

SDL Integration Guidelines

Table of Contents

| | |
|--|----|
| Table of Contents | 2 |
| Introduction..... | 38 |
| Abbreviations and Definitions..... | 39 |
| | |
| I. SDL Programmer`s Guide | 41 |
| 1 Overview | 41 |
| 2 Architecture | 41 |
| 3 Transport Manager..... | 42 |
| 3.1 General description | 42 |
| 3.1.1 Transport layer features | 42 |
| 3.1.2 Transport Level Structure | 43 |
| 3.1.2.1 Transport Manager Structure..... | 44 |
| 3.1.2.2 Transport Adapter Structure | 45 |
| 3.1.3 Operation Examples (Message Sequence Chart)..... | 46 |
| 3.1.3.1 New device search..... | 46 |
| 3.1.3.2 Device-originating connection..... | 48 |
| 3.1.3.3 Connection close command..... | 49 |
| 3.2 Transport Manager (TM) Usage | 49 |
| 3.2.1 General Processing Description | 49 |
| 3.2.1.1 TM initialization | 49 |
| 3.2.1.2 TM structure | 50 |
| 3.2.1.3 TA initialization..... | 50 |
| 3.2.1.4 Getting started | 50 |
| 3.2.1.5 Errors in TM | 51 |
| 3.2.1.6 Messages in TM..... | 51 |
| 3.2.1.7 Connection identifiers | 51 |
| 3.2.1.8 Connection closing..... | 52 |
| 3.2.1.9 SDL shutting down..... | 52 |
| 3.2.2 Create Default TM Instance..... | 52 |
| 3.2.3 Add Custom Listener to TM..... | 52 |
| 3.2.4 Add Custom Transport Adapter to Default TM | 53 |
| 3.2.4.1 About TA in general | 53 |
| 3.2.4.2 Several instances of TA..... | 53 |
| 3.2.4.3 Add Custom TA..... | 53 |
| 3.2.4.4 Initialization | 54 |
| 3.2.5 Custom TA implementation from scratch | 54 |
| 3.2.6 Add a New Listener to TA..... | 55 |
| 3.2.7 Create TM with Custom TAs Only (with No Default Adapter)..... | 56 |
| 3.2.8 Transport Manager Customizing | 57 |
| 3.2.8.1 Basic information | 57 |

| | | |
|---------|--|----|
| 3.2.8.2 | Queues as a fundamental of TM..... | 57 |
| 3.2.8.3 | Rules for developer..... | 58 |
| 3.2.9 | Transport Adapter Customizing..... | 58 |
| 3.2.9.1 | Basic information | 58 |
| 3.2.9.2 | Workers of TA | 58 |
| 3.2.9.3 | Connection..... | 59 |
| 3.2.9.4 | Descriptor..... | 59 |
| 3.2.9.5 | Create custom Transport Adapter..... | 59 |
| 4 | Build and Run SDL and Set Up the Environment..... | 62 |
| 4.1 | General..... | 62 |
| 4.2 | Known issues | 62 |
| 4.3 | Preparation steps | 62 |
| 4.3.1 | To prepare the Linux Host..... | 62 |
| 4.3.2 | To set up QNX cross platform build for Linux..... | 63 |
| 4.3.3 | To run specially prepared QNX in virtual machine | 63 |
| 4.3.4 | To make QNX virtual machine from scratch..... | 63 |
| 4.3.5 | To set up Android Simulator for using instead of real device | 64 |
| 4.4 | Configuration steps | 64 |
| 4.4.1 | HTML5 HMI and SDL to be running on separate computers | 64 |
| 4.4.2 | QML HMI and SDL to be running on separate computers | 64 |
| 4.5 | Build SDL from source on Linux Host | 65 |
| 4.5.1 | QNX with HTML5 HMI | 65 |
| 4.5.2 | QNX with QML HMI | 65 |
| 4.5.3 | Linux with HTML5 HMI | 65 |
| 4.5.4 | Linux with QML HMI | 65 |
| 4.5.5 | Build SDL..... | 65 |
| 4.6 | Run SDL..... | 65 |
| 4.6.1 | Linux host for both SDL and HMI (Qt / HTML5) | 65 |
| 4.6.2 | QNX host for both SDL and Qt HMI | 66 |
| 4.6.3 | QNX host for both SDL and HTML5 HMI | 66 |
| 4.6.4 | Run SDL on QNX Server and QML HMI on Linux host..... | 66 |
| 5 | Audio/Video Streaming over SDL..... | 67 |
| 5.1 | PulseAudio and GStreamer libraries: how to include or exclude from project | 67 |
| 5.1.1 | Short overview..... | 67 |
| 5.1.2 | Include | 67 |
| 5.1.3 | Exclude..... | 67 |
| 5.2 | Use SyncProxyTester to check audio/video streaming over SDL | 67 |
| 5.3 | Configuring audio/video streaming parameters in smartDeviceLink.ini file | 67 |
| II | SDL-HMI Integration Guide | 69 |
| 1 | Overview | 69 |
| 1.1 | Operation System..... | 69 |

| | | |
|---------|--|-------------------------------------|
| 1.2 | Transports Supported..... | 69 |
| 1.3 | Communication | 69 |
| 2 | WebSocket Transport..... | 70 |
| 2.1 | Connection Opening..... | 70 |
| 2.1.1 | Requirements to HMI adapter | 70 |
| 2.1.2 | Handshake | 70 |
| 2.1.3 | Components registering | 70 |
| 2.1.4 | Examples..... | 71 |
| 2.2 | JSON Message Format | 73 |
| 2.2.1 | Request..... | 73 |
| 2.2.1.1 | Examples..... | 74 |
| 2.2.2 | Notification..... | 74 |
| 2.2.2.1 | Examples..... | 75 |
| 2.2.3 | Response | 75 |
| 2.2.3.1 | Examples..... | 76 |
| 2.2.4 | Error..... | 77 |
| 2.2.4.1 | Examples..... | 78 |
| 4 | D-Bus Transport | 79 |
| 4.1 | Overview | 79 |
| 4.2 | HMI Integration Aspects..... | 79 |
| 4.3 | Low Level D-Bus processing | 80 |
| 5 | Getting Started | 82 |
| 5.1 | Readiness confirming | 82 |
| 5.1.1 | Sequence diagram | 82 |
| 6 | BasicCommunication Component Description | 84 |
| 6.1 | UpdateDeviceList..... | 84 |
| 6.1.1 | Description..... | 84 |
| 6.1.2 | Request | 84 |
| 6.1.2.1 | Behavior | 84 |
| 6.1.2.2 | Parameters | 84 |
| 6.1.2.3 | DeviceInfo Structure | 85 |
| 6.1.3 | Response | 85 |
| 6.1.4 | Sequence Diagrams | 85 |
| 6.1.4.1 | UpdateDeviceList on SDL's Periodic Search Routine..... | 85 |
| 6.1.4.2 | UpdateDeviceList on User's Request | Error! Bookmark not defined. |
| 6.1.5 | JSON Messages Examples | 86 |
| 6.1.5.1 | Request | 86 |
| 6.1.5.2 | Response | 87 |
| 6.1.5.3 | Error message | 87 |
| 6.1.6 | D-Bus Messages Examples | 87 |
| 6.1.6.1 | Request | 87 |
| 6.1.6.2 | Response | 87 |
| 6.1.6.3 | Failure..... | 87 |

| | |
|---|------------------------------|
| 6.2 UpdateAppList | Error! Bookmark not defined. |
| 6.2.1 Description..... | Error! Bookmark not defined. |
| 6.2.2 Request | Error! Bookmark not defined. |
| 6.2.2.1 Behavior | Error! Bookmark not defined. |
| 6.2.2.2 Parameters | Error! Bookmark not defined. |
| 6.2.2.3 HMIApplication Structure..... | Error! Bookmark not defined. |
| 6.2.2.4 AppHMIType Enumeration | Error! Bookmark not defined. |
| 6.2.3 Response | Error! Bookmark not defined. |
| 6.2.4 Sequence Diagrams | Error! Bookmark not defined. |
| 6.2.4.1 UpdateAppList after application has just registered with SDL..... | Error! Bookmark not defined. |
| 6.2.4.2 UpdateDeviceList on User's Request | Error! Bookmark not defined. |
| 6.2.5 JSON Messages Examples..... | Error! Bookmark not defined. |
| 6.2.5.1 Request | Error! Bookmark not defined. |
| 6.2.5.2 Response | Error! Bookmark not defined. |
| 6.2.5.3 Error message | Error! Bookmark not defined. |
| 6.2.6 D-Bus Messages Examples | Error! Bookmark not defined. |
| 6.2.6.1 Request | Error! Bookmark not defined. |
| 6.2.6.2 Response | Error! Bookmark not defined. |
| 6.2.6.3 Failure..... | Error! Bookmark not defined. |
| 6.3 MixingAudioSupported | 95 |
| 6.3.1 Description..... | 95 |
| 6.3.2 Request | 96 |
| 6.3.2.1 Behavior | 96 |
| 6.3.3 Response | 96 |
| 6.3.3.1 Behavior | 96 |
| 6.3.3.2 Parameters | 96 |
| 6.3.4 Sequence Diagrams | 96 |
| 6.3.4.1 MixingAudioSupported Messaging | 96 |
| 6.3.5 JSON Messages Examples | 97 |
| 6.3.5.1 Request | 97 |
| 6.3.5.2 Response | 97 |
| 6.3.5.3 Error message | 97 |
| 6.3.6 D-Bus Messages Examples | 97 |
| 6.3.6.1 Request | 97 |
| 6.3.6.2 Response | 97 |
| 6.3.6.3 Failure..... | 97 |
| 6.4 AllowDeviceToConnect..... | 97 |
| 6.4.1 Description..... | 97 |
| 6.4.2 Request | 98 |
| 6.4.2.1 Behavior | 98 |
| 6.4.2.2 Parameters | 98 |
| 6.4.2.3 DeviceInfo Structure | 98 |
| 6.4.3 Response | 98 |

| | |
|--|-------------------------------------|
| 6.4.3.1 Behavior | 98 |
| 6.4.3.2 Parameters | 98 |
| 6.4.4 Sequence Diagrams | 99 |
| 6.4.4.1 AllowDeviceToConnect Messaging | 99 |
| 6.4.5 JSON Messages Examples | 99 |
| 6.4.5.1 Request | 99 |
| 6.4.5.2 Response | 99 |
| 6.4.5.3 Error message | 99 |
| 6.4.6 D-Bus Messages Examples | 100 |
| 6.4.6.1 Request | 100 |
| 6.4.6.2 Response | 100 |
| 6.4.6.3 Failure | 100 |
| 6.5 AllowAllApps | 100 |
| 6.5.1 Description | 100 |
| 6.5.2 Request | 100 |
| 6.5.3 Response | 101 |
| 6.5.3.1 Behavior | 100 |
| 6.5.3.2 Parameters | Error! Bookmark not defined. |
| 6.5.4 Sequence Diagrams | 101 |
| 6.5.4.1 AllowDeviceToConnect Messaging | 101 |
| 6.5.5 JSON Messages Examples | 101 |
| 6.5.5.1 Request | 101 |
| 6.5.5.2 Response | 101 |
| 6.5.5.3 Error message | 101 |
| 6.5.6 D-Bus Messages Examples | 102 |
| 6.5.6.1 Request | 102 |
| 6.5.6.2 Response | 102 |
| 6.5.6.3 Failure | 102 |
| 6.6 AllowApp | 102 |
| 6.6.1 Description | 102 |
| 6.6.2 Request | 102 |
| 6.6.2.1 Behavior | 102 |
| 6.6.2.2 Parameters | 102 |
| 6.6.2.3 HMIApplication Structure | Error! Bookmark not defined. |
| 6.6.2.4 AppHMIType Enumeration | Error! Bookmark not defined. |
| 6.6.3 Response | 103 |
| 6.6.3.1 Behavior | 103 |
| 6.6.3.2 Parameters | Error! Bookmark not defined. |
| 6.6.4 Sequence Diagrams | 103 |
| 6.6.4.1 AllowApp Messaging | 103 |
| 6.6.5 JSON Messages Examples | 103 |
| 6.6.5.1 Request | 103 |
| 6.6.5.2 Response | 103 |
| 6.6.5.3 Error message | 104 |

| | |
|---|-------------------------------------|
| 6.6.6 D-Bus Messages Examples | 104 |
| 6.6.6.1 Request | 104 |
| 6.6.6.2 Response | 104 |
| 6.6.6.3 Failure..... | 104 |
| 6.7 ActivateApp..... | 104 |
| 6.7.1 Description..... | 104 |
| 6.7.2 Request | 104 |
| 6.7.2.1 Behavior | 104 |
| 6.7.2.2 Parameters | Error! Bookmark not defined. |
| 6.7.3 Response | 104 |
| 6.7.4 Sequence Diagrams | 105 |
| 6.7.4.1 ActivateApp as a result of VR command recognition | 105 |
| 6.7.4.2 ActivateApp as a result of User's choice on UI for the application having been deactivated..... | 105 |
| 6.7.5 JSON Messages Examples | 106 |
| 6.7.5.1 Request | 106 |
| 6.7.5.2 Response | 106 |
| 6.7.5.3 Error message | 106 |
| 6.7.6 D-Bus Messages Examples | 106 |
| 6.7.6.1 Request | 106 |
| 6.7.6.2 Response | 106 |
| 6.7.6.3 Failure..... | 106 |
| 6.9 OnReady..... | 106 |
| 6.9.1 Description..... | 106 |
| 6.9.2 Sequence Diagrams | 107 |
| 6.9.2.1 OnReady after WebSocket connection establishment | 107 |
| 6.9.2.2 OnReady after D-Bus service publishing..... | 107 |
| 6.9.3 JSON Messages Examples | 107 |
| 6.9.4 D-Bus Messages Examples | 108 |
| 6.9.6.1 Request | 108 |
| 6.9.6.2 Response | 108 |
| 6.9.6.3 Failure..... | 108 |
| 6.10 OnStartDeviceDiscovery..... | 108 |
| 6.10.1 Description..... | 108 |
| 6.9.2 Sequence Diagrams | 108 |
| 6.9.2.1 OnStartDeviceDiscovery upon User's request and resulting UpdateDeviceList | 108 |
| 6.9.3 JSON Messages Examples | 109 |
| 6.9.4 D-Bus Messages Examples | 109 |
| 6.10 OnDevcieChosen..... | 109 |
| 6.10.1 Description..... | 109 |
| 6.10.1.1 Parameters | 110 |
| 6.10.1.2 DeviceInfo Structure | 110 |
| 6.10.2 Sequence Diagrams | 110 |
| 6.10.2.1 OnDevcieChosen with preceding UpdateDeviceList | 110 |

| | |
|--|-----|
| 6.10.3 JSON Messages Examples | 111 |
| 6.10.4 D-Bus Messages Examples | 111 |
| 6.9.9.1 Request | 111 |
| 6.9.6.2 Response | 111 |
| 6.9.6.3 Failure..... | 111 |
| 6.11 OnUpdateDeviceList..... | 134 |
| 6.11.1 Description..... | 134 |
| 6.11.2 Sequence Diagrams | 135 |
| 6.11.2.1 OnUpdateDeviceList upon User`'s request and resulting UpdateDeviceList | 135 |
| 6.11.3 JSON Messages Examples | 135 |
| 6.11.4 D-Bus Messages Examples | 135 |
| 6.12 OnFindApplications | 111 |
| 6.12.1 Description..... | 111 |
| 6.12.1.1 Parameters | 111 |
| 6.12.1.2 DeviceInfo Structure | 111 |
| 6.12.2 Sequence Diagrams | 112 |
| 6.12.2.1 OnFindApplications upon User`'s choice and resulting OnAppRegistered | 112 |
| 6.12.2.2 OnFindApplications upon User`'s choosing the device in the list of found devices and resulting OnAppRegistered | 112 |
| 6.12.3 JSON Messages Examples | 113 |
| 6.12.4 D-Bus Messages Examples | 114 |
| 6.9.9.1 Request | 114 |
| 6.9.6.2 Response | 114 |
| 6.9.6.3 Failure..... | 114 |
| 6.13 OnAppActivated | 114 |
| 6.13.1 Description..... | 114 |
| 6.13.1.1 Parameters | 114 |
| 6.13.2 Sequence Diagrams | 114 |
| 6.13.2.1 OnAppActivated upon User`'s choice and resulting ActivateApp..... | 114 |
| 6.13.3 JSON Messages Examples | 115 |
| 6.13.4 D-Bus Messages Examples | 115 |
| 6.9.9.1 Request | 115 |
| 6.9.6.2 Response | 116 |
| 6.9.6.3 Failure..... | 116 |
| 6.14 OnAppDeactivated | 116 |
| 6.14.1 Description..... | 116 |
| 6.14.1.1 Parameters | 116 |
| 6.14.1.2 DeactivateReason Enumeration..... | 116 |
| 6.14.2 Sequence Diagrams | 117 |
| 6.14.2.1 OnAppDeactivated Simple Messaging Example | 117 |
| 6.14.2.1 OnAppDeactivated with preceding registering, activation, adding UI/VR commands and resulting deleting UI/VR commands..... | 117 |
| 6.14.3 JSON Messages Examples | 118 |
| 6.14.4 D-Bus Messages Examples | 119 |

| | |
|---|-------------------------------------|
| 6.15 OnAppRegistered | 119 |
| 6.15.1 Description..... | 119 |
| 6.15.1.1 Parameters | 120 |
| 6.15.1.2 HMIApplication Structure..... | 121 |
| 6.15.1.3 AppHMIType Enumeration | 122 |
| 6.15.2 Sequence Diagrams | 122 |
| 6.15.2.1 OnAppRegistered on SDL`s search and application name as a VR synonym adding for voice recognition | 123 |
| 6.15.2.2 OnAppRegistered on User`s Request and application name as a VR synonym adding for voice recognition | Error! Bookmark not defined. |
| 6.15.3 JSON Messages Examples..... | 125 |
| 6.15.4 D-Bus Messages Examples | 125 |
| 6.16 OnAppUnregistered | 125 |
| 6.16.1 Description..... | 125 |
| 6.16.1.1 Parameters | 126 |
| 6.16.2 Sequence Diagrams | 126 |
| 6.16.2.1 OnAppUnregistered of active application | 126 |
| 6.16.3 JSON Messages Examples..... | 127 |
| 6.16.4 D-Bus Messages Examples | 127 |
| 6.17 OnExitApplication..... | 127 |
| 6.17.1 Description..... | 127 |
| 6.17.1.1 Parameters | 127 |
| 6.17.2 Sequence Diagrams | 128 |
| 6.17.2.1 OnExitApplication and the resulting OnAppUnregistered..... | 128 |
| 6.17.3 JSON Messages Examples..... | 128 |
| 6.17.4 D-Bus Messages Examples | 129 |
| 6.18 OnExitAllApplications | 129 |
| 6.18.1 Description..... | 129 |
| 6.18.1.1 Parameters | 129 |
| 6.18.1.2 ApplicationsCloseReason Enumeration | 129 |
| 6.18.2 Sequence Diagrams | 129 |
| 6.18.2.1 OnExitAllApplications | 129 |
| 6.18.3 JSON Messages Examples..... | 130 |
| 6.18.4 D-Bus Messages Examples | 130 |
| 6.19 PlayTone..... | 130 |
| 6.19.1 Description..... | 130 |
| 6.19.2 Sequence Diagrams | 130 |
| 6.19.2.1 PlayTone with Alert request..... | 130 |
| 6.19.3 JSON Messages Examples..... | 131 |
| 6.19.4 D-Bus Messages Examples | 131 |
| 6.20 OnSystemRequest..... | 131 |
| 6.20.1 Description..... | 131 |
| 6.20.1.1 Parameters | 131 |
| 6.20.1.2 RequestType Enumeration | 132 |

| | |
|--|------------|
| 6.20.1.3 FileType Enumeration | 132 |
| 6.20.2 Sequence Diagrams | 132 |
| 6.20.2.1 OnSystemRequest | 132 |
| 6.20.2.2 OnSystemRequest | 133 |
| 6.20.3 JSON Messages Examples | 133 |
| 6.15.4 D-Bus Messages Examples | 133 |
| 7 UI Component Description | 145 |
| 7.1 IsReady..... | 145 |
| 7.1.1 Description..... | 145 |
| 7.1.2 Request | 145 |
| 7.1.2.1 Behavior | 145 |
| 7.1.3 Response | 145 |
| 7.1.3.1 Parameters | 145 |
| 7.1.4 Sequence Diagrams | 145 |
| 7.1.4.1 UI.IsReady and preceding OnReady..... | 145 |
| 7.1.5 JSON Messages Examples | 146 |
| 7.1.5.1 Request | 146 |
| 7.1.5.2 Response | 146 |
| 7.1.5.3 Error message | 146 |
| 7.1.6 D-Bus Messages Examples | 146 |
| 7.1.6.1 Request | 146 |
| 7.1.6.2 Response | 147 |
| 7.1.6.3 Failure..... | 147 |
| 7.2 GetCapabilities..... | 147 |
| 7.2.1 Description..... | 147 |
| 7.2.2 Request | 147 |
| 7.2.2.1 Behavior | 147 |
| 7.2.3 Response | 147 |
| 7.2.3.1 Parameters | 148 |
| 7.2.3.2 DisplayCapabilities Structure..... | 148 |
| 7.2.3.3 DisplayType Enumeration | 149 |
| 7.2.3.4 TextFieldName Enumeration..... | 149 |
| 7.2.3.7 ImageField Structure | 150 |
| 7.2.3.4 ImageFieldName Enumeration..... | 151 |
| 7.2.3.4 FileType Enumeration | 151 |
| 7.2.3.7 ImageResolution Structure | 151 |
| 7.2.3.5 MediaClockFormat Enumeration | 151 |
| 7.2.3.6 ImageType Enumeration | 152 |
| 7.2.3.7 ScreenParams Structure | 153 |
| 7.2.3.7 TouchEventCapabilities Structure | 153 |
| 7.2.3.7 SoftButtonCapabilities Structure..... | 153 |
| 7.2.3.8 HmiZoneCapabilities Enumeration | 153 |
| 7.2.3.9 AudioPassThruCapabilities Structure..... | 153 |
| 7.2.3.10 SamplingRate Enumeration..... | 154 |

| | |
|--|-----|
| 7.2.3.11 BitsPerSample Enumeration | 154 |
| 7.2.3.12 AudioType Enumeration | 154 |
| 7.2.4 Sequence Diagrams | 154 |
| 7.2.4.1 UI.GetCapabilities and preceding IsReady..... | 154 |
| 7.2.5 JSON Messages Examples | 155 |
| 7.2.5.1 Request | 155 |
| 7.2.5.2 Response | 155 |
| 7.2.5.3 Error message | 156 |
| 7.2.6 D-Bus Messages Examples | 156 |
| 7.2.6.1 Request | 156 |
| 7.2.6.2 Response | 156 |
| 7.2.6.3 Failure..... | 156 |
| 7.3 GetSupportedLanguages..... | 156 |
| 7.3.1 Description..... | 156 |
| 7.3.2 Request | 157 |
| 7.3.2.1 Behavior | 157 |
| 7.3.3 Response | 157 |
| 7.3.3.1 Parameters | 157 |
| 7.3.3.2 Language Enumeration | 157 |
| 7.3.4 Sequence Diagrams | 158 |
| 7.1.4.1 GetSupportedLanguages with preceding IsReady | 158 |
| 7.3.5 JSON Messages Examples | 158 |
| 7.3.5.1 Request | 158 |
| 7.3.5.2 Response | 158 |
| 7.3.5.3 Error message | 159 |
| 7.3.6 D-Bus Messages Examples | 159 |
| 7.1.6.1 Request | 159 |
| 7.1.6.2 Response | 159 |
| 7.1.6.3 Failure..... | 159 |
| 7.4 Alert..... | 159 |
| 7.4.1 Description..... | 159 |
| 7.4.2 Request | 159 |
| 7.4.2.1 Behavior | 159 |
| 7.4.2.2 Parameters | 160 |
| 7.4.2.3 TextFieldStruct Structure | 161 |
| 7.4.2.4 TextFieldName Enumeration | 161 |
| 7.4.3 Response | 162 |
| 7.4.3.1 Parameters | 162 |
| 7.4.4 Sequence Diagrams | 163 |
| 7.4.4.1 Alert together with PlayTone and Speak RPCs and closed by the timeout..... | 163 |
| 7.4.4.2 Alert displayed and closed by DEFAULT_ACTION soft button press | 164 |
| 7.4.4.3 Alert for non-activa application displayed and closed by STEAL_FOCUS soft button press | 165 |
| 7.4.4.4 Alert is displayed and then aborted by VR session started | 166 |
| 7.4.4.5 Alert is rejected because of RPC of a higher priority currently presented on UI | 166 |

| | |
|---|-----|
| 7.4.5 Possible Layout | 167 |
| 7.4.5.1 Alert dialog with text fields and NO soft buttons | 167 |
| 7.4.5.2 Alert dialog with text fields, one soft button and progress indicator..... | 167 |
| 7.4.5.3 Alert dialog with text fields and four soft buttons | 167 |
| 7.4.6 JSON Messages Examples | 168 |
| 7.4.6.1 Request | 168 |
| 7.4.6.2 Response | 168 |
| 7.4.6.3 Error message | 168 |
| 7.4.7 D-Bus Messages Examples | 169 |
| 7.4.7.1 Request | 169 |
| 7.4.7.2 Response | 169 |
| 7.4.7.3 Failure..... | 169 |
| 7.5 Show | 169 |
| 7.5.1 Description..... | 169 |
| 7.5.2 Request | 169 |
| 7.5.2.1 Behavior | 169 |
| 7.5.2.2 Parameters | 170 |
| 7.5.2.3 TextFieldStruct Structure | 171 |
| 7.5.2.4 TextFieldName Enumeration..... | 171 |
| 7.5.2.5 TextAlign Enumeration | 171 |
| 7.5.3 Response | 171 |
| 7.5.4 Sequence Diagrams | 172 |
| 7.5.4.1 Show for the active application that is then deactivated and activated again..... | 172 |
| 7.5.4.2 Show for the application currently not active on HMI | 173 |
| 7.5.4.3 Requested with Show soft button press | 175 |
| 7.5.4.4 Show text fields expected behavior | 176 |
| 7.5.5 Possible Layout | 177 |
| 7.5.5.1 Persistent display areas updated with Show for media applications | 177 |
| 7.5.5.2 Persistent display areas updated with Show for non-media applications | 177 |
| 7.5.6 JSON Messages Examples | 177 |
| 7.5.6.1 Request | 177 |
| 7.5.6.2 Response | 178 |
| 7.5.6.3 Error message | 178 |
| 7.5.6 D-Bus Messages Examples | 179 |
| 7.1.6.1 Request | 179 |
| 7.1.6.2 Response | 179 |
| 7.1.6.3 Failure..... | 179 |
| 7.8 AddCommand | 179 |
| 7.8.1 Description..... | 179 |
| 7.8.2 Request | 179 |
| 7.8.2.1 Behavior | 179 |
| 7.8.2.2 Parameters | 180 |
| 7.8.2.3 MenuParams | 180 |
| 7.8.3 Response | 181 |

| | |
|--|-----|
| 7.8.4 Sequence Diagrams | 181 |
| 7.8.4.1 AddCommand for the active application on HMI, the command is chosen by the User..... | 182 |
| 7.8.4.2 AddCommand for the application not active on HMI | 183 |
| 7.8.4.3 AddCommand: adding command to sub menu | 184 |
| 7.8.4.4 AddCommand: expected behavior of adding commands depending on position parameter | 185 |
| 7.8.4.5 AddCommand rejected because of the limit of menu items exhausted..... | 186 |
| 7.8.5 Possible Layout | 186 |
| 7.8.5.1 Application main menu with sub menus and commands..... | 186 |
| 7.8.5.2 Application sub menu with commands | 187 |
| 7.8.6 JSON Messages Examples | 187 |
| 7.8.6.1 Request | 187 |
| 7.8.6.2 Response | 187 |
| 7.8.6.3 Error message | 187 |
| 7.8.7 D-Bus Messages Examples | 188 |
| 7.8.7.1 Request | 188 |
| 7.8.7.2 Response | 188 |
| 7.8.7.3 Failure..... | 188 |
| 7.9 DeleteCommand | 188 |
| 7.9.1 Description..... | 188 |
| 7.9.2 Request | 188 |
| 7.9.2.1 Behavior | 188 |
| 7.9.2.2 Parameters | 189 |
| 7.9.3 Response | 189 |
| 7.9.4 Sequence Diagrams | 189 |
| 7.9.4.1 DeleteCommand, for the application active on HMI, for the command from the menu currently open on UI and preceding AddCommand..... | 189 |
| 7.9.4.2 DeleteCommand for the application not active on HMI | 190 |
| 7.9.5 JSON Messages Examples | 192 |
| 7.9.5.1 Request | 192 |
| 7.9.5.2 Response | 192 |
| 7.9.5.3 Error message | 192 |
| 7.9.6 D-Bus Messages Examples | 192 |
| 7.9.6.1 Request | 192 |
| 7.9.6.2 Response | 192 |
| 7.9.6.3 Failure..... | 192 |
| 7.10 AddSubMenu | 192 |
| 7.10.1 Description..... | 192 |
| 7.10.2 Request | 193 |
| 7.10.2.1 Behavior | 193 |
| 7.10.2.2 Parameters | 193 |
| 7.10.2.3 MenuParams | 193 |
| 7.10.3 Response | 194 |
| 7.10.4 Sequence Diagrams | 194 |
| 7.10.4.1 AddSubMenu for the application active on HMI | 194 |

| | |
|--|------------|
| 7.10.4.2 AddSubMenu for the application not active on HMI | 195 |
| 7.10.4.3 AddSubMenu: expected behavior of adding sub menus depending on position parameter | 196 |
| 7.10.4.4 AddSubMenu rejected because of the limit of menu items exhausted..... | 197 |
| 7.10.5 Possible Layout | 198 |
| 7.10.5.1 Application main menu with sub menus and commands..... | 198 |
| 7.10.6 JSON Messages Examples..... | 198 |
| 7.10.6.1 Request | 198 |
| 7.10.6.2 Response | 199 |
| 7.10.6.3 Error message | 199 |
| 7.10.7 D-Bus Messages Examples | 199 |
| 7.10.7.1 Request | 199 |
| 7.10.7.2 Response | 199 |
| 7.10.7.3 Failure..... | 199 |
| 7.11 DeleteSubMenu | 199 |
| 7.11.1 Description..... | 199 |
| 7.11.2 Request | 199 |
| 7.11.2.1 Behavior | 199 |
| 7.11.2.2 Parameters | 200 |
| 7.11.3 Response | 200 |
| 7.11.4 Sequence Diagrams | 201 |
| 7.11.4.1 DeleteSubMenu: the application is active on HMI, the msin menu is currently open on UI, AddSubMenu precedes | 201 |
| 7.11.4.2 DeleteCommand for the application not active on HMI | 201 |
| 7.11.5 JSON Messages Examples | 203 |
| 7.11.5.1 Request | 203 |
| 7.11.5.2 Response | 203 |
| 7.11.5.3 Error message | 203 |
| 7.11.6 D-Bus Messages Examples | 203 |
| 7.11.6.1 Request | 203 |
| 7.11.6.2 Response | 203 |
| 7.11.6.3 Failure..... | 203 |
| 7.12 PerformInteraction | 203 |
| 7.12.1 Description..... | 203 |
| 7.12.2 Request | 204 |
| 7.12.2.1 Behavior | 204 |
| 7.12.2.2 Parameters | 207 |
| 7.4.2.3 TextFieldStruct Structure..... | 208 |
| 7.12.2.4 TextFieldName | 208 |
| 7.12.2.5 Choice | 208 |
| 7.12.2.6 VrHelpItem..... | 209 |
| 7.12.2.7 LayoutMode | 209 |
| 7.12.3 Response | 209 |
| 7.12.4 Parameters | 209 |
| 7.12.4 Sequence Diagrams | 210 |

| | |
|--|------------|
| 7.12.4.1 UI.PerformInteraction followed with TTS.Performinteraction and TTS.Speak and User's choice before the timeout..... | 210 |
| 7.12.5 JSON Messages Examples | 212 |
| 7.12.5.1 Request | 212 |
| 7.12.5.2 Response | 213 |
| 7.12.5.3 Error message | 214 |
| 7.12.6 D-Bus Messages Examples | 214 |
| 7.12.6.1 Request | 214 |
| 7.12.6.2 Response | 214 |
| 7.12.6.3 Failure..... | 214 |
| 7.13 SetMediaClockTimer..... | 214 |
| 7.13.1 Description..... | 214 |
| 7.13.2 Request | 214 |
| 7.13.2.1 Behavior | 214 |
| 7.13.2.2 Parameters | 215 |
| 7.13.2.3 TimeFormat | 216 |
| 7.13.2.4 ClockUpdateMode | 216 |
| 7.13.3 Response | 216 |
| 7.13.4 Sequence Diagrams | 217 |
| 7.13.4.1 SetMediaClockTimer of COUNTUP and COUNTDOWN modes for the cases of active and background applications..... | 218 |
| 7.13.4.2 SetMediaClockTimer of PAUSE and RESUME for the application active on HMI | 219 |
| 7.13.4.3 SetMediaClockTimer of COUNTDOWN for the active application that is then deactivated and activated again and SetMediaClockTimer of CLEAR for thw active application..... | 220 |
| 7.13.5 JSON Messages Examples | 220 |
| 7.13.5.1 Request | 220 |
| 7.13.5.2 Response | 221 |
| 7.13.5.3 Error message | 221 |
| 7.13.6 D-Bus Messages Examples | 221 |
| 7.13.6.1 Request | 221 |
| 7.13.6.2 Response | 221 |
| 7.13.6.3 Failure..... | 221 |
| 7.14 SetGlobalProperties..... | 222 |
| 7.14.1 Description..... | 222 |
| 7.14.2 Request | 222 |
| 7.14.2.1 Behavior | 222 |
| 7.14.2.2 Parameters | 222 |
| 7.14.2.6 VrHelpItem..... | 223 |
| 7.14.2.4 KeyboardProperties..... | 223 |
| 7.14.2.5 KeyboardLayout | 224 |
| 7.14.3 Response | 224 |
| 7.14.4 Sequence Diagrams | 224 |
| 7.14.4.1 SetGlobalProperties for the application active on HMI and the later VR activation | 224 |
| 7.14.5 JSON Messages Examples | 225 |
| 7.14.5.1 Request | 225 |

| | |
|--------------------------------------|-----|
| 7.14.5.2 Response | 226 |
| 7.14.5.3 Error message | 226 |
| 7.14.6 D-Bus Messages Examples | 227 |
| 7.14.6.1 Request | 227 |
| 7.14.6.2 Response | 228 |
| 7.14.6.3 Failure..... | 228 |
| 7.15 ChangeRegistration | 228 |
| 7.15.1 Description..... | 228 |
| 7.15.2 Request | 228 |
| 7.15.2.1 Behavior | 228 |
| 7.15.2.2 Parameters | 229 |
| 7.15.2.3 Language..... | 229 |
| 7.15.3 Response | 230 |
| 7.15.4 Sequence Diagrams | 230 |
| 7.15.4.1 ChangeRegistration..... | 230 |
| 7.15.5 JSON Messages Examples | 230 |
| 7.15.5.1 Request | 230 |
| 7.15.5.2 Response | 230 |
| 7.15.5.3 Error message | 231 |
| 7.15.6 D-Bus Messages Examples | 231 |
| 7.15.6.1 Request | 231 |
| 7.15.6.2 Response | 231 |
| 7.15.6.3 Failure..... | 231 |
| 7.16 GetLanguage | 231 |
| 7.16.1 Description..... | 231 |
| 7.16.2 Request | 232 |
| 7.16.2.1 Behavior | 232 |
| 7.16.3 Response | 232 |
| 7.16.3.1 Parameters..... | 232 |
| 7.16.3.2 Language..... | 233 |
| 7.16.4 Sequence Diagrams | 233 |
| 7.16.4.1 GetLanguage..... | 233 |
| 7.16.5 JSON Messages Examples | 234 |
| 7.16.5.1 Request | 234 |
| 7.16.5.2 Response | 234 |
| 7.16.5.3 Error message | 234 |
| 7.16.6 D-Bus Messages Examples | 235 |
| 7.16.6.1 Request | 235 |
| 7.16.6.2 Response | 235 |
| 7.16.6.3 Failure..... | 235 |
| 7.17 SetAppIcon | 235 |
| 7.17.1 Description..... | 235 |
| 7.17.2 Request | 235 |
| 7.17.2.1 Behavior | 235 |

| | |
|--|-----|
| 7.17.2.2 Parameters | 236 |
| 7.17.2.3 Image..... | 236 |
| 7.17.2.4 ImageType..... | 236 |
| 7.17.3 Response | 236 |
| 7.17.4 Sequence Diagrams | 237 |
| 7.17.4.1 SetAppIcon | 237 |
| 7.17.5 Possible Layout | 237 |
| 7.17.5.1 The list of registered applications with and without icons..... | 237 |
| 7.17.6 JSON Messages Examples | 238 |
| 7.17.6.1 Request | 238 |
| 7.17.6.2 Response | 238 |
| 7.17.6.3 Error message | 238 |
| 7.17.7 D-Bus Messages Examples | 238 |
| 7.17.7.1 Request | 238 |
| 7.17.7.2 Response | 239 |
| 7.17.7.3 Failure..... | 239 |
| 7.18 Slider..... | 239 |
| 7.18.1 Description..... | 239 |
| 7.18.2 Request | 239 |
| 7.18.2.1 Behavior | 239 |
| 7.18.2.2 Parameters | 240 |
| 7.18.3 Response | 240 |
| 7.18.3.1 Parameters | 241 |
| 7.18.4 Sequence Diagrams | 241 |
| 7.18.4.1 Slider with static footer closed by the timeout | 241 |
| 7.18.4.2 Slider with dynamic footer aborted | 242 |
| 7.18.5 Possible Layout | 243 |
| 7.18.5.1 Slider | 243 |
| 7.18.5.2 Slider with dynamic footer | 244 |
| 7.18.5 JSON Messages Examples | 244 |
| 7.18.5.1 Request | 244 |
| 7.18.5.2 Response | 244 |
| 7.18.5.3 Error message | 244 |
| 7.18.6 D-Bus Messages Examples | 245 |
| 7.18.6.1 Request | 245 |
| 7.18.6.2 Response | 245 |
| 7.18.6.3 Failure..... | 246 |
| 7.19 ScrollableMessage..... | 246 |
| 7.19.1 Description..... | 246 |
| 7.19.2 Request | 246 |
| 7.19.2.1 Behavior | 246 |
| 7.19.2.2 Parameters | 247 |
| 7.19.2.3 TextFieldStruct Structure | 247 |
| 7.19.2.4 TextFieldName Enumeration..... | 247 |

| | |
|---|-----|
| 7.19.3 Response | 247 |
| 7.19.4 Sequence Diagrams | 248 |
| 7.19.4.1 ScrollableMessage displayed, scrolled by the User and closed by the timeout | 248 |
| 7.19.4.2 ScrollableMessage with soft buttons of DEFAULT_ACTION and KEEP_CONTEXT system action pressed by the User | 249 |
| 7.19.4.3 ScrollableMassage with STEAL_FOCUS soft button for the application not active on HMI..... | 250 |
| 7.19.5 JSON Messages Examples | 251 |
| 7.19.5.1 Request | 251 |
| 7.19.5.2 Response | 252 |
| 7.19.5.3 Error message | 252 |
| 7.19.6 D-Bus Messages Examples | 253 |
| 7.19.6.1 Request | 253 |
| 7.19.6.2 Response | 254 |
| 7.19.6.3 Failure..... | 254 |
| 7.20 PerformAudioPassThru | 255 |
| 7.20.1 Description..... | 255 |
| 7.20.2 Request | 255 |
| 7.20.2.1 Behavior | 255 |
| 7.20.2.2 Parameters | 255 |
| 7.20.2.3 TextFieldStruct Structure..... | 256 |
| 7.20.2.4 TextFieldName Enumeration..... | 256 |
| 7.20.3 Response | 256 |
| 7.20.4 Sequence Diagrams | 257 |
| 7.20.4.1 PerformAudioPassTru requested together with TTS.Speak, processed and then finished by 'Retry' soft button press..... | 258 |
| 7.20.4.2 PerformAudioPassTru for the application not active on HMI, processed and then finished by EndAudioPassThru..... | 259 |
| 7.20.5 JSON Messages Examples | 259 |
| 7.20.5.1 Request | 259 |
| 7.20.5.2 Response | 260 |
| 7.20.5.3 Error message | 260 |
| 7.20.6 D-Bus Messages Examples | 260 |
| 7.20.6.1 Request | 260 |
| 7.20.6.2 Response | 260 |
| 7.20.6.3 Failure..... | 261 |
| 7.21 EndAudioPassThru | 261 |
| 7.21.1 Description..... | 261 |
| 7.21.2 Request | 261 |
| 7.21.2.1 Behavior | 261 |
| 7.21.3 Response | 261 |
| 7.21.4 Sequence Diagrams | 262 |
| 7.21.4.1 EndAudioPassThru that stops the audio capturing | 262 |
| 7.21.4.2 EndAudioPassThru for the case when audio capturing is already over | 262 |
| 7.21.5 JSON Messages Examples | 263 |

| | |
|--|-----|
| 7.21.5.1 Request | 263 |
| 7.21.5.2 Response | 263 |
| 7.21.5.3 Error message | 264 |
| 7.21.6 D-Bus Messages Examples | 264 |
| 7.21.6.1 Request | 264 |
| 7.21.6.2 Response | 264 |
| 7.21.6.3 Failure | 264 |
| 7.22 ClosePopUp | 264 |
| 7.22.1 Description | 264 |
| 7.22.2 Request | 265 |
| 7.22.2.1 Behavior | 265 |
| 7.22.2.2 Parameters | 265 |
| 7.22.3 Response | 265 |
| 7.22.4 Sequence Diagrams | 265 |
| 7.22.4.1 ClosePopUp for UI.PerformIntercation | 265 |
| 7.22.5 JSON Messages Examples | 267 |
| 7.22.5.1 Request | 267 |
| 7.22.5.2 Response | 267 |
| 7.22.5.3 Error message | 267 |
| 7.22.6 D-Bus Messages Examples | 267 |
| 7.22.6.1 Request | 267 |
| 7.22.6.2 Response | 267 |
| 7.22.6.3 Failure | 267 |
| 7.23 OnCommand | 268 |
| 7.23.1 Description | 268 |
| 7.23.1.1 Parameters | 268 |
| 7.23.2 Sequence Diagrams | 269 |
| 7.23.2.1 OnCommand | 269 |
| 7.23.3 JSON Messages Examples | 269 |
| 7.23.4 D-Bus Messages Examples | 269 |
| 7.24 OnSystemContext | 270 |
| 7.24.1 Description | 270 |
| 7.24.1.1 Parameters | 270 |
| 7.24.1.2 SystemContext | 270 |
| 7.24.2 Sequence Diagrams | 271 |
| 7.24.2.1 OnSystemContext on different HMI states | 271 |
| 7.24.3 JSON Messages Examples | 272 |
| 7.24.4 D-Bus Messages Examples | 272 |
| 7.25 OnLanguageChange | 272 |
| 7.25.1 Description | 272 |
| 7.25.1.1 Parameters | 272 |
| 7.25.1.2 Language | 272 |
| 7.25.2 Sequence Diagrams | 273 |

| | |
|---|-------------------------------------|
| 7.25.2.1 OnLanguageChange | 273 |
| 7.25.3 JSON Messages Examples | 274 |
| 7.25.4 D-Bus Messages Examples | 274 |
| 7.26 OnKeyboardInput..... | 275 |
| 7.26.1 Description..... | 275 |
| 7.26.1.1 Parameters | 275 |
| 7.26.1.2 KeyboardEvent | 275 |
| 7.26.2 Sequence Diagrams | 277 |
| 7.26.2.1 OnKeyboardInput for SINGLE_KEYPRESS keypressMode | 277 |
| 7.26.2.2 OnKeyboardInput for QUEUE_KEYPRESSES keypressMode | 278 |
| 7.26.2.3 OnKeyboardInput for RESEND_CURRENT_ENTRY keypressMode | 279 |
| 7.26.2.3 OnKeyboardInput for cancelled entry | 280 |
| 7.26.3 JSON Messages Examples | 280 |
| 7.26.4 D-Bus Messages Examples | 280 |
| 7.27 OnTouchEvent..... | 281 |
| 7.27.1 Description..... | 281 |
| 7.27.1.1 Parameters | 281 |
| 7.27.1.2 TouchEvent | 281 |
| 7.27.1.2 TouchCoord | 282 |
| 7.27.1.2 TouchType | 282 |
| 7.27.2 Sequence Diagrams | 282 |
| 7.27.2.1 OnTouchEvent..... | 282 |
| 7.27.3 JSON Messages Examples | 282 |
| 7.27.4 D-Bus Messages Examples | 282 |
| 7.28 OnResetTimeout..... | 282 |
| 7.28.1 Description..... | 282 |
| 7.28.1.1 Parameters | 283 |
| 7.28.2 Sequence Diagrams | 283 |
| 7.28.2.1 OnResetTimeout upon keypress during PerformInteraction (KEYBOARD) | 283 |
| 7.28.2.2 OnResetTimeout upon User's scrolling the message during ScrollableMessage..... | 283 |
| 7.28.2.2 OnResetTimeout upon KEEP_CONTEXT soft button press during Alert..... | 284 |
| 7.28.3 JSON Messages Examples | 285 |
| 7.28.4 D-Bus Messages Examples | 285 |
| 7.30 OnDriverDistraction..... | 285 |
| 7.30.1 Description..... | 285 |
| 7.30.1.1 Parameters | 286 |
| 7.30.1.2 DriverDistractionState..... | 286 |
| 7.30.2 Sequence Diagrams | 286 |
| 7.30.2.1 OnDriverDistraction notification | 286 |
| 7.30.3 JSON Messages Examples | 286 |
| 7.30.4 D-Bus Messages Examples | 286 |
| 7.31 ShowNotification | Error! Bookmark not defined. |
| 7.31.1 Description..... | Error! Bookmark not defined. |

| | |
|---|------------------------------|
| 7.31.1.1 Parameters | Error! Bookmark not defined. |
| 7.20.2.3 TextFieldStruct Structure..... | Error! Bookmark not defined. |
| 7.31.1.1 TextFieldName | Error! Bookmark not defined. |
| 7.17.2.3 Image..... | Error! Bookmark not defined. |
| 7.17.2.4 ImageType..... | Error! Bookmark not defined. |
| 7.31.2 Sequence Diagrams | Error! Bookmark not defined. |
| 7.31.2.1 ShowNotification displayed and then closed by ClosePopUp | Error! Bookmark not defined. |
| 7.31.3 JSON Messages Examples..... | Error! Bookmark not defined. |
| 7.31.4 D-Bus Messages Examples | Error! Bookmark not defined. |
| 7.32 SetDisplayLayout..... | 287 |
| 7.32.1 Description..... | 287 |
| 7.32.2 Request | 288 |
| 7.32.2.1 Behavior | 288 |
| 7.32.2.2 Parameters | 288 |
| 7.32.3 Response | 288 |
| 7.32.3.1 Parameters | 289 |
| 7.21.4 Sequence Diagrams | 297 |
| 7.21.4.1 EndAudioPassThru that stops the audio capturing | 297 |
| 7.21.4.2 EndAudioPassThru for the case when audio capturing is already over | 298 |
| 7.21.5 JSON Messages Examples..... | 299 |
| 7.21.5.1 Request | 299 |
| 7.21.5.2 Response | 300 |
| 7.21.5.3 Error message | 302 |
| 7.21.6 D-Bus Messages Examples | 302 |
| 7.21.6.1 Request | 302 |
| 7.21.6.2 Response | 302 |
| 7.21.6.3 Failure..... | 302 |
| 8 Buttons Component Description | 302 |
| 8.1 GetCapabilities..... | 302 |
| 8.1.1 Description..... | 302 |
| 8.1.2 Request | 303 |
| 8.1.2.1 Behavior | 303 |
| 8.1.3 Response | 303 |
| 8.1.3.1 Parameters | 303 |
| 8.1.3.2 ButtonCapabilities..... | 304 |
| 8.1.3.3 ButtonName..... | 304 |
| 8.1.3.4 PresetBankCapabilities | 305 |
| 8.1.4 Sequence Diagrams | 305 |
| 8.1.4.1 GetCapabilities on system startup | 305 |
| 8.1.5 JSON Messages Examples..... | 305 |
| 8.1.5.1 Request | 305 |
| 8.1.5.2 Response | 305 |
| 8.1.5.3 Error message | 306 |

| | |
|---|-----|
| 8.1.6 D-Bus Messages Examples | 306 |
| 8.1.6.1 Request | 306 |
| 8.1.6.2 Response | 306 |
| 8.1.6.3 Failure..... | 306 |
| 8.2 OnButtonEvent..... | 306 |
| 8.2.1 Description..... | 306 |
| 8.2.1.1 Parameters | 307 |
| 8.1.3.3 ButtonName..... | 307 |
| 8.2.1.2 ButtonEventMode | 308 |
| 8.2.2 Sequence Diagrams | 308 |
| 8.2.2.1 OnButtonEvent for CUSTOM_BUTTON being pressed and released | 308 |
| 8.2.2.1 OnButtonEvent for hardware button being pressed and released..... | 308 |
| 8.2.3 JSON Messages Examples | 309 |
| 8.2.4 D-Bus Messages Examples | 309 |
| 8.3 OnButtonPress..... | 309 |
| 8.3.1 Description..... | 309 |
| 8.3.1.1 Parameters | 310 |
| 8.1.3.3 ButtonName..... | 310 |
| 8.3.1.2 ButtonPressMode | 311 |
| 8.3.2 Sequence Diagrams | 311 |
| 8.3.2.1 OnButtonPress (SHORT) for CUSTOM_BUTTON..... | 311 |
| 8.3.2.2 OnButtonPress (LONG) for hardware button | 312 |
| 8.3.2.2 OnButtonPress for hardware button that supports SHORT press mode only | 312 |
| 8.3.3 JSON Messages Examples | 313 |
| 8.3.4 D-Bus Messages Examples | 313 |
| 9 VR Component Description..... | 314 |
| 9.1 IsReady..... | 314 |
| 9.1.1 Description..... | 314 |
| 9.1.2 Request | 314 |
| 7.1.2.1 Behavior | 314 |
| 9.1.3 Response | 314 |
| 9.1.3.1 Parameters | 314 |
| 9.1.4 Sequence Diagrams | 314 |
| 9.1.4.1 VR.IsReady | 314 |
| 9.1.5 JSON Messages Examples | 315 |
| 9.1.5.1 Request | 315 |
| 9.1.5.2 Response | 315 |
| 9.1.5.3 Error message | 315 |
| 9.1.6 D-Bus Messages Examples | 315 |
| 9.1.6.1 Request | 315 |
| 9.1.6.2 Response | 316 |
| 9.1.6.3 Failure..... | 316 |
| 9.2 GetCapabilities..... | 316 |

| | |
|--|-----|
| 9.2.1 Description..... | 316 |
| 9.2.2 Request | 316 |
| 9.2.2.1 Behavior | 316 |
| 9.2.3 Response | 316 |
| 9.2.3.1 Parameters..... | 316 |
| 9.2.3.2 VrCapabilities | 317 |
| 9.2.4 Sequence Diagrams | 317 |
| 9.2.4.1 VR.GetCapabilities..... | 317 |
| 9.2.5 JSON Messages Examples | 317 |
| 9.2.5.1 Request | 317 |
| 9.2.5.2 Response | 317 |
| 9.2.5.3 Error message | 318 |
| 9.2.6 D-Bus Messages Examples | 318 |
| 9.2.6.1 Request | 318 |
| 9.2.6.2 Response | 318 |
| 9.2.6.3 Failure..... | 318 |
| 9.3 GetSupportedLanguages..... | 318 |
| 9.3.1 Description..... | 318 |
| 9.3.2 Request | 318 |
| 7.1.2.1 Behavior | 318 |
| 9.3.3 Response | 319 |
| 9.3.3.1 Parameters..... | 319 |
| 9.3.3.2 Language Enumeration | 319 |
| 9.3.4 Sequence Diagrams | 320 |
| 9.3.4.1 VR.GetSupportedLanguages | 320 |
| 9.3.5 JSON Messages Examples | 320 |
| 9.3.5.1 Request | 320 |
| 9.3.5.2 Response | 320 |
| 9.3.5.3 Error message | 320 |
| 9.3.6 D-Bus Messages Examples | 321 |
| 9.3.6.1 Request | 321 |
| 9.3.6.2 Response | 321 |
| 9.3.6.3 Failure..... | 321 |
| 9.4 AddCommand | 321 |
| 9.4.1 Description..... | 321 |
| 9.4.2 Request | 321 |
| 9.4.2.1 Behavior | 321 |
| 9.4.2.2 Parameters..... | 321 |
| 9.4.3 Response | 322 |
| 9.4.4 Sequence Diagrams | 322 |
| 9.4.4.1 VR.AddCommand..... | 322 |
| 9.4.5 JSON Messages Examples | 323 |
| 9.4.5.1 Request | 323 |

| | |
|-------------------------------------|-----|
| 9.4.5.2 Response | 324 |
| 9.4.5.3 Error message | 324 |
| 9.4.6 D-Bus Messages Examples | 324 |
| 9.4.6.1 Request | 324 |
| 9.4.6.2 Response | 324 |
| 9.4.6.3 Failure..... | 324 |
| 9.5 DeleteCommand | 324 |
| 9.5.1 Description..... | 324 |
| 9.5.2 Request | 324 |
| 9.5.2.1 Behavior | 324 |
| 9.5.2.2 Parameters | 325 |
| 9.5.3 Response | 325 |
| 9.5.4 Sequence Diagrams | 325 |
| 9.1.4.1 VR.DeleteCommand..... | 325 |
| 9.5.5 JSON Messages Examples | 326 |
| 9.5.5.1 Request | 326 |
| 9.5.5.2 Response | 326 |
| 9.5.5.3 Error message | 327 |
| 9.5.6 D-Bus Messages Examples | 327 |
| 9.5.6.1 Request | 327 |
| 9.5.6.2 Response | 327 |
| 9.5.6.3 Failure..... | 327 |
| 9.6 ChangeRegistration | 327 |
| 9.6.1 Description..... | 327 |
| 9.6.2 Request | 327 |
| 9.6.2.1 Behavior | 327 |
| 9.6.2.2 Parameters | 328 |
| 9.6.2.2 Language..... | 328 |
| 9.6.3 Response | 328 |
| 9.6.4 Sequence Diagrams | 329 |
| 9.6.4.1 VR.ChangeRegistration..... | 329 |
| 9.6.5 JSON Messages Examples | 330 |
| 9.6.5.1 Request | 330 |
| 9.6.5.2 Response | 330 |
| 9.6.5.3 Error message | 331 |
| 9.6.6 D-Bus Messages Examples | 331 |
| 9.6.6.1 Request | 331 |
| 9.6.6.2 Response | 331 |
| 9.6.6.3 Failure..... | 331 |
| 9.7 GetLanguage | 331 |
| 9.7.1 Description..... | 331 |
| 9.7.2 Request | 331 |
| 9.7.2.1 Behavior | 331 |
| 9.7.3 Response | 331 |

| | |
|---|-----|
| 9.7.3.1 Parameters | 332 |
| 9.7.3.2 Language..... | 332 |
| 9.7.4 Sequence Diagrams | 333 |
| 9.7.4.1 VR.GetLanguage..... | 333 |
| 9.7.5 JSON Messages Examples..... | 333 |
| 9.7.5.1 Request | 333 |
| 9.7.5.2 Response | 333 |
| 9.7.5.3 Error message | 333 |
| 9.7.6 D-Bus Messages Examples | 334 |
| 9.7.6.1 Request | 334 |
| 9.7.6.2 Response | 334 |
| 9.7.6.3 Failure..... | 334 |
| 9.8 Started | 334 |
| 9.8.1 Description..... | 334 |
| 9.8.2 Sequence Diagrams | 335 |
| 9.8.2.1 VR.Started on PTT button press | 335 |
| 9.8.3 JSON Messages Examples..... | 335 |
| 9.8.4 D-Bus Messages Examples | 335 |
| 9.9 Stopped..... | 336 |
| 9.9.1 Description..... | 336 |
| 9.9.2 Sequence Diagrams | 337 |
| 9.8.2.1 VR.Stopped on VR session ending | 337 |
| 9.9.3 JSON Messages Examples..... | 337 |
| 9.9.4 D-Bus Messages Examples | 337 |
| 9.10 OnCommand..... | 338 |
| 9.10.1 Description..... | 338 |
| 9.10.1.1 Parameters | 338 |
| 9.10.2 Sequence Diagrams | 339 |
| 9.10.2.1 VR.OnCommand | 339 |
| 9.10.3 JSON Messages Examples..... | 339 |
| 9.10.4 D-Bus Messages Examples | 340 |
| 9.11 OnLanguageChange..... | 340 |
| 9.11.1 Description..... | 340 |
| 9.11.1.1 Parameters | 340 |
| 9.11.1.2 Language..... | 340 |
| 9.11.2 Sequence Diagrams | 341 |
| 9.11.2.1 VR.OnLanguageChange | 341 |
| 9.11.3 JSON Messages Examples..... | 341 |
| 9.11.4 D-Bus Messages Examples | 342 |
| 10 TTS Component Description..... | 349 |
| 10.1 IsReady..... | 349 |
| 10.1.1 Description..... | 349 |
| 10.1.2 Request | 349 |

| | |
|--|-----|
| 10.1.2.1 Behavior | 349 |
| 10.1.3 Response | 349 |
| 10.1.3.1 Parameters | 349 |
| 10.1.4 Sequence Diagrams | 349 |
| 10.1.4.1 TTS.IsReady and preceding OnReady | 349 |
| 10.1.5 JSON Messages Examples | 350 |
| 10.1.5.1 Request | 350 |
| 10.1.5.2 Response | 350 |
| 10.1.5.3 Error message | 350 |
| 10.1.6 D-Bus Messages Examples | 350 |
| 10.1.6.1 Request | 350 |
| 10.1.6.2 Response | 350 |
| 10.1.6.3 Failure | 351 |
| 10.2 GetCapabilities | 351 |
| 10.2.1 Description | 351 |
| 10.2.2 Request | 351 |
| 7.1.2.1 Behavior | 351 |
| 10.2.3 Response | 351 |
| 10.2.3.1 Parameters | 351 |
| 10.2.3.2 SpeechCapabilities | 352 |
| 10.2.3.3 PrerecordedSpeech | 352 |
| 10.2.4 Sequence Diagrams | 352 |
| 10.1.4.1 TTS.GetCapabilities | 352 |
| 10.2.5 JSON Messages Examples | 352 |
| 10.2.5.1 Request | 352 |
| 10.2.5.2 Response | 353 |
| 10.2.5.3 Error message | 353 |
| 10.2.6 D-Bus Messages Examples | 353 |
| 10.2.6.1 Request | 353 |
| 10.2.6.2 Response | 353 |
| 10.2.6.3 Failure | 353 |
| 10.3 GetSupportedLanguages | 353 |
| 10.3.1 Description | 353 |
| 10.3.2 Request | 354 |
| 7.1.2.1 Behavior | 354 |
| 10.3.3 Response | 354 |
| 10.3.3.1 Parameters | 354 |
| 10.3.3.2 Language | 354 |
| 10.3.4 Sequence Diagrams | 355 |
| 10.3.4.1 TTS.GetSupportedLanguages | 355 |
| 10.3.5 JSON Messages Examples | 355 |
| 10.3.5.1 Request | 355 |
| 10.3.5.2 Response | 356 |
| 10.3.5.3 Error message | 356 |

| | |
|--------------------------------------|-----|
| 10.3.6 D-Bus Messages Examples | 356 |
| 10.3.6.1 Request | 356 |
| 10.3.6.2 Response | 356 |
| 10.3.6.3 Failure..... | 356 |
| 10.4 Speak..... | 356 |
| 10.4.1 Description..... | 356 |
| 10.4.2 Request | 356 |
| 10.4.2.1 Behavior | 356 |
| 10.4.3.1 Parameters..... | 357 |
| 10.4.3.2 TTSShunk | 357 |
| 10.4.3.2 SpeechCapabilities..... | 357 |
| 10.4.3 Response | 358 |
| 10.4.4 Sequence Diagrams | 358 |
| 10.4.4.1 Speak | 358 |
| 10.4.5 JSON Messages Examples | 359 |
| 10.4.5.1 Request | 359 |
| 10.4.5.2 Response | 359 |
| 10.4.5.3 Error message | 359 |
| 10.4.6 D-Bus Messages Examples | 359 |
| 10.4.6.1 Request | 359 |
| 10.4.6.2 Response | 359 |
| 10.4.6.3 Failure..... | 359 |
| 10.5 StopSpeaking..... | 359 |
| 10.5.1 Description..... | 359 |
| 10.5.2 Request | 360 |
| 10.5.2.1 Behavior | 360 |
| 10.5.3 Response | 360 |
| 10.5.4 Sequence Diagrams | 360 |
| 10.5.4.1 StopSpeaking | 360 |
| 10.5.5 JSON Messages Examples | 361 |
| 10.5.5.1 Request | 361 |
| 10.5.5.2 Response | 361 |
| 10.5.5.3 Error message | 361 |
| 10.5.6 D-Bus Messages Examples | 362 |
| 10.5.6.1 Request | 362 |
| 10.5.6.2 Response | 362 |
| 10.5.6.3 Failure..... | 362 |
| 10.6 ChangeRegistration | 362 |
| 10.6.1 Description..... | 362 |
| 10.6.2 Request | 362 |
| 10.6.2.1 Behavior | 362 |
| 10.6.2.2 Parameters..... | 362 |
| 10.6.2.3 Language..... | 363 |
| 10.6.3 Response | 363 |

| | |
|---|-----|
| 10.6.4 Sequence Diagrams | 364 |
| 10.6.4.1 TTS.ChangeRegistration after OnAppRegistered | 364 |
| 10.6.5 JSON Messages Examples | 364 |
| 10.6.5.1 Request | 364 |
| 10.6.5.2 Response | 365 |
| 10.6.5.3 Error message | 365 |
| 10.6.6 D-Bus Messages Examples | 365 |
| 10.6.6.1 Request | 365 |
| 10.6.6.2 Response | 365 |
| 10.6.6.3 Failure | 365 |
| 10.7 GetLanguage | 365 |
| 10.7.1 Description | 365 |
| 10.7.2 Request | 366 |
| 10.7.2.1 Behavior | 366 |
| 10.7.3 Response | 366 |
| 10.7.3.1 Parameters | 366 |
| 10.6.2.3 Language | 366 |
| 10.7.4 Sequence Diagrams | 367 |
| 10.7.4.1 TTS.GetLanguage | 367 |
| 10.7.5 JSON Messages Examples | 367 |
| 10.7.5.1 Request | 367 |
| 10.7.5.2 Response | 367 |
| 10.7.5.3 Error message | 368 |
| 10.7.6 D-Bus Messages Examples | 368 |
| 10.7.6.1 Request | 368 |
| 10.7.6.2 Response | 368 |
| 10.7.6.3 Failure | 368 |
| 10.8 SetGlobalProperties | 368 |
| 10.8.1 Description | 368 |
| 10.8.2 Request | 368 |
| 10.8.2.1 Behavior | 368 |
| 10.8.2.2 Parameters | 368 |
| 10.8.3 Response | 369 |
| 10.8.4 Sequence Diagrams | 369 |
| 10.8.4.1 TTS.SetGlobalProperties and corresponding HMI processing | 369 |
| 10.8.5 JSON Messages Examples | 370 |
| 10.8.5.1 Request | 370 |
| 10.8.5.2 Response | 371 |
| 10.8.5.3 Error message | 371 |
| 10.8.6 D-Bus Messages Examples | 371 |
| 10.8.6.1 Request | 371 |
| 10.8.6.2 Response | 371 |
| 10.8.6.3 Failure | 371 |

| | |
|---|------------------------------|
| 10.9 PerformInteraction | Error! Bookmark not defined. |
| 10.9.1 Description..... | Error! Bookmark not defined. |
| 10.9.2 Request | Error! Bookmark not defined. |
| 10.9.2.1 Behavior | Error! Bookmark not defined. |
| 10.9.2.2 Parameters | Error! Bookmark not defined. |
| 10.9.2.3 TTSCChunk | Error! Bookmark not defined. |
| 10.9.2.4 SpeechCapabilities..... | Error! Bookmark not defined. |
| 10.9.3 Response | Error! Bookmark not defined. |
| 10.9.4 Sequence Diagrams | Error! Bookmark not defined. |
| 10.9.4.1 TTS.PerformInteraction: initial and timeout prompts during UI_ONLY UI.PerformInteraction..... | Error! Bookmark not defined. |
| 10.9.4.1 TTS.PerformInteraction: initial, help and timeout prompts during VR_ONLY UI.PerformInteraction | Error! Bookmark not defined. |
| 10.9.5 JSON Messages Examples | Error! Bookmark not defined. |
| 10.9.5.1 Request | Error! Bookmark not defined. |
| 10.9.5.2 Response | Error! Bookmark not defined. |
| 10.9.5.3 Error message | Error! Bookmark not defined. |
| 10.9.6 D-Bus Messages Examples | Error! Bookmark not defined. |
| 10.9.6.1 Request | Error! Bookmark not defined. |
| 10.9.6.2 Response | Error! Bookmark not defined. |
| 10.9.6.3 Failure..... | Error! Bookmark not defined. |
| 10.10 OnLanguageChange..... | 372 |
| 10.10.1 Description..... | 372 |
| 10.10.1.1 Parameters | 372 |
| 10.10.1.2 Language..... | 372 |
| 10.10.2 Sequence Diagrams | 373 |
| 10.10.2.1 TTS.OnLanguageChange | 373 |
| 10.10.3 JSON Messages Examples | 373 |
| 10.10.4 D-Bus Messages Examples | 373 |
| 10.11 Started | 373 |
| 10.10.1 Description..... | 373 |
| 10.10.2 Sequence Diagrams | 374 |
| 10.10.2.1 TTS.Started upon TTS.Speak request from SDL | 374 |
| 10.10.2.2 TTS.Started during PerfromInteraction..... | 374 |
| 10.10.3 JSON Messages Examples | 375 |
| 10.10.4 D-Bus Messages Examples | 375 |
| 10.12 Stopped..... | 376 |
| 10.10.1 Description..... | 376 |
| 10.10.2 Sequence Diagrams | 376 |
| 10.10.2.1 TTS.Stopped after TTS.StopSpeaking from SDL | 376 |
| 10.10.3 JSON Messages Examples | 376 |
| 10.10.4 D-Bus Messages Examples | 376 |
| 11 VehicleInfo Component Description..... | 377 |
| 11.1 IsReady | 377 |

| | |
|---|-----|
| 11.1.1 Description | 377 |
| 11.1.2 Request | 377 |
| 11.1.2.1 Behavior | 377 |
| 11.1.3 Response | 377 |
| 11.1.3.1 Parameters | 377 |
| 11.1.4 Sequence Diagrams | 377 |
| 11.1.4.1 VehicleInfo.IsReady and preceding OnReady..... | 377 |
| 11.1.5 JSON Messages Examples | 378 |
| 11.1.5.1 Request | 378 |
| 11.1.5.2 Response | 378 |
| 11.1.5.3 Error message | 378 |
| 11.1.6 D-Bus Messages Examples | 378 |
| 11.1.6.1 Request | 378 |
| 11.1.6.2 Response | 379 |
| 11.1.6.3 Failure..... | 379 |
| 11.2 GetVehicleType | 379 |
| 11.2.1 Description | 379 |
| 11.2.2 Request | 379 |
| 11.2.2.1 Behavior | 379 |
| 11.2.3 Response | 379 |
| 11.2.3.1 Parameters | 379 |
| 11.2.3.2 VehicleType..... | 379 |
| 11.2.4 Sequence Diagrams | 380 |
| 11.2.4.1 GetVehicleType | 380 |
| 11.2.5 JSON Messages Examples | 380 |
| 11.2.5.1 Request | 380 |
| 11.2.5.2 Response | 380 |
| 11.2.5.3 Error message | 381 |
| 11.2.6 D-Bus Messages Examples | 381 |
| 11.2.6.1 Request | 381 |
| 11.2.6.2 Response | 381 |
| 11.2.6.3 Failure..... | 381 |
| 11.3 ReadDID | 381 |
| 11.3.1 Description | 381 |
| 11.3.2 Request | 381 |
| 11.3.2.1 Behavior | 381 |
| 11.3.2.1 Parameters | 382 |
| 11.3.3 Response | 382 |
| 11.3.3.1 Parameters | 382 |
| 11.3.3.2 DIDResult | 382 |
| 11.3.3.3 VehicleDataresultCode | 383 |
| 11.3.4 Sequence Diagrams | 383 |
| 11.3.4.1 ReadDID general processing | 383 |

| | |
|---|-----|
| 11.3.4.2 ReeadDID with expanded didResult in response | 383 |
| 11.3.5 JSON Messages Examples | 384 |
| 11.3.5.1 Request | 384 |
| 11.3.5.2 Response | 384 |
| 11.3.5.3 Error message | 385 |
| 11.3.6 D-Bus Messages Examples | 385 |
| 11.3.6.1 Request | 385 |
| 11.3.6.2 Response | 385 |
| 11.3.6.3 Failure | 385 |
| 11.4 GetDTCs | 385 |
| 11.4.1 Description | 385 |
| 11.4.2 Request | 385 |
| 11.4.2.1 Behavior | 385 |
| 11.4.2.1 Parameters | 386 |
| 11.4.3 Response | 386 |
| 11.4.3.1 Parameters | 386 |
| 11.4.4 Sequence Diagrams | 386 |
| 11.4.4.1 GetDTC | 386 |
| 11.4.5 JSON Messages Examples | 387 |
| 11.4.5.1 Request | 387 |
| 11.4.5.2 Response | 387 |
| 11.4.5.3 Error message | 387 |
| 11.4.6 D-Bus Messages Examples | 388 |
| 11.4.6.1 Request | 388 |
| 11.4.6.2 Response | 388 |
| 11.4.6.3 Failure | 388 |
| 11.5 DiagnosticMessage | 388 |
| 11.5.1 Description | 388 |
| 11.5.2 Request | 388 |
| 11.5.2.1 Behavior | 388 |
| 11.5.2.1 Parameters | 388 |
| 11.5.3 Response | 388 |
| 11.5.3.1 Parameters | 389 |
| 11.5.4 Sequence Diagrams | 389 |
| 11.5.4.1 GetDTC | 389 |
| 11.5.5 JSON Messages Examples | 389 |
| 11.5.5.1 Request | 389 |
| 11.5.5.2 Response | 389 |
| 11.5.5.3 Error message | 390 |
| 11.5.6 D-Bus Messages Examples | 390 |
| 11.5.6.1 Request | 390 |
| 11.5.6.2 Response | 390 |
| 11.5.6.3 Failure | 390 |
| 11.6 SubscribeVehicleData | 390 |

| | |
|--|-----|
| 11.6.1 Description..... | 390 |
| 11.6.2 Request | 390 |
| 11.6.2.1 Behavior | 390 |
| 11.6.2.1 Parameters..... | 391 |
| 11.6.3 Response | 392 |
| 11.5.3.1 Parameters | 393 |
| 11.6.3.2 VehicleDataResult..... | 394 |
| 11.6.3. VehicleDataType Enumeration..... | 394 |
| 11.6.3. VehicleDataresultCode Enumeration..... | 395 |
| 11.5.4 Sequence Diagrams | 395 |
| 11.5.4.1 SubscribeVehicleData | 395 |
| 11.5.5 JSON Messages Examples..... | 395 |
| 11.5.5.1 Request | 395 |
| 11.5.5.2 Response | 396 |
| 11.5.5.3 Error message | 397 |
| 11.5.6 D-Bus Messages Examples | 397 |
| 11.5.6.1 Request | 397 |
| 11.5.6.2 Response | 397 |
| 11.5.6.3 Failure..... | 397 |
| 11.7 UnSubscribeVehicleData..... | 398 |
| 11.7.1 Description..... | 398 |
| 11.7.2 Request | 398 |
| 11.7.2.1 Behavior | 398 |
| 11.7.2.1 Parameters..... | 398 |
| 11.7.3 Response | 399 |
| 11.7.3.1 Parameters | 400 |
| 11.7.3.2 VehicleDataResult..... | 401 |
| 11.7.3. VehicleDataType Enumeration..... | 401 |
| 11.7.3. VehicleDataresultCode Enumeration..... | 402 |
| 11.7.4 Sequence Diagrams | 402 |
| 11.7.4.1 UnSubscribeVehicleData..... | 402 |
| 11.7.5 JSON Messages Examples..... | 402 |
| 11.7.5.1 Request | 402 |
| 11.7.5.2 Response | 403 |
| 11.7.5.3 Error message | 404 |
| 11.7.6 D-Bus Messages Examples | 404 |
| 11.7.6.1 Request | 404 |
| 11.7.6.2 Response | 404 |
| 11.7.6.3 Failure..... | 405 |
| 11.8 GetVehicleData..... | 405 |
| 11.8.1 Description..... | 405 |
| 11.8.2 Request | 405 |
| 11.8.2.1 Behavior | 405 |
| 11.8.2.1 Parameters | 405 |

| | |
|---|-----|
| 11.8.3 Response | 406 |
| 11.8.3.1 Parameters | 407 |
| 11.8.4 Sequence Diagrams | 412 |
| 11.8.4.1 GetVehicleData | 412 |
| 11.8.5 JSON Messages Examples | 412 |
| 11.8.5.1 Request | 412 |
| 11.8.5.2 Response | 412 |
| 11.8.5.3 Error message | 414 |
| 11.8.6 D-Bus Messages Examples | 414 |
| 11.8.6.1 Request | 414 |
| 11.8.6.2 Response | 414 |
| 11.8.6.3 Failure | 414 |
| 11.9 OnVehicleData | 414 |
| 11.9.1 Description | 414 |
| 11.9.1.1 Parameters | 414 |
| 11.9.2 Sequence Diagrams | 416 |
| 11.9.2.1 OnVehicleData | 416 |
| 11.9.3 JSON Messages Examples | 416 |
| 11.9.4 D-Bus Messages Examples | 417 |
| 12 Navigation Component Description | 418 |
| 12.1 IsReady | 418 |
| 12.1.1 Description | 418 |
| 12.1.2 Request | 418 |
| 12.1.2.1 Behavior | 418 |
| 12.1.3 Response | 418 |
| 12.1.3.1 Parameters | 418 |
| 12.1.4 Sequence Diagrams | 418 |
| 12.1.4.1 Navigation.IsReady and preceding OnReady | 418 |
| 12.1.5 JSON Messages Examples | 419 |
| 12.1.5.1 Request | 419 |
| 12.1.5.2 Response | 419 |
| 12.1.5.3 Error message | 419 |
| 12.1.6 D-Bus Messages Examples | 419 |
| 12.1.6.1 Request | 419 |
| 12.1.6.2 Response | 420 |
| 12.1.6.3 Failure | 420 |
| 12.3 ShowConstantTBT | 422 |
| 12.3.1 Description | 422 |
| 12.3.2 Request | 422 |
| 12.3.2.1 Behavior | 422 |
| 12.3.2.1 Parameters | 422 |
| 12.3.3 Response | 423 |
| 12.3.4 Sequence Diagrams | 423 |

| | |
|--------------------------------------|-----|
| 12.3.4.1 ShowConstantTBT | 423 |
| 12.3.5 JSON Messages Examples | 423 |
| 12.3.5.1 Request | 423 |
| 12.3.5.2 Response | 424 |
| 12.3.5.3 Error message | 424 |
| 12.3.6 D-Bus Messages Examples | 425 |
| 12.3.6.1 Request | 425 |
| 12.3.6.2 Response | 425 |
| 12.3.6.3 Failure | 425 |
| 12.4 UpdateTurnList | 425 |
| 12.4.1 Description | 425 |
| 12.4.2 Request | 425 |
| 12.4.2.1 Behavior | 425 |
| 12.4.2.1 Parameters | 425 |
| 12.4.3 Response | 425 |
| 12.4.4 Sequence Diagrams | 426 |
| 12.4.4.1 UpdateTurnList | 426 |
| 12.4.5 JSON Messages Examples | 426 |
| 12.4.5.1 Request | 426 |
| 12.4.5.2 Response | 427 |
| 12.4.5.3 Error message | 427 |
| 12.4.6 D-Bus Messages Examples | 427 |
| 12.4.6.1 Request | 427 |
| 12.4.6.2 Response | 427 |
| 12.4.6.3 Failure | 427 |
| 12.5 StartStream | 428 |
| 12.5.1 Description | 428 |
| 12.5.1.1 Parameters | 428 |
| 12.5.2 Sequence Diagrams | 428 |
| 12.5.2.1 StartStream | 428 |
| 12.5.3 JSON Messages Examples | 428 |
| 12.5.4 D-Bus Messages Examples | 428 |
| 12.6 StopStream | 428 |
| 12.6.1 Description | 428 |
| 12.6.1.1 Parameters | 428 |
| 12.6.2 Sequence Diagrams | 429 |
| 12.6.2.1 StopStream | 429 |
| 12.6.3 JSON Messages Examples | 429 |
| 12.6.4 D-Bus Messages Examples | 429 |
| 12.5 StartAudioStream | 429 |
| 12.5.1 Description | 429 |
| 12.5.1.1 Parameters | 429 |
| 12.5.2 Sequence Diagrams | 429 |

| | |
|---------------------------------------|-----|
| 12.5.2.1 StartStream | 429 |
| 12.5.3 JSON Messages Examples | 430 |
| 12.5.4 D-Bus Messages Examples | 430 |
| 12.6 StopAudioStream..... | 430 |
| 12.6.1 Description..... | 430 |
| 12.6.1.1 Parameters | 430 |
| 12.6.2 Sequence Diagrams | 430 |
| 12.6.2.1 StopStream..... | 430 |
| 12.6.3 JSON Messages Examples | 430 |
| 12.6.4 D-Bus Messages Examples | 430 |
| 12.7 OnTBTClientState..... | 431 |
| 12.7.1 Description..... | 431 |
| 12.7.1.1 Parameters | 431 |
| 12.7.2 Sequence Diagrams | 431 |
| 12.7.2.1 OnTBTClientState | 431 |
| 12.7.3 JSON Messages Examples | 431 |
| 12.7.4 D-Bus Messages Examples | 431 |
| 13 Common Component Description | 467 |
| 13.1 Enumerations..... | 467 |
| 13.1.1 Result | 467 |
| 13.1.2 AppHMIType..... | 468 |
| 13.1.3 DeactivateReason | 469 |
| 13.1.4 ApplicationsCloseReason..... | 469 |
| 13.1.5 ClockUpdateMode | 469 |
| 13.1.6 SystemContext | 470 |
| 13.1.7 DisplayType | 470 |
| 13.1.8 MediaClockFormat | 471 |
| 13.1.9 HmiZoneCapabilities | 472 |
| 13.1.10 SpeechCapabilities | 472 |
| 13.1.11 VrCapabilities | 473 |
| 13.1.12 DriverDistractionState..... | 473 |
| 13.1.11 SoftButtonType | 473 |
| 13.1.12 SystemAction..... | 474 |
| 13.1.13 TextAlignement..... | 474 |
| 13.1.14 Language..... | 475 |
| 13.1.15 TextFieldName | 476 |
| 13.1.17 ImageFieldName | 478 |
| 13.1.18 ImageType..... | 479 |
| 13.1.17 ButtonName..... | 479 |
| 13.1.18 ButtonEventMode | 480 |
| 13.1.19 ButtonPressMode | 481 |
| 13.1.20 LayoutMode | 481 |

| | | |
|---------|-------------------------------------|-----|
| 13.1.21 | TouchEvent | 482 |
| 13.1.22 | SamplingRate | 482 |
| 13.1.23 | BitsPerSample | 482 |
| 13.1.24 | AudioType | 483 |
| 13.1.25 | KeyboardLayout | 483 |
| 13.1.26 | KeyboardEvent | 483 |
| 13.1.27 | KeypressMode | 484 |
| 13.1.28 | AmbientLightStatus | 484 |
| 13.1.29 | ECallConfirmationStatus | 485 |
| 13.1.30 | VehicleDataNotificationStatus | 485 |
| 13.1.31 | EmergencyEventType | 486 |
| 13.1.32 | FuelCutoffStatus | 486 |
| 13.1.33 | PowerModeQualificationStatus | 486 |
| 13.1.34 | CarModeStatus | 487 |
| 13.1.35 | PowerModeStatus | 487 |
| 13.1.36 | ComponentVolumeStatus | 488 |
| 13.1.37 | PRNDL | 488 |
| 13.1.38 | WarningLightStatus | 489 |
| 13.1.39 | VehicleDataEventStatus | 489 |
| 13.1.40 | IgnitionStableStatus | 490 |
| 13.1.41 | IgnitionStatus | 490 |
| 13.1.42 | DeviceLevelStatus | 490 |
| 13.1.43 | Primary AudioSource | 491 |
| 13.1.44 | VehicleDataStatus | 491 |
| 13.1.45 | WiperStatus | 491 |
| 13.1.46 | CompassDirection | 492 |
| 13.1.47 | Dimension | 493 |
| 13.1.48 | VehicleDataResultCode | 493 |
| 13.1.49 | TBTState | 493 |
| 13.2 | Structures | 494 |
| 13.2.1 | HMIApplication | 494 |
| 13.2.2 | DeviceInfo | 495 |
| 13.2.3 | MenuParams | 495 |
| 13.2.4 | Choice | 496 |
| 13.2.5 | TimeFormat | 497 |
| 13.2.6 | VrHelpItem | 497 |
| 13.2.7 | DisplayCapabilities | 498 |
| 13.2.7 | TouchEventCapabilities | 499 |
| 13.2.8 | ImageResolution | 499 |
| 13.2.9 | ScreenParams | 499 |
| 13.2.10 | ImageField | 500 |

| | | |
|---------|---------------------------------|-----|
| 13.2.11 | TextFieldStruct | 500 |
| 13.2.12 | SoftButtonCapabilities | 501 |
| 13.2.13 | SoftButton | 501 |
| 13.2.14 | Image | 503 |
| 13.2.15 | ButtonCapabilities | 503 |
| 13.2.16 | PresetBankCapabilities | 504 |
| 13.2.17 | AudioPassThruCapabilities | 504 |
| 13.2.18 | TouchLists | 504 |
| 13.2.19 | Coordinate | 505 |
| 13.2.20 | TouchArea | 505 |
| 13.2.21 | TouchEventInfo | 506 |
| 13.2.22 | KeyboardProperties | 506 |
| 13.2.23 | TTSChunk | 507 |
| 13.2.23 | VehicleType | 507 |
| 13.2.24 | DIDResult | 507 |
| 13.2.25 | GPSData | 508 |
| 13.2.26 | TireStatus | 509 |
| 13.2.27 | SingleTireStatus | 509 |
| 13.2.28 | BeltStatus | 510 |
| 13.2.29 | BodyInformation | 511 |
| 13.2.30 | DeviceStatus | 511 |
| 13.2.31 | HeadLampStatus | 512 |
| 13.2.32 | MyKey | 512 |
| 13.2.33 | Turn | 512 |
| 13.2.34 | ECallInfo | 513 |
| 13.2.35 | AirbagStatus | 513 |
| 13.2.36 | EmergencyEvent | 514 |
| 13.2.37 | ClusterModeStatus | 515 |
| | References | 516 |
| | Change History | 517 |

Introduction

This document provides the information for integrating the system of SmartDeviceLink (SDL) with the vehicle Head Unit (HU).

SDL is a system that installed on vehicle head unit allows the application on mobile device to utilize some vehicle functionality such as text-to-speech, voice recognition, display, hardware buttons, etc.

With the help of SDL the applications running on the mobile devices with:

- Operating system of Android / iOS / BlackBerry
- Connection over Bluetooth / USB / WiFi

might be presented to vehicle HMI.

The given guideline describes:

- The types of transports supported by SDL for communication with HMI
- How to connect over one of the supported transports
- The types of messages and the message formats used for communication
- The API that needs to be supported by vehicle HMI.

The document also provides:

- The drawings showing the exemplary display layout for illustrating the expected HMI behavior.
- The sequence diagrams for showing the sequence of messages to be received/responded by both HMI and SDL.
- The examples for each function call in different message formats corresponding to the transports currently supported.

Abbreviations and Definitions

Abbreviations used in this document are collected in the table below.

| Abbreviation | Expansion |
|--------------|----------------------------------|
| BT | Bluetooth |
| CID | Center Information Display |
| DID | Data Identifier |
| DTC | Diagnostic Trouble Code |
| ECU | Electronic Control Unit |
| GPS | Global Positioning System |
| GUI | Graphical User Interface |
| HDOP | Horizontal Dilution Of Precision |
| HMI | Human Machine Interface |
| IVI | In-Vehicle Infotainment |
| JSON | JavaScript Object Notation |
| PDOP | Positional Dilution Of Precision |
| RPC | Remote Procedure Call |
| SDE | Software Development Environment |
| SDL | SmartDeviceLink |
| SEE | Software Engineering Environment |
| TTS | Text To Speech |
| VDOP | Vertical Dilution Of Precision |
| VR | Voice Recognition |
| UTC | Universal Time Coordinate |

The list of terms used in the document and their definitions is provided in the table below.

| Term | Definition |
|-------------------------|--|
| Data Identifier (DID) | Addressed memory locations of the ECU. |
| Diagnostic Trouble Code | In the automotive industry, codes that are prescribed by SAE standards to help track problems in a vehicle detected by its on-board computer |
| JSON | - Is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. - Is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages. |

| | |
|---------------|---|
| GPS | GPS – The Global Positioning System – is a space-based satellite navigation system that provides location (using not less than four satellites simultaneously) and time information to anyone who exploits GPS receiver. When installed in a car, a GPS unit can provide useful information about the car's position and the best travel routes to a given destination. |
| E911 override | Occurs in case of a car accident – when the airbag deploys or, in certain vehicles, the emergency fuel pump shutoff is activated. The vehicle system (in case of relevant version) should react in appropriate way [3] when this override happens. |
| VR | Voice recognition is the technology by which sounds, words or phrases spoken by humans are converted into electrical signals and these signals are transformed into coding patterns to which meaning has been assigned. Relating to automotive, voice recognition aims to enable drivers to control entertainment and navigation systems simply by using their voices. |
| Command | This is an option that a user can select either via voice recognition (VR) or through the on-screen menu during a user-initiated interaction. |
| Phoneme | A <i>phoneme</i> is the basic unit of sound in a language. |

I. SDL Programmer`s Guide

1 Overview

2 Architecture

3 Transport Manager

3.1 General description

3.1.1 Transport layer features

Several definitions:

- 1) 'Connection establishing' with the application or device:
 - Means that transport layer creates a physical interconnection for sending and receiving 'handshake' data.
 - Does not mean that this application or device shall be marked as "connected" for the user.
 - May have two options:
 - One-time connection: when the data related to a device is cleared up after disconnection
 - Persistent connection: when the information about the device is stored even in the power off mode.
- 2) 'Connection closing':
 - Means that connection is not terminated until 'goodbye' data transmission is finished (all accumulated data is sent to / received from the application).
 - Physical disconnection happens when there is no more data left for transmission.

Table #1: List of features provided by transport layer

| N | Feature |
|-----|--|
| 1. | Search for applications on devices using all available transports |
| a) | Notify with the list of applications found on device |
| b) | Notify on search finished in adapter |
| c) | Notify on search done |
| d) | Inform for which transport the connection has failed |
| 2. | Establish a connection with the user-selected application for specific device using specific transport |
| a) | Notify on connection successful setup |
| b) | Notify on connection failure |
| 3. | Establish a connection with the user-selected application for specific device on all transports supported by the application |
| a) | Notify on connection successful setup |
| b) | Inform for which transport the connection has failed |
| 4. | Establish a connection with all applications for the user-selected device (using specific or all supported transports) |
| a) | Notify on connection successful setup |
| b) | Inform for which application (and transport) the connection has failed |
| 5. | Establish a connection with all applications on all user-selected devices (using specific or all supported transports) |
| a) | Notify on connection successful setup |
| b) | Inform for which application, device and/or transport the connection has failed |
| 6. | Establish a connection with the application that requests an access to the system by all available transports |
| a) | Notify on connection request from the application on unknown device |
| 7. | Remember the device (device is marked in a special way and is kept in the system until the user asks to clear all data connected with it) |
| 8. | 'Forget' the device (removing the mark from the device and removing the device from internal storage) |
| 9. | Enable/Disable SDL visibility for devices (similar to BT visible/invisible) |
| 10. | Close connection for specific application on specific device of specific transport |
| a) | Notify on successful disconnection |
| b) | Notify on disconnection failure |
| 11. | Close connection for specific application on specific device of all transports |
| a) | Notify on successful disconnection |
| b) | Notify on disconnection failure |
| 12. | Close connection for specific device of specific transport |
| a) | Notify on successful disconnection |

| | |
|-----|---|
| b) | Notify on disconnection failure |
| 13. | Close connection for specific device of all available transports |
| a) | Notify on successful disconnection |
| b) | Notify on disconnection failure |
| 14. | Close connection for all devices of all transports |
| a) | Notify on successful disconnection |
| b) | Notify on disconnection failure |
| 15. | Transmit data to a connected device |
| a) | Transmit data with the use of priority |
| b) | Notify on successful data transmission |
| c) | Inform which part of transmitted data stream has failed |
| 16. | Receive data from a connected device |
| a) | Process received data with the use of priority |
| b) | Notify on successful data receipt |
| 17. | Reconnect the unexpectedly disconnected application (device) |
| a) | Accumulate data destined for unexpectedly disconnected device |
| b) | Transmit all accumulated data in correct order if the application has reconnected |
| c) | Inform that the accumulated data has not been sent if there is no connection established after the timeout. |
| 18. | Restore previous state |
| a) | Save the internal information about the connected devices in persistent memory before the shutdown |
| b) | Load the information about connected devices from persistent memory |
| 19. | Reconnect applications configured to auto reconnect on power on |
| 20. | Automatically select transport according to priority for multi-transport applications (if appropriately configured) |
| 21. | Automatically switch the data flow for unexpectedly disconnected transport to another transport for multi-transport applications (if appropriately configured) |
| 22. | Provide current state information |
| a) | List the connected devices providing the list of applications for each device |
| b) | List the existing connections providing the information on their status |
| c) | List the available transports |

3.1.2 Transport Level Structure

The *Figure 1* demonstrates the structure of the Transport Level:

- Transport Manager has no limitations on transports number.
- Each transport has no limitations on devices number.
- Any device may be connected through unlimited number of transports. In this case each connection established between the application on a device and the Transport Manager has the unique ID.

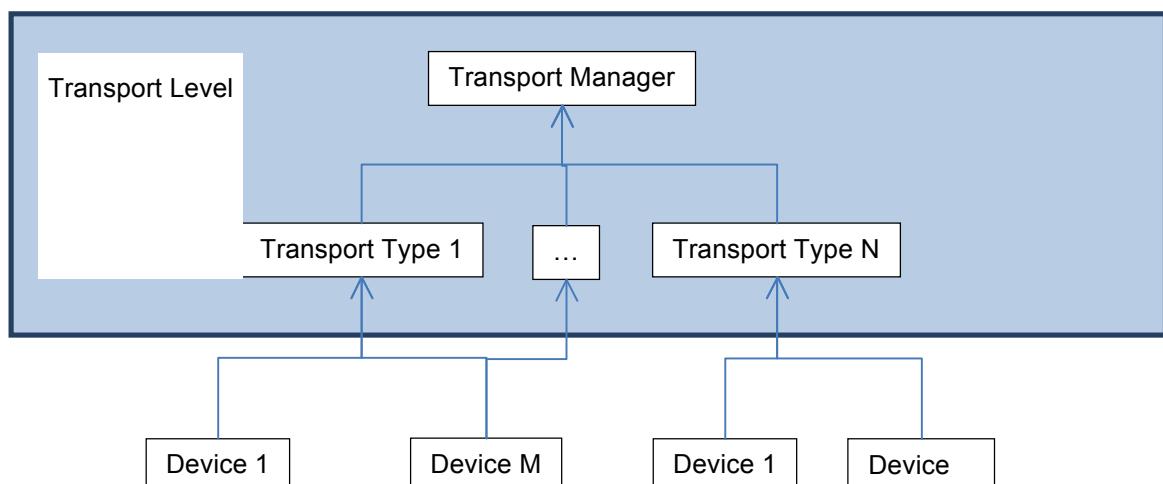


Figure 1: Transport Level Structure Diagram

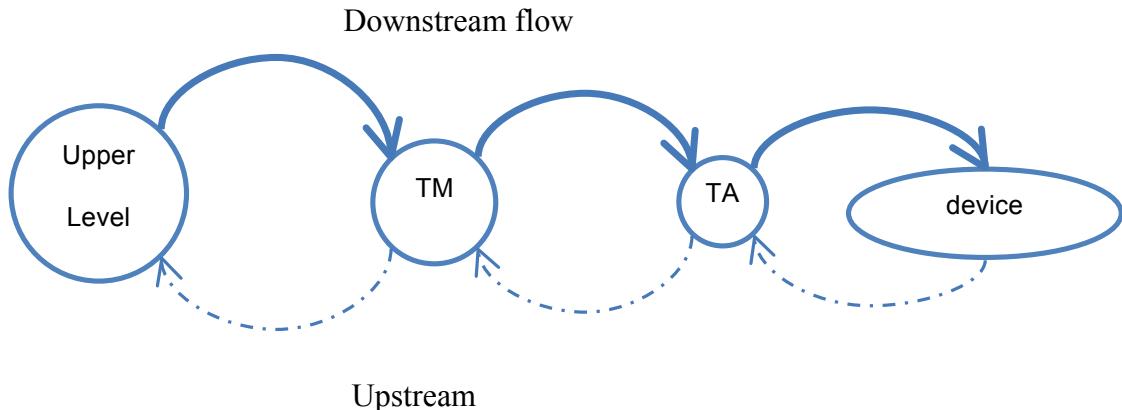


Figure 2: Transport level control and data flow (downstream – to device, upstream – from device)

The *Figure 2* demonstrates the data flows:

- Each abstraction level communicates only with the next and the previous abstraction level.
- Direct communication of two abstraction levels that are not the direct neighbors is prohibited.

For example:

The “Upper Level” cannot access the “Transport Adapter (TA)”

The “Transport Manager (TM)” cannot access the device and so forth.

The structure of the Transport Manager and the Transport Adapter is described in the below sections

3.1.2.1 Transport Manager Structure

The following diagram represents the structure of the Transport Manager.

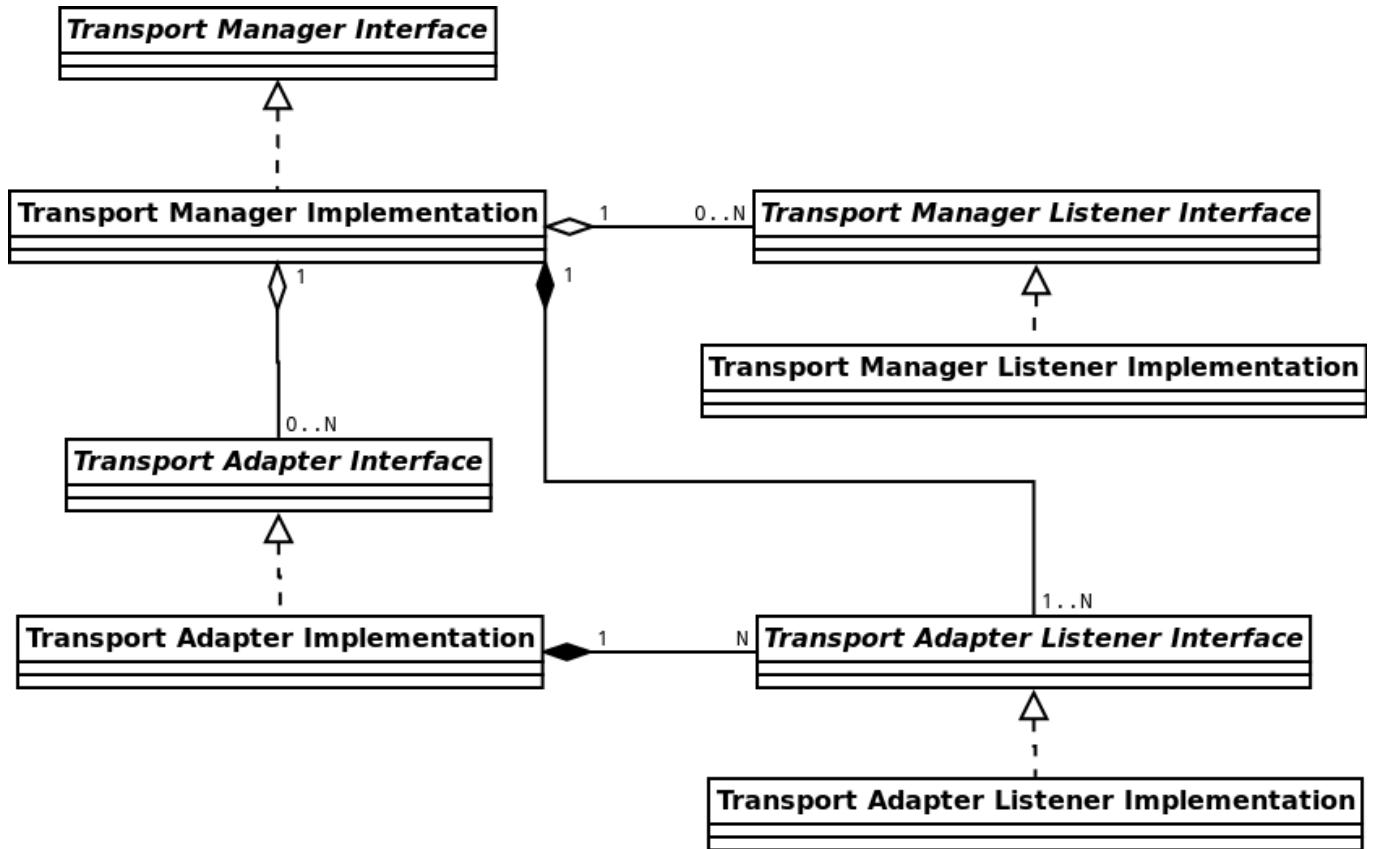


Figure 3: Transport Manager Class Structure Diagram

3.1.2.2 Transport Adapter Structure

The following diagram represents the structure of the Transport Adapter.

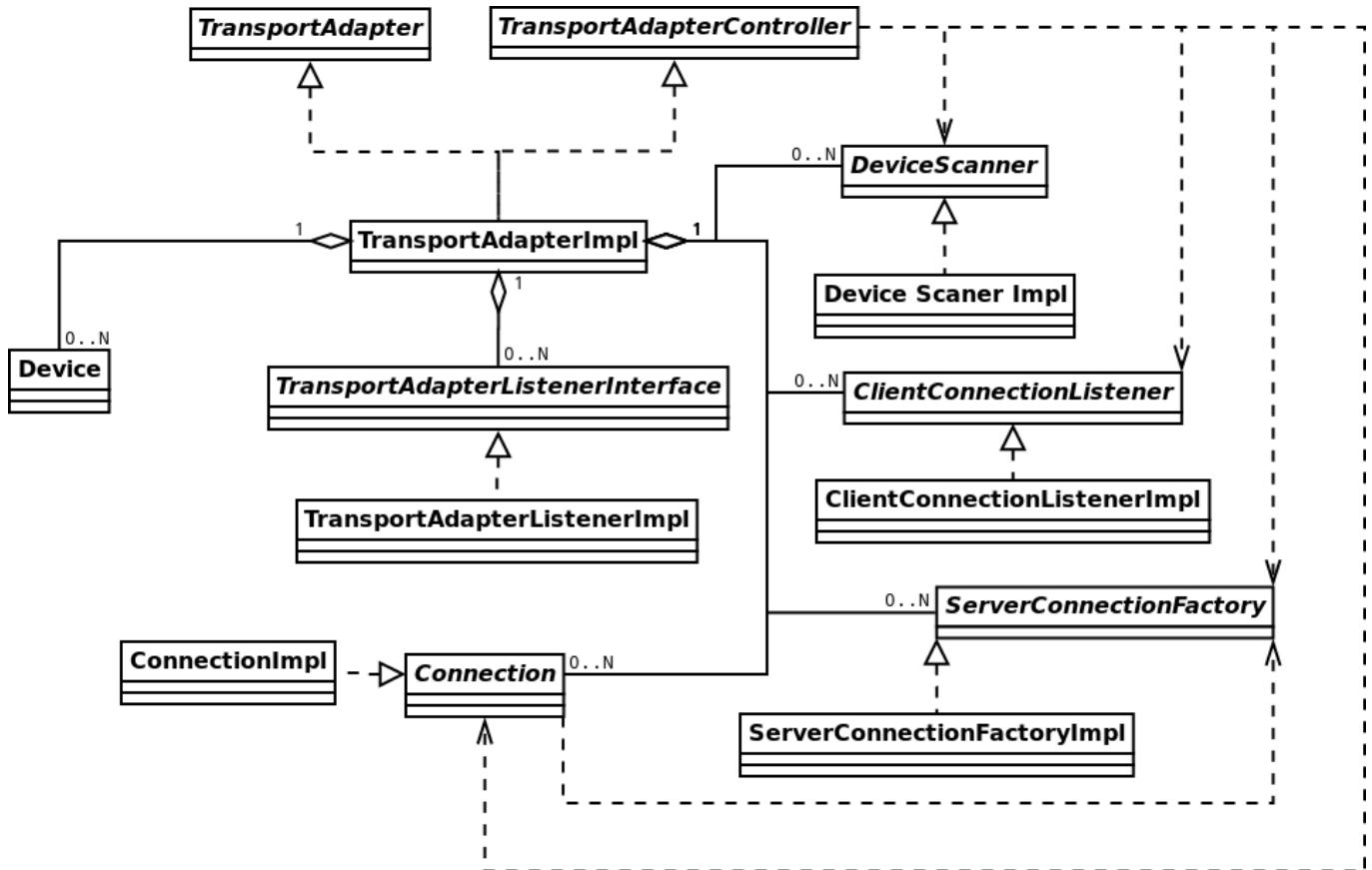


Figure 4: Transport Adapter Class Structure Diagram

3.1.3 Operation Examples (Message Sequence Chart)

3.1.3.1 New device search

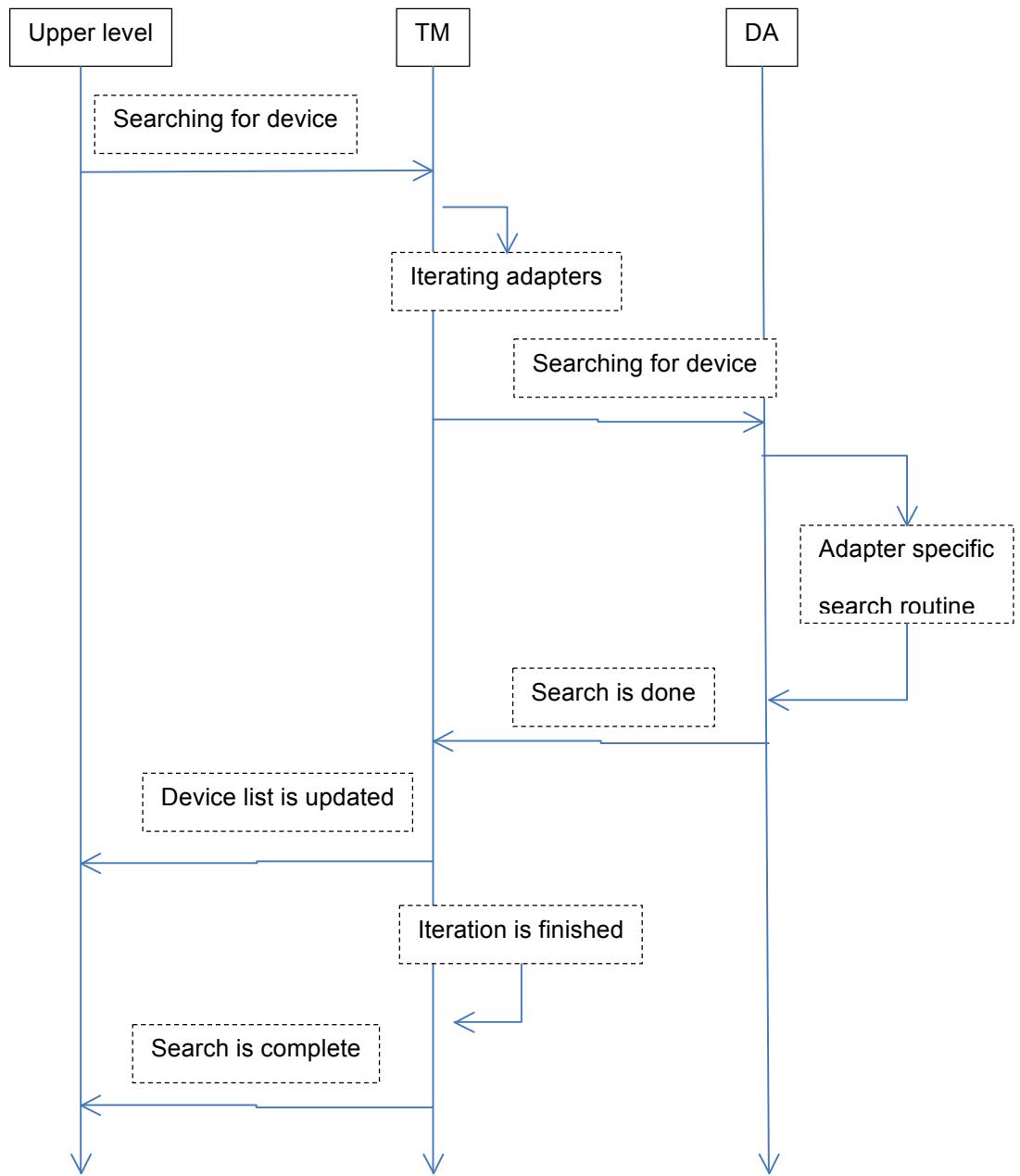


Figure 5: Transport Manager Message Sequence Chart for a New Device Search

3.1.3.2 Device-originating connection

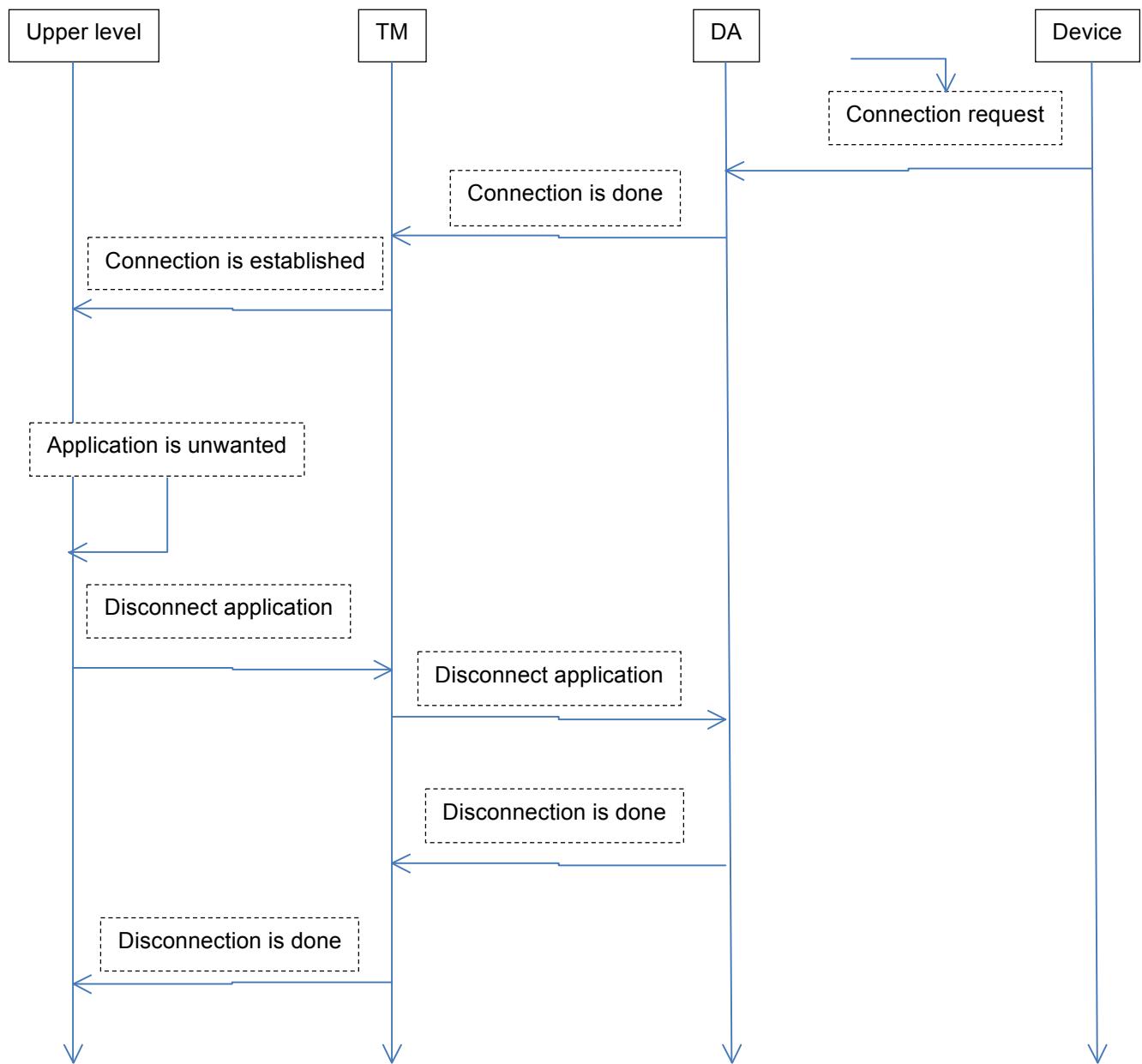


Figure 6: Transport Manager Message Sequence Chart for Device-Originated Connection

3.1.3.3 Connection close command

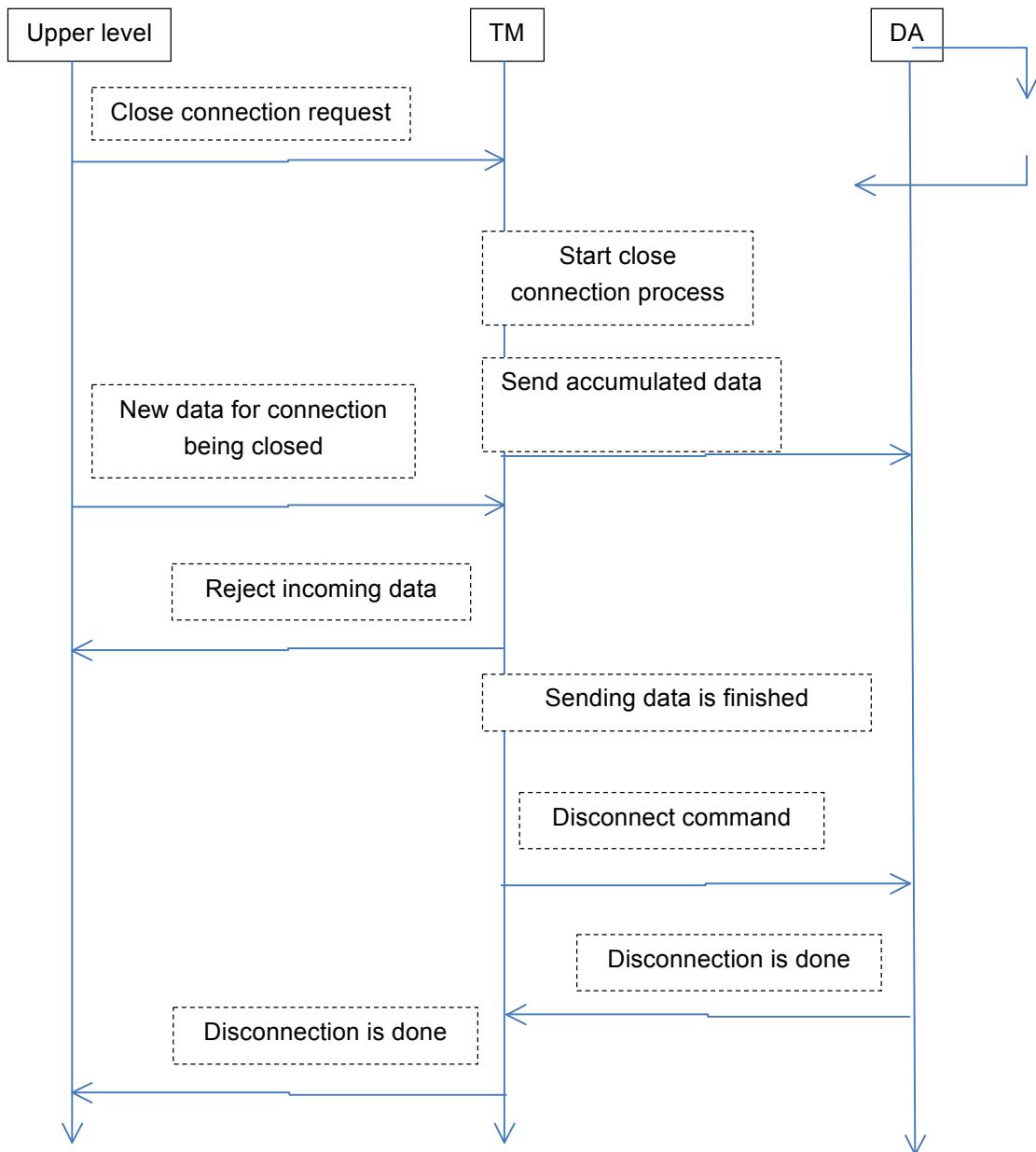


Figure 7: Transport Manager Message Sequence Chart for Connection Close Command

3.2 Transport Manager (TM) Usage

3.2.1 General Processing Description

3.2.1.1 TM initialization

- Every time SDL starts working it calls the creation and initialization of TM (the default or developer-defined one).
- TM uses singleton pattern and the instance of TM is created and initialized while being retrieved. And the pointer is provided to the developer.

- If TM is initialized once it must not be initialized again.
- The developer may initialize the customized TM by calling the appropriate ‘init()’ function:
 - If the customized TM is based on default implementation, ‘init()’ must NOT be called twice for one and the same TM
 - If the customized TM is created from scratch, it is up to developer to choose the initialization mechanism.
- During the initialization process TM creates two threads:
 - For processing the message queue of commands from the upper level
 - For processing the events coming from devices.
- Also TM creates and initializes all available Transport Adapters (the default and/or customized ones).
- If appropriately configured TM may load the information about previous state and perform necessary actions (e.g. reconnect the last connected application).
- TM becomes initialized:
 - When its internal threads are created and are ready for working (i.e. the ‘init()’ function is serial and if it returns the control, the initialization has completed successfully and TM is ready for working).
 - And it does not watch whether the Transport Adapter(s) has initialized by this time:
 - If any of TAs failed to initialize, this is informed into the log file. TM answers with erroneous messages to the requests related to such TA.
 - If there are NO TAs (either not initialized or not defined), TM answers with the error messages to all of the requests.

3.2.1.2 *TM structure*

- *TM Core*: it is a `TransportManagerImpl` class. It contains the processing mechanisms (e.g., for procedures of sending/receiving the data, for connecting/disconnecting procedures, etc.).It does not have the embedded default Adapters and Listeners. And it would not work apart of the Wrapper.
- *TM Wrapper*: it is implemented in `TransportManagerDefault` class. It adds the default Transport Adapter(s) and Listener to the Core completing the fully-featured functionality.

3.2.1.3 *TA initialization*

- The default Adapters are created and initialized by the default TM Wrapper.
- The customized Adapters should be initialized by the developer himself:
 - TM is initialized first
 - Then the `init()` function for the custom Adapter is called
 - Then the initialized custom TA and the Listener are added to the initialized TM.
- Transport adapter in its turn creates and initializes all available workers.
 - ‘Search Device’ worker creates a separate thread and waits for the ‘Start’ command.
 - ‘Client Listener’ creates a separate thread and waits for connections from devices.
 - ‘Server Connection’ worker executes all actions on caller’s thread (does not create a thread).

3.2.1.4 *Getting started*

- When the initialization is complete, TM starts waiting for:
 - The user’s command
 - The device connection.
 - Resumption from “last state” singleton (if such resumption is specified via profile)
- When one of the above happens, TA:
 - Creates a separate thread for each connection with device’ application(s)

- Notifies TM on connection created.
- When the connection is established, the Upper Level:
 - May start a handshake routine with the application and then notify the user about the application is connected.
 - May close the connection by sending an appropriate command to TM if the application or device is unwanted.

3.2.1.5 Errors in TM

- ‘*Immediate*’ error:
 - When TM is not able to execute any command it will immediately return the appropriate error code. For example: when `connect(app_id)` is called and TM is not initialized yet this type of error occurs.
- ‘*Postponed*’ error:
 - TM is able to execute a command and there is some error occurred in downstream.
 - The component, where the error has occurred, sends the appropriate information to TM.
 - Then TM provides a notification to the Upper Level.

3.2.1.6 Messages in TM

- TM is ready to send and receive data after the connection is opened,
- Sending messages:
 - Each message destined for the device is posted into the message queue.
 - The message is removed from the queue after it is successfully sent to the device.
 - The message(s) is returned to the caller if for some reason TM is not able to send it (e.g., unexpected device disconnection),
- Receiving messages:
 - TM redirects messages from the Transport Adapter to the Upper Level via notification mechanism.

3.2.1.7 Connection identifiers

- TM uses the pair “Device ID” and “Application ID” for accessing the application on a device and internally for connection establishing.
- Device ID:
 - Stands for global unique identifier based on MAC address for network adapters and MAC address like for BT.
 - In the default implementation `Device ID` is logically split into two parts:
 - Internal device ID – that is a MAC address string
 - External device ID – that is an integer value.
 - When the new MAC address is found, the integer value is assigned to it correspondingly (starting from 1 and incremented with every new assignment).
 - For persistent connection,
 - When the device is marked as “known”, the correspondence of internal and external IDs is stored (even after power off) until the user explicitly asks to “forget about <this> device”.
 - When the user requests to ‘forget’ the known device, the MAC address may be assigned with the new integer value on being connected for the next time.
 - For one-time connection this correspondence is not saved in long term storage. And if connected later the same MAC address may get the new integer value in correspondence.
 - It is not defined what happens if two devices with the same MAC address would connect.
- Application ID: is the application unique identifier. It is the incremental integer value assigned and used internally in TM.
- Connection ID:
 - Connection is represented with a unique pair of “Device ID” and “Application ID”.

- Connection ID is a system wide unique identifier (incremented integer value) assigned to each new connection.
- This ID is used for sending/receiving data and for closing the connection.
- There are no certain rules to define how exactly this ID is assigned.

3.2.1.8 *Connection closing*

- When TM receives ‘close connection’ or ‘disconnect’ commands it tries to finish sending accumulated messages and then closes the connection.
- If connection is lost TM will drain all accumulated messages and confirm connection closure.

3.2.1.9 *SDL shutting down*

- When SDL is going to shutdown, TM clears all objects which it has created (e.g. default transport adapters).
- Objects created by developer (e.g. developer’s transport adapter or listener) are not removed by TM. The developer must take care of destructing his customized objects by himself.

3.2.2 Create Default TM Instance

For creating the instance of TM with default configuration it is necessary to use the following code:

```
#include "transport_manager.h"
#include "transport_manager_default.h"

{
...
TransportManager* transport_manager =
    TransportManagerDefault::instance();
...
}
```

Initialization of TM may take some time due to threads creation and initialization routine. After all is set and done Transport Manager is ready to be used. Till that time any command will be rejected with the error code “NOT INITIALIZED” in return value.

TM implementation uses singleton pattern, thus only one instance of TM will exist at a time. Nevertheless, this is not the rule and may be changed in custom TM.

3.2.3 Add Custom Listener to TM

- A Listener allows monitoring the events that take place in TM.
- The number of TM Listeners is not limited and may be zero.
- These listeners are provided to TM and are used by any module that needs to receive the notifications from TM.
- The list of Listeners that are called upon any event occurred in TM is stored in TM.
- TM does not create the own listener by default.
- Custom Listener:
 - Its logic should be related with implementation of the module that uses this Listener.

- Should implement Transport Manager Listener Interface.

To set up the Listener the following code should be used:

```
#include "transport_manager.h"
#include "transport_manager_default.h"

class MyTransportManagerListenerImpl : public TransportManagerListener
{
    //implement here entire interface
}

{
...
TransportManager *tm_impl = TransportManagerDefault::instance();
TransportManagerListener *my_tm_listener = new MyTransportManagerListenerImpl();
tm_impl->addEventListenet(my_tm_listener);
...
}
```

To remove the Listener `removeEventListener` should be used.

3.2.4 Add Custom Transport Adapter to Default TM

3.2.4.1 *About TA in general*

- TM may contain zero to N Transport Adapters (TM with zero Adapters returns the erroneous messages for all of the requests).
- Each Adapter corresponds to specific type of transport (e.g. Bluetooth, LAN, USB, etc.).
- TA implements transport specific search, connect, disconnect and data transfer routines.

3.2.4.2 *Several instances of TA*

- It is allowed to have several adapters of the same type in one TM.
- The results of using several instances of default TAs are not defined.
- It is developer's responsibility to create a custom Adapter that operates well under conditions of using several instances of it.

3.2.4.3 *Add Custom TA*

- The simplest way to add a custom Transport Adapter is to derive the existing implementation of TCP or BT Adapter, adjust some behavior in derived class and add it to TM:

```

#include "transport_manager.h"

#include "transport_manager_default.h"


{

...


//note: the default implementation of TCP adapter is used in this example.

//developer is free to create his own implementation of transport adapter from
scratch

TransportAdapter* my_transport_adapter =
    static_cast<TransportAdapter*>(new TcpTransportAdapter);

my_transport_adapter->init(); //note: TA must be initialized before getting
instance of TM.

TransportManager *tm_impl =
    TransportManagerDefault::instance();

//note: when custom TA will be added to TM, the default listener will be
assigned to it automatically

tm_impl->addTransportAdapter(my_transport_adapter);

...
}

```

3.2.4.4 Initialization

- By default TA is initialized during TM initialization.
- If TA needs to be added to the TM already initialized, this TA should be initialized first.

Note:

- *TM has a TCP adapter by default. Adding the new instance of the same Adapter may result the unexpected behavior (because default Adapters are not designed for working in such configuration).*
- *The developer must change the standard behavior of TCP adapter to eliminate the potential problems.*

3.2.5 Custom TA implementation from scratch

If default implementation of adapter is not suitable for particular case one can create new adapter from scratch:

- Create custom class that implements transport adapter interface. Note that custom TA shall use defined interface `TransportAdapterListener` for notifying Transport Manager.
- The instance of transport adapter listener will be set by Transport Manager automatically when custom transport adapter is added. If custom adapter does not store its listener then adapter will not be able to notify Transport Manager about events such as `OnDataReceiveDone` or `OnConnectDone`.

```

#include "transport_adapter.h"

class MyTransportAdapterImpl : public TransportAdapter
{
    // implement all interface functions here
    // use TransportAdapterListener interface to notify transport manager about event
    // happening
}

```

- Add the following code where the Transport Manager is initialized

```

#include "transport_manager.h"
#include "my_transport_adapter_impl.h"

{
    ...
    TransportAdapter* my_transport_adapter =
        static_cast< MyTransportAdapterImpl*>(new MyTransportAdapterImpl);
    //note: TA must be initialized before getting instance of TM
    my_transport_adapter->init();
    TransportManager *tm_impl = TransportManagerDefault::instance();
    //note: when custom TA will be added to TM, the default listener will be
    assigned to it automatically
    tm_impl->addTransportAdapter(my_transport_adapter);
    ...
}

```

All internal logic implementation is up to developer. Developer is responsible to implement:

- Communication with device
- Notification of state changes
- Error handling and so forth.

3.2.6 Add a New Listener to TA

- A Listener allows monitoring the events that take place in TA.
- The number of TA Listeners is not limited.
- The list of Listeners that are called upon any event occurred in TA is stored in TA.
- Custom Listener for TA:
 - The developer can add the Listener to TA through the customizing procedure only. I.e., the developer needs to create his own TA on the base of the default one, and then to add the Listener to it.
 - Should implement Transport Manager Listener Interface.

Important Note:

- Working with TA Listener by-passing the TM is dangerous and may lead to the asynchronous behavior of TM and the Adapter.
- The custom Listener should be added only together with the new custom Adapter and/or the new custom Transport Manager.

To set up the Listener the following code should be used:

```
#include "transport_manager.h"
#include "transport_manager_default.h"

class MyTransportAdapterListener :public TransportAdapterListenerImpl
{
    //customize listener here if necessary
}

...
//note: the default implementation of TCP adapter and the default implementation
of adapter listener are used in this example.

//developer is free to create his own implementation of transport adapter from
scratch

TransportAdapter* my_transport_adapter =
    static_cast<TransportAdapter*>(new TcpTransportAdapter);
my_transport_adapter->init();

TransportManager *tm_impl =
    TransportManagerDefault::instance();

TransportAdapterListener* my_ta_listener =
    new MyTransportAdapterListener(tm_impl);
my_transport_adapter->addListener(my_listener)
tm_impl->addTransportAdapter(my_transport_adapter);
...
}
```

The developer should implement “TransportAdapterListener” interface with the custom logic before the Listener can be used.

3.2.7 Create TM with Custom TAs Only (with No Default Adapter)

If for some reason the default Adapters are not good enough they can be replaced with developer's defined Adapter(s). To do this the developer

- Must implement the Adapter's Interface using own logic
- Provide this new adapter to TM:

```

#include "transport_manager.h"
#include "transport_manager_impl.h"

class MyTransportManager : public TransportManagerImpl {
    virtual int init();
    virtual ~MyTransportManager();

    MyTransportAdapter *my_adapter_;
    explicit MyTransportManager(const TransportManagerAttr &config)
        : TransportManagerImpl(config),
        my_adapter_ (nullptr){
    }

public:
    static MyTransportManager* instance();
};

//note: the implementation of all methods above is not defined here just for
making the code look simpler

//Obvious MyTransportAdapter should be created and initialized somewhere.

{
...
...
TransportManager *tm_impl = MyTransportManagerImpl::instance();
...
}

```

More detailed information on custom Transport Adapter creation is provided in section 3.2.9**Error! Reference source not found..**

3.2.8 Transport Manager Customizing

3.2.8.1 Basic information

- TM is responsible for all complex logic and decisions, while Transport Adapter is a primitive entity that operates only with transport specifics.
- Communication interface between TM and TA:
 - TM sends a command to TA.
 - If TA is unable to execute this command it returns the error code not processing the command.
 - Otherwise, TA starts executing the command. Then TA notifies TM on executing completion by using the appropriate callback function.

3.2.8.2 Queues as a fundamental of TM

- Message Queue: for commands coming from the Upper Level.
- Event Queue: for events coming from devices.

3.2.8.3 Rules for developer

Customizing TM, the developer:

- Should implement transport manager interface.
- Should use the Transport Adapter Interface and the Transport Listener Interface for making his implementation work with default Adapters and Listeners.
- Has two options:
 - Creating the TM from scratch.
 - Deriving from the default implementation:
 - If not particularly changed, the default Adapters and Listeners will be used.

3.2.9 Transport Adapter Customizing

3.2.9.1 Basic information

- TA is highly adaptable to any specific of a real transport.
- TA consists of
 - So called *worker classes* that perform a single operation (e.g., device search)
 - *Controller* that
 - Accumulates event handling from all worker classes,
 - Controls the state of all internal data and
 - Notifies the Upper Level via callbacks
 - *Internal data structures* that contain the information about the device, the connection and other necessary information.

3.2.9.2 Workers of TA

- Device Scanner:
 - Implements transport dependent search procedure initiated by the appropriate command from the Transport Manager.
 - It is developer's responsibility to implement this worker.
 - It may be absent for transport types that do not support searching.
 - When the device is found (and in the current default implementation when all the devices are found) this worker:
 - creates a notification and directs this notification to the Controller
 - The Controller notifies TM using the TA Listener
 - TM receives the notification and sends a command to TA for connecting all available applications
 - TM and TA perform a chain of notifications
 - TM notifies the Upper Level using TM Listener with the information on each application connected: the connection ID, the application name, the device name.
- Client Connection Listener:
 - Implements the transport dependent connection that was originated by device.
 - It is developer's responsibility to implement this worker.
 - If transport does not support such ability this worker may be absent.
 - Working procedure:
 - This worker waits for the connection of a mobile device.
 - When the connection request from any of the mobile devices arrives, TA establishes a connection (creates the data path) with this device and the Connection Listener sends a notification through the Controller to the Upper Level with the device and application IDs.
- Server Connection Factory:
 - Implements transport dependent connection that was originated by the user.
 - It is developer's responsibility to implement this worker.
 - If transport does not support such ability this worker may be absent.
 - Creates a connection with the device and the application specified by the user:
 - Both the device and the application must be already known to TA by this moment.
 - TA may know about the device after 'search' routine or after 'restore previous state' routine.
 - If device is not known Transport Adapter returns the error immediately.

- When the connection is created TA sends a notification through the Controller to the Upper Level.

3.2.9.3 Connection

- The main responsibility of Client and Server connection workers is to create a **Connection**.
- Connection is the entity that is responsible for data transmitting between the core and the device.
- Workers and Connection they use are closely related.
- Customizing:
 - Custom implementation of Connection must be used in custom worker(s) only.
 - It is not possible to use other types of data exchange in default workers.
 - It is possible to use default Connection implementation in custom worker.
- Default connection implementation is based on sockets isolated in separate thread (threaded socket connection).
- Each transport specific worker shall use transport dependent initialization of threaded socket connection.
- If the default implementation is not convenient to developer he can create his own Connection implementing any suitable way of data exchanging (e.g. shared memory connection).
- In this case custom workers shall be also created.

3.2.9.4 Descriptor

Descriptor is used for manipulating with devices and connections inside of the adapter.

3.2.9.5 Create custom Transport Adapter

The create custom TA using the Adapter Concept provided by SDK, the developer should do the following:

- Create a class that is derived from “TransportAdapterImpl” and add the implementation of necessary virtual methods. Let it be “getDeviceType”.

```
#include "transport_adapter_impl.h"

class MyTransportAdapter : public TransportAdapterImpl
{
protected:
    virtual DeviceType getDeviceType() const;
}
```

- Create a connection class deriving from “ThreadedSocketConnection” and implement virtual methods.

```
#include "transport_adapter_impl.h"

class MySocketConnection : public ThreadedSocketConnection
{
    virtual ~MySocketConnection();
protected:
    virtual bool establish(ConnectError** error);
}
```

- Create a class for the device that will be used by Controller to manage devices and implement all virtual methods.

```
#include "transport_adapter_impl.h"

class MyDevice : public Device
{
    virtual ~Device();

    virtual bool isSameAs(const Device* other_device) const;

    virtual ApplicationList getApplicationList() const;
}
```

- Create the necessary worker classes by deriving from appropriate basic worker and fill them with necessary functionality.

```
#include "transport_adapter_impl.h"

class MyDeviceScanner : public DeviceScanner
{
}

class MyServerConnectionFactory : public ServerConnectionFactory
{
}

class MyClientListener : public ClientConnectionListener
{
}
```

These workers should use connection and device created in previous steps. For instance scanner should add a list of devices to controller. This list will be used later when ‘connect’ request will be received. To create a data path the connection class should be used.

When connection actually starts it will update Controller with the pointer to this connection.

- When everything is created it is time to combine all together.

```
#include "transport_adapter_impl.h"

MyTransportAdapter::MyTransportAdapter()
    : TransportAdapterImpl(
        new MyDeviceScanner(),
        new MyServerConnectionFactory(),
        new MyClientListener())
{}
```

- Create an instance of the new adapter and provide it to the Transport Manager.

```
#include "transport_manager.h"
#include "transport_manager_impl.h"
#include "my_transport_adapter_impl.h"

{
...
    TransportAdapter* my_transport_adapter =
        static_cast<TransportAdapter*>(new MyTransportAdapter);
    TransportManager *tm_impl =
        TransportManagerDefault::instance();
    tm_impl-> addTransportAdapter(my_transport_adapter);
...
}
```

This adapter will use the default connection implementation and default Adapter Listener but three worker classes will implement developer's logic. Also Transport Adapter will provide device type in developer's defined way.

`TransportAdapterImpl` virtual (but not pure virtual) methods `Store()` and `Restore()` can be reimplemented to provide resumption mechanism. Default implementations for both methods do nothing.

4 Build and Run SDL and Set Up the Environment

This instruction contains the information on how to build and run SDL depending on target OS and HMI emulator type.

4.1 General

To build SDL for QNX, cross-compiling only shall be used.

| Target OS | HMI emulator type | Steps to perform |
|-----------------|-----------------------------|---|
| Linux | HTML5 | 4.3.1 → 4.5.3 → 4.5.5 → 4.6.1 |
| Linux | QML | 4.3.1 → 4.5.4 → 4.5.5 → 4.6.1 |
| virtualized QNX | HTML5 HMI under Linux Host | 4.3.1 → 4.3.2 → 4.3.3 → 4.4.1 → 4.5.1 → 4.5.5 → 4.6.2 → 4.6.3 |
| virtualized QNX | QML HMI under Linux Host | 4.3.1 → 4.3.2 → 4.3.3 → 4.4.2 → 4.5.2 → 4.5.5 → 4.5.4 → 4.5.5 → 4.6.4. |
| virtualized QNX | with QML HMI under QNX Host | 4.3.1 → 4.3.2 → 4.3.3 → 4.5.2 → 4.5.5 → 4.6.2. |

To turn media features (libraries of PulseAudio and GStreamer) on/off please set `EXTENDED_MEDIA_MODE` to ON/OFF, for example:

```
cmake -DEXTEDDED_MEDIA_MODE=OFF
```

4.2 Known issues

1. HTML5 HMI does not display icons in case it is running on distant computer (the reason: SDL sends absolute paths on local computer).
2. Bluetooth does not work under QNX (the reason: BT stack absence for QNX).

4.3 Preparation steps

Please see section “4.1 General” to determine which of the below steps must be used.

4.3.1 To prepare the Linux Host

Note:

In case Ubuntu/Xubuntu 12.04 is already installed, git repository is cloned and is switched to <Main_Develop> branch, proceed from the step 6.

1. Install Ubuntu/Xubuntu 12.04 or higher.
2. Configure source repositories (check "Canonical Partners" repositories in Update Manager->Settings->SW source).
3. Execute "`sudo apt-get install git`" command in terminal.
4. Clone git repository with "`git clone git@adc.luxoft.com:applink`".

Note:

The permissions to download the repository are required. Apply PSavyelyev@luxoft.com on this matter.

5. Switch to <Main_Develop> branch: "`git checkout <Main_Develop>`".
6. Run "`sudo ./setup_env.sh -a`" to install all mandatory and optional packages.

Note:

Read the built-in setup environment help ("./setup_env.sh --help") which provides the description of predefined options of setup packages to be installed.

Executing `./setup_env.sh` without a key will result installing the apckages necessary for building SDL under Linux, with HTML5 HMI.

4.3.2 To set up QNX cross platform build for Linux

1. Download and install QNX SDP 6.5.0 SP1 cross platform tools for Linux (<http://www.qnx.com/download/download/21179/qnxsdp-6.5.0-201007091524-linux.bin>).

Note:

- a) It is necessary to obtain license key: either trial or commercial one (Register on <http://www.qnx.com/> website, enter 'QNX product evaluation' and apply for a license key).
- b) During tools installation mark "GNU Public Licence Utility" check-box.

2. Configure SSH on Linux host:

- 2.1. Install Open-SSH server:

- For Xubuntu run "sudo apt-get install ssh"
- for Ubuntu run "sudo apt-get install openssh-server"

- 2.2. Restart computer if necessary.

3. Unpack crossbuild archive of log4cxx on Linux host :

```
"unzip ./log4cxx.qnx.cross.build.files.zip -d /"
```

Note:

*log4cxx crossbuild archive "log4cxx.qnx.cross.build.files.zip" is located in git folder
git@adc.luxoft.com:applink/src/thirdPartyLibs/*

4.3.3 To run specially prepared QNX in virtual machine

1. Download and install VMware Player from <https://my.vmware.com/web/vmware/downloads>
2. Download the archived QNX image for VMware from ftp://ford-applink.luxoft.com/Distrs/QNX/QNX_SDP_6.5.0.tar.gz
3. Extract image from archive and open it in VMware

Note:

QNX image has root with empty pass and user with pass 'user'. SSH is configured.

4. Log in as a root.

Note:

- a) SSH server shall be started automatically. (Common command is "/usr/sbin/sshd").
- b) Verify that your QNX host ip adress is 192.168.243.131 (if has not changed, see section 4.4.2, step1.)

4.3.4 To make QNX virtual machine from scratch

1. Configure QNX server:

- Download VMware Player <https://my.vmware.com/web/vmware/downloads>
- Download the QNX Software Development Platform 6.5.0 bootable iso image from <http://www.qnx.com/download/download/21182/qnxsdp-6.5.0-x86-201007091524-nto.iso>
- Create a new VMware virtual machine and boot from QNX Software Development Platform bootable iso image

Note:

It is necessary to obtain license key: either trial or commercial one (Register on <http://www.qnx.com/> website, enter 'QNX product evaluation' and apply for a license key).

2. Configure SSH on QNX server:

- Run "passwd user" and set the new user id (just press enter), group (just press enter) and non-empty password.
- Run "ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key".
- Run "ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key".
- Run SSH server if it is not running "/usr/sbin/sshd".

3. Download 'D-Bus' version 1.7.8 and 'Expat' version 2.1.0.

Note:

for more details see <https://adc.luxoft.com/confluence/display/APPLINK/How+to+install+D-Bus>

4. Install log4cxx for QNX

- Download log4cxx development archive "log4cxx.qnx.cross.build.files.zip" and runtime archive "log4cxx.qnx.runtime.files.zip"

Note:

For development use git folder `git@adc.luxoft.com:applink/src/thirdPartyLibs/`
For releases use delivery svn folder

- Unpack crossbuild archive of log4cxx on Linux host :
`unzip ./log4cxx.qnx.cross.build.files.zip -d /`
 - Download log4cxx runtime archive to QNX
`scp <linuxhost_user>@<linuxhost_ip>:<Delivery_Package_Path>/log4cxx.qnx.runtime.files.zip ./`
 - Unpack on QNX server log4cxx binary files of log4cxx QNX port:
`unzip ./log4cxx.qnx.runtime.files.zip -d /`
(this file is also located in git)
5. Set up dbus-daemon and sshd to run at time login
- Create file `"/etc/profile.d/sshd.sh"` with command:
`/usr/sbin/sshd`
 - Create file `"/etc/profile.d/dbus.sh"` with command:

```
if test -z "$DBUS_SESSION_BUS_ADDRESS" ;  
then  
DBUS_SESSION_BUS_ADDRESS=`dbus-daemon --fork --session --print-  
address`  
export DBUS_SESSION_BUS_ADDRESS  
fi
```
6. Set up USB access mode
- If file `"./etc/rc.d/rc.local"` is either absent or empty then copy file `"<Project_Root_Src>/qnx/rc.local"` to folder `"./etc/rc.d"`.
 - If file `"./etc/rc.d/rc.local"` exists and is nonempty then append the last line of file `"<Project_Root_Src>/qnx/rc.local"` to its end.

