|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  |  | | |  |
|  | Mode Card | | |  |
|  | ( SYNC+\_CF067) | | |  |
|  |  | | |  |
|  |  | | |  |
| Document Type | **Feature Implementation Specification (FIS)** | | |  |
| Template Version | **6.1a** | | |  |
| Document ID | **/** | | |  |
| Document Location | **/** | | |  |
| Document Owner | **Geng Dekang (dgeng1)** | | |  |
| Document Version | **1.0** | | |  |
| Document Status | **Draft** | | |  |
| Date Issued | **2023/02/01** | | |  |
| Date Revised |  | | |  |
| Document Classification | GIS1 Item Number: | **27.60/35** | |  |
| GIS2 Classification: | **Confidential** | |
|  | | | | |
|  | | | | |
| Document Approval | | | | |
| Person | Role | | Email Confirmation | Date |
|  |  | |  |  |
|  |  | |  |  |

This document contains Ford Motor Company Confidential information. Disclosure of the information contained in any portion of this document is not permitted without the expressed, written consent of a duly authorized representative of Ford Motor Company, Dearborn, Michigan, U.S.A.

Copyright © 2021, Ford Motor Company

Printed Copies Are Uncontrolled

**Disclaimer**

**This document contains Ford Motor Company Confidential information. Disclosure of the information contained in any portion of this document is not permitted without the expressed, written consent of a duly authorized representative of Ford Motor Company, Dearborn, Michigan, U.S.A.**

This document contains information developed and accumulated by and for FORD MOTOR COMPANY. As such, it is a proprietary document, which, if disseminated to unauthorized persons, would provide others with restricted information, data, or procedures not otherwise available, exposing the FORD MOTOR COMPANY to potential harm.

Employees and suppliers having custody of this specification or authorized to use it must be cognizant of its proprietary nature and ensure that the information herein is not made available to unauthorized persons.

FORD MOTOR COMPANY reserves the right to protect this work as an unpublished copyrighted work in the event of an inadvertent or deliberate unauthorized publication. FORD MOTOR COMPANY also reserves its rights under copyright laws to protect this work as a published work.

This document or portions thereof shall not be distributed outside FORD MOTOR COMPANY without prior written consent. Refer all questions concerning disclosure to the author(s) or any duly authorized representative of Ford Motor Company.

**Copyright** © **2021 Ford Motor Company**

# Contents

[Contents 3](#_Toc35000551)

[1 Introduction 6](#_Toc35000552)

[1.1 Document Purpose 6](#_Toc35000553)

[1.2 Document Scope 6](#_Toc35000554)

[1.3 Document Audience 6](#_Toc35000555)

[1.3.1 Stakeholder List 6](#_Toc35000556)

[1.4 Document Organization 6](#_Toc35000557)

[1.4.1 Document Context 6](#_Toc35000558)

[1.4.2 Document Structure 6](#_Toc35000559)

[1.5 Document Conventions 7](#_Toc35000560)

[1.5.1 Requirements Templates 7](#_Toc35000561)

[1.5.1.1 Identification of requirements 7](#_Toc35000562)

[1.5.1.2 Requirements Attributes 7](#_Toc35000563)

[1.6 References 7](#_Toc35000564)

[1.6.1 Ford Documents 8](#_Toc35000565)

[1.6.2 External Documents and Publications 8](#_Toc35000566)

[1.7 Glossary 8](#_Toc35000567)

[1.7.1 Definitions 8](#_Toc35000568)

[1.7.2 Abbreviations 8](#_Toc35000569)

[2 Feature Implementation Overview 9](#_Toc35000570)

[2.1 Description 9](#_Toc35000571)

[2.2 Input Requirements/Documents 9](#_Toc35000572)

[2.3 Lessons Learned 9](#_Toc35000573)

[2.4 Assumptions 9](#_Toc35000574)

[3 Feature Implementation Architecture 11](#_Toc35000575)

[3.1 Functional Architecture 11](#_Toc35000576)

[3.1.1 Description 11](#_Toc35000577)

[3.1.2 Function List 11](#_Toc35000578)

[3.1.3 Signal List 12](#_Toc35000579)

[3.2 Physical Architecture 12](#_Toc35000580)

[3.2.1 E/E Architecture 12](#_Toc35000581)

[3.2.1.1 E/E Architecture Variants 12](#_Toc35000582)

[3.2.1.1.1 E/E Architecture “Architecture Variant 1” 12](#_Toc35000583)

[3.2.1.1.2 E/E Architecture “Architecture Variant 2” 14](#_Toc35000584)

[3.2.1.2 E/E Components 14](#_Toc35000585)

[3.2.1.3 E/E Connections 15](#_Toc35000586)

[3.2.1.4 Signal List 16](#_Toc35000587)

[3.2.2 Software Component Architecture 16](#_Toc35000588)

[3.2.2.1 Description 16](#_Toc35000589)

[3.3 Function Deployment 16](#_Toc35000590)

[3.3.1 Deployment Variants 17](#_Toc35000591)

[3.3.1.1 Deployment “Variant 1” 17](#_Toc35000592)

[3.3.2 Function Allocation 17](#_Toc35000593)

[4 Feature Implementation Modeling 20](#_Toc35000594)

[4.1 Component Interaction Diagrams 20](#_Toc35000595)

[4.1.1 Scenario: “System Startup / Shutdown” 20](#_Toc35000596)

[4.1.2 Scenario: “Normal Operation” 20](#_Toc35000597)

[4.2 Component Interface Behavior Diagrams 21](#_Toc35000598)

[5 Feature Implementation Requirements 22](#_Toc35000599)

[5.1 Functional Safety 22](#_Toc35000600)

[5.1.1 ASIL Decomposition of Technical Safety Requirements 22](#_Toc35000601)

[5.2 Requirements on Components 23](#_Toc35000602)

[5.2.1 Component 1 23](#_Toc35000603)

[5.2.1.1 Technology Function “MyLogicalFunctionA\_Component1” 23](#_Toc35000604)

[5.2.1.1.1 Function Interfaces 23](#_Toc35000605)

[5.2.1.1.1.1 Inputs 24](#_Toc35000606)

[5.2.1.1.1.2 Outputs 24](#_Toc35000607)

[5.2.1.1.1.3 Parameters 25](#_Toc35000608)

[5.2.1.1.1.4 Interface Requirements 25](#_Toc35000609)

[5.2.1.1.2 Function Requirements 26](#_Toc35000610)

[5.2.1.1.2.1 Component Specific Requirements 26](#_Toc35000611)

[5.2.1.2 (Technology) Function “MyLogicalFunctionB\_Component1” 26](#_Toc35000612)

[5.2.2 Component 2 26](#_Toc35000613)

[5.3 Requirements on Connections 26](#_Toc35000614)

[5.3.1 Networks 26](#_Toc35000615)

[5.3.1.1 “CAN Bus xxx” 26](#_Toc35000616)

[5.3.1.1.1 Protocol Requirements 27](#_Toc35000617)

[5.3.1.1.2 Electrical Requirements 27](#_Toc35000618)

[5.3.1.2 “LIN Bus xxx” 27](#_Toc35000619)

[5.3.1.2.1 Protocol Requirements 27](#_Toc35000620)

[5.3.1.2.1.1 Schedule Table 27](#_Toc35000621)

[5.3.1.2.2 Electrical Requirements 27](#_Toc35000622)

[5.3.1.3 “Ethernet xxx” 27](#_Toc35000623)

[5.3.2 HW I/Os 27](#_Toc35000624)

[5.3.2.1 “HW I/O xxx” 27](#_Toc35000625)

[5.4 Requirements on Development Process 27](#_Toc35000626)

[6 Open Concerns 28](#_Toc35000627)

[7 Revision History 29](#_Toc35000628)

[8 Appendix 30](#_Toc35000630)

[8.1 Data Dictionary 30](#_Toc35000631)

[8.1.1 Logical Signals 30](#_Toc35000632)

[8.1.2 Logical Parameters 30](#_Toc35000633)

[8.1.3 Technical Signals 30](#_Toc35000634)

[8.1.3.1 GSDB Signals 30](#_Toc35000635)

[8.1.3.2 HW I/Os 30](#_Toc35000636)

[8.1.3.3 Diagnostic Interfaces 30](#_Toc35000637)

[8.1.3.3.1 DTCs 30](#_Toc35000638)

[8.1.3.3.2 DIDs 31](#_Toc35000639)

[8.1.4 Technical Parameters 31](#_Toc35000640)

[8.1.5 Mappings 31](#_Toc35000641)

[8.1.6 Technical Interfaces 31](#_Toc35000642)

[8.1.6.1 AIS Interfaces 31](#_Toc35000643)

[8.1.6.1.1 Publisher Interfaces 31](#_Toc35000644)

[8.1.6.1.2 Subscriber Interfaces 31](#_Toc35000645)

[8.1.6.2 AUTOSAR Ports 31](#_Toc35000646)

[8.1.7 Messages/APIs 31](#_Toc35000647)

[8.1.7.1 CAN Bus “<Bus Name>” 31](#_Toc35000648)

[8.1.7.2 LIN Bus “<Bus Name>” 32](#_Toc35000649)

[8.1.7.3 AUTOSAR Interfaces 32](#_Toc35000650)

[8.1.7.4 SOA Service Contracts 32](#_Toc35000651)

[8.1.8 Encoding Types 32](#_Toc35000652)

**List of Figures**

[Figure 3‑1: Functional Architecture 11](#_Toc35000653)

[Figure 3‑2 E/E Architecture (SysML Style) 13](#_Toc35000654)

[Figure 3‑3 E/E Architecture (Network Topology Style) 14](#_Toc35000655)

[Figure 3‑4: AUTOSAR compliant SW Component Architecture 16](#_Toc35000656)

**List of Tables**

[Table 1‑1: Electrical Architecture(s) referenced in this document 6](#_Toc35000657)

[Table 1‑2: Ford internal Documents 8](#_Toc35000658)

[Table 1‑3: External documents and publications 8](#_Toc35000659)

[Table 1‑4: Definitions used in this document 8](#_Toc35000660)

[Table 1‑5: Abbreviations used in this document. 8](#_Toc35000661)

[Table 6: Input Requirements/Documents 9](#_Toc35000662)

[Table 3‑1: List of Functions 12](#_Toc35000663)

[Table 3‑2: Electrical Components 15](#_Toc35000664)

[Table 3‑3: E/E Connections 15](#_Toc35000665)

[Table 3‑4: Sample Deployment Diagram 17](#_Toc35000666)

[Table 3‑5: Function Allocation Table (Basic) 18](#_Toc35000667)

[Table 3‑6: Function Allocation Table (Functional Safety Extension) 19](#_Toc35000668)

[Table 5‑1: ASIL Decomposition Table 23](#_Toc35000669)

[Table 5‑2: Input Signal mappings of Function “MyLogicalFunctionA\_Component1” 24](#_Toc35000670)

[Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A 25](#_Toc35000671)

[Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1” 25](#_Toc35000672)

[Table 5‑5: Component Specific Requirements 26](#_Toc35000673)

[Table 5‑6: Inherited Requirements 26](#_Toc35000674)

[Table 6‑1: Open Concerns 28](#_Toc35000675)

# Introduction

## Document Purpose

The Feature Implementation Specification (FIS) specifies the deployment of the logical functions of a feature to an electrical architecture. The FIS specifies all interactions between the ECUs of the electrical architecture required for the feature including the technical signals and the interfaces. It also gives interface and integration requirements, which are specific to the feature for the electrical architecture.

To get more information about the concept of feature, function and component level abstraction refer to the [Ford RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Engineering+for+SW+Enabled+Features).

## Document Scope

This FIS describes the deployment of the feature <Mode Card> to the following electrical architecture(s):

| **Electrical Architecture Name** | **Owner** | **Reference** |
| --- | --- | --- |
| FNV2 | Dgeng1 | / |
| FNV2.1 | Dgeng1 | / |

Table 1‑1: Electrical Architecture(s) referenced in this document

## Document Audience

The FIS is authored by Emma Chen / Feature Owner Supervisor>. All Stakeholders, i.e., all people who have a valid interest in the feature implementation should read and, if possible, review the FIS. It needs to be guaranteed, that all stakeholders have access to the currently valid version of the FIS.

### Stakeholder List

For the latest list of the function stakeholders and their roles & responsibilities refer to <TBD>.

## Document Organization

### Document Context

Refer to the [Specification Structure page](http://wiki.ford.com/display/RequirementsEngineering/Specification+templates) in the [Ford RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Engineering+for+SW+Enabled+Features) to understand how the FIS relates to other Ford Requirements Documents and Specifications.

