Document Title	Requirements on Communication Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	716

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	17-03

Document Change History			
Date	**Release**	**Changed by**	**Description**
2017-03-31	17-03	AUTOSAR Release Management	Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Scope of this document

This document specifies requirements on Communication Management of the AUTOSAR Adaptive Platform.

2 Conventions to be used

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability [1].

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability [1].

3 Terms, acronyms and abbreviations

The glossary below includes terms, acronyms and abbreviations relevant to AP_RS_CommunicationManagement that are not included in the AUTOSAR Glossary [2].

Terms:	**Description:**
Fully qualified service ID	A fully qualified name of a service used as a system-wide unique identifier, e.g. 'com.someOEM.adas.collisionwarner'.

4 Functional overview

The AUTOSAR Adaptive Platform Communication Management provides services for the network and protocol independent communication between applications. This document therefore includes requirements on

- Communication between applications
  - Signature of the communication API
  - Behavior of the communication API
  - On-wire protocol for inter-ECU and inter-machine data communication to classic and adaptive platform
- Service discovery
  - Scope
  - Service registry
  - On-wire protocol for inter-ECU and inter-machine service discovery
- Configuration of middleware for communication aspects (register services)

5 Requirements traceability

The following table references the AUTOSAR main requirements specified in [3] and links to the fulfillment of these.

Feature	Description	Satisfied by
[RS_Main_00060]	AUTOSAR shall provide a standardized software interface for communication between Applications	[RS_CM_00001] [RS_CM_00002] [RS_CM_00101] [RS_CM_00102] [RS_CM_00103] [RS_CM_00104] [RS_CM_00105] [RS_CM_00201] [RS_CM_00202] [RS_CM_00203] [RS_CM_00211] [RS_CM_00212] [RS_CM_00213] [RS_CM_00214] [RS_CM_00215]
[RS_Main_00140]	AUTOSAR shall provide network independent communication mechanisms for applications	[RS_CM_00101] [RS_CM_00102] [RS_CM_00103] [RS_CM_00104] [RS_CM_00105] [RS_CM_00200] [RS_CM_00201] [RS_CM_00204] [RS_CM_00211] [RS_CM_00212] [RS_CM_00213]
[RS_Main_01001]	AUTOSAR shall support intra ECU communication	[RS_CM_00204]
[RS_Main_01002]	AUTOSAR shall support service-oriented communication	[RS_CM_00101] [RS_CM_00102] [RS_CM_00103] [RS_CM_00104] [RS_CM_00105] [RS_CM_00201] [RS_CM_00202] [RS_CM_00203] [RS_CM_00205] [RS_CM_00211] [RS_CM_00212] [RS_CM_00213] [RS_CM_00214] [RS_CM_00215]

6 Requirements specification

6.1 General requirements

[RS_CM_00001] The Communication Management shall provide a standardized header file structure for each service. ⌈

Type:	draft
Description:	The Communication Management shall provide a standardized header file structure for each service. The application uses the standardized header files which are independent of the underlying actual Communication Management implementation.
Rationale:	The application code shall be reusable for different AUTOSAR Adaptive platform implementations.
Dependencies:	–
Use Case:	The application developers implement their code against the standardized header files.
Supporting Material:	–

⌋*(RS_Main_00060)*

[RS_CM_00002] The service header files shall define the namespace for the respective service. ⌈

Type:	draft
Description:	The service header files shall define the namespace for the respective service to uniquely identify each service instance.
Rationale:	The application code shall be reusable for different AUTOSAR Adaptive platform implementations and for different vehicle lines.
Dependencies:	–
Use Case:	To avoid conflicts with other applications and other services each service shall have its own namespace.
Supporting Material:	–

⌋*(RS_Main_00060)*

6.2 Communication between applications

[RS_CM_00200] The Communication Management shall transform Fully Qualified Service IDs to communication protocol specific Service IDs ⌈

Type:	draft

Description:	The Communication Management shall transform Fully Qualified Service IDs to communication protocol specific Service IDs. Fully Qualified Service IDs are used within the application code by the developer and need to be defined to enable cooperation of services of different vendors. Communication protocol specific Service IDs may be used within the messages on the network and may be needed if the communication protocol service ID space was not designed for Fully Qualified Service IDs.
Rationale:	Binary of application shall be unaware of communication protocol specific Service IDs.
Dependencies:	–
Use Case:	One platform is used in multiple vehicle lines but the service IDs are different for the platform in the two vehicle lines. Communication Binding still uses Fully Qualified Service IDs. Communication Management transforms Fully Qualified to SOME/IP Service IDs.
Supporting Material:	see Adaptive Platform Scenarios

