FACE RECOGNITION WITH DSMC (DRIVER STATE MONITOR CAMERA)

SPECIFICATION

Issued by (dept, name, phone)
EESE China
Elaine Jia, +86-025-84922287
Yu Diven

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Purpose

This specification document is used to describe and specify the behaviour of the Driver State Monitor Camera (DSMC) to support various in driver recognition.

The requirements discussed in this document will describe all aspects of the driver recognition with DSMC, including interface, performance and functional.

1.2 Document Organization

1.2.1 Document Structure

The structure of this document is explained below:

Section 1 – Introduction how to use this document, including responsibilities and requisite documents. Explains the terminology. Gives a clarification of the definitions, concepts, and abbreviations used in the document.

Section 2 – Technology Description. States briefly the background and the purpose of the technology.

Section 3 – Technology Requirements, which defines the boundaries of the feature and includes functional/non-functional requirements of the feature

Section 4 – Manufacturing requirements

Section 5 – Revision History

1.2.2 Ford Documents

List here all Ford internal documents, which are directly related to the feature.

Table 1.2.2-1: Referenced Ford Documents

Reference	Title	Doc. ID	Revision
AN-0340	CAN Signals Rolling Counter Requirements	CAN RC	2010-05-21
	Driver State Monitor Camera Specification	RQT-190402-705877	

1.2.3 External Documents And Publications

The list of external documents could include books, reports and online sources.

Table 1.2.3-1: Referenced External Documents

Reference	Document/Publication
ISO 8855:2011	Road vehicles – Vehicle dynamics and road-holding ability
IEC 62471	Photobiological safety of lamps and lamp systems

1.3 Terminology

1.3.1 Definitions

1.3.1.1 Shall, Should, May Definition

This document contains both explanatory text and explicit requirements. The word "shall" denotes a mandatory design requirement. The word "should" denotes a recommended design requirement. The word "may" denotes an optional design requirement.

1.3.1.2 ISO Vehicle Coordinate definition

For the purpose of this requirement orthogonal body-fixed axes are defined such that, for a vehicle on flat ground, the origin is located at the geometric center of the camera imager, the longitudinal (X) axis points forward, the vertical (z) axis points up, and the lateral (Y) axis points to the left.

This specification follows the ISO standard vehicle coordinates as shown in the Figure 1.3.1-1 below:
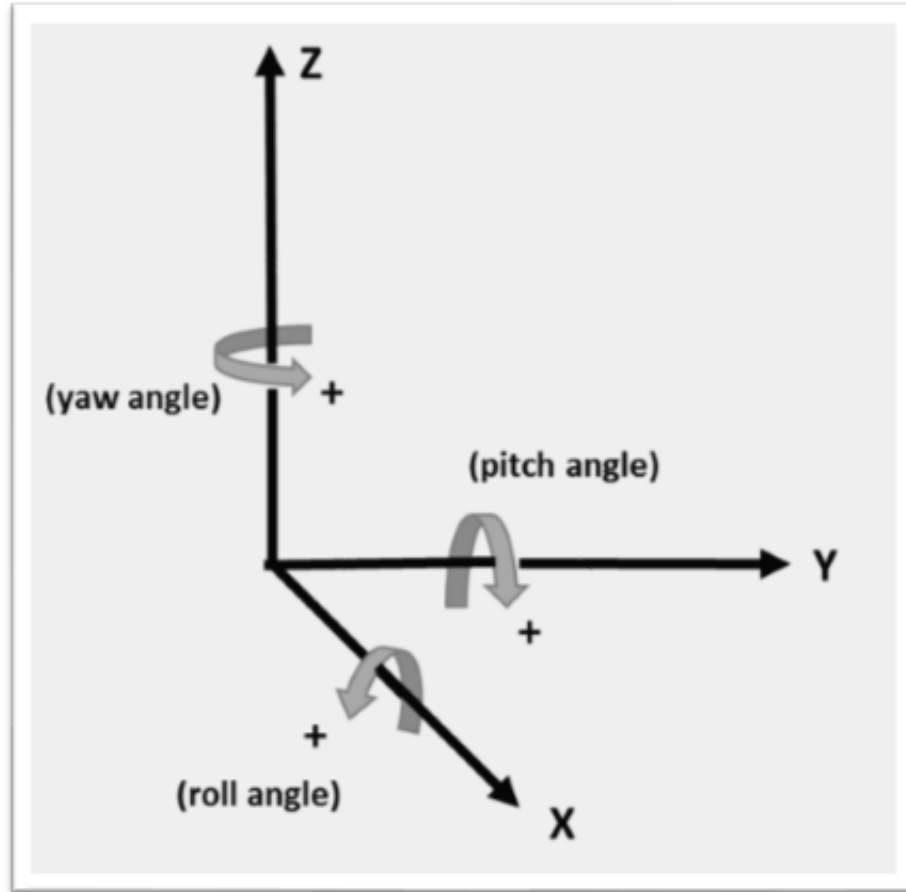
Figure 1.3.1-1: ISO Coordinate System for DSMC

The point of origin used for the coordinate system for the detection and tracking of head, pupils, and other necessary objects is defined as follows:

- Longitudinal (x): Focal point of the camera
- Lateral (y): Lateral center of the vehicle (independent of camera mounting location)
- Vertical (z): Focal point of the camera

1.3.1.3 Availability

If the driver's head is within the defined field of view (Section 5.1), all features and signals shall meet the requirements outlined in this document for the specified percentage of the time the head is detected.

1.4 Regions & Markets

Table 1.3.1-1: Applicable Regions for Driver Recognition Variants

Market / **Region** **Variant Name**	North America	South America	Europe	Middle East / Africa	Asia / Pacific	China
Driver Recognition (All variants)	-	-	-	-	-	X

1.5 Abbreviations

Table 1.3.1-1: Abbreviations

Abbr.	Stands for	Description
DSMC	Driver State Monitor Camera	System consist of camera, IR illuminator(s), and processing controller
FMM	Fault Mode Management	
FOV	Field of View	Space within the vehicle where the camera must meet output performance criteria
IVI	in-Vehicle Infotainment	

Abbr.	Stands for	Description

2 Architectural Design

2.1 Overview of Technology

Facial recognition, is a feature that use the driver state monitor camera to track the driver's face for recognizing the driver's identification. The DSMC as server will provide the enrollment and recognition services for the applicaton in IVI (Client). DSMC will transfer the driver's identification on CAN bus to IVI.

2.1.1 Driver Recognition Client (DRC)

The IVI as client provides the user interface to request enrollment, manager enrolled data and also with any warning and errors, such as IVI internal system errors, DSMC syste errors and so on.

2.1.2 Driver Recognition Server (DRS)

The DSMC as server is responsible for the tasks listed below:
1. Utilizing camera equipped with IR sensors and illuminators for driver face.
2. Doing enrolment of new driver.
3. Recognizing driver's identification:
4. Providing interfaces to manager all enrolled driver models with relevant information.
5. Outputing errors.

2.2 Boundary Diagram

This boundary diagram defines the interfaces between Client and the Server with respect to the following features:
- Vehicle CAN signals
- Imager and IRs LEDs
- Driver recognition algorithms
- Driver recognition services in Client.
- Configuration signals
- Diagnostics

