RSDL-HMI Guidelines

1. BasicCommunication.OnAppRegistered

1.1. Description

Type:	Notification
Sender:	SDL
Purpose:	Provide the application-specific data after its registration

Disclaimer

The below describes the behavior related to RSDL functionality only.

For standard SDL behavior and parameters, please see "6.13 OnAppRegistered" of SDL Integration Guidelines .

Changes

1. New `appHMIType:` **`REMOTE_CONTROL`** was added: all and any remote-control applications must register with this type.
2. Registration of **identical applications** (that is, with the same name, same VRSynonyms, same policies_appID), one from driver's device and another one from passenger's device, **is allowed** by RSDL.
3. Passenger's applications are not allowed to be activated to FULL level.

1.2. Behavior

i. Application from driver's device registers - the vehicle must:

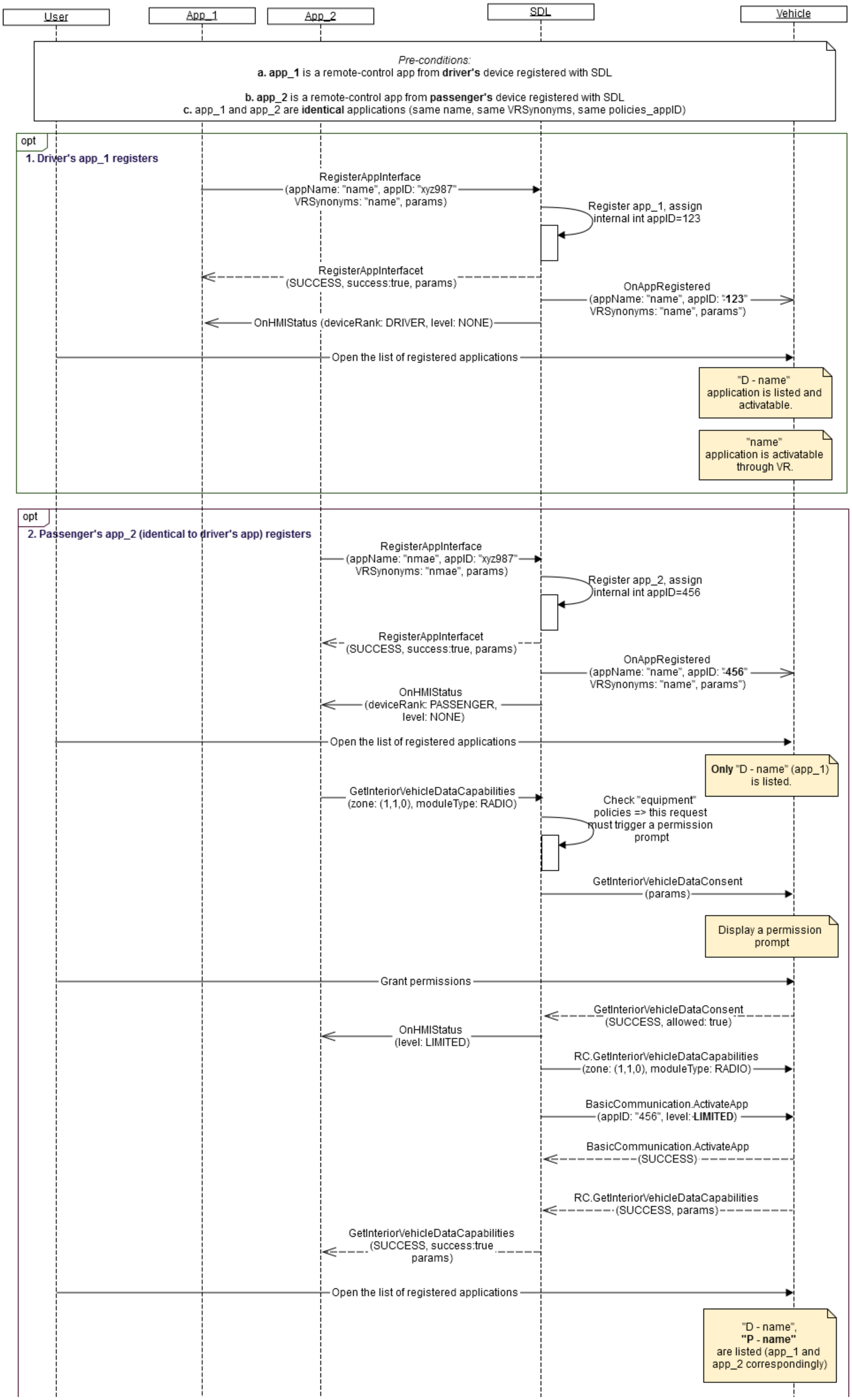
1. Know the device rank is driver's (see also `RC.OnDevcieRankChanged`)
2. Mark each application registered from driver's device with **"D - <appName>"** when displaying in the list of registered applications (all required information comes from `BasicCommunication.OnAppRegistered`)
3. Send `SDL.ActivateApp` to RSDL in case the driver activates such application through either UI or VR
4. Send `OnExitApplication(USER_EXIT)` in case the driver decides to exit such application through either UI or VR.

II. Application from passenger's device registers - the vehicle must:

1. Know the device rank is passenger's (see also `RC.OnDevcieRankChanged`)
2. Display only applications in LIMITED level (meaning, only passenger's applications that received driver's permission to remotely control. RSDL notifies about changed HMILevel via BasicCommunication.ActivateApp("level"), see "6.2 ActivateApp" of SDL Integration Guidelines)
3. In case displayed, mark each application registered from passenger's device with **"P - <appName>"**
4. Ignore passenger's application `VRSynonyms` in case the identical driver's application is also registered
5. Never allow to activate passenger's application.
6. Allow only "User Exit" for passenger's application (see "6.15 OnExitApplication" of SDL Integration Guidelines).

1.3. Sequence Diagrams

A. Registration of driver's and passenger's RC applications


User
App_1
App_2
SDL
Vehicle
Pre-conditions:
a. app_1 is a remote-control app from driver's device registered with SDL
b. app_2 is a remote-control app from passenger's device registered with SDL
c. app_1 and app_2 are identical applications (same name, same VRSynonyms, same policies_appID)
opt
1. Driver's app_1 registers
RegisterAppInterface (appName: "name", appID: "xyz987" VRSynonyms: "name", params)
Register app_1, assign internal int appID=123
RegisterAppInterfacet (SUCCESS, success:true, params)
OnAppRegistered (appName: "name", appID: "123" VRSynonyms: "name", params")
OnHMIStatus (deviceRank: DRIVER, level: NONE)
Open the list of registered applications
"D - name" application is listed and activatable.
"name" application is activatable through VR.
opt
2. Passenger's app_2 (identical to driver's app) registers
RegisterAppInterface (appName: "nmae", appID: "xyz987" VRSynonyms: "nmae", params)
Register app_2, assign internal int appID=456
RegisterAppInterfacet (SUCCESS, success:true, params)
OnAppRegistered (appName: "name", appID: "456" VRSynonyms: "name", params")
OnHMIStatus (deviceRank: PASSENGER, level: NONE)
Open the list of registered applications
Only "D - name" (app_1) is listed.
GetInteriorVehicleDataCapabilities (zone: (1,1,0), moduleType: RADIO)
Check "equipment" policies => this request must trigger a permission prompt
GetInteriorVehicleDataConsent (params)
Display a permission prompt
Grant permissions
GetInteriorVehicleDataConsent (SUCCESS, allowed: true)
OnHMIStatus (level: LIMITED)
RC.GetInteriorVehicleDataCapabilities (zone: (1,1,0), moduleType: RADIO)
BasicCommunication.ActivateApp (appID: "456", level: LIMITED)
BasicCommunication.ActivateApp (SUCCESS)
RC.GetInteriorVehicleDataCapabilities (SUCCESS, params)
GetInteriorVehicleDataCapabilities (SUCCESS, success:true params)
Open the list of registered applications
"D - name", "P - name" are listed (app_1 and app_2 correspondingly)

Only "name" application (**app_1**) is activatable through VR.

1.4. JSON Messages Examples

Notification

```
{
     "jsonrpc" : "2.0",
     "method" : "BasicCommunication.OnAppRegistered",
     “params” :
     {
            "application" :
            {
                   "appName" : "TryMe",
                   "ngnMediaScreenAppName" : "TryMe",
                   "appType" : "REMOTE_CONTROL",
                   "appID" : 65540,
                   "hmiDisplayLanguageDesired" : "ES-ES",
                   "isMediaApplication" : false ,
                   "policyAppID" : "xyz098"
                   "deviceInfo " :
                   {
                          "name" : “GT-I9300”,
                          "id" : 1563462,
                          "transportType" : “BLUETOOTH”
                   },
            },
            "vrSynonyms" : "TryMe"
     }
}
```

2. RC.GetInteriorVehicleDataCapabilities

2.1. Description

Type:	Function
Sender:	SDL
Purpose:	Obtain vehicle capabilities on available modules controls in available locations

Trigger to send:

RSDL sends `RC.GetInteriorVehicleDataCapabilities` to the vehicle after receiving a valid `GetInteriorVehicleDataCapabilities` request from remote-control application.

Back-up:

RSDL has a logic to read the capabilities from the predefined file so that to respond to the application *in case* the vehicle fails to respond or responds with error to `RC.GetInteriorVehicleDataCapabilities`.

To switch this logic on, the OEM must ensure the following:

1. `"InteriorVehicleDataCapabilities.json"` file that lists all available locations and available modules must be created (please see "Response" in "JSON message example" below for the expected structure)
2. `"InteriorVDCapabilitiesFile"` parameter of `SmartDeviceLink.ini` file must contain the path to this file.
   Example: `InteriorVDCapabilitiesFile = ./plugins/InteriorVehicleDataCapabilities.json`

In case any of the above 2 steps is omitted, RSDL will:

- transfer vehicle's error to the mobile app in case the vehicle responds `GetInteriorVehicleDataCapabilities` with an error.
- respond with `GENERIC_ERROR` result to the mobile app in case the vehicle does not respond during RSDL's internal timeout for vehicle RPCs (`"DefaultTimeout"` parameter in `SmartDeviceLink.ini` file; set to 10 sec by default)

2.2. Request

Behavior:

1. The vehicle must:
   1.1. Provide the list:
      a. Of available modules for controlling in the requested zone - in case the request comes with `"zone"` parameter only
      b. Of available zones for controlling with the requested module - in case the request comes with `"moduleTypes"` parameter only
      c. Of available modules for controlling in the available zones - in case the request comes without parameters
   1.2. Respond with SUCCESS result (see example "response" and diagram A) during 10 sec after getting `RC.GetInteriorVehicleDataCapabilities` request

2. The vehicle may:
   2.1. Respond with
      a. Either REJECTED resultCode
      b. Or do not respond at all
   in order to trigger RSDL reading capabilities from a predefined file (see diagram B) for responding to mobile application.
   Important: *conditions from "Back-up" of "Description" must be satisfied, otherwise RSDL will respond with error to mobile app (see diagram C).*

Parameters

Request Parameters

Param Name	Type	Mandatory	Additional	Description
`zone`	`Common.InteriorZone`	`false`	-	If included, only the corresponding modules able to be controlled by that zone will be sent back. If not included, all modules will be returned regardless of their ability to be controlled by specifc zones.

moduleTypes	Common.ModuleType	false	array = true
			minSize = 1
			maxSize = 1000
				If included, only the corresponding type of modules a will be sent back. If not included, all module types will be returned.

appID	Integer	true	-	Internal SDL-assigned Id of the application that requested this RPC.

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

col	Integer	true	minValue = 0
			maxValue = 100
				The value of column from "column-row-level" grid that defines the location.

row	Integer	true	minValue = 0
			maxValue = 100
				The value of row from "column-row-level" grid that defines the location.

level	Integer	true	minValue = 0
			maxValue = 100
				The value of level from "column-row-level" grid that defines the location.

colspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of columns in "column-row-level" grid that defines the location.

rowspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of rows in "column-row-level" grid that defines the location.

levelspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of levels in "column-row-level" grid that defines the location.

ModuleType Enum

Param Name	Description
CLIMATE	Names the Climate module of the vehicle
RADIO	Names the Radio module of the vehicle

2.3. Response

Behavior:
The below table lists resultCodes that the vehicle may use to respond:

Result	Description	Message type		Message Params	Notes
		WebSocket	D-Bus		
Success	SUCCESS The vehicle provides the list of zones and modules per application's request.	JSON response	Regular response	interiorVehicleDataCapabilities code: 0	See example "response".
Failure	REJECTED The vehicle intends to trigger "read capabilities from file" RSDL's logic.	JSON error message	Regular response	code: 4	Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. See example "error message".
	GENERIC_ERROR The unknown issue occurred or other codes are not applicable.			code: 22	

Parameters

Response parameters

Param Name	Type	Mandatory	Additional	Description

interiorVehicleDataCapabilities	Common.ModuleDescription	true	array = true
			minSize = 1
			maxSize = 1000
				Capabilities of the vehicle: which module is available at which location.

ModuleDescription Struct

Param Name	Type	Mandatory	Additional	Description
moduleZone	Common.InteriorZone	true	-	Defines the location where the module is available for controlling.
moduleType	Common.ModuleType	true	-	Defines the module available for controlling.

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

col	Integer	true	minValue = 0
			maxValue = 100
				The value of column from "column-row-level" grid that defines the location.

row	Integer	true	minValue = 0
			maxValue = 100
				The value of row from "column-row-level" grid that defines the location.

level	Integer	true	minValue = 0
			maxValue = 100
				The value of level from "column-row-level" grid that defines the location.

colspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of columns in "column-row-level" grid that defines the location.

rowspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of rows in "column-row-level" grid that defines the location.

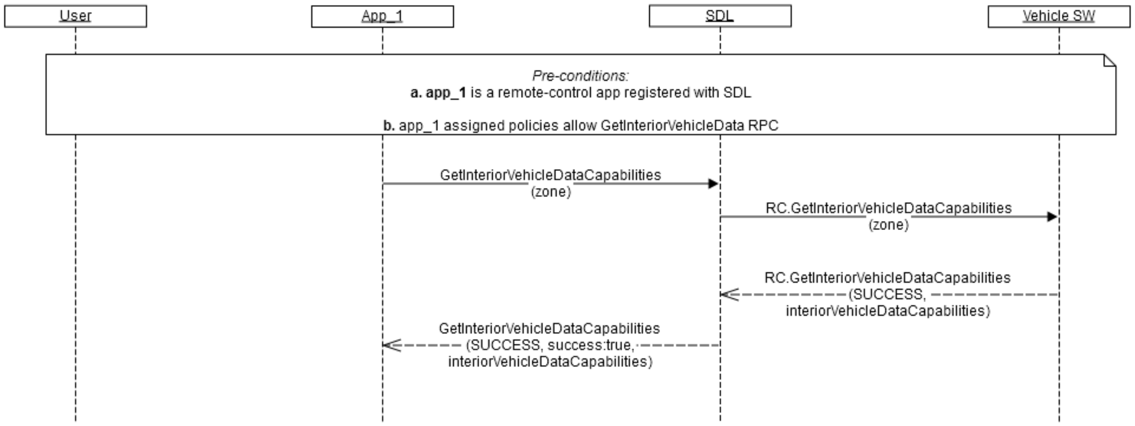
levelspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of levels in "column-row-level" grid that defines the location.