⌋*(RS_Main_00140)*

[RS_CM_00204] The Communication Management shall map the protocol independent Service Oriented Communication to the configured protocol binding and shall execute the protocol accordingly. ⌈

Type:	draft
Description:	The Communication Management shall map the protocol independent Service Oriented Communication to the configured protocol binding and shall execute the protocol accordingly. The application code shall use service oriented communication independently of the actually configured protocol. It is the responsibility of the Communication Management to realize the specific protocol.
Rationale:	Binary of application shall be unaware of communication protocol specifics.
Dependencies:	–
Use Case:	One application is used in multiple vehicle lines but the used communication protocols are different in the two vehicle lines, e.g. in one case it uses SOME/IP and in another case it uses local IPC.
Supporting Material:	–

⌋*(RS_Main_00140, RS_Main_01001)*

[RS_CM_00205] The Communication Management shall realize the SOME/IP service discovery protocol and the SOME/IP protocol. ⌈

Type:	draft
Description:	The Communication Management shall realize the SOME/IP service discovery protocol and the SOME/IP protocol. The protocols are described in AUTOSAR SOME/IP Service Discovery Protocol specification and AUTOSAR SOME/IP Protocol specification.
Rationale:	SOME/IP is supported in both AUTOSAR classic and adaptive.
Dependencies:	–
Use Case:	Radar, Camera and SensorFusion applications communicate by Ethernet using SOME/IP protocol

Supporting Material:	–

⌋(*RS_Main_01002*)

6.3 Service discovery

[RS_CM_00101] Communication Management shall provide an interface to offer services ⌈

Type:	draft
Description:	Application developers shall be able to offer services provided by their application for usage by other applications. The services shall be offered using a Fully Qualified Service ID for identification purposes.
Rationale:	To support communication a mechanism is needed to offer provided services to other applications, which are able to use them.
Dependencies:	–
Use Case:	Application “A” offers a wall clock service to other applications.
Supporting Material:	–

s ⌋(*RS_Main_01002*, *RS_Main_00060*, *RS_Main_00140*)

[RS_CM_00102] Communication Management shall provide an interface to find services ⌈

Type:	draft
Description:	Application developers shall be able to find all service instances provided by other applications at runtime.
Rationale:	To establish communication during runtime a mechanism is needed to find provided services based on the type of the service and the concrete service instance.
Dependencies:	–
Use Case:	Application “A” searches for a wall clock service provided by another application. Communication Management finds all available matching service instances and the application can select the right one.
Supporting Material:	–

⌋(*RS_Main_01002*, *RS_Main_00060*, *RS_Main_00140*)

[RS_CM_00103] Communication Management shall provide an interface to subscribe to a specific event provided by an instance of a certain service ⌈

Type:	draft
Description:	Application developers shall be able to subscribe to one specific event inside of one selected service instance.
Rationale:	After finding instances of a service type, it shall be possible to subscribe to certain events of a specific instance.
Dependencies:	–

Use Case:	Application “A” subscribes to the power on event of the application controlling the ignition lock.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00140)*

[RS_CM_00104] Communication Management shall provide an interface to stop the subscription to an event of a service instance ⌈

Type:	draft
Description:	Application developers shall be able to stop an active subscription to an event of a service instance by the application.
Rationale:	After subscribing to an event of a specific service instance, it shall be possible to stop the subscription later on.
Dependencies:	–
Use Case:	Application “A” stops the subscription to the power on event and receives no longer such events.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00140)*

[RS_CM_00105] Communication Management shall provide an interface to stop offering services ⌈

Type:	draft
Description:	Application developers shall be able to stop offering services which the application started to offer before.
Rationale:	After offering a service, it shall be possible to stop the offering of the service later on.
Dependencies:	–
Use Case:	Application “A” stops offering the wall clock service.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00140)*

6.4 Communication

6.4.1 Events

[RS_CM_00201] Communication Management shall provide an API to send events to other applications ⌈

Type:	draft
Description:	Application developers shall be able to provide data in form of events to other applications.
Rationale:	After offering a service, it shall be possible to send events of the respective service to all subscribed applications.
Dependencies:	–