4.3.5 To set up Android Simulator for using instead of real device

1. Install and configure Android Simulator in keeping with the following instructions
<https://adc.luxoft.com/confluence/display/APPLINK/How+to+install+and+setup+android+emulator+on+Ubuntu>

4.4 Configuration steps

Please see section “4.1 General” to determine which of the below steps must be used.

4.4.1 HTML5 HMI and SDL to be running on separate computers

Note:

Virtualized QNX is also considered to be running on a separate computer

1. LinHost: update SDL config file `"<Project_Root_Src>/src/appMain/smartDeviceLink.ini"`: find `"ServerAddress = "` there and replace the value with `<SDL server address>`.
2. LinHost: update HTML5 HMI network parameters: in file `"<Project_Root_Src>/src/components/HMI/ffw/RPCClient.js"` find `"url: "` and replace the value (`"ws://localhost:8087"` or `FLAGS.WEBSOCKET_URL`) with `<SDL server address>`. (e.g. `"url: "ws://192.168.177.128:8087""`)

Note:

`<SDL server address>` is the IP of the computer where SDL is running.

4.4.2 QML HMI and SDL to be running on separate computers

Note:

Virtualized QNX is also considered to be running on a separate computer

1. QNXServer: add the following lines into the file `"/usr/etc/dbus-1/session.conf"`:
`"<listen>tcp:host=IP-OF-YOUR-QNX-PC,port=PORT-FOR-
LISTEN</listen><allow_anonymous/>"`
(for example,
`"<listen>tcp:host=192.168.243.131,port=8087</listen><allow_anonymous/>"`)

2. LinHost: set up "export DBUS_SESSION_BUS_ADDRESS=tcp:host=<HOST>,port=<PORT>" on HMI side
(e.g. "export DBUS_SESSION_BUS_ADDRESS=tcp:host=192.168.243.131,port=8087").

4.5 Build SDL from source on Linux Host

Note:

In below cases the <Project_Root_Src> is the path to 'Project_Root_Src' git folder (e.g. "/home/user_name/Work/Project_Root_src").

Note:

For release build add "-DCMAKE_BUILD_TYPE=Release" to cmake arguments.

For QNX+QtHMI build flag "-DCMAKE_BUILD_TYPE=Release" is plainly required.

4.5.1 QNX with HTML5 HMI

Run command:

```
mkdir <Project_QNX_Build_WebHMI> && cd <Project_QNX_Build_WebHMI>
cmake <Project_Root_Src> -
DCMAKE_TOOLCHAIN_FILE=<Project_Root_Src>/qnx_6.5.0_linux_x86.cmake
```

4.5.2 QNX with QML HMI

Run command:

```
mkdir <Project_QNX_Build_Qt> && cd <Project_QNX_Build_Qt>
cmake <Project_Root_Src> -
DCMAKE_TOOLCHAIN_FILE=<Project_Root_Src>/qnx_6.5.0_linux_x86.cmake -DHMI=qt -
DCMAKE_BUILD_TYPE=Release
```

4.5.3 Linux with HTML5 HMI

Run command:

```
mkdir <Project_Linux_Build_WebHMI> && cd <Project_Linux_Build_WebHMI>
cmake <Project_Root_Src> -DEXTENDED_MEDIA_MODE=ON
```

4.5.4 Linux with QML HMI

Run command:

```
mkdir <Project_Linux_Build_Qt> && cd <Project_Linux_Build_Qt>
cmake <Project_Root_Src> -DEXTENDED_MEDIA_MODE=ON -DHMI=qt
```

4.5.5 Build SDL

Run build process:

make install

from previously created <Project_(OS-version)_Build_(HMI-type)> directory.

4.6 Run SDL

4.6.1 Linux host for both SDL and HMI (Qt / HTML5)

1. Go to the binary folder:
cd <Project_Linux_Build_WebHMI>/bin
or
cd <Project_Linux_Build_Qt>/bin
2. Run SDL:
. ./smartDeviceLinkCore

Note:

HMI will be started automatically. Type of started HMI depends on build options described in 4.5.

4.6.2 QNX host for both SDL and Qt HMI

1. Log in as a root and make sure you are not running Photon
2. QnxServ: download binary folder:
`scp -r <linux_host_user>@<linux_host_ip>:<Project_QNX_Build_Qt>/bin/* . /`
3. QnxServ: start SDL:
`su -c ./smartDeviceLinkCore`

Note:

For QML usage start SDL as root:

```
su -c ./smartDeviceLinkCore
```

4.6.3 QNX host for both SDL and HTML5 HMI

1. Log in as a root and make sure you are not running Photon
2. QnxServ: download binary folder:
`scp -r <linux_host_user>@<linux_host_ip>:<Project_QNX_Build_WebHMI>/bin/* . /`
3. QnxServ: start SDL “`su -c ./smartDeviceLinkCore`”.

Note:

For USB transport usage start SDL as root:

```
su -c ./smartDeviceLinkCore
```

4. LinHost: start Chrome:
`chromium-browser '<Project_Root_Src>/src/components/HMI/index.html'`

Note:

In case there is no connection with HTML5 HMI probably it is necessary to add QNX Server IP to "no_proxy" config.

4.6.4 Run SDL on QNX Server and QML HMI on Linux host

1. LinHost: make “`<Project_QNX_Build_Qt>/bin/start_hmi.sh`” file empty.
2. Log in as a root and make sure you are not running Photon.
3. QnxServ: download binary folder:
`scp -r <linux_host_user>@<linux_host_ip>:<Project_QNX_Build_Qt>/bin/* . /`
4. QnxServ: start SDL:
`su -c ./smartDeviceLinkCore`

Note:

For USB transport usage start SDL as root:

```
su -c ./smartDeviceLinkCore
```

Note:

If SDL unexpectedly terminates (with any error) check that real IP of QNX Server is the same to IP in “`/usr/etc/dbus-1/session.conf`”. If not, perform steps 2-4 and restart QNX server.

5. LinHost: goto “`<Project_Linux_Build_Qt>/bin`” folder and execute the command:
`<QT_HOME_DIR>/qmlscene -I ./hmi/plugins ./hmi/MainWindow.qml`

5 Audio/Video Streaming over SDL

5.1 PulseAudio and GStreamer libraries: how to include or exclude from project

5.1.1 Short overview

PulseAudio library enables streaming audio through BlueTooth A2DP over SDL.
GStreamer library enables audio capturing from the HU's microphone over SDL.

5.1.2 Include

To have SDL project with both libraries included, `EXTENDED_MEDIA_MODE` flag must be set to `ON` when building the project (see section '*4.5 Build SDL from source on Linux Host* '):

```
-DEXTENDED_MEDIA_MODE=ON
```

5.1.3 Exclude

To have SDL project with both libraries excluded, `EXTENDED_MEDIA_MODE` flag must be set to `OFF` when building the project (see section '*4.5 Build SDL from source on Linux Host* '):

```
-DEXTENDED_MEDIA_MODE=OFF
```

5.2 Use SyncProxyTester to check audio/video streaming over SDL

1. Follow the appropriate instructions from chapter '*4 Build and Run SDL and Set Up the Environment*' to build SDL project (mind to set `EXTENDED_MEDIA_MODE` flag to `ON`).
2. Make sure SDL console logging is switched off (e.g. by directing them to file:
 `./smartDeviceLinkCore > SDL.log`), as it is a known fact that logging lowers the performance.
3. Start SyncProxyTester (SPT) of the latest version on mobile device:
 - Check 'Mobile Navi' box when starting SPT (as only navigation applications are allowed to stream audio/video over SDL)
 - Use USB or WiFi transports only (as BlueTooth is not the target transport for the feature).
4. Once SPT is registered, activate it on HMI (choose SPT in the HMI list of registered applications).
5. From SPT on mobile device perform the following:
 - For audio streaming:
 - Check the 'Turn On Audio Service' box (unchecked by default): for SPT to apply for starting audio service. Once SPT receives SDL's acknowledgement on audio service started, it enables the 'Start Streaming' button.
 - Press 'Start Streaming' button.
 - For video streaming:
 - Check the 'Turn On Video Service' box (unchecked by default): for SPT to apply for starting video service. Once SPT receives SDL's acknowledgement on video service started, it enables the 'Start Streaming' button.
 - Press 'Start Streaming' button.
6. The audio and/or video streaming is performed and processed by SDL. Depending on configuration in `smartDeviceLink.ini` file, SDL may forward the received stream to file/ through socket/ through pipe. Please see section 5.2 for details.

5.3 Configuring audio/video streaming parameters in `smartDeviceLink.ini` file

The below describes the possible configurations in `smartDeviceLink.ini` file for performing audio/video streaming from SDL to HMI. All the updates in this file might be done after the project is built.

1. `smartDeviceLink.ini` file is stored in the same folder where the project executable is.

2. The name of the file where audio/video stream is saved to (see p.1.6):

[MEDIA MANAGER]

```
...
VideoStreamFile = video_stream_file //the value after "=" might be changed
AudioStreamFile = audio_stream_file //the value after "=" might be changed
...
```

3. Choosing socket or pipe:

- The default values (set when the project is built):

[MEDIA MANAGER]

```
...
;VideoStreamConsumer = socket //video streaming through socket is commented
;AudioStreamConsumer = socket //audio streaming through socket is commented
;VideoStreamConsumer = file //video streaming to file is commented
;AudioStreamConsumer = file //audio streaming to file is commented
VideoStreamConsumer = pipe //video streaming is performed through pipe
AudioStreamConsumer = pipe //audio streaming is performed through pipe
...
```

- The updates may look like the following:

[MEDIA MANAGER]

```
...
VideoStreamConsumer = socket //video streaming is performed through socket
AudioStreamConsumer = socket //audio streaming is performed through socket
;VideoStreamConsumer = pipe //video streaming through pipe is commented
;AudioStreamConsumer = pipe //audio streaming through pipe is commented
...
```

4. Pipe name:

[MEDIA MANAGER]

```
...
;Temp solution: if you change NamedPipePath also change path to pipe in
src/components/qt_hmi/qml_model_qtXX/views/SDLNavi.qml
NamedVideoPipePath = /tmp/video_stream_pipe //the value after "=" might be changed
NamedAudioPipePath = /tmp/audio_stream_pipe //the value after "=" might be changed
...
```

5. Socket port:

[HMI]

```
...
VideoStreamingPort = 5050 //the value after "=" might be changed
AudioStreamingPort = 5080 //the value after "=" might be changed
```

II SDL-HMI Integration Guide

1 Overview

1.1 Operation System

SDL is installed to vehicle HU operation system, which might be:

- Linux Ubuntu
- QNX
- Any POSIX-compliant system.

1.2 Transports Supported

SDL supports the following transports for communication with HMI:

- 1) WebSocket (more details please find in section 3)
- 2) D-Bus (more details please find in section 4)

To communicate with SDL over one of the above transports, the Transport Adapter needs to be created on HMI side with the responsibilities to:

- Establish a connection with SDL
- Send/receive messages of corresponding to chosen transport format
- Translate SDL commands to format/calls understandable by HMI components.

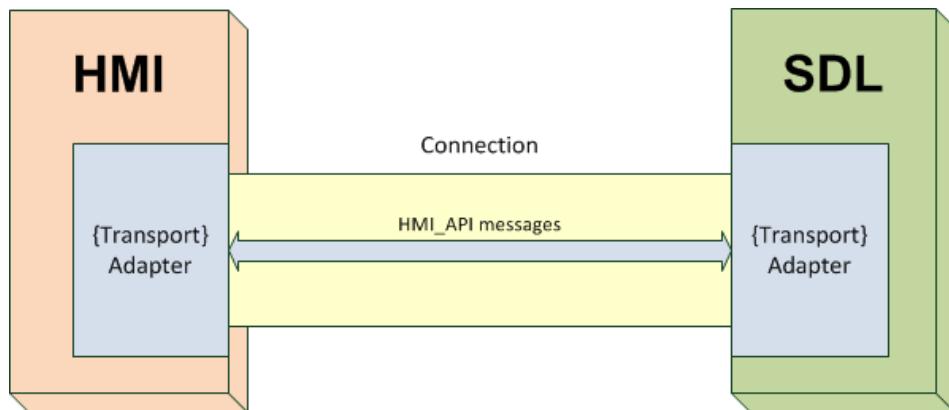


Figure 2.1 SDL-HMI Transport Adapters ({Transport} is the specific transport type).

The Transport Adapters of both WebSocket and D-Bus types are implemented in SDL.

For HMI it is enough to create one Adapter of the transport type chosen for communication.

Still, if there is a necessity, it is possible to expand the SDL with the Transport Adapter of another required type.

1.3 Communication

There are several requirements to HMI for performing the successful communication with SDL:

- 1) Establish a connection
- 2) Confirm its readiness to SDL
- 3) Correctly respond to the messages

2 WebSocket Transport

2.1 Connection Opening

2.1.1 Requirements to HMI adapter

Once decided to use WebSocket as a transport for communication between SDL and HMI, the below must be taken into consideration on HMI adapter creation.

HMI adapter must:

- Be installed on the same vehicle HU's OS where SDL is installed (Linux Ubuntu, QNX, or any POSIX-compliant system).
- Create and initialize the components named in correspondence to HMI_API (i.e. BasicCommunication, UI, Buttons, VR, TTS, Navigation, VehicleInfo).
- Establish a separate WebSocket connection with SDL for each of the above components (see 'Handshake' Example #1 and Example #3).
- To use the appropriate connection for sending requests/responses/notifications for the definite component (see Example #2).

2.1.2 Handshake

For opening a WebSocket connection, a handshake must be performed:

1. Client-Server Relations:
 - SDL is the Server (the one who confirms the WebSocket connection establishing)
 - HMI adapter is the Client (the one who requests the WebSocket connection opening).
2. Host:
 - When started and initialized SDL comes listening on:
127.0.0.1:8087
3. WebSocket Protocol:
 - [Version 13](#) is used.
4. Sequence of messages:
 - HMI adapter sends the handshake requests to SDL for every of the components (see Example #1 and Example #3). Each request must be sent:
 - To the mentioned host
 - With the individual Sec-WebSocket-Key
 - SDL provides the responses to every request with individual Sec-WebSocket-Accept.

2.1.3 Components registering

Once all the requested connections are opened, HMI adapter must send the JSON request for registering the component (see section 4.2 for JSON format details and Example #4 for registering example).

Here are the rules applicable to component registering messages only:

- "id" – a multiple of 100 integer value, e.g.:
 - 100 – for the first registering component
 - 200 – for the second one
 - 700 – for the seventh component
- "jsonrpc" : "2.0" - constant field and value for all JSON messages between SDL and HMI adapter.
- "method" : "MB.registerComponent" – the request is assigned to SDL's MessageBroker (MB), where the component name will be associated with the socket ID. Further SDL will send messages related to the named component over the corresponding connection.
- "componentName" – the name of the component being registered. Must correspond to the appropriate component name described in the current Guidelines.

SDL provides the response (Example 4.2) with:

- "id" – the value taken from the corresponding request.
- "result" – value of id multiplied by 10. Should be treated by HMI adapter as the success of registering.

2.1.4 Examples

Example #1 – WebSocket Handshake Headers

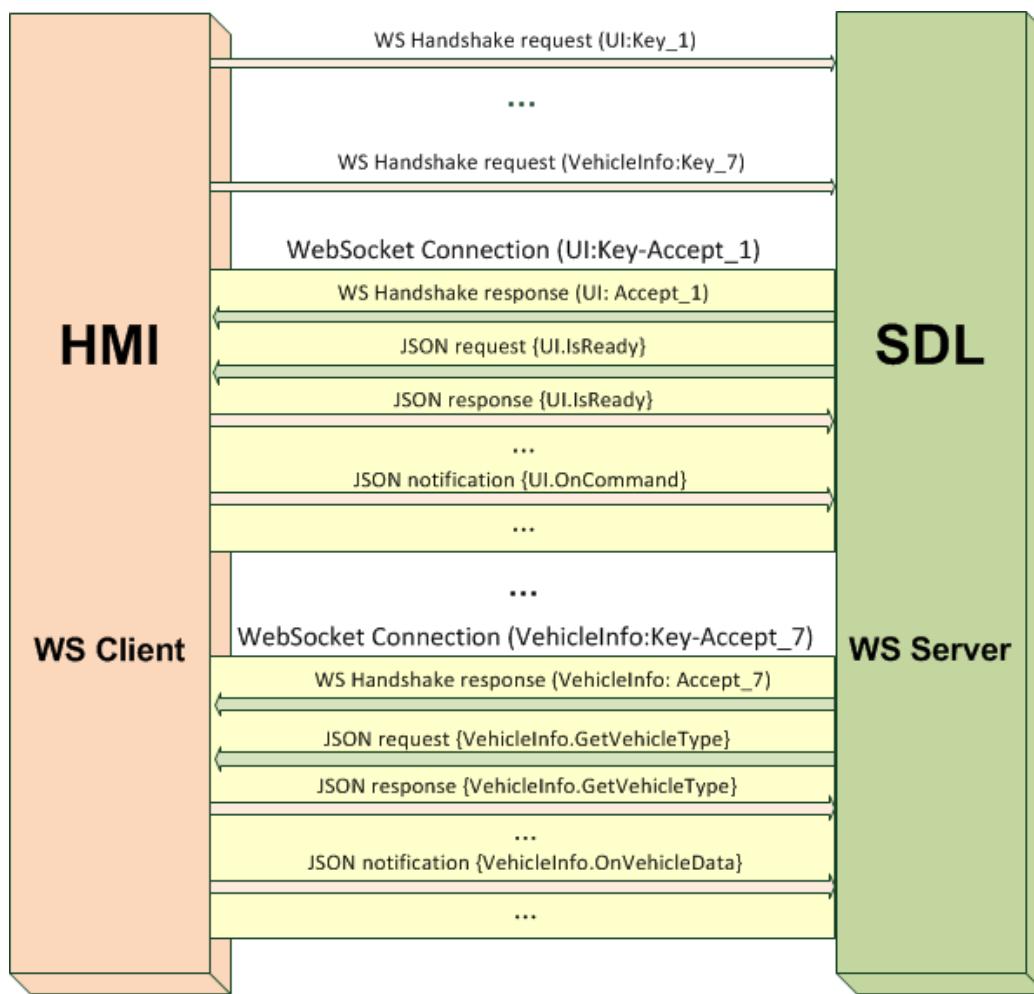
1.1 WS Handshake Request Header (sent from HMI)

```
GET / HTTP/1.1
Upgrade: websocket
Connection: Upgrade
Host: 127.0.0.1:8087
Origin: null
Pragma: no-cache
Cache-Control: no-cache
Sec-WebSocket-Key: b0atvDirYAt6OFFJ3DSGnw==
Sec-WebSocket-Version: 13
Sec-WebSocket-Extensions: x-webkit-deflate-frame
```

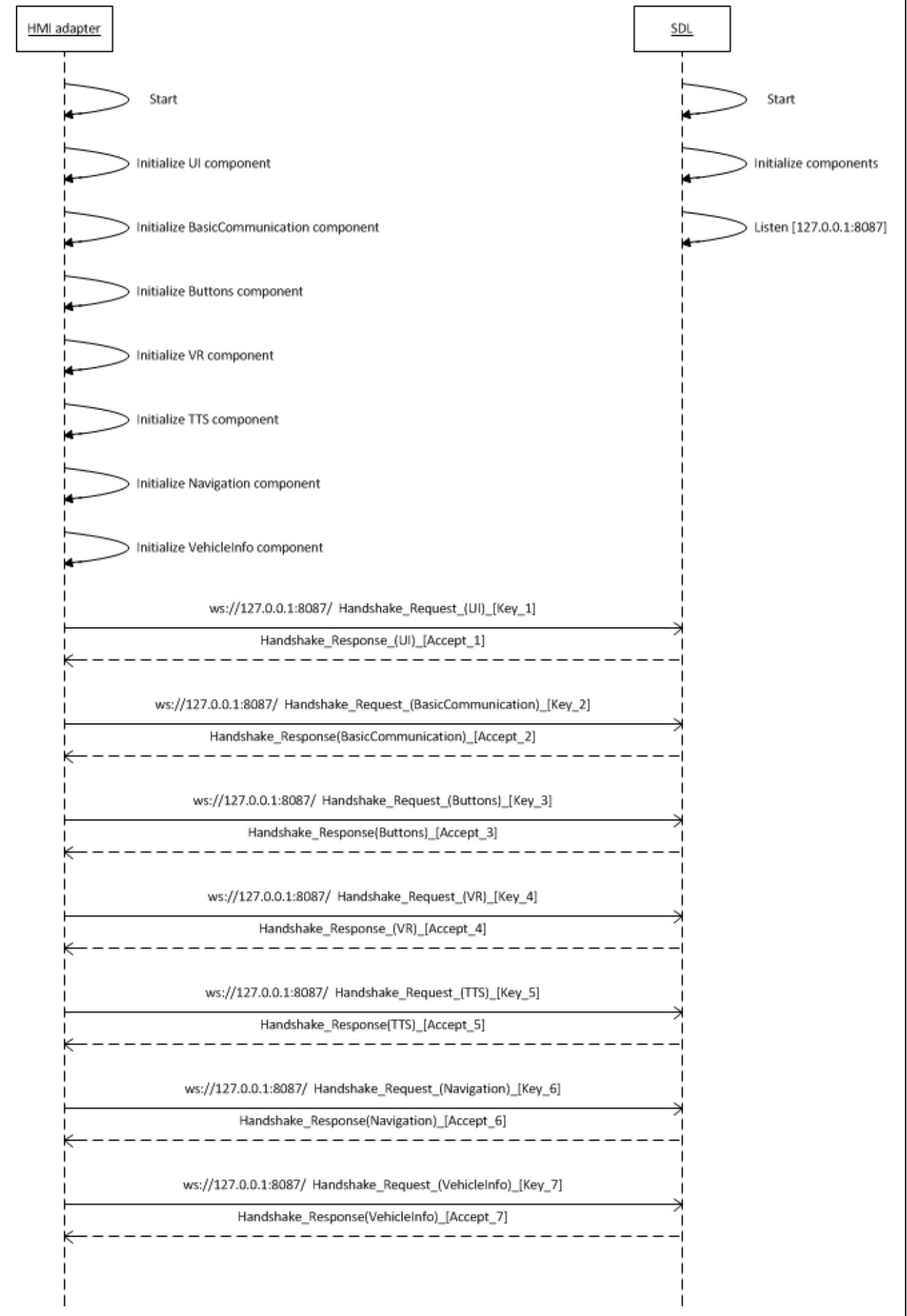
1.2 WS Handshake Response Header (sent from SDL)

```
Connection: Upgrade
Sec-WebSocket-Accept: h195GcH5yhZwfS16giEqe9HiYRk=
Upgrade: WebSocket
```

Example #2 – WebSocket Connection Diagram



Example #3 – WebSocket Handshake sequence diagram



Example #4 – Register Component JSON Messages

4.1 Request

```
{
    "id" : 700,
    "jsonrpc" : "2.0",
    "method" : "MB.registerComponent",
    "params" :
    {
        "componentName" : "VehicleInfo"
    }
}
```

4.2 Response

```
{
    "id" : 700,
    "jsonrpc" : "2.0",
    "result" : 7000
}
```

2.2 JSON Message Format

This chapter describes message structure for communication between SDL and HMI.
JSON RPC 2.0 format is taken as a basis.

Hereinafter within this chapter the actors for exchanging messages will be considered:

- The Client – the one who sends requests and notifications
- The Server – the one who provides the responses and sends notifications

Both SDL and HU components may act as a Client or as a Server.

2.2.1 Request

A RPC call is represented by sending a Request object to a Server. The Request object has the following members [1]:

| Member name | Description |
|--------------------|--|
| id | <p>An identifier established by the Client.</p> <p>This value MUST be of UNSIGNED INT type in the frames of communication between SDL and HU. This value should normally not be Null.</p> <p>If 'id' is not included the message is assumed to be a notification and should not be responded by the Server.</p> <p>(e.g. "id" : 65382).</p> |
| jsonrpc | <p>A String specifying the version of the JSON-RPC protocol.</p> <p>MUST be exactly "2.0":</p> <p>(e.g. "jsonrpc" : "2.0").</p> |
| method | <p>A String containing the information of the method to be invoked.</p> <p>The format is: "Component_Name.Method_Name"</p> <p>(e.g. "method" : "UI.IsReady").</p> |

| Member name | Description |
|-------------|---|
| params | A Structured value that holds the parameter values to be used during the invocation of the method. This member MAY be omitted. (e.g. "params" : { "cmdID" : 2318, "appID" : 409 }). |

The request might be sent with parameters or without any – please see the examples below.

2.2.1.1 Examples

Example #1 - A request with NO parameters:

```
{
    "id" : 125,
    "jsonrpc" : "2.0",
    "method" : "Buttons.GetCapabilities"
}
```

Example #2 - A request WITH parameters:

```
{
    "id" : 92,
    "jsonrpc" : "2.0",
    "method" : "UI.Alert",
    "params" :
    {
        "alertStrings" :
        [
            {
                "fieldName" : alertText1,
                "fieldText" : "WARNING"
            },
            {
                "fieldName" : alertText2,
                "fieldText" : "Hard weather conditions"
            }
        ],
        "duration" : 4000,
        "softButtons" :
        [
            {
                "type" : TEXT,
                "text" : "OK",
                "softButtonID" : 697,
                "systemAction" : STEAL_FOCUS
            },
            "appID" : 8218
        ]
    }
}
```

2.2.2 Notification

A Notification is a Request object without an ‘id’ member. For all the other members that must be included to notification please see the [section 4.1](#).

The Server must not reply to a Notification, so no Response object needs to be returned to the Client.

2.2.2.1 Examples

Example #1 - A notification with NO parameters:

```
{  
    "jsonrpc" : "2.0",  
    "method" : "UI.OnReady"  
}
```

Example #2 - A notification WITH parameters:

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnAppActivated",  
    "params" :  
    {  
        "appID" : 6578  
    }  
}  
  
{  
    "jsonrpc" : "2.0",  
    "method" : "Buttons.OnButtonPress",  
    "params" :  
    {  
        "mode" : "SHORT",  
        "name" : "OK"  
    }  
}
```

2.2.3 Response

On receipt of the Request message, the Server must reply with a Response (except for in the case of Notifications). The Response is expressed as a single JSON Object, with the following members

| Member name | Description |
|-------------|---|
| id | This member is REQUIRED. It MUST be the same as the value of the ' id ' member in the Request Object. If there was an error in detecting the id in the Request object (e.g. Parse error/Invalid Request), it MUST be Null. (e.g. " "id" : 65382"). |
| jsonrpc | A String specifying the version of the JSON-RPC protocol. MUST be exactly "2.0": (e.g. " jsonrpc " : "2.0"). |

| | |
|--------|--|
| result | <p>This member :</p> <ul style="list-style-type: none"> - is REQUIRED on success. - MUST NOT exist if there was an error invoking the method. <p>The 'result' field MUST contain:</p> <ul style="list-style-type: none"> - The 'method' field that has to include the same "Component_Name.Method_Name" as in originating Request. (e.g. "method" : "UI.IsReady"). - The 'code' field with '0' = SUCCESS value. No other result codes (described in the Result Enumeration in section 5.1.1) must be sent within the Response object. (e.g. "code" : 0). <p>This member MAY include the structured values determined by the method invoked on the Server. (e.g. "result" : <pre> { "available" : true, "code" : 0, "method" : "UI.IsReady" }).</pre></p> |
|--------|--|

2.2.3.1 Examples

Example #1 - A response with NO parameters (code and method are necessary):

```
{
    "id" : 167,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method": "UI.Alert"
    }
}
```

Example #2: - A response WITH extra parameters

```
{
  "id" : 125,
  "jsonrpc" : "2.0",
  "result" :
  {
    "capabilities" :
    [
      {
        "longPressAvailable" : true,
        "name" : "PRESET_0",
        "shortPressAvailable" : true,
        "upDownAvailable" : true
      },
      {
        "longPressAvailable" : true,
        "name" : "TUNEDOWN",
        "shortPressAvailable" : true,
        "upDownAvailable" : true
      }
    ],
    "presetBankCapabilities" :
    {
      "onScreenPresetsAvailable" : true
    },
    "code" : 0,
    "method" : "Buttons.GetCapabilities"
  }
}
```

2.2.4 Error

When a RPC call encounters an error, the Response Object MUST contain the ‘error’ member instead of the ‘result’ one. The error object has the following members:

| Member name | Description |
|-------------|---|
| id | <p>This member is REQUIRED.</p> <p>It MUST be the same as the value of the ‘id’ member in the Request Object.</p> <p>If there was an error in detecting the id in the Request object (e.g. Parse error/Invalid Request), it MUST be Null.</p> <p>(e.g. <code>"id" : 65382</code>).</p> |
| jsonrpc | <p>A String specifying the version of the JSON-RPC protocol.</p> <p>MUST be exactly “2.0”:</p> <p>(e.g. <code>"jsonrpc" : "2.0"</code>).</p> |

| | |
|-------|---|
| | <p>This member</p> <ul style="list-style-type: none"> - is REQUIRED on error. - MUST NOT exist if there was no error triggered during invocation. <p>The ‘error’ field MUST contain:</p> <ul style="list-style-type: none"> - The ‘code’ field with the value that indicates the error type that occurred. The result codes ‘1’ – ‘23’ from the Result Enumeration (described in section 5.1.1) should be used. (e.g. “code” : 14). - The ‘message’ field, containing the string that provides a short description of the error (from the same Result Enumeration). The message SHOULD be limited to a concise single sentence. (e.g. “message” : “There was a conflict with an already registered name”). - The ‘data’ field that MUST contain the ‘method’ member. The name in the form of “Component_Name.Method_Name” MUST be the same as in originating request. (e.g. “data” : { “method” : “UI.Alert” }). |
| error | |

2.2.4.1 Examples

Example #1 - Erroneous response

```
{
  "id" : 103,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : 13,
    "message" : "One of the provided IDs is not valid",
    "data" :
    {
      "method" : "VehicleInfo.GetDTCs"
    }
  }
}
```

4 D-Bus Transport

Hereinafter within this chapter the actors for exchanging messages will be considered:

- 1) The Client – the one who sends requests (Method call in D-Bus notation) and notifications (Signal in D-Bus notation)
- 2) The Server – the one who provides the responses (Method return in D-Bus notation) and sends notifications

SDL acts as a Client, HMI – as a Server.

4.1 Overview

D-Bus is an interprocess message exchange subsystem where communication between source and destination high level application objects is presented as a regular function call.

At sender's side at a low level such call is converted to message with service name, object name, interface name and function name. This message is posted into D-Bus subsystem.

At receiver's side at a low level this message is converted back to the appropriate function call.

4.2 HMI Integration Aspects

To communicate with SDL over D-Bus HMI must publish a service in D-Bus subsystem with "com.ford.sdl.hmi" name. This service must contain one object with "/" (root) name. This object shall provide interfaces according to this document. Each interface name shall start with "com.ford.sdl.hmi." prefix and provide functions according to API specification described below.

Example #1 - High level implementation of D-Bus initialization (at Server side, using Qt)

```
QDBusConnection::sessionBus().registerObject("/", this);
QDBusConnection::sessionBus().registerService("com.ford.sdl.hmi");
```

Any HMI that implements HMI API defined within this document uses optionality of arguments, meaning that some arguments may be present in the function call and others may be not. Such optionality is not supported by D-Bus. To resolve this, the special parameter is used for determining whether the argument is present:

```
{bool isPresent; valueType value}
```

- a) true isPresent – means the argument is present within function call
- b) false isPresent – means the argument is not present.

Example #2 – IsPresent parameter

```
{isPresent = true; appID = 123}
{isPresent = false; customPresets = ["ab", "cd"]}
```

This special type is mapped to D-Bus "struct" type with one Boolean field and one user's type field – "(b<user_type>)" in D-Bus type notation. For instance let it be optional integer. Then D-Bus signature of such argument will look like "(bi)".

According to this document SDL Core emits a few notifications. For this purpose SDL core registers a service with "com.ford.sdl.core" name and with one object named "/" in D-Bus subsystem. HMI must subscribe to those notifications by means of D-Bus subsystem (taking into account service name, object name and notification name).

Example #3 - Low level implementation of D-Bus initialization (at Client side)

```

/***
 * Well-known bus types. See dbus_bus_get().
 */
typedef enum
{
    DBUS_BUS_SESSION,      /**< The login session bus */
    DBUS_BUS_SYSTEM,       /**< The systemwide bus */
    DBUS_BUS_STARTER       /**< The bus that started us, if any */
} DBusBusType;
...
//init error var
dbus_error_init(&err);
//init connection in terms of D-Bus
DBusConnection* conn_ = dbus_bus_get(DBUS_BUS_SESSION, &err);
int ret = dbus_bus_request_name(conn_,
                                "com.ford.sdl.core",
                                DBUS_NAME_FLAG_DO_NOT_QUEUE,
                                &err);

```

SDL core expects that HMI supports the following behavior:

- A request to call the remote function can be synchronous or asynchronous
- All data transfer is performed through function's arguments
- The successfully executed function returns a common (successful) response with “success” return code
- The failed call returns the successful response with “error” return code
- The notification is distributed on definite event occurred.

4.3 Low Level D-Bus processing

Simplified code snippets are presented below.

Example #4 - Low level implementation of remote function call at Client side

```

DBusMessage* msg;
//create D-Bus message
msg = dbus_message_new_method_call(
    hmi_service_name,                      // "com.ford.sdl.hmi"
    hmi_object_path,                       // "/"
    (hmi_service_name+"."+hmi_interface_name), // "com.ford.sdl.hmi.BasicCommunication"
    hmi_function_name);

//send message to D-Bus
dbus_uint32_t serial;
dbus_connection_send(conn_id, msg, &serial);

```

Example #5 - Low level implementation of remote function call at Server side

```

// D-Bus message reading infinite loop
while(true) {
    // get message from D-Bus
    DBusMessage* msg = dbus_connection_pop_message(conn_id);
    // get type of call
    switch (dbus_message_get_type(msg)) {
        case DBUS_MESSAGE_TYPE_METHOD_CALL:
            // if it is function call
            //get interface and function name
            std::string method = dbus_message_get_member(msg);
            std::string interface = dbus_message_get_interface(msg);

```

```
if (interface == "org.freedesktop.DBus.Introspectable"
    && method == "Introspect") {
    //call appropriate function at Server side
    Introspect(msg);
    return false;
}
...
}
```

5 Getting Started

5.1 Readiness confirming

To start communication with SDL HMI must notify about its readiness via BasicCommunication.OnReady notification.

By receipt of OnReady notification SDL starts checking the availability of HMI components via the chain of RPCs:

- UI.IsReady –the display availability
- VR.IsReady –the voice recognition module availability
- TTS.IsReady – the Text-To-Speech module availability
- Navigation.Isready – check whether the navigation engine is available
- VehicleInfo.IsReady – check whether the vehicle information can be collected and provided.

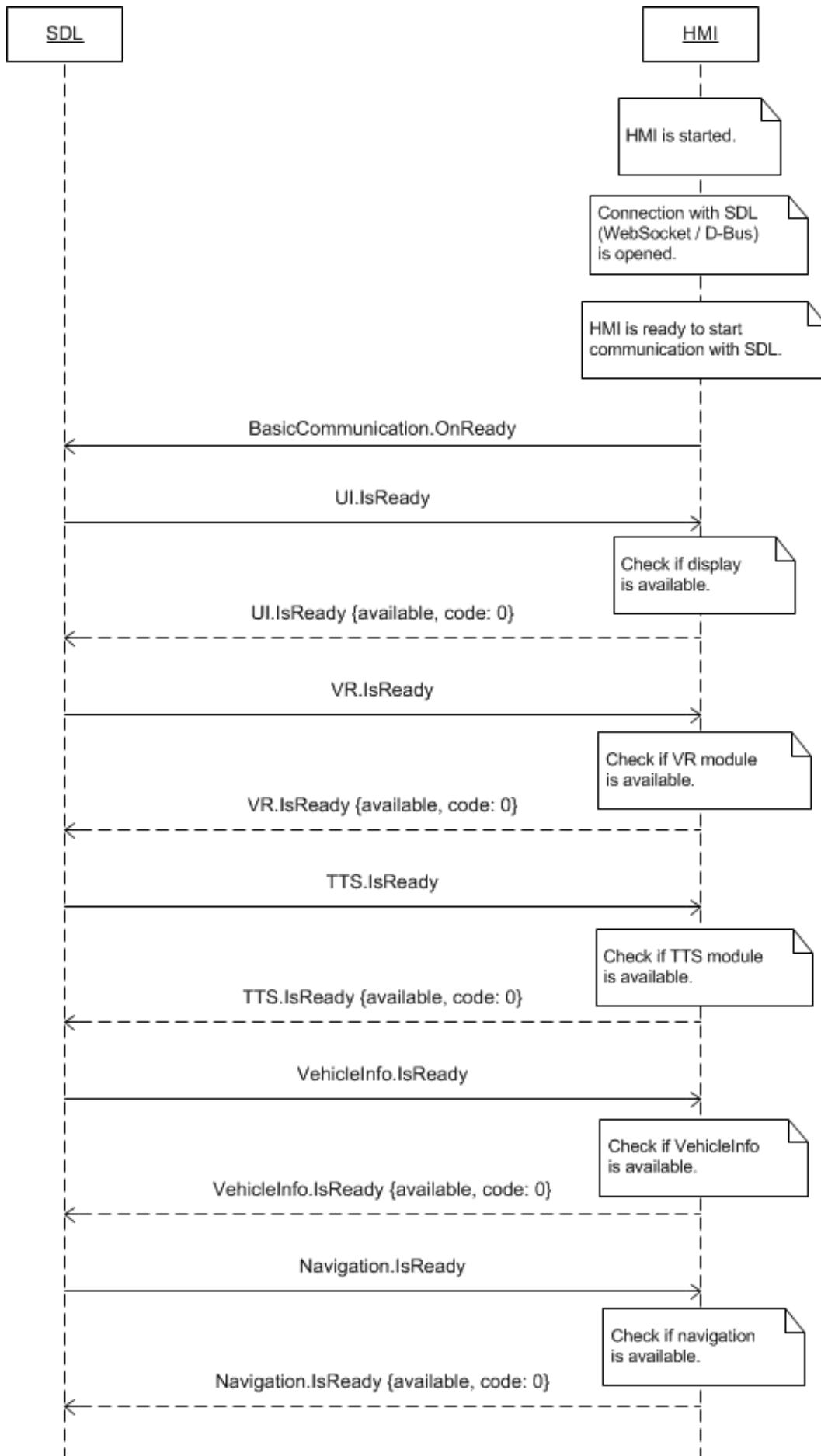
HMI must respond each of the above RPCs.

If the component is responded to be unavailable ("available": false), SDL will not further communicate with one.

5.1.1 Sequence diagram

Note:

In case of WebSocket connection, the RPCs to each of the components is sent within the separate WS connection



6 BasicCommunication Component Description

6.1 UpdateDeviceList

6.1.1 Description

| | |
|----------|-------------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Update HMI's list of found devices. |

The request comes after SDL has found a new device over one of available transports:

Notes about device searching:

With default SDL's Transport Manager (TM) and default Transport Adapters (TAs), SDL behaves in the following way:

Note:

Transport Manager as well as Transport Adapters may be customized or created from scratch, see chapter 3

1. BlueTooth transport:

- 1.1. SDL's TM makes a periodic search routine over BlueTooth.
- 1.2. Once the device is found, SDL starts the procedure of searching SDL-enabled applications on such device.
- 1.3. SDL sends BasicCommunication.UpdateDeviceList with the name and id of the discovered device to HMI.
- 1.4. SDL needs to receive OnDeviceChosen(deviceInfo) or OnFindApplications(deviceInfo) from HMI to allow registering applications running on this device.
- 1.5. After getting OnDeviceChosen(deviceInfo) or OnFindApplications(deviceInfo) from HMI SDL allows registering applications from the named device and sends BasicCommunication.OnAppRegistered to HMI.

2. USB transport:

- 2.1. SDL learns about new device connected over USB (notification from TA over TM).
- 2.2. SDL sends BasicCommunication.UpdateDeviceList with the name and id of the discovered device to HMI.
- 2.3. SDL sends BasicCommunication.OnAppRegistered to HMI right after (not waiting for OnDeviceChosen or OnFindApplications notifications).

3. WiFi transport:

- 3.1. SDL learns about new device connected over WiFi (notification from TA over TM).
- 3.2. SDL sends BasicCommunication.UpdateDeviceList with the name and id of the discovered device to HMI.
- 3.3. SDL sends BasicCommunication.OnAppRegistered to HMI right after (not waiting for OnDeviceChosen or OnFindApplications notifications).

6.1.2 Request

6.1.2.1 Behavior

HMI must:

1. Update its list of found devices (that is, store the provided information).
2. Use this information later when providing the OnDeviceChosen or OnFindApplications notifications.
3. Provide the User with the possibility of choosing among the found devices through either display or voice recognition or both.

6.1.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|-------------------|-----------|--|--|
| deviceList | Common.DeviceInfo | true | Array = true minsize = 0 maxsize = 100 | The list of devices (name and ID) found. If contains the empty array, it means that all the devices have been disconnected or are not connected yet. See DeviceInfo. |

6.1.2.3 DeviceInfo Structure

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|--|
| name | String | true | The name of the device. |
| id | Integer | true | The ID of the device. It remains unique during the ignition cycle. |

6.1.3 Response

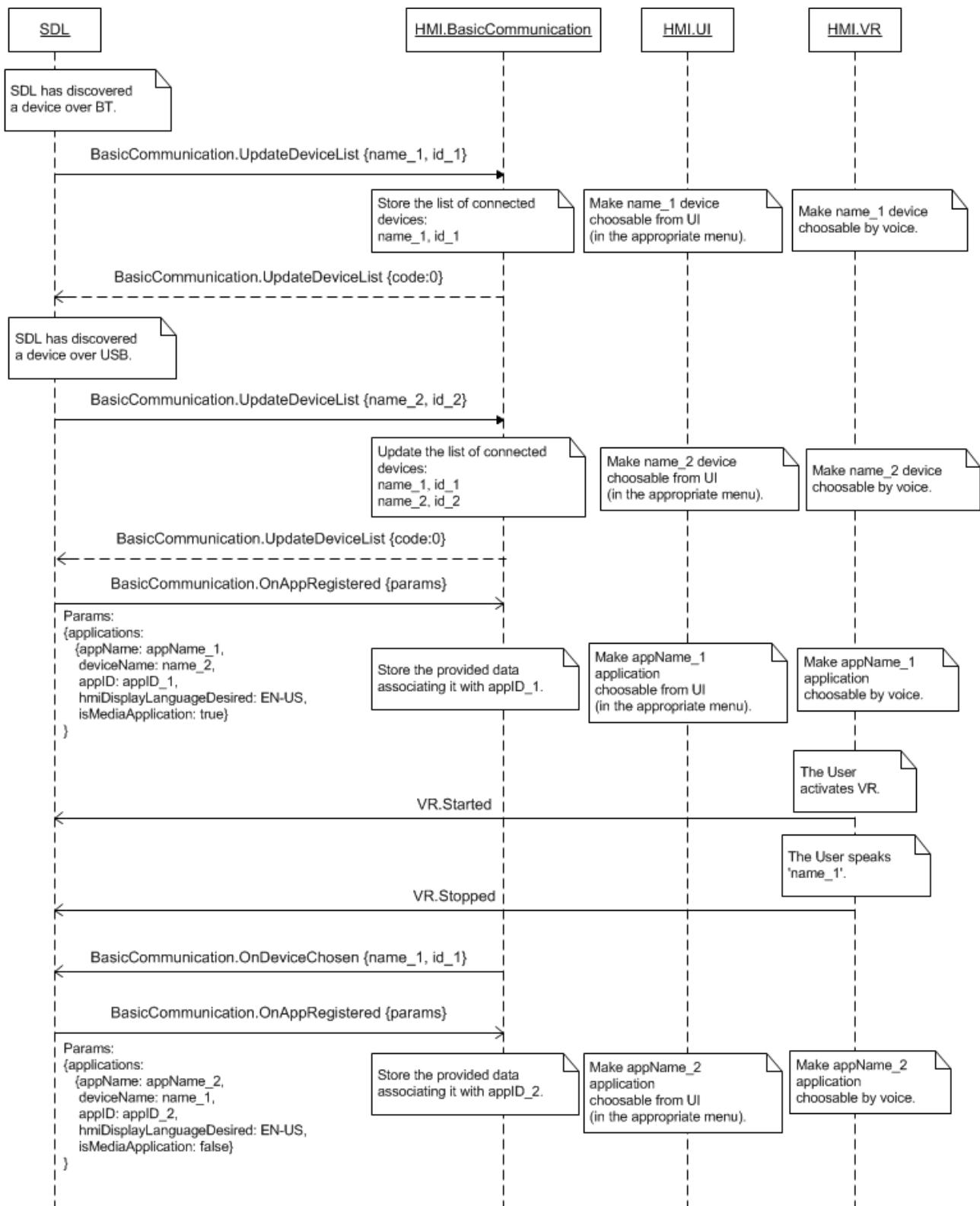
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|------------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI has stored the list of found devices provided within request. | JSON response | Regular response | code: 0 | - |
| Failure | INVALID_DATA: The data sent is invalid: | JSON error message | Regular response | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

6.1.4 Sequence Diagrams

6.1.4.1 UpdateDeviceList after SDL finds a new device over BT and USB



6.1.5 JSON Messages Examples

6.1.5.1 Request

```
{
    "id" : 64,
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.UpdateDeviceList",
    "params" :
    {
        "devicelist" :
        [

```

```
{
    {
        "name" : "Jerry's Phone",
        "id" : 3
    },
    {
        "name" : "XT910",
        "id" : 4
    }
}
}
```

6.1.5.2 Response

```
{
    "id" : 64,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "BasicCommunication.UpdateDeviceList"
    }
}
```

6.1.5.3 Error message

```
{
    "id" : 64,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "The data sent is invalid.",
        "data" :
        {
            "method" : "BasicCommunication.UpdateDeviceList"
        }
    }
}
```

6.1.6 D-Bus Messages Examples

6.1.6.1 Request

```
method call sender=:1.192 -> dest=com.ford.sdl.hmi serial=61 path=/;
interface=com.ford.sdl.hmi.BasicCommunication; member=UpdateDeviceList
array [
    struct {
        string "10.10.0.13"
        int32 1
    }
]
```

6.1.6.2 Response

```
method return sender=:1.193 -> dest=:1.192 reply_serial=61
    int32 0
    string ""
```

6.1.6.3 Failure

```
method return sender=:1.193 -> dest=:1.192 reply_serial=61
    int32 11
    string "The data sent is invalid."
```

6.2 ActivateApp

6.2.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Activate the named application on HMI. |

Activated application is assumed to receive access to head unit's display, audio and control systems/modules:

- supported by the head unit
- confirmed to be supported via responses to `IsReady` requests sent by SDL to every of components (i.e. UI, VR, TTS, Buttons, VehicleInfo, Navigation).

The request may follow:

- After SDL restores application's previous-to-ignition-off state on HMI.

Note:

By the time `ActivateApp` comes, HMI is already provided with the detailed information on every registered application with `OnAppRegistered` RPC.

6.2.2 Request

6.2.2.1 Behavior

HMI must:

1. Activate the named application on HMI:

- 1.1. Display the application-related screen.
 - 1.1.a. Display `UI.Show` related parameters associated with the named `appID` in case previously requested within ignition cycle..
 - 1.1.b. Display the corresponding template in case previously requested within ignition cycle by `UI.SetDispalyLayout` for the named application (`appID`).
 - 1.1.c. Apply `UI.SetGlobalProperties` associated with the named `appID` in case previously requested within ignition cycle.
 - 1.1.d. Apply `UI.AddCommand` associated with the named `appID` in case previously requested within ignition cycle.
 - 1.1.e. Apply `UI.AddSubMenu` associated with the named `appID` in case previously requested within ignition cycle.

- 1.2. Make VR commands accessible in case previously requested within ignition cycle by `VR.AddCommand` for the named `appID`.

- 1.3. Apply `TTS.SetGlobalProperties` associated with the named `appID` in case previously requested within ignition cycle.

2. Assign priority from the `priority` parameter to the named application. If request comes with omitted parameter, HMI must assign the priority of `NONE` by default.

3. Respond with `SUCCESS` result code right after HMI activated the named application. For all of applicable to this RPC result codes please see section 6.2.3. Response.

6.2.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|-----------------------|---------------------------------|--------------------|--|
| <code>appID</code> | <code>Integer</code> | <code>true</code> | The ID of the application requested to be activated. |
| <code>priority</code> | <code>Common.AppPriority</code> | <code>false</code> | Send to HMI so that it can coordinate order of requests/notifications correspondingly. |

| | | | |
|-------|-----------------|-------|---|
| level | Common.HMILevel | false | This parameter is omitted in case the app needs to be activated in FULL. The parameter is sent by SDL in case the app needs to be placed in one of LIMITED, BACKGROUND or NONE on HMI. |
|-------|-----------------|-------|---|

6.2.2.3 AppPriority

| Element name | Value | Short Description |
|---------------------|-------|-------------------|
| EMERGENCY | 0 | |
| NAVIGATION | 1 | |
| VOICE_COMMUNICATION | 2 | |
| COMMUNICATION | 3 | |
| NORMAL | 4 | |
| NONE | 5 | |

6.2.2.4 HMILevel

| Element name | Value | Short Description |
|--------------|-------|--|
| FULL | 0 | App must be activated to FULL level, that is the app must be granted the access to all of HMI supported resources: UI, VR, TTS, audio. |
| LIMITED | 1 | App must be activated to LIMITED level, that is the app must be granted the access to audio only. |
| BACKGROUND | 2 | App must be activated to LIMITED level, that is the app must be listed in corresponding HMI menu, have no acces to HMI supported resources, allowed to call Alerts only. |
| NONE | 3 | App must be activated to LIMITED level, that is the app must be listed in corresponding HMI menu, have no acces to HMI supported resources. |

6.2.3 Response

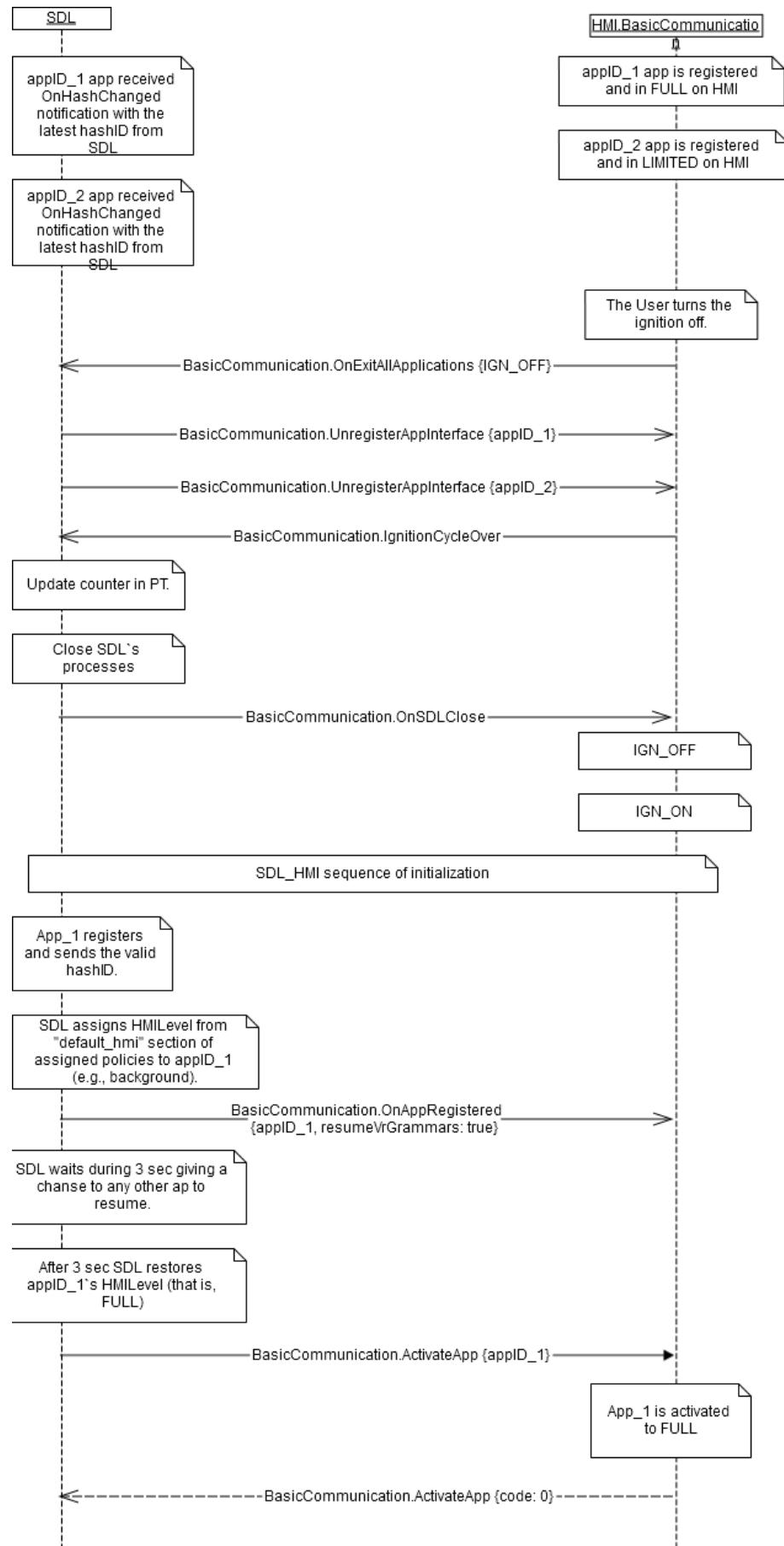
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

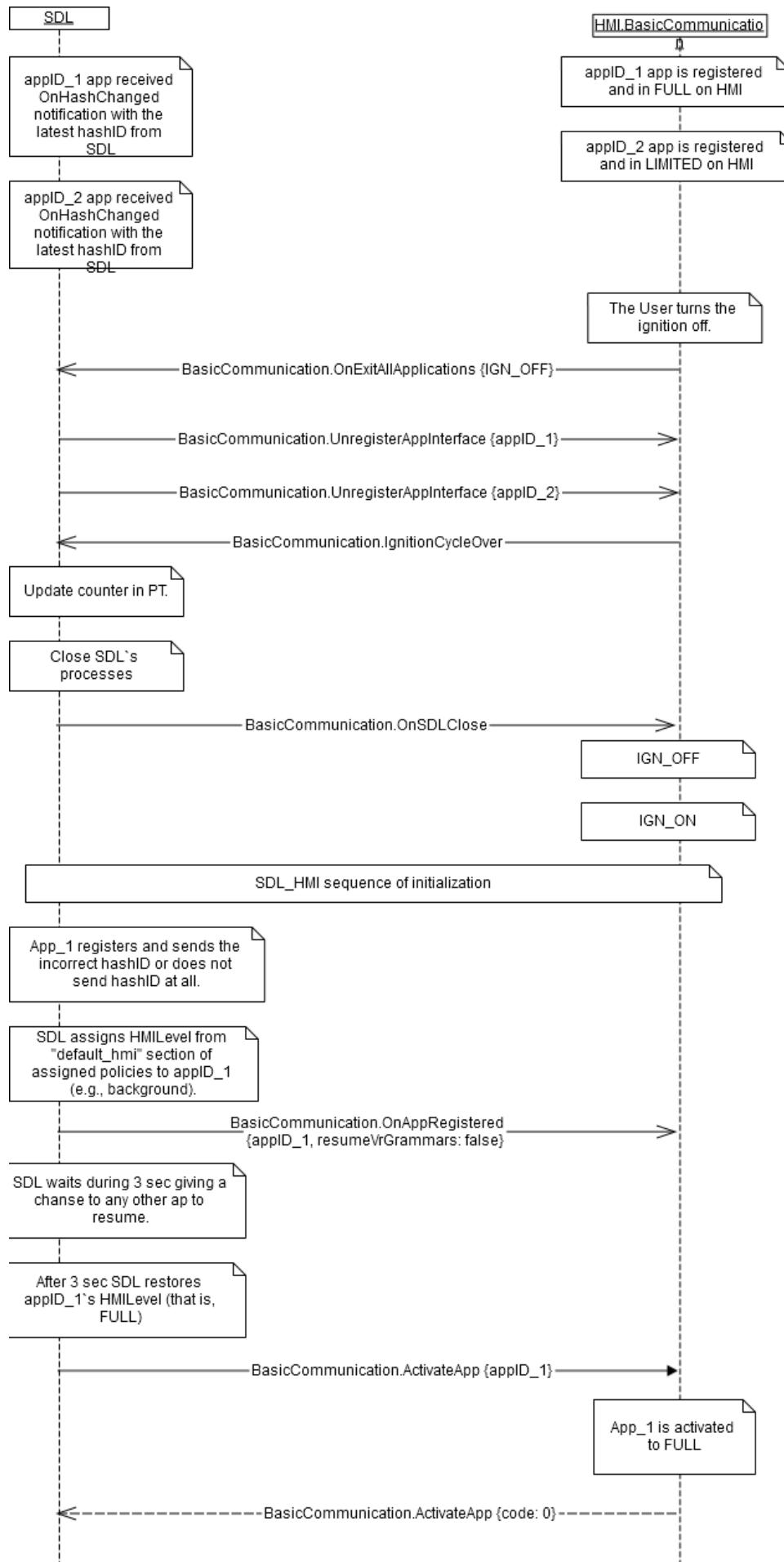
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|------------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI has activated the requested application (tsep 1. Of sub-section 6.2.2.1. is performed successfully).. | JSON response | Regular response | code: 0 | - |
| Failure | IGNORED: The named application is already acitvie on HMI: | JSON error message | Regular response | code: 6 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes |
| | APPLICATION_NOT_REGISTERED: HMI does not have the named application ID in its list of registered applications (that is, there were no OnAppRegistered with such applID from SDL previously). | | | code: 15 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

6.2.4 Sequence Diagrams

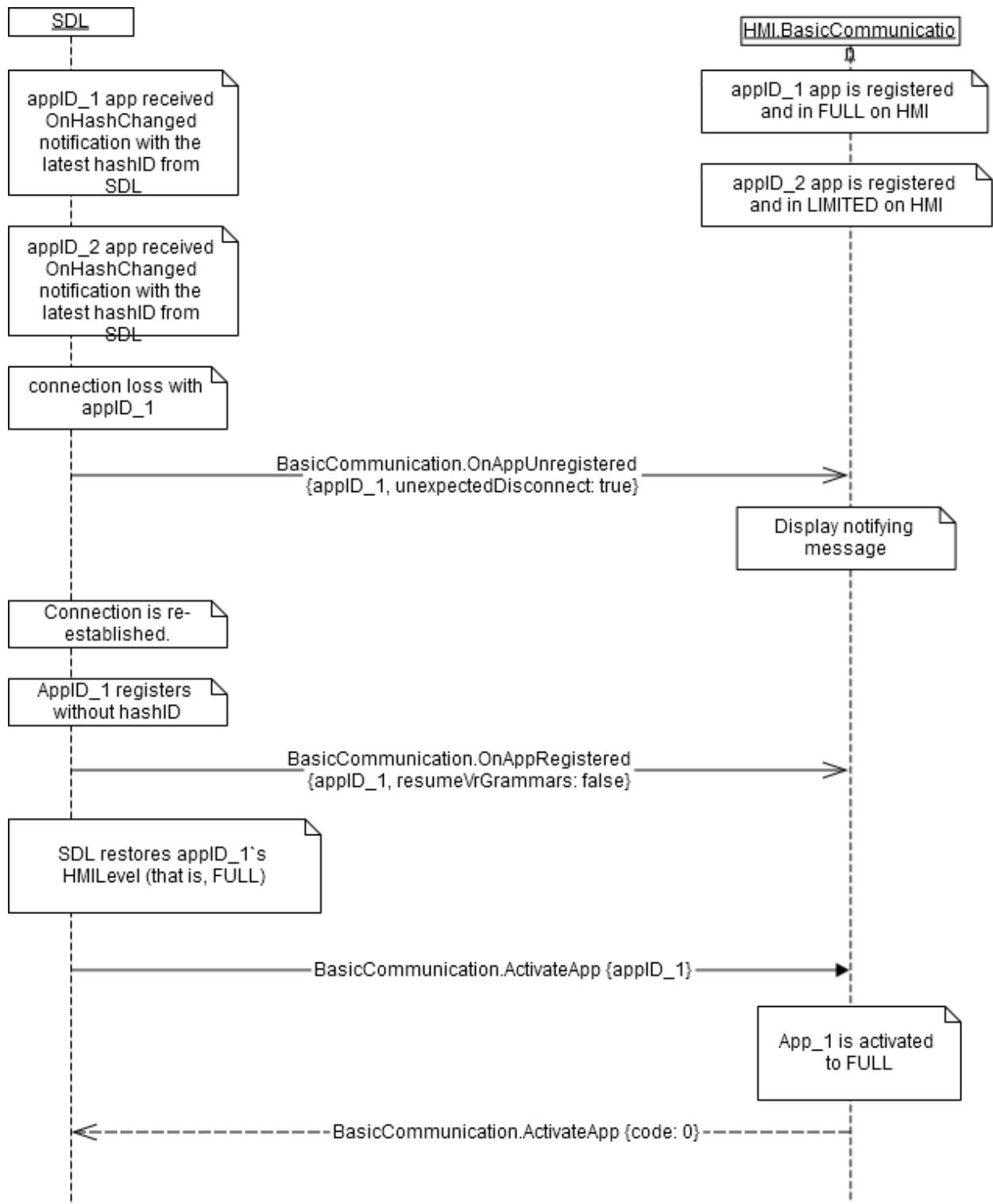
6.2.4.1 BasicCommunication.ActivateApp after successful resumption.



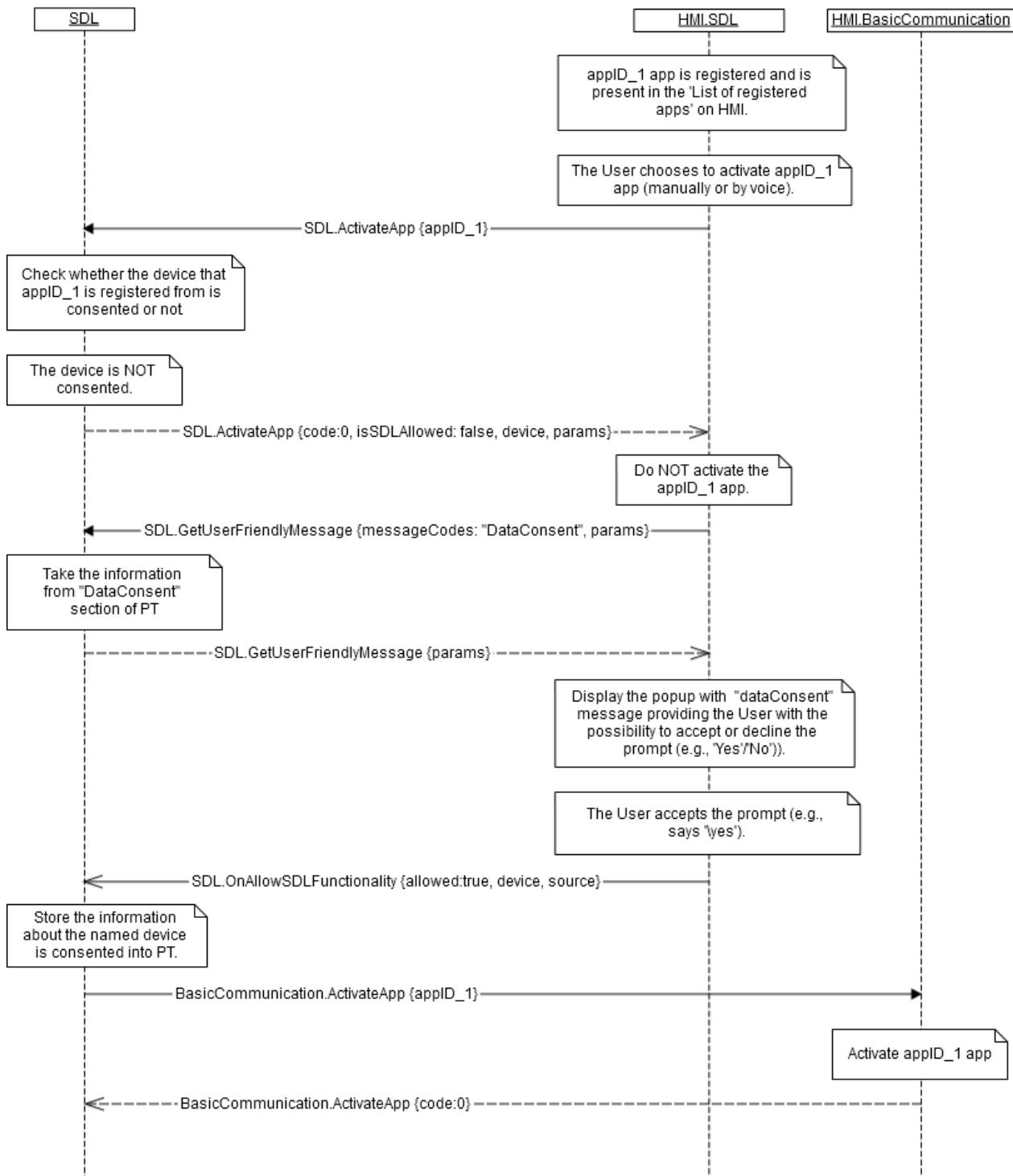
6.2.4.2 BasicCommunication.ActivateApp after UNsuccessful resumption.



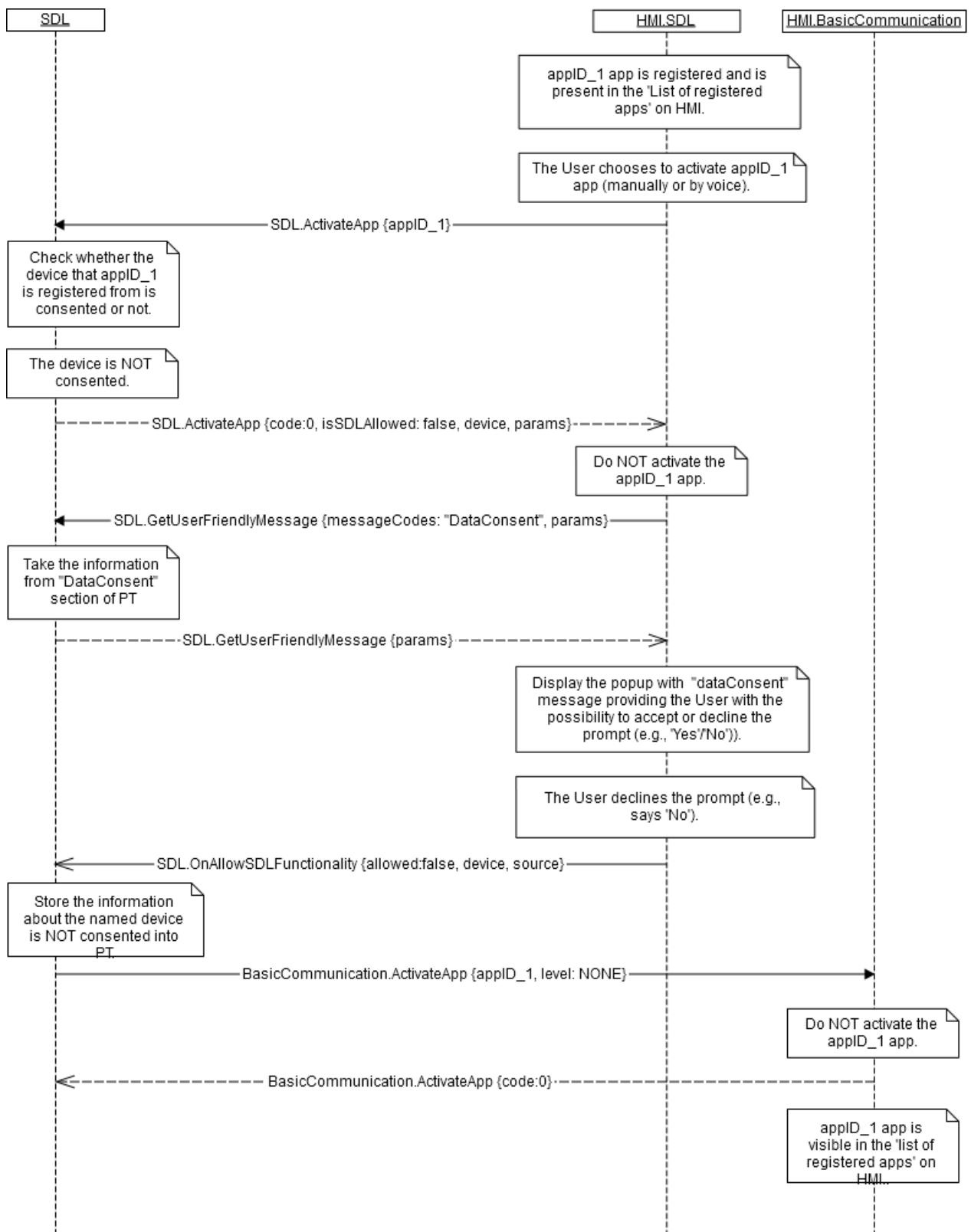
6.2.4.3 BasicCommunication.ActivateApp after unexpected disconnect



6.2.4.4 BasicCommunication.ActivateApp after the User accepted the data consent prompt



6.2.4.5 BasicCommunication.ActivateApp after the User declined the data consent prompt



6.2.5 JSON Messages Examples

6.2.5.1 Request

```
{
  "id" : 47,
  "jsonrpc" : "2.0",
  "method" : "BasicCommunication.ActivateApp",
  "result" :
  {
  }
```

```

        "appID" : 65368
    }
}
```

6.2.5.2 Response

```

{
    "id" : 47,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "BasicCommunication.ActivateApp"
    }
}
```

6.2.5.3 Error message

```

{
    "id" : 47,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 13,
        "message" : "One of the provided IDs is not valid.",
        "data" :
        {
            "method" : "BasicCommunication.ActivateApp"
        }
    }
}
```

6.2.6 D-Bus Messages Examples

6.2.6.1 Request

6.2.6.2 Response

6.2.6.3 Failure

6.3 MixingAudioSupported

6.3.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Find out if HMI supports mixing audio. |

'Mixing audio' names the ability of vehicle audio system to speak the TTS prompting or to listen to and recognize the VR command while playing audio.

Note:

SDL is able to get the information about 'mixing audio' capability of HMI from `smartDeviceLink.ini` file (this file is usually stored in the same folder where SDL executable is).

6.3.2 Request

6.3.2.1 Behavior

HMI must:

Check its mixing audio capabilities and provide the correct response.

Note:

If the request is NOT responded and there is no record in `smartDeviceLink.ini` file, SDL will consider that mixing audio is NOT supported.

6.3.3 Response

6.3.3.1 Behavior

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

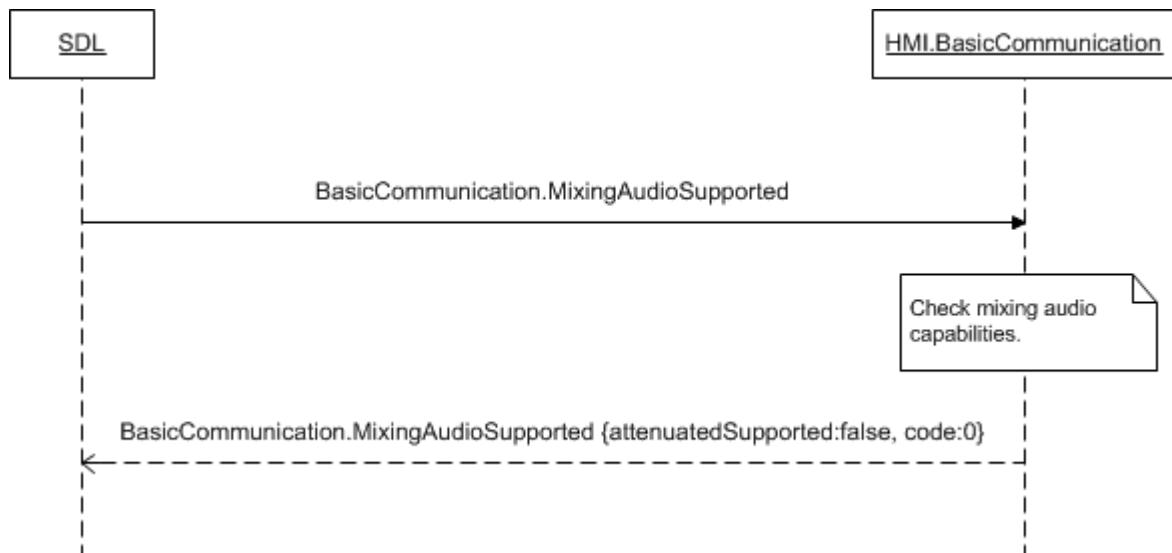
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|------------------|---------------------------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the information about 'mixing audio' supporting. | JSON response | Regular response | attenuatedSupported, code: 0 | - |
| Failure | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | JSON error message | Regular response | code: 22 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes |

6.3.3.2 Parameters

| Param Name | Type | Mandatory | Description |
|---------------------|---------|-----------|---|
| attenuatedSupported | Boolean | true | Must be - 'true' if HMI supports mixing audio - 'false' if not. |

6.3.4 Sequence Diagrams

6.3.4.1 MixingAudioSupported Messaging



6.3.5 JSON Messages Examples

6.3.5.1 Request

```
{  
    "id" : 27,  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.MixingAudioSupported"  
}
```

6.3.5.2 Response

```
{  
    "id" : 27,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "attenuatedSupported" : true,  
        "code" : 0,  
        "method" : "BasicCommunication.MixingAudioSupported"  
    }  
}
```

6.3.5.3 Error message

```
{  
    "id" : 27,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 22,  
        "message" : "An unknown error occurred",  
        "data" :  
        {  
            "method" : "BasicCommunication.MixingAudioSupported"  
        }  
    }  
}
```

6.3.6 D-Bus Messages Examples

6.3.6.1 Request

6.3.6.2 Response

6.3.6.3 Failure

6.4 AllowDeviceToConnect

6.4.1 Description

| | |
|---------|----------|
| Type: | Function |
| Sender: | SDL |

| | |
|-----------------|---------------------------------|
| Purpose: | Permit device to connect to HU. |
|-----------------|---------------------------------|

6.4.2 Request

6.4.2.1 Behavior

HMI must:

- Check in one of predefined ways whether to allow the named device to connect to HU.

HMI may:

- Request the User in one of accessible ways (VR, buttons, etc.)
- Have the definite settings to know whether to allow the connection.

6.4.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|------------|-------------------|-----------|--|
| deviceInfo | Common.DeviceInfo | true | The information about the device requested to be connected: its name and ID. See DeviceInfo . |

6.4.2.3 DeviceInfo Structure

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|--|
| name | String | true | The name of the device. |
| id | Integer | true | The ID of the device. It remains unique during the ignition cycle. |

6.4.3 Response

6.4.3.1 Behavior

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

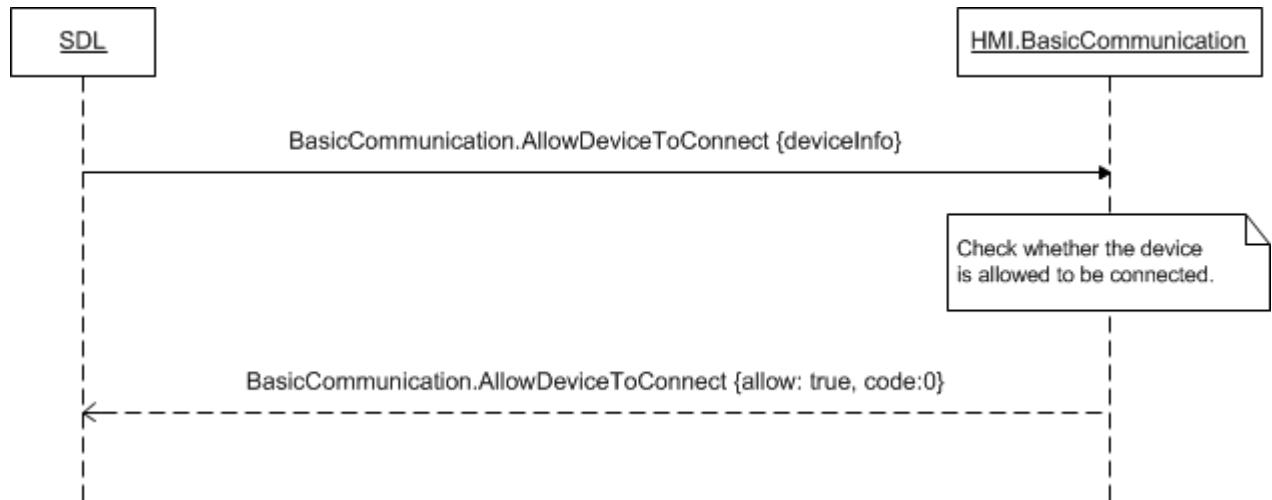
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|------------------|-------------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the information about whether the named device is allowed. | JSON response | Regular response | allow, code: 0 | - |
| Failure | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | JSON error message | Regular response | code: 22 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes |

6.4.3.2 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|--|
| allow | Boolean | true | Must be - 'true' if connecting the requested device is allowed - 'false' if not. |

6.4.4 Sequence Diagrams

6.4.4.1 AllowDeviceToConnect Messaging



6.4.5 JSON Messages Examples

6.4.5.1 Request

```
{
    "id" : 87,
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.AllowDeviceToConnect",
    "params" :
    [
        "deviceInfo" :
        {
            "name" : "Mary's Phone",
            "id" : 8
        }
    ]
}
```

6.4.5.2 Response

```
{
    "id" : 87,
    "jsonrpc" : "2.0",
    "result" :
    {
        "allow" : true,
        "code" : 0,
        "method" : "BasicCommunication.AllowDeviceToConnect"
    }
}
```

6.4.5.3 Error message

```
{
    "id" : 87,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 22,
        "message" : "An unknown error occurred",
        "data" :
        {
            "method" : "BasicCommunication.AllowDeviceToConnect"
        }
    }
}
```

```

    }
}
}
```

6.4.6 D-Bus Messages Examples

6.4.6.1 Request

| |
|--|
| |
|--|

6.4.6.2 Response

| |
|--|
| |
|--|

6.4.6.3 Failure

| |
|--|
| |
|--|

6.5 SystemRequest

6.5.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Provide the path to the system file that SDL has received from mobile application |

SDL sends BasicCommunication.SystemRequest to HMI in case SDL receives SystemRequest RPC from mobile application.

6.5.2 Request

6.5.2.1 Behavior

HMI must:

- Decrypt the fileName file

Send OnReceivedPolicyUpdate to SDL right after the fileName file is successfully decrypted.

HMI may:

6.5.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-------------|--------------------|-----------|----------------------------------|---|
| requestType | Common.RequestType | true | - | The type of system request. |
| fileName | String | true | minlength = 1 maxlength = 255 | The path to the file that SDL stored .in predefined in smartDeviceLink.ini file folder. |
| appID | String | false | minlength = 1 maxlength = 50 | ID of application that sent the file. |

6.5.2.2. RequestType

| Element name | Value | Short Description |
|--------------|-------|-------------------|
| HTTP | 0 | |
| FILE_RESUME | 1 | |
| AUTH_REQUEST | 2 | |

| Element name | Value | Short Description |
|----------------|-------|-------------------|
| AUTH_CHALLENGE | 3 | |
| AUTH_ACK | 4 | |
| PROPRIETARY | 5 | |

6.5.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|---------------------------|-------------------------|----------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI recognizes the path to the file and the corresponding request type.. | JSON response | Regular response | code: 0 | - |
| Failure | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | JSON <u>error message</u> | Regular <u>response</u> | code: 22 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes |

6.5.4 Sequence Diagrams

6.5.4.1 SystemRequest Messaging

6.5.5 JSON Messages Examples

6.5.5.1 Request

```
{
  "id" : 59,
  "jsonrpc" : "2.0",
  "method" : "BasicCommunication.SystemRequest"
  "params" :
  {
    "requestType" : FILE_RESUME,
    "fileName" : "/tmp/fs/mp/images/ivsu_cache/123.json",
    "appID" : "824587763458"
  }
}
```

6.5.5.2 Response

```
{
  "id" : 59,
  "jsonrpc" : "2.0",
  "result" :
  {
    "code" : 0,
    "method" : "BasicCommunication.SystemRequest"
  }
}
```

6.5.5.3 Error message

```
{
```

```

    "id" : 59,
    "jsonrpc" : "2.0",
    "error" :
    {
      "code" : 11,
      "message" : "Invalid data",
      "data" :
      {
        "method" : "BasicCommunication.SystemRequest"
      }
    }
}

```

6.5.6 D-Bus Messages Examples

6.5.6.1 Request

[REDACTED]

6.5.6.2 Response

[REDACTED]

6.5.6.3 Failure

[REDACTED]

6.6 PolicyUpdate

6.6.1 Description

| | |
|----------|----------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | |

From: SDL

To: SYNC Manager

Asks SYNC Manager to send given Policy Table snapshot to the backend

6.6.2 Request

6.6.2.1 Behavior

HMI must:

6.6.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|----------------------------------|-----------------------------------|
| file | String | true | minlength = 1 maxlength = 255 | Location of policy table snapshot |
| timeout | Integer | true | minvalue = 0 maxvalue = 65535 | Send attempt timeout in seconds |

| | | | | |
|-------|---------|------|--|---|
| retry | Integer | true | array = true minvalue = 0 maxvalue = 65535 minsize = 1 maxsize = 5 | Array of delays to wait after failed attempts |
|-------|---------|------|--|---|

6.6.3 Response

6.6.3.1 Behavior

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|---------------------------|-------------------------|-------------------|--|
| | | WebSocket | D-Bus | | |
| Success | The information is available and provided. | JSON response | Regular response | allow, Code: 0 | See section 6.6.3.2 |
| Failure | The information is not available or some failure occurred. | JSON <u>error message</u> | Regular <u>response</u> | Code: 1...23 | Result code must correspond to the failure occurred. |

6.6.4 Sequence Diagrams

6.6.4.1 AllowApp Messaging

6.6.5 JSON Messages Examples

6.6.5.1 Request

```
{
  "id" : 103,
  "jsonrpc" : "2.0",
  "method" : "BasicCommunication.AllowApp",
  "params" :
  {
    "app" :
    {
      "appName" : "Pandora",
      "icon" : "tmp/SDL/app/Pandora/icon.jpeg",
      "deviceName" : "Jerry`s Phone",
      "appID" : 65537,
      "hmiDisplayLanguageDesired" : FR-FR,
      "isMediaApplication" : true
    },
    "appPermissions" : ["Streaming audio", "VR", "TTS"]
  }
}
```

6.6.5.2 Response

```
{
  "id" : 103,
  "jsonrpc" : "2.0",
  "result" :
  {
    "allowed" : true,
    "code" : 0,
    "method" : "BasicCommunication.AllowApp"
  }
}
```

6.6.5.3 Error message

```
{  
    "id" : 103,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 16,  
        "message" : "Wrong language: the requested language is currently not  
supported",  
        "data" :  
        {  
            "method" : "BasicCommunication.AllowApp"  
        }  
    }  
}
```

6.6.6 D-Bus Messages Examples

6.6.6.1 Request

6.6.6.2 Response

6.6.6.3 Failure

6.7 GetSystemInfo

6.7.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get information about system installed on HU.. |

Request from SDL to HMI to obtain information about head unit system.

6.7.2 Request

6.7.2.1 Behavior

HMI must:

6.7.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|------------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | The named application has been activated on HMI. | JSON response | Regular response | Code: 0 | - |
| Ignored | The requested application is currently active on HMI. | JSON error message | Regular response | Code: 6 | - |
| Failure | The named application is NOT active on HMI. | JSON error message | Regular response | Code: 1...23 | Result code must correspond to the failure occurred. |

6.7.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-----------------|-----------------|-----------|-----------------|--|
| ccpu_version | String | true | maxlength = 500 | Software version of the module |
| language | Common.Language | true | - | ISO 639-1 combined with ISO 3166 alpha-2 country code (i.e. en-us) |
| wersCountryCode | String | true | maxlength = 500 | Country code from the Ford system WERS (i.e.WAEGB). |

6.7.3.2 Language Enumeration

| Element name | Value | Short Description |
|--------------|-------|--------------------------------|
| AR-SA | 0 | Arabic – Saudi Arabia |
| CS-CZ | 1 | Czech – Czech Republic |
| DA-DK | 2 | Danish – Denmark |
| DE-DE | 3 | German – Germany |
| EN-AU | 4 | English – Australia |
| EN-GB | 5 | English – GB |
| EN-US | 6 | English – US |
| ES-ES | 7 | Spanish – Spain |
| ES-MX | 8 | Spanish – Mexico |
| FR-CA | 9 | French – Canada |
| FR-FR | 10 | French – France |
| IT-IT | 11 | Italian – Italy |
| JA-JP | 12 | Japanese – Japan |
| KO-KR | 13 | Korean – South Korea |
| NL-NL | 14 | Dutch (Standard) – Netherlands |
| NO-NO | 15 | Norwegian - Norway |
| PL-PL | 16 | Polish – Poland |
| PT-PT | 17 | Portuguese – Portugal |
| PT-BR | 18 | Portuguese – Brazil |
| RU-RU | 19 | Russian - Russia |
| SV-SE | 20 | Swedish – Sweden |
| TR-TR | 21 | Turkish – Turkey |
| ZH-CN | 22 | Mandarin – China |
| ZH-TW | 23 | Mandarin – Taiwan |

6.7.4 Sequence Diagrams

6.7.4.1 GetSystemInfo

6.7.4.2 GetSystemInfo

6.7.5 JSON Messages Examples

6.7.5.1 Request

```
{  
    "id" : 47,  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.GetSystemInfo"  
}
```

6.7.5.2 Response

```
{  
    "id" : 47,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "BasicCommunication.GetSystemInfo"  
    }  
}
```

6.7.5.3 Error message

```
{  
    "id" : 47,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 13,  
        "message" : "One of the provided IDs is not valid.",  
        "data" :  
        {  
            "method" : "BasicCommunication.GetSystemInfo"  
        }  
    }  
}
```

6.7.6 D-Bus Messages Examples

6.7.6.1 Request

6.7.6.2 Response

6.7.6.3 Failure

6.8 OnReady

6.8.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about readiness to communicate. |

This is the first message in SDL-HMI communication.

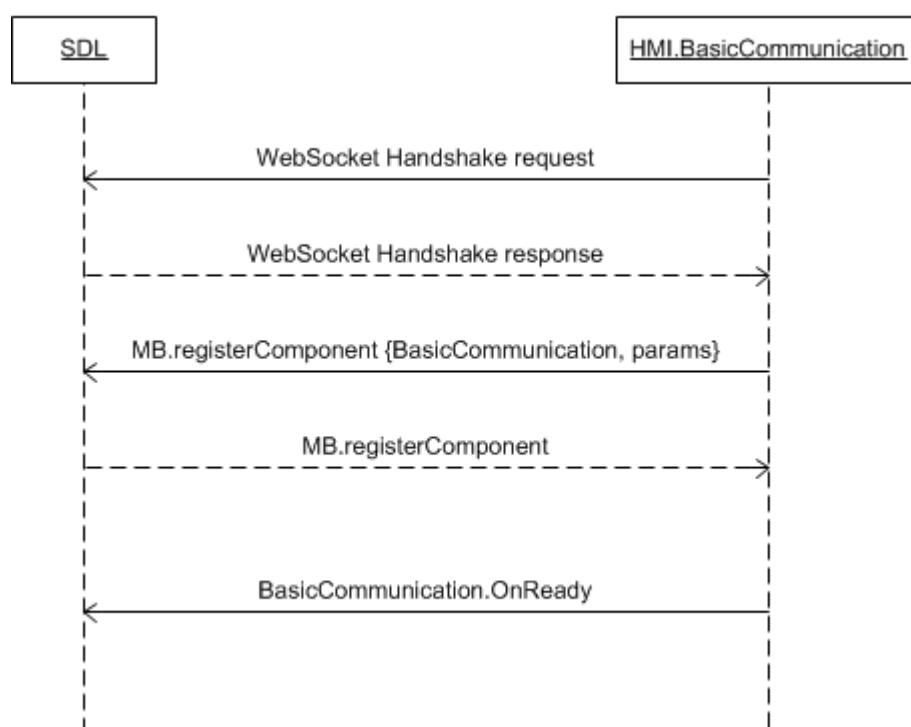
HMI must:

Send this notification after the connection is established and HMI is ready for communication:

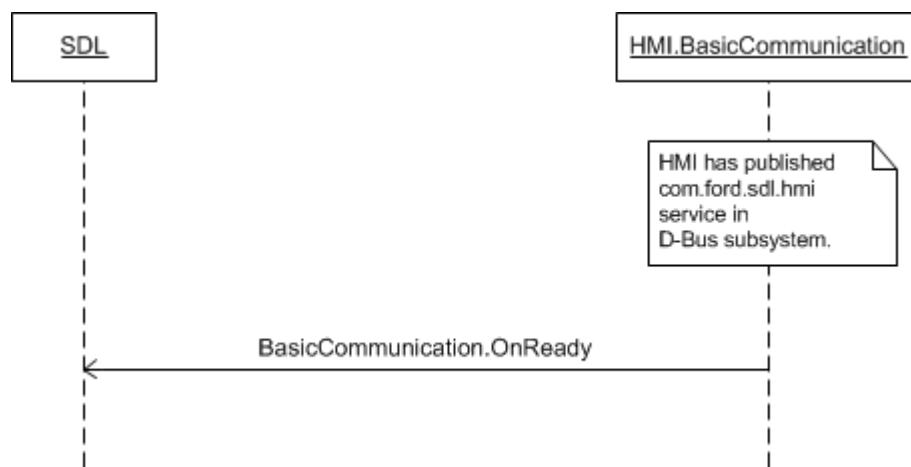
- For WebSocket connection: right after the handshake for HMI's BasicCommunication component and its registering with SDL (see [section 3.1.2](#) and [section 3.1.3](#)).
- For D-Bus connection: right after the service `com.ford.sdl.hmi` is published.

6.8.2 Sequence Diagrams

6.8.2.1 OnReady after WebSocket connection establishment



6.8.2.2 OnReady after D-Bus service publishing



6.8.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.OnReady"
}
```

6.8.4 D-Bus Messages Examples

6.8.6.1 Request

6.8.6.2 Response

6.8.6.3 Failure

6.9 OnStartDeviceDiscovery

6.9.1 Description

| | |
|----------|-------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Initiate device search. |

On receipt of this notification SDL starts searching for devices on all of available transports. Afterwards, SDL provides the search results via [UpdateDeviceList](#) RPC.

HMI must:

- Provide the User with the possibility to choose the device discovery in one of accessible ways (VR, buttons, etc.).
- Send `OnStartDeviceDiscovery` notification when the User has chosen to search for devices.

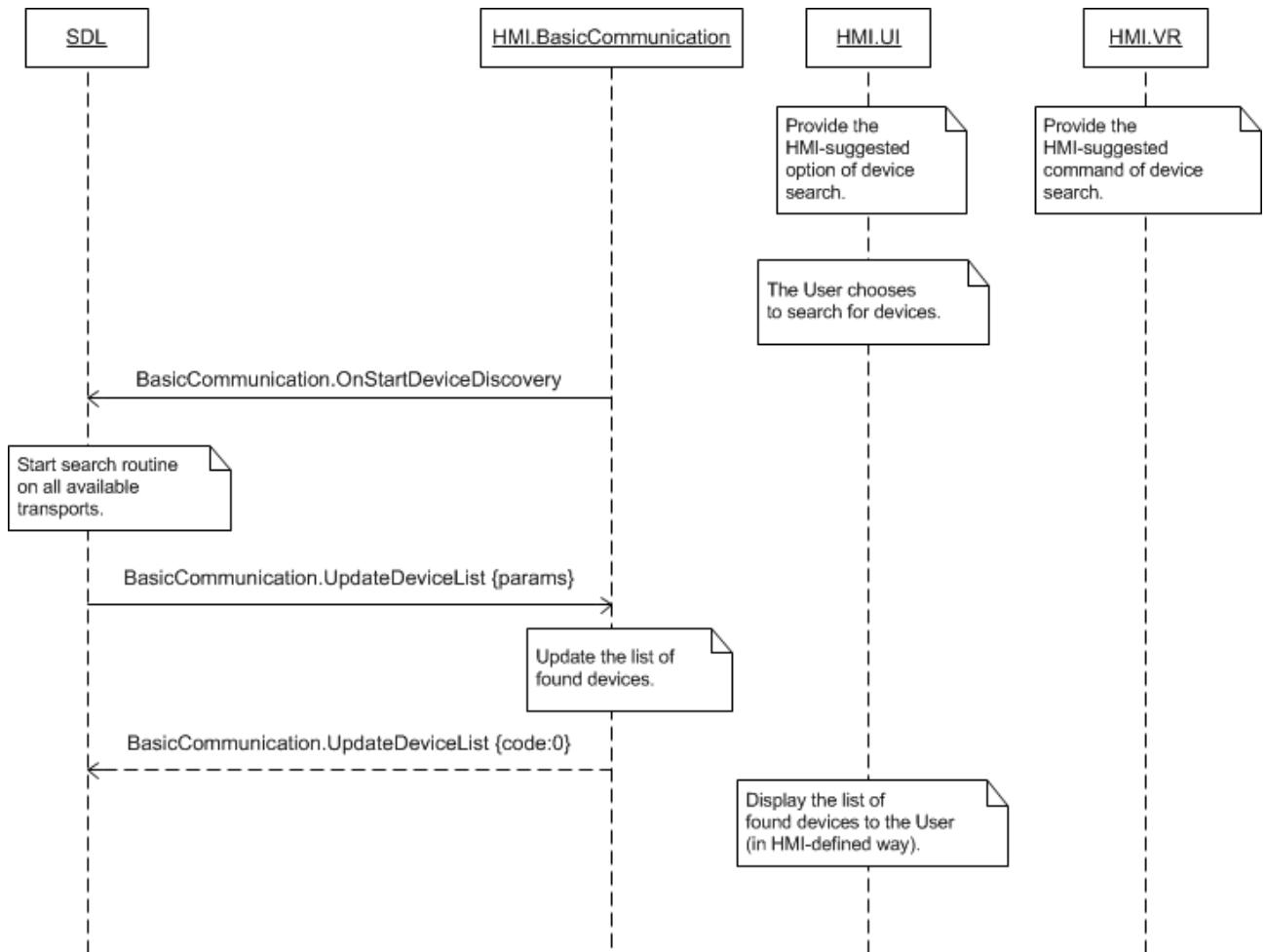
Note:

The difference:

- *OnStartDeviceDiscovery initiates SDL to start the new search procedure*
- *OnUpdateDeviceList results SDL to provide the updates of the recent search.*

6.9.2 Sequence Diagrams

6.9.2.1 *OnStartDeviceDiscovery upon User's request and resulting UpdateDeviceList*



6.9.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnStartDeviceDiscovery"  
}
```

6.9.4 D-Bus Messages Examples

```
method call sender=:1.167 -> dest=org.freedesktop.DBus serial=12 path=/org/freedesktop/DBus;
interface=org.freedesktop.DBus; member=AddMatch
    string "type='signal', sender='com.ford.sdl.hmi', path='/', interface='com.ford.sdl.hmi.BasicCommunication',
member='OnStartDeviceDiscovery'"
```

6.10 OnDeviceChosen

6.10.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Open the connection with the named device. |

On receipt of this notification SDL opens the connection with the named device and starts session with the application(s) running on it.

HMI must:

- Provide the User with the possibility to choose among the list of found devices in one of accessible ways (VR, buttons, etc.).
- Send `OnDeviceChosen` notification when the User has chosen the device.

Note:

The list of found devices (that HMI may provide to the User for choosing) is provided by SDL within `UpdateDeviceList` request.

6.10.1.1 Parameters

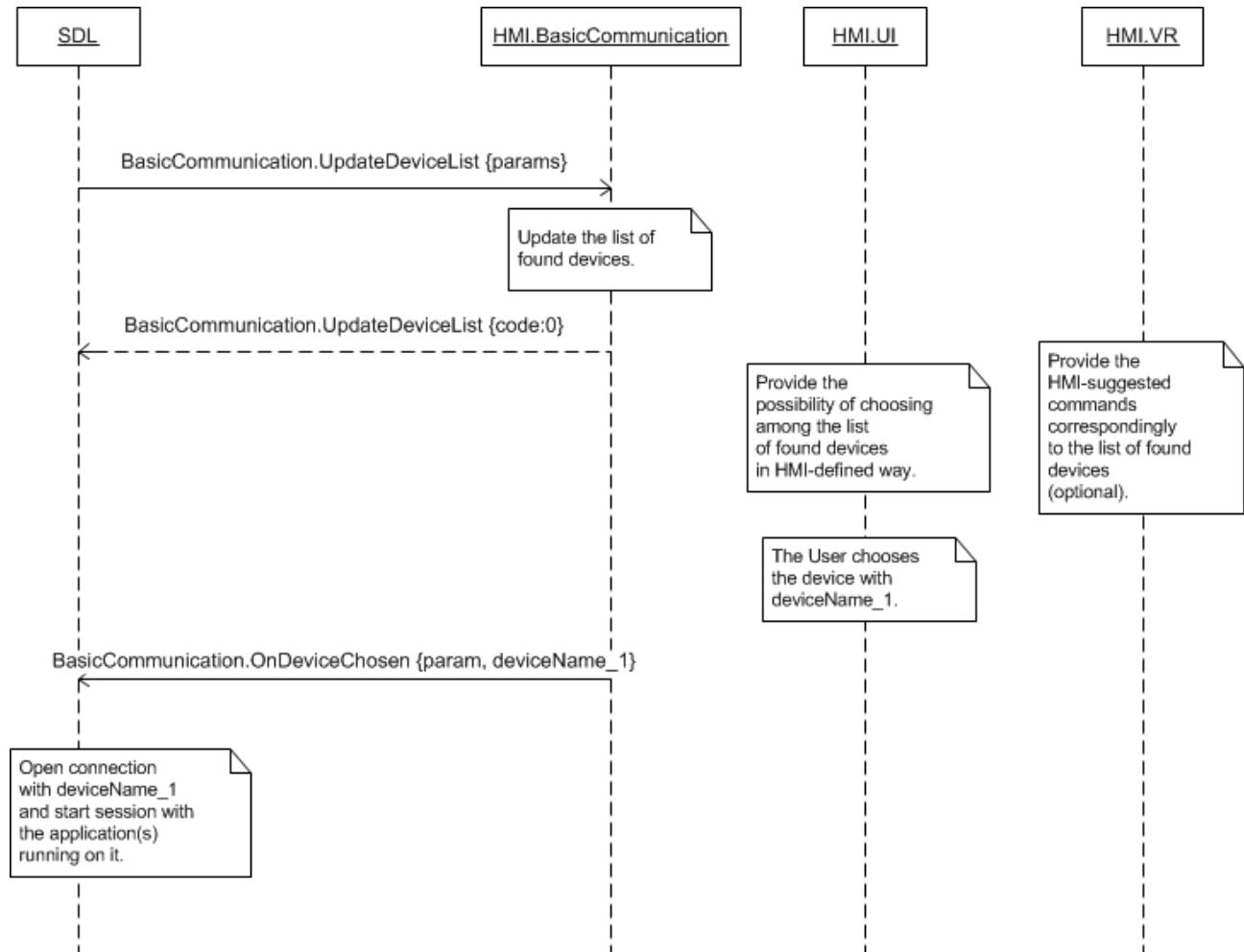
| Param Name | Type | Mandatory | Description |
|------------|-------------------|-----------|--|
| deviceInfo | Common.DeviceInfo | true | The information about the device chosen by the User: its name and ID. See <code>DeviceInfo</code> . |

6.10.1.2 DeviceInfo Structure

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|--|
| name | String | true | The name of the device. |
| id | Integer | true | The ID of the device. It remains unique during the ignition cycle. |

6.10.2 Sequence Diagrams

6.10.2.1 OnDeviceChosen with preceding UpdateDeviceList



6.10.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnDeviceChosen",  
    "params" :  
    {  
        "deviceInfo" :  
        {  
            "name" : "Jerry's Phone",  
            "id" : 3  
        }  
    }  
}
```

6.10.4 D-Bus Messages Examples

6.9.9.1 Request

6.9.6.2 Response

6.9.6.3 Failure

6.11 OnFindApplications

6.11.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Initiate the search for applications on the named device. |

On receipt of this notification SDL provides the list of registered applications for the named device via [UpdateAppList](#) RPC.

HMI must:

- Send OnFindApplications notification upon User's request (e.g. after choosing the device in the list of found devices or on definite button click)..

6.11.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|-------------------|-----------|--|
| deviceInfo | Common.DeviceInfo | true | The information about the device chosen by the User: its name and ID. See DeviceInfo. |

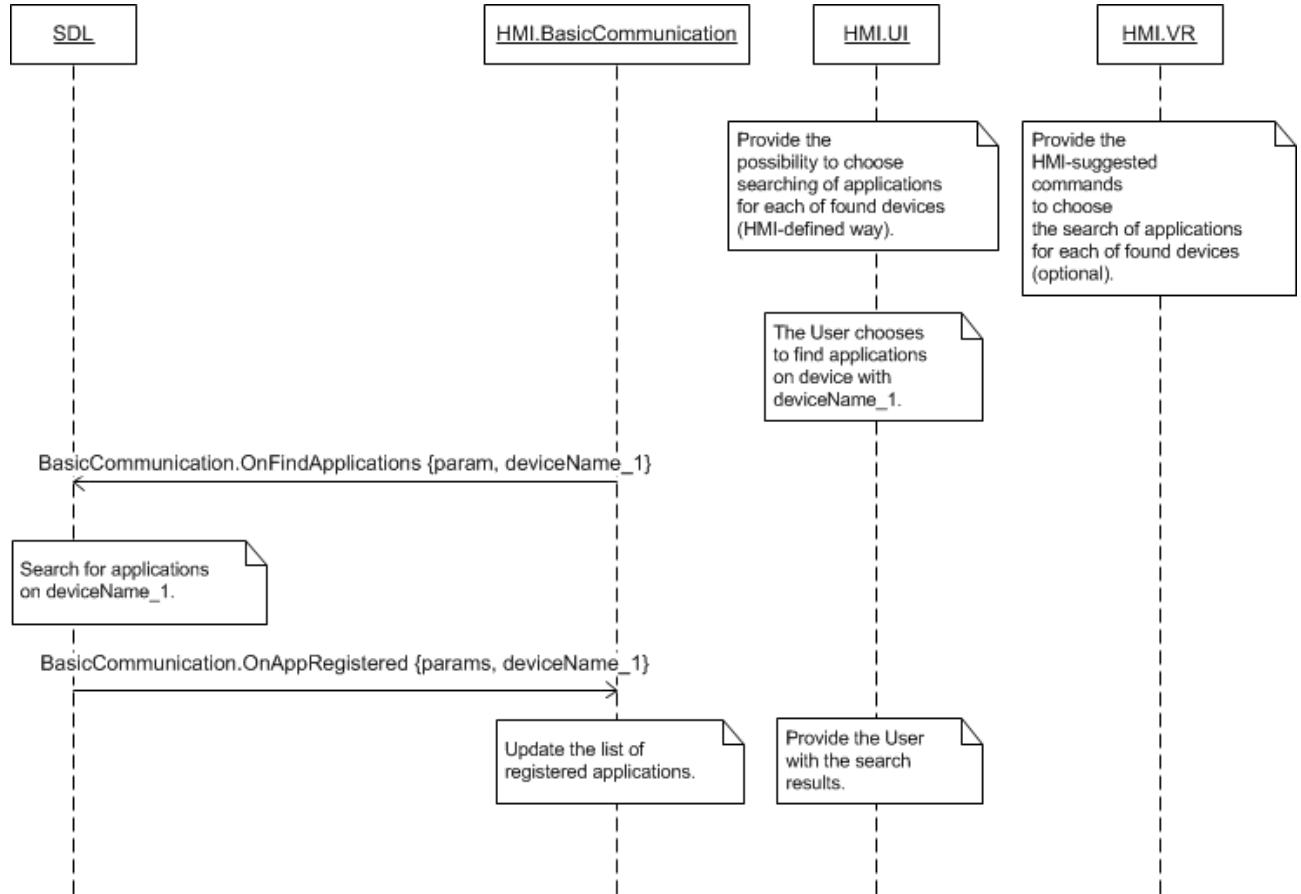
6.11.1.2 DeviceInfo Structure

| Param Name | Type | Mandatory | Description |
|------------|------|-----------|-------------|
|------------|------|-----------|-------------|

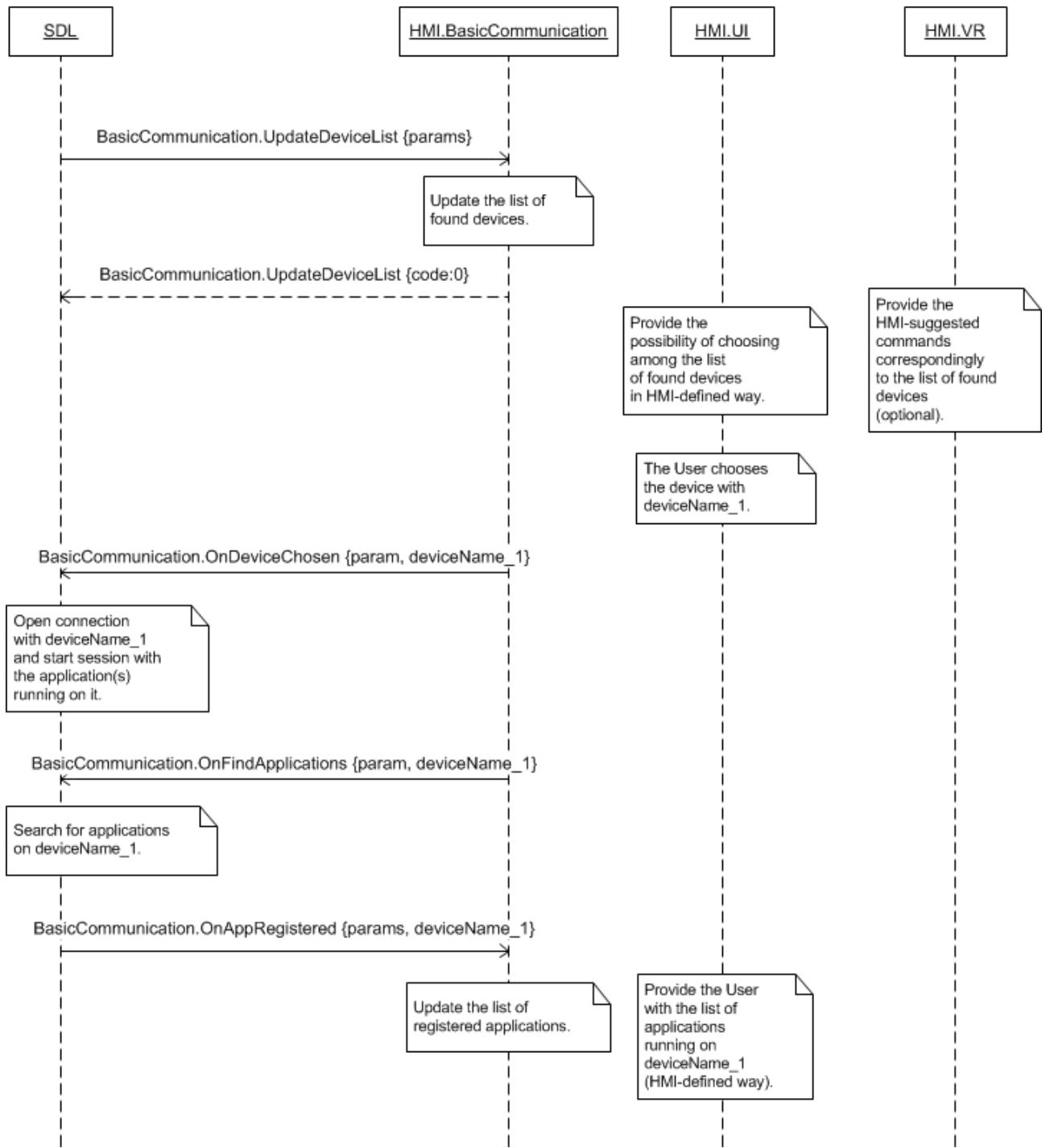
| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|--|
| name | String | true | The name of the device. |
| id | Integer | true | The ID of the device. It remains unique during the ignition cycle. |

6.11.2 Sequence Diagrams

6.11.2.1 OnFindApplications upon User's choice and resulting OnAppRegistered



6.11.2.2 OnFindApplications upon User's choosing the device in the list of found devices and resulting OnAppRegistered



6.11.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.OnFindApplications",
    "params" :
    {
        "deviceinfo" :
        {
            "name" : "XT910",
            "id" : 4
        }
    }
}
```

6.11.4 D-Bus Messages Examples

6.11.9.1 Request

6.11.6.2 Response

6.11.6.3 Failure

6.12 OnAppActivated

6.12.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the User has chosen to activate the application. |

SDL requires this notification for allowing the application on mobile device to send the RPCs related to HMI's functionality (e.g. add commands for VR, start interaction with the User, speak the text via TTS, etc.).

On receipt of `OnAppActivated` notification SDL provides [ActivateApp](#) RPC confirming the named application may be activated.

HMI must:

- Send `OnAppActivated` notification when the User chooses (in one of HMI-suggested ways).to activate the application on HMI.
- Wait for [ActivateApp](#) RPC and only then display the named application related screen.

Note:

The information about the application (name, ID, etc.) is provided by SDL via either [UpdateAppList](#) or `OnAppRegistered` RPCs.

Note:

The difference:

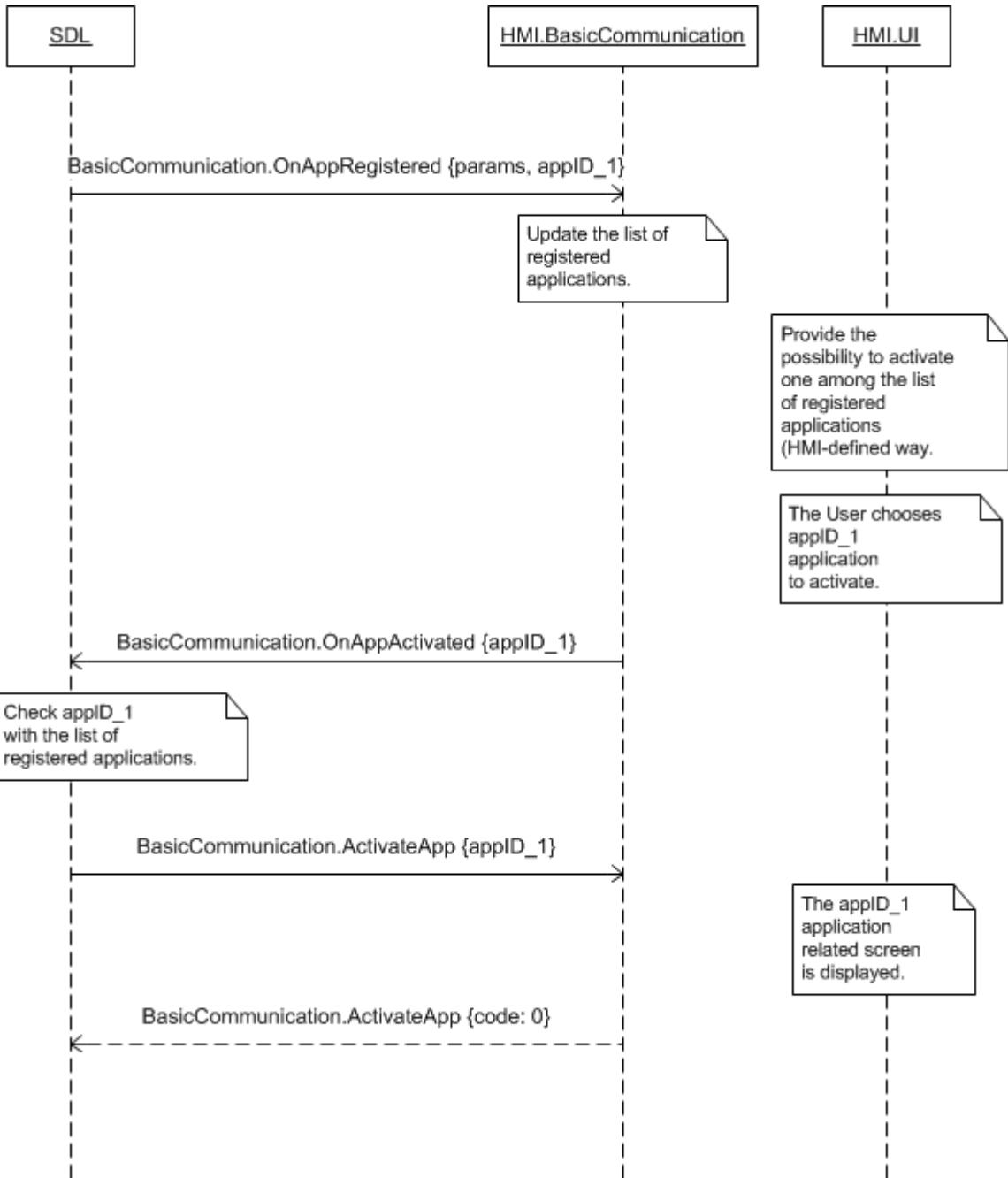
- `OnAppActivated` shows User's intension to activate the named application.
- [ActivateApp](#) is SDL's request and permission for the named application to be granted with the access to HMI's functionality.

6.12.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|---|
| appID | Integer | true | ID of the application to be activated on HMI. |

6.12.2 Sequence Diagrams

6.12.2.1 OnAppActivated upon User's choice and resulting ActivateApp



6.12.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "BasicCommunication.OnAppActivated",
  "params" :
  {
    "appID" : 65544
  }
}
```

6.12.4 D-Bus Messages Examples

6.12.9.1 Request

| |
|--|
| |
|--|

6.12.6.2 Response

| |
|--|
| |
|--|

6.12.6.3 Failure

| |
|--|
| |
|--|

6.13 OnAppDeactivated

6.13.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the application is no longer active on HMI. |

SDL requires this notification for restricting the application on mobile device from sending the RPCs related to HMI's functionality (e.g. adding commands for VR, starting interaction with the User, speaking the text via TTS, etc.).

On receipt of `OnAppDeactivated` SDL deletes the application-related temporary data (e.g. VR commands, UI submenus/commands, changes VR/TTS properties to default ones, etc.) using the appropriate RPCs described in the current document.

HMI must:

- Send `OnAppDeactivated` notification when the User has navigated away from the application persistent screen.

Note:

The information about the application (name, ID, etc.) is provided by SDL via either [UpdateApplList](#) or `OnAppRegistered` RPCs.

6.13.1.1 Parameters

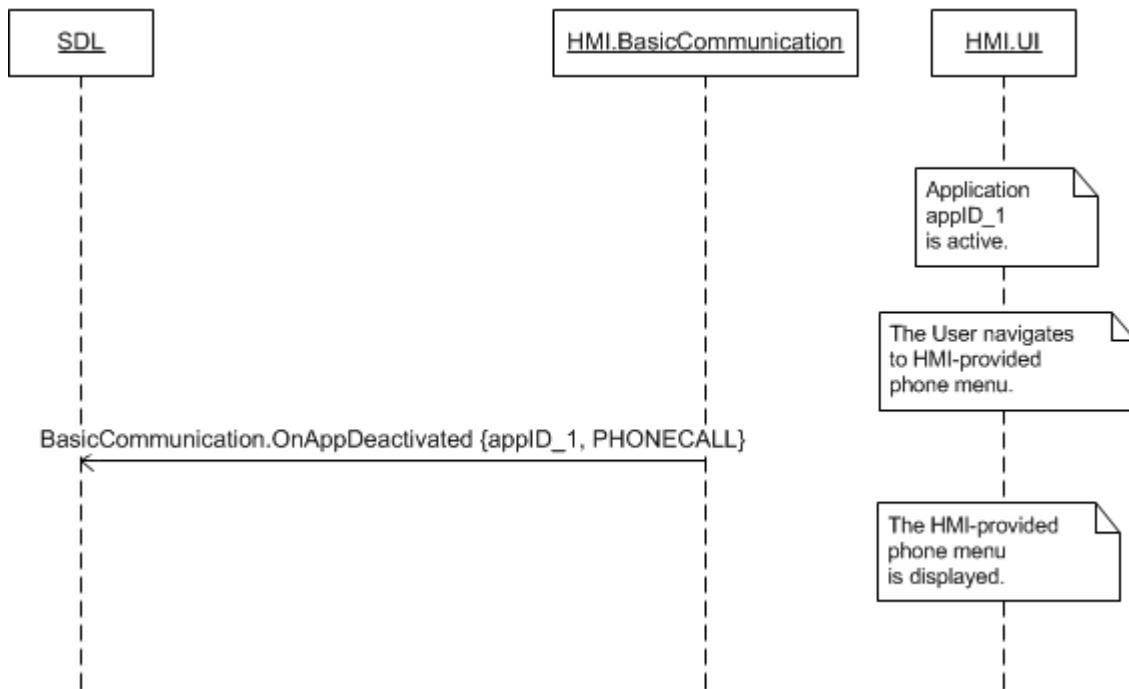
| Param Name | Type | Mandatory | Description |
|------------|-------------------------|-----------|--|
| appID | Integer | true | ID of the application deactivated. |
| reason | Common.DeactivateReason | true | Specifies the functionality the User has switched to. See <code>DeactivateReason</code> . |

6.13.1.2 DeactivateReason Enumeration

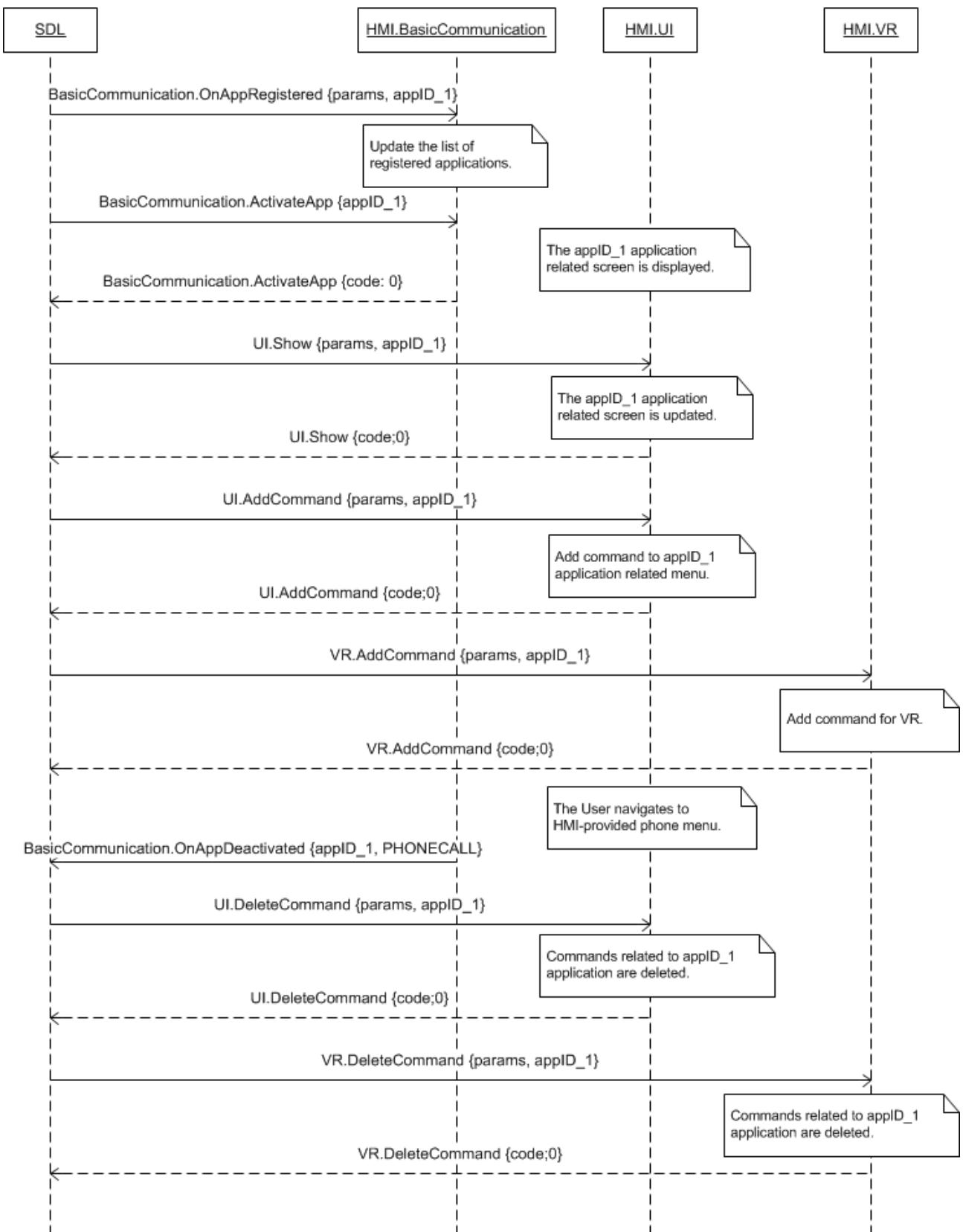
| Element name | Short Description |
|---------------|---|
| AUDIO | The User has navigated to audio (radio, etc.) |
| PHONECALL | The User has navigated to make a call. |
| NAVIGATIONMAP | The User has navigated to navigation screen. |
| PHONEMENU | The User has navigated to phone menu. |
| SYNCSETTINGS | The User has navigated to HU settings menu. |
| GENERAL | Other screens navigation apart from other mobile app. |

6.13.2 Sequence Diagrams

6.13.2.1 OnAppDeactivated Simple Messaging Example



6.13.2.1 OnAppDeactivated with preceding registering, activation, adding UI/VR commands and resulting deleting UI/VR commands



6.13.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.OnAppDeactivated",
    "params" :
    {
        "appID" : 65544,
        "reason" : PHONECALL
    }
}
```

```
    }  
}
```

6.13.4 D-Bus Messages Examples

```
signal sender=:1.151 -> dest=(null destination) serial=60 path=/;  
interface=com.ford.sdl.hmi.BasicCommunication;  
member=OnAppDeactivated  
    int32 65537  
    int32 5
```

6.14 OnAppRegistered

6.14.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Update the HMI's list of registered applications and resume the Audio Source on HMI. |

SDL may send the request:

- After SDL connected the device and the SDL-enabled application has registered (or re-registered itself with SDL successfully).
- After SDL re-connected the device after unexpected disconnect and the SDL-enabled application has re-registered (with or without resumption) itself with SDL successfully.
- Upon User's request delivered by HMI via `OnFindApplications` notification, in case any new applications are registered.

Note about resumption:

- *SDL keeps storing the application-related data during three ignition cycles after unexpected disconnect or ignition off. Upon the forth ignition on SDL clears all corresponding-application-related data for resumption.*
- *If the application resumes successfully, SDL restores application-related data like its state on HMI (also known as HMIlevel): active or background, params of Show, SetGlobalProperties and other.*
 - *SDL will provide `OnAppRegistered` with `resume:true`.*
 - *SDL will send `ActivateApp` RPC right afterwards, if SDL restores the application remembered to be active. When successful response is received from HMI, SDL will immediately send `Show`, `SetGlobalProperties`, `AddCommand` and other that is restored for the named application.*
 - *SDL will NOT send `ActivateApp` RPC to HMI while performing resumption If SDL remembers the application to be in background. (Unless further requested from HMI via `OnAppActivated`).*
- *If the application does NOT resume successfully, SDL restores just its state on HMI (also known as HMIlevel): active or background.*
 - *SDL will provide `OnAppRegistered` with `resume:false` or without `resume` parameter.*
 - *SDL will send `ActivateApp` RPC right afterwards, if SDL restores the application remembered to be active.*
 - *SDL will NOT send `ActivateApp` RPC, If SDL remembers the application to be in background. (Unless further requested from HMI via `OnAppActivated`).*

HMI must:

- Update its list of registered applications.
- Store the application data sent with `applications` parameter (application name, icon, name of corresponding device, appID and etc. that is described in section 6.15.1.2 HMIApplication).

Note:

The value of appID for one and the same application most often varies between ignition cycles and/or reconnections while the other values of HMIApplication parameter remain the same.

- Remember the value of ttsName parameter and speak this information (which is the application name) when the User speaks 'Help' in the VR layer that corresponds 'the list of registered applications' on UI.
- Remember the value of vrSynonyms parameter and use them when arranging the possibility for the User to choose among the registered applications by VR. The commands must be accessible on the layer correspondent to the following entries on UI:
 - The list of registered applications
 - Any active application (that is, HMI must provide the possibility to activate any of registered applications by VR when whichever another application is currently active).
- Provide the User with the possibility to choose among registered applications on UI.

Note:

HMI must provide OnAppActivated with the corresponding appID whenever the User activates the application from UI or via VR.

- Use appID provided within this request when sending definite notifications or requests (e.g. OnAppActivated, OnCommand and other) related to the corresponding application to SDL.
- In case of resume: true:
 - Understand that provided appID relates to the last Audio Source remembered by HMI (SDL mostly sends different appID with the same other values of applications parameter if to compare with 'before-resumption' one).
 - Restore the last Audio Source.

6.14.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|--------------|-----------------------|-----------|--|---|
| applications | Common.HMIApplication | true | - | <p>The parameter contains the detailed information about each application: its name, ID, type and other. See HMIApplication.</p> <p>Note: the value of appID for one and the same application most often varies between ignition cycles and/or reconnections with the same other values of HMIApplication parameter.</p> |
| ttsName | Common.TTSChunk | false | Array = true Minsize = 1 Maxsize = 100 | TTS string. HMI must speak this information (which is the application name) when the User speaks 'Help' in the VR layer that corresponds 'the list of registered applications' on UI. |

| | | | | |
|------------------|--------------------|-------|---|--|
| | | | | Voice recognition commands that HMI must use when arranging the possibility for the User to choose among the registered applications by VR. The commands must be accessible on the layer correspondent to: - UI: 'list of registered applications' - UI: any active application (that is, HMI must provide the possibility to activate any of registered applications by VR when whichever another application is currently active). When any of provided commands is recognized (that is, application is chosen by VR), HMI must send OnAppActivated with the corresponding appID to SDL. |
| resumeVrGrammars | Boolean | false | - | Informs whether the application-related VR grammars must be restored. |
| priority | Common.AppPriority | false | - | Send to HMI so that it can coordinate order of requests/notifications correspondingly. |

6.14.1.2 HMI Application Structure

| Param Name | Type | Mandatory | Additional | Description |
|---------------------------|-----------------|-----------|-----------------|---|
| appName | String | true | Maxlength = 100 | The mobile application name. It is unique over all applications. |
| ngnMediaScreenAppName | String | false | Maxlength = 100 | Provides an abbreviated version of the application name (may be displayed on the NGN media screen). If not provided, the appName should be used instead (and may be truncated if too long) |
| icon | String | false | - | Path to the application icon stored on HU hard disc. |
| deviceName | String | true | - | The name of the device where the identified application is running on. |
| appID | Integer | true | - | The application ID that remains unique during the ignition cycle. This ID will be sent by SDL further and must be provided by HMI within all the RPCs related to this application. |
| hmiDisplayLanguageDesired | Common.Language | true | - | The language that the application intends to use. See Language. |
| isMediaApplication | Boolean | true | - | Indicates whether the application is a media or a non-media one. Only media applications are allowed by SDL to stream audio to HU that is audible outside of the BT media source. |

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|-------------------|------------------|--|---|
| appType | Common.AppHMIType | false | array = true Minsize = 1 maxsize = 100 | The HMI may use this information for determining what functionality should be available for the application (e.g. NAVIGATION type of application will require displaying the information and not playing the audio). See AppHMIType. |

6.14.1.3 AppHMIType Enumeration

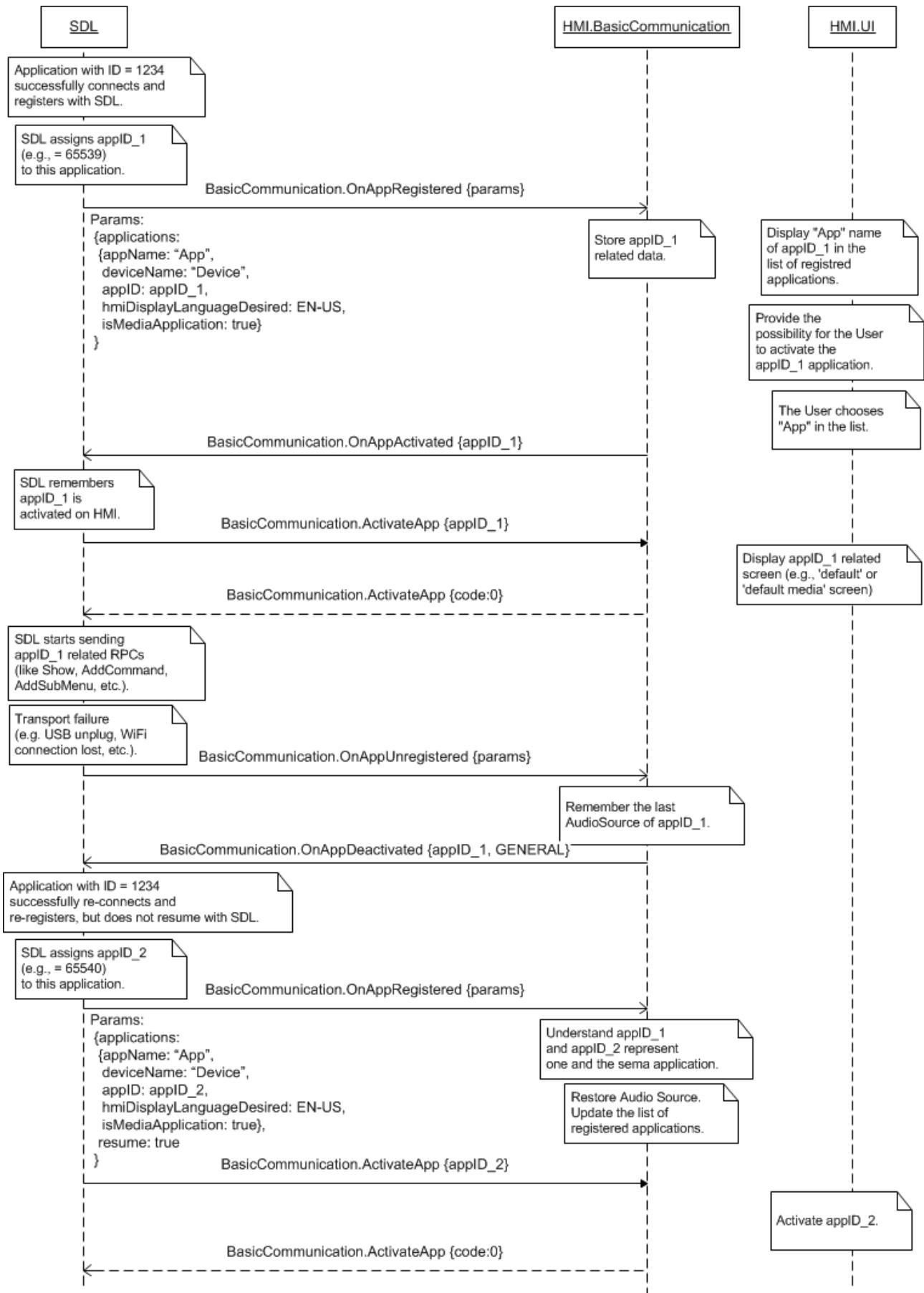
| Element name | Short Description |
|---------------------|---|
| DEFAULT | The application of default type. |
| COMMUNICATION | The application for communication |
| MEDIA | The media application |
| MESSAGING | The application of messaging type |
| NAVIGATION | The application of navigation type |
| INFORMATION | The application of information type |
| SOCIAL | The application of social type |
| BACKGROUND_PROCESS | The application does not require displaying the information |
| TESTING | The application of testing type |
| SYSTEM | The application of system type |

6.14.1.4 AppPriority Enumeration

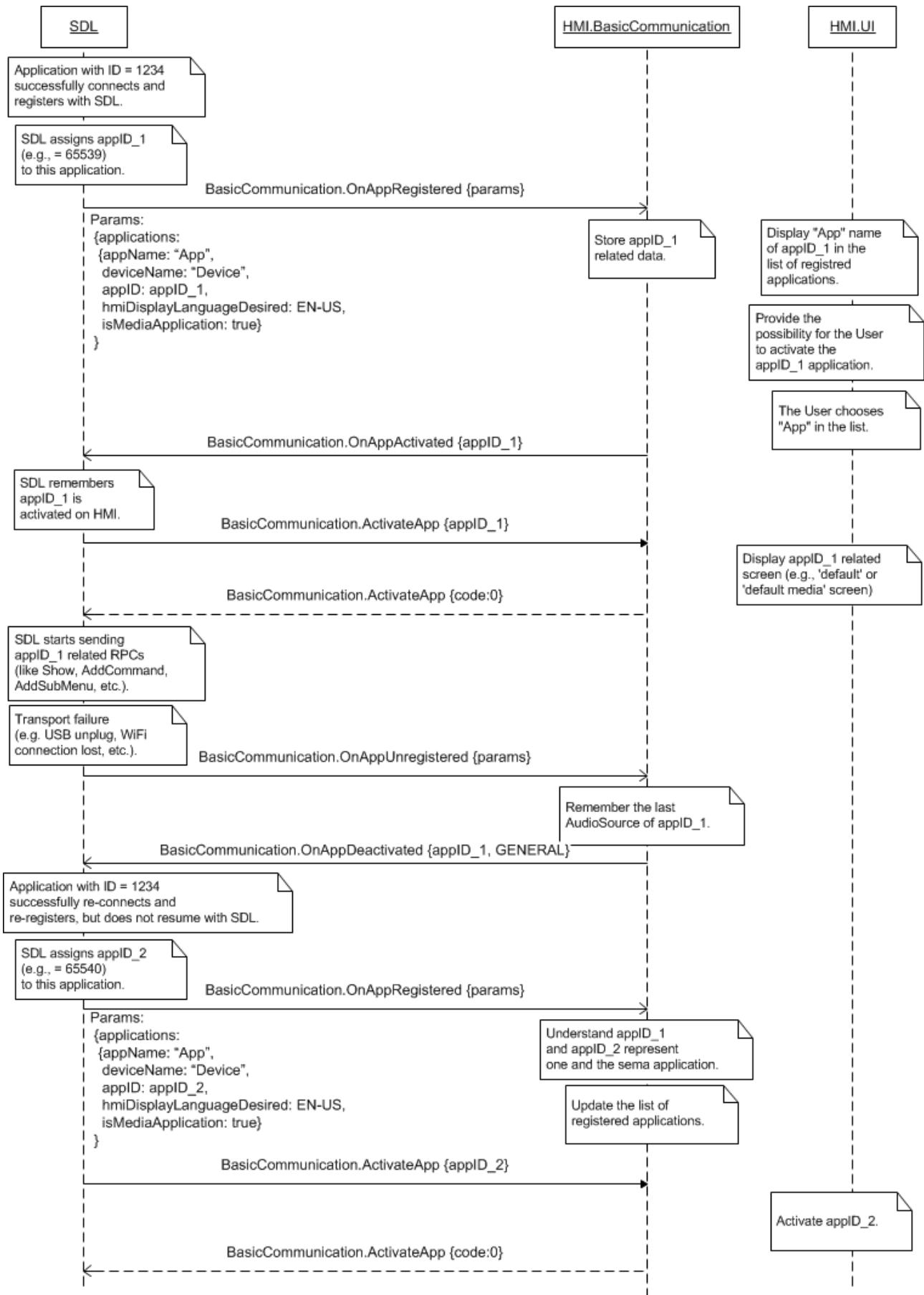
| Element name | Value | Short Description |
|---------------------|--------------|--------------------------|
| EMERGENCY | 0 | |
| NAVIGATION | 1 | |
| VOICE_COMMUNICATION | 2 | |
| COMMUNICATION | 3 | |
| NORMAL | 4 | |
| NONE | 5 | |

6.14.2 Sequence Diagrams

6.14.2.1 OnAppRegistered with resume=true after unexpected disconnect



6.14.2.2 OnAppRegistered without 'resume' parameter after unexpected disconnect



6.14.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnAppRegistered",  
    "params" :  
    {  
        "application" :  
        {  
            "appName" : "TryMe",  
            "ngnMediaScreenAppName" : "TryMe",  
            "deviceName" : "GT-I9300",  
            "appID" : 65540,  
            "hmiDisplayLanguageDesired" : ES-ES,  
            "isMediaApplication" : false  
        }  
        "resume" : true  
    }  
}
```

6.14.4 D-Bus Messages Examples

```
signal sender=:1.196 -> dest=(null destination) serial=100 path=/;  
interface=com.ford.sdl.core.BasicCommunication; member=OnAppRegistered  
    struct {  
        string "Hugo"  
        struct {  
            boolean true  
            string "Hugo"  
        }  
        struct {  
            boolean true  
            string ""  
        }  
        string "127.0.0.1"  
        int32 65538  
        int32 0  
        boolean true  
        struct {  
            boolean false  
            array [  
            ]  
        }  
    }  
}
```

6.15 OnAppUnregistered

6.15.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about the application has unregistered. |

SDL sends this notification when

- The named application has unregistered from the side of mobile device via appropriate RPC
- The connection and the corresponding session(s) are closed due to transport issues (for example, WiFi/BT connection closing on mobile device, USB unplugging, etc.).
- The HeartBeat Timeout between mobile application and SDL occurs (one of SDL or application does not respond with HeartBeat message during timeout defined in *smartDeviceLink.ini* file, thus the session is closed by the responsible side).

- The User chooses the 'Master Reset' or 'Factory Defaults' options on HMI. See OnExitAllApplications (MASTER_RESET and FACTORY_DEFAULTS) for more details
- The User ends the key-on session (that is, ignition cycle is over). See OnExitAllApplications (IGNITION_OFF) for more details.

1. **HMI must:** Distinguish 'unexpected disconnect' and 'regular exit' events by "unexpectedDisconnect" parameter:
 - 1.1. In case of "**unexpectedDisconnect: true**" arrives for the active app: HMI must
 - a) Display the informational HMI-defined popup of the kind: "The connection with "%AppName%" application has been unexpectedly lost".
 - b) Switch the display to HMI-defined screen/page (e.g. 'Home' page, screen showing the list of registered applications, etc.).
 - 1.2. In case "**unexpectedDisconnect: false**": HMI must proceed with step 2.
2. Switch the display to HMI-defined screen/page (e.g. 'Home' page, screen showing the list of registered applications, etc.) if the named application is active when this notification comes. .
3. Remove the named application from the list of registered applications.
4. Delete all the data related to this application.

