TELEMATICS CONTROL UNIT IN MODERN VEHICLES

جامعة حلوان
HELWAN UNIVERSITY

Team members

Ahmed sherif

Ahmed abdellah

Mariam mostafa

Nada mohamed

Samaa khaled

our sponsors & mentors

supervised by:
DR: Mohamed Eldakrory

Mentored by:
ENG:Bishoy Wasfy
ENG:Bahaa sehsah

Valeo

introduction

Automotive dead reckoning

Emergency call system

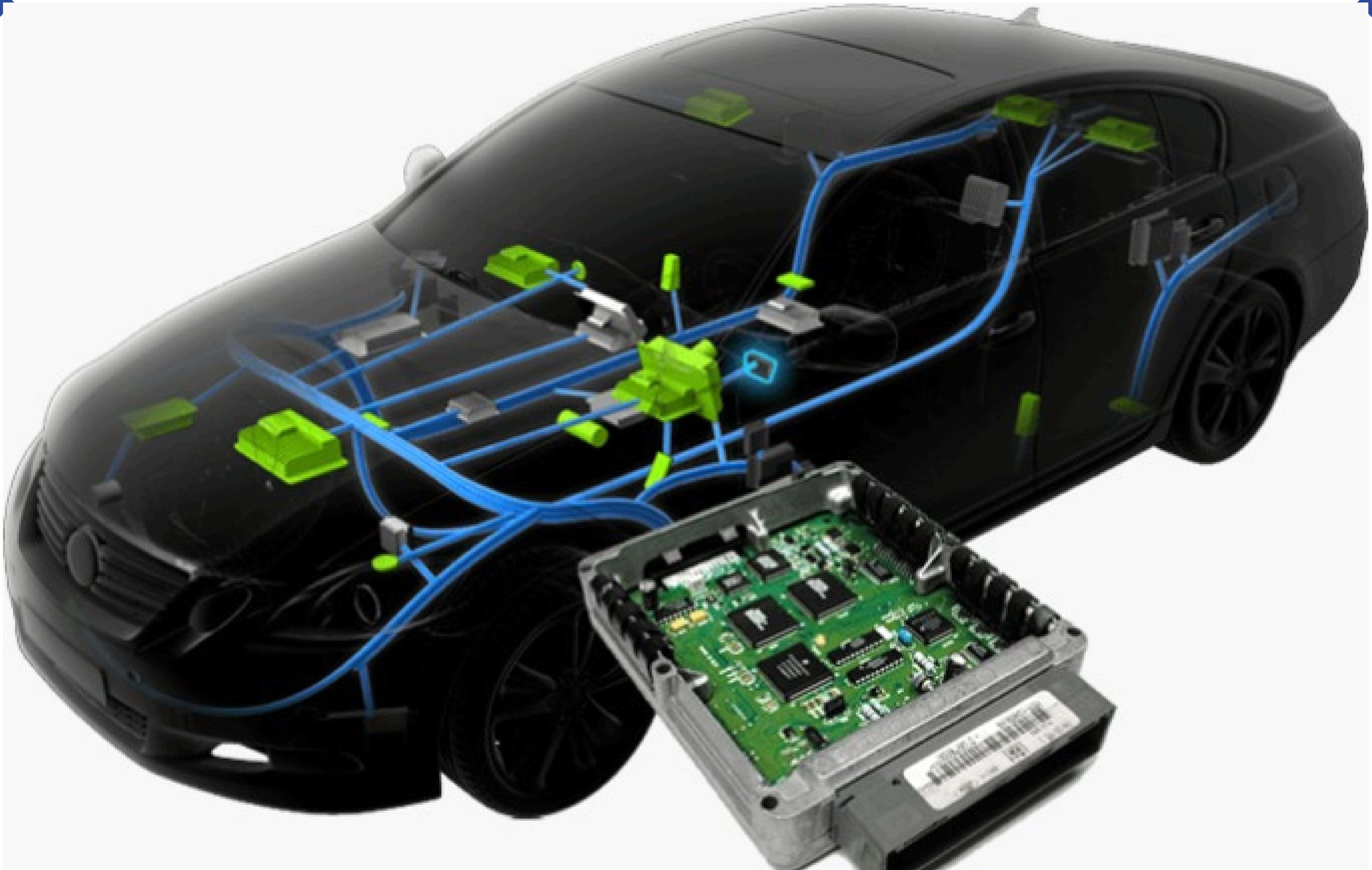
Socket programming

Inter process communication

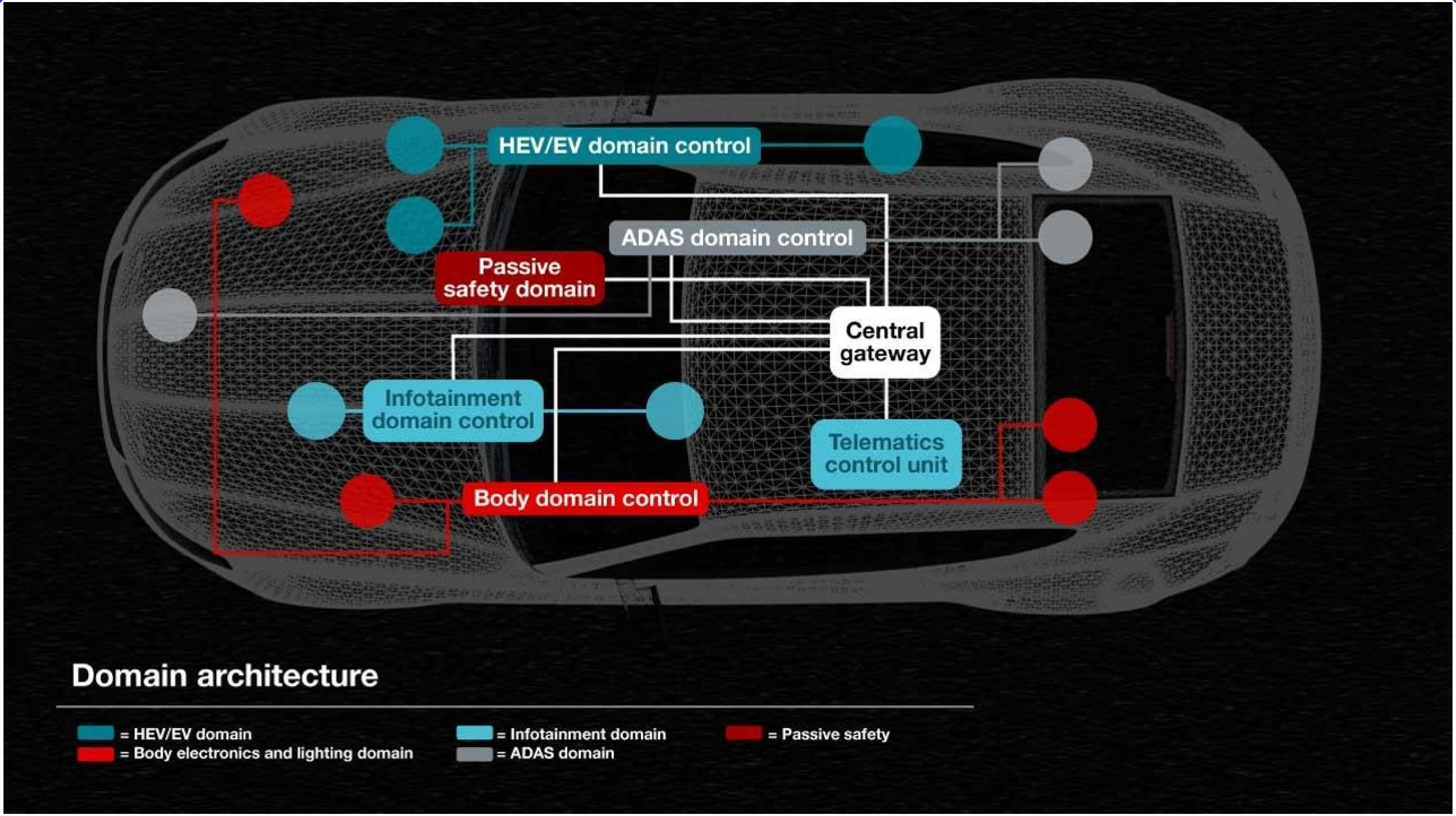
Problem statments

Project idea

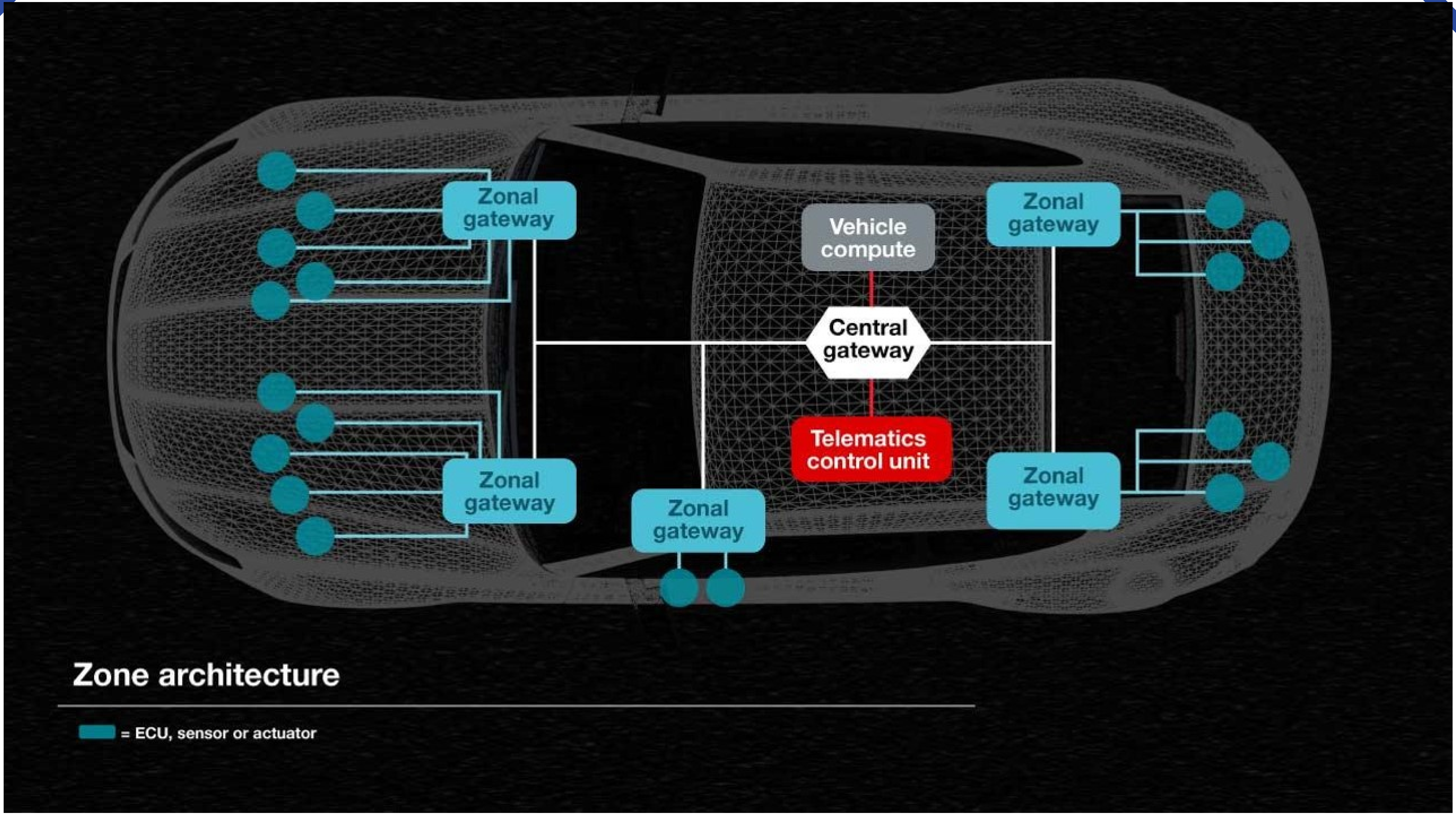
TELEMATICS CONTROL UNIT



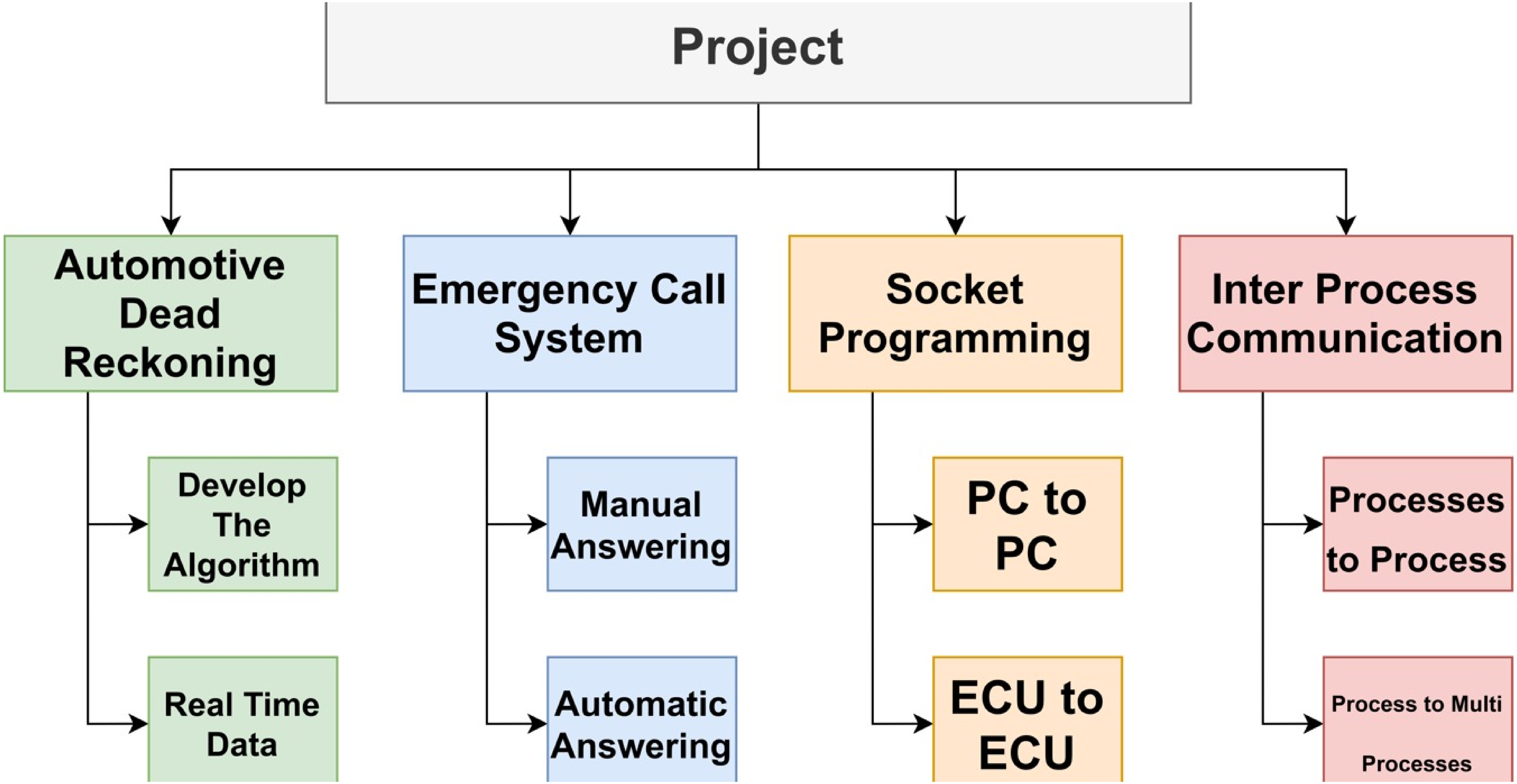
CLASSICAL VEHICLES