ModuleType Enum

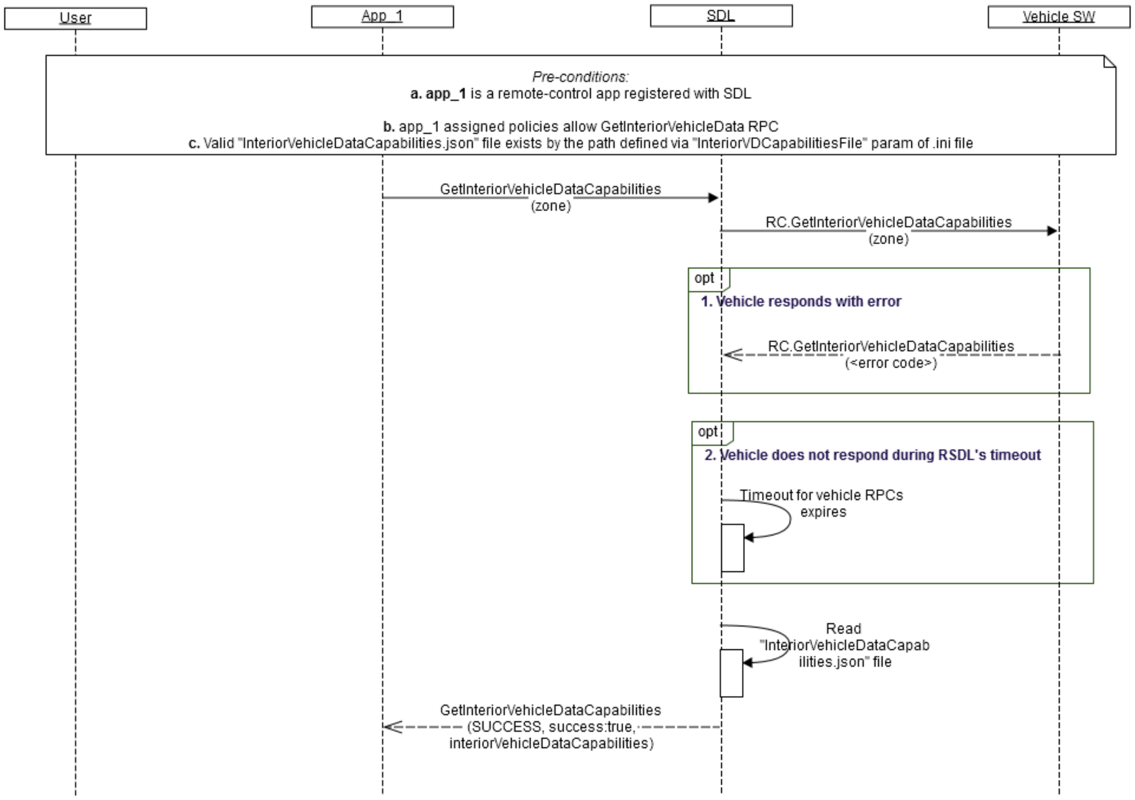
Param Name	Description
CLIMATE	Names the Climate module of the vehicle
RADIO	Names the Radio module of the vehicle

2.4. Sequence Diagrams

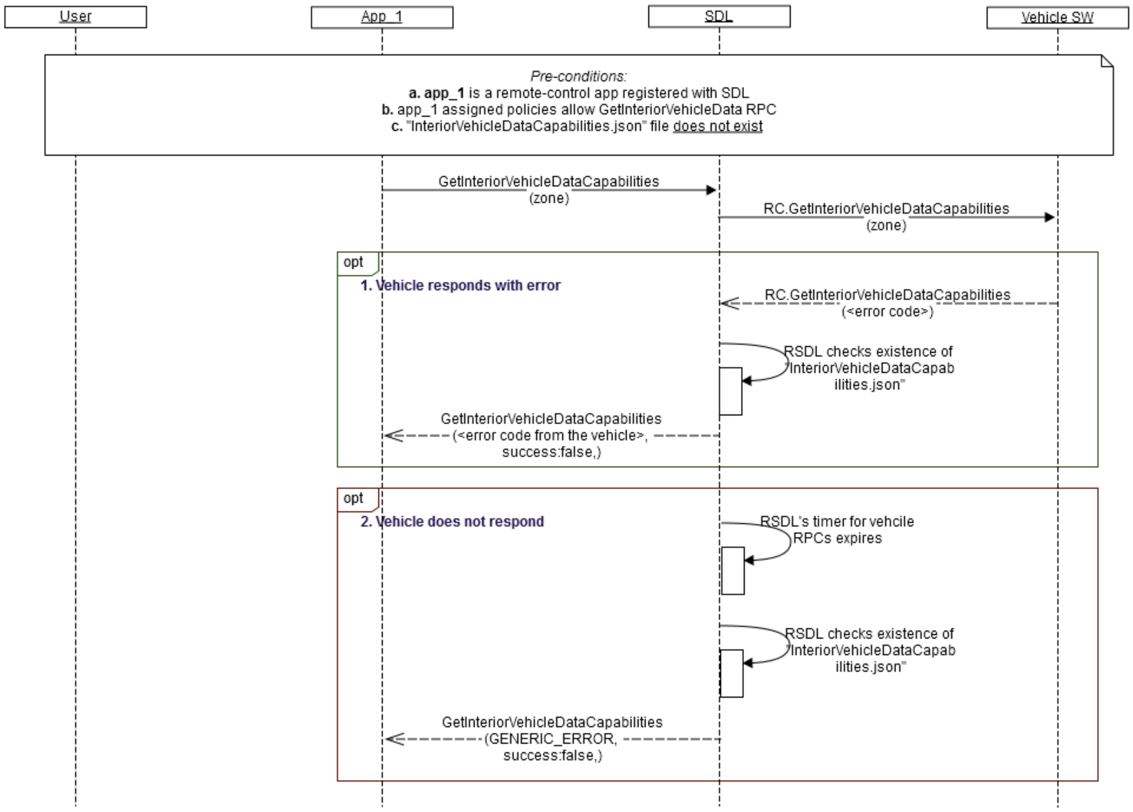
A. RSDL transfers vehcile's successful response to the mobile application


User
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is a remote-control app registered with SDL
b. app_1 assigned policies allow GetInteriorVehicleData RPC
GetInteriorVehicleDataCapabilities (zone)
RC.GetInteriorVehicleDataCapabilities (zone)
RC.GetInteriorVehicleDataCapabilities (SUCCESS, interiorVehicleDataCapabilities)
GetInteriorVehicleDataCapabilities (SUCCESS, success:true, interiorVehicleDataCapabilities)

B. RSDL reads capabilities from file in case of erroneous response or non-response from the vehicle


User
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is a remote-control app registered with SDL
b. app_1 assigned policies allow GetInteriorVehicleData RPC
c. Valid "InteriorVehicleDataCapabilities.json" file exists by the path defined via "InteriorVDCapabilitiesFile" param of .ini file
GetInteriorVehicleDataCapabilities (zone)
RC.GetInteriorVehicleDataCapabilities (zone)
opt
1. Vehicle responds with error
RC.GetInteriorVehicleDataCapabilities (<error code>)
opt
2. Vehicle does not respond during RSDL's timeout
Timeout for vehicle RPCs expires
Read "InteriorVehicleDataCapabilities.json" file
GetInteriorVehicleDataCapabilities (SUCCESS, success:true, interiorVehicleDataCapabilities)

C. RSDL sends erroneous response to the application


User
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is a remote-control app registered with SDL
b. app_1 assigned policies allow GetInteriorVehicleData RPC
c. "InteriorVehicleDataCapabilities.json" file does not exist
GetInteriorVehicleDataCapabilities (zone)
RC.GetInteriorVehicleDataCapabilities (zone)
opt
1. Vehicle responds with error
RC.GetInteriorVehicleDataCapabilities (<error code>)
RSDL checks existence of "InteriorVehicleDataCapabilities.json"
GetInteriorVehicleDataCapabilities (<error code from the vehicle>, success:false,)
opt
2. Vehicle does not respond
RSDL's timer for vehcile RPCs expires
RSDL checks existence of "InteriorVehicleDataCapabilities.json"
GetInteriorVehicleDataCapabilities (GENERIC_ERROR, success:false,)

2.5. JSON Messages Examples

Request

```
{
    "id" : 64,
    "jsonrpc" : "2.0",
    "method" : "RC.GetInteriorVehicleDataCapabilities",
    “params” :
    {
     "zone" :
     {
               "col" : “0”,
               "row" : "1",
               "level" : "0",
               "colspan" : "2",
               "rowspan" : "2",
               "levelspan" : "0"
     }
    }
}
```

Response

```
{
     "id" : 64,
     "jsonrpc" : "2.0",
     "result" :
     {
      "interiorVehicleDataCapabilities" :
     [
{
      "moduleZone" :
      {
                 "col" : “0”,
                 "row" : "1",
                 "level" : "0",
                 "colspan" : "2",
                 "rowspan" : "2",
                 "levelspan" : "0"
      },
     "moduleType" : RADIO
      },
{
      "moduleZone" :
      {
                 "col" : “0”,
                 "row" : "1",
                 "level" : "0",
                 "colspan" : "2",
                 "rowspan" : "2",
                 "levelspan" : "0"
      },
     "moduleType" : CLIMATE
      }
     ],
      "code" : 0,
      "method" : "RC.GetInteriorVehicleDataCapabilities"
     }
}
```

Error message

```
{
    "id" : 64,
    "jsonrpc" : "2.0",
    "error" :
    {
     "code" : 4,
     "message" : "The request is rejected",
     "data" :
     {
              "method" : "RC.GetInteriorVehicleDataCapabilities"
     }
    }
}
```

3. RC.GetInteriorVehicleData

3.1. Description

Type:	Function
Sender:	SDL
Purpose:	Obtain vehicle data and/or subscribe to notifications from the specified module in the specified location

Trigger to send:

RSDL sends `RC.GetInteriorVehicleData` to the vehicle:

1. After receiving a valid `GetInteriorVehicleData` request from remote-control application
2. Following the internal logic with the purpose to unsubscribe the application from notifications

Error handling:

1. RSDL validates response from the vehicle and in case it's invalid, RSDL returns `GENERIC_ERROR` result to the corresponding mobile application's request.

   Validation includes:

   - Mandatory parameters to be present

   - Type of parameters values to correspond HMI_API

   - Parameters values to be in bounds with HMI_API

   - JSON format to be valid.

   - Module and ControlData to match *(for example, response with "`RadioControlData`" sent together with `CLIMATE moduleType` is invalid)*

2. RSDL waits for response from the vehicle during RSDL's internal timeout for vehicle RPCs ("`DefaultTimeout`" parameter in `SmartDeviceLink.ini` file; set to 10 sec by default) and in case it's not received, RSDL returns `GENERIC_ERROR` result to the corresponding mobile application's request.

3.2. Request

Behavior:

1. *Happy path* (see diagram A) - the vehicle must:
 1.1. Process the request in the following way:
  a. Read the data from the requested module in the requested location (see example "request").
  b. In case "`subscribe: true`" exists in the request - subscribe the application with the requested `appID` to `OnInteriorVehicleData` notifications from the requested module in the requested location.
  c. In case "`subscribe: false`" exists in the request - unsubscribe the application with the requested `appID` from `OnInteriorVehicleData` notifications from the requested module in the requested location.
   Information: Initially, the vehicle obtains application's `appID` via `BasicCommunication.OnAppRegistered` notification from SDL.

 1.2. Form a valid JSON response with
  a. Retrieved data put as parameters values (see "parameters" of Response).
  b. "`isSubscribed: true`" - *in case* the "subscribe" parameter exists in the request *and* the vehicle subscribed the application (either successfully subscribed or failed to unsubscribe) to `OnInteriorVehicleData` notifications from the requested module in the requested location.
  c. "`isSubscribed: false`" - *in case* the "subscribe" parameter exists in the request *and* the vehicle unsubscribed the application (either successfully unsubscribed or failed to subscribe) from `OnInteriorVehicleData` notifications from the requested module in the requested location.

1.3. Send a response during 10 sec from the time of request receipt.

2. *Module is busy* (see diagram B) - the vehicle must:
   2.1. Respond with REJECTED resultCode in case the request is <u>from passenger's</u> application *and* <u>another passenger's</u> application is currently controlling the same module (no matter in what location).
   <u>Information:</u> The vehicle is expected to allow only one passenger's application and one driver's application to control the same module at the same time.

3. *Other result codes* - the vehicle may:
   3.1. Implement additional checks and return the resultCodes described in "applicable result codes" table of Response.

Parameters

Request Parameters

Param Name	Type	Mandatory	Additional	Description
`moduleDescription`	`Common.ModuleDescription`	`true`	-	The zone and module data to retrieve from the vehicle for that zone.
`subscribe`	`Boolean`	`false`	-	If subscribe is true, the head unit will send `onInteriorVehicleData` notifications for the moduleDescription
`appID`	`Integer`	`true`	-	Internal SDL-assigned Id of the application related to this RPC.

ModuleDescription Struct

Param Name	Type	Mandatory	Additional	Description
`moduleZone`	`Common.InteriorZone`	`true`	-	Defines the location where the module is available for controlling.
`moduleType`	`Common.ModuleType`	`true`	-	Defines the module available for controlling.

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

`col`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of column from "column-row-level" grid that defines the location.

`row`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of row from "column-row-level" grid that defines the location.

`level`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of level from "column-row-level" grid that defines the location.

`colspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of columns in "column-row-level" grid that defines the location.

`rowspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of rows in "column-row-level" grid that defines the location.

`levelspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of levels in "column-row-level" grid that defines the location.

ModuleType Enum

Param Name	Description
`CLIMATE`	Names the Climate module of the vehicle
`RADIO`	Names the Radio module of the vehicle

3.3. Response

Behavior:
The below table lists resultCodes that the vehicle may use to respond:

Result	Description	Message type		Message Params	Notes
		WebSocket	D-Bus		
Success	SUCCESS The vehicle successfully processes the request: - Provides the data for the requested module in the requested location - Subscribes or unsubscribes the application from notifications if requested.	JSON response	Regular response	moduleData, isSubscribed, code: 0	See example "response".
Failure	UNSUPPORTED_RESOURCE The vehicle does not support the requested module in the requested location.	JSON error message	Regular response	code: 2	Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. See example "error message".
	REJECTED The vehicle rejects GetInteriorVehicleData request because of - Other tasks of higher priority - The requested module is busy with another passenger's application.			code: 4	
	IGNORED The vehicle ignores the request because the data and subscription status has not changed since the previous request.			code: 6	
	GENERIC_ERROR The unknown issue occurred or other codes are not applicable.			code: 22	

.

Parameters

Response parameters

Param Name	Type	Mandatory	Additional	Description
moduleData	Common.ModuleData	true	-	The data from the requested module in the requested location.
isSubscribed	Boolean	true	-	Is a conditional-mandatory parameter: must be returned in case "subscribe" parameter was present in the related request. - if "true" - the "moduleDescription" from request is successfully subscribed and the head unit will send onInteriorVehicleData notifications for the moduleDescription. - if "false" - the "moduleDescription" from request is either unsubscribed or failed to subscribe.

.

Module:

ModuleData Struct

Param Name	Type	Mandatory	Additional	Description
moduleType	Common.ModuleType	true	-	Defines the module the data is provided for.
moduleZone	Common.InteriorZone	true	-	Defines the zone the data is provided for.
radioControlData	Common.RadioControlData	false	-	Must be provided in case the module is RADIO.
climateControlData	Common.ClimateControlData	false	-	Must be provided in case the module is CLIMATE.

ModuleType Enum

Param Name	Description
CLIMATE	Names the Climate module of the vehicle
RADIO	Names the Radio module of the vehicle

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

col	Integer	true	minValue = 0
			maxValue = 100
				The value of column from "column-row-level" grid that defines the location.

row	Integer	true	minValue = 0
			maxValue = 100
				The value of row from "column-row-level" grid that defines the location.

level	Integer	true	minValue = 0
			maxValue = 100
				The value of level from "column-row-level" grid that defines the location.

colspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of columns in "column-row-level" grid that defines the location.

rowspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of rows in "column-row-level" grid that defines the location.

levelspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of levels in "column-row-level" grid that defines the location.

Radio:

RadioControlData Struct

Param Name	Type	Mandatory	Additional	Description

frequencyInteger	Integer	false	minValue = 0
			maxValue = 1710
				The integer part of frequency value (for 101.7 this value should be 101).

frequencyFraction	Integer	false	minValue = 0
			maxValue = 9
				The fractional part of frequency value (for 101.7 is 7).

band	Common.RadioBand	false	-	The band: AM, FM or XM.
rdsData	Common.RdsData	false	-	RDS (Radio Data System) information.

availableHDs	Integer	false	minValue = 1
			maxValue = 3
				Number of HD sub-channels if available

hdChannel	Integer	false	minValue = 1
			maxValue = 3
				Current HD sub-channel if available.

signalStrength	Integer	false	minValue = 0
			maxValue = 100
				Current frequency signal strength.

signalChangeThreshold	Integer	false	minValue = 0
			maxValue = 100
				If the signal strength falls below the set value for this parameter, the radio will tune to an alternative frequency

radioEnable	Boolean	false	-	True if the radio is on, false is the radio is off.
state	Common.RadioState	false	-	The radio state: acquiring, acquired, multi-cast.