Note:

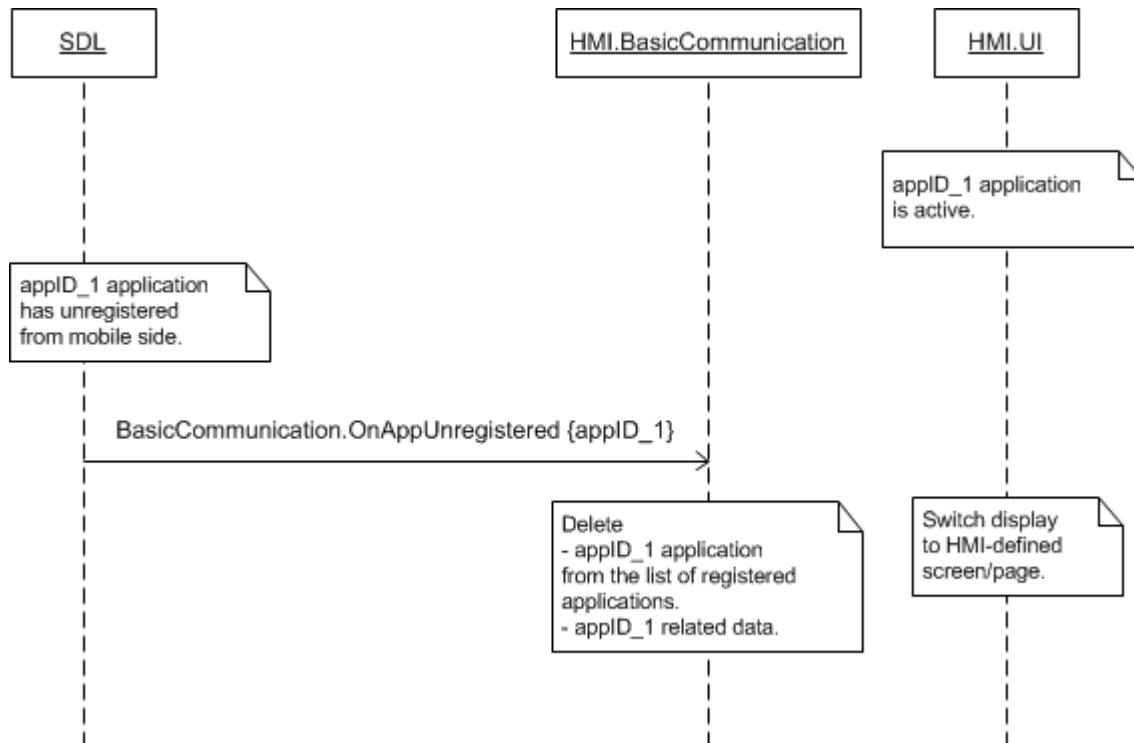
The information about the application (name, ID, etc.) is provided by SDL via either [UpdateAppList](#) or [OnAppRegistered](#) RPCs.

6.15.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|----------------------|---------|-----------|---|
| appID | Integer | true | ID of the application having been unregistered. |
| unexpectedDisconnect | Boolean | true | Defines whether the connection with the app was unexpectedly lost or this was the regular app's exit. |

6.15.2 Sequence Diagrams

6.15.2.1 OnAppUnregistered of active application



6.15.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnAppUnregistered",  
    "params" :  
    {  
        "appID" : 65539  
    }  
}
```

6.15.4 D-Bus Messages Examples

6.16 OnExitApplication

6.16.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about either the User has chosen to exit the application or the driver distraction rules has been violated. |

HMI must:

- Provide the possibility for the User to choose exiting any of registered applications (e.g. HMI-defined button, VR command, etc.).
- Track that the application being active or in background on HMI conforms with driving distraction rules.
- Send `OnExitApplication` when
 - 1) The User chooses to exit the application in one of HMI-suggested ways.
 - 2) The application violates one of the driving distraction rules on HMI.

Note:

The information about the application (name, ID, etc.) is provided by SDL via either `UpdateAppList` or `OnAppRegistered` RPCs.

6.16.1.1 Parameters

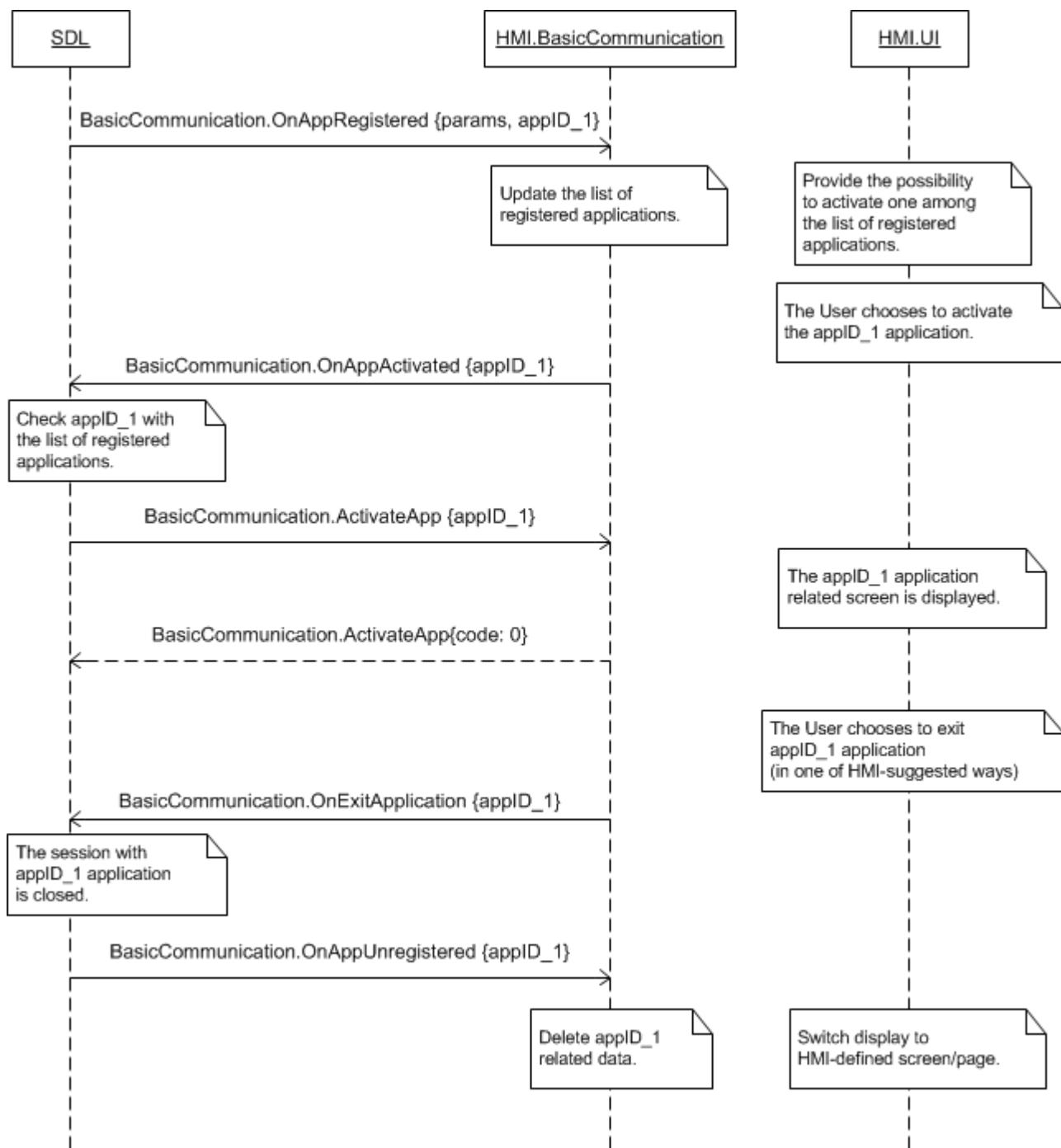
| Param Name | Type | Mandatory | Description |
|------------|--------------------------------|-----------|---|
| reason | Common.ApplicationToNONEReason | true | The reason why the named application should be exited from HMI. |
| appID | Integer | true | ID of the application chosen to be exited. |

6.16.1.2 ApplicationToNONEReason

| Element name | Value | Short Description |
|------------------------------|-------|--|
| DRIVER_DISTRACTION_VIOLATION | 0 | The application should be exited because of driver distraction rules violation |
| USER_EXIT | 1 | The application should be exited because so is requested by the User. |

6.16.2 Sequence Diagrams

6.16.2.1 OnExitApplication and the resulting OnAppUnregistered



6.16.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "BasicCommunication.OnExitApplication",
  "params" :
  {
    "appID" : 65544,
    "reason" : USER_EXIT
  }
}
```

6.16.4 D-Bus Messages Examples

| |
|--|
| |
|--|

6.17 OnExitAllApplications

6.17.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the event that causes exit of all applications. |

SDL requires this notification to accurately close the sessions with the registered applications before reloading/shutting down by the reason of User's actions.

HMI must:

- Send the notification upon one of the following results of User's actions:
 - Key off
 - Master reset
 - Return to factory defaults.

6.17.1.1 Parameters

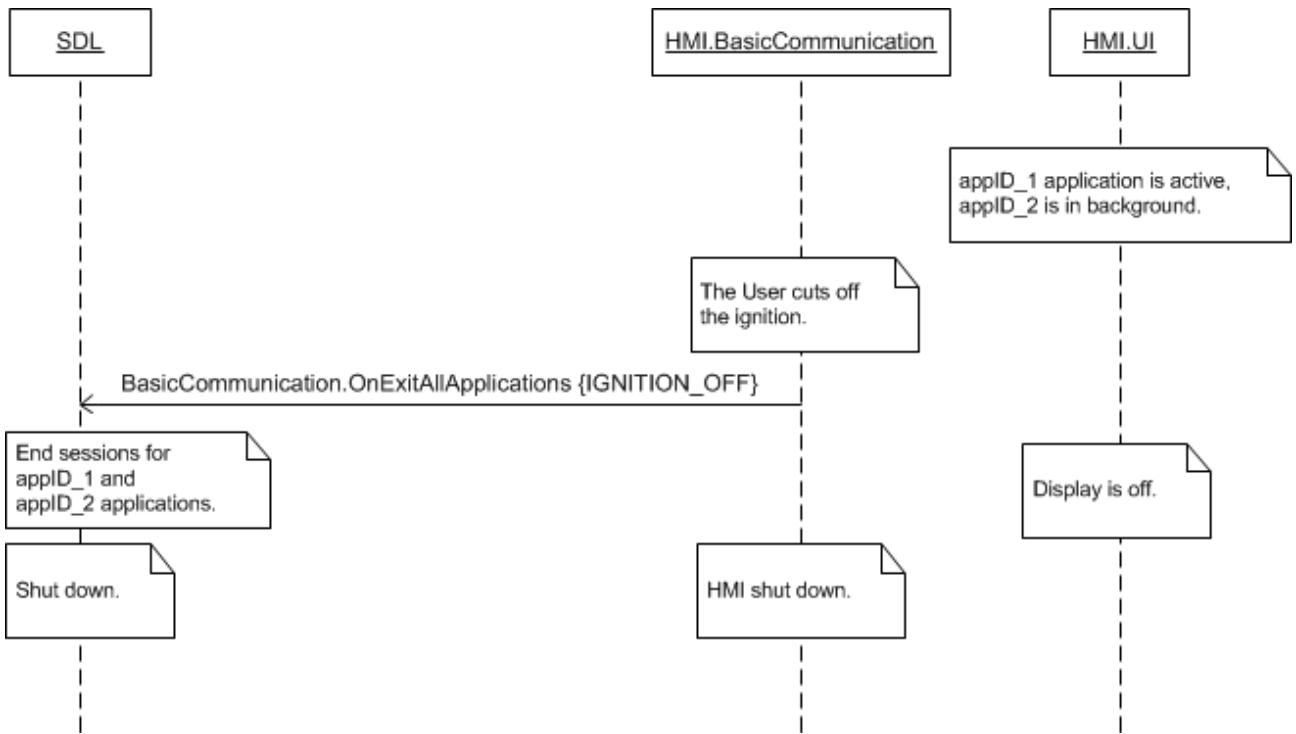
| Param Name | Type | Mandatory | Description |
|------------|---------------------------------|-----------|--|
| reason | Common.Application.sCloseReason | true | The reason for exiting all the applications. See ApplicationsCloseReason. |

6.17.1.2 ApplicationsCloseReason Enumeration

| Element name | Short Description |
|------------------|---|
| IGNITION_OFF | Key off. |
| MASTER_RESET | Master reset. |
| FACTORY_DEFAULTS | Return to factory defaults. |
| SUSPEND | Notifies SDL to perform persistence of data for resumption. |

6.17.2 Sequence Diagrams

6.17.2.1 OnExitAllApplications



6.17.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.OnExitAllApplications",
    "params" :
    {
        "reason" : IGNITION_OFF
    }
}
```

6.17.4 D-Bus Messages Examples

6.18 PlayTone

6.18.1 Description

| | |
|-----------------|------------------------------------|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Play HMI standard notifying sound. |

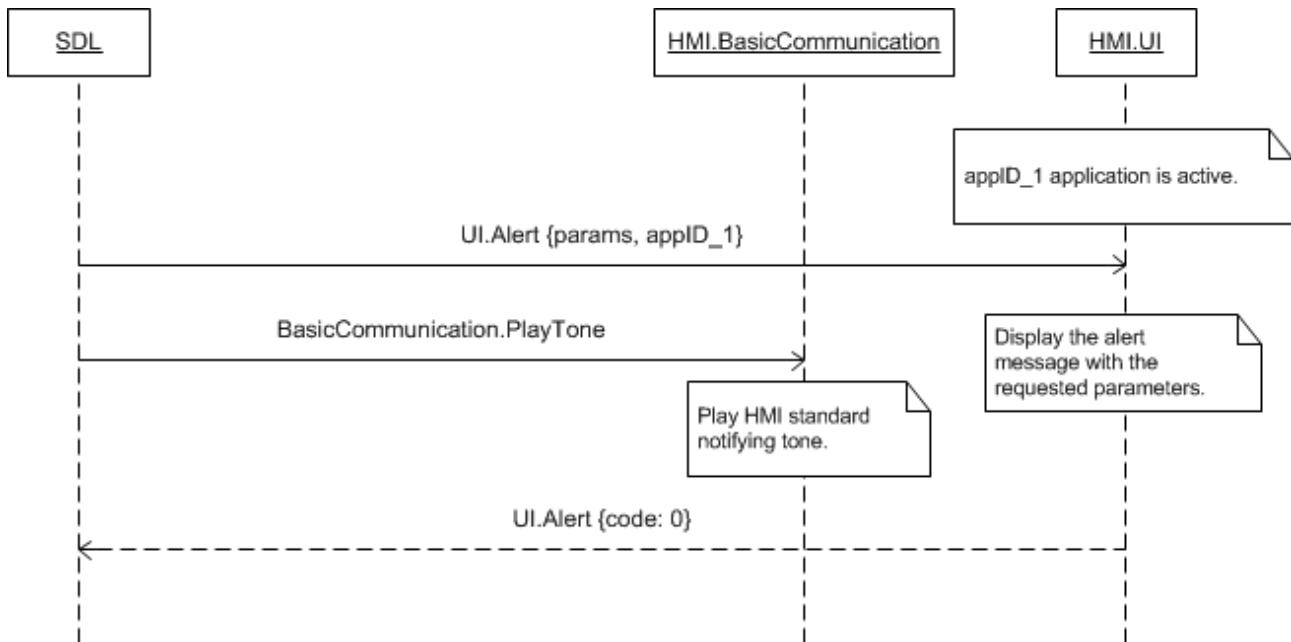
This notification may follow some SDL request that brings changes on HMI and is desirable to be noticed or reacted by the User (e.g. alert message displayed, audio capturing started, etc.).

HMI must:

- Play HMI standard notifying sound.

6.18.2 Sequence Diagrams

6.18.2.1 PlayTone with Alert request



6.18.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "BasicCommunication.PlayTone"
}
```

6.18.4 D-Bus Messages Examples



6.19 OnSystemRequest

6.19.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Transfer to mobile app for initiating the download of data. |

An asynchronous request from the system for specific data from the device or the cloud or response to a request from the device or cloud

Binary data can be included in hybrid part of message for some requests (such as Authentication request responses)

HMI must:

Note:

6.19.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| | | | | |
|-------------|--------------------|-------|--|--|
| requestType | Common.RequestType | true | - | The type of system request. |
| url | string | false | minlength = 1 maxlength = 1000 | Optional array of URL(s) for HTTP requests. |
| fileType | Common.FileType | false | - | Optional file type (meant for HTTP file requests). |
| offset | Integer | false | minvalue = 0 maxvalue = 1000000000000 | Optional offset in bytes for resuming partial data chunks |
| length | Integer | false | minvalue = 0 maxvalue = 1000000000000 | Optional length in bytes for resuming partial data chunks. |
| timeout | Integer | false | minvalue = 0 maxvalue = 2000000000 | |
| fileName | string | true | minlength = 1 maxlength = 255 | File reference name. |
| appID | string | false | minlength = 1 maxlength = 50 | ID of application that requested this RPC. |

6.19.1.2 RequestType Enumeration

| Element name | Value | Short Description |
|----------------|-------|-------------------|
| HTTP | 0 | |
| FILE_RESUME | 1 | |
| AUTH_REQUEST | 2 | |
| AUTH_CHALLENGE | 3 | |
| AUTH_ACK | 4 | |

6.19.1.3 FileType Enumeration

| Element name | Value | Short Description |
|--------------|-------|-------------------|
| GRAPHIC_BMP | 0 | |
| GRAPHIC_JPEG | 1 | |
| GRAPHIC_PNG | 2 | |
| AUDIO_WAVE | 3 | |
| AUDIO_MP3 | 4 | |
| AUDIO_AAC | 5 | |
| BINARY | 6 | |
| JSON | 7 | |

6.19.2 Sequence Diagrams

6.19.2.1 OnSystemRequest

6.19.2.2 OnSystemRequest

6.19.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnSystemRequest",  
    "params" :  
    {  
        "requestType" : HTTP  
  
    }  
}
```

6.19.4 D-Bus Messages Examples

6.20 OnPutFile

6.20.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about application-related file storing |

Notification, that comes to HMI when mobile sends file with systemFile flag

HMI must:

6.20.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|----------------|-----------------|-----------|---|--|
| offset | Integer | false | minvalue = 0 maxvalue = 100000000000 | Optional offset in bytes for resuming partial data chunks |
| length | Integer | false | minvalue = 0 maxvalue = 100000000000 | Optional length in bytes for resuming partial data chunks |
| fileSize | Integer | false | minvalue = 0 maxvalue = 100000000000 | Full Size of file. sends in first OnPutFile notification if file is splitted into many PutFiles |
| fileName | String | true | maxlength = 255 | File reference name. |
| syncFileName | String | true | maxlength = 255 | File reference name. |
| fileType | Common.FileType | true | - | Selected file type. |
| persistentFile | Boolean | false | - | Indicates if the file is meant to persist between sessions / ignition cycles. If set to TRUE, then the system will aim to persist this file through session / cycles. While files with this designation will have priority over others, they are |

| | | | | |
|--|--|--|--|--|
| | | | | subject to deletion by the system at any time. In the event of automatic deletion by the system, the app will receive a rejection and have to resend the file. If omitted, the value will be set to false. |
|--|--|--|--|--|

6.20.2 Sequence Diagrams

6.20.2.1 OnPutFile

6.20.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "BasicCommunication.OnPutFile"
}
```

6.20.4 D-Bus Messages Examples

6.21 OnFileRemoved

6.21.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about application-related file removing |

HMI must:

6.21.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|-----------------|-----------|---------------------------------|--|
| fileName | String | true | minlength = 1 maxlength = 30 | The name of the file that has been removed |
| fileType | Common.FileType | true | - | The file type. See FileType. |
| appID | Integer | true | - | ID of the application. |

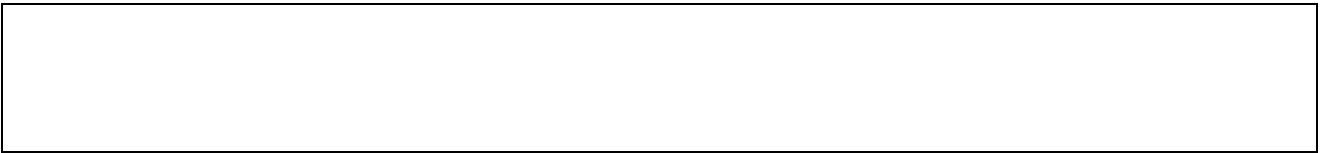
6.21.2 Sequence Diagrams

6.21.2.1 OnFileRemoved

6.21.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnFileRemoved"  
}
```

6.21.4 D-Bus Messages Examples



6.22 OnSystemInfoChanged

6.22.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about application-related file removing |

Issued by system to SDL to notify that some system param has changed. Currently applied for Sync Language.

HMI must:

6.22.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|-----------------|-----------|-------------|
| language | Common.Language | true | |

6.22.2 Sequence Diagrams

6.22.2.1 OnSystemInfoChanged

6.22.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnSystemInfoChanged"  
}
```

6.22.4 D-Bus Messages Examples

| |
|--|
| |
|--|

6.23 OnIgnitionCycleOver

6.23.1 Description

| | |
|-----------------|-------------------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about ignition cycle is over |

Notification from system to SDL to let it know that ignition cycle is over.

NOTE:

HMI must send **both** of the notifications to SDL:

- BasicCommunication.OnExitAllApplications(IGNITION_OFF) for SDL to start the procedure of unregistering all of registered applications and then close.
- BasicCommunication.OnIgnitionCycleOver for SDL to update the ignition-cycle-related fields in Policy Table.

HMI must:

6.23.2 Sequence Diagrams

6.23.2.1 OnIgnitionCycleOver

6.23.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnSystemInfoChanged"  
}
```

6.23.4 D-Bus Messages Examples

| |
|--|
| |
|--|

6.24 OnSDLClose

6.24.1 Description

| | |
|----------------|--------------|
| Type: | Notification |
| Sender: | SDL |

| | |
|-----------------|-------------------------|
| Purpose: | Inform about SDL closes |
|-----------------|-------------------------|

HMI must:

6.24.2 Sequence Diagrams

6.24.2.1 OnSDLClose

6.24.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.OnSDLClose"
}
```

6.24.4 D-Bus Messages Examples

6.25 OnUpdateDeviceList

6.25.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Get the updated list of found devices. |

On receipt of this notification SDL provides the list of devices found during the recent search procedure sending `UpdateDeviceList` RPC.

HMI must:

- Provide the User with the possibility to initiate the device list updating in one of accessible ways (VR, buttons, etc.).
- Send `OnUpdateDeviceList` notification when the User has chosen to update the device list.

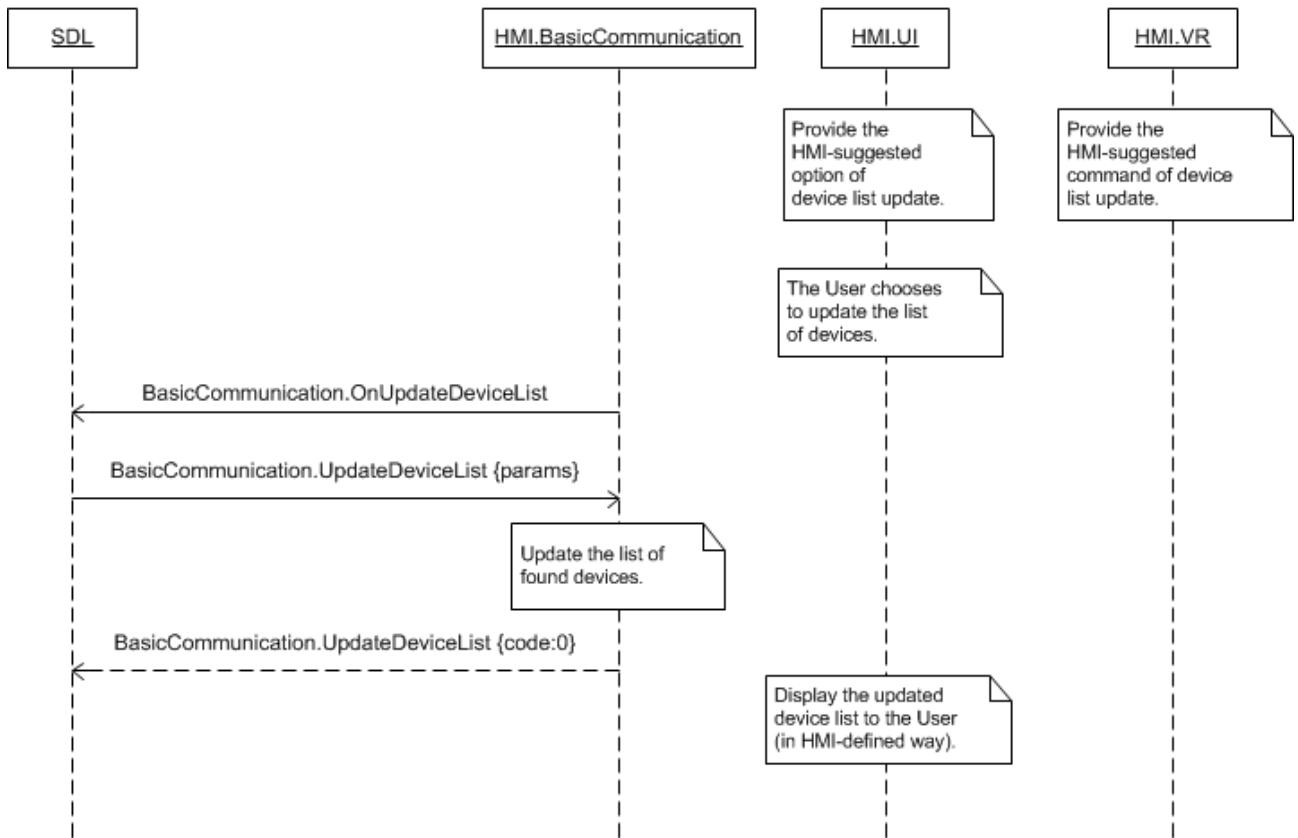
Note:

The difference:

- `OnUpdateDeviceList` results `SDL` to provide the updates of the recent search.
- `OnStartDeviceDiscovery` initiates `SDL` to start the new search procedure

6.25.2 Sequence Diagrams

6.25.2.1 OnUpdateDeviceList upon User's request and resulting UpdateDeviceList



6.25.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.OnUpdateDeviceList"
}
```

6.25.4 D-Bus Messages Examples



6.26 OnResume AudioSource

6.26.1 Description

| | |
|-----------------|---|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about the named application being in background needs to become audible on HMI |

SDL sends **OnResume AudioSource** to HMI only in case when in result of 'resumption' procedure SDL has detected the application had the state of 'LIMITED, AUDIBLE' on HMI before the previous ignition off.

HMI must:

- Activate the audio source for the named application.
- Not activate the application itself. The named application must stay in background on HMI.

6.26.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|------------|--|
| appID | Integer | true | - | The ID of the application that HMI must activate the audio source for. |

6.26.2 Sequence Diagrams

6.26.2.1 OnResume AudioSource

6.26.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnResume AudioSource"  
}
```

6.26.4 D-Bus Messages Examples



6.27 UpdateAppList

6.27.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Update HMI's list of registered applications. |

The request may come:

- After SDL connected the device and the SDL-enabled application has registered itself with SDL successfully.
- Upon User's request delivered by HMI via `OnFindApplications` notification.

6.27.2 Request

6.27.2.1 Behavior

HMI must:

- Update its list of registered applications.
- Use the `appID` provided within this request when sending the definite notifications or requests (e.g. `ActivateApp`, `OnCommand` and other) related to the corresponding application to SDL.
- Provide the User with the possibility to choose among the registered applications on UI.

Note:

SDL adds the VR synonyms of all registered applications to HMI via `VR.AddCommand`.

6.27.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| | | | | |
|--------------|----------------------|------|--|---|
| applications | Common.HMIApplcation | true | Array = true Minsize = 1 Maxsize = 100 | The list of registered applications. The array contains the detailed information about each application: its name, ID, type and other. See HMIApplcation . |
|--------------|----------------------|------|--|---|

6.27.2.3 HMIApplcation Structure

| Param Name | Type | Mandatory | Additional | Description |
|---------------------------|-------------------|-----------|--|--|
| appName | String | true | Maxlength = 100 | The mobile application name. It is unique over all applications. |
| ngnMediaScreenAppName | String | false | Maxlength = 100 | Provides an abbreviated version of the application name (may be displayed on the NGN media screen). If not provided, the appName should be used instead (and may be truncated if too long) |
| icon | String | false | – | Path to the application icon stored on HU hard disc. |
| deviceName | String | true | – | The name of the device where the identified application is running on. |
| appID | Integer | true | – | The application ID that remains unique during the ignition cycle. This ID will be sent by SDL further and must be provided by HMI within all the RPCs related to this application. |
| hmiDisplayLanguageDesired | Common.Language | true | – | The language that the application intends to use. See Language . |
| isMediaApplication | Boolean | true | – | Indicates whether the application is a media or a non-media one. Only media applications are allowed by SDL to stream audio to HU that is audible outside of the BT media source. |
| appType | Common.AppHMIType | false | array = true Minsize = 1 maxsize = 100 | The HMI may use this information for determining what functionality should be available for the application (e.g. NAVIGATION type of application will require displaying the information and not playing the audio). See AppHMIType . |

6.27.2.4 AppHMIType Enumeration

| Element name | Short Description |
|---------------|-------------------------------------|
| DEFAULT | The application of default type. |
| COMMUNICATION | The application for communication |
| MEDIA | The media application |
| MESSAGING | The application of messaging type |
| NAVIGATION | The application of navigation type |
| INFORMATION | The application of information type |

| Element name | Short Description |
|--------------------|---|
| SOCIAL | The application of social type |
| BACKGROUND_PROCESS | The application does not require displaying the information |
| TESTING | The application of testing type |
| SYSTEM | The application of system type |

6.27.3 Response

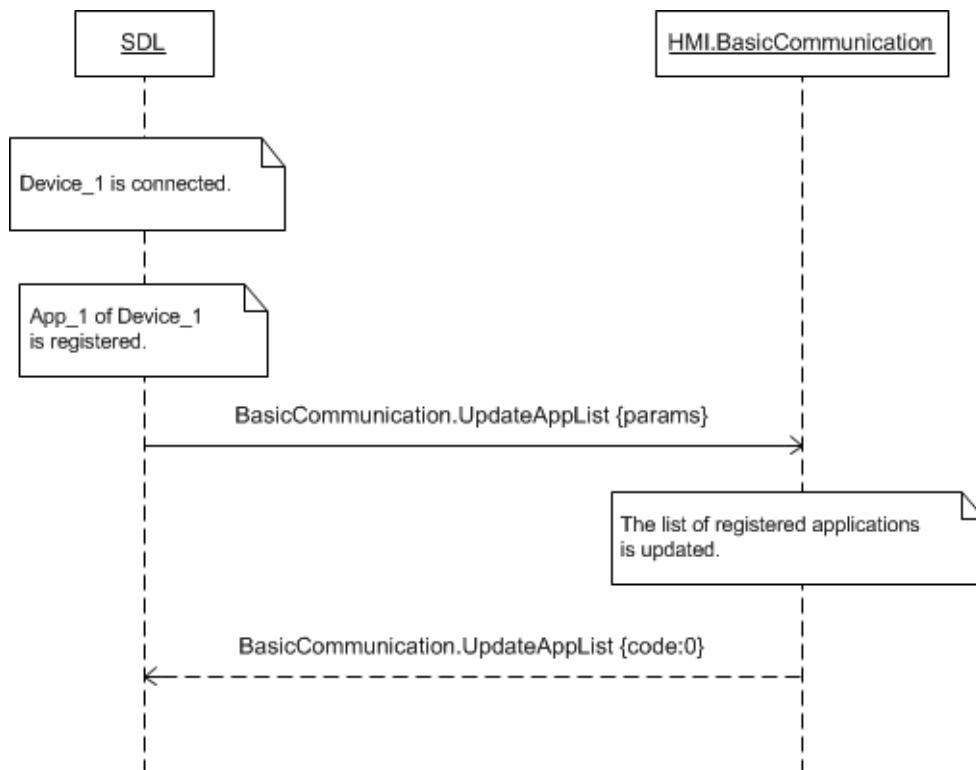
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

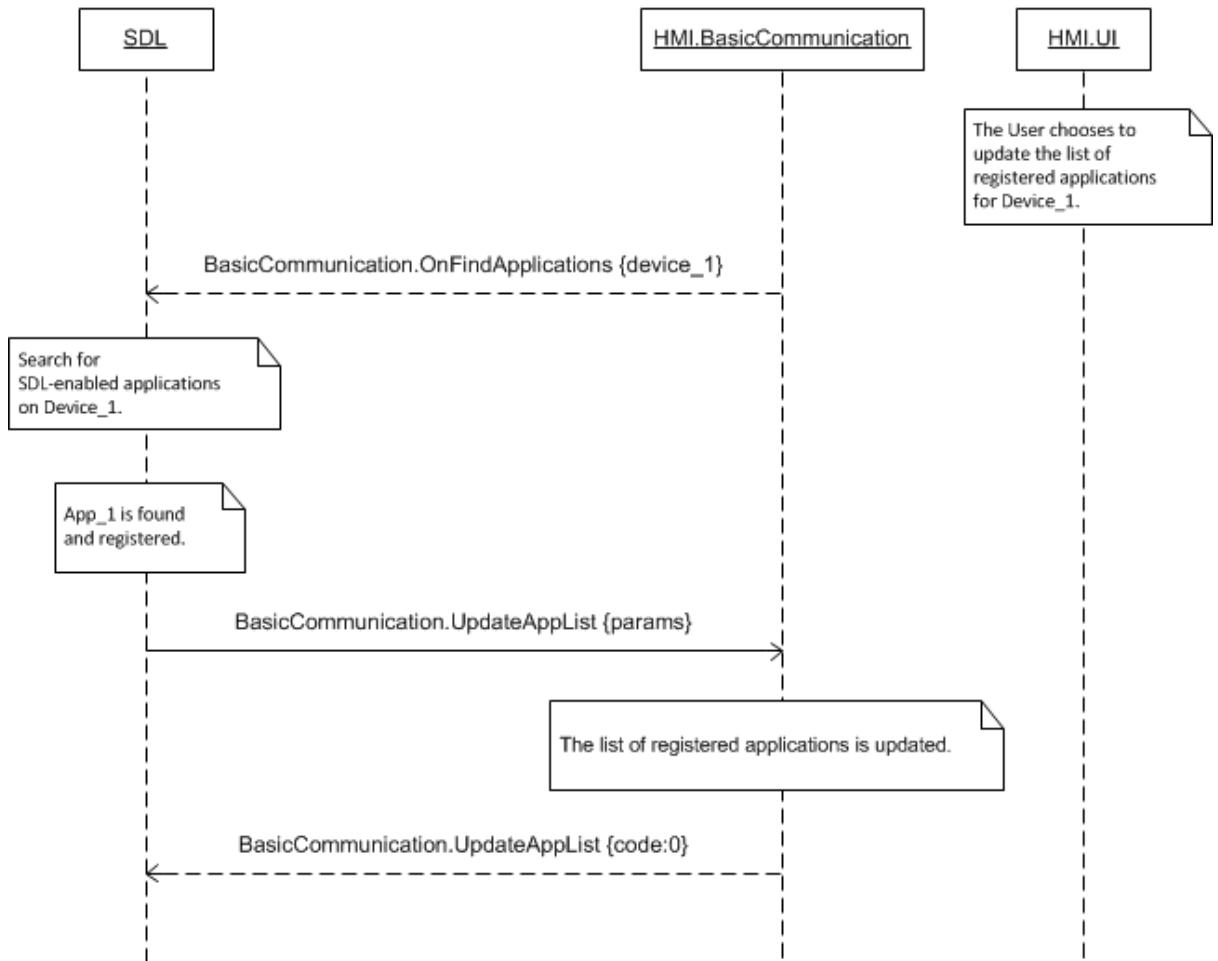
| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|------------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | The HMI's list of registered applications is updated. | JSON response | Regular response | Code: 0 | - |
| Failure | The HMI's list of registered applications is NOT updated | JSON error message | Regular response | Code: 1...23 | Result code must correspond to the failure occurred. |

6.27.4 Sequence Diagrams

6.27.4.1 UpdateAppList after application has just registered with SDL



6.27.4.2 UpdateAppList on User's Request



6.27.5 JSON Messages Examples

6.27.5.1 Request

```
{
    "id" : 75,
    "jsonrpc" : "2.0",
    "method" : "BasicCommunication.UpdateAppList",
    "params" :
    {
        "applications" :
        [
            {
                "appName" : "Beautiful Sound",
                "ngnMediaScreenAppName" : "BeauSo",
                "deviceName" : "Jerry's Phone",
                "appID" : 65544,
                "hmiDisplayLanguageDesired" : DE-DE,
                "isMediaApplication" : true
            },
            {
                "appName" : "Go Travel",
                "icon" : "tmp/SDL/app/Go_Travel/icon.png",
                "deviceName" : "XT910",
                "appID" : 65545,
                "hmiDisplayLanguageDesired" : EN-US,
                "isMediaApplication" : false,
                "appType" : INFORMATION
            }
        ]
    }
}
```

6.27.5.2 Response

```
{  
    "id" : 75,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "BasicCommunication.UpdateAppList"  
    }  
}
```

6.27.5.3 Error message

```
{  
    "id" : 75,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 22,  
        "message" : "During the API call the unknown error has occurred.",  
        "data" :  
        {  
            "method" : "BasicCommunication.UpdateAppList"  
        }  
    }  
}
```

6.27.6 D-Bus Messages Examples

6.27.6.1 Request

```
[REDACTED]
```

6.27.6.2 Response

```
[REDACTED]
```

6.27.6.3 Failure

```
[REDACTED]
```

6.28 OnSDLPersistenceComplete

6.28.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about the data persistence are completed |

SDL sends OnSDLPersistenceComplete right after all the data persistence operations are completed.

Information:

HMI must initiate SDL to start persistence operations by sending `OnExitAllApplications{SUSPEND}`.

HMI must:

- Wait for OnSDLPersistenceComplete before sending `OnExitAllApplications{IGNITION_OFF}`

6.28.2 Sequence Diagrams

6.28.2.1 *OnSDLPersistenceComplete*

6.28.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "BasicCommunication.OnSDLPersistenceComplete"  
}
```

6.28.4 D-Bus Messages Examples



7 UI Component Description

7.1 IsReady

7.1.1 Description

| | |
|----------|-----------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Know if UI module is ready. |

The request comes after HMI's readiness is confirmed via [OnReady](#) notification. SDL requires the information about whether the display is physically present on HU and if so whether it is working.

Note:

If UI module is responded to be unavailable, SDL will not further send the requests related to it.

7.1.2 Request

7.1.2.1 Behavior

HMI must:

- Check whether UI module is present and ready
- Respond correspondingly to results of this check.

7.1.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

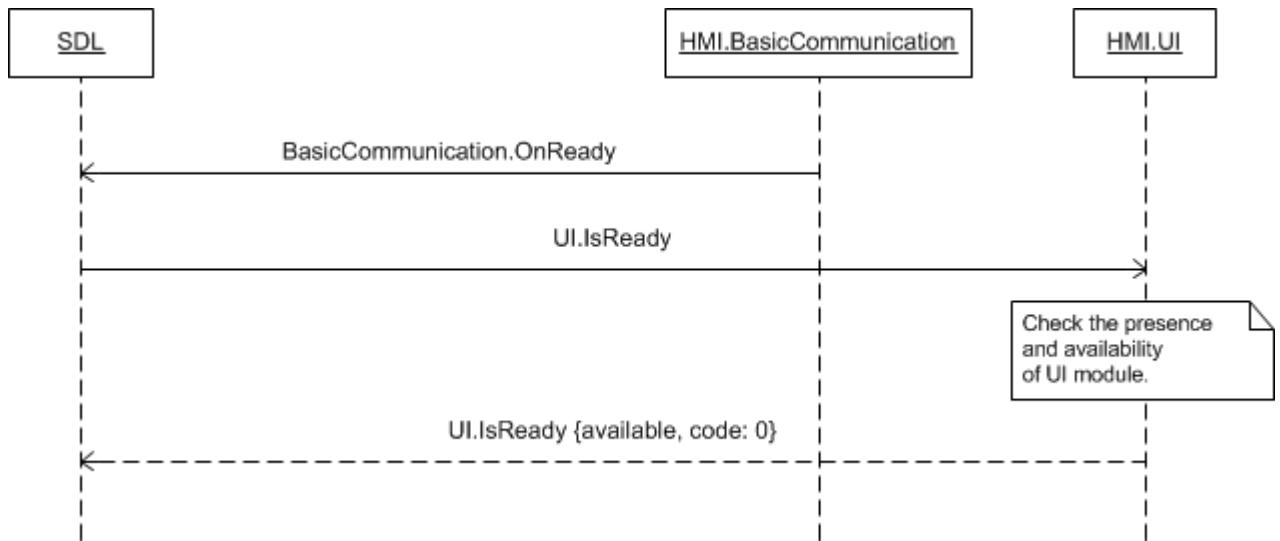
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|------------------|-----------------------|--|
| | | WebSocket | D-Bus | | |
| Success | HMI has the information about UI availability. | JSON response | Regular response | available, Code: 0 | See the table below. |
| Failure | HMI does NOT have the information about UI availability or some failure occurred. | JSON error message | Regular response | Code: 1...23 | Result code must correspond to the failure occurred. |

7.1.3.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|---|
| availabe | Boolean | true | Must be - 'true' if UI is present and ready - 'false' if not. |

7.1.4 Sequence Diagrams

7.1.4.1 UI.IsReady and preceding OnReady



7.1.5 JSON Messages Examples

7.1.5.1 Request

```
{
    "id" : 8,
    "jsonrpc" : "2.0",
    "method" : "UI.IsReady"
}
```

7.1.5.2 Response

```
{
    "id" : 8,
    "jsonrpc" : "2.0",
    "result" :
    {
        "available" : false,
        "code" : 0,
        "method" : "UI.IsReady"
    }
}
```

7.1.5.3 Error message

```
{
    "id" : 8,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "Invalid data",
        "data" :
        {
            "method" : "UI.IsReady"
        }
    }
}
```

7.1.6 D-Bus Messages Examples

7.1.6.1 Request

7.1.6.2 Response

| |
|--|
| |
|--|

7.1.6.3 Failure

| |
|--|
| |
|--|

7.2 GetCapabilities

7.2.1 Description

| | |
|----------|----------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Learn the UI capabilities of HU. |

Once UI module is confirmed to be ready (via response to [IsReady](#) RPC) SDL starts discovering its capabilities via `GetCapabilities` and `GetSupportedLanguages` RPCs.

The response to `UI.GetCapabilities` is assumed to bring all the information required for correct configuring the ‘display’ requests to UI.

7.2.2 Request

7.2.2.1 Behavior

HMI must:

- Check the UI capabilities of:
 - Display: its type, the fields supported, the format of media clock supported, whether displaying images is supported and of what type if so.
 - Zone the UI is located in (front/back).
 - Soft buttons: whether the notifications on events of button depress/release and on button presses of short/long are supported.
 - Audio capturing: the sampling rate, bits per sample, audio type supported.
- Respond correspondingly to results of this check.

Note:

The expected UI capabilities are described in the section [7.2.3.1 Parameters](#) linked to the corresponding structures and enumerations with more detailed information.

7.2.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|------------------|--|--|
| | | WebSocket | D-Bus | | |
| Success | HMI has the information about UI capabilities. | JSON response | Regular response | displayCapabilities, hmiZoneCapabilities, softButtonCapabilities, audioPassThruCapabilities, Code: 0 | See section 7.2.3.1 Parameters . |
| Failure | Some failure occurred. | JSON error message | Regular response | Code: 1...23 | Result code must correspond to the failure occurred. |

7.2.3.1 Parameters

| Param Name | Type | Mandatory | Description |
|---------------------------|----------------------------------|-----------|---|
| displayCapabilities | Common.DisplayCapabilities | true | The capabilities of HMI's display: its type, supported textfields, whether the graphics displaying is supported, the supported formats of media clock. See DisplayCapabilities . |
| softButtonCapabilities | Common.SoftButtonCapabilities | false | Must be returned if the platform supports on-screen soft buttons. Contains the soft buttons capabilities: whether the up/down events, long/short press, referencing image are supported. See SoftButtonCapabilities . |
| hmiZoneCapabilities | Common.HmiZoneCapabilities | true | Specifies HMI Zones in the vehicle (front/back). See HmiZoneCapabilities . |
| audioPassThruCapabilities | Common.AudioPassThruCapabilities | true | Specifies the capabilities of audio capturing: sampling rate, bits per sample, audio type. See AudioPassThruCapabilities . |

7.2.3.2 DisplayCapabilities Structure

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|-------------------------|-----------|---|--|
| displayType | Common.DisplayType | true | - | The type of the display that is installed on HU. See DisplayType . |
| textFields | Common.TextField | true | Array = true minsize = 0 maxsize = 100 | A set of all fields that support displaying text data. If there are no textfields supported HMI must send the empty array. See TextField . |
| imageFields | Common.ImageField | false | Array = true minsize = 1 maxsize = 100 | A set of all fields that support images. See ImageField |
| mediaClockFormats | Common.MediaClockFormat | true | Array = true minsize = 1 maxsize = 100 | A set of all supported formats of the media clock. See MediaClockFormat . |
| imageCapabilities | Common.ImageType | false | Array = true minsize = 0 maxsize = 2 | The array of supported image types (static and/or dynamic). The empty array should be returned if the platform does not support displaying images. See ImageType . |
| graphicSupported | Boolean | true | - | Must be: - 'true' if display's persistent screen supports referencing a static or dynamic image. - 'false' if not. |
| templatesAvailable | String | true | Array = true minsize = 0 maxsize = 100 maxlength = 100 | A set of all predefined persistent display templates available on headunit. To be referenced in SetDisplayLayout. |

| Param Name | Type | Mandatory | Additional | Description |
|---------------------------|---------------------|-----------|--------------------------------|---|
| screenParams | Common.ScreenParams | false | - | A set of all parameters related to a prescribed screen area (e.g. for video / touch input). |
| numCustomPresetsAvailable | Integer | false | minvalue = 1 maxvalue = 100 | |

[Back to Parameters](#)

7.2.3.3 DisplayType Enumeration

| Element name | Short Description |
|--------------|---|
| CID | Center Information Display. This display type provides a 2-line x 20 character "dot matrix" display. |
| TYPE2 | TYPE II display. 1 line older radio head unit. |
| TYPE5 | TYPE V display Old radio head unit. |
| NGN | Next Generation Navigation display. |
| GEN2_8_DMA | GEN-2, 8 inch display. |
| GEN2_6_DMA | GEN-2, 6 inch display. |
| MFD3 | 3 inch GEN1.1 display |
| MFD4 | 4 inch GEN1.1 display |
| MFD5 | 5 inch GEN1.1 display |
| GEN3_8-INCH | GEN-3, 8 inch display. |

[Back to DisplayCapabilities](#)

7.2.3.4 TextFieldName Enumeration

| Element name | Short Description |
|--------------|--|
| mainField1 | The text that must be displayed in a single or upper display line. If this value is not set, the text of mainField1 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show , section 7.5. |
| mainField2 | The text that must be displayed on the second display line. If this text is not set, the text of mainField2 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show , section 7.5. |
| mainField3 | The text that must be displayed on the second "page" first display line. If this text is not set, the text of mainField3 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show , section 7.5. |
| mainField4 | The text that must be displayed on the second "page" second display line. If this text is not set, the text of mainField4 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show , section 7.5. |
| statusBar | The text is placed in the status bar area. Note: This relates to navigation displays If this parameter is omitted, the status bar text must remain unchanged. If this parameter is an empty string, the field must be cleared. If provided and the display has no status bar, this parameter must be ignored. Applies to Show , section 7.5. |

| Element name | Short Description |
|---------------------------|--|
| mediaClock | <p>Text value for MediaClock field. Shall arrive in the form as described in the MediaClockFormat enumeration.</p> <p>If this text is set, any automatic media clock updates previously set with SetMediaClockTimer must be stopped.</p> <p>Applies to Show, section 7.5.</p> |
| mediaTrack | <p>The text that should be displayed in the track field. This field should be valid only for media applications on.</p> <p>If this text is not set, the text of mediaTrack must stay unchanged.</p> <p>If this text is empty "", the field must be cleared.</p> <p>Applies to Show, section 7.5.</p> |
| alertText1 | <p>The text that must be displayed in the top field of the display during the Alert.</p> <p>Applies to Alert, section 7.4.</p> |
| alertText2 | <p>The text that must be displayed in the bottom field of the display during the Alert.</p> <p>Applies to Alert, section 7.4.</p> |
| alertText3 | <p>The optional third line of the alert text field.</p> <p>Applies to Alert, section 7.4.</p> |
| scrollableMessageBody | <p>The long form body of text that can include newlines and tabs.</p> <p>Applies to ScrollableMessage, section 7.19.</p> |
| initialInteractionText | <p>Must be displayed when the interaction begins. The text must be displayed on the first line of a multiline display, and must be centered.</p> <p>Applies to PerformInteraction, section 7.12.</p> |
| navigationText1 | <p>The text that must be displayed on the first line of navigation text.</p> <p>Applies to ShowConstantTBT, section 12.3.</p> |
| navigationText2 | <p>The text that must be displayed on the second line of navigation text.</p> <p>Applies to ShowConstantTBT, section 12.3.</p> |
| ETA | <p>Estimated Time of Arrival for navigation.</p> <p>Applies to ShowConstantTBT, section 12.3.</p> |
| totalDistance | <p>Total distance to destination for navigation.</p> <p>Applies to ShowConstantTBT, section 12.3.</p> |
| navigationText | <p>Navigation text for UpdateTurnList.</p> <p>Applies to Turn, section 12.4.</p> |
| audioPassThruDisplayText1 | <p>The first line of text that must be displayed during audio capture.</p> <p>Applies to PerformAudioPassThru, section 7.20.</p> |
| audioPassThruDisplayText2 | <p>The second line of text that must be displayed during audio capture.</p> <p>Applies to PerformAudioPassThru, section 7.20.</p> |
| sliderHeader | <p>The text that must be displayed on the header of slider.</p> <p>Applies to Slider, section 7.18.</p> |
| sliderFooter | <p>The text that must be displayed on the footer of slider.</p> <p>Applies to Slider, section 7.18.</p> |
| notificationText | <p>The text that must be displayed to notify the User on some event.</p> <p>Applies to ShowNotification, section 7.31.</p> |
| menuName | <p>Primary text for Choice.</p> <p>Applies to PerformInteraction, section 7.12.</p> |
| secondaryText | <p>Secondary text for Choice.</p> <p>Applies to PerformInteraction, section 7.12.</p> |
| tertiaryText | <p>Tertiary text for Choice.</p> <p>Applies to PerformInteraction, section 7.12.</p> |
| menuTitle | <p>Optional text to label an app menu button (for certain touchscreen platforms).</p> |
| timeToDestination | <p>The line to display the time to destination.</p> <p>Applies to ShowConstantTBT, section 12.3.</p> |
| turnText | Currently not used. |

[Back to Display Capabilities](#)

7.2.3.5 ImageField Structure

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|----------------------------|------------------|--|---|
| name | Common.ImageFileDialogName | true | - | The name that identifies the field. See ImageFieldName. |
| imageTypeSupported | Common.FileType | false | array = true minsize = 1 maxsize = 100 | The image types that are supported in this field. See FileType. |
| imageResolution | Common.ImageResolution | false | - | The image resolution of this field. |

7.2.3.6 ImageFieldName Enumeration

| Element name | Value | Short Description |
|-----------------------------|--------------|--|
| softButtonImage | 0 | The image field for SoftButton |
| choiceImage | 1 | The first image field for Choice |
| choiceSecondaryImage | 2 | The secondary image field for Choice |
| vrHelpItem | 3 | The image field for vrHelpItem |
| turnIcon | 4 | The image field for Turn |
| menuIcon | 5 | The image field for the menu icon in SetGlobalProperties |
| cmdIcon | 6 | The image field for AddCommand |
| appIcon | 7 | The image field for the app icon (set by setAppIcon) |
| graphic | 8 | The image field for Show |
| showConstantTBTIcon | 9 | The primary image field for ShowConstantTBT |
| showConstantTBTNextTurnIcon | 10 | The secondary image field for ShowConstantTBT |
| locationImage | 11 | The optional image of a destination / location |

7.2.3.7 FileType Enumeration

| Element name | Value | Short Description |
|---------------------|--------------|--------------------------|
| GRAPHIC_BMP | 0 | |
| GRAPHIC_JPEG | 1 | |
| GRAPHIC_PNG | 2 | |
| AUDIO_WAVE | 3 | |
| AUDIO_MP3 | 4 | |
| AUDIO_AAC | 5 | |
| BINARY | 6 | |
| JSON | 7 | |

7.2.3.8 ImageResolution Structure

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|-------------|------------------|----------------------------------|------------------------------|
| resolutionWidth | Integer | true | minvalue = 1 maxvalue = 10000 | The image resolution width. |
| resolutionHeight | Integer | true | minvalue = 1 maxvalue = 10000 | The image resolution height. |

7.2.3.9 MediaClockFormat Enumeration

| Element name | Short Description |
|---------------------|--------------------------|
|---------------------|--------------------------|

| Element name | Short Description |
|---------------------|--|
| CLOCK1 | <p>minutesFieldWidth = 2; minutesFieldMax = 19; secondsFieldWidth = 2; secondsFieldMax = 99;</p> <p>maxHours = 19; maxMinutes = 59; maxSeconds = 59;</p> <p>Is used for Type II, NGN and CID head units.</p> |
| CLOCK2 | <p>minutesFieldWidth = 3; minutesFieldMax = 199; secondsFieldWidth = 2; secondsFieldMax = 99;</p> <p>maxHours = 59; maxMinutes = 59; maxSeconds = 59;</p> <p>Is used for Type V head units.</p> |
| CLOCK3 | <p>minutesFieldWidth = 2; minutesFieldMax = 59; secondsFieldWidth = 2; secondsFieldMax = 59;</p> <p>maxHours = 9; maxMinutes = 59; maxSeconds = 59;</p> <p>Is used for GEN1.1 (i.e. MFD3/4/5) head units.</p> |
| CLOCKTEXT1 | <p>5 characters possible</p> <p>Format: 1 sp c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space</p> <p>Is used for Type II head unit</p> |
| CLOCKTEXT2 | <p>5 chars possible</p> <p>Format: 1 sp c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space</p> <p>Is used for CID and NGN head unit.</p> |
| CLOCKTEXT3 | <p>6 chars possible</p> <p>Format: 1 sp c c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space</p> <p>Is used for Type V head unit.</p> |
| CLOCKTEXT4 | <p>6 chars possible</p> <p>Format: c : sp c c : c c : sp : colon or space c : character out of following character set: sp 0-9 [letters].</p> <p>Is used for GEN1.1 (i.e. MFD3/4/5) head units.</p> |

[Back to Display Capabilities](#)

7.2.3.10 ImageType Enumeration

| Element name | Short Description |
|---------------------|--|
| STATIC | Static image. The image that is sent as the binary or hex code within the request. |

| Element name | Short Description |
|--------------|---|
| DYNAMIC | Dynamic image. The image that is stored on HMI and just a link to it is further used within requests. |

[Back to DisplayCapabilities](#)

7.2.3.11 ScreenParams Structure

| Param Name | Type | Mandatory | Description |
|---------------------|-------------------------------|-----------|--|
| resolution | Common.ImageResolution | true | The resolution of the prescribed screen area. See ImageResolution . |
| touchEventAvailable | Common.TouchEventCapabilities | false | Types of screen touch events available in screen area. |

7.2.3.12 TouchEventCapabilities Structure

| Param Name | Type | Mandatory | Description |
|----------------------|---------|-----------|-------------|
| pressAvailable | Boolean | true | |
| multiTouchAvailable | Boolean | true | |
| doublePressAvailable | Boolean | true | |

7.2.3.13 SoftButtonCapabilities Structure

| Param Name | Type | Mandatory | Description |
|---------------------|---------|-----------|--|
| shortPressAvailable | Boolean | true | Must be - 'true' if soft buttons support a short press - 'false' if not. See ButtonPressMode for more information. |
| longPressAvailable | Boolean | true | Must be - 'true' if soft buttons support a LONG press - 'false' if not. See ButtonPressMode for more information. |
| upDownAvailable | Boolean | true | Must be - 'true' if soft buttons support "button down" and "button up". - 'false' if not. See ButtonEventMode for more information. |
| imageSupported | Boolean | true | Must be - 'true' if soft buttons support referencing image - 'false' if not. |

[Back to Parameters](#)

7.2.3.14 HmiZoneCapabilities Enumeration

| Element name | Short Description |
|--------------|--|
| FRONT | Indicates UI available to front seat passengers. |
| BACK | Indicates UI available to rear seat passengers. |

[Back to Parameters](#)

7.2.3.15 AudioPassThruCapabilities Structure

| Param Name | Type | Mandatory | Description |
|------------|------|-----------|-------------|
| | | | |

| | | | |
|---------------|----------------------|------|--|
| samplingRate | Common.SamplingRate | true | The number of samples per second. Defines the sampling rate supported by the HU for audio capturing (e.g. 8 kHz, 16 kHz, etc.). See SamplingRate . |
| bitsPerSample | Common.BitsPerSample | true | The number of bits of information in each sample. Defines the quality of audio capturing supported by HU. See BitsPerSample . |
| audioType | Common.AudioType | true | Specifies the type of audio data being requested. See AudioType . |

[Back to Parameters](#)

7.2.3.16 SamplingRate Enumeration

| Element name | Short Description |
|--------------|-------------------------|
| 8KHZ | Sampling rate of 8 kHz |
| 16KHZ | Sampling rate of 16 kHz |
| 22KHZ | Sampling rate of 22 kHz |
| 44KHZ | Sampling rate of 44 kHz |

[Back to AudioPassThruCapabilities](#)

7.2.3.17 BitsPerSample Enumeration

| Element name | Short Description |
|--------------|-------------------|
| 8_BIT | 8 bit per sample |
| 16_BIT | 16 bit per sample |

[Back to AudioPassThruCapabilities](#)

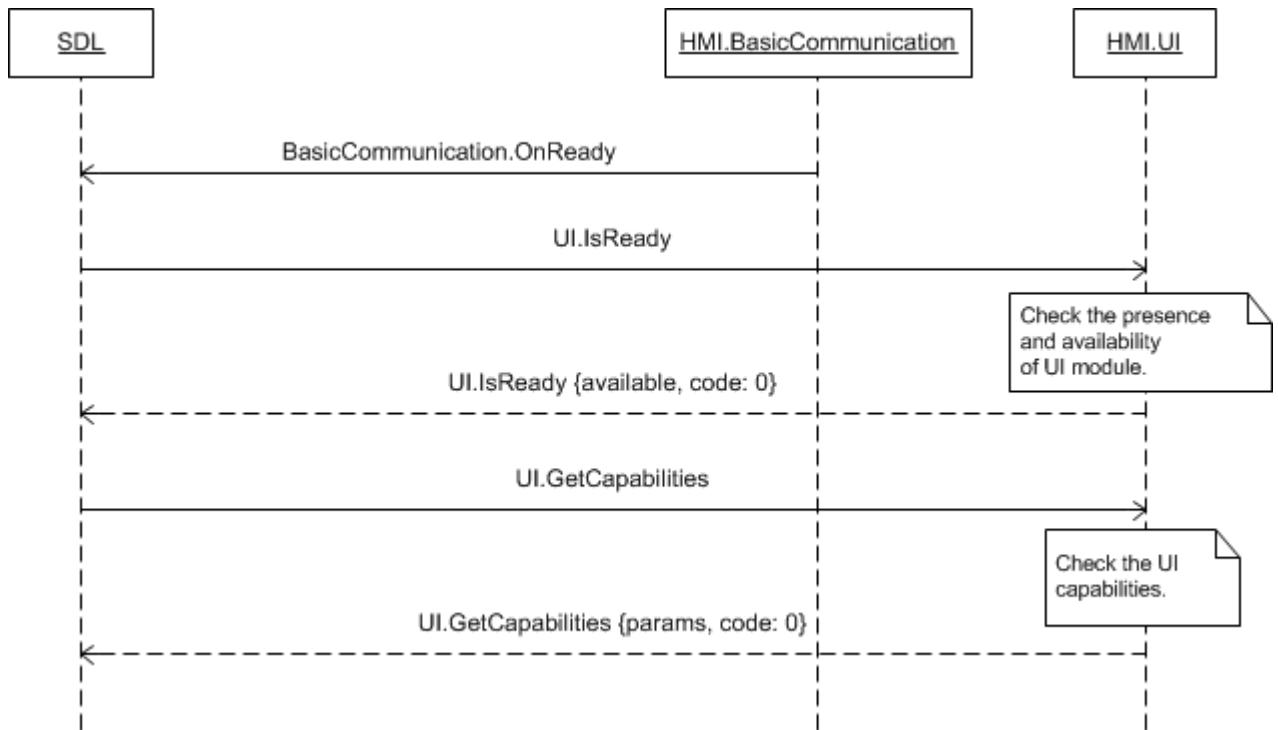
7.2.3.18 AudioType Enumeration

| Element name | Short Description |
|--------------|-----------------------------|
| PCM | Pulse Code Modulated audio. |

[Back to AudioPassThruCapabilities](#)

7.2.4 Sequence Diagrams

7.2.4.1 UI.GetCapabilities and preceding IsReady



7.2.5 JSON Messages Examples

7.2.5.1 Request

```
{
    "id" : 18,
    "jsonrpc" : "2.0",
    "method" : "UI.GetCapabilities"
}
```

7.2.5.2 Response

```
{
    "id" : 18,
    "jsonrpc" : "2.0",
    "result" :
    {
        "displayCapabilities" :
        {
            "displayType" : GEN2_8_DMA,
            "textFields" : [mainField1, mainField2, mediaclock,
                           mediaTrack], alertText1, alertText2,
                           alertText3, scrollableMessageBody,
                           initialInteractionText, navigationText1,
                           navigationText2, audioPassThruDisplayText1,
                           audioPassThruDisplayText2, notificationText]
            "mediaClockFormats" : [CLOCK1, CLOCKTEXT4],
            "graphicSupported" : true,
            "imageCapabilities": [DYNAMIC]
        },
        "softButtonCapabilities" :
        {
            "shortPressAvailable" : true,
            "longPressAvailable" : true,
            "upDownAvailable" : true,
            "imageSupported" : true
        },
    }
}
```

```

    "hmiZoneCapabilities" : FRONT,
    "audioPassThruCapabilities" :
    {
        "samplingRate" : 44KHZ,
        "bitsPerSample" : 8_BIT,
        "audioType" : PCM
    },
    "code" : 0,
    "method" : "UI.GetCapabilities"
}
}

```

7.2.5.3 Error message

```

{
    "id" : 18,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 22,
        "message" : "During API call the unknown error has occurred",
        "data" :
        {
            "method" : "UI.GetCapabilities"
        }
    }
}

```

7.2.6 D-Bus Messages Examples

7.2.6.1 Request

[REDACTED]

7.2.6.2 Response

[REDACTED]

7.2.6.3 Failure

[REDACTED]

7.3 GetSupportedLanguages

7.3.1 Description

| | |
|----------|---------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get the UI supported languages. |

Once UI module is confirmed to be ready (via response to [IsReady](#) RPC) SDL starts discovering its capabilities via [GetCapabilities](#) and [GetSupportedLanguages](#) RPCs.

Response to `UI.GetSupportedLanguages` is assumed to bring the information about what languages are supported for displaying the text information on UI. Having obtained this information SDL will monitor the language parameter within RPCs from mobile application(s) and reject the requests containing language not supported by HMI.

Note:

The list of languages recognized by SDL is provided in the section [7.3.3.2 Language Enumeration](#).

7.3.2 Request

7.3.2.1 Behavior

HMI must:

- Check the UI supported languages
- Respond correspondingly to results of this check.

7.3.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|------------------|--------------------|---|
| | | WebSocket | D-Bus | | |
| Success | HMI has the information about UI supported languages. | JSON response | Regular response | languages, code: 0 | See the table below. |
| Failure | The information about UI supported languages is not available. | JSON error message | Regular response | code: 9 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | The data sent is invalid (invalid JSON syntax) | | | code: 11 | |
| | The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.3.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------------------|-----------|--|--|
| languages | Common.Lang uage | true | Array = true minsize = 1 maxsize = 100 | List of UI supported languages. See Language. |

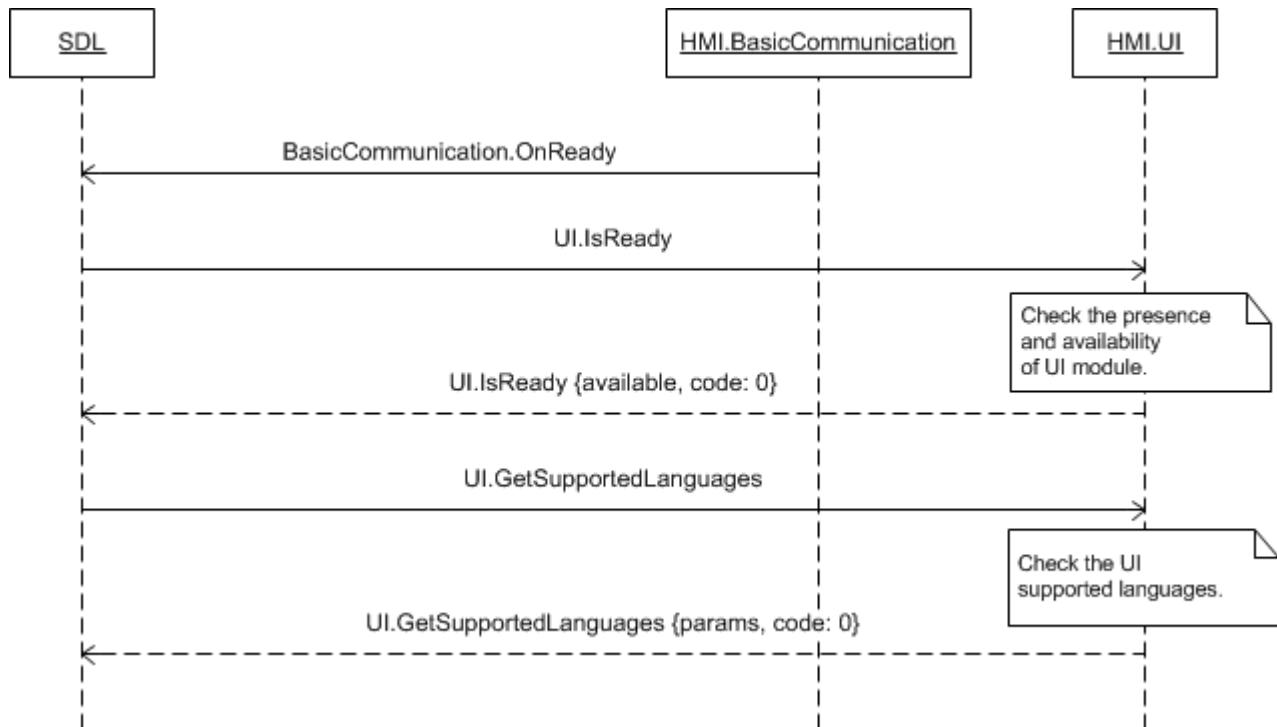
7.3.3.2 Language Enumeration

| Element name | Short Description |
|--------------|--------------------------------|
| AR-SA | Arabic – Saudi Arabia |
| CS-CZ | Czech – Czech Republic |
| DA-DK | Danish – Denmark |
| DE-DE | German – Germany |
| EN-AU | English – Australia |
| EN-GB | English – GB |
| EN-US | English – US |
| ES-ES | Spanish – Spain |
| ES-MX | Spanish – Mexico |
| FR-CA | French – Canada |
| FR-FR | French – France |
| IT-IT | Italian – Italy |
| JA-JP | Japanese – Japan |
| KO-KR | Korean – South Korea |
| NL-NL | Dutch (Standard) – Netherlands |
| NO-NO | Norwegian - Norway |
| PL-PL | Polish – Poland |
| PT-PT | Portuguese – Portugal |

| Element name | Short Description |
|--------------|---------------------|
| PT-BR | Portuguese – Brazil |
| RU-RU | Russian - Russia |
| SV-SE | Swedish – Sweden |
| TR-TR | Turkish – Turkey |
| ZH-CN | Mandarin – China |
| ZH-TW | Mandarin – Taiwan |

7.3.4 Sequence Diagrams

7.1.4.1 GetSupportedLanguages with preceding IsReady



7.3.5 JSON Messages Examples

7.3.5.1 Request

```
{
  "id" : 99,
  "jsonrpc" : "2.0",
  "method" : "UI.GetSupportedLanguages"
}
```

7.3.5.2 Response

```
{
  "id" : 99,
  "jsonrpc" : "2.0",
  "result" :
  {
    "languages" : [AR-SA, DE-DE, EN-GB, EN-US, ES-ES, FR-FR, IT-IT],
    "code" : 0,
    "method" : "UI.GetSupportedLanguages"
  }
}
```

7.3.5.3 Error message

```
{  
    "id" : 99,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 9,  
        "message" : "The requested data is not available",  
        "data" :  
        {  
            "method" : "UI.GetSupportedLanguages"  
        }  
    }  
}
```

7.3.6 D-Bus Messages Examples

7.1.6.1 Request

```
[REDACTED]
```

7.1.6.2 Response

```
[REDACTED]
```

7.1.6.3 Failure

```
[REDACTED]
```

7.4 Alert

7.4.1 Description

| | |
|----------|-------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Display alert message on HMI. |

This RPC represents a request to provide the displayable part of the important information to the User.

Note:

SDL uses the information received via response to [UI.GetCapabilities](#) for forming the request (uses the supported text fields names, provides image for soft buttons once confirmed to be supported by HMI, etc.).

Together with UI.Alert method SDL may also send

- BasicCommunication.PlayTone for turning the User's attention to the message with the help of HMI standard notifying sound
 - TTS.Speak for providing the audible part of the important information to the User.
- More details about around Alert messaging please find in section 7.4.4 Sequence Diagrams.

Note:

If Alert request comes together with PlayTone and Speak RPCs, HMI must:

- At first play the notifying sound
- Then speak the requested text.

7.4.2 Request

7.4.2.1 Behavior

HMI must:

1. Notify about the Alert context via OnSystemContext notification.

2. Display the alert dialog with:

- The text information in the [named fields](#) (up to three fields may be requested)
- Up to four soft buttons (optional)
- Progress indicator showing the time to dialog closing.

Note:

The Alert dialog possible view is provided in the section 7.4.5 Possible Layout.

3. React on requested soft button(s) press with:

- Sending `OnButtonPress/OnButtonEvent` notifications
- Predefined behavior depending on the soft button type:
 - `DEFAULT_ACTION` type button press is assumed to close the alert dialog and return to the previously displayed screen
 - `KEEP_CONTEXT` type button press is assumed to renew the timeout for the alert dialog being displayed.
 - `STEAL_FOCUS` type button press is assumed to
 - Activate the application that has sent the Alert message while being not active on HMI
 - Return to the screen previously displayed if the application that sent the Alert RPC is active on HMI.

Note:

The diagrams describing the sequence of button press, notifications, messages is provided in section 7.4.4 Sequence Diagrams.

4. Display the alert dialog and then close it

- By the timeout requested by SDL
- Or upon the events:
 - One of the SDL-defined soft buttons press
 - HMI-defined 'Close'/'Back'/'Return' button press
 - VR activation
 - Another application activation or switching to any of HMI screens
 - Another RPC of a higher priority arrival.

5. Respond correspondingly after the definite event occurred (timeout, soft button press, etc.).

6. Notify via `OnSystemContext` about the system context is changed to main.

Note:

The detailed information about the events occurred and the applicable response codes are described in the section [7.4.3 Response](#).

HMI may:

Display the HMI-defined 'Close'/'Back'/'Return' button for providing the User with the possibility to dismiss the dialog. When such button is pressed

- The notification `OnButtonPress/OnButtonEvent` should not be sent to SDL
- The dialog must be closed
- The Alert RPC must be responded.

7.4.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|------------------------|-----------|--|---|
| alertStrings | Common.TextFieldStruct | true | Array = true Minsize = 0 Maxsize = 3 | Array of lines of alert text fields that must be displayed. See TextFieldStruct . Uses alertText1, alertText2, alertText3 from TextFieldName . If empty array is sent, the displayed alert message should not contain any text. |
| duration | Integer | true | Minvalue = 3000 Maxvalue = 10000 | Timeout in milliseconds. |
| softButtons | Common.SoftButton | false | Array = true Minsize = 0 Maxsize = 4 | Application-defined soft buttons. If omitted or the empty array is sent, the displayed alert message should not have any soft buttons. See section 5. Soft Buttons Behavior on HMI |
| progressIndicator | Boolean | false | - | If supported on the given platform, the alert dialog should include some sort of animation indicating that loading of a feature is progressing. e.g. a spinning wheel or hourglass, etc. |
| alertType | Common.AlertType | true | - | Defines if only UI or BOTH portions of the Alert request are being sent to HMI Side |
| appID | Integer | true | - | ID of the application related to this RPC. |

7.4.2.3 *TextFieldStruct* Structure

| Param Name | Type | Mandatory | Additional | Description |
|------------|----------------------|-----------|-----------------|--|
| fieldName | Common.TextFieldName | true | - | The name of the field where the text must be displayed in. |
| fieldText | String | true | Maxlength = 500 | The text to be displayed. |

7.4.2.4 *TextFieldName* Enumeration

Only the text fields applicable to Alert RPC are described within this section. All the text fields names recognized by SDL are described in the section 13.1.14 *TextFieldName*.

| Element name | Short Description |
|--------------|--|
| alertText1 | The first line in the top of the alert dialog for displaying the text. |
| alertText2 | The second line of the alert dialog for displaying the text. |
| alertText3 | The third line of the alert dialog for displaying the text. |

7.4.2.4 *AlertType* Enumeration

| Element name | Short Description |
|--------------|--|
| UI | Alert RPC needs to be displayed only (no TTS portion). |

| Element name | Short Description |
|--------------|--|
| BOTH | Alert RPC has TTS portion in addition to displayed pop-up. |

7.4.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

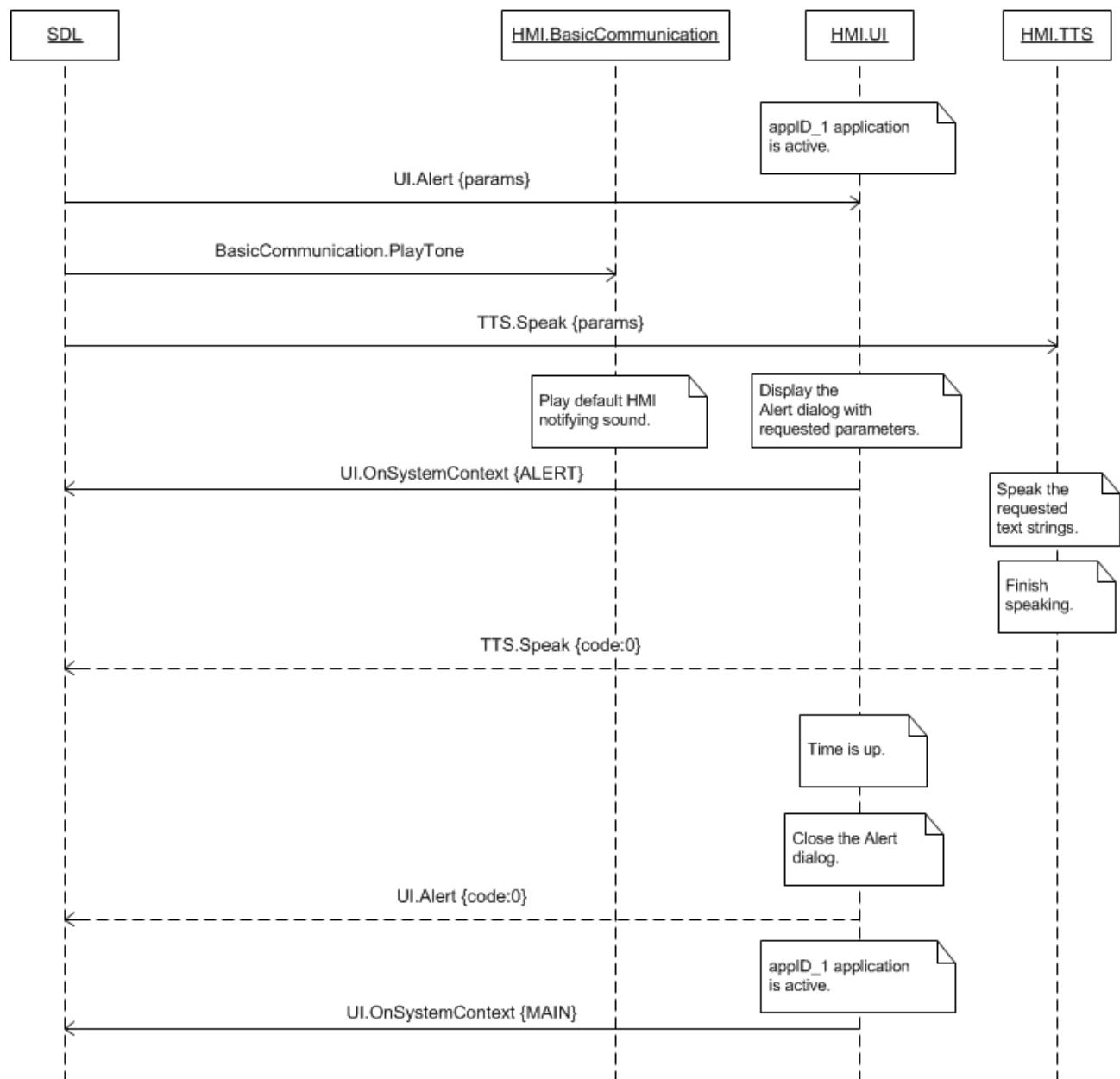
| Result | Description | Message type | | Message Params | Notes |
|---------|--|---------------------------|------------------|--------------------------|---|
| | | WS | D-Bus | | |
| Success | <ul style="list-style-type: none"> - The alert dialog has been displayed AND - Now is closed by one of the following reasons: <ul style="list-style-type: none"> • The timeout. • The soft button with DEFAULT_ACTION system action press. • The soft button with STEAL_FOCUS system action press. | JSON response | Regular response | code: 0 | - |
| Failure | <ul style="list-style-type: none"> - The alert dialog has been displayed AND - Now is aborted by: <ul style="list-style-type: none"> • HMI-defined 'Close' button press. • VR session. • Another application activation or switching to HMI screen • Another RPC of a higher priority. | JSON <u>error</u> message | Regular response | code: 5 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | The Alert RPC is rejected because the higher priority RPC is now displayed on HMI. | | | tryAgainTime, code: 4 | See section 7.4.3.1 Parameters |
| | The data sent is invalid . | | | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.4.3.1 Parameters

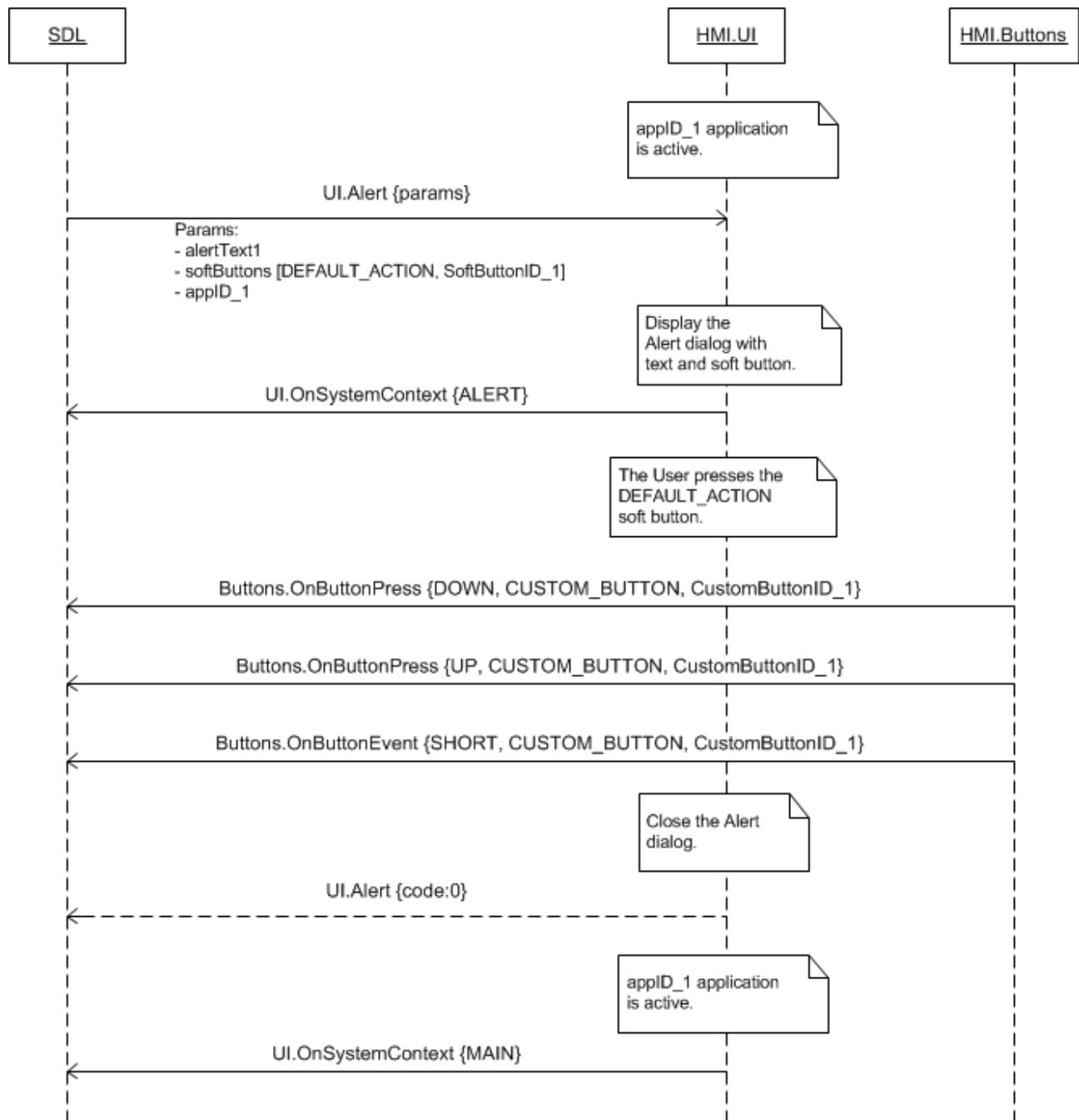
| Param Name | Type | Mandatory | Additional | Description |
|--------------|---------|-----------|-----------------------------------|--|
| tryAgainTime | Integer | false | Minvalue = 0 Maxvalue = 360000 | Amount of time (in milliseconds) that SDL must wait before resending an alert. Must be returned <u>only</u> with REJECTED (code: 4) result. |

7.4.4 Sequence Diagrams

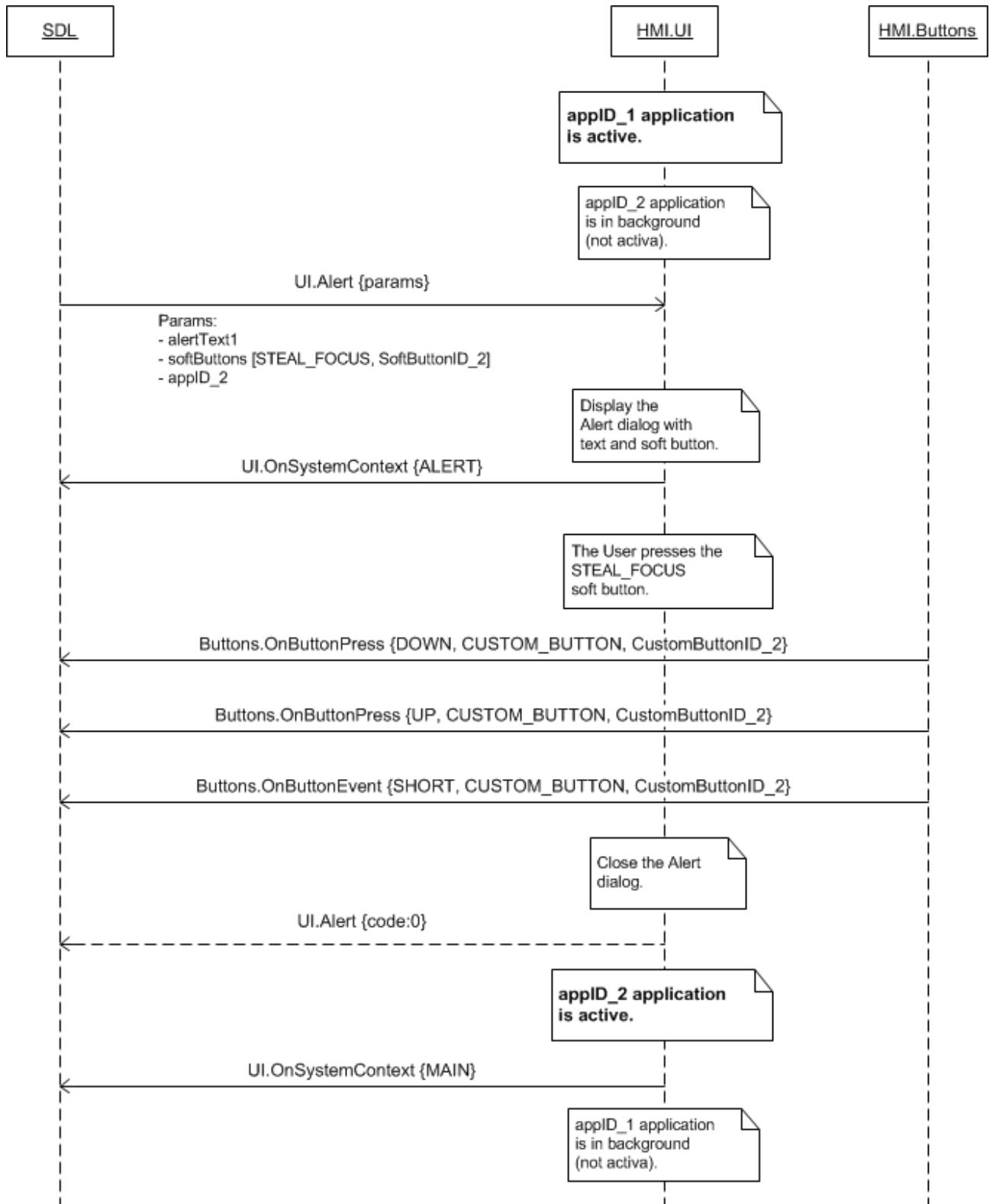
7.4.4.1 Alert together with PlayTone and Speak RPCs and closed by the timeout



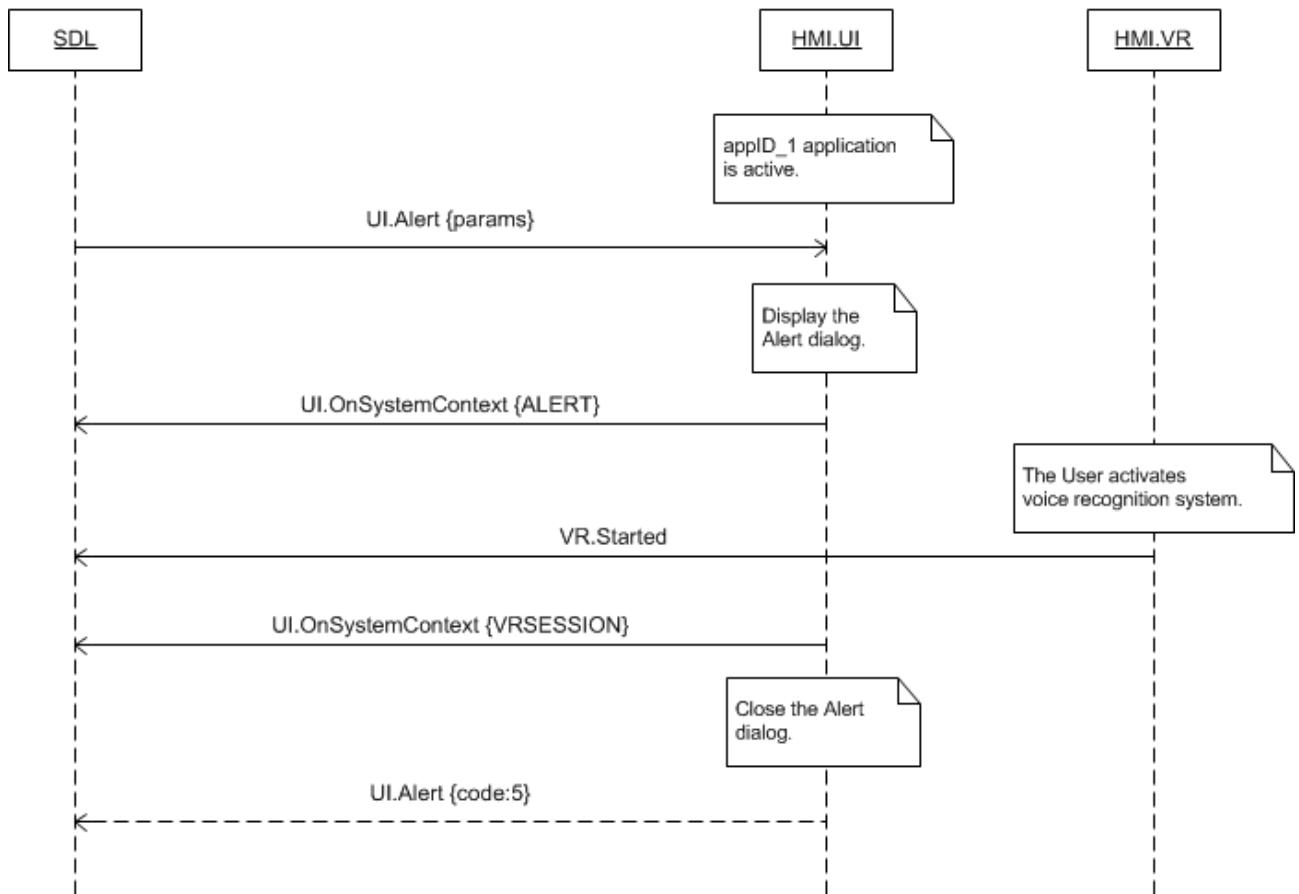
7.4.4.2 Alert displayed and closed by DEFAULT_ACTION soft button press



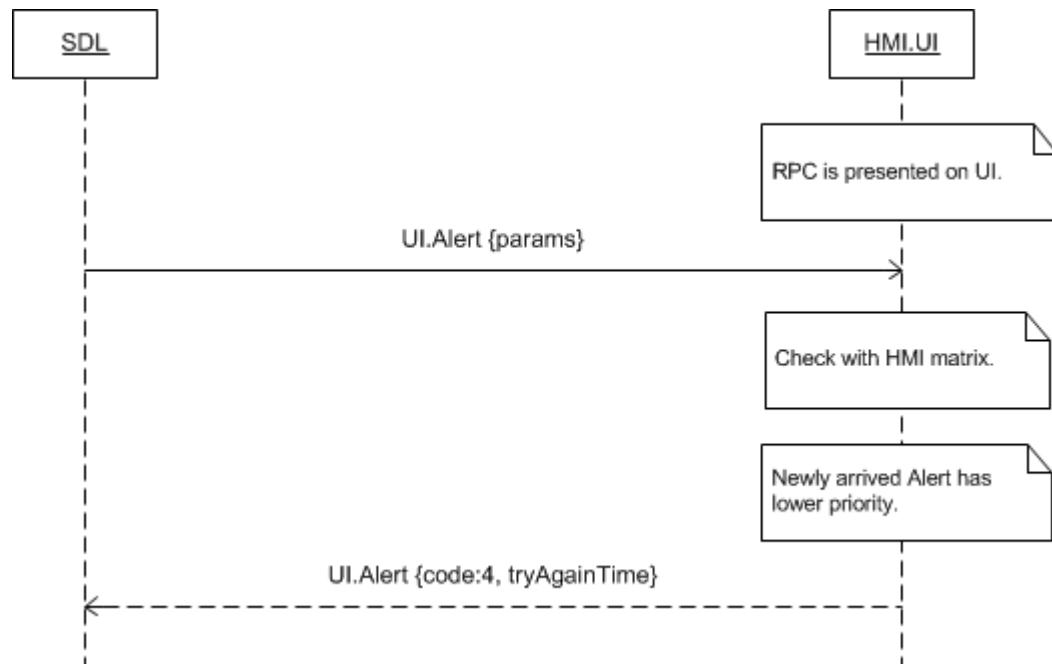
7.4.4.3 Alert for non-activa application displayed and closed by STEAL_FOCUS soft button press



7.4.4.4 Alert is displayed and then aborted by VR session started



7.4.4.5 Alert is rejected because of RPC of a higher priority currently presented on UI

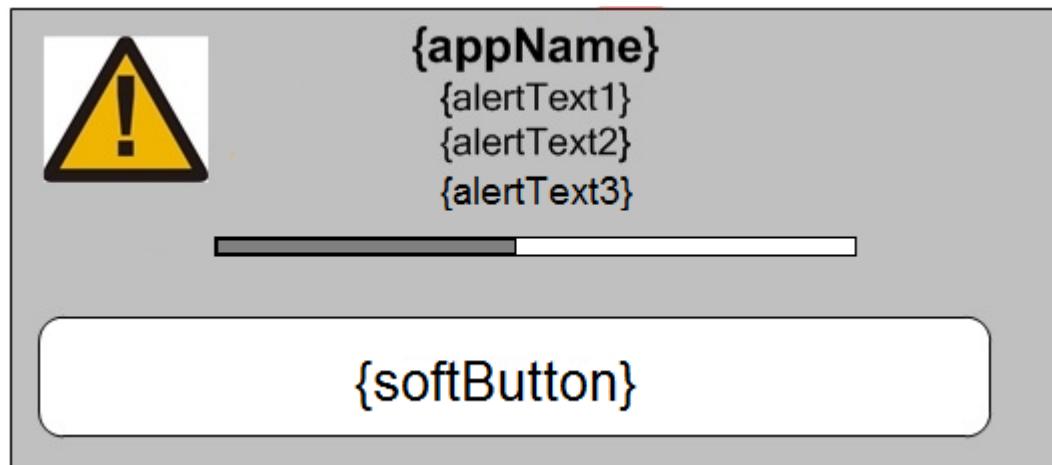


7.4.5 Possible Layout

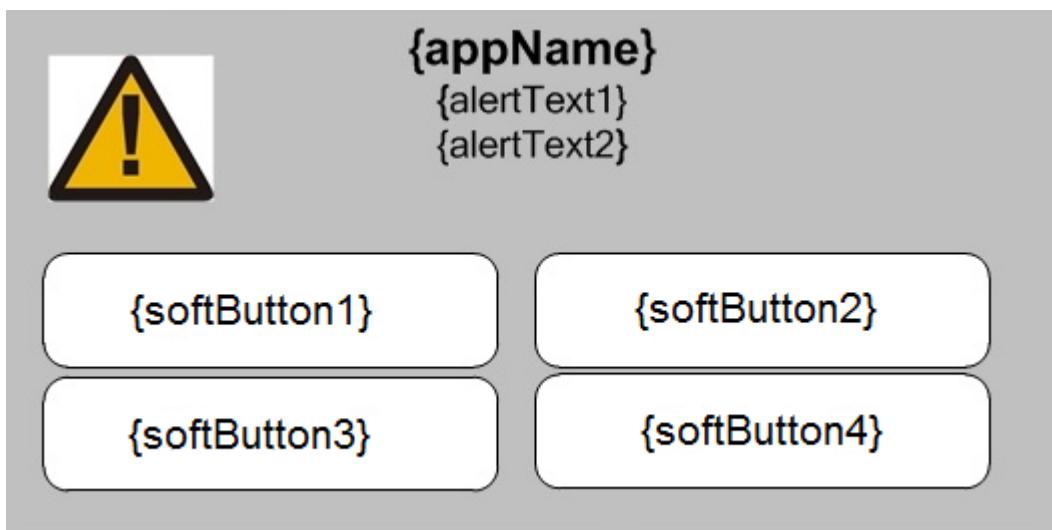
7.4.5.1 Alert dialog with text fields and NO soft buttons



7.4.5.2 Alert dialog with text fields, one soft button and progress indicator



7.4.5.3 Alert dialog with text fields and four soft buttons



7.4.6 JSON Messages Examples

7.4.6.1 Request

```
{  
    "id" : 92,  
    "jsonrpc" : "2.0",  
    "method" : "UI.Alert",  
    "params" :  
    {  
        "alertStrings" :  
        [  
            {  
                "fieldName" : alertText1,  
                "fieldText" : "WARNING"  
            },  
            {  
                "fieldName" : alertText2,  
                "fieldText" : "Hard weather conditions"  
            }  
        ],  
        "duration" : 5000,  
        "softButtons" :  
        {  
            "type" : TEXT,  
            "text" : "OK",  
            "softButtonID" : 697,  
            "systemAction" : DEFAULT_ACTION  
        },  
        "appID" : 65539  
    }  
}
```

7.4.6.2 Response

```
{  
    "id" : 92,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "UI.Alert"  
    }  
}
```

7.4.6.3 Error message

```
{  
    "id" : 92,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 4,  
        "message" : "The requested command was rejected.",  
        "data" :  
        {  
            "tryAgainTime" : 10000,  
            "method" : "UI.Alert"  
        }  
    }  
}
```

7.4.7 D-Bus Messages Examples

7.4.7.1 Request

| |
|--|
| |
|--|

7.4.7.2 Response

| |
|--|
| |
|--|

7.4.7.3 Failure

| |
|--|
| |
|--|

7.5 Show

7.5.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Update the application persistent display on HMI. |

With this RPC SDL requests HMI to update the supported fields for the application persistent display.

Note:

SDL uses the information received via response to [UI.GetCapabilities](#) for forming the request (uses the supported text fields names, provides the images of supported type, etc.).

The request may follow

- After the application is first activated on HMI
- For the application that is not currently active on HMI

7.5.2 Request

7.5.2.1 Behavior

HMI must:

1. Store the data sent within RPC
2. Be able to correlate the stored information with the application identified with appID.
3. Display the requested data (text, image, soft buttons, presets):
 - Right after the Show RPC arrived if the application is active on HMI
 - After the application is activated if the Show RPC:
 - Arrives while the application is not active on HMI.
 - Arrived for the active application that has then been deactivated.

Note:

Together with items requested with Show, HMI must also display the in-application commands menu (e.g. named 'Options') providing the possibility for the User to enter this menu (empty by default) and to choose among its elements (once added via [AddCommand](#) and/or [AddSubMenu](#)).

Note:

Pressing the soft button of any system action must cause

- NO changes for application related persistent display on HMI
- Sending OnButtonPress/OnButtonEvent notifications.

Note:

The sequence diagrams describing the expected HMI behavior are provided in the section 7.5.4 Sequence Diagrams

7.5.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------------|------------------------|-----------|--|---|
| showStrings | Common.TextFieldStruct | true | Array = true Minsize = 0 Maxsize = 7 | <p>Array of text lines to be displayed in the named text fields.</p> <p>The previously displayed text must not be changed on this RPC receipt if:</p> <ul style="list-style-type: none"> - The corresponding textfield parameter is not provided within a request. - The empty array is sent. <p>The previously displayed text must be cleared if the corresponding textfield parameter has the value of "" (empty string).</p> <p>See TextFieldStruct, TextFieldName</p> |
| alignment | Common.TextAlignment | false | - | <p>Specifies the way mainField1 and mainField2 texts must be aligned on the display.</p> <p>If omitted, texts must be centered.</p> <p>See TextAlignment.</p> |
| graphic | Common.Image | false | - | <p>Path to the optional dynamic image or the static binary image itself.</p> <p>If omitted, the displayed graphic must not be changed.</p> <p>See Image.</p> |
| secondaryGraphic | Common.Image | false | - | <p>Image struct determining whether static or dynamic secondary image to display in app.</p> <p>If omitted on supported displays, the displayed secondary graphic shall not change.</p> |
| softButtons | Common.SoftButton | false | Array = true Minsize = 0 Maxsize = 8 | <p>App defined soft buttons.</p> <p>Pressing the soft button with defined system action must cause no changes on HMI.</p> <p>For all of the soft buttons the notifications OnButtonPress/OnButtonEvent must be provided by HMI.</p> <p>If omitted or the empty array is sent, the currently displayed soft button values must not be changed.</p> <p>See SoftButton.</p> |
| customPresets | String | false | Array = true Minsize = 0 Maxsize = 10 Maxlength = 500 | <p>App labeled on-screen presets.</p> <p>If omitted or the empty array is sent, the presets must be shown as not defined (with default names "PRESET_1", "PRESET_2", "PRESET_3" etc).</p> |
| appID | Integer | true | - | ID of the application requested this RPC. |

7.5.2.3 TextFieldStruct Structure

| Param Name | Type | Mandatory | Additional | Description |
|------------|-----------------------|-----------|-----------------|--|
| fieldName | Common.TextFileldName | true | - | The name of the field where the text must be displayed in. |
| fieldText | String | true | Maxlength = 500 | The text to be displayed. |

7.5.2.4 TextFieldName Enumeration

Only the text fields applicable to Alert RPC are described within this section. All the text fields names recognized by SDL are described in the section 13.1.14 *TextFieldName*.

| Element name | Short Description |
|--------------|---|
| mainField1 | The text that must be displayed in a single or upper display line. If this value is not set, the text of mainField1 must stay unchanged. If this text is empty "", the field must be cleared.. |
| mainField2 | The text that must be displayed on the second display line. If this text is not set, the text of mainField2 must stay unchanged. If this text is empty "", the field must be cleared.. |
| mainField3 | The text that must be displayed on the second "page" first display line. If this text is not set, the text of mainField3 must stay unchanged. If this text is empty "", the field must be cleared.. |
| mainField4 | The text that must be displayed on the second "page" second display line. If this text is not set, the text of mainField4 must stay unchanged. If this text is empty "", the field must be cleared.. |
| statusBar | The text is placed in the status bar area. Note: This relates to navigation displays If this parameter is omitted, the status bar text must remain unchanged. If this parameter is an empty string, the field must be cleared. If provided and the display has no status bar, this parameter must be ignored.. |
| mediaClock | Text value for MediaClock field. Shall arrive in the form as described in the MediaClockFormat enumeration If this text is set, any automatic media clock updates previously set with SetMediaClockTimer must be stopped.. |
| mediaTrack | The text that should be displayed in the track field. This field should be valid only for media applications on. If this text is not set, the text of mediaTrack must stay unchanged. If this text is empty "", the field must be cleared.. |

7.5.2.5 TextAlignment Enumeration

| Element name | Short Description |
|---------------|---|
| LEFT_ALIGNED | Text of mainField1 and mainField2 is left aligned. |
| RIGHT_ALIGNED | Text of mainField1 and mainField2 is right aligned. |
| CENTERED | Text of mainField1 and mainField2 is centered. |

7.5.3 Response

Note:

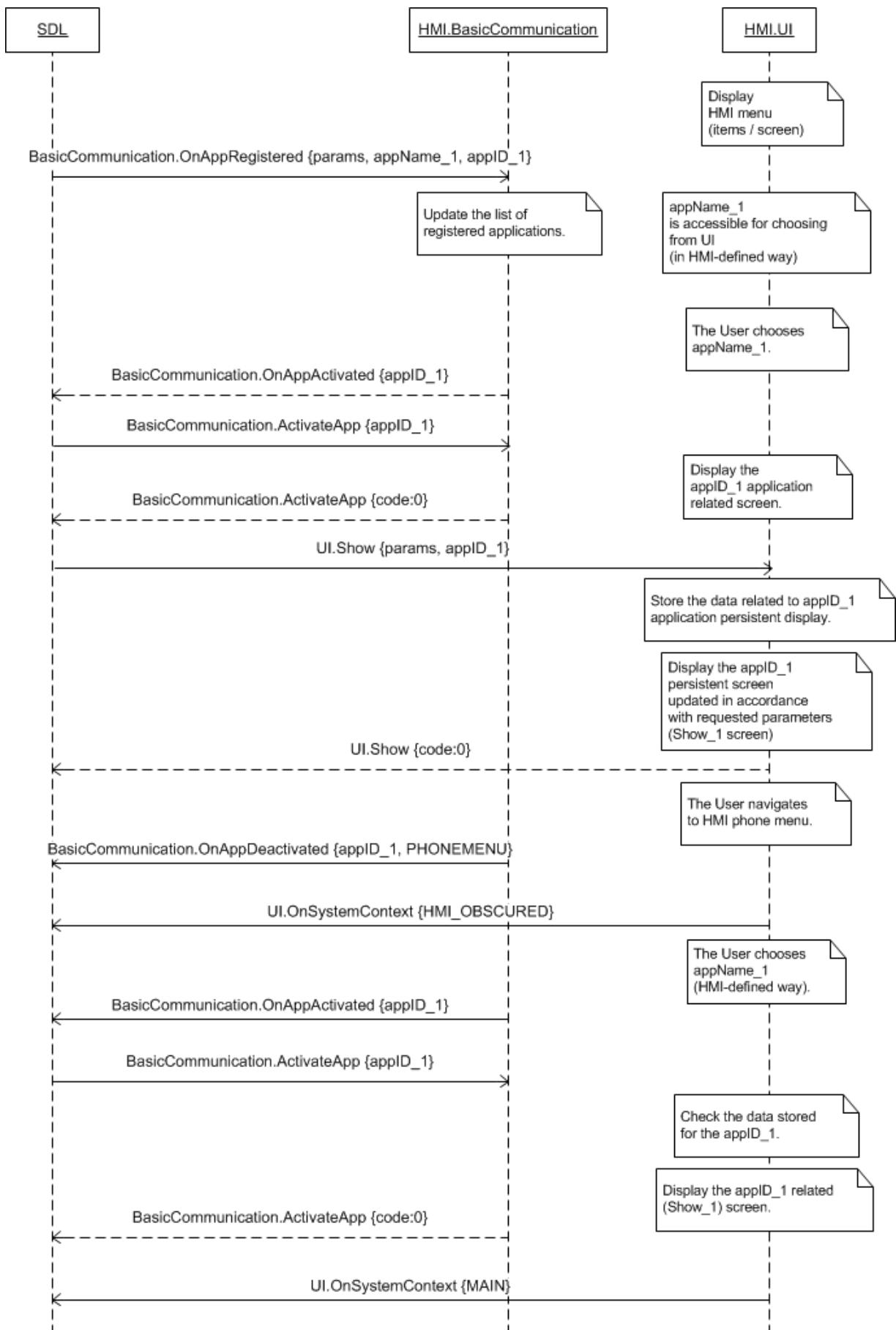
There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|--------|-------------|--------------|-------|----------------|-------|
| | | WebSocket | D-Bus | | |

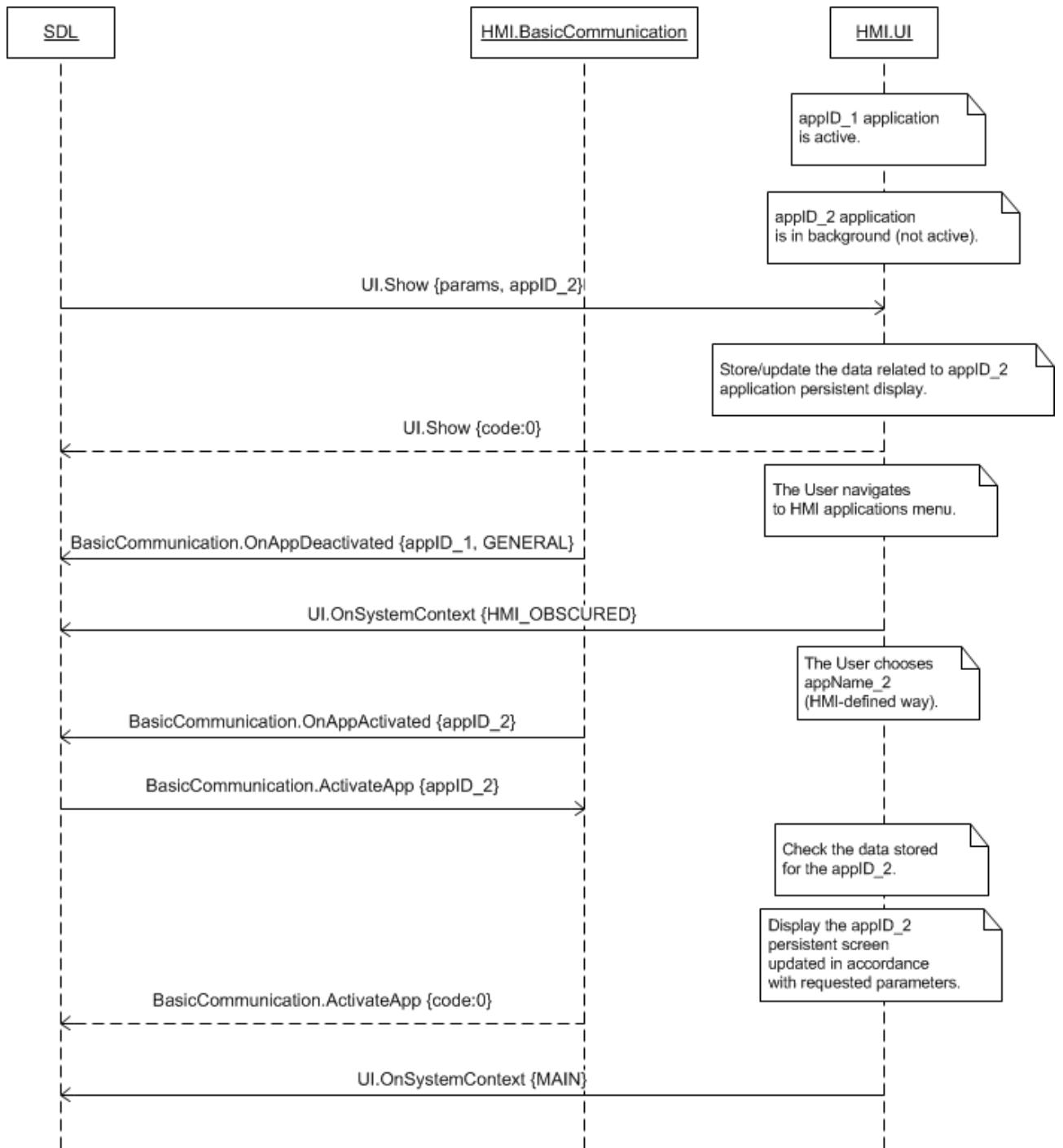
| | | | | | |
|---------|--|--------------------|---------------|----------|---|
| Success | HMI has stored the requested data correlating it with the application ID and updated the display areas if the requested application is currently active. | JSON response | Method return | code: 0 | |
| Failure | The data sent is invalid (invalid JSON syntax, parameters out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.5.4 Sequence Diagrams

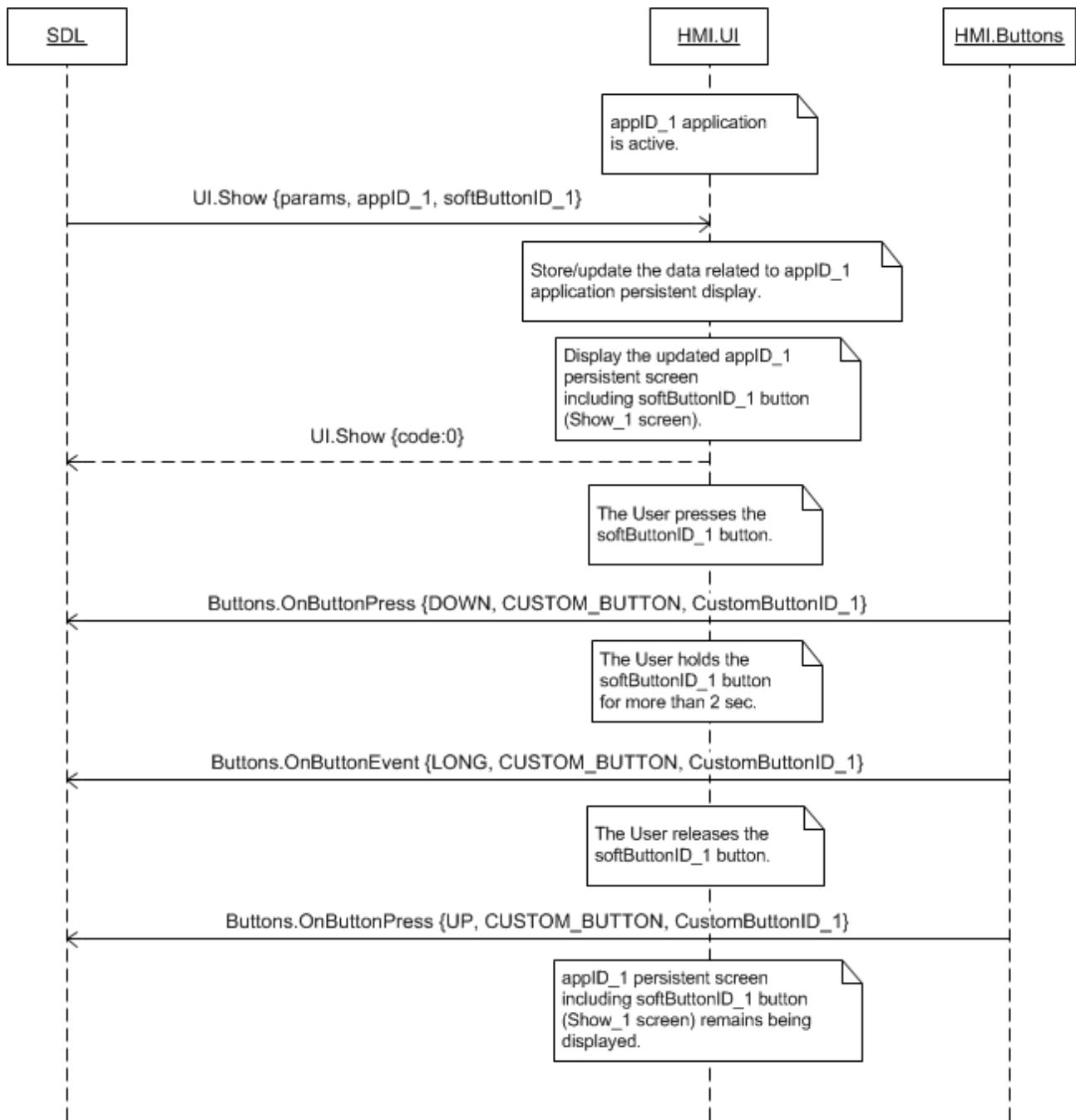
7.5.4.1 Show for the active application that is then deactivated and activated again



7.5.4.2 Show for the application currently not active on HMI



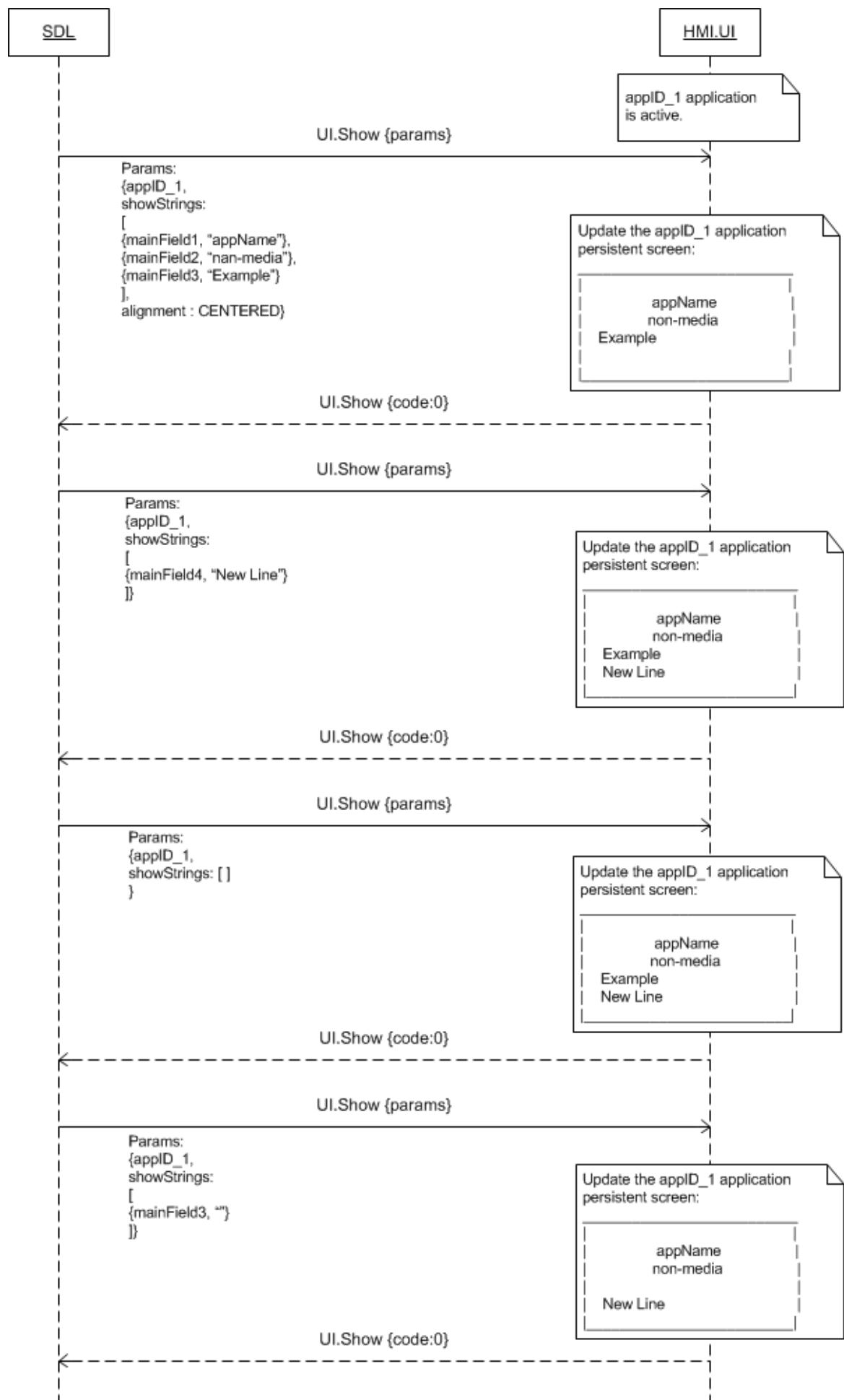
7.5.4.3 Requested with Show soft button press



Note:

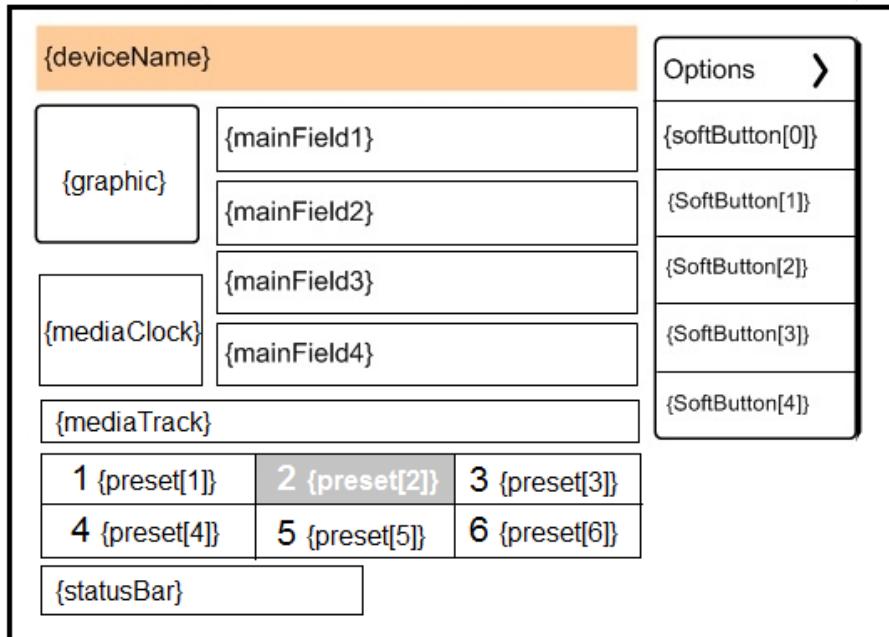
CustomButtonID parameter returned via OnButtonEvent/OnButtonPress notifications must have the same value as the softButtonID parameter sent via Show RPC.

7.5.4.4 Show text fields expected behavior

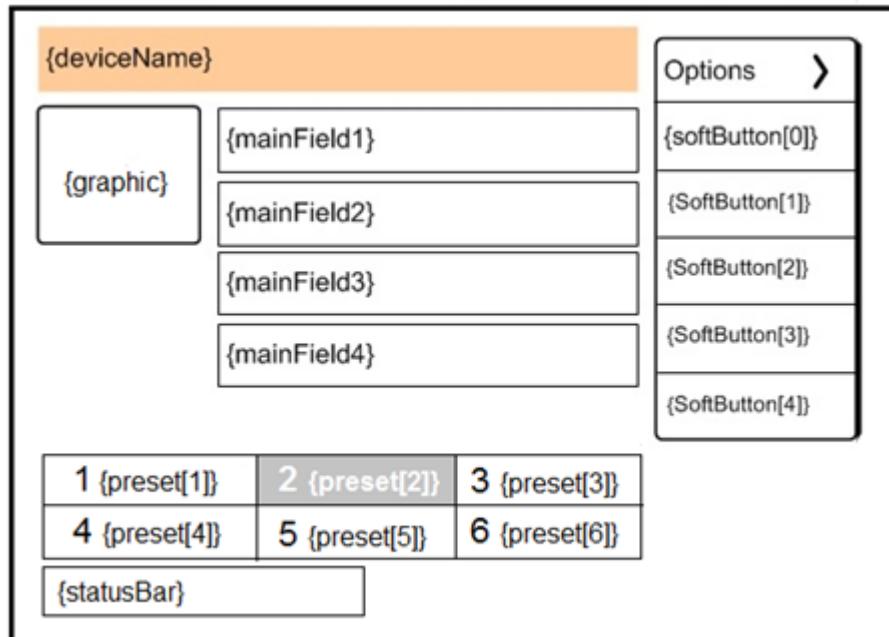


7.5.5 Possible Layout

7.5.5.1 Persistent display areas updated with Show for media applications



7.5.5.2 Persistent display areas updated with Show for non-media applications



7.5.6 JSON Messages Examples

7.5.6.1 Request

```
{  
    "id" : 120,  
    "jsonrpc" : "2.0",  
    "method" : "UI. Show",  
    "params" :  
    {  
        "showStrings" :
```

```

[

    {
        "fieldName" : mainField1,
        "fieldText" : "Favourite Album"
    },
    {
        "fieldName" : mediaClock,
        "fieldText" : "1:45:12"
    },
    {
        "fieldName" : mediaTrack,
        "fieldText" : "Ironic - The Collection - Alanis Morissette"
    }
],
"alignment" : LEFT_ALIGNED,
"graphic" :
{
    "value" : "tmp/SDL/app/Best_Media/AM-Collection-cover.png",
    "imageType" : DYNAMIC
},
"softButtons" :
[
    {
        "type" : BOTH,
        "text" : "Change Album",
        "image" :
        [
            {
                "value" : "tmp/SDL/app/Best_Media/change_alb_icon.jpg",
                "imageType" : DYNAMIC
            }
        ],
        "softButtonID" : 48,
        "systemAction" : DEFAULT_ACTION
    },
    {
        "type" : TEXT,
        "text" : "Change Artist",
        "softButtonID" : 57
    }
],
"customPresets" : ["Like Song", "Like Album"],
"appID" : 8726
}
]
}

```

7.3.5.2 Response

```
{
    "id" : 120,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "UI.Show"
    }
}
```

7.3.5.3 Error message

```
{
    "id" : 120,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 6,
        "message" : "Method not found"
    }
}
```

```

    "message" : "Ignored, because the intended result is already in effect",
    "data" :
    {
        "method" : "UI.Show"
    }
}

```

7.5.6 D-Bus Messages Examples

7.1.6.1 Request

7.1.6.2 Response

7.1.6.3 Failure

7.8 AddCommand

7.8.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Add a command to the in-application menu or sub menu. |

This RPC represents a request to HMI to add a command to in-application menu or sub-menu (the latter follows after AddSubMenu RPC).

The request may arrive in both cases of activated and deactivated application on HMI.

7.8.2 Request

7.8.2.1 Behavior

HMI must:

1. Provide the in-application menu (e.g. named 'Options') displayed right after the application is activated on HMI.
2. Provide the possibility for the User to enter this menu (empty by default) and to choose among its elements (once added via UI.AddCommand and/or AddSubMenu).
3. Check that the limit of menu / sub menu (depending on where the command is requested to be added to, see point 5. below) items is not exhausted by the command being requested (and if so, reject the request with the corresponding result code).
4. Store the data provided within this RPC correlating it with the `appID`.
5. Add the command with requested parameters (name, position) to:
 - The top level menu (e.g. 'Options') if `parentID` is not provided within RPC.
 - The sub menu with the ID corresponding to the `parentID` provided within RPC.
6. Provide the response correspondingly to the result of RPC execution.

Note:

The applicable to this RPC result codes are provided in section [7.8.3 Response](#).

7. Provide the UI.OnCommand notification with the cmdID corresponding to the command chosen by the User.

Important Note:

Once commands are added for the named application, they must remain accessible for the User (until UI.DeleteCommand comes) when this application is activated after having been deactivated.

7.8.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|-------------------|-----------|--------------|--|
| cmdID | Integer | true | Minvalue = 0 | <p>ID of the command to be added.</p> <p>This ID is unique within the application related to this request.</p> <p>This ID:</p> <ul style="list-style-type: none"> - Must be returned by HMI later within OnCommand notification to identify the command selected by the user. - Might be provided by SDL later within DeleteCommand request. |
| menuParams | Common.MenuParams | false | - | <p>Defines the name of the command, its position, ID of the submenu for the command to be added to.</p> <p>If not provided, the command must be added to the end of the list of items of the top-level menu.</p> <p>See MenuParams.</p> |
| cmdIcon | Common.Image | false | - | <p>Image to be displayed for representing the command: either the path to dynamic image stored on HU or the static binary image itself.</p> <p>If not provided, no image or the default one if applicable should be displayed.</p> <p>See Image.</p> |
| appID | Integer | true | - | ID of the application related to this RPC. |

7.8.2.3 MenuParams

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|---------------------------------------|---|
| parentID | Integer | false | minvalue = 0 maxvalue = 2000000000 | <ul style="list-style-type: none"> - Unique ID of the sub menu, the command must be added to. - If not provided, the command must be added to the top-level application menu. |

| | | | | |
|----------|---------|-------|---------------------------------|--|
| position | Integer | false | minvalue = 0 maxvalue = 1000 | This value is the position within the elements of top-level application menu / sub menu where the command must be added to: - If 0, the item must be inserted to the first position. - If 1, the item must be inserted to the second position. - Etc. If the next command comes with the same position value, it must be added as the next item within corresponding 'position block'. (e.g. the command with position 0 must be added - after the previous item with position 0 - and before the first item with position 1). If the value is greater than or equal to the number of elements of the top-level application menu, the command must be appended to the end of the list. If omitted the item must be added to the end of the list. |
| menuName | String | true | maxlength = 500 | The text that must be shown as a name of a command. |

7.8.3 Response

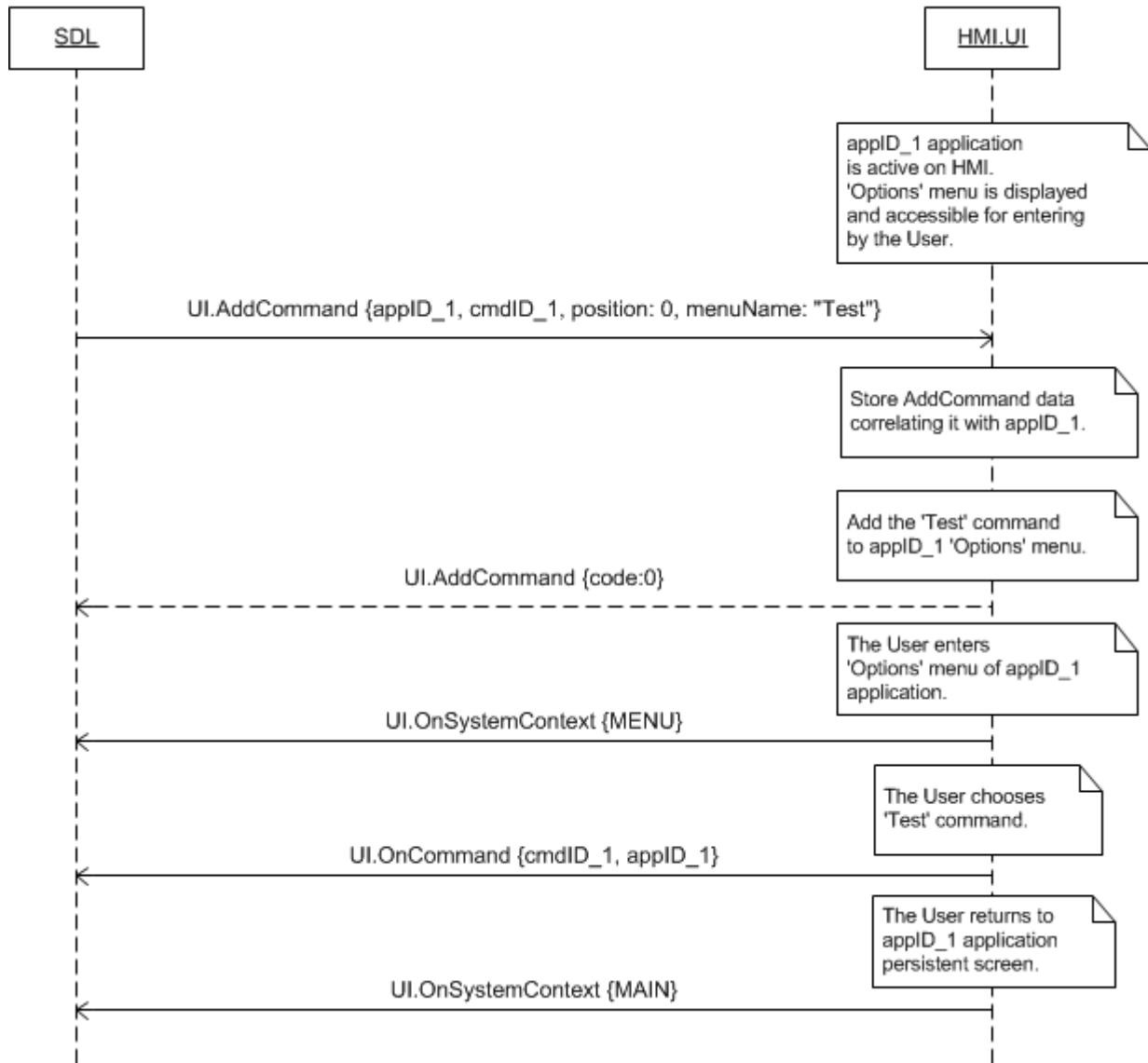
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

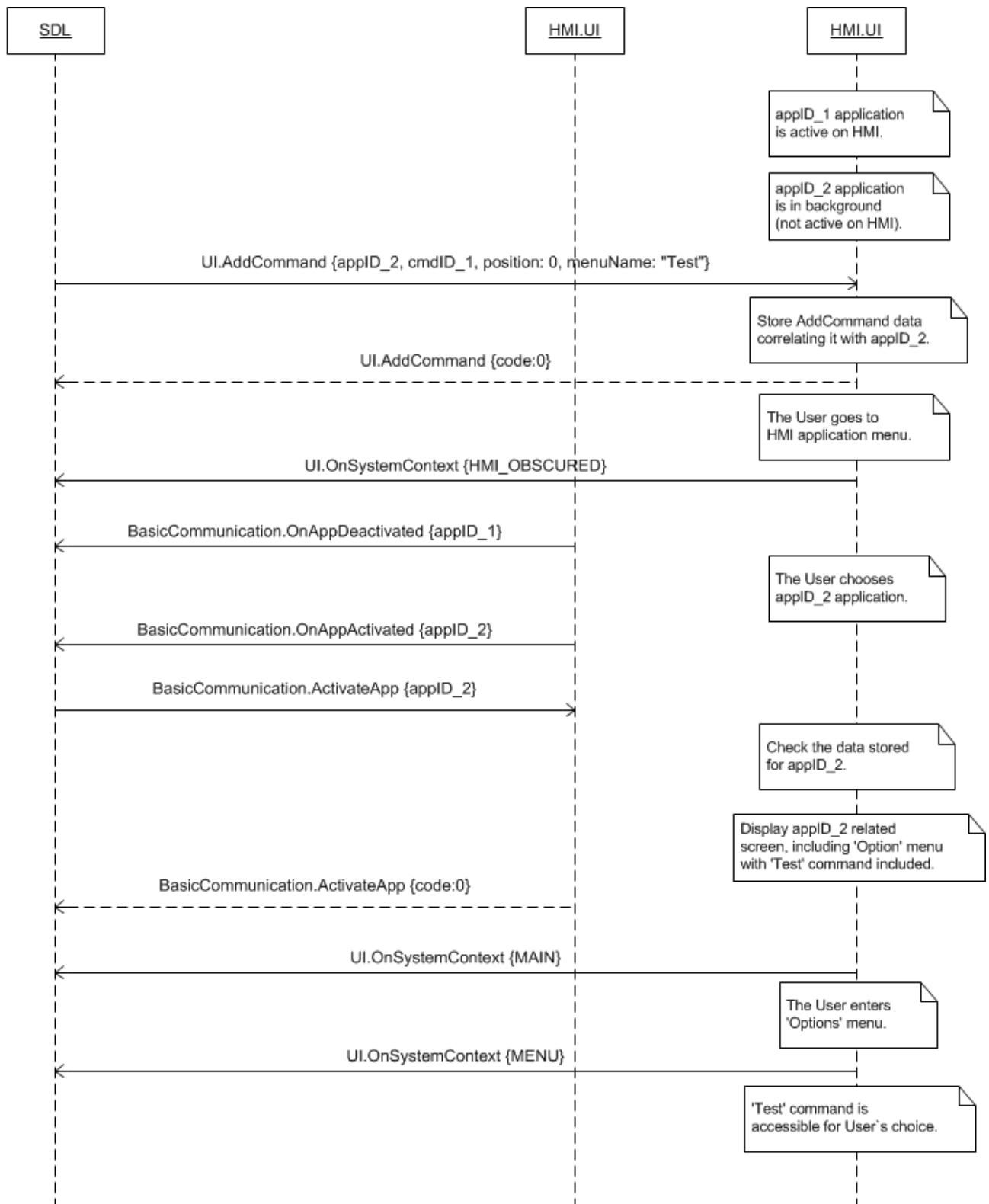
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | HMI has stored the data correlating it with <code>appID</code> and added the command with the corresponding parameters to the in-application menu/sub menu. | JSON response | Method return | code: 0 | |
| Failure | 1) The limit of position items of the top level menu is exhausted. 2) The limit of position items of the corresponding sub menu the command to be added to is exhausted. | JSON error message | Method return | Code: 4 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-recognized codes. |
| | The data sent is invalid (invalid JSON syntax, parameters out of bounds or of wrong type) | | | code: 11 | |
| | 1) The command with requested <code>cmdID</code> is already added for the named application. 2) The sub menu with requested <code>parentID</code> does not exist. | | | code: 13 | |
| | The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.8.4 Sequence Diagrams

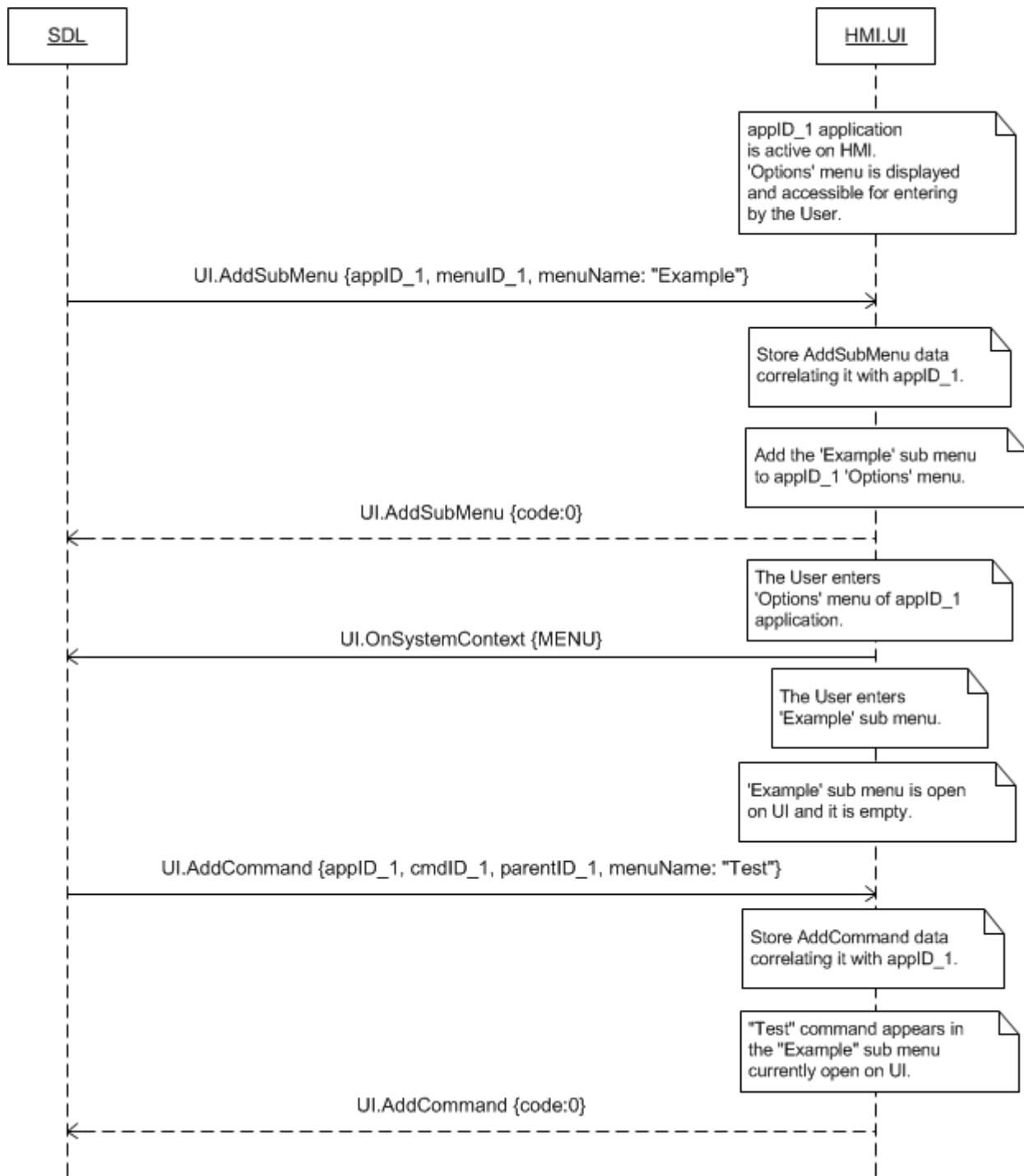
7.8.4.1 AddCommand for the active application on HMI, the command is chosen by the User.



7.8.4.2 AddCommand for the application not active on HMI



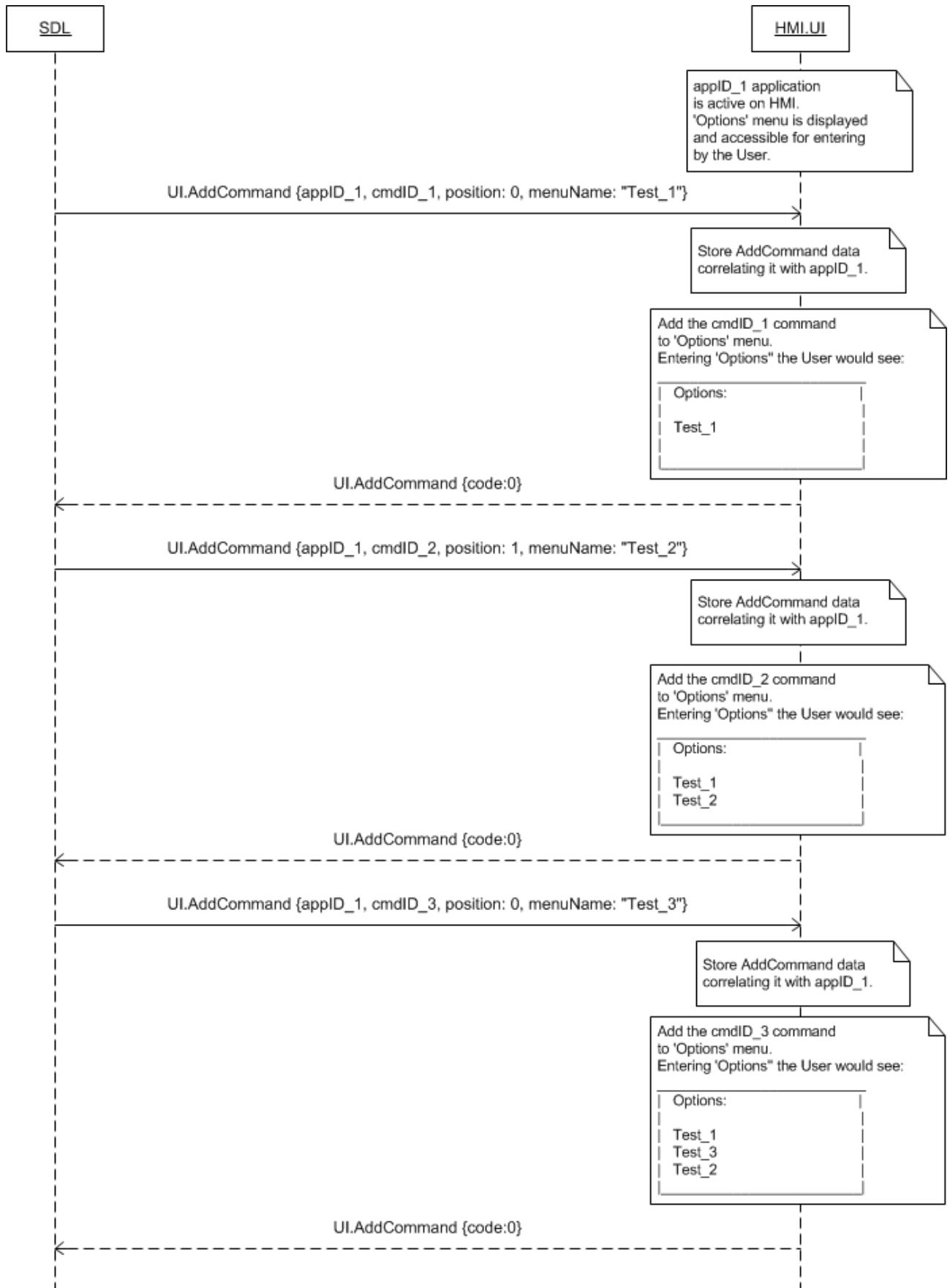
7.8.4.3 AddCommand: adding command to sub menu



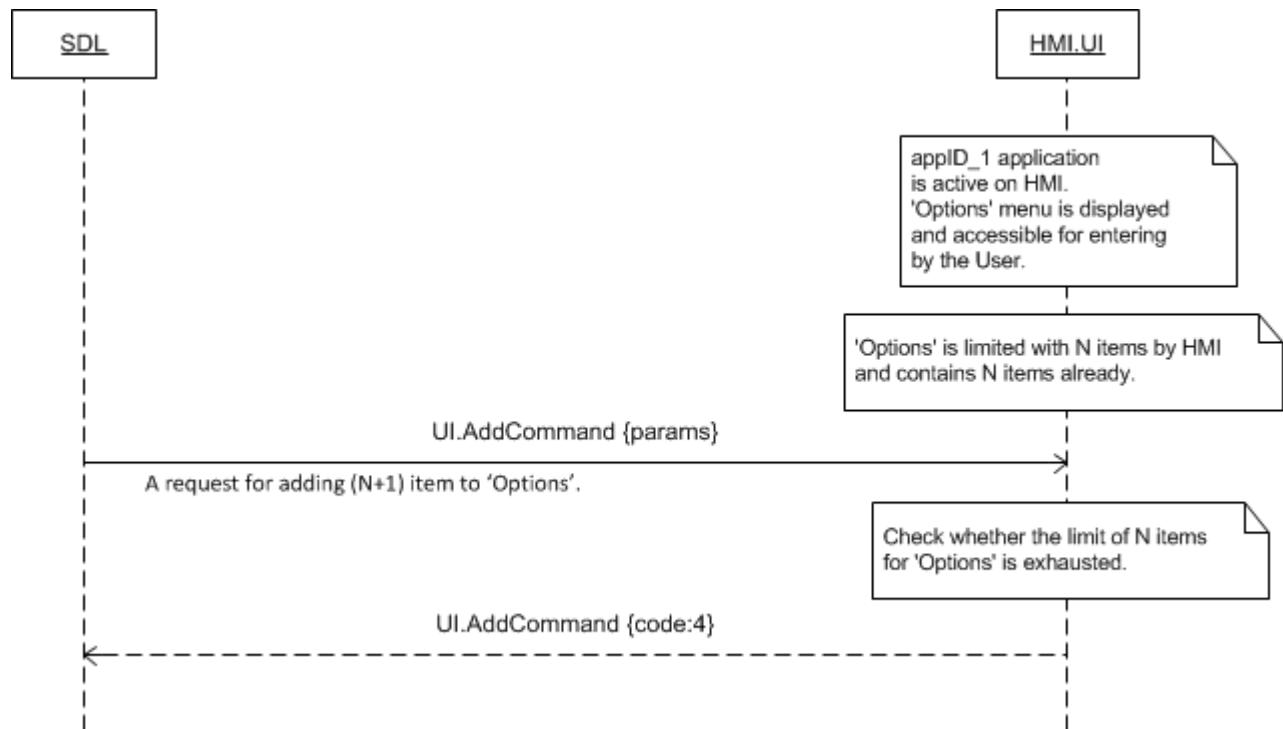
Note:

The value of `parentID` sent via `UI.AddCommand` is equal to the value of `menuID` previously provided via `AddSubMenu`.

7.8.4.4 AddCommand: expected behavior of adding commands depending on position parameter



7.8.4.5 AddCommand rejected because of the limit of menu items exhausted

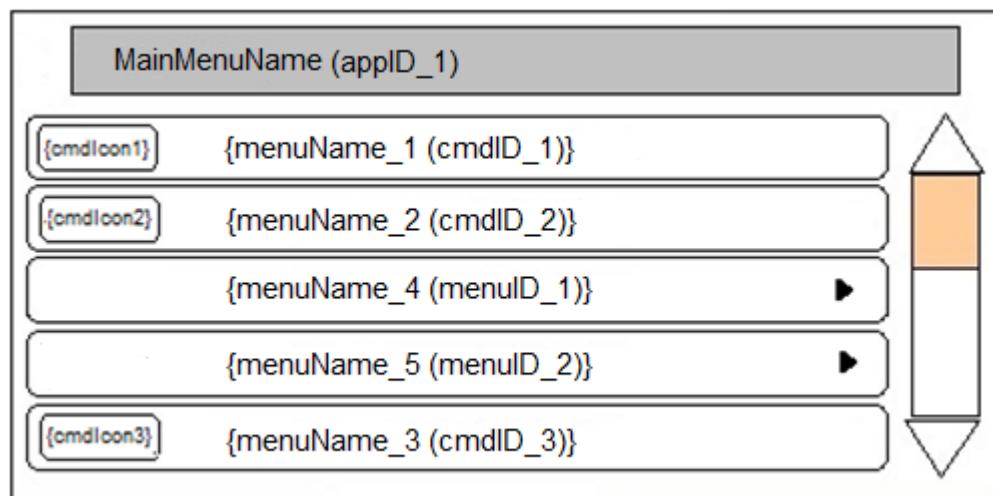


Note:

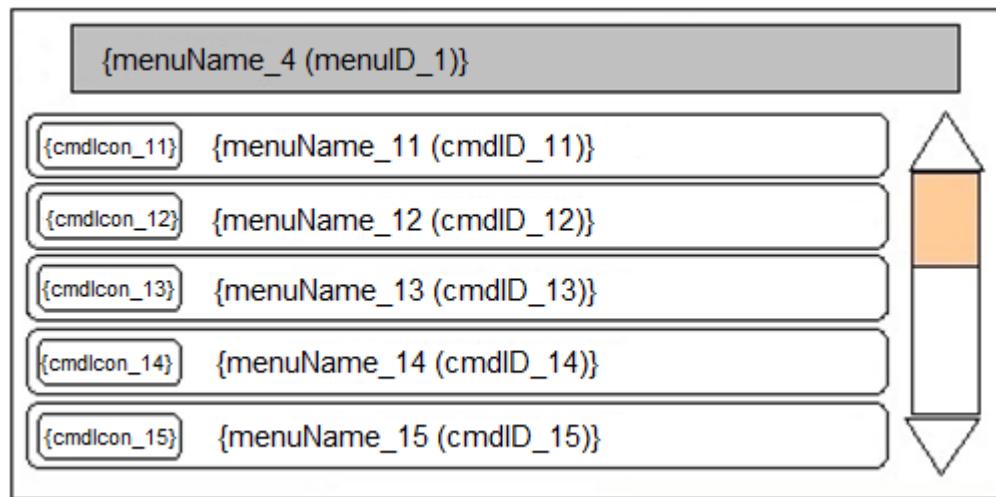
"Top level menu (e.g. 'Options') items" are considered to be both commands and sub menus.

7.8.5 Possible Layout

7.8.5.1 Application main menu with sub menus and commands



7.8.5.2 Application sub menu with commands



7.8.6 JSON Messages Examples

7.8.6.1 Request

