Document Title	System Tests of Adaptive Platform
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	890

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	17-10

Document Change History

Date	Release	Changed by	Change Description
2017-10-27	17-10	AUTOSAR Release Management	• Initial release

Disclaimer

Table of Contents

1 Acronyms and abbreviations

The glossary below includes terms, acronyms and abbreviations relevant to System Test Specification that are not included in the AUTOSAR Glossary [1].

Abbreviation / Acronym:	***Description:***
PCO	Point of Control and Observation
Rx	Reception
RS	Requirement Specification
SUT	System Under Test
SWC	Software Component
TCP	Test Coordination Procedures
Tx	Transmission
ST	System Test

2 Scope

The following test cases are used to validate acceptance criteria in visions in order to confirm whether original intention of features are satisfied by Demonstrator of AUTOSAR Adaptive Platform.

Each test case is applicable with coupled release of specification and implementation.

3 Test configuration and test steps for Communication Management

3.1 Acceptance criteria

The following acceptance criteria represent a snapshot from the JIRA system on 01/09/2017.

Vision ID	AP-4	
Title	Location independence of applications	
Acceptance Criteria	**ID**	Description
	AP-4-01	Three applications in demonstrator, two locally, one remote in a one-to-n communication topology where one local application communicates with the other local and the remote application
	AP-4-02	Three applications in demonstrator, two locally, one remote in a one-to-n communication topology where one local application communicates with the other local and the remote application (see AP-4-01) swap local and remote partners from the basic setting without adaptation of the source code
	AP-4-03	signature of the communication API does not exhibit specific items relevant for remote or local communication

Vision ID	AP-09	
Title	Discovery of local and remote Services	
Acceptance Criteria	**ID**	Description
	AP-09-01	Service offers and needs of an application can be described for the application, communicated by the application and accepted by the platform instance.
	AP-09-02	The platform instance performs service discovery and allows the involved applications to initiate communication.
	AP-09-03	Application A executed on platform instance p_1 issues provision of service S. Application B executed on platform instance p_2 issues request of service S. Platform instances interact and establish communication path between A and B via S.
	AP-09-04	Application A executed on platform instance p_1 issues provision of service S. Application B executed on platform instance p_1 issues request of service S. Platform instance establishes communication path between A and B via S.
	AP-09-05	Application A executed on platform instance p_1 issues provision of service S_1. Application B executed on platform instance p_1 issues request of service S_2. Platform instance analyses service request by B and notifies B that no service S_2 is available.
	AP-09-06	Application B executed on platform instance p_2 issues request of service S. Platform instance analyses service request by B and notifies B that no service S is available. Application A executed on platform instance p_1 issues provision of service S. Application B executed on platform instance p_2 issues second request of service S. Platform instances interact and establish communication path between A and B via S.
	AP-09-07	Static communication relationships can be defined without the need of a Service Discovery run. Setup with two platform instances running the adaptive platform and two Adaptive Applications establish a communication by services discovery and then communicate with each other a) Dynamic binding: Communication relationship is established after the applications are deployed on the same ecu or on different ecus

		b) Static binding: Communication relationship is established according to static definition of service binding before the applications are deployed on the same platform instance or on different platform instances.
	AP-09-08	Application B executed on platform instance p_2 issues request of service S. Platform instance analyses service request by B and notifies B that no service S is available. Application A executed on platform instance p_1 issues provision of service S. Platform instances interact and establish communication path between A and B via S.

Vision ID	AP-10	
Title	Service-Oriented Communication between Adaptive and Classic platform	
Acceptance Criteria	**ID**	Description
	AP-10-01	All communication will be executed via SOME/IP. Applications from the adaptive platform can communicate with other applications using service oriented communication via Ethernet.
	NOTE	ST could not ensure "all" communication is executed via SOME/IP. System test will evaluate communication can ve executed via SOME/IP. Classic platform and adaptive platform communication is not the scope of System test (for R17-10).

3.2 General test objective and approach

3.3 Test System

3.3.1 Test configurations

Configuration ID	STC_CM_00001
Description	Standard CI server for Communication Management test
ECU 1	Intel MinnowBoard Turbot, 192.168.100.5
ECU 2	Renesas R-Car H3 ULCB, 192.168.100.2
Jenkins	Jenkins Server, 192.168.100.10
SW information	Please see doxygen file (https://codeci.autosar.org/view/FT-ST/job/FT-ST_-_ADAS_Application_Test/HTML_Report/)

Configuration ID	STC_CM_00002
Description	Scenario 2 Variant 2 – Reference Deployment
ECU 1	Intel MinnowBoard Turbot, 192.168.100.5
ECU 2	Renesas R-Car H3 ULCB, 192.168.100.2
Jenkins	Jenkins Server, 192.168.100.10
SW information	Please see doxygen file (https://codeci.autosar.org/view/FT-ST/job/FT-ST_-_ADAS_Application_Test/HTML_Report/)

Configuration ID	STC_CM_00003
Description	Scenario 2 Variant 2 – Alternative Deployment
ECU 1	Renesas R-Car H3 ULCB, 192.168.100.5
ECU 2	Intel MinnowBoard Turbot, 192.168.100.2
Jenkins	Jenkins Server, 192.168.100.10
SW information	Please see doxygen file (https://codeci.autosar.org/view/FT-ST/job/FT-ST_-_ADAS_Application_Test/HTML_Report/)

The Jenkins Server, running the job with the Communication Management test (CM Tester) is connected via Ethernet to ECU1 hosting the System Test Application *Video Provider* (as well as *EBA*

depending on the configuration) and ECU2 hosting the System Test Applications *Video Adapter*, *PreProcessing*, *Computer Vision* and *EBA*.
The CM Tester is supposed to collect the results.
The communication between CM Tester and the applications on the ECU may take place over the Diagnostics functional cluster in form of diagnostic messages.