RadioBand Enum

Param Name	Description
AM	AM (amplitude modulation) band
FM	FM (frequency modulation) band
XM	XM Satellite Radio

RdsData Struct

Param Name	Type	Mandatory	Additional	Description

PS	String	false	minlength = 0
			maxlength = 8
				Program Service Name

RT	String	false	minlength = 0
			maxlength = 64
				Radio Text

CT	String	false	minlength = 24
			maxlength = 24
				The clock text in UTC format as YYYY-MM-DDThh:mm:ss.sTZD

PI	String	false	minlength = 0
			maxlength = 6
				Program Identification - the call sign for the radio station

PTY	Integer	false	minValue = 0
			maxValue = 31
				The program type - The region should be used to differentiate between EU and North America program types

TP	Boolean	false	-	Traffic Program Identification - Identifies a station that offers traffic
TA	Boolean	false	-	Traffic Announcement Identification - Indicates an ongoing traffic announcement

REG	String	false	minlength = 0
			maxlength = 255
				Region

RadioState Enum

Param Name	Description
ACQUIRING	-
ACQUIRED	-
MULTICAST	-
NOT_FOUND	-

Climate:

ClimateControlData Struct

Param Name	Type	Mandatory	Additional	Description

`fanSpeed`	`Integer`	`false`	`minValue = 0`
			`maxValue = 100`
				The speed of the fan.

`currentTemp`	`Integer`	`false`	`minValue = 0`
			`maxValue = 100`
				The temperature currently set on the module.

`desiredTemp`	`Integer`	`false`	`minValue = 0`
			`maxValue = 100`
				The temperature to be set on the module (must be used with `SetInteriorVehicleData` only).

`temperatureUnit`	`Common.TemperatureUnit`	`false`	-	Units the temperature is measured in: Kelvin, Fahrenheit or Celsius
`acEnable`	`Boolean`	`false`	-	If "true" - air conditioning is ON.
`circulateAirEnable`	`Boolean`	`false`	-	If "true" - air circulation is ON.
`autoModeEnable`	`Boolean`	`false`	-	If "true" - the climate system is in auto mode.
`defrostZone`	`Common.DefrostZone`	`false`	-	Defines the defrost zone: front, rear or all.
`dualModeEnable`	`Boolean`	`false`	-	If "true" - the dual mode in ON and driver's and passenger's zones can have separate climate settings.

TemperatureUnit Enum

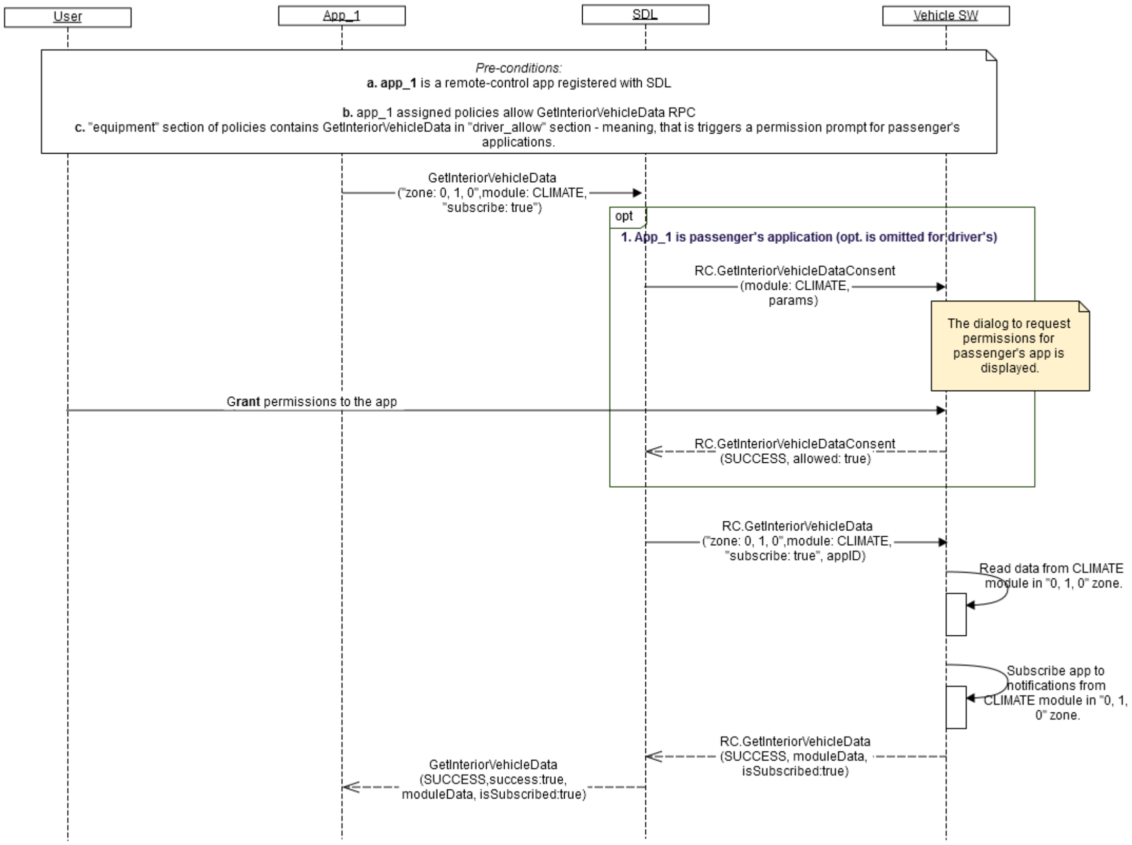
Param Name	Description
`KELVIN`	-
`FAHRENHEIT`	-
`CELSIUS`	-

DefrostZone Enum

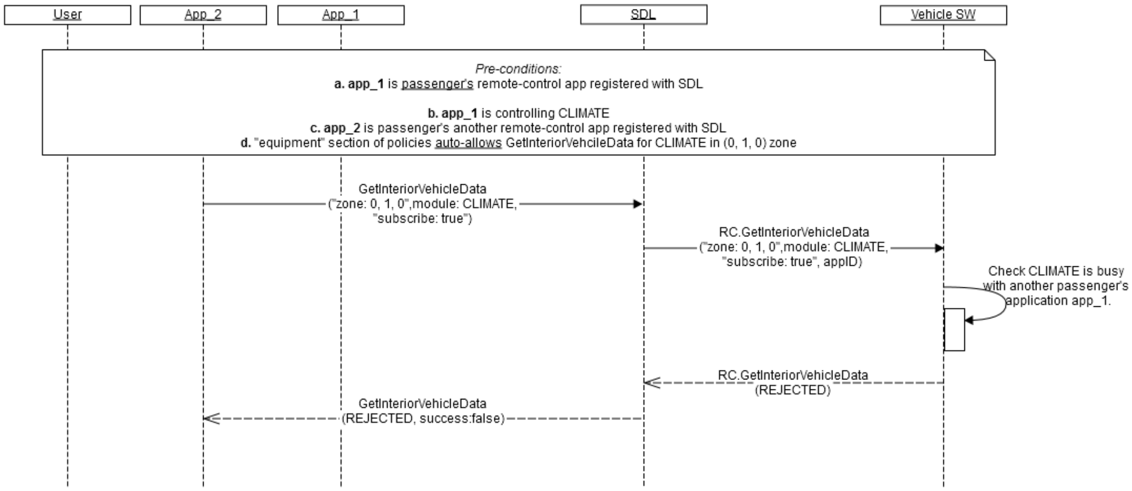
Param Name	Description
`FRONT`	-
`REAR`	-
`ALL`	-

3.4. Sequence Diagrams

A. RSDL transfers vehcile's successful response to the mobile application


User
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is a remote-control app registered with SDL
b. app_1 assigned policies allow GetInteriorVehicleData RPC
c. "equipment" section of policies contains GetInteriorVehicleData in "driver_allow" section - meaning, that is triggers a permission prompt for passenger's applications.
GetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, "subscribe: true")
opt
1. App_1 is passenger's application (opt. is omitted for driver's)
RC.GetInteriorVehicleDataConsent (module: CLIMATE, params)
The dialog to request permissions for passenger's app is displayed.
Grant permissions to the app
RC.GetInteriorVehicleDataConsent (SUCCESS, allowed: true)
RC.GetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, "subscribe: true", appID)
Read data from CLIMATE module in "0, 1, 0" zone.
Subscribe app to notifications from CLIMATE module in "0, 1, 0" zone.
RC.GetInteriorVehicleData (SUCCESS, moduleData, isSubscribed:true)
GetInteriorVehicleData (SUCCESS,success:true, moduleData, isSubscribed:true)

B. The vehcile rejects the request from another passenger's application for the busy module


User
App_2
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is passenger's remote-control app registered with SDL
b. app_1 is controlling CLIMATE
c. app_2 is passenger's another remote-control app registered with SDL
d. "equipment" section of policies auto-allows GetInteriorVehcileData for CLIMATE in (0, 1, 0) zone
GetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, "subscribe: true")
RC.GetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, "subscribe: true", appID)
Check CLIMATE is busy with another passenger's application app_1.
RC.GetInteriorVehicleData (REJECTED)
GetInteriorVehicleData (REJECTED, success:false)

5. JSON Messages Examples

Request

```
{
  "id" : 68,
  "jsonrpc" : "2.0",
  "method" : "RC.GetInteriorVehicleData",
  “params” :
  {
   "moduleDescription" :
   {
        "moduleZone" :
        {
              "col" : 0,
              "row" : 1,
              "level" : 0,
              "colspan" : 2,
              "rowspan" : 2,
              "levelspan" : 0

         },
        "moduleType" : "CLIMATE"
   },
   "appID" : 65128
  }
}
```

Response

```
{
     "id" : 68,
     "jsonrpc" : "2.0",
     "result" :
     {
            "moduleData" :
            {
                    "moduleType" : "CLIMATE",
                    "moduleZone" :
                    {
                            "col" : 0,
                            "row" : 1,
                            "level" : 0,
                            "colspan" : 2,
                            "rowspan" : 2,
                            "levelspan" : 0
                    },
                    "climateControlData" :
                    {
                            "fanSpeed" : 40,
                            "currentTemp" : 20,
                            "temperatureUnit" : “CELSIUS”,
                            "acEnable" : true
                    }
            },
      "code" : 0,
      "method" : "RC.GetInteriorVehicleData"
     }
}
```

Error message

```
{
     "id" : 68,
     "jsonrpc" : "2.0",
     "error" :
     {
      "code" : 4,
      "message" : "The request is rejected because the module is busy",
      "data" :
      {
                 "method" : "RC.GetInteriorVehicleData"
      }
     }
}
```

4. RC.SetInteriorVehicleData

4.1. Description

Type:	Function
Sender:	SDL
Purpose:	Control the specified module in the specified location

Trigger to send:

RSDL sends `RC.SetInteriorVehicleData` to the vehicle after receiving a valid `SetInteriorVehicleData` request from remote-control application

Error handling:

1. RSDL validates response from the vehicle and in case it's invalid, RSDL returns `GENERIC_ERROR` result to the corresponding mobile application's request.

   Validation includes:

   - Mandatory parameters to be present

   - Type of parameters values to correspond HMI_API

   - Parameters values to be in bounds with HMI_API

   - JSON format to be valid.

   - Module and ControlData to match *(for example, response with "`RadioControlData`" sent together with `CLIMATE moduleType` is invalid)*

2. RSDL waits for response from the vehicle during RSDL's internal timeout for vehicle RPCs ("`DefaultTimeout`" parameter in `SmartDeviceLink.ini` file; set to 10 sec by default) and in case it's not received, RSDL returns `GENERIC_ERROR` result to the corresponding mobile application's request.

"Read only" validation:

RSDL validates the `SetInteriorVehicleData` request from remote-control application with the purpose to filter (never send to the vehicle) the read-only parameters, which are the following:

1. RADIO module read-only parameters:
   - "rdsData"
   - "availableHDs"
   - "signalStrength"
   - "signalChangeThreshold"
   - "state"
   - "radioEnable"

2. CLIMATE mudule read-only parameters:
   - "currentTemp"

4.2. Request

Behavior:

1. *Happy path* (see diagram A) - the vehicle must:
   1.1. Process the request in the following way:
   a. Set the data to the requested module in the requested location (see "parameters" of Request and example "request").
   b. Read the data (with the purpose to know whether the requested data is successfully set) from the requested module in the requested location.

1.2. Form a valid JSON response with
a. Retrieved data put as parameters values (see "parameters" of Response).

1.3. Send a response during 10 sec from the time of request receipt.

2. *Module is busy* (see diagram B and example "error message") - the vehicle must:
2.1. Respond with REJECTED resultCode in case the request is from passenger's application *and* another passenger's application is currently controlling the same module (no matter in what location).
Information: The vehicle is expected to allow only one passenger's application and one driver's application to control the same module at the same time.

3. *Other result codes* - the vehicle may:
3.1. Implement additional checks and return the resultCodes described in "applicable result codes" table of Response.

Parameters

Request parameters

Param Name	Type	Mandatory	Additional	Description
`moduleData`	`Common.ModuleData`	`true`	-	Data to be set for the specified module in the specified location
`appID`	`Integer`	`true`	-	Internal SDL-assigned Id of the application related to this RPC.

.

Module:

ModuleData Struct

Param Name	Type	Mandatory	Additional	Description
`moduleType`	`Common.ModuleType`	`true`	-	Defines the module the data is provided for.
`moduleZone`	`Common.InteriorZone`	`true`	-	Defines the zone the data is provided for.
`radioControlData`	`Common.RadioControlData`	`false`	-	Must be provided in case the module is RADIO.
`climateControlData`	`Common.ClimateControlData`	`false`	-	Must be provided in case the module is CLIMATE.

ModuleType Enum

Param Name	Description
`CLIMATE`	Names the Climate module of the vehicle
`RADIO`	Names the Radio module of the vehicle

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

`col`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of column from "column-row-level" grid that defines the location.

`row`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of row from "column-row-level" grid that defines the location.

`level`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of level from "column-row-level" grid that defines the location.

`colspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of columns in "column-row-level" grid that defines the location.

`rowspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of rows in "column-row-level" grid that defines the location.

`levelspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of levels in "column-row-level" grid that defines the location.

Radio:

RadioControlData Struct

Param Name	Type	Mandatory	Additional	Description

`frequencyInteger`	`Integer`	`false`	`minValue = 0`
			`maxValue = 1710`
				The integer part of frequency value (for 101.7 this value should be 101).

`frequencyFraction`	`Integer`	`false`	`minValue = 0`
			`maxValue = 9`
				The fractional part of frequency value (for 101.7 is 7).

`band`	`Common.RadioBand`	`false`	-	The band: AM, FM or XM.
`rdsData`	`Common.RdsData`	`false`	-	RDS (Radio Data System) information. Read-only parameter.

`availableHDs`	`Integer`	`false`	`minValue = 1`
			`maxValue = 3`
				Number of HD sub-channels if available. Read-only parameter.

`hdChannel`	`Integer`	`false`	`minValue = 1`
			`maxValue = 3`
				Current HD sub-channel if available.

`signalStrength`	`Integer`	`false`	`minValue = 0`
			`maxValue = 100`
				Current frequency signal strength. Read-only parameter.

`signalChangeThreshold`	`Integer`	`false`	`minValue = 0`
			`maxValue = 100`
				If the signal strength falls below the set value for this parameter, the radio will tune to an alternative frequency. Read-only parameter.

`radioEnable`	`Boolean`	`false`	-	True if the radio is on, false is the radio is off. Read-only parameter.
`state`	`Common.RadioState`	`false`	-	The radio state: acquiring, acquired, multi-cast. Read-only parameter.