```
{
  "id" : 215,
  "jsonrpc" : "2.0",
  "method" : "UI.AddCommand",
  "params" :
  {
    "cmdID" : 2318,
    "menuParams" :
    {
      "parentID" : 6,
      "position" : 0,
      "menuName" : "Show weather for tomorrow"
    },
    "cmdIcon" :
    {
      "value" : "tmp/SDL/app/Gis_meteo/1245_28.jpeg",
      "imageType" : DYNAMIC
    },
    "appID" : 65409
  }
}
```

7.8.6.2 Response

```
{
  "id" : 215,
  "jsonrpc" : "2.0",
  "result" :
  {
    "code" : 0,
    "method" : "UI.AddCommand"
  }
}
```

7.8.6.3 Error message

```
{
  "id" : 215,
  "jsonrpc" : "2.0",
  "error" :
  {
```

```

    "code" : 19,
    "message" : "The request cannot be executed because no application data
                 has been received",
    "data" :
    {
        "method" : "UI.AddCommand"
    }
}

```

7.8.7 D-Bus Messages Examples

7.8.7.1 Request

[REDACTED]

7.8.7.2 Response

[REDACTED]

7.8.7.3 Failure

[REDACTED]

7.9 DeleteCommand

7.9.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Delete the command from in-application menu |

SDL requests to delete the command from in-application menu or sub menu of the named application previously added via UI.AddCommand.

The request may arrive in both cases of activated and deactivated application on HMI.

7.9.2 Request

7.9.2.1 Behavior

HMI must:

1. Update the named application stored data correspondingly to the RPC arrived.
2. Delete the command identified with cmdID.
3. Display updates:
 - Right away if the named application is active and the corresponding menu/sub menu is open on UI.
 - After the corresponding menu/sub menu is opened on UI upon User's request:
 - If the RPC arrived when the named application was active and had another menu or persistent display visible on UI
 - If the RPC arrived when the named application was not active on HMI.
5. Provide the response corresponding to the result of RPC execution.

Note:

The applicable to this RPC result codes are provided in section 7.9.3 Response.

Note:

- The value of cmdID is previously sent via UI.AddCommand.

- The value of `appID` is previously sent via `UpdateAppList` or `OnAppRegistered`.

7.9.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|--------------------|---------|-----------|--|
| <code>cmdID</code> | Integer | true | ID of the command to be deleted (the one sent within the <code>AddCommand</code> request). |
| <code>appID</code> | Integer | true | ID of the application that concerns this RPC. |

7.9.3 Response

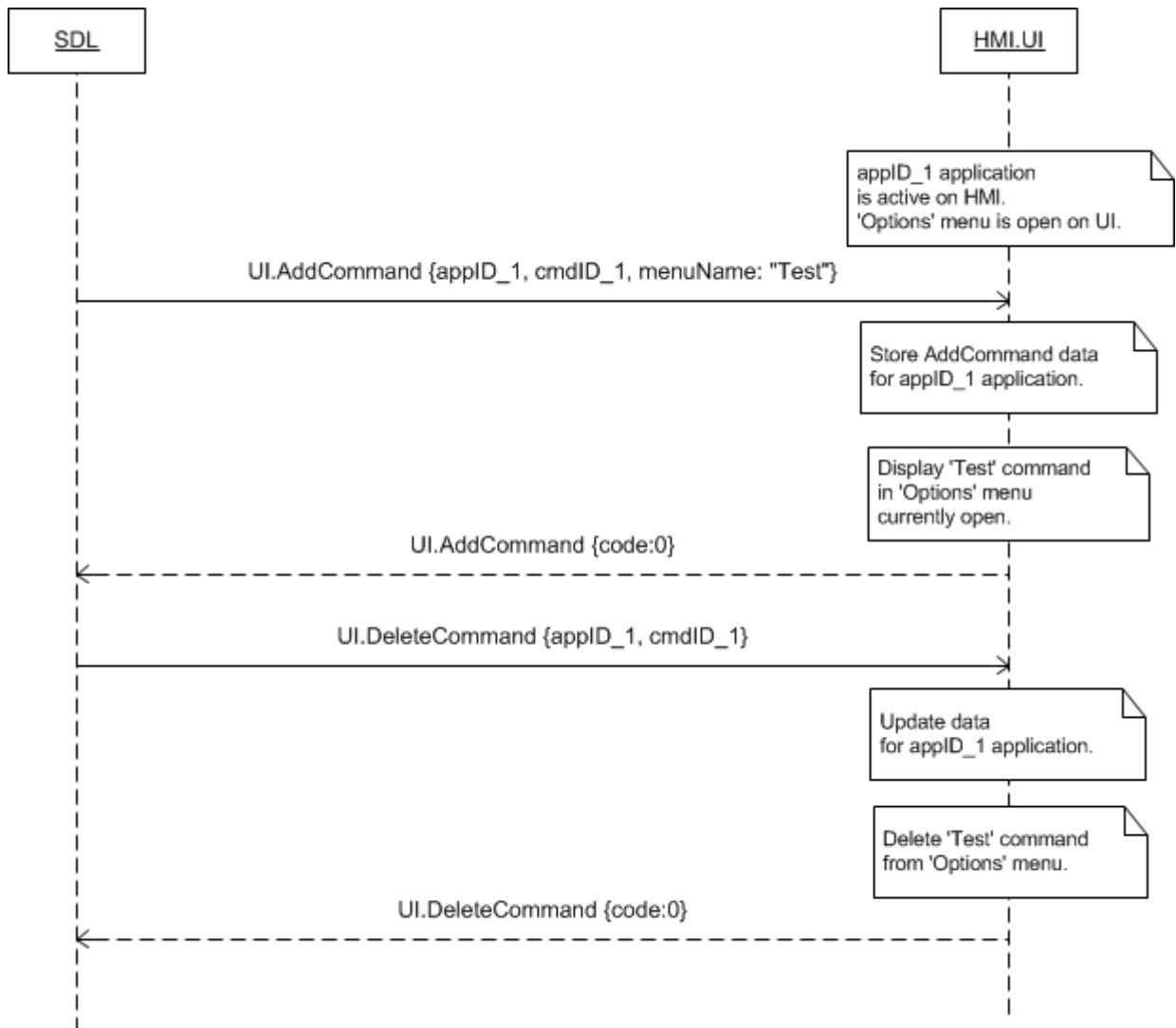
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

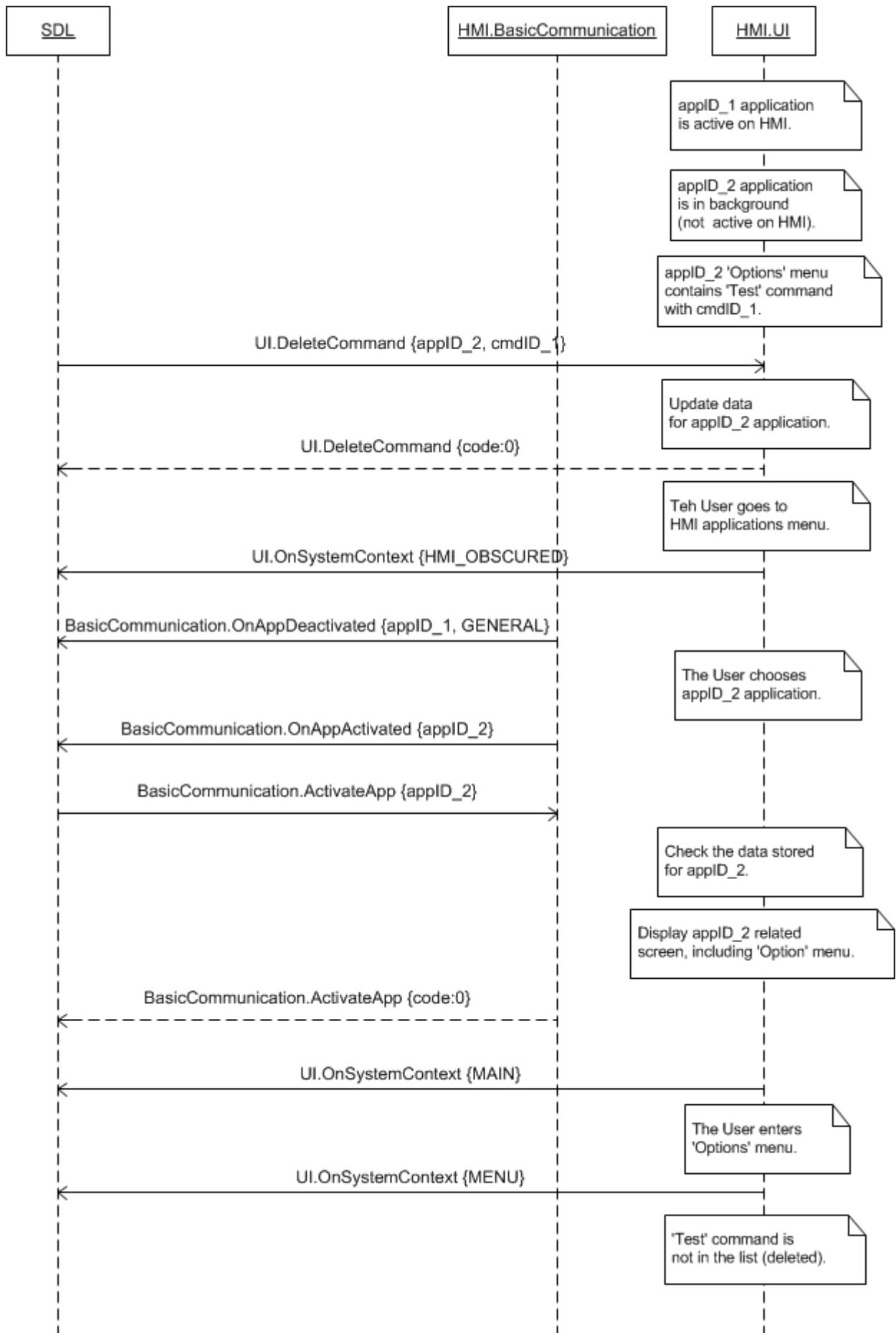
| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|---|
| | | WebSocket | D-Bus | | |
| Success | HMI has updated the named application related data and deleted the requested command. | JSON response | Method return | code: 0 | . |
| Failure | The data sent is invalid (invalid JSON syntax, parameters are out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | The command with requested <code>cmdID</code> does not exist on HMI for the named application. | | | Code: 13 | Please see Result Enumeration for all SDL-recognized codes. |
| | The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.9.4 Sequence Diagrams

7.9.4.1 DeleteCommand, for the application active on HMI, for the command from the menu currently open on UI and preceding AddCommand



7.9.4.2 DeleteCommand for the application not active on HMI



7.9.5 JSON Messages Examples

7.9.5.1 Request

```
{  
    "id" : 70,  
    "jsonrpc" : "2.0",  
    "method" : "UI. DeleteCommand",  
    "params" :  
    {  
        "cmdID" : 2318,  
        "appID" : 65409  
    }  
}
```

7.9.5.2 Response

```
{  
    "id" : 70,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "UI. DeleteCommand"  
    }  
}
```

7.9.5.3 Error message

```
{  
    "id" : 70,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 13,  
        "message" : "One of the provided IDs is not valid",  
        "data" :  
        {  
            "method" : "UI. DeleteCommand"  
        }  
    }  
}
```

7.9.6 D-Bus Messages Examples

7.9.6.1 Request

```
[REDACTED]
```

7.9.6.2 Response

```
[REDACTED]
```

7.9.6.3 Failure

```
[REDACTED]
```

7.10 AddSubMenu

7.10.1 Description

| | |
|---------|----------|
| Type: | Function |
| Sender: | SDL |

| | |
|-----------------|---------------------------------------|
| Purpose: | Add a sub menu to in-application menu |
|-----------------|---------------------------------------|

This RPC represents a request to HMI to add a command to in-application main menu. The sub menu is never requested to be added to sub menu.

Further SDL may request to add commands via AddCommands RPC to this sub menu identified with menuID.

The request may arrive in both cases of activated and deactivated application on HMI.

7.10.2 Request

7.10.2.1 Behavior

HMI must:

1. Provide the in-application main menu (e.g. named 'Options') displayed right after the application is activated on HMI.
2. Provide the possibility for the User to enter this menu (empty by default) and to choose among its elements (once added via UI.AddCommand and/or AddSubMenu).
3. Check that the limit of the main menu items is not exhausted by the sub menu being requested (and if so, reject the request with the corresponding result code).
4. Store the data provided within this RPC correlating it with the appID.
5. Add the sub menu with requested parameters (name, position) to the top level in-application menu.
6. Provide the response correspondingly to the result of RPC execution.

Note:

The applicable to this RPC result codes are provided in section 7.10.3 Response.

7. Provide the possibility for the User to enter this sub menu (empty by default) and choose among its elements (once added by SDL via UI.AddCommand).

Important Note:

Once the sub menus are added for the named application, they must remain accessible for the User (until UI.DeleteSubMenu comes) when this application is activated after having been deactivated.

7.10.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|-------------------|-----------|---------------------------------------|--|
| menuID | Integer | true | Minvalue = 1 Maxvalue = 2000000000 | ID of the sub menu to be added. It is unique within the application concerned. |
| menuParams | Common.MenuParams | true | - | The name of the sub-menu to be added and its positions within menu items. See MenuParams. |
| appID | Integer | true | - | ID of the application requested this RPC. |

7.10.2.3 MenuParams

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| | | | | |
|----------|---------|-------|---------------------------------|---|
| position | Integer | false | minvalue = 0 maxvalue = 1000 | <p>This value is the position within the elements of top-level application menu where the sub-menu must be added to:</p> <ul style="list-style-type: none"> - If 0, the item must be inserted to the first position. - If 1, the item must be inserted to the second position. - Etc. <p>If the next sub menu comes with the same position, it must be added as the next item within corresponding 'position block'. (e.g. the sub menu with position 0 must be added</p> <ul style="list-style-type: none"> - after the previous item with position 0 - and before the first item with position 1). <p>If the value is greater than or equal to the number of elements of the top-level application menu, the sub menu must be appended to the end of the list.</p> <p>If omitted the entry must be added to the end of the list.</p> |
| menuName | String | true | maxlength = 500 | The text that must be shown as a name of a sub - menu. |

7.10.3 Response

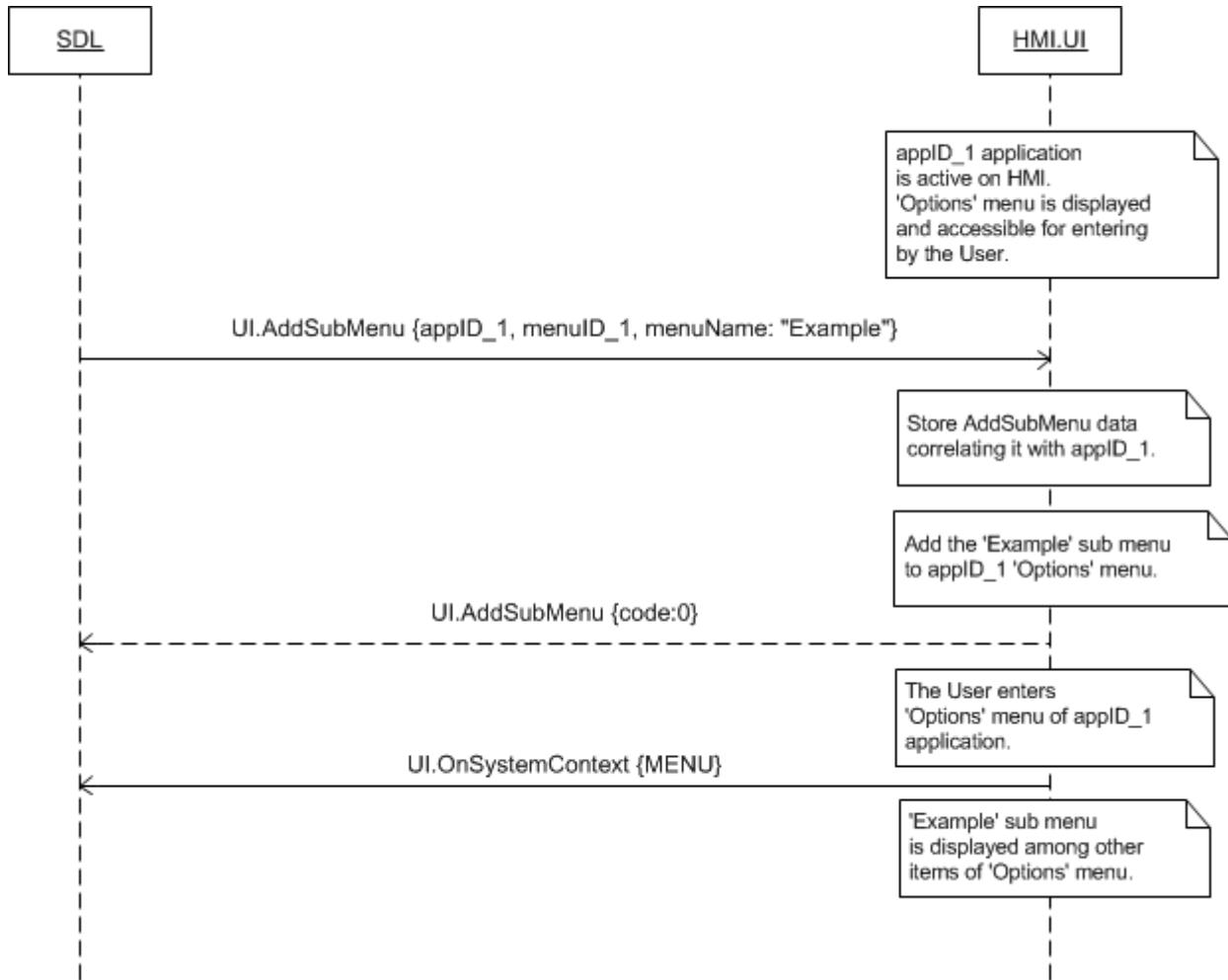
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

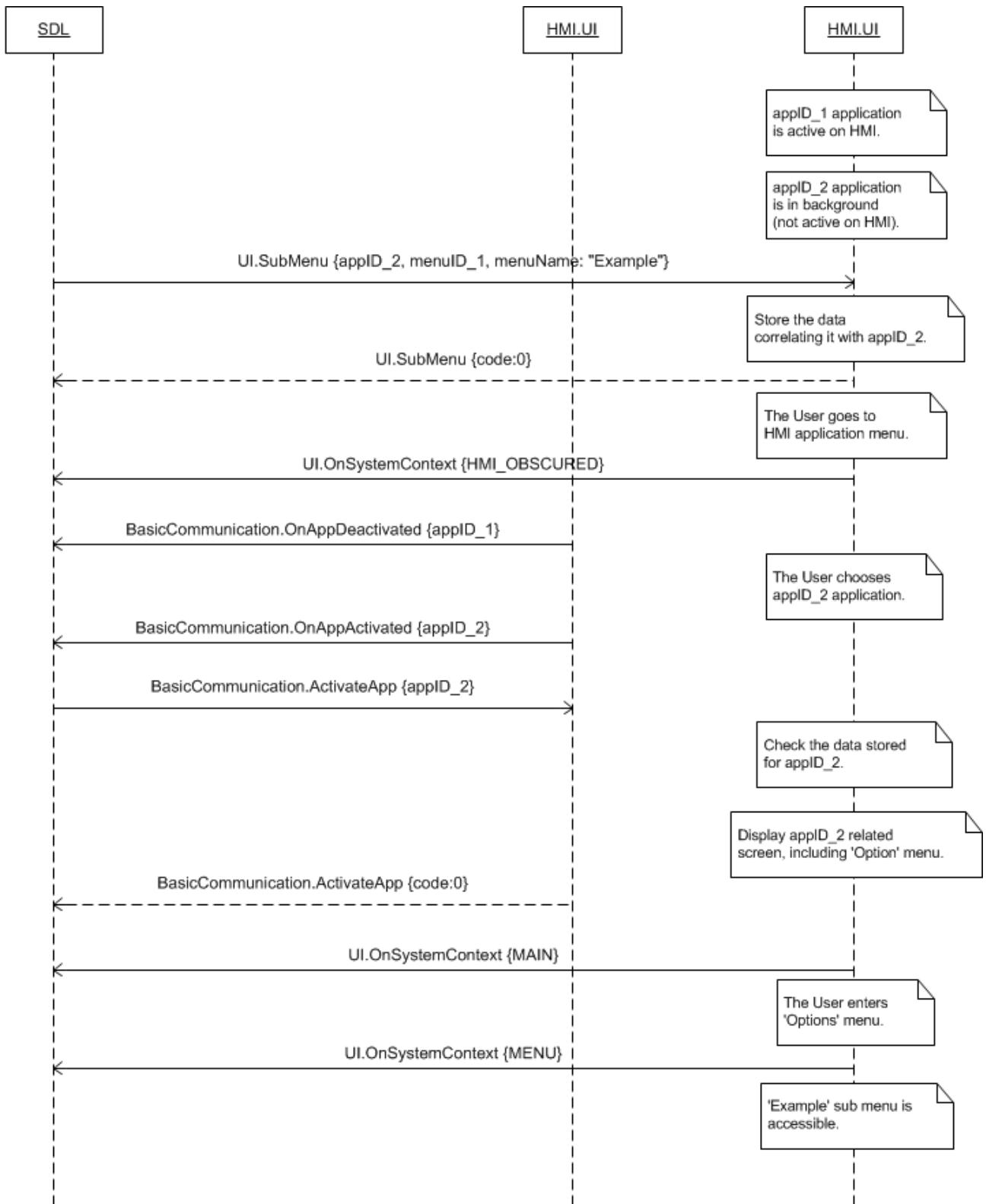
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | HMI has stored the data correlating it with appID and added the sub menu with the corresponding parameters to the in-application main menu. | JSON response | Method return | code: 0 | |
| Failure | The limit of position items of the top level menu is exhausted. | JSON error message | Method return | Code: 4 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-recognized codes. |
| | The data sent is invalid (invalid JSON syntax, parameters out of bounds or of wrong type) | | | code: 11 | |
| | The sub menu with requested menuID is already added for the named application. | | | code: 13 | |
| | The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.10.4 Sequence Diagrams

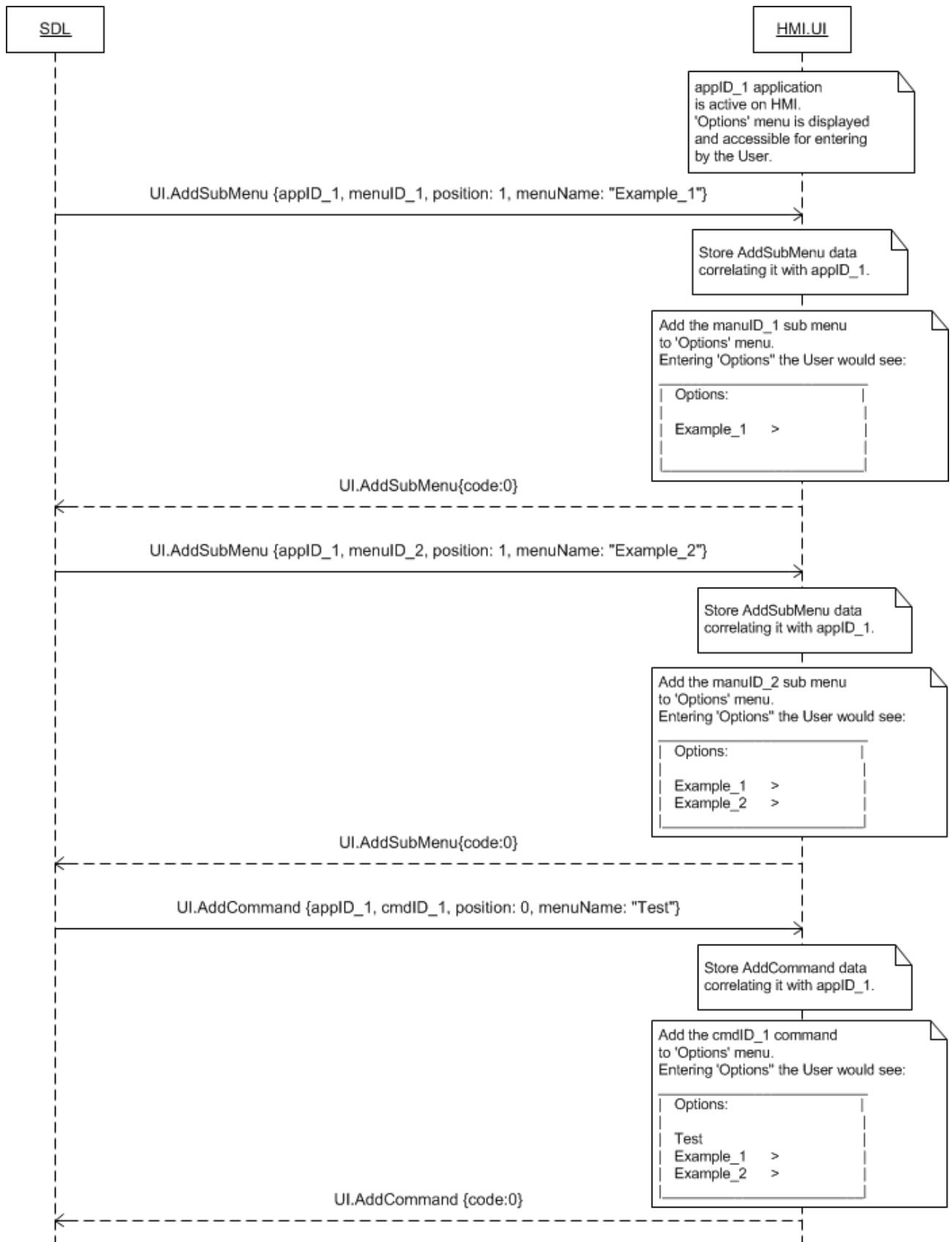
7.10.4.1 AddSubMenu for the application active on HMI



7.10.4.2 *AddSubMenu for the application not active on HMI*



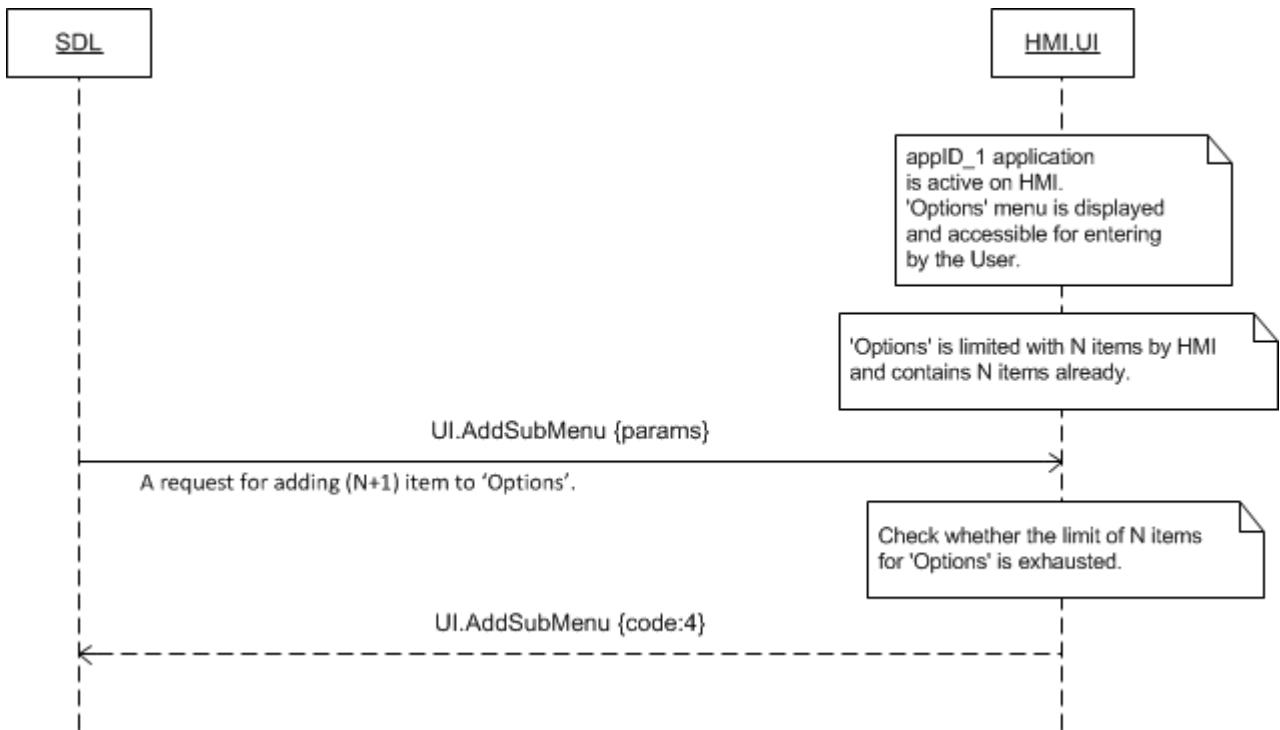
7.10.4.3 AddSubMenu: expected behavior of adding sub menus depending on position parameter



Note:

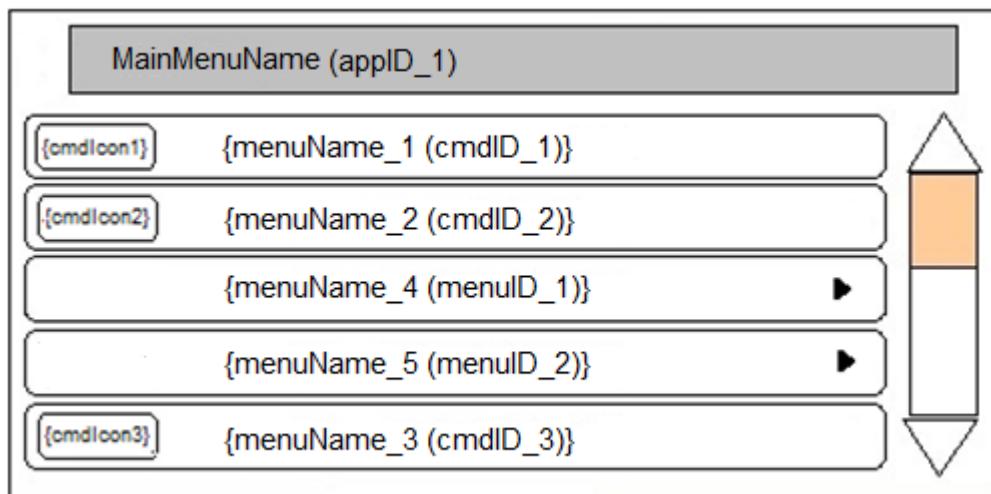
HMI should sort the in-application main menu items by position parameter independently of item type (submenu/command).

7.10.4.4 AddSubMenu rejected because of the limit of menu items exhausted



7.10.5 Possible Layout

7.10.5.1 Application main menu with sub menus and commands



7.10.6 JSON Messages Examples

7.10.6.1 Request

```
{
    "id" : 112,
    "jsonrpc" : "2.0",
    "method" : "UI.AddSubMenu",
    "params" :
    {
        "menuID" : 345,
        "menuParams" :
        {
            "position" : 2,
            "menuName" : "Settings"
        },
        "appID" : 65464
    }
}
```

7.10.6.2 Response

```
{  
    "id" : 112,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "UI.AddSubMenu"  
    }  
}
```

7.10.6.3 Error message

```
{  
    "id" : 112,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 14,  
        "message" : "Duplicate name: there was a conflict with an already  
registered name",  
        "data" :  
        {  
            "method" : "UI.AddSubMenu"  
        }  
    }  
}
```

7.10.7 D-Bus Messages Examples

7.10.7.1 Request

```
[REDACTED]
```

7.10.7.2 Response

```
[REDACTED]
```

7.10.7.3 Failure

```
[REDACTED]
```

7.11 DeleteSubMenu

7.11.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Delete the sub menu from in-application menu. |

SDL requests to delete the command from in-application menu of the named application previously added via UI.AddSubMenu.

The request may arrive in both cases of activated and deactivated application on HMI.

7.11.2 Request

7.11.2.1 Behavior

HMI must:

1. Check whether the named sub menu is currently open on UI and:
 - If so, NOT delete this sub menu responding with appropriate result code (see section 7.11.3).
 - If not, continue processing the request.
2. Update the named application stored data correspondingly.
3. Delete
 - The sub menu identified with `menuID`.
 - The commands that are related to this sub menu (which have `parentID` of the named `menuID` value)
4. Display updates:
 - Right away if the named application is active and the main menu is open on UI.
 - After the main menu is opened on UI upon User's request:
 - If the RPC arrived when the named application was active and had another menu or persistent display visible on UI
 - If the RPC arrived when the named application was not active on HMI.
5. Provide the response corresponding to the result of RPC execution.

Note:

The applicable to this RPC result codes are provided in section 7.11.3 Response.

Note:

- The value of `menuID` is previously sent via `AddSubMenu`.
- The value of `appID` is previously sent via `UpdateAppList` or `OnAppRegistered`.

7.11.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|---------------------|---------|-----------|---|---|
| <code>menuID</code> | Integer | true | Minvalue = 1 Maxvalue = 2000000000 | ID that identifies the submenu to be deleted (the one that was previously provided via <code>AddSubMenu</code>). |
| <code>appID</code> | Integer | true | - | ID of the application that requested this RPC. |

7.11.3 Response

Note:

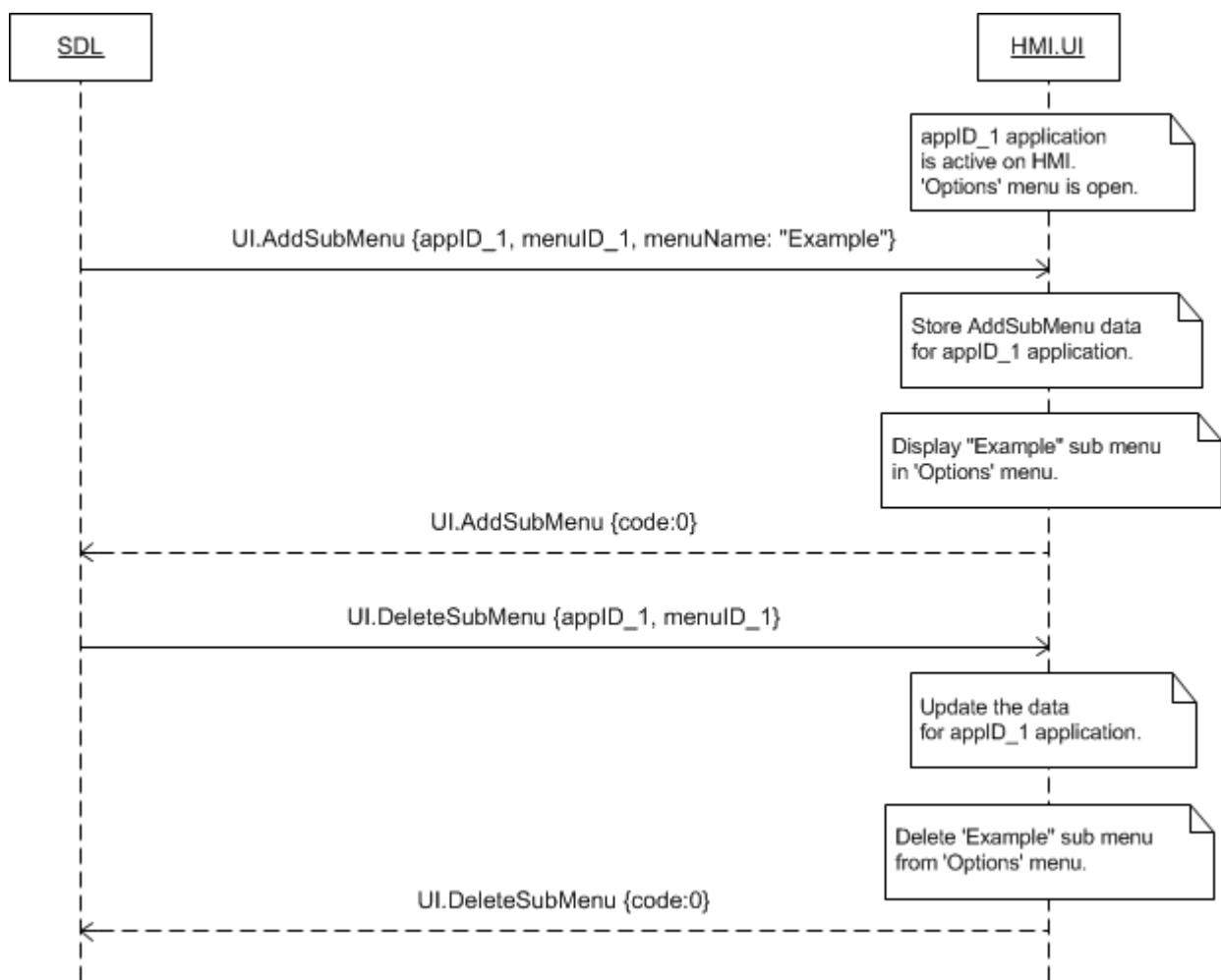
There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|---|
| | | WebSocket | D-Bus | | |
| Success | HMI has updated the named application related data and deleted the requested sub menu. | JSON response | Method return | code: 0 | . |
| Failure | The named sub menu is currently open on UI. | JSON error message | Method return | Code: 8 | Applicable for this RPC result codes. Please see <code>Result</code> Enumeration for all SDL-recognized codes. |
| | The data sent is invalid (invalid JSON syntax, parameters are out of bounds or of wrong type) | | | code: 11 | |
| | The sub menu with requested <code>menuID</code> does not exist on HMI for the named application. | | | Code: 13 | |

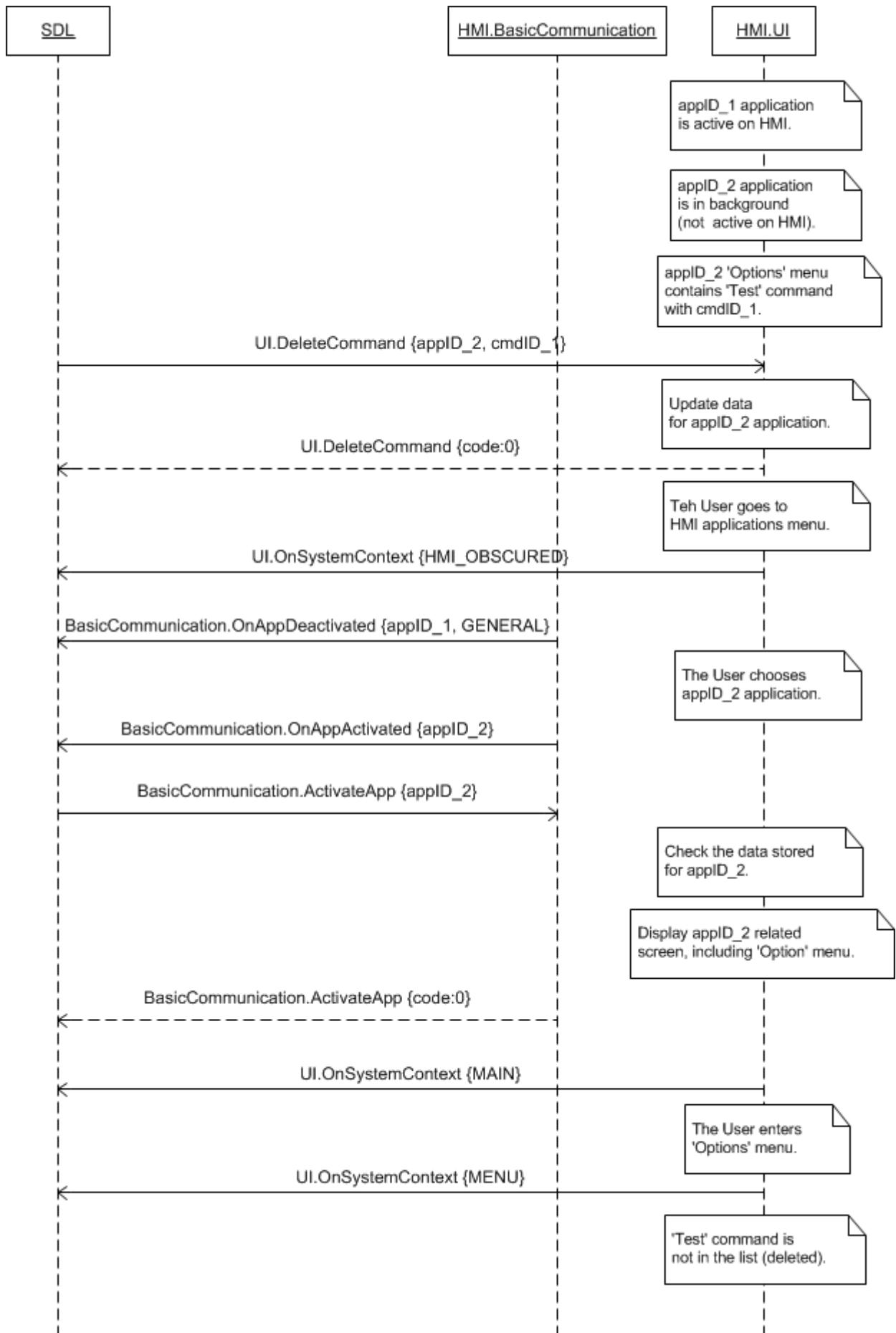
| | | | | | |
|--|---|--|--|----------|--|
| | The unknown issue occurred or other codes are not applicable. | | | code: 22 | |
|--|---|--|--|----------|--|

7.11.4 Sequence Diagrams

7.11.4.1 DeleteSubMenu: the application is active on HMI, the msin menu is currently open on UI, AddSubMenu precedes



7.11.4.2 DeleteCommand for the application not active on HMI



7.11.5 JSON Messages Examples

7.11.5.1 Request

```
{  
    "id" : 70,  
    "jsonrpc" : "2.0",  
    "method" : "UI. DeleteSubMenu",  
    "params" :  
    {  
        "menuID" : 345,  
        "appID" : 65464  
    }  
}
```

7.11.5.2 Response

```
{  
    "id" : 70,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "UI. DeleteCommand"  
    }  
}
```

7.11.5.3 Error message

```
{  
    "id" : 70,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 8,  
        "message" : "The data may not be changed because it is currently in use",  
        "data" :  
        {  
            "method" : "UI. DeleteCommand"  
        }  
    }  
}
```

7.11.6 D-Bus Messages Examples

7.11.6.1 Request

```
[REDACTED]
```

7.11.6.2 Response

```
[REDACTED]
```

7.11.6.3 Failure

```
[REDACTED]
```

7.12 UI.PerformInteraction

7.12.1 Description

| | |
|-------|----------|
| Type: | Function |
|-------|----------|

| | |
|-----------------|---|
| Sender: | SDL |
| Purpose: | Perform a UI interaction with the User. |

SDL uses PerformInteraction for requesting an interaction with the User, for example:

- To confirm a command
- To make a choice between several suggested ones.

The request concerns the application currently active on HMI.

Notes about PerformInteraction

1. *SDL may intend to perform one of three modes of interaction:*
 - 1.1. *'Manual only' – when the User is expected to make a choice manually only (display, buttons, soft buttons are involved).*
 - 1.2. *'VR only' – when the User is expected to make a choice by voice only (voice recognition module is involved).*
 - 1.3. *'Both' – when the User is expected to make a choice whichever by voice or manually (display, buttons, soft buttons, VR module are involved).*
2. *SDL sends both VR.PerformInteraction and UI.PerformInteraction for each of above types of interaction. HMI's expected behavior differs depending on interaction mode (see section 7.12.2.1 Behavior).*
3. *SDL provides HMI with the set(s) of choices to be presented to the User. See the below Notes of CreateInteractionChoiceSet.*

Notes about CreateInteractionChoiceSet:

1. *Mobile application in advance creates a set (or several sets) of choices on SDL to be further used in PerformInteraction.*
2. *SDL in advance provides HMI with VR synonyms of created choices via VR.AddCommand (see section 9.4 for details). They are then referenced in VR.PerformInteraction (see section 9.12).*
3. *SDL sends UI choices (those to be displayed) within UI.PerformInteraction directly.*

7.12.2 Request

7.12.2.1 Behavior

HMI must:

1. Distinguish the interaction mode by the following characteristics:

| Mode | SDL sends both RPCs with the following parameters: | Note |
|-------------|---|----------------------|
| VR_ONLY | VR.PerformInteraction {grammarID, params} | grammarID is present |
| | UI.PerformInteraction {params} | No choiceSet |
| MANUAL_ONLY | VR.PerformInteraction {params} | No grammarID |
| | UI.PerformInteraction {choiceSet, params} | choiceSet is present |
| BOTH | VR.PerformInteraction {grammarID, params} | grammarID is present |
| | UI.PerformInteraction {choiceSet, params} | choiceSet is present |

I. 'VR only' interaction mode:

HMI must:

1. Display the dialog with the information provided within:
 - vrHelpTitle parameter – as a title of the message
 - vrHelp parameter – as a list of commands accessible for voice recognition.

Note:

In 'VR only' interaction mode HMI must display the help information only.

2. Respond with SUCCESS result code right after the dialog is displayed (for all of applicable result codes please see section 7.12.3 Response).

3. Close the dialog upon the corresponding VR.PerformInteraction completes (by either the User's choice, or by timeout, or being aborted by system event of a higher priority).

Note:

Please see diagram in section 7.12.4.1 UI.PerformInteraction in 'VR only' mode.

II. 'Manual only' interaction mode:

1. Start timeout provided within `timeout` parameter.
2. Display the PerformInteraction dialog which layout must depend on `interactionLayout` parameter:
 - 2.1. **ICON_ONLY** layout:
 - 2.1.a) Use value of `initialText` parameter – to display the text on the first line of the display (mostly used as a prompting to the User to start the interaction).
 - 2.1.b) Use values of `choiceSet` parameter – to display the choices-related information (text and/or image) in the form of icons.
 - 2.1.c) Proceed with step 3. when the User clicks one of the choice-icons.
 - 2.2. **ICON_WITH_SEARCH** layout:
 - 2.2.a) Use value of `initialText` parameter – to display the text on the first line of the display (mostly used as a prompting to the User to start the interaction).
 - 2.2.b) Use values of `choiceSet` parameter – to display the choices-related information (text and/or image) in the form of icons.
 - 2.2.c) Display a 'Search tab' together with 'Search' button in the top of the dialog.
 - 2.2.c.1) Display the touchscreen keyboard overlaying both choices and text when the user clicks the 'Search tab' area.
 - 2.2.c.2) Provide OnKeyboardInput notification (s) (the sequence depends on KeyboardProperties defined by SDL within UI.SetGlobalProperties. Please see section 7.26 OnKeyboardInput for description and diagrams).
 - 2.2.c.3) Proceed with step 3. When the User presses the 'Search' button.
 - 2.2.d) Proceed with step 3. when the User clicks one of the choices in the list.
 - 2.3. **LIST_ONLY** layout:
 - 2.3.a) Use value of `initialText` parameter – to display the text on the first line of the display (mostly used as a prompting to the User to start the interaction).
 - 2.3.b) Use values of `choiceSet` parameter – to display the choices-related information (text and/or image) in the form of the list.
 - 2.3.c) Proceed with step 3. when the User clicks one of the choices in the list.
 - 2.4. **LIST_WITH_SEARCH** layout:
 - 2.4.a) Use value of `initialText` parameter – to display the text on the first line of the display (mostly used as a prompting to the User to start the interaction).
 - 2.4.b) Use values of `choiceSet` parameter – to display the choices-related information (text and/or image) in the form of the list.
 - 2.4.c) Display a 'Search tab' together with 'Search' button in the top of the dialog.
 - 2.4.c.1) Display the touchscreen keyboard overlaying both choices and text when the user clicks the 'Search tab' area.
 - 2.4.c.2) Provide OnKeyboardInput notification(s) (the sequence depends on KeyboardProperties defined by SDL within UI.SetGlobalProperties. Please see section 7.26 OnKeyboardInput for description and diagrams).
 - 2.4.c.3) Proceed with step 3. when the User presses the 'Search' button.
 - 2.4.d) Proceed with step 3. When the User clicks one of the choices in the list.
 - 2.5. **KEYBOARD** layout:
 - 2.5.a) Display the touchscreen keyboard with 'Search' button.
 - 2.5.b) Provide OnKeyboardInput notification(s) (the sequence depends on KeyboardProperties defined by SDL within UI.SetGlobalProperties. Please see section 7.26 OnKeyboardInput for description and diagrams).
 - 2.5.c) Proceed with step 3. when the User presses the 'Search' button.

3. Respond with **SUCCESS** result code after the User has made a choice AND provide the value of:

3.1. `choiceID` parameter for the steps of '2.1.c)', '2.2.d)', '2.3.c)', '2.4.d)'.

3.2. `manualTextEntry` parameter for the steps of '2.2.c.3)', '2.4.c.3)', '2.5.c'.

(for all of applicable result codes please see section 7.12.3 Response).

III. 'Both' interaction mode:

1. Display the dialog with the information provided within:

- `vrHelpTitle` parameter – as a title of the message
- `vrHelp` parameter – as a list of commands accessible for voice recognition.

Note:

This 'VR Help'-dialog must be kept in correspondence with currently active VR session that SDL requests via VR.PerformInteraction.

2. Close 'VR Help'-dialog when the corresponding VR session:

- 2.1. Completes successfully by the User's VR choice. Proceed with **step 3**.
- 2.2. Completes by the timeout. Proceed with **step 4**.
- 2.3. Is aborted by the User. Proceed with **step 4**.

3. Respond with **SUCCESS** result code (for all of applicable result codes please see section 7.12.3 Response).

4. Start 'Manual only' interaction mode.

4.1. Start timeout provided within `timeout` parameter.

4.2. Display the PerformInteraction dialog which layout must depend on `interactionLayout` parameter:

4.2.1. `ICON_ONLY` layout:

- 4.2.1.a) Use value of `initialText` parameter – to display the text on the first line of the display (mostly used as a prompting to the User to start the interaction).
- 4.2.1.b) Use values of `choiceSet` parameter – to display the choices-related information (text and/or image) in the form of icons.
- 4.2.1.c) Proceed with step 4.3. when the User clicks one of the choice-icons.

4.2.2. `ICON_WITH_SEARCH` layout:

- 4.2.2.a) Use value of `initialText` parameter – to display the text on the first line of the display (mostly used as a prompting to the User to start the interaction).
- 4.2.2.b) Use values of `choiceSet` parameter – to display the choices-related information (text and/or image) in the form of icons.
- 4.2.2.c) Display a 'Search tab' together with 'Search' button in the top of the dialog.
 - 4.2.2.c.1) Display the touchscreen keyboard overlaying both choices and text when the user clicks the 'Search tab' area.
 - 4.2.2.c.2) Provide `OnKeyboardInput` notification (s) (the sequence depends on `KeyboardProperties` defined by SDL within `UI.SetGlobalProperties`. Please see section 7.26 `OnKeyboardInput` for description and diagrams).
 - 4.2.2.c.3) Proceed with step 4.3. When the User presses the 'Search' button.
- 4.2.2.d) Proceed with step 4.3. when the User clicks one of the choices in the list.

4.2.3. `LIST_ONLY` layout:

- 4.2.3.a) Use value of `initialText` parameter – to display the text on the first line of the display (mostly used as a prompting to the User to start the interaction).
- 4.2.3.b) Use values of `choiceSet` parameter – to display the choices-related information (text and/or image) in the form of the list.
- 4.2.3.c) Proceed with step 4.3. when the User clicks one of the choices in the list.

4.2.4. `LIST_WITH_SEARCH` layout:

- 4.2.4.a) Use value of `initialText` parameter – to display the text on the first line of the display (mostly used as a prompting to the User to start the interaction).
- 4.2.4.b) Use values of `choiceSet` parameter – to display the choices-related information (text and/or image) in the form of the list.

- 4.2.4.c) Display a ‘Search tab’ together with ‘Search’ button in the top of the dialog.
- 4.2.4.c.1) Display the touchscreen keyboard overlaying both choices and text when the user clicks the ‘Search tab’ area.
- 4.2.4.c.2) Provide OnKeyboardInput notification(s) (the sequence depends on KeyboardProperties defined by SDL within UI.SetGlobalProperties. Please see section 7.26 OnKeyboardInput for description and diagrams).
- 4.2.4.c.3) Proceed with step 4.3. when the User presses the ‘Search’ button.
- 4.2.4.d) Proceed with step 4.3. When the User clicks one of the choices in the list.

4.2.5. KEYBOARD layout:

- 4.2.5.a) Display the touchscreen keyboard with ‘Search’ button.
- 4.2.5.b) Provide OnKeyboardInput notification(s) (the sequence depends on KeyboardProperties defined by SDL within UI.SetGlobalProperties. Please see section 7.26 OnKeyboardInput for description and diagrams).
- 4.2.5.c) Proceed with step 4.3. when the User presses the ‘Search’ button.

- 4.3. Respond with SUCCESS result code after the User has made a choice AND provide the value of:
- 4.3.1. choiceID parameter for the steps of ‘4.2.1.c’, ‘4.2.2.d’, ‘4.2.3.c’, ‘4.2.4.d’.
- 4.3.2. manualTextEntry parameter for the steps of ‘4.2.2.c.3’, ‘4.2.4.c.3’, ‘4.2.5.c’.
(for all of applicable result codes please see section 7.12.3 Response).

7.12.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-------------|------------------------|-----------|--|---|
| initialText | Common.TextFieldStruct | false | - | <p>The text that must be displayed:</p> <ul style="list-style-type: none"> - When the interaction begins - On the first line of a multiline screen - Centered. <p>See TextFieldStruct, TextFieldName (uses initialInteractionText).</p> |
| choiceSet | Common.Choice | false | Array = true MinSize = 1 MaxSize = 100 | <p>Array of one or more Choice Sets. Each one is represented with:</p> <ul style="list-style-type: none"> - The mandatory ID (to be returned by HMI in the response if the corresponding Choice Set is selected by the User) - Optional name and/or picture. <p>See Choice.</p> <p>HMI must provide the possibility for the User to choose among these choices.</p> |
| vrHelpTitle | String | false | MaxLength = 500 | <p>VR Help Title text.</p> <p>If omitted on supported displays, the default HU system help title should be used.</p> |
| vrHelp | Common.VrHelpItem | false | Array = true MinSize = 1 MaxSize = 100 | <p>VR Help Items that must be displayed on-screen after HMI recognizes ‘Help’ command during PerformInteraction.</p> <p>See VrHelpItem.</p> |

| | | | | |
|-------------------|-------------------|-------|--|--|
| timeout | Integer | true | Minvalue = 5000 Maxvalue = 100000 Defvalue = 10000 | Timeout in milliseconds. The amount of time the HMI must wait for the User to make a choice. After the time is up HMI respond with TIMED_OUT result code. |
| interactionLayout | Common.LayoutMode | false | - | See LayoutMode. |
| appID | Integer | true | - | ID of the application related to this RPC. |

7.4.2.3 TextFieldStruct Structure

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------------------------|-----------|-----------------|--|
| fieldName | Common.TextFileDialogName | true | - | The name of the field where the text must be displayed in. |
| fieldText | String | true | Maxlength = 500 | The text to be displayed. |

7.12.2.4 TextFieldName

| Element name | Short Description |
|------------------------|---|
| initialInteractionText | Must be displayed when the interaction begins. The text must be displayed on the first line of a multiline display, and must be centered. Applies to <code>PerformInteraction</code> |

7.12.2.5 Choice

| Param Name | Type | Mandatory | Additional | Description |
|----------------|--------------|-----------|----------------------------------|---|
| choiceID | Integer | true | minvalue = 0 maxvalue = 65535 | The unique within the concerned application identifier for this choice |
| menuName | String | false | Maxlength = 500 | The text to be displayed in the onscreen menu indicating the name of the choice (e.g. 'Yes'). |
| image | Common.Image | false | - | Image that must appear in the menu, representing this choice. See <code>Image</code> |
| secondaryText | String | false | Maxlength = 500 | Optional secondary text to display; e.g. address of POI in a search result entry |
| tertiaryText | String | false | Maxlength = 500 | Optional tertiary text to display; e.g. distance to POI for a search result entry |
| secondaryImage | Common.Image | false | - | Optional secondary image struct for choice |

7.12.2.6 VrHelpItem

| Param Name | Type | Mandatory | Additional | Description |
|------------|--------------|-----------|--------------------------------|---|
| text | String | True | maxlength = 500 | The text that must be displayed as a name for VR Help item. |
| image | Common.Image | False | - | Image that must be displayed by HU to represent the VR Help item. See Image |
| position | Integer | false | minvalue = 1 maxvalue = 100 | This value is the position within the elements of VR Help menu where the item must be added to: |

7.12.2.7 LayoutMode

| Element name | Short Description |
|------------------|---|
| ICON_ONLY | This mode causes the interaction to display the previous set of choices as icons |
| ICON_WITH_SEARCH | This mode causes the interaction to display the previous set of choices as icons along with a search field in the HMI. |
| LIST_ONLY | This mode causes the interaction to display the previous set of choices as a list |
| LIST_WITH_SEARCH | This mode causes the interaction to display the previous set of choices as a list along with a search field in the HMI. |
| KEYBOARD | This mode causes the interaction to immediately display a keyboard entry through the HMI. |

7.12.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|------------------|--|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS The cases described in 7.12.2: 'I.2', 'II.3', 'III.3', 'III.4.3'. | JSON response | Regular response | choiceID, manualTextEntry, code: 0 | See section 7.12.4 Parameters. |
| Failure | ABORTED The interaction is aborted by the User or system event of higher priority. | JSON error message | Regular response | code: 5 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | TIMED_OUT The User has not made a choice during timeout defined within request. | | | code: 10 | |
| | GENERIC_ERROR The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

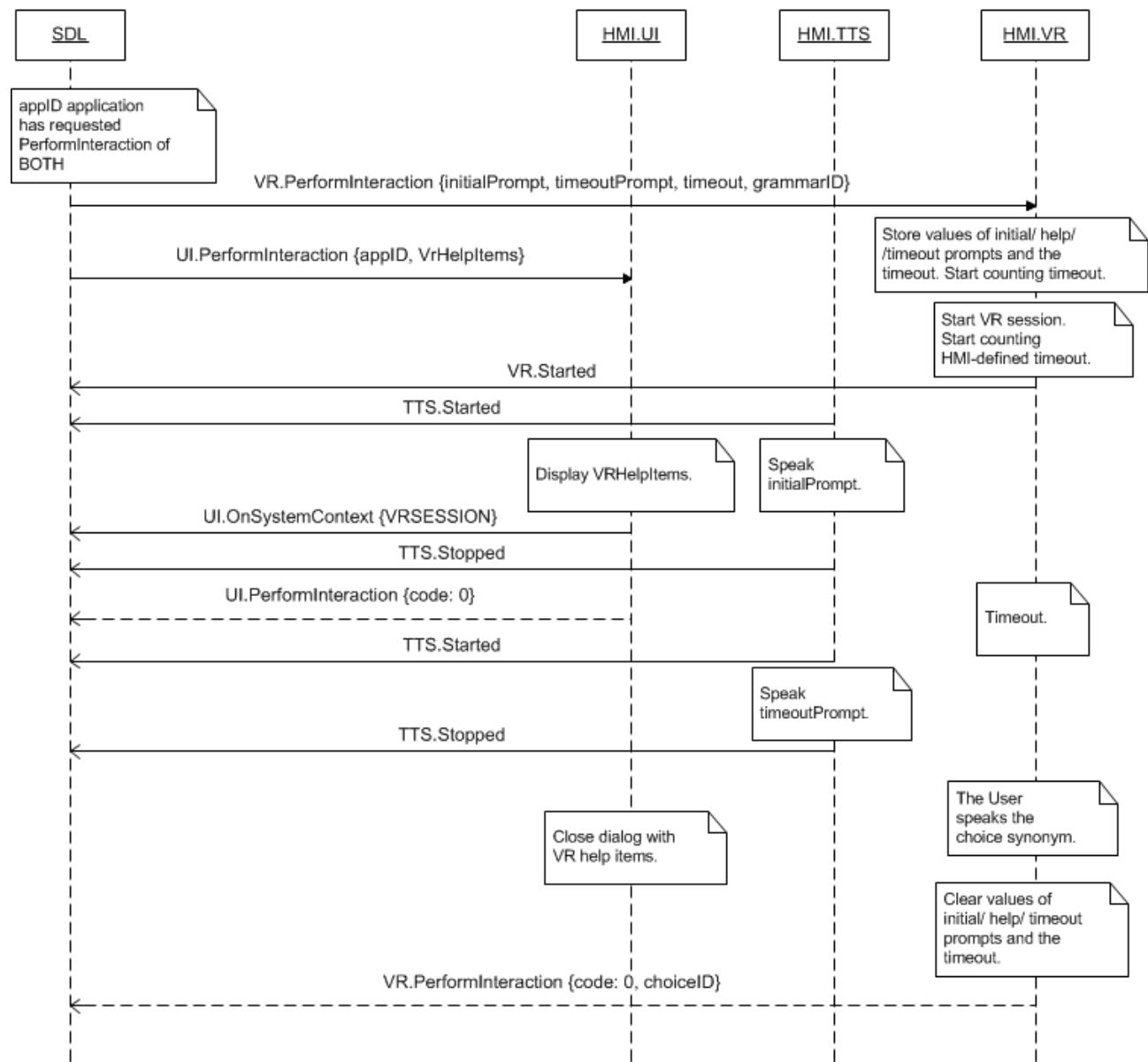
7.12.4 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

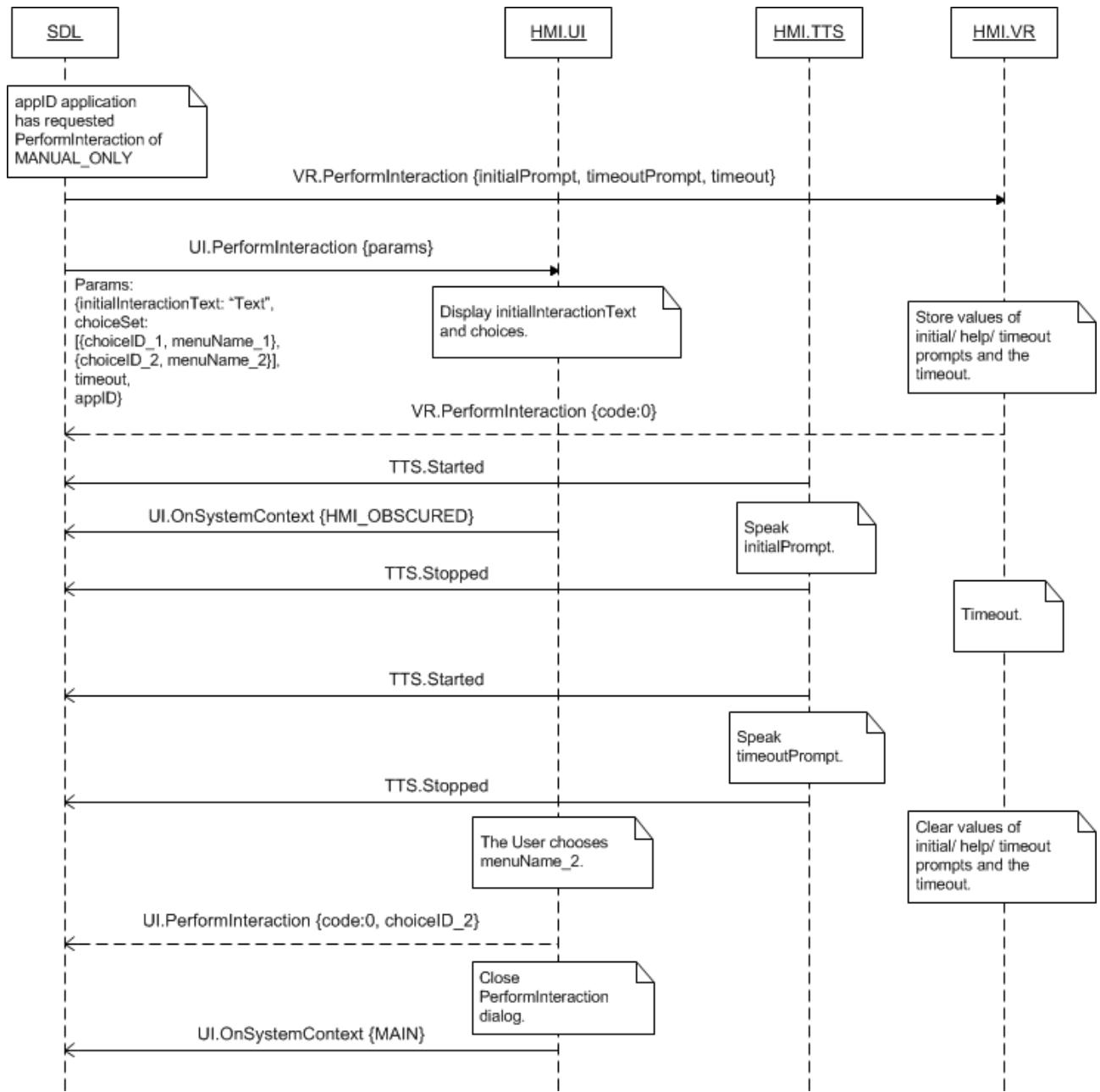
| Param Name | Type | Mandatory | Additional | Description |
|-----------------|---------|-----------|---------------------------------------|--|
| choiceID | Integer | false | minvalue = 0 maxvalue = 2000000000 | ID that represents the choice made by the User (one among those provided within the request). |
| manualTextEntry | String | false | minlength = 0 maxlength = 500 | Manually entered text selection, e.g. through keyboard Can be returned in lieu of choiceID, depending on trigger source |

7.12.4 Sequence Diagrams

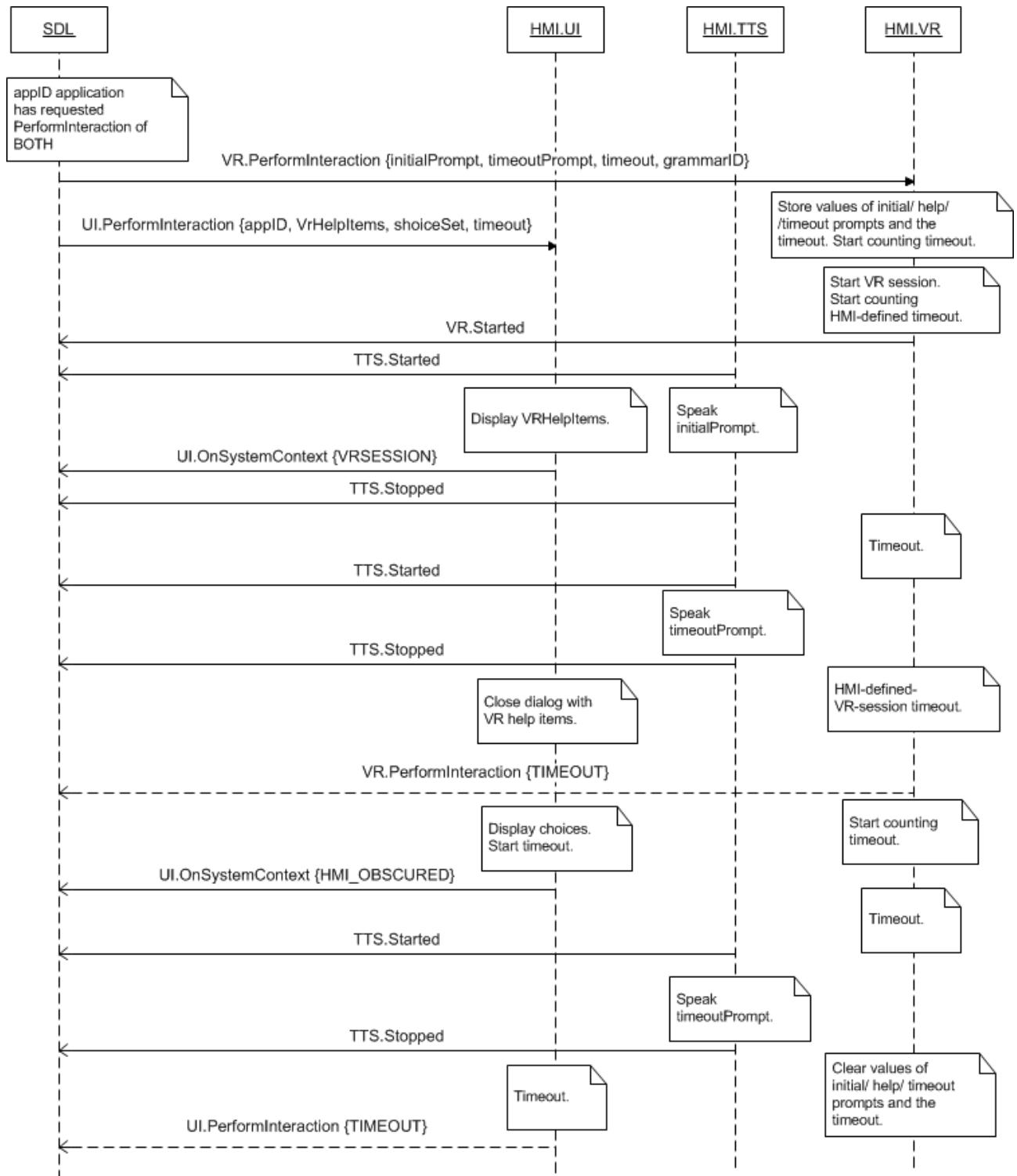
7.12.4.1 UI.PerformInteraction in ‘VR only’ mode successfully completed



7.12.4.2 UI.PerformInteraction in ‘Manual only’ mode successfully completed



7.12.4.3 UI.PerformInteraction in 'Both' mode timed out



7.12.5 JSON Messages Examples

7.12.5.1 Request

```
{
    "id" : 79,
    "jsonrpc" : "2.0",
    "method" : "UI.PerformInteraction",
    "params" :
    {
        "initialText" :
        {
            "fieldName" : initialInteractionText,
            "fieldText" : "Choose the station:"
        },
    }
}
```

```

"choiceSet" :
[
    {
        "choiceID" : 2415,
        "menuName" : "Sky.FM"
    },
    {
        "choiceID" : 2416,
        "menuName" : "Paradise"
    },
    {
        "choiceID" : 2417,
        "menuName" : "100 XR"
    }
],
"vrHelp" :
[
    {
        "text" : "Sky FM",
        "image" :
            [
                {
                    "value" : "tmp/SDL/app/Pandora/icon_5410.jpg",
                    "imageType" : DYNAMIC
                },
                "position" : 1
            },
        {
            "text" : "Paradise",
            "image" :
                [
                    {
                        "value" : "tmp/SDL/app/Pandora/icon_5423.jpeg",
                        "imageType" : DYNAMIC
                    },
                    "position" : 2
                },
            {
                "text" : "100 XR",
                "image" :
                    [
                        {
                            "value" : "tmp/SDL/app/Pandora/icon_5465.jpeg",
                            "imageType" : DYNAMIC
                        },
                        "position" : 3
                    }
            ],
            "timeout" : 15000,
            "appID" : 6493
        }
    }
]

```

7.12.5.2 Response

```
{
    "id" : 79,
    "jsonrpc" : "2.0",
    "result" :
    {
        "choiceID" : 2416
    }
}
```

```

        "code" : 0,
        "method" : "UI.PerformInteraction"
    }
}

```

7.12.5.3 Error message

```

{
    "id" : 79,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 10,
        "message" : "Overlay reached the maximum timeout and closed",
        "data" :
        {
            "method" : "UI.PerformInteraction"
        }
    }
}

```

7.12.6 D-Bus Messages Examples

7.12.6.1 Request

[REDACTED]

7.12.6.2 Response

[REDACTED]

7.12.6.3 Failure

[REDACTED]

7.13 SetMediaClockTimer

7.13.1 Description

| | |
|-----------------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Set a value and the update mode of the media clock. |

With this RPC SDL requests to

- set the initial value of media clock
- update it correspondingly to the provided parameters.

The request may concern the application not being currently active on HMI.

Note:

SDL uses the information about:

- *mediaClock field*
- *the format of time value*

confirmed to be supported by HMI within response to UI.GetCapabilities

SDL will not send the request if mediaClock field is not supported.

7.13.2 Request

7.13.2.1 Behavior

HMI must:

1. Perform the following depending on the `updateMode` parameter:

Important Note:

If the request comes for the application

- Currently active on HMI, the requested updates must be immediately displayed
- Currently NOT active (being in background) on HMI, HMI must
 - Perform the requested updates in background
 - Display the value of the media clock valid at the time the corresponding application becomes activated on HMI.
- COUNTUP / COUNTDOWN modes:
 - Start counting up / down from the requested `startTime` value with the step of 1 second
 - Continue counting up / down until:
 - The next request of `SetMediaClockTimer` with appropriate parameters comes
 - .Zero value in case of COUNTDOWN.
- PAUSE mode:
 - Pause the clock that is counting up / down.
- RESUME mode:
 - Continue counting up / down (in the mode that was in effect before pausing) the clock that has been paused.
- CLEAR mode:
 - Clear the clock value (whatever being currently updated or paused). It is up to HMI to determine the way the media clock timer is cleared: either to remove it from display or to set it to zero.

Important Note:

HMI must remember the mode and the value (continuing to update it in case of COUNTUP / COUNTDOWN) of the media clock timer associated with `appID` and display the accurate values whenever the `appID` application is activated after having been deactivated.

Note:

Initially, the `appID` together with other application-related information is provided by SDL within one of `UpdateAppList` and `OnAppRegistered` RPCs.

2. Respond with the result code correspondingly to the results of this RPC execution.

Note:

The applicable to this RPC result codes are provided in section 7.13.3 Response.

Note:

The sequence diagrams describing the expected HMI behavior are provided in the section 7.13.4 Sequence Diagrams

7.13.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|-------------------------|-------------------------------------|-----------|---|
| <code>startTime</code> | <code>Common.TimeFormat</code> | false | The starting time value to be counted up / down from. See <code>TimeFormat</code> |
| <code>updateMode</code> | <code>Common.ClockUpdateMode</code> | true | The update method of the clock value: COUNTUP, COUNTDOWN, PAUSE, RESUME, CLEAR. In case of PAUSE, RESUME, or CLEAR, the <code>startTime</code> value should be ignored and left out if provided. See <code>ClockUpdateMode</code> . |
| <code>appID</code> | <code>Integer</code> | true | ID of the application that concerns this RPC. |

7.13.2.3 TimeFormat

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|-------------------------------|------------------------------|
| hours | Integer | true | minvalue = 0 maxvalue = 59 | The hour of the media clock. |
| minutes | Integer | true | minvalue = 0 maxvalue = 59 | The minute. |
| seconds | Integer | true | minvalue = 0 maxvalue = 59 | The second. |

7.13.2.4 ClockUpdateMode

| Element name | Short Description |
|--------------|--|
| COUNTUP | HMI must start the media clock timer counting upwards, as in time elapsed, in increments of 1 second. |
| COUNTDOWN | HMI must start the media clock timer counting downwards, as in time remaining, in decrements of 1 second. |
| PAUSE | HMI must pause the media clock timer. |
| RESUME | HMI must resume the media clock timer. The timer must resume counting in whatever mode was in effect before pausing (i.e. COUNTUP or COUNTDOWN). |
| CLEAR | HMI must clear the media clock timer. |

7.13.3 Response

Note:

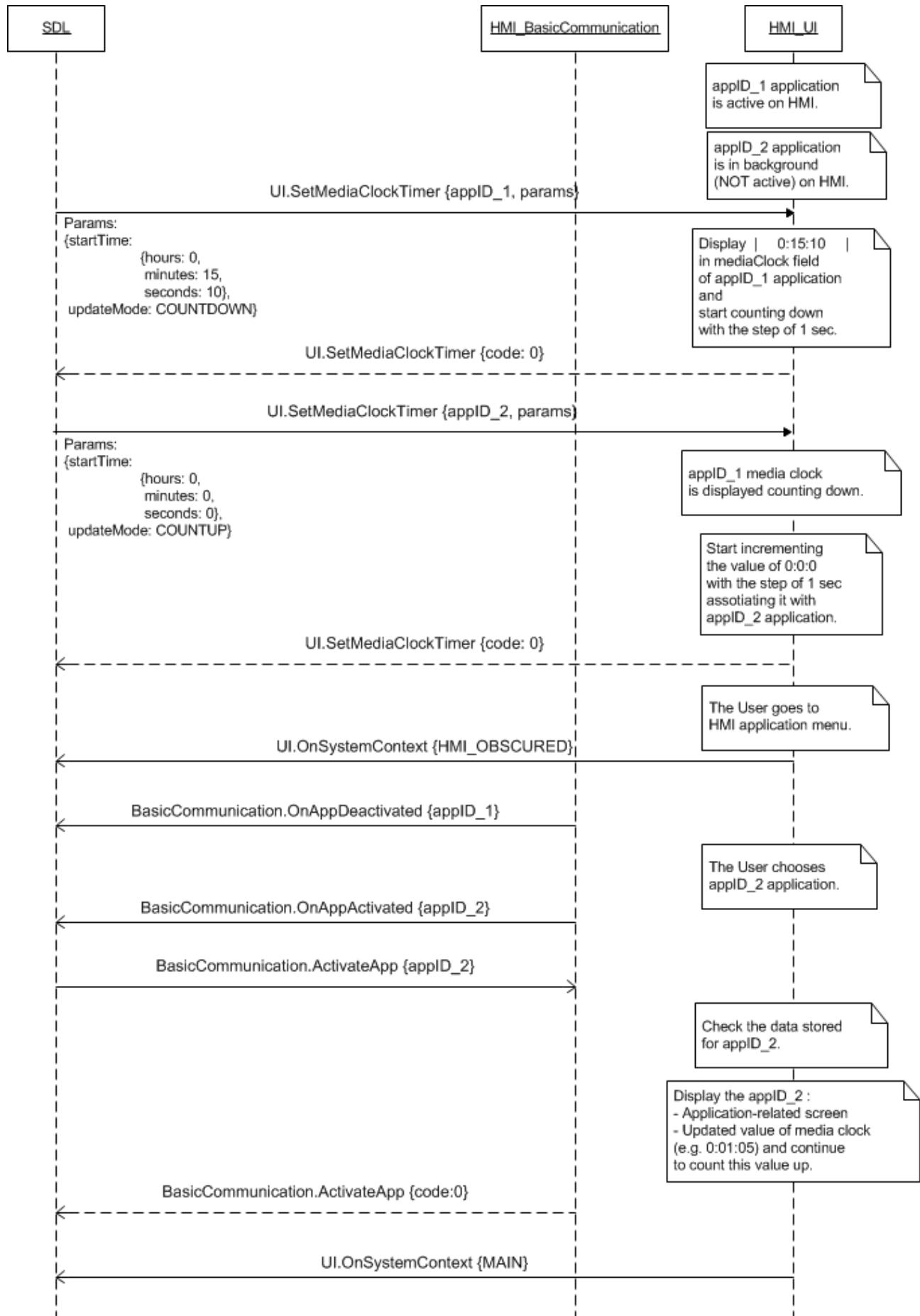
There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: 1) HMI has started counting up / down the media clock from the requested value (in case of COUNTUP / COUNTDOWN requested mode). 2) HMI has paused the media clock value (PAUSE requested mode). 3) HMI has resumed the media clock in the mode that had been in effect before pausing, i.e. has started counting up or down (RESUME requested mode). 4) HMI has cleared the existing media clock value (CLEAR requested mode). | JSON response | Method return | code: 0 | HMI must respond right after the action requested with updateMode parameter has been performed. See section 7.13.4 Sequence Diagrams for details. |
| Failure | IGNORED: 1) Request with RESUME mode arrives when the media clock is counting, already resumed or cleared with the previous request. 2) Request with PAUSE mode arrives when the media clock is paused or cleared with the previous request. | JSON error message | Method return | code: 6 | Applicable to this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |

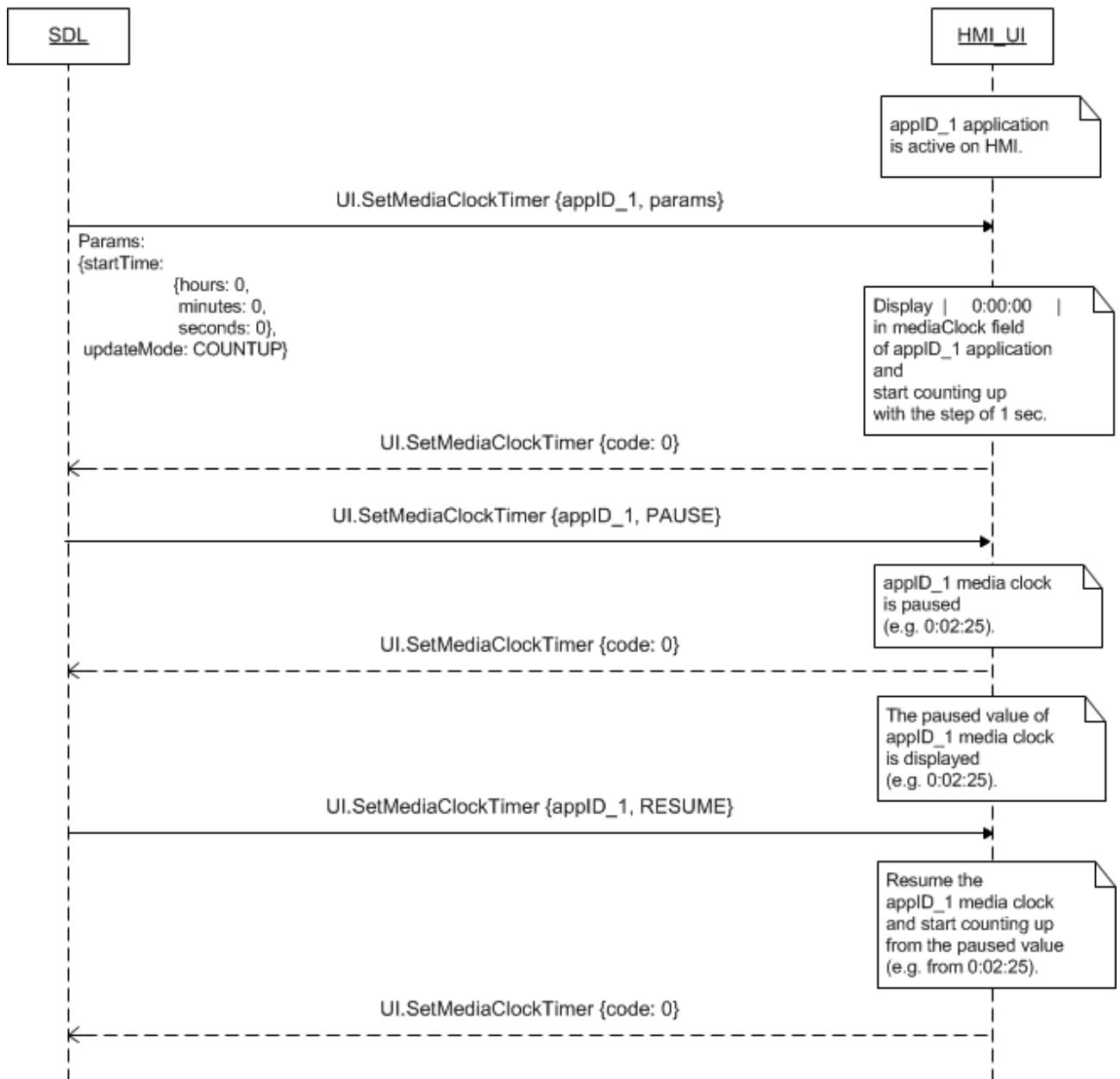
| | | | | |
|--|--|--|----------|--|
| | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | | code: 11 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | code: 22 | |

7.13.4 Sequence Diagrams

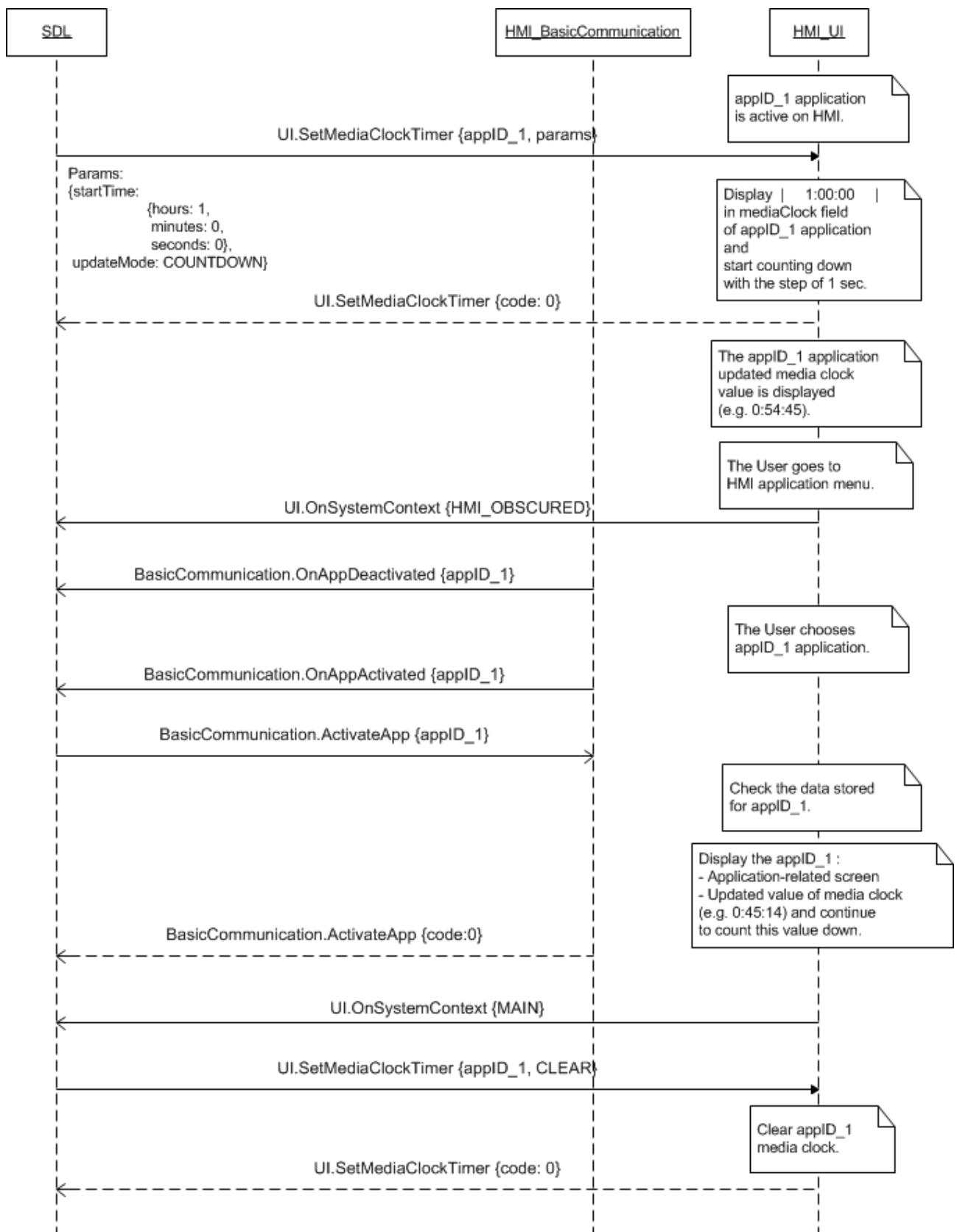
7.13.4.1 SetMediaClockTimer of COUNTUP and COUNTDOWN modes for the cases of active and background applications



7.13.4.2 SetMediaClockTimer of PAUSE and RESUME for the application active on HMI



7.13.4.3 SetMediaClockTimer of COUNTDOWN for the active application that is then deactivated and activated again and SetMediaClockTimer of CLEAR for the active application.



7.13.5 JSON Messages Examples

7.13.5.1 Request

```
{
}
```

```

    "id" : 109,
    "jsonrpc" : "2.0",
    "method" : "UI.SetMediaClockTimer",
    "params" :
    {
        "startTime" :
        {
            "hours" : 0,
            "minutes" : 18,
            "seconds" : 17
        },
        "updateMode" : COUNTUP,
        "appID" : 65146
    }
}

```

7.13.5.2 Response

```

{
    "id" : 109,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "UI.SetMediaClockTimer"
    }
}

```

7.13.5.3 Error message

```

{
    "id" : 109,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "Invalid data",
        "data" :
        {
            "method" : "UI.SetMediaClockTimer"
        }
    }
}

```

7.13.6 D-Bus Messages Examples

7.13.6.1 Request

7.13.6.2 Response

7.13.6.3 Failure

7.14 SetGlobalProperties

7.14.1 Description

| | |
|----------|---------------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Set the UI properties of application. |

SDL requests to set the values for VR help, the name and icon for in-application menu and the properties of touchscreen keyboard.

The request may arrive for the application whatever being active or in background on HMI.

7.14.2 Request

7.14.2.1 Behavior

HMI must:

1. Store the information associating it with appID.

Note:

Initially, the appID together with other application-related information is provided by SDL within UpdateAppList or OnAppRegistered RPCs.

2. Set the requested values for:

- VR help: whenever the User activates VR, HMI must display the list of commands available for voice recognition. SDL provides the title for this list (vrHelpTitle parameter) and the list of commands itself (vrHelp parameter which is the array of VrHelpItem-s).

Important Note:

If HMI-defined VR commands are accessible together with those provided by SDL via VR.AddCommand, HMI must:

- *Add the corresponding VR synonyms to the list of VR help items provided by SDL via UI.SetGlobalProperties*
- *Display the complete list of available VR commands (SDL-defined and HMI-defined ones) when the User activates VR.*

- In-application menu: HMI must display the in-application menu for every active application, where the SDL-requested commands (UI.AddCommand) and sub menus (UI.AddSubMenu) must be added to. SDL provides the values for the name (menuTitle parameter) and for the icon (menuIcon parameter) of this in-application menu.
- Touchscreen keyboard: if supported HMI displays the onscreen keyboard upon User's request. SDL provides the properties of this keyboard:

Note:

The values for in-application menu and touchscreen keyboard will be allowed by SDL for navigation type of application only.

3. Respond to the request.

Note:

The applicable to this RPC result codes are provided in section 7.14.3 Response.

Note:

The sequence diagrams describing the expected HMI behavior are provided in the section 7.14.4 Sequence Diagrams

7.14.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-------------|--------|-----------|-----------------|---|
| vrHelpTitle | String | false | Maxlength = 500 | The text that must be displayed as a title of VR Help Menu. |

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|---------------------------|-----------|--|--|
| vrHelp | Common.VrHelpItem | false | Array = true MinSize = 1 MaxSize = 100 | VR help items – i.e. the text strings to be displayed, and when pronounced by the user the recognition of any of which must trigger the corresponding VR command. See VrHelpItem. |
| menuTitle | String | false | MaxLength = 500 | Optional text to label an app menu button (for certain touchscreen platforms). |
| menuIcon | Common.Image | false | - | Optional icon to draw on an app menu button (for certain touchscreen platforms). |
| keyboardProperties | Common.KeyboardProperties | false | - | On-screen keyboard configuration (if available). |
| appID | Integer | true | - | ID of the application that concerns this RPC. |

7.14.2.6 VrHelpItem

| Param Name | Type | Mandatory | Additional | Description |
|------------|--------------|-----------|--------------------------------|---|
| text | String | True | maxLength = 500 | The text that must be displayed as a name for VR Help item. |
| image | Common.Image | False | - | Image that must be displayed by HU to represent the VR Help item. See Image |
| position | Integer | false | minValue = 1 maxValue = 100 | This value is the position within the elements of VR Help menu where the item must be added to: |

7.14.2.4 KeyboardProperties

| Param Name | Type | Mandatory | Additional | Description |
|----------------------|-----------------------|-----------|---|--|
| language | Common.Language | false | - | The keyboard language. |
| keyboardLayout | Common.KeyboardLayout | false | - | Desired keyboard layout. |
| sendDynamicEntry | Boolean | false | - | In this mode, all keypresses will be sent as they occur. If disabled, entire string of text will be returned only once submitted by user. If omitted, this value will be set to FALSE. |
| keypressMode | Common.KeyPressMode | false | - | Desired keypress mode.If omitted, this value will be set to RESEND_CURRENT_ENTRY. |
| limitedCharacterList | String | false | Array = true maxLength = 1 minSize = 1 maxSize = 100 | Array of keyboard characters to enable. All omitted characters will be greyed out (disabled) on the keyboard. If omitted, the entire keyboard will be enabled. |
| autoCompleteText | String | false | maxLength = 1000 | Allows an app to prepopulate the text field with a suggested or completed entry as the user types |

7.14.2.5 KeyboardLayout

| Element name | Short Description |
|--------------|---|
| QWERTY | QWERTY layout (the name comes from the first six keys appearing on the top left letter row of the keyboard and read from left to right) |
| QWERTZ | QWERTZ layout (the name comes from the first six keys appearing on the top left letter row of the keyboard and read from left to right) |
| AZERTY | AZERTY layout (the name comes from the first six keys appearing on the top left letter row of the keyboard and read from left to right) |

7.14.3 Response

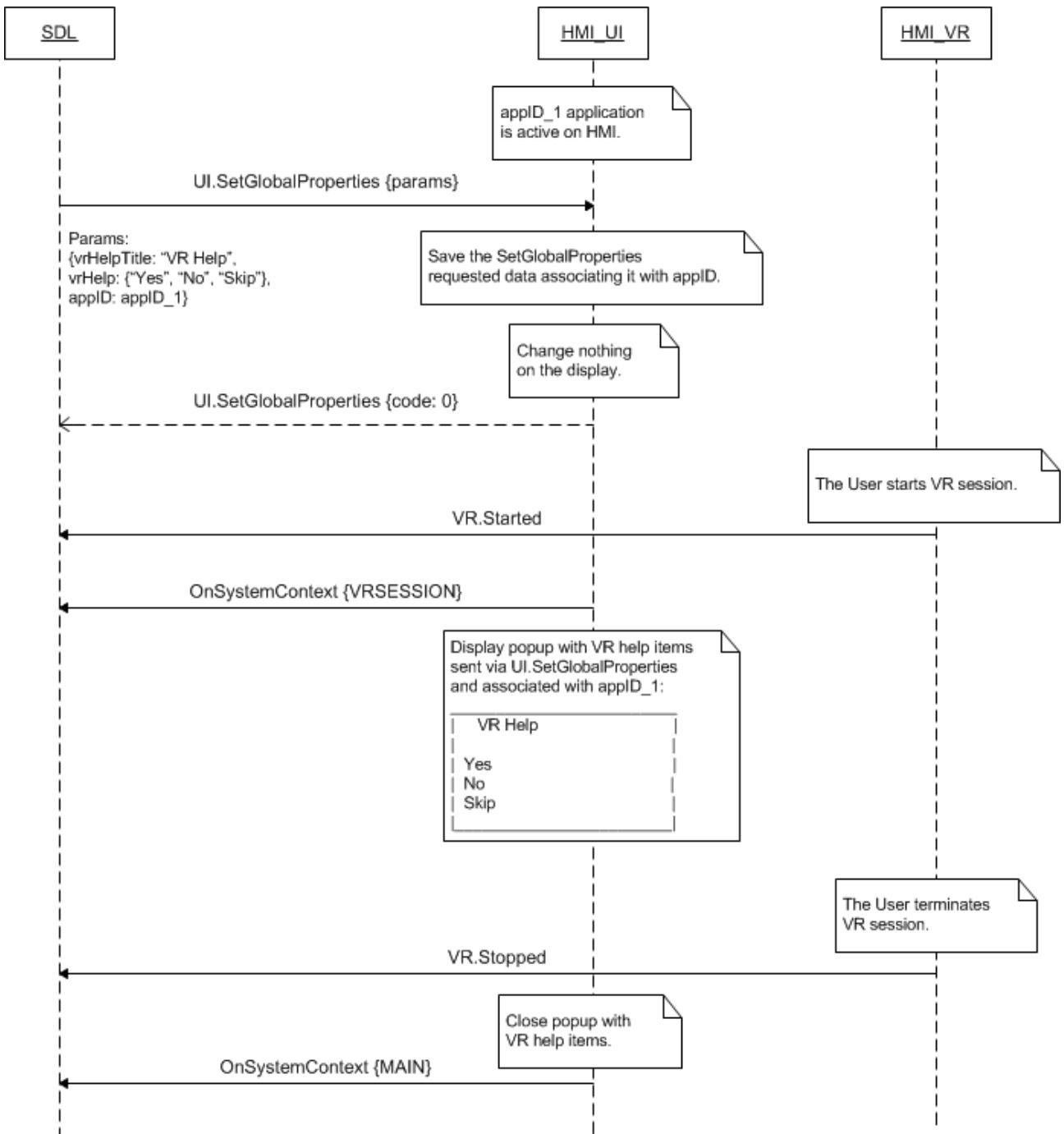
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI has set the requested properties.. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

7.14.4 Sequence Diagrams

7.14.4.1 SetGlobalProperties for the application active on HMI and the later VR activation



7.14.5 JSON Messages Examples

7.14.5.1 Request

```
{
    "id" : 116,
    "jsonrpc" : "2.0",
    "method" : "UI.SetGlobalProperties",
    "params" :
    {
        "vrHelpTitle" : "Choose the action",
        "vrHelp" :
        [
            {
                "text" : "Pause",
                "image" :
                [
                    {
                        "value" : "tmp/SDL/app/Pandora/icon_1067.jpg",

```

```

        "imageType" : DYNAMIC
    ],
    "position" : 1
},
{
    "text" : "Resume",
    "image" :
    [
        {
            "value" : "tmp/SDL/app/Pandora/icon_1083.jpeg",
            "imageType" : DYNAMIC
        },
        "position" : 2
},
{
    "text" : "Skip",
    "image" :
    [
        {
            "value" : "tmp/SDL/app/Pandora/icon_1013.jpeg",
            "imageType" : DYNAMIC
        },
        "position" : 3
},
{
    "text" : "Bookmark",
    "image" :
    [
        {
            "value" : "tmp/SDL/app/Pandora/icon_1046.jpeg",
            "imageType" : DYNAMIC
        },
        "position" : 4
}
],
"appID" : 53880
}
}

```

7.14.5.2 Response

```
{
    "id" : 116,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "UI.SetGlobalProperties"
    }
}
```

7.14.5.3 Error message

```
{
    "id" : 116,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "Invalid data",
        "data" :
        {
            "method" : "UI.SetGlobalProperties"
        }
    }
}
```

7.14.6 D-Bus Messages Examples

7.14.6.1 Request

```
method call sender=:1.128 -> dest=com.ford.sdl.hmi serial=85 path=/;
interface=com.ford.sdl.hmi.UI;
member=SetGlobalProperties
    struct {
        boolean true
        string "This is the VR help title"
    }
    struct {
        boolean true
        array [
            struct {
                string "VR help item"
                struct {
                    boolean true
                    struct {
                        string
                    }
                }
                int32 1
            }
        ]
        int32 1
    }
    struct {
        string "Help Item 2"
        struct {
            boolean true
            struct {
                string
            }
        }
        int32 1
    }
    struct {
        string "third item"
        struct {
            boolean true
            struct {
                string
            }
        }
        int32 2
    }
    struct {
        string "Hugo/action.png"
        int32 1
    }
    struct {
        string "Hugo/action.png"
        int32 1
    }
    struct {
        string "Hugo/action.png"
        int32 1
    }
    struct {
        boolean false
        string ""
    }
    struct {
        boolean false
        struct {
            string ""
            int32 -1
        }
    }
    struct {
        boolean false
        struct {
            struct {
                string
            }
        }
    }
"/home/abritanova/Work/smartdevicelink/applink/RB_E3.0/bin/Hugo/action.png"
"/home/abritanova/Work/smartdevicelink/applink/RB_E3.0/bin/Hugo/action.png"
"/home/abritanova/Work/smartdevicelink/applink/RB_E3.0/bin/Hugo/action.png"
```

```

        boolean false
        int32 -1
    }
    struct {
        boolean false
        int32 -1
    }
    struct {
        boolean false
        boolean false
    }
    struct {
        boolean false
        int32 -1
    }
    struct {
        boolean false
        array [
        ]
    }
    struct {
        boolean false
        string ""
    }
}
int32 65537

```

7.14.6.2 Response

```

method return sender=:1.129 -> dest=:1.128 reply_serial=85
    int32 0
    string ""

```

7.14.6.3 Failure

```

method return sender=:1.129 -> dest=:1.128 reply_serial=85
    int32 11
    string "The data sent is invalid"

```

7.15 ChangeRegistration

7.15.1 Description

| | |
|-----------------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Change the display language for the named application on HMI |

SDL requests to set a display language for the named application on HMI.

The request may arrive for the application whatever being active or in background on HMI.

Note:

SDL will send the language value confirmed to be supported by HMI via UI.GetCapabilities.

7.15.2 Request

7.15.2.1 Behavior

HMI must:

1. Store the provided information associating it with application's appID.

2. Respond to the request.

Note:

The applicable to this RPC result codes are provided in section 7.14.3 Response.

Note:

The sequence diagrams describing the expected HMI behavior are provided in the section 7.14.4 Sequence Diagrams

7.15.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-----------------------|-----------------|-----------|-----------------|--|
| language | Common.Language | true | | The language requested to be switched to. See Language. |
| appName | string | false | maxlength = 100 | Request new app name registration Needs to be unique over all applications. May not be empty. May not start with a new line character. May not interfere with any name or synonym of any registered applications. Applications with the same name will be rejected. (SDL makes all the checks) |
| ngnMediaScreenAppName | string | false | maxlength = 100 | Request new app short name registration |
| appID | Integer | true | | ID of the application that relates to this RPC. |

7.15.2.3 Language

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |

| Element Name | Value | Description |
|--------------|-------|------------------------|
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

7.15.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|---------------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI has changed the UI language for the named application. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | JSON error message | Method return | code: 11 message | Applicable for this RPC result codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 message | Please see Result Enumeration for all SDL-supported codes. |

7.15.4 Sequence Diagrams

7.15.4.1 ChangeRegistration

7.15.5 JSON Messages Examples

7.15.5.1 Request

```
{
  "id" : 117,
  "jsonrpc" : "2.0",
  "method" : "UI.ChangeRegistration",
  "params" :
  {
    "Language" : PT-PT,
    "appID" : 65146
  }
}
```

7.15.5.2 Response

```
{
  "id" : 117,
  "jsonrpc" : "2.0",
```

```

    "result" :
    {
        "code" : 0,
        "method" : "UI.ChangeRegistration"
    }
}

```

7.15.5.3 Error message

```

{
    "id" : 117,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "Invalid data",
        "data" :
        {
            "method" : "UI.ChangeRegistration"
        }
    }
}

```

7.15.6 D-Bus Messages Examples

Note:

7.15.6.1 Request

```

method call           //message type (request)
sender=:1.139         //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
-> dest=com.ford.sdl.hmi //the name (requested by and assigned to HMI) to receive a message
serial=108            //serial number of the message
path=/;               //the path to the object instance
interface=com.ford.sdl.hmi.UI;
member=ChangeRegistration
    int32 2           // language (FR-CA)
    int32 65538        // appID

```

7.15.6.2 Response

```

method return          //message type (successful response)
sender=:1.140          //unique for-the-lifetime-of-the-bus name automatically assigned to HMI
-> dest=:1.139          //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
reply_serial=108        //serial number of the corresponding method_call message
    int32 0             //code (SUCCESS)
    string ""            //message

```

7.15.6.3 Failure

```

method return          //message type (error response)
sender=:1.140          //unique for-the-lifetime-of-the-bus name automatically assigned to HMI
-> dest=:1.139          //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
reply_serial=108        //serial number of the corresponding method_call message
    int32 22            // code (GENERIC_ERROR)
    string "During the API call an unknown error has occurred."

```

7.16 GetLanguage

7.16.1 Description

| | |
|-------|----------|
| Type: | Function |
|-------|----------|

| | |
|-----------------|--|
| Sender: | SDL |
| Purpose: | Find out the current HMI display language. |

SDL inquires the current HMI display language.

Note:

'HMI display language' is meant to be the language of HMI-defined menus, screens, popups, etc. HMI display language and application display language (requested via UI.ChangeRegistration) might be different.

This RPC is sent by SDL after UI readiness is confirmed by UI.IsReady. If later the User changes the HMI display language, HMI must inform SDL about this event via UI.OnLanguageChange notification.

7.16.2 Request

7.16.2.1 Behavior

HMI must:

1. Check the HMI display language currently in effect.
2. Respond providing SDL with the results of this check.

Note:

The applicable to this RPC result codes are provided in section 7.16.3 Response.

Note:

The sequence diagrams describing the expected HMI behavior are provided in the section 7.14.4 Sequence Diagrams

7.16.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: The current HMI display language is provided. | JSON response | Method return | language, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | DATA_NOT_AVAILABLE: The information of the current HMI display language cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.16.3.1 Parameters

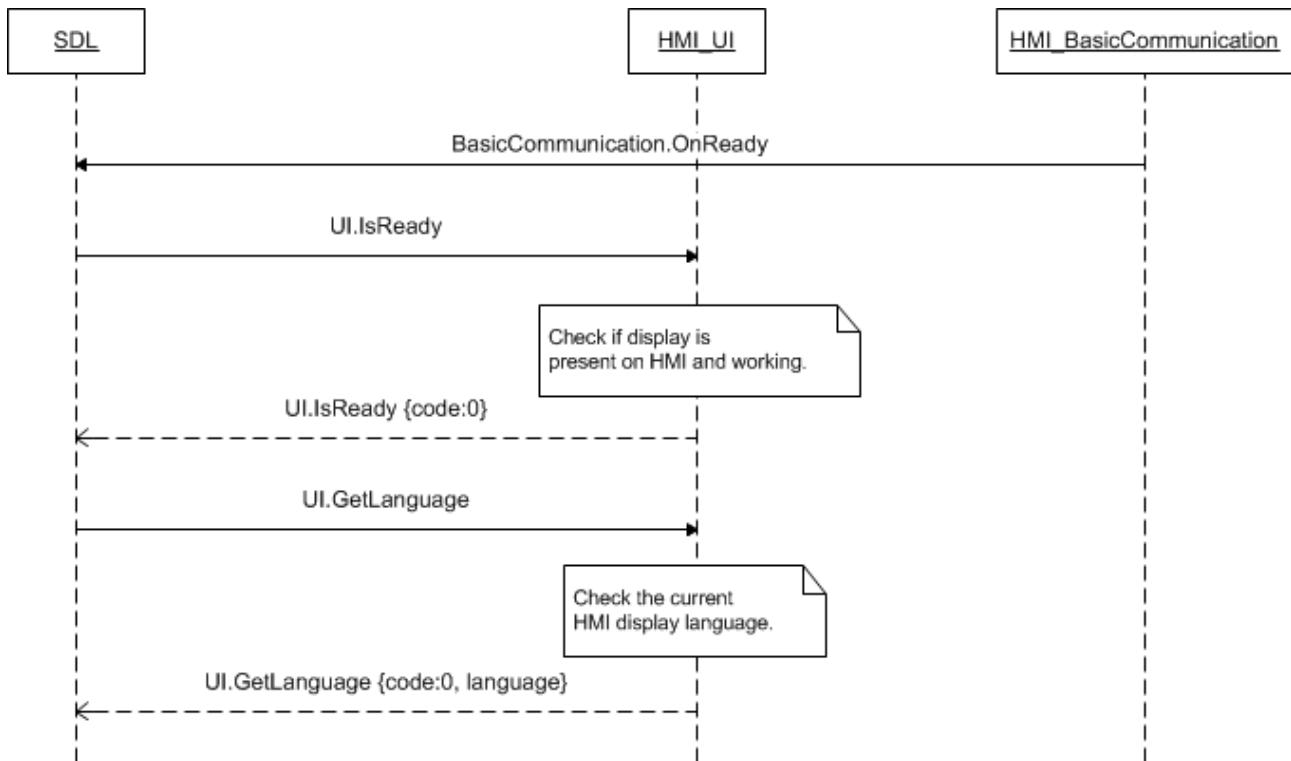
| Param Name | Type | Mandatory | Description |
|------------|-----------------|-----------|--|
| language | Common.Language | true | UI language that is currently active. See Language. |

7.15.2.3 Language

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

7.16.4 Sequence Diagrams

7.16.4.1 GetLanguage



7.16.5 JSON Messages Examples

7.16.5.1 Request

```
{
    "id" : 167,
    "jsonrpc" : "2.0",
    "method" : "UI.GetLanguage"
}
```

7.16.5.2 Response

```
{
    "id" : 167,
    "jsonrpc" : "2.0",
    "result" :
    {
        "language" : ES-ES,
        "code" : 0,
        "method" : "UI.GetLanguage"
    }
}
```

7.16.5.3 Error message

```
{
    "id" : 167,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "Invalid data",
        "data" :
        {
            "method" : "UI.GetLanguage"
        }
    }
}
```

7.16.6 D-Bus Messages Examples

7.16.6.1 Request

```
method call          //message type (request)
sender=:1.139        //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
-> dest=com.ford.sdl.hmi //the name (requested by and assigned to HMI) to receive a message
serial=68           //serial number of the message
path=/;             //the path to the object instance
interface=com.ford.sdl.hmi.UI;
member=GetLanguage
```

7.16.6.2 Response

Note:

In successful response the first returned value is always code parameter and the second one is language parameter.