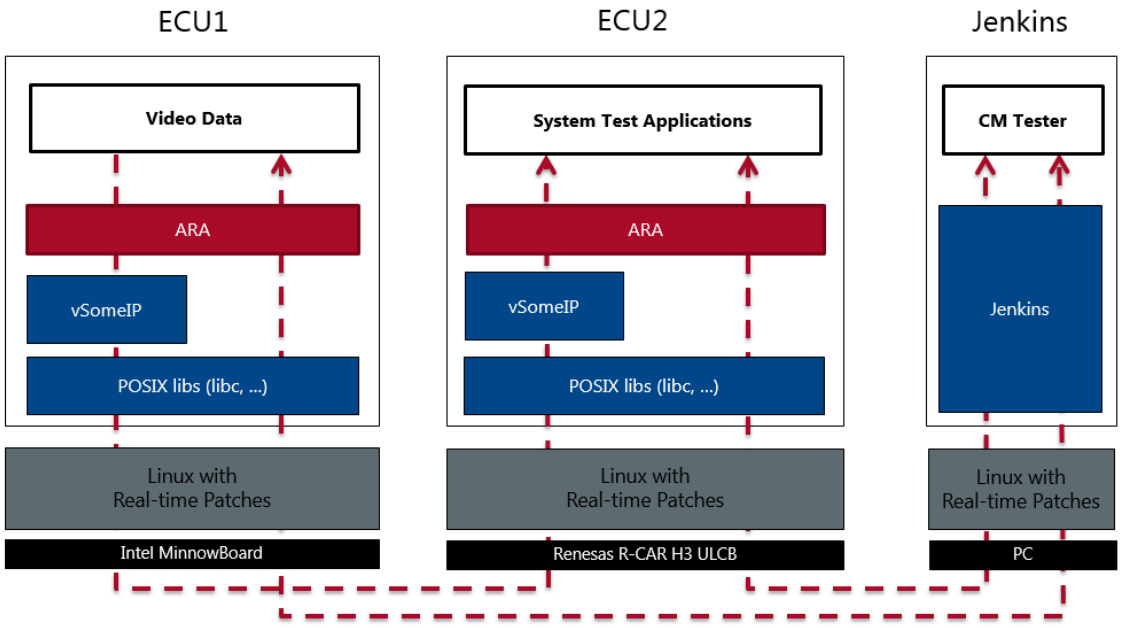

Figure 1: Illustration of test setup for Communication Management.

3.4 Test cases

3.4.1 [STS_CM_00001] Local and remote service discovery.

Test Objective	Verification, that applications are able to offer and request services and that service discovery works, establishing the correct communication paths.		
ID	STS_CM_00001	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Communication	**State**	Reviewed
Trace to Acceptance Criteria	AP-09-01 AP-09-02 AP-09-03 AP-09-04 AP-09-05 AP-09-06 AP-10-01		
Reference to Test Environment	STC_CM_00001		
Configuration Parameters	- Networking is configured properly. - The existing communication services comprise the following (service names are arbitrary): *Alive*: Offered by Video Adapter, requested by Video Provider. *Image*: Offered by Video Adapter, requested by PreProcessing. *Video*: Offered by Video Provider, requested by Video Adapter. *Unknown*: Not available, requested by PreProcessing. - ARXML model and VSomeIP configuration:		

	https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)
Summary	First, the Video Adapter and PreProcessing applications on ECU2 are started. - The Video Adapter offers the services *Alive* and *Image* and requests the service *Video*. - PreProcessing requests the service *Image* and *Unknown*. Then the Video Provider application on ECU1 is started. - The Video Provider offers the service *Video* and requests the service *Alive*. All services are supposed to be found once available. If a service is not available, the requesting application is expected to be notified. Note : As for order of offering, no particular order of offering and requesting is necessary.
Pre-conditions	- CM Tester is connected to both ECUs. - Both ECUs are in System State *Living*. - Video Provider, Video Adapter and PreProcessing are shut down according to System State.
Post-conditions	TCP connections between CM Tester and both ECUs are closed.

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[CM Tester] Query change of System State to *Driving* for ECU2.	System State for ECU2 is changed to *Driving*.
Step 2	[Video Adapter] Video Adapter is started.	
Step 3	[PreProcessing] PreProcessing is started.	
Step 4	[Video Adapter] Offer service *Alive*.	OfferService returns.
Step 5	[Video Adapter] Offer service *Image*.	OfferService returns.
Step 6	[PreProcessing] Request service *Image*.	Service discovery callback with a handle for service *Image* is received by PreProcessing.
Step 7	[CM Tester] Trigger Application Video Adapter to Stop Offering service *Image*.	
Step 8	[Video Adapter] Stop Offering service *Image*.	
Step 9	[PreProcessing] Request service *Image*.	Service is not available.
Step 10	[Video Adapter] Offer service *Image*.	OfferService returns.
Step 11	[Video Adapter]	Service is not available.

	Request service *Video*.	
Step 12	[CM Tester] Query change of System State to *Driving* for ECU1.	System State for ECU1 is changed to *Driving*.
Step 13	[Video Provider] Video Provider is started.	
Step 14	[Video Provider] Offer service *Video*.	OfferService returns.
Step 15	[Video Adapter] Request service *Video*.	Service discovery callback with a handle for service *Video* received.
Step 16	[Video Provider] Request service *Alive*.	Service discovery callback with a handle for service *Alive* is received by Video Provider.
Step 17	[PreProcessing] Request service *Unknown*.	Service is not available.

3.4.2 [STS_CM_00002] One-to-n communication topology.

Test Objective	Verification, that applications are able to offer and request services and that service discovery as well as communication work in a one-to-n communication topology.		
ID	STS_CM_00002	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Communication Management	**State**	Reviewed
Trace to Acceptance Criteria	AP-4-01 AP-4-03		
Reference to Test Environment	STC_CM_00002		
Configuration Parameters	- Networking is configured properly. - Diagnostics Read DID (brake event) and Routine Control (switch video) services configured. - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)		
Summary	The Computer Vision application on ECU2 offers the service *Vehicles*. This service is requested by one EBA instance on ECU2 and another EBA instance on ECU1. Through successful service discovery, a one-to-n communication topology is established. Note : As for order of offering, no particular order of offering and requesting is necessary.		
Pre-conditions	- CM Tester is connected to both ECUs. - Both ECUs are in System State *Living*. - Computer Vision, EBA on ECU2 and EBA on ECU1 are shut down according to System State.		
Post-conditions	TCP connections between CM Tester and both ECUs are closed.		
Main Test Execution			

Test Steps		Pass Criteria
Step 1	[CM Tester] Query change of System State to *Driving* for ECU2.	System State for ECU2 is changed to *Driving*.
Step 2	[Computer Vision] Computer Vision is started.	
Step 3	[EBA ECU2] EBA on ECU2 is started.	
Step 4	[Computer Vision] Offer service *Vehicle*.	OfferService returns.
Step 5	[EBA ECU2] Request service *Vehicle*.	Service discovery callback with a handle for service *Vehicle* is received by EBA ECU2.
Step 6	[CM Tester] Query change of System State to *Driving* for ECU1.	System State for ECU1 is changed to *Driving*.
Step 7	[EBA ECU1] EBA on ECU1 is started.	
Step 8	[EBA ECU1] Request service *Vehicle*.	Service discovery callback with a handle for service *Vehicle* is received by EBA ECU1.
Step 9	[CM Tester] Request *Video Provider* to switch to the video without brake events.	
Step 10	[CM Tester] Wait (synchronously) for the video completion.	
Step 11	[CM Tester] Query *EBA* (ECU1) about the number of brake events (BE_1A as symbolic name).	
Step 12	[CM Tester] Query EBA (ECU2) about the number of brake events (BE_2A as symbolic name).	BE_1A = BE_2A = 0
Step 13	[CM Tester] Request *Video Provider* to switch to the video with brake events.	
Step 14	[CM Tester] Wait (synchronously) for the video completion.	
Step 15	[Computer Vision] Send vehicle data over *Vehicle*.	
Step 16	[EBA ECU1] Detect threat and emit brake event.	
Step 17	[EBA ECU2]	