RadioBand Enum

Param Name	Description
`AM`	AM (amplitude modulation) band
`FM`	FM (frequency modulation) band
`XM`	XM Satellite Radio

RdsData Struct

Param Name	Type	Mandatory	Additional	Description

`PS`	`String`	`false`	`minlength = 0`
			`maxlength = 8`
				Program Service Name

RT	String	false	minlength = 0
			maxlength = 64
				Radio Text

CT	String	false	minlength = 24
			maxlength = 24
				The clock text in UTC format as YYYY-MM-DDThh:mm:ss.sTZD

PI	String	false	minlength = 0
			maxlength = 6
				Program Identification - the call sign for the radio station

PTY	Integer	false	minValue = 0
			maxValue = 31
				The program type - The region should be used to differentiate between EU and North America program types

TP	Boolean	false	-	Traffic Program Identification - Identifies a station that offers traffic
TA	Boolean	false	-	Traffic Announcement Identification - Indicates an ongoing traffic announcement

REG	String	false	minlength = 0
			maxlength = 255
				Region

RadioState Enum

Param Name	Description
ACQUIRING	-
ACQUIRED	-
MULTICAST	-
NOT_FOUND	-

Climate:

ClimateControlData Struct

Param Name	Type	Mandatory	Additional	Description

fanSpeed	Integer	false	minValue = 0
			maxValue = 100
				The speed of the fan.

currentTemp	Integer	false	minValue = 0
			maxValue = 100
				The temperature currently set on the module. Read-only parameter.

desiredTemp	Integer	false	minValue = 0
			maxValue = 100
				The temperature to be set on the module (must be used with SetInteriorVehicleData only).

temperatureUnit	Common.TemperatureUnit	false	-	Units the temperature is measured in: Kelvin, Fahrenheit or Celsius
acEnable	Boolean	false	-	If "true" - air conditioning is ON.
circulateAirEnable	Boolean	false	-	If "true" - air circulation is ON.
autoModeEnable	Boolean	false	-	If "true" - the climate system is in auto mode.

`defrostZone`	`Common.DefrostZone`	`false`	-	Defines the defrost zone: front, rear or all.
`dualModeEnable`	`Boolean`	`false`	-	If "true" - the dual mode in ON and driver's and passenger's zones can have separate climate settings.

TemperatureUnit Enum

Param Name	Description
`KELVIN`	-
`FAHRENHEIT`	-
`CELSIUS`	-

DefrostZone Enum

Param Name	Description
`FRONT`	-
`REAR`	-
`ALL`	-

4.3. Response

Behavior:
The below table lists resultCodes that the vehicle may use to respond:

Result	Description	Message type		Message Params	Notes
		WebSocket	D-Bus		

Success	`SUCCESS`
	The vehicle successfully processes the request:
	- Sets the data for the requested module in the requested location
		JSON response	Regular response	`moduleData,`
				`code: 0`
					See example "response".

Failure	`UNSUPPORTED_RESOURCE` The vehicle does not support the requested module in the requested location.	JSON error message	Regular response	`code: 2`	Applicable for this RPC result codes. Please see `Result` Enumeration for all SDL-supported codes. See example "error message".
	`REJECTED` The vehicle rejects GetInteriorVehicleData request because of - Other tasks of higher priority - The requested module is busy with another passenger's application.			`code: 4`	
	`IGNORED` The vehicle ignores the request because the requested data to be set are the same as in the previous request.			`code: 6`	
	`GENERIC_ERROR` The unknown issue occurred or other codes are not applicable.			`code: 22`	
	`READ_ONLY` The vehicle has all of requested parameters as "read-only" ones for the requested module in the requested location.			`code: 25`	

.

Parameters

Response parameters

Param Name	Type	Mandatory	Additional	Description
`moduleData`	`Common.ModuleData`	`true`	-	The data from the requested module in the requested location.

.

Module:

ModuleData Struct

Param Name	Type	Mandatory	Additional	Description
`moduleType`	`Common.ModuleType`	`true`	-	Defines the module the data is provided for.
`moduleZone`	`Common.InteriorZone`	`true`	-	Defines the zone the data is provided for.
`radioControlData`	`Common.RadioControlData`	`false`	-	Must be provided in case the module is RADIO.
`climateControlData`	`Common.ClimateControlData`	`false`	-	Must be provided in case the module is CLIMATE.

ModuleType Enum

Param Name	Description
`CLIMATE`	Names the Climate module of the vehicle
`RADIO`	Names the Radio module of the vehicle

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

`col`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of column from "column-row-level" grid that defines the location.

`row`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of row from "column-row-level" grid that defines the location.

`level`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of level from "column-row-level" grid that defines the location.

`colspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of columns in "column-row-level" grid that defines the location.

`rowspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of rows in "column-row-level" grid that defines the location.

`levelspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of levels in "column-row-level" grid that defines the location.

Radio:

RadioControlData Struct

Param Name	Type	Mandatory	Additional	Description

`frequencyInteger`	`Integer`	`false`	`minValue = 0`
			`maxValue = 1710`
				The integer part of frequency value (for 101.7 this value should be 101).

`frequencyFraction`	`Integer`	`false`	`minValue = 0`
			`maxValue = 9`
				The fractional part of frequency value (for 101.7 is 7).

band	Common.RadioBand	false	-	The band: AM, FM or XM.
rdsData	Common.RdsData	false	-	RDS (Radio Data System) information.

availableHDs	Integer	false	minValue = 1
			maxValue = 3
				Number of HD sub-channels if available

hdChannel	Integer	false	minValue = 1
			maxValue = 3
				Current HD sub-channel if available.

signalStrength	Integer	false	minValue = 0
			maxValue = 100
				Current frequency signal strength.

signalChangeThreshold	Integer	false	minValue = 0
			maxValue = 100
				If the signal strength falls below the set value for this parameter, the radio will tune to an alternative frequency

radioEnable	Boolean	false	-	True if the radio is on, false is the radio is off.
state	Common.RadioState	false	-	The radio state: acquiring, acquired, multi-cast.

RadioBand Enum

Param Name	Description
AM	AM (amplitude modulation) band
FM	FM (frequency modulation) band
XM	XM Satellite Radio

RdsData Struct

Param Name	Type	Mandatory	Additional	Description

PS	String	false	minlength = 0
			maxlength = 8
				Program Service Name

RT	String	false	minlength = 0
			maxlength = 64
				Radio Text

CT	String	false	minlength = 24
			maxlength = 24
				The clock text in UTC format as YYYY-MM-DDThh:mm:ss.sTZD

PI	String	false	minlength = 0
			maxlength = 6
				Program Identification - the call sign for the radio station

PTY	Integer	false	minValue = 0
			maxValue = 31
				The program type - The region should be used to differentiate between EU and North America program types

TP	Boolean	false	-	Traffic Program Identification - Identifies a station that offers traffic
TA	Boolean	false	-	Traffic Announcement Identification - Indicates an ongoing traffic announcement

REG	String	false	minlength = 0
			maxlength = 255
				Region

RadioState Enum

Param Name	Description
ACQUIRING	-
ACQUIRED	-
MULTICAST	-
NOT_FOUND	-

Climate:

ClimateControlData Struct

Param Name	Type	Mandatory	Additional	Description

fanSpeed	Integer	false	minValue = 0
			maxValue = 100
				The speed of the fan.

currentTemp	Integer	false	minValue = 0
			maxValue = 100
				The temperature currently set on the module.

desiredTemp	Integer	false	minValue = 0
			maxValue = 100
				The temperature to be set on the module (must be used with SetInteriorVehicleData only).

temperatureUnit	Common.TemperatureUnit	false	-	Units the temperature is measured in: Kelvin, Fahrenheit or Celsius
acEnable	Boolean	false	-	If "true" - air conditioning is ON.
circulateAirEnable	Boolean	false	-	If "true" - air circulation is ON.
autoModeEnable	Boolean	false	-	If "true" - the climate system is in auto mode.
defrostZone	Common.DefrostZone	false	-	Defines the defrost zone: front, rear or all.
dualModeEnable	Boolean	false	-	If "true" - the dual mode in ON and driver's and passenger's zones can have separate climate settings.

TemperatureUnit Enum

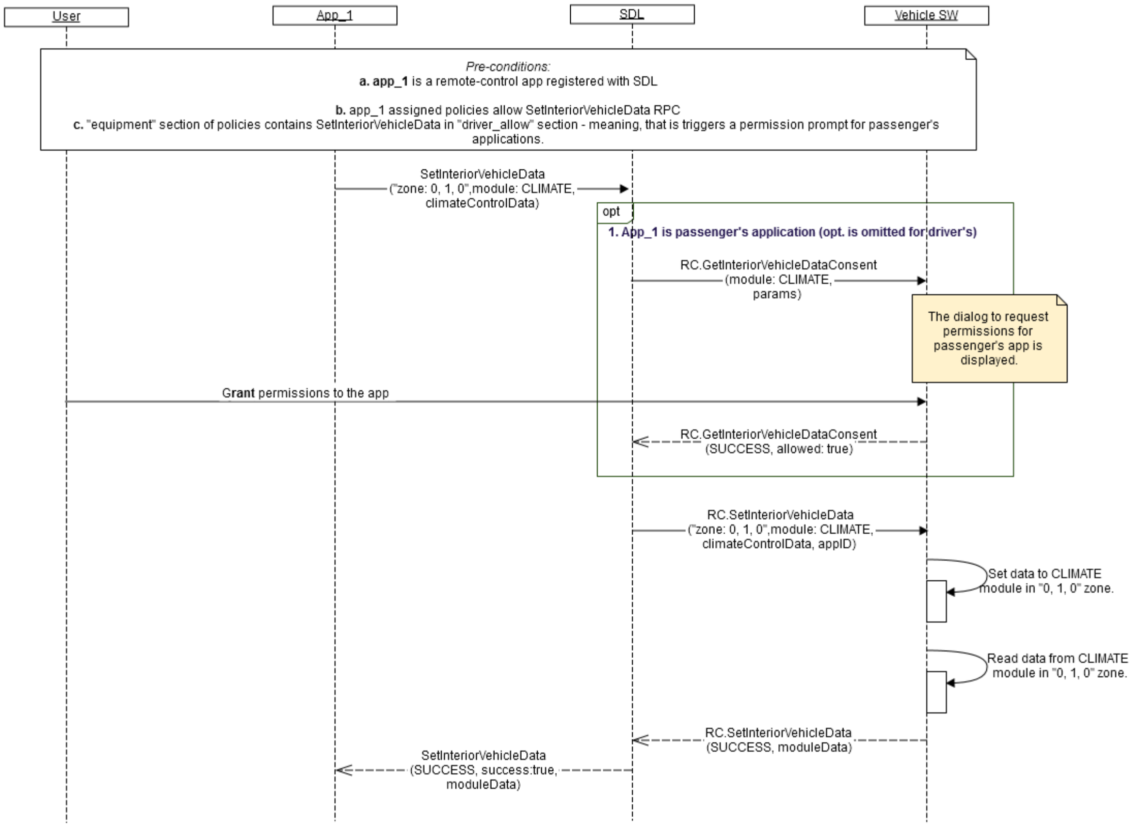
Param Name	Description
KELVIN	-
FAHRENHEIT	-
CELSIUS	-

DefrostZone Enum

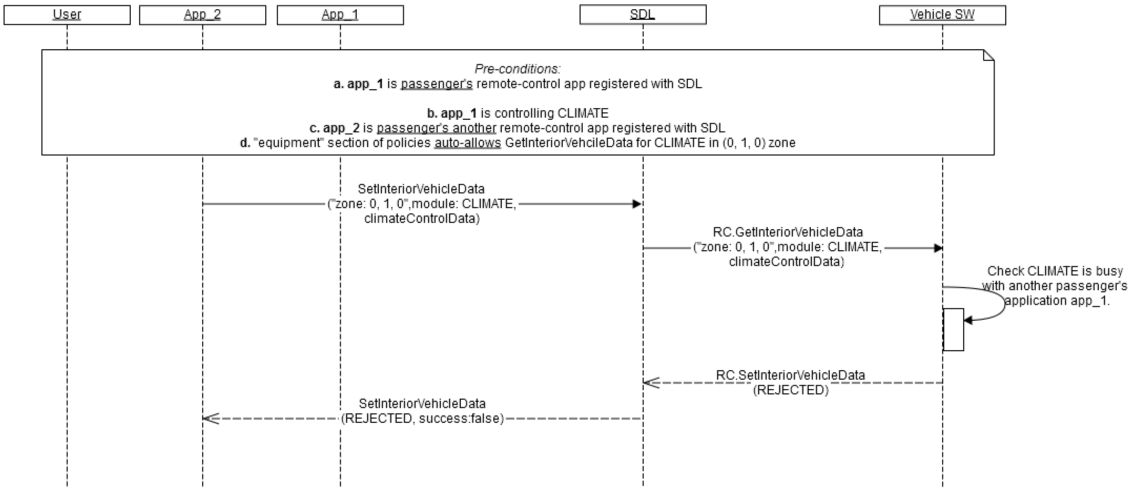
Param Name	Description
FRONT	-
REAR	-
ALL	-

4.4. Sequence Diagrams

A. RSDL transfers vehcile's successful response to the mobile application


User
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is a remote-control app registered with SDL
b. app_1 assigned policies allow SetInteriorVehicleData RPC
c. "equipment" section of policies contains SetInteriorVehicleData in "driver_allow" section - meaning, that is triggers a permission prompt for passenger's applications.
SetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, climateControlData)
opt
1. App_1 is passenger's application (opt. is omitted for driver's)
RC.GetInteriorVehicleDataConsent (module: CLIMATE, params)
The dialog to request permissions for passenger's app is displayed.
Grant permissions to the app
RC.GetInteriorVehicleDataConsent (SUCCESS, allowed: true)
RC.SetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, climateControlData, appID)
Set data to CLIMATE module in "0, 1, 0" zone.
Read data from CLIMATE module in "0, 1, 0" zone.
RC.SetInteriorVehicleData (SUCCESS, moduleData)
SetInteriorVehicleData (SUCCESS, success:true, moduleData)

B. The vehcile rejects the request from another passenger's application for the busy module


User
App_2
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is passenger's remote-control app registered with SDL
b. app_1 is controlling CLIMATE
c. app_2 is passenger's another remote-control app registered with SDL
d. "equipment" section of policies auto-allows GetInteriorVehcileData for CLIMATE in (0, 1, 0) zone
SetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, climateControlData)
RC.GetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, climateControlData)
Check CLIMATE is busy with another passenger's application app_1.
RC.SetInteriorVehicleData (REJECTED)
SetInteriorVehicleData (REJECTED, success:false)

4.5. JSON Messages Examples

Request

```
{
     "id" : 70,
     "jsonrpc" : "2.0",
     "method" : "RC.SetInteriorVehicleData",
     “params” :
     {
            "moduleData" :
            {
                    "moduleType" : "CLIMATE",
                    "moduleZone" :
                    {
                            "col" : 0,
                            "row" : 1,
                            "level" : 0,
                            "colspan" : 2,
                            "rowspan" : 2,
                            "levelspan" : 0
                    },
                    "climateControlData" :
                    {
                            "desiredTemp" : 20,
                            "temperatureUnit" : “CELSIUS”
                    }
            },
            "appID" : 65128
      }
     }
}
```

Response

```
{
     "id" : 70,
     "jsonrpc" : "2.0",
     "result" :
     {
            "moduleData" :
            {
                    "moduleType" : "CLIMATE",
                    "moduleZone" :
                    {
                            "col" : 0,
                            "row" : 1,
                            "level" : 0,
                            "colspan" : 2,
                            "rowspan" : 2,
                            "levelspan" : 0
                    },
                    "climateControlData" :
                    {
                            "fanSpeed" : 40,
                            "currentTemp" : 20,
                            "temperatureUnit" : “CELSIUS”,
                            "acEnable" : true
                    }
            },
      "code" : 0,
      "method" : "RC.SetInteriorVehicleData"
     }
}
```

Error message

```
{
     "id" : 70,
     "jsonrpc" : "2.0",
     "error" :
     {
      "code" : 4,
      "message" : "The request is rejected because the module is busy",
      "data" :
      {
                 "method" : "RC.SetInteriorVehicleData"
      }
     }
}
```

5. Buttons.ButtonPress

5.1. Description

Type:	Function
Sender:	SDL
Purpose:	Press a button remotely

Trigger to send:

RSDL sends `Buttons.ButtonPress` to the vehicle:after receiving a valid `ButtonPress` request from a remote-control application.

Error handling:

1. RSDL validates response from the vehicle and in case it's invalid, RSDL returns `GENERIC_ERROR` result to the corresponding mobile application's request.

   Validation includes:

   - JSON format to be valid.

2. RSDL waits for response from the vehicle during RSDL's internal timeout for vehicle RPCs ("`DefaultTimeout`" parameter in `SmartDeviceLink.ini` file; set to 10 sec by default) and in case it's not received, RSDL returns `GENERIC_ERROR` result to the corresponding mobile application's request.

