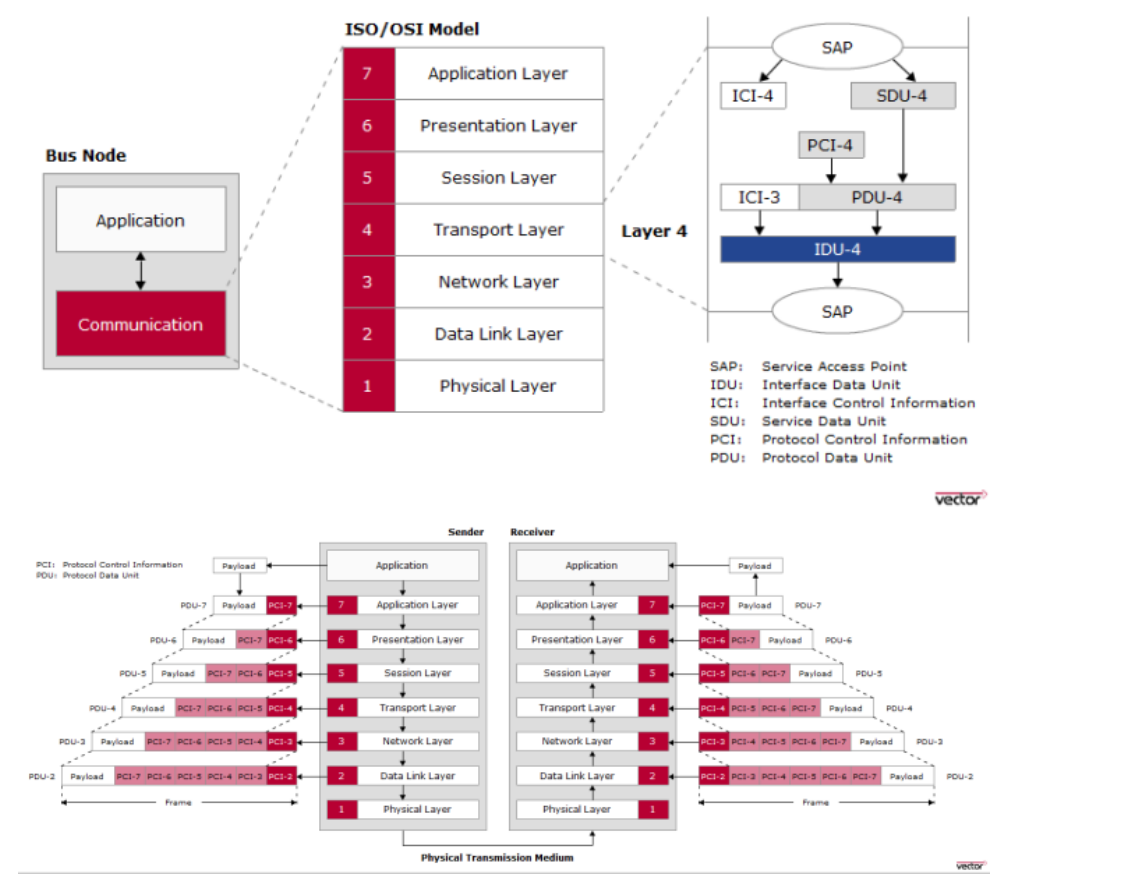
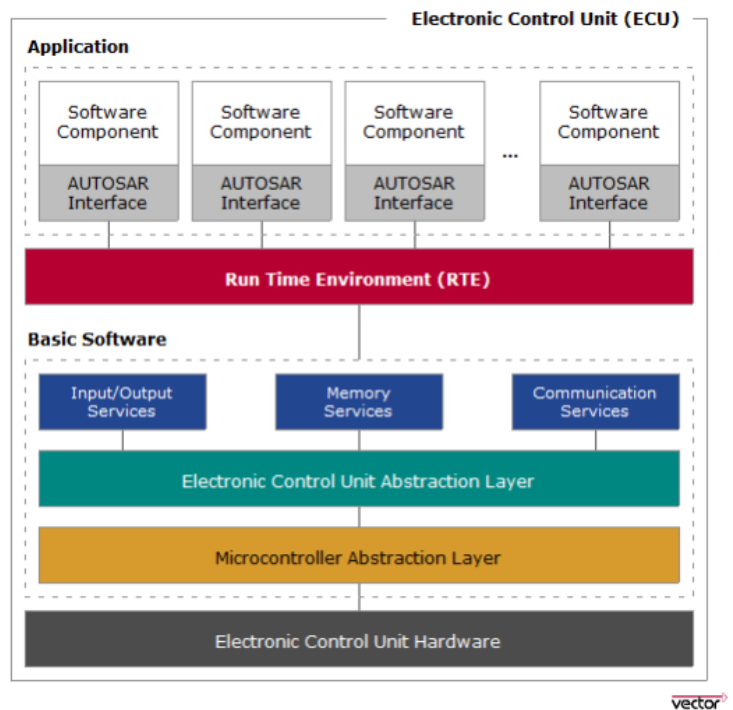
**DSM configuration**