	Detect threat and emit brake event.	
Step 18	[CM Tester] Query *EBA* (ECU1) about the number of brake events (BE_1B as symbolic name).	
Step 19	[CM Tester] Query *EBA* (ECU2) about the number of brake events (BE_2B as symbolic name).	BE_1B=BE_2B and BE_1B=BE_1A+1

3.4.3 [STS_CM_00003] Swapping

The system test consists in another hardware deployment of STS_CM_00002.

Test Objective	Verification, that the deployment of applications is hardware independent on top of the communication management (no code modification required).		
ID	STS_CM_00003	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Communication Management	**State**	Reviewed
Trace to Acceptance Criteria	AP-4-01 AP-4-02 AP-4-03		
Reference to Test Environment	STC_CM_00003		
Configuration Parameters	See [STS_CM_00002] One-to-n communication topology.		
Summary	See [STS_CM_00002] One-to-n communication topology.		
Pre-conditions	See [STS_CM_00002] One-to-n communication topology.		
Post-conditions	See [STS_CM_00002] One-to-n communication topology.		
Main Test Execution			
Test Steps			**Pass Criteria**
Steps	See [STS_CM_00002] One-to-n communication topology.		No code modification required.

4 Test configuration and test steps for Execution Management

4.1 Acceptance criteria

The following acceptance criteria represent a snapshot from the JIRA system on 01/09/2017.

Vision ID	AP-60	
Title	Start and stop of applications	
Acceptance Criteria	**ID**	Description
	AP-60-02	applications will be initiated based on specific dynamic conditions as being defined by a system state
	AP-60-03	applications will be started at system start-up

4.2 General test objective and approach

4.3 Test System

4.3.1 Test configurations

Configuration ID	STC_EMO_00001
Description	Standard CI server for Execution Management test
ECU 2	Renesas R-Car H3 ULCB, 192.168.100.2
Jenkins	Jenkins Server, 192.168.100.10
SW information	Please see doxygen file (https://codeci.autosar.org/view/FT-ST/job/FT-ST_-_ADAS_Application_Test/HTML_Report/)

The Jenkins Server, running the job with the Execution Management test (Exec Tester) is connected via Ethernet to ECU2 hosting the System Test Applications *Video Adapter*, *PreProcessing*, *Computer Vision* and *EBA*.
The Exec Tester is supposed to check the pass criteria.
The communication between Exec Tester and the applications on the ECU may take place over the Diagnostics functional cluster in form of diagnostic messages.

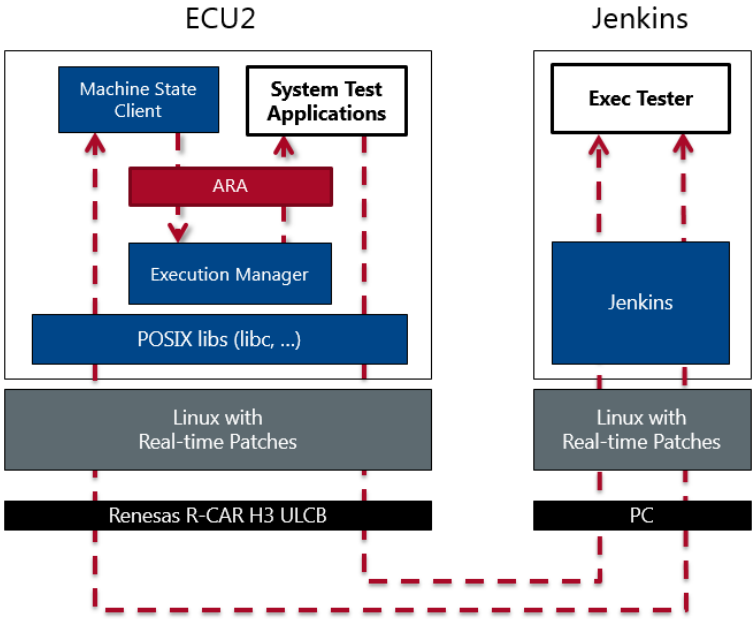

Figure 2:Illustration of test setup for Execution Management.

4.4 Test cases

4.4.1 [STS_EMO_00001] Startup of applications with change of system state.

Test Objective	Verification, that the execution management functional cluster can perform a change of System State and that applications associated with the new System State are started.		
ID	STS_EMO_00001	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Persistency	**State**	Reviewed
Trace to Acceptance Criteria	AP-60-02 AP-60-03		
Reference to Test Environment	STC_EMO_00001		
Configuration Parameters	- Networking is configured properly. - The System State *Living*, in which all System Test Applications are shut down, is defined. - System State *Driving*, in which all System Test Applications are running, is defined. - Application Manifests: https://code.autosar.org/tf-apd/sample-application/blob/FT-ST_development/ST_Scenarios/Video_Adapter/manifest/MANIFEST.json https://code.autosar.org/tf-apd/sample-application/blob/FT-ST_development/ST_Scenarios/PreProcessing/manifest/MANIFEST.json https://code.autosar.org/tf-apd/sample-application/blob/FT-ST_development/ST_Scenarios/Computer_Vision/manifest/MANIFEST.json https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/EBA/manifest (Revision 305d1fb5d48c765f756f921b2e8f1257f3ad6aa8) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model		

	(Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)
Summary	A change of System State from Living to Driving is requested and the startup of the applications associated with this System State is verified.
Pre-conditions	- Exec Tester is connected to ECU2. - Software components on ECU2 are initialized. - ECU2 is in System State *Living*. - Video Adapter, PreProcessing, Computer Vision and EBA are shut down according to System State. - Operating system on ECU2 has booted.
Post-conditions	TCP connection between Exec Tester and ECU2 is closed.

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[Exec Tester] Query change of System State to *Driving* for ECU2.	
Step 2	[Machine State Manager] Request change of System State from Execution Manager.	System State for ECU2 is changed to *Driving*.
Step 3	[Exec Tester] Query execution status of Video Adapter.	[Video Adapter] is executed.
Step 4	[Exec Tester] Query execution status of PreProcessing.	[PreProcessing] is executed.
Step 5	[Exec Tester] Query execution status of Computer Vision.	[Computer Vision] is executed.
Step 6	[Exec Tester] Query execution status of EBA.	[EBA] is executed.

