

## Software Design

### *The Invincibles*

*May 23, 2021*

#### **Important Notes:**

- *The descriptions in italics in this document (except for some section headings) are exemplary and explanatory and must be removed from the completed report.*
- *Identify which section of this report was created by which team member*

## 1 Product description

Our project consists of three main parts, which are communication, formation of platooning and driving. For each of them, we have planned some simulation.

These are some of the communications scenarios that might be simulated:

- Trucks can share information for their location and real-time positions.
- Trucks can share information for general street conditions, especially in difficult weather conditions.
- Trucks can share information for an emergency caused by external factors in a specific part of the street (Working canter, Infrastructure damage etc.)
- Trucks can share information for their directions and trajectory so the driving maneuvers can be done smoother.

List of processes that might be simulated in platooning:

- Trucks can join the platooning as a follower as well as a leader.
- At a specific point in time, a truck might need to leave the platooning.
- The leader selection is made by using a specific election algorithm.

List of processes that might be simulated in the driving domain.

- A brake simulation is performed when a user through the main controller gives this command.
- An accelerates simulation is performed when a user through the main controller gives this command.
- A full stop simulation is performed when a user through the main controller gives this command.

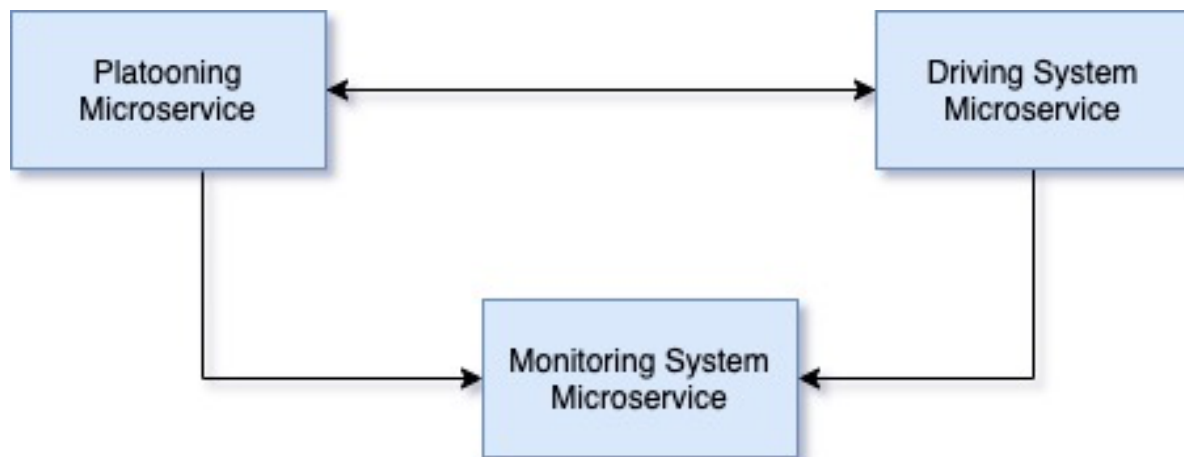
## 2 Technologies

Describe the technological approaches you will use to implement your project.

### **a. Architecture Patterns**

In this project, we will use the microservices architecture pattern.

We have identity three microservices :



Microservice Architecture by Tunde Aluko

**Platooning Microservice:** In platooning microservice, we would implement the join and leave functions. The platooning is responsible for the election algorithm of selecting a new leader. It handles the overview of the platooning system.

**Driving System Microservice:** We have the motion control function, that is, a function that controls the speed of the trucks to the desired results, and it includes the state of the trucks such as the fuel level, current speed, and others. This also implements the event channel communication by publishing the trucks states regularly so other components like the monitoring system can get the real-time status of the trucks.

**Monitoring System Microservice:**

In the monitoring system microservice, the platooning system's real-time status and driving system real-time status is being monitored. This microservice is essential for debugging faults and failures and checking the status of the systems in real-time.

**b. Programming Languages**

- All microservices will be programmed in C++ 11 or later versions.
- Local server side and querying will be programmed with Node-JS.
- In a dynamic web page we might use HTML, JavaScript, CSS

**c. Frameworks**

- Microservice Framework: C++ Microservice Framework, Docker(Maybe)
- Webservice Framework: NodeJS and Express.js)
- Website Framework: ReactJS, Bootstrap

**d. DevOps tools**

- CMake (To build the C++ project)
- Git (To for team collaboration)