Serial bus system

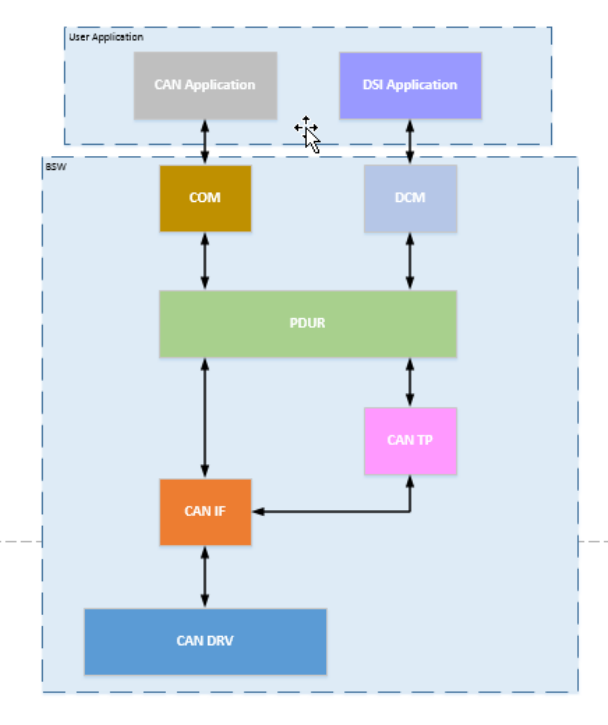




# 1/ Overview

### A/ Definition

DIAGNOSIS is an offline tool used to check the healthiness of the ECU components. It controls functions such as test, inspection, monitoring, or diagnosis of an on-vehicle ECU.



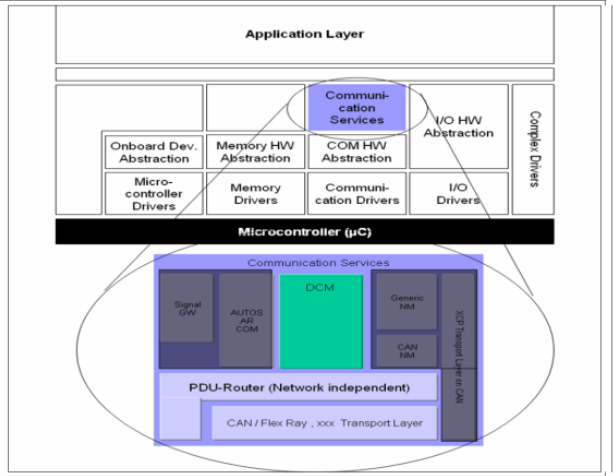
The DCM module provides a common API for Diagnostic service. The functionality ò DCM module is used by external diagnostic tool during development, manufacturing or services.

DCM has some functions as below:

-DCM module ensures diagnostic data flow and manages the diagnostic states, especially diagnostic session and security state.

-DCM check if the service requested is supported

In the AUTOSAR Architecture the Diagnostic Communication Manager is located in the Communication Services (Service Layer).



The DCM module is network-independent.

All network-specific functionality (the specification of networks like CAN, LIN, FlexRay or MOST) is handled outside of the DCM module.

The PDU Router (PduR) module provides a network-independent interface to the DCM.

The DCM module receives a diagnostic message from PduR module, the DCM process and check the message internally.

As part of processing the diagnostic service, the DCM will interact with other BSW module or SWC to obtain request data or execute the request command.