5 Test configuration and test steps for Diagnostics

5.1 Acceptance criteria

The following acceptance criteria represent a snapshot from the JIRA system on 01/09/2017.

Vision ID	AP-11	
Title	Support Diagnostics according to ISO 14229-x	
Acceptance Criteria	**ID**	Description
	AP-11-1	Diagnostics Tester is connected to vehicle network via DoIP. Diagnostics Tester establishes communication paths via vehicle port 13400 and searches for DoIP nodes on Adaptive Platform instances. Diagnostics tester selects particular DoIP node on vehicle and platform instance and requests communication connection to DoIP node for exchange of diagnostics information between Diagnostics Tester and Adaptive Platform instance.
	AP-11-2	After establishment of communication path to Adaptive Platform instance (e.g. via DoIP), Diagnostics Tester reads DTC related information from the Adaptive Platform instance.
	AP-11-3	For the application exists a Diagnostics Extract (DEXT) which defines the DTCs written by the application. The application writes DTCs, which are specified in the DEXT, in a controlled manner. All DTCs must be visible at the Diagnostics tester.
	AP-11-4	Further UDS services shall be demonstrated for acceptance.

5.2 General test objective and approach

5.3 Test System

5.3.1 Test configurations

Configuration ID	STC_DIAG_00001
Description	Standard CI server for diagnostic test
ECU 1	Intel Minnowboard Turbot, 192.168.100.5
ECU 2	Renesas R-Car H3 ULCB, 192.168.100.2
Jenkins	Jenkins Server, 192.168.100.10
SW information	Please see doxygen file (https://codeci.autosar.org/view/FT-ST/job/FT-ST_-_ADAS_Application_Test/HTML_Report/)

The Jenkins Server running the job with the Diagnostic Tester is connected via Ethernet to ECU1 hosting the System Test Application *Video Provider* and ECU2 hosting the System Test Application *Emergency Brake Assistant* (EBA). The Diagnostic Tester opens TCP connections on port 13400 and sends diagnostic data as UDS requests in DoIP packets.

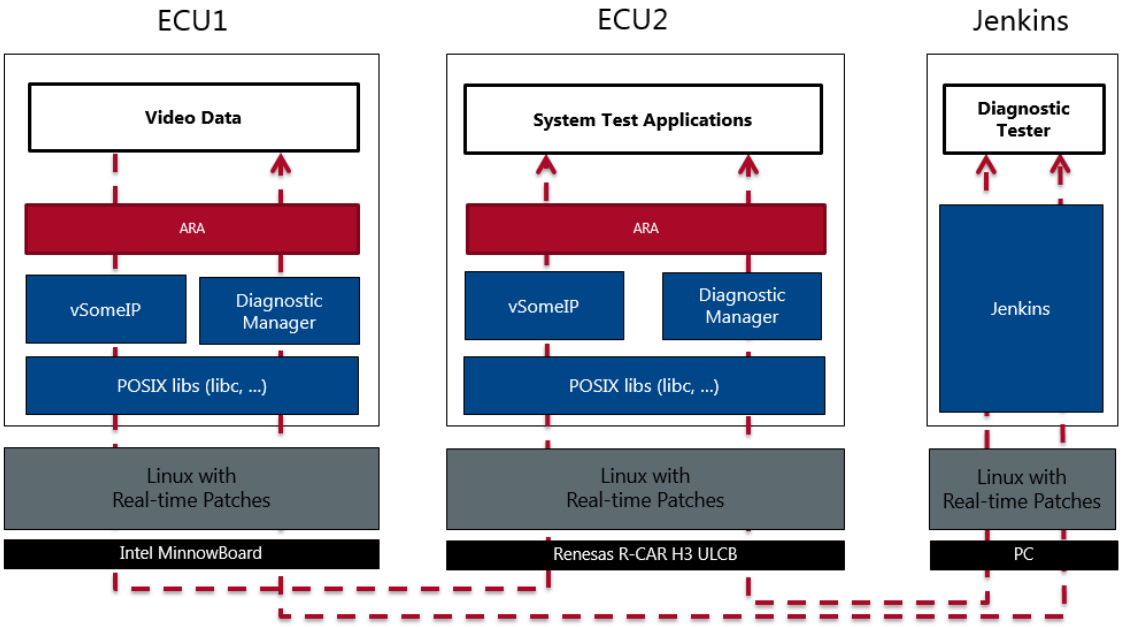

Figure 3: Illustration of test setup for Diagnostics

5.4 Test cases

5.4.1 [STS_DIAG_00001] Utilization of Diagnostic service ReadDataByIdentifier (0x22) by external Tester via UDS messages over DoIP.

Test Objective	Verification of correct behavior of Diagnostic service ReadDataByIdentifier (0x22) by external Tester via UDS messages over DoIP.		
ID	STS_DIAG_00001	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Diagnostic	**State**	Reviewed
Trace to Acceptance Criteria	AP-11-1 AP-11-4		
Reference to Test Environment	STC_DIAG_00001		
Configuration Parameters	- Networking is configured properly. - Diagnostics module: Service instance for service ReadDataByIdentifier with DID 0x0001 is configured. https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json (Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML Model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)		
Summary	This basic test tries to query the number of brake events detected by the EBA on ECU2 over the AP Diagnostics Module. The UDS service ReadDataByIdentifier (0x22) is used. The AP Diagnostics Module has to call a service in the Application Layer to retrieve the requested information and send it back as UDS Response. If an unknown		

	identifier is queried, a Negative Response must be sent. The data read by the service is the number of brake events, detected by the EBA. In order to control this number, a connection to the Video Provider on ECU1, which can switch between a video without emergency brake event and a video with emergency brake event, is necessary.	
Pre-conditions	- Diagnostic Tester is connected to ECU2 via TCP socket on DoIP-Port. - Software components on both ECUs are initialized. - Video Provider sends video without emergency brake event.	
Post-conditions	TCP connection between Diagnostic Tester and ECU2 is closed.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[Diagnostic Tester] Send UDS Request to query number of brake events: UDS-Service: ReadDataByIdentifier UDS-Payload: 0x22 0x0001	
Step 2	[EBA] Wait for invocation.	Implementation of method *Read* for DID 0x0001 is invoked.
Step 3	[EBA] Return number of brake events as DID data.	
Step 4	[Diagnostic Tester] Receive UDS response and store received DID data in variable *numBrakesT1*.	Positive response received (0x62 0xXX …). Payload of UDS response contains DID data.
Step 5	[Diagnostic Tester] Change video to the one with emergency brake event.	
Step 6	[EBA] Run until the full video was processed.	Number of brake events is increased by 1.
Step 7	[Diagnostic Tester] Change back to video without emergency brake event.	
Step 8	[Diagnostic Tester] Send UDS Request to query number of brake events: UDS-Service: ReadDataByIdentifier UDS-Payload: 0x22 0x0001	
Step 9	[EBA] Wait for invocation.	Implementation of method *Read* for DID 0x0001 is invoked.
Step 10	[EBA] Return number of brake events as DID data.	