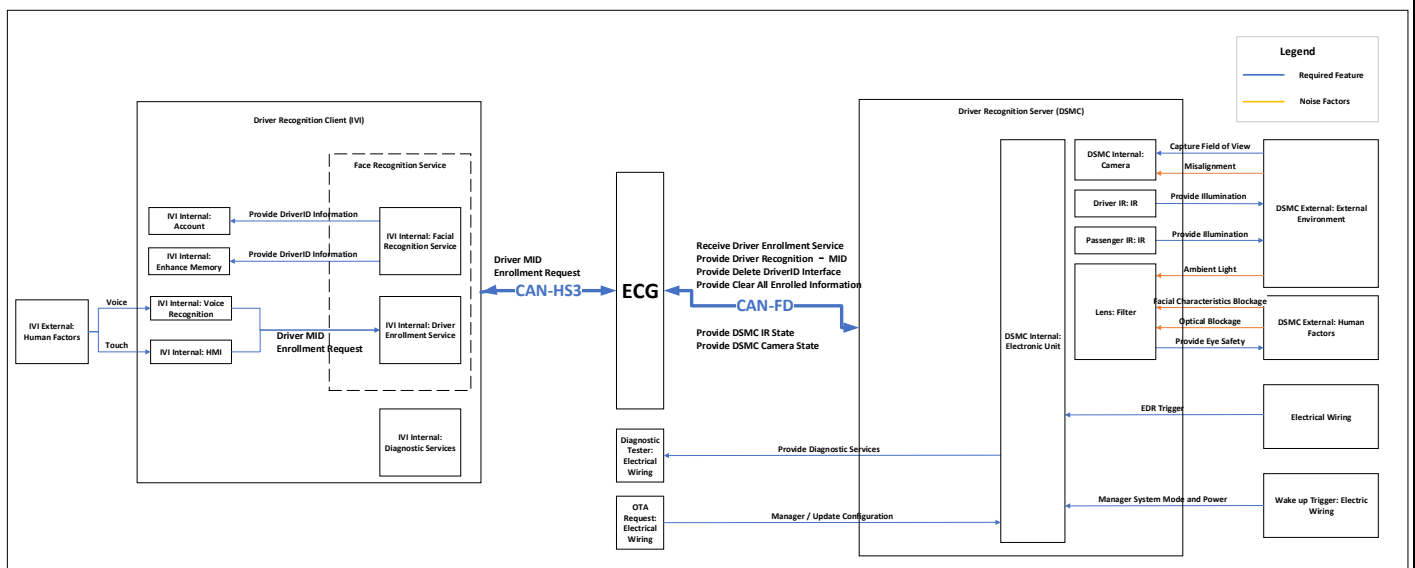**Figure 2.2 : Feature Boundary Diagram**

2.3 General Requirements

2.3.1 Requirement Compliance
Requirements (R2.3.1-x):
1. The driver recognition server shall complete and submit to Ford a design requirement compliance matrix.

EESE

2. The Driver Recognition Server shall include design requirement compliance evidence in the design requirement compliance matrix.

2.3.2 Feature Requirements

Requirements (R 2.3.2 – x):
1. The Driver Recognition Server shall be triggered to do enrolment of new driver data (Driver model) and respond with an identification number (MID – Model ID).
2. The Driver Recognition Server shall be capable of clearing all enrolled driver models with relevant information.
4. The Driver Recognition Server shall be capable of deleting a specified MID.
5. The Driver Recognition Server shall be capable of transferring all system and camera errors (defined in Error Handling) to Driver Recognition Client on CAN bus.
6. The Driver Recognition Server shall be triggered to start up VP or skip from Awake Mode to Sleep by Driver Recognition Client.

2.3.3 Software reflash and Over the Air Update

Requirements (R2.3.3 – x):
1. In OTA process or software reflash by special tool, driver recognition server shall meet the requirements below:
   1) Firstly, driver recognition server shall meet the requirements delineated in DSMC specification (RQT-190402-705877).
   2) The enrolled information saved in database shall be kept while DSMC implements update by Over-The-Air (OTA) or Software reflash.
   3) The driver recognition server shall calculate a checksum for Driver ID database. Checksum value will be sent along with Driver ID recognized result through CAN. The Checksum shall be performed based on Driver ID dataset for all 10 driver proifile slots.
   4) The driver recognition server shall provide the Checksum calculation method to the driver recognition client to check if the received Checksum is correct or not.

2. In OTA process or software reflash, driver recognition client shall meet the requirements below:
   1) The enrolled information shall be kept while the client implements update by Over-The-Air (OTA) or Software reflash.
   2) The driver recognition client shall request the driver recognition server to check data consistency.
   3) The driver recognition client shall compare checksum to verify if the database between the driver recognition server and the driver recognition client are still align before accepting the Driver ID signal.
   4) The driver recognition client shall clear all the enrolled data in database and request the driver recognition server to clear data while the data consistency is fail.

2.3.4 End of Line Configuration

The Driver Recognition Server and the Driver Recognition Client shall meet the configuration requirements as below:

Requirements (R2.3.4 - x):
1. The Driver Recognition Server shall meet the config requirements in below Table:

Parameters Requirement	value
DriverID_Enable_Status	Disable (Default)
	Enable
Recognition_Max_Observation_Time	[0.0,30.0] in 2 second intervals
	Default value: 2s
Recognition_Timeout	[0.0,30.0] in 2 second intervals
	Default value: 14s

Enroll_Timeout	[0.0,30.0] in 2 second intervals Default value: 10s
Left blank to Add	

2. The Driver Recognition Client shall meet the config requirements in below Table:

Parameters Requirement	value
Facial_Recognition	Disable (Default)
	Enable
CRM-DSMC	Not Present (Default)
	Present
Left blank to Add	

3 Logical Signal

The logical signals output from Driver Recognition Server (DSMC) are defined in table below:

3.1 Logical outputs of DSMC

1. Logical name and physical name

ID	Logical Name	Physical Name
1	DriverId_Mode_Status	DrvIdMde_D_Stat
2	PoseExpression_Constrains_Check	DrvExpression_D_Stat
3	DriverId_Recognition_Result	DrvId_D_Stat
4	DriverId_Recognition_Mid	DrvIdMatch_No_Actl
5	DriverId_Data_Consistent_Checksum	DrvId_No_Cs
6	DriverId_Action_Result	DrvIdActn_D_Stat
7	DrvCamera_D_Stat	DrvCamera_D_Stat
8	DrvCamDrvIR_D_Stat	DrvCamDrvIR_D_Stat
9	DrvCamPassIR_D_Stat	DrvCamPassIR_D_Stat

2. Signal definition

ID	Name	Description	Data Type	Values	Notes
1	DriverId_Mode_Status	Response signal stating the current operational mode of driver identification function.	enum	Null, Recognize, Enroll, ClearAll, DeleteMid	Null - feature is disabled, Enroll - feature is attempting to enroll a driver, Recognize- feature is performing the matching process.
2	PoseExpression_Constrains_Check	Current status of enrollment process. Vehicle application can read the status while enrollment process is in progress.	enum	Null, FaceNotInView, TooManyFaces, OutsidePoseLimits, MouthOpen, FaceBlockage, EyesNotVisible, PoorTrackingQuality, NotAFace, ConstraintsOk, HasRegister, FailNoSlot, FailMultipleMatches	Null – when enrollment or recognition function is not active ConstraintsOk – face image has passed all constraints for enrollment. Enrollment is in progress NotAFace is set when the spoofing is detected (liveness). OutsidePoseLimits: when PE constraints are not met and recognition is prevented. MouthOpen: PE constraints are not met due to driver mouth wide open. FaceBlockage: driver face is blocked. PoorTrackingQuality: is reported when tracking confidence being too low.
3	DriverId_Recognition_Result	Shows the result of recognition process.	enum	Null, OkMatched, OkNoMatch, FailTimeout	Null - when no recognition has yet occurred, FailTimeout is set when the system times out waiting for PE Constraints to be met.
4	DriverId_Recognition_Mid	MID of the driver when	32bit		

ID	Name	Description	Data Type	Values	Notes
		recognition result is matched			
5	DriverId_Data_Consistent_Checksum	The signal is the driver mode ID(MID) which is matched by recognition algorithm.	8 bit		
6	DriverId_Action_Result	Result of enroll, deleteMid and clear all data process.	enum	Null, Fail, Success	Null - Default value when no action. Fail - When enroll_result = FAIL TIMEOUT, detailed reason will be in check_PE_status. Success – when enroll_result = OK NEW DRIVER MODEL ADDED delete_result = OK clear_result = OK.
7	DrvCamera_D_Stat	Camera error		TemporaryUnavailable, Ok, Blocked, Faulty	
8	DrvCamDrvIR_D_Stat	Status of the IR light on the driver's side	enum	TemporaryUnavailable, Ok, Blocked, Faulty	
9	DrvCamPassIR_D_Stat	Status of the IR light on the passenger's side	enum	TemporaryUnavailable, Ok, Blocked, Faulty	
	Left blank to Add				

3.2 Logical inputs of DSMC

1. Logical name and physical name

ID	Logical Name	Physical Name
1	DriverId_Mode_Request	DrvIdMde_D_Rq
2	Delete_DriverId_MID	DrvIdDelete_No_Rq
	Left blank to Add	

2. Signal definition

ID	Name	Description	Data Type	Values	Notes
1	DriverId_Mode_Request	External command to set the operational modes	enum	Disable, Recognize, Enroll, ClearAll, DeleteMid	ClearAll – clear all data stored in DSMC database. DeleteMid – delete all data related to MID which received from signal Delete_DriverId_MID
2	Delete_DriverId_MID	MID of the driver when recognition result is Matched.	numeric 32 bit		
	Left blank to Add				