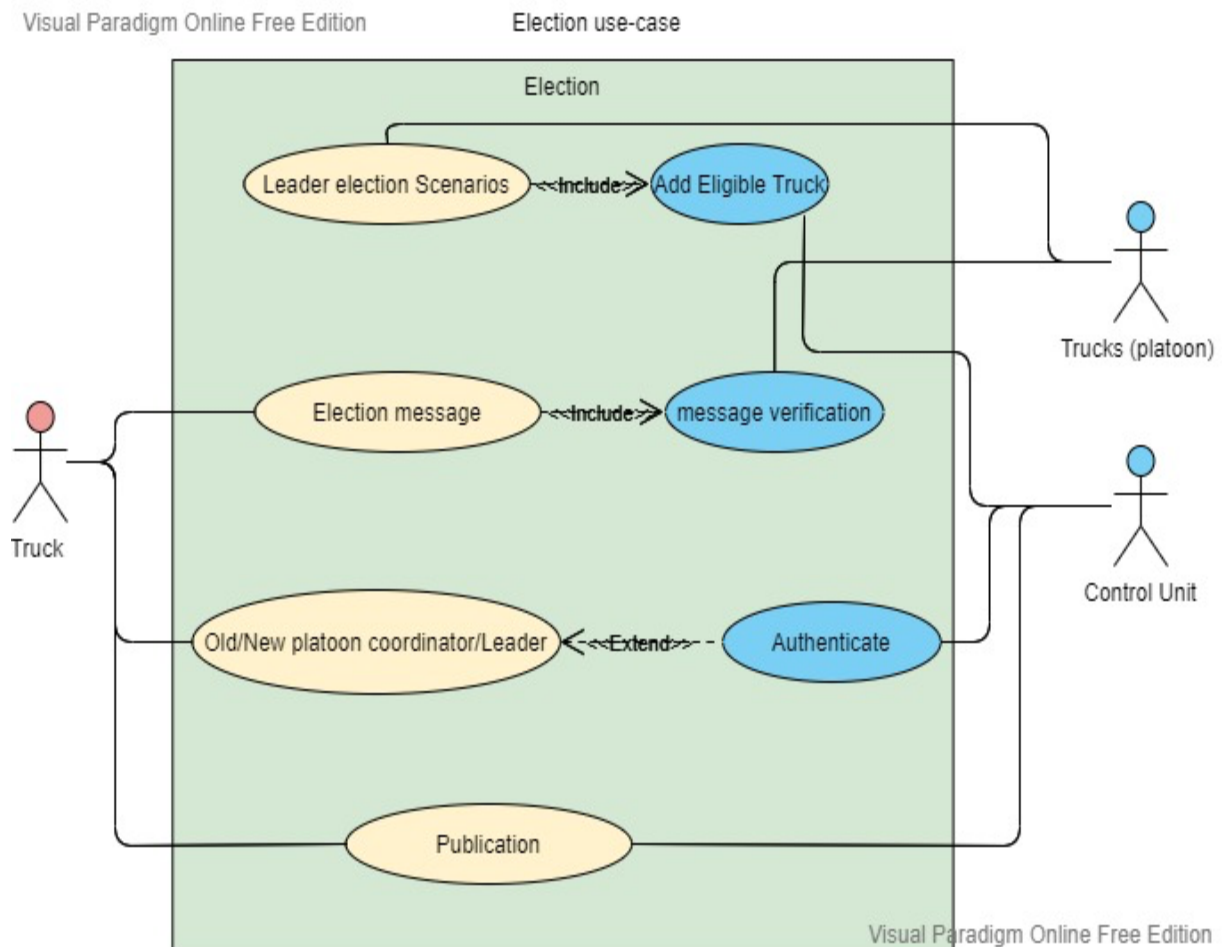
### 3 Use-Cases

**a. Truck Platooning System Use-Case**



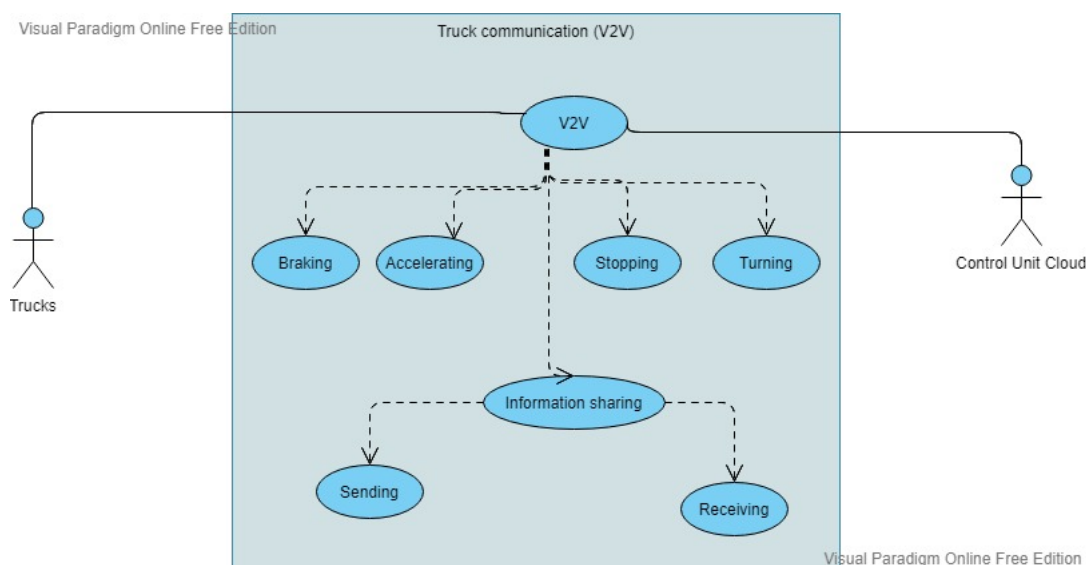
The use-case diagram above describes the truck platooning system scenario. The first truck starts the platooning and automatically becomes the leader. Other trucks can join or leave the platooning, and the leader truck. When the leader truck leaves, there will be an election to elect a new leader.

#### b. Election Algorithm Use-Case



### c. Communication Use-Case Diagram

In the communication use case is demonstrated how the trucks can share different information based on their needs. One case might be turning information in this case they can perform V2V communication with on board communication system if they are in range. If the trucks are not in range, they still can communicate, but this time through a cloud-based control unit. And in this case, we make sure that communication is established, and it is stable in our system.