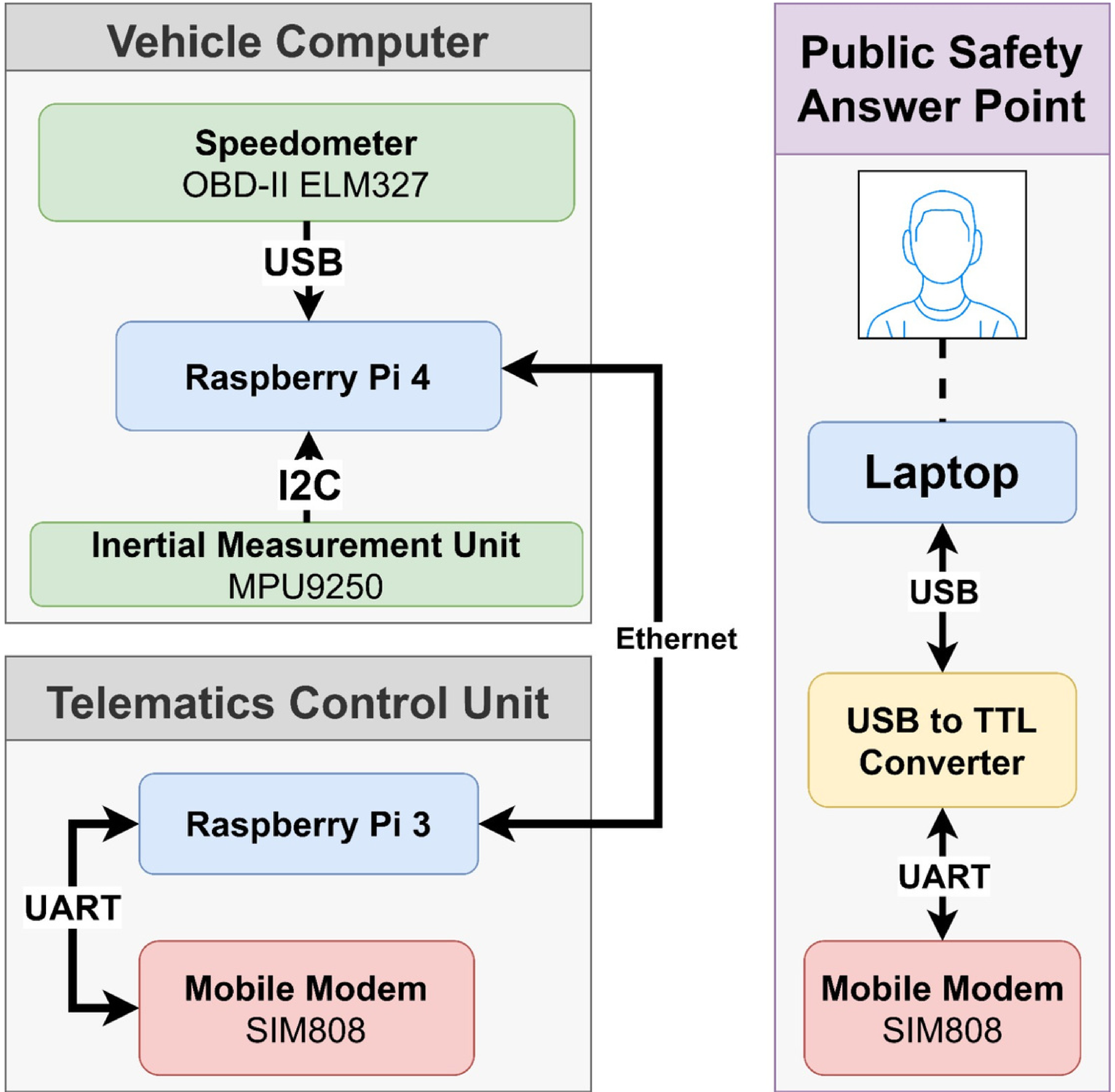
MODERN VEHICLES



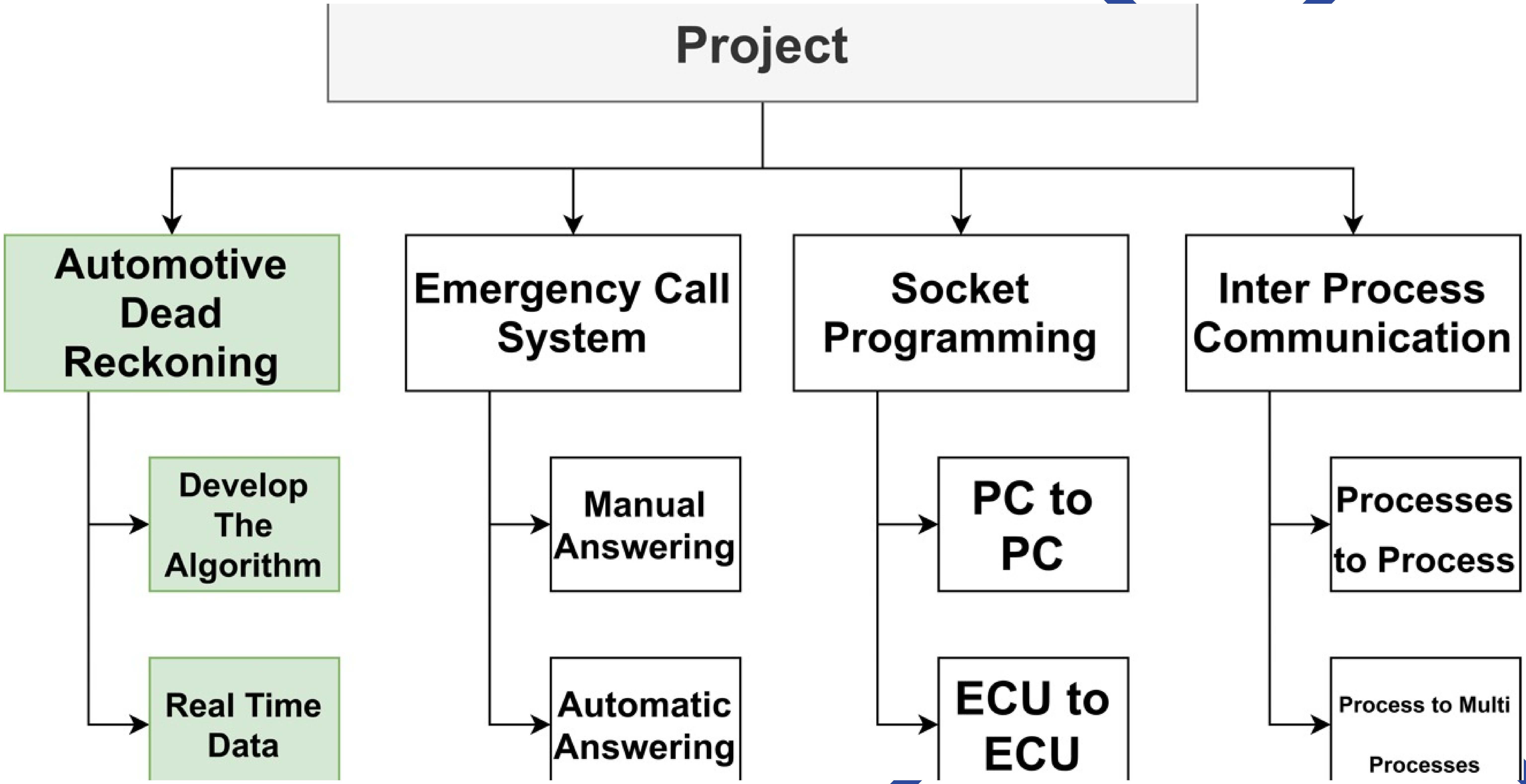
WORK BREAKDOWN STRUCTURE



SYSTEM ARCHITECTURE



Part 1: Automotive Dead Reckoning



Problem statement

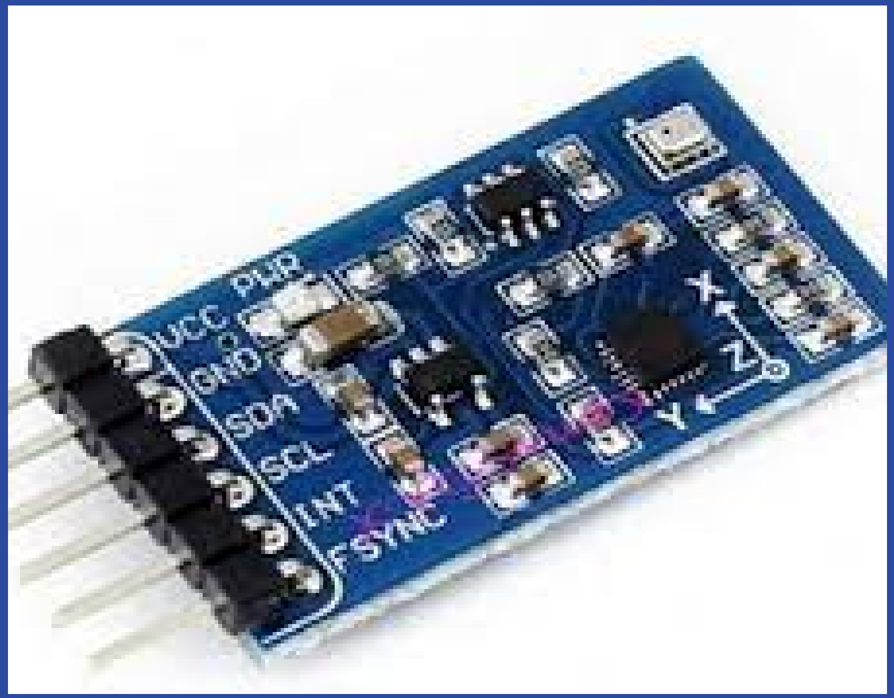
1)obstruction of GPS signals

2)inaccurate location information

3)effectively & quickly respond of emergency services means the difference between life and death.

fusing the data

GPS



IMU

SPEEDOMETER

Why use IMU&GPS and Speedometer ?

1. Their error dynamics are completely different and uncorrelated that mean of one of these failed The others are unlikely to fail.
