Document Title	Specification of Automotive API Gateway
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	1116

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R25-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2025-11-27	R25-11	AUTOSAR Release Management	• Renamed kNetworkBindingFailure to kCommunicationFailure
2024-11-27	R24-11	AUTOSAR Release Management	• Initial release

Disclaimer

Table of Contents

1 Introduction and functional overview

This specification describes the functionality, API and the configuration for the vehicle service functional cluster , which provide access to a vehicle for non-AUTOSAR clients using a data-centric approach via the `VISS` protocol.

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations that are only relevant to the Automotive API Gateway. A general list of acronyms and abbreviations is available in the [1, AUTOSAR glossary].

Abbreviation / Acronym:	Description:
VISS	The Vehicle Information Service Specification; This document refers to the version 2 of VISS as defined in [2] and [3])
Associated Field	The Field that was created for the VSS leaf according to [4]
Associated ClientServerOperation	The ClientServerOperation that was created for the VSS leaf according to [4].

Table 2.1: Acronyms and abbreviations used in the scope of this Document

3 Related documentation

3.1 Input documents & related standards and norms

[1] Glossary
AUTOSAR_FO_TR_Glossary

[2] Vehicle Information Service Specification - Core
https://github.com/COVESA/vehicle-information-service-specification/releases/download/v2.0/VISSv2_Core.pdf

[3] Vehicle Information Service Specification - Transport
https://github.com/COVESA/vehicle-information-service-specification/releases/download/v2.0/VISSv2_Transport.pdf

[4] Technical Report on VSS Representation
AUTOSAR_AP_TR_VSSRepresentation

[5] Specification of Adaptive Platform Core
AUTOSAR_AP_SWS_Core

[6] Vehicle Signal Specification
https://covesa.github.io/vehicle_signal_specification/

[7] Explanation of Adaptive Platform Software Architecture
AUTOSAR_AP_EXP_SWArchitecture

[8] Requirements on Automotive API Gateway
AUTOSAR_AP_RS_AutomotiveAPIGateway

[9] General Requirements specific to Adaptive Platform
AUTOSAR_AP_RS_General

[10] Specification of Communication Management
AUTOSAR_AP_SWS_CommunicationManagement

[11] Vehicle Information Service Specification
https://github.com/COVESA/vehicle-information-service-specification/releases/tag/v2.0

[12] Specification of Manifest
AUTOSAR_AP_TPS_ManifestSpecification

3.2 Further applicable specification

AUTOSAR provides a core specification [5] which is also applicable for the Automotive API Gateway. The chapter "General requirements for all FunctionalClusters" of this specification shall be considered an additional and required specification for implementing the Automotive API Gateway.

4 Constraints and assumptions

4.1 Known limitations

VSS *aggregates* are not considered by the `Automotive API Gateway`. It treats VSS catalogs as though that parameter did not exist. (see also [6])

The `Automotive API Gateway` rejects update and subscribe requests that address a VSS branch ([AP_SWS_AAG_03322]).

There is currently no standardized mechanism for the `Automotive API Gateway` to know the original capture time of a value. The time that the VISS network binding receives the value is used as a less-precise surrogate timestamp. It is still suitable to make limited assessments about the freshness of the value.

The declarative mapping possibilities are currently not sufficiently elaborated. Although the specification allows for their use it is expected that the declarative style does not yet support all requirements.

5 Dependencies to other Functional Clusters

This chapter defines the dependencies of this functional cluster to other functional clusters. AUTOSAR decided not to standardize interfaces which are exclusively used between functional clusters (on platform-level only), to allow efficient implementations, which might depend e.g., on the used operating system. The goal of this chapter is to provide an informative guideline for the interactions, without specifying syntactical details. This ensures compatibility between documents specifying different functional clusters and supports parallel implementation of different functional clusters. Details of internal interfaces are up to the platform provider. Additional internal interfaces, parameters and return values can be added. A detailed technical architecture documentation of the overall AUTOSAR Adaptive Platform is provided in [7].

5.1 Provided Interfaces

There is no interfaces which will be usable for other functional clusters.

5.2 Required Interfaces

The `Automotive API Gateway` requires interfaces from `Communication Management` and `Log and Trace`.