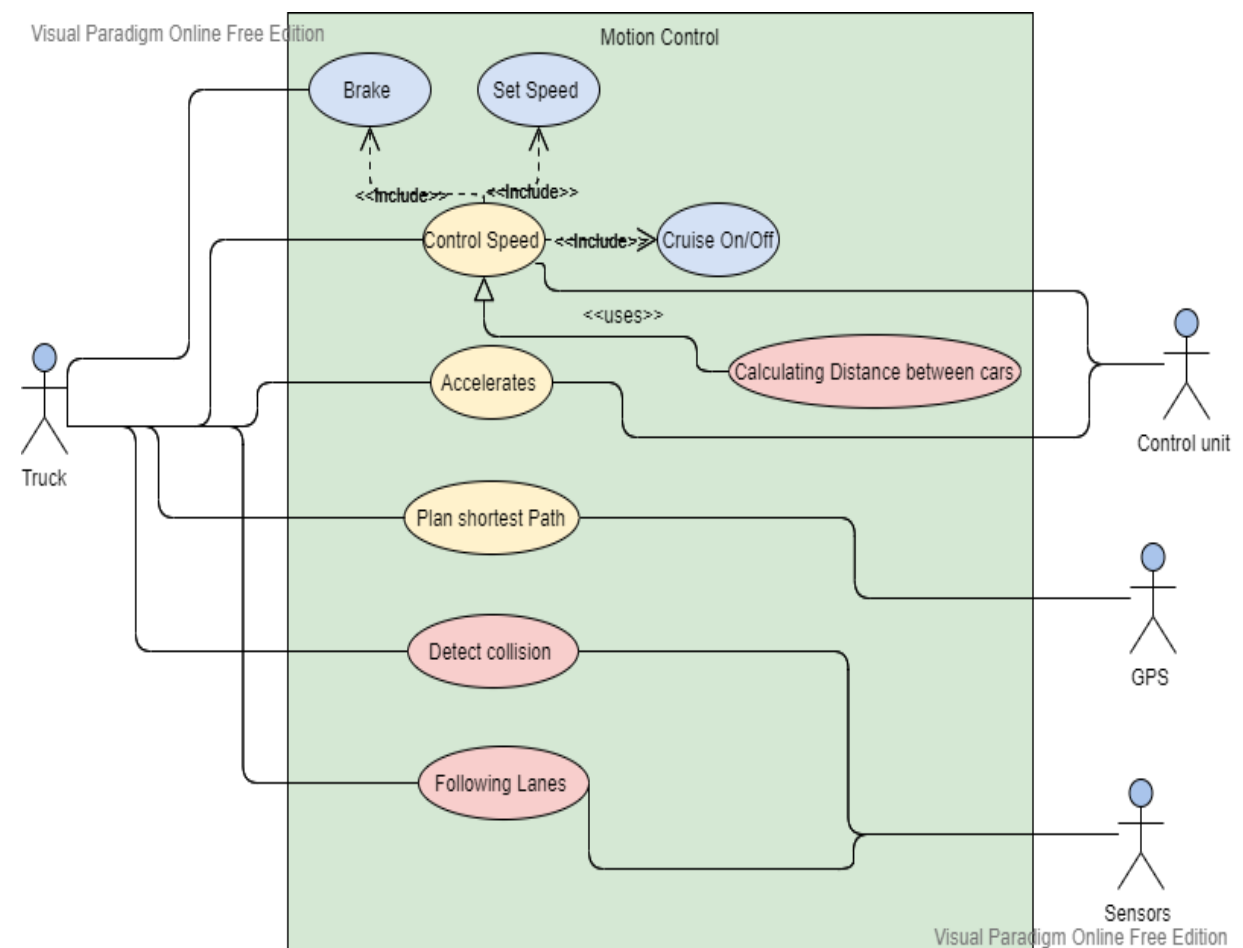


#### d. Motion Use-Case Diagram

Here we have the main actor which is the Truck, and secondary actors which are the Control unit, GPS, and Sensors.

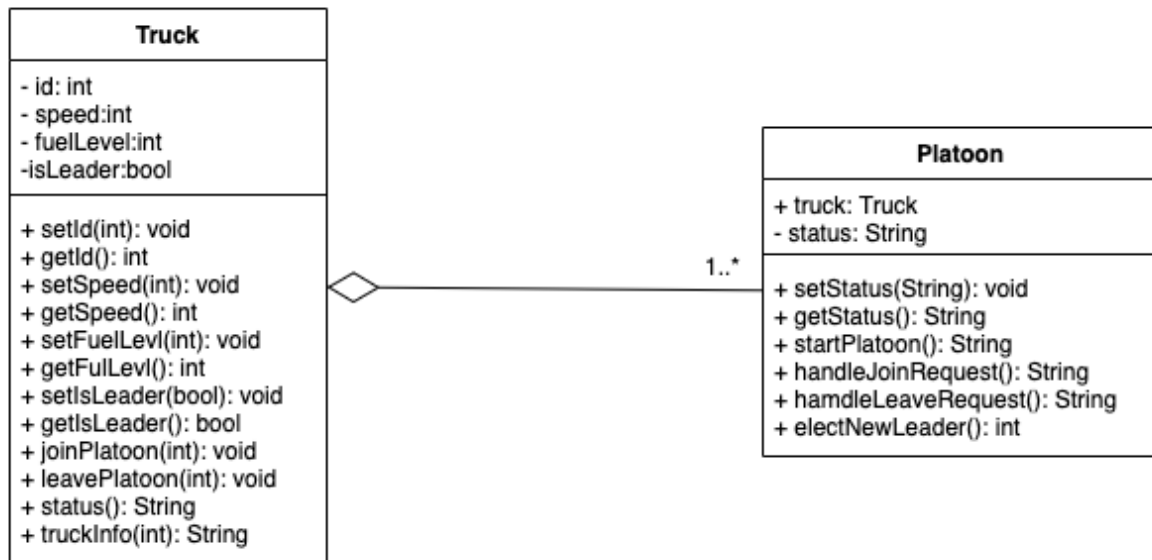
The events involved are as follows:

1. The speed is controlled by the car, which could brake, set speed and Cruise On/Off.
2. Control Unit then communicates the distance that should be between each car.
3. Then the Car accelerates. While moving the GPS with the aid of an Algorithm detects the shortest path possible for the platoon to get to its Destination.
4. Furthermore, we have sensors attached to the Trucks which can detect other objects and avoid collision.
5. Also, they can detect pavement markings and follow these lanes.