Typically, the DCM will assemble gathered information and send a message back through the PduR module.

|  |  |
| --- | --- |
| Abbreviation | Description |
| API | Application programming interface |
| DCM | Diagnostic communication manager |
| DID | Data Identifier |
| DSD | Diagnostic Service Dispatcher |
| DSL | Diagnostic Session Layer |
| DSP | Diagnostic Service Processing |
| DTC | Diagnostic Trouble Code |
| NRC | Negative Response Code |
| OBD | On-Board Diagnostic |
| PDU | Protocol Data Unit |
| ROE | Response On Event |

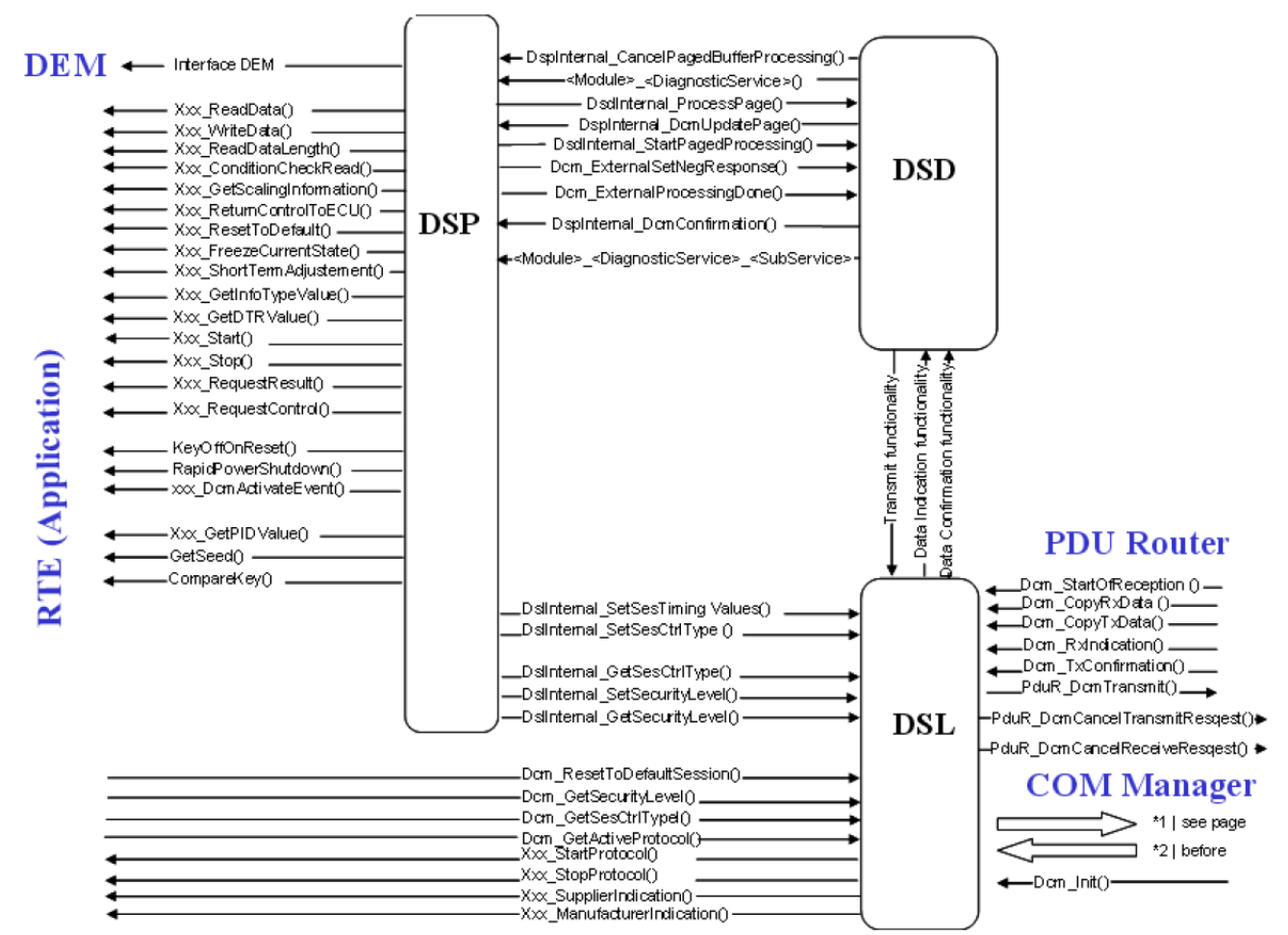
### B/ DCM architecture

#### Interaction between DCM and other modules

To define the functionality of the DCM module, The DCM SWS models the DCM module as consisting of the following sub-modules:

* Diagnostic Session Layer (DSL) sub-module: The DSL ensures data flow concerning diagnostic requests and responses, supervises and guarantees diagnostic protocol timing as diagnostic state (especially diagnostic session and security)
* Diagnostic Service Dispatcher (DSD) sub-module: The DSD sub-module process a stream of diagnostic data.
  + Receives a new diagnostic request over the network and forward it to a data processer.
  + Transmits a diagnostic response over a network when triggered by the data processor (e.g. by the DSP sub-module).
* Diagnostic Service Processing (DSP) sub-module: The DSP sub-module handles the actual diagnostic service (respectively subservice) requests.