6 Requirements Tracing

The following tables reference the requirements specified in AUTOSAR RS Automotive API Gateway [8] and the AUTOSAR RS General [9] , and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[AP_RS_AAG_00001]	Enforcement of Datatype Restrictions	[AP_SWS_AAG_00101] [AP_SWS_AAG_00102] [AP_SWS_AAG_00103]
[AP_RS_AAG_00010]	Datatype Mapping	[AP_SWS_AAG_01001]
[AP_RS_AAG_00011]	Unit Mapping	[AP_SWS_AAG_01001]
[AP_RS_AAG_00020]	Mapping to Field	[AP_SWS_AAG_01001] [AP_SWS_AAG_09003] [AP_SWS_AAG_09005]
[AP_RS_AAG_00021]	Mapping to Event	[AP_SWS_AAG_01001]
[AP_RS_AAG_00022]	Mapping to Methods	[AP_SWS_AAG_01001] [AP_SWS_AAG_09004]
[AP_RS_AAG_00023]	Mapping to Structured Datatype	[AP_SWS_AAG_01001]
[AP_RS_AAG_00024]	Mapping of Structured Datatype	[AP_SWS_AAG_02401]
[AP_RS_AAG_00031]	Possible Mapping Targets	[AP_SWS_AAG_01001]
[AP_RS_AAG_00032]	Association to Concrete Instance of a Service Interface	[AP_SWS_AAG_01001]
[AP_RS_AAG_00033]	Mapping of Functionality	[AP_SWS_AAG_03301] [AP_SWS_AAG_03302] [AP_SWS_AAG_03310] [AP_SWS_AAG_03311] [AP_SWS_AAG_03319] [AP_SWS_AAG_03320] [AP_SWS_AAG_03321] [AP_SWS_AAG_03322] [AP_SWS_AAG_03323] [AP_SWS_AAG_03324] [AP_SWS_AAG_03331] [AP_SWS_AAG_03332] [AP_SWS_AAG_03381] [AP_SWS_AAG_03382] [AP_SWS_AAG_03383] [AP_SWS_AAG_09001] [AP_SWS_AAG_09003] [AP_SWS_AAG_09004] [AP_SWS_AAG_09005]
[AP_RS_AAG_00034]	Mapping of Errors	[AP_SWS_AAG_03401] [AP_SWS_AAG_03403] [AP_SWS_AAG_03404] [AP_SWS_AAG_03444] [AP_SWS_AAG_03450] [AP_SWS_AAG_03451] [AP_SWS_AAG_03452]
[AP_RS_AAG_00040]	`VISS` interface	[AP_SWS_AAG_20001] [AP_SWS_AAG_20003] [AP_SWS_AAG_20005]
[AP_RS_AAG_00050]	`VISS` Read of Multiple Leaves	[AP_SWS_AAG_03311]
[AP_RS_AAG_00051]	`VISS` Subscriptions	[AP_SWS_AAG_03320] [AP_SWS_AAG_03323] [AP_SWS_AAG_03324] [AP_SWS_AAG_03331] [AP_SWS_AAG_03332] [AP_SWS_AAG_03381] [AP_SWS_AAG_03382] [AP_SWS_AAG_03383] [AP_SWS_AAG_03403] [AP_SWS_AAG_03444] [AP_SWS_AAG_07001] [AP_SWS_AAG_20011] [AP_SWS_AAG_20012] [AP_SWS_AAG_20013] [AP_SWS_AAG_20014]
[AP_RS_AAG_00052]	Ignore `VSS` Aggregates	[AP_SWS_AAG_01001] [AP_SWS_AAG_20010] [AP_SWS_AAG_20015]
[AP_RS_AAG_00060]	`VISS` Change Filter	[AP_SWS_AAG_20001] [AP_SWS_AAG_20005] [AP_SWS_AAG_20017] [AP_SWS_AAG_20018]
[AP_RS_AAG_00061]	`VISS` Timebased filter	[AP_SWS_AAG_03381] [AP_SWS_AAG_03382] [AP_SWS_AAG_03383] [AP_SWS_AAG_20001] [AP_SWS_AAG_20005] [AP_SWS_AAG_20017] [AP_SWS_AAG_20023]

△

Requirement	Description	Satisfied by
[AP_RS_AAG_00062]	VISS Server Capabilities	[AP_SWS_AAG_04002] [AP_SWS_AAG_20005] [AP_SWS_AAG_20025]
[AP_RS_AAG_00063]	Signal Freshness	[AP_SWS_AAG_03324]
[AP_RS_AAG_00070]	Protection from Overloads	[AP_SWS_AAG_07001]
[AP_RS_AAG_00081]	Signal to Service Capability	[AP_SWS_AAG_01001]

Table 6.1: Requirements Tracing

7 Functional specification

7.1 VISS Interface

[AP_SWS_AAG_20001] VISS Definition

Status: DRAFT
Upstream requirements: AP_RS_AAG_00040, AP_RS_AAG_00060, AP_RS_AAG_00061