### Document Structure

The structure of this document is explained below:

**Section 1** – Introduction – Giving an explanation how to use this document including responsibilities and the scope of the document. Additionally it contains the revision history and a list of unsettled but known issues that have to be consolidated in future versions. It explains the terminology and gives a clarification of the definitions, concepts and abbreviations used in the document.

**Section 2** – Feature Implementation Description – Giving an overview of the platform and listing assumptions, constraints or dependencies

**Section 3** – Feature Implementation Architecture – Describing 3 Architecture Views:

* Functional Architecture – Showing the logical architecture of functions
* Physical Architecture – Showing the physical architecture (first of all the E/E Architecture), which the Logical Functions get allocated to.
* Software Architecture – Showing the software architecture relevant for the feature (for features with in-house development only)
* Function Deployment – Presenting the allocation of logical functions and signals to the electrical and other components

**Section 4** – Deployment Specific Modeling –Modeling techniques providing additional detail on e.g. interface behavior

**Section 5** – Deployment Specific Requirements – Deployment specific requirements for ECUs, Network Communication, and Process

**Section 6** – List of Open Concerns

**Section 7** – Revision History

**Section 8** – Appendix - Presenting additional data mainly in a tabular form, e.g., a data dictionary

## Document Conventions

### Requirements Templates

Refer to “[How to use the Specification Templates](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates?src=contextnavpagetreemode)” on how to use the specification templates and the VBA macros to create/edit the requirements in the specifications.

The VBA macro enable the import of the specification to VSEM (refer to ["How to import specifications into VSEM as separate requirements"](http://wiki.ford.com/pages/viewpage.action?pageId=104991616&src=contextnavpagetreemode)).

#### Identification of requirements

The unique requirement ID given in the headline of any requirement follows the requirement throughout the development process. The requirement ID format follows a well-defined syntax.

All identifiers in an FIS shall be composed of 4 parts:

* A leading prefix, which indicates the type of requirement (R=Requirement, UC=Use Case, SC=Scenario, …)
* A prefix, which indicates the abstraction level (F=Feature, FNC=Function, CMP = component).
* Followed by a name, indicating the scope, which the requirement belongs to (e.g. feature or function name )
* Ending with the actual requirement number

*Example:*

*R\_CMP\_LockArbitrator\_00004* This is the fourth requirement on component level for the function Lock Arbitrator.

#### Requirements Attributes

Additionally attributes can be added to each requirement. This helps to classify requirements. A [list of available attributes](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes?src=contextnavpagetreemode) is given in the RE Wiki.

## References

### Ford Documents

The list of all Ford internal documents, which are directly related.

| **Reference** | **Title** | **Doc. ID** | **Revision** | **Document Location** |
| --- | --- | --- | --- | --- |
| 1 | Mode Card Feature Boundary Diagram | 1 | V1.0 | GitHub |
| 2 | Mode Card Feature Level Specification | 2 | V1.0 | GitHub |
| 3 | Mode Card Feature implementation specification | 3 | V1.0 | GitHub |
| 4 | Mode Card Full\_Range\_interface\_List | 4 | V1.5 | GitHub |

Table 1‑2: Ford internal Documents

### External Documents and Publications

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

| **Reference** | **Document / Publication** |
| --- | --- |
| / | / |
|  |  |

Table 1‑3: External documents and publications

## Glossary

### Definitions

| **Definition** | **Description** |
| --- | --- |
|  |  |
|  |  |
|  |  |

Table 1‑4: Definitions used in this document

### Abbreviations

|  |  |
| --- | --- |
| ABS | Automatic Braking System |
| AHU | Audio Head Unit |
| AHUD | Advanced Heads Up Display |
| APIM | Auxiliary Protocol Interface Module (SYNC Module) |
| BCM | Body Control Module |
| CAN | Controller Area Network |
| DTE | Distance to empty |
| D&R | Design and Release Engineer |
| DDM | Driver Door Module |
| DSM | Door Seat Module |
| DSP AMP | Digital Signal Processing Amplifier |
| DTC | Diagnostic Trouble Code |
| DCU | Door Control Unit |
| ECU | Electronic Control Unit |
| EM | Enhanced Memory |
| FCIM | Front Control Interface Module |
| FNV2 | Fully Networked Vehicle II |
| FS | Feature Specification Document |
| GSDB | Global Signal Database |
| HCM | Headlamp Control Module |
| HMI | Human Machine Interface |
| AEIS | Automatic Engine Idle Shutdown |
| IPC | Instrument Panel Cluster |
| IPMA | Image Processing Module A |
| IVI | In Vehicle Infotainment |
| MY23 | Model Year 2023 |
| PEPS | Passive Entry Passive Start |
| HVAC | Heating Ventilating and Air Conditioning |
| RCCM | Remote Climate Control Module |
| VR | Voice Recognition |

Table 1‑5: Abbreviations used in this document.

# Feature Implementation Overview

## Description

The Mode Card feature is designed for FNV2.x architecture and SYNC+ platform, it could provide customers with a combination of functions to achieve diversified and personalized scene services, and one-button control or/and automatic operation to provide customers with a more convenient usage.

## Input Requirements/Documents

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference**  (Reference as listed in ch. “References”) | **Section/Requirement** | **Description** | **Derived Requirement**  (optional – reference to requirement in ch. “Feature Implementation Requirements”) |
| **Feature/Function Requirements** | | | |
| 1 | Mode Card 1.0 PRD | Product Requirement Description | / |
| 2 | FBMP APIM SPSS v1.3 May 8, 2017 | Feature Based Message Protocol  APIM Infotainment Subsystem Part Specific | / |
|  |  |  |  |
| **Ford Engineering Standards** | | | |
|  | <Example: some SDS (requirement)> |  |  |
|  |  |  |  |
| **Legal Regulations** | | | |
| 1 | GB/T 4094.2-2017 | Electric vehicles -- Symbols for controls, indicators, and tell-tales | To identify the indicator display when running camp mode |
|  |  |  |  |
| **Industry Standards** | | | |
|  |  |  |  |
| **Other Sources** | | | |
|  |  |  |  |

Table 6: Input Requirements/Documents

## Lessons Learned

Mode Card is a new feature in Ford, so no Ford lesson learned exist, benchmark is the reference for L&L.

## Assumptions

Mode Card is assumed to enhance and extend the original smart scene feature, which is to provide customers with a combination of functions to achieve diversified and personalized scene services, and one-button control or/and automatic operation to provide customers with a more convenient usage. However, the interface capability depends on the vehicle EE architecture.

# Feature Implementation Architecture

## Functional Architecture

### Description

**#Hint**: Provide some informal description of the characteristics of the chosen Functional Architecture. Also give some graphical representation of the Functional Architecture. Either SysML activity diagrams or [Data Flow Diagrams](http://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemode) could be used to depict such a Functional Architecture.

**#Link:** *[SysML - Activity Diagrams](https://pd3.spt.ford.com/sites/SystemsEngineering/SEC/sysml-teamsite/SysML%20Wiki/Activity%20Diagram%20Basics.aspx)* or [*RE Wiki - Data Flow Diagrams*](http://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemodehttp://wiki.ford.com/display/RequirementsEngineering/Data+Flow+Diagram?src=contextnavpagetreemode)



Figure 3‑1: Functional Architecture

### Function List

The following functions from the [Global Feature & Function List](https://www.vsemweb.ford.com:443/tc/launchapp?-attach=true&-s=226TCSession&-o=ZmZNi0JHx3NrTDAAAAAAAAAAAAA) are referenced in this Feature Implementation Specification:

|  |  |  |
| --- | --- | --- |
| **Function ID** | **Function Name** | **Function Description** |
| 1 | Recommend Scene | These scenes can provide fixed and convenient scene for customers to select, such as camp mode etc. |
| 2 | Custom Scene | Customers can combine different conditions and application to create a new scene by themselves. |
| 3 | Platform API capability | It can provide the APIs based on vehicle EE architecture, such as ignition status、door status、clock etc. |
|  |  |  |
|  |  |  |

Table 3‑1: List of Functions

### Signal List

Refer to < Mode Card Full\_Range\_interface\_List >.

## Physical Architecture

### E/E Architecture

#### E/E Architecture Variants

|  |  |  |
| --- | --- | --- |
| E/E Architecture Variant Name | Variant Description | Variant Condition (optional) |
| FNV2 | Based on FNV2 platform | Network Topology = FNV2 |
| FNV2.1 | Based on FNV2.1 platform | Network Topology = FNV2.1 |
|  |  |  |

##### E/E Architecture

For detailed program needs to follow specific architecture, especially ambient light, Massage, EV functions and other new change for each program.

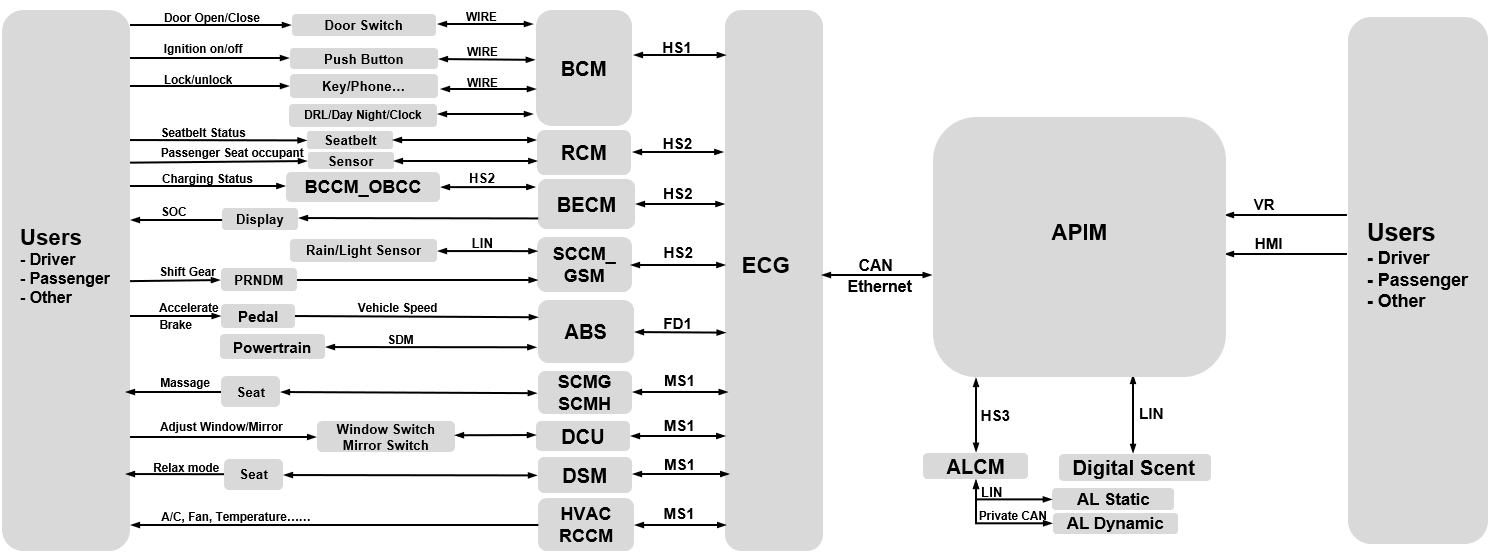


Figure 3‑2 E/E Architecture (FNV2.x)

##### E/E Architecture “Architecture Variant 2”

#### E/E Components

|  |  |
| --- | --- |
| Component Name | **Description** |
| APIM | Auxiliary Protocol Interface Module |
| ECG | Enhanced Center Gateway |
| Ford APP | Ford APP like EV/FordPass/LinclonWay on Smart Phone |
| ABS | Automatic Braking System |
| HPCM | Hybrid Powertrain Control Module |
| BECM | In Vehicle Infotainment |
| GSM | Audio Head Unit |
| HUD | Heads Up Display |
| SCCM | Steering Column Control Module |
| BCM | Body Control Module |
| RCM | Restraint Control Module |
| DCU | Door Control Unit |
| DSM | Driver Seat Module |
| HVAC | Heating Ventilating and Air Conditioning |
| RCCM | Remote Climate Control Module |
| SCMG | Seat Control Module G |
| SCMH | Seat Control Module H |
| ALCM | Ambient Light Control Module |
| Digital Scent | Digital Scent |
|  |  |

Table 3‑2: Electrical Components

#### E/E Connections

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Connection Name | **Connection Type** | **Protocol**  Only if ‘Connection Type’ is “Network”/”RF-Digital” | **Description** | **Allocated Messages**  Only if ‘Connection Type’ is “Network”/”RF-Digital” | **Connected Nodes** |
| HS-CAN1 | Network | CAN (High Speed) | High Speed CAN bus | … | BCM… |
| HS-CAN2 | Network | CAN (High Speed) | High Speed CAN bus | … | RCM… |
| HS-CAN3 | Network | CAN (High Speed) | Infotainment High Speed CAN bus | … | IPC/HUD… |
| HS-CAN4 | Network | CAN (High Speed) | High Speed CAN bus | … | TCU… |
| MS-CAN | Network | CAN (Mid Speed) | \High Speed CAN bus | … | HVAC/RCCM/… |
| CANFD | Network | CAN FD | CAN with Flexible Data-rate | … | ABS… |
| Ethernet | Network | Ethernet (MQTT) | Infotainment High Speed CAN bus | … | ECG… |

Table 3‑3: E/E Connections

#### Signal List

Refer to < Mode Card Full\_Range\_interface\_List >.