4 Functional REQUIREMENTS

4.1 Driver Recognition Operational Mode

4.1.1 Digram
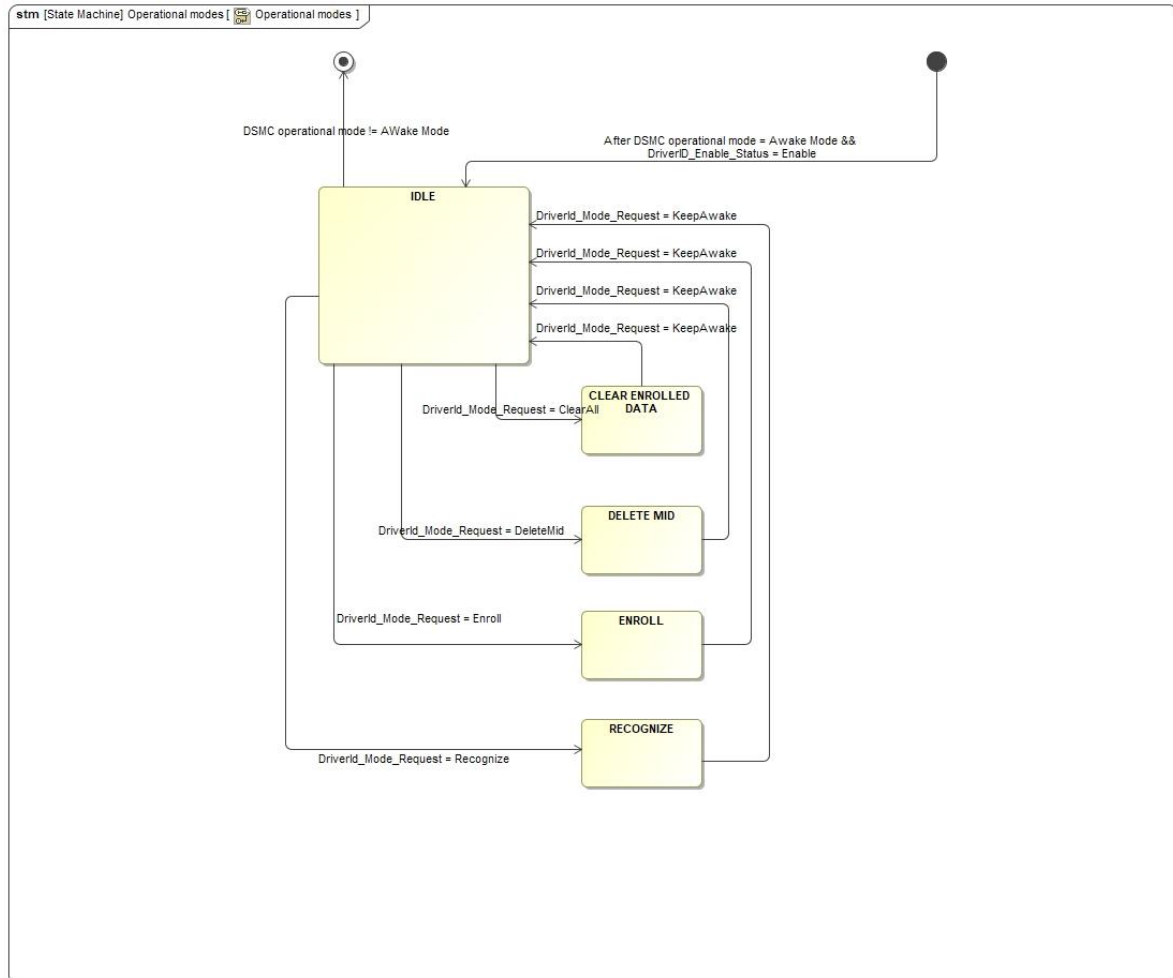
Requirements:
1. The Driver Recognition Server shall start to run Driver ID function within 8 seconds of wakeup trigger.
2. If the config parameter DriverID_Enable_Status=Enable, the DRS shall start to recognize the driver and the Driver ID recognition result shall be outputed within 3 seconds after the DRS is fully operational and receives the request from IVI with driver looking forward.
3. The driver recognition operational modes include:
   - IDLE
   - RECOGNIZE
   - ENROLL
   - CLEAR ENROLLED DATA
   - DELETE MID

4. When DRS receives DriverId_Mode_Request=KeepAwake, the Driver Recognize Server shall enter IDLE mode.
5. The jump conditions between IDLE, RECOGNIZE, CLEAR ENROLLED DATA, DELETE MID, ENROLL and RECOGNIZE are defined in table below:

Table: Jump conditions between IDLE, INACTIVE and RECOGNIZE

From	Results Mode	Condition

Initial state	RECOGNIZE	And	(DRS operational_mode == Awake_Mode)
			(DriverID_Enable_Status == Enable)
RECOGNIZE	IDLE		(DriverId_Mode_Request == KeepAwake)
IDLE	CLEAR ENROLLED DATA		(Driverid_Mode_Request == ClearAll)
CLEAR ENROLLED DATA	IDLE		(DriverId_Mode_Request == KeepAwake)
IDLE	DELETE MID		(Driverid_Mode_Request == DeleteMid)
DELETE	IDLE		(DriverId_Mode_Request == KeepAwake)
IDLE	ENROLL		(Driverid_Mode_Request == Enroll)
ENROLL	IDLE		(DriverId_Mode_Request == KeepAwake)
IDLE	RECOGNIZE		(DriverId_Mode_Request == KeepAwake)
IDLE	exit	Or	(DRS operational_mode != Awake_Mode)
Left blank to Add			

4.2 Enable / Disable driver recognition feature

The Driver Recognition Client shall enable/disable the feature status by user and the Driver Recognition Server shall stop running the Driver ID functions while (Driverid_Mode_Request = Disable) is received from Driver Recognition Client.

Requirements

1. The Driver Recognition Server shall provide the EOL config parameter (DriverID_Enable_Status) for enabling or disabling the Driver Recognition feature, the default value is Disable.
2. The Driver Recogniton Client shall provide the EOL config parameter (facial_recognition and CRM-DSMC) to indicate if the driver recognition function is enable or not in Driver Recognition Server.
3. The Driver Recognition Client shall enable/disable the driver recognition function by user.
4. The Driver Recognition Server shall disable the driver recognition function after completing the current process or timeout while (DriverId_Mode_Request = Disable) is received from Driver Recognition Client.
5. The Driver Recognition Server shall transfer the status (DriverId_Mode_Status) of the currently executed function by CAN bus timely, the status include Null / Recognize / Enroll / ClearAll / DeletedMid. The value of Null means there is no process in Driver Recognition Server.
6. The DRS shall output the Null result in 2s through (DriverId_Mode_Status) to DRC.