⌈The `Automotive API Gateway` shall offer a VISS service according to specifications [2] and [3].⌋

[AP_SWS_AAG_04002] Support VISS Server Capabilities

Status: DRAFT
Upstream requirements: AP_RS_AAG_00062

⌈The `Automotive API Gateway` shall support the `dynamic-metadata` filter with *parameter* set to `server_capabilities` according to the specification [2].⌋

[AP_SWS_AAG_20003] Authorization

Status: DRAFT
Upstream requirements: AP_RS_AAG_00040

⌈The `Automotive API Gateway` shall ensure authorization of requests. See chapter *Access Control Model* of [2].⌋

7.1.1 VISS Features

[AP_SWS_AAG_20005] Rejection of Unsupported Filters

Status: DRAFT
Upstream requirements: AP_RS_AAG_00040, AP_RS_AAG_00060, AP_RS_AAG_00061, AP_RS_AAG_00062

⌈The `Automotive API Gateway` shall respond with an Error Message to a VISS request in case it contains a filter other than

- `timebased`,
- `change`, or
- `dynamic-metadata` with *parameter* set to `server_capabilities`.

The Error Response in this case shall have the Error Reason *forbidden_request* and the associated Error Code and Message according to [3].⌋

[AP_SWS_AAG_03322] Cannot Update or Subscribe to VSS Branches

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033

⌈`Automotive API Gateway` shall respond with an Error Message to a VISS Update or Subscribe request in case it address a VSS branch. The Error Response in this

case shall have the Error Reason *not_implemented*, the Error Code `501`, and Error Message "Update and Subscribe to Branches is not supported".⌋

7.1.2 VISS Errors

[AP_SWS_AAG_00102] Reject Update Requests Violating Datatype Restrictions

Status: DRAFT
Upstream requirements: AP_RS_AAG_00001

⌈The `Automotive API Gateway` shall respond with an Error Message to a VISS Update request in case the given value violates VSS datatype restrictions as defined in [AP_SWS_AAG_00101]. The Error Response in this case shall have the Error Reason *invalid_data* and the associated Error Code and Message according to [3].⌋

[AP_SWS_AAG_00103] Read Values Violating Datatype Restrictions

Status: DRAFT
Upstream requirements: AP_RS_AAG_00001

⌈The `Automotive API Gateway` shall respond with an Error Response to a VISS Read request in case at least one of the response values violates VSS datatype restrictions as defined in [AP_SWS_AAG_00101]. The Error Response in this case shall have the Error Reason *bad_gateway*, the Error Code `502`, and the Error Message "The upstream server response was invalid".⌋

[AP_SWS_AAG_20010] Malformed Subscription Request

Status: DRAFT
Upstream requirements: AP_RS_AAG_00052

⌈The `Automotive API Gateway` shall respond with an Error Response to a VISS Subscribe request in case the request does not contain a filter. The Error Response in this case shall have the Error Reason *missing_trigger*, the Error Code `400`, and the Error Message "Subscription requests require a triggering filter".⌋

[AP_SWS_AAG_20015] Malformed Subscription Request

Status: DRAFT
Upstream requirements: AP_RS_AAG_00052

⌈The `Automotive API Gateway` shall respond with an Error Response to a VISS Subscribe request in case the request contains an invalid filter. The Error Response in this case shall have the Error Reason *invalid_trigger*, the Error Code `400`, and the Error Message "Subscription requests require a valid triggering filter".⌋

[AP_SWS_AAG_20011] Denied Subscription

Status: DRAFT
Upstream requirements: AP_RS_AAG_00051

⌈The `Automotive API Gateway` shall respond with an Error Message to a VISS Subscribe request in case the given VSS path is not available for the subscriber. The

Error Response in this case shall have the Error Reason *forbidden_request* and the associated Error Code and Message according to [3].⌋

[AP_SWS_AAG_20013] Subscription Removal of the Invalid Subscription

Status: DRAFT
Upstream requirements: AP_RS_AAG_00051

⌈The `Automotive API Gateway` shall respond with an Error Message to a VISS Unsubscribe request in case the given `subscriptionId` is not a currently valid subscription identifier. The Error Response in this case shall have the Error Reason *unavailable_data* and the associated Error Code and Message according to [3].⌋