2. IMU and speedometer provides smoothing of the GPS signal and continue to estimate the car state during GPS outage.
3. GPS provides absolute positioning information to reduce IMU drifts.
4. IMU and speedometer provided with the proper vehicle’s motion dynamics into the information fusion process can provide a good estimation of the car’s position during GPS outage

Why use ES-KF

1)Better performance compared to vanilla EKF

2)less computation compared to UKF+

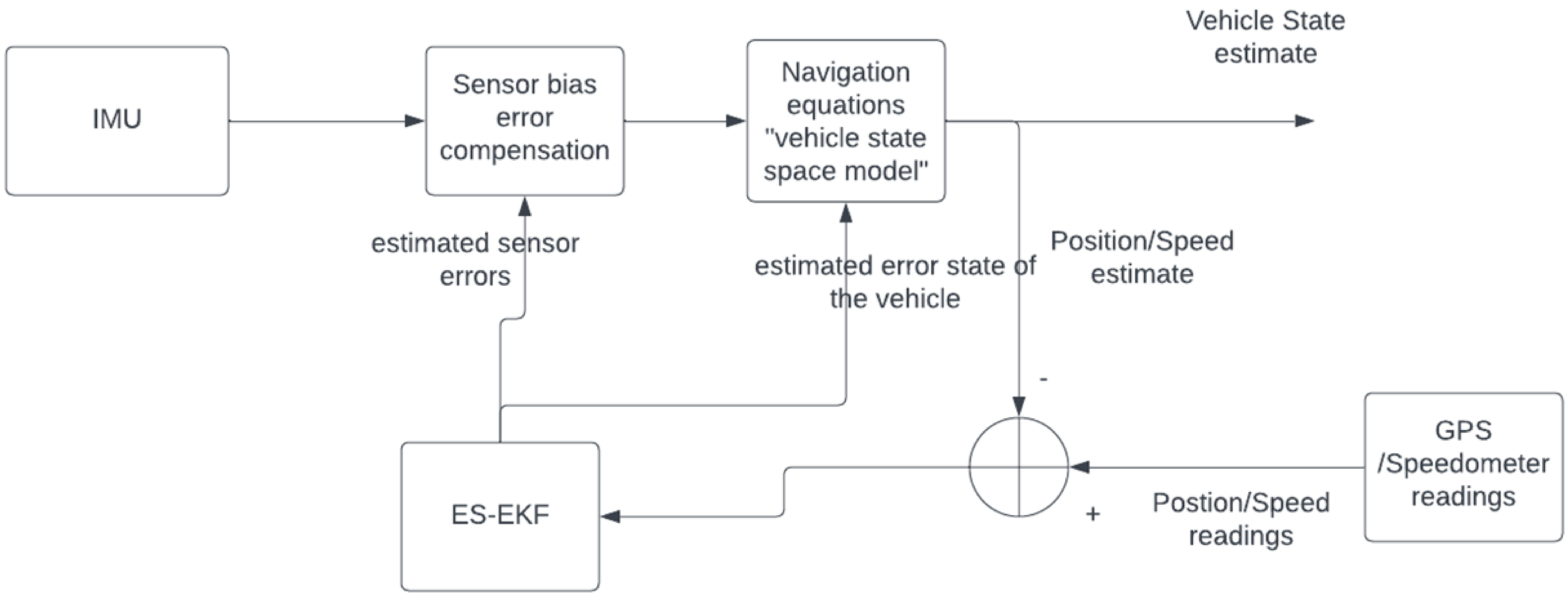
3)Easy to Work with rotational quantities like Quaternion

ES-EKF coupling

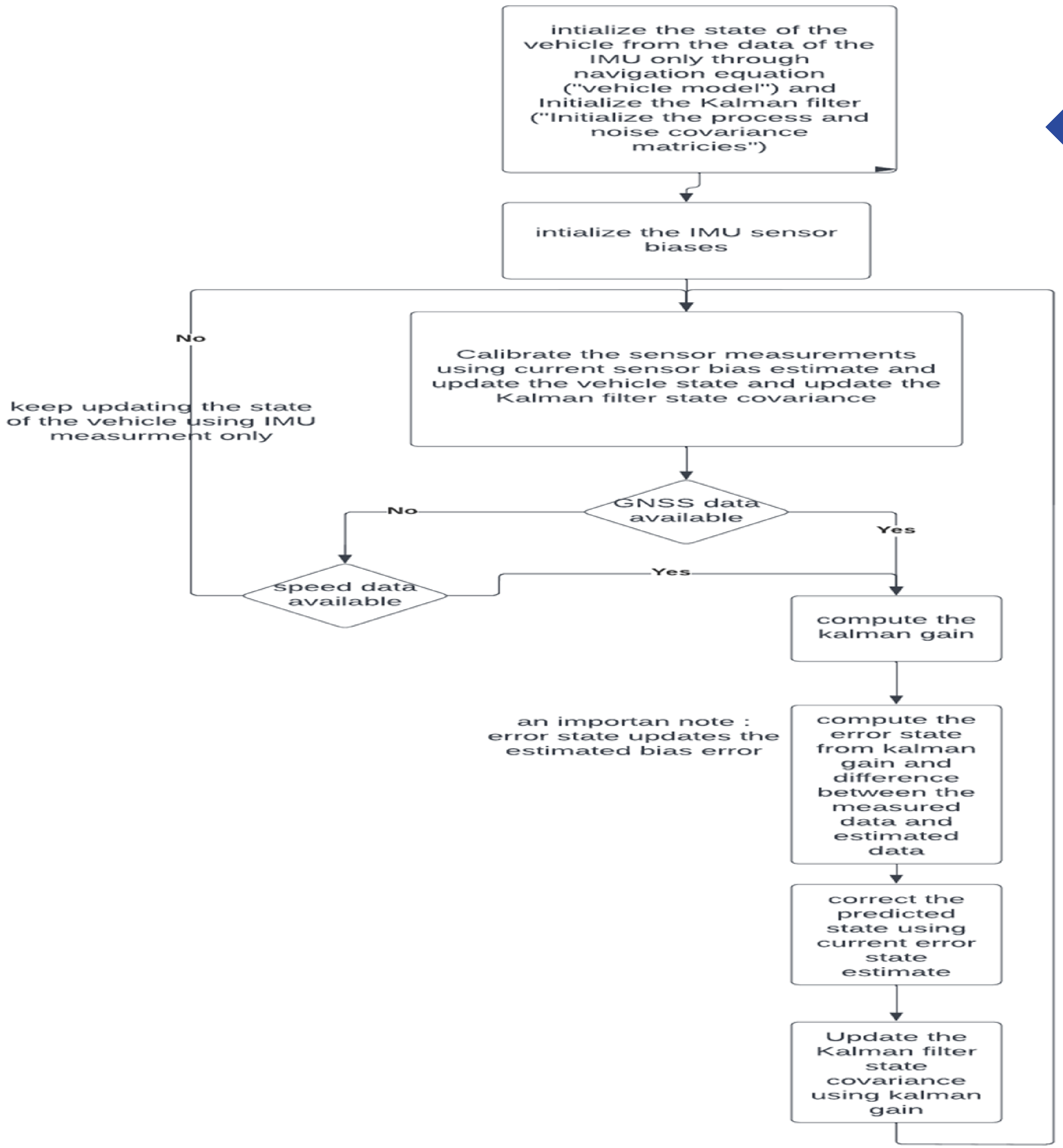
1)lossy coupled sensor fusion approach

2)why a loosely coupled?