Step 11	[Diagnostic Tester] Receive UDS response and store received DID data in variable *numBrakesT2*.	Positive response received (0x62 0xXX …). Payload of UDS response contains DID data.
Step 12	[Diagnostic Tester] Compare values of *numBrakesT2* and *numBrakesT1*.	*numBrakesT2* = *numBrakesT1* + 1
Step 13	[Diagnostic Tester] Send UDS Request to query data with a nonimplemented DID: UDS-Service: ReadDataByIdentifier UDS-Payload: 0x22 0x00 0x99	Tester receives Negative Response: 0x7F 0x22 0x31

5.4.2 [STS_DIAG_00002] Utilization of Diagnostic service RoutineControl (0x31) by external Tester via UDS messages over DoIP.

Test Objective	Verification of correct behavior of Diagnostic service RoutineControl (0x31) by external Tester via UDS messages over DoIP.		
ID	STS_DIAG_00002	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Diagnostic	**State**	Reviewed
Trace to Acceptance Criteria	AP-11-1 AP-11-4		
Reference to Test Environment	STC_DIAG_00001		
Configuration Parameters	- Networking is configured properly. - Diagnostics module: Service instance for service RoutineControl with RID 0x0001 is configured and only available in Extended Diagnostic Session. Service Diagnostic Session Control is configured. https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json (Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)		
Summary	This test tries to start a routine in the Video Provider over the AP Diagnostics Module and the UDS service RoutineControl (0x31). In *DefaultSession*, execution is not allowed and a Negative Response is sent. After switching to *ExtendedDiagnosticSession*, the routine is started and a Positive Response is sent.		
Pre-conditions	- Diagnostic Tester is connected to ECU1 via TCP socket on DoIP-Port. - Software components on both ECUs are initialized. - Video Provider sends video without emergency brake event.		
Post-conditions	TCP connection between Jenkins server and ECU1 is closed.		
Main Test Execution			

Test Steps		Pass Criteria
Step 1	[Diagnostic Tester] Send UDS request to change video to the one with emergency brake event: UDS-Service: RoutineControl UDS-Payload: 0x31 0x01 0x00 0x01	Negative Response received: Service Not Supported in Active Session (0x7F 0x31 0x7F).
Step 2	[Diagnostic Tester] Send UDS request to start an Extended Diagnostic Session: UDS-Service: DiagnosticSessionControl UDS-Payload: 0x10 0x03	Positive Response received (0x50 0x03).
Step 3	[Diagnostic Tester] Send UDS request to change video to the one with emergency brake event: UDS-Service: RoutineControl UDS-Payload: 0x31 0x01 0x00 0x01	
Step 4	[Video Provider] Wait for invocation.	Subfunction *Start* of Routine with RID 0x0001 is invoked.
Step 5	[Video Provider] Change video to the one with emergency brake event.	Video is changed to the one with emergency brake event.
Step 6	[Video Provider] Return from Subfunction *Start* of Routine with RID 0x0001.	
Step 7	[Diagnostic Tester] Receive UDS response.	Positive response received (0x71 0xXX …).

5.4.3 [STS_DIAG_00003] Utilization of Diagnostic service TesterPresent (0x3E) by External Tester via UDS messages over DoIP.

Test Objective	Verification of correct behavior of Diagnostic service Tester Present (0x3E) by External Tester via UDS messages over DoIP.		
ID	STS_DIAG_00003	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Diagnostic	**State**	Reviewed
Trace to Acceptance Criteria	AP-11-1 AP-11-4		
Reference to Test Environment	STC_DIAG_00001		
Configuration Parameters	- Networking is configured properly. - Diagnostics module: Service instance for service RoutineControl with RID 0x0001 is configured and only available in Extended Diagnostic Session. Service Diagnostic Session Control and Extended Diagnostic Session time out is configured. https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json		

	(Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)
Summary	Tester present request is sent to indicate that previously activated non-default (e.g. extended) session will still be active. The UDS service RoutineControl (0x31) is executed to check if Extended session is active. Positive response is received for the Tester present request if suppressPosRspMsgIndicationBit is set to FALSE.
Pre-conditions	- Diagnostic Tester is connected to ECU1 via TCP socket on DoIP-Port. - Software components on both ECUs are initialized.
Post-conditions	TCP connection between Jenkins server and ECU1 is closed.
Main Test Execution	
Test Steps	**Pass Criteria**
Step 1	[Diagnostic Tester] Send UDS request to start an Extended Diagnostic Session: UDS-Service: DiagnosticSessionControl UDS-Payload: 0x10 0x03
	Positive Response received (0x50 0x03).
Step 2	[Diagnostic Tester] Wait for some time (e.g. 3 sec) before Extended Diagnostic Session time out
Step 3	[Diagnostic Tester] Send UDS request Tester Present with suppressPosRspMsgIndicationBit is set to FALSE UDS-Service: TesterPresent UDS-Payload: 0x3E 0x00
	Positive Response received (0x7E 0x00).
Step 4	[Diagnostic Tester] Wait for some time (e.g. 3 sec) for Extended Diagnostic Session to time out. Note: Since the Tester Present is sent Extended Session will still be active.
Step 5	[Diagnostic Tester] Send UDS request RoutineControl to confirm if Extended Session is active. UDS-Service: RoutineControl UDS-Payload: 0x31 0x01 0x00 0x01
Step 6	[Diagnostic Tester] Check UDS response.
	Positive response received (0x71 0x01 0x00 0x01 …).
Step 7	[Diagnostic Tester] Stop sending tester present and wait for Extended Diagnostic Session to time out

Step 8	[Diagnostic Tester] Send UDS request RoutineControl to confirm if Extended Session is active. UDS-Service: RoutineControl UDS-Payload: 0x31 0x01 0x00 0x01	Negative Response received: Service Not Supported in Active Session (0x7F 0x31 0x7F).
Step 9	[Diagnostic Tester] Send UDS request to start an Extended Diagnostic Session: UDS-Service: DiagnosticSessionControl UDS-Payload: 0x10 0x03	Positive Response received (0x50 0x03).
Step 10	[Diagnostic Tester] Send UDS request Tester Present with suppressPosRspMsgIndicationBit is set to TRUE UDS-Service: TesterPresent UDS-Payload: 0x3E 0x80	No Response received for UDS request Tester Present
Step 11	[Diagnostic Tester] Send UDS request RoutineControl to confirm if Extended Session is active. UDS-Service: RoutineControl UDS-Payload: 0x31 0x01 0x00 0x01	
Step 12	[Diagnostic Tester] Check UDS response.	Positive response received (0x71 0x01 0x00 0x01 …).