[AP_SWS_AAG_20014] Reject Requests Addressing Non-Existent VSS Nodes

Status: DRAFT
Upstream requirements: AP_RS_AAG_00051

⌈The `Automotive API Gateway` shall respond with an Error Message to a VISS request in case it addresses at least one non-existent VSS node. The Error Response in this case shall have the Error Reason *unavailable_data* and the associated Error Code and Message according to [3].⌋

[AP_SWS_AAG_20017] Read Request with Filters

Status: DRAFT
Upstream requirements: AP_RS_AAG_00060, AP_RS_AAG_00061

⌈The `Automotive API Gateway` shall respond with an Error Message to a VISS Read request in case it contains a change filter or time-based filter. The Error Response in this case shall have the Error Reason *bad_request* and the associated Error Code and Message according to [3].⌋

[AP_SWS_AAG_20018] Change Filter Requests

Status: DRAFT
Upstream requirements: AP_RS_AAG_00060

⌈The `Automotive API Gateway` shall respond with an Error Message to a VISS request in case it contains a filter with invalid logical operators or missing *diff* field (according to [2]). The Error Response in this case shall have the Error Reason *bad_request* and the associated Error Code and Message according to [3].⌋

[AP_SWS_AAG_20023] Malformed Time-based Filter Request

Status: DRAFT
Upstream requirements: AP_RS_AAG_00061

⌈The `Automotive API Gateway` shall respond with an Error Message to a VISS request in case it contains a time-based filter without a time period. The Error Response in this case shall have the Error Reason *bad_request* and the associated Error Code and Message according to [3].⌋

[AP_SWS_AAG_20025] Invalid Dynamic-filter Request

Status: DRAFT
Upstream requirements: AP_RS_AAG_00062

⌈The `Automotive API Gateway` respond with an Error Message to a VISS request in case it contains an invalid dynamic-filter.⌋

The Error Response in this case shall follow this table:

Use case	Error code	Error reason
dynamic-filter combined with path-filter	`403`	*forbidden*
dynamic-filter combined with filters supported by `Automotive API Gateway`	`400`	*bad_request*
dynamic-filter without parameter field	`400`	*bad_request*

7.2 VSS

[AP_SWS_AAG_01001] Model Import of the VSS Catalog

Status: DRAFT
Upstream requirements: AP_RS_AAG_00010, AP_RS_AAG_00011, AP_RS_AAG_00020, AP_RS_AAG_00021, AP_RS_AAG_00022, AP_RS_AAG_00052, AP_RS_AAG_00023, AP_RS_AAG_00031, AP_RS_AAG_00032, AP_RS_AAG_00081

⌈The `Automotive API Gateway` shall be modeled according to [4].⌋

By using the modeling as specified in [4] many model elements including `ServiceInterface` are created. This can then be used to either generate a Skeleton and implement the possibilities required in code or to use existing Metamodel mapping capabilities like the `PassThroughSwConnector` to make the required connections. Either way, this mapping fulfills or helps to fulfill the multiple mapping requirements that this specification item traces up to.

[AP_SWS_AAG_00101] VSS Datatype Restrictions

Status: DRAFT
Upstream requirements: AP_RS_AAG_00001

⌈The `Automotive API Gateway` shall consider a value of a VSS leaf as violating VSS datatype restrictions if one of the conditions is true:

- The value is not in the defined range of the defined VSS datatype according to [6]
- For the leaf the *min* attribute is set and the value of the request is smaller than the value of the *min* attribute
- For the leaf the *max* attribute is set and the value of the request is bigger than the value of the *max* attribute

- The leaf has the datatype string and the value of the request contains characters that are not part of the unicode character set
- For the leaf the *allowed* attribute is set and the value is not part of the specified allowed values
- For the leaf the datatype is an array type and the *arraysize* attribute is set and the value contains an array of a different size

⌋

7.3 VISS Binding

[AP_SWS_AAG_03403] Request to a Service that is not Offered

Status: DRAFT
Upstream requirements: AP_RS_AAG_00034, AP_RS_AAG_00051

⌈When a client requests to read, update, or subscribe to a VSS leaf and the ServiceInterface of the `Associated Field` / `Associated ClientServerOperation` is currently not offered, the `Automotive API Gateway` shall respond with an Error Message with the Error Reason *unavailable_data* and the associated Error Number and Message according to [3].⌋

[AP_SWS_AAG_03324] Timestamp of a VISS Read Response and Subscription Event Message

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00051, AP_RS_AAG_00063

⌈The `Automotive API Gateway` shall set the timestamp of the data point within the VISS read success response and subscription event message to be the timestamp of when the network binding captured the value.⌋

Note: It is planned in a future release to add facilities for a more precise capture time. Currently it is not supported to have the original capture time of the value.