## 4 Structure of the software

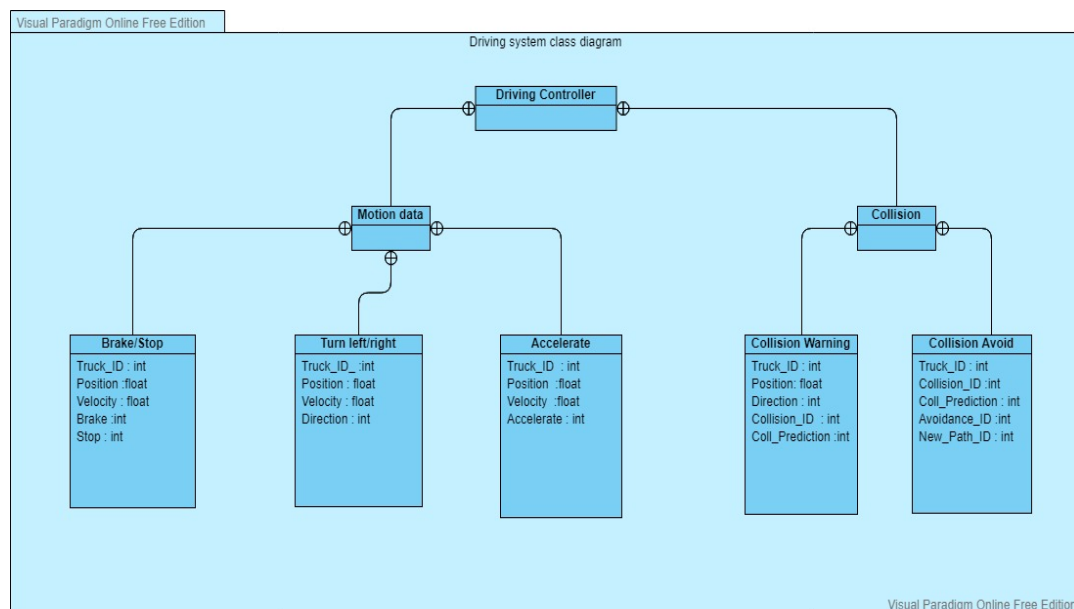
### a. Platooning Class Diagram



Platooning Class Diagram by Tunde Aluko

We have the class truck, and the class is instantiated in the platoon class. In the Platoon class, we handle the join and leave request to join the platooning service and the election method.

## b. Truck Driving System Class Diagram

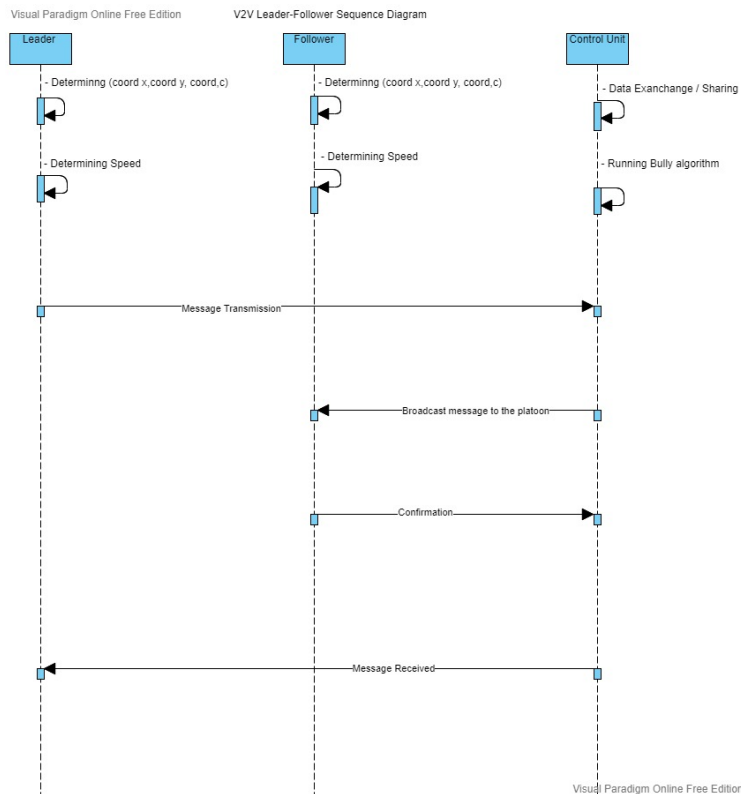