4.2.1 **Dataflow Diagram**
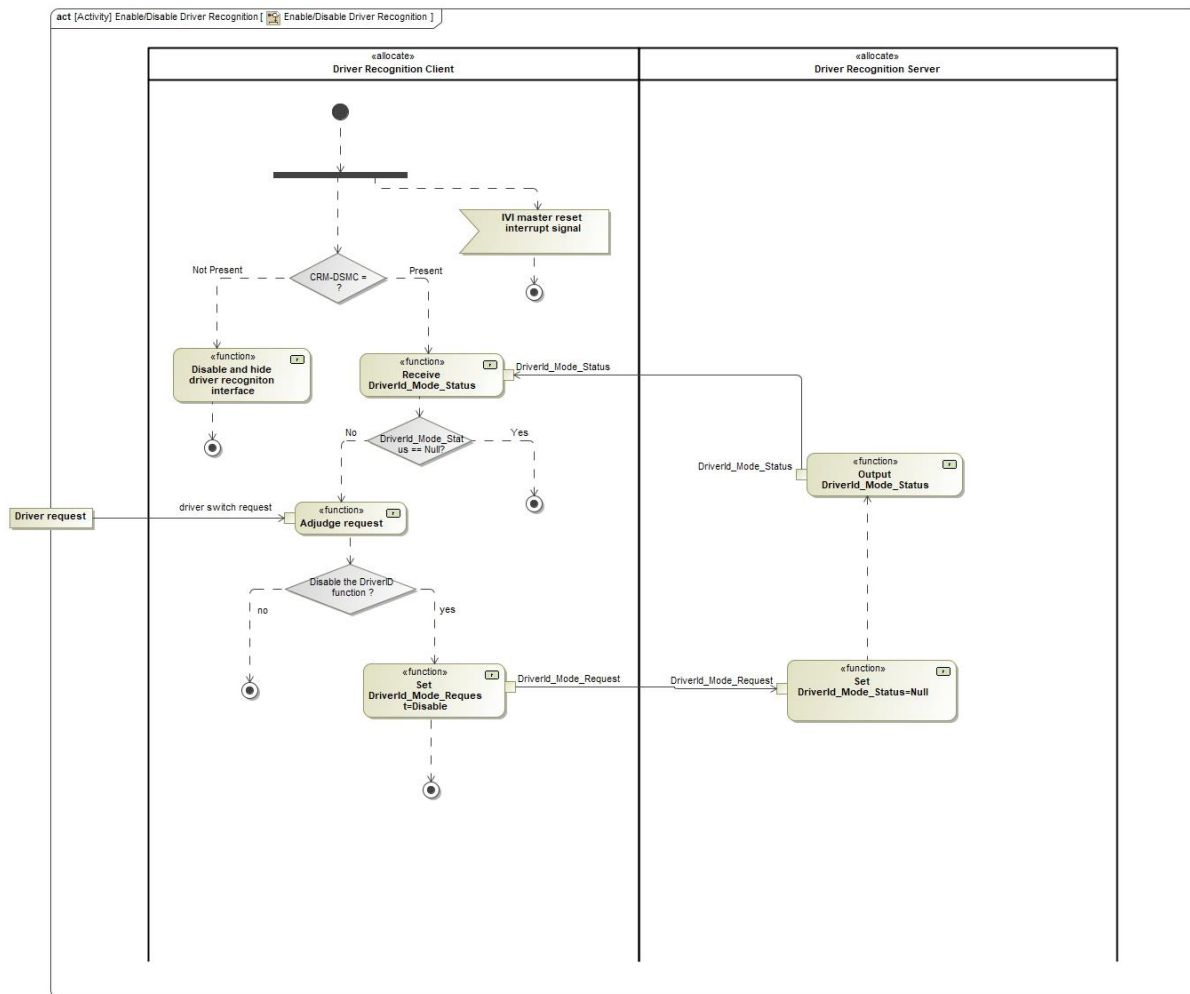
4.2.2 **Functions in Driver Recognition Server**

Fun_ id	Fun_Name	Description
1	Output DriverId_Mode_Status	The function outputs the status of facial recognition function process status to IVI through CAN.

4.2.3 **Functions in Driver Recognition Client**

Fun_ id	Fun_Name	Description
1	Receive DriverId_Mode_Status	The function allows the driver recognition client to get the currently executed function in driver recognition server from CAN.
2	Set DriverId_Mode_Request=Disable	The function allows user to disable the facial recognition feature.

4.3 Enrollment Process

The enrollment process mainly implements tracking driver face and capture biometric observation to generate a driver model and an unique identified model ID stored in the Driver Recognition Server database.

Requirement for Driver Recognition Server (DRS)

1. The DRS shall provide the interface for vehicle application in the Driver Recognition Client (DRC) to trigger the enrollment process.
2. The DRS shall provide the EOL config parameters (Enroll_Timeout) for control the enroll process. The default value of Enroll_Timeout is **12** seconds. When the DRS receives the Enroll request from DRC, the DRS shall output the enrollment result before time out as soon as possible.
3. The DRS shall output (DriverId_Mode_Status = Enroll) if it receives the enrolled request (DriverId_Mode_Request = Enroll) from DRC and start the enrollment process.
4. The DRS shall provide enough storage slots for 10 driver models at least.
5. The DRS shall output (PoseExpression_Constrains_Check = FailNoSlot and DriverId_Action_Result = Fail) while there is no slot for enrolling a new driver ID.
6. In the process of enrolling MID, it is necessary in DRS to detect the spoofing behaviors that include 2D simulated face models and the DRS shall output (PoseExpression_Constrains_Check = NotAFace and DriverId_Action_Result = Fail) and stop enrollment as long as the spoofing behaviors are detected.
7. In the enroll process, DRS shall output the enrolled status of pose and emotion detection on CAN bus (PoseExpression_Constrains_Check) continuously, the detected status and the priority of outputting rule are defined as follows: (0: The highest priority)

Detected&Exported Priority	Detected Action	Description
0	NotAFace	When the spoofing is detected (liveness)
1	FailNoSlot	
2	HasRegister	
3	FaceNotInView	
4	TooManyFaces	
5	FaceBlockage	
6	MouthOpen	
7	EyesNotVisible	
8	OutsidePoseLimits	When PE constraints are not met and the process is prevented.
9	PoorTrackingQuality	When tracking confidence is too low.
10	ConstraintsOk	

8. The DRS shall detect if the driver has already registered (PoseExpression_Constrains_Check = HasRegister) and reject the duplicated enrollment request (DriverId_Action_Result = Fail).
9. The DRS shall stop the enrollment process while the pose and emotion detection is timeout and output (DriverId_Action_Result = Fail).
10. While the enrollment process was completed, DRS shall output the enrolled results (DriverId_Action_Result) on CAN bus timely, the value of this signal includes:
    - Null
    - Fail
    - Success
11. The DRS shall update the checksum based on the newest database and transmit it with the enrollment result by "DriverId_Data_Consistent_Checksum".

Requirement for Driver Recognition Client (DRC)

1. The Driver Recognition Client shall reject the enrolled request from user while (DriverId_Mode_Status <> Null) or any camera/IR issues occurs (DrvCamera_D_Stat, DrvCamDrvIR_D_Stat and DrvCamPassIR_D_Stat), otherwise the DRC shall transfer the (DriverId_Mode_Request = Enroll) while user needs to enroll the driver ID.
2. The DRC shall monitor the status of (PoseExpression_Constrains_Check) while (DriverId_Mode_Status = Enroll), if the (PoseExpression_Constrains_Check = FailNoSlot), the DRC shall stop the enrollment timely and feedback the result to user.
3. The DRC shall monitor the status of (PoseExpression_Constrains_Check) while (DriverId_Mode_Status = Enroll), if the PE constrains is not met the DRC shall remind the user to correct posture timely.

4. The DRC shall remind the failed enrollment while (PoseExpression_Constrains_Check = NotAFace and DriverId_Action_Result = Fail) and stop the enrollment timely.
5. The vehicle application in driver recognition client shall guide drivers to complete the following actions during the enrollment process.
6. The DRC shall remind the failed enrollment while (PoseExpression_Constrains_Check = HasRegister) and stop the enrollment timely.
7. The DRC shall store the enrolled MID received from (DriverId_Recognition_Result) in database and maintain the association with in DRC.
8. The DRC shall set a 20s timer to wait the enrolled result while it sends (DriverId_Mode_Request = Enroll). If it is timeout the DRC shall stop the enrollment and feedback the failed result to user.
9. The DRC shall monitor the signal values of (DrvCamera_D_Stat, DrvCamDrvIR_D_Stat and DrvCamPassIR_D_Stat) while request DRS to execute enrollment process. If any issue of camera or IR light is occurred, DRC shall prompt to driver.
10. The DRC shall calculate checksum and verify database consistency after enrollment is completed successfully.
11. If user stops the enrollment process before DRS feedback the result, the DRC shall send a Delete request after receiving the successful result in the background to keep consistency of database.