7.3.1 VISS Read

[AP_SWS_AAG_03310] VSS Leaf Read Corresponds To Field Get

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033

⌈For the `Automotive API Gateway` a client read request to a VSS leaf shall correspond with getting the value of the `Associated Field`.⌋

[AP_SWS_AAG_03401] Field Get Error

Status: DRAFT
Upstream requirements: AP_RS_AAG_00034

⌈In case getting the value of an `Associated Field` results in an error the value can not be retrieved. In this case the `Automotive API Gateway` shall respond to the client that requested the read with a VISS Error Message with the Error Reason *service_unavailable* and the associated Error Number and Message according to [3].⌋

[AP_SWS_AAG_03311] VSS Branch Read Correponds to Single Reads

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00050

⌈For the `Automotive API Gateway` a client read request to a VSS branch shall correspond with read requests to all leaves of that branch.
Additional information:

- If the value of leaves can not be retrieved they shall not be part of the response.
- If the value of no leaf can be retrieved an error response according to one of the leaf errors shall be returned to the client.

⌋

Note: That together with [AP_SWS_AAG_03310] *this means it corresponds to reads of the associated fields of all leaves.*

Note: The value of single leaves might not be retrievable e.g., because the associated ServiceInterface is not currently offered, because the client doesn't have the required authorization, or because the retrieval failed with an error. Still, all leaves that could successfully be accessed are intended to be part of the response.

7.3.2 VISS Update

[AP_SWS_AAG_03321] Mapping of an Update Request

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033

⌈A client update request to a VSS leaf of the `Automotive API Gateway` shall correspond with a call of the `Associated ClientServerOperation` with the argument set to the *value* of the update request.⌋

[AP_SWS_AAG_03404] Reaction to an Update Request Function Error

Status: DRAFT
Upstream requirements: AP_RS_AAG_00034

⌈If a `ClientServerOperation` that corresponds to a client update request returns an error the `Automotive API Gateway` shall answer the associated VISS request with a VISS Error Message with the Error Reason *bad_gateway*, the Error Number 502, and the Error Message "The upstream server response was an error.".⌋

7.3.3 VISS Subscription

[AP_SWS_AAG_03320] Mapping of a Subscription Request

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00051

⌈A successful client subscription without the *timebased* filter to a VSS leaf of the `Automotive API Gateway` shall correspond with a subscription to the `Associated Field`.⌋

[AP_SWS_AAG_03381] Mapping of a Timebased Subscription Request

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00051, AP_RS_AAG_00061

⌈A successful client subscription with the *timebased* filter to a VSS leaf of the `Automotive API Gateway` shall correspond with periodically getting the value of the `Associated Field`.⌋

[AP_SWS_AAG_03331] Mapping of an Unsubscribe Request

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00051

⌈A successful client unsubscribe request for a subscription to the `Automotive API Gateway` shall correspond to the termination of the subscription according to [AP_SWS_AAG_03332] or [AP_SWS_AAG_03382].⌋

[AP_SWS_AAG_03332] Mapping of the Termination of a Subscription

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00051

⌈The termination of a subscription without the *timebased* filter to a VSS leaf offered by the `Automotive API Gateway` shall correspond to an unsubscribe from the `Associated Field`.⌋

[AP_SWS_AAG_03382] Mapping of the Termination of a Timebased Subscription

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00051, AP_RS_AAG_00061

⌈The termination of a subscription with the *timebased* filter to a VSS leaf offered by the `Automotive API Gateway` shall correspond to stopping the periodic getting of the value of the `Associated Field` and stopping to send subscription update events to the client.⌋

[AP_SWS_AAG_20012] Subscription after Token Expiriation

Status: DRAFT
Upstream requirements: AP_RS_AAG_00051

⌈If the authorization token has a expiration time the `Automotive API Gateway` shall terminate active subscriptions when the clients authorization token expires.⌋

[AP_SWS_AAG_03319] Initial Value of a Subscription

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033

⌈When a client successfully subscribes to a VSS leaf the `Automotive API Gateway` shall send a subscription event to that client using the current value of the `Associated Field`.⌋

Note: This is not mandated by the VISS standard but is considered useful behavior of the `Automotive API Gateway`.

