Dortmund International Summer School 2020

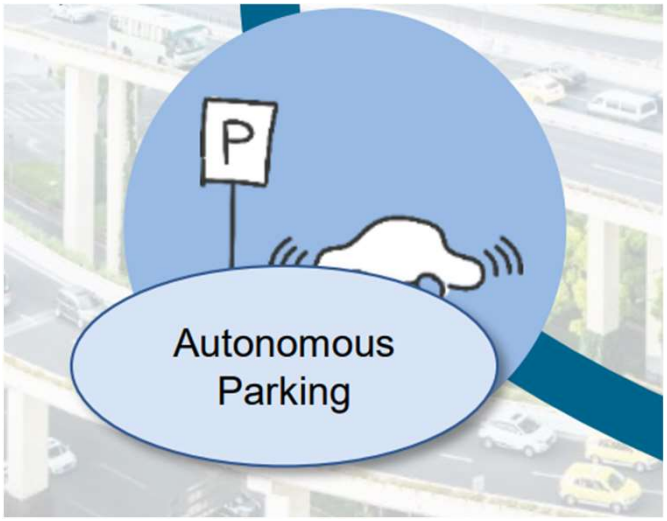
Automotive Software Engineering

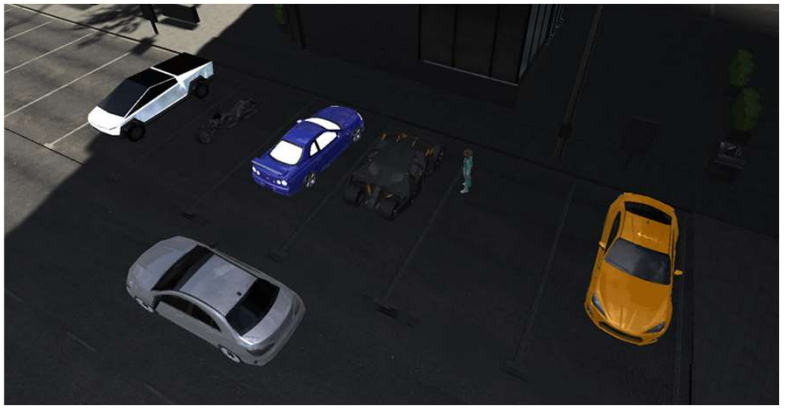
Model-based Requirements Analysis

Group 5: Ernest Hunko
Swathi Rao
Tejas
Torben Müller
Truong Hoang

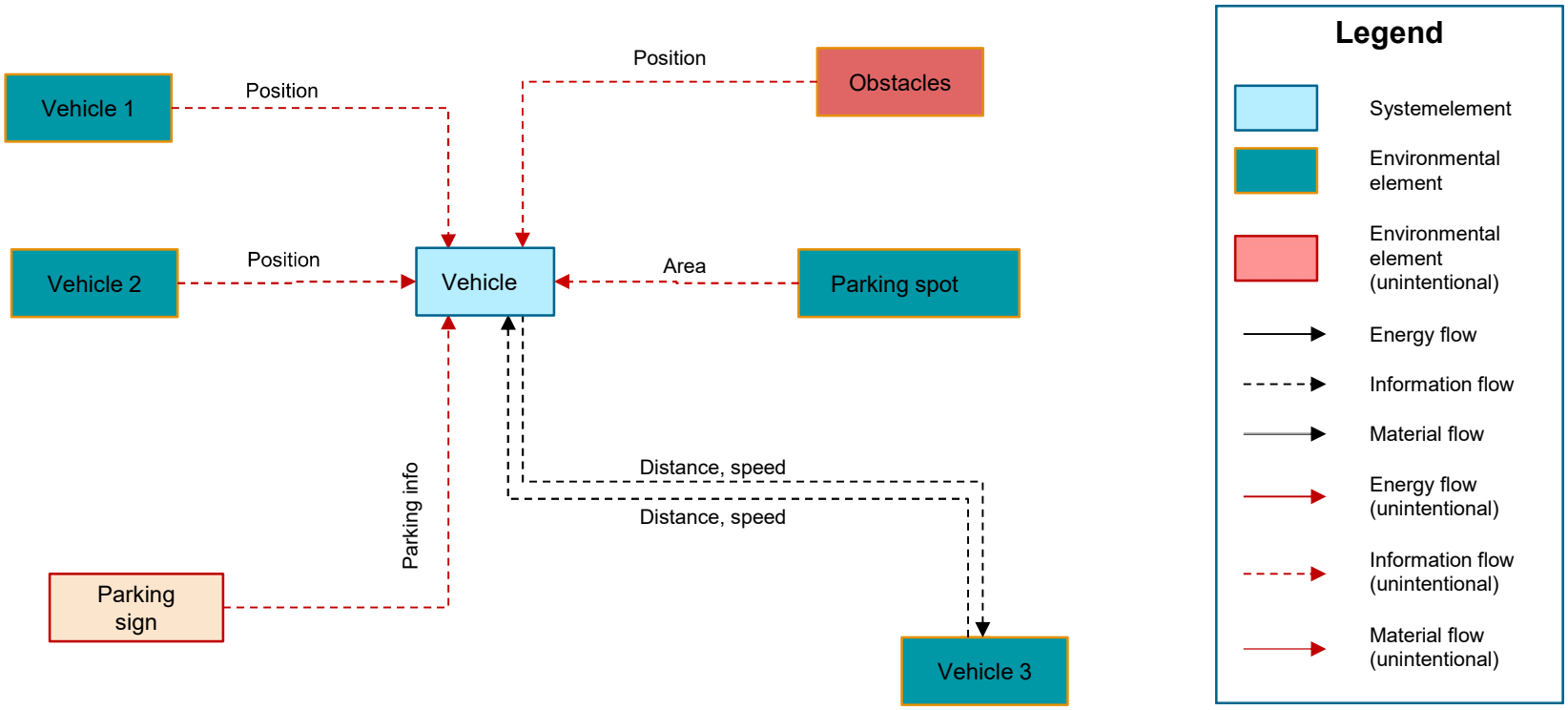
Fachhochschule
Dortmund

Autonomous Parking Scenario



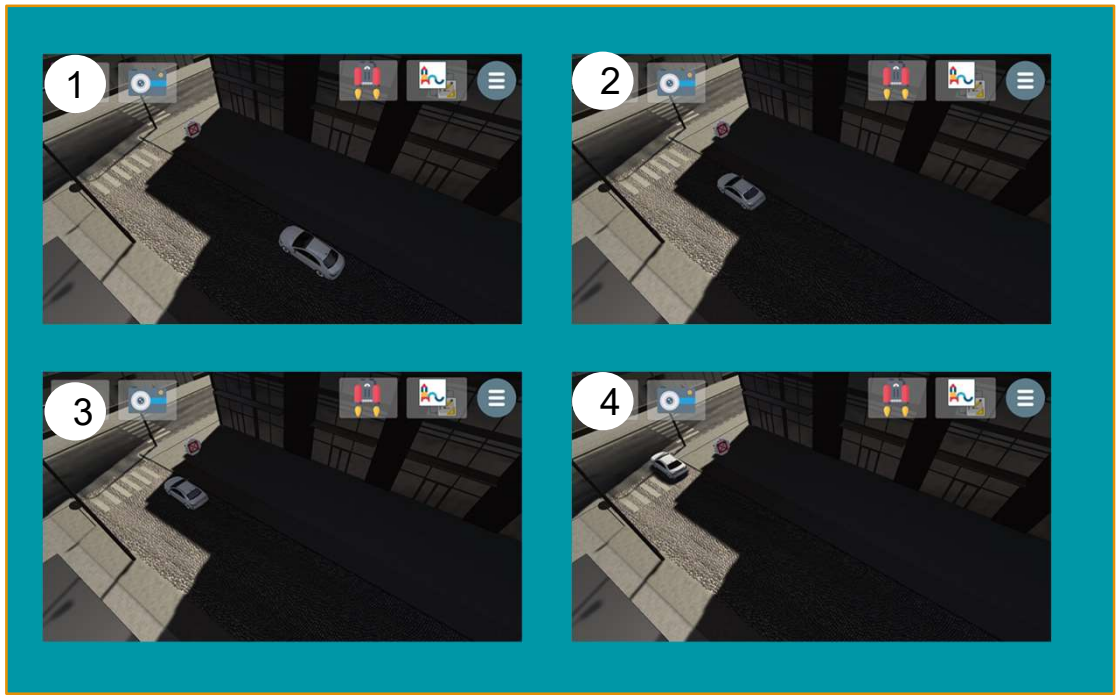


Application Environment Model (Information Flow)