Dataflow Diagram
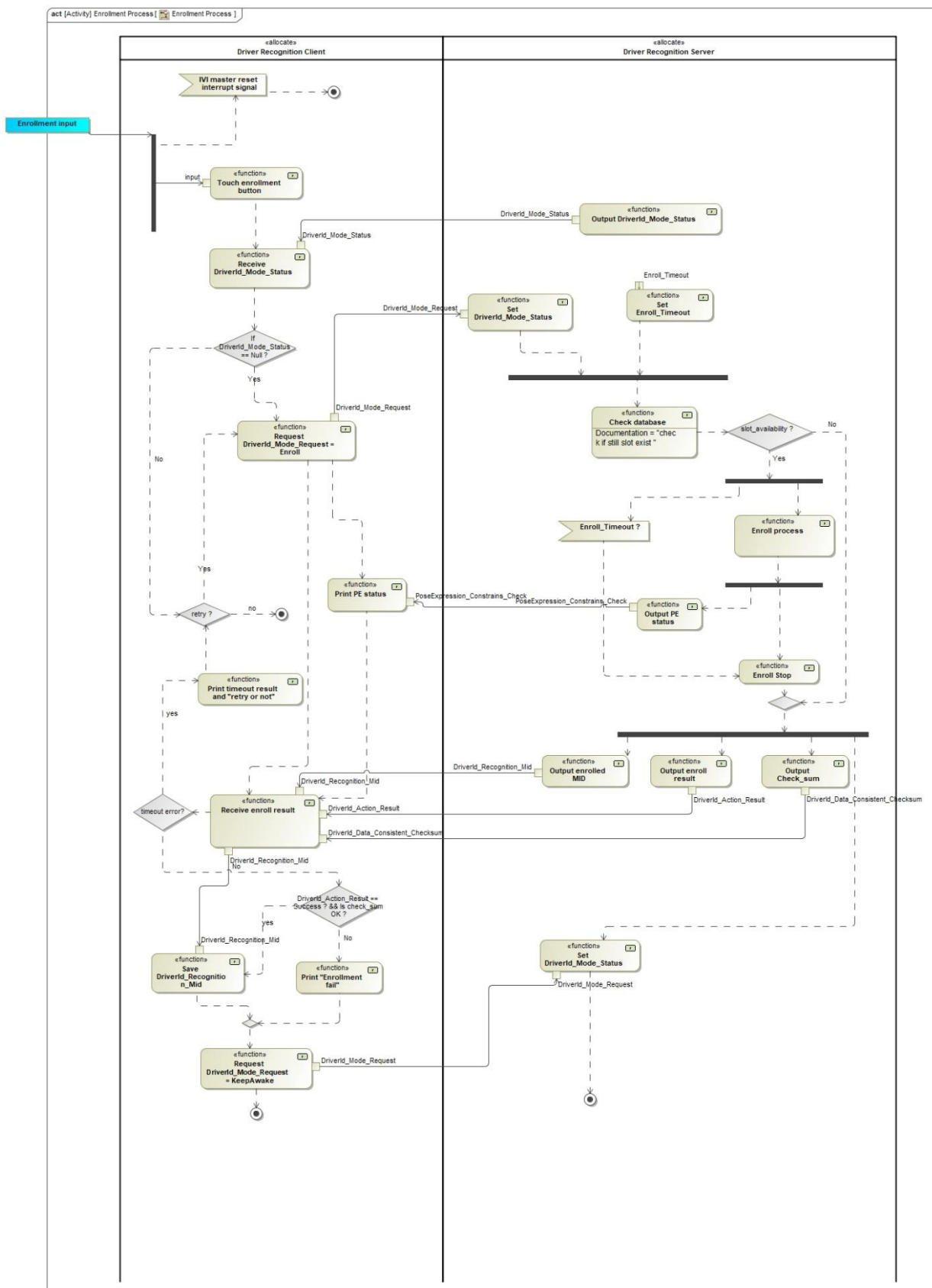
4.3.2 Functions in Driver Recognition Server

4.3.2.1 Functions Description

Fun_id	Fun_Name	Description
1	Output DriverId_Mode_Status	This function is used to output the current operational mode executed in driver recognition server. The states of operational modes include Disable, Recognition, Enroll, ClearAll, DeleteMid.
2	Set Driver_Mode_Status	This function is used for DRS to confirm it has entered Enrollment process and communicate it to DRC.
4	Set Enroll_Timeout	The driver recognition server shall get the config parameter enroll_PE_timeout which is the time limit for PE constraints while the enrollment is requested.
5	Output PE status	This function is used to output the PE check status regularly before enrollment completed.
6	Output enrolled MID	This function is used to output the new generated model ID through CAN.
7	Output enroll result	This function is used to output the result of the enrollment process, such as Null, Fail, Success.
8	Output Check_sum	This function is used to calculate the newest checksum based on the whole database and output to DRC.

4.3.3 Functions in Driver Recognition Client

4.3.3.1 Functions Description

Fun_id	Fun_Name	Description
1	Receive DriverId_Mode_Status	This function is used to get the current operational mode executed in driver recognition server. If the current mode is not Null, the driver recognition client shall not request the DRS to execute enrollment.
2	Request DriverId_Mode_Request = Enroll	This function is used to request driver recognition server to do enrollment.
3	Print PE status	This function is used to receive the PE check result from DSMC and show to user to complete the enrolled process.
4	Verify check_sum	This function is used to adjudge that the enrolled process is completed successfully and verify the checksum to keep data consistency.
5	Save DriverId_Recognition_Mid	The driver recognition client shall save the MID while the enrollment reresult is success and maintain the association with account in client.
6	Print "Enrollment fail"	This function is used to interact with user about the enrolled result, and receive the user's requirement of retrying registration.
7	Request DriverId_Mode_Request = KeepAwake	While the enrollment process is completed, the driver recognition client shall reset the operational mode as Disable to the driver recognition server.

4.4 Delete MID Process

Delete MID process mainly implements deleting the facial biometric information related to the specific MID from the internal database of Driver Recognition Server.

Requirement for Driver Recognition Server (DRS)
1. The Driver Recognition Server shall provide an interface to receive the request of deleting a specific MID with the biometric information.

2. The DRS shall response the deleting request from the Driver Recognition Client and complete the action in 1 second.
3. The DRS shall accept the deleting request while the state of recognition operational models is in Null.
4. The DRS shall output the deleting result on CAN bus timely (DriverId_Action_Result), including Fail and Success.
5. The DRS shall update the checksum based on the newest database and transmit it by "DriverId_Data_Consistent_Checksum".

Requirement for Driver Recognition Client (DRC)
1. The Driver Recognition Client shall reject the delete MID request from user while (DriverId_Mode_Status < > Null), otherwise while (DriverId_Mode_Status = Null) the DRC shall transfer the (DriverId_Mode_Request = DeleteMid and the value of Delete_DriverId_MID) while user needs to delete a specific driver ID.
2. The vehicle application in driver recognition client shall guide drivers to double confirm the delete request.
3. The DRC shall monitor the status of (DriverId_Action_Result) while (DriverId_Mode_Status = DeleteMid), if the (DriverId_Action_Result = Fail), the DRC shall stop the delete process and feedback the result to user.
4. The DRC shall delete the related data (Delete_DriverId_MID) from its own database while it receives (DriverId_Action_Result == Success).
5. The DRC shall set a 3s timer to wait the enrolled result while it sends (DriverId_Mode_Request = DeleteMid). If it is timeout the DRC shall stop the delete process and feedback the failed result to user.
6. The DRC shall verify database consistency after it receives the result of Delete.