The next graphic gives an overview of the interfaces between the sub-modules DSP, DSD, and DSL within the DCM module.



#### Diagnostic Session Layer (DSL)

##### Functionality

The DSL sub-module provides the following functionalities:

###### Request Handling

* **Forward requests from the PduR module to the DSD sub- module**

The PduR module indicates the DCM module whenever a reception of a new diagnostic request content is started on a DcmRxPduId, which is assign to the DCM module by calling Dcm\_StartOfReception(), this function inform the DCM module of the data size to be received and allow the DCM to reject the reception if the data size overflow its buffer size.  
The Further call DCM\_CoppyRxData request DCM copy the data from the provide buffer to the DCM buffer.  
If the reception of diagnostic request is finished, the PduR module will call Dcm\_TpRxIndication() to give a receive indication to DCM module.  
The DSL sub-module shall forward received data to the DSD sub-module only after a call of Dcm\_TpRxIndication() with parameter Result = NTFRSLT\_OK.

* **Concurrent "Tester Present"**

"Tester Present" commands are sent by the tester in parallel to physical requests/responses. This is called "keep alive logic" in ISO14229-1 [14]. This functional "Tester Present" will be received on a separate DcmRxPduId with a separate receive buffer. Due to that reason, the functional "Tester Present" (and only functional "Tester Present" without response) is handled in the following way:  
When the PduR module calls Dcm\_TpRxIndication() with parameter Result=NTFRSLT\_OK, if the request is a Tester present command with "suppressPosRspMsgIndicationBit" set to TRUE (SID equal to 0x3E, sub-function equal to 0x80), the DSL sub-module shall reset the session timeout timer (S3Server) and not forward this request to DSD sub-module.

###### Response Handling

* **Forward responses from the DSD sub-module to the PduR module**

The DSD sub-module shall request DSL sub-module for transmission of responses.  
When the diagnostic response is ready the DSL will trigger the transmission of the diagnostic response to the PduR module by calling PduRDcmTransmit()  
If the DSL sub-module receives a confirmation after the complete DCM PDU has successfully been transmitted or an error occurred by a call of Dcm\_TpTxConfirmation(), then the DSL sub-module shall forward this confirmation to the DSD sub-module. In case of a failed transmission (failed PduR\_DcmTransmit() request) or error confirmation (Dcm\_TpTxConfirmation() with error), the DSD sub-module shall not repeat the diagnostic response transmission.

* **Guarantee timing to tester by sending busy responses**

If the application or DSP sub-module is able to perform a requested diagnostic task, but need additional more time to finish the task and prepare response, sub-module DSL shall send a negative response with NRC 0x78 when reaching the response time.

* **Support of periodic transmission.**

The UDS service ReadDataByPeriodicIdentifier (0x2A) allows the tester to request the periodic transmission of data record values from the ECU identified by one or more periodicDataIdentifiers.  
The DCM module shall send responses for periodic transmissions using a separate protocol and a separate buffer of configurable size.  
If a pending message for normal diagnostic response (high priority) exists, then the DSL will wait for the transmission confirmation for this normal diagnostic response before transmit a periodic message response.

* **Support ROE transmission**

With the UDS Service ResponseOnEvent (0x86) tester requests an ECU start or stop transmission of responses initiated by a specified event. Upon registering an event for transmission, the tester also specifies the corresponding service to respond to (e.g: UDS Service ReadDataByIdentifier (0x22)).  
The ROE functionality is enable only if the container DcmDslResponseOnEvent exists.

* **Support of segmented response (paged-buffer**

If enabled (DcmPagedBuferEnable = TRUE) the DCM module shall provide a mechanism to send response larger than configured and located diagnostic buffer.  
With paged-buffer handling the ECU is not forced to provide a buffer, which is as large as the maximum length of response.  
Page-buffer handling is for transmit only- not for reception  
Page-buffer handling is not available for application (DCM internal-use only)

* **Support of ResponsePending response triggered by the Application**

In some cases, e.g.in case of routine execution, an application need to request an immediate NRC 0x78 (Response Pending), which will be sent immediately and not just before reaching response time.  
When the DCM module calls an operation and gets an error status E\_FORCE\_RCRRP, the DSL sub-module will trigger the transmission of a negative response with NRC 0x78 (Response pending).