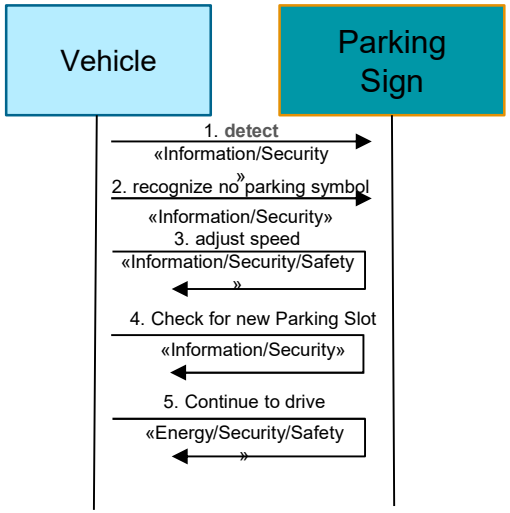


Sequence Diagram 1: Use Case Parallel Parking With A Sign

UC01

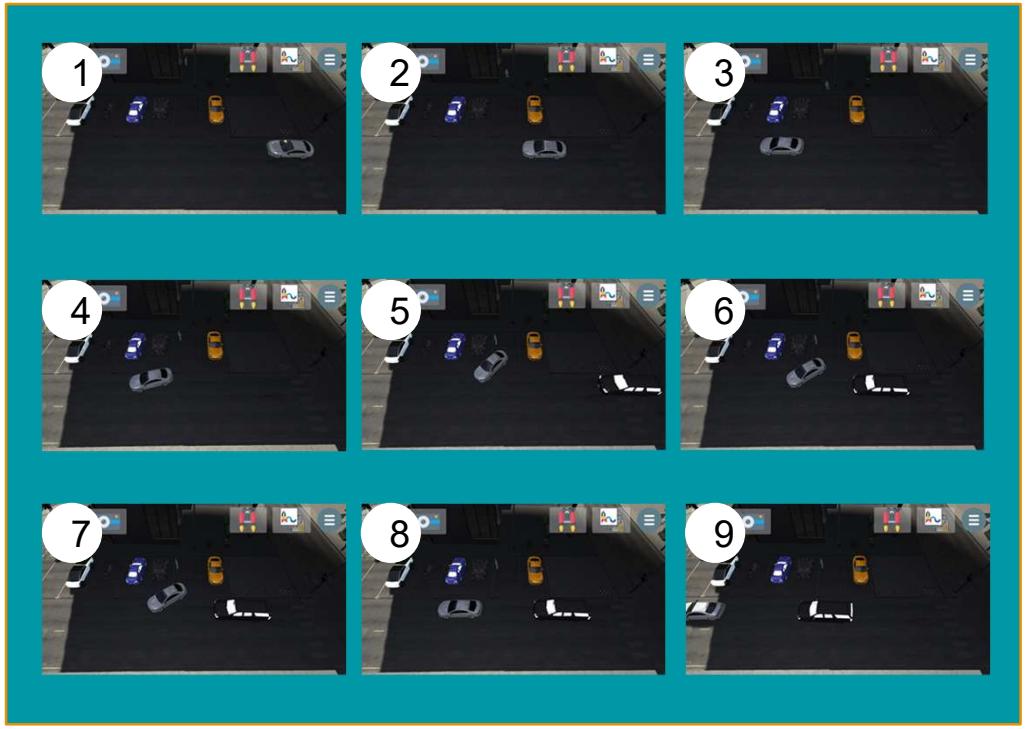


SQ1

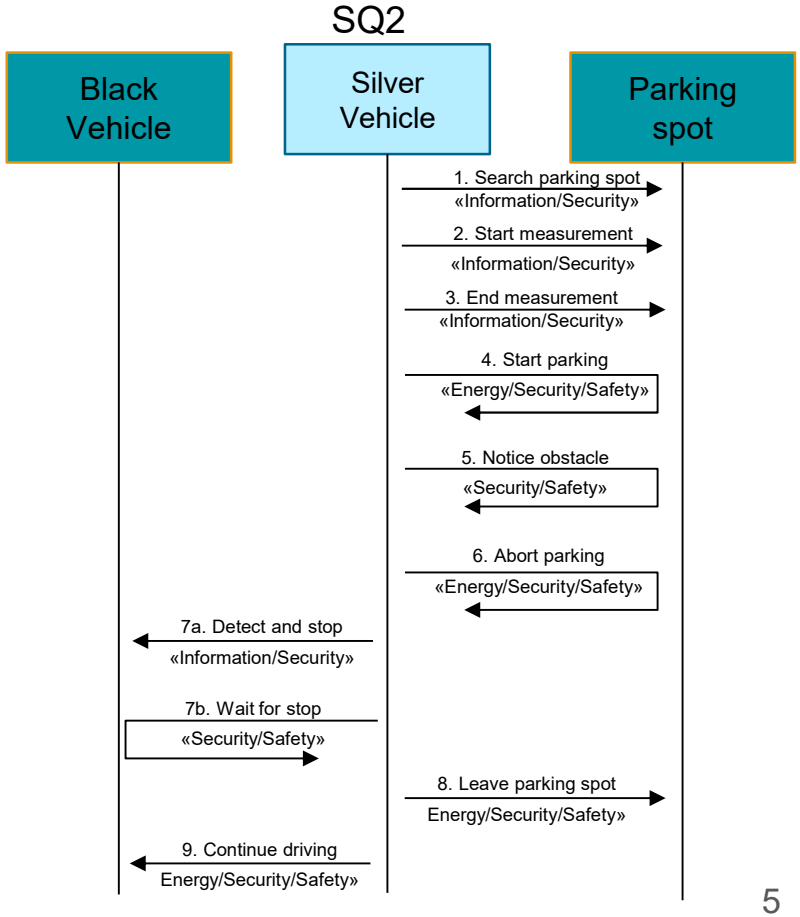


Sequence Diagram 2: Threat Case Parking With An Obstacle

TC01



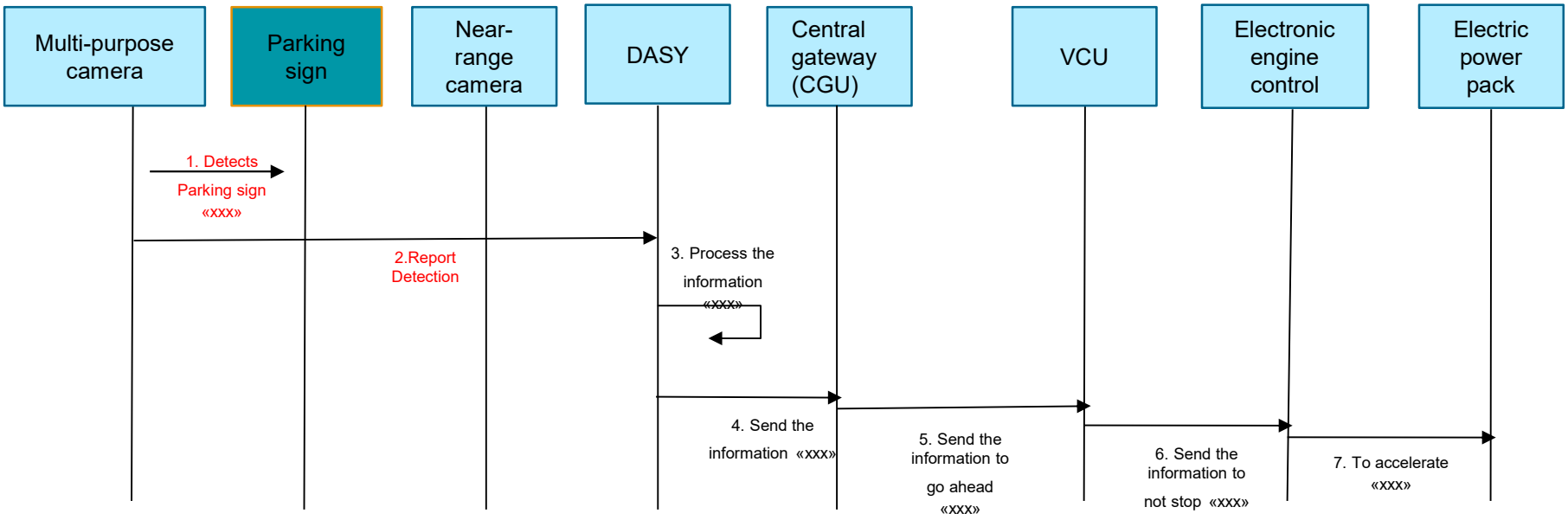
SQ2



White Box Sequence Diagram For SD1: UC01

No	Function	Stereotype	K	R	T/S	SecL	S	E	C	ASIL
1	Detect parking sign	Information	1	3	3	3	2	2	3	B