### Software Component Architecture

#### Description

This is the Software Component Architecture for Ford SYNC+ Phase4 FNV2 programs and Phase6 FNV2.1 programs, other specific programs can also refer to it.

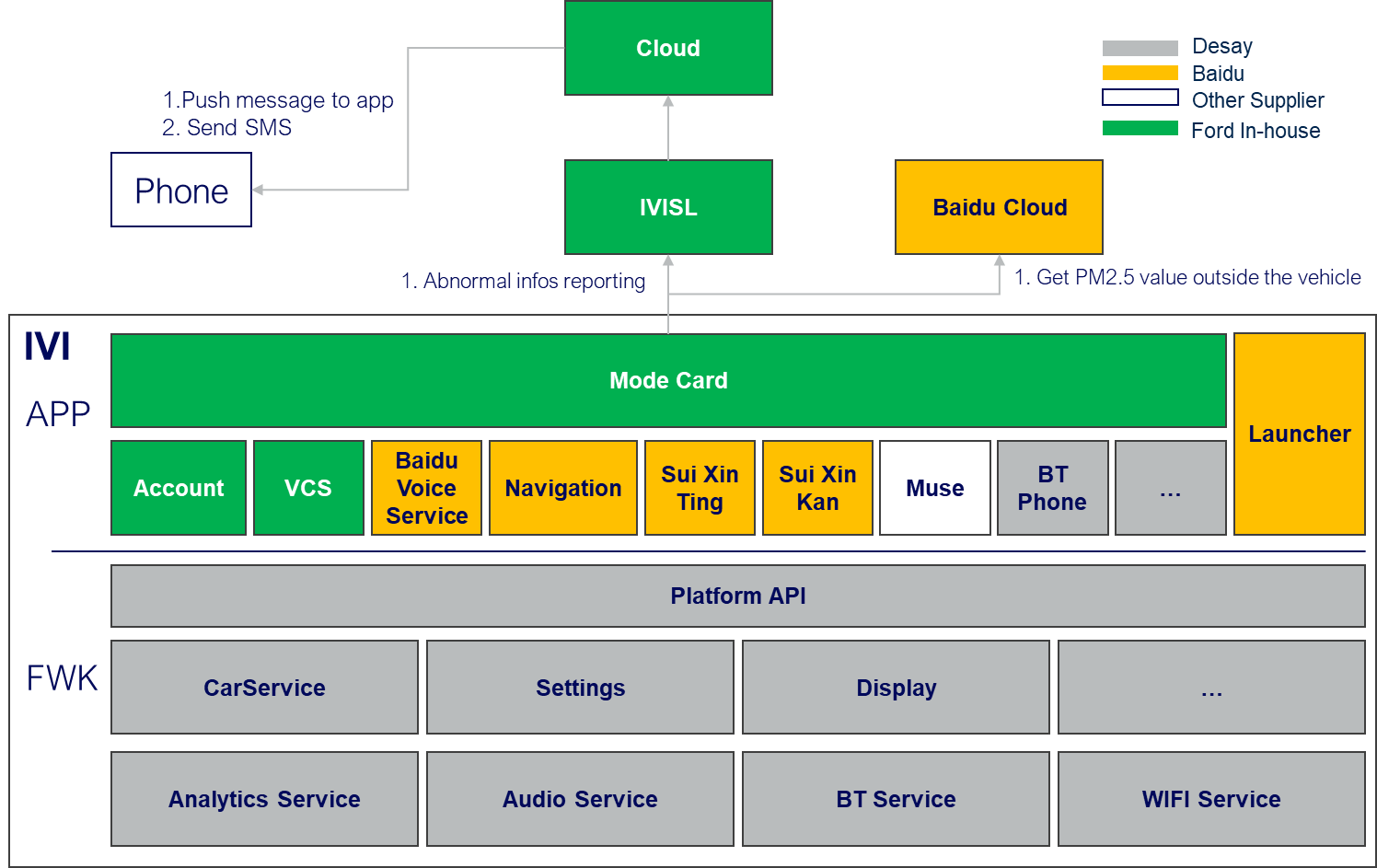


Figure 3‑3 SW Architecture (Based on FNV2)

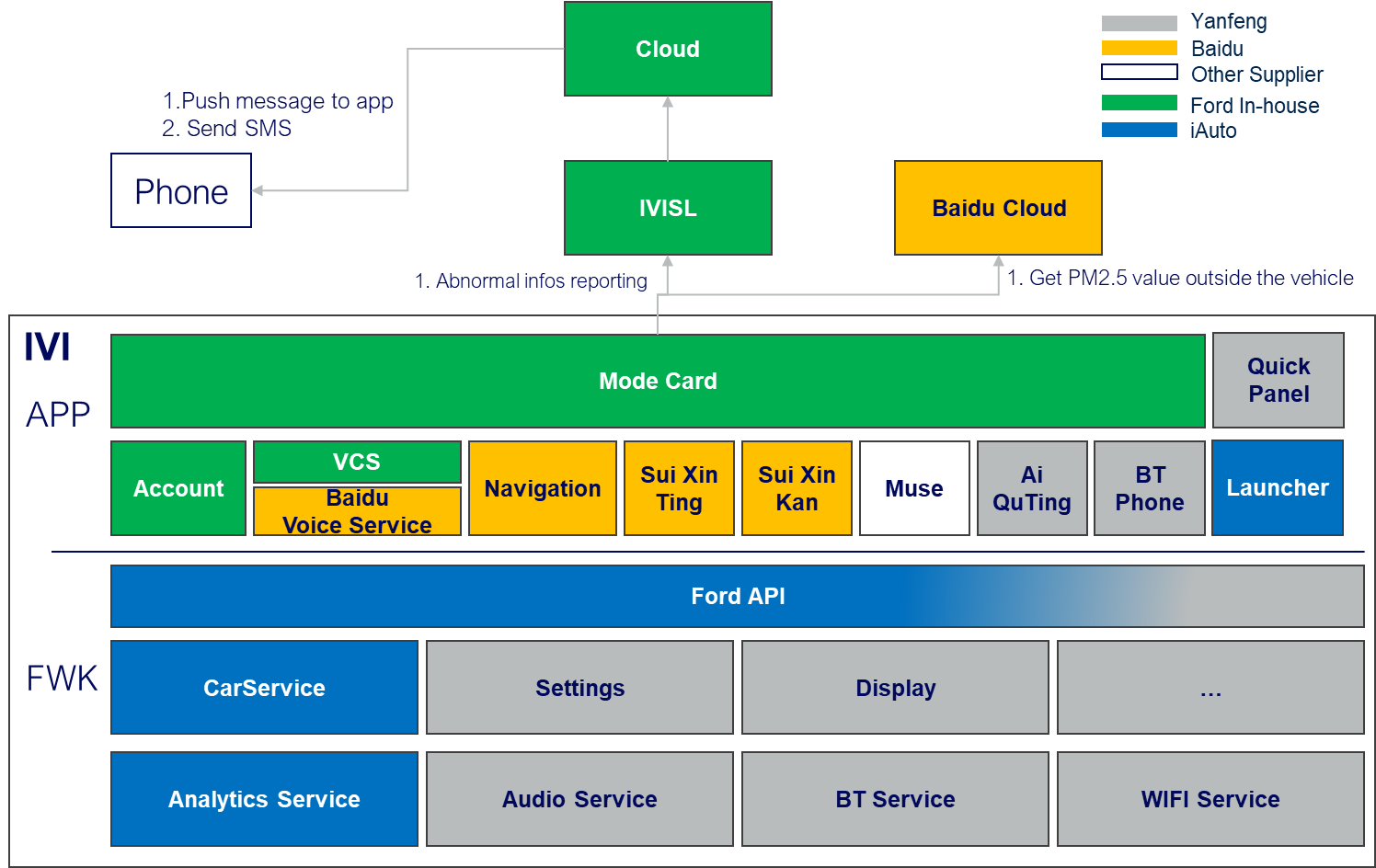
****

Figure 3‑4 SW Architecture (Based on FNV2.1)

## Function Deployment

### Deployment Variants

|  |  |  |
| --- | --- | --- |
| **Deployment Variant Name** | Variant Description | Variant Condition (optional) |
| FNV2 | Based on FNV2 platform | Network Topology = FNV2 |
| FNV2.1 | Based on FNV2.1 platform | Network Topology = FNV2.1 |
|  |  |  |

### Function Allocation

***#Hint:*** *The “Function Allocation Table” shows the mapping of the Logical Functions and the corresponding Technology Functions of a feature to components of the physical architecture as also shown in the deployment diagrams. Typically, there is a 1:1 relationship between (Atomic) Logical and Technology Functions. For details refer to the* *RE Wiki pages* [*“Deriving Implemented Functions from Logical Functions”*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions) *and “*[*Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements%23CascadeRequirements-CascadingVsTraceability)*” When applying MBSE methods please refer to Guideline for Alignment of SW QoS with Ford Starting Model (SysML) for how Logical and Technology Functions in the Ford Starting Model align to Atomic Logical Functions and Technology Functions in RE.*

*For Functional Safety critical features the second table (“Function Allocation Table (Functional Safety Extension)”) has to be additionally filled in*

* *to map Technical Safety Requirements (TSRs) to Technology Functions and hence Components and*
* *to assign an ASIL level to Components and TSRs*

***#Link:***[*RE Wiki - Deriving Implemented Functions from Logical Functions*](http://wiki.ford.com/display/RequirementsEngineering/Deriving+Implemented+Functions+from+Logical+Functions)

[*RE Wiki - Cascade Requirements*](http://wiki.ford.com/display/RequirementsEngineering/Cascade+Requirements#CascadeRequirements-CascadingVsTraceability)

[*Functional Safety Sharepoint*](https://pd3.spt.ford.com/sites/GlobalFunctionalSafety/Pages/default.aspx)

[*Guideline for Alignment of SW QoS with Ford Starting Model (SysML)*](http://wiki.ford.com/display/RequirementsEngineering/Alignment+with+the+Ford+Starting+Model)

| Component | Technology Function Name | Logical Function Name |
| --- | --- | --- |
|
| APIM | Voice Recognition | * Open/Close/Run/End functions via VR commands |
| Navigation | * Distance to destination * Time to destination * Launch Navigation * Cruise mode |
| IVI Cloud | * External PM2.5 value |
| Vehicle setting | * SDM selection * HUD setting * AEIS setting * Digital scent setting * Massage mode/intensity adjustment * DRL setting * Ambient Light setting |
| System setting | * Volume adjustment * Brightness * Calm screen * Screen split * Bluetooth setting * WIFI setting * Hotspot setting * Balance Attenuation * TTS |
| Climate Control | * Temperature * A/C * Dual * Blowing Mode * Auto * Fan speed * Recirc * Defrost * Steering Wheel heating * Cool/Heated Seat |
| Multi-media | * Bluetooth Music * Online news * QQ Music * USB music * Online video |
| BCM | Clock | * Display detailed time * Timer |
| Door status | * Door ajar/closed |
| Window | * Window status |
| Lock | * Vehicle Lock status |
| Frunk | * Frunk status |
| Power Liftgate | * Power Liftgate Status |
| ADAS | Electrical Horizon | * Bridge * Tunnel * Speed limit value |
| SCCM | Rain Sensor | * Judge rain status |
| HVAC/RCCM | PM2.5 | * PM2.5 status |
| External temperature | * External temperature status |
| Internal temperature | * Internal temperature status |
| ABS | Vehicle Speed | * Vehicle speed value |
| EPB | * EPB status |
| HPCM | Engine status | * Engine On/Off |
| Gear Position | * Gear Position value |
| EV | * Charge status * Time to full charge (DC) * Time to full charge (AC) |
| IPC | DTE | * DTE value |
| BECM | Soc (HV) | * Soc percentage value |
| RCM | Seatbelt | * Seatbelt status for each seat |
| Seat occupant | * Seat occupant for passenger |
| DSM | Relax mode | * Active/restore relax mode |
| SCMH/SCMG | Massage | * Active/adjust massage function |

Table 3‑5: Function Allocation Table (condition)

# Feature Implementation Modeling

## Component Interaction Diagrams

For Mode Card application, it can be split into three layers, and “Card Manager”, “Trigger Manger”, “Executor Manager” and “Terminator Manager” are the four important components as component layer. For different modes in business layer, such as commute mode (work/off work),Camp mode, Pet mode, Battery save mode, Glamour studio mode, Relax mode(TBD) and custom mode, the architecture will be more efficient to implement these functions based on mode card platform capability.