The main components in driving controller that might be simulated are motion data and collision. Motion has three main sub-classes which are braking/stopping, Turning left/right and acceleration. Each sub-class has its own attributes. The second class is

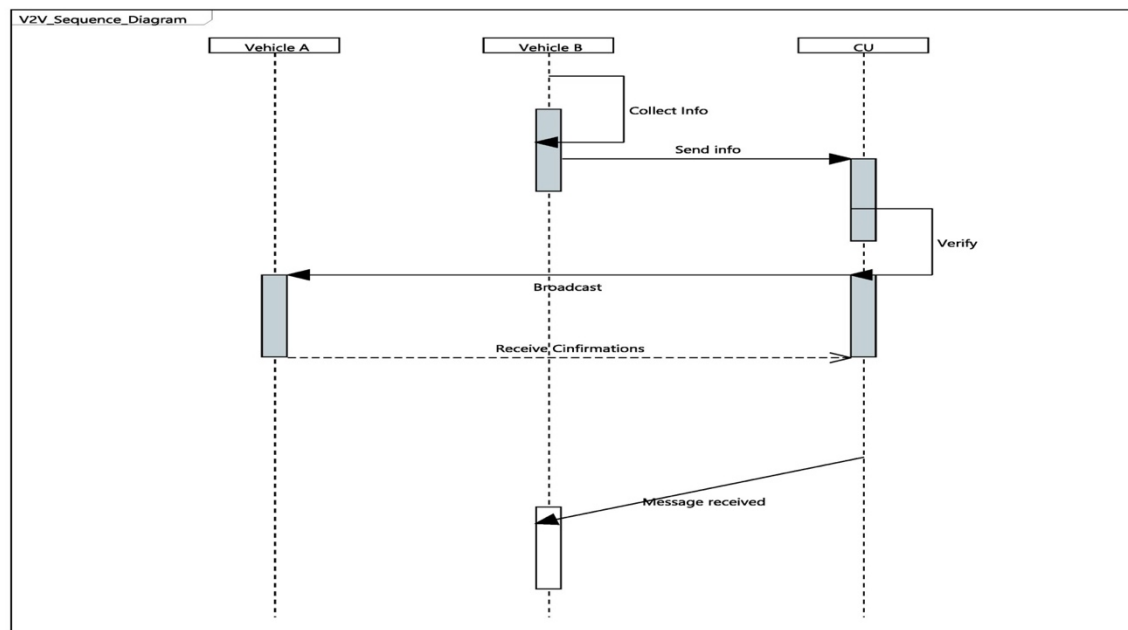
Collision which has two sub-classes. The sub-classes are collision warning and collision avoiding. Each sub-class has its own attributes as well.