2	Report detection	Information	2	2	2	2	1	2	2	A
3	Process the information	Information	2	3	1	0	-	-	-	-
4	Send the information	Information	2	2	1	0	-	-	-	-
5	Send the information to go ahead	Information	2	2	1	0	-	-	-	-
6	Send the information to not stop	Information	2	2	1	0	-	-	-	-
7	To accelerate	Information	2	2	1	0	-	-	-	-

White Box Sequence Diagram For SD1



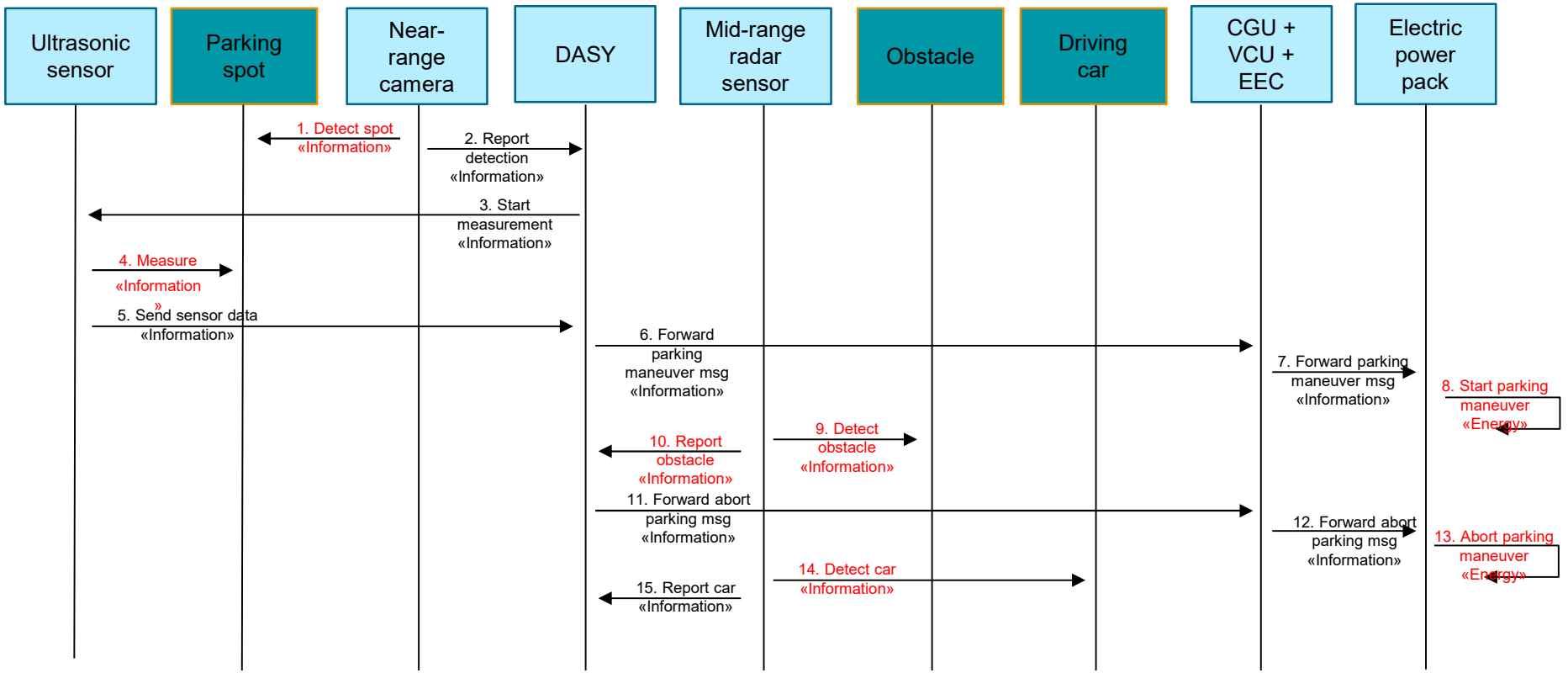
White Box Sequence Diagram For SD2: TC01