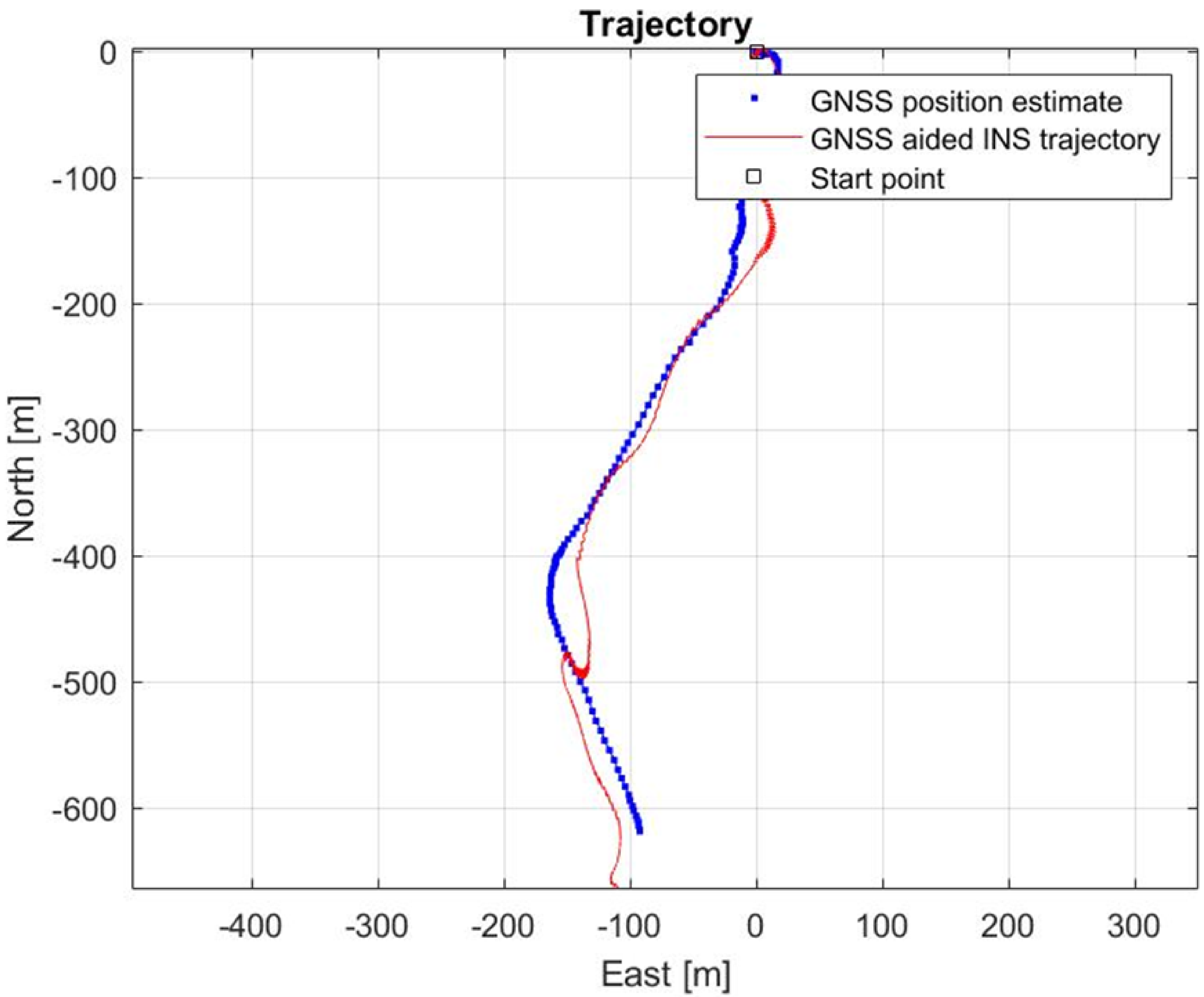
Proposed algorithm for GPS outage



Algorithm flow graph



Output


Trajectory
GNSS position estimate
GNSS aided INS trajectory
Start point
North [m]
East [m]
0
-100
-200
-300
-400
-500
-600
-400
-300
-200
-100
0
100
200
300

IMPORTANT MATH EQUATIONS

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k),$$

where

$$\begin{aligned}
\mathbf{p}_k &= \mathbf{p}_{k-1} + T_s \mathbf{v}_{k-1} + \frac{T_s^2}{2}\Big(\mathbf{R}_b^n(\mathbf{q}_{k-1})\mathbf{s}_k - \mathbf{g}\Big) \\
\mathbf{v}_k &= \mathbf{v}_{k-1} + T_s\Big(\mathbf{R}_b^n(\mathbf{q}_{k-1})\mathbf{s}_k - \mathbf{g}\Big) \\
\mathbf{q}_k &= \Big(\cos(0.5\,T_s\,\|\boldsymbol{\omega}_k\|)\mathbf{I}_4 + \frac{1}{\|\boldsymbol{\omega}_k\|}\sin(0.5\,T_s\,\|\boldsymbol{\omega}_k\|)\boldsymbol{\Omega}_k\Big)\mathbf{q}_{k-1}
\end{aligned}$$

and

$$\boldsymbol{\Omega}_k = \begin{bmatrix} 0 & [\boldsymbol{\omega}_k]_z & -[\boldsymbol{\omega}_k]_y & [\boldsymbol{\omega}_k]_x \\ -[\boldsymbol{\omega}_k]_z & 0 & [\boldsymbol{\omega}_k]_x & [\boldsymbol{\omega}_k]_y \\ [\boldsymbol{\omega}_k]_y & -[\boldsymbol{\omega}_k]_x & 0 & [\boldsymbol{\omega}_k]_z \\ -[\boldsymbol{\omega}_k]_x & -[\boldsymbol{\omega}_k]_y & -[\boldsymbol{\omega}_k]_z & 0 \end{bmatrix}.$$

$$\mathbf{z}_k = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k)\mathbf{z}_{k-1} + \mathbf{G}_k(\mathbf{x}_k)\mathbf{w}_k,$$

where

$$\mathbf{z}_k \triangleq \begin{bmatrix} \delta\mathbf{x}_k \\ \delta\mathbf{u}_k \end{bmatrix} \in R^{15}, \quad \mathbf{w}_k \triangleq \begin{bmatrix} \mathbf{w}_k^{(1)} \\ \mathbf{w}_k^{(2)} \end{bmatrix} \in R^{12}$$

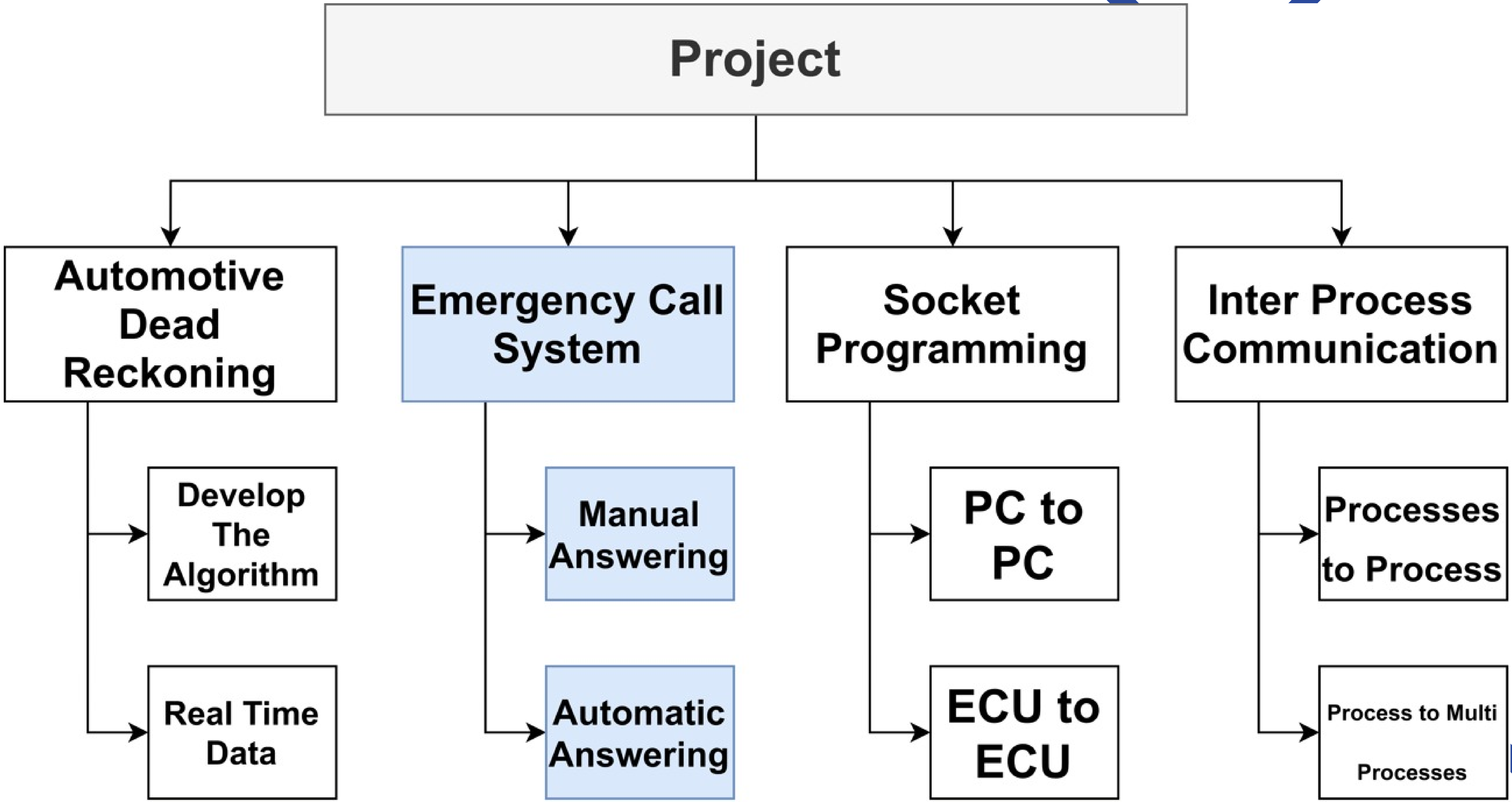
and the state transition matrix and noise gain matrix are defined as

$$\mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathbf{I}_3 & T_s\mathbf{I}_3 & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I}_3 & T_s[\mathbf{R}_b^n(\mathbf{q}_k)\mathbf{s}_k]_\times & T_s\mathbf{R}_b^n(\mathbf{q}_k) & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3} & -T_s\mathbf{R}_b^n(\mathbf{q}_k) \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 \end{bmatrix}$$

and

$$\mathbf{G}_k(\mathbf{x}_k) = \begin{bmatrix} \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ T_s\mathbf{R}_b^n(\mathbf{q}_k) & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & T_s\mathbf{R}_b^n(\mathbf{q}_k) & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{0}_{3,3} & \mathbf{I}_3 \end{bmatrix},$$