## 5 Communication flow

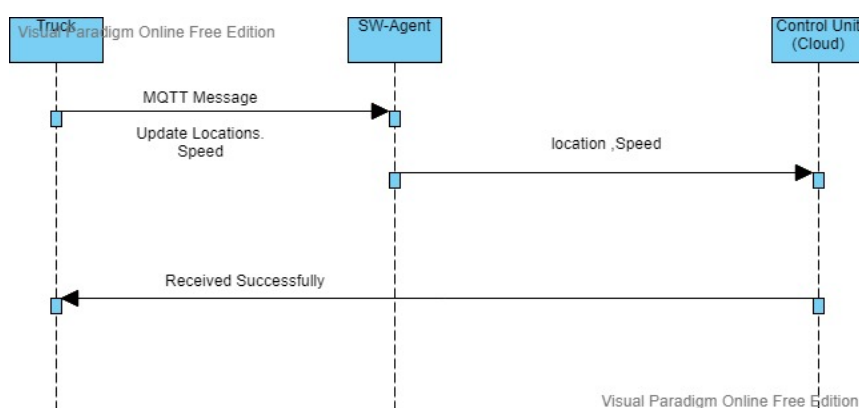
### a. V2V Sequence Diagram (Leader-Follower)



### b. V2V Sequence Diagram-Data share



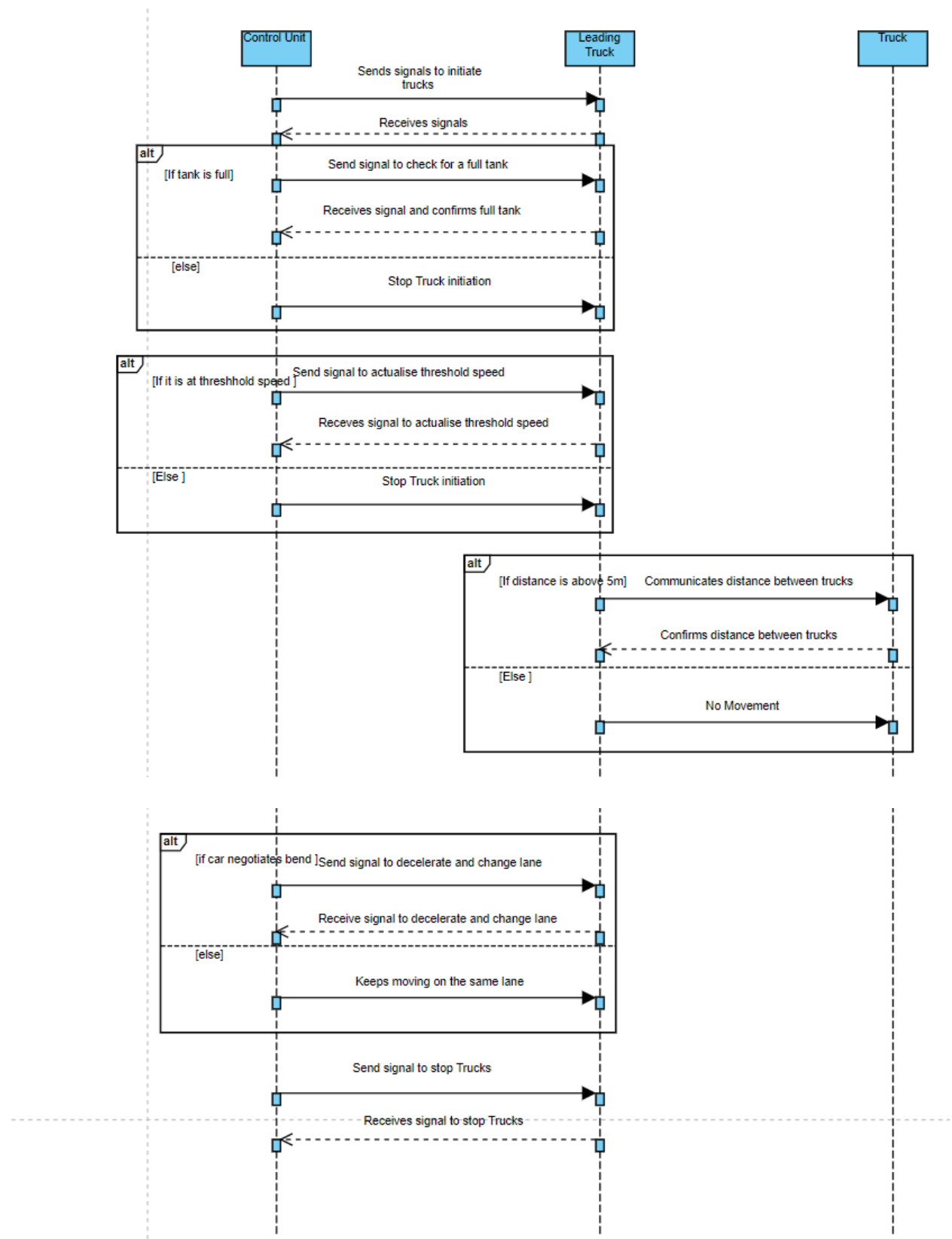
Based in the use case we developed two different sequence diagrams. In this case the communication is done through on-board communication system that each truck has. Truck B collect some information regarding a specific scenario (Speed, obstacles, Road signs, Position). This information is forwarded to on board control unit of the truck B. The information is processed and verified in the control unit. After this step the control unit can broadcast this information to Truck A. In the next step the truck A send a confirmation to control unit if the information was received successfully.