No	Function	Stereotype	K	R	T/S	SecL	S	E	C	ASIL
1	Detect spot	Information	1	2	3	2	3	1	3	A
2	Report detection	Information	2	2	2	0	-	-	-	-
3	Start measurement	Information	1	3	2	0	-	-	-	-
4	Measure	Energy	1	2	3	2	1	3	2	A
5	Send sensor data	Information	2	2	2	0	-	-	-	-
6	Forward parking maneuver msg	Information	2	2	2	0	-	-	-	-
7	Forward parking maneuver msg	Information	2	2	2	0	-	-	-	-
8	Start parking maneuver	Energy	2	2	3	1	2	4	1	A
9	Detect obstacle	Information	1	1	3	3	2	3	3	B
10	Report obstacle	Information	1	3	3	1	2	3	2	A
11	Forward abort parking msg	Information	2	2	2	0	-	-	-	-
12	Forward abort parking msg	Information	2	2	2	0	-	-	-	-

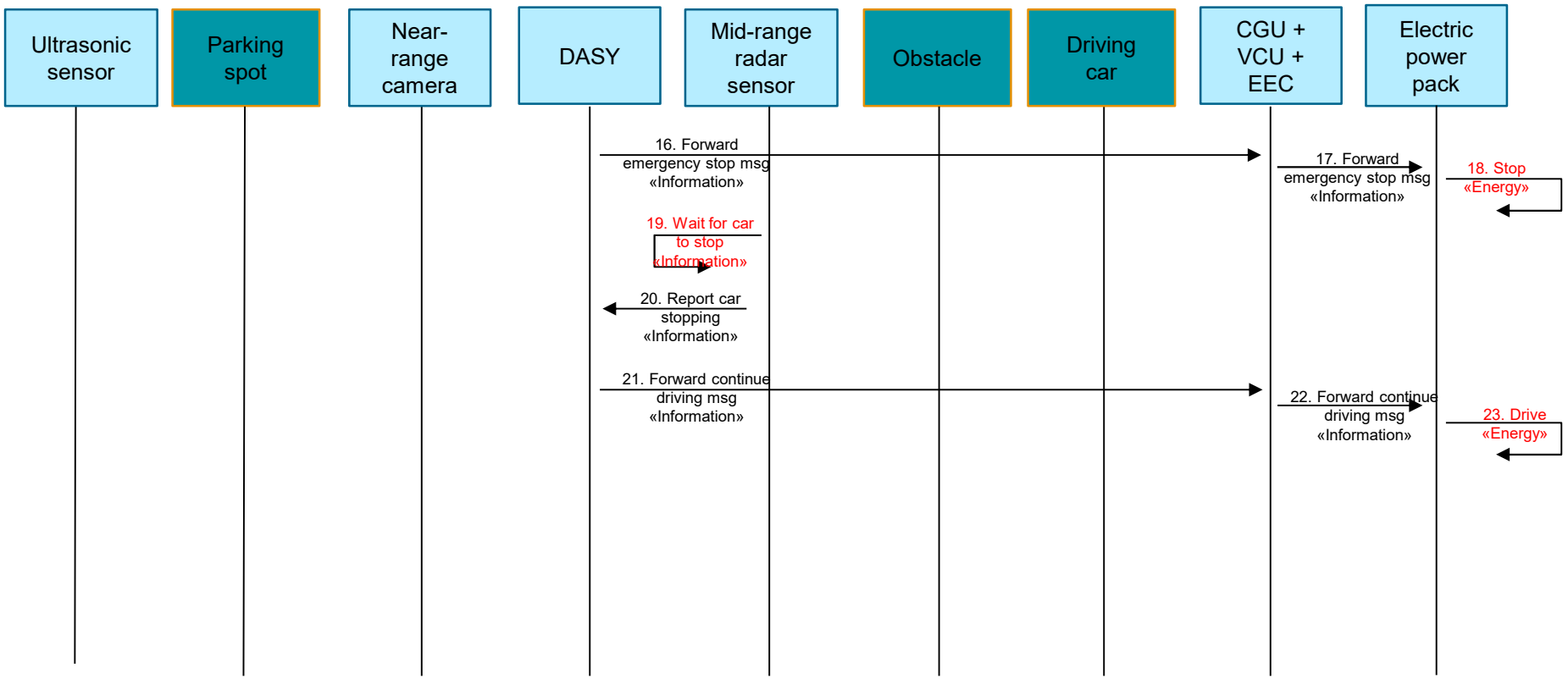
White Box Sequence Diagram For SD2: TC01

No	Function	Stereotype	K	R	T/S	SecL	S	E	C	ASIL
13	Abort parking maneuver	Energy	2	3	3	1	3	1	3	A
14	Detect car	Information	1	2	3	2	1	3	2	A
15	Report car	Information	2	2	2	0	-	-	-	-
16	Forward emergency stop msg	Information	2	2	2	0	-	-	-	-
17	Forward emergency stop msg	Information	2	2	2	0	-	-	-	-
18	Stop	Energy	2	3	3	1	3	1	3	A
19	Wait for car to stop	Information	1	2	3	2	1	3	2	A
20	Report stopping car	Information	2	2	2	0	-	-	-	-
21	Forward continue driving msg	Information	2	2	2	0	-	-	-	-
22	Forward continue driving msg	Information	2	2	2	0	-	-	-	-
23	Drive	Energy	2	3	3	1	3	1	3	A