6 Test configuration and test steps for Logging and Tracing

6.1 Acceptance criteria

The following acceptance criteria represent a snapshot from the JIRA system on 01/09/2017.

Vision ID	AP-289	
Title	Debug Log and Trace	
Acceptance Criteria	**ID**	Description
	AP-289-01	A sample application will be set up and create debug information in terms of active logs to DLT. The DLT viewer connected to the network shall be able to display the debug information created by the application even after the sending application has been terminated.
	AP-289-02	A sample application creates debug information and instantaneously be visible at run time by use of a DLT viewer.
	AP-289-03	A sample application creates debug information at a wrong level which is in turn neither displayed at the DLT viewer nor at bus level.
	AP-289-04	A DLT viewer connects to the network, retrieves the available communication partners and displays the DLT messages.
	AP-289-05	A DLT viewer shall change the log level to retrieve and display the DLT messages for the selected level.

6.2 General test objective and approach

6.3 Test System

6.3.1 Test configurations

Configuration ID	STC_LT_00001
Description	Standard CI server for LT test
ECU 1	Intel MinnowBoard Turbot, 192.168.100.5
ECU 2	Renesas R-Car H3 ULCB, 192.168.100.2
Jenkins	Jenkins Server, 192.168.100.10
SW information	Please see doxygen file (https://codeci.autosar.org/view/FT-ST/job/FT-ST_-_ADAS_Application_Test/HTML_Report/)

The Jenkins Server, running the job with the LT Tester, is connected via Ethernet to ECU1 hosting the System Test Application *Video Provider* and ECU2 hosting the System Test Application *Emergency Brake Assistant* (EBA). The LT Tester opens TCP connections on port 3490 and receives log messages from the LT module.

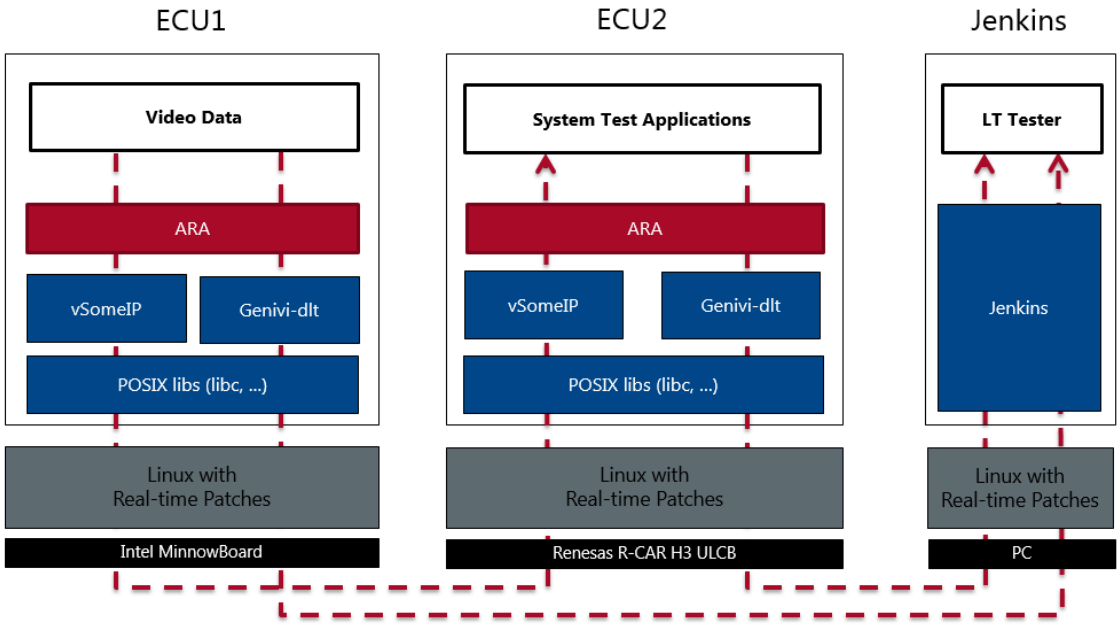

Figure 4: Illustration of test setup for Logging and Tracing.

6.4 Test cases

6.4.1 [STS_LT_00001] Receiving of log messages from LT module by external Tester and remote control of application's default log level.

Test Objective	Verification that all sent log messages from LT module are received by external Tester, that they carry the correct attributes like Application ID and ECU ID, and that the remote control of the application's default log level works.		
ID	STS_LT_00001	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Logging and Tracing	**State**	Reviewed
Trace to Acceptance Criteria	AP-289-01 AP-289-02 AP-289-03 AP-289-04 AP-289-05		
Reference to Test Environment	STC_LT_00001		
Configuration Parameters	- Networking is configured properly. - LT module in ECU1 is configured properly: https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-extended/genivi-dlt/genivi-dlt/dlt-ecu1.conf (Revision 7b852e23558134fb87e9476c1a27b94ebdcf685a) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9) - Video Provider has LT Application ID *VPrv*.		
Summary	The LT Tester has to connect to the LT module, which has to receive and forward the log messages from the Application Layer. First, log messages on all log levels with correct attributes are expected. Then the applications default log level is consecutively		

	lowerd to more restrictive values and it is checked, whether the respective log messages disappear.
Pre-conditions	- LT Tester is connected to ECU1 via TCP socket on Port 3490. - Software components on ECU1 are initialized. - Video Provider's default log level is set to *Verbose*.
Post-conditions	TCP connection between LT Tester and ECU1 is closed.

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT Tester] Receive log messages.	Tester receives log messages every 0.5 seconds. The messages are received for all log levels in context with ID *CTX1* and contain ECU ID *ECU1*, and Application ID *VPrv*.
Step 2	[LT Tester] Send request to query change of Video Provider's default log level to *Debug*.	Messages with log level *Verbose* are no longer received. Messages with lower log level are still coming in.
Step 3	[LT Tester] Send request to query change of Video Provider's default log level to *Info*.	Messages with log level *Debug* are no longer received. Messages with lower log level are still coming in.
Step 4	[LT Tester] Send request to query change of Video Provider's default log level to *Warn*.	Messages with log level *Info* are no longer received. Messages with lower log level are still coming in.
Step 5	[LT Tester] Send request to query change of Video Provider's default log level to *Error*.	Messages with log level *Warn* are no longer received. Messages with lower log level are still coming in.
Step 6	[LT Tester] Send request to query change of Video Provider's default log level to *Fatal*.	Messages with log level *Error* are no longer received. Messages with lower log level are still coming in.
Step 7	[LT Tester] Send request to query change of Video Provider's default log level to *Off*.	No log messages are received.