* **Manage security level**

The DSL shall save the level of the current active security level  
For accessing this level, DCM provides 2 interface to:

* Get the current active security level Dcm\_GetSecurityLevel()
* Set a new security level: DslInternal\_SetSecurityLevel()

During DCM initialization the security level is set to value 0x00.

* **Manage session state**

The DSL submodule shall save the state of the current active session.  
For accessing this variable, the DSL sub-module shall provide interface to

* + **Get current active session:** DCM\_GetSesCtrlType()
  + **Set a new session:** DslInternal\_SetSesCtrlType()

During DCM initialization the session state is set to value 0x01 ("DefaultSession").

* **Keep track of active non-default sessions**
  + Whenever a non-default session is active and when a session timeout is reached without receive any diagnostic request. The DSL shall reset to the default session state (Default Session, 0x01).
* **Allow to modify timings**

All protocols (OBD, enhanced diagnosis) share the same protocol timing values (given by default session configuration).  
These timing values are set when the protocol is started. These protocol timing parameters have influence on the session layer timing (no influence on Transport Layer timing). Some of these timing parameters can be modified while protocol is active with the following means:

* + UDS Service DiagnosticSessionControl (0x10)
  + UDS Service AccessTimingParameter (0x83)

The DSL sub-module provides the following functionalities to modify the timing parameters:

* + Provide the active timing parameters,
  + Set the new timing parameters. Activation of new timing values is only allowed after sending the response
* **Handling of different diagnostic protocol**

Different protocols shall be used for UDS Services versus OBD service.  
With every protocol initialization, the DSL sub-module set a link to corresponding to the service table.  
If a running diagnostic request have a higher priority of another protocol the DSL sub-module shall call all configured Xxx\_StopProtocol()functions.  
In order to cancel pending transmission in lower-layer, related to the lower prior request, the DCM module shall call PduR\_DCMCancelTransmit()

* **Manage resources**

Due to limited resources, the following points should be considered as hints for the design:

* + It is allowed to use and allocate only one diagnostic buffer in the DCM module. This buffer is then used for processing the diagnostic requests and responses.
  + Output of NRC 0x78 (Response pending) responses is done with a separate buffer.
  + paged-buffer handling
* **Communication Mode Handling**
  + No Communication  
    The ComM module will indicate the No Communication Mode to the DCM module by calling Dcm\_ComM\_NoComModeEntered (In response, the DCM will immediately disable all transmissions
  + Silent Communication  
    The ComM module will indicate the Silent Communication Mode to the DCM module by calling Dcm\_ComM\_SilentComModeEntered(). In response, the DCM will immediately disable all transmissions
  + Full Communication  
    The ComM module will indicate the Full Communication Mode to the DCM module by calling Dcm\_ComM\_FullComModeEntered().

#### **Diagnostic Session Dispatcher** **(DSD)**

##### Functionality

* **Support checking the diagnostic service identifier and adapting the diagnostic message**

The DSD sub-module shall be triggered by the DSL sub-module if a new diagnostic message is recognized. The DSD sub-module will start processing by analyzing the diagnostic service identifier contained in the received diagnostic message

* **Handling of "suppressPosRspMsgIndicationBit"**

The "suppressPosRspMsgIndicationBit" is part of the subfunction parameter structure (Bit 7 based on second byte of the diagnostic message, see ISO14229-1 [9] Section 6.5: Server response implementation rules)  
If the "suppressPosRspMsgIndicationBit" is TRUE, the DSD sub-module shall NOT send a positive response message  
  
The DSD sub-module shall remove the "suppressPosRspMsgIndicationBit" (by masking the Bit) from the diagnostic message.  
In case of responsePending the DCM module shall clear the "suppressPosRspMsgIndicationBit."  
The "suppressPosRspMsgIndicationBit" is only available if a service has a sub-function