Part 2: Emergency Call System



ECALL & BCALL

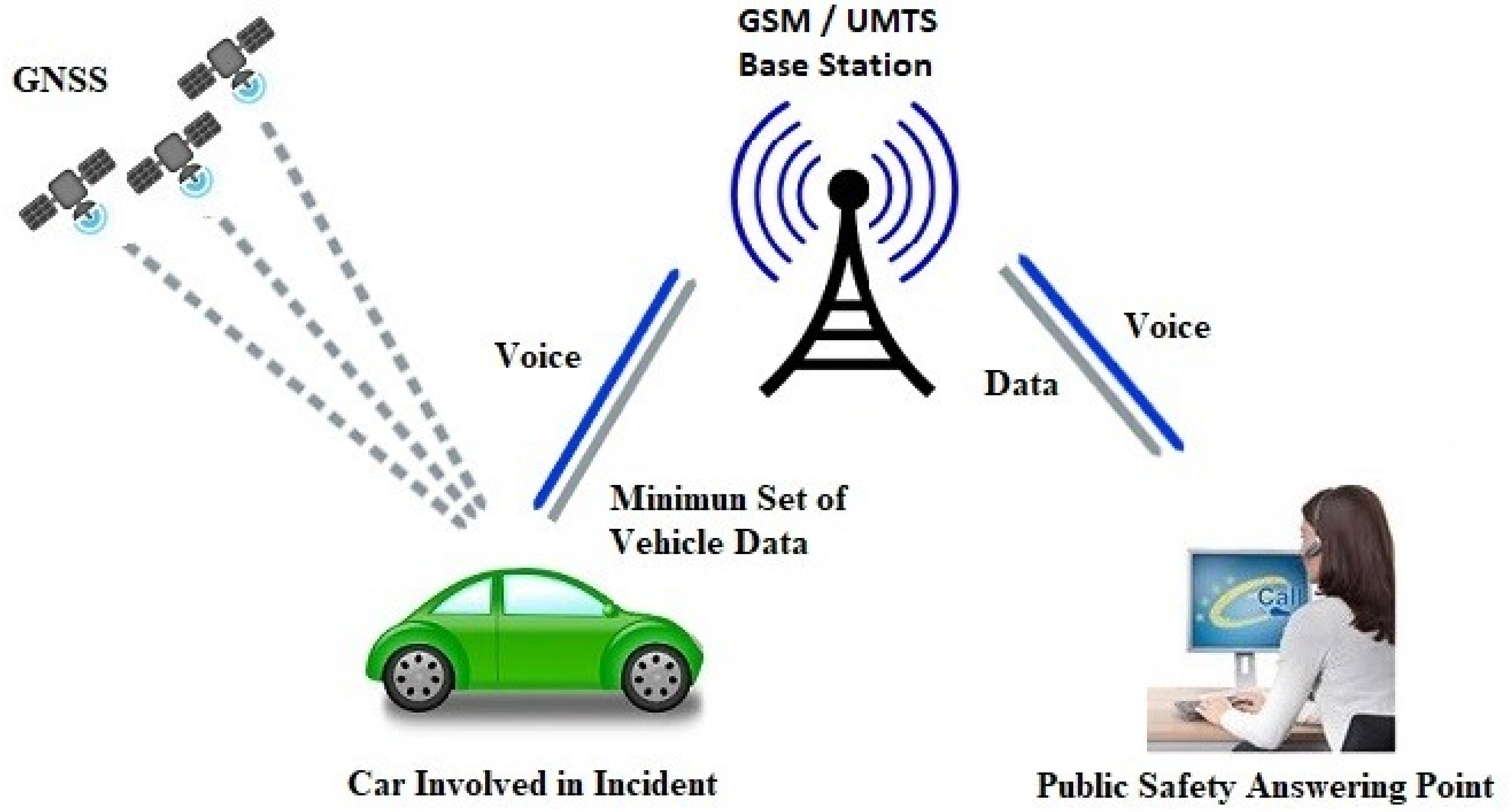
ECALL

If your vehicle is involved in a serious accident, you will be connected with the nearest emergency response network.

BCALL

It allows drivers to call for 24-hour roadside assistance in the event that the car breaks down or in case of the need of help.

ECALL & BCALL

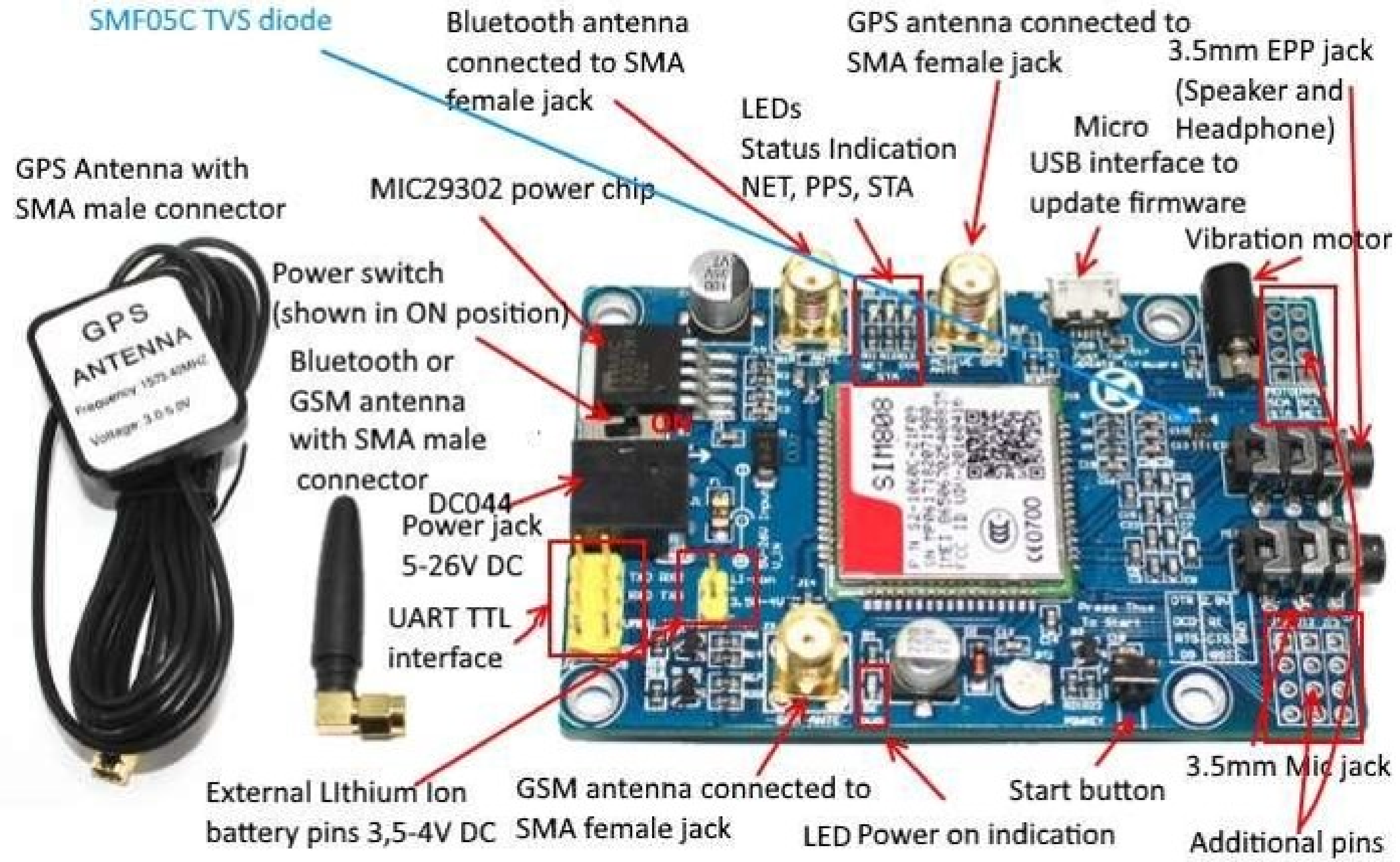

GNSS
GSM / UMTS
Base Station
Voice
Minimun Set of
Vehicle Data
Voice
Data
Car Involved in Incident
Public Safety Answering Point