In the second sequence diagram the communication is done through a cloud based main control unit. In this case the truck sends information to the SW agent maybe an update in location or speed or another type of information. In the next step the SW agent deliver this information to the main control unit. The web-based control unit can now distribute the information to different trucks if they need this type of information. After the control unit receive the information it sends back to the truck a notification that the information was received successfully.



## c. Sequence Diagram for Control Unit



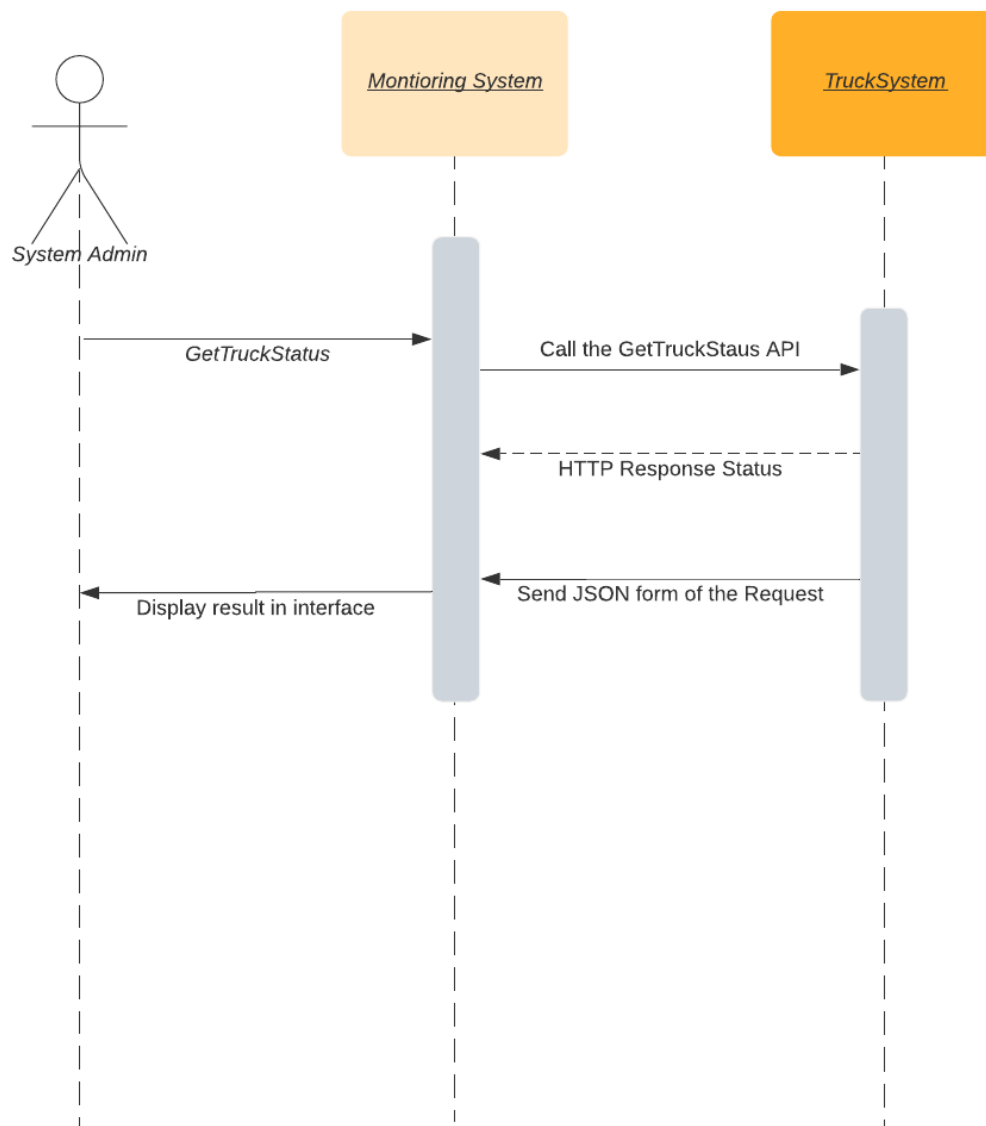
- In this section we see how the Trucks, Leader Truck and Following Trucks, and the Control unit.
- First the Trucks are initialized then signals are sent to check for fuel availability. If available, it moves else it stops initialization.