3. RSDL transfers only button names that match the specified module to the vehicle *(for example, the button `REPEAT` will be sent with module `RADIO` only):*

   Radio buttons: `VOLUME_UP`, `VOLUME_DOWN`, `EJECT`, `SOURCE`, `SHUFFLE`, `REPEAT`

   Climate buttons: `AC_MAX`, `AC`, `RECIRCULATE`, `FAN_UP`, `FAN_DOWN`, `TEMP_UP`, `TEMP_DOWN`, `DEFROST_MAX`, `DEFROST`, `DEFROST_REAR`, `UPPER_VENT`, `LOWER_VENT`

5.2. Request

Behavior:

1. *Happy path* (see diagram A) - the vehicle must:
   1.1. Process the request in the following way:
   a. Send the signal to CAN about the named button is pressed with the named press mode

   1.2. Form a valid JSON response with SUCCESS resultCode

   1.3. Send a response during 10 sec from the time of request receipt.

2. *Module is busy* - the vehicle must:
   2.1. Respond with REJECTED resultCode in case the request is from passenger's application *and* another passenger's application is currently controlling the same module (no matter in what location).
   Information: The vehicle is expected to allow only one passenger's application and one driver's application to control the same module at the same time.

3. *Other result codes* - the vehicle may:
   3.1. Implement additional checks and return the resultCodes described in "applicable result codes" table of Response.

Parameters

Request Parameters

Param Name	Type	Mandatory	Additional	Description
zone	Common.InteriorZone	true	-	The zone where the button press should occur.
moduleType	Common.ModuleType	true	-	Defines the module where the button should be pressed.
buttonName	Common.ButtonName	true	-	Name of the button.
buttonPressMode	Common.ButtonPressMode	true		Indicates whether this is a LONG or SHORT press event.
appID	Integer	true	-	Internal SDL-assigned Id of the application related to this RPC.

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

col	Integer	true	minValue = 0
			maxValue = 100
				The value of column from "column-row-level" grid that defines the location.

row	Integer	true	minValue = 0
			maxValue = 100
				The value of row from "column-row-level" grid that defines the location.

level	Integer	true	minValue = 0
			maxValue = 100
				The value of level from "column-row-level" grid that defines the location.

colspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of columns in "column-row-level" grid that defines the location.

rowspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of rows in "column-row-level" grid that defines the location.

levelspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of levels in "column-row-level" grid that defines the location.

ModuleType Enum

Param Name	Description
CLIMATE	Names the Climate module of the vehicle
RADIO	Names the Radio module of the vehicle

ButtonName Enum

Param Name	Description
AC_MAX	Max mode of Air Conditioning. Climate button.
AC	Air Conditioning. Climate button.
RECIRCULATE	Recirculation mode. Climate button.
FAN_UP	Increment the fan speed. Climate button.
FAN_DOWN	Decrement the fan speed. Climate button.
TEMP_UP	Increment the temperature. Climate button.
TEMP_DOWN	Decrement the temperature. Climate button.
DEFROST_MAX	Max mode of windshield defrost. Climate button.
DEFROST	Windshield defrost. Climate button.
DEFROST_REAR	Rear screen defrost. Climate button.

UPPER_VENT	Upper Vent System mode. Climate button.
LOWER_VENT	Lower Vent System mode. Climate button.
VOLUME_UP	Increment the volume. Radio button.
VOLUME_DOWN	Decrement the volume. Radio button.
EJECT	Eject. Radio button.
SOURCE	Source. Radio button.
SHUFFLE	Shuffle. Radio button.
REPEAT	Repeat. Radio button.

ButtonPressMode Enum

Param Name	Description
LONG	A button was released, after it was pressed for a long time. Actual timing is defined by head unit and may vary
SHORT	A button was released, after it was pressed for a short time. Actual timing is defined by head unit and may vary

5.3. Response

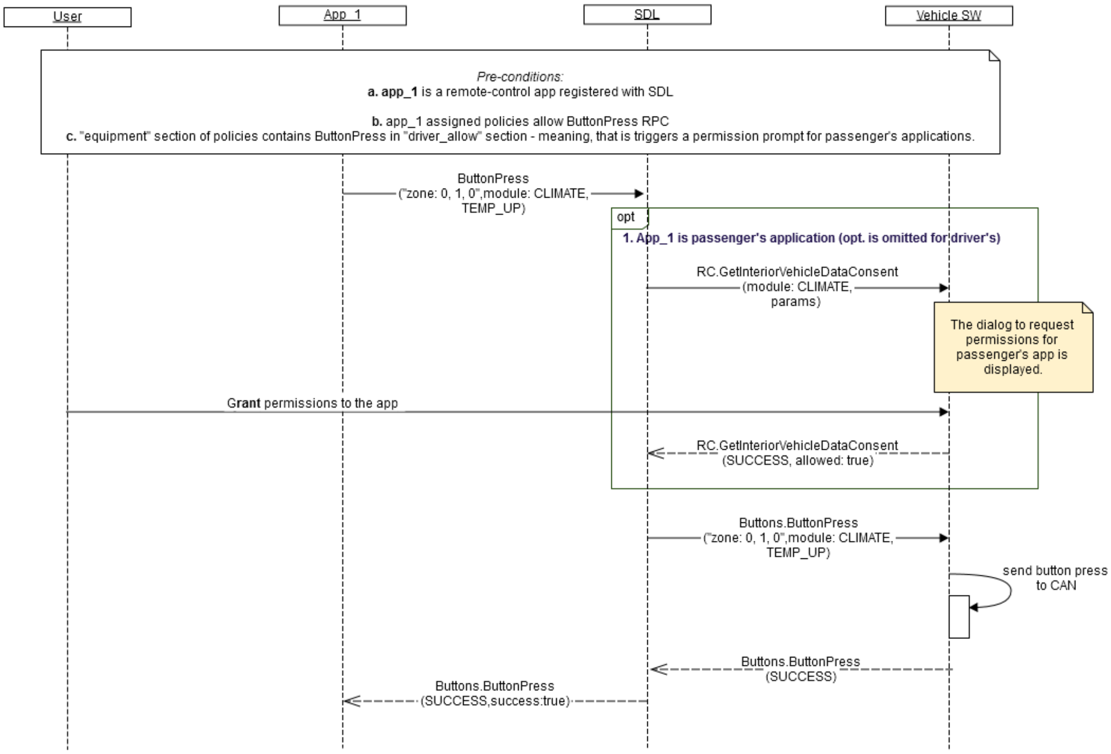
Behavior:
The below table lists resultCodes that the vehicle may use to respond:

Result	Description	Message type		Message Params	Notes
		WebSocket	D-Bus		
Success	SUCCESS The vehicle successfully processes the request: - The requested button press was sent to the CAN bus.	JSON response	Regular response	moduleData, isSubscribed, code: 0	See example "response".
Failure	UNSUPPORTED_RESOURCE The vehicle does not support the requested button.	JSON error message	Regular response	code: 2	Applicable for this RPC result codes. Please see Result Enumeration for all SDL-supported codes. See example "error message".
	REJECTED The vehicle rejects GetInteriorVehicleData request because of - Other tasks of higher priority - The requested module is busy with another passenger's application.			code: 4	
	IGNORED The vehicle ignores the repeated request.			code: 6	
	GENERIC_ERROR The unknown issue occurred or other codes are not applicable.			code: 22	

.

5.4. Sequence Diagrams

A. RSDL transfers vehcile's successful response to the mobile application


User
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is a remote-control app registered with SDL
b. app_1 assigned policies allow ButtonPress RPC
c. "equipment" section of policies contains ButtonPress in "driver_allow" section - meaning, that is triggers a permission prompt for passenger's applications.
ButtonPress ("zone: 0, 1, 0",module: CLIMATE, TEMP_UP)
opt
1. App_1 is passenger's application (opt. is omitted for driver's)
RC.GetInteriorVehicleDataConsent (module: CLIMATE, params)
The dialog to request permissions for passenger's app is displayed.
Grant permissions to the app
RC.GetInteriorVehicleDataConsent (SUCCESS, allowed: true)
Buttons.ButtonPress ("zone: 0, 1, 0",module: CLIMATE, TEMP_UP)
send button press to CAN
Buttons.ButtonPress (SUCCESS)
Buttons.ButtonPress (SUCCESS,success:true)

5.5. JSON Messages Examples

Request

```
{
     "id" : 89,
     "jsonrpc" : "2.0",
     "method" : "Buttons.ButtonPress",
     “params” :
     {
      "zone" :
      {
                  "col" : 0,
                  "row" : 1,
                  "level" : 0,
                  "colspan" : 2,
                  "rowspan" : 2,
                  "levelspan" : 0
      },
      "moduleType" : "CLIMATE",
      "buttonName" : "TEMP_DOWN",
      "buttonPressMode" : "SHORT",
      "appID" : 65128
     }
}
```

Response

```
{
     "id" : 89,
     "jsonrpc" : "2.0",
     "result" :
     {
      "code" : 0,
      "method" : "Buttons.ButtonPress"
     }
}
```

Error message

```
{
     "id" : 89,
     "jsonrpc" : "2.0",
     "error" :
     {
      "code" : 4,
      "message" : "The request is rejected because the module is busy",
      "data" :
      {
                 "method" : "Buttons.ButtonPress"
      }
     }
}
```

6. RC.GetInteriorVehicleDataConsent

6.1. Description

Type:	Function
Sender:	SDL
Purpose:	Obtain driver's permission for the passenger's application to control the specified module in the specified zone.

Trigger to send:

RSDL sends `RC.GetInteriorVehicleDataConsent` to the vehicle after receiving a valid remote-control request from passenger's application that requires driver's approval.

Error handling:

1. RSDL validates response from the vehicle and in case it's invalid, RSDL returns `GENERIC_ERROR` result to the corresponding mobile application's request.

   Validation includes:

   - Mandatory parameters to be present

   - Type of parameters values to correspond HMI_API

   - Parameters values to be in bounds with HMI_API

   - JSON format to be valid.

2. RSDL waits for response from the vehicle during RSDL's internal timeout for vehicle RPCs ("`DefaultTimeout`" parameter in `SmartDeviceLink.ini` file; set to 10 sec by default) and in case it's not received, RSDL treats this as driver's denial..

6.2. Request

Behavior:

1. *Happy path* (see diagram A) - the vehicle must:
   1.1. Process the request (see example "request") in the following way:
   a. Display the permission prompt to the driver based on parameters from request and with the possibility to either confirm or reject the prompt.

   1.2. Form a valid JSON response (see example "response") with
   a. "`allowed: true`" - *in case* the driver confirms the prompt (grant the permission to the specified application)
   b. "`allowed: false`" - *in case* the driver rejects the prompt (denies the permission to the specified application)

   1.3. Send a response during 10 sec from the time of request receipt.

   Note: In case the vehicle needs to display the permission prompt longer than 10 sec, it may renew SDL's default timeout for vehicle RPCs via `BasicCommunication.OnResetTimeout` notification.

2. *Safety* - the vehicle must:
   2.1. Respond with REJECTED resultCode in case the `RC.GetInteriorVehicleDataConsent`
   a. Causes driver distraction
   b. Comes again from the same application after driver's denial
   c. Comes for controlling the module already busy with another passenger's application
   Information: The vehicle is expected to allow only one passenger's application and one driver's application to control the same module at the same time.

3. *Other result codes* - the vehicle may:
   3.1. Implement additional checks and return the resultCodes described in "applicable result codes" table of Response.

Parameters

Request Parameters

Param Name	Type	Mandatory	Additional	Description
`moduleType`	`Common.ModuleType`	`true`	-	The module that the app requests to control
`zone`	`Common.InteriorZone`	`true`	-	The zone in which the app requests to control
`appID`	`Integer`	`true`	-	Internal SDL-assigned Id of the application related to this RPC.

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

`col`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of column from "column-row-level" grid that defines the location.

`row`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of row from "column-row-level" grid that defines the location.

`level`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of level from "column-row-level" grid that defines the location.

`colspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of columns in "column-row-level" grid that defines the location.

`rowspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of rows in "column-row-level" grid that defines the location.

`levelspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of levels in "column-row-level" grid that defines the location.

ModuleType Enum

Param Name	Description
`CLIMATE`	Names the Climate module of the vehicle
`RADIO`	Names the Radio module of the vehicle

6.3. Response

Behavior:
The below table lists resultCodes that the vehicle may use to respond:

Result	Description	Message type		Message Params	Notes
		WebSocket	D-Bus		

Success	`SUCCESS`
	The vehicle successfully processes the request:
	- Provides the data for the requested module in the requested location
	- Subscribes or unsubscribes the application from notifications if requested.
		JSON response	Regular response	`allowed,`
				`code: 0`
					See example "response".

Failure	`UNSUPPORTED_RESOURCE` The vehicle does not support the requested module in the requested location.	JSON error message	Regular response	`code: 2`	Applicable for this RPC result codes. Please see `Result` Enumeration for all SDL-supported codes.

	`REJECTED` The vehicle rejects GetInteriorVehicleData request because of - Other tasks of higher priority - Safety rules in effect - The requested module is busy with another passenger's application.			`code: 4`	See example "error message".
	`GENERIC_ERROR` The unknown issue occurred or other codes are not applicable.			`code: 22`	

.

Parameters

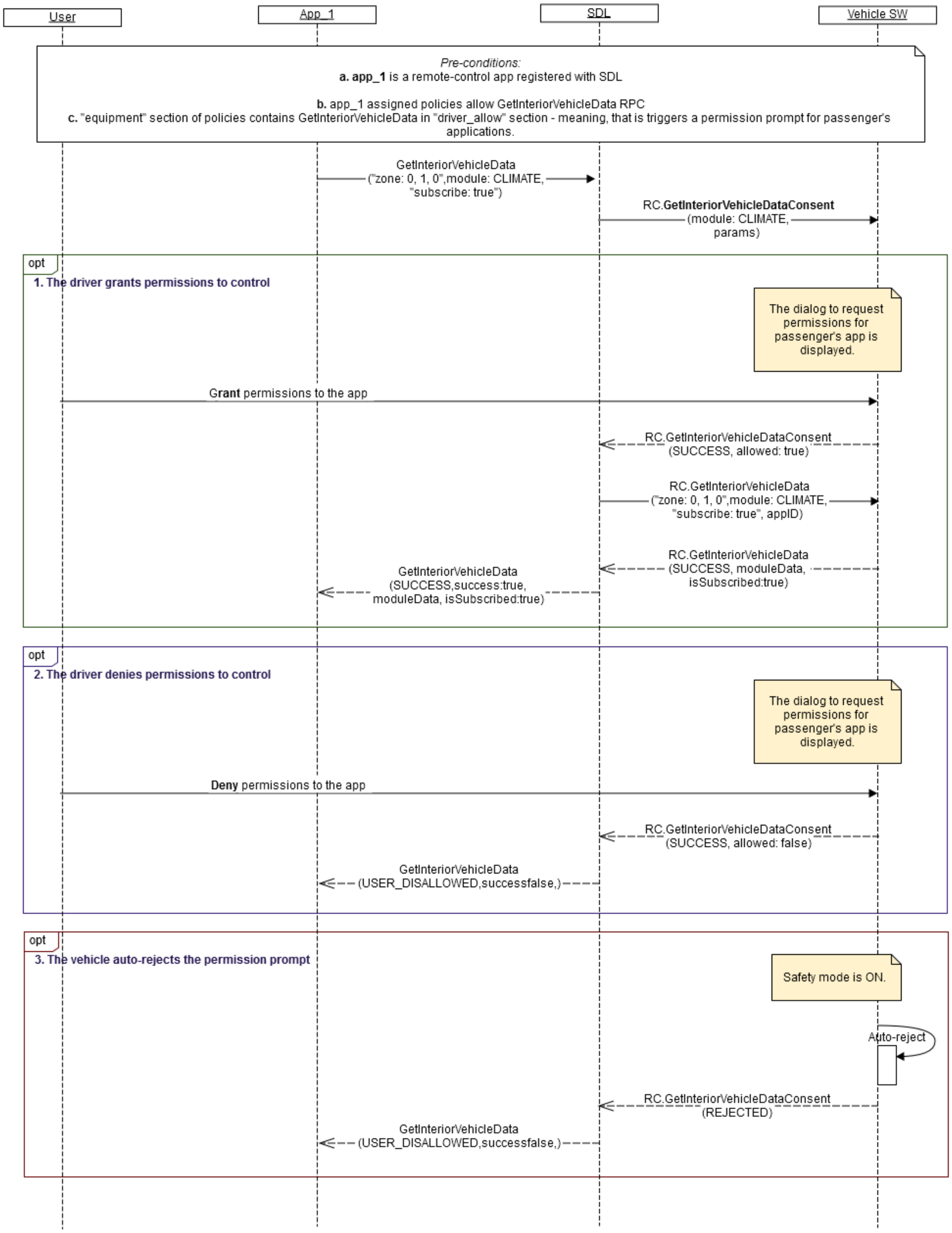
Response parameters

Param Name	Type	Mandatory	Additional	Description
`allowed`	`Boolean`	`true`	-	"true" - if the driver grants the permission for controlling to the named app; "false" - in case the driver denies the permission for controlling to the named app.