```
method return        //message type (successful response)
sender=:1.140        //unique for-the-lifetime-of-the-bus name automatically assigned to HMI
-> dest=:1.139        //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
reply_serial=68      //serial number of the corresponding method_call message
    int32 0            //code (SUCCESS)
    int32 1            //language
    string ""           //message
```

7.16.6.3 Failure

```
method return        //message type (erroneous response)
sender=:1.140        //unique for-the-lifetime-of-the-bus name automatically assigned to HMI
-> dest=:1.139        //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
reply_serial=68      //serial number of the corresponding method_call message
    int32 9            //code (DATA_NOT_AVAILABLE)
    string ""           //message
```

7.17 SetAppIcon

7.17.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Display the icon together with application name. |

SDL requests to display the icon together with the specified application name (sent via OnAppRegistered or UpdateAppList to HMI) in the HMI list of registered applications. The path to the icon stored on HU is provided within request.

7.17.2 Request

7.17.2.1 Behavior

HMI must:

1. Store the provided data associating it with application appID.
2. Display the requested icon together with the application name in the HMI list of registered applications:
 - Right away if the screen with the list of registered applications is currently displayed
 - When the list of registered applications is displayed upon User's request on HMI.
3. Respond the request.

HMI may:

Display the HMI default icons for those applications that did not send the SetAppIcon request.

Note:

- The applicable to this RPC result codes are provided in section 7.17.3 Response.
- The sequence diagrams describing the expected HMI behavior are provided in the section 7.17.4 Sequence Diagrams
- The picture of "Slider" screen is added to the section 7.17.5 Possible Layout.

7.17.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|--------------|--------------|-----------|---|
| syncFileName | Common.Image | true | - The path to the image stored on HU - Or the binary image itself. See Image. |
| appID | Integer | true | ID of the application related to this RPC. |

7.17.2.3 Image

| Param Name | Type | Mandatory | Additional | Description |
|------------|------------------|-----------|-------------------|---|
| value | String | true | maxlength = 65535 | - The path to the dynamic image stored on HU - Or the static binary image itself |
| imageType | Common.ImageType | true | - | Describes, whether it is a static or dynamic image. |

7.17.2.4 ImageType

| Element name | Value | Short Description |
|--------------|-------|---|
| STATIC | 0 | Static image. The image that is sent as the binary or hex code within the request. |
| DYNAMIC | 1 | Dynamic image. The image that is stored on HMI and just a link to it is further used within requests. |

7.17.3 Response

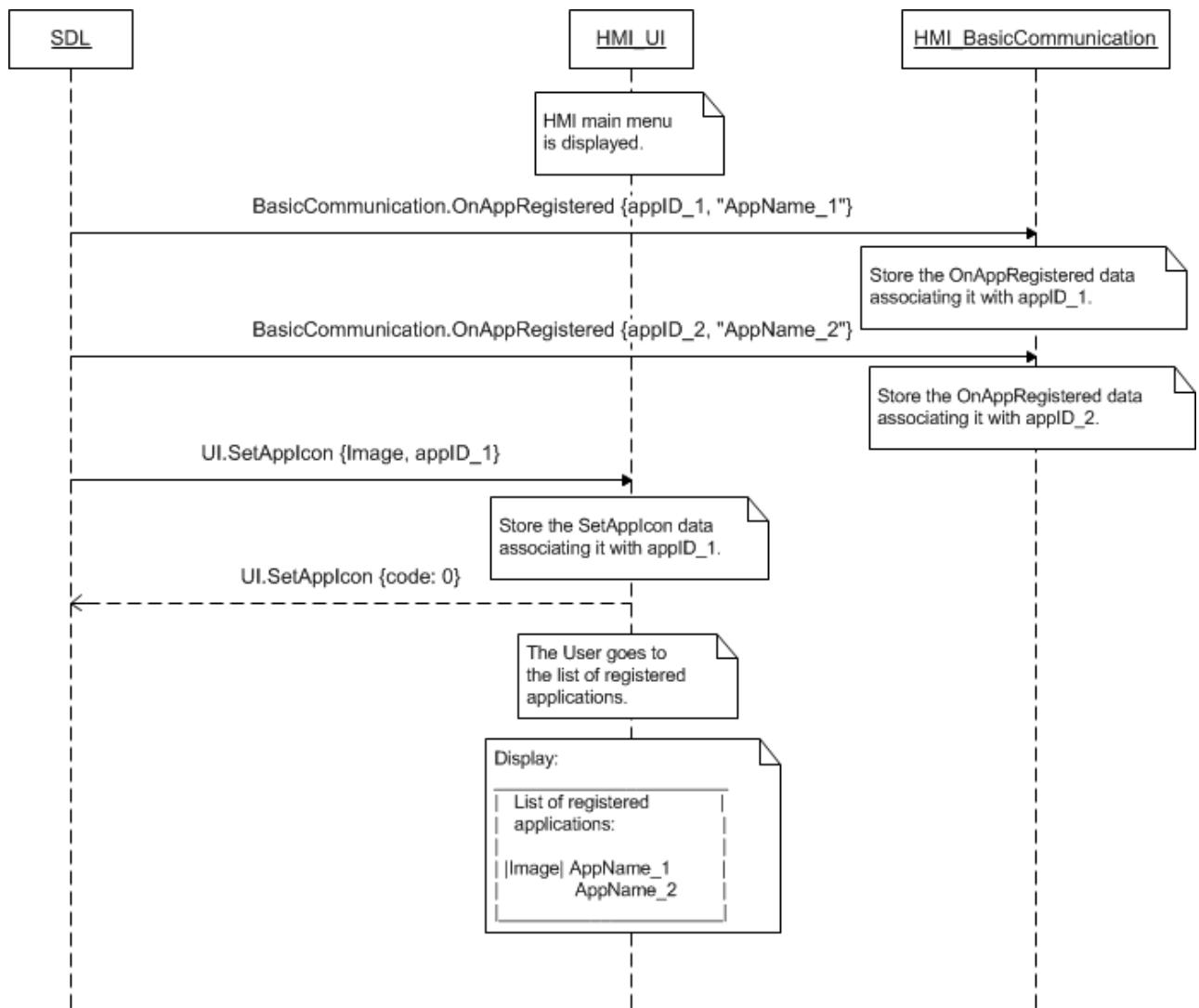
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: The requested icon is displayed for the named application in the HMI list of registered applications. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

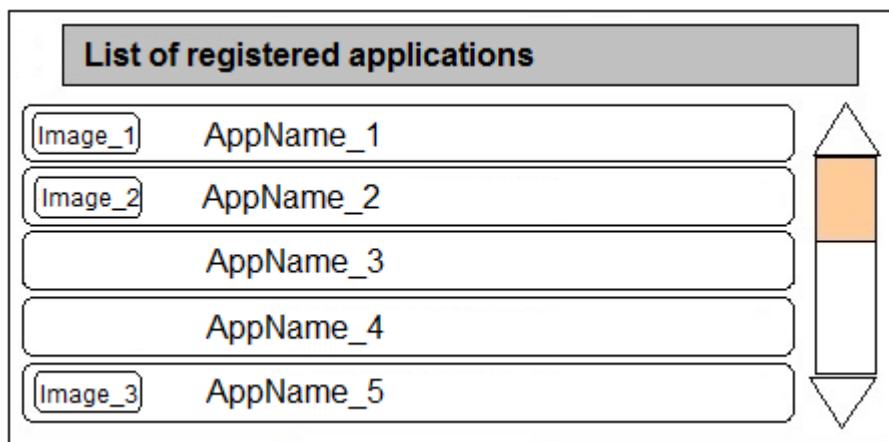
7.17.4 Sequence Diagrams

7.17.4.1 SetAppIcon



7.17.5 Possible Layout

7.17.5.1 The list of registered applications with and without icons



7.17.6 JSON Messages Examples

7.17.6.1 Request

```
{  
    "id" : 88,  
    "jsonrpc" : "2.0",  
    "method" : "UI.SetAppIcon",  
    "params" :  
    {  
        "syncFileName" :  
        {  
            "value" : "tmp/SDL/app/Best_Media/12345.jpg",  
            "imageType" : DYNAMIC  
        },  
        "appID" : 65146  
    }  
}
```

7.17.6.2 Response

```
{  
    "id" : 88,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "UI.SetAppIcon"  
    }  
}
```

7.17.6.3 Error message

```
{  
    "id" : 88,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 2,  
        "message" : "Unsupported resource",  
        "data" :  
        {  
            "method" : "UI.SetAppIcon"  
        }  
    }  
}
```

7.17.7 D-Bus Messages Examples

7.17.7.1 Request

```
method call          //message type (request)  
sender=:1.111         //unique for-the-lifetime-of-the-bus name automatically assigned to SDL  
-> dest=com.ford.sdl.hmi //the name (requested by and assigned to HMI) to receive a message  
serial=84             //serial number of the message  
path=/;               //the path to the object instance  
interface=com.ford.sdl.hmi.UI;  
member=SetAppIcon  
    struct  
    {  
        string "tmp/SDL/app/Best_Media/12345.png"      //value  
        int32 1           //imageType  
    }  
    int32 65537       //appID
```

7.17.7.2 Response

```
method return          //message type (successful response)
sender=:1.112          //unique for-the-lifetime-of-the-bus name automatically assigned to HMI
-> dest=:1.111          //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
reply_serial=84        //serial number of the corresponding method_call message
    int32 0              //code (SUCCESS)
    string ""             //message
```

7.17.7.3 Failure

```
method return          //message type (erroneous response)
sender=:1.112          //unique for-the-lifetime-of-the-bus name automatically assigned to HMI
-> dest=:1.111          //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
reply_serial=84        //serial number of the corresponding method_call message
    int32 11             //code (INVALID_DATA)
    string "The data sent is invalid"           //message
```

7.18 Slider

7.18.1 Description

| | |
|----------|-------------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Display the user-controlled slider. |

SDL requests to display the user-controlled full screen or popup slider (see the picture in section 7.18.5 Possible Layout).

The request arrives only for the application currently active on HMI.

7.18.2 Request

7.18.2.1 Behavior

HMI must:

1. Display the user-controlled slider (see the picture in section 7.18.5 Possible Layout) with:
 - The number of ticks defined by `nnumTicks` parameter
 - The initial position of the slider control corresponding to `position` parameter
 - The header defined by `sliderHeader` parameter
 - The footer defined by `sliderFooter` parameter:
 - Static – a single text string that must be displayed throughout the positions chosen.
 - Dynamic – an array of strings that must be assigned for every position and displayed when position is set by the User.
2. Keep displaying slider until one of the following occurs:
 - Timeout defined by `timeout` parameter
 - The slider is aborted by the User or any RPC of the higher priority
 - The User presses HMI-defined 'OK' button.
3. Respond the request (providing `position` parameter in case the slider is closed by 'OK' button press or aborted, see section 7.18.3 Response).

HMI may:

Display the HMI-defined 'Close'/'Back' button by the press of which the slider popup/screen must be closed (HMI must not provide any notifications to SDL upon such button press).

Note:

- The applicable to this RPC result codes are provided in section 7.18.3 Response.
- The sequence diagrams describing the expected HMI behavior are provided in the section 7.18.4 Sequence Diagrams

- The picture of “Slider” screen is added to the section 7.18.5 Possible Layout.

7.18.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|--------------|---------|-----------|--|--|
| numTicks | Integer | true | minvalue = 2 maxvalue = 26 | Number of selectable items on a horizontal axis to be displayed. |
| position | Integer | true | minvalue = 1 maxvalue = 26 | Initial position of the slider control. |
| sliderHeader | String | true | maxlength = 500 | Text header of the slider to be displayed. |
| sliderFooter | String | false | Array = true maxlength = 500 minsize = 1 maxsize = 26 | <p>Text footer to be displayed.</p> <p>For the <i>static</i> text footer, only one footer string is provided in the array.</p> <p>For a <i>dynamic</i> text footer:</p> <ul style="list-style-type: none"> - The number of footer text strings in the array always matches the numTicks value. - Text array string should correlate with potential slider position index. <p>If omitted, no footer text must be displayed.</p> |
| timeout | Integer | true | minvalue = 1000 maxvalue = 65535 | <p>The timeout in milliseconds.</p> <p>It is the amount of time the HMI must wait for User's actions while displaying the slider.</p> <p>When the amount of time defined has elapsed, the HMI must close the dialog with slider and return to the screen previously displayed.</p> |
| appID | Integer | true | - | ID of the application related to this RPC. |

7.18.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI displayed the slider with requested parameters and has just closed it by 'OK' button press. The last slider position must be returned. | JSON response | Method return | sliderPosition, code: 0 | |
| Failure | ABORTED: HMI has displayed the slider with requested parameters and has just closed it by: 1) HMI-defined 'Return'/'Back' button press. 2) A higher priority RPC. The last slider position must be returned. | JSON error message | Method return | sliderPosition, code: 5 | <p>Applicable for this RPC result codes.</p> <p>Please see Result Enumeration for all SDL-supported codes.</p> |

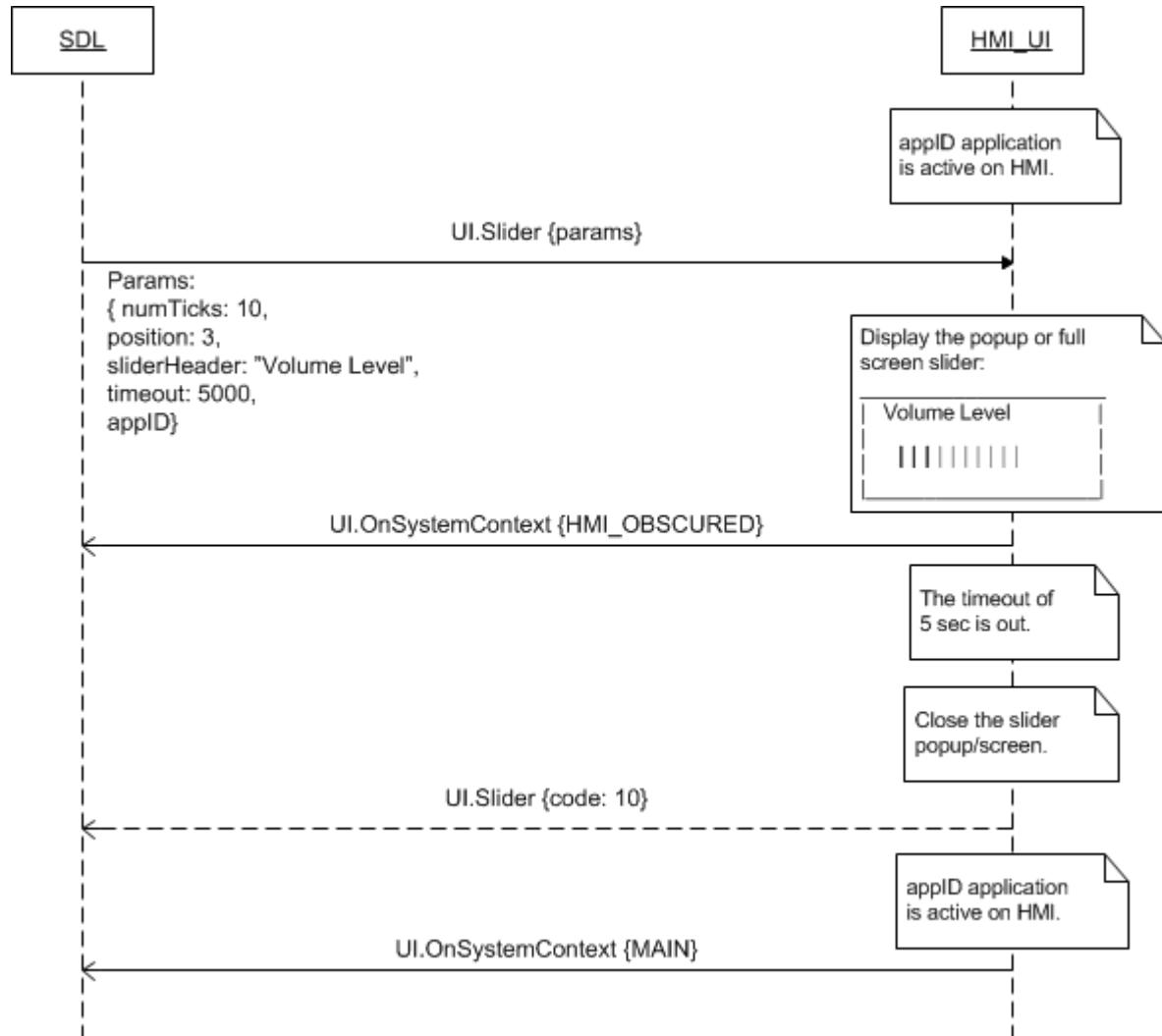
| | | | | |
|--|---|--|----------|--|
| | <p>TIMED_OUT: HMI displayed the slider with requested parameters and has just closed it by timeout.</p> | | code: 10 | |
| | <p>INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type)</p> | | code: 11 | |
| | <p>GENERIC_ERROR: The unknown issue occurred or other codes are not applicable.</p> | | code: 22 | |

7.18.3.1 Parameters

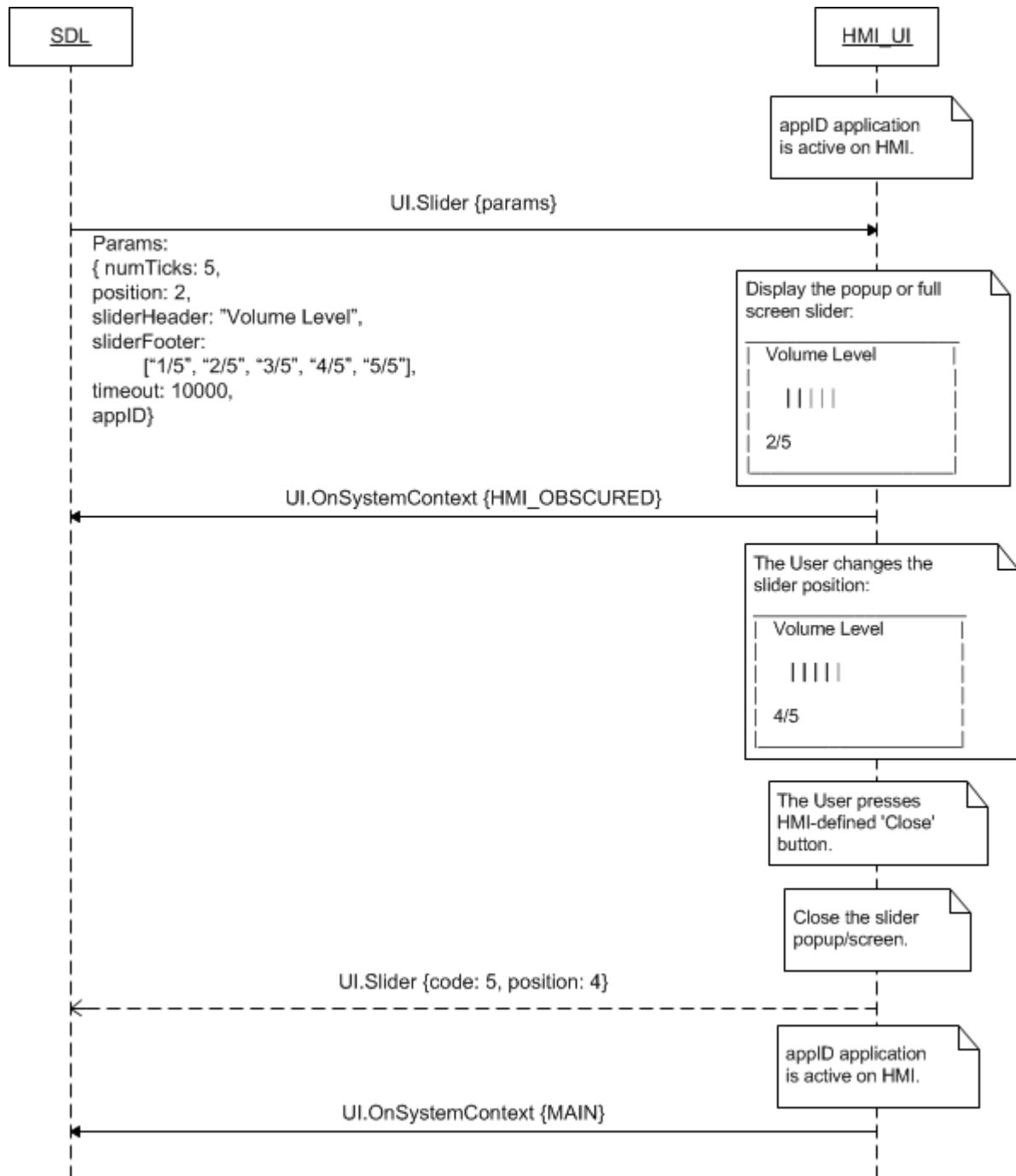
| Param Name | Type | Mandatory | Additional | Description |
|----------------|---------|-----------|-------------------------------|--|
| sliderPosition | Integer | true | minvalue = 1 maxvalue = 26 | Current slider value to be returned when the slider is closed by the timeout or aborted. |

7.18.4 Sequence Diagrams

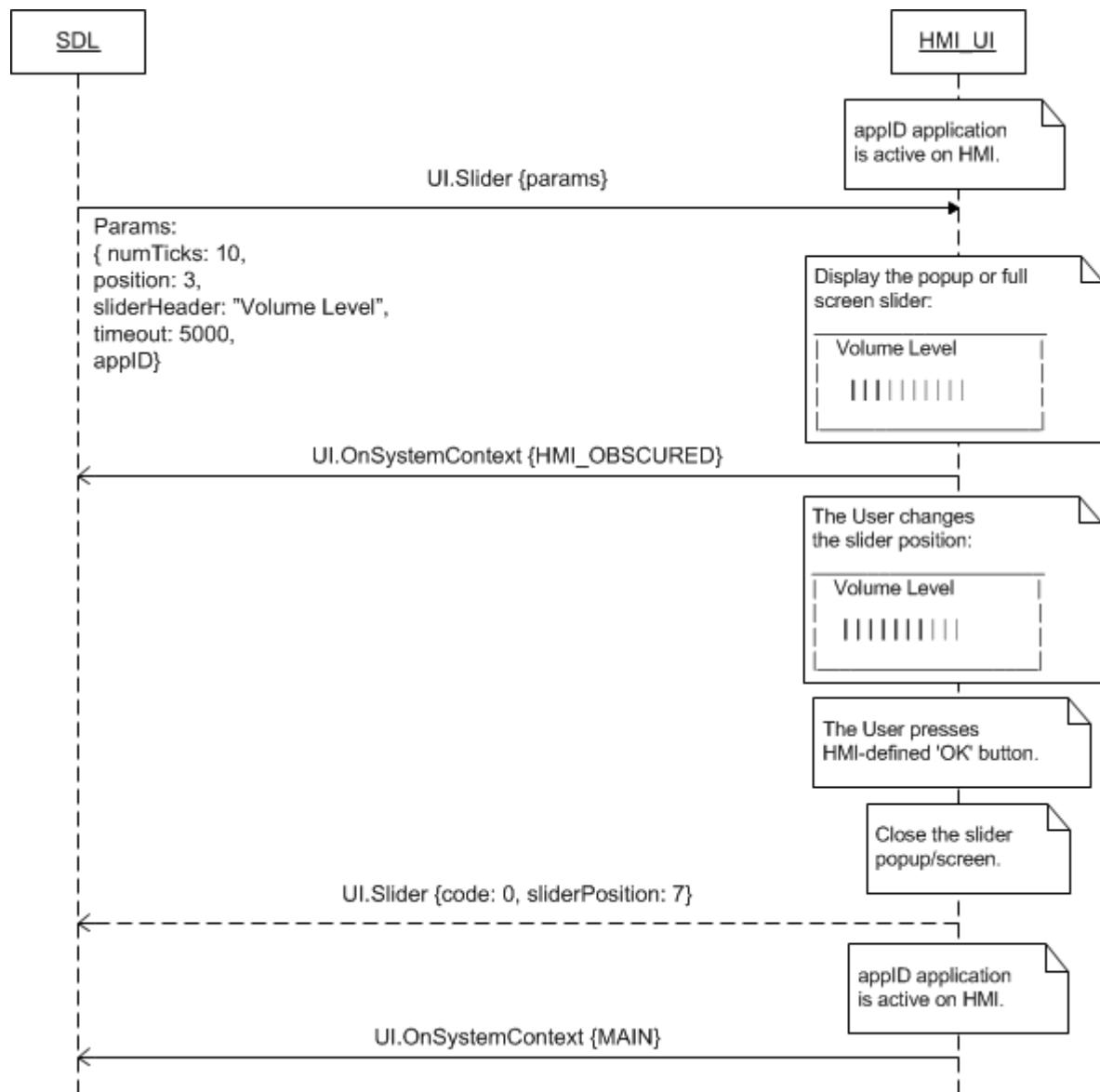
7.18.4.1 Slider with static footer displayed then closed by the timeout



7.18.4.2 Slider with dynamic footer displayed then aborted

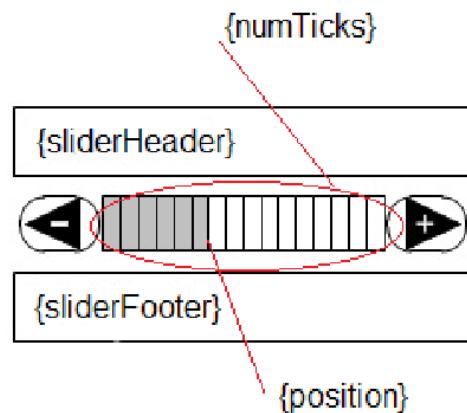


7.18.4.2 Slider with static footer displayed then closed by 'OK' button press

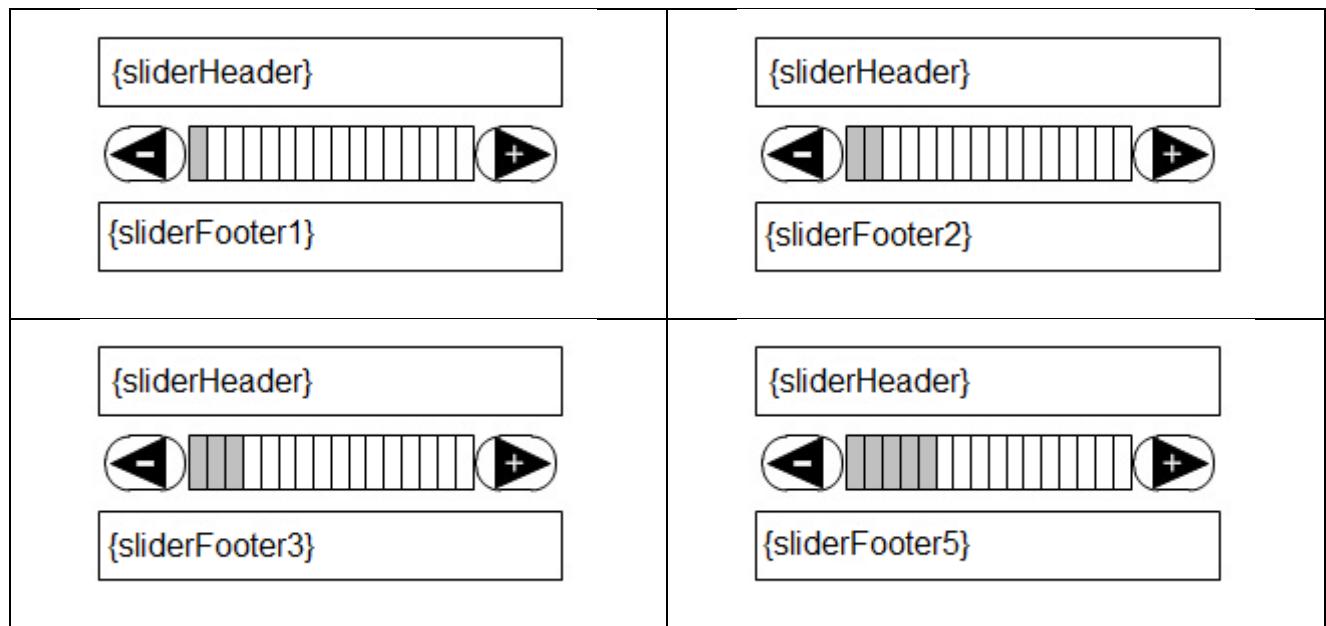


7.18.5 Possible Layout

7.18.5.1 Slider



7.18.5.2 Slider with dynamic footer



7.18.6 JSON Messages Examples

7.18.6.1 Request

```
{  
    "id" : 133,  
    "jsonrpc" : "2.0",  
    "method" : "UI.SSlider",  
    "params" :  
    {  
        "numTicks" : 5,  
        "position" : 2,  
        "sliderHeader" : "Volume Level",  
        "sliderFooter" : [ "1/5", "2/5", "3/5", "4/5", "5/5" ],  
        "timeout" : 10000,  
        "appID" : 4328  
    }  
}
```

7.18.6.2 Response

```
{  
    "id" : 133,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "sliderPosition" : 4,  
        "code" : 0,  
        "method" : "UI.SSlider"  
    }  
}
```

7.18.6.3 Error message

```
{  
    "id" : 133,  
    "jsonrpc" : "2.0",  
    "error" :  
    {
```

```
{
  "code" : 5,
  "message" : "A command was aborted due to user interaction",
  "data" :
  {
    "sliderPosition" : 5
    "method" : "UI.SLider"
  }
}
```

```
{
  "id" : 133,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : 13,
    "message" : "One of the provided IDs is not valid",
    "data" :
    {
      "method" : "UI.SLider"
    }
  }
}
```

7.18.7 D-Bus Messages Examples

7.18.7.1 Request

```
method call          //message type (request)
sender=:1.111        //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
-> dest=com.ford.sdl.hmi //the name (requested by and assigned to HMI) to receive a message
serial=90           //serial number of the message
path=/;             //the path to the object instance
interface=com.ford.sdl.hmi.UI;
member=Slider
  int32 24          //numTicks
  int32 1            //position
  string "Slider Header" //sliderHeader
  struct           //sliderFooter array
  {
    boolean true     //IsPresent (d-bus indicator)
    array
    [
      string "Slider Footer"
    ]
  }
  int32 5000         //timeout
  int32 65537        //appID
```

7.18.7.2 Response

```
method return
sender=:1.112
-> dest=:1.111
reply_serial=90
  int32 0          //code (SUCCESS)
  string ""        //message
  struct
  {
    boolean true   //IsPresent (d-bus indicator: value for param is provided)
    int32 19       //position
  }
```

7.18.7.3 Failure

```
method return
sender=:1.112
-> dest=:1.111
reply_serial=90
    int32 11      //code (INVALID_DATA)
    string "The data sent is invalid"      //message
    struct
    {
        boolean false   //IsPresent (d-bus indicator: value of the param is NOT present)
        int32 -1      //'trash'
    }
```

7.19 ScrollableMessage

7.19.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Display the dialog that may contain new lines and tabs. |

SDL requests to display a popup or full screen overlay with the text that might be formatted with the tabs and new lines.

The request may arrive in both cases of active and background application on HMI.

7.19.2 Request

7.19.2.1 Behavior

HMI must:

1. Display the popup or full screen overlay of formatted text together with up to 8 soft buttons (or without any, depending on parameters received).

Important Note:

HMI must provide OnButtonPress/OnButtonEvent notifications for every soft button defined within request.

2. Keep displaying the message until:

- The timeout defined by the timeout parameter

Important Note:

HMI must renew the timeout informing SDL about this event via UI.OnResetTimeout notification after

- a) Every User's action over the message (e.g. the User scrolls the message)
 - b) KEEP_CONTEXT soft button (defined within this RPC) press.
- One of the below occurs:
 - HMI-defined 'Close' button press
 - ClosePopup RPC from SDL
 - RPC or system event of a higher priority
 - Soft button of DEFAULT_ACTION or STEL_FOCUS press (if the soft button of such system action is defined within ScrollableMessage request)

3. Respond the request.

Note:

- The applicable to this RPC result codes are provided in section 7.18.3 Response.
- The sequence diagrams describing the expected HMI behavior are provided in the section 7.18.4 Sequence Diagrams

- The picture of scrollable message screen is added to the section 7.18.5 Possible Layout.

7.19.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-------------|------------------------|-----------|--|--|
| messageText | Common.TextFieldStruct | true | - | The body of text that can include newlines and tabs and can be scrolled up. See TextFieldStruct. Uses scrollableMessageBody (see TextFieldName). |
| timeout | Integer | true | Minvalue = 1000 maxvalue = 65535 | The timeout in milliseconds defined by the application. It is the amount of time that - HMI must keep displaying the message - Must be renewed after every User's action (e.g. scrolling the message) with sending UI.OnResetTimeout notification to SDL. When the defined amount of time has elapsed, HMI must - close the message popup/screen - send the successful response. |
| softButtons | Common.SoftButton | false | Array = true Minsize = 0 Maxsize = 8 | Application defined soft buttons (See SoftButton). HMI must provide OnButtonPress/OnButtonEvent notifications for every soft button defined within request. If omitted HMI may display the system-defined "Close" soft button providing the response with ABORTED Result code when it is pressed. HMI must not send any notifications to SDL for such button. |
| appID | Integer | true | - | ID of the application that concerns this RPC. |

7.19.2.3 TextFieldStruct Structure

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------------------------|-----------|-----------------|--|
| fieldName | Common.TextFileDialogName | true | - | The name of the field where the text must be displayed in. |
| fieldText | String | true | Maxlength = 500 | The text to be displayed. |

7.19.2.4 TextFieldName Enumeration

Only the text fields applicable to Alert RPC are described within this section. All the text fields names recognized by SDL are described in the section 13.1.14 TextFieldName.

| Element name | Short Description |
|-----------------------|---|
| scrollableMessageBody | The long form body of text that can include newlines and tabs. Applies to ScrollableMessage, section 7.1.17. |

7.19.3 Response

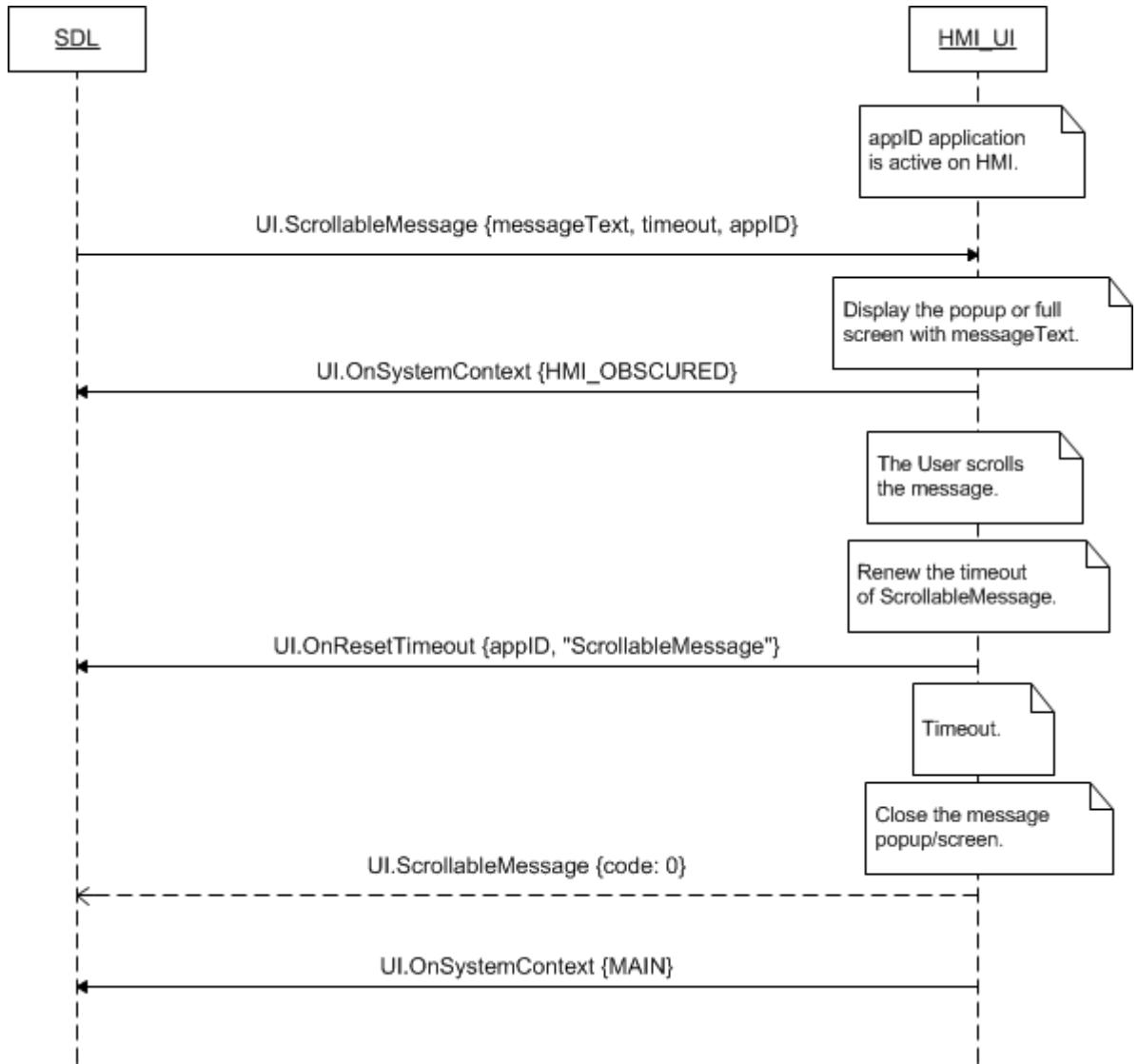
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

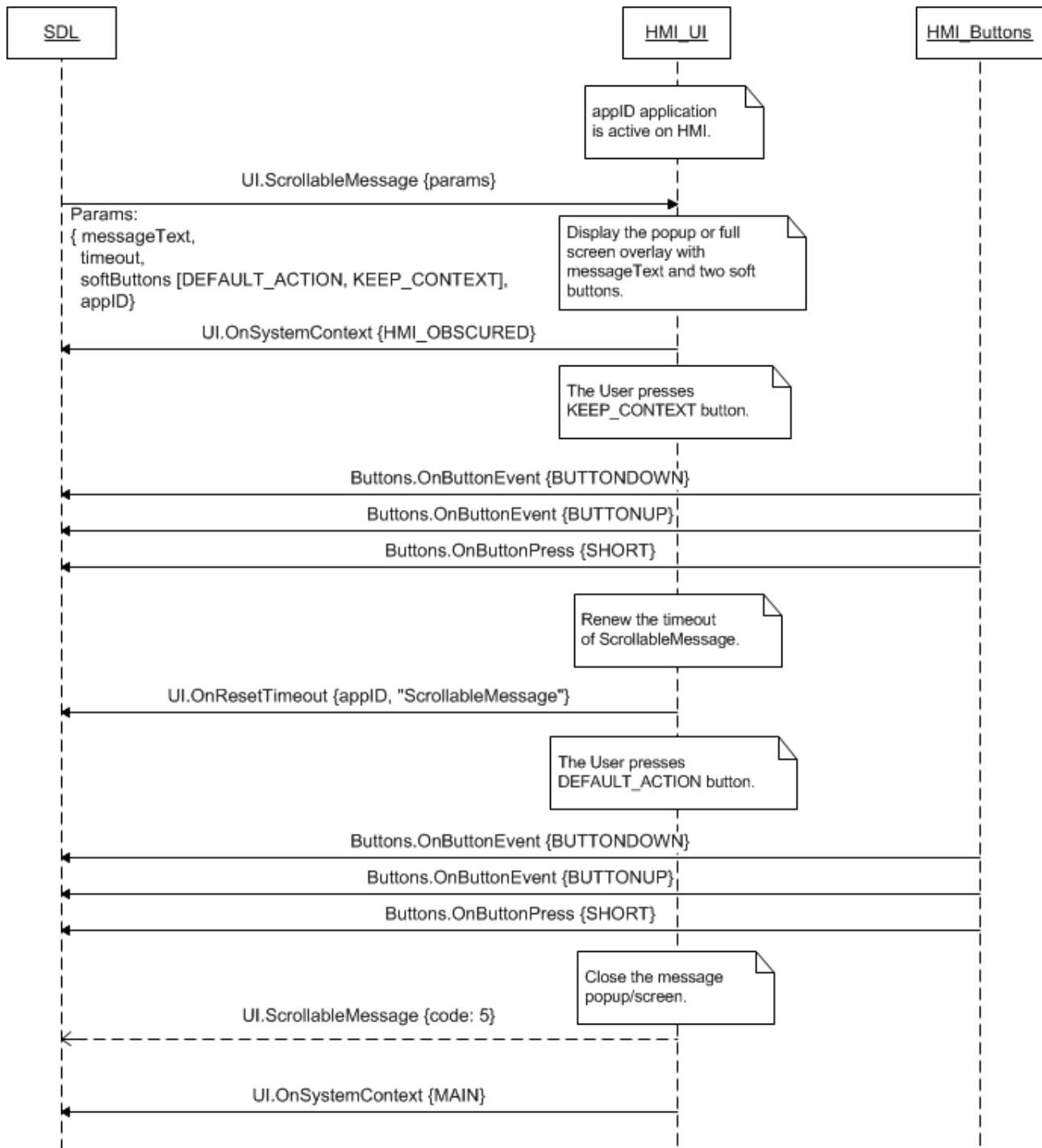
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI displayed the message with requested parameters and has just closed it by 1) The timeout. 2) STEAL_FOCUS soft button (defined within this RPC) press. | JSON response | Method return | code: 0 | |
| Failure | ABORTED: HMI displayed the message with requested parameters and has just closed it by: 1) HMI-defined 'Close'/'Back' button press. 2) A higher priority RPC or system event 3) DEFAULT_ACTION soft button (defined within this RPC) press. | JSON error message | Method return | code: 5 | Applicable for this RPC result codes. |
| | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | | | code: 11 | Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.19.4 Sequence Diagrams

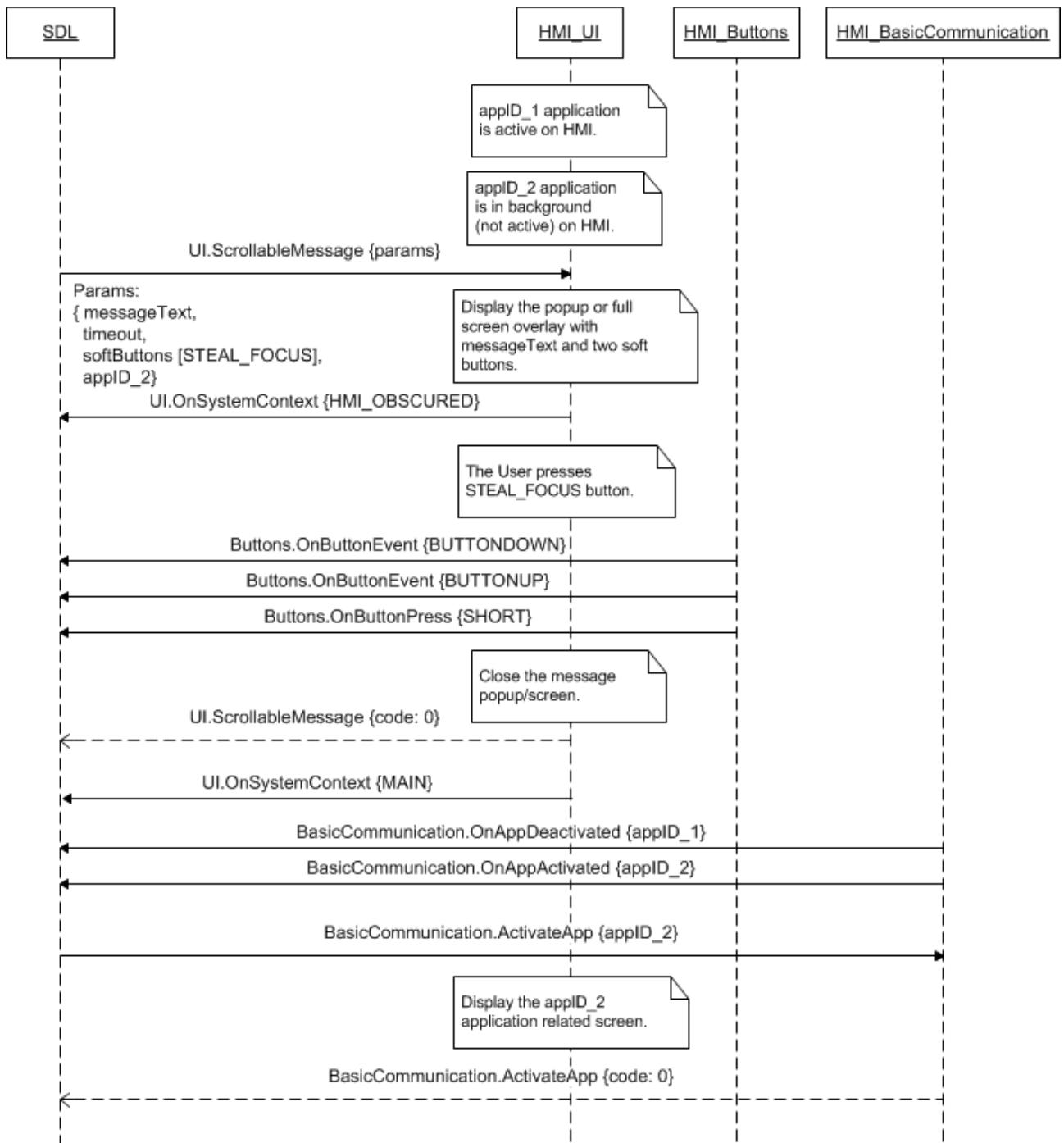
7.19.4.1 ScrollableMessage displayed, scrolled by the User and closed by the timeout



7.19.4.2 ScrollableMessage with soft buttons of *DEFAULT_ACTION* and *KEEP_CONTEXT* system action pressed by the User



7.19.4.3 ScrollableMassage with STEAL_FOCUS soft button for the application not active on HMI



7.19.5 JSON Messages Examples

7.19.5.1 Request

```
{
    "id" : 138,
    "jsonrpc" : "2.0",
    "method" : "UI.ScrollableMessage",
    "params" :
    {
        "messageText" :
        {
            "fieldName" : scrollableMessageBody,
            "fieldText" : "Create a Station
Enter an artist, song or composer in the
Search box in the top left corner. We'll
create a radio station featuring that music"
        }
    }
}
```

and more like it.

You can also create a new station from the song or artist currently playing by hovering over the album artwork, clicking the white up-arrow and selecting New Station—you can choose From Song or From Artist.”

```
{
  "timeout" : 10000,
  "softButtons" :
  [
    {
      "type" : TEXT,
      "text" : "Leave onscreen",
      "softButtonID" : 15,
      "systemAction" : KEEP_CONTEXT
    },
    {
      "type" : TEXT,
      "text" : "Cancel",
      "softButtonID" : 16,
      "systemAction" : STEAL_FOCUS
    }
  ],
  "appID" : 6527
}
```

7.19.5.2 Response

```
{
  "id" : 138,
  "jsonrpc" : "2.0",
  "result" :
  {
    "code" : 0,
    "method" : "UI.ScrollableMessage"
  }
}
```

7.19.5.3 Error message

```
{
  "id" : 138,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : 12,
    "message" : "The string data is too long",
    "data" :
    {
      "method" : "UI.ScrollableMessage"
    }
  }
}
```

7.19.6 D-Bus Messages Examples

7.19.6.1 Request

```
method call
sender=:1.138
-> dest=com.ford.sdl.hmi
serial=111
path=/;
interface=com.ford.sdl.hmi.UI;
member=ScrollableMessage

    struct                      //messageText
    {
        int32 10                //fieldName
        string "Text text text." //fieldText
    }

Text text.

Text text text.

Text."
}

int32 30000                  //timeout

struct                      //softButtons
{
    boolean true             //isPresent: the optional param is provided
    array
    [
        struct                //the first element of soft button in the array
        {
            int32 0              //type: TEXT
            struct               //text to display on the button
            {
                boolean true     //isPresent: the optional param is provided
                string "Keep"    //text
            }
            struct               //image to display on the button
            {
                boolean false   //isPresent: the optional param is not provided
                struct           //the values that must be ignored
                {
                    string ""
                    int32 -1
                }
            }
        }
        struct               //to highlight a button or not
        {
            boolean true     //isPresent: the optional param is provided
            boolean true     //isHighlighted
        }
    ]
}

int32 5500                  //softButtonID

struct                      //system action to be assigned
{
    boolean true             //isPresent: the optional param is provided
    int32 2                  //systemAction: KEEP_CONTEXT
```

```

        }

    struct          //the second element of soft button in the array
    {
        int32 0      //type: TEXT

        struct      //text to display on the button
        {
            boolean true //isPresent: the optional param is provided
            string "Close" //text
        }

        struct      //image to display on the button
        {
            boolean false //isPresent: the optional param is not provided
            struct      //the values that must be ignored
            {
                string ""
                int32 -1
            }
        }

        struct      //to highlight a button or not
        {
            boolean true //isPresent: the optional param is provided
            boolean false //isHighlighted
        }

        int32 5501      //softButtonID

        struct      //system action to be assigned
        {
            boolean true //isPresent: the optional param is provided
            int32 0      //systemAction: DEFAULT_ACTION
        }
    }
}

int32 65540      //appID

```

7.19.6.2 Response

```

method return
sender=:1.139
-> dest=:1.138
reply_serial=111
    int32 0      //code (SUCCESS)
    string ""      //message

```

7.19.6.3 Failure

```

method return
sender=:1.139
-> dest=:1.138
reply_serial=111
    int32 5      //code (ABORTED)
    string "The method was aborted" //message

```

7.20 PerformAudioPassThru

7.20.1 Description

| | |
|----------|------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Start audio capturing. |

SDL prompts HMI to start audio capturing and inform the User about the event by displaying the dialog with the requested text information.

The request may arrive in both cases of active and background application on HMI.

Note:

This RPC may arrive together with TTS.Speak which purpose is to inform the User about start of audio capturing by the means of TTS module.

7.20.2 Request

7.20.2.1 Behavior

HMI must:

1. Attenuate or make any audio stream not audible except of TTS speaking (depending on HMI capabilities).
2. Display the dialog with requested text. (audioPassThruDisplayText1 and audioPassThruDisplayText2 text fields).
3. Start the audio capturing from the microphone.

Important Note:

If PerfromAudioPassThru request is accompanied with TTS.Speak RPC, HMI must start the audio capturing only after TTS finishes speaking the requested text.

4. Keep displaying the dialog and capturing the audio until:

- The value of maxDuration is reached.
- The User presses any of HMI-defined ‘Cancel’/‘Done’/‘Retry’ buttons.
- The request of EndAudioPassThru comes from SDL.

5. Respond the request.

HMI may:

Display the HMI-defined ‘Done’, ‘Retry’, ‘Cancel’ soft buttons. HMI must not return the OnButtonEvent/OnButtonPress notifications to SDL when such soft button is pressed by the User.

Note:

- The applicable to this RPC result codes are provided in section 7.20.3 Response.
- The sequence diagrams describing the expected HMI behavior are provided in the section 7.20.4 Sequence Diagrams
- The picture of performAudioPassThru screen is added to the section 7.18.5 Possible Layout.

7.20.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|---------------------------|------------------------|-----------|--|---|
| audioPassThruDisplayTexts | Common.TextFieldStruct | true | Array = true minsize = 0 maxsize = 2 | The text to be displayed upon the start of audio capturing. See TextFieldStruct. Uses audioPassThruDisplayText1 and audioPassThruDisplayText2 (See TextFieldName) |
| maxDuration | Integer | true | Minvalue = 1 Maxvalue = 1000000 | The maximum duration of audio recording in milliseconds |

| | | | | |
|-----------|---------|------|---|---|
| muteAudio | Boolean | true | – | Defines if the current audio source should be muted during the APT session. If not, the audio source will play without interruption. If omitted, the value is set to true |
| appID | Integer | true | – | ID of the application that requested this RPC. |

7.20.2.3 TextFieldStruct Structure

| Param Name | Type | Mandatory | Additional | Description |
|------------|-----------------------|-----------|-----------------|--|
| fieldName | Common.TextFileldName | true | – | The name of the field where the text must be displayed in. |
| fieldText | String | true | Maxlength = 500 | The text to be displayed. |

7.20.2.4 TextFieldName Enumeration

Only the text fields applicable to PerformAudioPassThru RPC are described in this sub-section. All the text fields names recognized by SDL are described in the section 13.1.14 *TextFieldName*.

| Element name | Short Description |
|---------------------------|--|
| audioPassThruDisplayText1 | The first line of text that must be displayed during audio capture. |
| audioPassThruDisplayText2 | The second line of text that must be displayed during audio capture. |

7.20.3 Response

Note:

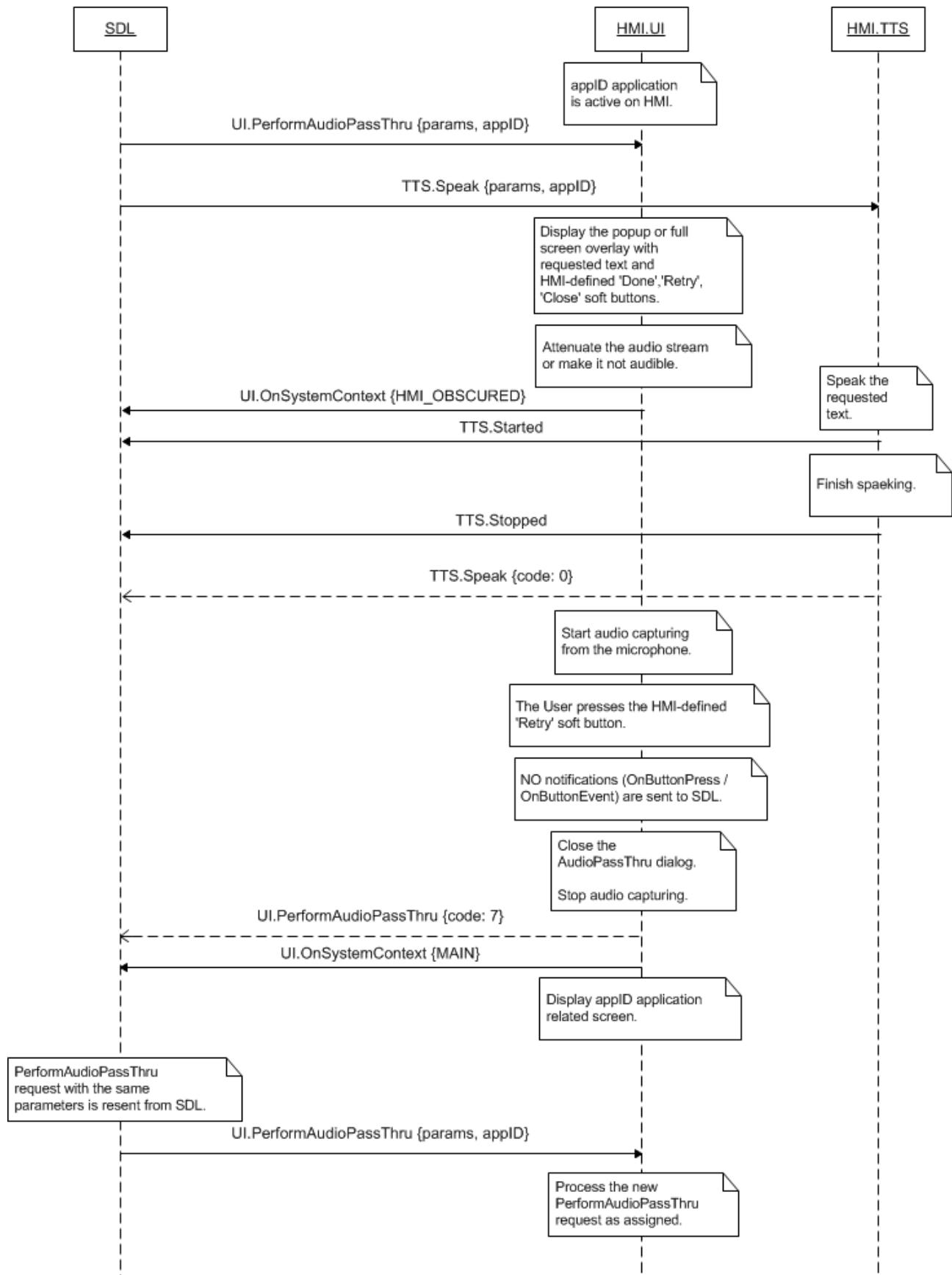
There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI displayed the PerformAudioPassTru dialog with requested parameters, has just closed it and stopped the audio capturing by: 1) The value of <code>maxDuration</code> reached 2) The HMI-defined 'Ok'/'Done' button press 3) EndAudioPassTru request from SDL. | JSON response | Method return | code: 0 | |
| Failure | ABORTED: HMI displayed the PerformAudioPassTru dialog with requested parameters, has just closed it and stopped the audio capturing by HMI-defined 'Cancel' button press. | JSON error message | Method return | code: 5 | Applicable for this RPC result codes. |
| | RETRY: HMI displayed the PerformAudioPassTru dialog with requested parameters, has just closed it and stopped the audio capturing by HMI-defined 'Retry' button press. | | | Code: 7 | Please see Result Enumeration for all SDL-supported codes. |

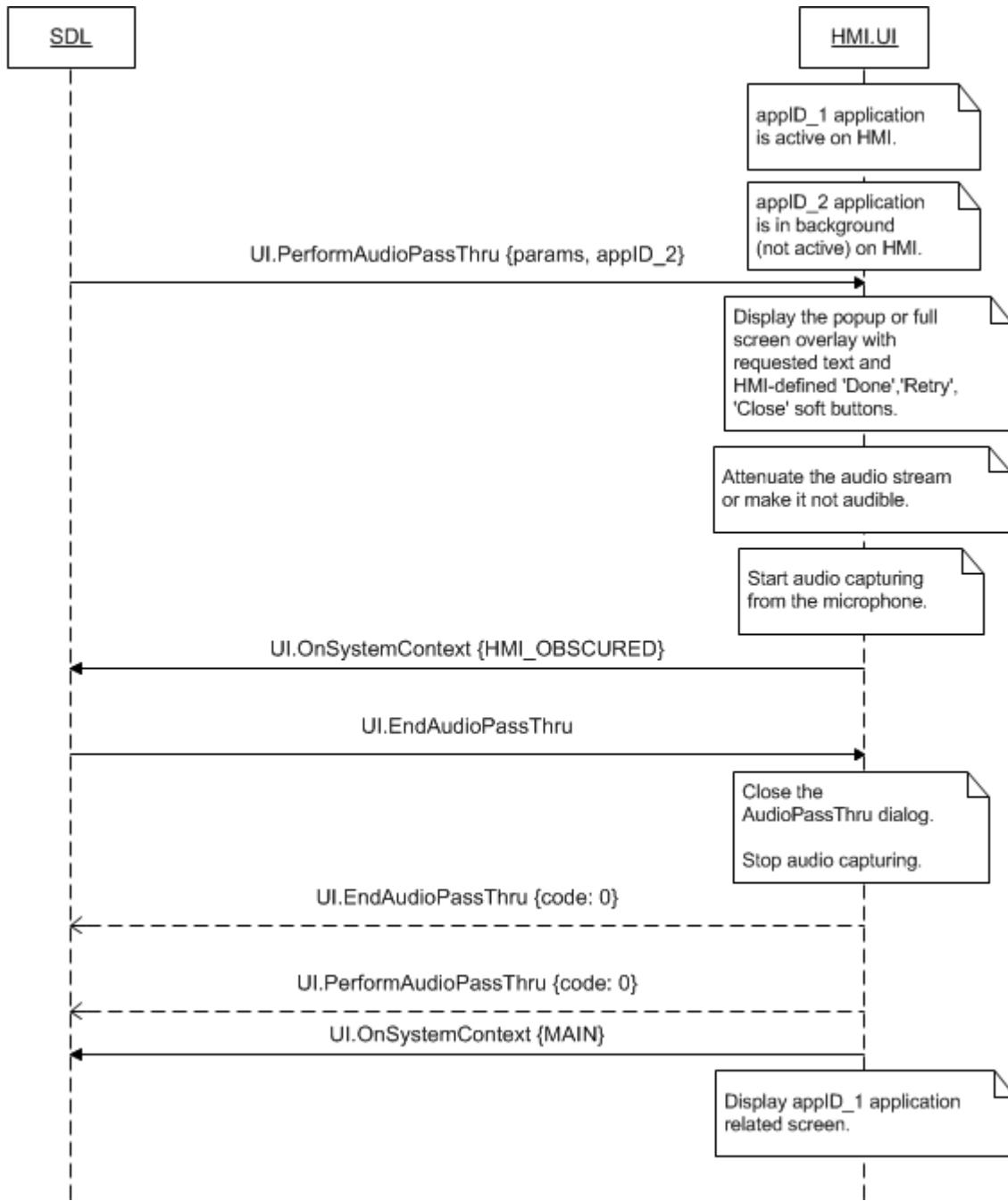
| | | | | |
|--|--|--|----------|--|
| | <p>INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type)</p> <p>GENERIC_ERROR: The unknown issue occurred or none of other SDL-supported codes are applicable.</p> | | code: 11 | |
| | | | code: 22 | |

7.20.4 Sequence Diagrams

7.20.4.1 PerformAudioPassTru requested together with TTS.Speak, processed and then finished by 'Retry' soft button press



7.20.4.2 PerformAudioPassThru for the application not active on HMI, processed and then finished by EndAudioPassThru



7.20.5 JSON Messages Examples

7.20.5.1 Request

```
{
    "id" : 79,
    "jsonrpc" : "2.0",
    "method" : "UI.PerformAudioPassThru",
    "params" :
    {
        "audioPassThruDisplayTexts" :
        {
            "fieldName" : audioPassThruDisplayText1,
            "fieldText" : "The audio capturing is in progress"
        },
        "maxDuration" : 10000,
    }
}
```

```
}
```

7.20.5.2 Response

```
{
  "id" : 79,
  "jsonrpc" : "2.0",
  "result" :
  {
    "code" : 0,
    "method" : "UI.PerformAudioPassThru"
  }
}
```

7.20.5.3 Error message

```
{
  "id" : 79,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : 7,
    "message" : "The user interrupted the RPC and indicated to start over",
    "data" :
    {
      "method" : "UI.PerformAudioPassThru"
    }
  }
}
```

7.20.6 D-Bus Messages Examples

7.20.6.1 Request

```
method call
sender=:1.186
-> dest=com.ford.sdl.hmi
serial=85
path=/;
interface=com.ford.sdl.hmi.UI;
member=PerformAudioPassThru
array                                //audioPassThruDisplayTexts
[
  struct                            //the first element
  {
    int32 17                         //fieldname: audioPassThruDisplayText1
    string "AudioPassThru Display Text 1" //fieldText
  }

  struct                            //the second element
  {
    int32 18                         //fieldname: audioPassThruDisplayText2
    string "AudioPassThru Display Text 2" //fieldText
  }
]

int32 10000                         //maxDuration
int32 65537                          //appID
```

7.20.6.2 Response

```
method return
```

```

sender=:1.187
-> dest=:1.186
reply_serial=85
    int32 0
    string ""

```

7.20.6.3 Failure

```

method return
sender=:1.187
-> dest=:1.186
reply_serial=85
    int32 7
    string "The User interrupted the method call and indicated to start over"

```

7.21 EndAudioPassThru

7.21.1 Description

| | |
|----------|-----------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Stop audio capturing. |

SDL prompts HMI to stop audio capturing and close the AudioPassThru dialog on UI.

7.21.2 Request

7.21.2.1 Behavior

HMI must:

1. Stop audio capturing started by PerformAudioPassThru RPC.
2. Close the PerformAudioPassThru dialog.
3. Respond the request.

Note:

HMI must first respond the EndAudioPassThru request, and then provide the response to the corresponding PerformAudioPassThru request.

Note:

- The applicable to this RPC result codes are provided in section 7.21.3 Response.
- The sequence diagrams describing the expected HMI behavior are provided in the section 7.21.4 Sequence Diagrams

7.21.3 Response

Note:

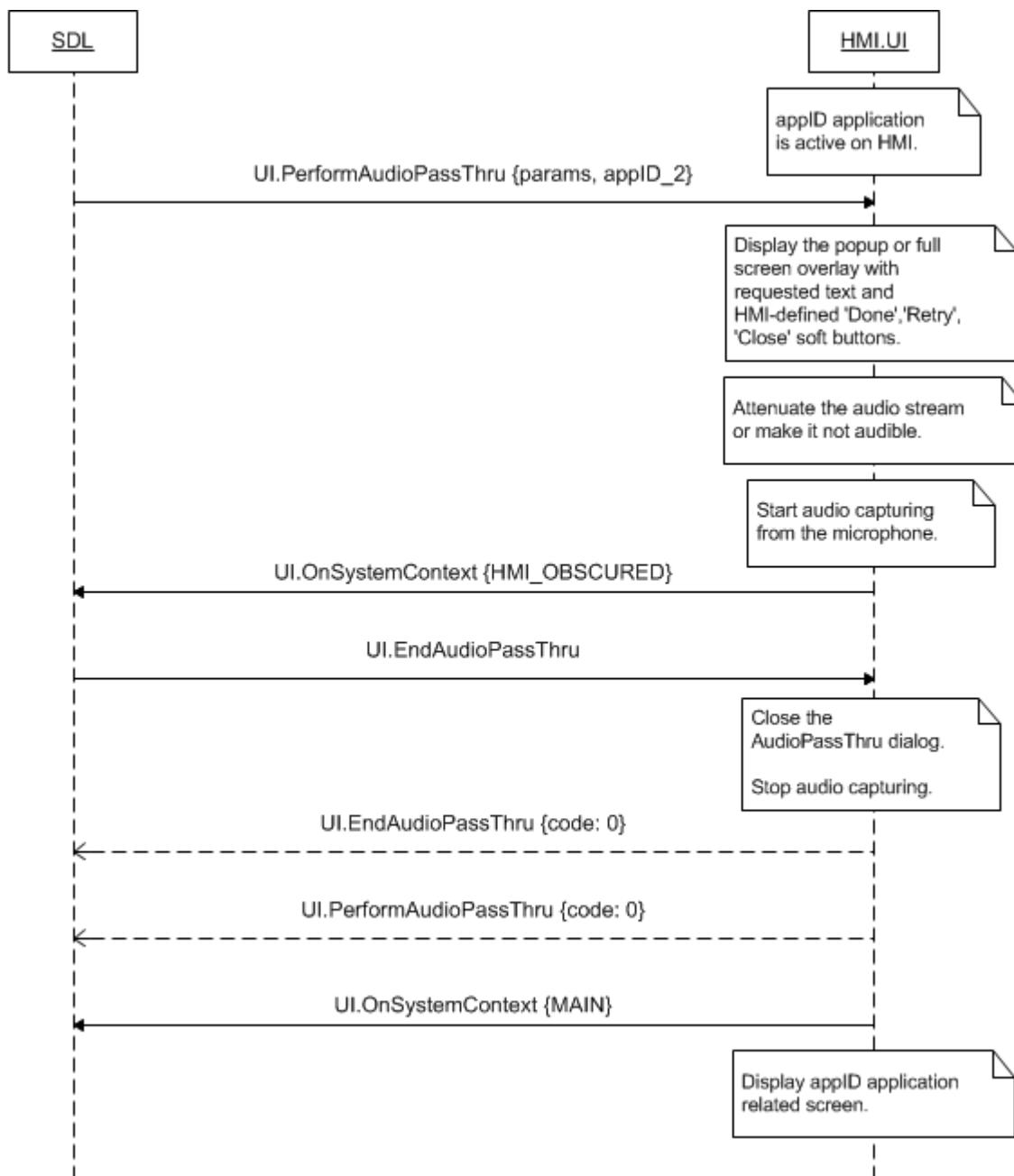
There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI has stopped audio capturing upon this request receipt. | JSON response | Method return | code: 0 | |
| Failure | REJECTED: PerformAudioPassThru has already finished by the time this request arrived. | JSON error message | Method return | code: 4 | Applicable for this RPC result codes. Please see |

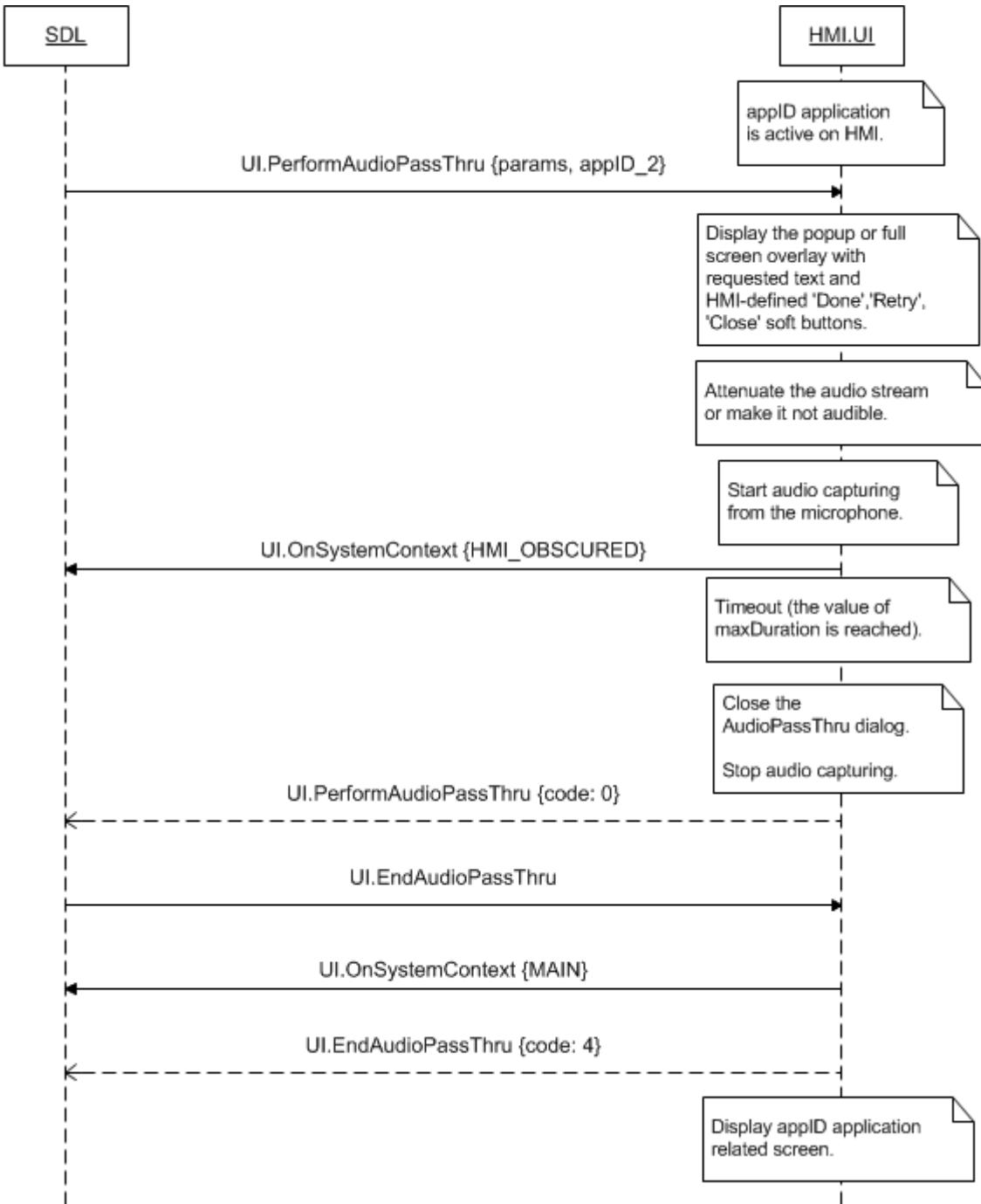
| | | | | |
|--|--|--|----------|--|
| | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | | code: 11 | Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | code: 22 | |

7.21.4 Sequence Diagrams

7.21.4.1 EndAudioPassThru that stops the audio capturing



7.21.4.2 EndAudioPassThru for the case when audio capturing is already over



7.21.5 JSON Messages Examples

7.21.5.1 Request

```
{
  "id" : 79,
  "jsonrpc" : "2.0",
  "method" : "UI.EndAudioPassThru",
}
```

7.21.5.2 Response

```
{
  "id" : 79,
  "jsonrpc" : "2.0",
  "result" :
```

```
{
  "code" : 0,
  "method" : "UI.EndAudioPassThru"
}
```

7.21.5.3 Error message

```
{
  "id" : 79,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : 4,
    "message" : "Rejected: no PerformAudioPassThru is now active",
    "data" :
    {
      "method" : "UI.EndAudioPassThru"
    }
  }
}
```

7.21.6 D-Bus Messages Examples

7.21.6.1 Request

```
method call          //message type (request)
sender=:1.186        //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
-> dest=com.ford.sdl.hmi //the name (requested by and assigned to HMI) to receive a message
serial=89           //serial number of the message
path=/;             //the path to the object instance
interface=com.ford.sdl.hmi.UI;
member=EndAudioPassThru
```

7.21.6.2 Response

```
method return
sender=:1.187
-> dest=:1.186
reply_serial=89
  int32 0          //code (SUCCESS)
  string ""         //message
```

7.21.6.3 Failure

```
method return
sender=:1.187
-> dest=:1.186
reply_serial=89
  int32 4          //code (REJECTED)
  string " Rejected: no PerformAudioPassThru is now active "           //message
```

7.22 ClosePopUp

7.22.1 Description

| | |
|-----------------|--------------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Close the popup currently displayed. |

SDL prompts HMI to close the application-related dialog (popup/screen) currently displayed on UI.

7.22.2 Request

7.22.2.1 Behavior

HMI must:

1. Close the application-related dialog (popup/screen) called by the `methodName` method and currently displayed on UI.
2. Respond the request.

Note:

HMI must first reply the `ClosePopUp` method, and then reply the method having been interrupted.

Note:

- The applicable to this RPC result codes are provided in section 7.21.3 Response.
- The sequence diagrams describing the expected HMI behavior are provided in the section 7.21.4 Sequence Diagrams

7.22.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|------------|--------|-----------|--------------------------|
| methodName | String | false | Method that to be closed |

7.22.3 Response

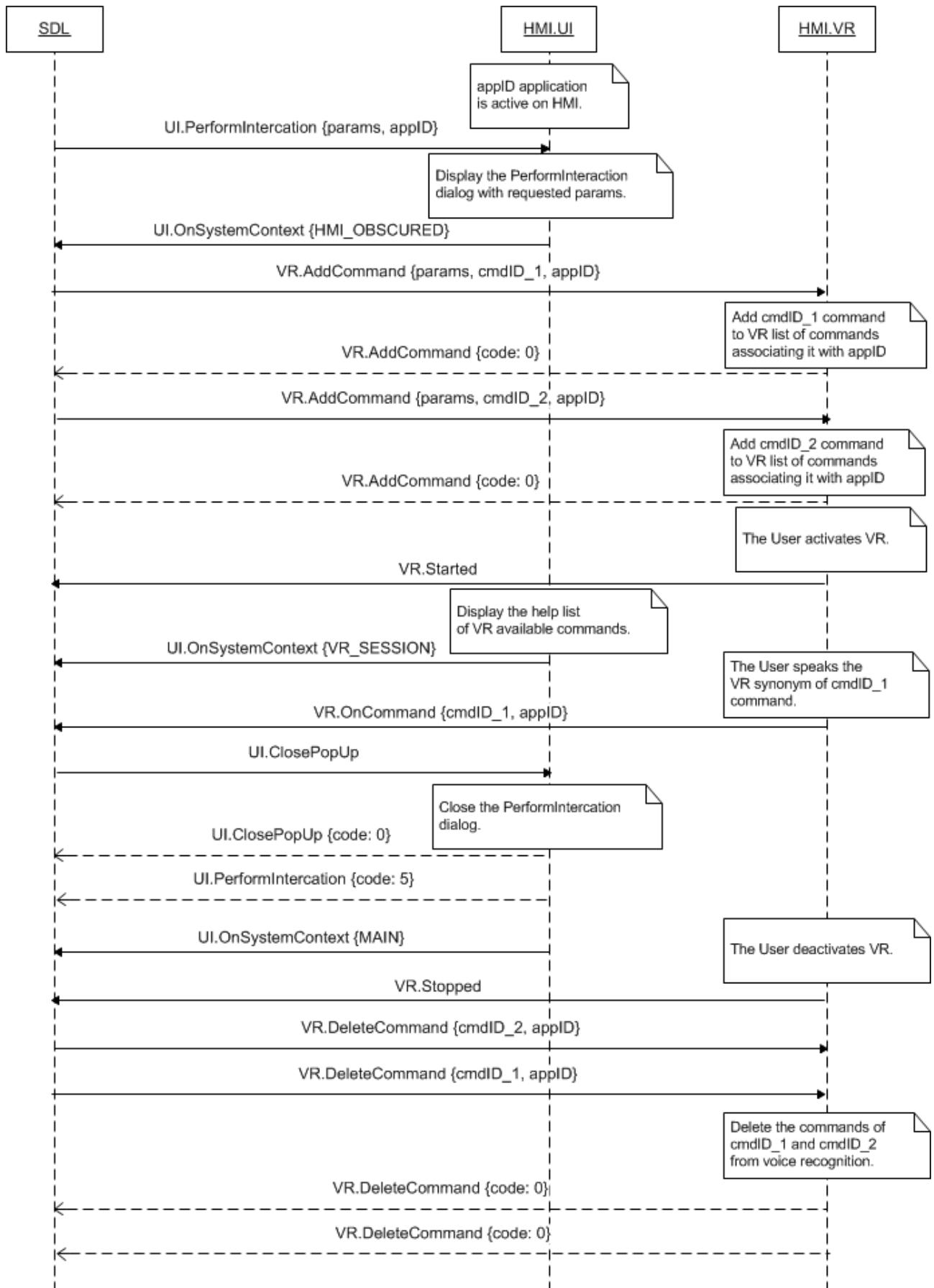
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI has closed the dialog (popup/screen) displayed on UI at the time the request arrived. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.22.4 Sequence Diagrams

7.22.4.1 ClosePopUp for UI.PerformInterception



7.22.5 JSON Messages Examples

7.22.5.1 Request

```
{  
    "id" : 79,  
    "jsonrpc" : "2.0",  
    "method" : "UI.ClosePopUp",  
}
```

7.22.5.2 Response

```
{  
    "id" : 79,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "UI.ClosePopUp"  
    }  
}
```

7.22.5.3 Error message

```
{  
    "id" : 79,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 22,  
        "message" : "During API call an unknown error has occurred",  
        "data" :  
        {  
            "method" : "UI.ClosePopUp"  
        }  
    }  
}
```

7.22.6 D-Bus Messages Examples

7.22.6.1 Request

```
method call          //message type (request)  
sender=:1.186        //unique for-the-lifetime-of-the-bus name automatically assigned to SDL  
-> dest=com.ford.sdl.hmi //the name (requested by and assigned to HMI) to receive a message  
serial=109          //serial number of the message  
path=/;             //the path to the object instance  
interface=com.ford.sdl.hmi.UI;  
member=ClosePopUp
```

7.22.6.2 Response

```
method return  
sender=:1.187  
-> dest=:1.186  
reply_serial=89  
    int32 0           //code (SUCCESS)  
    string ""         //message
```

7.22.6.3 Failure

```
method return  
sender=:1.187  
-> dest=:1.186  
reply_serial=89
```

| | |
|---|-----------|
| <pre>int32 22 //code (GENERIC_ERROR) string "During method call an unknown error has occurred"</pre> | //message |
|---|-----------|

7.23 OnCommand

7.23.1 Description

| | |
|-----------------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the command chosen by the User from UI. |

Initially SDL adds the commands to the in-application menu (see `UI.AddCommand`) or sub-menu (see `UI.AddSubMenu`) on UI. The `UI.OnCommand` notification is required by SDL to know the SDL-defined command has been chosen by the User.

HMI must:

Send `UI.OnCommand` notification upon User's choosing the SDL-defined command on UI providing the values of:

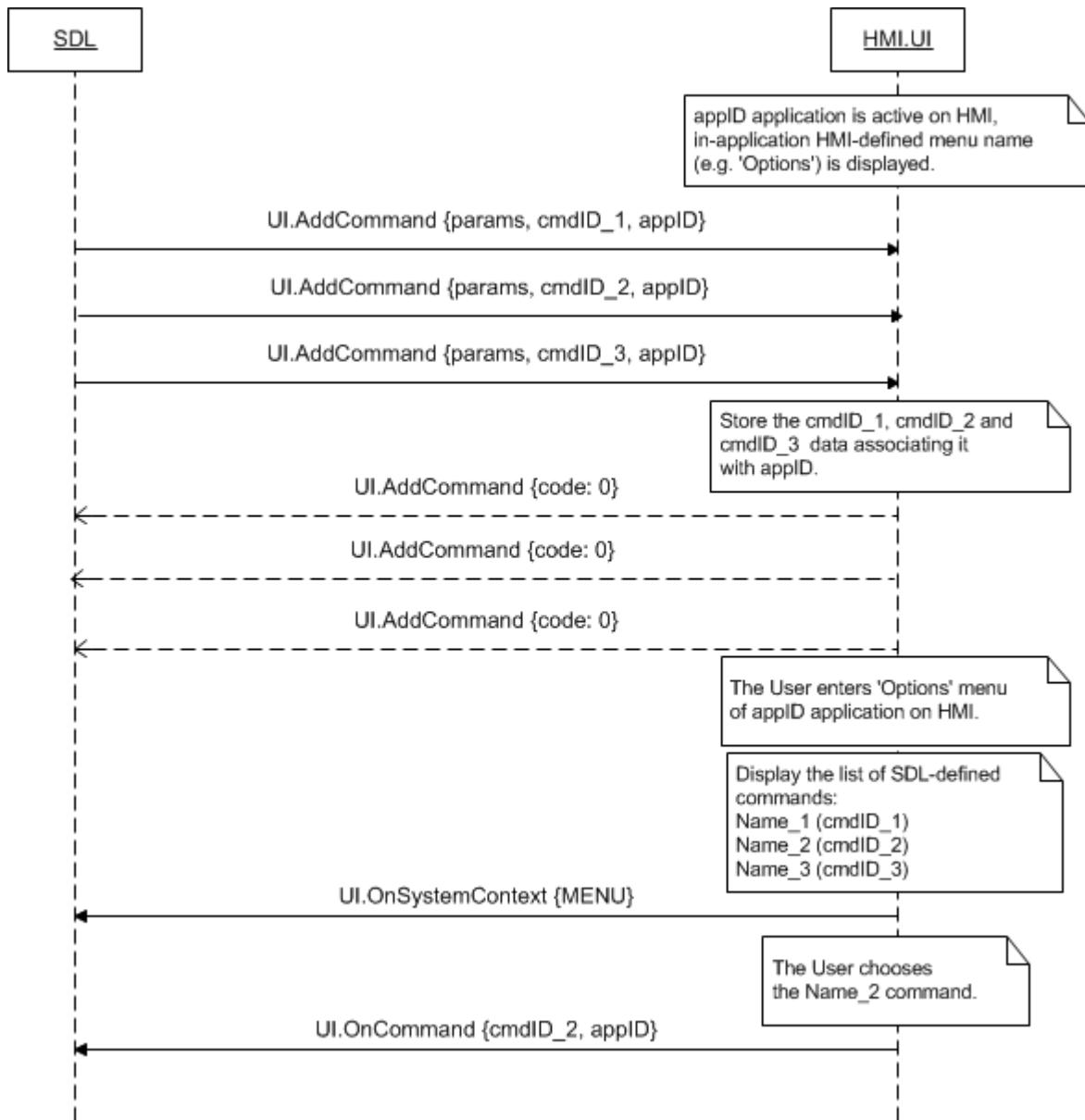
- Command ID (`cmdID`) initially provided by SDL via `UI.AddCommand`
- Application ID (`appID`) initially provided by SDL via `OnAppRegistered` or `UpdateAppList`.

7.23.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|---------|-----------|---|---|
| <code>cmdID</code> | Integer | true | <code>minvalue = 0</code> <code>maxvalue = 2000000000</code> | ID of the command that corresponds to the chosen menu item. This ID was previously sent by SDL within <code>AddCommand</code> request. |
| <code>appID</code> | Integer | true | – | ID of the application that concerns this RPC. Initially is provided by SDL via <code>OnAppRegistered</code> or <code>UpdateAppList</code> . |

7.23.2 Sequence Diagrams

7.23.2.1 OnCommand



7.23.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "UI.OnCommand",
  "params" :
  {
    "cmdID" : 2318,
    "appID" : 65409
  }
}
```

7.23.4 D-Bus Messages Examples

```
signal
sender=:1.209
-> dest=(null destination)
serial=83
```

```

path=/;
interface=com.ford.sdl.hmi.UI;
member=OnCommand
    int32 6          //cmdID
    int32 65537      //appID

```

7.24 OnSystemContext

7.24.1 Description

| | |
|-----------------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the User-initiated interaction on HMI. |

SDL requires the information about the state of HMI and whether the User-initiated interaction is currently taking place on it.

HMI must:

Send the OnSystemContext notification of:

- MAIN: when the User doesn't interact with application via persistant display (neither the in-application menu nor dialog/overlay is currently displayed).
- VRSESSION: when VR becomes active
 - as a result of PTT with help list of VR synonyms displayed
 - as a result of UI.PerformInteraction (of BOTH or VR_ONLY mode) request from SDL
- MENU: when the User enters in-application menu (e.g. 'Options') on UI.
- HMI_OBSCURED: when HMI is currently obscuring the application persistenr display with:
 - The dialog of ScrollableMessage, Slider, PerfromInteraction (MANUAL_ONLY and BOTH mode), PerfromAudioPassThru.
 - Any HMI-defined (system) dialog.
- ALERT: when SDL-requested Alert dialog is displayed.

Note:

The *OnSystemContext* must be sent right after the event has occurred.

7.24.1.1 Parameters

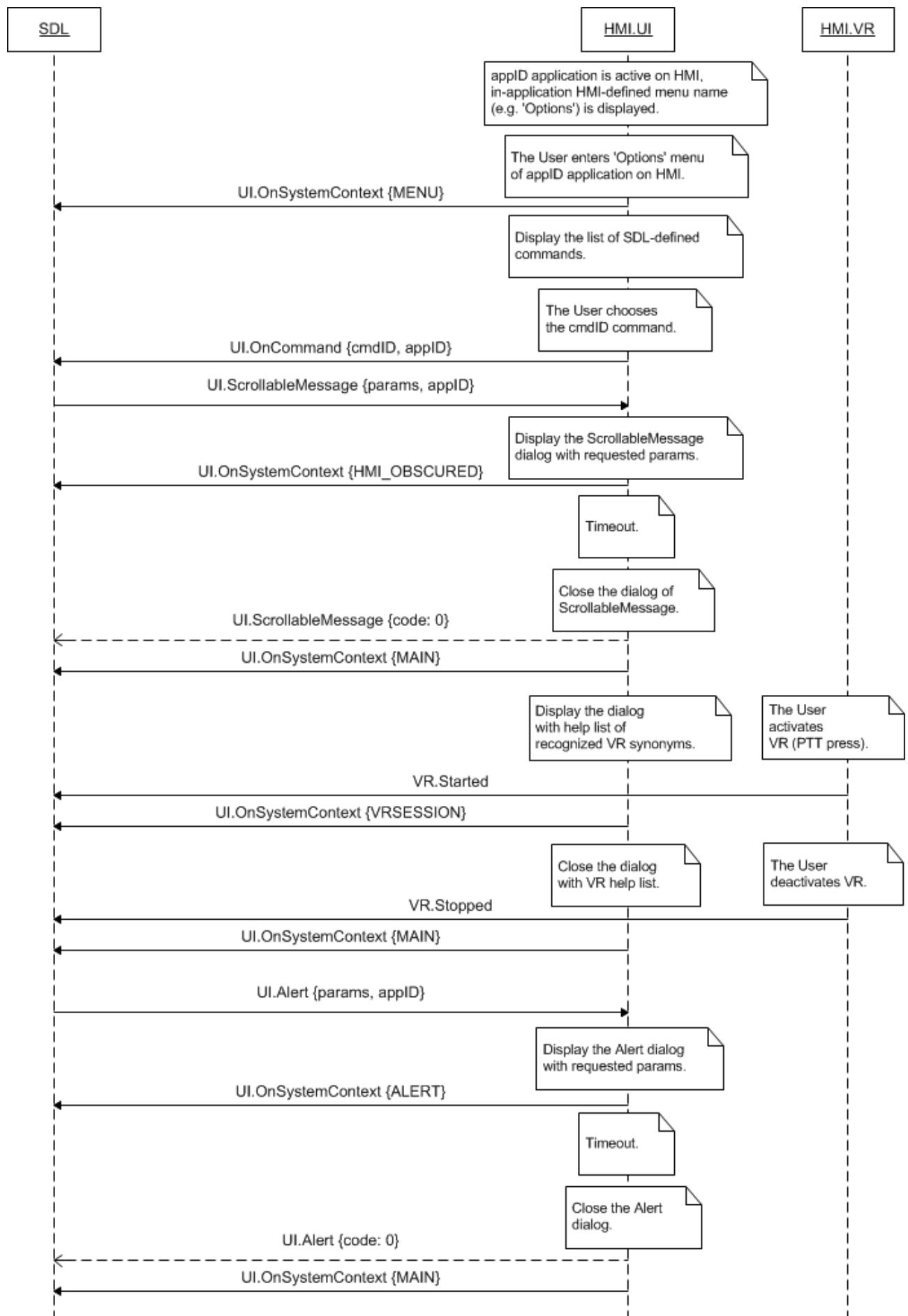
| Param Name | Type | Mandatory | Description |
|---------------|----------------------|-----------|---|
| systemContext | Common.SystemContext | true | The context the application is brought into. See SystemContext. |
| appID | integer | false | |

7.24.1.2 SystemContext

| Element name | Value | Short Description |
|--------------|-------|--|
| MAIN | 0 | If there is currently no user interaction (user-initiated or app-initiated) with the HMI, HMI must notify the SystemContext is MAIN. |
| VRSESSION | 1 | Must be sent if there is a current VR-oriented user interaction (VR becomes active as a result of PTT or PerformInteraction). |
| MENU | 2 | Must be sent if HMI is currently displaying an in-application menu onscreen. |
| HMI_OBSCURED | 3 | Must be sent if HMI is currently obscuring the application display either with a system or with application overlay (except of Alert element). |
| ALERT | 4 | Must be sent if the Alert message is currently displayed onscreen. |

7.24.2 Sequence Diagrams

7.24.2.1 OnSystemContext on different HMI states



7.24.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "UI.OnSystemContext",  
    "params" :  
    {  
        "systemContext" : VRSESSION  
    }  
}
```

7.24.4 D-Bus Messages Examples

```
signal  
sender=:1.209  
-> dest=(null destination)  
serial=84  
path=/;  
interface=com.ford.sdl.hmi.UI;  
member=OnSystemContext  
int32 0          //systemContext: MAIN
```

7.25 OnLanguageChange

7.25.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the UI language is changed. |

SDL needs to be in the know when the User changes the HMI display language: Upon the receipt of OnLanguageChange notification SDL will unregister the applications of different language to provide them with possibility to re-register with the correct (new HMI) display language.

HMI must:

Send the UI.OnLanguageChange notification when the User switches HMI to another language and provide this new value with the language parameter.

7.25.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|-----------------|-----------|-----------------------------------|
| language | Common.Language | true | Language that UI has switched to. |

7.25.1.2 Language

| Element Name | Value | Description |
|--------------|-------|------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |

| Element Name | Value | Description |
|---------------------|--------------|--------------------------------|
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

7.25.2 Sequence Diagrams

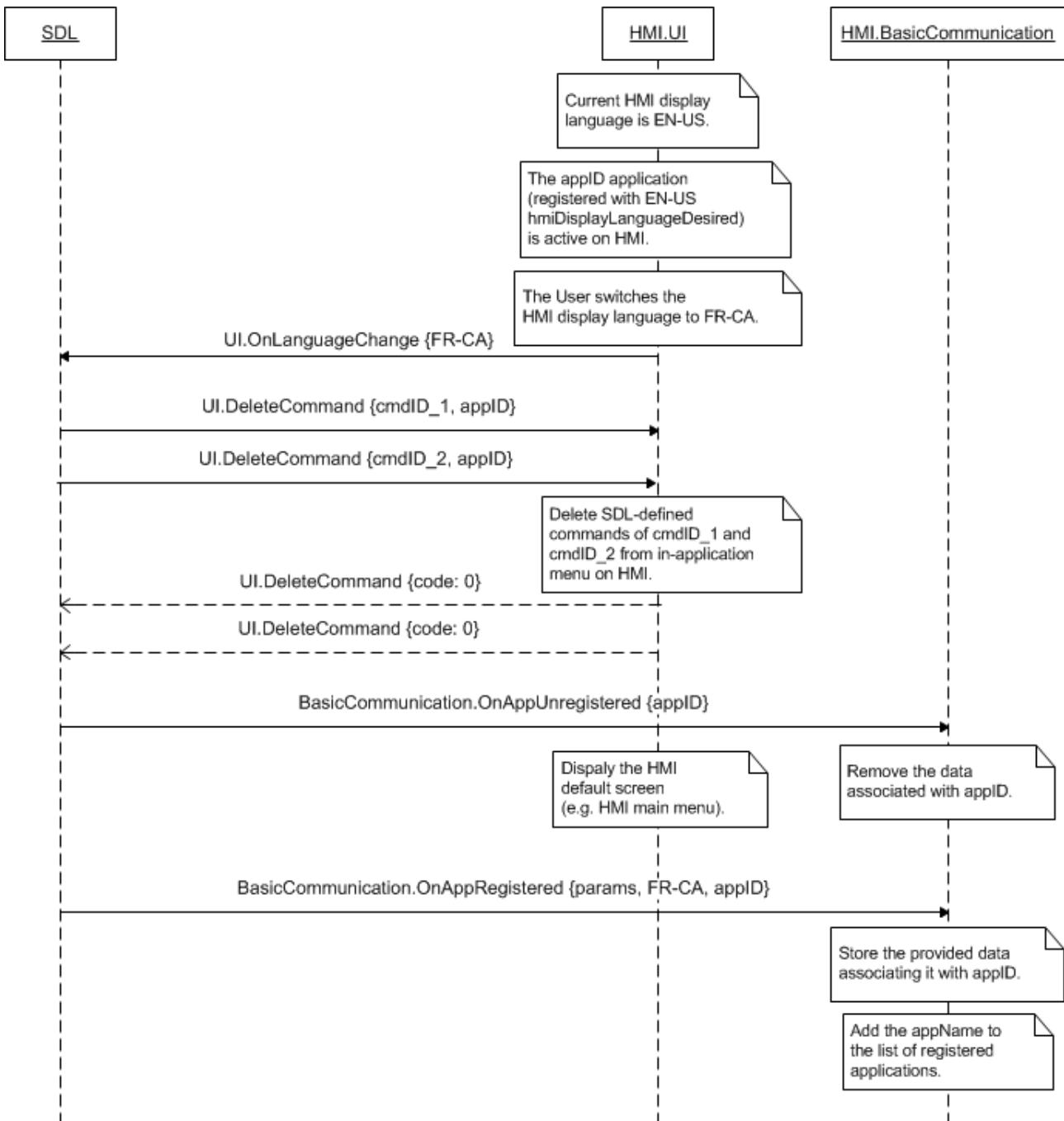
7.25.2.1 OnLanguageChange

Note:

In case of several applications registered with HMI, upon *OnLanguageChange* receipt SDL will send the following to HMI:

- RPCs to delete the sub-menus and UI/VR commands associated with every of appIDs
- *OnAppUnregistered* notifications for each of appIDs.

After the applications re-register with the new (provided via *OnLanguageChange*) *hmiDisplayLanguageDesired*, SDL will send the corresponding *OnAppRegistered* notifications to HMI.



7.25.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "UI.OnLanguageChange",
    "params" :
    {
        "language" : FR-CA
    }
}
```

7.25.4 D-Bus Messages Examples

```
signal
sender=:1.226
-> dest=(null destination)
serial=50
```

```

path=/;
interface=com.ford.sdl.hmi.UI;
member=OnLanguageChange
int32 2

```

7.26 OnKeyboardInput

7.26.1 Description

| | |
|----------|----------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform the keyboard event. |

HMI may be requested by SDL to display the touchscreen keyboard via:

- PerformInteraction with layoutMode: KEYBOARD – keyboard to be displayed
- PerformInteraction with layoutMode: ICONS_WITH_SEARCH – choices as icons to be displayed, the keyboard to be displayed upon User's clicking in search bar area.
- PerformInteraction with layoutMode: LIST_WITH_SEARCH – choices as a list to be displayed, the keyboard to be displayed upon User's clicking in search bar area.

The keyboard layout, language, way of sending the entry are defined by SDL via UI.SetGlobalProperties (keyboardProperties structure) RPC.

Depending on keypressMode value (from keyboardProperties structure of UI.SetGlobalProperties), HMI must send the onKeyboardInput notification with the following data:

- SINGLE_KEYPRESS: each and every User's keypress must be reported (new notification for every newly entered single symbol).
- QUEUE_KEYPRESSES: the whole entry is reported only after the User submits it (by 'Search' button click displayed on touchscreen keyboard)
- RESEND_CURRENT_ENTRY: the whole entry must be reported each and every time the User makes a new keypress (new notification with all previously entered symbols and a newly entered one appended).

HMI must:

Send the OnKeyboardInput notification providing the data entered from and the event occurred over the keyboard.

7.26.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|----------------------|-----------|----------------------------------|--|
| event | Common.KeyboardEvent | true | – | The event occurred over the touchscreen keyboard. |
| data | String | false | Maxlength = 500 Minlength = 0 | On-screen keyboard input data: - A single symbol in case keypressMode is defined as SINGLE_KEYPRESS via UI.SetGlobalProperties. - The whole entry after the User has pressed 'Search' button in case keypressMode is defined as QUEUE_KEYPRESS via UI.SetGlobalProperties. - The whole entry every time the User makes a new keypress in case keypressMode is defined as RESEND_CURRENT_ENTRY via UI.SetGlobalProperties. |

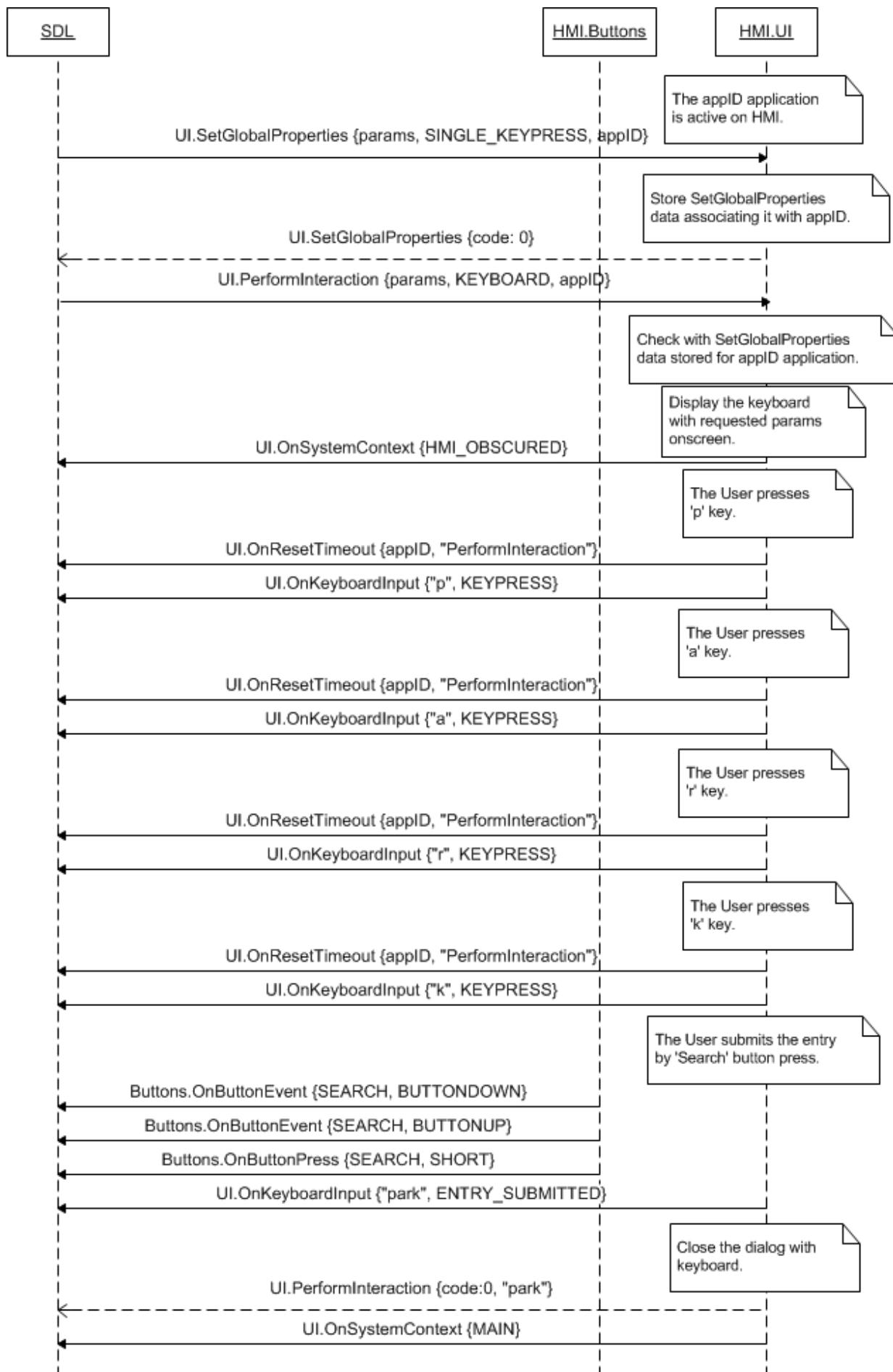
7.26.1.2 KeyboardEvent

| Element name | Short Description |
|--------------|-------------------|
|--------------|-------------------|

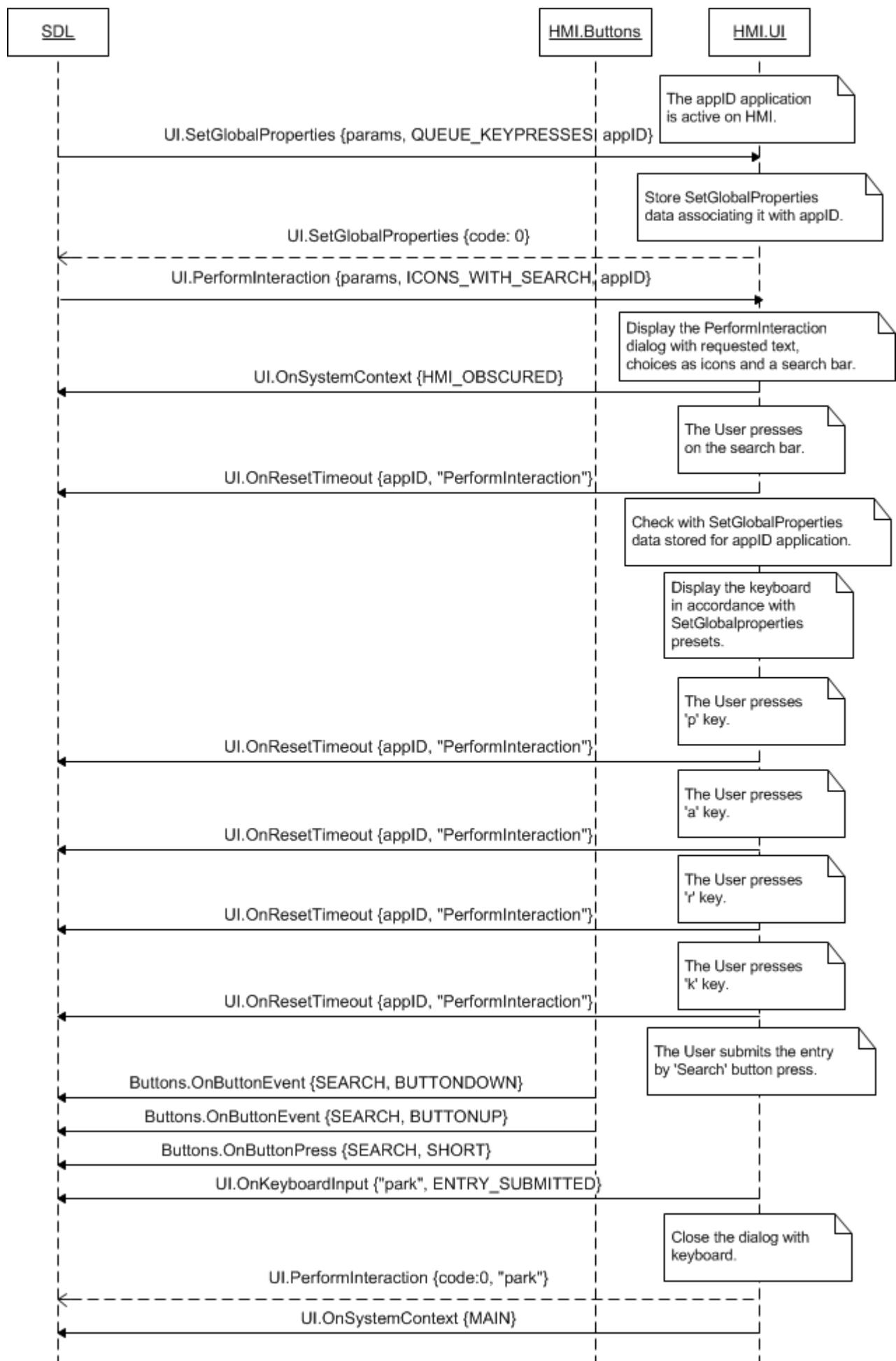
| | |
|-----------------|---|
| KEYPRESS | The User has pressed the keyboard key (applies to both SINGLE_KEYPRESS and RESEND_CURRENT_ENTRY modes). |
| ENTRY_SUBMITTED | The User has finished entering text from the keyboard and submitted the entry. |
| ENTRY_CANCELLED | The User has pressed the HMI-defined 'Cancel' button. |
| ENTRY_ABORTED | The User has not finished entering text and the keyboard is aborted with the event of higher priority. |

7.26.2 Sequence Diagrams

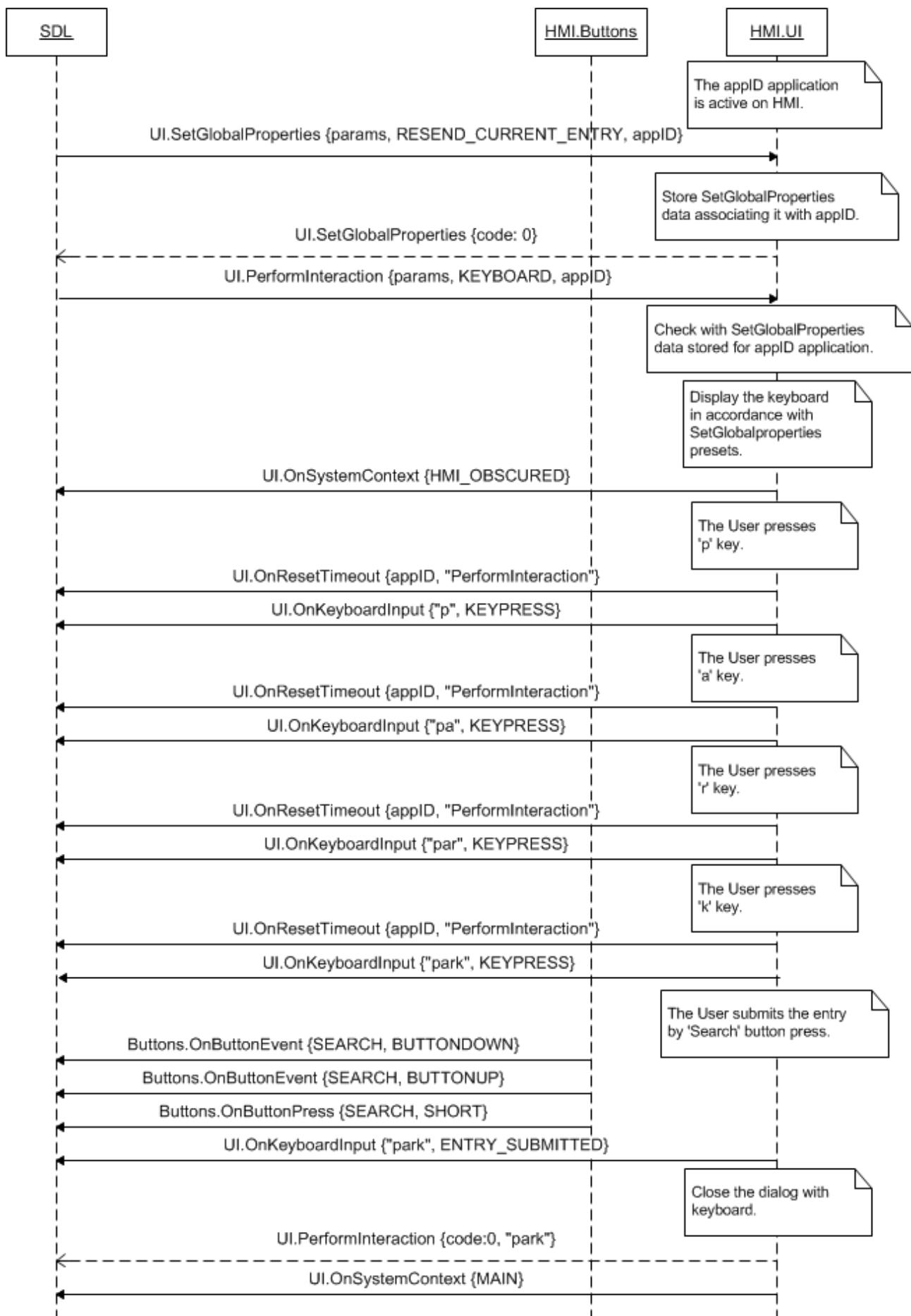
7.26.2.1 OnKeyboardInput for SINGLE_KEYPRESS keypressMode



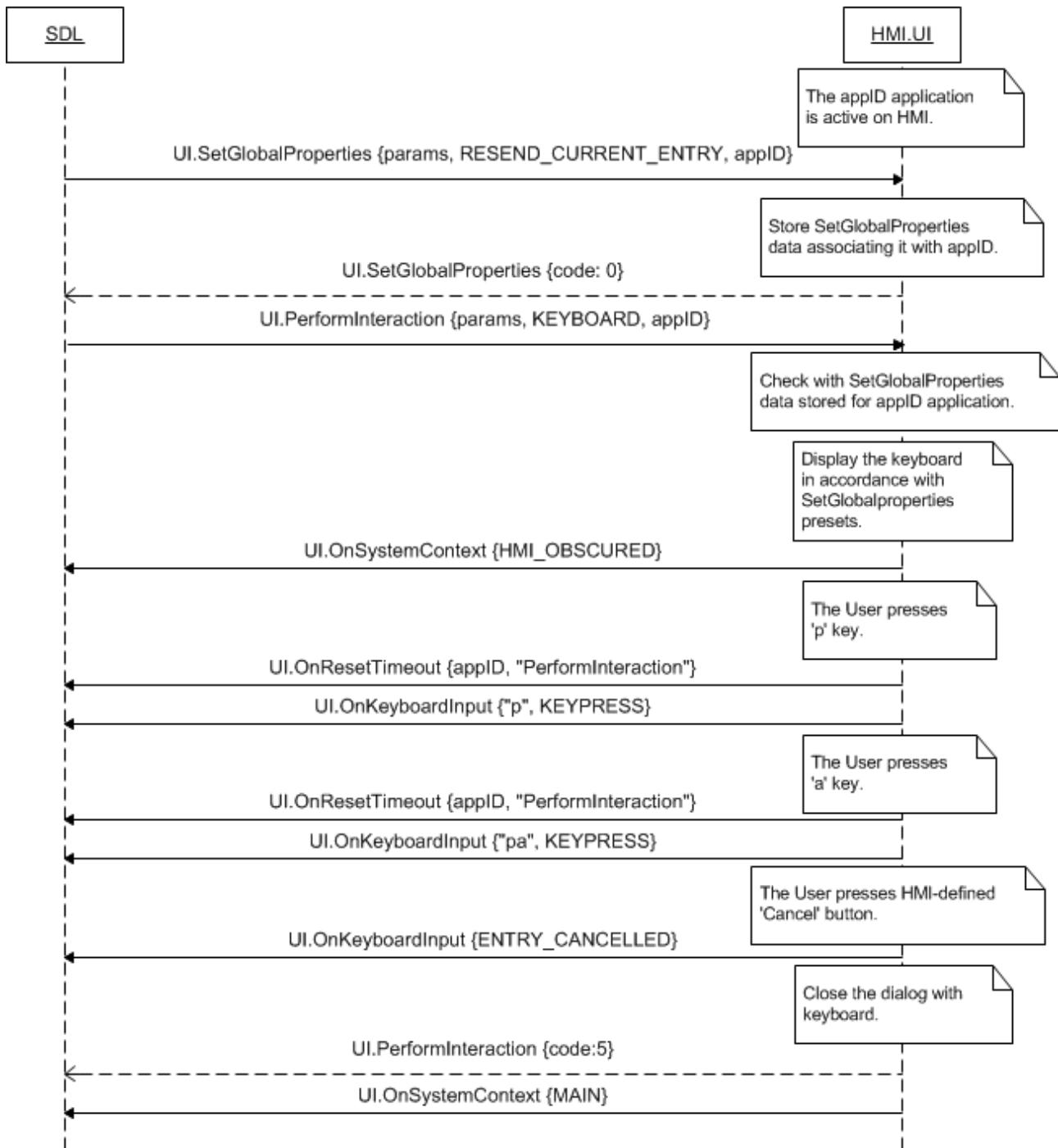
7.26.2.2 OnKeyboardInput for QUEUE_KEYPRESSES keypressMode



7.26.2.3 OnKeyboardInput for RESEND_CURRENT_ENTRY keypressMode



7.26.2.3 OnKeyboardInput for cancelled entry



7.26.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "UI.OnKeyboardInput",
  "params" :
  {
    "event" : ENTRY_CANCELLED
  }
}
```

7.26.4 D-Bus Messages Examples

```
signal
sender=:1.226
```

```

-> dest=(null destination)
serial=50
path=/;
interface=com.ford.sdl.hmi.UI;
member=OnLanguageChange
    int32 0                                //event: KEYPRESS
    struct
    {
        boolean true                      //IsPresent: the optional param is provided
        string "p"                         //data
    }

```

7.27 OnTouchEvent

7.27.1 Description

| | |
|-----------------|--------------------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform the touch event has occurred. |

SDL needs to be informed about every User's touching the touch screen.

HMI must:

7.27.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|-------------------|-----------|---|--|
| type | Common.TouchType | true | - | The type of touch event. |
| event | Common.TouchEvent | true | array = true minsize = 1 maxsize = 10 | List of all individual touches involved in this event. |

7.27.1.2 TouchEvent

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|--|--|
| id | Integer | true | minvalue = 0 maxvalue = 9 | A touch's unique identifier. The application can track the current touch events by id. If a touch event has type begin, the id should be added to the set of touches. If a touch event has type end, the id should be removed from the set of touches. |
| ts | Integer | true | Array = true minvalue = 0 maxvalue = 2147483647 minsize = 1 maxsize = 1000 | The time that the touch was recorded. This number can be the time since the beginning of the session or something else as long as the units are in milliseconds. The timestamp is used to determine the rate of change of position of a touch. The application also uses the time to verify whether two touches, with different ids, are part of a single action by the user. If there is only a single timestamp in this array, it is the same for every coordinate in the |

| | | | | |
|---|-------------------|------|---|--|
| | | | | coordinates array. |
| c | Common.TouchCoord | true | Array = true minsize = 1 maxsize = 1000 | The coordinates of the screen area where the touch event occurred. |

7.27.1.2 TouchCoord

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|----------------------------------|--------------------------------|
| x | Integer | true | minvalue = 0 maxvalue = 10000 | The x coordinate of the touch. |
| y | Integer | true | minvalue = 0 maxvalue = 10000 | The y coordinate of the touch. |

7.27.1.2 TouchType

| Element name | Value | Short Description |
|--------------|-------|--|
| BEGIN | 0 | The user has touched the screen. |
| MOVE | 1 | The User has moved his finger over the screen. |
| END | 2 | The User has removed his finger from the screen. |

7.27.2 Sequence Diagrams

7.27.2.1 OnTouchEvent

7.27.3 JSON Messages Examples

| |
|--|
| |
|--|

7.27.4 D-Bus Messages Examples

| |
|--|
| |
|--|

7.28 OnResetTimeout

7.28.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about timeout reset for the named function. |

SDL needs to be in the know when the timeout of the recently called UI RPC is reset by definite User's actions.

HMI must:

Send the OnResetTimeout notification in the following cases :

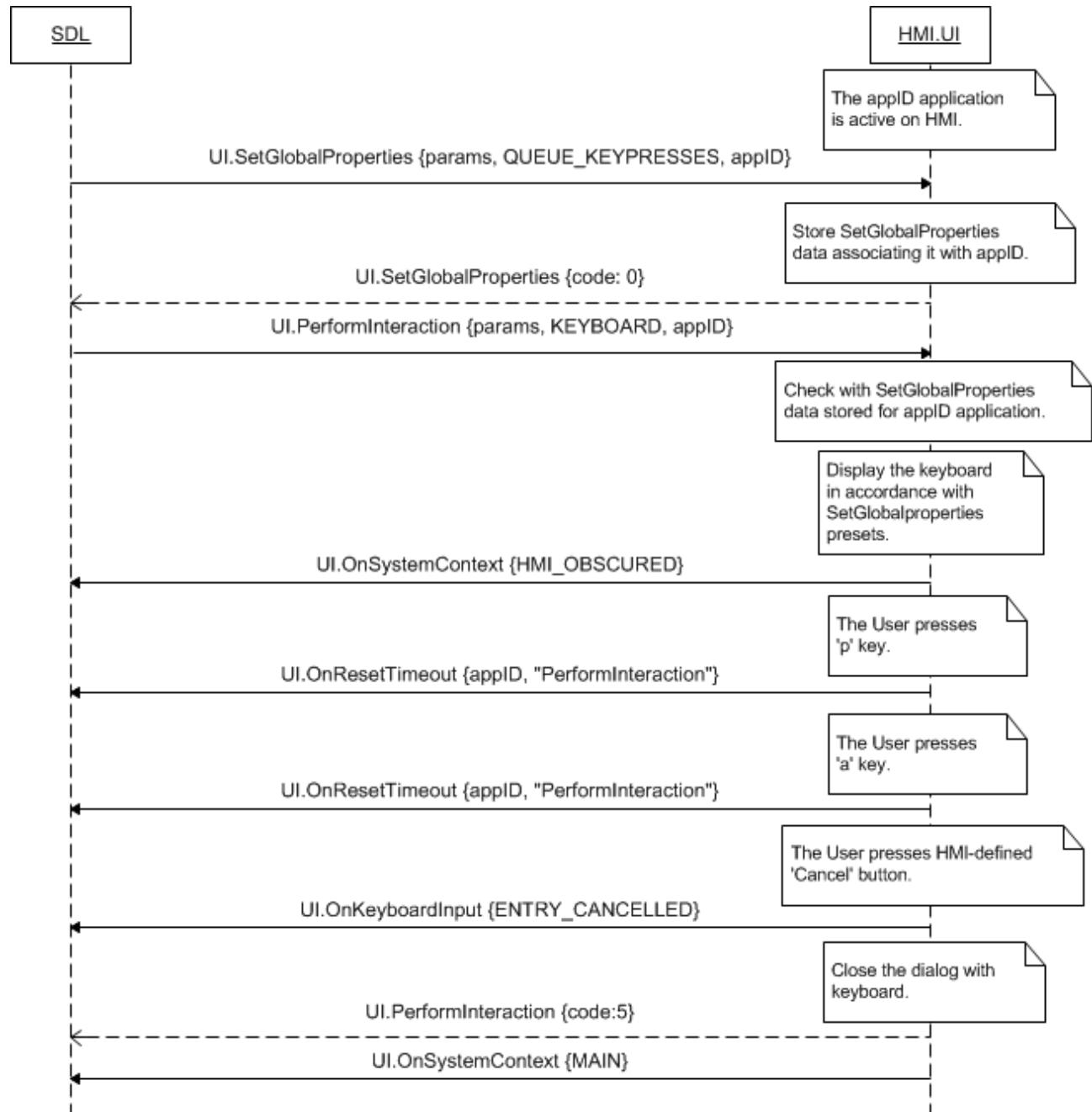
- PerformInteraction of KEYBOARD layoutMode – upon every User's keypress.
- ScrollableMessage:
 - Upon User's actions over the message (e.g. scrolling)
 - Upon KEEP_CONTEXT soft button (defined within ScrollableMessage RPC) press.
- Alert – upon KEEP_CONTEXT soft button (defined within Alert RPC) press

7.28.1.1 Parameters

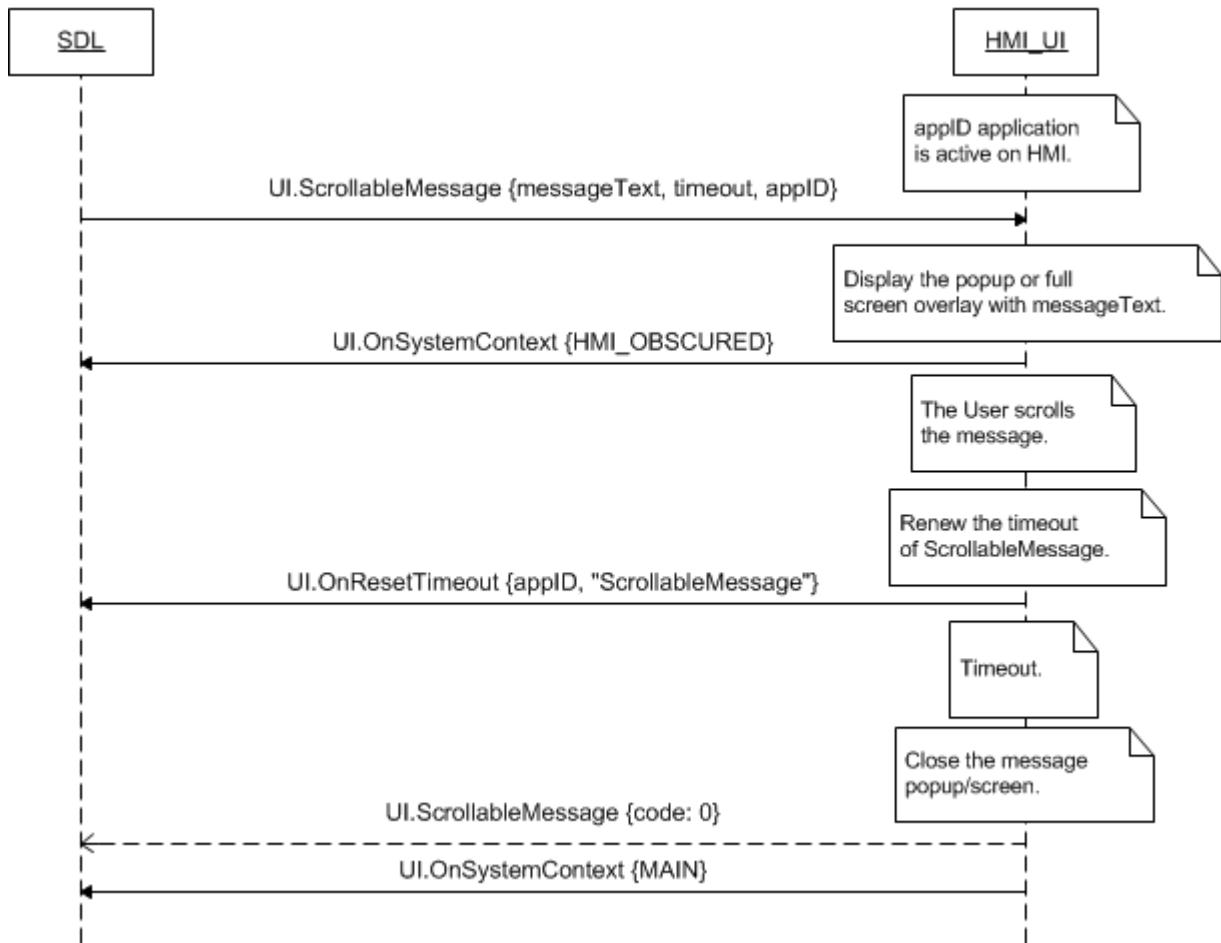
| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|--|
| appId | Integer | true | Id of application that invoked notification. |
| methodName | String | true | Currently used method name on which was triggered action |

7.28.2 Sequence Diagrams

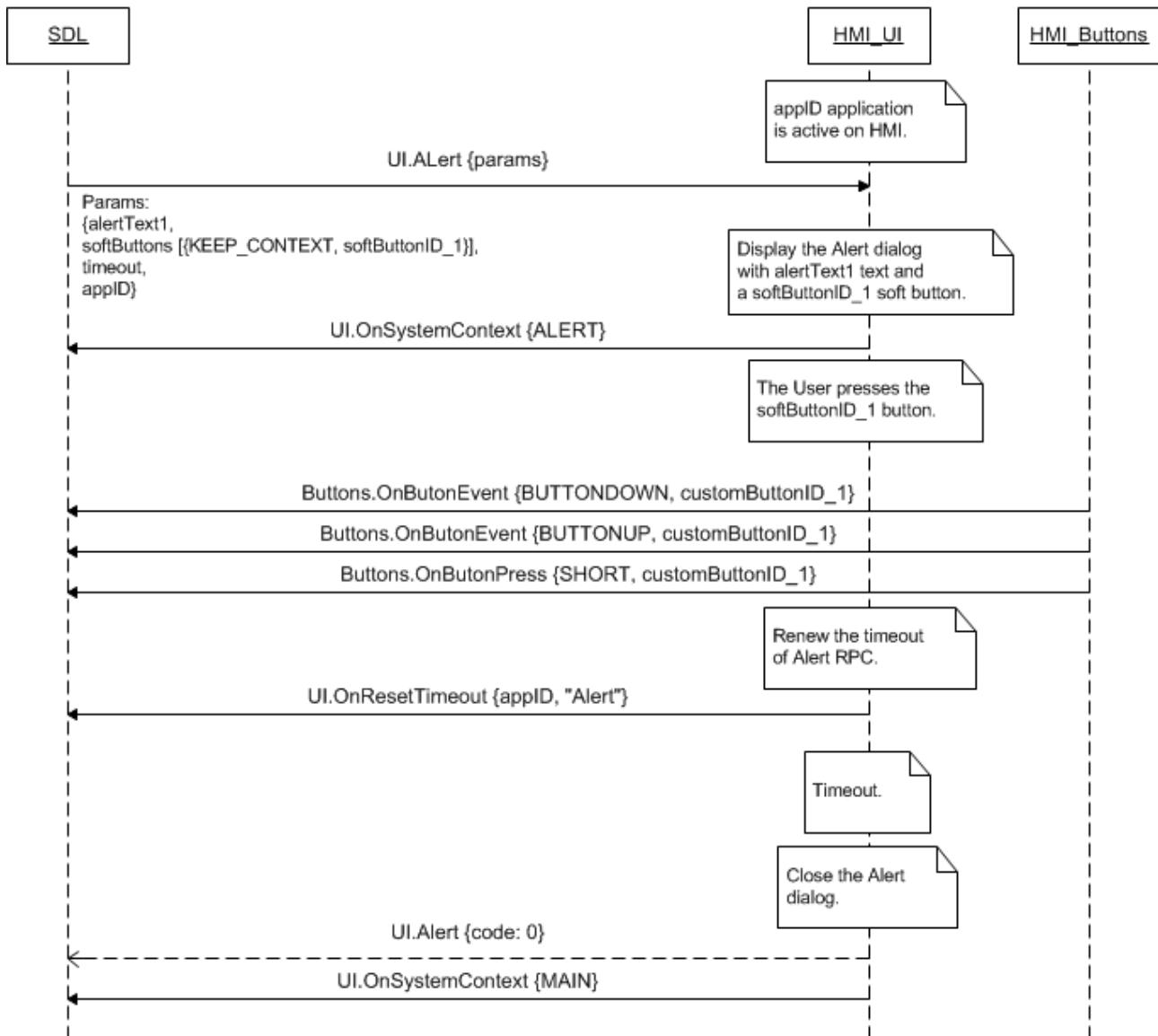
7.28.2.1 OnResetTimeout upon keypress during PerformInteraction (KEYBOARD)



7.28.2.2 OnResetTimeout upon User's scrolling the message during ScrollableMessage



7.28.2.2 OnResetTimeout upon KEEP_CONTEXT soft button press during Alert



7.28.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "UI.OnResetTimeout",
}
```

7.28.4 D-Bus Messages Examples

7.30 OnDriverDistraction

7.30.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about changes of driver distraction state. |

HMI must:

Inform SDL via OnDriverDistraction whenever the driver distraction rules are changed on HMI.

SDL requires this information for determining which RPCs to allow for the application.

7.30.1.1 Parameters

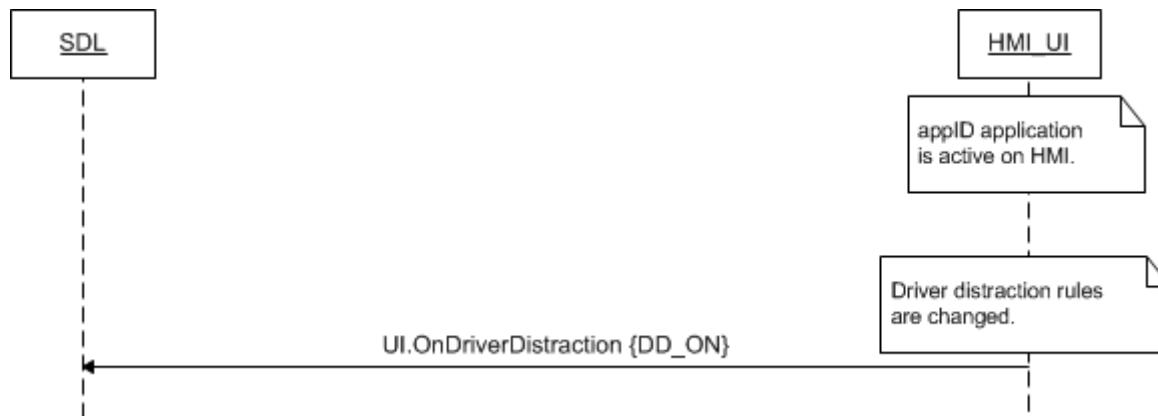
| Param Name | Type | Mandatory | Description |
|------------|-------------------------------|-----------|--|
| state | Common.DriverDistractionState | true | Notifies the application of the current driver distraction state |

7.30.1.2 DriverDistractionState

| Element name | Value | Short Description |
|--------------|-------|---|
| DD_ON | 0 | Driver Distraction rules are in effect. |
| DD_OFF | 1 | Driver Distraction rules are not in effect. |

7.30.2 Sequence Diagrams

7.30.2.1 OnDriverDistraction notification



7.30.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "UI.OnDriverDistraction",
    "params" :
    {
        "state" : DD_ON
    }
}
```

7.30.4 D-Bus Messages Examples

```
[REDACTED]
```

7.31 OnRecordStart

7.31.1 Description

| | |
|----------|---------------------------------|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Start capturing from microphone |

HMI must:

7.31.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|--|
| appID | Integer | true | ID of application related to this RPC. |

7.31.2 Sequence Diagrams

7.31.2.1 OnRecordStart

7.31.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "UI.OnRecordStart",  
    "params" :  
    {  
        "appID" : 65537  
    }  
}
```

7.31.4 D-Bus Messages Examples

7.32 SetDisplayLayout

7.32.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Set HMI-predefined display template for the named application. |

With `SetDisplayLayout` SDL requests HMI to use the named template when arranging the display layout of the named application.

Within `SetDisplayLayout` request SDL will use the template name (see section 7.32.2.2 Parameters) that HMI previously provides within response to `UI.GetCapabilities` (see section 7.2.3 Response and the below extract of corresponding JSON message).

Extract of JSON response to `UI.GetCapabilities` that contains templates names

```
"method" : "UI.GetCapabilities",  
"params":  
{  
    ...  
    "displayCapabilities" :  
    {  
        ...  
        "templatesAvailable" : ["template_1", "template_2"],  
        ...  
    },  
    ...
}
```

7.32.2 Request

7.32.2.1 Behavior

HMI must:

1. Check whether the `displayLayout` parameter (see table in section 7.32.2.2) contains the correct template name (string value):
 - If correct, proceed with steps 2,3,4.
 - If not correct or some error occurred, proceed with step 3 only.
2. Associate the named template of `displayLayout` with the provided application `appID` (see table in section 7.32.2.2).
3. Respond with one of the appropriate result codes (see section 7.32.3 Response for applicable result codes). And in case of `SUCCESS` return the capabilities of the named template:
 - `displayCapabilities`: the type of display, supported text and image fields, supported formats of media clock and other that is described in [section 7.32.3.2](#).
 - `buttonCapabilities`: capabilities of hardware buttons: the names of available buttons, information about whether notifications on short/long presses and up/down events are supported by HMI (see [section 7.32.3.13](#)).
 - `softButtonCapabilities`: capabilities of soft buttons: information about whether HMI supports referencing image and notifying on short/long presses and up/down events (see [section 7.32.3.15](#)).
 - `presetBankCapabilities`: information about whether HMI supports duplicating the hardware custom presets with onscreen buttons (see [section 7.32.3.16](#)).
4. When the `appID` application is activated on HMI, HMI must switch to the corresponding template (associated in step 2.) that implements the corresponding capabilities (reported in step 3.).

Note:

- The applicable to this RPC result codes are provided in section 7.32.3 Response.
- The sequence diagrams describing the expected HMI behavior are provided in the section 7.32.4 Sequence Diagrams

7.32.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|----------------------------|---------|-----------|-----------------|---|
| <code>displayLayout</code> | String | true | Maxlength = 500 | The name of predefined or dynamically created screen layout. Currently only predefined screen layouts are defined. See section 7.32.3.17 PredefinedLayout for the variants of template names. Still, HMI is not limited with these values and is free to create the name(s) that fit the template behavior/purpose/capabilities, etc. |
| <code>appID</code> | Integer | true | – | ID of the application that requested this RPC. |

7.32.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|---------------|---------------|---|-----------------------|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI recognizes the requested.template name and provides the capabilities of this template. | JSON response | Method return | <code>displayCapabilities</code> , <code>buttonCapabilities</code> , <code>softButtonCapabilities</code> , <code>presetBankCapabilities</code> | See section 7.32.3.1. |

| | | | | | |
|---------|--|--------------------|---------------|------------------------|---|
| | | | | pabilities, code: 0 | |
| Failure | UNSUPPORTED_RESOURCE: HMI does not support any display templates at all (response to UI.GetCapabilities contains an empty array of templatesAvailable). | JSON error message | Method return | code: 2 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | IGNORED: The named template is already associated with the named appID application | | | code: 6 | |
| | INVALID_DATA: The requested template name is not valid. | | | code: 11 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

7.32.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------------------|-------------------------------|-----------|--|---|
| displayCapabilities | Common.DisplayCapabilities | false | - | The type of display, supported text and image fields, formats of media clock and other that is described in section 7.32.3.2 . |
| buttonCapabilities | Common.ButtonCapabilities | false | Array = true Minsize = 1 Maxsize = 100 | Capabilities of hardware buttons: the names of available buttons, information about whether notifications on short/long presses and up/down events are supported by HMI (see section 7.32.3.13). |
| softButtonCapabilities | Common.SoftButtonCapabilities | false | Array = true Minsize = 1 Maxsize = 100 | Must be returned when the template supports on-screen SoftButtons. Capabilities of soft buttons: information about whether HMI supports referencing image and notifying on short/long presses and up/down events (see section 7.32.3.15). |
| presetBankCapabilities | Common.PresetBankCapabilities | false | - | Must be returned when the template supports custom on-screen presets. Information about whether HMI supports duplicating the hardware custom presets with onscreen buttons (see section 7.32.3.16). |

7.32.3.2 DisplayCapabilities

| Param Name | Type | Mandatory | Additional | Description |
|-------------|--------------------|-----------|------------|---|
| displayType | Common.DisplayType | true | - | The type of the display that is installed on HU. See section 7.32.3.3 DisplayType. |

| Param Name | Type | Mandatory | Additional | Description |
|---------------------------|-------------------------|-----------|---|---|
| textFields | Common.TextField | true | Array = true minsize = 0 maxsize = 100 | A set of all fields that support displaying text data. If there are no textfields supported HMI must send the empty array. See section 7.32.3.4 TextField. |
| imageFields | Common.ImageField | false | Array = true minsize = 1 maxsize = 100 | A set of all fields that support images. See section 7.32.3.5 ImageField |
| mediaClockFormats | Common.MediaClockFormat | true | Array = true minsize = 1 maxsize = 100 | A set of all supported formats of the media clock. See section 7.32.3.9 MediaClockFormat. |
| imageCapabilities | Common.ImageType | false | Array = true minsize = 0 maxsize = 2 | The array of supported image types (static and/or dynamic). The empty array should be returned if the platform does not support displaying images. See section 7.32.3.10 ImageType. |
| graphicSupported | Boolean | true | - | Must be: - 'true' if display's persistent screen supports referencing a static or dynamic image. - 'false' if not. |
| templatesAvailable | String | true | Array = true minsize = 0 maxsize = 100 maxlength = 100 | A set of all predefined persistent display templates available on headunit. See section 7.32.3.17 PredefinedLayout for the variants of template names. Still, HMI is not limited with these values and is free to create the name(s) that fit the template behavior/purpose/capabilities, etc. |
| screenParams | Common.ScreenParams | false | - | A set of all parameters related to a prescribed screen area (e.g. for video / touch input). See section 7.32.3.11 ScreenParams. |
| numCustomPresetsAvailable | Integer | false | minvalue = 1 maxvalue = 100 | The number of on-screen custom presets available (if any); otherwise omitted. |

7.32.3.3 DisplayType Enumeration

| Element name | Short Description |
|--------------|---|
| CID | Center Information Display. This display type provides a 2-line x 20 character "dot matrix" display. |
| TYPE2 | TYPE II display. 1 line older radio head unit. |
| TYPE5 | TYPE V display Old radio head unit. |
| NGN | Next Generation Navigation display. |
| GEN2_8_DMA | GEN-2, 8 inch display. |
| GEN2_6_DMA | GEN-2, 6 inch display. |
| MFD3 | 3 inch GEN1.1 display |
| MFD4 | 4 inch GEN1.1 display |
| MFD5 | 5 inch GEN1.1 display |
| GEN3_8-INCH | GEN-3, 8 inch display. |

7.32.3.4 TextField

| Param Name | Type | Mandatory | Additional | Description |
|--------------|------------------------|-----------|--------------------------------|--|
| name | Common.TextFieldIdName | true | - | The name that identifies the field. See TextFieldName. |
| characterSet | Common.CharacterSet | true | - | The character set that is supported in this field. See CharacterSet. |
| width | Integer | true | minvalue="1" maxvalue="500" | The number of characters in one row of this field. |
| rows | Integer | true | minvalue="1" maxvalue="3" | The number of rows of this field. |

7.32.3.5 TextFieldName Enumeration

| Element name | Short Description |
|--------------|---|
| mainField1 | The text that must be displayed in a single or upper display line. If this value is not set, the text of mainField1 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.5. |
| mainField2 | The text that must be displayed on the second display line. If this text is not set, the text of mainField2 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.5. |
| mainField3 | The text that must be displayed on the second "page" first display line. If this text is not set, the text of mainField3 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.5. |
| mainField4 | The text that must be displayed on the second "page" second display line. If this text is not set, the text of mainField4 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.5. |
| statusBar | The text is placed in the status bar area. Note: This relates to navigation displays If this parameter is omitted, the status bar text must remain unchanged. If this parameter is an empty string, the field must be cleared. If provided and the display has no status bar, this parameter must be ignored. Applies to Show, section 7.5. |
| mediaClock | Text value for MediaClock field. Shall arrive in the form as described in the MediaClockFormat enumeration If this text is set, any automatic media clock updates previously set with SetMediaClockTimer must be stopped. Applies to Show, section 7.5. |
| mediaTrack | The text that should be displayed in the track field. This field should be valid only for media applications on. If this text is not set, the text of mediaTrack must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.5. |
| alertText1 | The text that must be displayed in the top field of the display during the Alert. Applies to Alert, section 7.4. |
| alertText2 | The text that must be displayed in the bottom field of the display during the Alert. Applies to Alert, section 7.4. |

| Element name | Short Description |
|---------------------------|--|
| alertText3 | The optional third line of the alert text field. Applies to Alert , section 7.4. |
| scrollableMessageBody | The long form body of text that can include newlines and tabs. Applies to ScrollableMessage , section 7.19. |
| initialInteractionText | Must be displayed when the interaction begins. The text must be displayed on the first line of a multiline display, and must be centered. Applies to PerformInteraction , section 7.12. |
| navigationText1 | The text that must be displayed on the first line of navigation text. Applies to ShowConstantTBT , section 12.3. |
| navigationText2 | The text that must be displayed on the second line of navigation text. Applies to ShowConstantTBT , section 12.3. |
| ETA | Estimated Time of Arrival for navigation. Applies to ShowConstantTBT , section 12.3. |
| totalDistance | Total distance to destination for navigation. Applies to ShowConstantTBT , section 12.3. |
| navigationText | Navigation text for UpdateTurnList . Applies to Turn , section 12.4. |
| audioPassThruDisplayText1 | The first line of text that must be displayed during audio capture. Applies to PerformAudioPassThru , section 7.20. |
| audioPassThruDisplayText2 | The second line of text that must be displayed during audio capture. Applies to PerformAudioPassThru , section 7.20. |
| sliderHeader | The text that must be displayed on the header of slider. Applies to Slider , section 7.18. |
| sliderFooter | The text that must be displayed on the footer of slider. Applies to Slider , section 7.18. |
| notificationText | The text that must be displayed to notify the User on some event. Applies to ShowNotification , section 7.31. |
| menuName | Primary text for Choice. Applies to PerformInteraction , section 7.12. |
| secondaryText | Secondary text for Choice. Applies to PerformInteraction , section 7.12. |
| tertiaryText | Tertiary text for Choice. Applies to PerformInteraction , section 7.12. |
| timeToDestination | The line to display the time to destination. Applies to ShowConstantTBT , section 12.3. |
| turnText | Currently not used. |

7.32.3.6 CharacterSet

| Element name | Value | Short Description |
|---------------------|--------------|--------------------------|
| TYPE2SET | 0 | |
| TYPE5SET | 1 | |
| CID1SET | 2 | |
| CID2SET | 3 | |

7.32.3.7 ImageField Structure

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|----------------------------|------------------|--|---|
| name | Common.ImageFileDialogName | true | - | The name that identifies the field. See section 7.32.3.6 ImageFieldName . |
| imageTypeSupported | Common.FileType | false | array = true minsize = 1 maxsize = 100 | The image types that are supported in this field. See section 7.32.3.7 FileType . |

| Param Name | Type | Mandatory | Additional | Description |
|-----------------|------------------------|-----------|------------|---|
| imageResolution | Common.ImageResolution | false | - | The image resolution of this field. See section 7.32.3.8. |

7.32.3.8 ImageFieldName Enumeration

| Element name | Value | Short Description |
|-----------------------------|-------|--|
| softButtonImage | 0 | The image field for SoftButton |
| choiceImage | 1 | The first image field for Choice |
| choiceSecondaryImage | 2 | The secondary image field for Choice |
| vrHelpItem | 3 | The image field for vrHelpItem |
| turnIcon | 4 | The image field for Turn |
| menuIcon | 5 | The image field for the menu icon in SetGlobalProperties |
| cmdIcon | 6 | The image field for AddCommand |
| appIcon | 7 | The image field for the app icon (set by setAppIcon) |
| graphic | 8 | The image field for Show |
| showConstantTBTIcon | 9 | The primary image field for ShowConstantTBT |
| showConstantTBTNextTurnIcon | 10 | The secondary image field for ShowConstantTBT |
| locationImage | 11 | The optional image of a destination / location |

7.32.3.9 FileType Enumeration

| Element name | Value | Short Description |
|--------------|-------|-------------------|
| GRAPHIC_BMP | 0 | |
| GRAPHIC_JPEG | 1 | |
| GRAPHIC_PNG | 2 | |
| AUDIO_WAVE | 3 | |
| AUDIO_MP3 | 4 | |
| AUDIO_AAC | 5 | |
| BINARY | 6 | |
| JSON | 7 | |

7.32.3.10 ImageResolution Structure

| Param Name | Type | Mandatory | Additional | Description |
|------------------|---------|-----------|----------------------------------|------------------------------|
| resolutionWidth | Integer | true | minvalue = 1 maxvalue = 10000 | The image resolution width. |
| resolutionHeight | Integer | true | minvalue = 1 maxvalue = 10000 | The image resolution height. |

7.32.3.11 MediaClockFormat Enumeration

| Element name | Short Description |
|--------------|--|
| CLOCK1 | minutesFieldWidth = 2; minutesFieldMax = 19; secondsFieldWidth = 2; secondsFieldMax = 99; maxHours = 19; maxMinutes = 59; maxSeconds = 59; Is used for Type II, NGN and CID head units. |

| Element name | Short Description |
|---------------------|--|
| CLOCK2 | <p>minutesFieldWidth = 3; minutesFieldMax = 199; secondsFieldWidth = 2; secondsFieldMax = 99;</p> <p>maxHours = 59; maxMinutes = 59; maxSeconds = 59;</p> <p>Is used for Type V head units.</p> |
| CLOCK3 | <p>minutesFieldWidth = 2; minutesFieldMax = 59; secondsFieldWidth = 2; secondsFieldMax = 59;</p> <p>maxHours = 9; maxMinutes = 59; maxSeconds = 59;</p> <p>Is used for GEN1.1 (i.e. MFD3/4/5) head units.</p> |
| CLOCKTEXT1 | <p>5 characters possible</p> <p>Format: 1 sp c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space</p> <p>Is used for Type II head unit</p> |
| CLOCKTEXT2 | <p>5 chars possible</p> <p>Format: 1 sp c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space</p> <p>Is used for CID and NGN head unit.</p> |
| CLOCKTEXT3 | <p>6 chars possible</p> <p>Format: 1 sp c c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space</p> <p>Is used for Type V head unit.</p> |
| CLOCKTEXT4 | <p>6 chars possible</p> <p>Format: c : sp c c : c c : sp : colon or space c : character out of following character set: sp 0-9 [letters].</p> <p>Is used for GEN1.1 (i.e. MFD3/4/5) head units.</p> |

7.32.3.12 ImageType Enumeration

| Element name | Short Description |
|---------------------|---|
| STATIC | Static image. The image that is sent as the binary or hex code within the request. |
| DYNAMIC | Dynamic image. The image that is stored on HMI and just a link to it is further used within requests. |

7.32.3.13 ScreenParams Structure

| Param Name | Type | Mandatory | Description |
|-------------------|------------------------|------------------|---|
| resolution | Common.ImageResolution | true | The resolution of the prescribed screen area. |

| Param Name | Type | Mandatory | Description |
|---------------------|-------------------------------|------------------|---|
| | | | See section 7.32.3.8 ImageResolution. |
| touchEventAvailable | Common.TouchEventCapabilities | false | Types of screen touch events available in screen area. See section 7.32.3.12. |

7.32.3.14 TouchEventCapabilities Structure

| Param Name | Type | Mandatory | Description |
|----------------------|-------------|------------------|--------------------|
| pressAvailable | Boolean | true | |
| multiTouchAvailable | Boolean | true | |
| doublePressAvailable | Boolean | true | |

7.32.3.15 ButtonCapabilities

| Param Name | Type | Mandatory | Description |
|---------------------|-------------------|------------------|---|
| name | Common.ButtonName | true | The name of supported/existing hardware button. See section 7.32.3.14 ButtonName |
| shortPressAvailable | Boolean | true | Must be 'true' if the button supports short press mode. If 'true' HMI must notify SDL with OnButtonPress (SHORT) every time this button is pressed short. See section 8.3 OnButtonPress |
| longPressAvailable | Boolean | true | Must be 'true' if the button supports long press mode. If 'true' HMI must notify SDL with OnButtonPress (LONG) every time this button is pressed long. See section 8.3 OnButtonPress |
| upDownAvailable | Boolean | true | Must be 'true' if tracking the events of button being depressed/released is supported. If 'true' HMI must notify SDL with OnButtonEvent of DOWN/UP on every button depress/release. See section 8.4 OnButtonEvent |

7.32.3.16 ButtonName

| Element name | Value | Short Description |
|---------------------|--------------|---|
| OK | 0 | Represents the button usually labeled "OK". A typical use of this button is for the User to press it to make a selection. |
| SEEKLEFT | 1 | Represents the seek-left button. A typical use of this button is for the user to scroll to the left through menu choices, one menu item per press. |
| SEEKRIGHT | 2 | Represents the seek-right button. A typical use of this button is for the user to scroll to the right through menu choices one menu item per press. |
| TUNEUP | 3 | Represents a turn of the tuner knob in the clockwise direction one tick. |
| TUNEDOWN | 4 | Represents a turn of the tuner knob in the counter-clockwise direction one tick. |
| PRESET_0 | 5 | Represents the preset 0 button. |
| PRESET_1 | 6 | Represents the preset 1 button. |
| PRESET_2 | 7 | Represents the preset 2 button. |
| PRESET_3 | 8 | Represents the preset 3 button. |
| PRESET_4 | 9 | Represents the preset 4 button. |
| PRESET_5 | 10 | Represents the preset 5 button. |
| PRESET_6 | 11 | Represents the preset 6 button. |

| Element name | Value | Short Description |
|---------------|-------|---|
| PRESET_7 | 12 | Represents the preset 7 button. |
| PRESET_8 | 13 | Represents the preset 8 button. |
| PRESET_9 | 14 | Represents the preset 9 button. |
| CUSTOM_BUTTON | 15 | Represents any of onscreen buttons requested by SDL. |
| SEARCH | 16 | Represents the 'Search' button on the touchscreen keyboard. |

7.32.3.17 SoftButtonCapabilities Structure

| Param Name | Type | Mandatory | Description |
|---------------------|---------|-----------|--|
| shortPressAvailable | Boolean | true | Must be - 'true' if soft buttons support a short press - 'false' if not. See section 8.3 OnButtonPress. |
| longPressAvailable | Boolean | true | Must be - 'true' if soft buttons support a LONG press - 'false' if not. See section 8.3 OnButtonPress. |
| upDownAvailable | Boolean | true | Must be - 'true' if soft buttons support "button down" and "button up". - 'false' if not. See section 8.4 OnButtonEvent |
| imageSupported | Boolean | true | Must be - 'true' if soft buttons support referencing image - 'false' if not. |

7.32.3.18 PresetBankCapabilities

| Param Name | Type | Mandatory | Description |
|--------------------------|---------|-----------|---|
| onScreenPresetsAvailable | Boolean | true | Must be 'true' if HMI supports duplicating the hardware custom presets with onscreen buttons. |

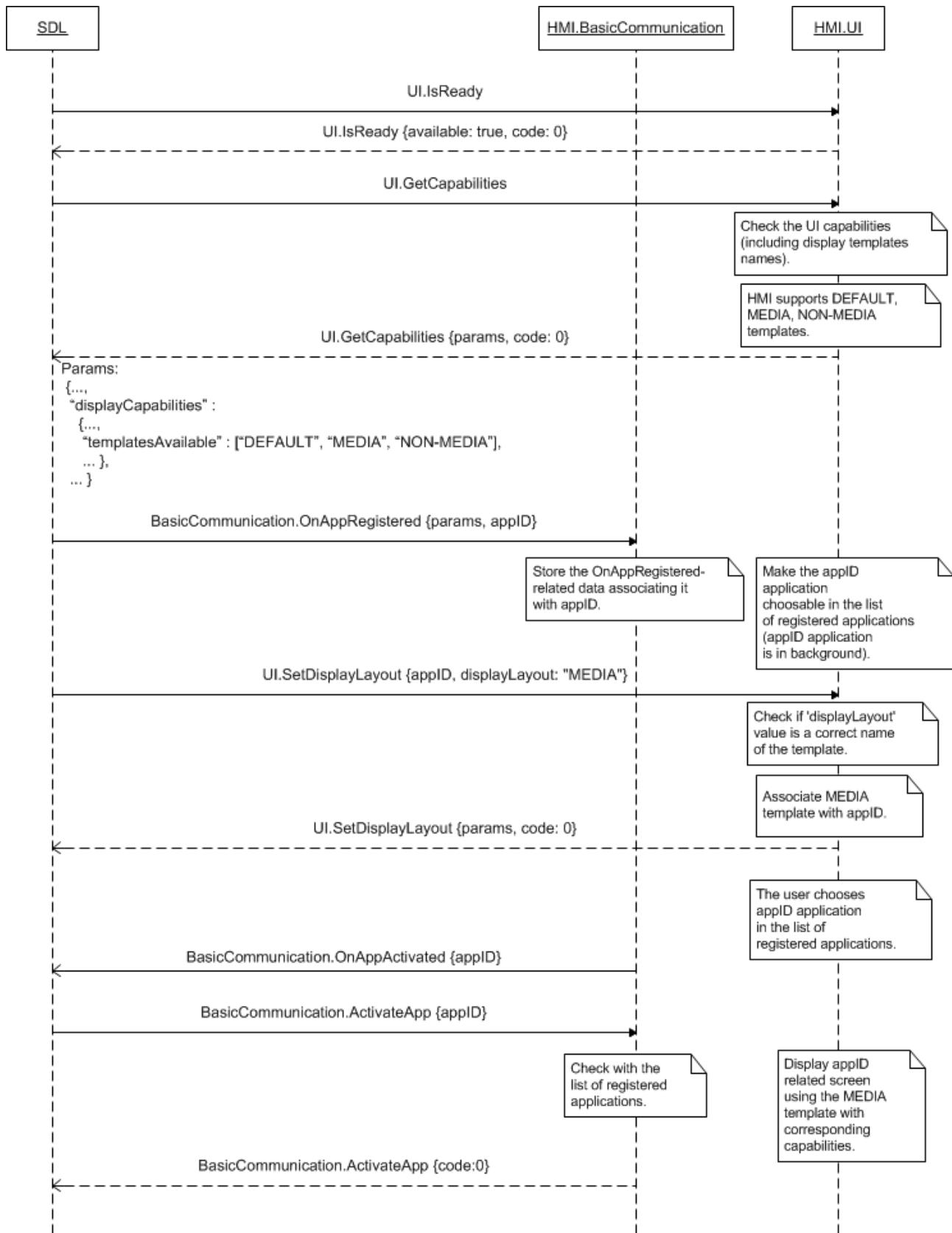
7.32.3.19 PredefinedLayout

| Element name | Short Description |
|--------------------|---|
| DEFAULT | Default media / non-media screen. Can be set as a root screen. |
| MEDIA | Default Media screen. Can be set as a root screen. |
| NON-MEDIA | Default Non-media screen. Can be set as a root screen. |
| ONSCREEN_PRESETS | Custom root media screen containing app-defined onscreen presets. Can be set as a root screen. |
| NAV_FULLSCREEN_MAP | Custom root template screen containing full screen map with navigation controls. Can be set as a root screen. |
| NAV_LIST | Custom root template screen containing video represented list. Can be set as a root screen. |
| NAV_KEYBOARD | Custom root template screen containing video represented keyboard. Can be set as a root screen. |
| GRAPHIC_WITH_TEXT | Custom root template screen containing half-screen graphic with lines of text. Can be set as a root screen. |
| TEXT_WITH_GRAPHIC | Custom root template screen containing lines of text with half-screen graphic. Can be set as a root screen. |
| TILES_ONLY | Custom root template screen containing only tiled SoftButtons. Can be set as a root screen. |

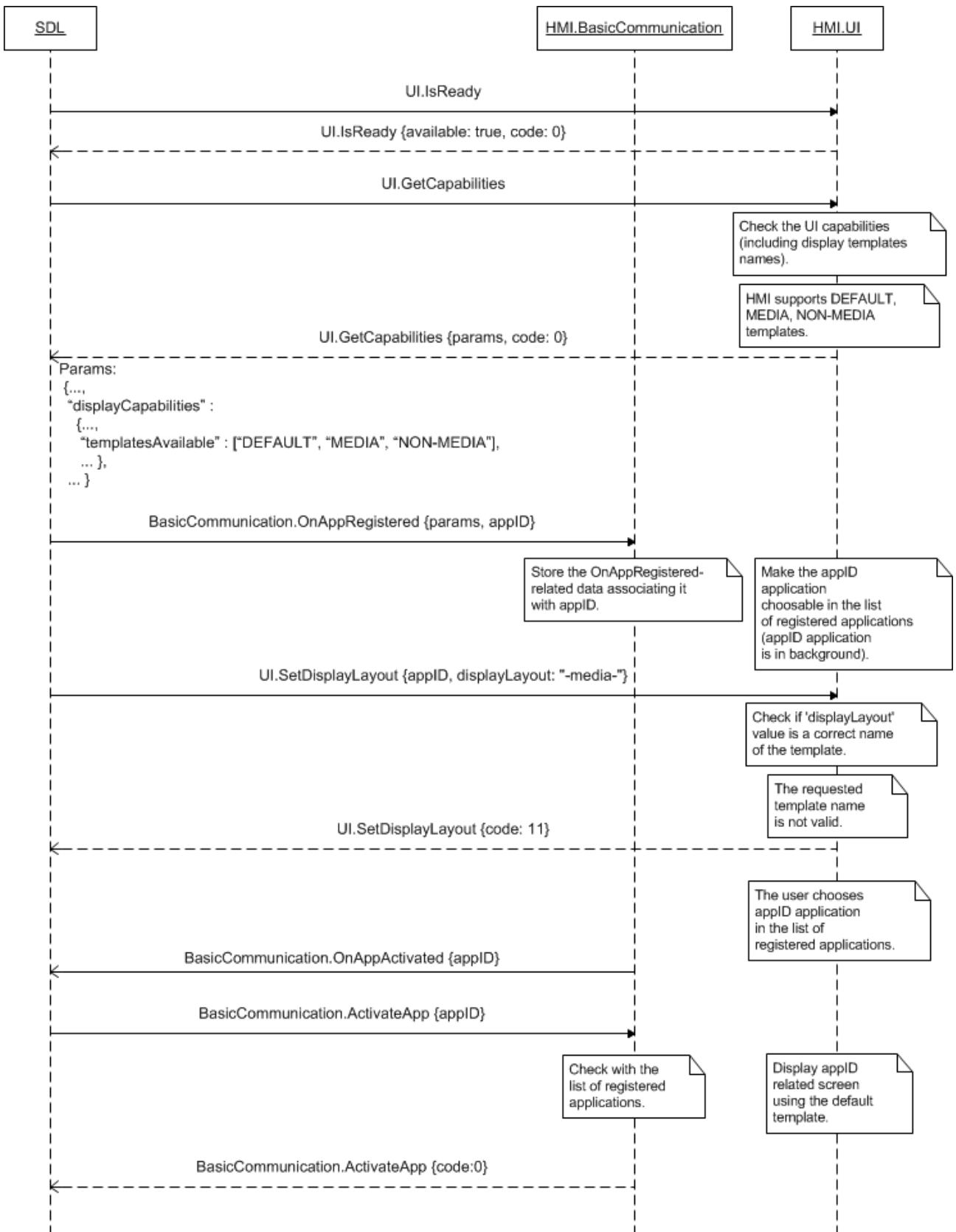
| Element name | Short Description |
|-----------------------------------|---|
| TEXTBUTTONS_ONLY | Custom root template screen containing only text SoftButtons. Can be set as a root screen. |
| GRAPHIC_WITH_TILES | Custom root template screen containing half-screen graphic with tiled SoftButtons. Can be set as a root screen. |
| TILES_WITH_GRAPHIC | Custom root template screen containing tiled SoftButtons with half-screen graphic. Can be set as a root screen. |
| GRAPHIC_WITH_TEXT_AND_SOFTBUTTONS | Custom root template screen containing half-screen graphic with text and SoftButtons. Can be set as a root screen. |
| TEXT_AND_SOFTBUTTONS_WITH_GRAPHIC | Custom root template screen containing text and SoftButtons with half-screen graphic. Can be set as a root screen. |
| GRAPHIC_WITH_TEXTBUTTONS | Custom root template screen containing half-screen graphic with text only SoftButtons. Can be set as a root screen. |
| TEXTBUTTONS_WITH_GRAPHIC | Custom root template screen containing text only SoftButtons with half-screen graphic. Can be set as a root screen. |
| LARGE_GRAPHIC_WITH_SOFTBUTTONS | Custom root template screen containing a large graphic and SoftButtons. Can be set as a root screen. |
| DOUBLE_GRAPHIC_WITH_SOFTBUTTONS | Custom root template screen containing two graphics and SoftButtons. Can be set as a root screen. |
| LARGE_GRAPHIC_ONLY | Custom root template screen containing only a large graphic. Can be set as a root screen. |

7.32.4 Sequence Diagrams

7.32.4.1 SetDisplayLayout (successful) with preceding UI.GetCapabilities



7.32.4.2 SetDisplayLayout (INVALID_DATA) with preceding UI.GetCapabilities



7.32.5 JSON Messages Examples

7.32.5.1 Request

```
{
    "id" : 47,
    "jsonrpc" : "2.0",
}
```

```

"method" : "UI.SetDisplayLayout",
"params" :
{
    "displayLayout" : "NON-MEDIA",
    "appID" : 65638
}
}

```

7.32.5.2 Response

```

{
    "id" : 47,
    "jsonrpc" : "2.0",
    "result" :
    {
        "displayCapabilities" :
        {
            "displayType" : GEN2_8_DMA,
            "textFields" : [mainField1, mainField2, alertText1,
                            alertText2, alertText3, scrollableMessageBody,
                            initialInteractionText, navigationText1,
                            navigationText2, audioPassThruDisplayText1,
                            audioPassThruDisplayText2, notificationText]
            "imageFields" :
            [
                {
                    "name" : "softButtonImage",
                    "imageTypeSupported" : ["GRAPHIC_BMP",
                                           "GRAPHIC_JPEG", "GRAPHIC_PNG"],
                    "imageResolution" :
                    {
                        "resolutionWidth" : 32,
                        "resolutionHeight" : 32
                    }
                },
                {
                    "name" : "vrHelpItem",
                    "imageTypeSupported" : ["GRAPHIC_BMP",
                                           "GRAPHIC_JPEG", "GRAPHIC_PNG"],
                    "imageResolution" :
                    {
                        "resolutionWidth" : 32,
                        "resolutionHeight" : 32
                    }
                },
                {
                    "name" : "appIcon",
                    "imageTypeSupported" : ["GRAPHIC_BMP",
                                           "GRAPHIC_JPEG", "GRAPHIC_PNG"],
                    "imageResolution" :
                    {
                        "resolutionWidth" : 64,
                        "resolutionHeight" : 64
                    }
                },
                "mediaClockFormats" : [CLOCK1, CLOCKTEXT4],
                "imageCapabilities" : [DYNAMIC],
                "graphicSupported" : true,
                "templatesAvailable" : ["DEFAULT", "MEDIA", "NON-MEDIA"],

                "screenParams" :
                {

```

```
        "resolution" :
        {
            "resolutionWidth" : 800,
            "resolutionHeight" : 480
        },

        "touchEventAvailable" :
        {
            "pressAvailable" : true,
            "multiTouchAvailable" : true,
            "doublePressAvailable" : false
        }
    },
    "numCustomPresetsAvailable" : 8
),

"buttonCapabilities" :
[
    {
        "name" : OK,
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true
    },
    {
        "name" : SEEKLEFT,
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true
    },
    {
        "name" : SEEKRIGHT,
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true
    },
    {
        "name" : TUNEUP,
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true
    },
    {
        "name" : TUNEDOWN,
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true
    },
],
"softButtonCapabilities" :
[
    {
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true,
        "imageSupported" : true
    }
],
"presetBankCapabilities" :
{
    "onScreenPresetsAvailable" : true
},
"code" : 0,
```

```

        "method" : "UI.SetDisplayLayout"
    }
}

```

7.32.5.3 Error message

```

{
    "id" : 47,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 6,
        "message" : "Ignored as the requested template is already associated with
the named appID",
        "data" :
        {
            "method" : "UI.SetDisplayLayout"
        }
    }
}

```

7.32.6 D-Bus Messages Examples

7.32.6.1 Request

```

method call          //message type (request)
sender=:1.186        //unique for-the-lifetime-of-the-bus name automatically assigned to SDL
-> dest=com.ford.sdl.hmi //the name (requested by and assigned to HMI) to receive a message
serial=89           //serial number of the message
path=/;             //the path to the object instance
interface=com.ford.sdl.hmi.UI;
member=SetDisplayLayout
    string "NON-MEDIA" //displayLayout
    int32 65638         //appId

```

7.32.6.2 Response

7.32.6.3 Failure

```

method return
sender=:1.187
-> dest=:1.186
reply_serial=89
    int32 6           //code (IGNORED)
    string "Ignored as the requested template is already associated with the
named appID"        //message

```

8 Buttons Component Description

8.1 GetCapabilities

8.1.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get the capabilities of hardware buttons. |

SDL can store the HMI capabilities in `smartDeviceLink.ini` file saved in SDL directory. Nevertheless, on its startup after getting OnReady notification from HMI, SDL sends the RPC for obtaining the buttons capabilities.