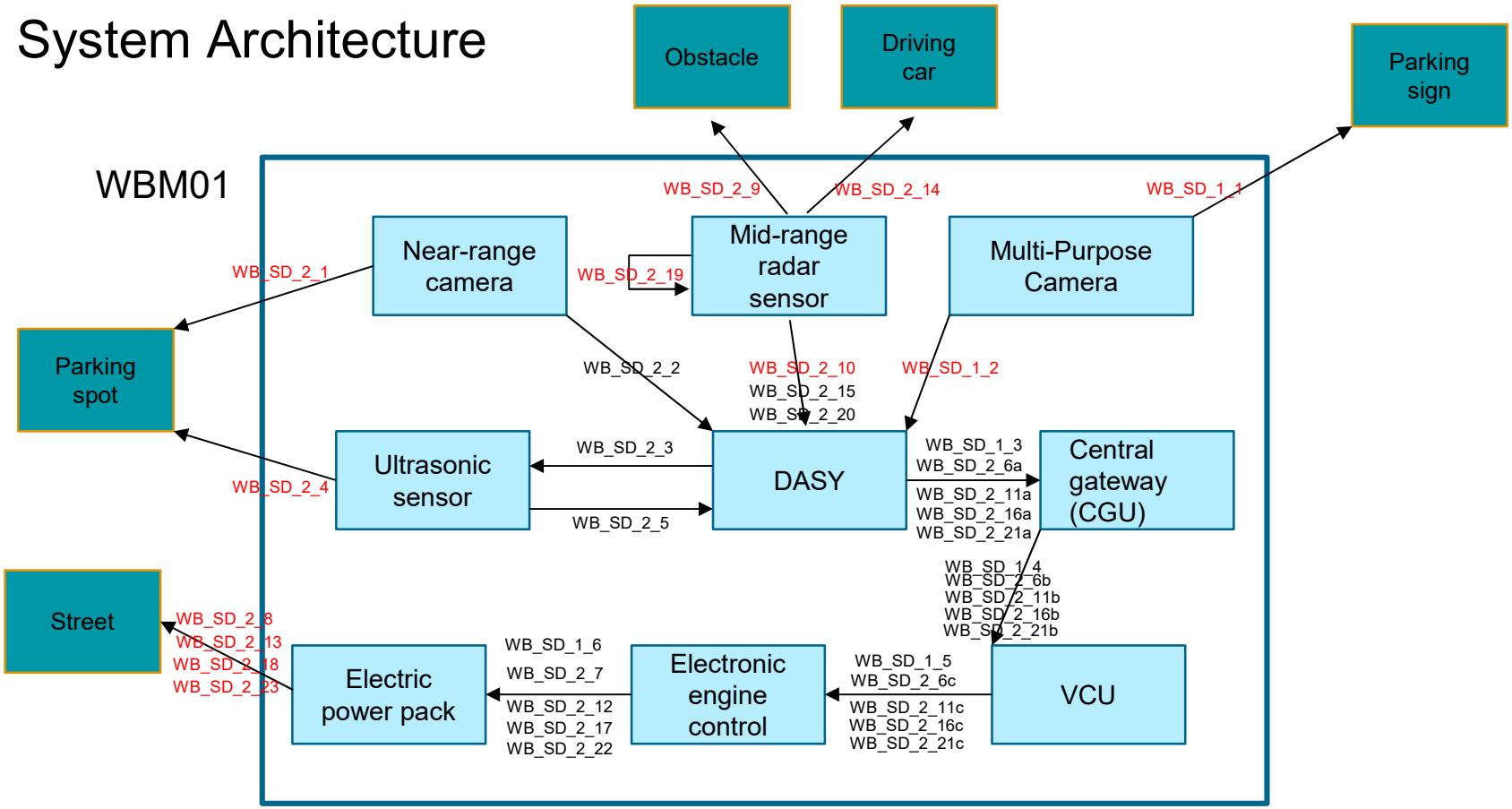
White Box Sequence Diagram For SD2



White Box Sequence Diagram For SD2



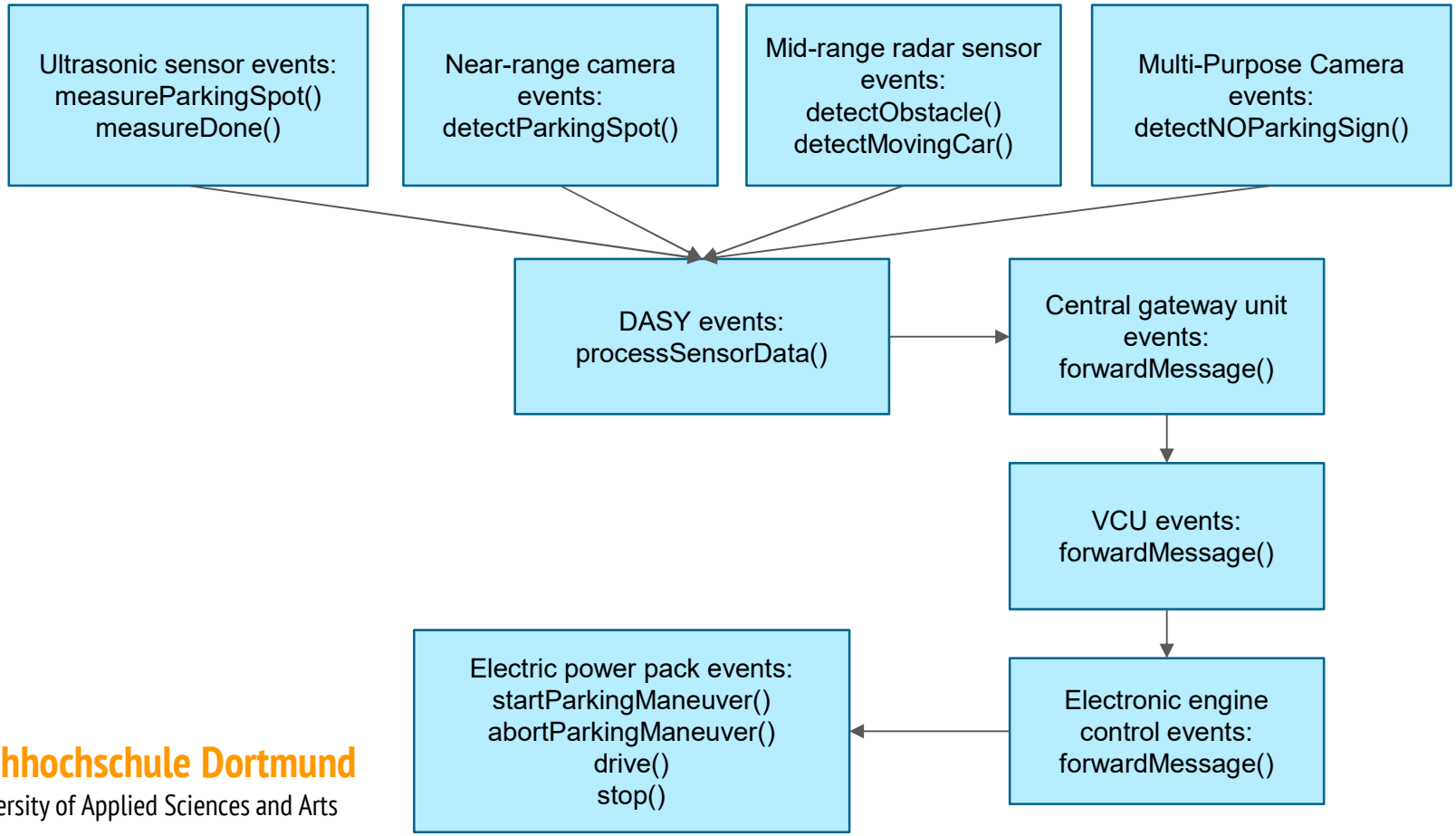
System Architecture



Requirement

		SAHARA	ISO 26262	Derived from			
ID	Description	Security Level (0-4)	Safety Level (0-4)	Requirements	System Architecture	Use Cases + Sequence Diagram	Threat Cases + Sequence Diagram
R01	The vehicle system shall be able to detect parking spot automatically, using a camera sensor system.	2	2	R01	WBM01	-	TC01, SQ2
R02	The vehicle shall be able to detect obstacle both in front and rear with a pre-defined range by using sensors	3	4	R02	WBM01	-	TC01, SQ2
R03	The vehicle shall be able to measure the distance to obstacles both in front and rear	3	4	R03	WBM01		TC01, SQ2
R04	The vehicle shall automatically stop within a pre-defined time in case the distance to obstacles less than a pre-defined distance	4	4	R04	WBM01		TC01, SQ2
R05	The vehicle shall be able to detect and understand the parking signs on the road	2	3	R05	WBM01	UC01, SQ1	
R06	In case user is searching for a parking spot, the vehicle shall NOT perform the parking action when "No parking sign" is nearby.	2	2	R06	WBM01	UC01, SQ1	