Use Case:	Application “A” sends the power on event upon turning the key in the ignition lock.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00140)*

[RS_CM_00202] Communication Management shall provide an API to the application to poll received events ⌈

Type:	draft
Description:	Application developers shall be able to query whether a certain event has been received from another application and read that data at the same time.
Rationale:	After subscribing to an event of a specific service instance, it shall be possible to receive events send by the server and access them in a polling-based style.
Dependencies:	–
Use Case:	Application “A” polls for the receiving of the power on event and is able to access the corresponding data.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060)*

[RS_CM_00203] Communication Management shall trigger the application on reception of an event ⌈

Type:	draft
Description:	Application developers shall be able to let the platform trigger the application when a new event has been received from another application. The platform shall not deliver the data directly, but instead provide a mechanism to read the data upon request.
Rationale:	After subscribing to an event of a specific service instance, it shall be possible to receive events send by the server and access them in a event-based style through triggering of a processing function.
Dependencies:	–
Use Case:	Application “A” gets triggered whenever receiving the power on event and is able to access the corresponding data.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060)*

6.4.2 Methods

[RS_CM_00211] Communication Management shall provide an interface to provide methods to other applications ⌈

Type:	draft
Description:	Application developers shall be able to provide methods which can be called by other applications.
Rationale:	After offering a service, it shall be possible for other applications to call methods of the service and get the respective result.

Dependencies:	–
Use Case:	Application “A” calls the “getCurrentTime” method of the wall clock service provided by application “B”.

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00140)*

[RS_CM_00212] Communication Management shall provide an interface to call methods of other applications synchronously ⌈

Type:	draft
Description:	Application developers shall be able to synchronously call methods provided by other applications. It is required that the result is available when the method call returns.
Rationale:	After finding a service, it shall be possible for an application to call a method of the service as a synchronous service call: the calling application wants to wait for the completion of the service method execution and have the result available before continuing.
Dependencies:	–
Use Case:	Application “A” calls the “getCurrentTime” method of the wall clock service provided by application “B” and wants to stop processing until the result has been received.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00140)*

[RS_CM_00213] Communication Management shall provide an interface to call service methods asynchronously ⌈

Type:	draft
Description:	Application developers shall be able to asynchronously call methods provided by other applications. It is not required that the result is available when the method call returns. Either the calling application checks for the completion of the service method execution by itself or it is notified on the completion.
Rationale:	After finding a service, it shall be possible for an application to call a method of the service as an asynchronous service call: the calling application does not want to wait for the completion of the service method execution and continues without the result available.
Dependencies:	–
Use Case:	Application “A” calls the “getCurrentTime” method of the wall clock service provided by application “B” asynchronously and can do further processing until the “getCurrentTime” method execution is finished.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060, RS_Main_00140)*

[RS_CM_00214] Communication Management shall provide an interface to query the result of an asynchronously called service method ⌈

Type:	draft

Description:	Application developers shall be able to retrieve the result of an asynchronously called method. The method to query the result can be called at any time after the called method has returned: if it is called before completion of the service method execution, it returns instantly; if it is called after completion, it returns providing the result.
Rationale:	After calling a service method asynchronously, the application shall be able to get the result of this service method.
Dependencies:	–
Use Case:	Application "A" calls the "getCurrentTime" method of the wall clock service provided by application "B" asynchronously. After "getCurrentTime" method execution has completed, Application "A" accesses the result.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060)*

[RS_CM_00215] Communication Management shall trigger the application on completion of an asynchronously called service method ⌈

Type:	draft
Description:	Application developers shall be able to let the platform trigger the application when the result of an asynchronously called method is available.
Rationale:	After asynchronously calling a method of a specific service instance, it shall be possible to trigger a processing function on the availability of the method result.
Dependencies:	–
Use Case:	Application "A" calls the "getCurrentTime" method of the wall clock service provided by application "B" asynchronously. After "getCurrentTime" method execution has completed, Application "A" a specific function of Application "A" is called to notify that the result of "getCurrentTime" is available.
Supporting Material:	–

⌋*(RS_Main_01002, RS_Main_00060)*

6.4.3 Fields

This feature is currrently not supported. It will be supported with R17-10.

7 References

[1] System Template
AUTOSAR_TPS_SystemTemplate

[2] Glossary
AUTOSAR_TR_Glossary

[3] Main Requirements
AUTOSAR_RS_Main