6.4.2 [STS_LT_00002] Receiving of log messages from LT modules of several ECUs.

Test Objective	Verification that all log messages from multiple ECUs are received and that they carry the correct attributes like Application ID and ECU ID.		
ID	STS_LT_00002	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Logging and Tracing	**State**	Reviewed
Trace to Acceptance	AP-289-01 AP-289-02		

Criteria	AP-289-04	
Reference to Test Environment	STC_LT_00001	
Configuration Parameters	- Networking is configured properly. - LT modules in both ECUs are configured properly. https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-extended/genivi-dlt/genivi-dlt/dlt-ecu1.conf https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-extended/genivi-dlt/genivi-dlt/dlt-ecu2.conf (Revision 7b852e23558134fb87e9476c1a27b94ebdcf685a) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9) - Video Provider has LT Application ID *VPrv*. - EBA has LT Application ID *EBA*.	
Summary	The LT Tester has to connect to the LT modules on the different ECUs. These have to receive and forward the log messages from the different applications in the Application Layers. First, log messages from ECU1 on all log levels with correct attributes are expected. Then a connection to ECU2 is established and additional messages with correct attributes are expected.	
Pre-conditions	- LT Tester is connected to ECU1 via TCP socket on Port 3490. - Video Provider's default log level is set to *Verbose*. - EBA's default log level is set to *Verbose*.	
Post-conditions	TCP connections between Jenkins server and both ECUs are closed.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[LT Tester] Receive log messages.	Tester receives log messages every 0.5 seconds. The messages are received for all log levels in context with ID *CTX1* and contain ECU ID *ECU1*, and Application ID *VPrv*.
Step 2	[LT Tester] Second LT Client connects to ECU2 on Port 3490 using TCP.	Client connected.
Step 3	[LT Tester] Receive log messages	Messages from ECU1 are still received every 0.5 seconds. Tester additionally receives log messages from ECU2 every 0.5 seconds. The additional messages are received for log level Verbose in context with ID *CTX2* and contain ECU ID *ECU2*, and Application ID *EBA*.

7 Test configuration and test steps for Persistency

7.1 Acceptance criteria

The following acceptance criteria represent a snapshot from the JIRA system on 01/09/2017.

Vision ID	AP-386	
Title	Persistent storage of data	
Acceptance Criteria	**ID**	Description
	N/A	*no acceptance criteria in vision*

7.2 General test objective and approach

7.3 Test System

7.3.1 Test configurations

Configuration ID	STC_PER_00001
Description	Standard CI server for Persistency test
ECU 1	Intel MinnowBoard Turbot, 192.168.100.5
Jenkins	Jenkins Server, 192.168.100.10
SW information	Please see doxygen file (https://codeci.autosar.org/view/FT-ST/job/FT-ST_-_ADAS_Application_Test/HTML_Report/)

The Jenkins Server, running the job with the Persistency Tester is connected via Ethernet to ECU1 hosting the Persistency Test Application (PTA). The Persistency Tester is supposed to check the pass criteria.
The communication with the PTA may take place over the Diagnostics functional cluster in form of diagnostic messages. The functionality of the PTA described in the test steps may for example entirely be contained in routines, that are implementation of subroutines of instances of the Diagnostic service *RoutineControl*. This service also provides a means to transport data from the Persistency Tester to the PTA and vice versa.

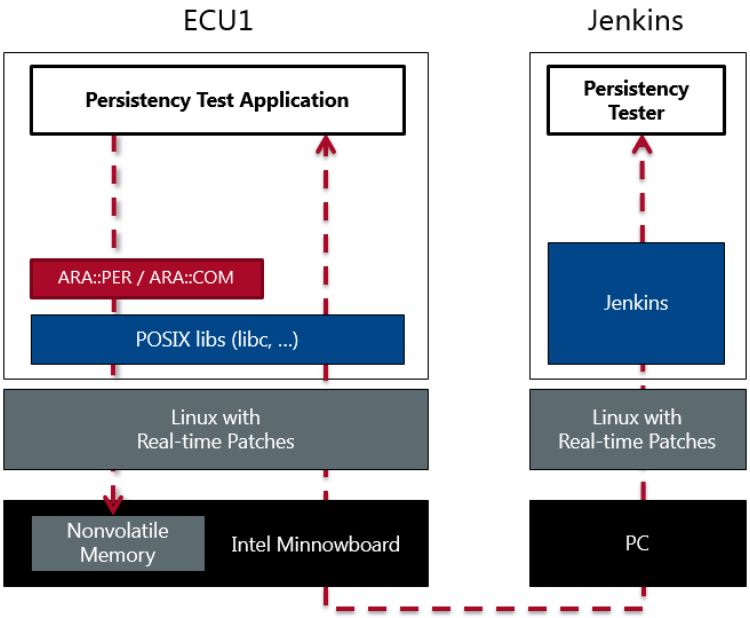

Figure 5: Illustration of test setup for Persistency.

7.4 Test cases

7.4.1 [STS_PER_00001] Storing an integer in a key-value database.

Test Objective	Verification, that integer data can be stored in a key-value database and that it can be retrieved again, using the associated key.		
ID	STS_PER_00001	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Persistency	**State**	Reviewed
Trace to Acceptance Criteria	No acceptance criteria in AP-386		
Reference to Test Environment	STC_PER_00001		
Configuration Parameters	- Networking is configured properly. - File system contains a file for the key-value database. - Diagnostics module in ECU is configured properly: https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json (Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)		
Summary	Integer data is stored in a key-value database. It is then retrieved again from the database using the associated key and the retrieved value is compared to the original one.		
Pre-conditions	- Persistency tester is connected to ECU1. - Software components on ECU1 are initialized. - File for key-value database opened successfully and the file should be empty		

Post-conditions	TCP connection between Persistency Tester and ECU1 is closed.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[PTA] Store integer *intData* with associated key *intKey* in key-value database.	
Step 2	[PTA] Retrieve integer from key-value database using the associated key and store it in variable *retIntData*.	Originally written integer value is returned. And values of *intData* and *retIntData* are equal.

7.4.2 [STS_PER_00002] Storing a float in a key-value database.