4.4.1 Dataflow Diagram
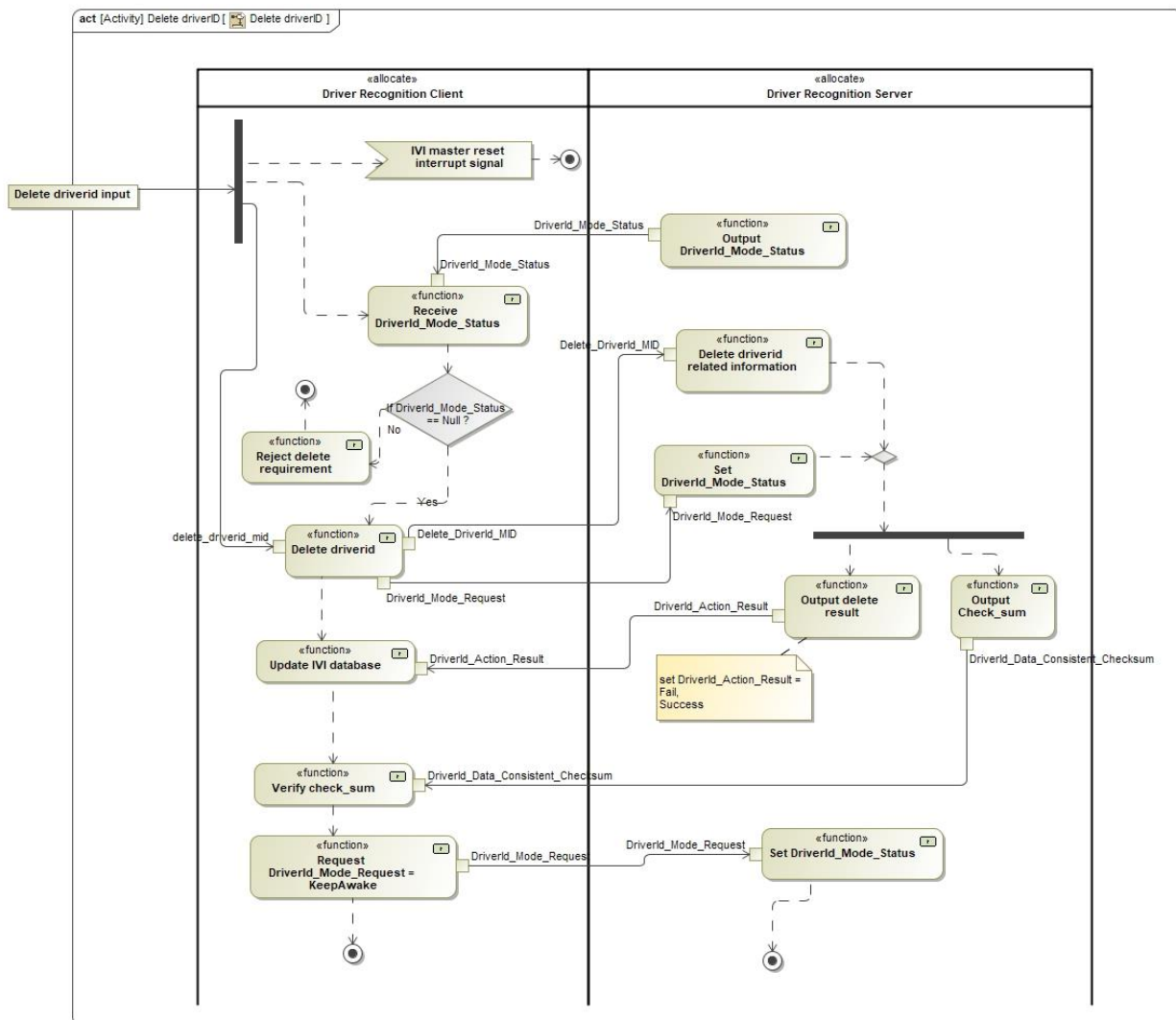
4.4.2 Functions in Driver Recognition Server

4.4.2.1 Functions Description

Fun_id	Fun_Name	Description
1	Set DriverId_Mode_Status	This function is used to set the operational modes of driver recognition system. The state of operational modes is set to DeleteMid.
2	Delete driverid related information	This function is requested by external command to delete the enrolled information relevant to the specific MID received from Driver Recognition Client.
3	Output delete result	This function is used to response the delete MID request and output the result of delete process.
4	Output Check_sum	This function is used to calculate the newest checksum based on the whole database and output to DRC.

4.4.3 Functions in Driver Recognition Client

4.4.3.1 Functions Description

EESE

Fun_id	Fun_Name	Description
1	Receive DriverId_Mode_Status	This function is used to get the current operational mode executed in driver recognition server. If the current mode is not Null, the driver recognition client shall not request the DRS to execute DeleteMid.
2	Update IVI database	This function receives the delete response from user and delete relevant information from the driver recognition client database.
3	Request DriverId_Mode_Request = Null	This function is used to terminate the DELETE MID state of the operational mode in DSMC and reset to IDLE.
4	Verify check_sum	This function is used to verify the checksum to keep data consistency.

4.5 Clear Enrolled Data

Clear enrolled data mainly implements clearing all enrolled models and relevant biometric information from the internal database.

Requirement for Driver Recognition Server (DRS)
1. The DRS shall provide an interface to receive the request of clearing all enrolled models and relevant biometric information.
2. The DRS shall response the clearing request from Driver Recognition Client and complete the action in 3 seconds.
3. The DRS shall output the clearing through (DriverId_Action_Result).
4. The DRS shall response back database checksum where the checksum value reverts to seed value for no enrolled drivers in database.

Requirement for Driver Recognition Client (DRC)
1. The Driver Recognition Client shall execute the ClearAll request while the data consistent check is fail (DriverId_Data_Consistent_Checksum).
2. The DRC shall guide drivers to know the clear operation is executed.
3. The DRC shall monitor the status of (DriverId_Action_Result) while (DriverId_Mode_Status = ClearAll), if the (DriverId_Action_Result = Fail), the DRC shall stop the clear process and feedback the result to user.
4. The DRC shall clear all related data from its own database while it receives (DriverId_Action_Result == Success).
5. The DRC shall set a 5s timer to wait the ClearAll result while it sends (DriverId_Mode_Request = ClearAll). If it is timeout the DRC shall stop the ClearAll process and feedback the failed result to user.
6. The DRC shall verify database checksum received from DRS matches seed value stored in DRC.

4.5.1 Dataflow Diagram
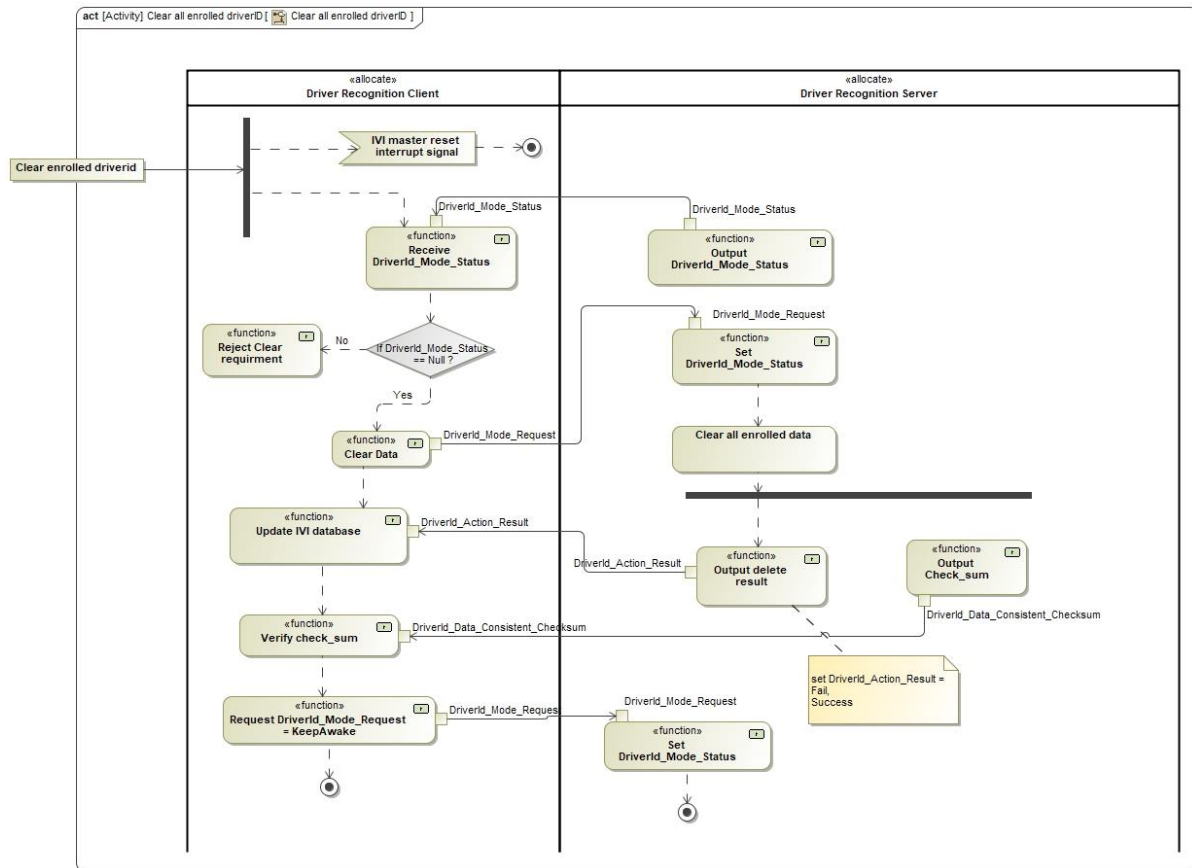
4.5.2 Functions in Driver Recognition Server

4.5.2.1 Functions Description

Fun_id	Fun_Name	Description
1	Set DriverId_Mode_Status	This function is used to set the operational modes of driver recognition system. The state of operational modes is set to DeleteMid.
2	Clear all enrolled data	This function is requested by external command to clear all enrolled data from database in DCS.
3	Output delete result	This function is used to output the result of DeleteMid process.
4	Output Check_sum	This function is used to calculate the newest checksum based on the whole database and output to DRC.

