distance from the vehicle. The ODS connects to the control module.

Hydraulic Actuators: Handles the flow of gear shifting and computes necessary changes to energy transfer between wheels and the vehicle. Interacts with the Engine Control Module (ECM) to manage various components related to engine control to ensure

quality of function. Also interacts with the Powertrain Control Module (PCM) to compute functions using input from sensors to trigger error checks.

Driver’s Interface: The Driver’s interface will showcase information the user will need.

Autopilot Button: Part of the driver’s interface, this component allows the toggling on or off of the BTA system.

NASS System: The NASS Control Module is the primary component that establishes relationships between components, and also starts the mapping process among software and hardware such as sensors, cameras, and ECUs.

OBD - II: The On-Board Diagnostic system is a California mandated component tracking every component that affects emissions.

9. Size and Performance

The NASS system supports the follow size and performance constraints:

- The NASS system should be able to generate available routes for auto-steering at least 95% of the time.
- The NASS system should be able to receive at least 3 position velocity messages in under 2 seconds.
- In the event of an unavoidable collision the system shall cease automatic steering in 100ms or less.
- The NASS system shall slow to 5mph for a 180 degree turn, 15 mph for a 90 degree turn, and 35 mph for a 45 degree turn.

The directly connected NASS components help make sure that the NASS system can quickly communicate between components to perform their required process.

10. Quality

The architecture of the NASS contributes to improving the security, availability, reliability, and portability.

10.1 Reliability

Description: The system shall have a mean time between failures of no less than 12 months. A failure is defined as any error except GPS Signal error that causes a cease of automatic steering or unsafe behavior.

Solution: Navigation Display Update message will accurately measure vehicle position 99.9999% of the time. Route generation will be accurate 95% of the time. Steering, Throttle, and braking actuators will be accurate 99.999% of the time. The OBD-II system shall be accurate 99.9% of the Time. The NASS shall run with less than 3 errors per 5 seconds, 99.9 % of the time. Upon 3 or more error messages being received to the command Module within 5 seconds, the system shall notify the User via driver's interface and disengage the NASS in 5 seconds.

10.2 Portability

Description: How the NASS can be reused in different Vehicles Environments.

Solution: NASS can be run and installed by the Sonny Motors Company in Standard vehicles that fit the requirements listed by Sonny Motors Company. A layered architecture will be used to abstract business logic from hardware design. The transversal business logic shall be designed to function on any hardware design.

10.3 Security

Description: CAN/ECU security, VANET security, Intrusion detection System.

Solution: implement CAN/ECU and VANET security, in addition to a intrusion detection system. The CAN/ECU security can be implemented in a variety of ways such as secure boots and updates. In addition the method considers real time traffic in the vehicle network, CAN frame structure and limited resources of the ECU. The ECU security can then use a message authentication code to provide guarantees of integrity and reliability. In addition, delayed data authentication is introduced to prevent interruption of real life traffic. VANET security can be

implemented to secure blue tooth and online connection through the Vehicle from attacks such as impersonation, modification, and offline password-guessing attacks. An intrusion detection System can provide a solution to basic network security solutions, firewall and intrusion detection systems into the autonomous vehicle [1].

10.4 Availability

Description: GPS availability, NASS error detection mechanism, NASS automatic steering suspension/resume.

Solution: The NASS will be available for the driver to turn on and off via the Driver's interface control settings. Upon multiple errors being detected by the System, it shall be suspended until the errors are eliminated. If the Driver were to manually take control of the vehicle the NASS will no longer be available for use until driver interaction ceases.

11. Appendix

Appendix A: Architectural Design Principles

Architectural Design Principles that will be used for the creation of our product Better Than AutoPilot or BTA include: Buy rather than build, reuse, and Single point of view, and separation of concerns..

A.1 Buy rather than build

Many features included within the Software Architecture Document such as the security features can be procured via third party products to ensure more tried and tested features are used. As a more widely known security system will have gone through a much more thorough testing period, and have a much longer standing exposure to potential attackers, thus making them more effective than building an intrusion detection system from scratch. Thus buying certain parts for our project can prove to be much more effective than building them. This can also include the cost as well, depending on how long it would take to develop those segments ourselves.

A.2 Reuse

Hardware components used within the Sonny Motors Company NASS system will have had prior uses within non-autonomous vehicles that can be reused when designing the NASS. Such components can include the steering actuators, obd-ii system, Command Module and Driver Interface. Reusing these hardware components and changing our software to make them work for our product can help lower costs on design.

A.3 Single point of view

BTA will adhere to a single point of view, and will combine a multitude of data Sources, such that it appears as though it has only one source. This will be done by linking several features and functions of our SAD to specific components that encompasses that data. Such an example will be linking Obstacle detection system features onto the LIDAR device as the LIDAR's sensors and cameras are required to detect the system.

A.4 Seperation of concerns

Implies that components should be responsible for a cohesive set of business focused Tasks. BTA will be separated along a horizontal Separation of Concerns where the NASS will be divided into logical layers of functionality.

Presentation Layer	Driver's Interface On/Off automatic steering
Business Layer	Acceleration Actuator Breaking Actuator Resume Automatic Steering.
Resource Access Layer	Internet connectivity Route generation GPS monitoring

The above diagram is a simple example of Separation of Concerns, where it is divided by the Presentation Layer, Business Layer, and Resource Access Layer. The Presentation Layer encompasses the process and components related to the user interface and govern components such as the Driver's interface. The business layer encompass the components or processes related to the NASS domain. Such examples include the actuators and features such as Resume Automatic steering which represent the logical flow/actions the NASS will undergo on use. The Resource Access Layer governs processes and components related to the access of external information, such as data access, internet connectivity, GPS monitoring and more from the NASS.