Test Objective	Verification, that float data can be stored in a key-value database and that it can be retrieved again, using the associated key.		
ID	STS_PER_00002	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Persistency	**State**	Reviewed
Trace to Acceptance Criteria	No acceptance criteria in AP-386		
Reference to Test Environment	STC_PER_00001		
Configuration Parameters	- Networking is configured properly. - File system contains a file for the key-value database. - Diagnostics module in ECU is configured properly: https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json (Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)		
Summary	Float data is stored in a key-value database. It is then retrieved again from the database using the associated key and the retrieved value is compared to the original one.		
Pre-conditions	- Persistency tester is connected to ECU1. - Software components on ECU1 are initialized. - File for key-value database opened successfully and the file should be empty		
Post-conditions	TCP connection between Jenkins server and ECU1 is closed.		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[PTA] Store float *floatData* with associated key *floatKey* in key-value database.		
Step 2	[PTA]		Originally written float value is

	Retrieve float from key-value database using the associated key and store it in variable *retFloatData*.	returned. And Values of *floatData* and *retFloatData* are equal

7.4.3 [STS_PER_00003] Storing a string in a key-value database.

Test Objective	Verification, that string data can be stored in a key-value database and that it can be retrieved again, using the associated key.		
ID	STS_PER_00003	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Persistency	**State**	Reviewed
Trace to Acceptance Criteria	No acceptance criteria in AP-386		
Reference to Test Environment	STC_PER_00001		
Configuration Parameters	- Networking is configured properly. - File system contains a file for the key-value database. - Diagnostics module in ECU is configured properly: https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json (Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)		
Summary	A string is stored in a key-value database. It is then retrieved again from the database using the associated key and the retrieved value is compared to the original one.		
Pre-conditions	- Persistency tester is connected to ECU1. - Software components on ECU1 are initialized. - File for key-value database opened successfully and the file should be empty		
Post-conditions	TCP connection between Jenkins server and ECU1 is closed.		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[PTA] Store string *stringData* with associated key *stringKey* in key-value database.		
Step 2	[PTA] Retrieve string from key-value database using the associated key and store it in variable *retStringData*.		Originally written string value is returned. And Values of *stringData* and *retStringData* are equal.

7.4.4 [*STS_PER_00004*] *Storing a string in a file.*

Test Objective	Verification that a string can be stored in a file and retrieved again, using a file stream.		
ID	STS_PER_00004	**AUTOSAR Releases**	R17-10
Affected	Persistency	**State**	Reviewed

Functional Cluster			
Trace to Acceptance Criteria	No acceptance criteria in AP-386		
Reference to Test Environment	STC_PER_00001		
Configuration Parameters	- Networking is configured properly. - File system contains a file for the file stream. - Diagnostics module in ECU is configured properly: https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json (Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)		
Summary	A string is stored in a file, using a file stream. It is then retrieved again from the file and the retrieved value is compared to the original one.		
Pre-conditions	- Persistency tester is connected to ECU1. - Software components on ECU1 are initialized. - File stream successfully opened file and the file should be empty		
Post-conditions	TCP connection between Jenkins server and ECU1 is closed.		
Main Test Execution			
Test Steps			**Pass Criteria**
Step 1	[PTA] Write string *stringData* to file via file stream.		
Step 2	[PTA] Close file.		
Step 3	[PTA] Open file.		File opened successfully.
Step 4	[PTA] Retrieve string from file via file stream and store it in variable *retStringData*.		Originally written string value is retrieved. And Values of *stringData* and *retStringData* are equal.

7.4.5 [STS_PER_00005] Storing an integer in a key-value database and retrieving it after reboot.

Test Objective	Verification, that integer data can be stored in a key-value database and, after a reboot, retrieved again using the associated key.		
ID	STS_PER_00005	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Persistency	**State**	Reviewed
Trace to Acceptance Criteria	No acceptance criteria in AP-386		
Reference	STC_PER_00001		

to Test Environment	
Configuration Parameters	- Networking is configured properly. - File system contains a file fort the key-value database. - Diagnostics module in ECU is configured properly: https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json (Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)
Summary	Integer data is stored in a key-value database. A reboot is performed and the integer data is retrieved again from the database. The retrieved value is then compared to the original one.
Pre-conditions	- Persistency tester is connected to ECU1. - Software components on ECU1 are initialized. - File for key-value database opened successfully and the file should be empty
Post-conditions	TCP connection between Jenkins server and ECU1 is closed.

Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[PTA] Store integer *intData* with associated key *intKey* in key-value database.	
Step 2	[Persistency Tester] Request reboot.	
Step 3	[Persistency Tester] Wait until ECU1 has rebooted and PTA is initialized.	
Step 4	[PTA] Open database.	Database file is opened.
Step 5	[PTA] Retrieve integer from key-value database using the associated key and store it in variable *retIntData*.	Originally written integer value is returned. And Values of *intData* and *retIntData* are equal.

7.4.6 [STS_PER_00006] Storing a string in a file and retrieving it after reboot.

Test Objective	Verification, that string data can be stored in a file and, after a reboot, retrieved again using a file stream.		
ID	STS_PER_00006	**AUTOSAR Releases**	R17-10
Affected Functional Cluster	Persistency	**State**	Reviewed
Trace to Acceptance Criteria	No acceptance criteria in AP-386		
Reference	STC_PER_00001		

to Test Environment		
Configuration Parameters	- Networking is configured properly. - File system contains a file for the for the file stream. - Diagnostics module in ECU is configured properly: https://code.autosar.org/tf-apd/ara-api/blob/master/diag/etc/dm_config.json (Revision 247f409adf1382b61e04c16556614168755f5365) https://code.autosar.org/tf-apd/yocto-layers/blob/FT-ST_development/meta-st/recipes-ara/diagnostic-manager/diagnostic-manager/configure_dm.patch (Revision 4948ffdb4600f018befd60832ed7e15902dea746) - ARXML model and VSomeIP configuration: https://code.autosar.org/tf-apd/sample-application/tree/FT-ST_development/ST_Scenarios/Model (Revision dbffbb671b2711e317cd577b2a5720b524ef6ce9)	
Summary	String data is stored in a file using a file stream provided by the Persistency Functional Cluster. A reboot is performed and the string data is retrieved again from the file. The retrieved value is then compared to the original one.	
Pre-conditions	- Persistency tester is connected to ECU1. - Software components on ECU1 are initialized. - File stream successfully opened file and the file should be empty	
Post-conditions	TCP connection between Jenkins server and ECU1 is closed.	
Main Test Execution		
Test Steps		**Pass Criteria**
Step 1	[PTA] Write string *stringData* to file via file stream.	
Step 2	[PTA] Close file.	
Step 3	[Persistency Tester] Request reboot.	
Step 4	[Persistency Tester] Wait until ECU1 has rebooted and PTA is initialized.	
Step 5	[PTA] Open file.	File opened successfully.
Step 6	[PTA] Retrieve string from file via file stream and store it in variable *retStringData*.	Originally written string value is retrieved. And Values of *stringData* and *retStringData* are equal.

8 Reference

[1] Glossary
AUTOSAR_TR_Glossary