[AP_SWS_AAG_03323] Conditions for Sending a VISS Subscription Event Message

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00051

⌈The `Automotive API Gateway` shall send a VISS subscription event message to a Client under the following conditions:

- The Client is subscribed without the *timebased* filter to a VSS leaf
- The `Associated Field` of that leaf receives an update
- The updated value does not violate VSS datatype restrictions ([AP_SWS_AAG_00101])
- No change filter for the subscription applies
  or the updated value satisfies the conditions of the change filter according to [2]

⌋

[AP_SWS_AAG_03383] Conditions for Sending a VISS Subscription Event Message for a Timebased subscription

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033, AP_RS_AAG_00051, AP_RS_AAG_00061

⌈The `Automotive API Gateway` shall send a VISS subscription event message to a Client under the following conditions:

- The Client is subscribed with the *timebased* filter to a VSS leaf
- Since the last VISS subscription event message that was sent for this subscription the *period* given in the *timebased* filter expired
- The value that the `Automotive API Gateway` gets of the `Associated Field` does not violate VSS datatype restrictions ([AP_SWS_AAG_00101])
- No change filter for the subscription applies
  or the value satisfies the conditions of the change filter according to [2]

⌋

[AP_SWS_AAG_03444] Reaction to Subscribed Fields Becoming Unavailable

Status: DRAFT
Upstream requirements: AP_RS_AAG_00034, AP_RS_AAG_00051

⌈If a Service Interface becomes unavailable the Automotive API Gateway shall send a VISS Error Message to the clients of all active subscriptions that contain at least one VSS leaf with a mapping to that service and terminate the subscriptions. In this case the Error Message shall contain the Error Reason "unavailable_data" and the associated Error Number and Message according to [3].⌋

Note: The ServiceInterface can become unavailable e.g. if StopOffer is called in the Skeleton.

Note: In case of subscriptions with timebased filter it is implementation defined when the client will be informed about the unavailability of the service. Some options are to send the error message as soon as the service becomes unavailable or to send the error message when the next value would have been sent.

As part of an effort to protect the Automotive API Gateway from getting overloaded a timeout for subscriptions is introduced. It is intended to prevent unused subscriptions from accumulating by automatically terminating them after an implementation defined time. Clients can easily re-subscribe if they still want to get the updates.

[AP_SWS_AAG_07001] Subscription Timeout

Status: DRAFT
Upstream requirements: AP_RS_AAG_00051, AP_RS_AAG_00070

⌈The Automotive API Gateway shall terminate active subscriptions after an implementation defined time after a successful VISS Subscribe request. The Automotive API Gateway shall send a VISS Error Message to the client of this subscription. In this case the Error Message shall contain the Error Reason *request_timeout*, the Error Number 408, and the Error Message "Subscription timed out.".⌋

7.4 C++ Binding

[AP_SWS_AAG_09001] C++ Binding

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033

⌈The Automotive API Gateway shall use the C++ binding as defined in [10, SWS Communication Management]. In case of conflicting specifications this specification shall take precedent over [10, SWS Communication Management] for the Automotive API Gateway.⌋

7.4.1 Field Read

[AP_SWS_AAG_09003] Binding of a Field Read

Status: DRAFT
Upstream requirements: AP_RS_AAG_00020, AP_RS_AAG_00033

⌈The getting of the value of a `Field` of a `ServiceInterface` with VISS deployment shall invoke the registered get handler (see [SWS_CM_00114]) of the `Field` class. The value of the `Field` shall be the returned value of the get handler.
Additional Information:

- In case a `MethodCallProcessingMode` of `kPoll` has been passed to the constructor of the Service Skeleton class (see [SWS_CM_00130]), the get handler shall only be executed upon a call to the `ProcessNextMethodCall` method (see [SWS_CM_00199]) of the Service Skeleton class.
- In case a `MethodCallProcessingMode` of `kEventSingleThread` has been passed to the constructor of the Service Skeleton class (see [SWS_CM_00130]), the get handler shall only be executed upon completion of all previous calls to get handlers of that Skeleton.

⌋

Note: The specification regarding the MethodCallProcessingMode is similar to [SWS_CM_10338], [SWS_CM_10339].

7.4.2 ClientServerOperation Execution

[AP_SWS_AAG_09004] Binding of a ClientServerOperation Execution

Status: DRAFT
Upstream requirements: AP_RS_AAG_00022, AP_RS_AAG_00033

⌈An execution of a `ClientServerOperation` in the role method of a `ServiceInterface` with VISS deployment shall correspond to the execution of the service method (see [SWS_CM_00191]) that was generated for this `ClientServerOperation`. The content of the resolved Future returned by the service method shall correspond to the return of the `ClientServerOperation`.
Additional Information:

- In case a `MethodCallProcessingMode` of `kPoll` has been passed to the constructor of the Service Skeleton class (see [SWS_CM_00130]), the service method shall only be executed upon a call to the `ProcessNextMethodCall` method (see [SWS_CM_00199]) of the Service Skeleton class.
- In case a `MethodCallProcessingMode` of `kEventSingleThread` has been passed to the constructor of the Service Skeleton class (see [SWS_CM_00130]), the service method shall only be executed upon completion of all previous calls to get handlers of that skeleton.