The software diagram as below：

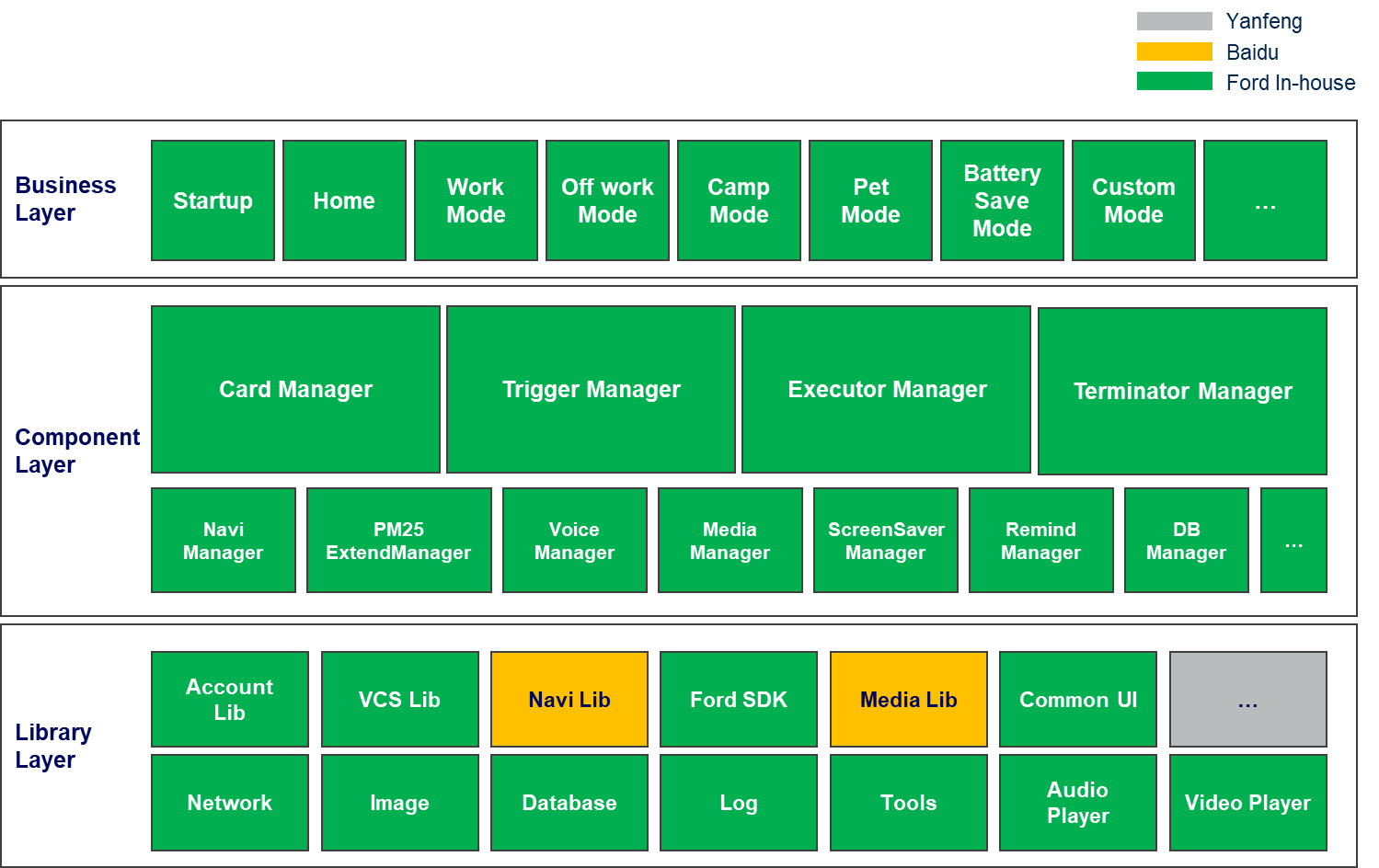


Figure 3‑5 SW Diagram

### Scenario: “Startup Mode Card Process”

|  |  |
| --- | --- |
| ModeCard\_Sequence\_01 | Functions |
| Operation outline | 1. Bootstrap 2. Initialize all card data and enable automatic triggering 3. Initialize the VR function |

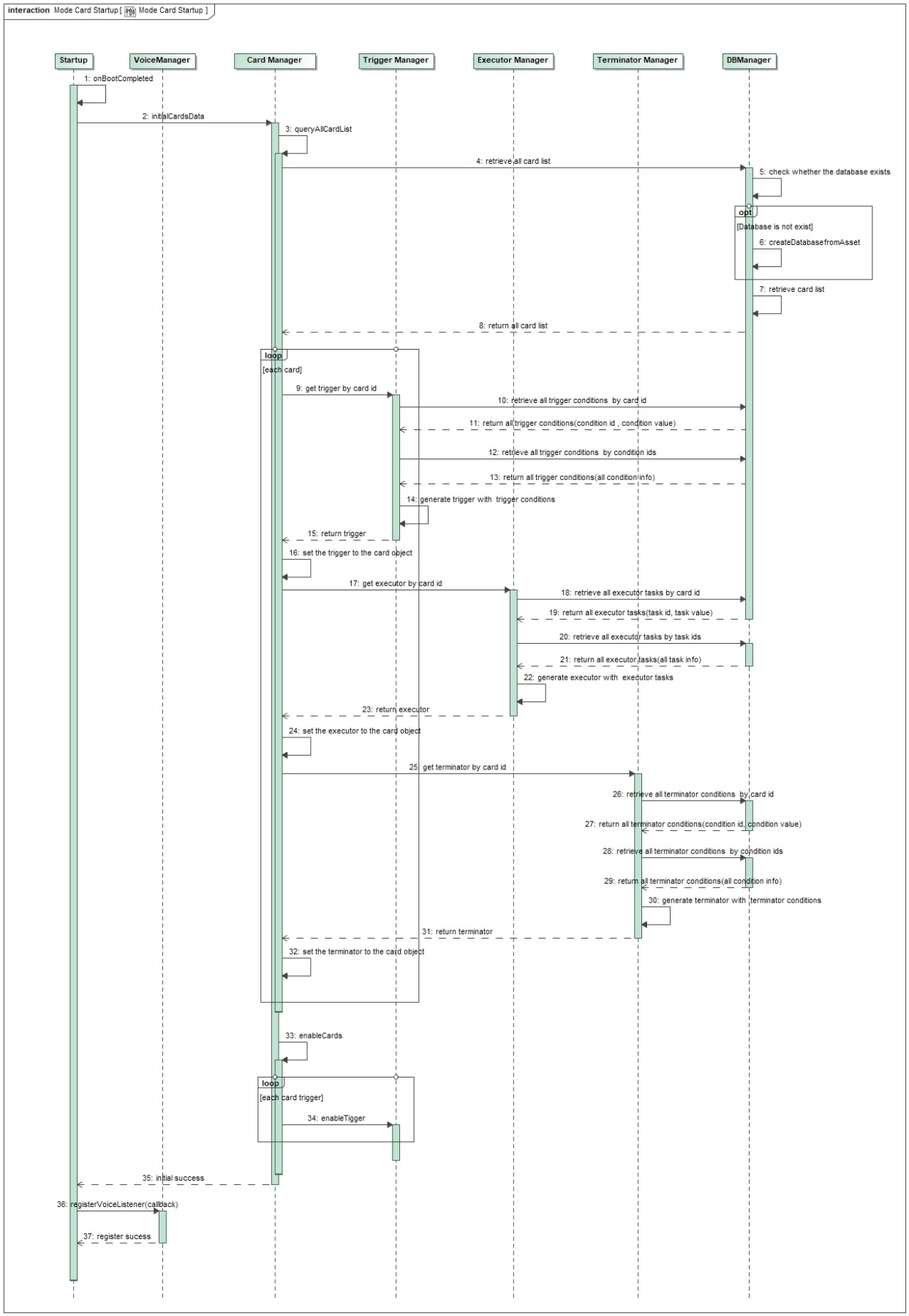


Figure 3‑6 Startup Mode Card Process

### Scenario: “View Recommend Card List”

|  |  |
| --- | --- |
| ModeCard\_Sequence\_02 | Functions |
| Operation outline | 1. Open mode Card homepage manually or by VR 2. Get the list of recommendation cards 3. Display the list of recommendation cards |

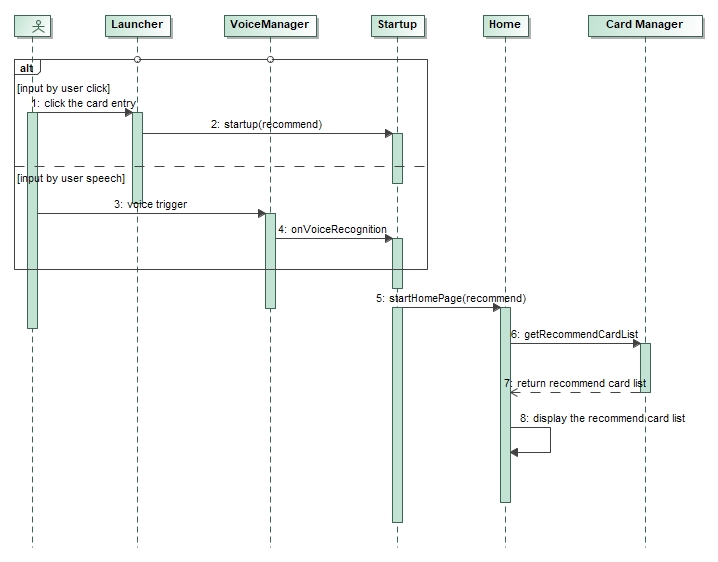


Figure 3‑7 View Recommend Card List

### Scenario: “View custom card list”

|  |  |
| --- | --- |
| ModeCard\_Sequence\_03 | Functions |
| Operation outline | 1. Open the custom card page manually or by voice 2. Get the list of custom mode 3. Display the list of custom mode |