R07	In case user is searching for a parking spot, the vehicle shall perform the parking action automatically when a space is greater than a pre-defined number and the "No parking sign" is not nearby.	2	1	R07	WBM01	UC01, SQ1	
R08	The vehicle shall abort parking if obstacle is in parking spot	3	4	R08	WBM01	UC01, SQ1	
R09	The vehicle shall be able to make a decision to stop or move forward in case another vehicle is nearby	3	4	R09	WBM01		TC01, SQ2
R10	The decisions regarding detection must be based on information from the camera sensor system and the ultrasonic sensor system.	3	2	R10	WBM01	UC01, SQ1	TC01, SQ2
R11	The decision message must be forwarded to the Electric Power Pack.	1	2	R11	WBM01	UC01, SQ1	TC01, SQ2
R12	The vehicle shall be able to measure a detected parking spot using the ultrasonic sensor	2	2	R12	WBM01		TC01, SQ2

Event Definition



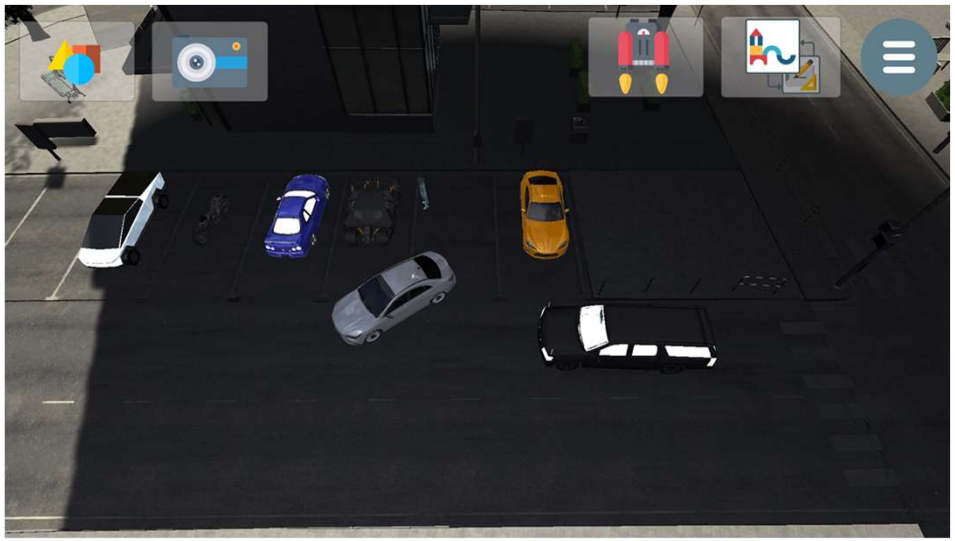
Example Event Definition

In ScenarioSpecification.kt

```
class MidRangeRadar{
    fun detectObstacle() = event(){}
    fun detectMovingCar() = event(){}
}

class DASY{
    var msg = " "
    fun processSensorData(message: String) = event(message){this.msg = message}
}
```

Test Case Scenario

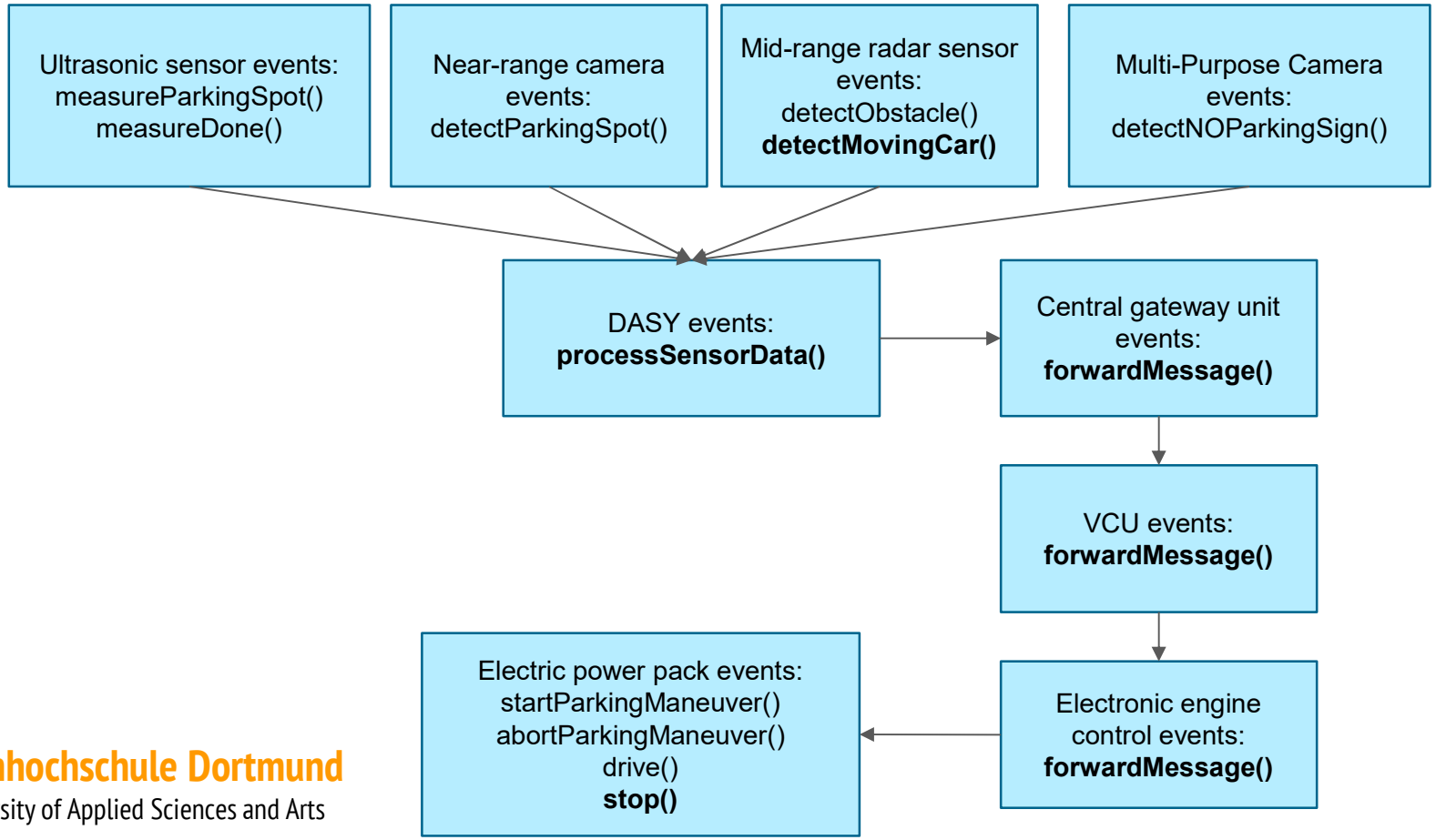
- Mid-range sensor detects moving car
- System propagates message
- Vehicle stops