.

6.4. Sequence Diagrams

A. GetInteriorVehicleDataConsent - vehcile response cases


User
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is a remote-control app registered with SDL
b. app_1 assigned policies allow GetInteriorVehicleData RPC
c. "equipment" section of policies contains GetInteriorVehicleData in "driver_allow" section - meaning, that is triggers a permission prompt for passenger's applications.
GetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, "subscribe: true")
RC.GetInteriorVehicleDataConsent (module: CLIMATE, params)
opt
1. The driver grants permissions to control
The dialog to request permissions for passenger's app is displayed.
Grant permissions to the app
RC.GetInteriorVehicleDataConsent (SUCCESS, allowed: true)
RC.GetInteriorVehicleData ("zone: 0, 1, 0",module: CLIMATE, "subscribe: true", appID)
RC.GetInteriorVehicleData (SUCCESS, moduleData, isSubscribed:true)
GetInteriorVehicleData (SUCCESS,success:true, moduleData, isSubscribed:true)
opt
2. The driver denies permissions to control
The dialog to request permissions for passenger's app is displayed.
Deny permissions to the app
RC.GetInteriorVehicleDataConsent (SUCCESS, allowed: false)
GetInteriorVehicleData (USER_DISALLOWED,successfalse,)
opt
3. The vehicle auto-rejects the permission prompt
Safety mode is ON.
Auto-reject
RC.GetInteriorVehicleDataConsent (REJECTED)
GetInteriorVehicleData (USER_DISALLOWED,successfalse,)

6.5. JSON Messages Examples

Request

```
{
     "id" : 76,
     "jsonrpc" : "2.0",
     "method" : "RC.GetInteriorVehicleDataConsent",
     “params” :
     {
      "moduleZone" :
      {
                 "col" : 0,
                 "row" : 1,
                 "level" : 0,
                 "colspan" : 2,
                 "rowspan" : 2,
                 "levelspan" : 0
      },
      "moduleType" : "CLIMATE",
      "appID" : 65128
     }
}
```

Response

```
{
     "id" : 76,
     "jsonrpc" : "2.0",
     "result" :
     {
            "allowed" : true
     },
      "code" : 0,
      "method" : "RC.GetInteriorVehicleDataConsent"
     }
}
```

Error message

```
{
     "id" : 76,
     "jsonrpc" : "2.0",
     "error" :
     {
      "code" : 4,
      "message" : "The request is rejected because the module is busy",
      "data" :
      {
                 "method" : "RC.GetInteriorVehicleDataConsent"
      }
     }
}
```

7. RC.OnInteriorVehicleData

7.1. Description

Type:	Function
Sender:	Vehicle
Purpose:	Notify the application subscribed to the specified module in the specified location about the data change.

Error handling:

1. RSDL validates notification from the vehicle and in case it's invalid, RSDL discards it (not transfer it to the specified mobile application).

   Validation includes:

   - Mandatory parameters to be present

   - Type of parameters values to correspond HMI_API

   - Parameters values to be in bounds with HMI_API

   - JSON format to be valid.

   - Module and ControlData to match *(for example, notification with "`RadioControlData`" sent together with `CLIMATE moduleType` is invalid)*

7.2. Behavior

1. The vehicle must (see diagram A):
   1.1. Monitor the data change of the module in the location that was subscribed by the application (see RC.GetInteriorVehicleData for details about subscription)
   1.2. Form a valid JSON notification with the changed data put as parameters values and with the `appID` of the subscribed application (see "parameters" below and example "notification")
   1.3. Send the notification to RSDL.

<u>*Information:*</u>

a. The vehicle is expected to send `OnInteriorVehicleData` notifications <u>only</u> in case there is at least one application currently subscribed via `RC.GetInteriorVehcileData`.

Parameters

Param Name	Type	Mandatory	Additional	Description
`moduleData`	`Common.ModuleData`	`true`	-	Data changed in the specified module in the specified location

.

Module:

ModuleData Struct

Param Name	Type	Mandatory	Additional	Description
`moduleType`	`Common.ModuleType`	`true`	-	Defines the module the data is provided for.
`moduleZone`	`Common.InteriorZone`	`true`	-	Defines the zone the data is provided for.
`radioControlData`	`Common.RadioControlData`	`false`	-	Must be provided in case the module is RADIO.
`climateControlData`	`Common.ClimateControlData`	`false`	-	Must be provided in case the module is CLIMATE.

ModuleType Enum

Param Name	Description
CLIMATE	Names the Climate module of the vehicle
RADIO	Names the Radio module of the vehicle

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

col	Integer	true	minValue = 0
			maxValue = 100
				The value of column from "column-row-level" grid that defines the location.

row	Integer	true	minValue = 0
			maxValue = 100
				The value of row from "column-row-level" grid that defines the location.

level	Integer	true	minValue = 0
			maxValue = 100
				The value of level from "column-row-level" grid that defines the location.

colspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of columns in "column-row-level" grid that defines the location.

rowspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of rows in "column-row-level" grid that defines the location.

levelspan	Integer	true	minValue = 0
			maxValue = 100
				Total number of levels in "column-row-level" grid that defines the location.

Radio:

RadioControlData Struct

Param Name	Type	Mandatory	Additional	Description

frequencyInteger	Integer	false	minValue = 0
			maxValue = 1710
				The integer part of frequency value (for 101.7 this value should be 101).

frequencyFraction	Integer	false	minValue = 0
			maxValue = 9
				The fractional part of frequency value (for 101.7 is 7).

band	Common.RadioBand	false	-	The band: AM, FM or XM.
rdsData	Common.RdsData	false	-	RDS (Radio Data System) information. Read-only parameter.

availableHDs	Integer	false	minValue = 1
			maxValue = 3
				Number of HD sub-channels if available. Read-only parameter.

hdChannel	Integer	false	minValue = 1
			maxValue = 3
				Current HD sub-channel if available.

signalStrength	Integer	false	minValue = 0
			maxValue = 100
				Current frequency signal strength. Read-only parameter.

signalChangeThreshold	Integer	false	minValue = 0
			maxValue = 100
				If the signal strength falls below the set value for this parameter, the radio will tune to an alternative frequency. Read-only parameter.

radioEnable	Boolean	false	-	True if the radio is on, false is the radio is off. Read-only parameter.
state	Common.RadioState	false	-	The radio state: acquiring, acquired, multi-cast. Read-only parameter.

RadioBand Enum

Param Name	Description
AM	AM (amplitude modulation) band
FM	FM (frequency modulation) band
XM	XM Satellite Radio

RdsData Struct

Param Name	Type	Mandatory	Additional	Description

PS	String	false	minlength = 0
			maxlength = 8
				Program Service Name

RT	String	false	minlength = 0
			maxlength = 64
				Radio Text

CT	String	false	minlength = 24
			maxlength = 24
				The clock text in UTC format as YYYY-MM-DDThh:mm:ss.sTZD

PI	String	false	minlength = 0
			maxlength = 6
				Program Identification - the call sign for the radio station

PTY	Integer	false	minValue = 0
			maxValue = 31
				The program type - The region should be used to differentiate between EU and North America program types

TP	Boolean	false	-	Traffic Program Identification - Identifies a station that offers traffic
TA	Boolean	false	-	Traffic Announcement Identification - Indicates an ongoing traffic announcement

REG	String	false	minlength = 0
			maxlength = 255
				Region

RadioState Enum

Param Name	Description
ACQUIRING	-
ACQUIRED	-
MULTICAST	-
NOT_FOUND	-

Climate:

ClimateControlData Struct

Param Name	Type	Mandatory	Additional	Description

`fanSpeed`	`Integer`	`false`	`minValue = 0`
			`maxValue = 100`
				The speed of the fan.

`currentTemp`	`Integer`	`false`	`minValue = 0`
			`maxValue = 100`
				The temperature currently set on the module. Read-only parameter.

`desiredTemp`	`Integer`	`false`	`minValue = 0`
			`maxValue = 100`
				The temperature to be set on the module (must be used with `SetInteriorVehicleData` only).

`temperatureUnit`	`Common.TemperatureUnit`	`false`	-	Units the temperature is measured in: Kelvin, Fahrenheit or Celsius
`acEnable`	`Boolean`	`false`	-	If "true" - air conditioning is ON.
`circulateAirEnable`	`Boolean`	`false`	-	If "true" - air circulation is ON.
`autoModeEnable`	`Boolean`	`false`	-	If "true" - the climate system is in auto mode.
`defrostZone`	`Common.DefrostZone`	`false`	-	Defines the defrost zone: front, rear or all.
`dualModeEnable`	`Boolean`	`false`	-	If "true" - the dual mode in ON and driver's and passenger's zones can have separate climate settings.

TemperatureUnit Enum

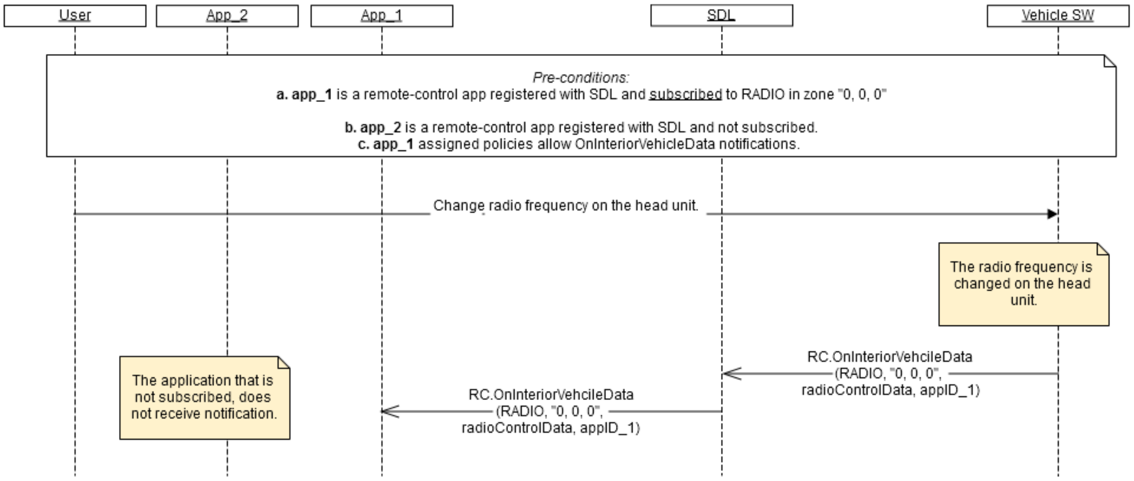
Param Name	Description
`KELVIN`	-
`FAHRENHEIT`	-
`CELSIUS`	-

DefrostZone Enum

Param Name	Description
`FRONT`	-
`REAR`	-
`ALL`	-

7.3. Sequence Diagrams

A. OnInteriorVehicleData notification is sent upon data change


User
App_2
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is a remote-control app registered with SDL and subscribed to RADIO in zone "0, 0, 0"
b. app_2 is a remote-control app registered with SDL and not subscribed.
c. app_1 assigned policies allow OnInteriorVehicleData notifications.
Change radio frequency on the head unit.
The radio frequency is changed on the head unit.
RC.OnInteriorVehcileData (RADIO, "0, 0, 0", radioControlData, appID_1)
The application that is not subscribed, does not receive notification.
RC.OnInteriorVehcileData (RADIO, "0, 0, 0", radioControlData, appID_1)

7.4. JSON Messages Examples

Notification

```
{
     "jsonrpc" : "2.0",
     "method" : "RC.OnInteriorVehicleData",
     “params” :
     {
            "moduleData" :
            {
                    "moduleType" : "RADIO",
                    "moduleZone" :
                    {
                            "col" : 0,
                            "row" : 0,
                            "level" : 0,
                            "colspan" : 2,
                            "rowspan" : 2,
                            "levelspan" : 0
                    },
                    "radioControlData" :
                    {
                            "frequencyInteger" : 105,
                            "frequencyFraction" : 3
                    }
            },
            "appID" : 65128
      }
     }
}
```

8. RC.OnReverseAppsAllowing

8.1. Description

Type:	Function
Sender:	Vehicle
Purpose:	Notify the remote-control for passenger's applications is either turned off or on.

Error handling:

1. RSDL validates notification from the vehicle and in case it's invalid, RSDL discards it (does not apply in its internal logic).

   Validation includes:

   - Mandatory parameters to be present

   - Type of parameters values to correspond HMI_API

   - JSON format to be valid.

Defaults:

1. By default (until `OnReverseAppsAllowing(false)` is received from the vehicle) RSDL allows RC for passenger's applications.
2. RSDL stores driver's settings notified via `OnReverseAppsAllowing` through ignition cycles.

Usage:

By `RC.OnReverseAppsAllowing` notification from the vehicle, RSDL:

1. Re-assigns policies to the passenger's application(s):
   1.a. the very limited set of permissions is assigned (that allow notifications from RSDL only) in case the parameter value is "`false`"
   1.b. normal set of permissions (obtained from the cloud) is assigned in case the parameter value is "`true`"
2. Resets passenger's application(s) subscription status (unsubscribes in case the related application(s) was subscribed. See `RC.GetInteriorVehicleData`) in case the parameter value is "`false`".

8.2. Behavior

1. The vehicle must (see diagram A):
   1.1. Implement the command that allows the driver to either turn off or turn on the Remote-Control for passenger's applications through the Settings Menu (UI or VR or both)
   1.2. After the above command is recognized, form a valid JSON notification with parameter value that corresponds driver's choice (see "parameters" below and example "notification")
   1.3. Send the notification to RSDL.

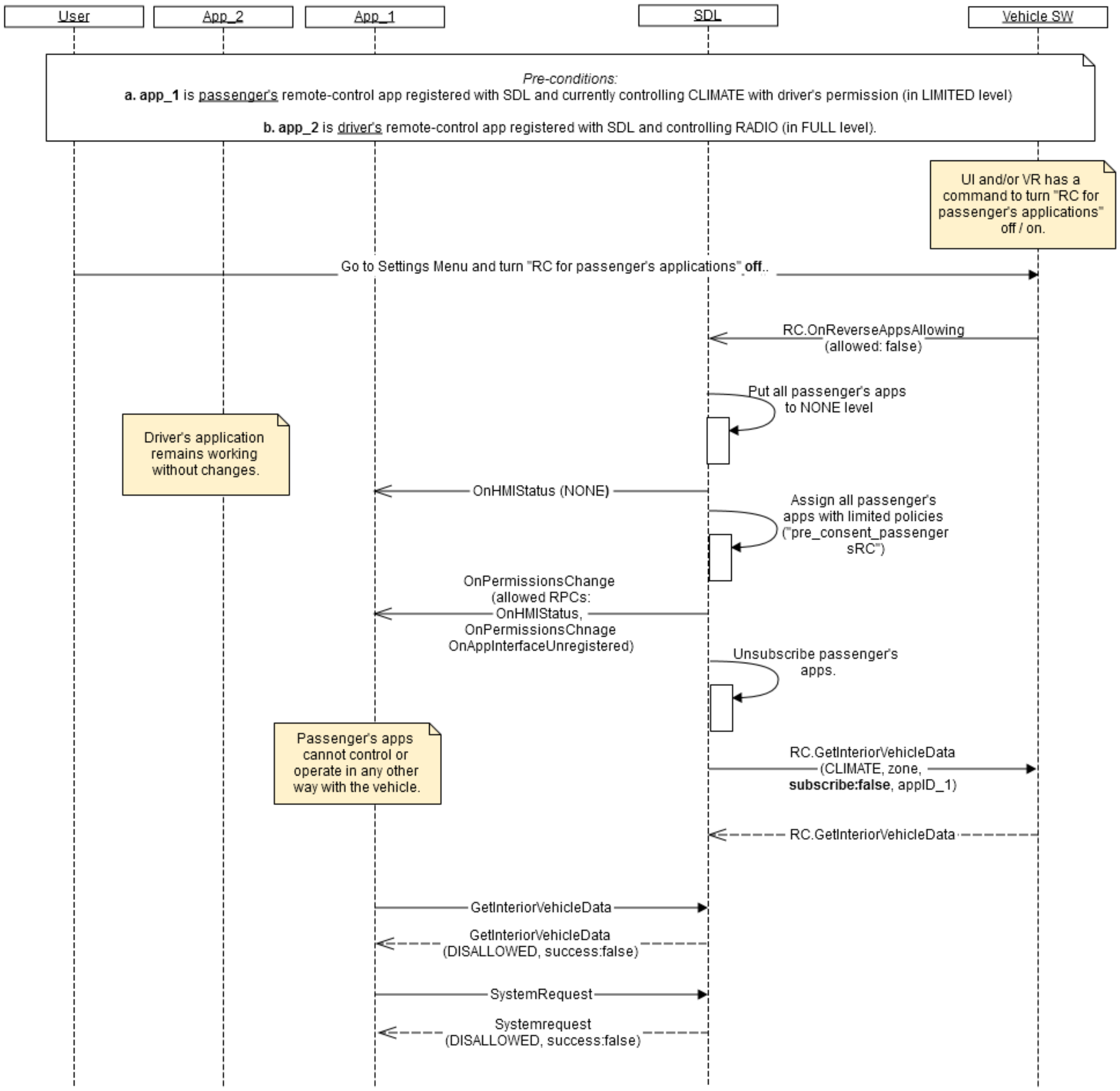
Parameters

Param Name	Type	Mandatory	Additional	Description
`allowed`	`Boolean`	`true`	-	If "true" - remote-control for passenger's applications is allowed. If "false" - disallowed.