8.1.2 Request

8.1.2.1 Behavior

HMI must:

- Check the capabilities of supported hardware buttons and provide the following information about each of them:
 - Name (from the `ButtonName` enumeration).

Note:

CUSTOM_BUTTON name and the corresponding capabilities must be provided for the case when HMI has the ability to draw custom buttons onscreen.

- Whether the short/long presses (see Chapter 1 “Abbreviations and Definitions”) are supported and distinguished by HMI. (If supported, HMI must notify SDL with `OnButtonPress` of `SHORT/LONG` on every button press).
- Whether tracking the events of button being depressed and released are supported. (If supported, HMI must notify SDL with `OnButtonEvent` of `DOWN/UP` on every button depress/release).

Note:

For the mobile application better performance on HMI, HMI is recommended to provide the ability of reporting on depress/release events and at least ‘short’ press mode for its supported hardware and custom onscreen buttons.

- Check whether the ability to duplicate the hardware preset buttons with onscreen buttons is supported.
- Respond correspondingly to results of this check.

8.1.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|---------------------------|---------------|--|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the buttons capabilities. | JSON response | Method return | <code>capabilities</code> , <code>presetBankCapabilities</code> , <code>code: 0</code> | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | <u>JSON error message</u> | Method return | code: 11 | Applicable for this RPC result codes. |
| | DATA_NOT_AVAILABLE: The information about buttons capabilities cannot be provided. | | | Code: 9 | Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

8.1.3.1 Parameters

| Param name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| Param name | Type | Mandatory | Additional | Description |
|------------------------|-------------------------------|-----------|--|--|
| capabilities | Common.ButtonCapabilities | true | array = true minsize = 1 maxsize = 100 | Each element of the array defines the button name and whether short/long presses and down/up events are supported. See ButtonCapabilities |
| presetBankCapabilities | Common.PresetBankCapabilities | false | - | Should be returned if the platform supports duplicating the hardware presets with onscreen buttons. See PresetBankCapabilities |

8.1.3.2 ButtonCapabilities

| Param Name | Type | Mandatory | Description |
|---------------------|-------------------|-----------|---|
| name | Common.ButtonName | true | The name of supported/existing hardware button. See ButtonName |
| shortPressAvailable | Boolean | true | Must be 'true' if the button supports short press mode. If 'true' HMI must notify SDL with OnButtonPress (SHORT) every time this button is pressed short. See ButtonPressMode |
| longPressAvailable | Boolean | true | Must be 'true' if the button supports long press mode. If 'true' HMI must notify SDL with OnButtonPress (LONG) every time this button is pressed long. See ButtonPressMode |
| upDownAvailable | Boolean | true | Must be 'true' if tracking the events of button being depressed/released is supported. If 'true' HMI must notify SDL with OnButtonEvent of DOWN/UP on every button depress/release. See ButtonEventMode |

8.1.3.3 ButtonName

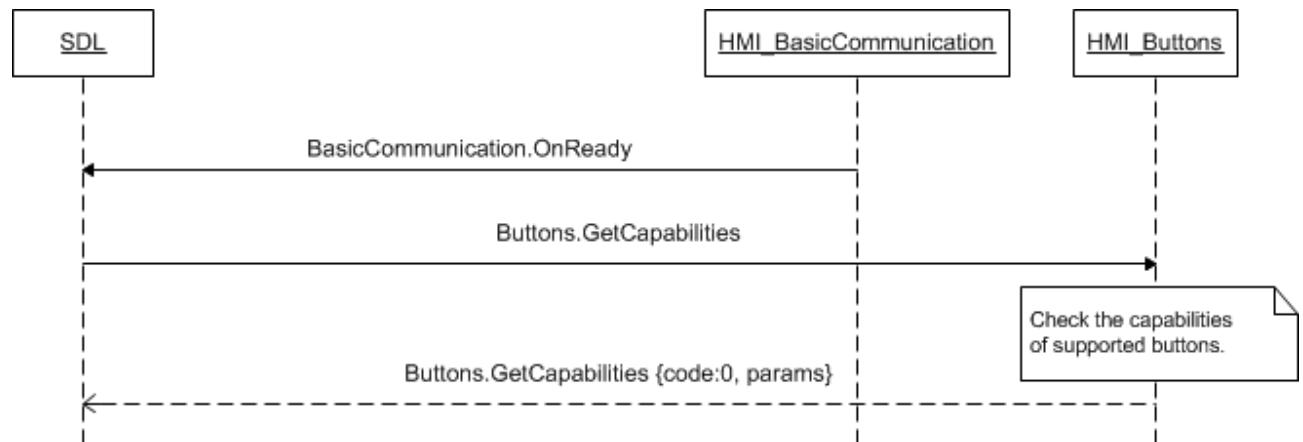
| Element name | Value | Short Description |
|---------------|-------|---|
| OK | 0 | Represents the button usually labeled "OK". A typical use of this button is for the User to press it to make a selection. |
| SEEKLEFT | 1 | Represents the seek-left button. A typical use of this button is for the user to scroll to the left through menu choices, one menu item per press. |
| SEEKRIGHT | 2 | Represents the seek-right button. A typical use of this button is for the user to scroll to the right through menu choices one menu item per press. |
| TUNEUP | 3 | Represents a turn of the tuner knob in the clockwise direction one tick. |
| TUNEDOWN | 4 | Represents a turn of the tuner knob in the counter-clockwise direction one tick. |
| PRESET_0 | 5 | Represents the preset 0 button. |
| PRESET_1 | 6 | Represents the preset 1 button. |
| PRESET_2 | 7 | Represents the preset 2 button. |
| PRESET_3 | 8 | Represents the preset 3 button. |
| PRESET_4 | 9 | Represents the preset 4 button. |
| PRESET_5 | 10 | Represents the preset 5 button. |
| PRESET_6 | 11 | Represents the preset 6 button. |
| PRESET_7 | 12 | Represents the preset 7 button. |
| PRESET_8 | 13 | Represents the preset 8 button. |
| PRESET_9 | 14 | Represents the preset 9 button. |
| CUSTOM_BUTTON | 15 | Represents any of onscreen buttons requested by SDL. |
| SEARCH | 16 | Represents the 'Search' button on the touchscreen keyboard. |

8.1.3.4 PresetBankCapabilities

| Param Name | Type | Mandatory | Description |
|--------------------------|---------|-----------|---|
| onScreenPresetsAvailable | Boolean | true | Must be 'true' if HMI supports duplicating the hardware custom presets with onscreen buttons. |

8.1.4 Sequence Diagrams

8.1.4.1 GetCapabilities on system startup



8.1.5 JSON Messages Examples

8.1.5.1 Request

```
{
  "id" : 20,
  "jsonrpc" : "2.0",
  "method" : "Buttons.GetCapabilities"
}
```

8.1.5.2 Response

```
{
  "id" : 20,
  "jsonrpc" : "2.0",
  "result" :
  {
    "capabilities" :
    [
      {
        "name" : OK,
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true
      },
      {
        "name" : SEEKLEFT,
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true
      },
      {
        "name" : SEEKRIGHT,
        "shortPressAvailable" : true,
        "longPressAvailable" : true,
        "upDownAvailable" : true
      }
    ]
  }
}
```

```

        },
        {
            "name" : TUNEUP,
            "shortPressAvailable" : true,
            "longPressAvailable" : true,
            "upDownAvailable" : true
        },
        {
            "name" : TUNEDOWN,
            "shortPressAvailable" : true,
            "longPressAvailable" : true,
            "upDownAvailable" : true
        },
    ],
    "presetBankCapabilities" :
    [
        "onScreenPresetsAvailable" : true
    ],
    "code" : 0,
    "method" : "Buttons.GetCapabilities"
}
}

```

8.1.5.3 Error message

```

{
    "id" : 20,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 9,
        "message" : "The requested data is not available",
        "data" :
        {
            "method" : "Buttons.GetCapabilities"
        }
    }
}

```

8.1.6 D-Bus Messages Examples

8.1.6.1 Request

[REDACTED]

8.1.6.2 Response

[REDACTED]

8.1.6.3 Failure

[REDACTED]

8.2 OnButtonEvent

8.2.1 Description

| | |
|-----------------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the event of depress/release has occurred for the button. |

Once HMI confirmed it supports reporting on events of button press/release for the named buttons (via response to `Buttons.GetCapabilities`), SDL expects to be notified about such events via `OnButtonEvent`.

HMI must:

Send the following information upon every event of press or release occurred for each of supported buttons:

- The name of the button (from `ButtonName` enumeration) the event has occurred for.
- The event occurred:
 - `BUTTONDOWN` – when the User has pressed the button
 - `BUTTONUP` – when the User has released the button.
- The ID of the custom button: for the case of `CUSTOM_BUTTON` name only.
- The `appId` in case the `ButtonName` is `CUSTOM_BUTTON` (if not sent, SDL will ignore the notification and will not transfer it to mobile app).

Note:

The value of custom button ID is provided by SDL within `softButton` structure (`softButtonID` parameter) via one of the following RPCs: `UI.Alert`, `UI.ScrollableMessage`, `UI.Show`.

8.2.1.1 Parameters

| Param name | Type | Mandatory | Additional | Description |
|-----------------------------|-------------------------------------|--------------------|--|--|
| <code>name</code> | <code>Common.ButtonName</code> | <code>true</code> | – | The name of the button involved into event. See <code>ButtonName</code> . |
| <code>mode</code> | <code>Common.ButtonEventMode</code> | <code>true</code> | – | Indicates whether this is an UP or DOWN event. See <code>ButtonEventMode</code> . |
| <code>customButtonID</code> | <code>Integer</code> | <code>false</code> | <code>minvalue = 0</code> <code>maxvalue = 65536</code> | This ID must be provided in case the event has occurred for the <code>CUSTOM_BUTTON</code> only. The value is previously provided by SDL within <code>softButton</code> structure via one of the following RPCs: <code>UI.Alert</code> , <code>UI.ScrollableMessage</code> , <code>UI.Show</code> . |
| <code>appId</code> | <code>Integer</code> | <code>false</code> | – | In case the <code>ButtonName</code> is <code>CUSTOM_BUTTON</code> , HMI must include <code>appId</code> parameters to <code>OnButtonPress</code> notification sent to SDL. Otherwise, if <code>appId</code> is not sent together with <code>CUSTOM_BUTTON</code> , this notification will be ignored by SDL. |

8.1.3.3 ButtonName

| Element name | Value | Short Description |
|------------------------|-------|---|
| <code>OK</code> | 0 | Represents the button usually labeled "OK". A typical use of this button is for the User to press it to make a selection. |
| <code>SEEKLEFT</code> | 1 | Represents the seek-left button. A typical use of this button is for the user to scroll to the left through menu choices, one menu item per press. |
| <code>SEEKRIGHT</code> | 2 | Represents the seek-right button. A typical use of this button is for the user to scroll to the right through menu choices one menu item per press. |
| <code>TUNEUP</code> | 3 | Represents a turn of the tuner knob in the clockwise direction one tick. |
| <code>TUNEDOWN</code> | 4 | Represents a turn of the tuner knob in the counter-clockwise direction one tick. |
| <code>PRESET_0</code> | 5 | Represents the preset 0 button. |
| <code>PRESET_1</code> | 6 | Represents the preset 1 button. |
| <code>PRESET_2</code> | 7 | Represents the preset 2 button. |
| <code>PRESET_3</code> | 8 | Represents the preset 3 button. |
| <code>PRESET_4</code> | 9 | Represents the preset 4 button. |
| <code>PRESET_5</code> | 10 | Represents the preset 5 button. |

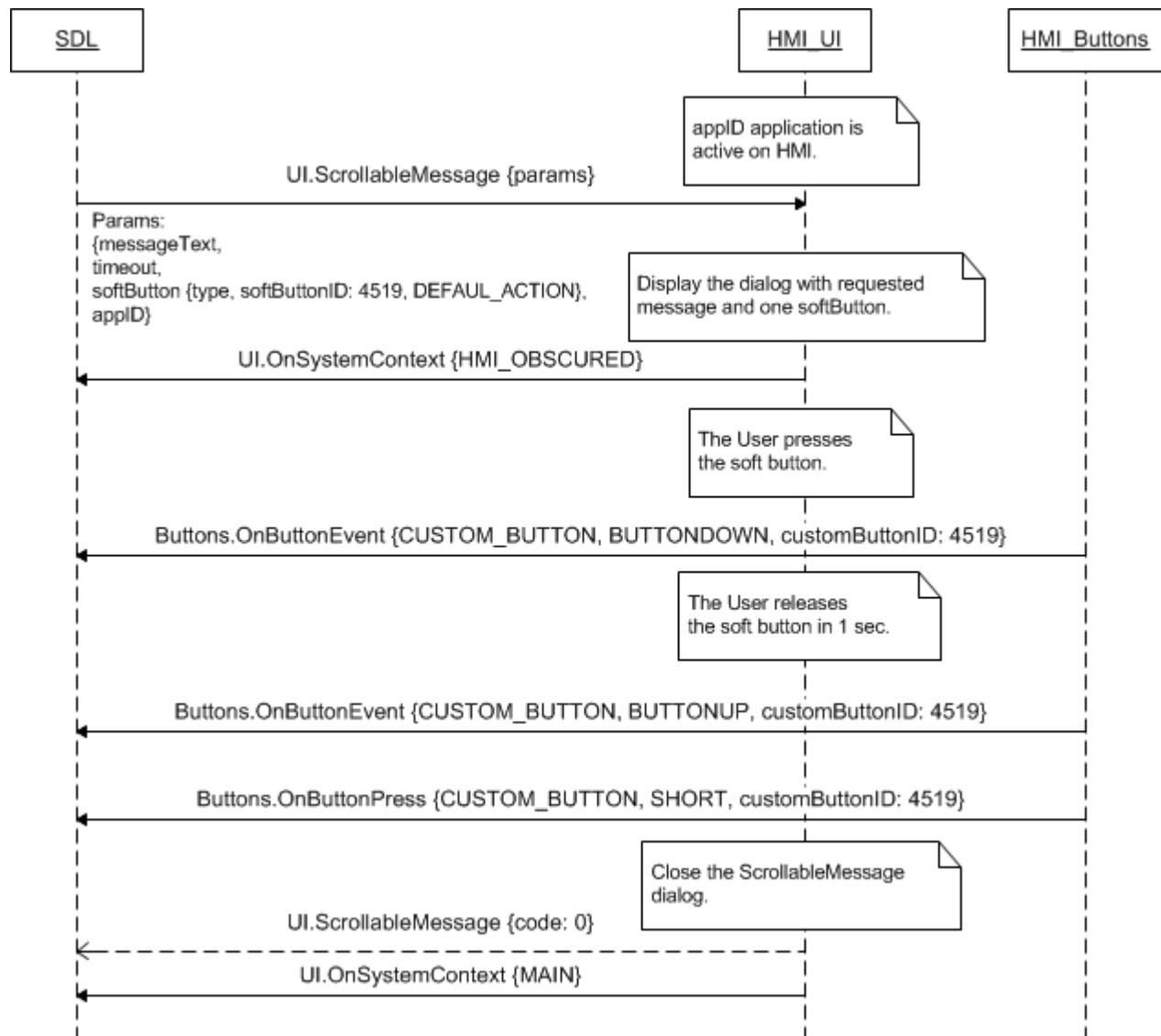
| Element name | Value | Short Description |
|---------------|-------|---|
| PRESET_6 | 11 | Represents the preset 6 button. |
| PRESET_7 | 12 | Represents the preset 7 button. |
| PRESET_8 | 13 | Represents the preset 8 button. |
| PRESET_9 | 14 | Represents the preset 9 button. |
| CUSTOM_BUTTON | 15 | Represents any of onscreen buttons requested by SDL. |
| SEARCH | 16 | Represents the 'Search' button on the touchscreen keyboard. |

8.2.1.2 ButtonEventMode

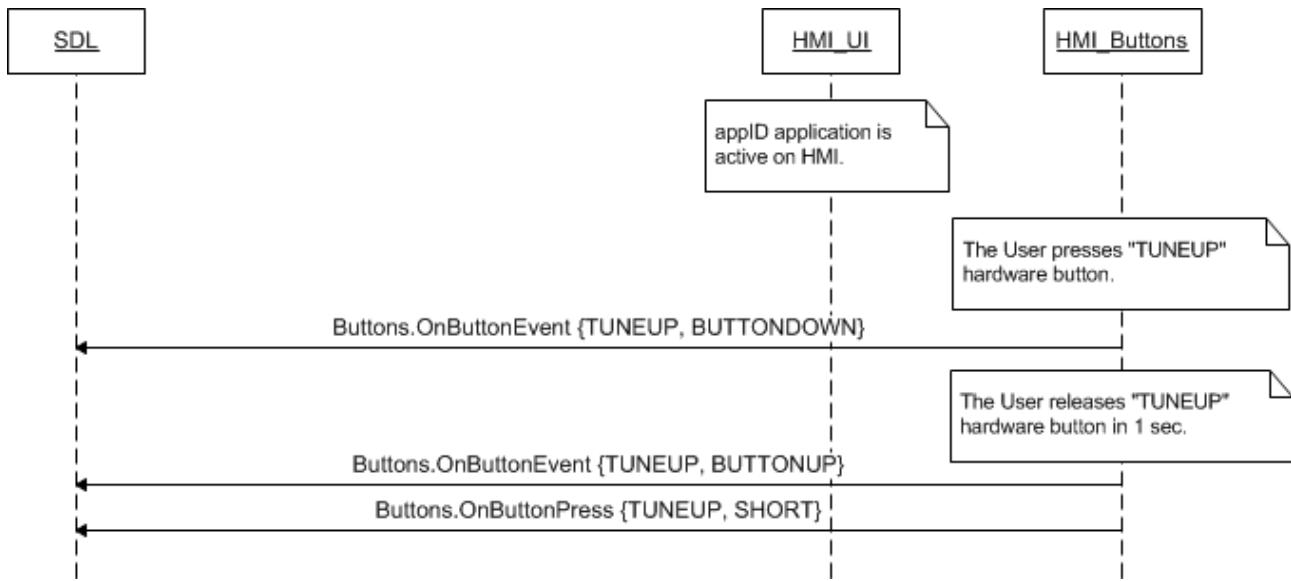
| Element name | Value | Short Description |
|--------------|-------|------------------------------|
| BUTTONDOWN | 0 | The button has been pressed |
| BUTTONUP | 1 | The button has been released |

8.2.2 Sequence Diagrams

8.2.2.1 OnButtonEvent for CUSTOM_BUTTON being pressed and released



8.2.2.1 OnButtonEvent for hardware button being pressed and released



8.2.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "Buttons.OnButtonEvent",
    "params" :
    {
        "name" : OK,
        "mode" : BUTTONDOWN,
    }
}
```

8.2.4 D-Bus Messages Examples

8.3 OnButtonPress

8.3.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the press mode detected for the button. |

Once HMI confirmed it supports short and/or long press modes for the named buttons (via response to `Buttons.GetCapabilities`), SDL expects to be notified with the information of the press mode occurred via `OnButtonPress`.

HMI must:

Send the following information upon every long or short press detected for each of supported buttons:

- The name of the button (from `ButtonName` enumeration) being pressed.
- The press mode detected:
 - SHORT – when the button is pressed and is being held during less than HMI-defined time threshold (e.g. during less than 2 sec.) and then released
 - LONG – when the button is pressed and is being held during more than HMI-defined time threshold (e.g. during more than 2 sec.) and then released
- The ID of the custom button: for the case of `CUSTOM_BUTTON` name only.
- The `appId` in case the `ButtonName` is `CUSTOM_BUTTON` (if not sent, SDL will ignore the notification and will not transfer it to mobile app).

Note:

The value of custom button ID is provided by SDL within `softButton` structure (`softButtonID` parameter) via one of the following RPCs: `UI.Alert`, `UI.ScrollableMessage`, `UI.Show`.

Important note:

It is expected that the sequence of `OnButtonEvent` and `OnButtonPress` notifications will depend on the button press mode detected:

- 1) The button is pressed **SHORT** (is held during less than HMI-defined time threshold):

- `OnButtonEvent (BUTTONDOWN)`
- `OnButtonEvent (BUTTONUP)`
- `OnButtonPress (SHORT)`

- 2) The button is pressed **LONG** (is held during more than HMI-defined time threshold):

- `OnButtonEvent (BUTTONDOWN)`
- `OnButtonPress (LONG)`
- `OnButtonEvent (BUTTONUP)`

Note:

In case the button supports **SHORT** press mode only, HMI must send `OnButtonPress (SHORT)` independently of the time this button is being held for.

8.3.1.1 Parameters

| Param name | Type | Mandatory | Additional | Description |
|----------------|------------------------|-----------|----------------------------------|---|
| name | Common.ButtonName | true | – | The name of the button the press mode is detected for. See <code>ButtonName</code> . |
| mode | Common.ButtonPressMode | true | – | Indicates whether this is a LONG or SHORT button press mode See <code>ButtonPressMode</code> . |
| customButtonID | Integer | false | minvalue = 0 maxvalue = 65536 | This ID must be provided in case the event has occurred for the CUSTOM_BUTTON only. |
| appID | Integer | false | – | In case the <code>ButtonName</code> is CUSTOM_BUTTON, HMI must include <code>appID</code> parameters to <code>OnButtonPress</code> notification sent to SDL. Otherwise, if <code>appID</code> is not sent together with CUSTOM_BUTTON, this notification will be ignored by SDL. |

8.1.3.3 ButtonName

| Element name | Value | Short Description |
|--------------|-------|---|
| OK | 0 | Represents the button usually labeled "OK". A typical use of this button is for the User to press it to make a selection. |
| SEEKLEFT | 1 | Represents the seek-left button. A typical use of this button is for the user to scroll to the left through menu choices, one menu item per press. |
| SEEKRIGHT | 2 | Represents the seek-right button. A typical use of this button is for the user to scroll to the right through menu choices one menu item per press. |
| TUNEUP | 3 | Represents a turn of the tuner knob in the clockwise direction one tick. |
| TUNEDOWN | 4 | Represents a turn of the tuner knob in the counter-clockwise direction one tick. |
| PRESET_0 | 5 | Represents the preset 0 button. |
| PRESET_1 | 6 | Represents the preset 1 button. |
| PRESET_2 | 7 | Represents the preset 2 button. |

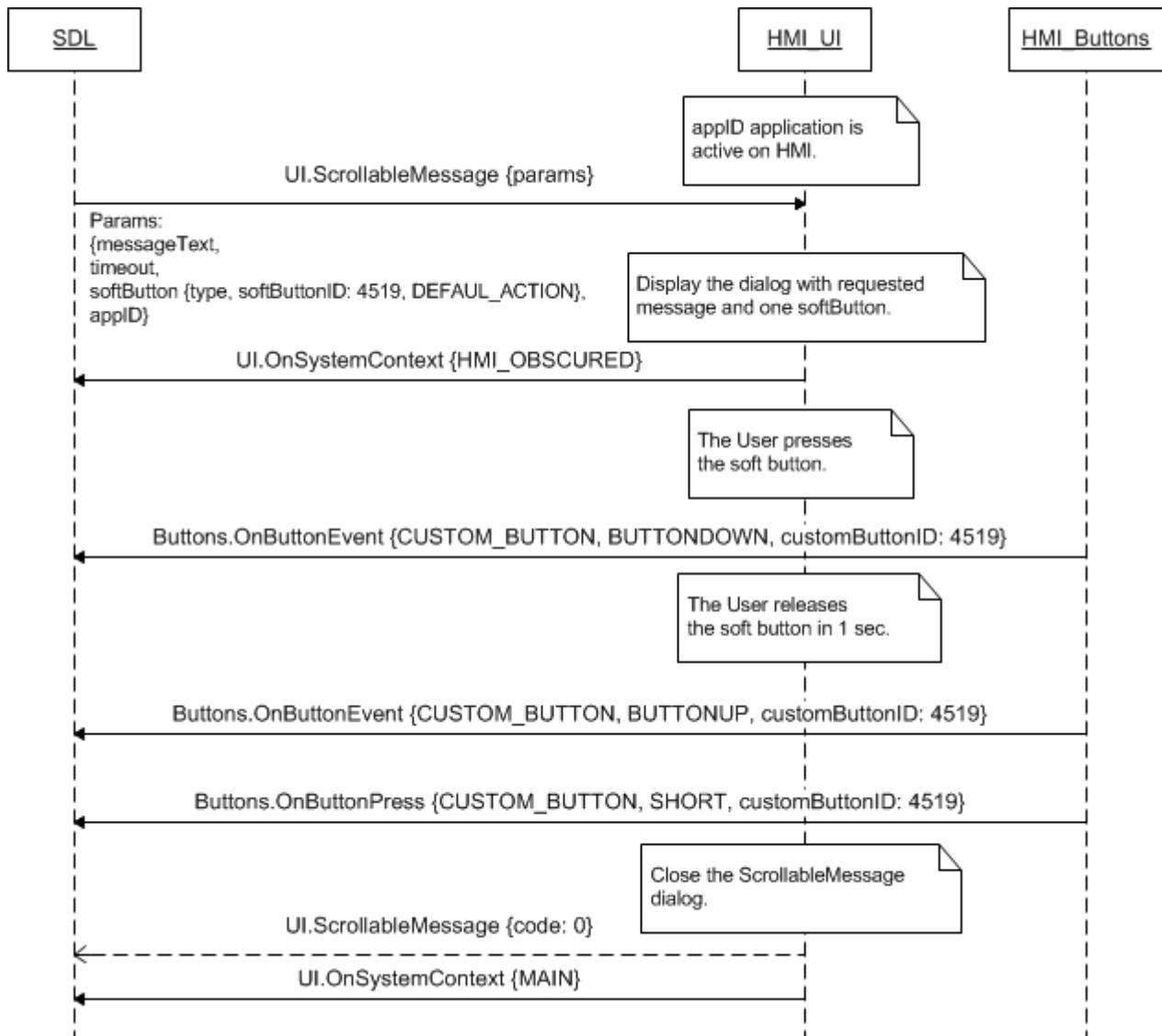
| Element name | Value | Short Description |
|---------------|-------|---|
| PRESET_3 | 8 | Represents the preset 3 button. |
| PRESET_4 | 9 | Represents the preset 4 button. |
| PRESET_5 | 10 | Represents the preset 5 button. |
| PRESET_6 | 11 | Represents the preset 6 button. |
| PRESET_7 | 12 | Represents the preset 7 button. |
| PRESET_8 | 13 | Represents the preset 8 button. |
| PRESET_9 | 14 | Represents the preset 9 button. |
| CUSTOM_BUTTON | 15 | Represents any of onscreen buttons requested by SDL. |
| SEARCH | 16 | Represents the 'Search' button on the touchscreen keyboard. |

8.3.1.2 ButtonPressMode

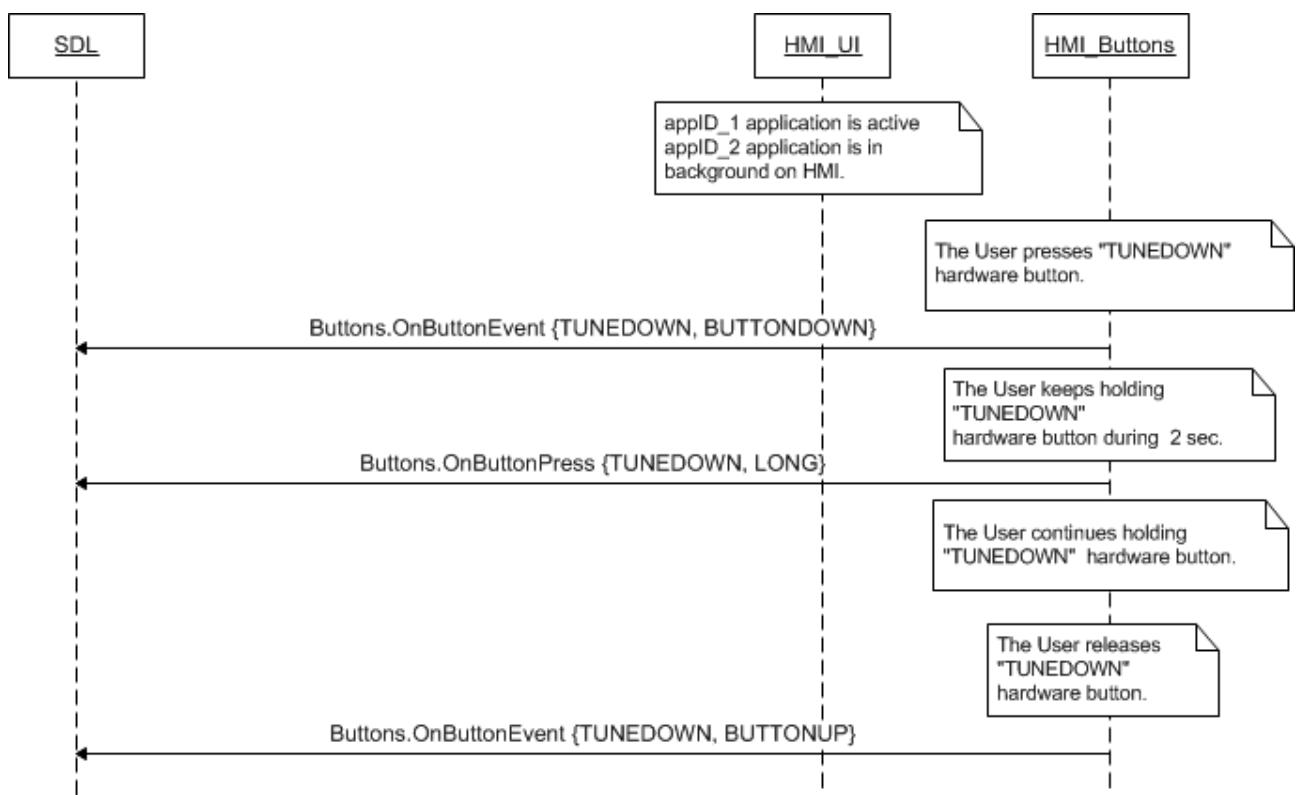
| Element name | Value | Short Description |
|--------------|-------|--|
| SHORT | 0 | Short-time button press: the button is being held during less than HMI-defined time threshold (e.g. during less than 2 sec.) and then released |
| LONG | 1 | Long-time button press: the button is being held during more than HMI-defined time threshold (e.g. during more than 2 sec.) and then released |

8.3.2 Sequence Diagrams

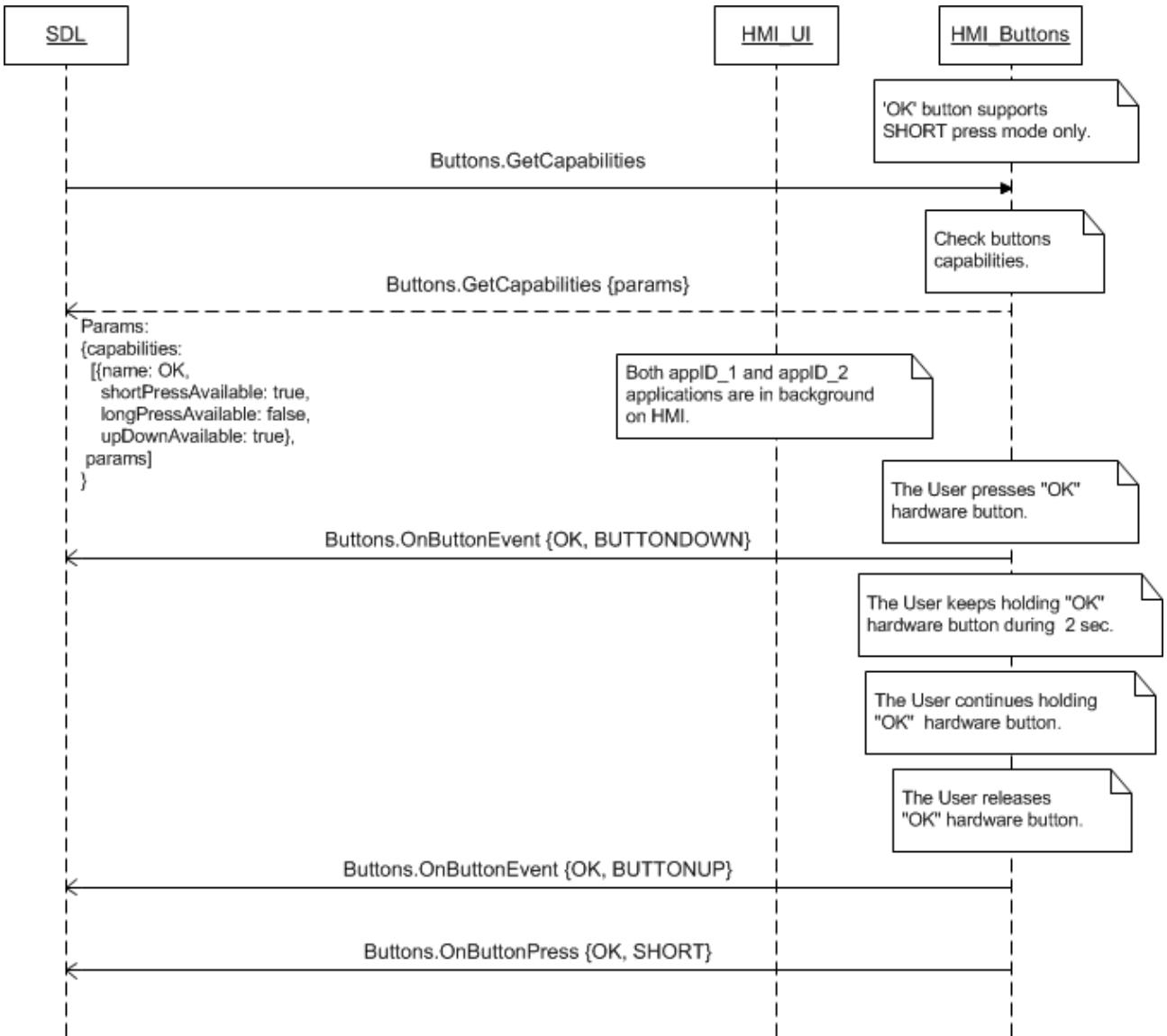
8.3.2.1 OnButtonPress (SHORT) for CUSTOM_BUTTON



8.3.2.2 OnButtonPress (LONG) for hardware button



8.3.2.2 OnButtonPress for hardware button that supports SHORT press mode only



8.3.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "Buttons.OnButtonPress",
    "params" :
    {
        "name" : CUSTOM_BUTTON,
        "mode" : SHORT,
        "customButtonID" : 564
    }
}
```

8.3.4 D-Bus Messages Examples

9 VR Component Description

9.1 IsReady

9.1.1 Description

| | |
|----------|-----------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Know if UI module is ready. |

The request comes after HMI's readiness is confirmed via [OnReady](#) notification. SDL requires the information about whether the voice recognition module is physically present on HU and if so whether it is working.

Note:

If VR module is responded to be unavailable or IsReady request is not responded at all, SDL will not further send any requests related to it.

9.1.2 Request

7.1.2.1 Behavior

HMI must:

- Check whether VR module is present and ready
- Respond correspondingly to results of this check.

9.1.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

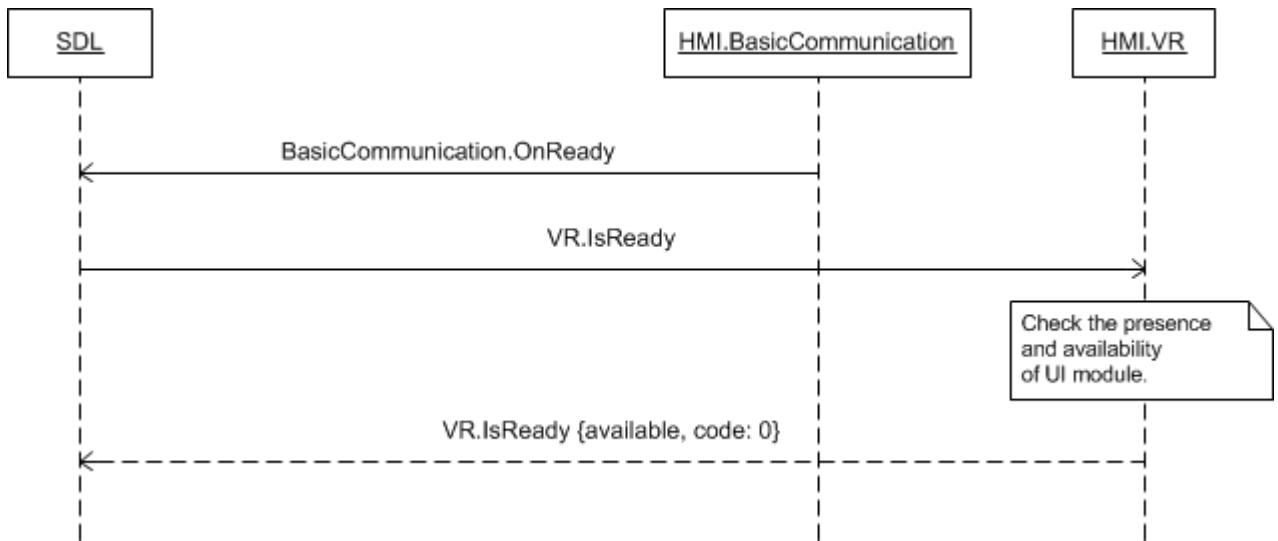
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|-----------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the information about VR module availability. | JSON response | Method return | available, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | DATA_NOT_AVAILABLE: The information about VR module availability cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

9.1.3.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|---|
| availabe | Boolean | true | Must be - 'true' if UI is present and ready - 'false' if not. |

9.1.4 Sequence Diagrams

9.1.4.1 VR.IsReady



9.1.5 JSON Messages Examples

9.1.5.1 Request

```
{
    "id" : 45,
    "jsonrpc" : "2.0",
    "method" : "VR.IsReady"
}
```

9.1.5.2 Response

```
{
    "id" : 45,
    "jsonrpc" : "2.0",
    "result" :
    {
        "available" : true,
        "code" : 0,
        "method" : "VR.IsReady"
    }
}
```

9.1.5.3 Error message

```
{
    "id" : 45,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "Invalid data",
        "data" :
        {
            "method" : "VR.IsReady"
        }
    }
}
```

9.1.6 D-Bus Messages Examples

9.1.6.1 Request

9.1.6.2 Response

| |
|--|
| |
|--|

9.1.6.3 Failure

| |
|--|
| |
|--|

9.2 GetCapabilities

9.2.1 Description

| | |
|----------|-------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get capabilities of VR. |

SDL can store the HMI capabilities in `smartDeviceLink.ini` file saved in SDL directory. Nevertheless, on its startup after getting OnReady notification and response to VR.IsReady from HMI, SDL sends this RPC for obtaining the capabilities of VR.

9.2.2 Request

9.2.2.1 Behavior

HMI must:

1. Check the VR capabilities.
2. Respond correspondingly to results of this check.

Note:

Currently only `TEXT` is SDL-supported capability of VR.

9.2.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the VR capabilities. | JSON response | Method return | vrCapabilities, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | DATA_NOT_AVAILABLE: The information about VR capabilities cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

9.2.3.1 Parameters

| Param name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

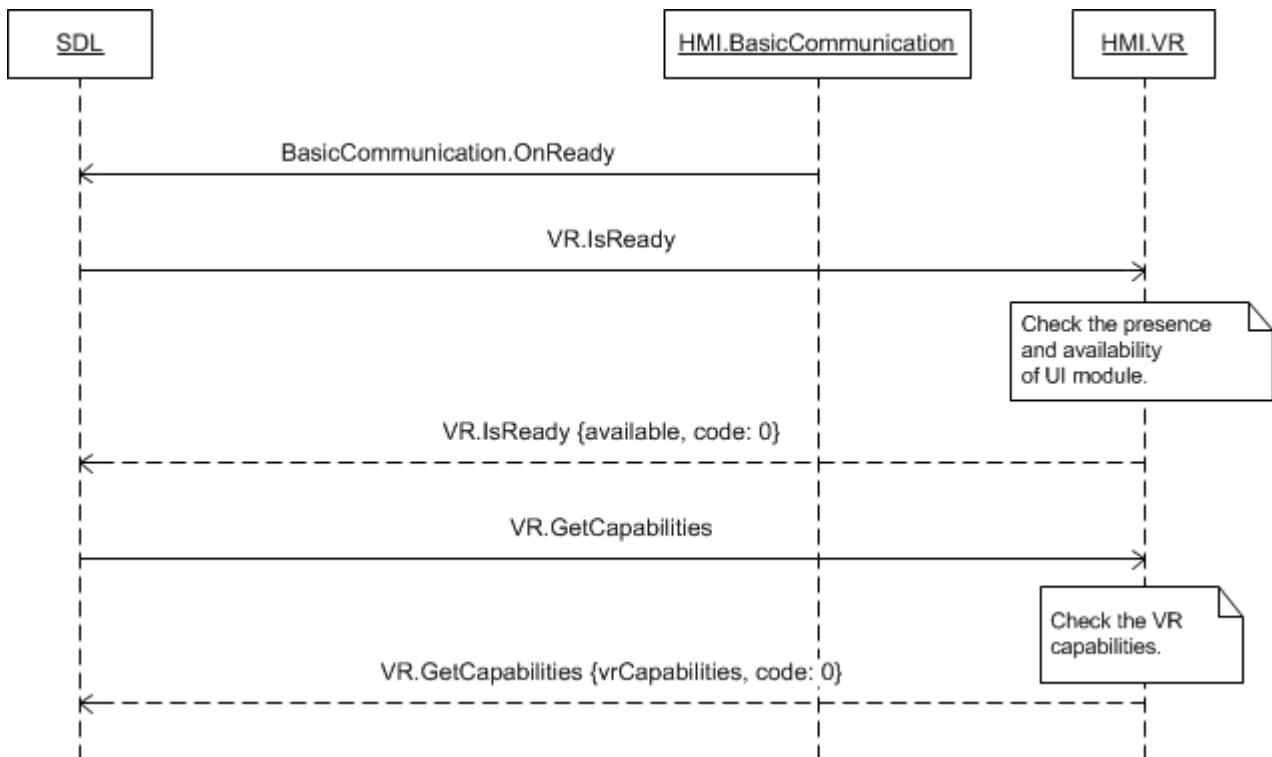
| Param name | Type | Mandatory | Additional | Description |
|----------------|-----------------------|-----------|--|---|
| vrCapabilities | Common.VrCapabilities | false | Array = true minsize = 1 maxsize = 100 | Types of input recognized by VR module. |

9.2.3.2 VrCapabilities

| Element name | Short Description |
|--------------|-------------------|
| TEXT | Text is supported |

9.2.4 Sequence Diagrams

9.2.4.1 VR.GetCapabilities



9.2.5 JSON Messages Examples

9.2.5.1 Request

```
{
  "id" : 9,
  "jsonrpc" : "2.0",
  "method" : "VR.GetCapabilities"
}
```

9.2.5.2 Response

```
{
  "id" : 9,
  "jsonrpc" : "2.0",
  "result" :
  {
    "vrCapabilities" : [TEXT],
    "code" : 0,
  }
}
```

```

        "method" : "VR.GetCapabilities"
    }
}

```

9.2.5.3 Error message

```

{
    "id" : 9,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "The data sent is invalid",
        "data" :
        {
            "method" : "VR.GetCapabilities"
        }
    }
}

```

9.2.6 D-Bus Messages Examples

9.2.6.1 Request

9.2.6.2 Response

9.2.6.3 Failure

9.3 GetSupportedLanguages

9.3.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get the list of languages supported for VR |

Once VR module is confirmed to be ready (via response to [IsReady](#) RPC) SDL starts discovering its capabilities via [GetCapabilities](#) and [GetSupportedLanguages](#) RPCs.

Response to [VR.GetSupportedLanguages](#) is assumed to bring the information about what languages are supported for voice recognition by VR module. Having obtained this information SDL will monitor the language parameter within RPCs from mobile application(s) and reject the requests containing language not supported by HMI.

Note:

The list of languages recognized by SDL is provided in the section 9.3.3.2 Language Enumeration.

9.3.2 Request

7.1.2.1 Behavior

HMI must:

- Check the VR supported languages
- Respond correspondingly to results of this check.

9.3.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the VR supported languages. | JSON response | Method return | language, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | DATA_NOT_AVAILABLE: The information about VR supported languages cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

9.3.3.1 Parameters

| Param name | Type | Mandatory | Additional | Description |
|------------|-----------------|-----------|--|---|
| languages | Common.Language | true | Array = true minsize = 1 maxsize = 100 | List of languages supported in VR. See Language. |

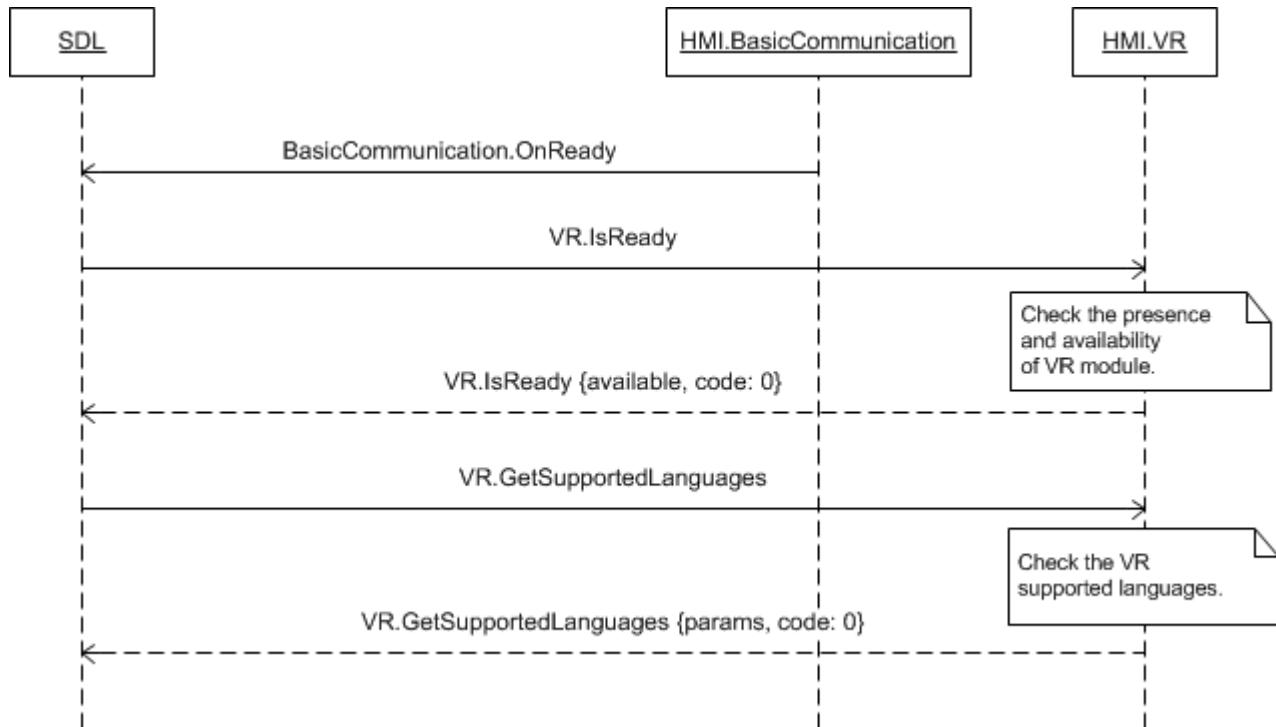
9.3.3.2 Language Enumeration

| Element name | Short Description |
|--------------|--------------------------------|
| AR-SA | Arabic – Saudi Arabia |
| CS-CZ | Czech – Czech Republic |
| DA-DK | Danish – Denmark |
| DE-DE | German – Germany |
| EN-AU | English – Australia |
| EN-GB | English – GB |
| EN-US | English – US |
| ES-ES | Spanish – Spain |
| ES-MX | Spanish – Mexico |
| FR-CA | French – Canada |
| FR-FR | French – France |
| IT-IT | Italian – Italy |
| JA-JP | Japanese – Japan |
| KO-KR | Korean – South Korea |
| NL-NL | Dutch (Standard) – Netherlands |
| NO-NO | Norwegian - Norway |
| PL-PL | Polish – Poland |
| PT-PT | Portuguese – Portugal |
| PT-BR | Portuguese – Brazil |
| RU-RU | Russian - Russia |
| SV-SE | Swedish – Sweden |

| Element name | Short Description |
|--------------|-------------------|
| TR-TR | Turkish – Turkey |
| ZH-CN | Mandarin – China |
| ZH-TW | Mandarin – Taiwan |

9.3.4 Sequence Diagrams

9.3.4.1 VR.GetSupportedLanguages



9.3.5 JSON Messages Examples

9.3.5.1 Request

```
{
    "id" : 18,
    "jsonrpc" : "2.0",
    "method" : "VR.GetSupportedLanguages"
}
```

9.3.5.2 Response

```
{
    "id" : 18,
    "jsonrpc" : "2.0",
    "result" :
    {
        "languages" : [AR-SA, DE-DE, EN-GB, EN-US, ES-ES, FR-FR, IT-IT],
        "code" : 0,
        "method" : "VR.GetSupportedLanguages"
    }
}
```

9.3.5.3 Error message

```
{
    "id" : 18,
    "jsonrpc" : "2.0",
```

```

"error" :
{
  "code" : 11,
  "message" : "The data sent is invalid",
  "data" :
  {
    "method" : "VR.GetSupportedLanguages"
  }
}

```

9.3.6 D-Bus Messages Examples

9.3.6.1 Request

9.3.6.2 Response

9.3.6.3 Failure

9.4 AddCommand

9.4.1 Description

| | |
|----------|--------------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Add a command for voice recognition. |

SDL requests to add a command for voice recognition.

9.4.2 Request

9.4.2.1 Behavior

HMI must:

1. Store the provided values of cmdID and corresponding vrCommands.
2. Keep these values stored until VR.DeleteCommand or BasicCommunication.OnAppUnregistered RPC comes from SDL.
3. Add the vrCommands to voice recognition system and make them accessible in the following layers:
 - If appId parameter is present in request: HMI must use the vrCommands in the layer of the appId application (i.e. when it is active on HMI).
 - If appId parameter is not present in request: HMI must use the vrCommands in all layers.

Important note:

HMI must return this cmdId via VR.OnCommand when the corresponding synonym is recognized.

9.4.2.2 Parameters

| Param name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|--------------|--|
| cmdId | Integer | true | minvalue = 0 | Unique ID that identifies the command. Must be returned in the OnCommand notification to identify the command selected by the User. |

| Param name | Type | Mandatory | Additional | Description |
|------------|-----------------------|-----------|--|--|
| vrCommands | String | true | minsize = 1 maxsize = 100 maxlength = 99 array = true | An array of strings to be used as VR synonyms for this command. Defines one or more VR phrases the recognition of any of which must trigger the <code>OnCommand</code> notification with the above cmdID. |
| type | Common.VRCOmmmandType | true | - | Type of added command. See <code>VRCommandType</code> . |
| grammarID | Integer | true | minvalue = 0 maxvalue = 2000000000 | ID of the specific grammar, whether top-level or choice set. |
| appID | Integer | false | - | ID of the application related to this RPC. |

9.4.2.3 VRCommandType

| Element name | Vlaue | Short Description |
|--------------|-------|---|
| Choice | 0 | The VR command must be used in VR.PerformInteraction that arrived with the corresponding grammarID and must NOT be accessible until then. |
| Command | 1 | The VR command must be accessible for the User upon VR activation while the application of the corresponding appID is active on HMI. |

9.4.3 Response

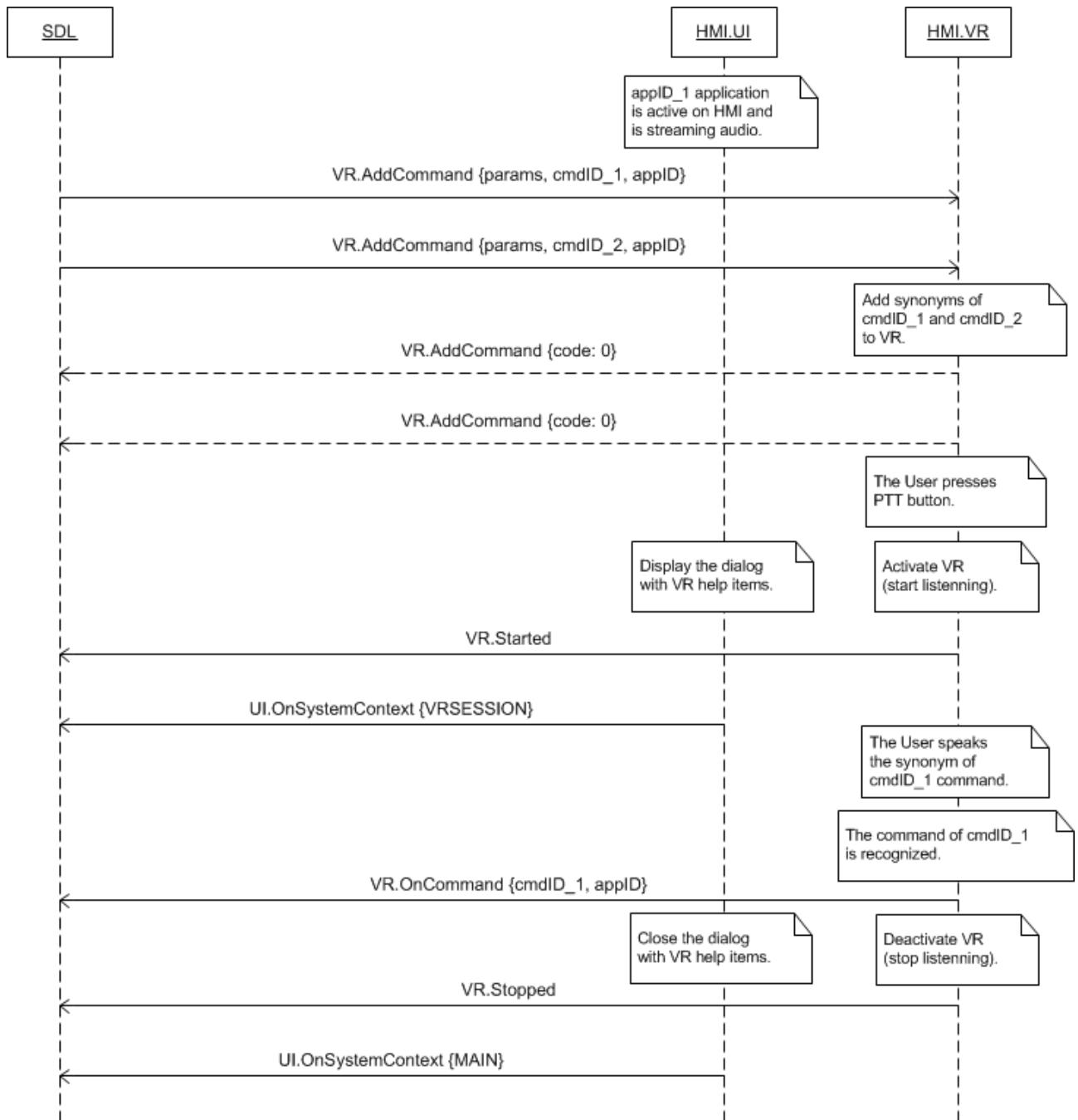
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: The command with requested parameters has been added for voice recognition for the named application. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

9.4.4 Sequence Diagrams

9.4.4.1 VR.AddCommand



9.4.5 JSON Messages Examples

9.4.5.1 Request

```
{
    "id" : 119,
    "jsonrpc" : "2.0",
    "method" : "VR.AddCommand",
    "params" :
    {
        "cmdID" : 4365,
        "vrCommands" :
        [
            "Leave",
            "Exit",
            "Quit"
        ],
        "appId" : 64467
    }
}
```

```
}
```

9.4.5.2 Response

```
{
    "id" : 119,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "VR.AddCommand"
    }
}
```

9.4.5.3 Error message

```
{
    "id" : 119,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 13,
        "message" : "One of the provided IDs is not valid",
        "data" :
        {
            "method" : "VR.AddCommand"
        }
    }
}
```

9.4.6 D-Bus Messages Examples

9.4.6.1 Request

9.4.6.2 Response

9.4.6.3 Failure

9.5 DeleteCommand

9.5.1 Description

| | |
|----------|---------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Delete a command from VR. |

SDL requests to delete the command from in-application menu or sub menu of the named application previously added via UI.AddCommand.

The request may arrive in both cases of activated and deactivated application on HMI.

9.5.2 Request

9.5.2.1 Behavior

HMI must:

1. Update the named application stored data correspondingly to the RPC arrived.
2. Delete the command identified with cmdID.
3. Display updates:
 - Right away if the named application is active and the corresponding menu/sub menu is open on UI.
 - After the corresponding menu/sub menu is opened on UI upon User's request:
 - If the RPC arrived when the named application was active and had another menu or persistent display visible on UI
 - If the RPC arrived when the named application was not active on HMI.
5. Provide the response corresponding to the result of RPC execution.

Note:

The applicable to this RPC result codes are provided in section 7.9.3 Response.

Note:

- The value of cmdID is previously sent via UI.AddCommand.
- The value of appID is previously sent via UpdateAppList or OnAppRegistered.

9.5.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|----------------------|-----------|---------------------------------------|--|
| cmdID | Integer | true | - | ID that identifies the command to be deleted (sent by AddCommand). |
| type | Common.VRCommandType | true | - | Type of added command. See VRCommandType. |
| grammarID | Integer | true | minvalue = 0 maxvalue = 2000000000 | ID of the specific grammar, whether top-level or choice set. |
| appID | Integer | true | - | ID of application that concerns this RPC. |

9.5.3 Response

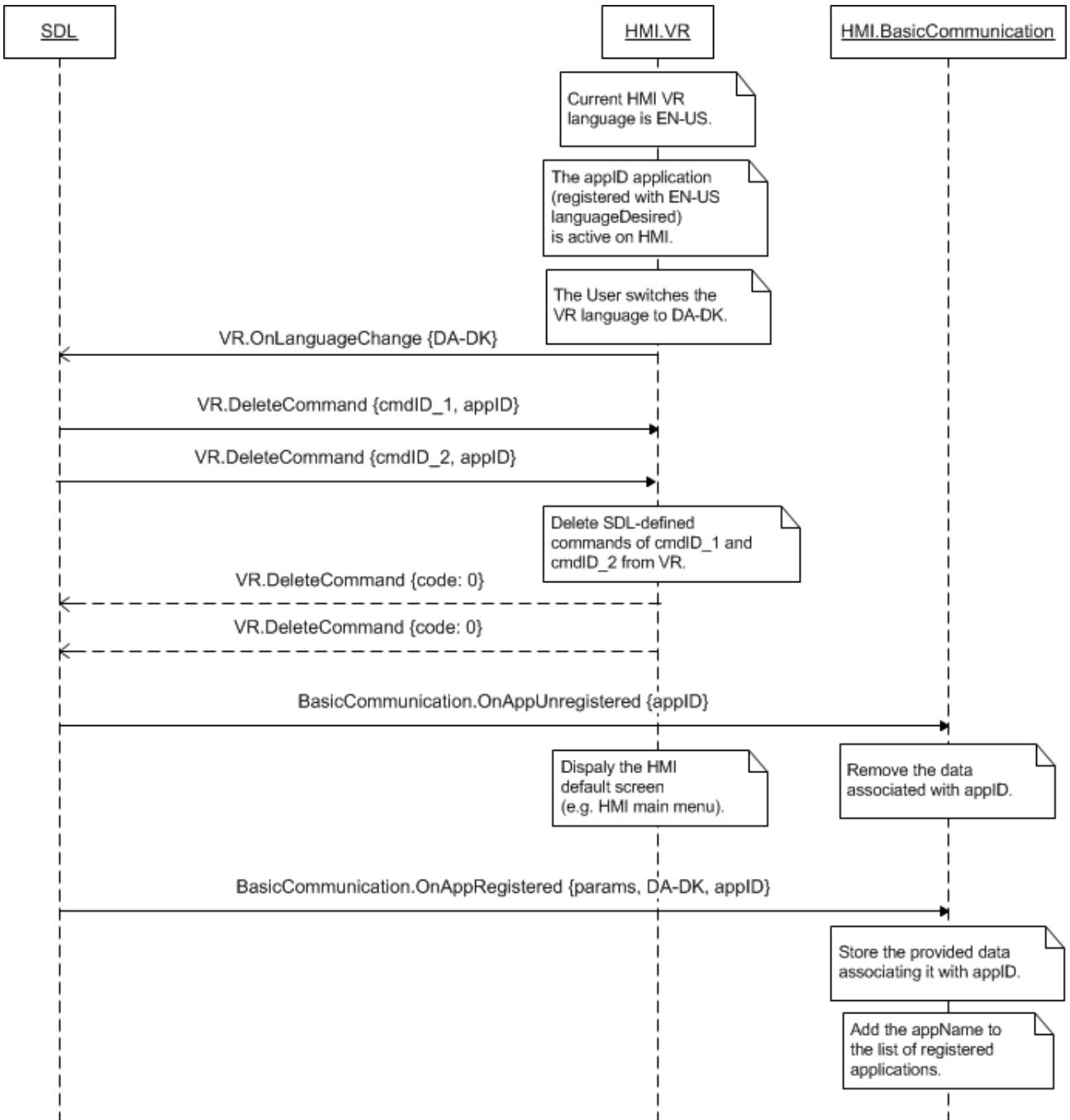
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: The named command has been deleted from VR for the named application. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

9.5.4 Sequence Diagrams

9.1.4.1 VR.DeleteCommand



9.5.5 JSON Messages Examples

9.5.5.1 Request

```
{
    "id" : 147,
    "jsonrpc" : "2.0",
    "method" : "VR.DeleteCommand",
    "params" :
    {
        "cmdID" : 4365,
        "appID" : 8764
    }
}
```

9.5.5.2 Response

```
{}
```

```

    "id" : 147,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "VR. DeleteCommand"
    }
}

```

9.5.5.3 Error message

```

{
    "id" : 147,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 13,
        "message" : "One of the provided IDs is not valid",
        "data" :
        {
            "method" : "VR. DeleteCommand"
        }
    }
}

```

9.5.6 D-Bus Messages Examples

9.5.6.1 Request

9.5.6.2 Response

9.5.6.3 Failure

9.6 ChangeRegistration

9.6.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Change the application language for voice recognition on HMI. |

SDL requests to set a display language for the named application on HMI.

The request may arrive for the application whatever being active or in background on HMI.

Note:

SDL will send the language value confirmed to be supported by HMI via UI.GetCapabilities.

9.6.2 Request

9.6.2.1 Behavior

HMI must:

1. Store the provided information associating it with application's applID.
2. Respond to the request.

9.6.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|-----------------|-----------|--|--|
| language | Common.Language | true | | The language requested to be switched to. See Language. |
| vrSynonyms | String | false | array=true maxlength=40 minsize=1 maxsize=100 | Request new VR synonyms registration Defines an additional voice recognition command. Must not interfere with any name of previously registered applications(SDL makes check). |
| appID | Integer | true | | ID of the application that relates to this RPC. |

9.6.2.2 Language

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

9.6.3 Response

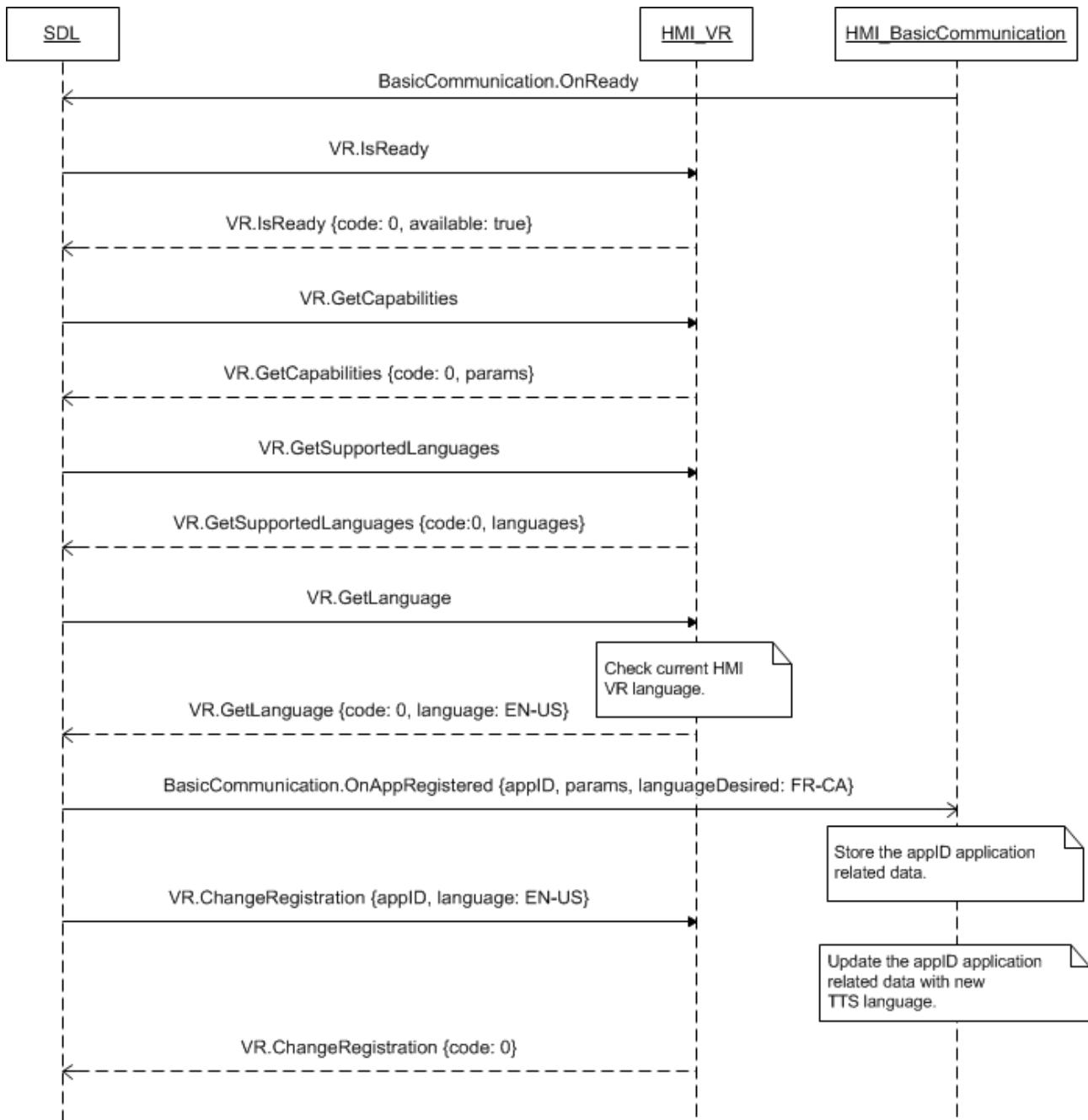
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: The requested language has been set for voice recognition for the named application. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax, parameters are out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

9.6.4 Sequence Diagrams

9.6.4.1 VR.ChangeRegistration



9.6.5 JSON Messages Examples

9.6.5.1 Request

```
{
    "id" : 206,
    "jsonrpc" : "2.0",
    "method" : "VR.ChangeRegistration",
    "params" :
    {
        "language" : DE-DE,
        "appID" : 13264
    }
}
```

9.6.5.2 Response

```
{
    "id" : 206,
```

```

"jsonrpc" : "2.0",
"result" :
{
    "code" : 0,
    "method" : "VR.ChangeRegistration"
}
}

```

9.6.5.3 Error message

```

{
    "id" : 206,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 22,
        "message" : "The unknown error occurred",
        "data" :
        {
            "method" : "VR.ChangeRegistration"
        }
    }
}

```

9.6.6 D-Bus Messages Examples

9.6.6.1 Request

9.6.6.2 Response

9.6.6.3 Failure

9.7 GetLanguage

9.7.1 Description

| | |
|----------|--------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get the current language of VR |

SDL inquires the current HMI VR language.

This RPC is sent by SDL after VR readiness is confirmed by VR.IsReady. If later the User changes the HMI language of VR, HMI must inform SDL about this event via VR.OnLanguageChange notification.

9.7.2 Request

9.7.2.1 Behavior

HMI must:

1. Check the HMI VR language currently in effect.
2. Respond providing SDL with the results of this check.

9.7.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the language currently active for VR. | JSON response | Method return | language, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax, parameters are out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

9.7.3.1 Parameters

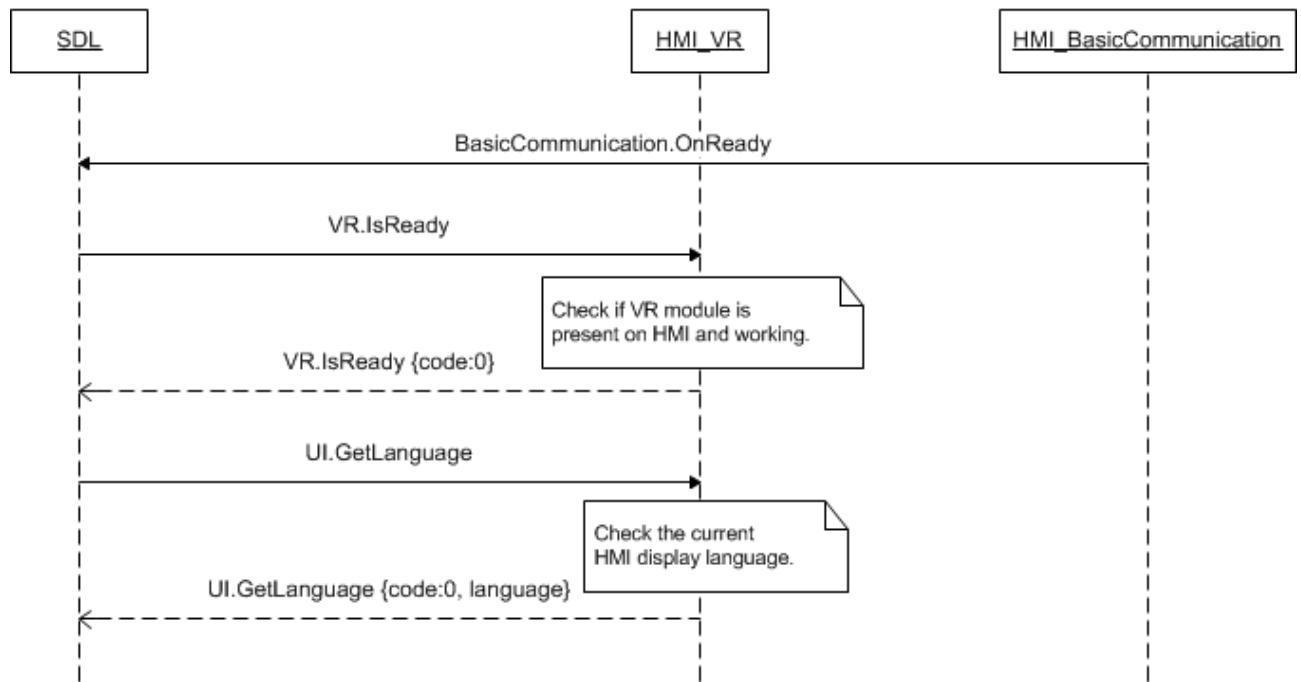
| Param Name | Type | Mandatory | Description |
|------------|-----------------|-----------|--|
| language | Common.Language | true | The language requested to be switched to. See Language. |

9.7.3.2 Language

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

9.7.4 Sequence Diagrams

9.7.4.1 VR.GetLanguage



9.7.5 JSON Messages Examples

9.7.5.1 Request

```
{
    "id" : 110,
    "jsonrpc" : "2.0",
    "method" : "VR.GetLanguage",
}
```

9.7.5.2 Response

```
{
    "id" : 110,
    "jsonrpc" : "2.0",
    "result" :
    {
        "language" : DE-DE,
        "code" : 0,
        "method" : "VR.GetLanguage"
    }
}
```

9.7.5.3 Error message

```
{
    "id" : 110,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 22,
        "message" : "During the API call the unknown error has occurred",
        "data" :
        {
            "method" : "VR.GetLanguage"
        }
}
```

```
    }  
}
```

9.7.6 D-Bus Messages Examples

9.7.6.1 Request

9.7.6.2 Response

9.7.6.3 Failure

9.8 Started

9.8.1 Description

| | |
|----------|----------------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about start of VR session |

Once VR is activated, HMI should attenuate the audio or make it not audible (depending on its capabilities). SDL needs to be notified about the event in order to provide a mobile application with the accurate information about audio streaming state on HMI.

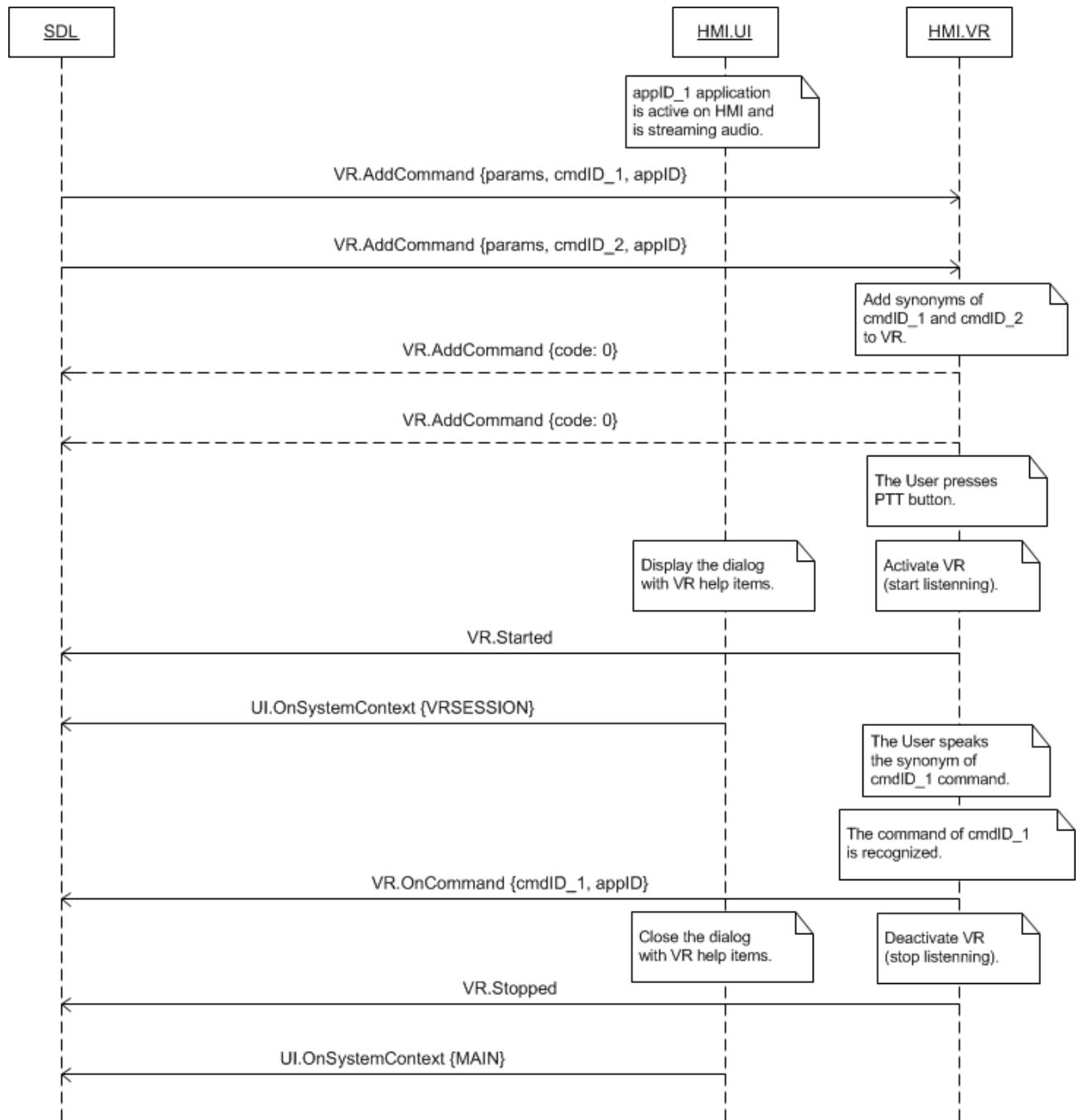
HMI must:

Send VR.Started notification when VR system is activated by:

- PTT button press
- UI.PerformInteraction RPC from SDL

9.8.2 Sequence Diagrams

9.8.2.1 VR.Started on PTT button press



9.8.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "VR.Started",
}
```

9.8.4 D-Bus Messages Examples

```
signal
sender=:1.209
-> dest=(null destination)
serial=94
path=/;
```

```
interface=com.ford.sdl.hmi.VR;  
member=Started
```

9.9 Stopped

9.9.1 Description

| | |
|----------|---------------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about stop of VR session |

After VR interaction with the User has completed, HMI should return the audio to be audible for the User. SDL requires to be notified about the event so that to inform the mobile application about the actual audio streaming state on HMI.

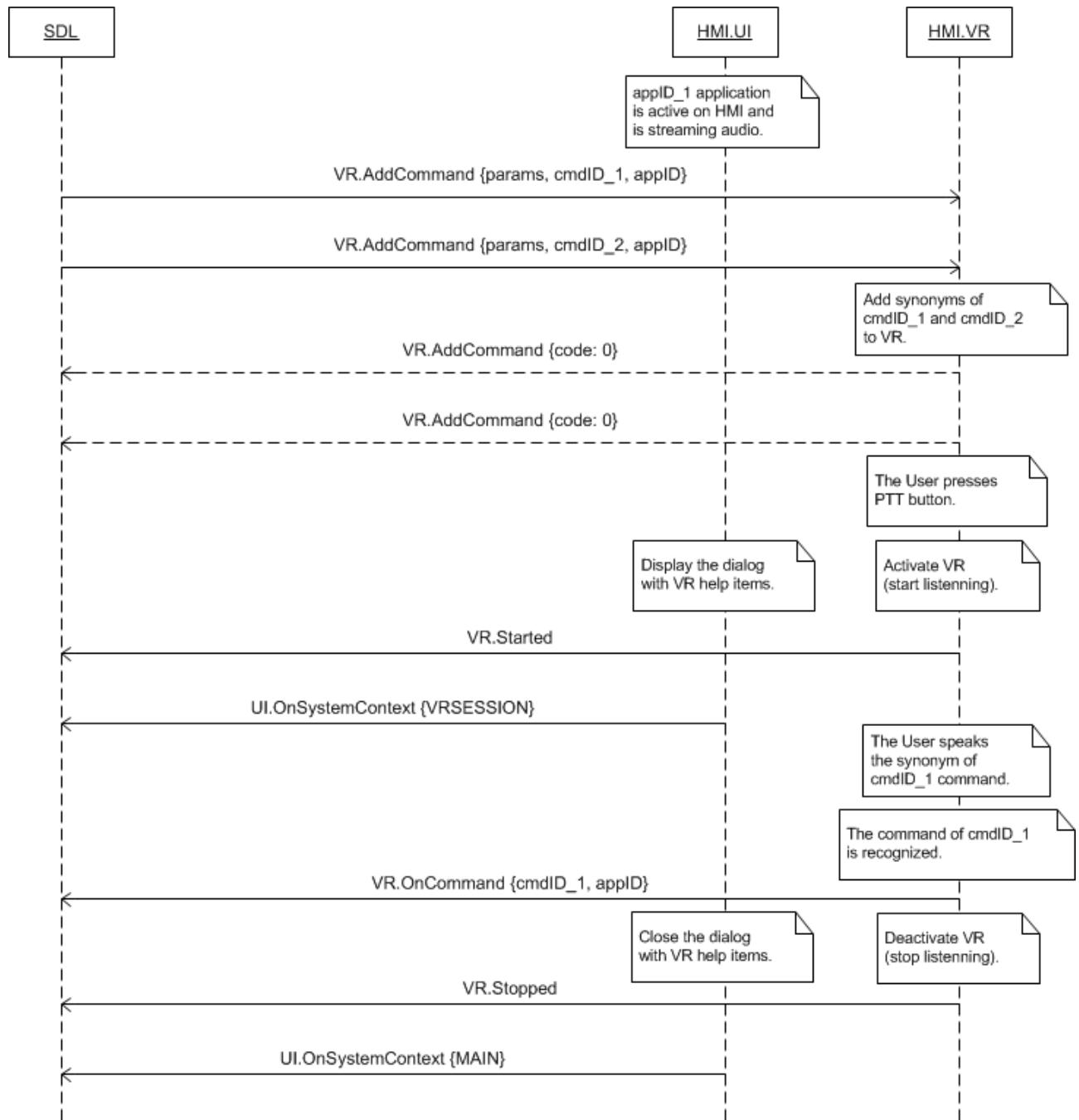
HMI must:

Send VR.Stopped notification when voice recognition is ended by:

- Successfully recognized command
- User cancelling the interaction
- Event of a higher priority

9.9.2 Sequence Diagrams

9.8.2.1 VR.Stopped on VR session ending



9.9.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "VR.Stopped",
}
```

9.9.4 D-Bus Messages Examples

```
signal
sender=:1.209
-> dest=(null destination)
serial=98
path=/;
interface=com.ford.sdl.hmi.VR;
member=Stopped
```

9.10 OnCommand

9.10.1 Description

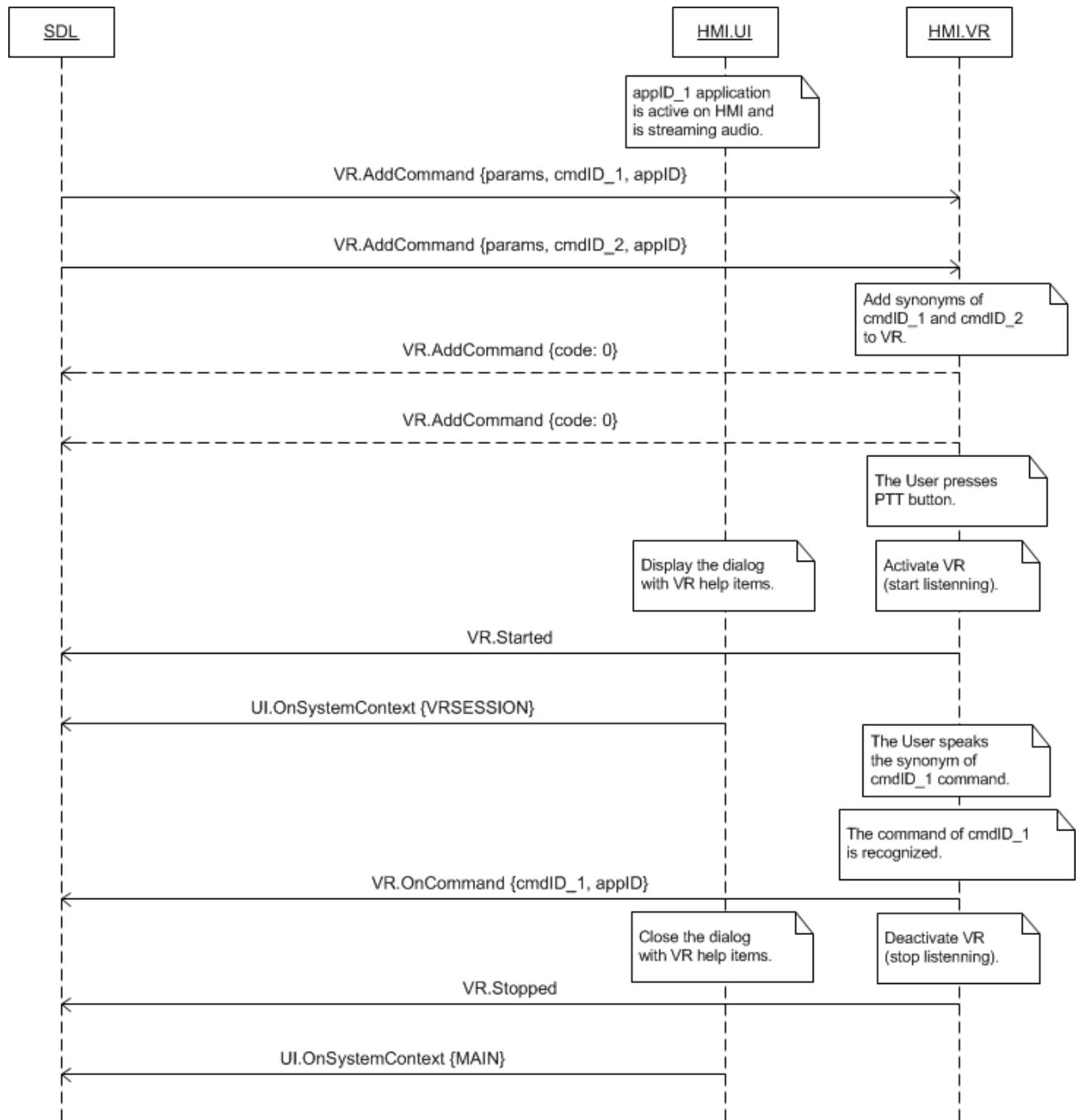
| | |
|-----------------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform the command is recognized by VR |

9.10.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|---|
| cmdID | Integer | true | The ID of the command selected by the User. |
| appID | Integer | true | ID of the application that relates to this RPC. |

9.10.2 Sequence Diagrams

9.10.2.1 VR.OnCommand



9.10.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "VR.OnCommand",
  "params" :
  {
    "cmdID" : 4365,
    "appId" : 12564
  }
}
```

9.10.4 D-Bus Messages Examples

| |
|--|
| |
|--|

9.11 OnLanguageChange

9.11.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about language of VR is changed. |

SDL needs to be in the know when the User changes the language of voice recognition: Upon the receipt of OnLanguageChange notification SDL will unregister the applications of different language to provide them with possibility to re-register with the correct (new HMI) VR language.

HMI must:

Send the VR.OnLanguageChange notification when the User switches VR to another language and provide this new value via language parameter.

9.11.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|-----------------|-----------|---|
| language | Common.Language | true | Language VR has been switched to. See Language |

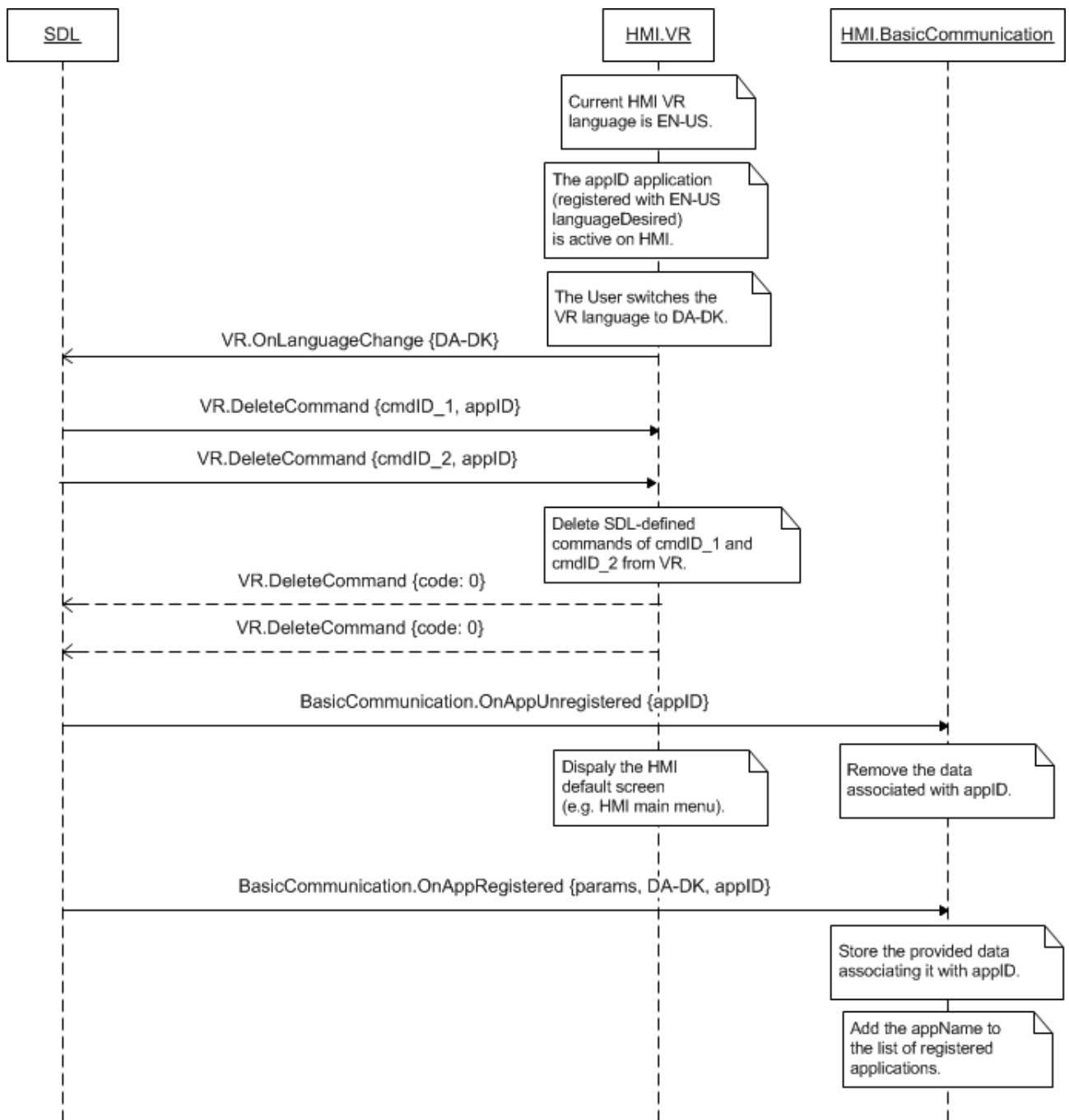
9.11.1.2 Language

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |

| Element Name | Value | Description |
|--------------|-------|--------------------|
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

9.11.2 Sequence Diagrams

9.11.2.1 VR.OnLanguageChange



9.11.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "VR.OnLanguageChange",
  "params" :
  {
    "language" : IT-IT
  }
}
```

```

    }
}
```

9.11.4 D-Bus Messages Examples

```

signal
sender=:1.226
-> dest=(null destination)
serial=250
path=/;
interface=com.ford.sdl.hmi.VR;
member=OnLanguageChange
int32 22
```

9.12 VR.PerformInteraction

9.12.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Perform a VR interaction with the User. |

SDL uses PerformInteraction for requesting an interaction with the User, for example:

- To confirm a command
- To make a choice between several suggested ones.

The request concerns the application currently active on HMI.

Notes about PerformInteraction

1. *SDL may intend to perform one of three modes of interaction:*
 - 1.1. 'Manual only' – when the User is expected to make a choice manually only (display, buttons, soft buttons are involved).
 - 1.2. 'VR only' – when the User is expected to make a choice by voice only (voice recognition module is involved).
 - 1.3. 'Both' – when the User is expected to make a choice whichever by voice or manually (display, buttons, soft buttons, VR module are involved).
2. *SDL sends both VR.PerformInteraction and UI.PerformInteraction for each of above types of interaction. HMI's expected behavior differs depending on interaction mode (see section 7.12.2.1 Behavior).*
3. *SDL provides HMI with the set(s) of choices to be presented to the User. See the below Notes of CreateInteractionChoiceSet.*

Notes about CreateInteractionChoiceSet:

1. *Mobile application in advance creates a set (or several sets) of choices on SDL to be further used in PerformInteraction.*
2. *SDL in advance provides HMI with VR synonyms of created choices via VR.AddCommand (see section 9.4 for details). They are then referenced in VR.PerformInteraction (see section 9.12).*
3. *SDL sends UI choices (those to be displayed) within UI.PerformInteraction directly.*

9.12.2 Request

9.12.2.1 Behavior

HMI must:

1. Distinguish the interaction mode by the following characteristics:

| Mode | SDL sends both RPCs with the following parameters: | Note |
|---------|--|----------------------|
| VR_ONLY | VR.PerformInteraction {grammarID, params} | grammarID is present |
| | UI.PerformInteraction {params} | No choiceSet |

| | | |
|-------------|---|----------------------|
| MANUAL_ONLY | VR.PerformInteraction {params} | No grammarID |
| | UI.PerformInteraction {choiceSet, params} | choiceSet is present |
| BOTH | VR.PerformInteraction {grammarID, params} | grammarID is present |
| | UI.PerformInteraction {choiceSet, params} | choiceSet is present |

I. ‘VR only’ interaction mode:

HMI must:

1. Store the provided text values of initial, help and timeout prompts (see below for how to use) as well as the value of timeout.
2. Start VR session using VR commands of the corresponding grammarID with HMI-defined timeout.
3. Speak the `initialPrompt` right after request is received (sending notifications of `TTS.Started` and `TTS.Stopped` before and after speaking the prompt correspondingly).
4. Start counting down the value provided within `timeout` parameter.
5. Speak the `timeoutPrompt` (sending notifications of `TTS.Started` and `TTS.Stopped` before and after speaking the prompt correspondingly) when `timeout` value runs out.
6. Respond the request with one of applicable result codes upon:
 - HMI-defined-VR-session timeout (ABORTED result)
 - User’s choice (SUCCESS result) providing the `choiceID` (that is equal to the `cmdID` of corresponding VR.AddCommand)
7. Forget the `prompts` data after the interaction completes.

II. ‘Manual only’ interaction mode:

1. Store the provided text values of initial, help and timeout prompts (see below for how to use) as well as the value of timeout.
2. Respond the request with `SUCCESS` result codes (for all of applicable result codes please see section 10.9.3 Response).
3. Speak the `initialPrompt` right after request is received (sending notifications of `TTS.Started` and `TTS.Stopped` before and after speaking the prompt correspondingly).
4. Start counting down the value provided within `timeout` parameter.
5. Speak the `timeoutPrompt` (sending notifications of `TTS.Started` and `TTS.Stopped` before and after speaking the prompt correspondingly) when `timeout` value runs out.
6. Forget the `prompts` data upon the corresponding `UI.PerformInteraction` is completed by user’s UI choice or aborted by the user or some event.

III. ‘Both’ interaction mode:

1. Store the provided text values of initial, help and timeout prompts (see below for how to use) as well as the value of timeout.
2. Start VR session using VR commands of the corresponding grammarID with HMI-defined timeout.

3. Speak the `initialPrompt` right after request is received (sending notifications of `TTS.Started` and `TTS.Stopped` before and after speaking the prompt correspondingly).
4. Start counting down the value provided within `timeout` parameter.
5. Speak the `timeoutPrompt` (sending notifications of `TTS.Started` and `TTS.Stopped` before and after speaking the prompt correspondingly) when `timeout` value runs out.
6. Respond the request with one of applicable result codes upon:
 - HMI-defined-VR-session timeout (ABORTED result) AND proceed with **step 7**.
 - User's choice (SUCCESS result) providing the `choiceID` (that is equal to the `cmdID` of corresponding VR.AddCommand)
7. Repeat step 4. -5. for corresponding UI.PerformInteraction and forget the prompts values when UI.PerformInteraction completes.

9.12.2.2 Parameters

| Param name | Type | Mandatory | Additional | Description |
|---------------|-----------------|-----------|---|--|
| helpPrompt | Common.TTSChunk | false | Array = true minsize = 1 maxsize = 100 | This is the array of strings to be spoken when the HMI-defined 'Help' VR command is recognized during PerformInteraction (see TTSChunk). |
| initialPrompt | Common.TTSChunk | true | Array = true minsize = 1 maxsize = 100 | The initial prompt that must be spoken to the User right after request comes to HMI (see TTSChunk). |
| timeoutPrompt | Common.TTSChunk | false | Array = true minsize = 1 maxsize = 100 | Must be spoken upon timeout defined within request (see TTSChunk). |
| timeout | Integer | true | - | Timeout. When ran out timeoutPrompt must be spoken. |
| grammarID | Integer | false | Array = true minsize = 1 maxsize = 100 minvalue = 0 maxvalue = 2000000000 | ID of VR choice set: contains a set of commands previously added via VR.AddCommand that have one and the same grammrID. |

9.12.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

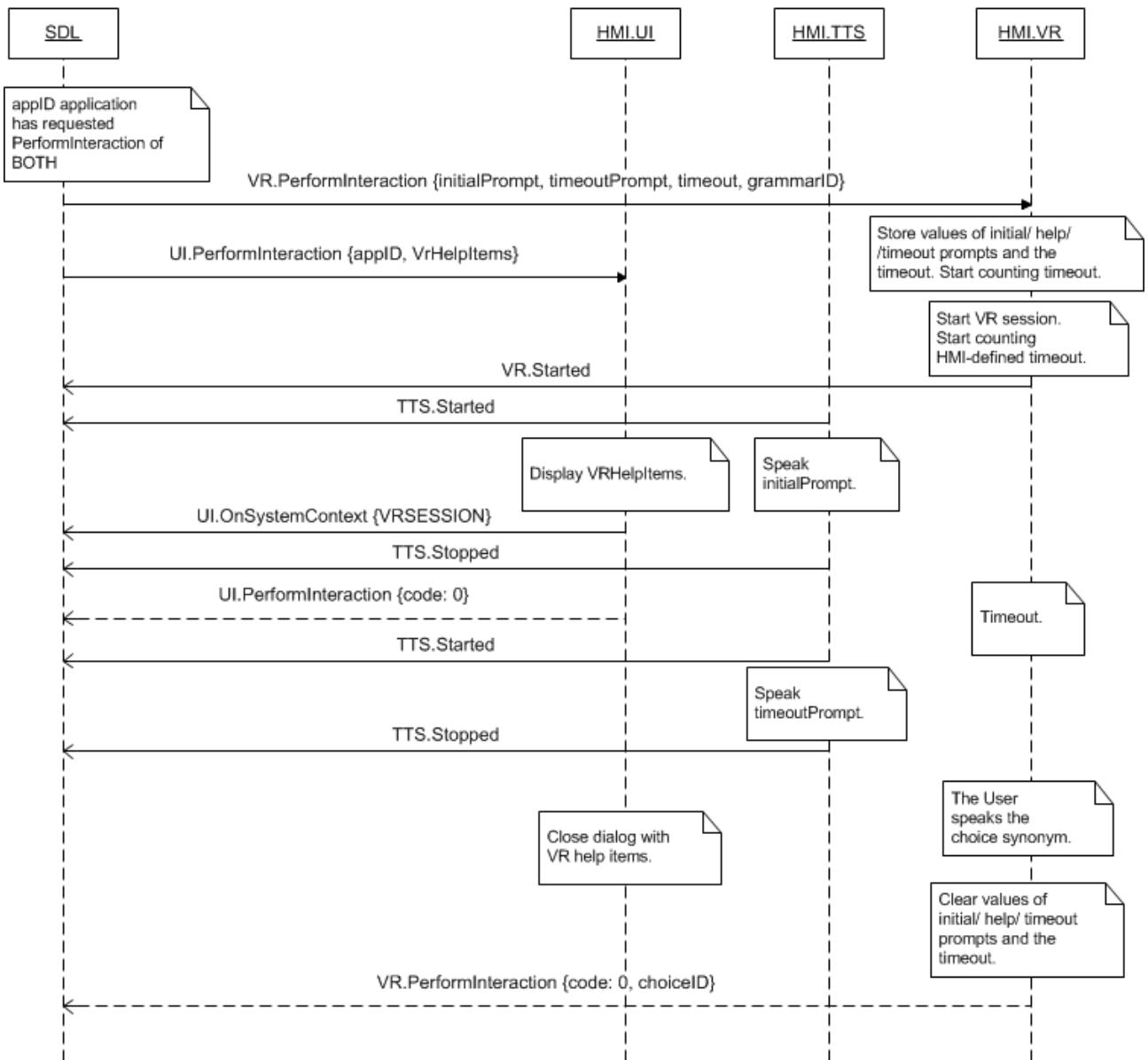
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|------------------|----------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS The cases described in 7.12.2: '1.2', 'II.3', 'III.3', 'III.4.3'. | JSON response | Regular response | choiceID, code: 0 | See section 9.12.4 Parameters. |
| Failure | ABORTED The interaction is aborted by the User or system event of higher priority. | JSON error message | Regular response | code: 5 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | TIMED_OUT The User has not made a choice during HMI-defined timeout. | | | code: 10 | |
| | GENERIC_ERROR The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

9.12.4 Parameters

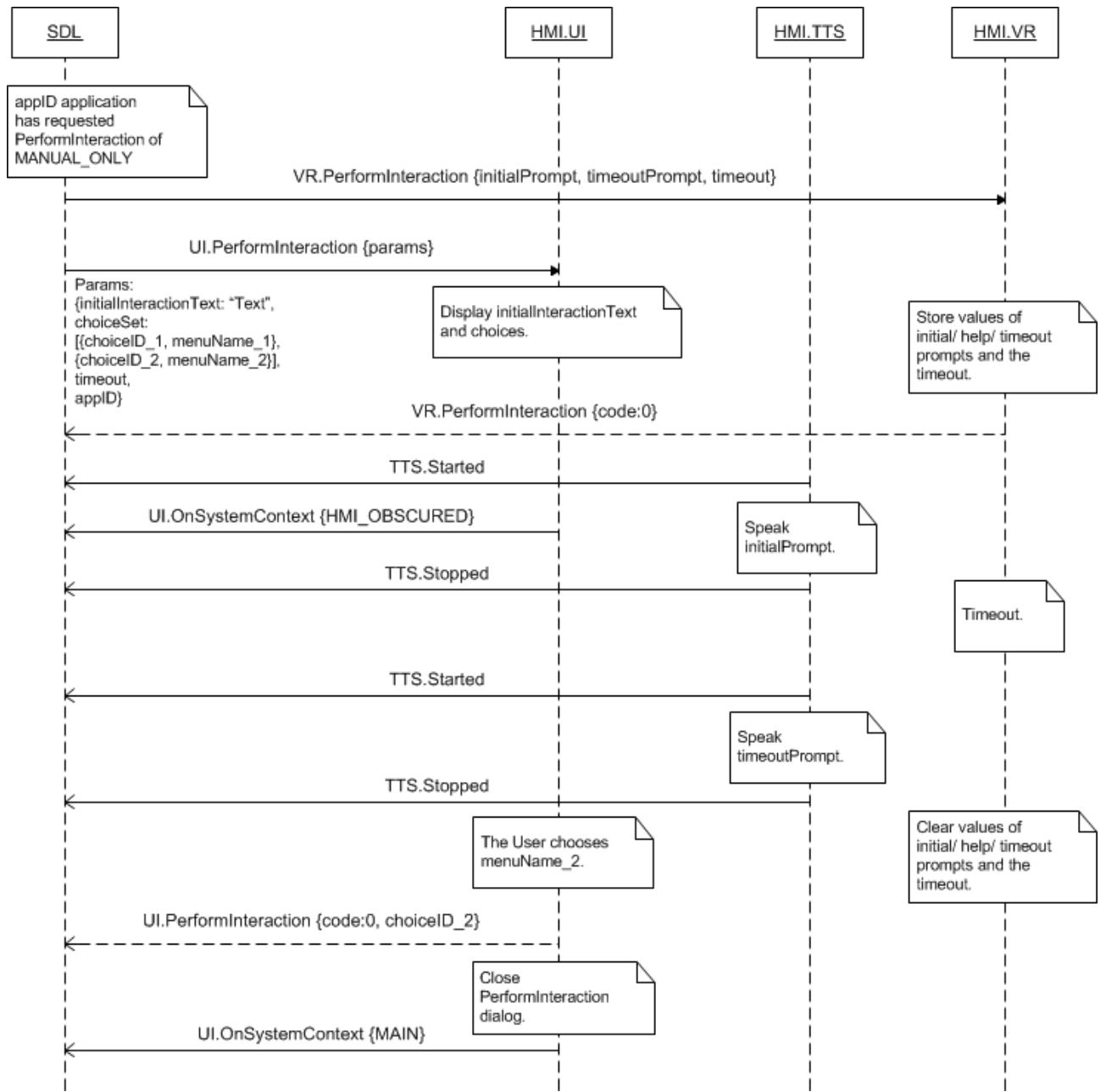
| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|---------------------------------------|---|
| choiceID | Integer | false | Minvalue = 0 Maxvalue = 2000000000 | ID that represents the choice made by the User (one among those provided within the request). |

9.12.4 Sequence Diagrams

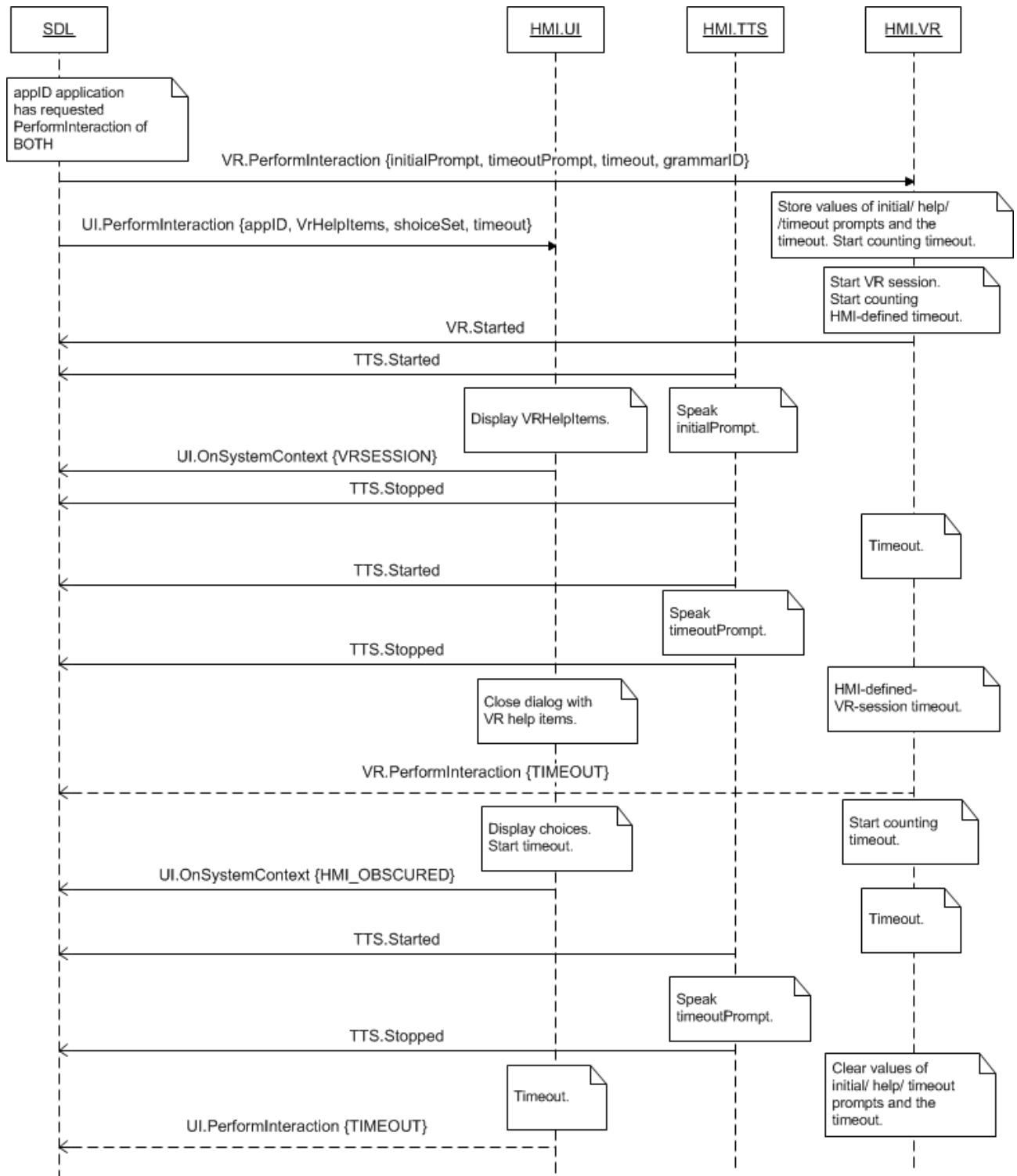
9.12.4.1 VR.PerformInteraction in ‘VR only’ mode successfully completed



9.12.4.2 VR.PerformInteraction in ‘Manual only’ mode successfully completed



9.12.4.3 VR.PerformInteraction in 'Both' mode timed out



9.12.5 JSON Messages Examples

9.12.5.1 Request

```
{
    "id" : 79,
    "jsonrpc" : "2.0",
    "method" : "VR.PerformInteraction",
    "params" :
    {
        "initialPrompt" :
        [
            {
                "text" : "Please make your choice by voice",
            },

```

```

    "helpPrompt" :
    [
        "text" : "Yes",
        "text" : "No",
        "text" : "Skip"
    ],
    "timeoutPrompt" :
    [
        "text" : "The time is about to run out"
    ],
    "timeout" : 10000
    "grammarID" : 245
}
}

```

9.12.5.2 Response

```
{
    "id" : 79,
    "jsonrpc" : "2.0",
    "result" :
    {
        "choiceID" : 2416
        "code" : 0,
        "method" : "VR.PerformInteraction"
    }
}
```

7.12.5.3 Error message

```
{
    "id" : 79,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 10,
        "message" : "Interaction reached the maximum timeout and closed",
        "data" :
        {
            "method" : "UI.PerformInteraction"
        }
    }
}
```

7.12.6 D-Bus Messages Examples

7.12.6.1 Request

```
[REDACTED]
```

7.12.6.2 Response

```
[REDACTED]
```

7.12.6.3 Failure

```
[REDACTED]
```

10 TTS Component Description

10.1 IsReady

10.1.1 Description

| | |
|----------|------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Know if TTS module is ready. |

The request comes after HMI's readiness is confirmed via [OnReady](#) notification. SDL requires the information about whether the TTS module is physically present on HU and if so whether it is working.

Note:

If TTS module is responded to be unavailable, SDL will not further send the requests related to it.

10.1.2 Request

10.1.2.1 Behavior

HMI must:

- Check whether TTS module is present and ready
- Respond correspondingly to results of this check.

10.1.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

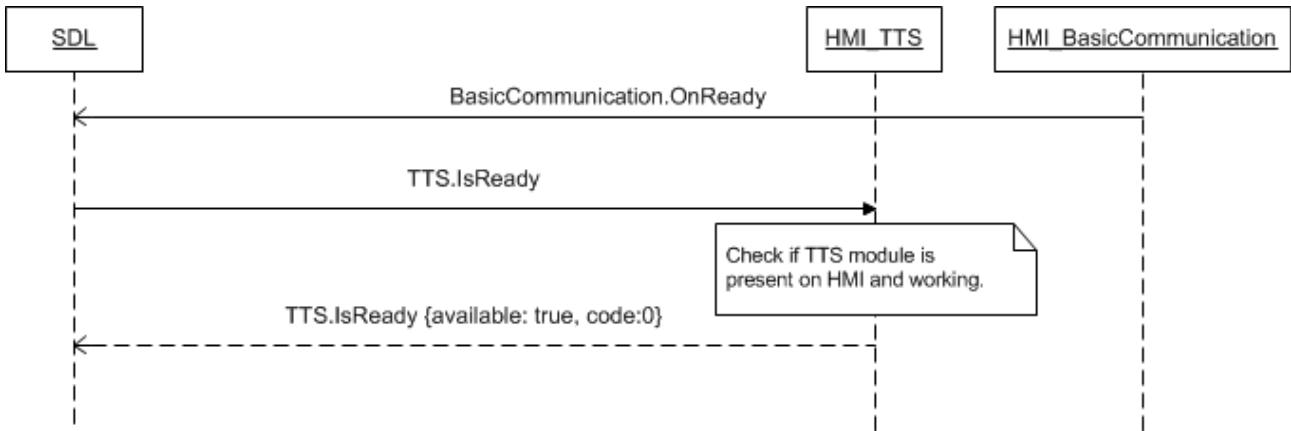
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|-----------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the information about TTS availability. | JSON response | Method return | available, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | DATA_NOT_AVAILABLE: The information about TTS availability cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

10.1.3.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|---|
| availabe | Boolean | true | Must be - 'true' if UI is present and ready - 'false' if not. |

10.1.4 Sequence Diagrams

10.1.4.1 TTS.IsReady and preceding OnReady



10.1.5 JSON Messages Examples

10.1.5.1 Request

```
{
    "id" : 45,
    "jsonrpc" : "2.0",
    "method" : "TTS.IsReady"
}
```

10.1.5.2 Response

```
{
    "id" : 45,
    "jsonrpc" : "2.0",
    "result" :
    {
        "available" : true,
        "code" : 0,
        "method" : "TTS.IsReady"
    }
}
```

10.1.5.3 Error message

```
{
    "id" : 45,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "Invalid data",
        "data" :
        {
            "method" : "TTS.IsReady"
        }
    }
}
```

10.1.6 D-Bus Messages Examples

10.1.6.1 Request

10.1.6.2 Response

10.1.6.3 Failure

| |
|--|
| |
|--|

10.2 GetCapabilities

10.2.1 Description

| | |
|----------|----------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get TTS capabilities |

SDL can store the HMI capabilities in `smartDeviceLink.ini` file saved in SDL directory. Nevertheless, on its startup after getting OnReady notification and response to `TTS.IsReady` from HMI, SDL sends this RPC for obtaining the capabilities of TTS.

10.2.2 Request

7.1.2.1 Behavior

HMI must:

1. Check the TTS capabilities.
2. Respond correspondingly to results of this check.

10.2.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|--|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the TTS capabilities. | JSON response | Method return | capabilities, prerecordedSpeechCapabilities, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | DATA_NOT_AVAILABLE: The TTS capabilities cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

10.2.3.1 Parameters

| Param name | Type | Mandatory | Additional | Description |
|-------------------------------|----------------------------|-----------|--|--|
| capabilities | Common.Speech Capabilities | true | array = true minsize = 1 maxsize = 5 | Contains information about the TTS capabilities. |
| prerecordedSpeechCapabilities | Common.PrerecordedSpeech | true | array = true minsize = 1 maxsize = 5 | Contains a list of prerecorded speech items present on the platform. |

10.2.3.2 SpeechCapabilities

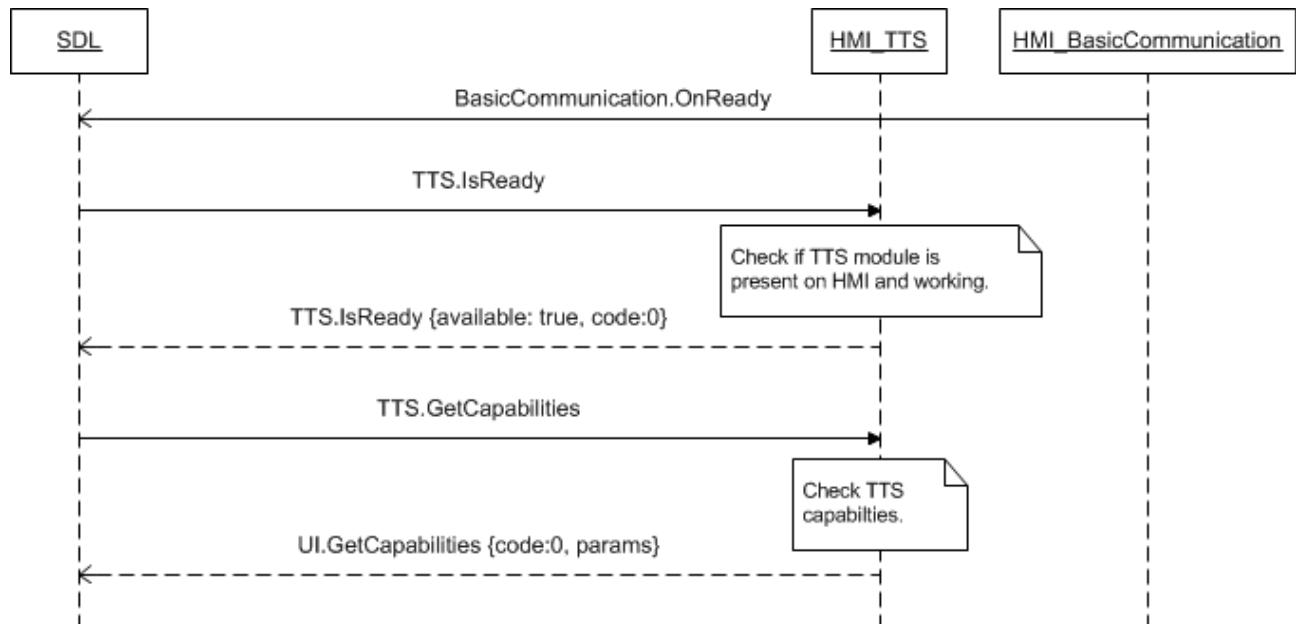
| Element name | Value | Short Description |
|-----------------|-------|---|
| TEXT | 0 | HMI has the capability to speak the provided text. |
| SAPI_PHONEMES | 1 | Speech API phonemes: used for creating pronunciations for words that are not currently in the lexicon. |
| LHPLUS_PHONEMES | 2 | LH Plus API phonemes: used for creating pronunciations for words that are not currently in the lexicon. |
| PRE_RECORDED | 3 | HMI has the capability to speak the pre-recorded phrases. |
| SILENCE | 4 | HMI has the capability to play silence over TTS during the default HMI-defined period of time. |

10.2.3.3 PrerecordedSpeech

| Element name | Value | Short Description |
|-----------------|-------|-------------------|
| HELP_JINGLE | 0 | |
| INITIAL_JINGLE | 1 | |
| LISTEN_JINGLE | 2 | |
| POSITIVE_JINGLE | 3 | |
| NEGATIVE_JINGLE | 4 | |

10.2.4 Sequence Diagrams

10.1.4.1 TTS.GetCapabilities



10.2.5 JSON Messages Examples

10.2.5.1 Request

```
{
  "id" : 13,
  "jsonrpc" : "2.0",
  "method" : "TTS.GetCapabilities"
}
```

10.2.5.2 Response

```
{  
    "id" : 13,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "capabilities" : [TEXT],  
        "prerecordedSpeechCapabilities" : [HELP_JINGLE, INITIAL_JINGLE,  
                                            LISTEN_JINGLE, POSITIVE_JINGLE,  
                                            NEGATIVE_JINGLE],  
        "code" : 0,  
        "method" : "TTS.GetCapabilities"  
    }  
}
```

10.2.5.3 Error message

```
{  
    "id" : 28,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 11,  
        "message" : "The data sent is invalid",  
        "data" :  
        {  
            "method" : "TTS.GetCapabilities"  
        }  
    }  
}
```

10.2.6 D-Bus Messages Examples

10.2.6.1 Request

```
[REDACTED]
```

10.2.6.2 Response

```
[REDACTED]
```

10.2.6.3 Failure

```
[REDACTED]
```

10.3 GetSupportedLanguages

10.3.1 Description

| | |
|----------|--------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get languages supported by TTS |

Once TTS module is confirmed to be ready (via response to IsReady RPC) SDL starts discovering its capabilities via GetCapabilities and GetSupportedLanguages RPCs.

Response to TTS.GetSupportedLanguages is assumed to bring the information about what languages are supported for voice recognition by VR module. Having obtained this information SDL will monitor the language parameter within RPCs from mobile application(s) and reject the requests containing language not supported by HMI.

Note:

The list of languages recognized by SDL is provided in the section 10.3.3.2 Language Enumeration.

10.3.2 Request

7.1.2.1 Behavior

HMI must:

- Check the TTS supported languages
- Respond correspondingly to results of this check.

10.3.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|-----------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the TTS supported languages. | JSON response | Method return | languages, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | DATA_NOT_AVAILABLE: The TTS supported languages cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

10.3.3.1 Parameters

| Param name | Type | Mandatory | Additional | Description |
|------------|-----------------|-----------|--|--|
| languages | Common.Language | true | Array = true minsize = 1 maxsize = 100 | List of languages supported in TTS. See Language. |

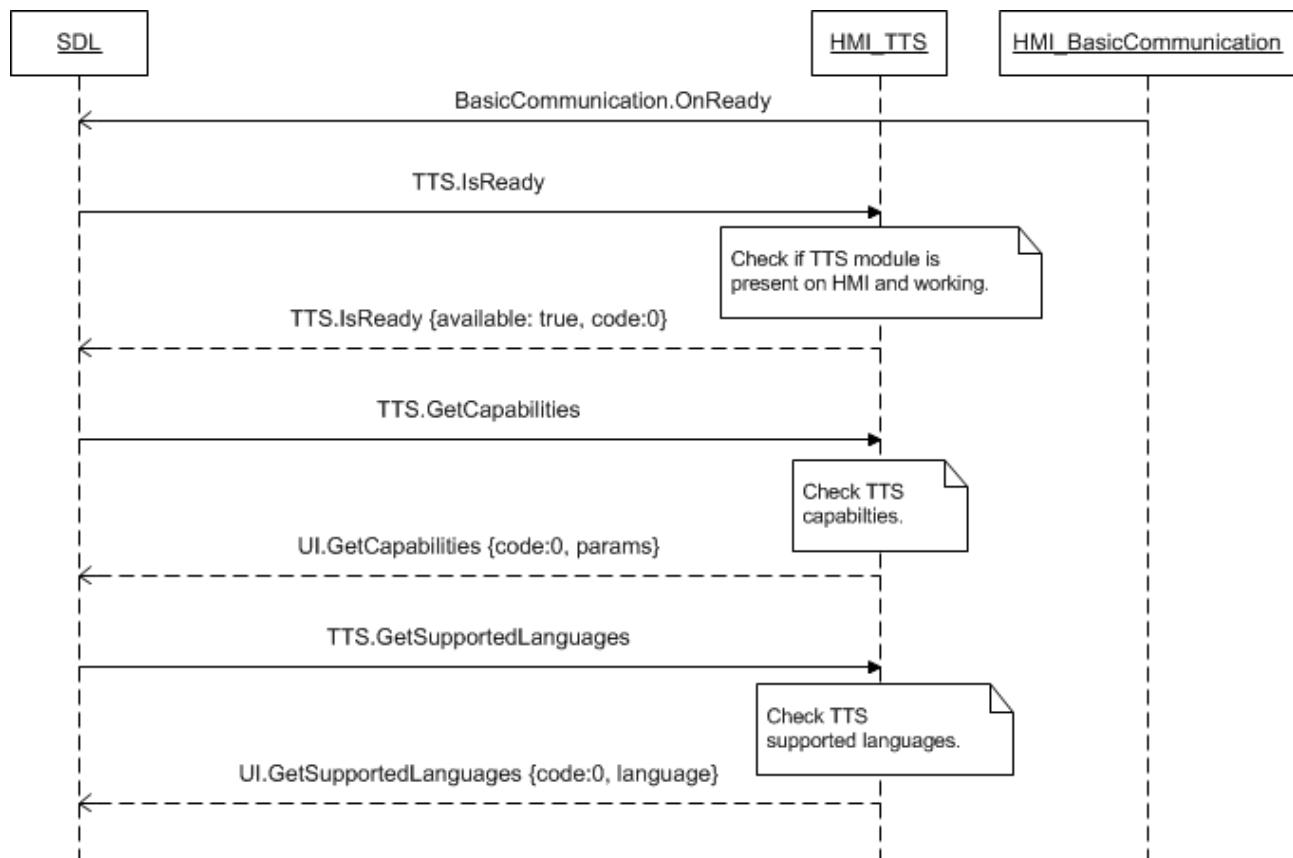
10.3.3.2 Language

| Element Name | Value | Description |
|--------------|-------|------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

10.3.4 Sequence Diagrams

10.3.4.1 TTS.GetSupportedLanguages



10.3.5 JSON Messages Examples

10.3.5.1 Request

```
{
  "id" : 19,
  "jsonrpc" : "2.0",
  "method" : "TTS.GetSupportedLanguages"
```

```
}
```

10.3.5.2 Response

```
{
    "id" : 19,
    "jsonrpc" : "2.0",
    "result" :
    {
        "languages" : [AR-SA, DE-DE, EN-GB, EN-US, ES-ES, FR-FR, IT-IT],
        "code" : 0,
        "method" : "TTS.GetSupportedLanguages"
    }
}
```

10.3.5.3 Error message

```
{
    "id" : 19,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "The data sent is invalid",
        "data" :
        {
            "method" : "TTS.GetSupportedLanguages"
        }
    }
}
```

10.3.6 D-Bus Messages Examples

10.3.6.1 Request

10.3.6.2 Response

10.3.6.3 Failure

10.4 Speak

10.4.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Prompt TTS to speak the requested text |

SDL provides the text information for HMI to speak it to the User via TTS engine.

10.4.2 Request

10.4.2.1 Behavior

HMI must:

1. Notify SDL via TTS.Started about speaking is started

2. Depending on HMI capabilities, attenuate or make the audio (if streamed by the application or HMI itself) not audible.
3. Distinguish the type of request (please see section 10.2.3.3 SpeakType):
 - 3.1. **ALERT**-type requests have the highest priority: HMI must speak the requested text by all means and right away.
 - 3.2. **SPEAK** type requests have the normal priority: HMI must speak the requested text, still the delays are acceptable.
 - 3.3. If type is omitted within request, HMI must speak the requested text; still the negative response is acceptable.
4. Speak the text requested within **TTSChunk** parameter. The text value is specified together with 'type' parameter that defines how exactly this text must be used by HMI:
 - **TEXT** – the text is provided and must be pronounced by HMI (letter-to-sound).
 - **SAPI_PHONEMES / LHPLUS_PHONEMES** – the phonemes are provided and must be pronounced by HMI.
 - **PRE_RECORD** – the provided text must be pronounced by pre-recorded phrases.
 - **SILENCE** – HMI must play silence over TTS during the default HMI-defined period of time.
5. Respond the request with successful result code (the cases of non-successful results are described in section 10.4.3 Response).
6. Notify SDL via **TTS.Stopped** about speaking is finished.

10.4.3.1 Parameters

| Param name | Type | Mandatory | Additional | Description |
|------------|------------------|-----------|--|---|
| ttsChunks | Common.TTSChunk | true | Array = true minsize = 1 maxsize = 100 | List of strings to be spoken. See TTSChunks |
| speakType | Common.SpeakType | false | - | |
| appID | Integer | false | - | ID of the application that requested this RPC. |

10.4.3.2 TTSChunk

| Param name | Type | Mandatory | Additional | Description |
|------------|---------------------------|-----------|-----------------|--|
| text | String | true | Maxlength = 500 | The text or phonemes to speak |
| type | Common.SpeechCapabilities | true | - | Describes, whether it is text or a specific phoneme set. See SpeechCapabilities . |

10.2.3.2 SpeechCapabilities

| Element name | Value | Short Description |
|-----------------|-------|---|
| TEXT | 0 | Text is provided to HMI to be spoken. |
| SAPI_PHONEMES | 1 | Speech API phonemes are provided and must be used by HMI for pronouncing. |
| LHPLUS_PHONEMES | 2 | LH Plus phonemes are provided and must be used by HMI for pronouncing. |
| PRE_RECORD | 3 | The text is provided and must be spoken as a pre-recorded phrase by HMI. |
| SILENCE | 4 | HMI must play silence over TTS during the default HMI-defined period of time. |

10.2.3.3 SpeakType

| Element name | Value | Short Description |
|--------------|-------|--|
| ALERT | 0 | TTS.Speak has the <u>alert</u> type (that is, either accompanies UI.Alert RPC or is requested independently from UI as an alert TTS prompting to the User) |

| Element name | Value | Short Description |
|--------------|-------|--|
| SPEAK | 1 | TTS.Speak has the ‘regular’ speak type (that is, a friendly reminding or notification or informing, etc., by TTS to the User). |

10.4.3 Response

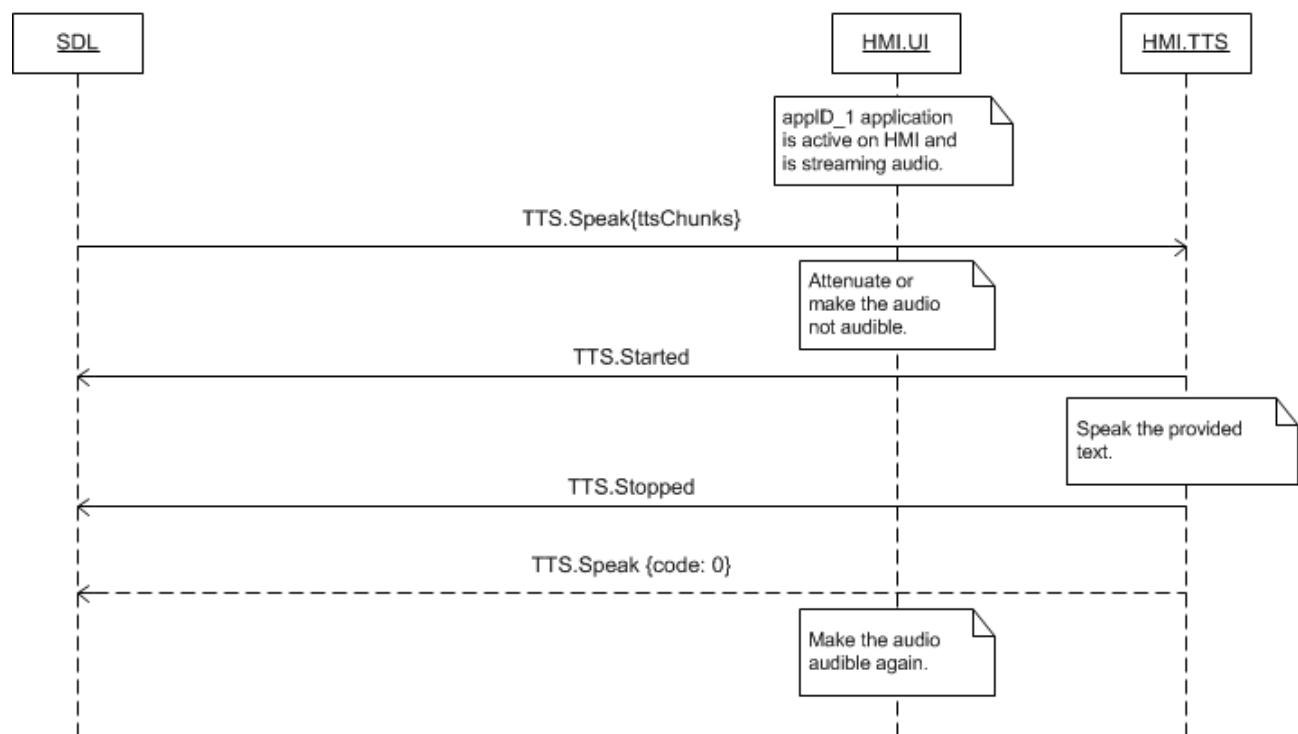
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI speaks the requested text and responds when finished. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax, parameters are out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | ABORTED: RPC is interrupted by: 1) StopSpeaking RPC 2) VR session started 3) Another RPC of higher priority | | | Code: 5 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

10.4.4 Sequence Diagrams

10.4.4.1 Speak



10.4.5 JSON Messages Examples

10.4.5.1 Request

```
{  
    "id" : 144,  
    "jsonrpc" : "2.0",  
    "method" : "TTS.Speak",  
    "params" :  
    {  
        "ttsChunks" :  
        [  
            {"text" : "Please say a command"}  
        ]  
    }  
}
```

10.4.5.2 Response

```
{  
    "id" : 144,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "code" : 0,  
        "method" : "TTS.Speak"  
    }  
}
```

10.4.5.3 Error message

```
{  
    "id" : 144,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 5,  
        "message" : "The command was aborted",  
        "data" :  
        {  
            "method" : "TTS.Speak"  
        }  
    }  
}
```

10.4.6 D-Bus Messages Examples

10.4.6.1 Request

10.4.6.2 Response

10.4.6.3 Failure

10.5 StopSpeaking

10.5.1 Description

| | |
|-------|----------|
| Type: | Function |
|-------|----------|

| | |
|-----------------|--------------------------|
| Sender: | SDL |
| Purpose: | Interrupt the Speak RPC. |

SDL sends StopSpeaking RPC to abort TTS playing initiated by TTS.Speak. This RPC is not intended to interrupt HMI-initiated TTS speaking.