Test case:

1. send event **detectMovingCar()** of mid-range sensor
2. evaluate event **stop()** of electric power pack

Event Chain


Ultrasonic sensor events:
measureParkingSpot()
measureDone()
Near-range camera events:
detectParkingSpot()
Mid-range radar sensor events:
detectObstacle()
detectMovingCar()
Multi-Purpose Camera events:
detectNOParkingSign()
DASY events:
processSensorData()
Central gateway unit events:
forwardMessage()
VCU events:
forwardMessage()
Electronic engine control events:
forwardMessage()
Electric power pack events:
startParkingManeuver()
abortParkingManeuver()
drive()
stop()

Scenario Definition

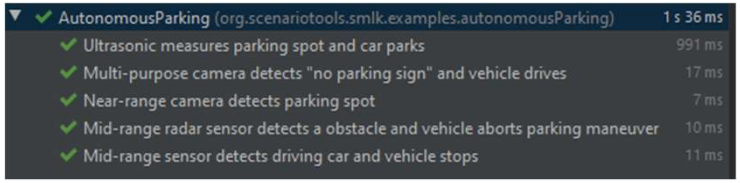
In ScenarioSpecification.kt under function guaranteeScenarios

```
scenario(midRangeRadar.detectMovingCar()){ this: Scenario
    request(dasy.processSensorData("movingCar"))
},
scenario(dasy.processSensorData("movingCar")) { this: Scenario
    request(centralGatewayUnit.forwardMessage("movingCar"))
    request(vehicleControlUnit.forwardMessage("movingCar"))
    request(electronicEngineControl.forwardMessage("movingCar"))
},
scenario(electronicEngineControl.forwardMessage("movingCar")) { this: Scenario
    request(electricPowerPack.stop())
},
```

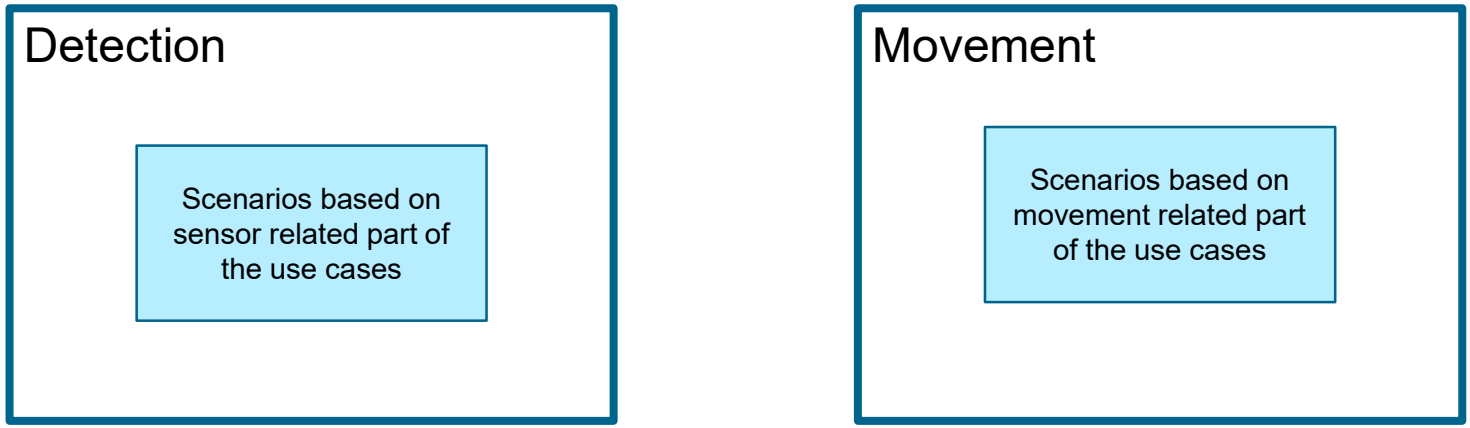
Test Case Definition

In TestSpecification.kt

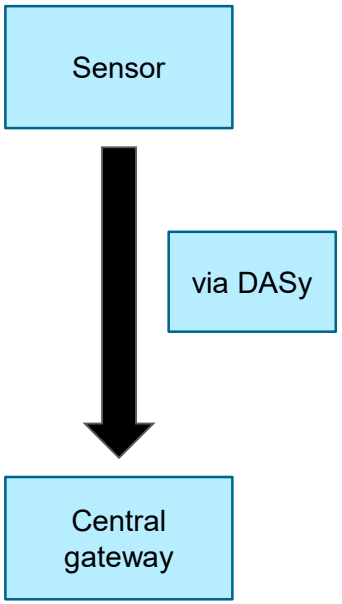
```
class AutonomousParking{
    init {
        connectSpecificationModel()
    }

    @Test(timeout=1000)
    fun `Mid-range sensor detects driving car and vehicle stops`(){
        send(midRangeRadar.detectMovingCar())
        eventually(electricPowerPack.stop(), electricPowerPack.drive())
    }

    @Test(timeout=1000)
    fun `Mid-range radar sensor detects a obstacle and vehicle aborts parking maneuver`(){...}

    @Test(timeout=1000)
    fun `Near-range camera detects parking spot`(){...}

    @Test(timeout=1000)
    fun `Ultrasonic measures parking spot and car parks`(){...}

    @Test(timeout=1000)
    fun `Multi-purpose camera detects "no parking sign" and vehicle drives`(){...}
}
```