* **Verification functionality**
  + **Verification of the Diagnostic Session**  
    On receiving a service request, the DSD module will obtain the current Diagnostic Session with Dcm\_GetSesCtrlType() and will verify whether the execution of the requested service (NOT the UDS Service DiagnosticSessionControl (0x10)) is allowed in the current diagnostic session or not.  
    If the newly received diagnostic service is not allowed in the current Diagnostic Session (according to the configuration parameter DcmDsdSidTabSessionLevelRef), the DSD sub-module shall transmit a negative response with NRC 0x7F (Service not supported in active session) to the DSL sub-module.  
    If the newly received diagnostic service is allowed in the current Diagnostic Session, but the requested subservice is not current Diagnostic Session the DSD sub-module shall transmit a negative response with NRC 0x7E (SubFunctionNotSupportedInActiveSession) to the DSL sub-module.
  + **Verification of the Service Security Access levels**  
    If the newly received diagnostic service is not allowed in the current Security level (according to the configuration parameter DcmDsdSidTabSecurityLevelRef), the DSD sub-module shall transmit a negative response with NRC 0x33 (Security access denied) to the DSL sub-module  
      
    If the newly received diagnostic service is allowed in the current Security level but the requested subservice is not allow in the current Security level (according to the configuration parameter DcmDsdSubServiceSecurityLevelRef), the DSD sub-module shall transmit a negative response with NRC 0x33 (Security access denied) to the DSL sub-module
  + **Check format and sub-function support**  
    The DSD sub-module checks whether a specific sub-function is supported before executing the requested command.  
    The DSD sub-module shall trigger a negative response with NRC 0x12 (Sub-Function not supported), when the analysis of the request message results in sub-function not supported.  
    The DSD sub-module will check for the minimum message length before executing the requested command.  
    The DSD sub-module shall trigger a negative response with NRC 0x13 (Incorrect message length or invalid format), if the length of the request is inferior the minimum length of the request.
  + **Distribution of diagnostic message to DSP sub-module**  
    The DSD sub-module shall search for the DSP sub-module for newly received diagnostic service identifier and shall call the corresponding DSP service interpreter.
  + **Assemble positive or negative response**  
    **Positive Response**  
    The DSP sub-module indicates a positive result by calling Dcm\_ExternalProcessingDone(). The parameter "Dcm\_MsgContextType" comprises the diagnostic (response) message.  
    **Negative Response**  
    The DSD submodule shall handle all NRCssupported from the Application and defined in Dcm\_NegativeResponseCodeType  
    **Suppression of response**  
      
    In the case of a Negative Result of the execution and active Functional Addressing the DSD sub-module shall activate the suppression of the following Negative Responses:  
    NRC 0x11 (Service not supported),  
    NRC 0x12 (SubFunction not supported),  
    NRC 0x31 (Request out of range).
* **Initiate transmission**

The DSD submodule shall forward the diagnostic (response) message (positive or negative response) to the DSL sub-module.  
The DSL submodule shall forward the diagnostic (response) message (positive or negative response) further to the PduR module by executing a DSL transmit functionality  
  
The DSL sub-module shall forward the received confirmation from the PduR module to the DSD sub-module.  
The DSD sub-module shall forward the confirmation via the internal function DspInternal\_DcmConfirmation()to the DSP sub-module.

#### **Diagnostic Session Processing** **(DSP)**

##### Functionality

When receiving a function call from the DSD sub-module requiring the DSP sub-module to process a diagnostic service request, the DSP always carries out following basic process steps:

* analyze the received request message,
* check format and whether the addressed sub-function is supported,
* acquire data or execute the required function call on the DEM, SW-Cs or other BSW modules
* assemble the response
* **Check format and sub-function support**

The DSP sub-module will check for appropriate message length and structure before executing the requested command.  
The DSP sub-module shall trigger a negative response with NRC 0x13 (Incorrect message length or invalid format), when the analysis of the request message results in formatting or length failure.

* **Assemble response**

The DSP sub-module shall assemble the response message excluding response service identifier and determine the response message length.  
The DSP sub-module shall confirm the completion of the request processing with the function call DsdInternal\_ProcessingDone().

* **Additional Negative response code**
  + The DSP submodule shall trigger a negative response with NRC 0x22 (Conditions not correct), when the API calls made to execute the service do not return OK.
  + The DSP submodule shall trigger a negative response with NRC 0x7F (Service not supported in active session), when the analysis of the request message results in subfunction not supported in active diagnostic session.
  + The DSP submodule shall trigger a negative response with NRC 0x31 (Request out of range), when the analysis of the request message results in other unsupported message parameters.
  + The DSP submodule shall trigger a negative response with NRC 0x33 (Security access denied), when the analysis of the request message results in subfunction not supported in active security level

# 2/ BCT configured

### DcmDsl

### DcmDsd

### DcmDsp

# 3/ Functionality