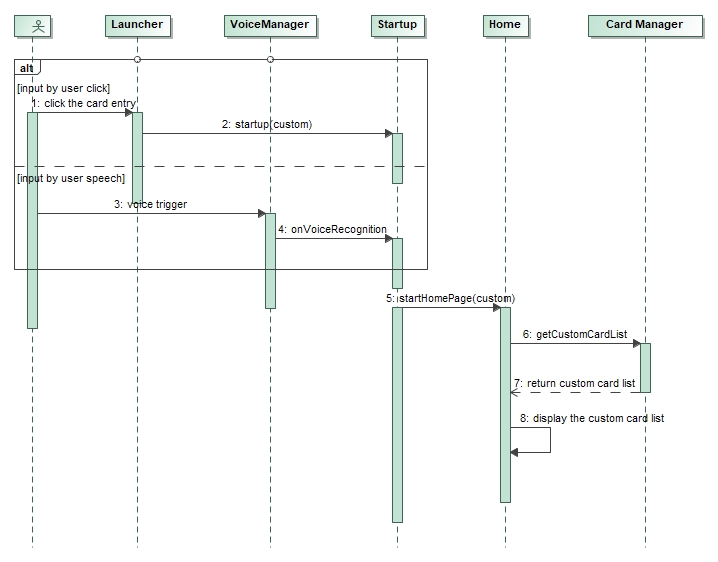
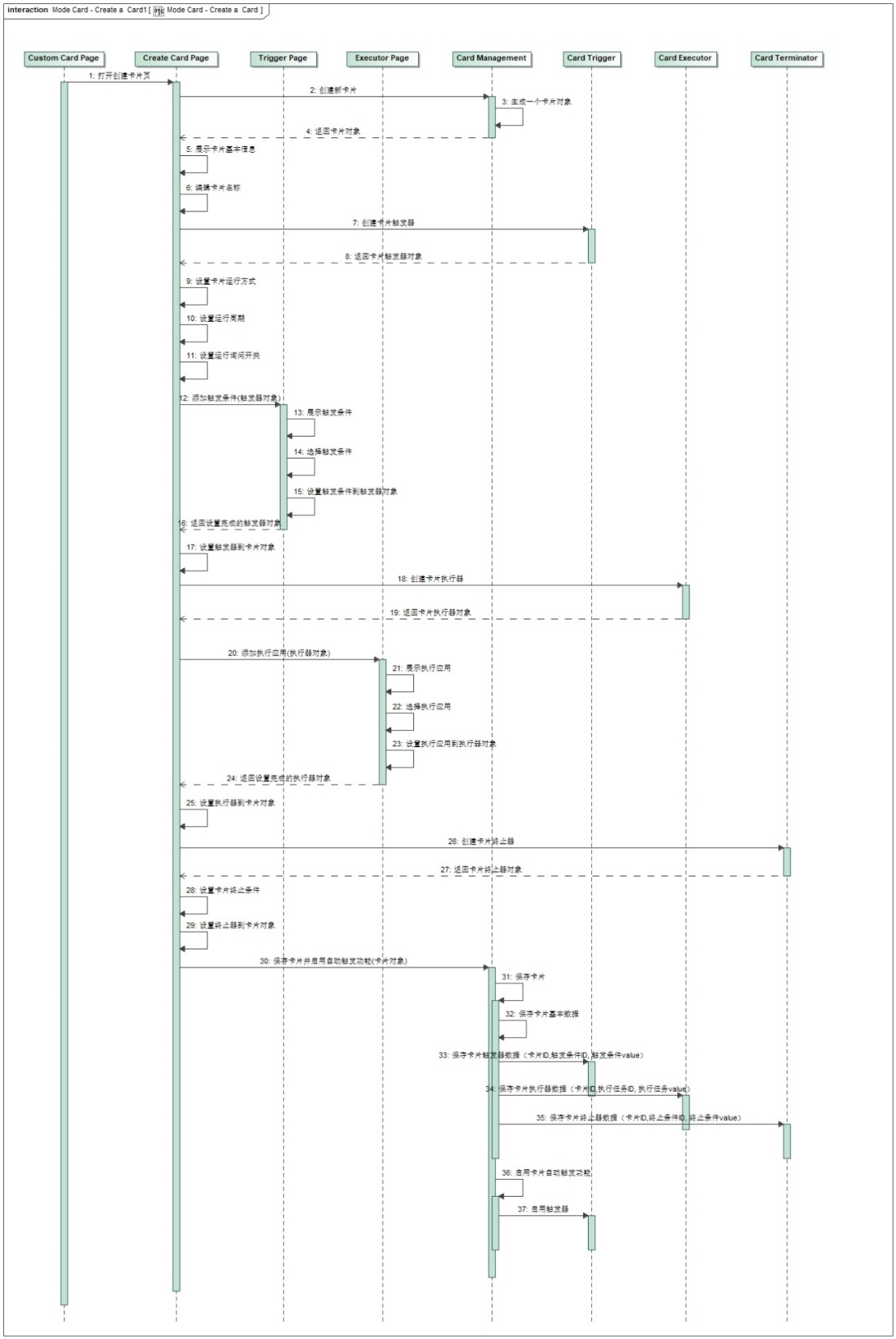


Figure 3‑8 View Custom Card List

### Scenario: “Create a custom Card Mode”

|  |  |
| --- | --- |
| ModeCard\_Sequence\_04 | Functions |
| Operation outline | 1. Create a blank card 2. Edit Card name 3. Set the mode of operation 4. Set the trigger condition 5. Set the cycle of operation 6. Set the query button before running 7. Set the operation task 8. Set the method of termination 9. Save Card 10. Run the Card |



**Figure 3-9:** Create a custom Card Mode

### Scenario: “Run a Card Mode”

|  |  |
| --- | --- |
| ModeCard\_Sequence\_05 | Functions |
| Operation outline | 1. Run the Mode Manually/VR/Automatically 2. Operation environment check 3. Cache the current cockpit setting 4. Execute card tasks |

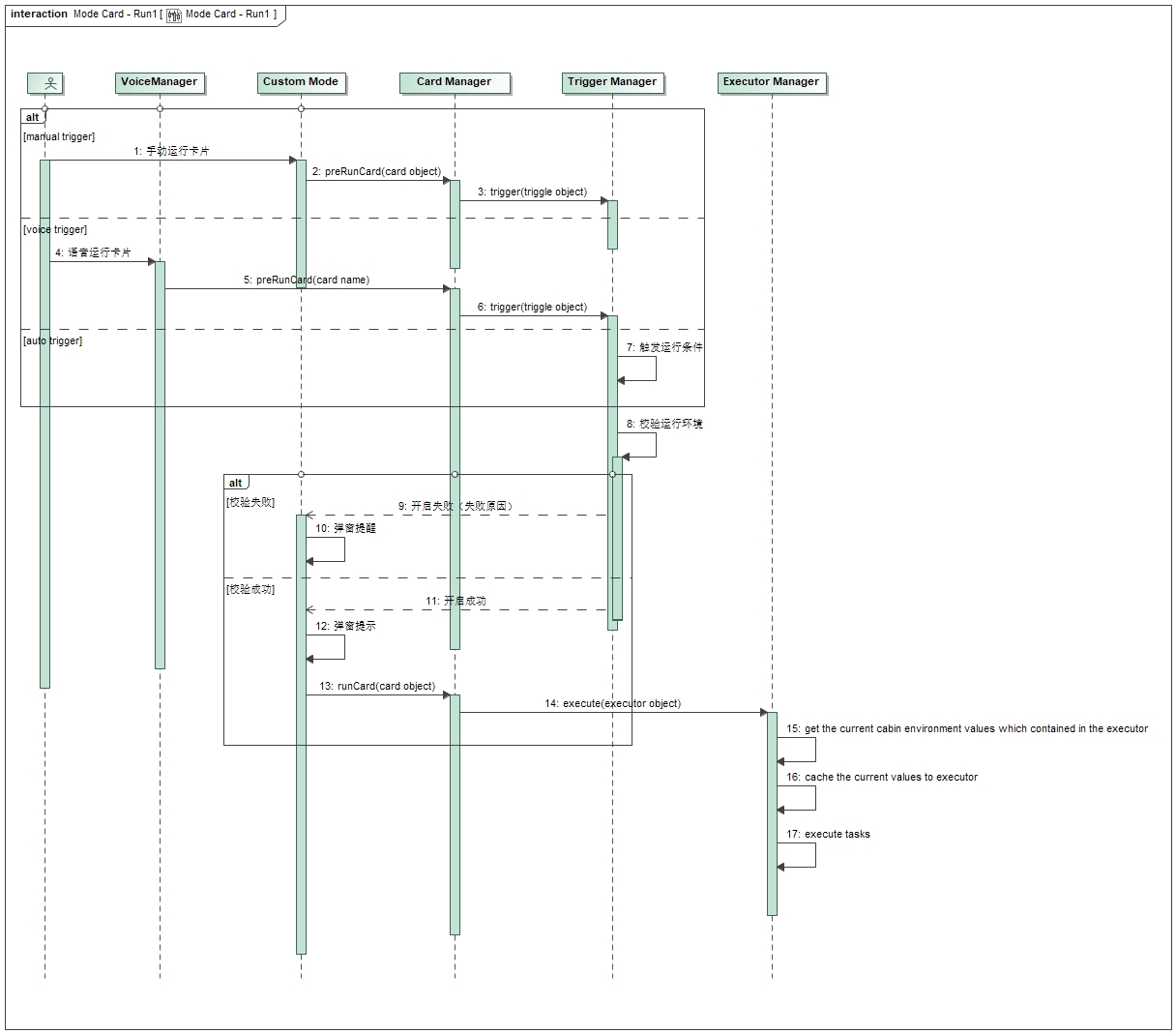


Figure 3‑9 Run a Card Mode

### Scenario: “Terminate a Card Mode”

|  |  |
| --- | --- |
| ModeCard\_Sequence\_06 | Functions |
| Operation outline | 1. Terminate the Mode Manually/VR/Automatically 2. Terminate the running tasks 3. Restore the cockpit setting |

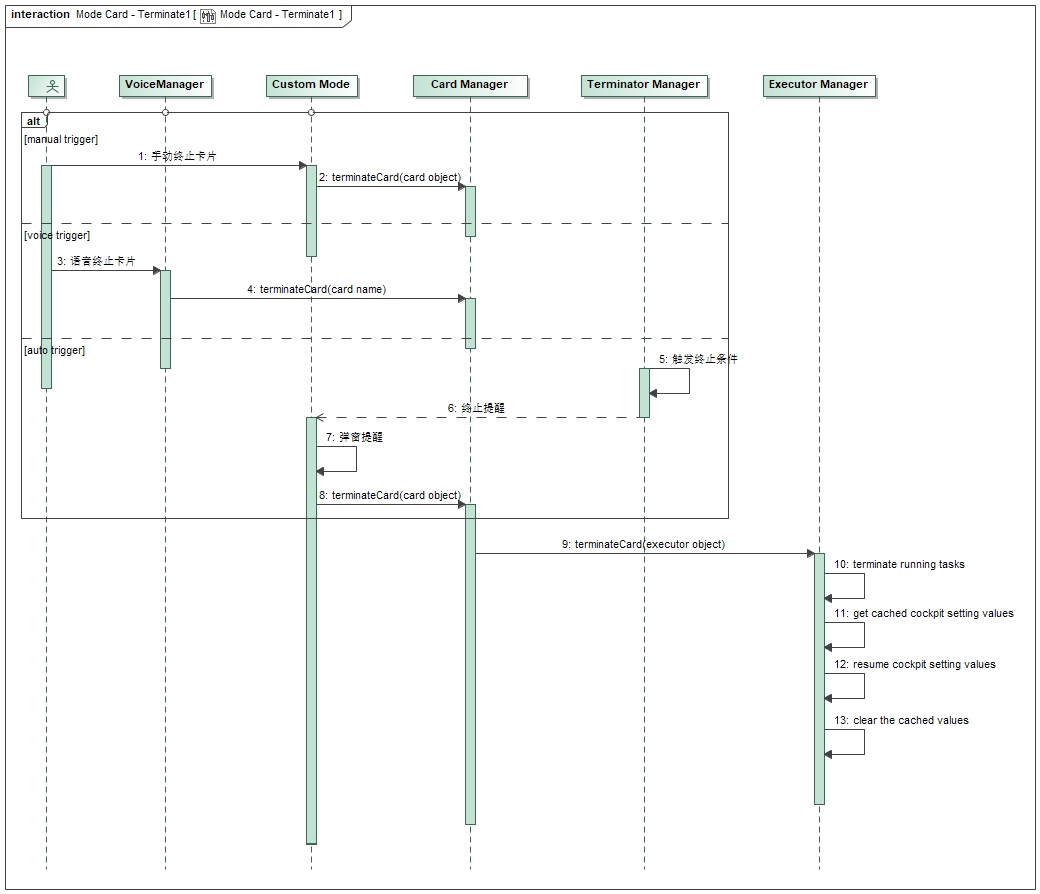


Figure 3‑10 Terminate a Card Mode

## Component Interface Behavior Diagrams

*#Hint: For complex (application level) interface protocols a protocol state machine would be more appropriate than a bunch of sequence diagrams to illustrate the interactions between components. So, this section would typically show a (protocol) state machine.*

# Feature Implementation Requirements

## Functional Safety

Feature is not Function Safety critical – QM level. NA.

## Requirements on Components

For Mode card application, it can be split into “operation condition” and “execution application”. When set any mode of Mode Card, the input is usually the operation condition, and the output is the execution application.