.

8.3. Sequence Diagrams

A. Disable remote-control for passenger's applications


User
App_2
App_1
SDL
Vehicle SW
Pre-conditions:
a. app_1 is passenger's remote-control app registered with SDL and currently controlling CLIMATE with driver's permission (in LIMITED level)
b. app_2 is driver's remote-control app registered with SDL and controlling RADIO (in FULL level).
UI and/or VR has a command to turn "RC for passenger's applications" off / on.
Go to Settings Menu and turn "RC for passenger's applications" off..
RC.OnReverseAppsAllowing (allowed: false)
Put all passenger's apps to NONE level
Driver's application remains working without changes.
OnHMIStatus (NONE)
Assign all passenger's apps with limited policies ("pre_consent_passenger sRC")
OnPermissionsChange (allowed RPCs: OnHMIStatus, OnPermissionsChnage OnAppInterfaceUnregistered)
Unsubscribe passenger's apps.
Passenger's apps cannot control or operate in any other way with the vehicle.
RC.GetInteriorVehicleData (CLIMATE, zone, subscribe:false, appID_1)
RC.GetInteriorVehicleData
GetInteriorVehicleData
GetInteriorVehicleData (DISALLOWED, success:false)
SystemRequest
Systemrequest (DISALLOWED, success:false)

8.4. JSON Messages Examples

Notification

```
{
     "jsonrpc" : "2.0",
     "method" : "RC.OnReverseAppsAllowing",
     “params” :
     {
            "allowed" : false
     }
}
```

9. RC.OnDeviceRankChanged

9.1. Description

Type:	Function
Sender:	Vehicle
Purpose:	Notify the connected device's rank is changed to either driver's or passenger's.

Error handling:

1. RSDL validates notification from the vehicle and in case it's invalid, RSDL discards it (does not apply in its internal logic).

   Validation includes:

   - Mandatory parameters to be present

   - Type of parameters values to correspond HMI_API

   - JSON format to be valid.

Defaults:

1. By default (until `OnDeviceRankChanged(DRIVER)` is received from the vehicle) RSDL treats any connected device as passenger's.
2. RSDL does not store device rank notified via `OnDeviceRankChanged` through ignition cycles (meaning, the device ranked as "driver's" in the previous ignition cycle will be defaulted to "passenger's" in the next ignition cycle by RSDL).
3. RSDL allows only one device to be "driver's" at a time (meaning, current "driver's" device will start to be treated as passenger's by RSDL in case the vehicle sends `OnDeviceRankChanged(DRIVER)` for another device).

Usage:

By `RC.OnDeviceRank` notification from the vehicle, RSDL:

1. Re-assigns policies to the related applications:
   -> the same application will get different set of permissions depending on whether the related device is ranked as "driver's" or as "passenger's"
2. Sends `OnHMIStatus("deviceRank")` notification to the related applications
3. Resets application's subscription status (unsubscribes in case the related application(s) was subscribed. See RC.GetInteriorVehicleData)

9.2. Behavior

1. The vehicle must (see diagram A):
   1.1. Implement the command that allows the driver to rank a device to either "driver's" or "passenger's" through the Settings Menu (UI or VR or both).
   1.2. Form a valid JSON notification with parameters values that corresponds driver's choice (see "parameters" below and example "notification") in case:
   - the above command is recognized
   - the vehicle receives `UpdateDeviceList` with `name` and `id` of the device that is stored to be "driver's" per earlier choice.

   1.3. Send the notification to RSDL.

2. The vehicle may:
   2.1. Avoid sending notification - in case the device rank of connected device is remembered as passenger's by the vehicle settings.

Parameters

Param Name	Type	Mandatory	Additional	Description

deviceRank	Common.DeviceRank	true	-	If the value is "DRIVER" - the named by DeviceInfo device is set as driver's. If the value is "PASSENGER" - the named by DeviceInfo device is set as passenger's.
device	Common.DeviceInfo	true	-	The device info. Initially is sent by SDL via UpdateDeviceList to the vehicle. Only name and id must be provided.

.

DeviceRank Enum

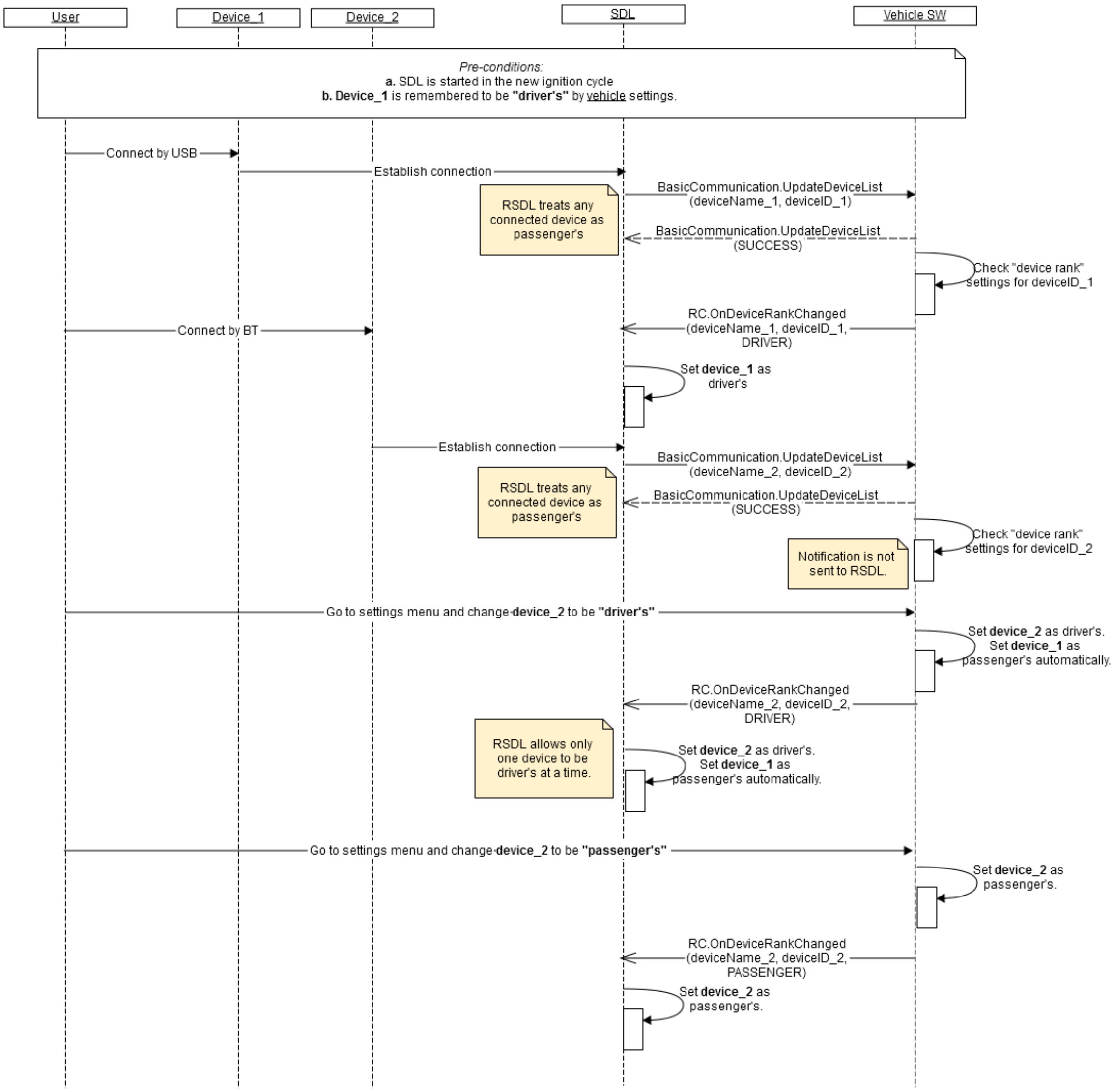
Param Name	Description
DRIVER	The device is ranked as driver's
PASSENGER	The device is ranked as passenger's

DeviceInfo Struct

Param Name	Type	Mandatory	Additional	Description
name	String	true	maxlength = 500	The name of the device connected.
id	Integer	true	-	The ID of the device connected.
transportType	Common.TransportType	false	-	The transport type the named-app's-device is connected over HU(BlueTooth, USB or WiFi). It must be provided in OnAppRegistered and in UpdateDeviceList
isSDLAllowed	Boolean	false	-	Sent by SDL in UpdateDeviceList. 'true' – if device is allowed for PolicyTable Exchange; 'false' – if device is NOT allowed for PolicyTable Exchange

9.3. Sequence Diagrams

A. OnDeviceRankChanged upon vehicle settings and upon driver's command


User
Device_1
Device_2
SDL
Vehicle SW
Pre-conditions:
a. SDL is started in the new ignition cycle
b. Device_1 is remembered to be "driver's" by vehicle settings.
Connect by USB
Establish connection
BasicCommunication.UpdateDeviceList
(deviceName_1, deviceID_1)
RSDL treats any connected device as passenger's
BasicCommunication.UpdateDeviceList
(SUCCESS)
Check "device rank" settings for deviceID_1
RC.OnDeviceRankChanged
(deviceName_1, deviceID_1,
DRIVER)
Connect by BT
Set device_1 as driver's
Establish connection
BasicCommunication.UpdateDeviceList
(deviceName_2, deviceID_2)
RSDL treats any connected device as passenger's
BasicCommunication.UpdateDeviceList
(SUCCESS)
Check "device rank" settings for deviceID_2
Notification is not sent to RSDL.
Go to settings menu and change device_2 to be "driver's"
Set device_2 as driver's.
Set device_1 as passenger's automatically.
RC.OnDeviceRankChanged
(deviceName_2, deviceID_2,
DRIVER)
RSDL allows only one device to be driver's at a time.
Set device_2 as driver's.
Set device_1 as passenger's automatically.
Go to settings menu and change device_2 to be "passenger's"
Set device_2 as passenger's.
RC.OnDeviceRankChanged
(deviceName_2, deviceID_2,
PASSENGER)
Set device_2 as passenger's.

9.4. JSON Messages Examples

Notification

```
{
     "jsonrpc" : "2.0",
     "method" : "RC.OnDeviceRankChanged",
     “params” :
     {
            "deviceRank" : "DRIVER",
            "device" :
            {
                   "name" : "Paul's phone",
                   "id" : 123456
            }
     }
}
```

10. RC.OnDeviceLocationChanged

10.1. Description

Type:	Function
Sender:	Vehicle
Purpose:	Notify the connected device's location is changed.

Error handling:

1. RSDL validates notification from the vehicle and in case it's invalid, RSDL discards it (does not apply in its internal logic).

   Validation includes:

   - Mandatory parameters to be present

   - Type of parameters values to correspond HMI_API

   - Parameters values to be in bounds with HMI_API

   - JSON format to be valid.

Defaults:

1. By default (until `OnDeviceLocationChanged()` is received from the vehicle) RSDL treats the location provided by the application via one of remote-control RPCs as device location.

Usage:

By `RC.OnDeviceLocationChanged` notification from the vehicle, RSDL:

1. Starts checking the "permission prompt triggers" for the applications from the named passenger's device against the location from notification (See also RC.GetInteriorVehicleDataConsent)
2. Resets application's subscription status (unsubscribes in case the related application(s) was subscribed. See RC.GetInteriorVehicleData)

10.2. Behavior

1. The vehicle must (see diagram A):
   1.1. Implement the mechanism to define the mobile device's location in the vehicle interior..
   1.2. Form a valid JSON notification with parameters values that correspond the device location (see "parameters" below and example "notification") in case:
   - the vehicle receives `UpdateDeviceList` with `name` and `id` of connected device
   - the location of the known (via `UpdateDeviceList`) device is changed.
   1.3. Send the notification to RSDL.

Parameters

Param Name	Type	Mandatory	Additional	Description
`deviceLocation`	`Common.InteriorZone`	`true`	-	If the value is "`DRIVER`" - the named by `DeviceInfo` device is set as driver's. If the value is "`PASSENGER`" - the named by `DeviceInfo` device is set as passenger's.
`device`	`Common.DeviceInfo`	`true`	-	The device info. Initially is sent by SDL via `UpdateDeviceList` to the vehicle. Only `name` and `id` must be provided.

.

InteriorZone Struct

Param Name	Type	Mandatory	Additional	Description

`col`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of column from "column-row-level" grid that defines the location.

`row`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of row from "column-row-level" grid that defines the location.

`level`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				The value of level from "column-row-level" grid that defines the location.

`colspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of columns in "column-row-level" grid that defines the location.

`rowspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of rows in "column-row-level" grid that defines the location.

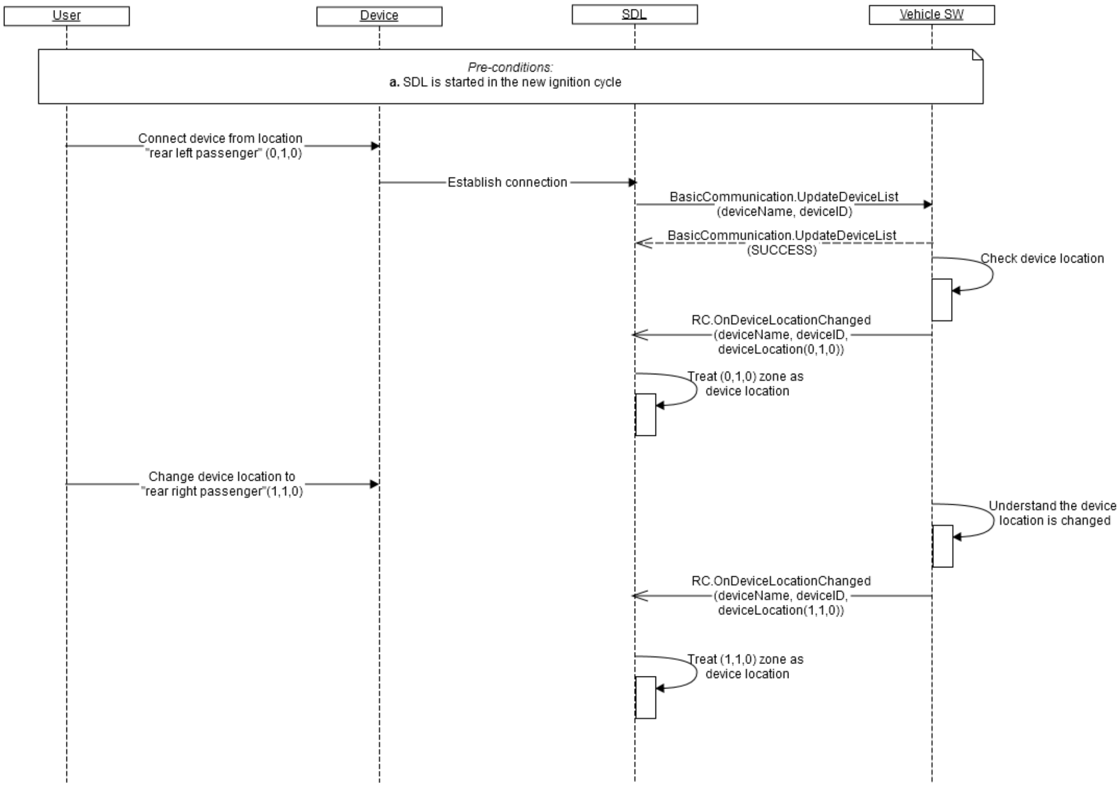
`levelspan`	`Integer`	`true`	`minValue = 0`
			`maxValue = 100`
				Total number of levels in "column-row-level" grid that defines the location.