hardware used for ECALL

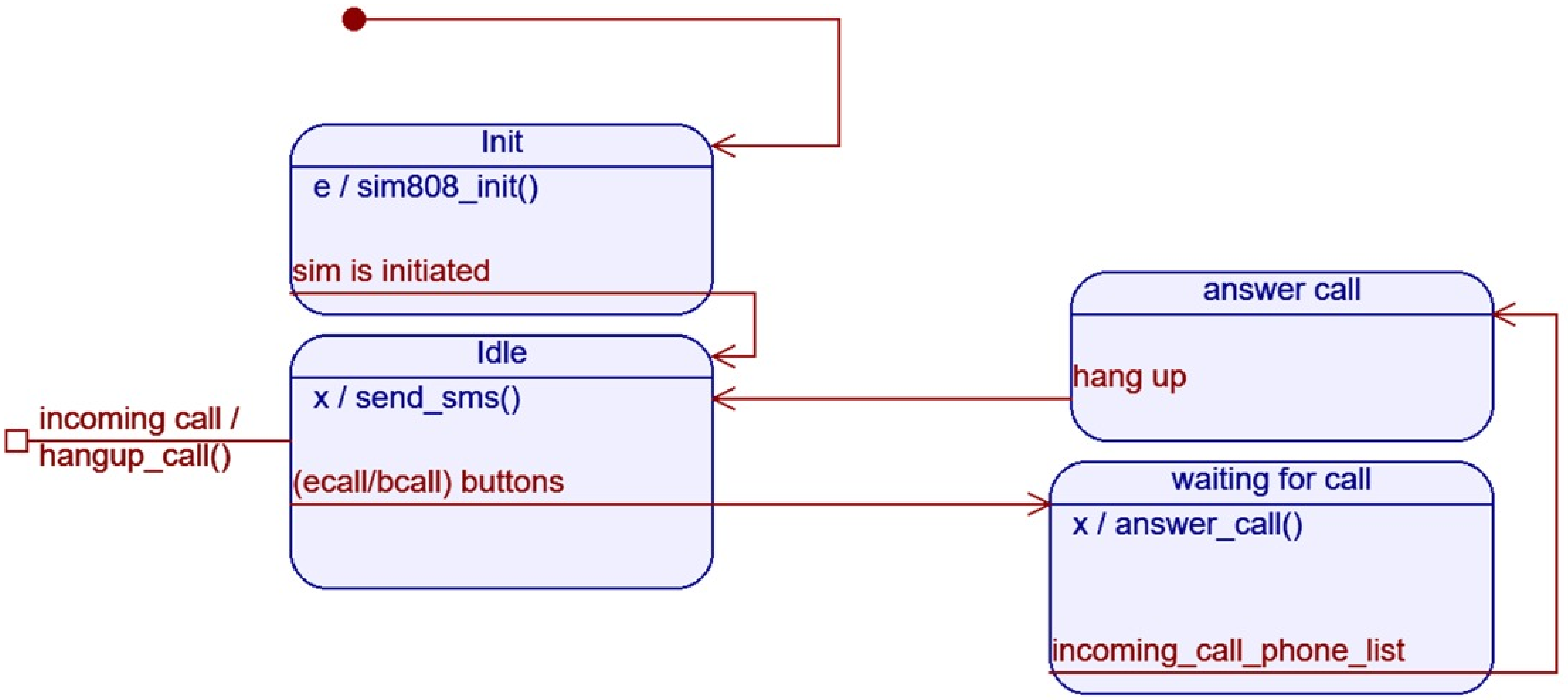
SIM808

- GPS
- GSM
- GPRS

Controlled by AT commands



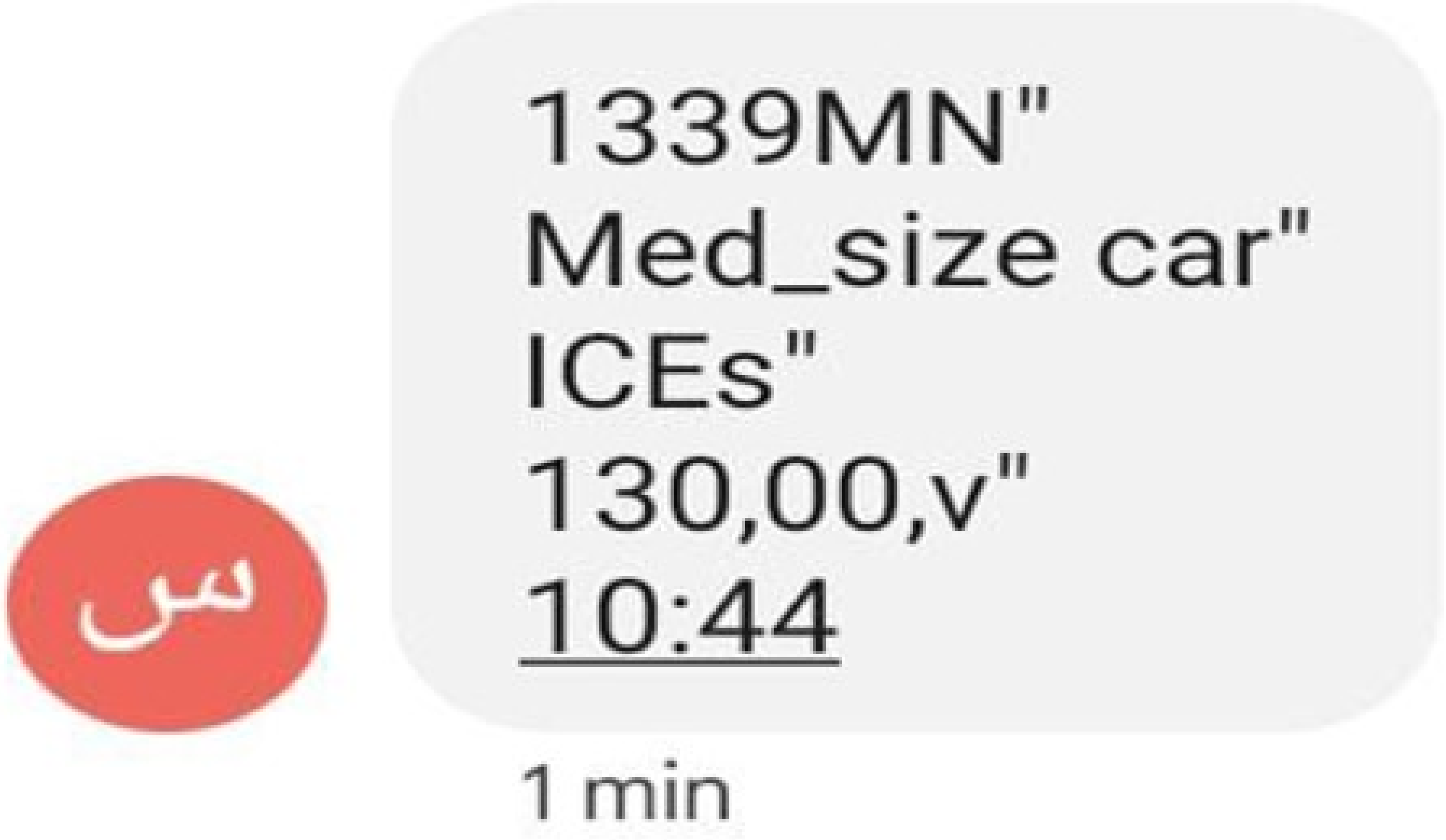
STATE MACHINE



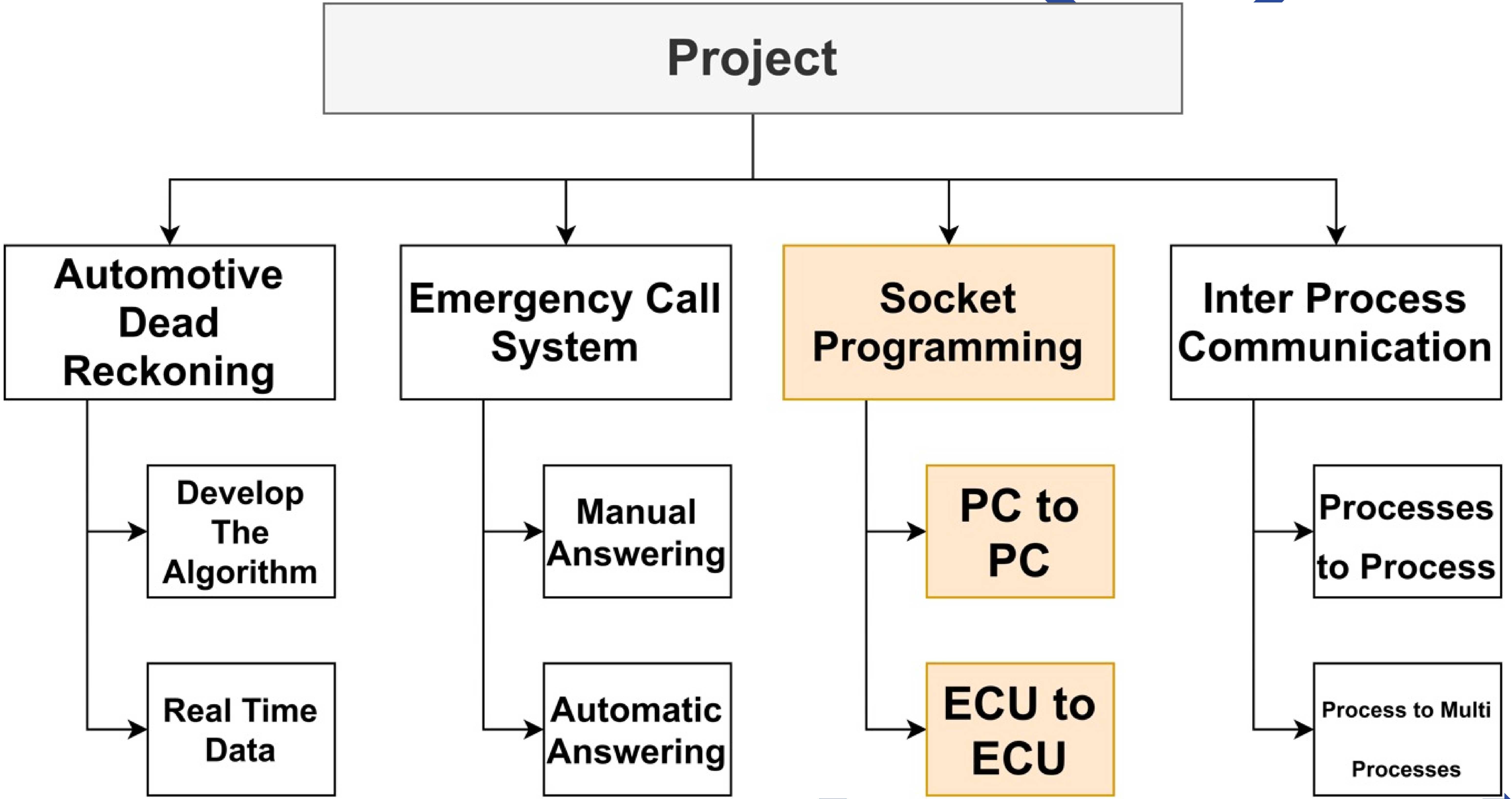
sending MSD

MINIMUM SETUP DATA

- VEHICLE REGISTRATION
- VEHICLE TYPE
- PROPULSION TYPE
- POSITION DATA
- TIME OF THE INCIDENT
- SOS TYPE
- POSITION CONFIDENCE