### APIM

#### Technology Function “Operation conditions”

Mode Card provides many available “Operation conditions” for customer based on different vehicle E/E architecture, which can make it as the operation conditions for both recommend scene and custom scene. Operations

##### Function Interfaces

***#Hint:***

*The subsections “Inputs”, “Outputs” and “Parameters” below map the Logical Signals / Parameters their Technical (i.e. physical) counterparts, which are sent and received by the Technology Function. This is done by mapping tables, which reference Data Dictionary entries. If Data Dictionary entries are not sufficient, the Interface Requirements section may be used to specify additional requirements. If possible also link those requirements to the tables in the subsections “Inputs”, “Outputs” and “Parameters”.*

*How to …:*

1. *If the technical signal / parameter does not yet exist in the GSDB or elsewhere in VSEM, create those in the “Technical Signals”/”Technical Parameters” section of the “Data Dictionary”. Use* [*Add Ins -> Add Requirement macro*](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#AddNewRequirement) *(select “Technical Signal”/”Technical Parameter” as type).*

*Note: Bookmarks are automatically created for the ID, the name and the description of each signal / parameter in the “Data Dictionary”, if you use the macros.*

*Note: The corresponding Logical Signal/Parameter should already exist in the Data Dictionary, because it should have been created, when specifying the Logical Function (refer to the Logical Signals/Technical Signals).*

1. *Reference the signal / parameter name bookmark from the Data Dictionary in the tables below.*
2. *If the mapping is not 1:1 (e.g. a Logical Signal gets split into 2 Technical Signals) the Mapping Details need to be specified by a Mappings object in the Data Dictionary.*
3. *For “Publisher Interfaces” and “Subscriber Interfaces” column you may (optionally) link to AIS Interfaces in the Data Dictionary.*
4. *For “Messages” to be referenced in the “E/E Connections” column proceed similarly. Example: <ConnectionName>::<MessageName> refers to the Message which is sent on bus <Connection Name> and which is given in section Messages of the “Data Dictionary”. Alternatively, for CAN you could link directly to the message from the CMDB (e.g.* [*CGEA 1.3*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=n0SJN9h0x3NrTDAAAAAAAAAAAAA&servername=Production_Server) *or* [*FNV2*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*).*

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

**#Link:** [*RE Wiki – Adding a Signal or Parameter Mapping*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

###### Inputs

For Mode card application, the input is usually the operation condition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| GlobalClock\_Data | GlblClkYr\_No\_Actl  GlblClkDay\_No\_Actl  GlblClkHr\_No\_Actl  GlblClkMnte\_No\_Actl  GlblClkScnd\_No\_Actl | N. A | N. A | N. A |
| VR\_data | IVI internal command | N. A | N. A | N. A |
| ElecHorizon\_Data | 桥梁：adas\_mini\_Bridge  隧道：adas\_mini\_Tunnel  限速：adas\_mini\_EffSpdLim | N. A | N. A | N. A |
| Nav\_Send\_data | Nav\_DistanToDestUnit\_St  Nav\_DistancToDestVal\_St  Nav\_RemainTTDestDays\_St Nav\_RemainTTDestHour\_St Nav\_RemainTTDestMin\_St | N. A | N. A | N. A |
| Light\_data | Day\_Night\_Status | N. A | N. A | N. A |
| Rain Wiper\_data | WiprFront\_D\_Stat | N. A | N. A | N. A |
| Outside\_Pm\_data | Cloud data | N. A | N. A | N. A |
| Inside\_Pm\_data | CabnAmb\_Te\_Actl | N. A | N. A | N. A |
| Outside\_Temp\_data | Outside\_Air\_Temp\_Stat | N. A | N. A | N. A |
| Inside\_Temp\_data | CabnAmb\_Te\_Actl | N. A | N. A | N. A |
| Veh\_speed\_data | Veh\_V\_ActlEng | N. A | N. A | N. A |
| HV\_Bat\_data | BattTracSoc\_Pc\_Dsply | N. A | N. A | N. A |
| Range\_data | VehElRnge\_L2\_Dsply | N. A | N. A | N. A |
| Gear\_position | GearLvrPos\_D\_Actl | N. A | N. A | N. A |
| GearPos\_D\_Trg | N. A | N. A | N. A |
| Park\_Brake | PrkBrkStatus | N. A | N. A | N. A |
| Ignition\_data | Ignition\_Status | N. A | N. A | N. A |
| Eng\_D\_Stat | N. A | N. A | N. A |
| Door\_data | DrStatDrv\_B\_Actl  DrStatPsngr\_B\_Actl  DrStatRl\_B\_Actl  DrStatRr\_B\_Actl | N. A | N. A | N. A |
| Trunk\_data | TrunkFrontOpen\_D\_Stat  DrStatHood\_B\_Actl | N. A | N. A | N. A |
| Tailgate\_data | DrStatInnrTgate\_B\_Actl | N. A | N. A | N. A |
| Window\_data | PassWindowPosition  RearPassWindowPos  DriverWindowPosition  RearDriverWindowPos | N. A | N. A | N. A |
| Lock\_data | Veh\_Lock\_Status |  | N. A | N. A |
| Seatbelt\_data | VedsDrvBelt\_D\_Ltchd  VedsRw1PasBckl\_D\_Ltchd  VedsRw2dBckl\_D\_Ltchd  VedsRw2mBckl\_D\_Ltchd  VedsRw2pBckl\_D\_Ltchd |  | N. A | N. A |
| Psngr\_detect\_data | PsngrFrntDetct\_D\_Actl |  | N. A | N. A |
| EV\_data | ChrgStat\_D2\_Dsply | Charge status | N. A | N. A |
| ActChrgStrtYr\_No\_Actl  ActChrgStrMnth\_No\_Actl  ActChrgStrtDay\_No\_Actl  ActChrgStrtHr\_No\_Actl  ActChrgStrtMin\_No\_Actl | Charge start time | N. A | N. A |
| ActChrgEndYr\_No\_Actl  ActChrgEndMnth\_No\_Actl  ActChrgEndDay\_No\_Actl  ActChrgEndHr\_No\_Actl  ActChrgEndMin\_No\_Actl | Charge end time | N. A | N. A |
| ChargeNowDuration\_St | AC charge duration | N. A | N. A |
| EstmChrgTimeHP\_St | N. A | N. A |
| EstmChrgTimeLP\_St | N. A | N. A |
| FstChrgCmplt\_T\_Est | DC charge 100% | N. A | N. A |

Table 5‑2: Input Signal mappings of Function “MyLogicalFunctionA\_Component1”

###### Outputs

For Mode card application, the output is usually the execution application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
| Navi\_data | internal data | * Launch Navigation * Cruise mode | N. A | N. A |
| Media\_data | internal data | * QQ Music * FM * BT music * Xi Ma La Ya * Online news * Muse (optional) | N. A | N. A |
| Volume\_data | internal data | * Prompt volume * VR volume * Media volume * Phone volume | N. A | N. A |
| Video\_data | internal data | * Ai Qi Yi | N. A | N. A |
| Display\_data | internal data | * Brightness * Calm Screen * Screen split * TTS | N. A | N. A |
| SDM\_data | SelDrvMde\_D\_RqDrv | * Request | N. A | N. A |
| ActvDrvMde\_D2\_Stat | * Status | N. A | N. A |
| Setting\_data | internal data | * BT * WIFI * Hotspot * Balance Attenuation * HUD setting * BT secret mode | N. A | N. A |
| AEIS\_data | **Tx**:  CtrStkDsplyOp\_D\_Rq  CtrStkFeatNoActl  CtrStkFeatConfigActl  CtrStkPersIndex\_D\_Actl  **Rx**:  FeatNoIpcActl  FeatConfigIpcActl  PersIndexIpc\_D\_Actl | * CX727 (FNV2) | N. A | N. A |
| EngIdleShutDown\_B\_RqDrv | * CX821 (FNV2.1) | N. A | N. A |
| Scent\_data | AC\_1\_FGA\_OperationReq  AC\_1\_FGAChanTypSelect  AC\_1\_FGAIntensityReq | * Lin Data | N. A | N. A |
| Climate\_data | Drv\_Set\_Temp  Psngr\_Set\_Temp  Frt\_Blower\_Speed  Frt\_Btn\_Status\_1st |  | N. A | N. A |
| Seat\_data | Frt\_Btn\_Status\_1st 0x1C LHS\_Cld\_Seat\_Pressed  Frt\_Btn\_Status\_1st 0x1D RHS\_Cld\_Seat\_Pressed | * Cool seat * Heat Seat | N. A | N. A |
| SeatRelxPosActv\_B\_Rq  SeatRelxPosDactv\_B\_Rq  SeatRelxPosSet\_D\_Rq | * Relax mode | N. A | N. A |
| Massage\_data | SeatFnDrv\_D\_Rq  SeatFnDrv\_D\_Stat  SeatFnPsgr\_D\_Rq  SeatFnPsgr\_D\_Stat | * Massage on/off | N. A | N. A |
| SeatFnDrv\_D\_Rq  SeatMasgDrv\_D\_Rq SeatFnDrv\_D\_Stat  SeatMasgDrv\_D\_Stat  SeatFnPsgr\_D\_Rq  SeatMasgPsngr\_D\_Rq  SeatFnPsgr\_D\_Stat  SeatMasgPsngr\_D\_Stat | * Massage Mode selection | N. A | N. A |
| SeatFnChngDrv2\_D\_Rq  SeatIntnsDrv\_D\_Stat  SeatFnChngPsgr2\_D\_Rq  SeatIntnsPsngr\_D\_Stat | * Massage Intensity | N. A | N. A |
| DRL\_data | **Tx**:  CtrStkDsplyOp\_D\_Rq (0x2 Set)  CtrStkFeatNoActl  CtrStkFeatConfigActl  CtrStkPersIndex\_D\_Actl  **Rx**:  FeatNoIpcActl  FeatConfigIpcActl  PersIndexIpc\_D\_Actl | * DRL on/off | N. A | N. A |
| AL\_data | LightAmbColor\_No\_Rq  LightAmbIntsty\_No\_Rq | * CX727 Ambient Light | N. A | N. A |
| TBD | * CX821 Ambient Light | N. A | N. A |

Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

NA.

###### Interface Requirements

Refer to < Mode Card Full\_Range\_interface\_List >.

##### Function Requirements

###### Voice Recognition function

Mode Card should open/start/run/end via voice recognition：

* Open/Exit Mode Card entrance
* Run/end sub-scene (Camp mode, Commute mode, guest mode etc.)
* Abnormal scenario (VR commands fail)

###### Clock

Mode Card application could get the clock data via two ways, one is system time via IVI Software internal variate, the other is below BCM clock data：

* Customer can set the detailed start time or end time in Mode Card application
* Customer can set the running duration for any mode in Mode Card application

If use global clock strategy, please refer to <Global Clock Strategy Specification>

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| GlblClkYr\_No\_Actl | year | 0x00 – 0xFE | Offset <2000> |
| GlblClkDay\_No\_Actl | day | 0x01 – 0x16E |  |
| GlblClkHr\_No\_Actl | Hour | 0x00 – 0x17 |  |
| GlblClkMnte\_No\_Actl | Minute | 0x00 – 0x3B |  |
| GlblClkScnd\_No\_Actl | Second | 0x00 – 0x3B |  |
| Values for Month, Day… shall be interpreted as Julian date (e.g. ´Year is 2009 so BCM\_Year =0x9, Day is April 30th so BCM\_Julian\_Day = 0x78). Julian day goes from 1 – 366 with 0x0 being the initial startup value. | | | |
| The max. time jitter shall not exceed 100ms on the event distribution of the signal. | | | |

###### Navigation data

Mode Card should have below navigation related functions：

* Start Navigation to Home/Company
* Start Navigation only (default as cruise mode)
* Distance to destination
* Remain time to destination
* Judge the location (Bridge/Tunnel)
* Judge the speed limit (via ADAS map/HD map)

Mode Card application could get the Navigation data via two ways, one is Navigation application via IVI Software internal variate, the other is Transport protocol as below：

If use transport protocol strategy, please refer to < Transport Protocol APIM SPSS> and < NAV Repeater V2

APIM SPSS>.

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| Nav\_DistanToDestUnit\_St | Kilometers\_km | 0x0 |  |
| Meters\_m | 0x1 |  |
| Miles\_mi | 0x2 |  |
| Yards\_yd | 0x3 |  |
| Nav\_DistancToDestVal\_St | Distance | 0x0 – 0xFFFE |  |
| Nav\_RemainTTDestDays\_St | days | 0x0 – 0xD |  |
| Exceeding13days | 0xE |  |
| Nav\_RemainTTDestHour\_St | Hours | 0x0 – 0x1F |  |
| Nav\_RemainTTDestMin\_St | Minutes | 0x0 – 0x3F |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| adas\_mini\_Bridge | Not Part of Bridge | 0x0 |  |
| Part of Bridge | 0x1 |  |
| Unknow | 0x2 |  |
| N/A | 0x3 |  |
| adas\_mini\_Tunnel | Not Part of Tunnel | 0x0 – 0xFFFE |  |
| Part of Tunnel |  |  |
| Unknow |  |  |
| N/A |  |  |
| adas\_mini\_EffSpdLim | unknown | 0x0 |  |
| <=5 | 0x1 |  |
| 7 | 0x2 |  |
| 10-120 | 0x3-0x19 | Step 5 |
| 120-160 | 0x19-0x1D | Step 10 |
| Unlimited | 0x1E |  |
| Invalid | 0x1F |  |

###### Light status judgement function

Mode Card can judge the day/night status via below logical signal:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| Day\_Night\_Status | Null | 0x0 |  |
| Day | 0x1 |  |
| Night | 0x2 |  |
| NotUsed | 0x3 |  |

###### Front Wiper judgement status

Mode Card can judge the front wiper status to predict the rain status via below logical signals:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| WiprFront\_D\_Stat | OFF | 0x0 |  |
| Auto\_OFF | 0x1 |  |
| MAN\_INT\_OFF | 0x3 |  |
| MAN\_INT\_ON | 0x4 |  |
|  | MAN\_LOW | 0x5 |  |
|  | MAN\_High | 0x6 |  |
|  | AUTO\_LOW | 0x9 |  |
|  | AUTO\_High | 0xA |  |
| **Rain or not prediction rule**:  1. **Man\_Low/Auto low/Man\_High/Auto high**: Last 15 seconds  2. **Man\_int\_off --> on** exchange ≥4 times within 30 seconds  3. **Auto off--> auto low** exchange ≥4 times within 30 seconds | | | |

###### PM2.5 status judgement

Mode Card can judge the internal and external PM2.5 status.