⌋

Note: The return can either be a value or an error.

Note: The specification regarding the MethodCallProcessingMode is similar to [SWS_CM_10306], [SWS_CM_10307].

7.4.3 Subscription

[AP_SWS_AAG_09005] Binding of Field Update

Status: DRAFT
Upstream requirements: AP_RS_AAG_00020, AP_RS_AAG_00033

⌈In the context of the `Automotive API Gateway` an `Associated Field` shall be considered to receive an update when the `Update` function according to [SWS_CM_00119] is called successfully on the respective `Field` class.⌋

Note: This will trigger required subscription event messages according to [AP_SWS_AAG_03323].

Note: This specification is relevant for the programmed solution using a Skeleton. For the declarative approach the update of a field is expressed in relation to the update of a connected field.

7.5 Declarative Mapping

This chapter contains specification items that are only relevant in the declarative mapping case.

[AP_SWS_AAG_03301] Declarative Mapping

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033

⌈If a `providedInterface` of a `PPortPrototype` of an `Automotive API Gateway` is referenced by the `providedOuterPort` of a `PassThroughSwConnector` the `Automotive API Gateway` shall behave according to the semantics of that `PassThroughSwConnector`.⌋

Note: In future releases closer elaborations are planned to better describe how the modelling can be used to create a declarative mapping to other ServiceInterfaces.

Note: In this situation it might still be desired to generate Skeletons. This is allowed but not mandated by the standard so that implementations of the declarative mapping have full freedom for their implementation.

[AP_SWS_AAG_03450] Network Binding Failure Reaction

Status: DRAFT
Upstream requirements: AP_RS_AAG_00034

⌈If during the required interaction with the Service Interfaces due to a VISS request, the Automotive API Gateway receives a ComErrc::kCommunicationFailure it shall reject the VISS request with an Error Message with the Error Reason *bad_gateway*, the Error Number 502, and the Error Message "The upstream server response was invalid.".⌋

[AP_SWS_AAG_03451] Network Binding Failure Reaction for Subscriptions

Status: DRAFT
Upstream requirements: AP_RS_AAG_00034

⌈If during the required interaction with the Service Interfaces due to active VISS subscriptions, the Automotive API Gateway receives a ComErrc::kCommunicationFailure it shall send a VISS Error Message to the clients of these subscriptions and terminate the subscriptions. In this case the Error Message shall contain the Error Reason *bad_gateway*, the Error Number 502, and the Error Message "The upstream server response was invalid.".⌋

[AP_SWS_AAG_03452] Update Request with Invalid Data

Status: DRAFT
Upstream requirements: AP_RS_AAG_00034

⌈If a VISS Update request contains a value that after transformation is not valid according to the Service Interface that the VSS leaf is mapped to, the Automotive API Gateway shall reject the VISS request with an Error Message with the Error Reason *invalid_data*, the Error Number 502, and the Error Message "The upstream server response was invalid.".⌋

[AP_SWS_AAG_02401] Reject Splitting of VSS Struct

Status: DRAFT
Upstream requirements: AP_RS_AAG_00024

⌈The Automotive API Gateway shall reject configurations where a single structured datatype on the VSS side is mapped to multiple types of other ServiceInterfaces.⌋

Note: This still allows to do this mapping in code, it just excludes the declarative solution.

Note: This is currently not allowed because VSS leaves with struct type require atomic reading and writing of the leafs data and this can not be represented with the current configuration. Allowing this could therefore break the semantics of the VSS struct.

[AP_SWS_AAG_03302] Invalid and Unauthorized VISS Requests

Status: DRAFT
Upstream requirements: AP_RS_AAG_00033

⌈Invalid or unauthorized VISS requests shall not result in any interaction with the Service Interfaces of the Automotive API Gateway.⌋

7.6 Functional cluster life-cycle

This section defines behavior of this functional cluster during its life-cycle. Please note that there is a general behavior for ara::core::Initialize and ara::core::Deinitialize defined in [5] by [SWS_CORE_15005] and [SWS_CORE_90022].

7.6.1 Startup

In this release of the Automotive API Gateway the startup of the functional cluster is not yet standardized.