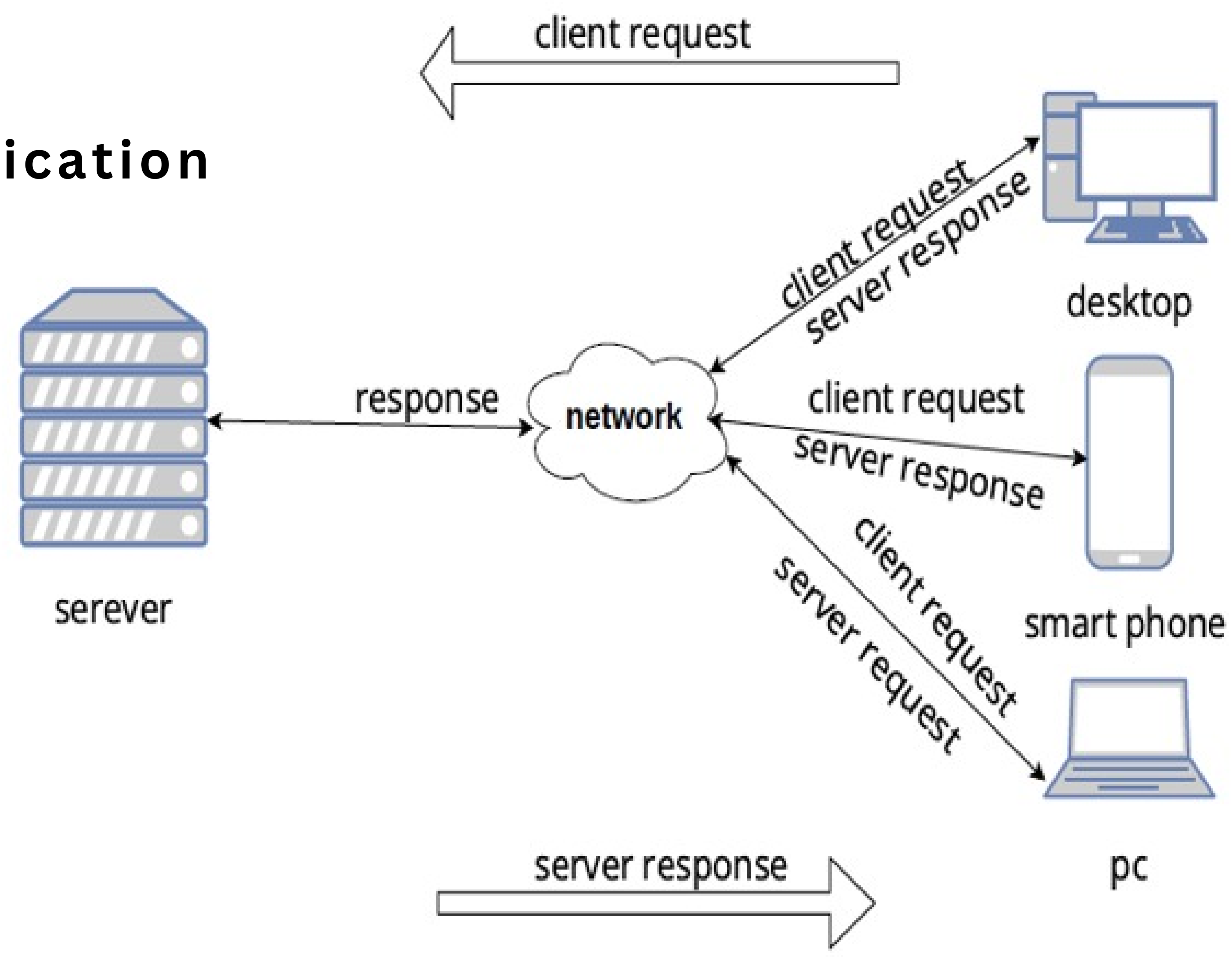
Part 3: Socket Programming



Part 3: Socket Programming

1 **DEFINITION**

Is a method to connect two nodes over a network to establish a means of communication between those two nodes.
A node represents a computer or a phys_ ical device with an internet connection.
A socket is the endpoint used for connecting to a node.



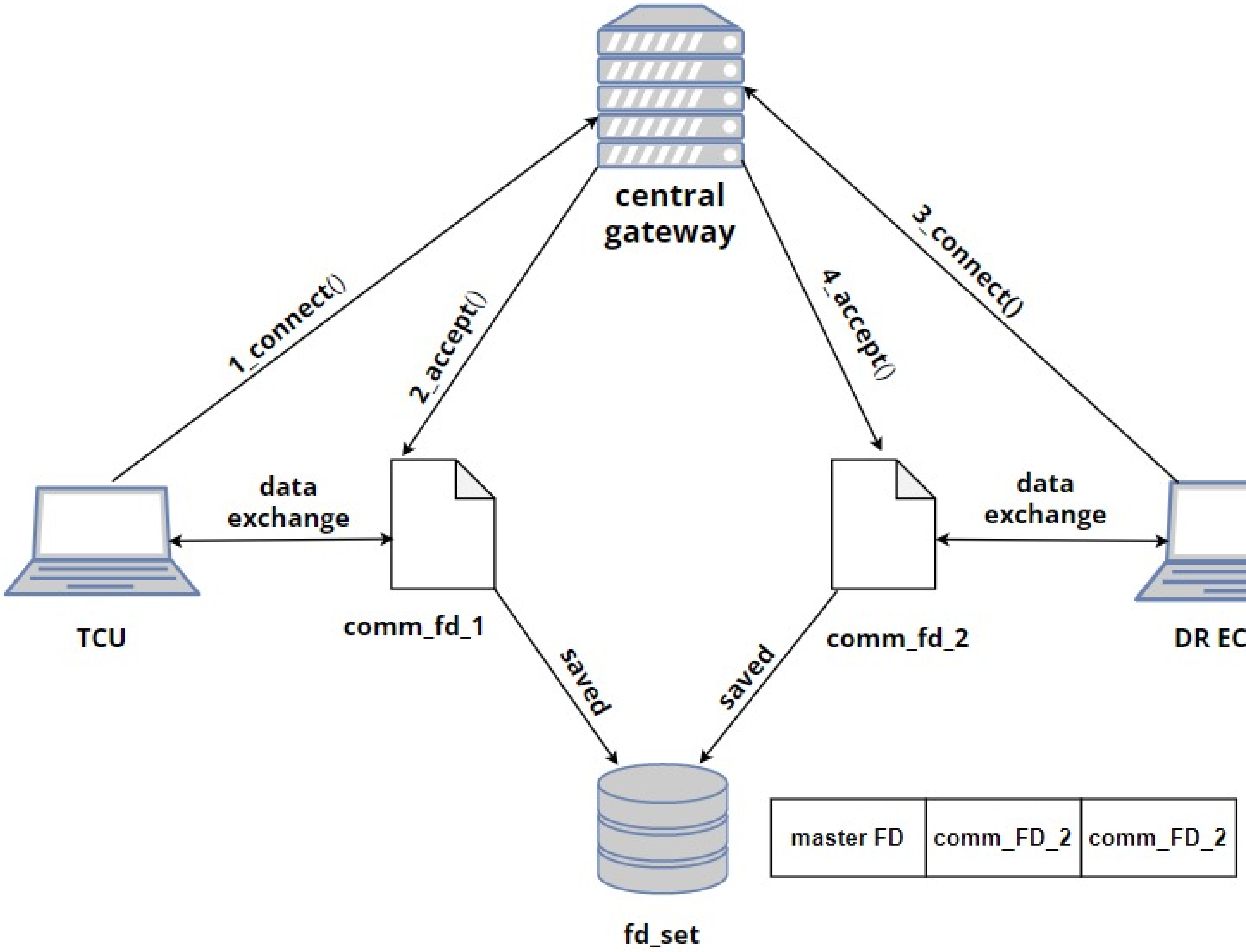
2 **IDENTIFICATION**

IP address & Port number

3 **EXAMPLE**

communicate between two raspberry PIs

connection between TCU ECU&gateway ECU



TCP VS UDP

Features	TCP	UDP
Connection status	Connection oriented	Connectionless
Retransmission of data	Retransmission of lost packets is possible	No retransmission of lost packets
Speed	Slower than UDP	Faster than TCP
Transmission modes	Allows two-way data exchange, once the connection is established. (full-duplex)	Allows data to be transferred in one direction at once. (half-duplex)
Real time application	Not used in real time application	Used in real time application

sending sensor data between two raspberry pi

The sensor readings

Ax	Ay	Az	Gx	Gy	Gz
0.06	0.06	0.85	-0.04	0.06	1.61
0.06	0.05	0.87	-0.08	0.13	1.61
0.06	0.06	0.0.86	0.02	0.15	1.66
0.06	0.07	0.86	0.01	0.19	1.66
0.06	0.06	0.85	-0.04	0.06	1.61
0.06	0.05	0.87	-0.08	0.13	1.61
0.06	0.06	0.0.86	0.02	0.15	1.66
0.06	0.07	0.86	0.01	0.19	1.66
0.06	0.06	0.85	-0.04	0.06	1.61
0.06	0.05	0.87	-0.08	0.13	1.61
0.06	0.05	0.84	0	0	0
0.07	0.06	0.85	0	0.15	1.08
0.06	0.06	0.84	-0.01	0.11	1.56
0.06	0.07	0.85	-0.02	0.18	1.69
0.06	0.06	0.84	0.04	0.24	1.63
0.08	0.06	0.84	-0.02	0.1	1.66
0.06	0.06	0.84	-0.03	0.13	1.69
0.07	0.06	0.85	-0.04	0.11	1.66
0.07	0.05	0.85	0.02	0.15	1.66
0.07	0.06	0.83	-0.02	0.19	1.72
0.07	0.06	0.84	-0.07	0.14	1.69
0.06	0.06	0.84	-0.07	0.27	1.62
0.06	0.06	0.86	-0.02	0.19	1.63
0.06	0.07	0.86	0	0.24	1.69