* For external PM2.5 status, it usually get from cloud side
* For internal PM2.5 status, can get via below logical signal:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| PmCabn\_Conc\_Actl | 0-509 | 0x0-0x1F | parts/million |
| NoDataExists | 0xFE |  |
| Faulty | 0xFF |  |

###### Temperature judgement

Mode Card can judge the internal and external temperature status via below logical signals.

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| CabnAmb\_Te\_Actl | -57 – 70℃ | 0x0-0xFE | 1. Internal Temperature 2. Resolution 0.5℃ |
| Fault | 0xFE |
| Outside\_Air\_Temp\_Stat | -40 – 86.5℃ | 0x0-0xFD | 1. External Temperature 2. Resolution 0.5℃ |
| Unknown | 0xFE |
| Invalid | 0xFF |

###### Vehicle Speed Judgement

Mode Card can judge the vehicle speed via below logical signal:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| Veh\_V\_ActlEng | 0-655.35 | 0x0-0xFFFF | Kph |

###### HV battery percentage

Mode Card can judge the vehicle speed via below logical signal:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| BattTracSoc\_Pc\_Dsply | 0-126.5 | 0x0 - 0xFD | Resolution 0.5% |
| NoDataExists | 0xFE |
| Faulty | 0xFF |

###### DTE judgement

Mode Card can judge the DTE via below logical signal:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| VehElRnge\_L2\_Dsply | 0-818.9 | 0x0 - 0x1FFD | Resolution 0.1KM |
| NoDataExists | 0x1FFE |
| Faulty | 0x1FFF |

###### Gear Position judgement

Mode Card can judge the Gear position via below logical signal:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| GearLvrPos\_D\_Actl | Park | 0x0 | Only list the necessary value |
| Reverse | 0x1 |
| Neutral | 0x2 |
| Drive | 0x3 |
| Sport\_DriveSport | 0x4 |

###### EPB status judgement

Mode Card can judge the EPB status via below logical signal:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| PrkBrkStatus | Not\_Supported | 0x0 |  |
| Rear\_Caliper\_Closed | 0x1 |
| Rear\_Caliper\_Transition | 0x2 |
| RWU\_By\_EPB\_Active | 0x3 |
| Rear\_Caliper\_Open | 0x4 |
| EPB\_Limphome\_Active | 0x5 |
| ECD\_by\_Brake\_ECU\_Active | 0x6 |
| GeneralFault\_MaintenceMode | 0x7 |

###### Ignition status judgement

Mode Card can judge the ignition status via below logical signals:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| Ignition\_Status | Unknown | 0x0 |  |
| Off | 0x1 |
| Accessory | 0x2 |
| Run | 0x4 |
| Start | 0x8 |
| Invalid | 0xF |
| Eng\_D\_Stat | EngOff | 0x0 |  |
| EngOn | 0x1 |
| EngAutoStopped | 0x2 |
| NotUsed | 0x3 |

###### Door Status Judgement

Mode Card can judge the door status via below logical signals:

|  |  |  |  |
| --- | --- | --- | --- |
| **Logical Signal Name** | **Literals** | **Value** | **Description** |
| DrStatDrv\_B\_Actl | Closed | 0x0 | Driver Door |
| Ajar | 0x1 |
| DrStatPsngr\_B\_Actl | Closed | 0x0 | Passenger Door |
| Ajar | 0x1 |
| DrStatRl\_B\_Actl | Closed | 0x0 | Rear left Door |
| Ajar | 0x1 |
| DrStatRr\_B\_Actl | Closed | 0x0 | Rear right Door |
| Ajar | 0x1 |
| TrunkFrontOpen\_D\_Stat |  |  |  |
|  |  |  |
| DrStatHood\_B\_Actl |  |  |  |
|  |  |  |

#### (Technology) Function “MyLogicalFunctionB\_Component1”

### Cloud

## Requirements on Connections

### Networks

#### “CAN Bus xxx”

***#Hint:*** *For CAN most requirements are defined by Netcom and referenced in* [*VSEM “Multiplexing Specifications” section*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=xcbJ6OwAx3NrTDAAAAAAAAAAAAA&servername=Production_Server)*. Put in this section only those requirements, which deviate from that standard specification.*

*The CAN messages relevant for this feature are listed in the section “Messages” of the “Data Dictionary”.*

##### Protocol Requirements

**#Hint:** *For CAN Ford currently mandates FNOS as SW implementation for the CAN protocol stack. This includes the CAN Network Management and Transport Protocol. If you deviate from this assumption or if you have specific requirements on FNOS, which are not contained in the standard package, put requirements in this section. F*or details the FNOS user guide and application notes could be referenced.

##### Electrical Requirements

**#Hint:** List requirements here, only if they deviate from the SDS CAN.

#### “LIN Bus xxx”

**#Hint:** Place requirements here, which are common to all LIN nodes, but not covered by some SDS LIN.

*The LIN messages relevant for this feature are listed in the section “Messages” of the “Data Dictionary”.*

##### Protocol Requirements

###### Schedule Table

***#Hint:*** *The LIN Schedule Table should be documented in the LDF file. The LDF file could be referenced here*

##### Electrical Requirements

***#Hint:*** *The LIN Schedule Table should be documented in the LDF file. The LDF file could be referenced here*

#### “Ethernet xxx”

***#Hint:*** *On Ethernet (wired or wireless) we will see most likely the DoIP, MQTT or V2x protocols. Those protocols are described in separate specifications and are implemented in the Ford AUTOSAR stack. While DoIP might be not that relevant in this scope, MQTT (together with the Google Protocol Buffer (GPB) serialization of the payload) will become important for all features, which are mapped to a Service Oriented Architecture/Communcation (SoC). Application data (SOA APIs), which is transmitted via MQTT, are listed in the data dictionary section “AUTOSAR Interfaces*

#Hint: Those AUTOSAR Classic (Sender/Receiver and Client/Server) Interfaces, which are used by the feature but not managed in a central repository yet, should be listed here.

SOA Service *”.*

***#Link:***[*http://www.mqtt.org*](http://www.mqtt.org)*, https://developers.google.com/protocol-buffers/docs/proto*

### HW I/Os

**#Hin**t: This section lists all hardwired signals relevant for the feature deployment. Those get typically mapped to VSEM EDAS signals – refer to list of connections in corresponding table in chapter ”E/E Architecture → E/E Connections”. If any specific protocol is used to send/receive signal information or multiplex/demultiplex signals on the HW circuit.

#### “HW I/O xxx”

## Requirements on Development Process

# Open Concerns

***#Hint:*** *The following list presents known issues that have to be discussed or clarified over the course of the on-going requirements engineering.*

| ID | Concern Description | e-Tracker Reference | Status | Solution |
| --- | --- | --- | --- | --- |
| 1 | How to fully capture Wake/Sleep requirements. Currently we have a mismatch between what is captured in old EuCD SRD requirements and what is captured in AIS Publisher Interfaces (Publishing Network Sleep Inhibitor, Network Wake Up) |  | Open | Extend AIS attributes? |
| 2 | Clarify how to export Message list entries from CMDB in VSEM |  | Open |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

Table 6‑1: Open Concerns

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Description | Approved by | Responsible |
| A |  | Initial version |  | Jbaden1 |
|  |  |  |  |  |