10.5.2 Request

10.5.2.1 Behavior

HMI must:

1. Stop speaking the text requested via TTS.Speak.
2. Respond to TTS.StopSpeaking
3. Respond with ABORTED result code for TTS.Speak.
4. Notify SDL about TTS has finished speaking via TTS.Stopped.

10.5.3 Response

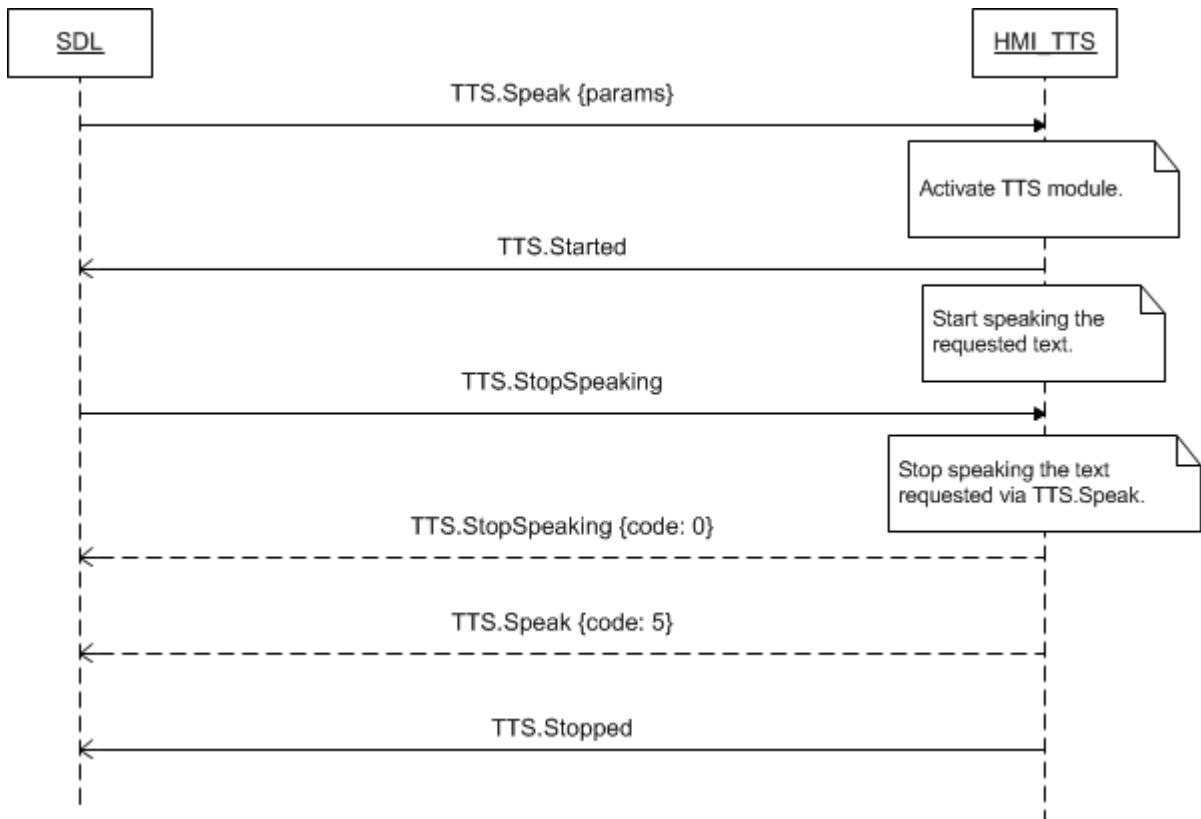
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: The Speak RPC is interrupted. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

10.5.4 Sequence Diagrams

10.5.4.1 StopSpeaking



10.5.5 JSON Messages Examples

10.5.5.1 Request

```
{
    "id" : 148,
    "jsonrpc" : "2.0",
    "method" : "TTS.StopSpeaking"
}
```

10.5.5.2 Response

```
{
    "id" : 148,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "TTS.StopSpeaking"
    }
}
```

10.5.5.3 Error message

```
{
    "id" : 148,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 11,
        "message" : "Invalid data: invalid JSON syntax",
        "data" :
        {
            "method" : "TTS.StopSpeaking"
        }
    }
}
```

```
}
```

10.5.6 D-Bus Messages Examples

10.5.6.1 Request

10.5.6.2 Response

10.5.6.3 Failure

10.6 ChangeRegistration

10.6.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Change the application language for TTS |

When the application registers and its `languageDesired` value is different from the current HMI TTS language, this application is provided with the possibility to change the language and re-register just this value. `TTS.ChangeRegistration` RPC is used for such purpose of re-registering the application with HMI current TTS language.

10.6.2 Request

10.6.2.1 Behavior

HMI must:

1. Update the appID application related data previously provided via `OnAppRegistered` with the new application language desired for TTS.
2. Respond the request.

Note:

There may be cases when the application requests to re-register it with the language different from current HMI TTS language (e.g. `TTS.ChangeRegistration` (ES-ES) while the current language is EN-US):

- *HMI must: confirm it accepts such registration (respond with `SUCCESS` result code)*
- *HMI is expected to:*
 - *Recognize the text provided in the registered language for speaking further.*
 - *Use the rules and pronunciation of the current HMI TTS language (and all the consequences of using TTS engine in the wrong language (e.g. in ES-ES while the current is EN-US) are on the application).*

10.6.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|-----------------|-----------|--|--|
| language | Common.Language | true | | The language requested to be switched to. See <code>Language</code> . |
| ttsName | Common.TTSChunk | false | array=true minsize=1 maxsize=100 | Request new VR synonyms registration Defines an additional voice recognition command. Must not interfere with any name of previously registered applications(SDL makes check). |

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|------------|---|
| appID | Integer | true | | ID of the application that relates to this RPC. |

10.6.2.3 Language

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

10.6.3 Response

Note:

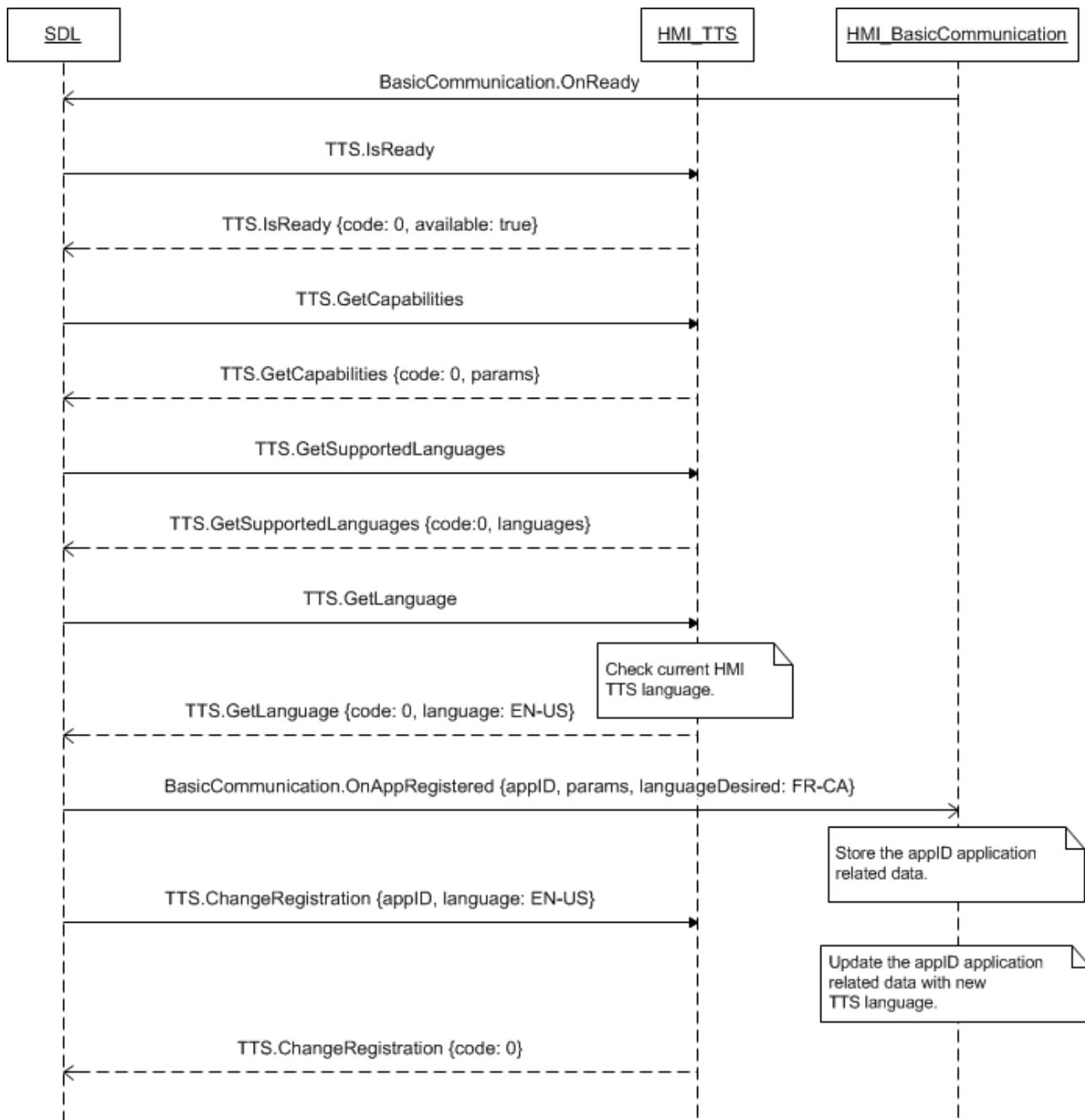
There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: The requested language has been set for TTS for the named application. | JSON response | Method return | code: 0 | |
| Failure | INVALID _ DATA: The data sent is invalid (invalid JSON syntax, parameters are out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result |

| | | | | | |
|--|---|--|--|----------|--|
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Enumeration for all SDL-supported codes. |
|--|---|--|--|----------|--|

10.6.4 Sequence Diagrams

10.6.4.1 TTS.ChangeRegistration after OnAppRegistered



10.6.5 JSON Messages Examples

10.6.5.1 Request

```
{
  "id" : 206,
  "jsonrpc" : "2.0",
  "method" : "TTS.ChangeRegistration",
  "params" :
```

```
{
  "language" : DE-DE,
  "appID" : 65539
}
```

10.6.5.2 Response

```
{
  "id" : 206,
  "jsonrpc" : "2.0",
  "result" :
  {
    "code" : 0,
    "method" : "TTS.ChangeRegistration"
  }
}
```

10.6.5.3 Error message

```
{
  "id" : 206,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : 22,
    "message" : "Unknown error occurred",
    "data" :
    {
      "method" : "TTS.ChangeRegistration"
    }
  }
}
```

10.6.6 D-Bus Messages Examples

10.6.6.1 Request

10.6.6.2 Response

10.6.6.3 Failure

10.7 GetLanguage

10.7.1 Description

| | |
|-----------------|------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get the current TTS language |

SDL inquires the current HMI TTS language.

This RPC is sent by SDL after TTS readiness is confirmed by `TTS.IsReady`. If later the User changes the HMI language of TTS, HMI must inform SDL about this event via `TTS.OnLanguageChange` notification.

10.7.2 Request

10.7.2.1 Behavior

HMI must:

1. Check the HMI TTS language currently in effect.
2. Respond providing SDL with the results of this check.

10.7.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the language currently active for TTS. | JSON response | Method return | language, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax, parameters are out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

10.7.3.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|-----------------|-----------|--|
| language | Common.Language | true | The current TTS language. See Language. |

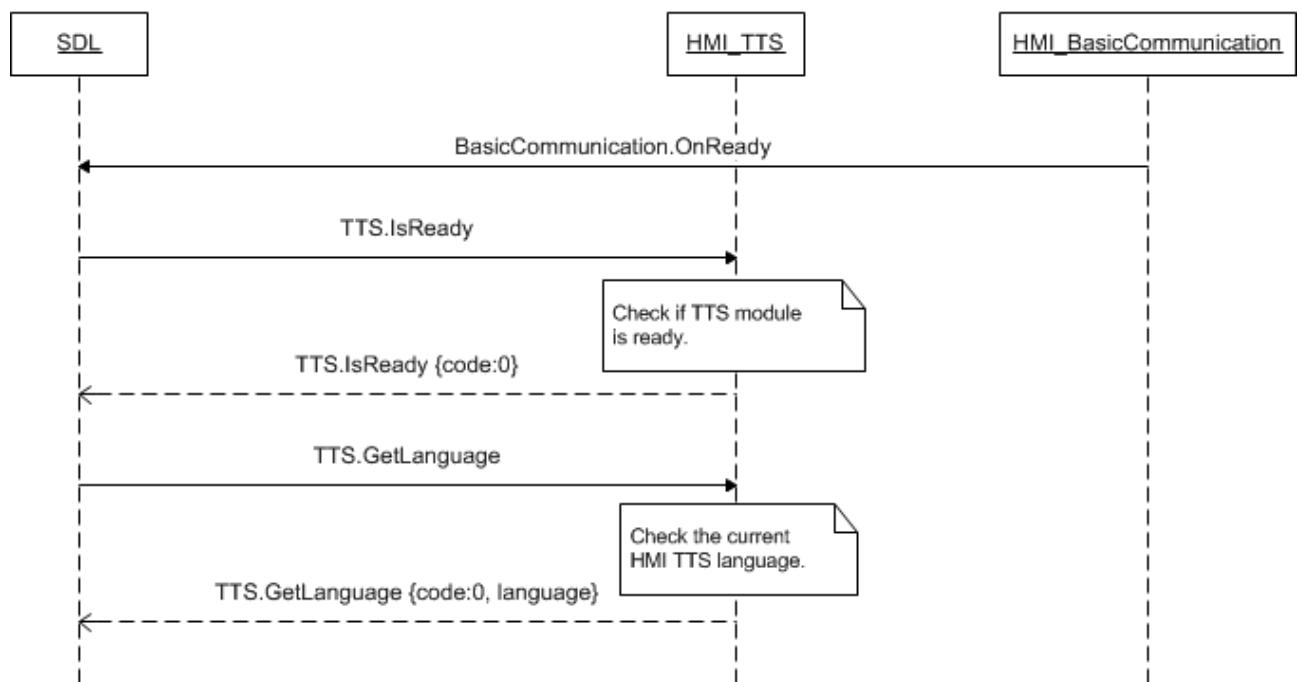
10.6.2.3 Language

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |

| Element Name | Value | Description |
|--------------|-------|------------------------|
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

10.7.4 Sequence Diagrams

10.7.4.1 TTS.GetLanguage



10.7.5 JSON Messages Examples

10.7.5.1 Request

```
{
  "id" : 110,
  "jsonrpc" : "2.0",
  "method" : "TTS.GetLanguage",
}
```

10.7.5.2 Response

```
{
  "id" : 110,
  "jsonrpc" : "2.0",
  "result" :
  {
    "language" : DE-DE,
    "code" : 0,
    "method" : "TTS.GetLanguage"
  }
}
```

10.7.5.3 Error message

```
{  
    "id" : 110,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 22,  
        "message" : "During the API call the unknown error has occurred",  
        "data" :  
        {  
            "method" : "VR.GetLanguage"  
        }  
    }  
}
```

10.7.6 D-Bus Messages Examples

10.7.6.1 Request

```
[REDACTED]
```

10.7.6.2 Response

```
[REDACTED]
```

10.7.6.3 Failure

```
[REDACTED]
```

10.8 SetGlobalProperties

10.8.1 Description

| | |
|----------|----------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Set the properties for TTS |

SDL requests to set the values for the prompts to be spoken by TTS in the definite moments during the User's interaction with the application over head unit.

10.8.2 Request

10.8.2.1 Behavior

HMI must:

1. Store the provided values associating them with the provided appID.
2. Speak the provided text during the following events related to the appID application:
 - helpPrompt – when the HMI-defined 'Help' command is recognized by VR.
 - timeoutPrompt – when the HMI-defined timeout for VR session is about going to end. After speaking this prompt HMI must provide some additional time for the user to be able to give a command by VR. After this additional amount of time is out, VR session must be closed.
3. Respond with SUCCESSFUL result code after the requested values have been stored. All of applicable result codes please see in the section 10.8.3 Response.

10.8.2.2 Parameters

| Param name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| Param name | Type | Mandatory | Additional | Description |
|---------------|------------------|-----------|--|---|
| helpPrompt | Common.TTSSchunk | false | Array = true minsize = 1 maxsize = 100 | This is the array of strings to be spoken when the HMI-defined 'Help' VR command is recognized (see TTSSchunk). |
| timeoutPrompt | Common.TTSSchunk | false | Array = true minsize = 1 maxsize = 100 | This is the array of strings to be spoken when the HMI-defined timeout for VR session is about going to end (see TTSSchunk). After speaking this prompt HMI must provide some additional time for the user to be able to give a command by VR. After this additional amount of time is out, VR session must be closed. |
| appID | Integer | true | - | ID of the application that requested this RPC. |

10.8.3 Response

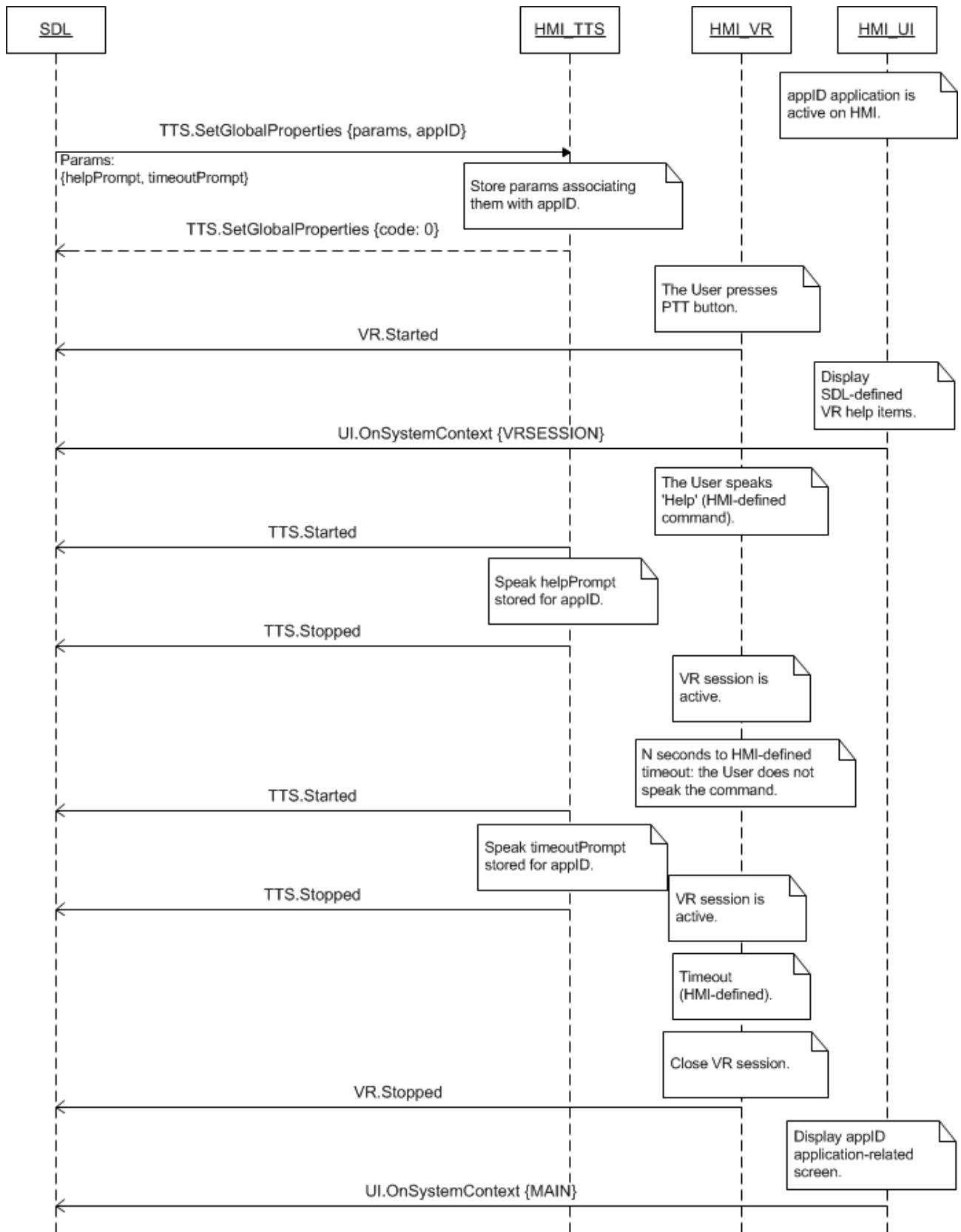
Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: The requested properties are set for TTS for the named application. | JSON response | Method return | code: 0 | |
| Failure | UNsupported_RESOURCE: The request comes for TTS which is not there on HU or not working. | JSON error message | Method return | code: 2 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | INVALID_DATA: The data sent is invalid (invalid JSON syntax, parameters are out of bounds or of wrong type) | | | code: 11 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

10.8.4 Sequence Diagrams

10.8.4.1 TTS.SetGlobalProperties and corresponding HMI processing



Note:

SDL-defined VR help items are provided by SDL via `UI.SetGlobalProperties` RPC.

10.8.5 JSON Messages Examples

10.8.5.1 Request

```
{
    "id" : 37,
```

```

"jsonrpc" : "2.0",
"method" : "TTS.SetGlobalProperties",
"params" :
{
    "helpPrompt" :
    [
        {"text" : "Yes",
        "text" : "No",
        "text" : "Skip"
    ],
    "timeoutPrompt" :
    [
        {"text" : "Please make a choice",
        "text" : "The time is about to expire"
    ],
    "appID" : 65542
}
}

```

10.8.5.2 Response

```
{
    "id" : 37,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "TTS.SetGlobalProperties"
    }
}
```

10.8.5.3 Error message

```
{
    "id" : 37,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 2,
        "message" : "TTS is not supported",
        "data" :
        {
            "method" : "TTS.SetGlobalProperties"
        }
    }
}
```

10.8.6 D-Bus Messages Examples

10.8.6.1 Request

```
[REDACTED]
```

10.8.6.2 Response

```
[REDACTED]
```

10.8.6.3 Failure

```
[REDACTED]
```

10.10 OnLanguageChange

10.10.1 Description

| | |
|----------|----------------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about TTS language change |

SDL needs to be in the know when the User changes the language of TTS: Upon the receipt of OnLanguageChange notification SDL will unregister the applications of different language to provide them with possibility to re-register with the correct (new HMI) TTS language.

HMI must:

Send the TTS.OnLanguageChange notification when the User switches TTS to another language and provide this new value via language parameter.

10.10.1.1 Parameters

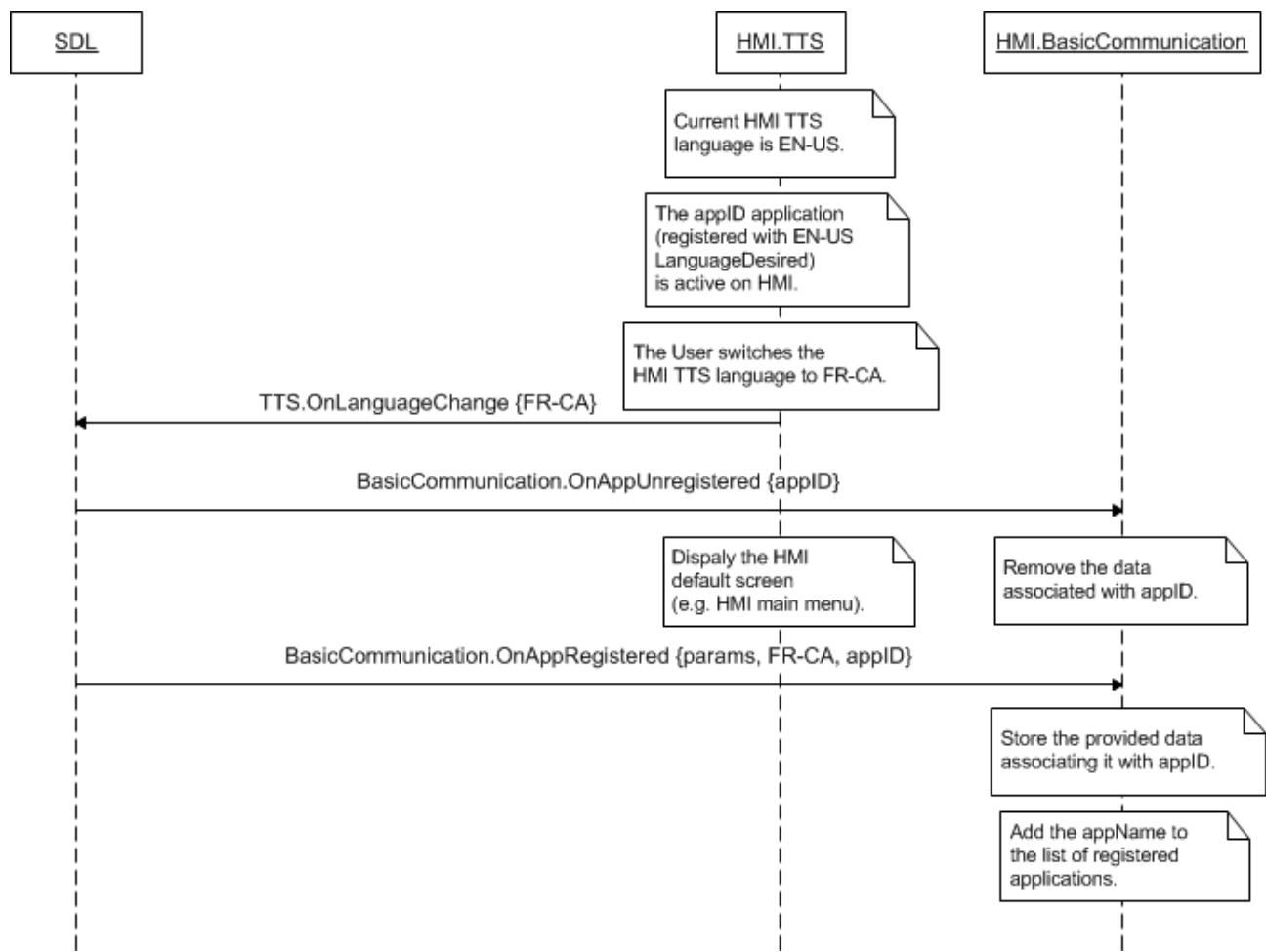
| Param Name | Type | Mandatory | Description |
|------------|-----------------|-----------|--|
| language | Common.Language | true | Language TTS has been switched to. See Language |

10.10.1.2 Language

| Element Name | Value | Description |
|--------------|-------|--------------------------------|
| EN-US | 0 | English – US |
| ES-MX | 1 | Spanish – Mexico |
| FR-CA | 2 | French – Canada |
| DE-DE | 3 | German – Germany |
| ES-ES | 4 | Spanish – Spain |
| EN-GB | 5 | English – GB |
| RU-RU | 6 | Russian - Russia |
| TR-TR | 7 | Turkish – Turkey |
| PL-PL | 8 | Polish – Poland |
| FR-FR | 9 | French – France |
| IT-IT | 10 | Italian – Italy |
| SV-SE | 11 | Swedish – Sweden |
| PT-PT | 12 | Portuguese – Portugal |
| NL-NL | 13 | Dutch (Standard) – Netherlands |
| EN-AU | 14 | English – Australia |
| ZH-CN | 15 | Mandarin – China |
| ZH-TW | 16 | Mandarin – Taiwan |
| JA-JP | 17 | Japanese – Japan |
| AR-SA | 18 | Arabic – Saudi Arabia |
| KO-KR | 19 | Korean – South Korea |
| PT-BR | 20 | Portuguese - Brazil |
| CS-CZ | 21 | Czech – Czech Republic |
| DA-DK | 22 | Danish – Denmark |
| NO-NO | 23 | Norwegian - Norway |

10.10.2 Sequence Diagrams

10.10.2.1 TTS.OnLanguageChange



10.10.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "TTS.OnLanguageChange",
    "params" :
    {
        "language" : IT-IT
    }
}
```

10.10.4 D-Bus Messages Examples

10.11 Started

10.10.1 Description

| | |
|---------|--------------|
| Type: | Notification |
| Sender: | HMI |

| | |
|-----------------|--------------------------|
| Purpose: | Inform about TTS started |
|-----------------|--------------------------|

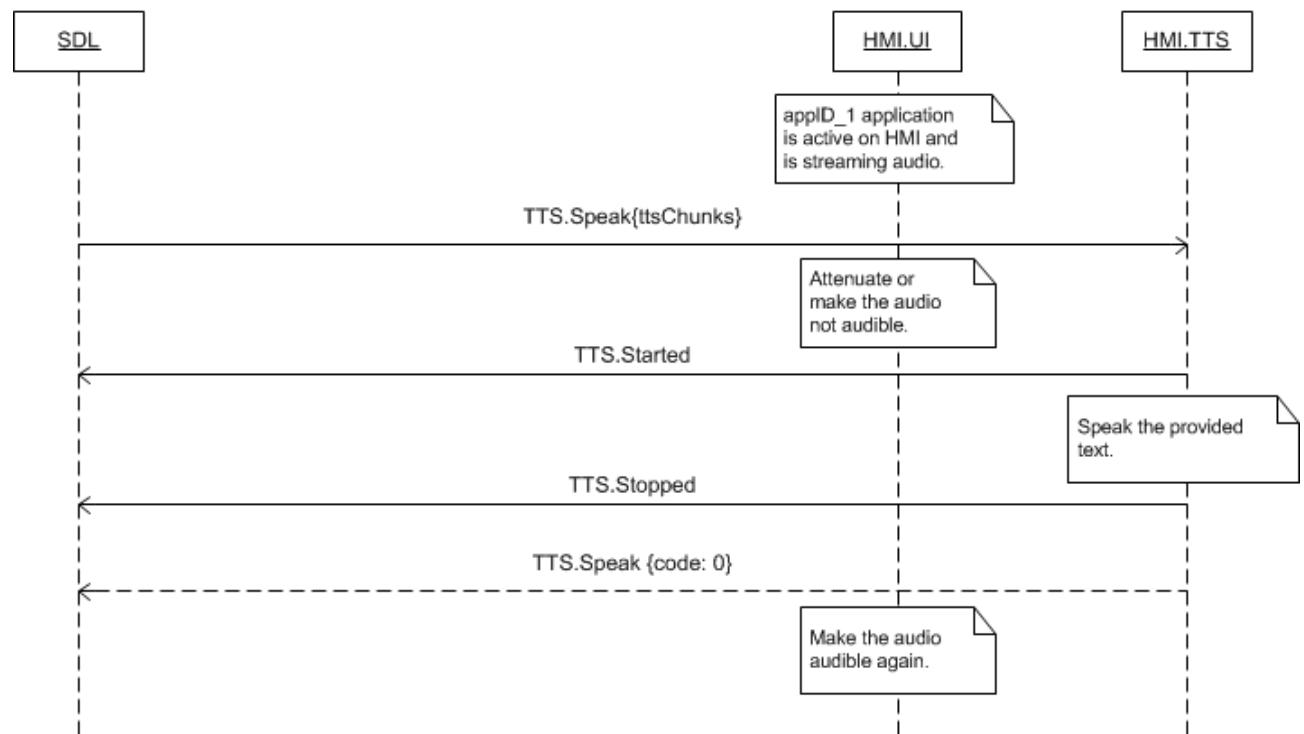
Once TTS starts speaking, HMI should attenuate the audio or make it not audible (depending on its capabilities). SDL needs to be notified about the event in order to provide a mobile application with the accurate information about audio streaming state on HMI.

HMI must:

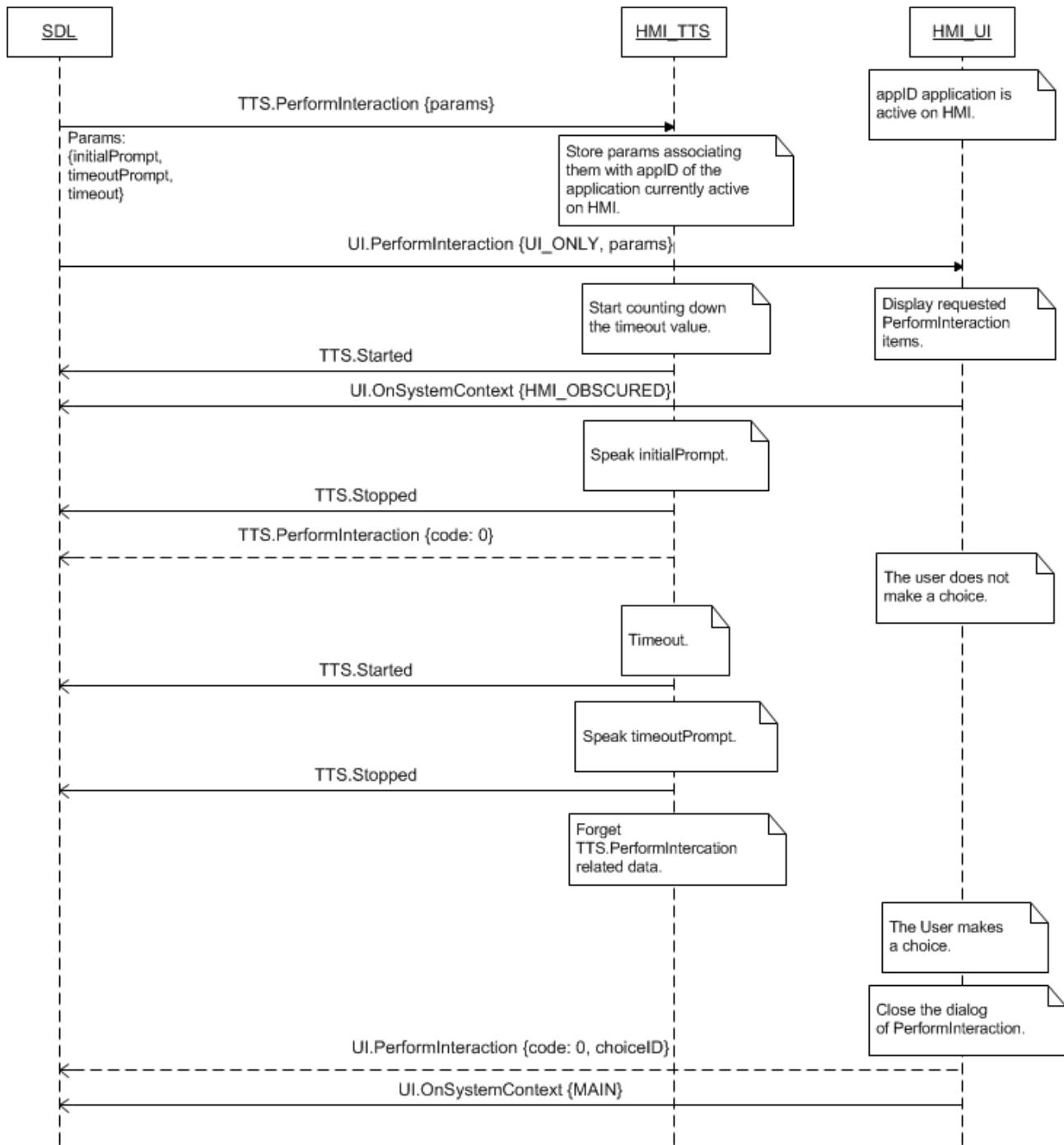
Send TTS.Started notification when TTS has started speaking either upon SDL's request (e.g. TTS.Speak) or upon HMI-defined prompting.

10.10.2 Sequence Diagrams

10.10.2.1 TTS.Started upon TTS.Speak request from SDL



10.10.2.2 TTS.Started during PerfromInteraction



10.10.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "TTS.Started",
}
```

10.10.4 D-Bus Messages Examples

10.12 Stopped

10.10.1 Description

| | |
|----------|---------------------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about TTS has stopped speaking |

When TTS module has finished speaking (started whatever by SDL via e.g. TTS.Speak or by HMI itself), HMI must notify SDL about the event via TTS.Stopped.

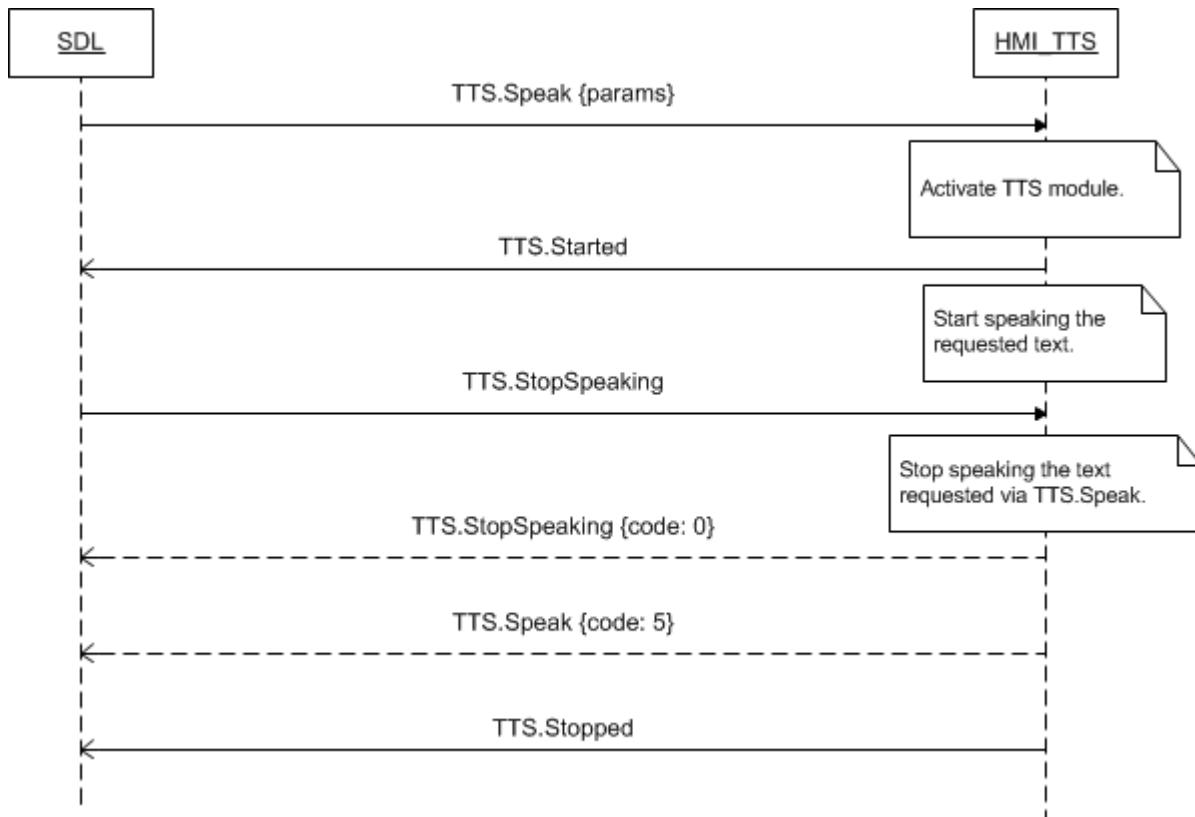
HMI must:

Send TTS.Stopped notification when TTS has stopped speaking

- By having finished speaking the designated text
- Having been aborted by
 - SDL (e.g. TTS.StopSpeaking)
 - User (e.g. PTT button press)
 - Event of higher priority (e.g. system or SDL-defined alert message)

10.10.2 Sequence Diagrams

10.10.2.1 TTS.Stopped after TTS.StopSpeaking from SDL



10.10.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "TTS.Stopped",
}
```

10.10.4 D-Bus Messages Examples

11 VehicleInfo Component Description

11.1 IsReady

11.1.1 Description

| | |
|----------|-----------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Know if component is ready. |

The request comes after HMI's readiness is confirmed via [OnReady](#) notification. SDL requires the information about whether the vehicle information can be collected and provided by HMI.

Note:

If VehicleInfo component is responded to be unavailable, SDL will not further send the requests related to it.

11.1.2 Request

11.1.2.1 Behavior

HMI must:

- Check whether VehicleInfo component is present and ready
- Respond correspondingly to results of this check.

11.1.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

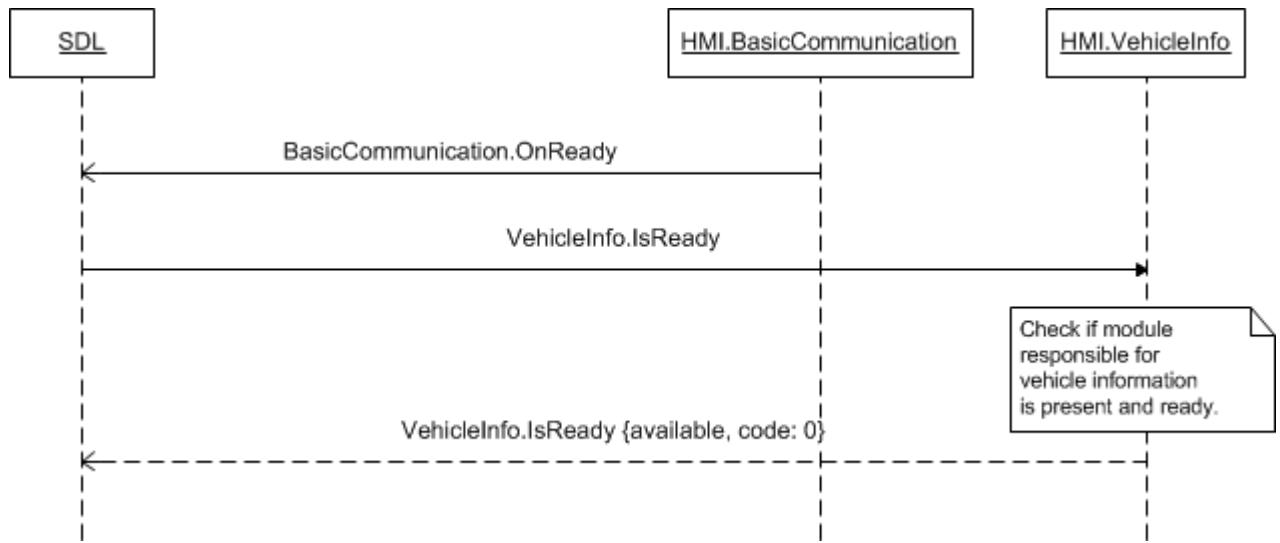
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|-----------------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the information about VehicleInfo availability. | JSON response | Method return | available, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | DATA_NOT_AVAILABLE: The information about VehicleInfo availability cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

11.1.3.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|---|
| availabe | Boolean | true | Must be <ul style="list-style-type: none">- 'true' if UI is present and ready- 'false' if not. |

11.1.4 Sequence Diagrams

11.1.4.1 VehicleInfo.IsReady and preceding OnReady



11.1.5 JSON Messages Examples

11.1.5.1 Request

```
{
    "id" : 17,
    "jsonrpc" : "2.0",
    "method" : "VehicleInfo.IsReady"
}
```

11.1.5.2 Response

```
{
    "id" : 17,
    "jsonrpc" : "2.0",
    "result" :
    {
        "available" : true,
        "code" : 0,
        "method" : "VehicleInfo.IsReady"
    }
}
```

11.1.5.3 Error message

```
{
    "id" : 17,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 9,
        "message" : "Data not available",
        "data" :
        {
            "method" : "VehicleInfo.IsReady"
        }
    }
}
```

11.1.6 D-Bus Messages Examples

11.1.6.1 Request

11.1.6.2 Response

| |
|--|
| |
|--|

11.1.6.3 Failure

| |
|--|
| |
|--|

11.2 GetVehicleType

11.2.1 Description

| | |
|----------|------------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Get the information about vehicle. |

SDL can store the information about the type of the host vehicle in `smartDeviceLink.ini` file stored in `SDL` directory. Still in current implementation, SDL will request the information via `VehicleInfo.GetVehicleType` after getting `OnReady` notification and response to `VehicleInfo.IsReady` from HMI.

11.2.2 Request

11.2.2.1 Behavior

HMI must:

1. Check for the information requested (make, model, trim and model year).
2. Provide this information in response message (see section 11.2.3).

11.2.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|-------------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the information about vehicle. | JSON response | Method return | vehicleType, code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | DATA_NOT_AVAILABLE: The information about vehicle cannot be provided. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

11.2.3.1 Parameters

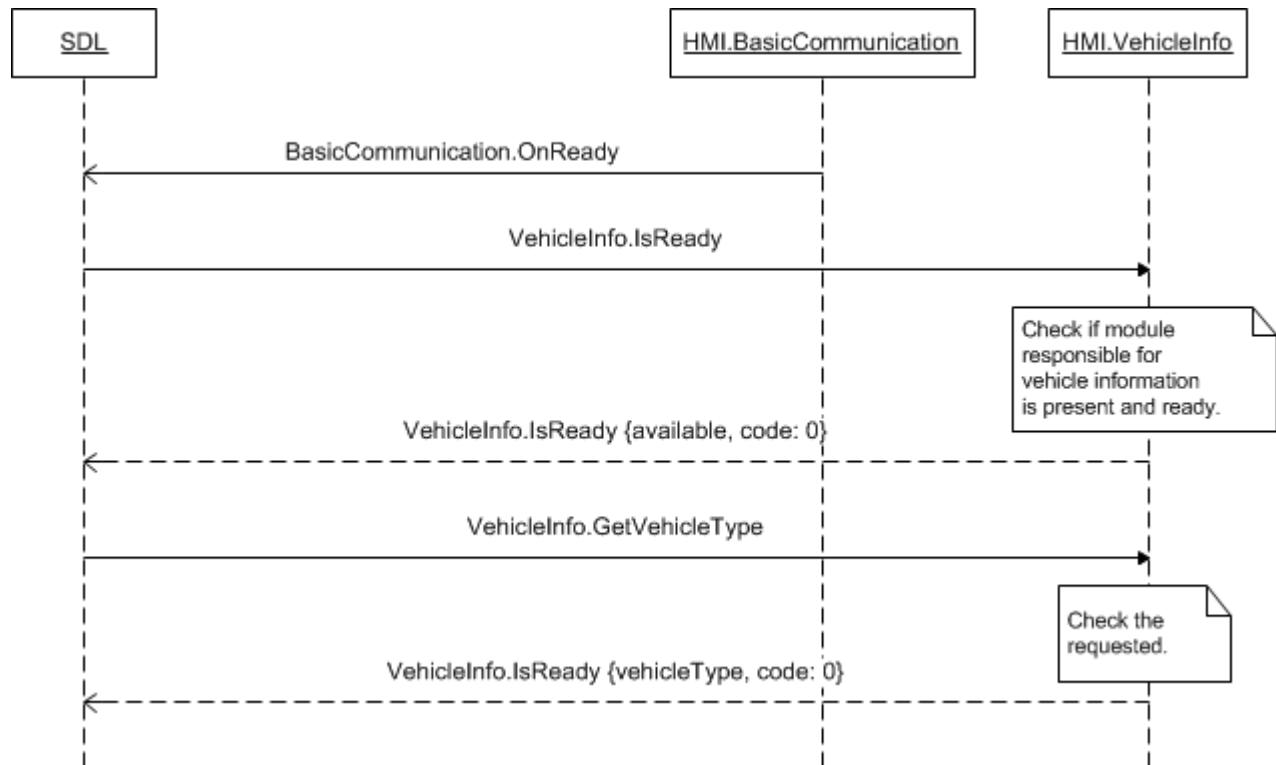
| Param Name | Type | Mandatory | Description |
|-------------|--------------------|-----------|-----------------|
| vehicleType | Common.VehicleType | true | See VehicleType |

11.2.3.2 VehicleType

| Param Name | Type | Mandatory | Additional | Description |
|------------|--------|-----------|-----------------|---------------------------------------|
| make | String | false | maxlength = 500 | Make of the vehicle (e.g. Ford) |
| model | String | false | maxlength = 500 | Model of the vehicle (e.g. Fiesta) |
| modelYear | String | false | maxlength = 500 | Model Year of the vehicle (e.g. 2013) |
| trim | String | false | maxlength = 500 | Trim of the vehicle (e.g. SE) |

11.2.4 Sequence Diagrams

11.2.4.1 GetVehicleType



11.2.5 JSON Messages Examples

11.2.5.1 Request

```
{
  "id" : 21,
  "jsonrpc" : "2.0",
  "method" : "VehicleInfo.GetVehicleType"
}
```

11.2.5.2 Response

```
{
  "id" : 21,
  "jsonrpc" : "2.0",
  "result" :
  {
    "vehicleType" :
    [
      {
        "make" : "Ford",
        "model" : "Fusion",
        "modelYear" : "2013",
        "trim" : "SE"
      }
    ]
  }
}
```

```

        "code" : 0,
        "method" : "VehicleInfo.GetVehicleType"
    }
}

```

11.2.5.3 Error message

```

{
    "id" : 21,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 9,
        "message" : "The requested data is not available",
        "data" :
        {
            "method" : "VehicleInfo.GetVehicleType"
        }
    }
}

```

11.2.6 D-Bus Messages Examples

11.2.6.1 Request

11.2.6.2 Response

11.2.6.3 Failure

11.3 ReadDID

11.3.1 Description

| | |
|----------|-------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Receive the known DIDs. |

SDL requests to provide the known DIDs, i.e. Data Identifiers: addressed memory locations of the Electronic Control Unit (ECU).

11.3.2 Request

11.3.2.1 Behavior

HMI must:

1. Check data in the requested locations (from `didLocation` parameter, which may be the array of addresses) on the named ECU (`ecuName` parameter).
2. Respond with one of the appropriate result codes (see section 11.3.3 Response for applicable result codes). And in case of SUCCESS return the array of `didResult`, each element of which refers the data from one of the requested locations:
 - Data itself (`data` parameter in `DIDResult` structure), which is the hex byte string of however many bytes stored at named location. May be missing in case of any kind of error occurred (see section 11.3.3.3 `VehicleDataresultCode`).
 - Address where it is taken from (`didLocation` parameter). Must be equal to the value from request.

- Individual result code (`resultCode` parameter), which defines the status of the data being provided.

11.3.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-------------|---------|-----------|---|--|
| ecuName | Integer | true | Minvalue = 0 Maxvalue = 65535 | Name of ECU. |
| didLocation | Integer | true | Array = true Minsize = 1 Maxsize = 1000 Minvalue = 0 Maxvalue = 65535 | The array of addresses of DID location on ECU. |
| appID | Integer | true | – | ID of the application that requested this RPC. |

11.3.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|-----------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides requested DIDs. | JSON response | Method return | didResult, code: 0 | |
| Failure | UNsupported_RESOURCE: The named ECU does not exist. | JSON error message | Method return | Code: 2 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | INVALID_DATA: The data sent is invalid (invalid JSON syntax, parameters out of bounds or of wrong type) | | | code: 11 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |
| | USER_DISALLOWED: Providing DIDs is disallowed by the User. | | | Code: 23 | |

11.3.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|------------------|-----------|---|---|
| didResult | Common.DIDResult | false | Array = true Minsize = 0 Maxsize = 1000 | Array of requested DID results (with data if available). See DIDResult |

11.3.3.2 DIDResult

| Param Name | Type | Mandatory | Additional | Description |
|------------|------------------------------|-----------|------------|---|
| resultCode | Common.VehicleDataResultCode | true | – | Individual DID result code. See VehicleDataResultCode. |

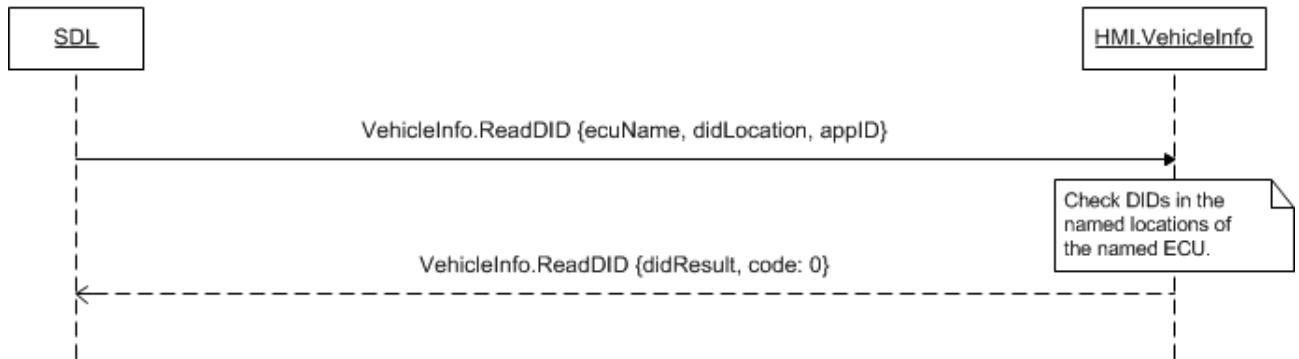
| Param Name | Type | Mandatory | Additional | Description |
|-------------|---------|-----------|----------------------------------|---|
| didLocation | Integer | true | minvalue = 0 maxvalue = 65535 | The address of DID location from the corresponding request. |
| data | String | false | maxlength = 5000 | The DID data which is the hex byte string of however many bytes are stored at that location |

11.3.3.3 VehicleDataResultCode

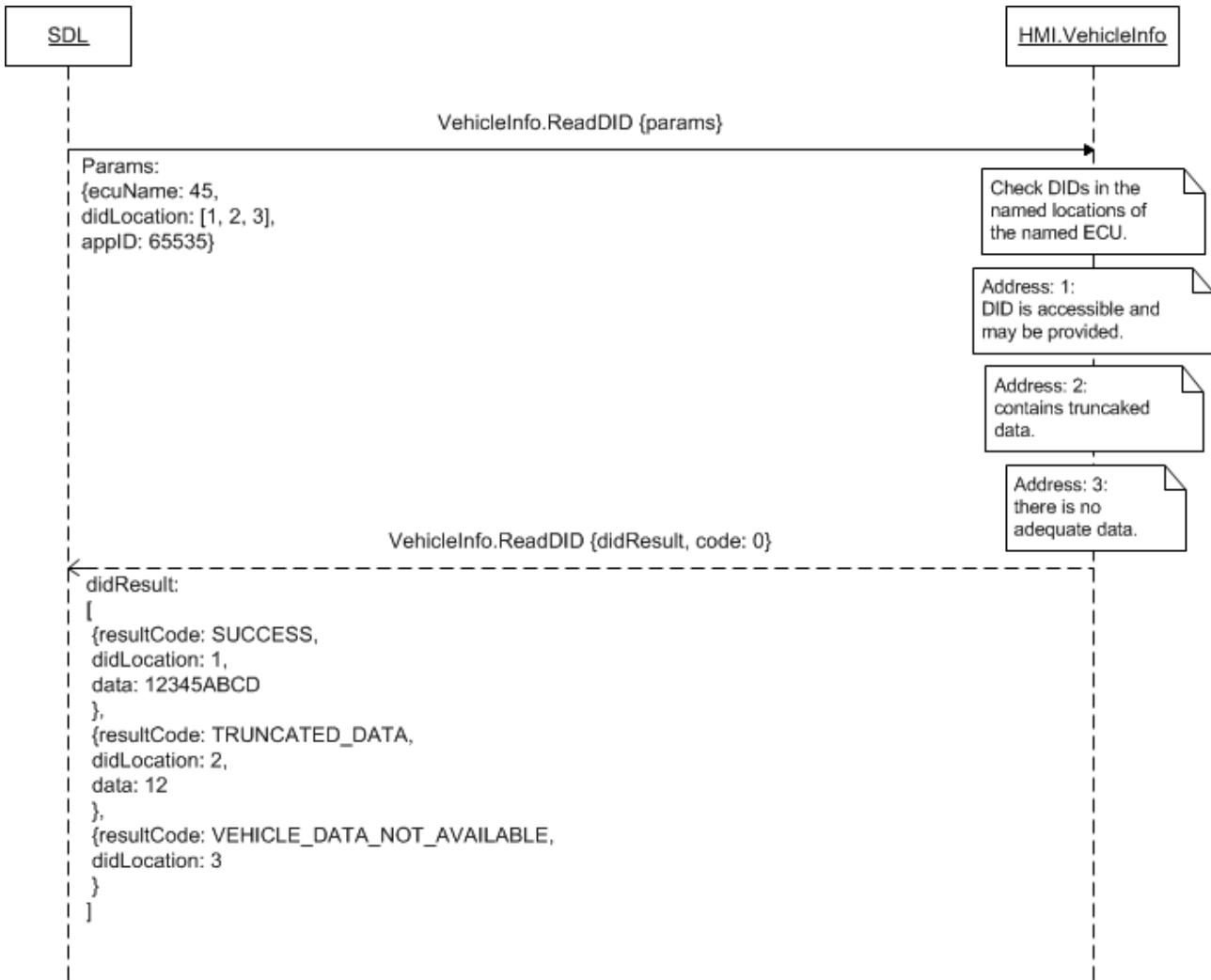
| Element name | Value | Short Description |
|----------------------------|-------|---|
| SUCCESS | 0 | The requested data is accessible. |
| TRUNCATED_DATA | 1 | The data is truncated: not all of the requested information is available. |
| DISALLOWED | 2 | The request is not authorized in local policies. |
| USER_DISALLOWED | 3 | The request is included in a functional group explicitly blocked by the User. |
| INVALID_ID | 4 | One of the provided IDs is not valid. |
| VEHICLE_DATA_NOT_AVAILABLE | 5 | The requested data is not available. |
| DATA_ALREADY_SUBSCRIBED | 6 | The requested data has been subscribed already. |
| DATA_NOT_SUBSCRIBED | 7 | The data is not subscribed. |
| IGNORED | 8 | The request is ignored because the intended result is already in effect. |

11.3.4 Sequence Diagrams

11.3.4.1 ReadDID general processing



11.3.4.2 ReeadDID with expanded didResult in response



11.3.5 JSON Messages Examples

11.3.5.1 Request

```
{
  "id" : 158,
  "jsonrpc" : "2.0",
  "method" : "VehicleInfo.ReadDID",
  "params" :
  {
    "ecuName" : 1287,
    "didLocation" : [35, 48, 182],
    "appID" : 93
  }
}
```

11.3.5.2 Response

```
{
  "id" : 158,
  "jsonrpc" : "2.0",
  "result" :
  {
    "didResult" :
    [
      {
        "resultCode" : SUCCESS,
        "didLocation" : 35,
        "data" : "38AF"
      },
      {
        "resultCode" : TRUNCATED_DATA,
        "didLocation" : 48
      }
    ]
  }
}
```

```

        "resultCode" : TRUNCATED_DATA,
        "didLocation" : 48,
        "data" : "35"
    },
    {
        "resultCode" : INVALID_ID,
        "didLocation" : 182
    }
],
"code" : 0,
"method" : "VehicleInfo.ReadDID"
}
}

```

11.3.5.3 Error message

```

{
    "id" : 158,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 2,
        "message" : "The requested ECU does not exist",
        "data" :
        {
            "method" : "VehicleInfo.ReadDID"
        }
    }
}

```

11.3.6 D-Bus Messages Examples

11.3.6.1 Request

[REDACTED]

11.3.6.2 Response

[REDACTED]

11.3.6.3 Failure

[REDACTED]

11.4 GetDTCs

11.4.1 Description

| | |
|-----------------|-------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Receive the known DTCs. |

SDL requests to provide the known Diagnostic Trouble Codes (DTCs) of the Electronic Control Unit (ECU).

11.4.2 Request

11.4.2.1 Behavior

HMI must:

1. Check the requested DTCs using provided information of `ecuName` and `dtcMask`.

2. Respond with one of the appropriate result codes (see section 11.3.3 Response for applicable result codes). And in case of SUCCESS return the following:
- 2 byte ECU Header (`ecuHeader` parameter) that must contain the information whether the below list of DTCs is truncated or not.
 - Array of all reported DTCs on module (`dtc` parameter), each of the elements is expected to consist of 4 bytes: 3 bytes of data and 1 byte of status.

11.4.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|----------------------------------|--|
| ecuName | Integer | true | minvalue = 0 maxvalue = 65535 | Name of ECU. |
| dtcMask | Integer | false | minvalue = 0 maxvalue = 255 | DTC Mask Byte to be sent in diagnostic request to module |
| appID | Integer | true | – | ID of the application that requested this RPC. |

11.4.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

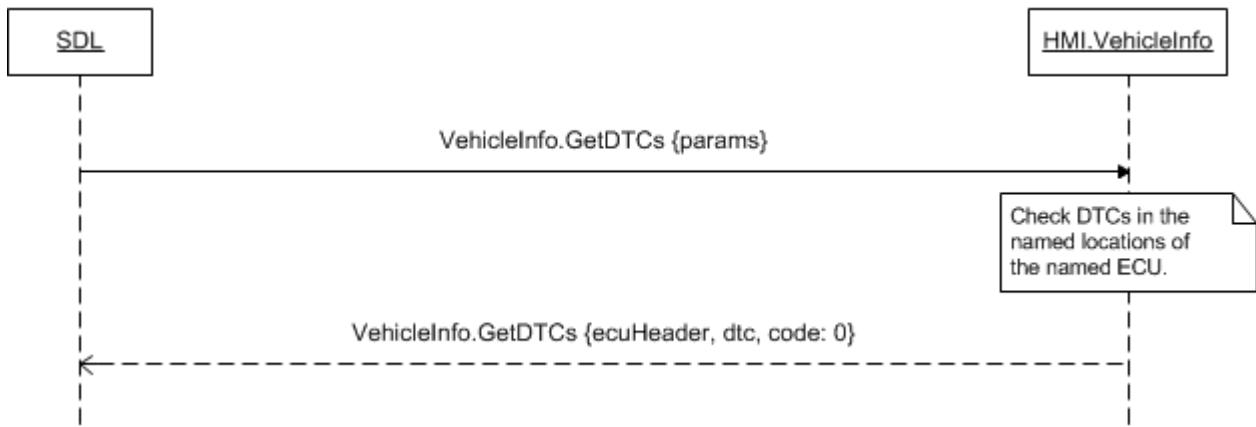
| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|-------------------------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides requested DTCs. | JSON response | Method return | ecuHeader, dtc, code: 0 | |
| Failure | UNsupported_RESOURCE: The named ECU does not exist. | JSON error message | Method return | code: 2 | Applicable for this RPC result codes. |
| | DATA_NOT_AVAILABLE: The named ECU exists, but no data can be provided.. | | | Code: 9 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |
| | USER_DISALLOWED: Providing DTCs is disallowed by the User. | | | Code: 23 | |

11.4.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|---|--|
| ecuHeader | Integer | true | minvalue = 0 maxvalue = 65535 | 2 byte ECU Header for DTC response. Contains the information whether the below list of DTCs is truncated. |
| dtc | String | false | Array = true Minsize = 1 Maxsize = 15 Maxlength = 10 | Array of all reported DTCs on module. Each DTC is represented with 4 bytes: - 3 bytes for data - 1 byte for status |

11.4.4 Sequence Diagrams

11.4.4.1 GetDTCs



11.4.5 JSON Messages Examples

11.4.5.1 Request

```
{
    "id" : 139,
    "jsonrpc" : "2.0",
    "method" : "VehicleInfo.GetDTCs",
    "params" :
    {
        "ecuName" : 56
        "dtcMask" : 84,
        "appID" : 65645
    }
}
```

11.4.5.2 Response

```
{
    "id" : 139,
    "jsonrpc" : "2.0",
    "result" :
    {
        "ecuHeader" : 6534,
        "dtc" : ["84752093", "28237", "748398"],
        "code" : 0,
        "method" : "VehicleInfo.GetDTCs"
    }
}
```

11.4.5.3 Error message

```
{
    "id" : 139,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 9,
        "message" : "Data not available",
        "data" :
        {
            "method" : "VehicleInfo.GetDTCs"
        }
    }
}
```

11.4.6 D-Bus Messages Examples

11.4.6.1 Request

| |
|--|
| |
|--|

11.4.6.2 Response

| |
|--|
| |
|--|

11.4.6.3 Failure

| |
|--|
| |
|--|

11.5 DiagnosticMessage

11.5.1 Description

| | |
|----------|--------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Vehicle diagnostic |

Non periodic vehicle diagnostic request.

11.5.2 Request

11.5.2.1 Behavior

HMI must:

1. Check the requested data using provided information of `targetID` (name of ECU), `messageLength` and `messageData`.
2. Respond with one of the appropriate result codes (see section 11.3.3 Response for applicable result codes). And in case of SUCCESS return `messageDataResult` which is an array of bytes comprising CAN message result.

11.5.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|---------------|---------|-----------|--|--|
| targetID | Integer | true | Minvalue = 0 Maxvalue = 65535 | Name of target ECU. |
| messageLength | Integer | true | Minvalue = 0 Maxvalue = 65535 | Length of message (in bytes). |
| messageData | Integer | true | Array = true Minsize = 1 Maxsize = 65535 Minvalue = 0 Maxvalue = 255 | Array of bytes comprising CAN message. |
| appID | Integer | true | - | ID of application that requested this RPC. |

11.5.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|---------------|---------------|-------------------------------|-------|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides requested DTCs. | JSON response | Method return | messageDataResult, code: 0 | |

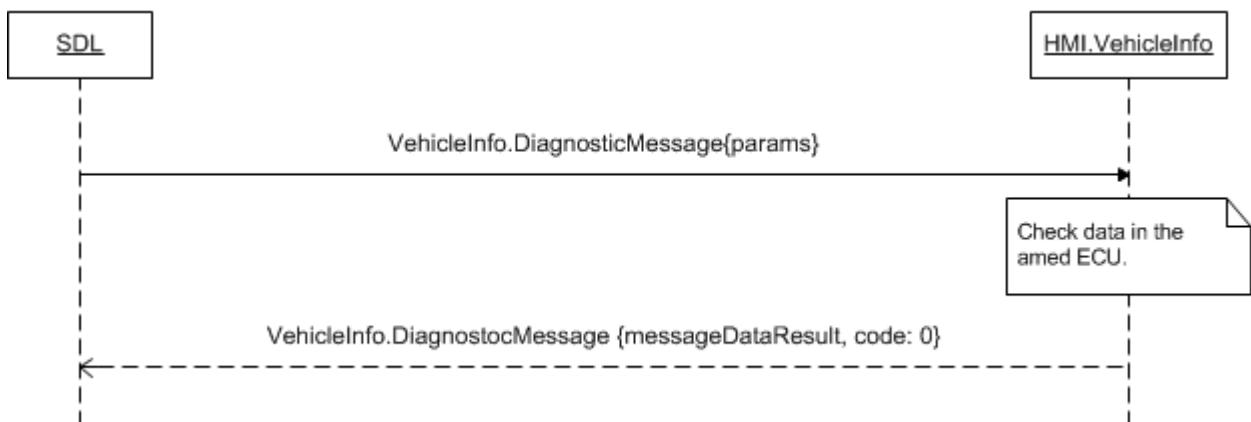
| | | | | | |
|---------|---|--------------------|---------------|----------|--|
| Failure | UNSUPPORTE_RESOURCE: The named ECU does not exist. | JSON error message | Method return | code: 2 | Applicable for this RPC result codes. |
| | DATA_NOT_AVAILABLE: The named ECU exists, but no data can be provided.. | | | Code: 9 | Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

11.5.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|---------|-----------|--|---|
| messageDataResult | Integer | true | Array = true Minsize = 1 Maxsize = 65535 Minvalue = 0 Maxvalue = 255 | Array of bytes comprising CAN message result. |

11.5.4 Sequence Diagrams

11.5.4.1 DiagnosticMessage



11.5.5 JSON Messages Examples

11.5.5.1 Request

```
{
  "id" : 139,
  "jsonrpc" : "2.0",
  "method" : "VehicleInfo.DiagnosticMessage",
  "params" :
  {
    "targetID" : 5456
    "messageLength" : 1084,
    "messageData" : [1,2,3,4,5,6,7,8,9]
  }
}
```

11.5.5.2 Response

```
{
  "id" : 139,
  "jsonrpc" : "2.0",
```

```

    "result" :
    {
        "messageDataResult" : [1,2,3,4,5,6],
        "code" : 0,
        "method" : "VehicleInfo.GetDTCs"
    }
}

```

11.5.5.3 Error message

```

{
    "id" : 139,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 9,
        "message" : "Data not available",
        "data" :
        {
            "method" : "VehicleInfo.GetDTCs"
        }
    }
}

```

11.5.6 D-Bus Messages Examples

11.5.6.1 Request

11.5.6.2 Response

11.5.6.3 Failure

11.6 SubscribeVehicleData

11.6.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Subscribe for definite vehicle data types updates |

SDL provides the data types for which periodic updates an application of requested appID would like to subscribe. In case of no errors, SDL expects to receive the information via OnVehicleData from HMI whenever the subscribed data type gets updates.

The request may come for the application being whatever active or in background on HMI.

11.6.2 Request

11.6.2.1 Behavior

HMI must:

1. Check whether the periodic updates can be provided for requested data type. The data type is defined with Boolean parameters of gps, speed, rpm and other that are provided in section 11.6.2.2:
 - Parameters of true value have to be subscribed

- One or several or all of the parameters from section 11.6.2.2 may be included into request..

Note:

Within a request SDL will provide at least one ‘ture’-value parameter of data type to be subscribed.

2. Remember the requested data type in the list of members that get OnVehicleData update notifications.

3. Respond with one of appropriate result codes (see section [11.6.3 Response](#)). And in case of SUCCESS return the parameter(s) of the name(s) equal to those requested for subscription. The returned parameters must contain

- Data type (see [section 11.6.3.3](#)) correspondent to the name of related parameter (for example, `gps : {VEHICLEDATA_GPS, SUCCESS}, speed : {VEHICLEDATA_SPEED, SUCCESS}`).
- An individual VehicleDataResult code from [section 11.6.3.4](#). The codes different from SUCCESS will inform SDL about the status of the named data type and whether the updates would be provided.

4. Provide OnVehicleData notifications whenever the successfully subscribed vehicle data type changes until SDL unsubscribes from updates via UnsubscribeVehicleData (see section 11.7).

HMI is recommended to:

Display the HMI-defined popup requesting the User for allowance to provide periodic updates of the named data type(s) for the named application.

11.6.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|------------------------|---------|-----------|---|
| gps | Boolean | false | Subscribe for GPS data updates. Information related to GPS data: number of satellites, compass direction, longitude, latitude, etc. See GPSData |
| speed | Boolean | false | Subscribe for vehicle speed updates to be provided in kilometers per hour. |
| rpm | Boolean | false | Subscribe for the number of engine revolutions per minute updates. |
| fuelLevel | Boolean | false | Subscribe for the fuel level in the tank updates (percentage). |
| fuelLevel_State | Boolean | false | Subscribe for the fuel level state updates: normal, low, etc. See ComponentVolumeStatus . |
| instantFuelConsumption | Boolean | false | Subscribe for the instantaneous fuel consumption updates to be provided in microliters. |
| externalTemperature | Boolean | false | Subscribe for the external temperature to be provided in degrees Celsius. |
| prndl | Boolean | false | Subscribe for gear stick position updates: first, second, etc. See PRNDL . |
| tirePressure | Boolean | false | Subscribe for tire pressure status updates: - The warning status (on, off, etc.) - The status of the tire pressure itself: normal, low, etc. See TireStatus . |
| odometer | Boolean | false | Subscribe for the information from odometer to be provided in km. |
| beltStatus | Boolean | false | Subscribe for the information of the seat belts status updates: deployed, buckled. See BeltStatus . |
| bodyInformation | Boolean | false | Subscribe for updates of body information including power modes: park brake status, ignition and ignition stable status. See BodyInformation . |
| deviceStatus | Boolean | false | Subscribe for updates of device status including signal and battery strength. See DeviceStatus . |
| driverBraking | Boolean | false | Subscribe for the information of the brake pedal status updates: on, off, etc. See VehicleDataEventStatus . |

| Param Name | Type | Mandatory | Description |
|--------------------|---------|-----------|---|
| wiperStatus | Boolean | false | Subscribe for updates of the wipers status: when they are manually on, manually off, stalled, etc. See WiperStatus. |
| headLampStatus | Boolean | false | Subscribe for the updates of the head lamps status: when they are on, off, etc. See HeadLampStatus. |
| engineTorque | Boolean | false | Subscribe for the updates of the torque value for engine (in Nm) on non-diesel variants |
| accPedalPosition | Boolean | false | Subscribe for the information of accelerator pedal position (percentage depressed). |
| steeringWheelAngle | Boolean | false | Subscribe for the information of current angle of the steering wheel (in degrees) |
| appID | Integer | true | Id of application that concerns this RPC. |

FORD specific data (extension of 11.6.2.2 table)

| Param Name | Type | Mandatory | Description |
|----------------|---------|-----------|---|
| eCallInfo | Boolean | false | Subscribe for Emergency Call notification and confirmation data. See ECallInfo. |
| airbagStatus | Boolean | false | Subscribe for the information on status of air bags updates. See AirbagStatus. |
| emergencyEvent | Boolean | false | Subscribe for the information related to an emergency event (if it occurred). See EmergencyEvent. |
| clusterModes | Boolean | false | Subscribe for the status modes of cluster updates. See ClusterModeStatus. |
| myKey | Boolean | false | Subscribe for the information related to MyKey feature. See MyKey. |

11.6.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|--|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI processes the request successfully and returns the information about the data type(s) and the corresponding individual subscription result code (see p.3 of section 11.6.2.1) | JSON response | Method return | Parameters of the names equal to requested ones, code: 0 | See 11.6.3.1 Parameters. |
| Failure | IGNORED: HMI has all of currently requested data types in the list of members that get OnVehicleData notifications after the previous request. | JSON error message | Method return | code: 6 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |
| | USER_DISALLOWED: The User disallowed providing periodic updates for all of the supported vehicle data types. | | | code: 23 | |

11.6.3.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------------------|--------------------------|-----------|---|
| gps | Common.VehicleDataResult | false | GPS data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_GPS must be used). |
| speed | Common.VehicleDataResult | false | Vehicle speed subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_SPEED must be used). |
| rpm | Common.VehicleDataResult | false | The engine number of revolutions per minute data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_RPM must be used). |
| fuelLevel | Common.VehicleDataResult | false | The fuel level data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_FUELLEVEL must be used). |
| fuelLevel_State | Common.VehicleDataResult | false | The fuel level state data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_FUELLEVEL_STATE must be used). |
| instantFuelConsumption | Common.VehicleDataResult | false | The instantaneous fuel consumption data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_FUELCONSUMPTION must be used). |
| externalTemperature | Common.VehicleDataResult | false | The external temperature data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_EXTERNTTEMP must be used). |
| prndl | Common.VehicleDataResult | false | The gear stick position data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_PRNDL must be used). |
| tirePressure | Common.VehicleDataResult | false | The tire pressure data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_TIREPRESSURE must be used). |
| odometer | Common.VehicleDataResult | false | Odometer data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_ODOMETER must be used). |
| beltStatus | Common.VehicleDataResult | false | The seat belts status data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_BELTSTATUS must be used). |
| bodyInformation | Common.VehicleDataResult | false | The body information data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_BODYINFO must be used). |
| deviceStatus | Common.VehicleDataResult | false | The device status data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_DEVICESTATUS must be used). |
| driverBraking | Common.VehicleDataResult | false | The brake pedal status data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_BRAKING must be used). |
| wiperStatus | Common.VehicleDataResult | false | The wipers status data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_WIPERSTATUS must be used). |
| headLampStatus | Common.VehicleDataResult | false | The head lamps status data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_HEADLAMPSTATUS must be used). |
| engineTorque | Common.VehicleDataResult | false | Torque value for engine data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_ENGINETORQUE must be used). |
| accPedalPosition | Common.VehicleDataResult | false | Accelerator pedal position data subscription status. See section 11.6.3.2 VehicleDataResult (VEHICLEDATA_ACCPEDAL must be used). |

| Param Name | Type | Mandatory | Description |
|--------------------|--------------------------|-----------|--|
| steeringWheelAngle | Common.VehicleDataResult | false | Current angle of the steering wheel data subscription status. See section 11.6.3.2 <i>VehicleDataResult</i> (VEHICLEDATA_STEERINGWHEEL must be used). |

FORD specific data (extension of 11.6.3.1 table)

| Param Name | Type | Mandatory | Description |
|----------------|--------------------------|-----------|---|
| eCallInfo | Common.VehicleDataResult | false | Emergency Call data subscription status. See section 11.6.3.2 <i>VehicleDataResult</i> (VEHICLEDATA_ECALLINFO must be used). |
| airbagStatus | Common.VehicleDataResult | false | The air bags status data subscription status. See section 11.6.3.2 <i>VehicleDataResult</i> (VEHICLEDATA_AIRBAGSTATUS must be used). |
| emergencyEvent | Common.VehicleDataResult | false | The emergency event data subscription status. See section 11.6.3.2 <i>VehicleDataResult</i> (VEHICLEDATA_EMERGENCYEVENT must be used). |
| clusterModes | Common.VehicleDataResult | false | Cluster modes status data subscription status. See section 11.6.3.2 <i>VehicleDataResult</i> (VEHICLEDATA_CLUSTERMODESTATUS must be used). |
| myKey | Common.VehicleDataResult | false | MyKey data subscription status. See section 11.6.3.2 <i>VehicleDataResult</i> (VEHICLEDATA_MYKEY must be used). |

11.6.3.2 *VehicleDataResult*

| Param Name | Type | Mandatory | Description |
|------------|------------------------------|-----------|---|
| dataType | Common.VehicleDataType | true | The data type being subscribed. Must correspond to the name of related parameter from section 11.6.3.1. |
| resultCode | Common.VehicleDataResultCode | true | Result code that defines the subscription status for the named data type. |

11.6.3.3 *VehicleDataType Enumeration*

| Element name | Value | Short Description |
|-----------------------------|-------|--|
| VEHICLEDATA_GPS | 0 | Parameter that must contain this value: gps |
| VEHICLEDATA_SPEED | 1 | Parameter that must contain this value: speed |
| VEHICLEDATA_RPM | 2 | Parameter that must contain this value: rpm |
| VEHICLEDATA_FUELLEVEL | 3 | Parameter that must contain this value: fuelLevel |
| VEHICLEDATA_FUELLEVEL_STATE | 4 | Parameter that must contain this value: fuelLevel_State |
| VEHICLEDATA_FUELCONSUMPTION | 5 | Parameter that must contain this value: instantFuelConsumption |
| VEHICLEDATA_EXTERNTTEMP | 6 | Parameter that must contain this value: externalTemperature |
| VEHICLEDATA_VIN | 7 | Not used for subscription |
| VEHICLEDATA_PRNDL | 8 | Parameter that must contain this value: prndl |
| VEHICLEDATA_TIREPRESSURE | 9 | Parameter that must contain this value: tirePressure |
| VEHICLEDATA_ODOMETER | 10 | Parameter that must contain this value: odometer |
| VEHICLEDATA_BELTSTATUS | 11 | Parameter that must contain this value: beltStatus |
| VEHICLEDATA_BODYINFO | 12 | Parameter that must contain this value: bodyInformation |
| VEHICLEDATA_DEVICESTATUS | 13 | Parameter that must contain this value: deviceStatus |
| VEHICLEDATA_ECALLINFO | 14 | Parameter that must contain this value: eCallInfo |

| Element name | Value | Short Description |
|-------------------------------|--------------|--|
| VEHICLEDATA_AIRBAGSTATUS | 15 | Parameter that must contain this value: airbagStatus |
| VEHICLEDATA_EMERGENCYEVENT | 16 | Parameter that must contain this value: emergencyEvent |
| VEHICLEDATA_CLUSTERMODESTATUS | 17 | Parameter that must contain this value: clusterModes |
| VEHICLEDATA_MYKEY | 18 | Parameter that must contain this value: myKey |
| VEHICLEDATA_BRAKING | 19 | Parameter that must contain this value: driverBraking |
| VEHICLEDATA_WIPERSTATUS | 20 | Parameter that must contain this value: wiperStatus |
| VEHICLEDATA_HEADLAMPSTATUS | 21 | Parameter that must contain this value: headLampStatus |
| VEHICLEDATA_BATTVOLTAGE | 22 | Not used for subscription |
| VEHICLEDATA_ENGINETORQUE | 23 | Parameter that must contain this value: engineTorque |
| VEHICLEDATA_ACCPEDAL | 24 | Parameter that must contain this value: accPedalPosition |
| VEHICLEDATA_STEERINGWHEEL | 25 | Parameter that must contain this value: steeringWheelAngle |

11.6.3.4 VehicleDataResultCode Enumeration

| Element name | Value | Short Description |
|----------------------------|--------------|--|
| SUCCESS | 0 | The named data type is successfully subscribed. Whenever it receives the updates, HMI will notify SDL about the event via OnVehicleData. |
| TRUNCATED_DATA | 1 | Not applicable. |
| DISALLOWED | 2 | Not applicable. |
| USER_DISALLOWED | 3 | Providing periodic updates for the named data type is disallowed by the User. |
| INVALID_ID | 4 | Not applicable. |
| VEHICLE_DATA_NOT_AVAILABLE | 5 | The named data type is not supported (or reported, published) by HMI. |
| DATA_ALREADY_SUBSCRIBED | 6 | The named data type is already marked by HMI to receive periodic update notifications. |
| DATA_NOT_SUBSCRIBED | 7 | Not applicable. |
| IGNORED | 8 | Not applicable. |

11.6.4 Sequence Diagrams

11.6.4.1 SubscribeVehicleData

11.6.5 JSON Messages Examples

11.6.5.1 Request

```
{
  "id" : 139,
  "jsonrpc" : "2.0",
  "method" : "VehicleInfo.SubscribeVehicleData",
  "params" :
  {
    "gps" : true,
    "speed" : true,
    "fuelLevel_State" : true,
    "externalTemperature" : true,
  }
}
```

```

        "prndl" : true,
        "tirePressure" : true,
        "odometer" : true,
        "beltStatus" : true,
        "bodyInformation" : true,
        "deviceStatus" : true,
        "wiperStatus" : true,
        "headLampStatus" : true,
        "accPedalPosition" : true,
    }
}

```

11.6.5.2 Response

```
{
    "id" : 139,
    "jsonrpc" : "2.0",
    "result" :
    {
        "gps" :
        {
            dataType : VEHICLEDATA_GPS,
            resultCode : SUCCESS
        },
        "speed" :
        {
            dataType : VEHICLEDATA_SPEED,
            resultCode : DATA_ALREADY_SUBSCRIBED
        },
        "fuelLevel_State" :
        {
            dataType : VEHICLEDATA_FUELLEVEL,
            resultCode : SUCCESS
        },
        "externalTemperature" :
        {
            dataType : VEHICLEDATA_EXTERNTemp,
            resultCode : VEHICLE_DATA_NOT_AVAILABLE
        },
        "prndl" :
        {
            dataType : VEHICLEDATA_PRNDL,
            resultCode : VEHICLE_DATA_NOT_AVAILABLE
        },
        "tirePressure" :
        {
            dataType : VEHICLEDATA_TIREPRESSURE,
            resultCode : SUCCESS
        },
        "odometer" :
        {
            dataType : VEHICLEDATA_ODOMETER,
            resultCode : SUCCESS
        },
        "beltStatus" :
        {
            dataType : VEHICLEDATA_BELTSTATUS,
            resultCode : SUCCESS
        },
    }
}
```

```

        "bodyInformation" :
        {
            dataType : VEHICLEDATA_BODYINFO,
            resultCode : SUCCESS
        },
        "deviceStatus" :
        {
            dataType : VEHICLEDATA_DEVICESTATUS,
            resultCode : DATA_ALREADY_SUBSCRIBED
        },
        "wiperStatus" :
        {
            dataType : VEHICLEDATA_WIPERSTATUS,
            resultCode : SUCCESS
        },
        "headLampStatus" :
        {
            dataType : HEADLAMPSTATUS,
            resultCode : SUCCESS
        },
        "accPedalPosition" :
        {
            dataType : VEHICLEDATA_ACCPEDAL,
            resultCode : VEHICLE_DATA_NOT_AVAILABLE
        },
        "code" : 0,
        "method" : "VehicleInfo.SubscribeVehicleData"
    }
}

```

11.6.5.3 Error message

```

{
    "id" : 139,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 6,
        "message" : "All of requested data types is subscribed already",
        "data" :
        {
            "method" : "VehicleInfo.SubscribeVehicleData"
        }
    }
}

```

11.6.6 D-Bus Messages Examples

11.6.6.1 Request

[REDACTED]

11.6.6.2 Response

[REDACTED]

11.6.6.3 Failure

[REDACTED]

11.7 UnSubscribeVehicleData

11.7.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Unsubscribe from definite vehicle data type(s) updates |

Initially SDL requests HMI via `SubscribeVehicleData` to provide periodic updates for the named data types. Once HMI responds successfully (see section 11.6 `SubscribeVehicleData`), it shall inform SDL upon every update of the named data type via `OnVehicleData` notification.

Via `UnsubscribeVehicleData` SDL requests HMI to stop sending updates for the named data type(s) previously subscribed.

The request may come for the application being whatever active or in background on HMI.

11.7.2 Request

11.7.2.1 Behavior

HMI must:

1. Check whether the named data type may be successfully unsubscribed. The data type is defined with Boolean parameters of `gps`, `speed`, `rpm` and other that are provided in section 11.7.2.2:

- Parameters of `true` value have to be unsubscribed
- One or several or all of the parameters from section 11.7.2.2 may be included into request.
- One or several or all of the parameters previously subscribed may be included into request.

Note:

Within a request SDL will provide at least one ‘true’-value parameter of data type to be unsubscribed.

2. Delete the requested data type from the list of members that get `OnVehicleData` update notifications.

3. Respond with one of appropriate result codes (see section 11.7.3 Response). And in case of `SUCCESS` return the parameter(s) of the name(s) equal to those requested for subscription. The returned parameters must contain

- Data type (see section 11.7.3.3) correspondent to the name of related parameter (for example, `gps : {VEHICLEDATA_GPS, SUCCESS}, speed : {VEHICLEDATA_SPEED, SUCCESS}`).
- An individual `VehicleDataResult` code from section 11.7.3.4. The codes different from `SUCCESS` will inform SDL about the status of the named data type and whether it is unsubscribed indeed.

4. Stop providing `OnVehicleData` notifications upon any changes of successfully unsubscribed vehicle data type.

11.7.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|------------------------------|---------|-----------|--|
| <code>gps</code> | Boolean | false | Unsubscribe from GPS data updates. Information related to GPS data: number of satellites, compass direction, longitude, latitude, etc. See <code>GPSData</code> |
| <code>speed</code> | Boolean | false | Unsubscribe from vehicle speed updates that are provided in kilometers per hour. |
| <code>rpm</code> | Boolean | false | Unsubscribe from the number of engine revolutions per minute updates. |
| <code>fuelLevel</code> | Boolean | false | Unsubscribe from the fuel level in the tank updates (percentage). |
| <code>fuelLevel_State</code> | Boolean | false | Unsubscribe from the fuel level state updates: normal, |

| Param Name | Type | Mandatory | Description |
|------------------------|---------|-----------|---|
| | | | low, etc. See ComponentVolumeStatus . |
| instantFuelConsumption | Boolean | false | Unsubscribe from the instantaneous fuel consumption updates to be provided in microliters. |
| externalTemperature | Boolean | false | Unsubscribe from the external temperature that is provided in degrees Celsius. |
| prndl | Boolean | false | Unsubscribe from gear stick position updates: first, second, etc. See PRNDL . |
| tirePressure | Boolean | false | Unsubscribe from tire pressure status updates: - The warning status (on, off, etc.) - The status of the tire pressure itself: normal, low, etc. See TireStatus . |
| odometer | Boolean | false | Unsubscribe from the information from odometer that is provided in km. |
| beltStatus | Boolean | false | Unsubscribe from the information of the seat belts status updates: deployed, buckled. See BeltStatus . |
| bodyInformation | Boolean | false | Unsubscribe from updates of body information including power modes: park brake status, ignition and ignition stable status. See BodyInformation . |
| deviceStatus | Boolean | false | Unsubscribe from updates of device status including signal and battery strength. See DeviceStatus . |
| driverBraking | Boolean | false | Unsubscribe from the information of the brake pedal status updates: on, off, etc. See VehicleDataEventStatus . |
| wiperStatus | Boolean | false | Unsubscribe from updates of the wipers status: when they are manually on, manually off, stalled, etc. See WiperStatus . |
| headLampStatus | Boolean | false | Unsubscribe from the updates of the head lamps status: when they are on, off, etc. See HeadLampStatus . |
| engineTorque | Boolean | false | Unsubscribe from the updates of the torque value for engine (in Nm) on non-diesel variants |
| accPedalPosition | Boolean | false | Unsubscribe from the information of accelerator pedal position (percentage depressed). |
| steeringWheelAngle | Boolean | false | Unsubscribe from the information of current angle of the steering wheel (in degrees) |
| appID | Integer | true | Id of application that requested this RPC. |

11.7.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|--|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI processes the request successfully and returns the information about the data type(s) and the corresponding individual unsubscription result code (see p.3 of section 11.7.2.1) | JSON response | Method return | Parameters of the names equal to requested ones, code: 0 | See 11.7.3.1 Parameters. |
| Failure | DATA_NOT_AVAILABLE: All of requested data types did not and will not receive update notifications since they are not supported (or reported, published) by HMI. | JSON error message | Method return | Code: 9 | Applicable for this RPC result codes. Please see Result |

| | | | | | |
|--|---|--|--|----------|--|
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Enumeration for all SDL-supported codes. |
|--|---|--|--|----------|--|

11.7.3.1 Parameters

| Param Name | Type | Mandatory | Description | | |
|------------------------|--------------------------|-----------|--|--|--|
| gps | Common.VehicleDataResult | false | GPS data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_GPS must be used). | | |
| speed | Common.VehicleDataResult | false | Vehicle speed unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_SPEED must be used). | | |
| rpm | Common.VehicleDataResult | false | The engine number of revolutions per minute data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_RPM must be used). | | |
| fuelLevel | Common.VehicleDataResult | false | The fuel level data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_FUELLEVEL must be used). | | |
| fuelLevel_State | Common.VehicleDataResult | false | The fuel level state data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_FUELLEVEL_STATE must be used). | | |
| instantFuelConsumption | Common.VehicleDataResult | false | The instantaneous fuel consumption data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_FUELCONSUMPTION must be used). | | |
| externalTemperature | Common.VehicleDataResult | false | The external temperature data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_EXTERNTEMP must be used). | | |
| prndl | Common.VehicleDataResult | false | The gear stick position data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_PRNDL must be used). | | |
| tirePressure | Common.VehicleDataResult | false | The tire pressure data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_TIREPRESSURE must be used). | | |
| odometer | Common.VehicleDataResult | false | Odometer data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_ODOMETER must be used). | | |
| beltStatus | Common.VehicleDataResult | false | The seat belts status data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_BELTSTATUS must be used). | | |
| bodyInformation | Common.VehicleDataResult | false | The body information data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_BODYINFO must be used). | | |
| deviceStatus | Common.VehicleDataResult | false | The device status data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_DEVICESTATUS must be used). | | |
| driverBraking | Common.VehicleDataResult | false | The brake pedal status data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_BRAKING must be used). | | |
| wiperStatus | Common.VehicleDataResult | false | The wipers status data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_WIPERSTATUS must be used). | | |
| headLampStatus | Common.VehicleDataResult | false | The head lamps status data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_HEADLAMPSTATUS must be used). | | |
| engineTorque | Common.VehicleDataResult | false | Torque value for engine data unsubscription status. See section 11.7.3.2 VehicleDataResult | | |

| Param Name | Type | Mandatory | Description |
|--------------------|--------------------------|-----------|--|
| | | | (VEHICLEDATA_ENGINETORQUE must be used). |
| accPedalPosition | Common.VehicleDataResult | false | Accelerator pedal position data sunscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_ACCPEDAL must be used). |
| steeringWheelAngle | Common.VehicleDataResult | false | Current angle of the steering wheel data unsubscription status. See section 11.7.3.2 VehicleDataResult (VEHICLEDATA_STEERINGWHEEL must be used). |

11.7.3.2 VehicleDataResult

| Param Name | Type | Mandatory | Description |
|------------|------------------------------|-----------|---|
| dataType | Common.VehicleDataType | true | The data type being unsubscribed. Must correspond to the name of related parameter from section 11.7.3.1. |
| resultCode | Common.VehicleDataResultCode | true | Result code that defines the unsubscription status for the named data type. |

11.7.3. VehicleDataType Enumeration

| Element name | Value | Short Description |
|-------------------------------|-------|--|
| VEHICLEDATA_GPS | 0 | Parameter that must contain this value: gps |
| VEHICLEDATA_SPEED | 1 | Parameter that must contain this value: speed |
| VEHICLEDATA_RPM | 2 | Parameter that must contain this value: rpm |
| VEHICLEDATA_FUELLEVEL | 3 | Parameter that must contain this value: fuelLevel |
| VEHICLEDATA_FUELLEVEL_STATE | 4 | Parameter that must contain this value: fuelLevel_State |
| VEHICLEDATA_FUELCONSUMPTION | 5 | Parameter that must contain this value: instantFuelConsumption |
| VEHICLEDATA_EXTERNTTEMP | 6 | Parameter that must contain this value: externalTemperature |
| VEHICLEDATA_VIN | 7 | Not used for unsubscription |
| VEHICLEDATA_PRNDL | 8 | Parameter that must contain this value: prndl |
| VEHICLEDATA_TIREPRESSURE | 9 | Parameter that must contain this value: tirePressure |
| VEHICLEDATA_ODOMETER | 10 | Parameter that must contain this value: odometer |
| VEHICLEDATA_BELTSTATUS | 11 | Parameter that must contain this value: beltStatus |
| VEHICLEDATA_BODYINFO | 12 | Parameter that must contain this value: bodyInformation |
| VEHICLEDATA_DEVICESTATUS | 13 | Parameter that must contain this value: deviceStatus |
| VEHICLEDATA_ECALLINFO | 14 | Parameter that must contain this value: eCallInfo |
| VEHICLEDATA_AIRBAGSTATUS | 15 | Parameter that must contain this value: airbagStatus |
| VEHICLEDATA_EMERGENCYEVENT | 16 | Parameter that must contain this value: emergencyEvent |
| VEHICLEDATA_CLUSTERMODESTATUS | 17 | Parameter that must contain this value: clusterModes |
| VEHICLEDATA_MYKEY | 18 | Parameter that must contain this value: myKey |
| VEHICLEDATA_BRAKING | 19 | Parameter that must contain this value: driverBraking |
| VEHICLEDATA_WIPERSTATUS | 20 | Parameter that must contain this value: wiperStatus |
| VEHICLEDATA_HEADLAMPSTATUS | 21 | Parameter that must contain this value: headLampStatus |
| VEHICLEDATA_BATTVOLTAGE | 22 | Not used for unsubscription |
| VEHICLEDATA_ENGINETORQUE | 23 | Parameter that must contain this value: engineTorque |

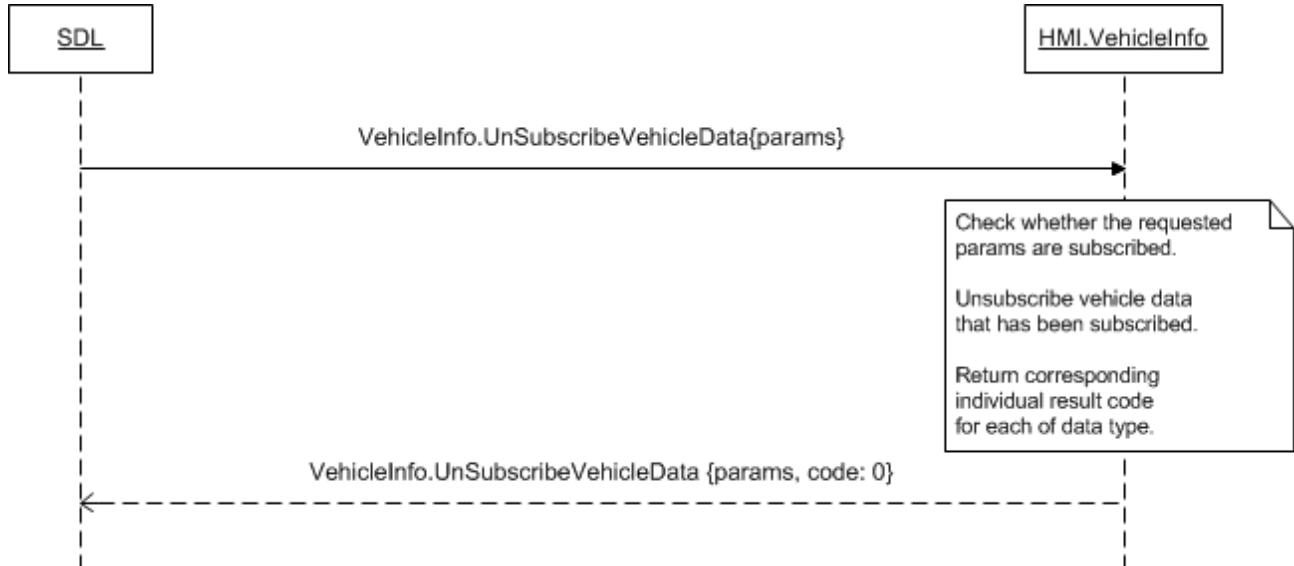
| Element name | Value | Short Description |
|---------------------------|-------|---|
| VEHICLEDATA_ACCPEDAL | 24 | Parameter that must contain this value: accPedalPosition |
| VEHICLEDATA_STEERINGWHEEL | 25 | Parameter that must contain this value: steeringWheelAngle |

11.7.3. VehicleDataresultCode Enumeration

| Element name | Value | Short Description |
|----------------------------|-------|--|
| SUCCESS | 0 | The named data type is successfully unsubscribed. HMI shall NOT notify SDL about any changes of this data type further more. |
| TRUNCATED_DATA | 1 | Not applicable. |
| DISALLOWED | 2 | Not applicable. |
| USER_DISALLOWED | 3 | Not applicable. |
| INVALID_ID | 4 | Not applicable. |
| VEHICLE_DATA_NOT_AVAILABLE | 5 | The named data type did not and will not receive update notifications since it is not supported (or reported, published) by HMI. |
| DATA_ALREADY_SUBSCRIBED | 6 | Not applicable. |
| DATA_NOT_SUBSCRIBED | 7 | The named data type cannot be unsubscribed because it has not been subscribed yet. |
| IGNORED | 8 | Not applicable. |

11.7.4 Sequence Diagrams

11.7.4.1 UnSubscribeVehicleData



11.7.5 JSON Messages Examples

11.7.5.1 Request

```
{
  "id" : 139,
  "jsonrpc" : "2.0",
  "method" : "VehicleInfo.UnsubscribeVehicleData",
  "params" :
  {
    "gps" : true,
  }
}
```

```

        "speed" : true,
        "fuelLevel_State" : true,
        "externalTemperature" : true,
        "prndl" : true,
        "tirePressure" : true,
        "odometer" : true,
        "beltStatus" : true,
        "bodyInformation" : true,
        "deviceStatus" : true,
        "wiperStatus" : true,
        "headLampStatus" : true,
        "accPedalPosition" : true,
        "myKey" : true
    }
}

```

11.7.5.2 Response

```

{
    "id" : 139,
    "jsonrpc" : "2.0",
    "result" :
    {
        "gps" :
        {
            "dataType" : VEHICLEDATA_GPS,
            "resultCode" : SUCCESS
        },
        "speed" :
        {
            "dataType" : VEHICLEDATA_SPEED,
            "resultCode" : DATA_NOT_SUBSCRIBED
        },
        "fuelLevel_State" :
        {
            "dataType" : VEHICLEDATA_FUELLEVEL,
            "resultCode" : SUCCESS
        },
        "externalTemperature" :
        {
            "dataType" : VEHICLEDATA_EXTERNTemp,
            "resultCode" : DATA_NOT_SUBSCRIBED
        },
        "prndl" :
        {
            "dataType" : VEHICLEDATA_PRNDL,
            "resultCode" : DATA_NOT_SUBSCRIBED
        },
        "tirePressure" :
        {
            "dataType" : VEHICLEDATA_TIREPRESSURE,
            "resultCode" : SUCCESS
        },
        "odometer" :
        {
            "dataType" : VEHICLEDATA_ODOMETER,
            "resultCode" : SUCCESS
        },
        "beltStatus" :

```

```

{
    dataType : VEHICLEDATA_BELTSTATUS,
    resultCode : SUCCESS
} ,

"bodyInformation" :
{
    dataType : VEHICLEDATA_BODYINFO,
    resultCode : SUCCESS
} ,

"deviceStatus" :
{
    dataType : VEHICLEDATA_DEVICESTATUS,
    resultCode : DATA_NOT_SUBSCRIBED
} ,

"wiperStatus" :
{
    dataType : VEHICLEDATA_WIPERSTATUS,
    resultCode : SUCCESS
} ,

"headLampStatus" :
{
    dataType : HEADLAMPSTATUS,
    resultCode : SUCCESS
} ,

"accPedalPosition" :
{
    dataType : VEHICLEDATA_ACCPEDAL,
    resultCode : DATA_NOT_SUBSCRIBED
} ,

"code" : 0,
"method" : "VehicleInfo.UnsubscribeVehicleData"
}
}

```

11.7.5.3 Error message

```

{
    "id" : 139,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 22,
        "message" : "An unknown error occurred",
        "data" :
        {
            "method" : "VehicleInfo.UnsubscribeVehicleData"
        }
    }
}

```

11.7.6 D-Bus Messages Examples

11.7.6.1 Request

11.7.6.2 Response

11.7.6.3 Failure

| |
|--|
| |
|--|

11.8 GetVehicleData

11.8.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Receive current values of vehicle data |

SDL sends GetVehicleData (in contrast to [SubscribeVehicleData](#)) for getting the current value(s) of the named data type(s).

11.8.2 Request

11.8.2.1 Behavior

HMI must:

1. Check whether the current values of requested data type can be provided. The data type is defined with Boolean parameters of gps, speed, rpm and other that are provided in section 11.8.2.2:
 - true value parameters are those which require information in response.
 - One or several or all of the parameters from section 11.8.2.2 may be included into request.

Note:

Within a request SDL will provide at least one ‘ture’-value parameter of data type the current values of which must be provided.

2. Respond with one of appropriate result codes (see section 11.8.3 Response).

2.1. HMI must return the parameter(s) of the name(s) equal to those requested, only together with result codes of:

- SUCCESS: when HMI can provide the current values of all requested data types.
- SUCCESS when HMI can provide just one or several values of the total requested (that is, less than requested). The example cases: when sensor is broken or HMI does not support or publish some data type.

See the table below for clarification.

| Request | Response | | |
|-------------|----------------------------|-----------------------------|-----------------------------|
| | code: SUCCESS | code: SUCCESS | code: SUCCESS |
| | data available: all | data available: some | data available: none |
| data_type_1 | data_type_1 | data_type_1 | |
| data_type_2 | data_type_2 | data_type_3 | |
| data_type_3 | data_type_3 | data_type_4 | |
| data_type_4 | data_type_4 | data_type_5 | |
| data_type_5 | data_type_5 | | |

- 2.2. Every of parameters returned must contain the appropriate values as described in section 11.8.3.1.

11.8.2.2 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|---|
| gps | Boolean | false | Get GPS data current values. Information related to GPS data: number of satellites, compass direction, longitude, latitude, etc. See GPSData |
| speed | Boolean | false | Get vehicle speed current value to be provided in kilometers per hour. |

| Param Name | Type | Mandatory | Description |
|------------------------|---------|-----------|---|
| rpm | Boolean | false | Get the number of engine revolutions per minute current value. |
| fuelLevel | Boolean | false | Get the fuel level in the tank current value (percentage). |
| fuelLevel_State | Boolean | false | Get the fuel level state current value: normal, low, etc. See ComponentVolumeStatus. |
| instantFuelConsumption | Boolean | false | Get the instantaneous fuel consumption current value to be provided in microliters. |
| externalTemperature | Boolean | false | Get the external temperature current value to be provided in degrees Celsius. |
| vin | Boolean | false | Vehicle identification number. |
| prndl | Boolean | false | Get the gear stick position current value: first, second, etc. See PRNDL. |
| tirePressure | Boolean | false | Get tire pressure status current values: - The warning status (on, off, etc.) - The status of the tire pressure itself: normal, low, etc. See TireStatus. |
| odometer | Boolean | false | Get the current information from odometer to be provided in km. |
| beltStatus | Boolean | false | Get the current information of the seat belts status: deployed, buckled. See BeltStatus. |
| bodyInformation | Boolean | false | Get the current values of body information including power modes: park brake status, ignition and ignition stable status. See BodyInformation. |
| deviceStatus | Boolean | false | Get the current values of device status including signal and battery strength. See DeviceStatus. |
| driverBraking | Boolean | false | Get the current information of the brake pedal status updates: on, off, etc. See VehicleDataEventStatus. |
| wiperStatus | Boolean | false | Get the current information of the wipers status: whether they are manually on, manually off, stalled, etc. See WiperStatus. |
| headLampStatus | Boolean | false | Get the current information of the head lamps status: when they are on, off, etc. See HeadLampStatus. |
| engineTorque | Boolean | false | Get the current information of the torque value for engine (in Nm) on non-diesel variants |
| accPedalPosition | Boolean | false | Get the current information of accelerator pedal position (percentage depressed). |
| steeringWheelAngle | Boolean | false | Get the current information of current angle of the steering wheel (in degrees) |
| eCallInfo | Boolean | false | Get the current information about whether the 'eCall' Event is currently active. |
| airbagStatus | Boolean | false | Get the current information of air-bags current status. |
| emergencyEvent | Boolean | false | Get the current information about whether the 'emergency' event is currently active. |
| clusterModeStatus | Boolean | false | Get the current information about the status modes of the cluster |
| myKey | Boolean | false | Get the current information about 'MyKey' feature. |
| appID | Integer | true | Id of application that requested this RPC. |

11.8.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|--|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides values of all requested vehicle data. | JSON response | Method return | Parameters of the names equal to requested ones, code: 0 | See 11.8.3.1 Parameters . |
| Failure | DATA_NOT_AVAILABLE: 1. HMI can provide just one or several values of the total requested (that is, less than requested). | JSON error message | Method return | Parameters of the names equal to requested ones, code: 9 | See 11.8.2.1. Behavior . |
| | DATA_NOT_AVAILABLE: 2. HMI can provide none of the requested data types. | | | code: 9 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |

11.8.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------------------|------------------------------|-----------|----------------------------------|---|
| gps | Common.GPSData | false | - | The current values of GPS data: number of satellites, compass direction, longitude, latitude, etc. See section 11.8.3.2 GPSData |
| speed | Float | false | minvalue = 0 maxvalue = 700 | The current value of vehicle speed in kilometers per hour. |
| rpm | Integer | false | minvalue = 0 maxvalue = 20000 | The current number of revolutions per minute of the engine. |
| fuelLevel | Float | false | minvalue = -6 maxvalue = 106 | The current value of fuel level in the tank (percentage). |
| fuelLevelState | Common.ComponentVolumeStatus | false | - | The current value of fuel level state: normal, low, etc. See ComponentVolumeStatus |
| instantFuelConsumption | Float | false | minvalue = 0 maxvalue = 25575 | The current value of instantaneous fuel consumption in microliters. |
| externalTemperature | Float | false | minvalue = -40 maxvalue = 100 | The current external temperature in degrees Celsius. |
| vin | String | false | maxlength = 17 | Vehicle identification number. |
| prndl | Common.PRNDL | false | - | The current position of gear stick: first, second, etc. See PRNDL . |
| tirePressure | Common.TireStatus | false | - | The current information of: - The warning status (on, off, etc.) - The status of the tire pressure itself: normal, low, etc. See TireStatus . |

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|-------------------------------|------------------|-------------------------------------|--|
| odometer | Integer | false | minvalue = 0 maxvalue = 17000000 | Current value of odometer in km. |
| beltStatus | Common.BeltStatus | false | - | The current status of the seat belts: deployed, buckled. See BeltStatus |
| bodyInformation | Common.BodyInformation | false | - | The cuurent body information: park brake status, ignition and ignition stable status. See BodyInformation |
| deviceStatus | Common.DeviceStatus | false | - | The current information of device status including signal and battery strength. See DeviceStatus. |
| driverBraking | Common.VehicleDataEventStatus | false | - | The current status of the brake pedal: on, off, etc. See VehicleDataEventStatus |
| wiperStatus | Common.WiperStatus | false | - | The current status of the wipers: whether they are manually on, manually off, stalled, etc. See WiperStatus |
| headLampStatus | Common.HeadLampStatus | false | - | The current status of the head lamps: whether they are on, off, etc. See HeadLampStatus |
| engineTorque | Float | false | minvalue = -1000 maxvalue = 2000 | The current torque value for engine (in Nm) for non-diesel models. |
| accPedalPosition | Float | false | minvalue = 0 maxvalue = 100 | The current value of accelerator pedal position (percentage depressed). |
| steeringWheelAngle | Float | false | minvalue = -2000 maxvalue = 2000 | Current angle of the steering wheel (in degrees). |
| eCallInfo | Common.ECallInfo | false | - | |
| airbagStatus | Common.AirbagStatus | false | - | |
| emergencyEvent | Common.EmergencyEvent | false | - | |
| clusterModeStatus | Common.ClusterModeStatus | false | - | |
| myKey | Common.MyKey | false | - | |

11.8.3.2 GPSData

| Param Name | Type | Mandatory | Maxvalue | Description |
|-------------------|-------------|------------------|-----------------------------------|--------------------------------|
| longitudeDegrees | Float | false | minvalue = -180 maxvalue = 180 | Position longitude in degrees. |

| Param Name | Type | Mandatory | Maxvalue | Description |
|------------------|-------------------------|-----------|---------------------------------------|--|
| latitudeDegrees | Float | false | minvalue = -90 maxvalue = 90 | Position latitude in degrees. |
| utcYear | Integer | false | minvalue = 2010 maxvalue = 2100 | The current UTC year. |
| utcMonth | Integer | false | minvalue = 1 maxvalue = 12 | The current UTC month. |
| utcDay | Integer | false | minvalue = 1 maxvalue = 31 | The current UTC day. |
| utcHours | Integer | false | minvalue = 0 maxvalue = 23 | The current UTC hour. |
| utcMinutes | Integer | false | minvalue = 0 maxvalue = 59 | The current UTC minute. |
| utcSeconds | Integer | false | minvalue = 0 maxvalue = 59 | The current UTC second. |
| compassDirection | Common.CompassDirection | false | - | The compass direction: north, east, etc. See section 11.8.3.3 CompassDirection |
| pdop | Float | false | minvalue = 0 maxvalue = 10 | Positional Dilution Of Precision. |
| hdop | Float | false | minvalue = 0 maxvalue = 10 | Horizontal Dilution Of Precision. |
| vdop | Float | false | minvalue = 0 maxvalue = 10 | Vertical Dilution Of Precision. |
| actual | Boolean | false | - | The information about actuality of GPS data: 'true', if actual 'false', if inferred |
| satellites | Integer | false | minvalue = 0 maxvalue = 31 | Number of satellites in view. |
| dimension | Common.Dimension | false | - | The supported GPS dimension: 2D, 3D. See section 11.8.3.4 Dimension |
| altitude | Float | false | minvalue = -10000 maxvalue = 10000 | Altitude in meters. |
| heading | Float | false | minvalue = 0 maxvalue = 359.99 | '0' is considered as heading North. SDL expects to receive data with resolution of 0.01. |
| speed | Float | false | minvalue = 0 maxvalue = 500 | The speed in KPH. |

11.8.3.3 CompassDirection

| Element name | Value | Short Description |
|--------------|-------|---|
| NORTH | 0 | Represents the North compass direction |
| NORTHWEST | 1 | Represents the North-West compass direction |
| WEST | 2 | Represents the West compass direction |
| SOUTHWEST | 3 | Represents the South-West compass direction |
| SOUTH | 4 | Represents the South compass direction |
| SOUTHEAST | 5 | Represents the South-East compass direction |

| Element name | Value | Short Description |
|---------------------|--------------|---|
| EAST | 6 | Represents the East compass direction |
| NORTHEAST | 7 | Represents the North-East compass direction |

11.8.3.4 Dimension

| Element name | Value | Short Description |
|---------------------|--------------|-------------------------------------|
| NO_FIX | 0 | No GPS at all |
| 2D | 1 | Longitude and latitude |
| 3D | 2 | Longitude and latitude and altitude |

11.8.3.5 ComponentVolumeStatus

| Element name | Value | Short Description |
|---------------------|--------------|--|
| UNKNOWN | 0 | The data is unknown. |
| NORMAL | 1 | The volume is normal. |
| LOW | 2 | The volume is low. |
| FAULT | 3 | The module/sensor is currently faulted. |
| ALERT | 4 | The component's volume is in critical level. |
| NOT_SUPPORTED | 5 | The data is not supported. |

11.8.3.6 PRNDL

| Element name | Value | Short Description |
|---------------------|--------------|---|
| PARK | 0 | Parking |
| REVERSE | 1 | Reverse gear |
| NEUTRAL | 2 | No gear |
| DRIVE | 3 | Drive Sport mode |
| SPORT | 4 | Sport mode |
| LOWGEAR | 5 | 1st gear hold |
| FIRST | 6 | 1st gear |
| SECOND | 7 | 2nd gear |
| THIRD | 8 | 3d gear |
| FOURTH | 9 | 4th gear |
| FIFTH | 10 | 5th gear |
| SIXTH | 11 | 6th gear |
| SEVENTH | 12 | 7th gear |
| EIGHTH | 13 | 8th gear |
| FAULT | 14 | The module/sensor is currently faulted. |

11.8.3.7 TireStatus

| Param Name | Type | Mandatory | Description |
|-------------------|---------------------------|------------------|---|
| pressureTelltale | Common.WarningLightStatus | false | Status of the Tire Pressure Telltale. See WarningLightStatus. |
| leftFront | Common.SingleTireStatus | false | The status of the left front tire. |
| rightFront | Common.SingleTireStatus | false | The status of the right front tire. |
| leftRear | Common.SingleTireStatus | false | The status of the left rear tire. |

| Param Name | Type | Mandatory | Description |
|-------------------|-------------------------|------------------|-------------------------------------|
| rightRear | Common.SingleTireStatus | false | The status of the right rear tire. |
| innerLeftRear | Common.SingleTireStatus | false | The status of the inner left rear. |
| innerRightRear | Common.SingleTireStatus | false | The status of the inner right rear. |

11.8.3.8 WarningLightStatus

| Element name | Value | Short Description |
|---------------------|--------------|--|
| OFF | 0 | The warning light is off |
| ON | 1 | The warning light is on |
| FLASH | 2 | The warning light is flashing |
| NOT_USED | 3 | There is no warning light for the event on HU. |

11.8.3.9 SingleTireStatus

| Param Name | Type | Mandatory | Description |
|-------------------|------------------------------|------------------|--|
| status | Common.ComponentVolumeStatus | true | See section 11.8.3.5 ComponentVolumeStatus |

11.8.3.10 BeltStatus

| Param Name | Type | Mandatory | Description |
|----------------------------|-------------------------------|------------------|---|
| driverBeltDeployed | Common.VehicleDataEventStatus | false | The driver seat belt is deployed. |
| passengerBeltDeployed | Common.VehicleDataEventStatus | false | The passenger seat belt is deployed. |
| passengerBuckleBelted | Common.VehicleDataEventStatus | false | The passenger seat belt is buckled. |
| driverBuckleBelted | Common.VehicleDataEventStatus | false | The driver seat belt is buckled. |
| leftRow2BuckleBelte d | Common.VehicleDataEventStatus | false | The left seat belt of the 2 nd row is buckled. |
| passengerChildDetected | Common.VehicleDataEventStatus | false | The child passenger is detected. |
| rightRow2BuckleBelt ed | Common.VehicleDataEventStatus | false | The right seat belt of the 2 nd row is buckled. |
| middleRow2BuckleBel ted | Common.VehicleDataEventStatus | false | The middle seat belt of the 2 nd row is buckled. |
| middleRow3BuckleBel ted | Common.VehicleDataEventStatus | false | The middle seat belt of the 3 rd row is buckled. |
| leftRow3BuckleBelte d | Common.VehicleDataEventStatus | false | The left seat belt of the 3 rd row is buckled. |
| rightRow3BuckleBelte d | Common.VehicleDataEventStatus | false | The right seat belt of the 3 rd row is buckled. |
| leftRearInflatableB elted | Common.VehicleDataEventStatus | false | The left rear inflatable is belted. |
| rightRearInflatableB elted | Common.VehicleDataEventStatus | false | The right rear inflatable is belted. |
| middleRow1BeltDeplo yed | Common.VehicleDataEventStatus | false | The seat belt of the middle row is deployed. |

| Param Name | Type | Mandatory | Description |
|------------------------|-------------------------------|-----------|---|
| middleRow1BuckleBelted | Common.VehicleDataEventStatus | false | The seat belt of the middle row is buckled. |

11.8.4 Sequence Diagrams

11.8.4.1 GetVehicleData

11.8.5 JSON Messages Examples

11.8.5.1 Request

```
{
  "id" : 139,
  "jsonrpc" : "2.0",
  "method" : "VehicleInfo.GetVehicleData",
  "params" :
  {
    "gps" : true,
    "speed" : true,
    "fuelLevel_State" : true,
    "externalTemperature" : true,
    "prndl" : true,
    "tirePressure" : true,
    "odometer" : true,
    "beltStatus" : true,
    "bodyInformation" : true,
    "deviceStatus" : true,
    "wiperStatus" : true,
    "headLampStatus" : true,
    "accPedalPosition" : true,
    "myKey" : true
  }
}
```

11.8.5.2 Response

```
{
  "id" : 139,
  "jsonrpc" : "2.0",
  "result" :
  {
    "gps" :
    [
      {
        "longitudeDegrees" : 46.4774700,
        "latitudeDegrees" : 30.7326200,
        "utcYear" : 2013,
        "utcMonth" : 12,
        "utcDay" : 31,
        "utcHours" : 23,
        "utcMinutes" : 50,
        "utcSeconds" : 5,
        "compassDirection" : NORTH,
        "pdop" : 0.15,
        "hdop" : 1.01,
        "vdop" : 1.56,
        "actual" : true,
        "satellites" : 8,
      }
    ]
  }
}
```

```

        "dimension" : 3D,
        "altitude" : 47,
        "heading" : 0,
        "speed" : 90
    ],

    "speed" : 90,
    "fuelLevel_State" : LOW,
    "externalTemperature" : -5,
    "prndl" : FOURTH,
    "tirePressure" :
    [
        "pressureTelltale" : ON,
        "leftFront" : NORMAL,
        "rightFront" : NORMAL,
        "leftRear" : LOW,
        "rightRear" : UNKNOWN
    ],
    "odometer" : 1065,
    "beltStatus" :
    [
        "driverBeltDeployed" : YES,
        "passengerBeltDeployed" : YES,
    ],
    "bodyInformation" :
    [
        "parkBrakeActive" : false,
        "ignitionStableStatus" : IGNITION_SWITCH_STABLE,
        "ignitionStatus" : RUN
    ],
    "deviceStatus" :
    [
        "voiceRecOn" : false,
        "btIconOn" : false,
        "callActive" : false,
        "phoneRoaming" : false,
        "textMsgAvailable" : true,
        "battleLevelStatus" : THREE_LEVEL_BARS,
        "stereoAudioOutputMuted" : true,
        "monoAudioOutputMuted" : false,
        "signalLevelStatus" : NOT_PROVIDED,
        "primary AudioSource" : MOBILE_APP,
        "eCallEventActive" : false
    ],
    "wiperStatus" : OFF,
    "headLampStatus" :
    [
        "lowBeamsOn" : true,
        "highBeamsOn" : false
    ],
    "accPedalPosition" : 80,
    "myKey" :
    [
        "e911Override" : OFF
    ],
    "code" : 0,
    "method" : "VehicleInfo.GetVehicleData"
}
}

```

11.8.5.3 Error message

```
{  
    "id" : 139,  
    "jsonrpc" : "2.0",  
    "error" :  
    {  
        "code" : 9,  
        "message" : "The requested data is not available",  
        "data" :  
        {  
            "method" : "VehicleInfo.GetVehicleData"  
        }  
    }  
}
```

11.8.6 D-Bus Messages Examples

11.8.6.1 Request

11.8.6.2 Response

11.8.6.3 Failure

11.9 OnVehicleData

11.9.1 Description

| | |
|----------|---------------------------------------|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about changes of vehicle data. |

Initially SDL sends SubscribeVehicleData for getting the periodic updates from HMI whenever each of subscribed data types changes. OnVehicleData is expected to bring such updated values to SDL

11.9.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|----------------|-----------|--------------------------------|---|
| gps | Common.GPSData | false | - | Information related to GPS data: number of satellites, compass direction, longitude, latitude, etc. For more details please see GPSData |
| speed | Float | false | minvalue = 0 maxvalue = 700 | The vehicle speed in kilometers per hour. |

| Param Name | Type | Mandatory | Additional | Description |
|------------------------|------------------------------|-----------|-------------------------------------|---|
| rpm | Integer | false | minvalue = 0 maxvalue = 20000 | The number of revolutions per minute of the engine. |
| fuelLevel | Float | false | minvalue = -6 maxvalue = 106 | The fuel level in the tank (percentage). |
| fuelLevelState | Common.ComponentVolumeStatus | false | - | The fuel level state should be returned: normal, low, etc. For more details please see ComponentVolumeStatus |
| instantFuelConsumption | Float | false | minvalue = 0 maxvalue = 25575 | The instantaneous fuel consumption in microliters. |
| externalTemperature | Float | false | minvalue = -40 maxvalue = 100 | The external temperature in degrees Celsius. |
| vin | String | false | maxlength = 17 | Vehicle identification number. |
| prndl | Common.PRNDL | false | - | The position of change-speed lever: first, second, etc. For more details please see PRNDL. |
| tirePressure | Common.TireStatus | false | - | The following information should be returned: - The warning status (on, off, etc.) - The status of the tire pressure itself: normal, low, etc. For more details please see TireStatus. |
| odometer | Integer | false | minvalue = 0 maxvalue = 17000000 | Odometer in km |
| beltStatus | Common.BeltStatus | false | - | The status of the seat belts: deployed, buckled. For more details please see BeltStatus |
| bodyInformation | Common.BodyInformation | false | - | The body information should be provided: park brake status, ignition and ignition stable status. For more details please see BodyInformation |
| deviceStatus | Common.DeviceStatus | false | - | The information on device status should be provided including signal and battery strength. For more details please see DeviceStatus. |

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|-------------------------------|-----------|-------------------------------------|--|
| driverBraking | Common.VehicleDataEventStatus | false | - | The status of the brake pedal should be provided: on, off, etc. For more details please see VehicleDataEventStatus |
| wiperStatus | Common.WiperStatus | false | - | The status of the wipers should be provided: if they are manually on, manually off, stalled, etc. For more details please see WiperStatus |
| headLampStatus | Common.HeadLampStatus | false | - | Status of the head lamps should be provided: if they are on, off, etc. For more details please see HeadLampStatus |
| engineTorque | Float | false | minvalue = -1000 maxvalue = 2000 | Torque value for engine (in Nm) for non-diesel models. |
| accPedalPosition | Float | false | minvalue = 0 maxvalue = 100 | Accelerator pedal position (percentage depressed). |
| steeringWheelAngle | Float | false | minvalue = -2000 maxvalue = 2000 | Current angle of the steering wheel (in degrees). |
| myKey | Common.MyKey | false | - | Information related to the MyKey feature. For more details please see MyKey |

11.9.2 Sequence Diagrams

11.9.2.1 OnVehicleData

11.9.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "VehicleInfo.OnVehicleData",
  "params" :
  {
    "speed" : 60,
    "externalTemperature" : -7,
    "prndl" : THIRD,
    "odometer" : 1066,
    "wiperStatus" : MAN_INT_ON,
    "accPedalPosition" : 70
  }
}
```

11.9.4 D-Bus Messages Examples

| |
|--|
| |
|--|

12 Navigation Component Description

12.1 IsReady

12.1.1 Description

| | |
|----------|-----------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Know if component is ready. |

The request comes after HMI's readiness is confirmed via [OnReady](#) notification. SDL requires the information about whether the navigation information could be processed or provided by HMI.

Note:

If Navigation component is responded to be unavailable, SDL will not further send the requests related to it.

12.1.2 Request

12.1.2.1 Behavior

HMI must:

- Check whether Navigation component is present and ready
- Respond correspondingly to results of this check.

12.1.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

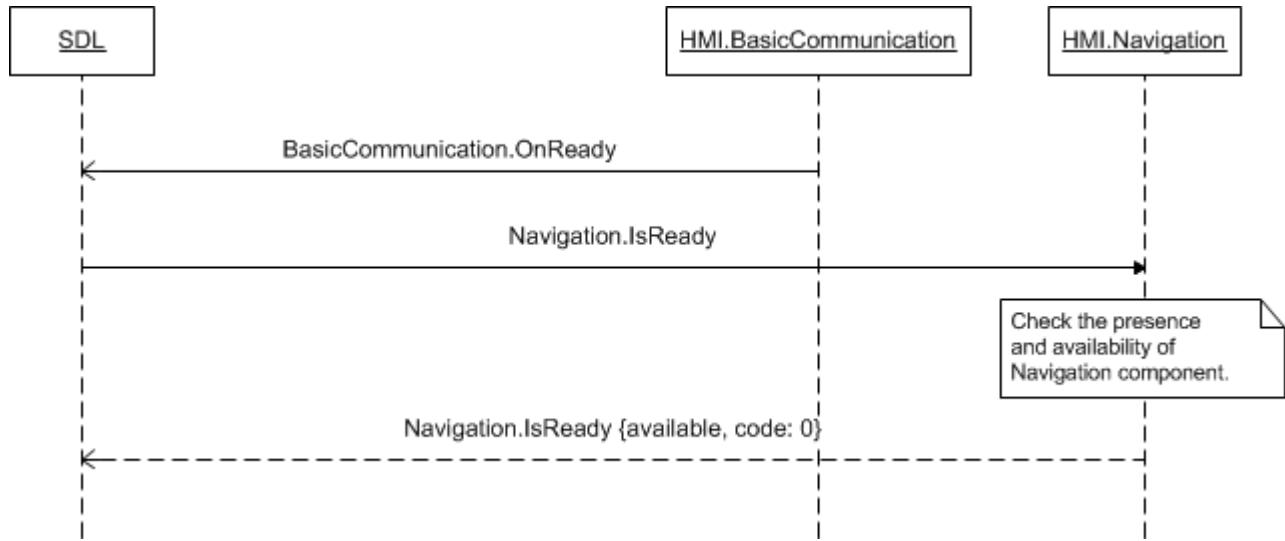
| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|-----------------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI provides the information about Navigation component availability. | JSON response | Method return | available, code: 0 | |
| Failure | DATA_NOT_AVAILABLE: The information about Navigation availability cannot be provided. | JSON error message | Method return | Code: 9 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

12.1.3.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|---|
| availabe | Boolean | true | Must be <ul style="list-style-type: none">- 'true' if Navigation component is present and ready- 'false' if not. |

12.1.4 Sequence Diagrams

12.1.4.1 Navigation.IsReady and preceding OnReady



12.1.5 JSON Messages Examples

12.1.5.1 Request

```
{
    "id" : 27,
    "jsonrpc" : "2.0",
    "method" : "Navigation.IsReady"
}
```

12.1.5.2 Response

```
{
    "id" : 27,
    "jsonrpc" : "2.0",
    "result" :
    {
        "available" : true,
        "code" : 0,
        "method" : "Navigation.IsReady"
    }
}
```

12.1.5.3 Error message

```
{
    "id" : 27,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 22,
        "message" : "The unknown error has occurred",
        "data" :
        {
            "method" : "Navigation.IsReady"
        }
    }
}
```

12.1.6 D-Bus Messages Examples

12.1.6.1 Request

12.1.6.2 Response

| |
|--|
| |
|--|

12.1.6.3 Failure

| |
|--|
| |
|--|

12.2 AlertManeuver

12.2.1 Description

| | |
|----------|----------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Announce the navigation maneuver |

SDL sends AlertManeuver together with TTS.Speak RPC. The purpose is to notify the user about the next navigation maneuver.

12.2.2 Request

12.2.2.1 Behavior

HMI must:

1. Display the dialog with HMI-defined alert icon and one of the following:
 - Up to three soft buttons once defined within request with `softButtons` parameter
 - HMI-defined 'Close' soft button once the empty request has arrived (that is, without `softButtons` parameter).
- 2.

12.1.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-------------|-------------------|-----------|--|---|
| softButtons | Common.SoftButton | false | Array = true minsize = 0 maxsize = 3 | Up to three soft buttons defined. If omitted, only the system defined "Close" soft button should be displayed (if applicable). |

12.2.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI displayed the requested dialog of navigation maneuver. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | ABORTED: The dialog is aborted with the hogher priority RPC. | | | Code: 5 | Please see Result Enumeration for all SDL- |

| | | | | | |
|--|---|--|--|----------|------------------|
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | supported codes. |
|--|---|--|--|----------|------------------|

12.2.4 Sequence Diagrams

12.2.4.1 GetVehicleType

12.2.5 JSON Messages Examples

12.2.5.1 Request

```
{
  "id" : 143,
  "jsonrpc" : "2.0",
  "method" : "Navigation.AlertManeuver",
  "params" :
  {
    "softButtons" :
    [
      {
        "type" : TEXT,
        "text" : "Leave onscreen",
        "softButtonID" : 45,
        "systemAction" : KEEP_CONTEXT
      },
      {
        "type" : TEXT,
        "text" : "Close",
        "softButtonID" : 46,
        "systemAction" : STEAL_FOCUS
      }
    ],
    "appID" : 96
  }
}
```

12.2.5.2 Response

```
{
  "id" : 143,
  "jsonrpc" : "2.0",
  "result" :
  {
    "code" : 0,
    "method" : "Navigation.AlertManeuver"
  }
}
```

12.2.5.3 Error message

```
{
  "id" : 143,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : 15,
    "message" : "A command cannot be executed because there is NO specified"
  }
}
```

```

        with appID application registered",
    "data" :
    {
        "method" : "Navigation.AlertManeuver"
    }
}

```

12.2.6 D-Bus Messages Examples

12.2.6.1 Request

[REDACTED]

12.2.6.2 Response

[REDACTED]

12.2.6.3 Failure

[REDACTED]

12.3 ShowConstantTBT

12.3.1 Description

| | |
|----------|-----------------------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Show information about navigation |

SDL requests to display

12.3.2 Request

12.3.2.1 Behavior

HMI must:

12.3.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|------------------------|-----------|--|--|
| navigationTexts | Common.TextFieldStruct | true | Array = true minsize = 0 maxsize = 5 | The text to be displayed in up to four fields on Navigation display: - navigationText1 - navigationText2 - ETA - totalDistance For more details please see TextFieldStruct, section 5.2.8 |
| turnIcon | Common.Image | true | - | The icon to be displayed. For more details please see Image, section 5.2.11 . |
| nextTurnIcon | Common.Image | false | - | |
| distanceToManeuver | Float | true | minvalue = 0 maxvalue = 1000000000 | The distance from the previous maneuver till the next one. HMI may use these data for calculating the progress bar. |
| distanceTo | Float | true | minvalue = 0 | Fraction of distance from the |

| Param Name | Type | Mandatory | Additional | Description |
|------------------|-------------------|-----------|--|--|
| ManeuverScale | | | maxvalue = 1000000000 | AlertManeuver triggering till the next maneuver. HMI may use these data for calculating the progress bar. |
| maneuverComplete | Boolean | false | - | If 'true', the maneuver has been completed and the AlertManeuver overlay must be cleared. If omitted the value must be assumed as 'false'. |
| softButtons | Common.SoftButton | false | Array = true minsize = 0 maxsize = 3 | SDL may request to draw up to three soft buttons for Navigation. If the empty array is sent or the parameter is omitted, the currently displayed soft button values must remain unchanged. For more details please see SoftButton , section 5.2.10 |
| appID | Integer | true | - | ID of the application that concerns this RPC. |

12.3.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI executed the request successfully. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax, parameters out of bounds or of wrong type) | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | ABORTED: This RPC is aborted by the higher priority RPC.. | | | Code: 5 | |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

12.3.4 Sequence Diagrams

12.3.4.1 ShowConstantTBT

12.3.5 JSON Messages Examples

12.3.5.1 Request

```
{
```

```

    "id" : 543,
    "jsonrpc" : "2.0",
    "method" : "Navigation.ShowConstantTBT",
    "params" :
    {
        "navigationTexts" :
        [
            {
                "fieldName" : navigationText1,
                "fieldText" : "Destination point: Berlin"
            },
            {
                "fieldName" : ETA,
                "fieldText" : "15:45"
            },
            {
                "fieldName" : totalDistance,
                "fieldText" : "658"
            }
        ],
        "turnIcon" :
        [
            {
                "value" : "tmp/SDL/app/Navi/icon_3245.jpeg",
                "imageType" : DYNAMIC
            },
            "distanceToManeuver" : 168,
            "distanceToManeuverScale" : 265,
            "softButtons" :
            [
                {
                    "type" : TEXT,
                    "text" : "Close",
                    "softButtonID" : 76,
                    "systemAction" : DEFAULT_ACTION
                },
                "appID" : 26743
            }
        ]
    }
}

```

12.3.5.2 Response

```
{
    "id" : 543,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "Navigation.ShowConstantTBT"
    }
}
```

12.3.5.3 Error message

```
{
    "id" : 543,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 5,
        "message" : "A command was aborted",
        "data" :
        {
            "method" : "Navigation.ShowConstantTBT"
        }
    }
}
```

12.3.6 D-Bus Messages Examples

12.3.6.1 Request

12.3.6.2 Response

12.3.6.3 Failure

12.4 UpdateTurnList

12.4.1 Description

| | |
|----------|------------------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | Update turn list |

12.4.2 Request

12.4.2.1 Behavior

HMI must:

12.4.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-------------|-------------------|-----------|--|--|
| turnList | Common.Turn | false | Array = true minsize = 1 maxsize = 100 | The array of objects that consist of: - text string to be displayed in navigation field - image for turn prompt. For more details please see Turn, section |
| softButtons | Common.SoftButton | false | Array = true minsize = 0 maxsize = 1 | One soft button defined. For more details regarding the type/highlighted or not/etc. please see SoftButton, section If omitted on supported displays, app-defined SoftButton will be left blank. |

12.4.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|--------------------|---------------|----------------|---------------------------------------|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: HMI executed the RPC successfully. | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: The data sent is invalid (invalid JSON syntax, parameters out of bounds or | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |

| | | | | |
|--|---|--|----------|--|
| | of wrong type) | | | Pleas see Result Enumeration for all SDL- supported codes. |
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | code: 22 | |

12.4.4 Sequence Diagrams

12.4.4.1 UpdateTurnList

12.4.5 JSON Messages Examples

12.4.5.1 Request

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "method" : "Navigation.UpdateTurnList",
  "params" :
  {
    "turnList" :
    [
      {
        "navigationText" :
        [
          {
            "fieldName" : navigationText,
            "fieldText" : "Turn Right"
          },
          "turnIcon" :
          [
            {
              "value" : "tmp/SDL/app/Navi/icon_turn_right.jpeg",
              "imageType" : DYNAMIC
            }
          ],
          {
            "navigationText" :
            [
              {
                "fieldName" : navigationText,
                "fieldText" : "Turn Left"
              },
              "turnIcon" :
              [
                {
                  "value" : "tmp/SDL/app/Navi/icon_turn_left.jpeg",
                  "imageType" : DYNAMIC
                }
              ],
              {
                "navigationText" :
                [
                  {
                    "fieldName" : navigationText,
                    "fieldText" : "Go Forward"
                  },
                  "turnIcon" :
                  [
                    {
                      "value" : "tmp/SDL/app/Navi/icon_go_forward.jpeg",
                      "imageType" : DYNAMIC
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  }
}
```

```

        ],
    },
    "softButtons" :
    [
        {
            "type" : BOTH,
            "text" : "Return",
            "image" :
            [
                {
                    "value" : "tmp/SDL/app/Navi/icon_583.jpg",
                    "imageType" : DYNAMIC
                },
                "isHighlighted" : true,
                "softButtonID" : 118,
                "systemAction" : DEFAULT_ACTION
            ]
        }
    ]
}

```

12.4.5.2 Response

```
{
    "id" : 176,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "Navigation.UpdateTurnList"
    }
}
```

12.4.5.3 Error message

```
{
    "id" : 176,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 4,
        "message" : "A command was rejected because a higher priority command is requested",
        "data" :
        {
            "method" : "Navigation.UpdateTurnList"
        }
    }
}
```

12.4.6 D-Bus Messages Examples

12.4.6.1 Request

12.4.6.2 Response

12.4.6.3 Failure

12.5 StartStream

12.5.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about start of video streaming. |

12.5.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|-----------------------------------|--|
| url | String | true | Minlength = 21 maxlength = 500 | URL that HMI start playing. |
| appID | Integer | true | - | ID of the application requested this RPC |

12.5.2 Sequence Diagrams

12.5.2.1 StartStream

12.5.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "Navigation.StartStream",  
    "params" :  
    {  
        "url" : SDL/application_directory/video/123.mp4,  
        "appID" : 65674  
    }  
}
```

12.5.4 D-Bus Messages Examples

| |
|--|
| |
|--|

12.6 StopStream

12.6.1 Description

| | |
|----------|---------------------------------------|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about stop of video streaming. |

12.6.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|------------|--|
| appID | Integer | true | - | ID of the application requested this RPC |

12.6.2 Sequence Diagrams

12.6.2.1 StopStream

12.6.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "Navigation.StopStream",
    "params" :
    {
        "appID" : 65674
    }
}
```

12.6.4 D-Bus Messages Examples

12.5 StartAudioStream

12.5.1 Description

| | |
|----------|--|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about start of video streaming. |

12.5.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|-----------------------------------|--|
| url | String | true | Minlength = 21 maxlength = 500 | URL that HMI start playing. |
| appID | Integer | true | - | ID of the application requested this RPC |

12.5.2 Sequence Diagrams

12.5.2.1 StartStream

12.5.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "Navigation.StartStream",  
    "params" :  
    {  
        "url" : SDL/application_directory/video/123.mp4,  
        "appID" : 65674  
    }  
}
```

12.5.4 D-Bus Messages Examples

```
[REDACTED]
```

12.6 StopAudioStream

12.6.1 Description

| | |
|----------|---------------------------------------|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about stop of video streaming. |

12.6.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|------------|--|
| appID | Integer | true | - | ID of the application requested this RPC |

12.6.2 Sequence Diagrams

12.6.2.1 StopStream

```
[REDACTED]
```

12.6.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "Navigation.StopStream",  
    "params" :  
    {  
        "appID" : 65674  
    }  
}
```

12.6.4 D-Bus Messages Examples

```
[REDACTED]
```

12.7 OnTBTClientState

12.7.1 Description

| | |
|-----------------|--|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Provide the information about TBT Client state |

12.7.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|-----------------|-----------|--|
| state | Common.TBTState | true | Current State of TBT client. For more details please see TBTState |

12.7.2 Sequence Diagrams

12.7.2.1 OnTBTClientState

12.7.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "Navigation.OnTBTClientState",  
    "params" :  
    {  
        "state" : NEXT_TURN_REQUEST  
    }  
}
```

12.7.4 D-Bus Messages Examples

| |
|--|
| |
|--|

12.8 SendLocation

12.8.1 Description

| | |
|-----------------|----------|
| Type: | Function |
| Sender: | SDL |
| Purpose: | |

That allows the app to send a destination to the embedded nav system.