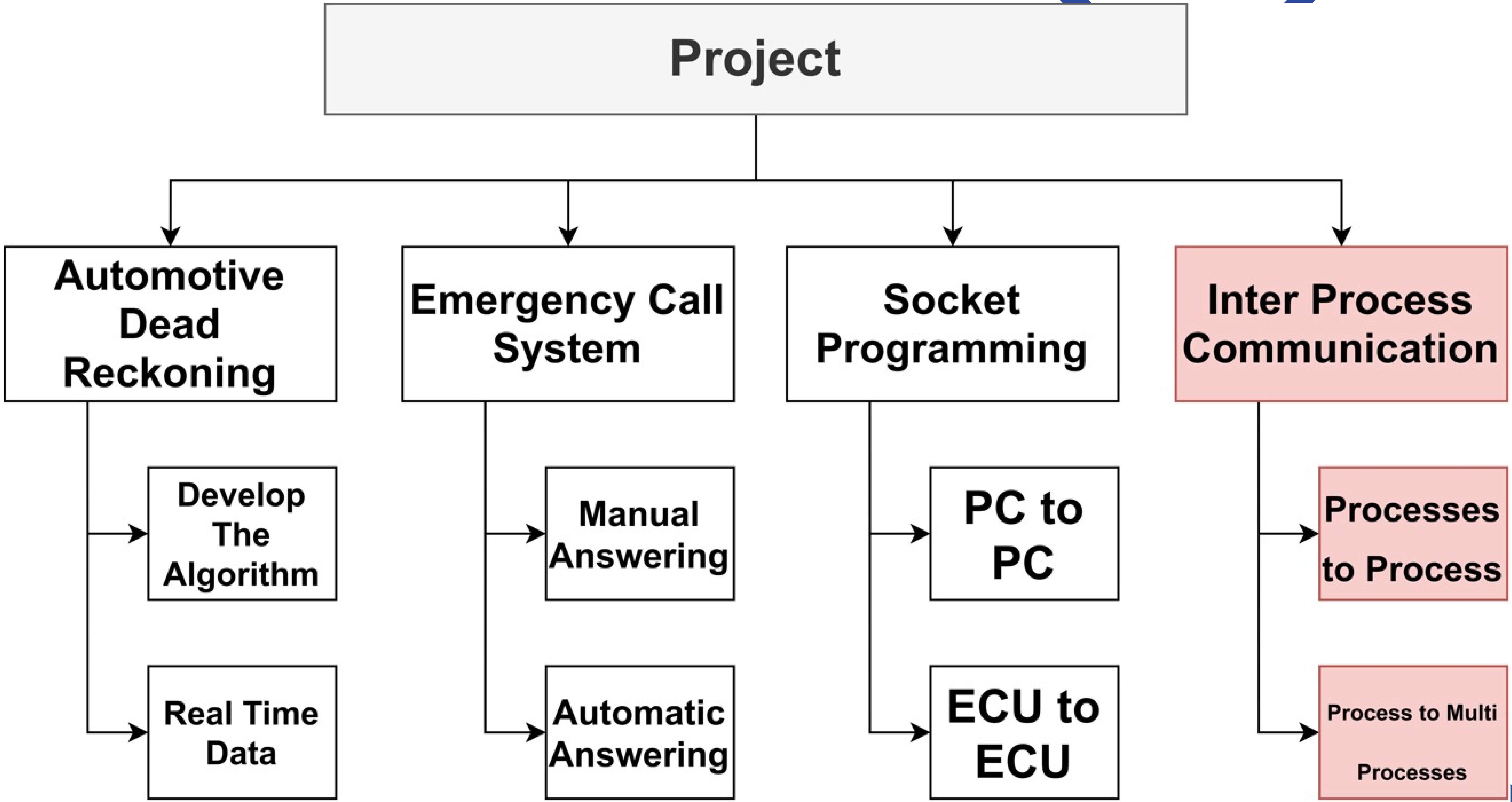
client side receiving data

```
nada@nada-VirtualBox:~/Desktop$ ./client 127.0.0.1 8080
Socket created
Connected

Server reply :

0.06 0.06 0.85 -0.04 0.06 1.61
0.06 0.05 0.87 -0.08 0.13 1.61
0.06 0.06 0.00 0.02 0.15 1.66
0.06 0.07 0.86 0.01 0.19 1.66
0.06 0.06 0.85 -0.04 0.06 1.61
0.06 0.05 0.87 -0.08 0.13 1.61
0.06 0.06 0.00 0.02 0.15 1.66
0.06 0.07 0.86 0.01 0.19 1.66
0.06 0.06 0.85 -0.04 0.06 1.61
0.06 0.05 0.87 -0.08 0.13 1.61
0.06 0.05 0.84 0.00 0.00 0.00
0.07 0.06 0.85 0.00 0.15 1.08
0.06 0.06 0.84 -0.01 0.11 1.56
0.06 0.07 0.85 -0.02 0.18 1.69
0.06 0.06 0.84 0.04 0.24 1.63
0.08 0.06 0.84 -0.02 0.10 1.66
0.06 0.06 0.84 -0.03 0.13 1.69
```

Part 4: Inter-Process Communication



WHY LINUX FOR TELEMATICS

1)Re-use & Features Extendibility

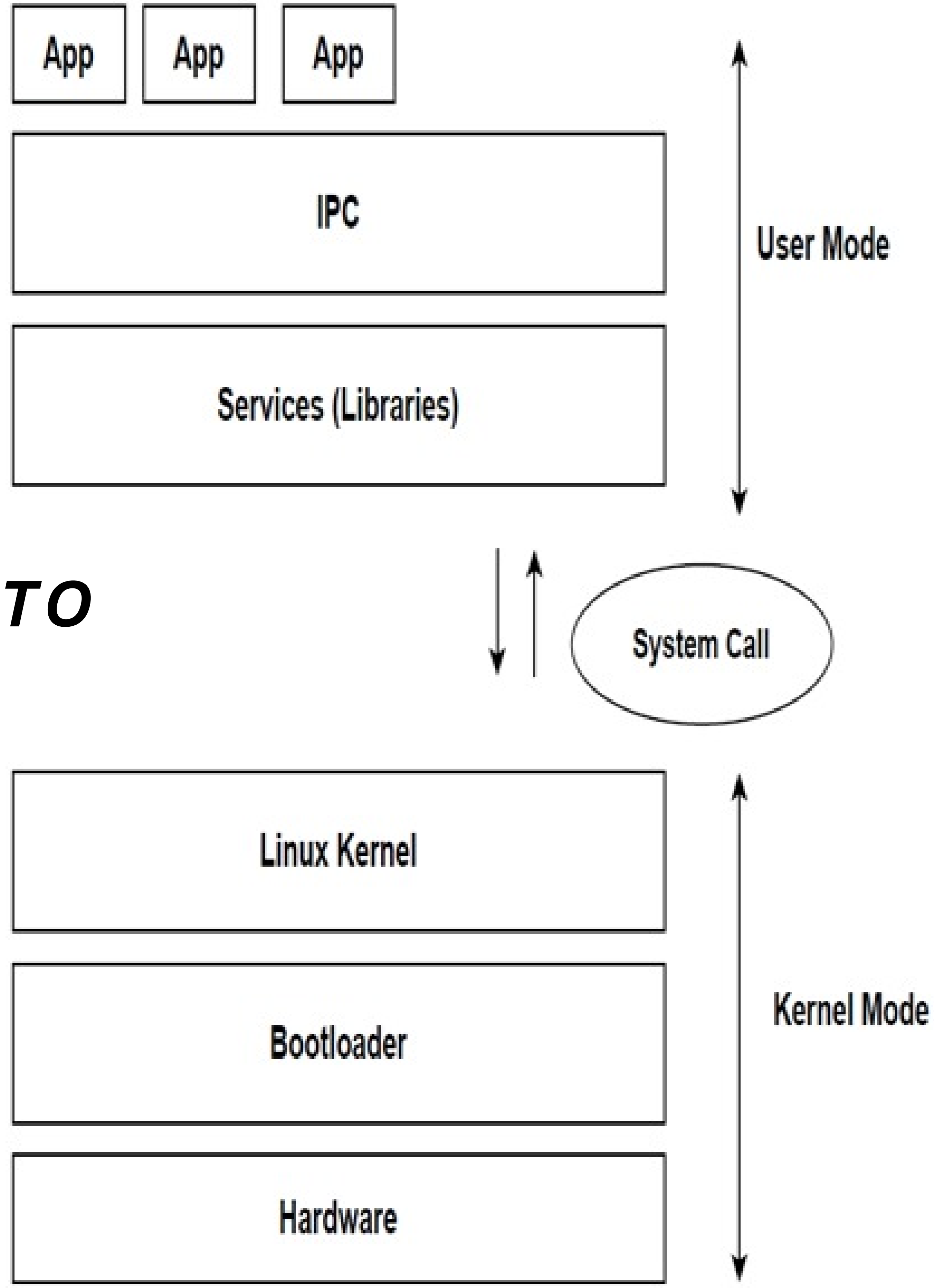
2)cost

3)Time to market

4)Full control

Generic Architecture for Embedded Linux

- BOOTLOADER IS THE FIRST PIECE OF CODE RUNNING BY THE PROCESSOR.
- LINUX KERNEL PROVIDES SUBSYSTEMS TO MANAGE ALL SW/HW RESOURCES.
- USER SPACE CAN COMMUNICATE WITH KERNEL BY SYSTEM CALL.



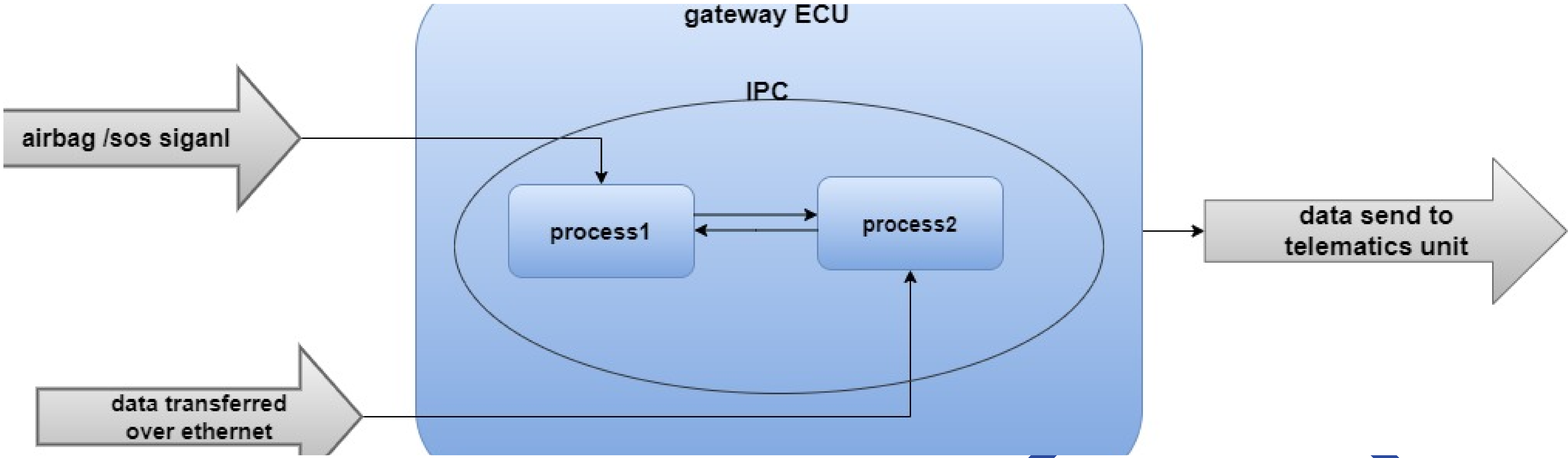
IPC

Linux runs processes /applications at the same time

Processes running on the same machine need to exchange data with each others

Linux provides IPC mechanisms for processes running on the same machine

IPC in our project



Project's Demo

Thank you