4.5.3 Functions in Driver Recognition Client

4.5.3.1 Functions Description

| Fun_id	Fun_Name	Description
1	Clear data	This function is used to request the driver reocognition server to clear all enrolled information and transfer DriverId_Mode_Request = ClearAll to DRS.
2	Update IVI database	This function receives the clear response (DriverId_Action_Result = Success) from DRS and clear all information from driver recognition database in DRC.
3	Request DriverId_Mode_Request = Null	This function is used to terminate the ClearAll state of the operational mode in DRC and reset to Null.
4	Verify check_sum	This function is used to verify the checksum to keep data consistency.

4.6 Camera Status

Requirement for Driver Recognition Server (DRS)
1. The DRS shall report camera issues, including camera failure or LED failure, connected errors and so on. Related signals are as below:
   - DrvCamera_D_Stat
   - DrvCamDrvIR_D_Stat
   - DrvCamPassIR_D_Stat
2. The DRS shall output the camera status real-time while the Enroll or Recognize processes are executed.
3. If the camera blockage is repaired, this status shall be output through CAN message.

Requirement for Driver Recognition Client (DRC)
1. The DRC shall monitor the camera status and shall report the camera blockage error to user timely while (DrvCamera_D_Stat == Blocked || DrvCamDrvIR_D_Stat == Blocked || DrvCamPassIR_D_Stat == Blocked).
2. The DRC shall prompt the camera temporary unavailable and disable the Driver ID feature while (DrvCamera_D_Stat == TemporaryUnavailable || DrvCamDrvIR_D_Stat == TemporaryUnavailable || DrvCamPassIR_D_Stat == TemporaryUnavailable).
3. If the camera or IR blockage is enabled, DRC shall not initiate an Enroll command to DRS (It is ok to still initiate a Recognize/Delete/Clear all command).

4.6.1 Dataflow Diagram
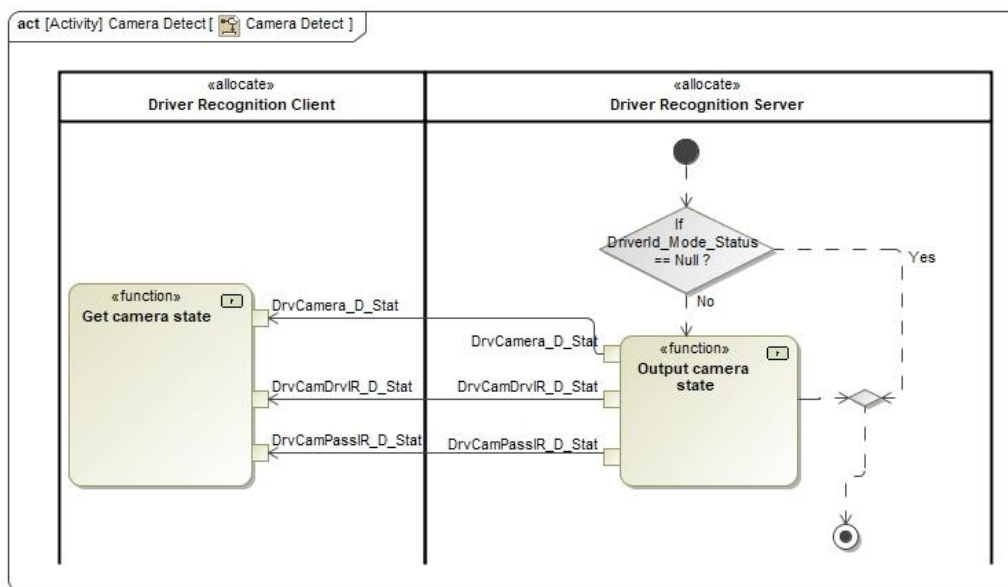
4.7 Driver Recognition Process

The driver recognition process recognizes the person in the driver's seat by comparing their appearance against a database of enrolled drivers. The Driver Recognition Server with camera identifies the driver by capturing image of the person, extracting biometric characteristics and comparing with existing driver models generated in enrollment process in databse which saved in internal memory of the DRS.

Requirement for Driver Recognition Server (DRS)
1. The DRS shall not execute the RECOGNIZE process while receives (DriverId_Mode_Request = Disable) from the driver recognition client.
2. The DRS shall provider an interface for DRC to actively request to perform face recognition function.
3. The driver recognition result signal shall be populated and transmitted (DriverId_Recognition_Result and DriverId_Recognition_Mid) out on CAN bus at completion of Recognition Process.

4. The DRS shall output the recognition result with the checksum signal (DriverId_Data_Consistent_Checksum) used for DRC to check if the data consistent is OK. And the method to calculate checksum shall be shared to DRC.
5. The DRS shall provide the config parameter Recognition_Timeout (default value is **10** seconds) for the entire face recognition process timeout if no cooperation is facilitated.
6. The DRS shall provide "Recognition_Max_Observation_Time" as an EOL configuration parameter which can be adjustable as 0-30s in 2s increments with default (and recommended value) of **2s** to meet 90% KPI.
7. The DRS shall output the status of pose and emotion detection on CAN bus (PoseExpression_Constrains_Check) continuously.
8. The Driver ID recognition result shall be populated in 3 seconds after the DSMC is fully operational with driver looking forward.
9. The driver recognition process needs to have the ability to prevent attacks and if the spoofing action is detected, the DRS shall output PoseExpression_Constrains_Check = NotAFace.
10. While DRS receives the recognition request, it shall detect the camera and output the status through (DrvCamera_D_Stat, DrvCamDrvIR_D_Stat and DrvCamPassIR_D_Stat).
11. The DRS shall output the checksum of database with the recognition result to DRC.

Requirement for Driver Recognition Client (DRC)
1. The DRC shall set a 20 seconds timer to wait for the recognition result from DRS while DRC sends (DriverId_Mode_Request=Recognize) to DRS. If the timer is timeout, the DRC shall stop the recognition process and feedback to user.
2. The DRC shall check the data consistent (DriverId_Data_Consistent_Checksum) before deal with the recognition result. If the data check is fail, the DRC shall request clear all the enrolled data (Please refer to the chapter of **Error Handling**).
3. The DRC shall verify the received checksum signal (DriverId_Data_Consistent_Checksum) to ensure database consistency.