- Later, a threshold for the speed is set by the Control Unit and sent to the Leading Truck which is then communicated to other trucks.
- Furthermore, the Leading Truck communicates between each other how far apart they must stay from each other during movement.
- Also, the control unit sends signals to decelerate and change lanes while negotiating a bend, it received and executed by the trucks.
- Finally signals are sent to stop the Platoon, it is received and stopped.

#### d. Monitoring Website Communication Flow Sequence Diagram

##### Monitoring website Communication Sequence Diagram

Tunde Oluwayemi Aluko | May 23, 2021



The monitoring website communicates via the REST protocol we use the GET method to the status from the truck.

## 6 Election algorithms

Bully algorithm for the election of the platoon's leader.

The major criteria for choosing the leader is based on the robustness and flexibility of the insurance terms between the insurance company and the vehicle owner. Thus, the stronger the truck insurance, the more likely his election will be.

Truck drivers of the same platoon will be identified by a tag number corresponding to the wellness of their insurance contract.

Leader election scenarios [1]

- Leadership change due to Leader Leave Maneuver:
- Leadership change due to termination of duties of existing leader
- Leadership change due to technical issues in existing leader
- Leadership change during merging of two platoons

An election of a new leader will be made if any of the above scenarios arise

The algorithm for electing a new platoon leader is established on the following assumption [2]:

- The platoon is a synchronous system and it uses timeout mechanism to keep track of coordinator failure detection
- Each truck has a unique ID number to distinguish it.
- Every truck knows the truck ID number of all other truck.
- truck do not know which trucks are currently up and which trucks are currently down
- In the election, a truck with the highest ID number is elected as a coordinator which is agreed by other alive trucks
- A truck that had technical issue can rejoin the system after repair

Bully algorithm for the platoon's leader [2]

When truck T determines that the current coordinator (Leader) is in one of the previously mentioned Leader election scenarios, it executes bully election algorithm using the following sequence of actions.

- Truck T sends an election message to trucks with higher ID number present in the platoon (if there are any). If T receives a response, it stops sending messages. Higher ID number will send back and forth election messages between them till the highest ID number wins the election. The winner will send a later message to notify all other that it is the new coordinator (leader).
- If truck T sends an election message to truck with higher ID number present in the platoon and gets no answer. It then sends an election message to truck with lower ID number to inform them of its new role in the platoon: the coordinator
- Immediately after returning in one of the changes mentioned in Leader election scenarios section, the bully algorithm runs

## **7 Sources**

[1] Pravav k and Sahil S, Leader Election in Cooperative Adaptive Cruise Control Based Platooning, C3VP'18, October 29, 2018, New Delhi, India.

[2] Muhammad Mahbubur Rahman, Afroza Nahar, Modified Bully Algorithm using Election Commission.

## **Who Does What?**

**Tunde Oluwayemi Aluko:**

**2a, 3a, 4a, 5d.**

**Enkeledi Mema:**

**1, 2d, 3c, 4b**

**Brian Kelein Ngwoh Visas:**

**2c, 3d, 5c**

**Anguiga Hermann:**

**2c, 3b, 6**