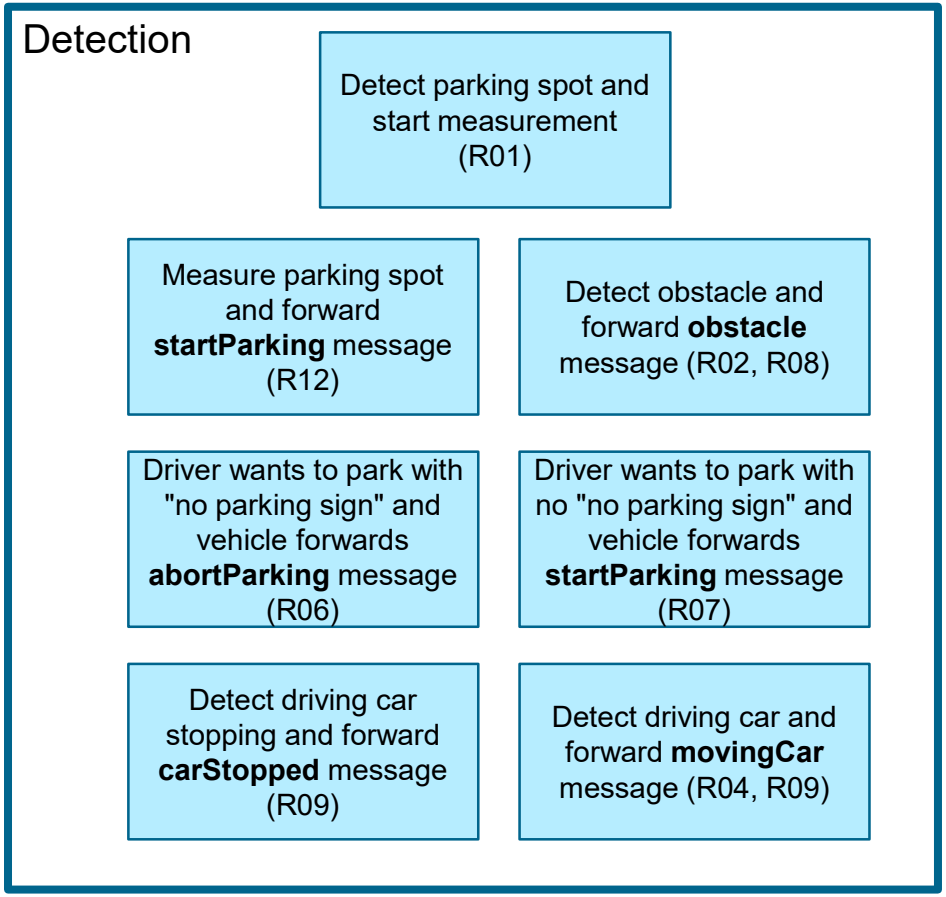


Scil feature definition

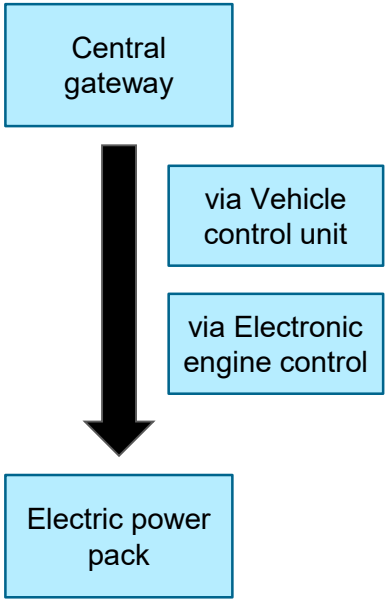


Detection feature





Movement feature



Movement

Message **startParking** is transmitted and the vehicle starts the parking maneuver (R07, R11)

Message **obstacle** is transmitted and the vehicle aborts the parking maneuver (R04, R08, R11)

Message **movingCar** is transmitted and the vehicle stops (R09, R11)

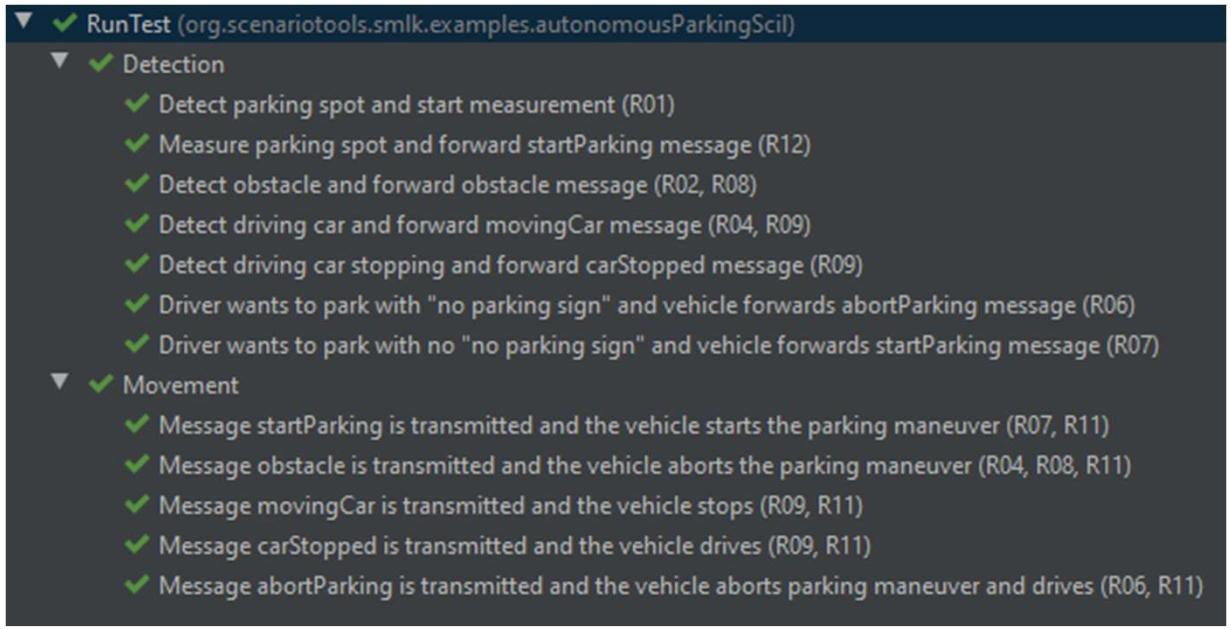
Message **carStopped** is transmitted and the vehicle drives (R09, R11)

Message **abortParking** is transmitted and the vehicle aborts parking maneuver and drives (R06, R11)

Example feature definition

```
Feature: Movement
    Background: init test setup
        Given init test setup

    Scenario: Message startParking is transmitted and the vehicle starts the parking maneuver (R07, R11)
        When the central gateway forwards startParking message
        Then the vehicle starts parking maneuver

    Scenario: Message obstacle is transmitted and the vehicle aborts the parking maneuver (R04, R08, R11)
        When the central gateway forwards obstacle message
        Then the vehicle aborts parking maneuver

    Scenario: Message movingCar is transmitted and the vehicle stops (R09, R11)
        When the central gateway forwards movingCar message
        Then the vehicle stops

    Scenario: Message carStopped is transmitted and the vehicle drives (R09, R11)
        When the central gateway forwards carStopped message
        Then the vehicle drives

    Scenario: Message abortParking is transmitted and the vehicle aborts parking maneuver and drives (R06, R11)
        When the central gateway forwards abortParking message
        Then the vehicle drives
```

Scil test cases



Summary

- We learnt how to visualize various scenarios related to automated driving using 3-D engineer tool. We identified the System and environment interactions.
- We were able to derive the use cases/ threat cases from the scenarios which were visualised.
- Modelled the black box and white box sequence diagrams for the use case and threat case.
- Developed the System architecture and derived detailed requirements for the test cases.
- Learnt Scenario modelling Language and were able to verify and validate our test cases by using Test driven scenario specification.
- We also learnt definition of feature files and validated our test cases by behavioral driven development.

Thank You