12.8.2 Request

12.8.2.1 Behavior

HMI must:

- 1.

Note:

- The applicable to this RPC result codes are provided in section 7.18.3 Response.
- The sequence diagrams describing the expected HMI behavior are provided in the section 7.18.4 Sequence Diagrams
- The picture of scrollable message screen is added to the section 7.18.5 Possible Layout.

12.8.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------------|--------------|-----------|--|-------------|
| longitudeDegrees | Float | true | Minvalue = -180 Maxvalue = 180 | |
| latitudeDegrees | Float | true | Minvalue = -90 Maxvalue = 90 | |
| locationName | String | false | Maxlength = 500 | |
| addressLines | String | false | Maxlength = 500 minsize= 0 maxsize= 4 array= true | |
| phoneNumber | String | false | Maxlength = 500 | |
| locationImage | Common.Image | false | | |

12.8.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|--|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: | JSON response | Method return | code: 0 | |
| Failure | | JSON error message | Method return | code: 5 | Applicable for this RPC result codes. |
| | INVALID_DATA: The data sent is invalid (invalid JSON syntax or parameters out of bounds or of wrong type) | | | code: 11 | Please see Result Enumeration for all SDL-supported codes. |

| | | | | | |
|--|---|--|--|----------|--|
| | GENERIC_ERROR: The unknown issue occurred or other codes are not applicable. | | | code: 22 | |
|--|---|--|--|----------|--|

12.8.4 Sequence Diagrams

12.8.4.1 SendLocation

12.8.4.2 SendLocation

12.8.4.3 SendLocation

12.8.5 JSON Messages Examples

12.8.5.1 Request

```
{
    "id" : 138,
    "jsonrpc" : "2.0",
    "method" : "Navigation.SendLocation",
    "params" :
    {
        "longitudeDegrees" : 139.34,
        "latitudeDegrees" : 35.36,
        "locationName" : "Ford Repair",
        "locationImage" :
        {
            "value" : "tmp/SDL/app/Navi/12345.jpg",
            "imageType" : DYNAMIC
        }
    }
}
```

12.8.5.2 Response

```
{
    "id" : 138,
    "jsonrpc" : "2.0",
    "result" :
    {
        "code" : 0,
        "method" : "Navigation.SendLocation"
    }
}
```

12.8.5.3 Error message

```
{
    "id" : 138,
    "jsonrpc" : "2.0",
    "error" :
```

```
{  
    "code" : 12,  
    "message" : "The string data is too long",  
    "data" :  
    {  
        "method" : "Navigation.SendLocation"  
    }  
}
```

12.8.6 D-Bus Messages Examples

12.8.6.1 Request

12.8.6.2 Response

12.8.6.3 Failure

13 SDL Component Description

Notes about ‘SDL’ component:

A. ‘SDL’ names a component that is a part of SDL. HMI has the ability to request and receive definite information from SDL using ‘SDL’ interface, in contrast to communicating using other interfaces of HMI_API.

B. HMI-SDL communication with ‘SDL’ component over WebSocket (WS):

B.1. Both parties should use the WS registered with the name of ‘BasicCommunication’ (see [section 2.1 Connection Opening of chapter 2 WebSocketTransport](#)). Separate WS connection should not be opened for communication with ‘SDL’ component.

B.2. Both parties should send JSON messages with the field of:

"method" : "SDL.<RPC_name>" (for example, "SDL.ActivateApp")
when communicating using ‘SDL’ ‘interface.

13.1 SDL.ActivateApp (Ford-specific)

13.1.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | HMI |
| Purpose: | Inform about User has activated the application and get the permissions for it. |

SDL needs SDL.ActivateApp request to:

1. Know that the named application has been chosen by the User.
2. Provide HMI with the policies-related data for the named application that SDL sends within response to SDL.ActivateApp.

13.1.2 Request

13.1.2.1 Behavior

HMI must:

1. Send `SDL.ActivateApp` to SDL when the User requests to activate the application either form UI or by VR. Include the `appID` of the application being activated to the request is a must.

Note:

The information about the application (name, ID, etc.) is provided by SDL via either `BasicCommunication.UpdateAppList` or `BasicCommunication.OnAppRegistered` RPCs.

2. Wait for the response from SDL before activating the named application.
3. Read the information from SDL’s response and behave in one of the following ways depending on parameters provided.

3.1. DO NOT activate the application in one of the following cases:

- 3.1.1. `isSDLAllowed: false` -
 - a) Do not activate the application
 - b) Request the appropriate user-friendly message from SDL (via `SDL.GetUserFriendlyMessage {messageCodes: "DataConsent"}`)
 - c) Display the message providing the User with the possibility to accept or decline the prompt (for example, “Yes” / “No” buttons)
 - d) After User’s making choice, send `SDL.OnAllowSDLFunctionality` to SDL notifying whether the User accepted the prompt or not (“allowed:true” or “allowed:false” correspondingly)
 - e) Wait for `BasicCommunication.ActivateApp` from SDL in both cases of User’s accepting or declining the prompt:
 - e-1) `BasicCommunication.ActivateApp{appID}` will come for the purpose to activate the app to FULL after the User has accepted the prompt.
 - e-2) otherwise, `BasicCommunication.ActivateApp{appID, level: NONE}` will come for the purpose to clearly restrict HMI from activating the app in case the User has declined the prompt.

Note:

Related diagrams: see sections #12.1.4.1 and #12.1.4.2

- 3.1.2. `isAppRevoked: true` –
- Do not activate the application
 - Request the appropriate user-friendly message from SDL (via `SDL.GetUserFriendlyMessage {messageCodes: "AppUnauthorized"}`)
 - Unregister the corresponding application upon `BasicCommunication.OnAppUnregistered{appId}` (*Information: while the appId has NULL permissions in PT, SDL will reject such app's registration afterwards*)

Note:

Related diagram: see section #12.1.4.5.

3.2. Activate the application in one of the following cases:

- 3.2.1. `isSDLAllowed: true` and `isPermissionsConsentNeeded: true`
- Activate the application
 - Request the appropriate user-friendly message from SDL (via `SDL.GetUserFriendlyMessage {messageCodes: "AppPermissions"}`)
 - Request the list of permissions that require User's consent from SDL (via `SDL.GetListOfPermissions`)
 - Display the message providing the User with the possibility to allow-disallow each of permission items (for example, using checkboxes)
 - After the User has finished choosing, notify SDL via `SDL.OnAppPermissionConsent`.

Note:

Related diagram: see section #12.1.4.3

- 3.2.2. `isSDLAllowed: true` and `isAppPermissionsRevoked: true`
- Activate the application
 - Request the appropriate user-friendly message from SDL (via `SDL.GetUserFriendlyMessage {messageCodes: "AppPermissionsRevoked"}`)
 - Display the message to the User.

Note:

Related diagram: see section #12.1.4.4

13.1.2.2 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|---------|-----------|------------|---|
| <code>appId</code> | Integer | true | – | ID of the application that the User has chosen. |

13.1.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|---|---------------------------|---------------|--|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: SDL has processed the request and provides the app's policies status information. | JSON response | Method return | <code>code: 0,</code> <code>params</code> | 'params' are described in section 12.1.3.1 Parameters. |
| Failure | GENERIC_ERROR: An error occurred. | JSON <u>error message</u> | Method return | <code>code: 22</code> | |

13.1.3.1 Parameters

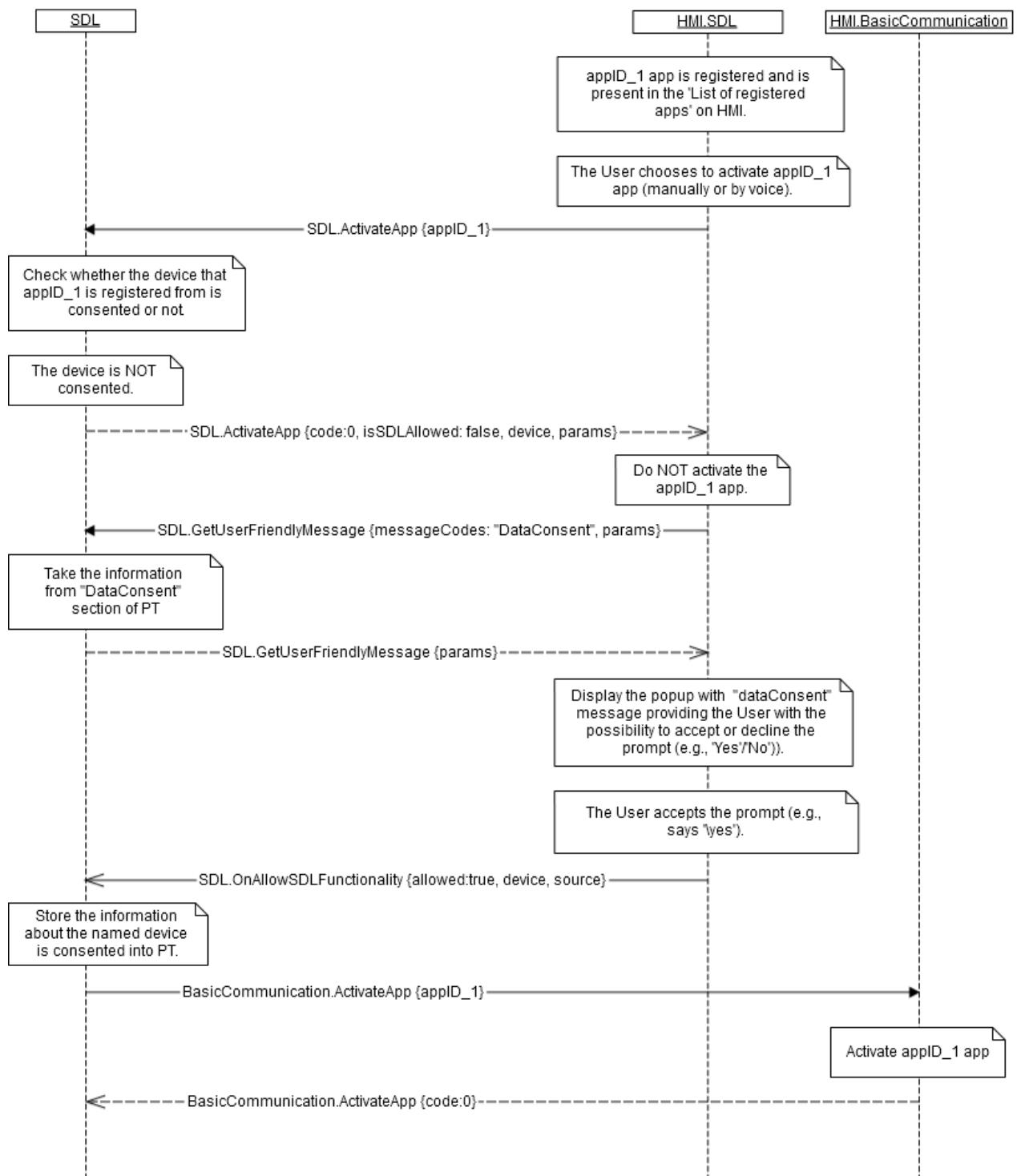
| Param Name | Type | Mandatory | Additional | Description |
|----------------------------|-----------------------|-----------|--|---|
| isSDLAllowed | Boolean | true | - | SDL returns: - ‘true’, in case the User has allowed using the device for PolicyTable Exchange. - ‘false’, in case the User has not yet been asked for or in case the User has disallowed using the device for PolicyTable Exchange. |
| device | Common.DeviceInfo | false | - | If isSDLAllowed is false, consent for sending PT through specified device is required. |
| isPermissionsConsentNeeded | Boolean | true | - | SDL returns: - ‘true’, in case the policies assigned to the named app require the User’s consent before taking into effect.. - ‘false’, in case there are no policy groups that require User’s consent. |
| isAppPermissionsRevoked | Boolean | true | - | SDL returns: - ‘true’, in case the latest PT Update removed one or more permission groups from the named app’s policies. - ‘false’, in case there are no permission groups removed from the named app’s policies. |
| appRevokedPermissions | Common.PermissionItem | false | array = true, minsize = 1, maxsize = 100 | If app permissions were reduced (isAppPermissionsRevoked == true), then this array specifies list of removed permissions. |
| isAppRevoked | Boolean | true | - | SDL returns: - ‘true’, in case the latest PT Update specified the named app to have NULL permissions (meaning the app became unauthorized). - ‘false’, in case the named app has different from NULL permissions. |
| priority | Common.AppPriority | false | - | Send to HMI so that it can coordinate order of requests/notifications correspondingly. |

13.1.3.3 AppPriority

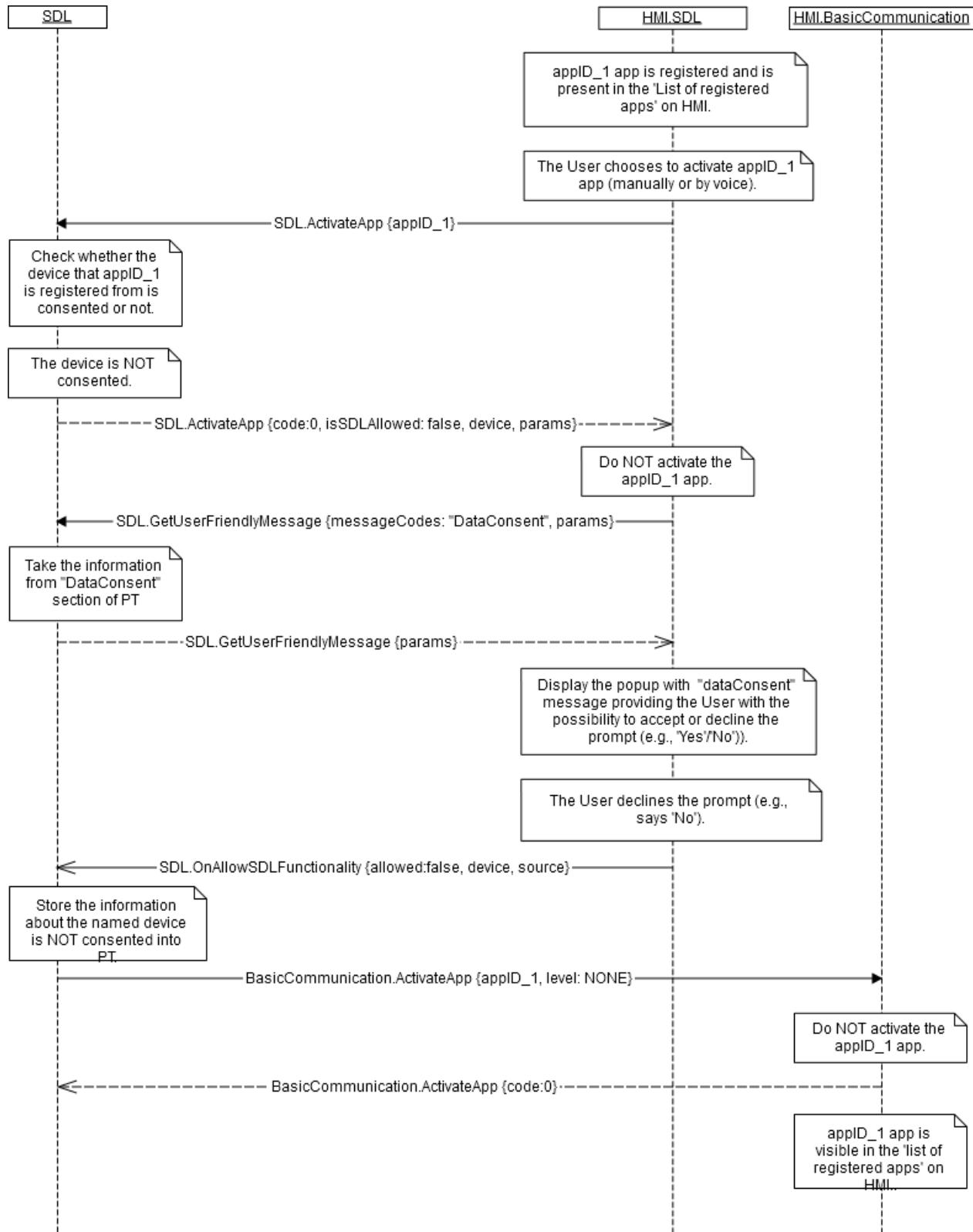
| Element name | Value | Short Description |
|---------------------|-------|-------------------|
| EMERGENCY | 0 | |
| NAVIGATION | 1 | |
| VOICE_COMMUNICATION | 2 | |
| COMMUNICATION | 3 | |
| NORMAL | 4 | |
| NONE | 5 | |

13.1.4 Sequence Diagrams

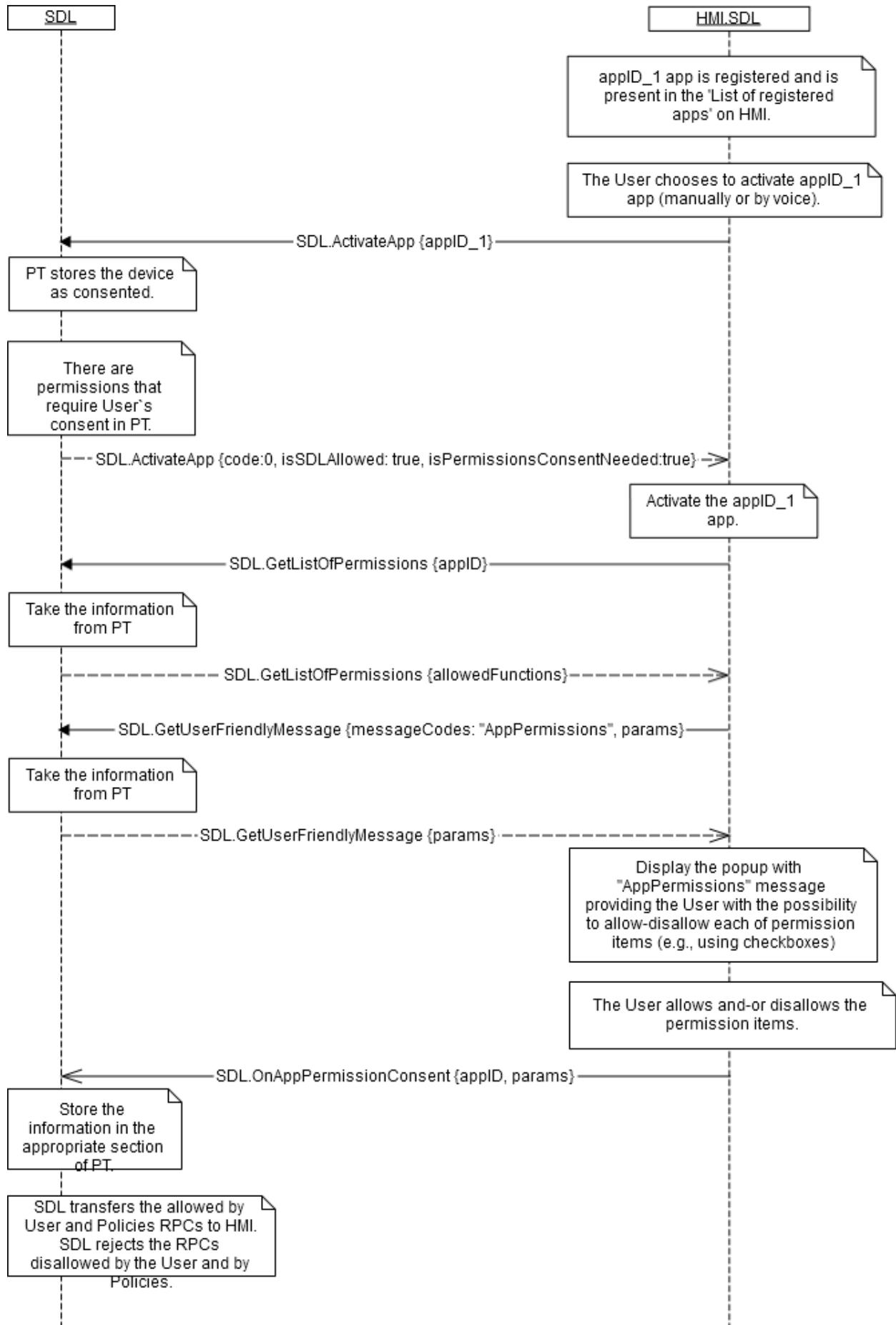
13.1.4.1 SDL.ActivateApp for the application registered from the non-consented device, which gets User's consent afterwards



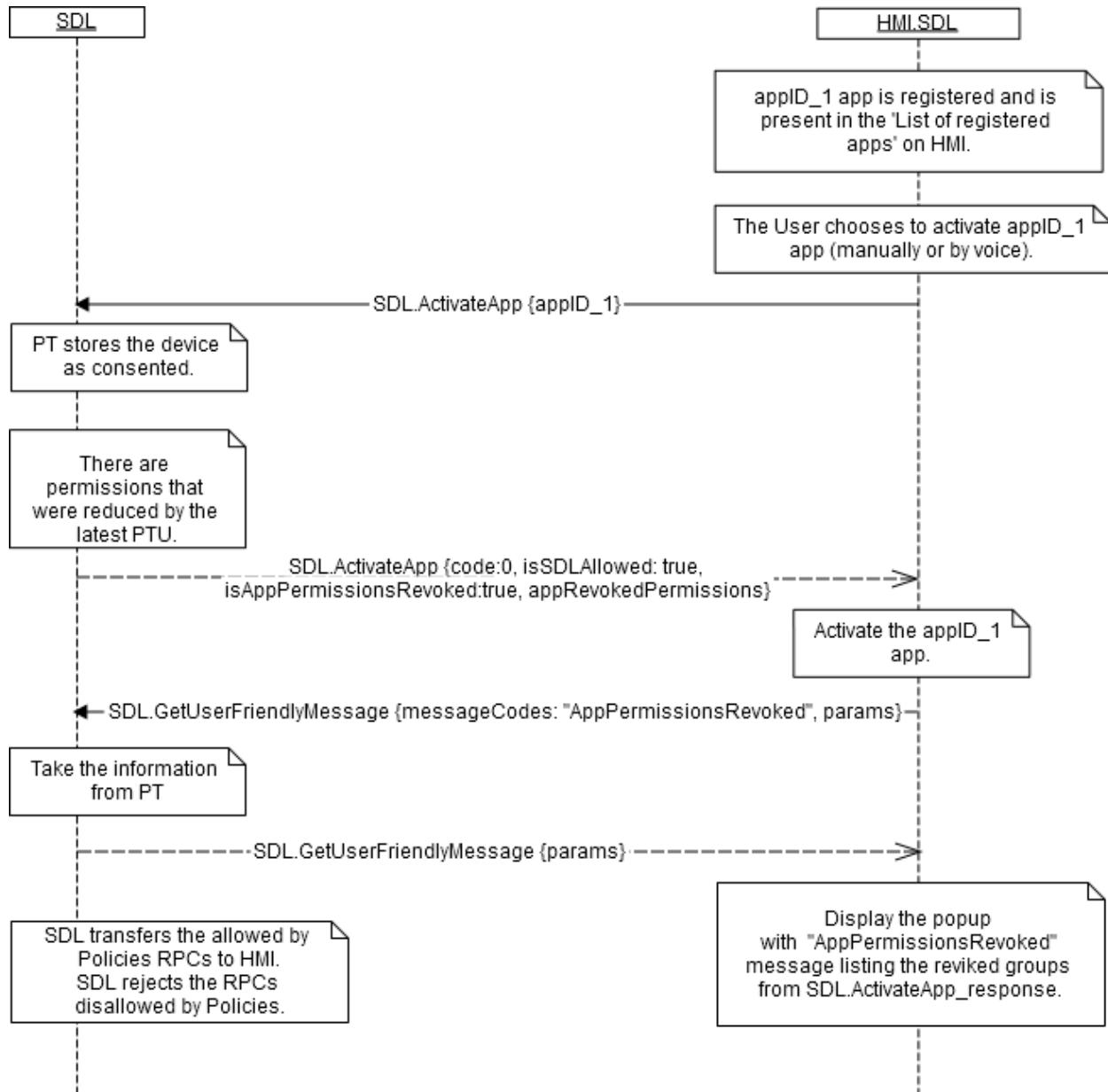
13.1.4.2 SDL.ActivateApp for the application registered from the non-consented device, which does NOT get User's consent afterwards



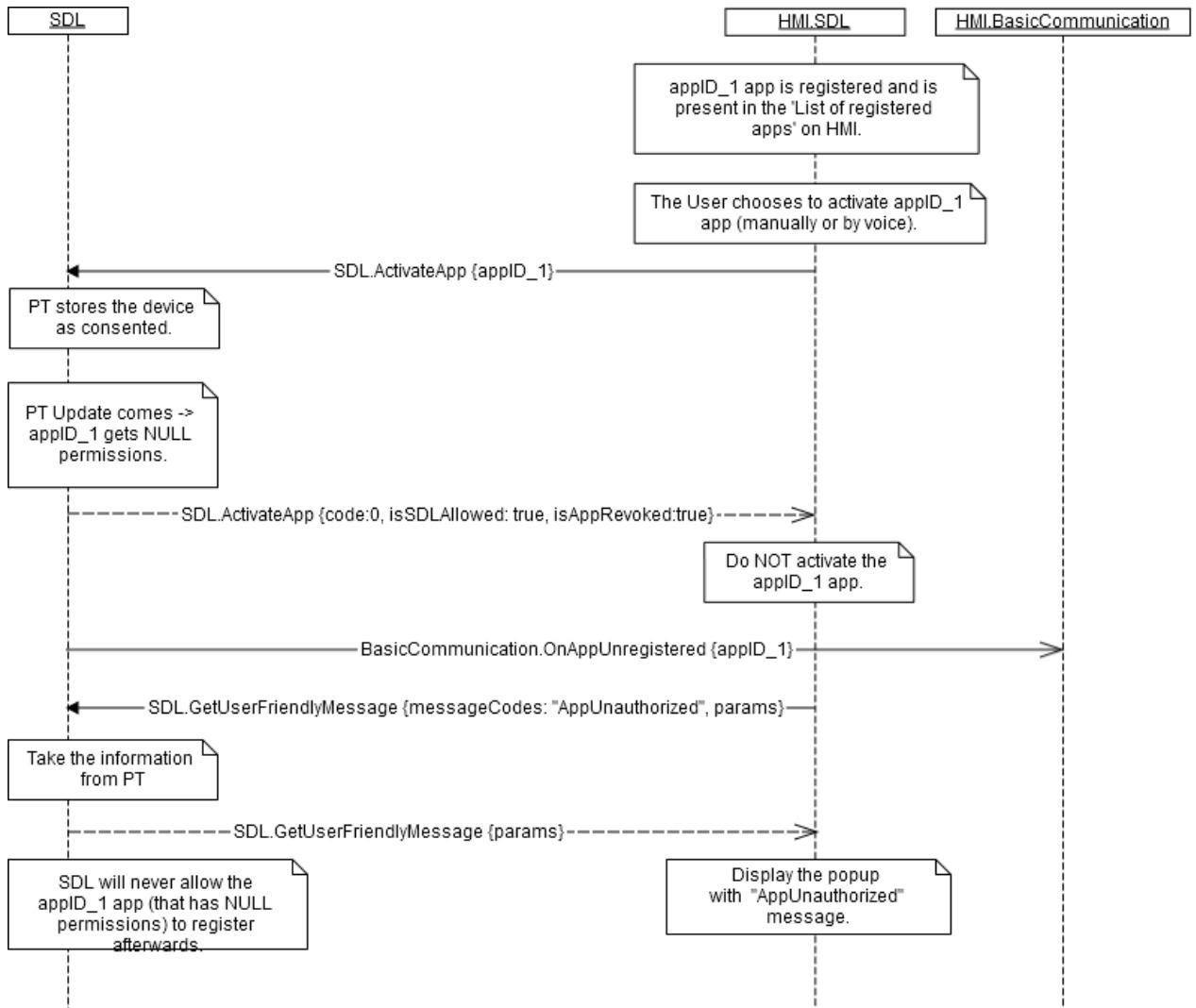
13.1.4.3 SDL.ActivateApp for the application registered from the consented device and this application requires User's consent for special permissions



13.1.4.4 *SDL.ActivateApp* for the application registered from the consented device and this application was reduced in permissions by the latest PT Update.



13.1.4.5 SDL.ActivateApp for the application registered from the consented device and this application was unauthorized by the latest PT Update.



13.1.5 JSON Messages Examples

13.1.5.1 Request

```
{
    "id" : 27,
    "jsonrpc" : "2.0",
    "method" : "SDL.ActivateApp"
    "params" :
    {
        "appId" : 12345
    }
}
```

13.1.5.2 Response

```
{
    "id" : 27,
    "jsonrpc" : "2.0",
    "result" :
    {
        "isSDLAllowed" : true,
        "isPermissionsConsentNeeded" : false,
        "isAppPermissionsRevoked" : false,
        "isAppRevoked" : false,
        "code" : 0,
    }
}
```

```

        "method" : "SDL.ActivateApp"
    }
}

```

13.1.5.3 Error message

```

{
    "id" : 27,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 22,
        "message" : "The unknown error has occurred",
        "data" :
        {
            "method" : "SDL.ActivateApp"
        }
    }
}

```

13.1.6 D-Bus Messages Examples

13.1.6.1 Request

13.1.6.2 Response

13.1.6.3 Failure

13.2 GetListOfPermissions

13.2.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | HMI |
| Purpose: | Get the list of permissions either for the named application or for all applications. |

The function returns the permission groups names the status of which (that is ‘allowed’ or ‘disallowed’) may be changed by the User.

13.2.2 Request

13.2.2.1 Behavior

HMI must:

- 1) Initiate sending GetListOfPermissions **with “appId” parameter** in the following cases:
 - 1.1) After receiving `SDL.ActivateApp{isPermissionsConsentNeeded: true}` from SDL.

Note:

appId is known from the corresponding `SDL.ActivateApp` request.

- 1.2) After receiving `SDL.OnAppPermissionChanged{appId, appPermissionsConsentNeeded: true}` from SDL.
- 1.3) After the User presses the button to change the permissions for the application.
- 2) Initiate sending GetListOfPermissions **without “appId” parameter** in the following cases:

2.1) After the User presses the button to change the permissions of all currently registered applications.

Note:

The information about the application (name, ID, etc.) is provided by SDL via either BasicCommunication.UpdateAppList or BasicCommunication.OnAppRegistered RPCs.

13.1.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|------------|--|
| appID | integer | false | - | ID of the application the list of permissions is required for. |

13.2.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|----------------|--------------------|---------------|------------------------------|---|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: | JSON response | Method return | allowedFunctions, code: 0 | |
| Failure | INVALID_DATA: | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. |
| | ABORTED: | | | Code: 5 | |
| | GENERIC_ERROR: | | | code: 22 | |

13.2.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------------|-----------------------|-----------|--|---|
| allowedFunctions | Common.PermissionItem | true | array="true" minsize="0" maxsize="100" | If no permissions were specified for application the array will come empty. |

13.2.4 Sequence Diagrams

13.2.4.1 2 GetListOfPermissions

13.2.5 JSON Messages Examples

13.2.5.1 Request

```
{
    "id" : 143,
    "jsonrpc" : "2.0",
    "method" : "SDL.GetListOfPermissions",
```

```

"params" :
{
  "appID" : 65596
}

```

13.2.5.2 Response

```

{
  "id" : 143,
  "jsonrpc" : "2.0",
  "result" :
  {
    "allowedFunctions" :
    {

    }
    "code" : 0,
    "method" : "SDL.GetListOfPermissions"
  }
}

```

13.2.5.3 Error message

```

{
  "id" : 143,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : 15,
    "message" : "A command cannot be executed because there is NO specified
                 with appID application registered",
    "data" :
    {
      "method" : "SDL.GetListOfPermissions"
    }
  }
}

```

13.2.6 D-Bus Messages Examples

13.2.6.1 Request

13.2.6.2 Response

13.2.6.3 Failure

13.3 UpdateSDL

13.3.1 Description

| | |
|-----------------|--------------------------|
| Type: | Function |
| Sender: | HMI |
| Purpose: | Update the Policy Table. |

13.3.2 Request

13.3.2.1 Behavior

HMI must:

12.3.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|----------------|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | ABORTED: | | | Code: 5 | Please see Result Enumeration for all SDL-supported codes. |
| | GENERIC_ERROR: | | | code: 22 | |

13.3.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------------------|-----------|------------|--|
| result | Common.UpdateResult | true | - | result: no update needed, update was successful/unsuccessful etc |

13.3.4 Sequence Diagrams

13.3.4.1 ShowConstantTBT

13.3.5 JSON Messages Examples

13.3.5.1 Request

```
{  
    "id" : 543,  
    "jsonrpc" : "2.0",  
    "method" : "SDL.UpdateSDL"  
}
```

13.3.5.2 Response

```
{  
    "id" : 543,  
    "jsonrpc" : "2.0",  
    "result" :  
    {  
        "result" :  
        "code" : 0,  
    }  
}
```

```

        "method" : "SDL.UpdateSDL"
    }
}

```

13.3.5.3 Error message

```

{
    "id" : 543,
    "jsonrpc" : "2.0",
    "error" :
    {
        "code" : 22,
        "message" : "The unknown error has occurred",
        "data" :
        {
            "method" : "SDL.UpdateSDL"
        }
    }
}

```

13.3.6 D-Bus Messages Examples

13.3.6.1 Request

13.3.6.2 Response

13.3.6.3 Failure

13.4 GetStatusUpdate

13.4.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | HMI |
| Purpose: | Get information about current status of PT update process. |

13.4.2 Request

13.4.2.1 Behavior

HMI must:

13.4.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|-------------|---------------|---------------|----------------|-------|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: | JSON response | Method return | code: 0 | |

| | | | | | |
|---------|----------------|--------------------|---------------|----------|--|
| Failure | INVALID_DATA: | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

13.4.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------------------|-----------|------------|-------------|
| status | Common.UpdateResult | true | - | |

13.4.4 Sequence Diagrams

13.4.4.1 UpdateTurnList

13.4.5 JSON Messages Examples

13.4.5.1 Request

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "method" : "SDL.GetStatusUpdate",
}
```

13.4.5.2 Response

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "result" :
  {
    "status" :
    "code" : 0,
    "method" : "SDL.GetStatusUpdate"
  }
}
```

13.4.5.3 Error message

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : ,
    "message" : " ",
    "data" :
    {
      "method" : "SDL.GetStatusUpdate"
    }
  }
}
```

13.4.6 D-Bus Messages Examples

13.4.6.1 Request

| |
|--|
| |
|--|

13.4.6.2 Response

| |
|--|
| |
|--|

13.4.6.3 Failure

| |
|--|
| |
|--|

13.5 GetURLS

13.5.1 Description

| | |
|----------|--|
| Type: | Function |
| Sender: | HMI |
| Purpose: | Get list of URL for specified service type and optionally policy's application id. |

13.5.2 Request

13.5.2.1 Behavior

HMI must:

13.5.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|--------------------------------|--|
| service | Integer | true | minvalue="0" maxvalue="100" | Service type the list of urls is required for. |

13.5.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|----------------|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: | | | code: 22 | Please see Result Enumeration for all SDL-supported codes. |

13.5.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| Param Name | Type | Mandatory | Additional | Description |
|------------|--------------------|-----------|--|-------------|
| urls | Common.ServiceInfo | false | array="true" minsize="1" maxsize="100" | |

13.5.4 Sequence Diagrams

13.5.4.1 UpdateTurnList

13.5.5 JSON Messages Examples

13.5.5.1 Request

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "method" : "SDL.GetStatusUpdate",
  "params" :
  {
    "service" :
  }
}
```

13.5.5.2 Response

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "result" :
  {
    "urls" :
    [
      {}
    ]
    "code" : 0,
    "method" : "SDL.GetStatusUpdate"
  }
}
```

13.5.5.3 Error message

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" :,
    "message" : " ",
    "data" :
    {
      "method" : "SDL.GetStatusUpdate"
    }
  }
}
```

13.4.6 D-Bus Messages Examples

13.4.6.1 Request

13.4.6.2 Response

13.4.6.3 Failure

13.6 GetUserFriendlyMessage

13.6.1 Description

| | |
|----------|---|
| Type: | Function |
| Sender: | HMI |
| Purpose: | Get the user-friendly message(s) reserved in Policy Table |

13.6.2 Request

13.6.2.1 Behavior

HMI must:

13.6.2.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|--------------|-----------------|-----------|---|--|
| messageCodes | String | true | array="true" minsize="1" maxsize="100" maxlength="500" | Id of message to be received according to Policy Table i.e. StatusNeeded, Notifications, DrivingCharacteristics etc. |
| language | Common.Language | false | - | Optional parameter if HMI wants message in some other language then its current one already known to SDL. |

13.6.3 Response

Note:

There is a difference in message type for WebSocket and D-Bus connection (see the table below).

| Result | Description | Message type | | Message Params | Notes |
|---------|----------------|--------------------|---------------|----------------|--|
| | | WebSocket | D-Bus | | |
| Success | SUCCESS: | JSON response | Method return | code: 0 | |
| Failure | INVALID_DATA: | JSON error message | Method return | code: 11 | Applicable for this RPC result codes. |
| | GENERIC_ERROR: | | | code: 22 | Please see Result Enumeration for all SDL- |

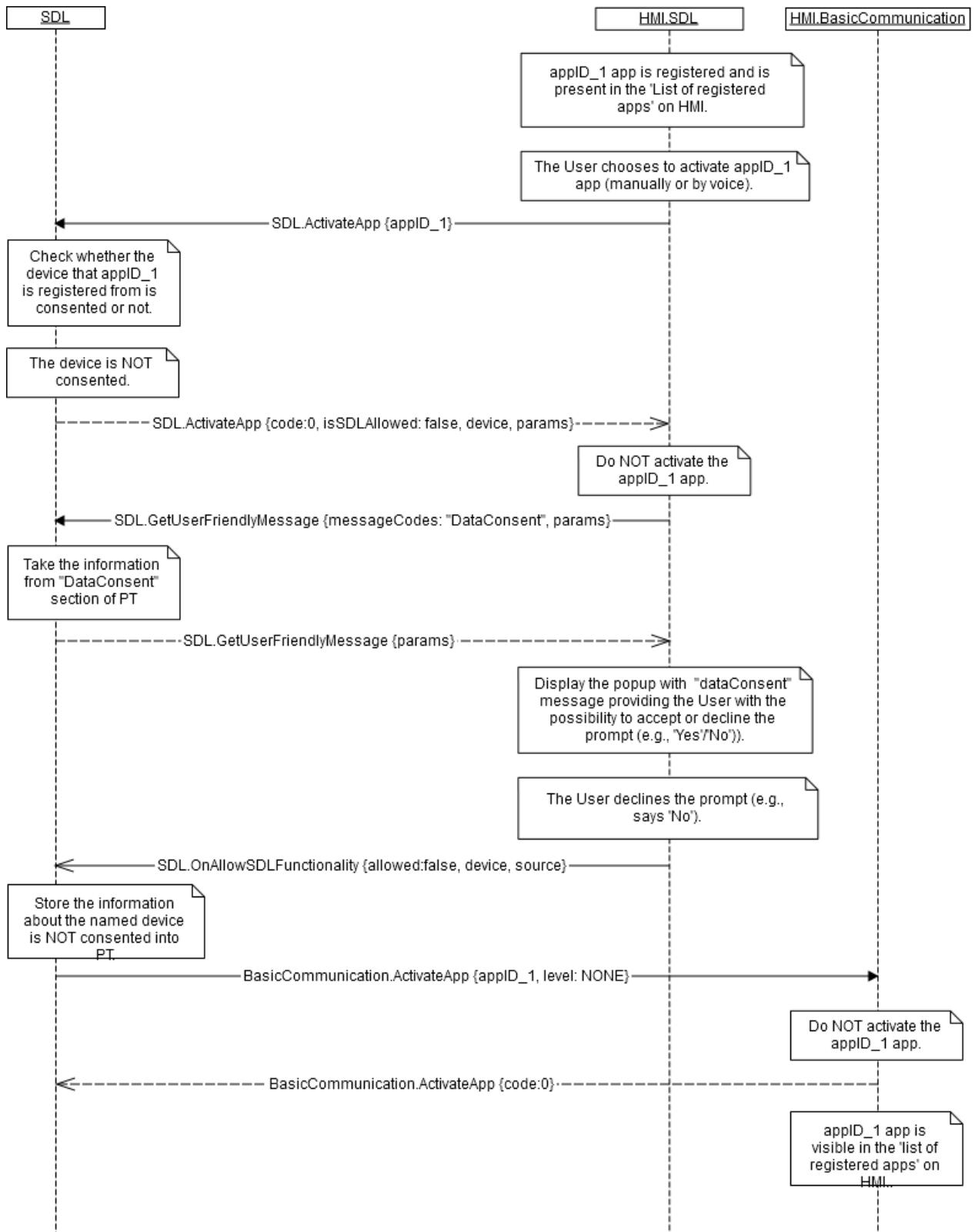
| | | | | | |
|--|--|--|--|--|------------------|
| | | | | | supported codes. |
|--|--|--|--|--|------------------|

13.6.3.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|----------------------------|-----------|--|---|
| messages | Common.UserFriendlyMessage | false | array="true" minsize="1" maxsize="100" | If no message was found in PT for specified message code and for HMI current or specified language, this parameter will be omitted. |

13.6.4 Sequence Diagrams

13.6.4.1 GetUserFriendlyMessage



13.6.5 JSON Messages Examples

13.6.5.1 Request

```
{
    "id" : 176,
    "jsonrpc" : "2.0",
    "method" : "SDL.GetUserFriendlyMessage",
    "params" :
```

```
{
  "messageCodes" :
  "language" : EN-GB
}

}
```

13.6.5.2 Response

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "result" :
  {
    "messages" :
    [
      {}
    ]
    "code" : 0,
    "method" : "SDL.GetUserFriendlyMessage"
  }
}
```

13.6.5.3 Error message

```
{
  "id" : 176,
  "jsonrpc" : "2.0",
  "error" :
  {
    "code" : ,
    "message" : " ",
    "data" :
    {
      "method" : "SDL.GetUserFriendlyMessage"
    }
  }
}
```

13.6.6 D-Bus Messages Examples

13.6.6.1 Request

13.6.6.2 Response

13.6.6.3 Failure

13.7 OnAllowSDLFunctionality

13.7.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about allowing SDL functionality |

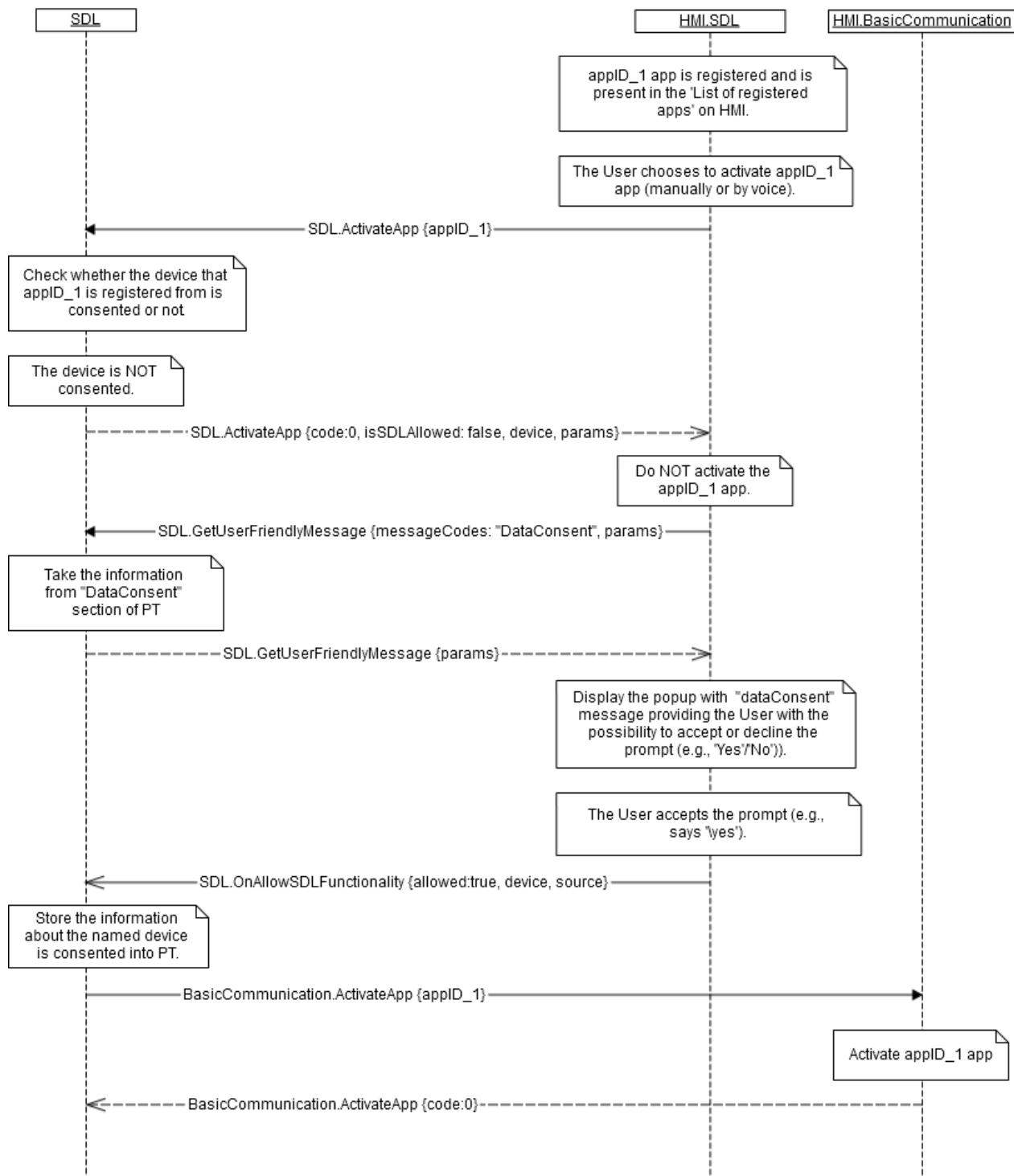
Notifies about user/HMI allowing SDL functionality or disallowing access to all mobile apps. Needed if HMI has additional ways of asking user about this (i.e. Settings etc)

13.7.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|----------------------|-----------|------------|---|
| device | Common.DeviceInfo | false | - | If no device is specified permission counts for SDL functionality in general. |
| allowed | Boolean | true | - | Must be true if allowed |
| source | Common.ContentSource | true | - | |

13.7.2 Sequence Diagrams

13.7.2.1 OnAllowSDLFunctionality



13.7.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "SDL.OnAllowSDLFunctionality",
  "params" :
  {
    "device" : SDL/application_directory/video/123.mp4,
    "allowed" : true,
  }
}
```

```

    "source" :
}
}
```

13.7.4 D-Bus Messages Examples

[REDACTED]

13.8 OnAppPermissionConsent

13.8.1 Description

| | |
|-----------------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the User permission of some functionality for the named application. |

Initiated by HMI for specifying the allowance for the application to perform some functionality. Duplicates functionality of the request, needed if HMI has specific ways to allow/disallow functionality (i.e. Setting Menu)

13.8.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|-----------------------|-----------|--|--|
| appID | Integer | true | - | ID of the application requested this RPC |
| consentedFunctions | Common.PermissionItem | true | array="true" minsize="1" maxsize="100" | |
| source | Common.ConsentSource | true | - | |

13.8.2 Sequence Diagrams

13.8.2.1 OnAppPermissionConsent

[REDACTED]

13.8.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "SDL.OnAppPermissionConsent",
  "params" :
  {
    "appID" : 65674
  }
}
```

13.8.4 D-Bus Messages Examples

[REDACTED]

13.9 OnAppPermissionChanged

13.9.1 Description

| | |
|-----------------|--|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about reduced permissions for the application |

Notification from SDL to HMI. Occurs when app permissions were reduced. If no permission specified means that app was disallowed and has to be unregistered.

13.9.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|-----------------------------|-----------------------|-----------|--|--|
| appID | Integer | true | | |
| isAppPermissionsRevoked | Boolean | false | | |
| appRevokedPermissions | Common.PermissionItem | false | array="true" minsize="1" maxsize="100" | If app permissions were reduced (isAppPermissionsRevoked == true), then this array specifies list of removed permissions |
| appRevoked | Boolean | false | | If present then specified application was prohibited to used with Sync. |
| appPermissionsConsentNeeded | Boolean | false | | If present specifies that permissions were added to application that require User Consent, then HMI can send GetListOfPermissions request to obtain list of permissions. |
| appUnauthorized | Boolean | false | | When present and set to true (should be if present) then this means that application was not authorized (nickname check failed.) |
| priority | Common.AppPriority | false | | Send to HMI so that it can coordinate order of requests/notifications correspondingly. |

13.9.1.3 AppPriority

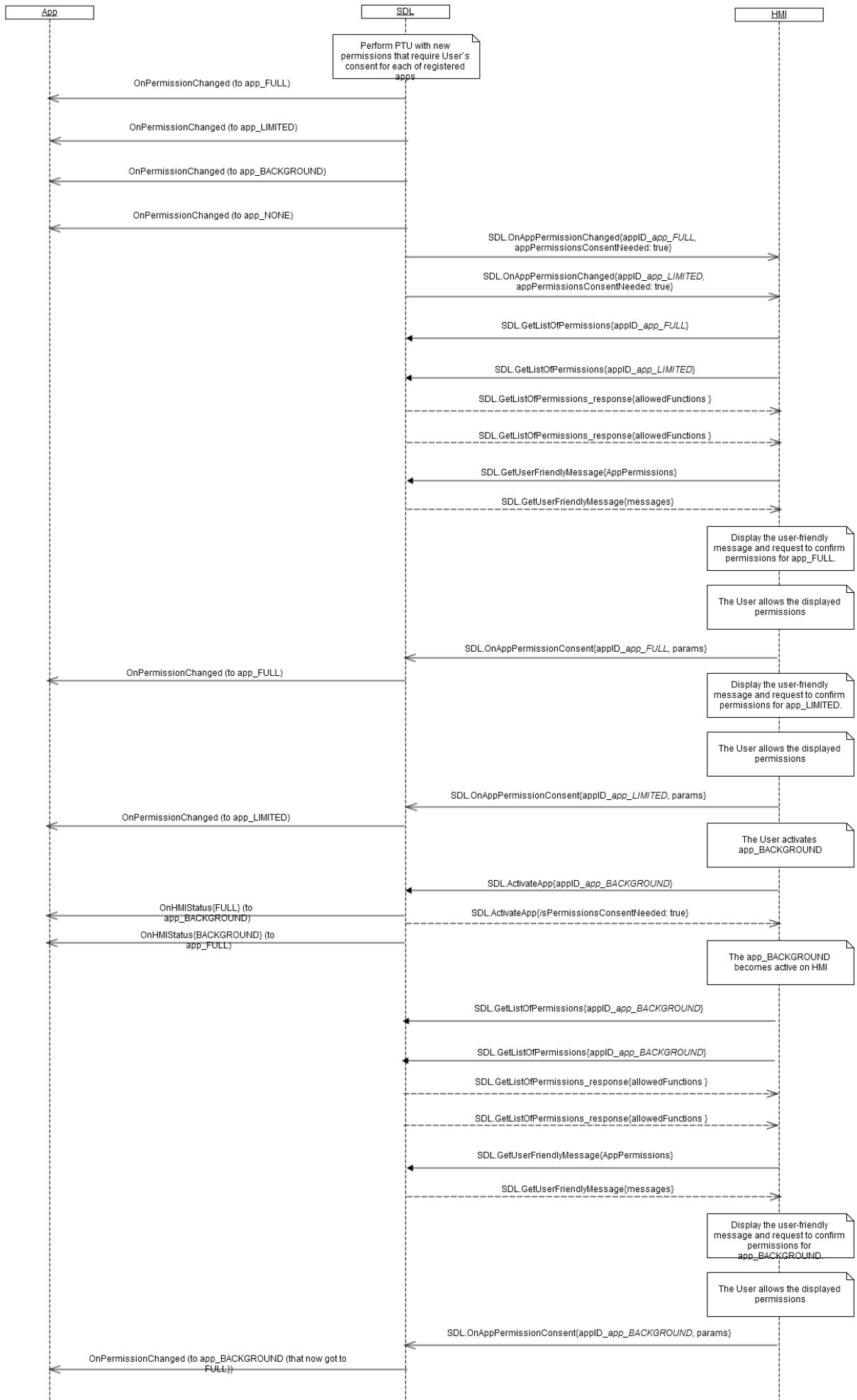
| Element name | Value | Short Description |
|---------------------|-------|-------------------|
| EMERGENCY | 0 | |
| NAVIGATION | 1 | |
| VOICE_COMMUNICATION | 2 | |
| COMMUNICATION | 3 | |
| NORMAL | 4 | |
| NONE | 5 | |

13.9.2 Sequence Diagrams

13.9.2.1 *OnAppPermissionChanged with appPermissionsConsentNeeded:true*

Pre-conditions to the sequence:

- a) SDL and HMI are started
- b) Device is connected to the System (SDL/HU) and is consented by the User.
- c) Four apps are registered with SDL and HMI ('name_HMILevel'): app_FULL, app_LIMITED, app_BACKGROUND, app_NONE.



13.9.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "SDL.OnAppPermissionChanged",  
    "params" :  
    {  
        "appID" : 65674,  
    }  
}
```

13.9.4 D-Bus Messages Examples

```
[REDACTED]
```

13.10 OnSDLConsentNeeded

13.10.1 Description

| | |
|----------|---------------------------------------|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about stop of video streaming. |

Send from SDL to HMI to notify that data consent is needed for device either because PT update or Retry strategy.

13.10.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|-------------------|-----------|------------|--|
| device | Common.DeviceInfo | true | - | Device that needs to receive the User's consent from HMI side. |

13.10.2 Sequence Diagrams

13.10.2.1 OnSDLConsentNeeded

13.10.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "SDL.OnSDLConsentNeeded",  
    "params" :  
    {  
        "device" :  
    }  
}
```

13.10.4 D-Bus Messages Examples

| |
|--|
| |
|--|

13.11 OnStatusUpdate

13.11.1 Description

| | |
|-----------------|--|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform the status of Policy Table update |

13.11.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|---------------------|-----------|---|
| status | Common.UpdateResult | true | Notification from SDL to HMI when current status of PT exchange changed (i.e. it Succeeded or Failed etc) |

13.11.2 Sequence Diagrams

13.11.2.1 OnStatusUpdate

13.11.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "SDL.OnStatusUpdate",
  "params" :
  {
    "status" :
  }
}
```

13.11.4 D-Bus Messages Examples

| |
|--|
| |
|--|

13.12 OnSystemError

13.12.1 Description

| | |
|-----------------|---------------------------------|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about the error occurred |

13.12.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|------------|------|-----------|-------------|
|------------|------|-----------|-------------|

| Param Name | Type | Mandatory | Description |
|------------|--------------------|-----------|----------------|
| error | Common.SystemError | true | Error occurred |

13.12.2 Sequence Diagrams

13.12.2.1 OnSystemError

13.12.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "SDL.OnSystemError",
    "params" :
    {
        "status" :
    }
}
```

13.12.4 D-Bus Messages Examples

| |
|--|
| |
|--|

13.13 AddStatisticsInfo

13.13.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Provide statistics data to be recorded in Policy Table. |

13.13.1.1 Parameters

| Param Name | Type | Mandatory | Description |
|---------------|-----------------------|-----------|---|
| statisticType | Common.StatisticsType | true | Sent by system to record statscs and error counts. Increases statistics specified by statisticType by one |

13.13.2 Sequence Diagrams

13.13.2.1 AddStatisticsInfo

13.13.3 JSON Messages Examples

```
{
    "jsonrpc" : "2.0",
    "method" : "SDL.AddStatisticsInfo",
    "params" :
    {
        "status" :
    }
}
```

```
}
```

13.13.4 D-Bus Messages Examples

13.14 OnDeviceStateChanged

13.14.1 Description

| | |
|----------|----------------------------------|
| Type: | Notification |
| Sender: | SDL |
| Purpose: | Inform about device state change |

13.14.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------------|--------------------|-----------|----------------------------------|-------------|
| deviceState | Common.DeviceState | true | | |
| deviceInternalId | String | true | minlength="0" maxlength="500" | |
| deviceId | Common.DeviceInfo | false | | |

13.14.1.2 DeviceState

| Element name | Value | Short Description |
|--------------|-------|-------------------|
| UNKNOWN | 0 | |
| UNPAIRED | 1 | |

13.14.2 Sequence Diagrams

13.14.2.1 OnDeviceStateChanged

13.14.3 JSON Messages Examples

```
{
  "jsonrpc" : "2.0",
  "method" : "SDL.OnDeviceStateChanged",
  "params" :
  {
    "status" :
  }
}
```

13.14.4 D-Bus Messages Examples

13.15 OnReceivedPolicyUpdate

13.15.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Inform about the PolicyTable is decrypted and can be applied by SDL's PoliciesManager |

13.15.1.1 Parameters

| Param Name | Type | Mandatory | Additional | Description |
|------------|--------|-----------|--------------------------------|--|
| policyfile | String | true | minlength= 1 maxlength= 255 | Location of decrypted policy table Json file on target |

13.15.2 Sequence Diagrams

13.15.2.1 OnReceivedPolicyUpdate

13.15.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "SDL.OnReceivedPolicyUpdate",  
    "params" :  
    {  
        "policyfile" : "example/path/to/the-file.json"  
    }  
}
```

13.15.4 D-Bus Messages Examples

```
[REDACTED]
```

13.16 OnPolicyUpdate

13.16.1 Description

| | |
|----------|---|
| Type: | Notification |
| Sender: | HMI |
| Purpose: | Informs about the new Policy Update flow must be triggered. |

13.16.2 Sequence Diagrams

13.16.2.1 *OnReceivedPolicyUpdate*

13.16.3 JSON Messages Examples

```
{  
    "jsonrpc" : "2.0",  
    "method" : "SDL.OnPolicyUpdate",  
}
```

13.16.4 D-Bus Messages Examples

```
[REDACTED]
```

14 Common Component Description

14.1 Enumerations

14.1.1 Result

The enumeration defines the possible result codes for all operations provided by API between SDL and HMI.

SUCCESS is the only result code that notifies that no errors, nonstandard situations or lack of resources/privileges were encountered.

| Element Name | Value | Description |
|----------------------|-------|---|
| SUCCESS | 0 | The request is executed successfully. |
| UNSUPPORTED_REQUEST | 1 | The request is not supported. |
| UNSUPPORTED_RESOURCE | 2 | The requested resource is not supported under the current system. E.g.: notifications from the button are requested but physically this button is not present on HU. |
| DISALLOWED | 3 | API call is not allowed by HMI. |
| REJECTED | 4 | The requested command is rejected. E.g.: - Because the mobile app is in background and cannot perform any commands from HMI. - Or an HU command (e.g. Speak) is rejected because a higher priority HU command (e.g. Alert) is playing. |
| ABORTED | 5 | A requested command was aborted. E.g.: - Due to user interaction (e.g. user pressed the button). - Or an HMI command (e.g. Speak) is aborted because a higher priority HMI command (e.g. Alert) was requested. |
| IGNORED | 6 | A command was ignored, because the intended result is already in effect. E.g.: - SetMediaClockTimer was used to pause the media clock although the clock is paused already. |
| RETRY | 7 | The user interrupted the RPC and indicated to start over. E.g. : PerformAudioPassThru . |
| IN_USE | 8 | The data may not be changed, because it is currently in use. E.g.: when trying to delete a command set that is currently involved in an interaction. |
| DATA_NOT_AVAILABLE | 9 | The requested data is not available. |

| Element Name | Value | Description |
|----------------------------|-------|---|
| TIMED_OUT | 10 | Overlay reached the maximum timeout and closed. |
| INVALID_DATA | 11 | <p>The data sent is invalid.</p> <p>E.g.:</p> <ul style="list-style-type: none"> - Invalid Json syntax - Parameters out of bounds (number or enum range) - Mandatory parameters not provided - Parameter provided with wrong type - Invalid characters - Empty string |
| CHAR_LIMIT_EXCEEDED | 12 | The string data is too long. |
| INVALID_ID | 13 | One of the provided IDs is not valid. |
| DUPLICATE_NAME | 14 | There was a conflict with an already registered name. |
| APPLICATION_NOT_REGISTERED | 15 | A command cannot be executed because there is no registered application specified with AppID. |
| WRONG_LANGUAGE | 16 | The requested language is currently not supported. |
| OUT_OF_MEMORY | 17 | The system could not process the request because the necessary memory RAM couldn't be allocated |
| TOO_MANY_PENDING_REQUESTS | 18 | There are too many requests pending (means, that the response cannot be delivered yet) |
| NO_APPS_REGISTERED | 19 | The request cannot be executed because no application interface has been registered. |
| NO_DEVICES_CONNECTED | 20 | No device has been connected. |
| WARNINGS | 21 | The RPC is executed successfully but one or more items have a warning or failure (e.g. Speak). |
| GENERIC_ERROR | 22 | During the API call an unknown or unspecified within the current Result Enumeration error has occurred. |
| USER_DISALLOWED | 23 | RPC is included in a functional group explicitly blocked by the user. |

Related items:

[Table of Contents](#).

14.1.2 AppHMIType

The enumeration that lists possible types of the application.

| Element name | Short Description |
|---------------|-----------------------------------|
| DEFAULT | The application of default type. |
| COMMUNICATION | The application for communication |
| MEDIA | The media application |
| MESSAGING | The application of messaging type |

| Element name | Short Description |
|--------------------|---|
| NAVIGATION | The application of navigation type |
| INFORMATION | The application of information type |
| SOCIAL | The application of social type |
| BACKGROUND_PROCESS | The application does not require displaying the information |
| TESTING | The application of testing type |
| SYSTEM | The application of system type |

Related items:

HMIApplication – [section 5.2.1](#)

[Table of Contents](#).

14.1.3 DeactivateReason

This enumeration specifies the non-application functionality the User may switch to.

| Element name | Short Description |
|---------------|---|
| AUDIO | The User has navigated to audio (radio, etc.) |
| PHONECALL | The User has navigated to make a call. |
| NAVIGATIONMAP | The User has navigated to navigation screen. |
| PHONEMENU | The User has navigated to phone menu. |
| SYNCSETTINGS | The User has navigated to HU settings menu. |
| GENERAL | Other screens navigation apart from other mobile app. |

Related items:

OnAppDeactivated – [section 6.2.3](#)

[Table of Contents](#).

14.1.4 ApplicationsCloseReason

This enumeration describes the possible reasons for exiting from all of registered applications.

| Element name | Short Description |
|------------------|---|
| IGNITION_OFF | When the ignition is off. |
| MASTER_RESET | When the master reset takes place. |
| FACTORY_DEFAULTS | When the factory default settings are chosen. |

Related items:

OnExitAllApplications – [section 6.2.11](#)

[Table of Contents](#).

14.1.5 ClockUpdateMode

The enumeration describes how HMI must update the media clock timer. The details are in the table below.

| Element name | Short Description |
|--------------|--|
| COUNTUP | HMI must start the media clock timer counting upwards, as in time elapsed, in increments of 1 second. |
| COUNTDOWN | HMI must start the media clock timer counting downwards, as in time remaining, in decrements of 1 second. |
| PAUSE | HMI must pause the media clock timer. |
| RESUME | HMI must resume the media clock timer. The timer must resume counting in whatever mode was in effect before pausing (i.e. COUNTUP or COUNTDOWN). |
| CLEAR | HMI must clear the media clock timer (previously set through setMediaClockTimer). |

Related Items:

`setMediaClockTimer` – [section 7.1.11](#)

[Table of Contents](#).

14.1.6 SystemContext

Enumeration that describes possible contexts the application might be in on HU.

| Element name | Short Description |
|--------------|---|
| MAIN | If there is currently no user interaction (user-initiated or app-initiated) with the head-unit, the application must be notified the the SystemContext is MAIN. |
| VRSESSION | Must be sent if there is a current user interaction that is VR-oriented (VR becomes active as a result of PTT or PerformInteraction). |
| MENU | Must be sent if HMI is currently displaying an in-application menu onscreen. |
| HMI_OBSCURED | Must be sent if HU is currently obscuring the application display either with a system or with another application overlay (except of Alert element). |
| ALERT | Must be sent if the Alert message is currently displayed onscreen. |

Related items:

`OnSystemContext` – [section 7.2.2](#)

[Table of Contents](#).

14.1.7 DisplayType

HMI must send the information about the supported displays within [DisplayCapabilities](#) structure of the [GetCapabilities](#) RPC. The display types are described in the enumeration below.

| Element name | Short Description |
|--------------|---|
| CID | Center Information Display. This display type provides a 2-line x 20 character "dot matrix" display. |
| TYPE2 | TYPE II display. 1 line older radio head unit. |

| Element name | Short Description |
|---------------------|--|
| TYPE5 | TYPE V display Old radio head unit. |
| NGN | Next Generation Navigation display. |
| GEN2_8_DMA | GEN-2, 8 inch display. |
| GEN2_6_DMA | GEN-2, 6 inch display. |
| MFD3 | 3 inch GEN1.1 display |
| MFD4 | 4 inch GEN1.1 display |
| MFD5 | 5 inch GEN1.1 display |
| GEN3_8-INCH | GEN-3, 8 inch display. |

Related items:

DisplayCapabilities – [section 5.2.7](#)

GetCapabilities – [section 7.1.2](#)

[Table of Contents](#).

14.1.8 MediaClockFormat

The enumeration describes the format of media clock value supported by HMI.

Media clock is used by media applications for indicating the total/remaining duration of a song/album.

Upon [GetCapabilities](#) request from SDL HMI must provide the information on supported media clock format sent within [DisplayCapabilities](#) structure. Further SDL will use the accurate values of the corresponding format for setting the time of media clock on UI.

| Element name | Short Description |
|---------------------|---|
| CLOCK1 | minutesFieldWidth = 2; minutesFieldMax = 19; secondsFieldWidth = 2; secondsFieldMax = 99; maxHours = 19; maxMinutes = 59; maxSeconds = 59; Is used for Type II, NGN and CID head units. |
| CLOCK2 | minutesFieldWidth = 3; minutesFieldMax = 199; secondsFieldWidth = 2; secondsFieldMax = 99; maxHours = 59; maxMinutes = 59; maxSeconds = 59; Is used for Type V head units. |
| CLOCK3 | minutesFieldWidth = 2; minutesFieldMax = 59; secondsFieldWidth = 2; secondsFieldMax = 59; maxHours = 9; maxMinutes = 59; maxSeconds = 59; Is used for GEN1.1 (i.e. MFD3/4/5) head units. |

| Element name | Short Description |
|---------------------|--|
| CLOCKTEXT1 | 5 characters possible Format: 1 sp c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space Is used for Type II head unit |
| CLOCKTEXT2 | 5 chars possible Format: 1 sp c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space Is used for CID and NGN head unit. |
| CLOCKTEXT3 | 6 chars possible Format: 1 sp c c : sp c c 1 sp : digit "1" or space c : character out of following character set: sp 0-9 [letters : sp : colon or space Is used for Type V head unit. |
| CLOCKTEXT4 | 6 chars possible Format: c : sp c c : c c : sp : colon or space c : character out of following character set: sp 0-9 [letters]. Is used for GEN1.1 (i.e. MFD3/4/5) head units. |

Related items:

DisplayCapabilities – [section 5.2.7](#)

TextFieldName – [section 5.1.14](#)

[Table of Contents](#).

14.1.9 HmiZoneCapabilities

Upon [GetCapabilities](#) request, HMI must provide the information about the zone it is available from:

| Element name | Short Description |
|---------------------|--|
| FRONT | Indicates HMI available for front seat passengers. |
| BACK | Indicates HMI available for rear seat passengers. |

Related items:

GetCapabilities – for UI, [section 7.1.2](#)

[Table of Contents](#).

14.1.10 SpeechCapabilities

Upon TTS . GetCapabilities request, HMI must provide the information about the zone it is available from:

| Element name | Short Description |
|---------------------|--------------------------|
| TEXT | |
| SAPI_PHONEMES | |
| LHPLUS_PHONEMES | |
| PRE_RECORDED | |
| SILENCE | |

Related items:

GetCapabilities – for TTS, section

[Table of Contents](#).

14.1.11 VrCapabilities

Upon TTS . GetCapabilities request, HMI must provide the information about the zone it is available from:

| Element name | Short Description |
|---------------------|--------------------------|
| TEXT | |

Related items:

GetCapabilities – for VR, section

[Table of Contents](#).

13.1.12 DriverDistractionState

HMI must provide the information about either the driver distraction rules are in effect or not. This enumeration describes the possible states of driver distraction.

| Element name | Short Description |
|---------------------|---|
| DD_ON | Driver Distraction rules are in effect. |
| DD_OFF | Driver Distraction rules are not in effect. |

Related items:

OnDriverDistraction – [section 7.2.4](#)

[Table of Contents](#).

13.1.11 SoftButtonType

The enumeration defines the types of the soft buttons to be displayed on UI component:

- 1) The text is displayed on the soft button
- 2) The image is displayed on the soft button
- 3) Both image and text are displayed on the soft button.

| Element name | Short Description |
|--------------|-------------------------------|
| TEXT | Text displayed |
| IMAGE | Image displayed |
| BOTH | Both text and image displayed |

Related items:

SoftButton – [section 5.2.10](#)

[Table of Contents](#).

13.1.12 SystemAction

The HMI may provide three possible ways of reacting on soft button click (the exact behavior may vary and be platform-dependent):

- 1) Default action (DEFAULT_ACTION). For example, if the soft button is preset for cleaning the overlay, the click on it should close the pop-up or clean the overlay.
- 2) Returning application to the full mode (STEAL_FOCUS). The soft button with presets for this action should close the pop-up window and return the application which caused that pop-up to the full mode.
- 3) Making the pop-up or event remain active (KEEP_CONTEXT). The pop-up or event should be left on the screen after clicking the soft button with presets for this action, renewing the timeout for this event.

The responsibility of behavior on System Action implementation is on HMI.

If HMI does not support some System Action it has to respond correspondingly.

The following enumeration defines the system actions for the soft button click:

| Element name | Short Description |
|----------------|---|
| DEFAULT_ACTION | Default action should occur. Standard behavior (e.g. SoftButton clears overlay) |
| STEAL_FOCUS | - The pop-up/dialog should be closed - The app, having been obscured with that pop-up/dialog, should be returned into the full mode. |
| KEEP_CONTEXT | - The pop-up/dialog should be left on the screen - The timeout for this pop-up/dialog should be renewed |

Related items:

SoftButton – [section 5.2.10](#)

[Table of Contents](#).

13.1.13 TextAlignement

This enumeration lists the variants of text alignment.

| Element name | Short Description |
|---------------|------------------------|
| LEFT_ALIGNED | Text is aligned left. |
| RIGHT_ALIGNED | Text is aligned right. |

| Element name | Short Description |
|---------------------|--------------------------|
| CENTERED | Text is centered. |

Related items:

Show – [section 7.1.5](#)

[Table of Contents](#).

13.1.14 Language

The enumeration defines the possible languages that the HU components might support and the SDL might request:

| Element name | Short Description |
|---------------------|--------------------------------|
| AR-SA | Arabic – Saudi Arabia |
| CS-CZ | Czech – Czech Republic |
| DA-DK | Danish – Denmark |
| DE-DE | German – Germany |
| EN-AU | English – Australia |
| EN-GB | English – GB |
| EN-US | English – US |
| ES-ES | Spanish – Spain |
| ES-MX | Spanish – Mexico |
| FR-CA | French – Canada |
| FR-FR | French – France |
| IT-IT | Italian – Italy |
| JA-JP | Japanese – Japan |
| KO-KR | Korean – South Korea |
| NL-NL | Dutch (Standard) – Netherlands |
| NO-NO | Norwegian - Norway |
| PL-PL | Polish – Poland |
| PT-PT | Portuguese – Portugal |
| PT-BR | Portuguese – Brazil |

| Element name | Short Description |
|---------------------|--------------------------|
| RU-RU | Russian - Russia |
| SV-SE | Swedish – Sweden |
| TR-TR | Turkish – Turkey |
| ZH-CN | Mandarin – China |
| ZH-TW | Mandarin – Taiwan |

Related items:

HMIApplication – [section 5.2.1](#)

GetSupportedLanguages – for UI, [section 7.1.3](#)
 – for VR, [section 9.1.2](#)
 – for TTS, [section 10.1.2](#)

ChangeRegistration – for UI, [section 7.1.15](#)
 – for VR, [section 9.1.5](#)
 – for TTS, [section 10.1.5](#)

GetLanguage – for UI, [section 7.1.16](#)
 – for VR, [section 9.1.6](#)
 – for TTS, [section 10.1.5](#)

OnLanguageChange – for UI, [section 7.2.3](#)
 – for VR, [section 9.2.4](#)
 – for TTS, [section 10.2.1](#)

[Table of Contents](#).

13.1.15 TextFieldName

The enumeration describes the possible fields for the text information to be displayed.

Upon SDL's request HMI must provide the list of supported fields.

Further SDL will request displaying the text information in these fields within different requests to HMI.

The table below contains the information about:

- The name of the field
- Conditions of displaying the text
- The name of the method concerned

| Element name | Short Description |
|---------------------|---|
| mainField1 | The text that must be displayed in a single or upper display line. If this value is not set, the text of mainField1 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.1.5 . |
| mainField2 | The text that must be displayed on the second display line. If this text is not set, the text of mainField2 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.1.5 . |

| Element name | Short Description |
|------------------------|--|
| mainField3 | <p>The text that must be displayed on the second "page" first display line. If this text is not set, the text of mainField3 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.1.5.</p> |
| mainField4 | <p>The text that must be displayed on the second "page" second display line. If this text is not set, the text of mainField4 must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.1.5.</p> |
| statusBar | <p>The text is placed in the status bar area. Note: This relates to navigation displays If this parameter is omitted, the status bar text must remain unchanged. If this parameter is an empty string, the field must be cleared. If provided and the display has no status bar, this parameter must be ignored. Applies to Show, section 7.1.5.</p> |
| mediaClock | <p>Text value for MediaClock field. Shall arrive in the form as described in the MediaClockFormat enumeration If this text is set, any automatic media clock updates previously set with SetMediaClockTimer must be stopped. Applies to Show, section 7.1.5.</p> |
| mediaTrack | <p>The text that should be displayed in the track field. This field should be valid only for media applications on. If this text is not set, the text of mediaTrack must stay unchanged. If this text is empty "", the field must be cleared. Applies to Show, section 7.1.5.</p> |
| alertText1 | <p>The text that must be displayed in the top field of the display during the Alert. Applies to Alert, section 7.1.4.</p> |
| alertText2 | <p>The text that must be displayed in the bottom field of the display during the Alert. Applies to Alert, section 7.1.4.</p> |
| alertText3 | <p>The optional third line of the alert text field. Applies to Alert, section 7.1.4.</p> |
| scrollableMessageBody | <p>The long form body of text that can include newlines and tabs. Applies to ScrollableMessage, section 7.1.17.</p> |
| initialInteractionText | <p>Must be displayed when the interaction begins. The text must be displayed on the first line of a multiline display, and must be centered. Applies to PerformInteraction, section 7.1.10.</p> |
| navigationText1 | <p>The text that must be displayed on the first line of navigation text. Applies to ShowConstantTBT, section 12.1.3.</p> |
| navigationText2 | <p>The text that must be displayed on the second line of navigation text. Applies to ShowConstantTBT, section 12.1.3.</p> |
| ETA | <p>Estimated Time of Arrival for navigation. Applies to ShowConstantTBT, section 12.1.3.</p> |
| totalDistance | <p>Total distance to destination for navigation. Applies to ShowConstantTBT, section 12.1.3.</p> |

| Element name | Short Description |
|---------------------------|---|
| navigationText | Navigation text for UpdateTurnList. Applies to Turn, section 5.2.25 . |
| audioPassThruDisplayText1 | The first line of text that must be displayed during audio capture. Applies to PerformAudioPassThru, section 7.1.20 . |
| audioPassThruDisplayText2 | The second line of text that must be displayed during audio capture. Applies to PerformAudioPassThru, section 7.1.20 . |
| sliderHeader | The text that must be displayed on the header of slider. Applies to Slider, section 7.1.18 . |
| sliderFooter | The text that must be displayed on the footer of slider. Applies to Slider, section 7.1.18 . |
| notificationText | The text that must be displayed to notify the User on some event. Applies to ShowNotification, section 7.2.6 . |
| menuName | Primary text for Choice |
| secondaryText | Secondary text for Choice |
| tertiaryText | Tertiary text for Choice |
| timeToDestination | |
| turnText | |

Related Items:

MediaClockFormat – [section 5.1.8](#)
 DisplayCapabilities – [section 5.2.7](#)
 TextFieldStruct – [section 5.2.8](#)
 Turn – [section 5.2.25](#)
 Alert – [section 7.1.4](#)
 Show – [section 7.1.5](#)
 PerformInteraction – [section 7.1.10](#)
 Slider – [section 7.1.16](#)
 ScrollableMessage – [section 7.1.17](#)
 PerformAudioPassThru – [section 7.1.18](#)
 ShowNotification – [section 7.2.5](#)
 ShowConstantTBT – [section 12.1.3](#)

[Table of Contents](#).

13.1.17 ImageFieldName

This enumeration contains the information about the type of the image.

| Element name | Short Description |
|---------------------|--------------------------------|
| softButtonImage | The image field for SoftButton |

| Element name | Short Description |
|-----------------------------|--|
| choiceImage | The first image field for Choice |
| choiceSecondaryImage | The secondary image field for Choice |
| vrHelpItem | The image field for vrHelpItem |
| turnIcon | The image field for Turn |
| menuIcon | The image field for the menu icon in SetGlobalProperties |
| cmdIcon | The image field for AddCommand |
| graphic | The image field for Show |
| showConstantTBTIcon | The primary image field for ShowConstantTBT |
| showConstantTBTNextTurnIcon | The secondary image field for ShowConstantTBT |
| nextTurnIcon | |

Related items:

DisplayCapabilities – [section 5.2.7](#)

Image – [section 5.2.11](#)

[Table of Contents](#).

13.1.18 ImageType

This enumeration contains the information about the type of the image.

| Element name | Short Description |
|---------------------|---|
| STATIC | Static image. The image that is sent as the binary or hex code within the request. |
| DYNAMIC | Dynamic image. The image that is stored on HMI and just a link to it is further used within requests. |

Related items:

DisplayCapabilities – [section 5.2.7](#)

Image – [section 5.2.11](#)

[Table of Contents](#).

13.1.17 ButtonName

The following enumeration defines the hard (physical) and soft (touchscreen) buttons that may be available from HMI.

| Element name | Short Description |
|---------------------|--------------------------|
|---------------------|--------------------------|

| Element name | Short Description |
|---------------|---|
| OK | Represents the button usually labeled "OK". A typical use of this button is for the User to press it to make a selection. |
| SEEKLEFT | Represents the seek-left button. A typical use of this button is for the user to scroll to the left through menu choices, one menu item per press. |
| SEEKRIGHT | Represents the seek-right button. A typical use of this button is for the user to scroll to the right through menu choices one menu item per press. |
| TUNEUP | Represents a turn of the tuner knob in the clockwise direction one tick. |
| TUNEDOWN | Represents a turn of the tuner knob in the counter-clockwise direction one tick. |
| PRESET_0 | Represents the preset 0 button. |
| PRESET_1 | Represents the preset 1 button. |
| PRESET_2 | Represents the preset 2 button. |
| PRESET_3 | Represents the preset 3 button. |
| PRESET_4 | Represents the preset 4 button. |
| PRESET_5 | Represents the preset 5 button. |
| PRESET_6 | Represents the preset 6 button. |
| PRESET_7 | Represents the preset 7 button. |
| PRESET_8 | Represents the preset 8 button. |
| PRESET_9 | Represents the preset 9 button. |
| CUSTOM_BUTTON | Represents any of touchscreen buttons provided by Mobile Application |
| SEARCH | Represents the 'Search' button. |

Related items:

ButtonCapabilities – [section 5.2.11](#)

PresetBankCapabilities – [section 5.2.13](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

OnButtonEvent – [section 8.2.1](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents](#).

13.1.18 ButtonEventMode

The following events must be supported by HMI for the hardware/soft buttons and must be reported within [UI.GetCapabilities](#) or [Buttons.GetCapabilities](#) RPCs:

- 2) The User has pressed the button (BUTTONDOWN event).
- 3) The User has released the button (BUTTONUP event).

HMI must provide the [OnButtonEvent](#) notification whenever the button is pressed/released.

This enumeration defines UP/DOWN events for the button.

| Element name | Short Description |
|--------------|-------------------|
|--------------|-------------------|

| Element name | Short Description |
|--------------|------------------------------|
| BUTTONDOWN | The button has been pressed |
| BUTTONUP | The button has been released |

Related Items:

[SoftButtonCapabilities – section 5.2.9](#)

[ButtonCapabilities – section 5.2.12](#)

[OnButtonEvent – section 8.2.1](#)

[Table of Contents.](#)

13.1.19 ButtonPressMode

The following press modes might be supported by HMI's hardware/soft buttons and if supported must be reported within [UI.GetCapabilities](#) or [Buttons.GetCapabilities](#) RPCs.

- 1) **Short** – occurs when a button is pressed, then released within two (varies depending on HMI) seconds. The event is considered to occur immediately after the button is released.
- 2) **Long** – occurs when a button is pressed and held for two seconds or more. The event is considered to occur immediately after the two (varies depending on HU) seconds threshold has been crossed, before the button is released

HMI must provide the [onButtonPress](#) notification whenever the Long/Short press occurs.

The following enumeration defines SHORT/LONG modes for the hardware/soft button:

| Element name | Short Description |
|--------------|-------------------------|
| SHORT | Short-time button press |
| LONG | Long-time button press |

Related Items:

[SoftButtonCapabilities – section 5.2.9](#)

[ButtonCapabilities – section 5.2.12](#)

[OnButtonPress – section 8.2.2](#)

[Table of Contents.](#)

13.1.20 LayoutMode

For touchscreen interactions, the mode of how the choices are presented.:.

| Element name | Short Description |
|------------------|---|
| ICON_ONLY | This mode causes the interaction to display the previous set of choices as icons |
| ICON_WITH_SEARCH | This mode causes the interaction to display the previous set of choices as icons along with a search field in the HMI. |
| LIST_ONLY | This mode causes the interaction to display the previous set of choices as a list |
| LIST_WITH_SEARCH | This mode causes the interaction to display the previous set of choices as a list along with a search field in the HMI. |

| | |
|----------|---|
| KEYBOARD | This mode causes the interaction to immediately display a keyboard entry through the HMI. |
|----------|---|

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.21 TouchEvent

For touchscreen interactions, the mode of how the choices are presented:

| Element name | Short Description |
|--------------|-------------------|
| TOUCHSTART | |
| TOUCHMOVE | |
| TOUCHEND | |
| DOUBLETOUCH | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.22 SamplingRate

Describes different sampling options for PerformAudioPassThru

| Element name | Short Description |
|--------------|-------------------|
| 8KHZ | |
| 16KHZ | |
| 22KHZ | |
| 44KHZ | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.23 BitsPerSample

Describes different quality options for PerformAudioPassThru.

| Element name | Short Description |
|--------------|-------------------|
| 8_BIT | |
| 16_BIT | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.24 AudioType

Describes different audio type options for PerformAudioPassThru

| Element name | Short Description |
|--------------|-------------------|
| PCM | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.25 KeyboardLayout

Enumeration listing possible keyboard layouts

| Element name | Short Description |
|--------------|-------------------|
| QWERTY | |
| QWERTZ | |
| AZERTY | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.26 KeyboardEvent

Enumeration listing possible keyboard events.

| Element name | Short Description |
|--------------|-------------------|
| KEYPRESS | |

| | |
|-----------------|--|
| ENTRY_SUBMITTED | |
| ENTRY_CANCELLED | |
| ENTRY_ABORTED | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.27 KeypressMode

Enumeration listing possible keyboard events.

| Element name | Short Description |
|----------------------|--|
| SINGLE_KEYPRESS | Each keypress is individually sent as the user presses the keyboard keys. |
| QUEUE_KEYPRESSES | The keypresses are queued and a string is eventually sent once the user chooses to submit their entry. |
| RESEND_CURRENT_ENTRY | The keypresses are queue and a string is sent each time the user presses a keyboard key; the string contains the entire current entry. |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.28 AmbientLightStatus

Reflects the status of the ambient light sensor.

| Element name | Short Description |
|--------------|-------------------|
| NIGHT | |
| TWILIGHT_1 | |
| TWILIGHT_2 | |
| TWILIGHT_3 | |
| TWILIGHT_4 | |
| DAY | |
| UNKNOWN | |
| INVALID | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.29 ECallConfirmationStatus

Reflects the status of the eCall Notification.

| Element name | Short Description |
|----------------------------|-------------------|
| NORMAL | |
| CALL_IN_PROGRESS | |
| CALL_CANCELLED | |
| CALL_COMPLETED | |
| CALL_UNSUCCESSFUL | |
| ECALL_CONFIGURED_OFF | |
| CALL_COMPLETE_DTMF_TIMEOUT | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.30 VehicleDataNotificationStatus

Reflects the status of a vehicle data notification.

| Element name | Short Description |
|---------------|-------------------|
| NOT_SUPPORTED | |
| NORMAL | |
| ACTIVE | |
| NOT_USED | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.31 EmergencyEventType

Reflects the emergency event status of the vehicle.

| Element name | Short Description |
|---------------|-------------------|
| NO_EVENT | |
| FRONTAL | |
| SIDE | |
| REAR | |
| ROLLOVER | |
| NOT_SUPPORTED | |
| FAULT | |

Related Items:

[SoftButtonCapabilities – section 5.2.9](#)

[ButtonCapabilities – section 5.2.12](#)

[OnButtonPress – section 8.2.2](#)

[Table of Contents.](#)

13.1.32 FuelCutoffStatus

Reflects the status of the RCM fuel cutoff.

| Element name | Short Description |
|------------------|-------------------|
| TERMINATE_FUEL | |
| NORMAL_OPERATION | |
| FAULT | |

Related Items:

[SoftButtonCapabilities – section 5.2.9](#)

[ButtonCapabilities – section 5.2.12](#)

[OnButtonPress – section 8.2.2](#)

[Table of Contents.](#)

13.1.33 PowerModeQualificationStatus

Reflects the status of the current power mode qualification.

| Element name | Short Description |
|----------------------|-------------------|
| POWER_MODE_UNDEFINED | |

| | |
|-----------------------------------|--|
| POWER_MODE_EVALUATION_IN_PROGRESS | |
| NOT_DEFINED | |
| POWER_MODE_OK | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.34 CarModeStatus

Reflects the status of the current car mode.

| Element name | Short Description |
|--------------|-------------------|
| NORMAL | |
| FACTORY | |
| TRANSPORT | |
| CRASH | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.35 PowerModeStatus

Reflects the status of the current power mode.

| Element name | Short Description |
|------------------|-------------------|
| KEY_OUT | |
| KEY_RECENTLY_OUT | |
| KEY_APPROVED_0 | |
| POST_ACCESORY_0 | |
| ACCESORY_1 | |
| POST IGNITION_1 | |
| IGNITION_ON_2 | |

| | |
|-----------|--|
| RUNNING_2 | |
| CRANK_3 | |

Related Items:

SoftButtonCapabilities – [section 5.2.9](#)

ButtonCapabilities – [section 5.2.12](#)

OnButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.1.36 ComponentVolumeStatus

The enumeration provides the cases for the component volume status (e.g., fuel level, tire pressure).

| Element name | Short Description |
|---------------|--|
| UNKNOWN | The data is unknown. |
| NORMAL | The volume is normal. |
| LOW | The volume is low. |
| FAULT | The module/sensor is currently faulted. |
| ALERT | The component's volume is in critical level. |
| NOT_SUPPORTED | The data is not supported. |

Related items:

SingleTireStatus – [section 5.2.19](#)

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents.](#)

13.1.37 PRNDL

The following enumeration describes the possible positions of vehicle's change-speed lever.

| Element name | Short Description |
|--------------|--|
| PARK | Parking |
| REVERSE | Reverse gear |
| NEUTRAL | No gear |
| DRIVE | Drive Sport mode |
| SPORT | Sport mode |
| LOWGEAR | 1st gear hold |
| FIRST | The change-speed lever is in the first position |
| SECOND | The change-speed lever is in the second position |
| THIRD | The change-speed lever is in the third position |
| FOURTH | The change-speed lever is in the fifth position |
| FIFTH | The change-speed lever is in the sixth position |

| Element name | Short Description |
|--------------|---|
| SIXTH | The change-speed lever is in the seventh position |
| SEVENTH | |
| EIGHTH | |
| FAULT | |

Related items:

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.1.38 WarningLightStatus

The enumeration reflects the status of a cluster instrument warning light.

| Element name | Short Description |
|--------------|-------------------------------|
| OFF | The warning light is off |
| ON | The warning light is on |
| FLASH | The warning light is flashing |
| NOT_USED | |

Related items:

TireStatus – [section 5.2.18](#)

[Table of Contents](#).

13.1.39 VehicleDataEventStatus

The following enumeration reflects the status of a vehicle data event (e.g. a seat belt event status).

| Element name | Short Description |
|---------------|---|
| NO_EVENT | The system does not have the adequate information to send valid 'YES' or 'NO' states. |
| NO | The requested event is in 'NO' state. |
| YES | The requested event is in 'YES' state. |
| NOT_SUPPORTED | The requested data is not supported |
| FAULT | The module/sensor is currently faulted. |

Related items:

BeltStatus – [section 5.2.20](#)

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.1.40 IgnitionStableStatus

This enumeration describes the states of the ignition switch.

| Element name | Short Description |
|----------------------------|--|
| IGNITION_SWITCH_NOT_STABLE | Ignition switch is not stable. |
| IGNITION_SWITCH_STABLE | Ignition switch is stable. |
| MISSING_FROM_TRANSMITTER | Either the data is not accessible or the sensor is broken. |

Related items:

BodyInformation – [section 5.2.21](#)

[Table of Contents](#).

13.1.41 IgnitionStatus

This enumeration reflects the status of ignition.

| Element name | Short Description |
|--------------|--|
| UNKNOWN | The information is not acceptable. The sensor cannot provide accurate data at this time. |
| OFF | The ignition is off. |
| ACCESSORY | The accessories are active (power windows, audio, display, etc.). |
| RUN | Ignition is active. |
| START | Starter is switched. |
| INVALID | The data is provided, but there is some sort of fault or problem. |

Related items:

BodyInformation – [section 5.2.21](#)

[Table of Contents](#).

13.1.42 DeviceLevelStatus

The enumeration describes the possible status of the battery if reported.

| Element name | Short Description |
|------------------|--|
| ZERO_LEVEL_BARS | The battery is low |
| ONE_LEVEL_BARS | The battery is in the first level bar. |
| TWO_LEVEL_BARS | The battery is in the second level bar |
| THREE_LEVEL_BARS | The battery is in the fourth level bar |
| FOUR_LEVEL_BARS | The battery is in the fifth level bar |
| NOT_PROVIDED | The data is not provided |

Related items:

DeviceStatus – [section 5.2.22](#)

[Table of Contents](#).

13.1.43 Primary AudioSource

Reflects the current primary audio source (if selected).

| Element name | Short Description |
|-----------------------|--|
| NO_SOURCE_SELECTED | No source for playing audio is selected. |
| USB | The audio is played from USB. |
| USB2 | The audio is played from USB2 |
| BLUETOOTH_STEREO_BTST | The audio in stereo format is played from Bluetooth. |
| LINE_IN | The audio is played from the line input. |
| IPOD | The audio is played from the IPod. |
| MOBILE_APP | The audio is played from the mobile application. |

Related items:

DeviceStatus – [section 5.2.22](#)

[Table of Contents](#).

13.1.44 VehicleDataStatus

The following enumeration reflects the status of a binary vehicle data item.

| Element name | Short Description |
|----------------|--|
| NO_DATA_EXISTS | The information is not acceptable. The sensor cannot provide accurate data at this time. |
| OFF | The vehicle item is in 'Off' state |
| ON | The vehicle item is in 'On' state |

Related items:

MyKey – [section 5.2.24](#)

[Table of Contents](#).

13.1.45 WiperStatus

This enumeration reflects the status of the wipers.

| Element name | Short Description |
|--------------|---|
| OFF | The wipers are off. |
| AUTO_OFF | The wipers are automatically off after detecting the wipers do not need to be engaged (rain stopped, etc.). |

| Element name | Short Description |
|----------------|---|
| OFF_MOVING | Means that though set to off, somehow the wipers have been engaged (physically moved enough to engage a wiping motion). |
| MAN_INT_OFF | The wipers are manually off after having been working. |
| MAN_INT_ON | The wipers are manually on. |
| MAN_LOW | The wipers are manually set to low speed. |
| MAN_HIGH | The wipers are manually set to high speed. |
| MAN_FLICK | The wipers are manually set for doing a flick. |
| WASH | The wipers are set to use the water from vehicle washer bottle for cleaning the windscreen. |
| AUTO_LOW | The wipers are automatically set to low speed. |
| AUTO_HIGH | The wipers are automatically set to high speed. |
| COURTESYWIPE | This is for when a user has just initiated a WASH and several seconds later a secondary wipe is automatically initiated to clear remaining fluid. |
| AUTO_ADJUST | This is set as the user moves between possible automatic wiper speeds. |
| STALLED | The wiper is stalled to its place. There may be an obstruction. |
| NO_DATA_EXISTS | The sensor / module cannot provide any information for wiper. |

Related items:

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.1.46 CompassDirection

The potential compass directions that IVI component may provide are listed below.

| Element name | Short Description |
|--------------|---|
| NORTH | Represents the North compass direction |
| NORTHWEST | Represents the North-West compass direction |
| WEST | Represents the West compass direction |
| SOUTHWEST | Represents the South-West compass direction |
| SOUTH | Represents the South compass direction |
| SOUTHEAST | Represents the South-East compass direction |
| EAST | Represents the East compass direction |
| NORTHEAST | Represents the North-East compass direction |

Related items:

GPSData – [section 5.2.17](#)

[Table of Contents](#).

13.1.47 Dimension

The enumeration defines the supported dimensions of GPS.

| Element name | Short Description |
|--------------|-------------------------------------|
| NO_FIX | No GPS at all |
| 2D | Longitude and latitude |
| 3D | Longitude and latitude and altitude |

Related items:

GPSData – [section 5.2.17](#)

[Table of Contents](#).

13.1.48 VehicleDataresultCode

Enumeration that describes possible result codes of a vehicle data entry request.

| Element name | Short Description |
|----------------------------|---|
| SUCCESS | The requested data is accessible. |
| TRUNCATED_DATA | The data is truncated: not all of the requested information is available. |
| DISALLOWED | The request is not authorized in local policies. |
| USER_DISALLOWED | The request is included in a functional group explicitly blocked by the User. |
| INVALID_ID | One of the provided IDs is not valid. |
| VEHICLE_DATA_NOT_AVAILABLE | The requested data is not available. |
| DATA_ALREADY_SUBSCRIBED | The requested data has been subscribed already. |
| DATA_NOT_SUBSCRIBED | The data is not subscribed. |
| IGNORED | The request is ignored because the intended result is already in effect. |

Related items:

DIDResult – [section 5.2.16](#)

ReadDID – [section 11.1.3](#)

[Table of Contents](#).

13.1.49 TBTState

The enumeration describes the possible needs of HMI turn-by-turn.

| Element name | Short Description |
|----------------------|--|
| ROUTE_UPDATE_REQUEST | Request from HMI for updating the route. |
| ROUTE_ACCEPTED | Confirmation from HMI about accepting the route. |
| ROUTE_REFUSED | Information from HMI about the route refusal. |
| ROUTE_CANCELLED | Information from HMI about cancelling the route. |
| ETA_REQUEST | Request from HMI for Estimated time of arrival. |

| Element name | Short Description |
|------------------------------|--|
| NEXT_TURN_REQUEST | Request from HMI for the information of the next turn. |
| ROUTE_STATUS_REQUEST | Request from HMI for the route status. |
| ROUTE_SUMMARY_REQUEST | Request from HMI for the route summary. |
| TRIP_STATUS_REQUEST | Request from HMI for the information about trip status. |
| ROUTE_UPDATE_REQUEST_TIMEOUT | Request from HMI for the timeout for waiting for the route updating. |

Related items:

OnTBTClientState – [section 12.2.1](#)

[Table of Contents](#).

13.2 Structures

13.2.1 HMIApplication

The structure defines the information about the application: its name, ID, the device concerned and etc. described in the table below.

| Param Name | Type | Mandatory | Additional | Description |
|---------------------------|-----------------|-----------|-----------------|---|
| appName | String | true | Maxlength = 100 | The mobile application name. |
| ngnMediaScreenAppName | String | false | Maxlength = 100 | Provides an abbreviated version of the application name (may be displayed on the NGN media screen). If not provided, the appName should be used instead (and may be truncated if too long) |
| icon | String | false | – | Path to the application icon stored on HU. |
| deviceName | String | true | – | The name of the device where the identified application is running on. |
| appID | Integer | true | – | The application ID that remains unique during the ignition cycle. This ID will be sent by SDL further and must be provided by HMI within all the RPCs related to this application. |
| hmiDisplayLanguageDesired | Common.Language | true | – | The language that the application intends to use. See Language . |

| Param Name | Type | Mandatory | Additional | Description |
|--------------------|-------------------|-----------|--|--|
| isMediaApplication | Boolean | true | - | <p>Indicates whether the application is a media or a non-media one.</p> <p>Only media applications are allowed by SDL to stream audio to HU that is audible outside of the BT media source.</p> |
| appType | Common.AppHMIType | false | array = true Minsize = 1 maxsize = 100 | <p>The HMI may use this information for determining what functionality should be available for the application (e.g. NAVIGATION type of application will require displaying the information and not playing the audio).</p> <p>See AppHMIType.</p> |

Related items:

UpdateApplist – [section 6.1.2](#)

AllowApp – [section 6.1.6](#)

OnAppRegistered – [section 6.2.8](#)

Language – [section 5.1.14](#)

AppHMIType – [section 5.1.2](#)

[Table of Contents](#).

13.2.2 DeviceInfo

The structure is used for providing the information about the device connected to SDL:

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|--|
| name | String | true | The name of the device. |
| id | Integer | true | The ID of the device. It remains unique during the ignition cycle. |

Related items:

UpdateDeviceList – [section 6.1.1](#)

AllowDeviceToConnect – [section 6.1.4](#)

OnDeviceChosen – [section 6.2.3](#)

OnFindApplications – [section 6.2.4](#)

[Table of Contents](#).

13.2.3 MenuParams

The structure is used for setting the position of the command/sub-menu and the name of the sub-menu to be added within the requests of AddCommand and AddSubMenu.

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| | | | | |
|----------|---------|-------|---------------------------------------|---|
| parentID | Integer | false | minvalue = 0 maxvalue = 2000000000 | <ul style="list-style-type: none"> - This value will NOT be provided for AddSubMenu request. - Unique ID of the sub menu, the command must be added to. - If not provided, the command must be added to the top-level application menu. |
| position | Integer | false | minvalue = 0 maxvalue = 1000 | <p>This value is the position within the elements of top-level application menu where the command/sub-menu must be added to:</p> <ul style="list-style-type: none"> - If 0, the item must be inserted to the first position. - If 1, the item must be inserted to the second position. - Etc. <p>If the value is greater than or equal to the number of elements of the top-level application menu, the sub menu or a command must be appended to the end of the list.</p> <p>If omitted the entry must be added to the end of the list.</p> |
| menuName | String | true | maxlength = 500 | The text that must be shown as a name of a command/sub -menu. |

Related items:

AddCommand – for UI, [section 7.1.6](#)

AddSubMenu – for UI, [section 7.1.8](#)

[Table of Contents](#).

13.2.4 Choice

A choice is an option given to the user which can be selected either by menu or through voice recognition system during an application initiated interaction. For example, the application may request for the user's choice among several suggested ones: 'Yes', 'No', 'Skip'.

The given structure defines the UI representation of the choice.

The choices for VR interaction are added by the AddCommand ([section 9.1.3](#)), where cmdID param is provided for the choice identifier.

| Param Name | Type | Mandatory | Additional | Description |
|------------|---------|-----------|----------------------------------|---|
| choiceID | Integer | true | minvalue = 0 maxvalue = 65535 | The unique within the concerned application identifier for this choice |
| menuName | String | false | Maxlength = 500 | The text to be displayed in the onscreen menu indicating the name of the choice (e.g. 'Yes'). |

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|--------------|------------------|-------------------|---|
| image | Common.Image | false | - | Image that must appear in the menu, representing this choice. See Image |
| secondaryText | String | false | Maxlength = 500 | Optional secondary text to display; e.g. address of POI in a search result entr |
| tertiaryText | String | false | Maxlength = 500 | Optional tertiary text to display; e.g. distance to POI for a search result entry |
| secondaryImage | Common.Image | false | - | Optional secondary image struct for choice |

Related items:

PerformInteraction – [section 7.1.10](#)

AddCommand – [section 9.1.3](#)

Image – [section 5.2.11](#)

[Table of Contents](#).

13.2.5 TimeFormat

The structure describes the format of time value to be set upon request from SDL with SetMediaClockTimer.

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|-------------|------------------|-------------------------------|--|
| hours | Integer | true | minvalue = 0 maxvalue = 59 | The hour of the media clock. Some display types only support a max of 19 hours. If out of range, the request must be rejected. |
| minutes | Integer | true | minvalue = 0 maxvalue = 59 | The minute. |
| seconds | Integer | true | minvalue = 0 maxvalue = 59 | The second. |

Related items:

SetMediaClockTimer – [section 7.1.11](#)

[Table of Contents](#).

13.2.6 VrHelpItem

The structure provides the information about VR Help Menu item that must be displayed on UI representing the command of VR to the User.

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|-------------|------------------|-------------------|---|
| text | String | True | maxlength = 500 | The text that must be displayed as a name for VR Help item. |

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|--------------|------------------|--------------------------------|---|
| image | Common.Image | False | - | Image that must be displayed by HU to represent the VR Help item. See Image |
| position | Integer | false | minvalue = 1 maxvalue = 100 | This value is the position within the elements of VR Help menu where the item must be added to: |

Related items:

Image – [section 5.2.11](#)

ShowVrHelp – [section 7.1.12](#)

PerformInteraction – [section 7.1.10](#)

Image – [section 5.2.11](#)

[Table of Contents](#).

13.2.7 DisplayCapabilities

The structure contains the information about the display capabilities.

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|-------------------------|------------------|--|--|
| displayType | Common.DisplayType | true | - | The type of the display that is installed on HU. See DisplayType . |
| textFields | Common.TextFieldName | true | Array = true minsize = 0 maxsize = 100 | A set of all fields that support displaying text data. If there are no textfields supported HMI must send the empty array. See TextFieldName . |
| mediaClockFormats | Common.MediaClockFormat | true | Array = true minsize = 1 maxsize = 100 | A set of all supported formats of the media clock. See MediaClockFormat . |
| imageCapabilities | Common.ImageType | false | Array = true minsize = 0 maxsize = 2 | The array of supported image types (static and/or dynamic). The empty array should be returned if the platform does not support displaying images. See ImageType . |
| graphicSupported | Boolean | true | - | The display's persistent screen supports referencing a static or dynamic image. |

Related items:

DisplayType – [section 5.1.7](#)

MediaClockFormat – [section 5.1.8](#)

TextFieldName – [section 5.1.15](#)

ImageType – [section 5.1.16](#)

GetCapabilities – for UI, [section 7.1.2](#)

[Table of Contents](#).

13.2.7 TouchEventCapabilities

The structure contains the information about the display capabilities.

| Param Name | Type | Mandatory | Description |
|----------------------|---------|-----------|-------------|
| pressAvailable | Boolean | true | |
| multiTouchAvailable | Boolean | true | |
| doublePressAvailable | Boolean | true | |

Related items:

DisplayType – [section 5.1.7](#)

MediaClockFormat – [section 5.1.8](#)

TextFieldName – [section 5.1.15](#)

ImageType – [section 5.1.16](#)

GetCapabilities – for UI, [section 7.1.2](#)

[Table of Contents](#).

13.2.8 ImageResolution

The structure contains the information about the display capabilities.

| Param Name | Type | Mandatory | Additional | Description |
|------------------|---------|-----------|----------------------------------|------------------------------|
| resolutionWidth | Integer | true | minvalue = 1 maxvalue = 10000 | The image resolution width. |
| resolutionHeight | Integer | true | minvalue = 1 maxvalue = 10000 | The image resolution height. |

Related items:

DisplayType – [section 5.1.7](#)

MediaClockFormat – [section 5.1.8](#)

TextFieldName – [section 5.1.15](#)

ImageType – [section 5.1.16](#)

GetCapabilities – for UI, [section 7.1.2](#)

[Table of Contents](#).

13.2.9 ScreenParams

The structure contains the information about the display capabilities.

| Param Name | Type | Mandatory | Description |
|---------------------|-------------------------------|------------------|--|
| resolution | Common.ImageResolution | true | The resolution of the prescribed screen area. |
| touchEventAvailable | Common.TouchEventCapabilities | true | Types of screen touch events available in screen area. |

Related items:

DisplayType – [section 5.1.7](#)

MediaClockFormat – [section 5.1.8](#)

TextFieldName – [section 5.1.15](#)

ImageType – [section 5.1.16](#)

GetCapabilities – for UI, [section 7.1.2](#)

[Table of Contents](#).

13.2.10 ImageField

The structure contains the information about the display capabilities.

| Param Name | Type | Mandatory | Additional | Description |
|---------------------|------------------------|------------------|--|---|
| name | Common.ImageFieldName | true | - | The name that identifies the field. See ImageFieldName. |
| imageTypesSupported | Common.FileType | false | Array = true minsize = 1 maxsize = 100 | The image types that are supported in this field. See FileType. |
| imageResolution | Common.ImageResolution | false | - | The image resolution of this field |

Related items:

DisplayType – [section 5.1.7](#)

MediaClockFormat – [section 5.1.8](#)

TextFieldName – [section 5.1.15](#)

ImageType – [section 5.1.16](#)

GetCapabilities – for UI, [section 7.1.2](#)

[Table of Contents](#).

13.2.11 TextFieldStruct

The structure is used in requests for HMI for displaying the text message in the appropriate text area of the display.

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|----------------------|------------------|-------------------|--|
| fieldName | Common.TextFiledName | true | - | The name of the field where the text must be displayed in. |
| fieldText | String | true | MaxLength = 500 | The text to be displayed. |

Related items:

TextFieldName – [section 5.1.14](#)

Turn – [section 5.2.25](#)

Alert – [section 7.1.4](#)

ShowNotification – [section 7.2.5](#)

Show – [section 7.1.5](#)

PerformInteraction – [section 7.1.10](#)

ScrollableMessage – [section 7.1.17](#)

PerformAudioPassThru – [section 7.1.18](#)

ShowConstantTBT – [section 12.1.3](#)

[Table of Contents.](#)

13.2.12 SoftButtonCapabilities

The structure contains the information about the soft buttons capabilities:

| Param Name | Type | Mandatory | Description |
|---------------------|-------------|------------------|--|
| shortPressAvailable | Boolean | true | Must be - 'true' if soft buttons support a short press - 'false' if not. See ButtonPressMode for more information. |
| longPressAvailable | Boolean | true | Must be - 'true' if soft buttons support a LONG press - 'false' if not. See ButtonPressMode for more information. |
| upDownAvailable | Boolean | true | Must be - 'true' if soft buttons support "button down" and "button up". - 'false' if not. See ButtonEventMode for more information. |
| imageSupported | Boolean | true | Must be - 'true' if soft buttons support referencing image - 'false' if not. |

Related items:

ButtonEventMode – [section 5.1.18](#)

ButtonPressMode – [section 5.1.19](#)

GetCapabilities – for UI, [section 7.1.2](#)

onButtonEvent – [section 8.2.1](#)

onButtonPress – [section 8.2.2](#)

[Table of Contents.](#)

13.2.13 SoftButton

The structure describes all the values sent to HMI for displaying a soft button on UI.

| Param Name | Type | Mandatory | Additional | Description |
|---------------|-----------------------|-----------|----------------------------------|--|
| type | Common.SoftButtonType | true | - | Defines the type of the soft button: whether it must be displayed with a text, image or both text and image. See SoftButtonType . |
| text | String | false | maxlength = 500 | If the value is present, UI must display the text on the corresponding soft button. This value is provided if the soft button type is defined as TEXT. |
| image | Common.Image | false | - | If the value is present, UI must display the image on the corresponding soft button. This parameter is provided if the soft button type is defined as IMAGE. See Image . |
| isHighlighted | Boolean | false | - | If ‘true’, the soft button must be highlighted. If ‘false’, must be not. If omitted, the button must NOT be highlighted. |
| softButtonID | Integer | true | minvalue = 0 maxvalue = 65535 | The unique ID provided for identifying the soft button within the application. This value must be returned via OnButtonPress / OnButtonEvent as a customButtonId parameter. |
| systemAction | Common.SystemAction | false | - | Parameter that indicates whether clicking the SoftButton must call a specific system action. See SystemAction . If omitted the default system action should occur. |

Related items:

SoftButtonType – [section 5.1.11](#)

SystemAction – [section 5.1.12](#)

Image – [section 5.2.11](#)

Alert – [section 7.1.4](#)

Show – [section 7.1.5](#)

ScrollableMessage – [section 7.1.17](#)

onButtonPress – [section 8.2.2](#)

onButtonEvent – [section 8.2.1](#)

AlertManeuver – [section 12.1.2](#)

ShowConstantTBT – [section 12.1.3](#)

UpdateTurnList – [section 12.1.4](#)

[Table of Contents](#).

13.2.14 Image

The structure describes the type of and the path to the image to be displayed on UI upon corresponding request of SDL.

| Param Name | Type | Mandatory | Additional | Description |
|------------|------------------|-----------|-------------------|---|
| value | String | true | maxlength = 65535 | - The path to the dynamic image stored on HU - Or the static binary image itself |
| imageType | Common.ImageType | true | - | Describes, whether it is a static or dynamic image. |

Related items:

ImageType – [section 5.1.15](#)
Choice – [section 5.2.4](#)
VrHelpItem – [section 5.2.6](#)
SoftButton – [section 5.2.10](#)
Turn – [section 5.2.25](#)
Show – [section 7.1.5](#)
AddCommand – [section 7.1.6](#)
SetAppIcon – [section 7.1.15](#)
ShowNotification – [section 7.2.5](#)
ShowConstantTBT – [section 12.1.3](#)

[Table of Contents.](#)

13.2.15 ButtonCapabilities

The structure describes the hardware buttons capabilities.

Upon the request HMI must provide the list of the following information:

- The names of all existing/supported hardware buttons as described in the section 5.1.2.
- The availability of LONG/SHORT press for each existing/supported hardware button correspondingly (as described in the section 5.1.4).
- The availability of UP/DOWN events for each existing/supported hardware button correspondingly (as described in the section 5.1.3).

| Param Name | Type | Mandatory | Description |
|---------------------|-------------------|-----------|--|
| name | Common.ButtonName | true | The name of supported/existing hardware buttons. See ButtonName |
| shortPressAvailable | Boolean | true | Must be 'true' if the button supports SHORT press mode. See ButtonPressMode |
| longPressAvailable | Boolean | true | The button supports LONG press mode. See ButtonPressMode |
| upDownAvailable | Boolean | true | The button supports "button down" and "button up". See ButtonEventMode |

Related items:

ButtonName – [section 5.1.16](#)
ButtonEventMode – [section 5.1.17](#)
ButtonPressMode – [section 5.1.18](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

[Table of Contents](#).

13.2.16 PresetBankCapabilities

Some HMI displays can duplicate hardware buttons with on-screen buttons. Such on-screen buttons have the same capabilities and functionality (e.g. play, pause, etc.) as hardware buttons (e.g. ‘short press’, ‘long press’).

HMI must provide the information about the availability of on-screen presets for supported hardware buttons.

| Param Name | Type | Mandatory | Description |
|--------------------------|---------|-----------|--|
| onScreenPresetsAvailable | Boolean | true | Must be ‘true’ if on-screen custom presets are available for the hardware button of the provided <code>ButtonName</code> . |

Related items:

`ButtonName` – [section 5.1.16](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

[Table of Contents](#).

13.2.17 AudioPassThruCapabilities

Describes different audio type configurations for `PerformAudioPassThru`.

e.g. 8kHz,8-bit,PCM

| Param Name | Type | Mandatory | Description |
|----------------------------|-----------------------------------|-----------|-------------|
| <code>samplingRate</code> | <code>Common.SamplingRate</code> | true | |
| <code>bitsPerSample</code> | <code>Common.BitsPerSample</code> | true | |
| <code>audioType</code> | <code>Common.AudioType</code> | true | |

Related items:

`ButtonName` – [section 5.1.16](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

[Table of Contents](#).

13.2.18 TouchLists

Describes different audio type configurations for `PerformAudioPassThru`.

e.g. 8kHz,8-bit,PCM

| Param Name | Type | Mandatory | Additional | Description |
|------------|------|-----------|------------|-------------|
|------------|------|-----------|------------|-------------|

| | | | | |
|----------------|---------|------|---|---|
| touches | Integer | true | Array = true minvalue = 0 maxvalue = 9 minsize = 1 maxsize = 10 | List of fingers currently on screen |
| changedTouches | Integer | true | Array = true minvalue = 0 maxvalue = 9 minsize = 1 maxsize = 10 | List of fingers involved in the current event. For example, in a touchend event, this will be the finger that was removed |

Related items:

ButtonName – [section 5.1.16](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

[Table of Contents](#).

13.2.19 Coordinate

Describes different audio type configurations for PerformAudioPassThru.
e.g. 8kHz,8-bit,PCM

| Param Name | Type | Mandatory | Description |
|------------|---------|-----------|-------------|
| xCoord | Integer | true | |
| yCoord | Integer | true | |

Related items:

ButtonName – [section 5.1.16](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

[Table of Contents](#).

13.2.20 TouchArea

Describes different audio type configurations for PerformAudioPassThru.
e.g. 8kHz,8-bit,PCM

| Param Name | Type | Mandatory | Additional | Description |
|---------------|-------------------|-----------|--------------------------------|-------------|
| rotationAngle | Float | false | minvalue = 0 maxvalue = 360 | |
| radiusCoord | Common.Coordinate | true | - | |

Related items:

ButtonName – [section 5.1.16](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

[Table of Contents](#).

13.2.21 TouchEventInfo

Describes different audio type configurations for PerformAudioPassThru.
e.g. 8kHz,8-bit,PCM

| Param Name | Type | Mandatory | Additional | Description |
|------------|-------------------|-----------|------------------------------|--|
| id | Integer | false | minvalue = 0 maxvalue = 9 | Finger / stylus ID |
| point | Common.Coordinate | true | | XY coordinate of where on screen the event happened |
| area | Common.TouchArea | false | | Describe the ellipse approximating touch input shape |

Related items:

ButtonName – [section 5.1.16](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

[Table of Contents](#).

13.2.22 KeyboardProperties

Configuration of on-screen keyboard (if available).

| Param Name | Type | Mandatory | Additional | Description |
|----------------------|-----------------------|-----------|---|--|
| language | Common.Language | false | | The keyboard language. |
| keyboardLayout | Common.KeyboardLayout | false | | Desired keyboard layout. |
| sendDynamicEntry | Boolean | false | | In this mode, all keypresses will be sent as they occur. If disabled, entire string of text will be returned only once submitted by user. If omitted, this value will be set to FALSE. |
| limitedCharacterList | String | false | Array = true maxlength = 1 minsize = 1 maxsize = 100 | Array of keyboard characters to enable. All omitted characters will be greyed out (disabled) on the keyboard. If omitted, the entire keyboard will be enabled. |
| autoCompleteText | String | false | maxlength = 1000 | Allows an app to prepopulate the text field with a suggested or completed entry as the user types |

Related items:

ButtonName – [section 5.1.16](#)

GetCapabilities – for Buttons, [section 8.1.1](#)

[Table of Contents.](#)

13.2.23 TTSCChunk

The following structure defines a TTS chunk that consists of the text/phonemes to be spoken by the TTS module of HU:

| Param Name | Type | Mandatory | Maxlength | Description |
|------------|---------------------------|-----------|-----------|--|
| text | String | true | 500 | The text or phonemes to speak |
| type | Common.SpeechCapabilities | true | - | Describes, whether it is text or a specific phoneme set. See SpeechCapabilities. |

Related items:

Speak – [section 10.1.3](#)

SetGlobalProperties – for TTS, [section 10.1.7](#)

[Table of Contents.](#)

13.2.23 VehicleType

The following structure describes the information about vehicle type: its manufacturer, model, model year and trim.

| Param Name | Type | Mandatory | Additional | Description |
|------------|--------|-----------|-----------------|---------------------------------------|
| make | String | false | maxlength = 500 | Make of the vehicle (e.g. Ford) |
| model | String | false | maxlength = 500 | Model of the vehicle (e.g. Fiesta) |
| modelYear | String | false | maxlength = 500 | Model Year of the vehicle (e.g. 2013) |
| trim | String | false | maxlength = 500 | Trim of the vehicle (e.g. SE) |

Related items:

GetVehicleType – [section 11.1.2](#)

[Table of Contents.](#)

13.2.24 DIDResult

The structure describes the information returned for the ReadDID request.

| Param Name | Type | Mandatory | Additional | Description |
|-------------|------------------------------|-----------|----------------------------------|--|
| resultCode | Common.VehicleDataResultCode | true | - | Individual DID result code (see VehicleDataResultCode) |
| didLocation | Integer | true | minvalue = 0 maxvalue = 65535 | The address of DID location from the ReadDID request. |

| Param Name | Type | Mandatory | Additional | Description |
|-------------------|-------------|------------------|-------------------|---|
| data | String | false | maxlength = 5000 | The DID data which is the hex byte string of however many bytes are stored at that location |

Related items:

VehicleDataresultCode – [section 5.1.31](#)

ReadDID – [section 11.1.3](#)

[Table of Contents](#).

13.2.25 GPSData

The following structure describes the GPS data, which IVI module is expected to return for the request of SDL.

| Param Name | Type | Mandatory | Maxvalue | Description |
|-------------------|-------------------------|------------------|------------------------------------|--|
| longitudeDegrees | Float | true | minvalue = -180 maxvalue = 180 | Position longitude in degrees |
| latitudeDegrees | Float | true | minvalue = -90 maxvalue = 90 | Position latitude in degrees |
| utcYear | Integer | false | minvalue = 2010 maxvalue = 2100 | The current UTC year. |
| utcMonth | Integer | false | minvalue = 1 maxvalue = 12 | The current UTC month. |
| utcDay | Integer | false | minvalue = 1 maxvalue = 31 | The current UTC day |
| utcHours | Integer | false | minvalue = 0 maxvalue = 23 | The current UTC hour. |
| utcMinutes | Integer | false | minvalue = 0 maxvalue = 59 | The current UTC minute |
| utcSeconds | Integer | false | minvalue = 0 maxvalue = 59 | The current UTC second |
| compassDirection | Common.CompassDirection | false | - | The information about compass direction (see section 5.1.25). |
| pdop | Float | false | minvalue = 0 maxvalue = 10 | Positional Dilution Of Precision |
| hdop | Float | false | minvalue = 0 maxvalue = 10 | Horizontal Dilution Of Precision |
| vdop | Float | false | minvalue = 0 maxvalue = 10 | Vertical Dilution Of Precision |
| actual | Boolean | false | - | The information about actuality of GPS data should be returned: - ‘true’, if actual - ‘false’, if inferred |
| satellites | Integer | false | minvalue = 0 maxvalue = 31 | Number of satellites in view |

| Param Name | Type | Mandatory | Maxvalue | Description |
|-------------------|------------------|------------------|---------------------------------------|--|
| dimension | Common.Dimension | false | - | The supported GPS dimension (see section 12.2.26) |
| altitude | Float | false | minvalue = -10000 maxvalue = 10000 | Altitude in meters |
| heading | Float | false | minvalue = 0 maxvalue = 359.99 | '0' is considered as heading North. Resolution accepted by SDL is 0.01. |
| speed | Float | false | minvalue = 0 maxvalue = 500 | The speed in KPH |

Related items:

CompassDirection – [section 5.1.29](#)

Dimension – [section 5.1.30](#)

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.2.26 TireStatus

The following structure describes the status and pressure of vehicle tires.

| Param Name | Type | Mandatory | Description |
|-------------------|---------------------------|------------------|--|
| pressureTelltaile | Common.WarningLightStatus | false | Status of the Tire Pressure Telltale. See WarningLightStatus. |
| leftFront | Common.SingleTireStatus | false | The status of the left front tire. |
| rightFront | Common.SingleTireStatus | false | The status of the right front tire. |
| leftRear | Common.SingleTireStatus | false | The status of the left rear tire. |
| rightRear | Common.SingleTireStatus | false | The status of the right rear tire. |
| innerLeftRear | Common.SingleTireStatus | false | The status of the inner left rear. |
| innerRightRear | Common.SingleTireStatus | false | The status of the inner right rear. |

Related items:

WarningLightStatus – [section 5.1.21](#)

SingleTireStatus – [section 5.2.19](#)

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.2.27 SingleTireStatus

The structure provides the status of component volume.

| Param Name | Type | Mandatory | Description |
|-------------------|-------------|------------------|--------------------|
| | | | |

| Param Name | Type | Mandatory | Description |
|------------|------------------------------|-----------|---------------------------|
| status | Common.ComponentVolumeStatus | true | See ComponentVolumeStatus |

Related items:

ComponentVolumeStatus – [section 5.1.19](#)

[Table of Contents](#).

13.2.28 BeltStatus

The structure references the signals from the sensors that detect whether the seat belt is deployed or buckled.

| Param Name | Type | Mandatory | Description |
|---------------------------|-------------------------------|-----------|---|
| driverBeltDeployed | Common.VehicleDataEventStatus | false | The driver seat belt is deployed. |
| passengerBeltDeployed | Common.VehicleDataEventStatus | false | The passenger seat belt is deployed. |
| passengerBuckleBelted | Common.VehicleDataEventStatus | false | The passenger seat belt is buckled. |
| driverBuckleBelted | Common.VehicleDataEventStatus | false | The driver seat belt is buckled. |
| leftRow2BuckleBelted | Common.VehicleDataEventStatus | false | The left seat belt of the 2 nd row is buckled. |
| passengerChildDetected | Common.VehicleDataEventStatus | false | The child passenger is detected. |
| rightRow2BuckleBelted | Common.VehicleDataEventStatus | false | The right seat belt of the 2 nd row is buckled. |
| middleRow2BuckleBelted | Common.VehicleDataEventStatus | false | The middle seat belt of the 2 nd row is buckled. |
| middleRow3BuckleBelted | Common.VehicleDataEventStatus | false | The middle seat belt of the 3 rd row is buckled. |
| leftRow3BuckleBelted | Common.VehicleDataEventStatus | false | The left seat belt of the 3 rd row is buckled. |
| rightRow3BuckleBelted | Common.VehicleDataEventStatus | false | The right seat belt of the 3 rd row is buckled. |
| leftRearInflatableBelted | Common.VehicleDataEventStatus | false | The left rear inflatable is belted. |
| rightRearInflatableBelted | Common.VehicleDataEventStatus | false | The right rear inflatable is belted. |
| middleRow1BeltDeployed | Common.VehicleDataEventStatus | false | The seat belt of the middle row is deployed. |
| middleRow1BuckleBelted | Common.VehicleDataEventStatus | false | The seat belt of the middle row is buckled. |

Related items:

VehicleDataEventStatus – [section 5.1.22](#)

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.2.29 BodyInformation

The structure defines the information about the park brake and ignition.

| Param Name | Type | Mandatory | Description |
|----------------------|-----------------------------|-----------|--|
| parkBrakeActive | Boolean | true | The information about the park brake: - 'true', if active - 'false' if not. |
| ignitionStableStatus | Common.IgnitionStableStatus | true | The information about stability of the ignition switch. See IgnitionStableStatus. |
| ignitionStatus | Common.IgnitionStatus | true | The information about ignition status. See IgnitionStatus. |
| driverDoorAjar | Boolean | false | |
| passengerDoorAjar | Boolean | false | |
| rearLeftDoorAjar | Boolean | false | |
| rearRightDoorAjar | Boolean | false | |

Related items:

IgnitionStableStatus – [section 5.1.23](#)

IgnitionStatus – [section 5.1.24](#)

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.2.30 DeviceStatus

The structure reflects the information about the Head Unit status.

| Param Name | Type | Mandatory | Description |
|------------------------|--------------------------|-----------|--|
| voiceRecOn | Boolean | false | Must be 'true' if the voice recording is on. |
| btIconOn | Boolean | false | Must be 'true' if Bluetooth icon is displayed. |
| callActive | Boolean | false | Must be 'true' if there is an active call. |
| phoneRoaming | Boolean | false | Must be 'true' if there is a phone roaming. |
| textMsgAvailable | Boolean | false | Must be 'true' if the text message is available. |
| battLevelStatus | Common.DeviceLevelStatus | false | Device battery level status. See DeviceLevelStatus. |
| stereoAudioOutputMuted | Boolean | false | Must be 'true' if stereo audio output is muted. |

| Param Name | Type | Mandatory | Description |
|----------------------|---------------------------|------------------|--|
| monoAudioOutputMuted | Boolean | false | Must be ‘true’ if mono audio output is muted. |
| signalLevelStatus | Common.DeviceLevelStatus | false | Device signal level status. See DeviceLevelStatus. |
| primary AudioSource | Common.PrimaryAudioSource | false | Primary audio source. See Primary AudioSource. |
| eCallEventActive | Boolean | false | Must be ‘true’ if emergency call event is active. |

Related items:

DeviceLevelStatus – [section 5.1.25](#)

Primary AudioSource – [section 5.1.26](#)

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.2.31 HeadLampStatus

The structure describes the status of the headlights.

| Param Name | Type | Mandatory | Description |
|--------------------------|---------------------------|------------------|-------------------------------------|
| lowBeamsOn | Boolean | true | Status of the low beam lamps. |
| highBeamsOn | Boolean | true | Status of the high beam lamps. |
| ambientLightSensorStatus | Common.AmbientLightStatus | true | Status of the ambient light sensor. |

Related items:

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

[Table of Contents](#).

13.2.32 MyKey

The following structure defines whether E911 override (see [section 2](#) for definition) is on.

| Param Name | Type | Mandatory | Description |
|-------------------|--------------------------|------------------|---|
| e911Override | Common.VehicleDataStatus | true | Indicates whether e911 override is on. See VehicleDataStatus. |

Related items:

GetVehicleData – [section 11.1.5](#)

OnVehicleData – [section 11.2.1](#)

VehicleDataStatus – [section 5.1.27](#)

[Table of Contents](#).

13.2.33 Turn

The structure represents the information for TBT navigation.

| Param Name | Type | Mandatory | Description |
|-------------------|-------------|------------------|--------------------|
| | | | |

| Param Name | Type | Mandatory | Description |
|----------------|------------------------|-----------|--|
| navigationText | Common.TextFieldStruct | false | Contains the information text and the field for the text to be displayed in. Uses navigationText from TextFieldName |
| turnIcon | Common.Image | false | The image that represents the information about the turn. See Image |

Related items:

UpdateTurnList – [section 12.1.4](#)

TextFieldStruct – [section 5.2.8](#)

TextFieldName – [section 5.1.14](#)

Image – [section 5.2.11](#)

[Table of Contents.](#)

13.2.34 ECallInfo

The structure represents the information for TBT navigation.

| Param Name | Type | Mandatory | Description |
|----------------------------|--------------------------------------|-----------|--|
| eCallNotificationStatus | Common.VehicleDataNotificationStatus | | References signal "eCallNotification_4A". See VehicleDataNotificationStatus. |
| auxECallNotificationStatus | Common.VehicleDataNotificationStatus | | References signal "eCallNotification". See VehicleDataNotificationStatus. |
| eCallConfirmationStatus | Common.ECallConfirmationStatus | | References signal "eCallConfirmation". See ECallConfirmationStatus. |

Related items:

UpdateTurnList – [section 12.1.4](#)

TextFieldStruct – [section 5.2.8](#)

TextFieldName – [section 5.1.14](#)

Image – [section 5.2.11](#)

[Table of Contents.](#)

13.2.35 AirbagStatus

The structure represents the information for TBT navigation.

| Param Name | Type | Mandatory | Description |
|----------------------|-------------------------------|-----------|---|
| driverAirbagDeployed | Common.VehicleDataEventStatus | | References signal "VedsDrvBag_D_Ltchd". See VehicleDataEventStatus. |

| Param Name | Type | Mandatory | Description |
|--------------------------------|-------------------------------|-----------|--|
| driverSideAirbagDeployed | Common.VehicleDataEventStatus | | References signal "VedsDrvSideBag_D_Ltchd". See VehicleDataEventStatus. |
| driverCurtainAirbagDeployed | Common.VehicleDataEventStatus | | References signal "VedsDrvCrtnBag_D_Ltchd". See VehicleDataEventStatus. |
| passengerAirbagDeployed | Common.VehicleDataEventStatus | | References signal "VedsPasBag_D_Ltchd". See VehicleDataEventStatus |
| passengerCurtainAirbagDeployed | Common.VehicleDataEventStatus | | References signal "VedsPasCrtnBag_D_Ltchd". See VehicleDataEventStatus. |
| driverKneeAirbagDeployed | Common.VehicleDataEventStatus | | References signal "VedsKneeDrvBag_D_Ltchd". See VehicleDataEventStatus. |
| passengerSideAirbagDeployed | Common.VehicleDataEventStatus | | References signal "VedsPassSideBag_D_Ltchd". See VehicleDataEventStatus. |
| passengerKneeAirbagDeployed | Common.VehicleDataEventStatus | | References signal "VedsKneePasBag_D_Ltchd". See VehicleDataEventStatus. |

Related items:

UpdateTurnList – [section 12.1.4](#)

TextFieldStruct – [section 5.2.8](#)

TextFieldName – [section 5.1.14](#)

Image – [section 5.2.11](#)

[Table of Contents](#).

13.2.36 EmergencyEvent

The structure represents the information for TBT navigation.

| Param Name | Type | Mandatory | Description |
|--------------------|---------------------------|-----------|---|
| emergencyEventType | Common.EmergencyEventType | | References signal "VedsEvntType_D_Ltchd". See EmergencyEventType. |
| fuelCutoffStatus | Common.FuelCutoffStatus | | References signal "RCM_FuelCutoff". See FuelCutoffStatus. |

| Param Name | Type | Mandatory | Description |
|-----------------------|-------------------------------|-----------|--|
| rolloverEvent | Common.VehicleDataEventStatus | | References signal "VedsEvntRoll_D_Ltchd". See VehicleDataEventStatus. |
| maximumChangeVelocity | Common.VehicleDataEventStatus | | References signal "VedsMaxDeltaV_D_Ltchd". See VehicleDataEventStatus. |
| multipleEvents | Common.VehicleDataEventStatus | | References signal "VedsMultiEvnt_D_Ltchd". See VehicleDataEventStatus. |

Related items:

UpdateTurnList – [section 12.1.4](#)

TextFieldStruct – [section 5.2.8](#)

TextFieldName – [section 5.1.14](#)

Image – [section 5.2.11](#)

[Table of Contents.](#)

13.2.37 ClusterModeStatus

The structure represents the information for TBT navigation.

| Param Name | Type | Mandatory | Description |
|------------------------------|-------------------------------------|-----------|-------------|
| powerModeActive | Boolean | | |
| powerModeQualificationStatus | Common.PowerModeQualificationStatus | | |
| carModeStatus | Common.CarModeStatus | | |
| powerModeStatus | Common.PowerModeStatus | | |

Related items:

UpdateTurnList – [section 12.1.4](#)

TextFieldStruct – [section 5.2.8](#)

TextFieldName – [section 5.1.14](#)

Image – [section 5.2.11](#)

[Table of Contents.](#)

References

- 1) JSON-RPC 2.0 Specification: <http://www.jsonrpc.org/specification>

Change History

| Version | Date | Status | Change description | Author/Editor |
|-----------|------------|-----------------|---|---------------|
| V.1.0 | 05/23/2013 | Initial Version | Template creation; Chapters 4, 8, 10,11 creation | A.Britanova |
| V.1.1 | 06/12/2013 | Draft | Chapters 6, 12 creation | A.Britanova |
| V.1.2 | 06/25/2013 | Draft | Chapters 5, 7 creation | A.Britanova |
| V.1.3 | 07/04/2013 | Draft | Chapters 'Contents', 'Introduction', 'Overview' creation. Descriptions updating. | A.Britanova |
| V.2.0 | 07/25/2013 | Draft | Ready for customer review | A.Britanova |
| V.2.1 | 08/20/2013 | Draft | Possible layout pictures are added. | A.Britanova |
| V.2.2 | 10/25/13 | Draft | Diagrams are added. | A.Britanova |
| V.2.3 | 11/08/13 | Draft | D-Bus overview added | Y.Kazakov |
| V.2.4 | 12/02/13 | Draft | Template is changed. | A.Britanova |
| V.2.5.1 | 12/10/13 | Draft | 'WebSocket Transport' is added | A.Britanova |
| V.2.5.2 | 12/23/13 | Draft | Sections 6.1 – 6.19 are updated: expanded description, requirements to HMI, sequence diagrams. | A.Britanova |
| V.2.5.3 | 02/07/14 | Draft | Sections 7.1 – 7.5 and 7.8 – 7.26 are updated: expanded description, requirements to HMI, sequence diagrams, pictures of possible layout. | A.Britanova |
| V.2.5.4 | 02/13/14 | Draft | - Sections #7.6 'ShowKeyboard" and #7.7 "ShowCustomForm" are deleted as not actual. The numbering of further sections is shifted. - Interfaces are updated in correspondence to HMI_API that implements protocol 3.0 revision L. - Sections #7.27 - #7.31; #8.1 - #8.3; #9.1. – #9.3 are updated: expanded description, requirements to HMI, sequence diagrams. | A.Britanova |
| | | | | |
| V.2.5.10 | 05/16/14 | Draft | - Add 'Chapter 13 SDL Component Description' - Update section 10.4 Speak (add sec.10.4.2.1 p.3, sec.10.2.3.3) - Add sections 6.25 OnUpdateDeviceList, 6.26. OnResume AudioSource 6.27 UpdateAppList | |
| V.2.5.11. | 08/26/14 | Draft | - Add 'level' param to "6.2.2.2 Parameters" section - Add diagram to "6.2.4.1 BasicCommunication.ActivateApp after successful resumption." - Add diagram to "6.2.4.2 BasicCommunication.ActivateApp after UNsuccessful resumption." - Add diagram to "6.2.4.3 BasicCommunication.ActivateApp after unexpected disconnect" - Add diagram to "6.2.4.4 | |

| | | | | |
|-----------|----------|-------|--|--|
| | | | <p>BasicCommunication.ActivateApp after the User accepted the data consent prompt”</p> <ul style="list-style-type: none"> - Add diagram to “6.2.4.5 BasicCommunication.ActivateApp after the User declined the data consent prompt” - Add description to “6.15.1 Description” of “6.15 OnAppUnregistered” - Add “unexpectedDisconnect” param to “6.15.1.1 Parameter” of “6.15 OnAppUnregistered” - Update description of “7.18.2.1 Behavior” in “7.18 Slider” section. - Update description of “7.18.3 Response” in “7.18 Slider” section. - Update the diagram “7.18.4.1 Slider with static footer displayed then closed by the timeout” - Add the diagram “7.18.4.2 Slider with static footer displayed then closed by ‘OK’ button press” - Add description to “13.1.2.1 Behavior” of “13.1 SDL.ActivateApp” section - Add description to “13.1.3 Response” - Add description to “13.1.3.1 Parameters” - Add diagram “13.1.4.1 SDL.ActivateApp for the application registered from the non-consented device, which gets User’s consent afterwards” - Add diagram “13.1.4.2 SDL.ActivateApp for the application registered from the non-consented device, which does NOT get User’s consent afterwards” - Add diagram “13.1.4.3 SDL.ActivateApp for the application registered from the consented device and this application requires User’s consent for special permissions” - Add diagram “13.1.4.4 SDL.ActivateApp for the application registered from the consented device and this application was reduced in permissions by the latest PT Update.” - Add diagram “13.1.4.5 SDL.ActivateApp for the application registered from the consented device and this application was unauthorized by the latest PT Update.” - Update “13.1.5 JSON Messages Examples” | |
| V.2.5.12. | 09/29/14 | Draft | <ul style="list-style-type: none"> - Add “6.2.2.4 HMILevel” enum description - Add description to “6.5.2.1 Behavior” of “6.5 SystemRequest” - Add “SUSPEND” value to “6.17.1.2 ApplicationsCloseReason Enumeration” - Add new section “6.28 | |

| | | | |
|--|--|---|--|
| | | <p>OnSDLPersistenceComplete" with all relevant sub-sections.</p> <ul style="list-style-type: none"> - Add "locationImage" to "7.2.3.6 ImageFieldName Enumeration" - Add "alertType" param to "7.4.2.2 Parameters" of "7.4 Alert" - Add "7.4.2.4 AlertType Enumeration" - Change "customPresets -> Maxsize = <u>10</u>" in "7.5.2.2 Parameters" of "7.5 Show" - Add new params to "7.15.2.2 Parameters" of "7.15 ChangeRegistration" - Add description to "8.2.1 Description" of "8.2 OnButtonEvent" - Add "appID" param to "8.2.1.1 Parameters" of "8.2 OnButtonEvent" section - Add description to "8.3.1 Description" of "8.3 OnButtonPress" - Add "appID" param to "8.3.1.1 Parameters" of "8.3 OnButtonPress" section - Add new params to "9.6.2.2 Parameters" of "9.6 ChangeRegistration" -- Add new params to "10.6.2.2 Parameters" of "10.6 ChangeRegistration" - Add "appID" to "11.5.2.1 Parameters" of "11.5 DiagnosticMessage" - Add new section "12.8 SendLocation" with all relevant sub-sections - Add description to "13.2.1 Description" of "13.2 GetListOfPermissions" - Add description to "13.2.2.1 Behavior" - Add diagram "13.6.4.1 GetUserFriendlyMessage" - Add diagram to "13.7.2.1 OnAllowSDLFunctionality" - Add diagram, to "13.9.2.1 OnAppPermissionChanged with appPermissionsConsentNeeded:true" - Add new section "13.15 OnReceivedPolicyUpdate" - Add new section "13.16 OnPolicyUpdate" | |
|--|--|---|--|