## Template Revisions

*#Important: Do not change this section*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Rev. | Date | Description | Responsible |
| 0 | 2 | 2015-08-05 | * TOC corrected * Document Properties adapted to match needs of VBA macros | Awegman1 |
| 1 | 0 | 2015-11-16 | * Revision History moved to chapter 7 * Table-Styles removed | Awegman1 |
| 1 | 1 | 2016-03-02 | * Rework according to PCL example | Jbaden1 |
| 1 | 2 | 2016-03-22 | * V1.3: Footer formating corrected (Issue 19) * “Constraints” chapter renamed to “Input Requirements” (Issue 20) | Jbaden1 |
| 1 | 3 | 2016-04-20 | * Broken Wiki links repaired | Jbaden1 |
| 2 | 0 | 2016-05-23 | * Prepared for Specification\_Macros.dotm v2.0 * Additional explanations added to ch. 2.2 “Input Requirements” (ARL and SDS requirements often go here) | Jbaden1 |
| 2 | 1 | 2016-07-08 | * Template version added to footer | Jbaden1 |
| 2 | 2 | 2016-07-15 | * Sample SysML diagrams added * Data Dictionary reworked * Alignment with relevant sections in SRD templated | Jbaden1 |
| 3 | 0 | 2016-09-05 | * Lessons learned from IPRB incorporated | Jbaden1 |
| 4 | 0 | 2016-09-27 | * Alignment with QPIP Feature Function Ownership workstream. Platform Spec renamed to Feature Implementation Spec | Jbaden1 |
| 4 | 1 | 2016-11-04 | * Chapters “Purpose” and “Scope” reworked. | Jbaden1 |
| 4 | 1 | 2016-11-10 | * Subsection for “Logical Service Interfaces” added. | Jbaden1 |
| 5 | 0 | 2017-01-13 | * Meta data updated for specification macros, version 3.1 * SW Unit chapter removed for the time being * Green boxes added for user hints | Jbaden1 |
| 5 | 1 | 2017-01-18 | * Minor editorial changes (e.g. hyperlinks highlighted in comments) | Jbaden1 |
| 5 | 1b | 2017-01-20 | * Some editorial corrections * Substructure of old Network Communication (now Connections) moved to Requirements on Connections | Jbaden1 |
| 6 | 0 | 2018-07-24 | * CR53: * Add new cover sheet * Add disclaimer section * Add the following meta-data to the doc properties for the the new cover sheet   + DocGis1ItemNumber   + DocGis2Classification   + DocType   + DocStatus   + DocIssueDate   + DocReleaseDate * CR63: Update FuSa sharepoint references in templates | Jbaden1 |
| 6 | 0 | 2018-08-06 | * CR81: Incorporate lessons learned from System Service Spec pilot (Vehicle Speed) into AFS and FIS | Jbaden1 |
| 6 | 0 | 2018-09-28 | * Broken links to RE Wiki repaired | Jbaden1 |
| 6 | 0 | 2018-10-31 | * Minor corrections on cover sheet and in footer to be more GIS compliant and VSEM aligned * “Overview” and “Description” exchanged in headings (following common sense) | Jbaden1 |
| 6 | 0 | 2018-11-30 | * Update of Functional Safety sections after review by Functional Safety Team * Initial support for variant handling | Jbaden1 |
| 6 | 0 | 2018-12-01 | * Variant condition fields added consistently * Links updated | Jbaden1 |
| 6 | 0 | 2018-12-11 | * Variant condition fields removed from mapping/allocation tables * Mapping tables simplified * Explanatory text for “Variants” sections revised | Jbaden1 |
| 6 | 0a | 2019-01-04 | * Chapter heading “Inherited Function Requirements” removed. Corresponding table renamed to “Requirements not cascaded”. * E/E Connection table got another column for allocated messages * Naming conventions for Implemented Functions corrected (FncName\_CmpName instead of FncName\_on\_CmpName) * Editorial corrections on the cover sheet * Explanatory text added to “Ethernet” section in chapter “Requirements on Connections” * AIS templates updated. Linked to Wiki page | Jbaden1 |
| 6 | 0a | 2019-01-04 | * Minor restructuring in FuSa chapter – after aligning with ECU Functional Spec * Bugfix: table 13 renamed from FTTI table to FHT table, includes a bug fix: each FSR is allocated to only one ECU/component | Jbaden1 |
| 6 | 0b | 2019-02-04 | * Change: Chapter “Interface Requirements” added to “Implemented Function xxx” section (to have a single chapter for to collect subscriber/publisher interface and mapping requirements which to not conform to the corresponding Data Dictionary objects) * Change: “CAN Interface” subsection renamed to “AIS Interfaces” again. Although several Subscriber/Publisher interface attributes are probably CAN bus specific, other attributes seem to be well suited for other networks than CAN. * Change: Chapter “ECU Specific Requirements” renamed to “Component Specific Requirements” in chapter “Implemented Function xxx”. Table “Requirements not cascaded” renamed to “Component Specific Requirements” and refined to describe changes from Logical Function requirements set more formally. This is also to help during VSEM import to identify those requirements of the Logical Function which cannot be simply carried over to the ECU. * Change: Explanatory text in section “Implemented Function xxx” improved. | Jbaden1 |
| 6 | 0c | 2019-02-05 | * Change: Layout of AIS Interfaces in Data Dictionary reworked to enable Excel Import | Jbaden1 |
| 6 | 0c | 2019-02-20 | * Bugfix: In AIS Interfaces none-picklist fields formatted as invisible | Jbaden1 |
| 6 | 1a | 2019-02-05 | Functional Safety related changes:   * Table “Architectural Redundancy Summary” updated * Section “Functional Flows for FTTI ‘xyz’” added to chapter “Component Interaction Diagrams” * Fault Tolerant Time Summary section added to Functional Safety chapter * Chapter “HW Metrics” added | Jbaden1 |
| 6 | 1a | 2019-04-02 | Headings of “Architectural Redundancy Summary” table clarified | Jbaden1 |
| 6 | 1a | 2019-04-10 | * ASIL Decomposition table moved from Function Spec into the Feature Implementation Spec (ASIL Decomposition of Technical Safety Requirements) * 2 alternative versions of the Function Allocation Table (Standard variant vs. Functional Safety variant) placed next to each other. | Jbaden1 |
| 6 | 1a | 2019-05-31 | * Function Allocation Table split into a base (non FuSa) part and a FuSa part to allow a more flexible mapping of MBSE functions (Logical and Technology) to RE functions (Atomic Logical and Implemented). | Jbaden1 |
| 6 | 1a | 2019-05-31 | * “Input Requirement” section reworked (symmetrically to all other templates). * Sections “Functional Flows for FTTI xyz” and “Fault Tolerant Time Summary” removed, because guidance is not available yet. * “Reference” and “Glossary” section moved back to introduction, i.e., to the very beginning of the document (such that also section 2 can already rely on it). * Some mostly editorial changes per request from FuSa team. | Jbaden1 |
| 6 | 1a | 2019-07-02 | * "Important" box added on cover sheet which points to the macros * “Input Requirements” section renamed to Input Information (after discussion with FuSa team) | Jbaden1 |
| 6 | 1a | 2019-07-17 | * Chapter “Message List” removed from CAN and LIN specific chapters of section “Requirements on Connections” | Jbaden1 |
| 6 | 1a | 2019-10-08 | * Chapter “ASIL Decomposition of Technical Safety Requirements”: Input TSRs are specified in the chapter right above the decomposition table. | Jbaden1 |
| 6 | 1a | 2019-10-09 | * Chapter “Service Oriented Communication” moved to section “Messages” in the Data Dictionary. Details from Central SW Wiki about FNV2 SOA added | Jbaden1 |
| 6 | 1a | 2019-10-25 | * Minor updates for HW IOs/Signals * Subsection “Functional Safety” removed from chapter “Feature Implementation Modeling”. Per requrest from FuSa team since no guidance is available how to model e.g. FHT timing diagram. | Jbaden1 |
| 6 | 1a | 2019-05-11 | * Copyright notice shortened and moved to cover sheet and added to footer (to be compliant [with Ford copyright guidelines](http://www.fgti.ford.com/client/NewFGTI/CopyrightNotice.html)) * Term “Disclaimer” no longer used for what is actually only a copyright notice | Jbaden1 |
| 6 | 1a | 2019-22-11 | * Some minor modifications for the SOA APIs/MQTT Messages in the section “Messages” of the Data Dictionary (section references Service Contracts via the API name) * Some minor updates of the Input/Output mapping tables in section “Requirements on Components” for mappings to SOA APIs and EDAS signals. | Jbaden1 |
| 6 | 1a | 2019-12-05 | * Upstream Documents section added to “Input Requirements/Documents” table * Custom style table formatting removed | Jbaden1 |
| 6 | 1a | 2020-01-07 | * Some fine tuning for naming conventions of E/E components and connections. * List of HW I/O signal types reduced to RF-A, RF-D, D, A, Networked and PWM. * Protocol column added to the E/E connection table | Jbaden1 |
| 6 | 1a | 2020-01-07 | * “HW Metric” and “Architecture Redundancy Summary” sections removed per request from the Functional Architecture Team (based on Governance Board decision [FSTGB-97](mailto:TrackLite%20%23%20FSTGB-97:%20https://www.tracklite.ford.com/prweb/PRAuth/TrackLiteSSO?pyActivity=@baseclass.RedirectAndRunWraper&ThreadName=WorkLinkThread&bPurgeTargetThread=true&AccessGroupName=FSTGB:ProjectAdministrators&Location=pyActivity%3DWork-.Open%26Action%3DReview%26HarnessPurpose%3DReview%26InsHandle%3DFORD-FSTGB-WORK+FSTGB-97)) * “Functional Safety” chapter moved to “Feature Implementation Requirements” section. “Function Allocation” chapter seemed no longer appropriate. | Jbaden1 |
| 6 | 1a | 2020-01-07 | * Ordering of fields in AIS interfaces tables modified to conform with the Macro Template and the Importer Sheet * Page Header: no longer in bold letters | Jbaden1 |
| 6 | 1a | 2020-03-09 | * Missing doc property “LatestSigMappingID” and “LatestAisInterfaceID” added * doc property “CopyrightDate” re-formatted to text and copyright date field in footer corrected * Version numbering re-initialized as 0.1 * Init value of version/revision date set to “yyyy/mm/dd” instead of “yyyy-mm-dd” to be in line with the “Edit Document Property” dialog * Type of “Latest….ID” doc properties changed from Text to Number | Jbaden1 |
| 6 | 1a | 2020-03-11 | * “Mapping” table removed from template. Has been migrated to macro. | Jbaden1 |
| 6 | 1a | 2020-03-13 | * Separate chapter “Technical Safety Requirements” removed. Content already covered by Allocation Table in chapter Function Allocation. * “Implemented Function” replaced by term “Technology Function” | Jbaden1 |

# Appendix

## Data Dictionary

### Logical Signals

**#Hint:** Logical Signals are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link**: [*RE Wiki – Adding a Logical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Logical+Signal+or+Parameter)

**#Macro**: Add Ins -> Add Requirement macro (select “Logical Signal” as type)

### Logical Parameters

**#Hint:** Logical Parameters are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link**: [*RE Wiki – Adding a Logical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Logical+Signal+or+Parameter)

**#Macro:** Add Ins -> Add Requirement macro (select “Logical Parameter” as type)

### Technical Signals

**#Hint:** This section lists all GSDB + GDT + SW signals relevant for the feature deployment.

**#Link**: [*RE Wiki – Adding a Technical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Signal+or+Parameter)

**#Macro:** Add Ins -> Add Requirement macro (select “Technical Signal” as type)

#### GSDB Signals

**#Hint:** This part of the Data Dictionary lists signals, which should go to the GSDB in VSEM, but do not exist in the GSDB in VSEM yet, but are or will be requested for the GSDB. Those would go temporarily to this section in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server) in VSEM.

#### HW I/Os

**#Hint:** This chapter lists signals, which will be mapped to hardwired I/Os. Those get typically refer to VSEM EDAS signals (or input/output signals of device transmittals in VSEM GDT).

#### Diagnostic Interfaces

**#Hint:** This chapter lists Diagnostic Interfaces (DTCs and DIDs), which get mapped to Logical Parameters in context of the Technology Functions in chapter “Parameters” of the Function Interfaces. Those DTC/DID names should match the names in the diagnostics specification (Part 2).

**#ToDo:** Currently the template below is just a proposal. A macro still needs to be created

##### DTCs

###<DTC\_<ID>>### <DTC Name>

<Some Description of the DTC.

Refer to VSEM document “[Diagnostic Fault Coverage and DTC Numbers](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=yAUtrNhnx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

[Design Consideration](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=yAUtrNhnx3NrTDAAAAAAAAAAAAA&servername=Production_Server)”, what to fill into the attributes below>

|  |  |
| --- | --- |
| **Test Period Time** |  |
| **Test Run Criteria,** |  |
| **Enable Criteria (EC)** |  |
| **Applicable** |  |
| **FailureTypeBytes** |  |
| **Test Period Time** |  |
| **Test Run Criteria,** |  |

##### DIDs

**#Hint**: This section lists diagnostic DID which Technical Parameters get mapped to.

**#Todo**: A proper template derived from the Part 2 spec still needs to be created.

### Technical Parameters

**#Hint:** This section lists all Method 2, Method 3 and calibration parameters relevant for the feature deployment.

**#Link**: [*RE Wiki – Adding a Technical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Signal+or+Parameter)

**#Macro:** [Add Ins -> Add Requirement macro](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#HowtousetheSpecificationTemplates-AddNewRequirement) (select “Technical Parameter” as type)

### Mappings

**#Hint**: This section lists mapping objects for Logical Signals / Parameters to their GSDB + GDT + SW counterparts (1:N mapping is supported). Mapping objects are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link:** [RE Wiki – Adding a Signal or Parameter Mapping](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

**#Macro:** Add Ins -> Add Requirement macro (select “Mapping” as type)

### Technical Interfaces

**#Hint:** This section lists port/interface details, which define how network/SW/HW signals are published / subscribed.

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

#### AIS Interfaces

**#Hint:** This chapter lists the AIS subscriber and publisher interface objects (managed in VSEM), which are needed to deploy the feature to the E/E architecture. If AIS interfaces do not yet exist in VSEM, those may temporarily be managed as a workaround in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link:** [System Engineering Portal – AIS Release 3.2](https://pd3.spt.ford.com/sites/fede/vsem-spls/Shared%20Documents/02-ais/methods/AIS%20Methods%20Document.pptx?web=1)  
[RE Wiki - AIS Interfaces](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface#AddingaTechnicalInterface-AisInterfaces)

[*Publisher Interface AIS in VSEM*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=zjYtY3Jcx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

[*Subscriber Interface AIS in VSEM*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=LSYtewY7x3NrTDAAAAAAAAAAAAA&servername=Production_Server)

**#Macro:** Add Ins -> Add Requirement macro (select “AIS Subscriber If” or “AIS Publisher If” as type)

##### Publisher Interfaces

##### Subscriber Interfaces

#### AUTOSAR Ports

**#Hint:** Those AUTOSAR Classic (provided and required) ports, which are used by the feature but are not managed in a central repository yet, could be listed here.

### Messages/APIs

#### CAN Bus “<Bus Name>”

**#Hint:** This section gives the relevant extract from the [Central Message Database (CMDB) in VSEM](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server) .

###<MSG\_MessageID### MessageName

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CAN ID** | **Transmission Mode** | **Period** | **Signal Names** | **Transmitter(s)** | **Receiver(s)** |
|  |  |  |  |  |  |
|  |
|  |
|  |

#### LIN Bus “<Bus Name>”

#### AUTOSAR Interfaces

**#Hint:** Those AUTOSAR Classic (Sender/Receiver and Client/Server) Interfaces, which are used by the feature but not managed in a central repository yet, should be listed here.

#### SOA Service Contracts

**#Hint:** This part of the Data Dictionary lists Service APIs/MQTT messages and embedded data elements, which are used for the Service Oriented Architecture (SOA). If those APIs/MQTT messages already exist e.g. in the [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog), simply add a reference to those yet.

Information on FNV2 SOA can be found in the ECG wiki page

* MQTT Topic Naming: [*FNV2-SOA: MQTT Topic and Message Structure*](https://www.eesewiki.ford.com/display/ecg/FNV2-SOA%3A+MQTT+Topic+and+Message+Structure?src=sidebar)
* message syntax and proper naming can be found [*SOA API Messaging Guidelines*](https://www.eesewiki.ford.com/x/Q7rKAg)

For examples what to fill into the table fields below refer to [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog)

###<ServiceContractID>### Service Contract Name

<Service contract purpose/behavior>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Messaging Pattern | Frequency  (For Data Broadcast Only) | Message Data Element(s)  (Must Match GPB) or applicable CAN signal | Description of Data Element(s) | Topic Name |
| Choose an item. |  | GBP Data element / CAN Signal name 1 | Detailed encoding of data element 1 |  |
| … |  |  |
| GBP Data element / CAN Signal name 1 | Detailed encoding of data element 3 |  |

### Encoding Types

**#Link:** [*RE Wiki – Adding Encoding Types*](http://wiki.ford.com/display/RequirementsEngineering/Adding+an+Encoding+Type)

**#Macro:** Add Ins -> Add Requirement macro (select “Encoding Type” as type)

Document ends here.