DeviceInfo Struct

Param Name	Type	Mandatory	Additional	Description
`name`	`String`	`true`	`maxlength = 500`	The name of the device connected.
`id`	`Integer`	`true`	-	The ID of the device connected.
`transportType`	`Common.TransportType`	`false`	-	The transport type the named-app's-device is connected over HU(BlueTooth, USB or WiFi). It must be provided in OnAppRegistered and in UpdateDeviceList
`isSDLAllowed`	`Boolean`	`false`	-	Sent by SDL in UpdateDeviceList. 'true' – if device is allowed for PolicyTable Exchange; 'false' – if device is NOT allowed for PolicyTable Exchange

10.3. Sequence Diagrams

A. OnDeviceRankChanged upon vehicle settings and upon driver's command


User
Device
SDL
Vehicle SW
Pre-conditions:
a. SDL is started in the new ignition cycle
Connect device from location "rear left passenger" (0,1,0)
Establish connection
BasicCommunication.UpdateDeviceList (deviceName, deviceID)
BasicCommunication.UpdateDeviceList (SUCCESS)
Check device location
RC.OnDeviceLocationChanged (deviceName, deviceID, deviceLocation(0,1,0))
Treat (0,1,0) zone as device location
Change device location to "rear right passenger"(1,1,0)
Understand the device location is changed
RC.OnDeviceLocationChanged (deviceName, deviceID, deviceLocation(1,1,0))
Treat (1,1,0) zone as device location

10.4. JSON Messages Examples

Notification

```
{
     "jsonrpc" : "2.0",
     "method" : "RC.OnDeviceRankChanged",
     “params” :
     {
            "deviceLocation" :
            {
                   "col" : 0,
                   "row" : 1,
                   "level" : 0,
                   "colspan" : 2,
                   "rowspan" : 2,
                   "levelspan" : 0
            },
            "device" :
            {
                   "name" : "Paul's phone",
                   "id" : 123456
            }
     }
}
```

Appendix 1: RSDL HMI_API.xml

```
<interfaces name="Reverse-SDL HMI API">

<interface name="Common" version="1.2" date="2015-11-16">

    <enum name="Result">
        <element name="READ_ONLY" value="25"/>
    </enum>

    <enum name="AppHMIType">
        <element name="REMOTE_CONTROL" />
    </enum>

    <enum name="ButtonName">
                <!-- Climate Buttons -->
        <element name="AC_MAX" />
        <element name="AC" />
        <element name="RECIRCULATE" />
        <element name="FAN_UP" />
        <element name="FAN_DOWN" />
        <element name="TEMP_UP" />
        <element name="TEMP_DOWN" />
        <element name="DEFROST_MAX" />
        <element name="DEFROST" />
        <element name="DEFROST_REAR" />
        <element name="UPPER_VENT" />
        <element name="LOWER_VENT" />
                <!-- Radio Buttons -->
        <element name="VOLUME_UP" />
        <element name="VOLUME_DOWN" />
        <element name="EJECT" />
        <element name="SOURCE" />
        <element name="SHUFFLE" />
        <element name="REPEAT" />
    </enum>

    <enum name="ModuleType">
        <element name="CLIMATE"/>
        <element name="RADIO"/>
    </enum>

    <enum name="RadioBand">
        <element name="AM"/>
        <element name="FM"/>
        <element name="XM"/>
    </enum>

    <enum name="RadioState">
        <element name="ACQUIRING"/>
        <element name="ACQUIRED"/>
        <element name="MULTICAST"/>
        <element name="NOT_FOUND"/>
    </enum>

    <enum name="DefrostZone">
        <element name="FRONT"/>
        <element name="REAR"/>
        <element name="ALL"/>
    </enum>

    <enum name="TemperatureUnit">
        <element name="KELVIN"/>
        <element name="FAHRENHEIT"/>
        <element name="CELSIUS"/>
    </enum>

    <enum name="DeviceRank">
        <element name="DRIVER">
            <description>The device is ranked as driver's</description>
        </element>
        <element name="PASSENGER">
            <description>The device is ranked as passenger's</description>
        </element>
    </enum>


    <struct name="InteriorZone">
        <description>Describes the origin and span of a zone in the vehicle. Vehicle
zones can be overlapping</description>
        <param name="col" type="Integer" minvalue="0" maxvalue="100">
        </param>
        <param name="row" type="Integer" minvalue="0" maxvalue="100">
        </param>
        <param name="level" type="Integer" minvalue="0" maxvalue="100">
        </param>
        <param name="colspan" type="Integer" minvalue="0" maxvalue="100">
        </param>
        <param name="rowspan" type="Integer" minvalue="0" maxvalue="100">
        </param>
        <param name="levelspan" type="Integer" minvalue="0" maxvalue="100">
        </param>
    </struct>

    <struct name="ModuleDescription">
        <param name="moduleZone" type="Common.InteriorZone">
        </param>
        <param name="moduleType" type="Common.ModuleType">
        </param>
    </struct>

    <struct name="ModuleData">
        <description>The moduleType indicates which type of data should be changed and
identifies which data object exists in this struct. For example, if the moduleType is
CLIMATE then a "climateControlData" should exist</description>
        <param name="moduleType" type="Common.ModuleType">
        </param>
        <param name="moduleZone" type="Common.InteriorZone">
        </param>
        <param name="radioControlData" type="Common.RadioControlData"
mandatory="false">
        </param>
        <param name="climateControlData" type="Common.ClimateControlData"
mandatory="false">
        </param>
    </struct>
    <struct name="RdsData">
        <param name="PS" type="String" minlength="0" maxlength="8">
            <description>Program Service Name</description>
        </param>
        <param name="RT" type="String" minlength="0" maxlength="64">
            <description>Radio Text</description>
        </param>
        <param name="CT" type="String" minlength="24" maxlength="24">
            <description>The clock text in UTC format as
YYYY-MM-DDThh:mm:ss.sTZD</description>
        </param>
        <param name="PI" type="String" minlength="0" maxlength="6">
            <description>Program Identification - the call sign for the radio
station</description>
        </param>
        <param name="PTY" type="Integer" minvalue="0" maxvalue="31">
            <description>The program type - The region should be used to differentiate
between EU and North America program types</description>
        </param>
        <param name="TP" type="Boolean">
            <description>Traffic Program Identification - Identifies a station that
offers traffic</description>
        </param>
        <param name="TA" type="Boolean">
            <description>Traffic Announcement Identification - Indicates an ongoing
traffic announcement</description>
        </param>
        <param name="REG" type="String">
            <description>Region</description>
        </param>
    </struct>

    <struct name="RadioControlData">
        <param name="frequencyInteger" type="Integer" minvalue="0" maxvalue="1710"
mandatory="false">
            <description>The integer part of the frequency ie for 101.7 this value
should be 101</description>
        </param>
        <param name="frequencyFraction" type="Integer" minvalue="0" maxvalue="9"
mandatory="false">
            <description>The fractional part of the frequency for 101.7 is
7</description>
        </param>
        <param name="band" type="Common.RadioBand" mandatory="false">
        </param>
        <param name="rdsData" type="Common.RdsData" mandatory="false">
        </param>
        <param name="availableHDs" type="Integer" minvalue="1" maxvalue="3"
mandatory="false">
            <description>number of HD sub-channels if available</description>
        </param>
        <param name="hdChannel" type="Integer" minvalue="1" maxvalue="3"
mandatory="false">
            <description>Current HD sub-channel if available</description>
        </param>
        <param name="signalStrength" type="Integer" minvalue="0" maxvalue="100"
mandatory="false">
        </param>
        <param name="signalChangeThreshold" type="Integer" minvalue="0" maxvalue="100"
mandatory="false">
            <description>If the signal strength falls below the set value for this
parameter, the radio will tune to an alternative frequency</description>
        </param>
        <param name="radioEnable" type="Boolean" mandatory="false">
            <description> True if the radio is on, false is the radio is
off</description>
        </param>
        <param name="state" type="Common.RadioState" mandatory="false">
        </param>
    </struct>

    <struct name="ClimateControlData">
        <param name="fanSpeed" type="Integer" minvalue="0" maxvalue="100"
mandatory="false">
        </param>
        <param name="currentTemp" type="Integer" minvalue="0" maxvalue="100"
mandatory="false">
        </param>
        <param name="desiredTemp" type="Integer" minvalue="0" maxvalue="100"
mandatory="false">
        </param>
        <param name="temperatureUnit" type="Common.TemperatureUnit" mandatory="false">
        </param>
        <param name="acEnable" type="Boolean" mandatory="false">
        </param>
        <param name="circulateAirEnable" type="Boolean" mandatory="false">
        </param>
        <param name="autoModeEnable" type="Boolean" mandatory="false">
        </param>
        <param name="defrostZone" type="Common.DefrostZone" mandatory="false">
        </param>
        <param name="dualModeEnable" type="Boolean" mandatory="false">
        </param>
    </struct>

</interface>


<interface name="RC" version="1.0" date="2015-10-13">

    <function name="GetInteriorVehicleDataCapabilities" messagetype="request">
        <description>Called to retrieve the available zones and supported control
types</description>
        <param name="zone" type="InteriorZone" mandatory="false">
            <description>If included, only the corresponding modules able to be
controlled by that zone will be sent back. If not included, all modules will be
returned regardless of their ability to be controlled by specifc zones.</description>
        </param>
        <param name="moduleTypes" type="ModuleType" array="true" mandatory="false"
minsize="1" maxsize="1000">
            <description>If included, only the corresponding type of modules a will be
sent back. If not included, all module types will be returned.</description>
        </param>
        <param name="appID" type="Integer" mandatory="true">
            <description>Internal SDL-assigned ID of the related
application</description>
        </param>
    </function>

    <function name="GetInteriorVehicleDataCapabilities" messagetype="response">
        <param name="interiorVehicleDataCapabilities" type="ModuleDescription"
array="true" minsize="1" maxsize="1000">
        </param>
    </function>


    <function name="GetInteriorVehicleData" functionID="GetInteriorVehicleDataID"
messagetype="request">
        <param name="moduleDescription" type="ModuleDescription">
            <description>The zone and module data to retrieve from the vehicle for
that zone</description>
        </param>
        <param name="subscribe" type="Boolean" mandatory="false" defvalue="false">
            <description>If subscribe is true, the head unit will send
onInteriorVehicleData notifications for the moduleDescription</description>
        </param>
        <param name="appID" type="Integer" mandatory="true">
            <description>Internal SDL-assigned ID of the related
application</description>
        </param>
    </function>

    <function name="GetInteriorVehicleData" functionID="GetInteriorVehicleDataID"
messagetype="response">
        <param name="moduleData" type="ModuleData">
        </param>
        <param name="isSubscribed" type="Boolean" mandatory="false" >
            <description>Is a conditional-mandatory parameter: must be returned in
case "subscribe" parameter was present in the related request.
            if "true" - the "moduleDescription" from request is successfully
subscribed and  the head unit will send onInteriorVehicleData notifications for the
moduleDescription.
            if "false" - the "moduleDescription" from request is either unsubscribed
or failed to subscribe.</description>
        </param>
    </function>


    <function name="SetInteriorVehicleData" functionID="SetInteriorVehicleDataID"
messagetype="request">
        <param name="moduleData" type="ModuleData">
            <description>The zone, module, and data to set for the (zone, module)
pair</description>
        </param>
        <param name="appID" type="Integer" mandatory="true">
            <description>Internal SDL-assigned ID of the related
application</description>
        </param>
    </function>

    <function name="SetInteriorVehicleData" functionID="SetInteriorVehicleDataID"
messagetype="response">
        <description>Used to set the values of one zone and one data type within that
zone</description>
        <param name="moduleData" type="ModuleData">
        </param>
```

```
     </function>


    <function name="GetInteriorVehicleDataConsent" messagetype="request">
        <description>Sender: SDL->HMI. </description>
        <description>HMI is expected to display a permission prompt to the driver
showing the module, zone, and app details (for example, app's name). The driver is
expected to have an ability to grant or deny the permission.</description>
        <param name="moduleType" type="Common.ModuleType" mandatory="true">
            <description>The module that the app requests to control.</description>
        </param>
        <param name="zone" type="Common.InteriorZone" mandatory="true">
            <description>A zone from which the app requests to control the named
module.</description>
        </param>
        <param name="appID" type="Integer" mandatory="true">
            <description>ID of the application that triggers the permission
prompt.</description>
        </param>
    </function>

    <function name="GetInteriorVehicleDataConsent" messagetype="response">
        <param name="allowed" type="Boolean" mandatory="true">
            <description>"true" - if the driver grants the permission for controlling
to the named app; "false" - in case the driver denies the permission for controlling
to the named app.</description>
        </param>
    </function>


    <function name="OnInteriorVehicleData" functionID="OnInteriorVehicleDataID"
messagetype="notification">
        <param name="moduleData" type="ModuleData">
        </param>
    </function>


    <function name="OnReverseAppsAllowing" messagetype="notification">
        <description>Sender: vehicle -> RSDL. Notification about remote-control for
passenger`s applications must be either turned off or turned on. Sent after User`s
choice through HMI.</description>
        <param name="allowed" type="Boolean" mandatory="true" >
            <description>If "true" - RC for passengers is allowed; if "false" -
disallowed.</description>
        </param>
    </function>


    <function name="OnDeviceRankChanged" messagetype="notification">
        <description>Sender: vehicle->RSDL. Purpose: inform about the device is set
either as driver`s or as passenger`s  device by the HMI settings or by the HMI User
through the appropriate menu.</description>
        <param name="deviceRank" type="Common.DeviceRank" mandatory="true">
            <description>If "DRIVER" - the named by "DeviceInfo" device is set as
driver`s. If "PASSENGER" -  the named by "DeviceInfo" device is set as passenger`s.
</description>
        </param>
        <param name="device" type="Common.DeviceInfo" mandatory="true">
            <description>The device info: name and ID. Initially is sent by SDL via
```

```
UpdateDeviceList to the vehicle.</description>
        </param>
    </function>


    <function name="OnDeviceLocationChanged" messagetype="notification">
        <description>Sender: vehicle->RSDL. Purpose: inform about the device location
in the vehicle interior.</description>
        <param name="deviceLocation" type="Common.InteriorZone" mandatory="true">
            <description>Defines the vehicle zone where the named device is located.
</description>
        </param>
        <param name="device" type="Common.DeviceInfo" mandatory="true">
            <description>The device info: name and ID. Initially is sent by SDL via
UpdateDeviceList to HMI.</description>
        </param>
    </function>

</interface>


<interface name="Buttons" version="1.1" date="2015-10-13">

    <function name="ButtonPress" messagetype="request">
        <param name="zone" type="Common.InteriorZone">
            <description>The zone where the button press should occur.</description>
        </param>
        <param name="moduleType" type="Common.ModuleType">
            <description>The module where the button should be pressed</description>
        </param>
        <param name="buttonName" type="Common.ButtonName"/>
        <param name="buttonPressMode" type="Common.ButtonPressMode">
            <description>Indicates whether this is a LONG or SHORT button press
event.</description>
        </param>
        <param name="appID" type="Integer" mandatory="true">
            <description>ID of the application that triggers the permission
prompt.</description>
        </param>
    </function>

    <function name="ButtonPress" messagetype="response">
    </function>
```

```
</interface>

</interfaces>
```

Change History

Date	Version	Status	Description	Author	Notes
13-Nov-2015	v1.0	Draft	Initial creation	A.Britanova	RSDL_HMI_Guidelines_v1.0.pdf
24-Nov-2015	v1.1.	Released	> add Buttons.ButtonPress > add BasicCommunication.OnAppRegistered > RC.OnReverseAppsAllowing -> add "Usage" to "Description" > RC.OnDeviceRankChanged -> add "Usage" to "Description" > RC.OnDeviceLocationChanged -> add "Usage" to "Description"	A.Britanova	Current version