4.7.1 Dataflow Diagram
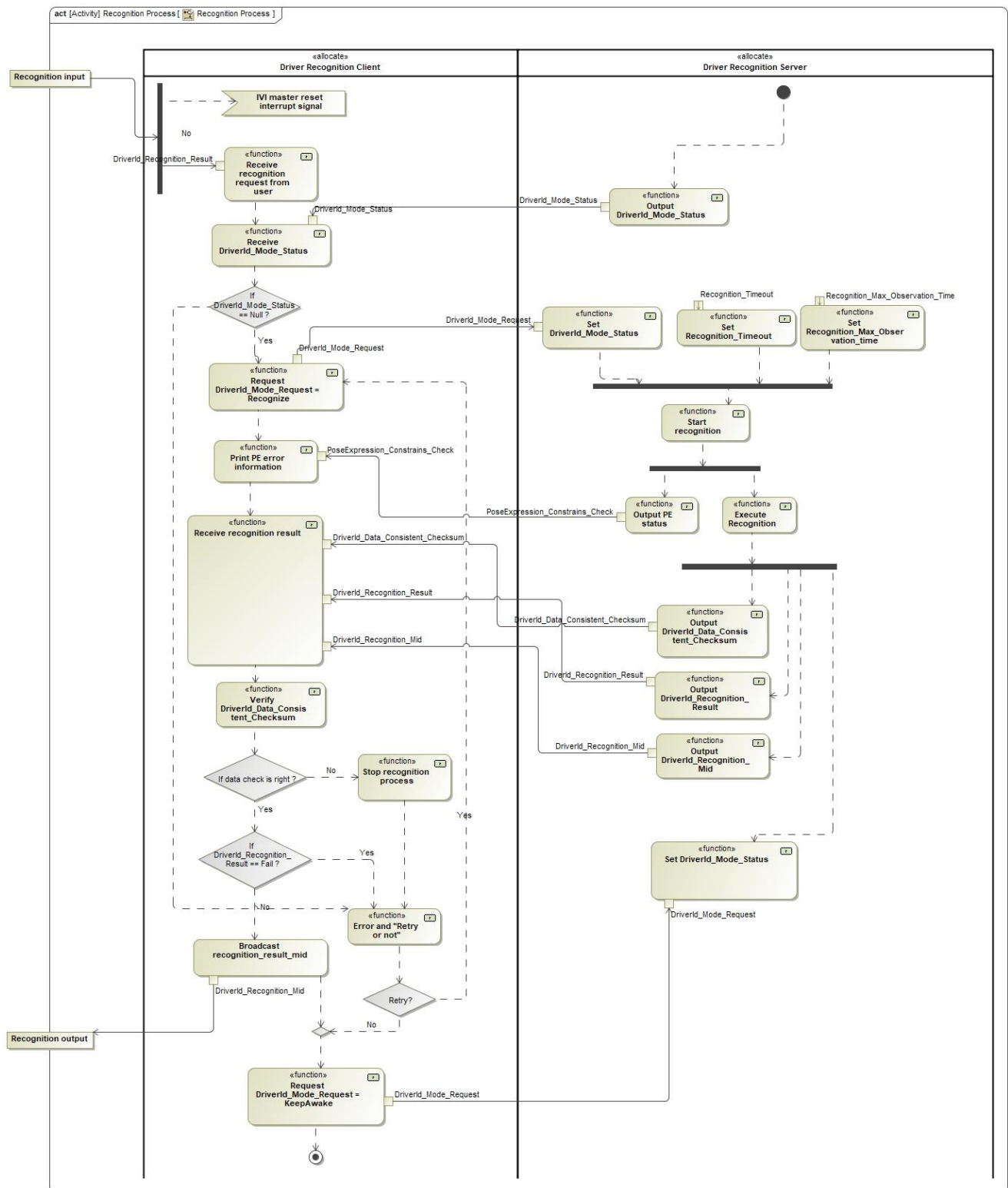
4.7.2 Functions in Driver Recognition Server

4.7.2.1 Functions Description

Fun_id	Fun_Name	Description

1	Set DriverId_Mode_Status	This function is used to set the operational modes of driver recognition system. The state of operational modes is set to Recognize.
2	Set Recognition_Timeout	The vehicle application (outside of DSMC) could use this function to set the duration allowed for the recognition process to be run. If the recognition_timeout occurs, the current recognition process must be interrupted immediately and restart a new recognition.
3	Set Recognition_Max_Observation_time	The driver recognition server shall get the config parameter Recognition_Max_Observation_time which is the time limit for observation while the recognition is requested.
	Output DriverId_Data_Consistent_Checksum	The driver recognition server shall caculate the checksum to verify data consistency.
16	Output DriverId_Recognition_Mid	This function is used to output the recognized result on CAN bus.
17	Output DriverId_Recognition_Result	This function is used to output the result of the recognition process, such as Null, OKMatched, OKNoMatch, and FailTimeout.
	Left blank to Add	

4.7.3 **Functions in Driver Recognition Client**

4.7.3.1 **Functions Description**

Fun_id	Fun_Name	Description
1	Receive DriverId_Mode_Status	This function is used to get the current operational mode executed in driver recognition server. If the current mode is not Null, the driver recognition client shall not request the DRS to execute enrollment.
2	Request DriverId_Mode_Request = Recognize	This function is used to request driver recognition server to do recognition.
3	Print PE error information status	This function is used to receive the PE check result from DSMC and show to user to complete the recognition process.
4	Verify DriverId_Data_Consistent_Checksum	This function is used to adjudge that the recognition process is completed successfully and verify the checksum to keep data consistency.
7	Request DriverId_Mode_Request = KeepAwake	While the recognition process is completed, the driver recognition client shall reset the operational mode as Disable to the driver recognition server.
	Left blank to Add	

5 Output Performance Requirements

5.1 Field of View

Requirements:
1. The Driver Recognition Server shall follows all performance requirement in **RQT-190402-705877.**

5.2 Driver Recognition Result

The driver recognition result is a signal which indicates driver identification. This signal is based on driver face information.

Requirements:
1. Driver recognition shall be provided accuracy performance listed in Table 4.7.3-1.

Table 4.7.3-1: Driver Recognition Status Accuracy

	False Positive (FPR)	True Positive (TPR)
Driver Recognition Result	<1.5%	>95%

Test-conditions for the requirement:
1) In typical vehicle environment, the KPI requirement shall be met.
2) The situation that the driver is on seat with looking at the Infortainment or looking on road shall meet the KPI requirement.
3) The recognition result shall be outputted as soon as possible and not more than 10 seconds from recognition request.

5.3 Camera and IR Obstruction Performance

Requirements:
1. The Driver Recognition Server shall detect the camera blockage and output the status to the Driver Recognition Client.
2. The Driver Recognition Client shall popup warning to user to check camera obstruction.

6 Error Handling

6.1 Camera Error Detection

It is possible that the driver monitor camera or illumination system can be obstructed on accident (e.g. something gets placed in front of it) or on purpose (e.g. the driver covers the camera with one hand) which will obscure the camera's view of the driver or reduce the driver illumination.

Requirement (R-X)

1. The DSMC shall report camera or illimunator blockage conditions when it detects an obstruction which significantly degrades driver tracking performance.
2. The DSMC shall report camera or illimunator fault messages while the below errors occur:
    1) Camera or illimunator fakra connector failure.
    2) Camera or illimunator internal EE failure.
    3) Camera temperature abnormally.

ID	Logical Name	Physical Name
1	DriverId_Mode_Status	DrvIdMde_D_Stat
2	PoseExpression_Constrains_Check	DrvExpression_D_Stat
3	DriverId_Recognition_Result	DrvId_D_Stat
4	DriverId_Recognition_Mid	DrvIdMatch_No_Actl
5	DriverId_Data_Consistent_Checksum	DrvId_No_Cs
6	DriverId_Action_Result	DrvIdActn_D_Stat
	Left blank to Add	

Memory swaps will only be initiated when specific criteria are met

6.2 Abnormal Data Loss

6.2.1 Driver Recognition Client Data Loss

Due to replacement or in response to master reset, etc, the driver recognition client may loss all enrolled data. In these situations, the driver recognition client and server shall take actions to guarantee the data consistency.

Requirement (R-X)

1. The driver recognition client shall check data consistency with the server by checking the Checksum value, if it is incorrect, the below actions shall be executed:
    1) The client shall request the server to clear all data relevant to the driver recognition functionalities.
    2) The client shall clear all enrolled data relevant to the driver recognition functionalities.

6.2.2 Driver Recognition Server Data Loss

Due to replacement or in response to master reset, etc, the driver recognition Server may loss all enrolled data. In these situations, the driver recognition client and server shall take actions to guarantee the data consistency.

Requirement (R-X)

1. DRS shall support a Clear command to clear all data.
2. DRS shall calculate a checksum based on database and send it to DRC within Enroll/Delete/Clear/Recogniton process to ensure alignment in database information between DRC and DRS.
3. The DRC shall verify database by checksum.

6.3 Hardware and Software Faults of DRS

Requirements:

1. The driver recognition client needs to popup the errors and disable the driver recognition functionalities.
2. The driver recognition client shall record the DRS connected error by DTC.

EESE

3. The driver recognition client shall record the DRS function failure by DTC (When the DRS contines to send signal DrvIdMde_D_Stat = 0x7 (Temp_unavailable) for more than 10 seconds).

EESE

7 REVISION HISTORY

Ver.	Date	Description	Responsible
1.0	February 28, 2020	Initial version	Jia,Elaine
2.0	March 5, 2020	Initial version	Jia Elaine Yu,Diven
3	April 21, 2020	1. Update enrolled process and signal requirements;	Jia Elaine Lee,Darren
4	Jun 20, 2020	1. Update the performance requirement	Jia Elaine Li, Cindy
5	July 9, 2020	1. Update all the diagram; 2. Update the requirems of enrollment and reocognition process; 3. Update the Open question list comments discussed with VNE	Jia Elaine Li, Cindy
6	July 23, 2020	1. Remove the operation mode of DSMC; 2. Update the start time of DSMC to 8s;	
7	August 11, 2020	1. Update the signal list; 2. Add timeout for IVI to receive the DSMC feedback; 3. Update the DSMC hardware (Camera and IR light) errors definition; 4. Update the Driver Reognition Process requirements;	Jia Elaine
8	Sep 17, 2020	1. Update contents depend on VNE's comments. 2. Update the KPI requirement.	Jia Elaine