7.6.2 Shutdown

In this release of the Automotive API Gateway the shutdown of the functional cluster is not yet standardized.

7.7 Reporting

In this release of the Automotive API Gateway the reporting is not yet standardized.

7.7.1 Security Events

This functional cluster does not define any security events.

7.7.2 Log Messages

This functional cluster does not define any non-verbose log messages (i.e., modelled DLT messages).

7.7.3 Violation Messages

This functional cluster does not define any violation messages (i.e., DLT messages logged for Violations according to [SWS_CORE_00021]).
Please note that concrete implementations might implement Non-Standardized Violations (see also [SWS_CORE_00003]).

7.7.4 Production Errors

This functional cluster does not define any production errors (i.e., Diagnostic Events).

8 API specification

This functional cluster does not define any C++ APIs for AUTOSAR Adaptive Applications. However, it offers an API for VISS clients as defined by the VISS specification.

The VISS client-server interactions are fully described in VISS specification [11]. As transport protocols it can use HTTP, WebSocket and MQTT which interaction models are publically known.

9 Service Interfaces

The Automotive API Gateway provides service interfaces which specification is out of the scope of standarization.

Service Interface the Automotive API Gateway requires and provides are modeled in ARXML as defined in [12].

The Automotive API Gateway has two kinds of Service Interfaces:

- VISS Service Interfaces (only provided)

  VISS Service Interfaces can be recognized by their VISS deployment and are offered as a VISS service to VISS clients. They are created according to [4].

- ara::com Service Interfaces

  The ara::com Service Interfaces are used by the Gateway to interact with the vehicle internals.

10 Configuration

The Automotive API Gateway at this moment does not specify any configuration.

A Mentioned Manifest Elements

This chapter contains the remaining set of meta-class tables which are not shown directly in the main body of this document.

This chapter is generated.

Class	**ClientServerOperation**			
Note	An operation declared within the scope of a client/server interface.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationInterface.command, *AtpClassifier*.atpFeature, ClientServerInterface.operation, DiagnosticDataElementInterface.read, DiagnosticDataIdentifierInterface.read, DiagnosticDataIdentifierInterface.write, DiagnosticExtendedDataRecordInterface.provide, DiagnosticRoutineInterface.requestResult, DiagnosticRoutineInterface.start, DiagnosticRoutineInterface.stop, PhmRecoveryActionInterface.recovery, ServiceInterface.method			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
argument (ordered)	ArgumentDataPrototype	*	aggr	An argument of this `ClientServerOperation`. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=argument.shortName, argument.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime
fireAndForget	Boolean	0..1	attr	This attribute defines whether this method is a fire&forget method (true) or not (false). This Attribute is only used by the AUTOSAR Adaptive Platform.
possibleApError	ApApplicationError	*	ref	This reference identifies AdaptivePlatformApplicationErrors as a possible error raised by the enclosing ClientServerOperation. This Attribute is only used by the AUTOSAR Adaptive Platform.
possibleApErrorSet	ApApplicationErrorSet	*	ref	This reference represents the ability to refer to an entire group of ApApplicationErrors as one model element instead of having to refer to all the represented ApApplicationErrors separately. This Attribute is only used by the AUTOSAR Adaptive Platform.

Table A.1: ClientServerOperation

Class	**Field**			
Note	This meta-class represents the ability to define a piece of data that can be accessed with read and/or write semantics. It is also possible to generate a notification if the value of the data changes. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationInterface.attribute, *AtpClassifier*.atpFeature, ServiceInterface.field			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
hasGetter	Boolean	0..1	attr	This attribute controls whether read access is foreseen to this field.
hasNotifier	Boolean	0..1	attr	This attribute controls whether a notification semantics is foreseen to this field.
hasSetter	Boolean	0..1	attr	This attribute controls whether write access is foreseen to this field.

Table A.2: Field

Class	**PPortPrototype**			
Note	Component port providing a certain port interface.			
Base	*ARObject*, *AbstractProvidedPortPrototype*, *AtpBlueprintable*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
provided Interface	PortInterface	0..1	tref	The interface that this port provides. **Stereotypes:** isOfType

Table A.3: PPortPrototype

Class	**PassThroughSwConnector**			
Note	This kind of `SwConnector` can be used inside a `CompositionSwComponentType` to connect two delegation `PortPrototypes`.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SwConnector*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
providedOuter Port	AbstractProvidedPort Prototype	0..1	ref	This represents the provided outer delegation Port Prototype of the PassThroughSwConnector.
requiredOuter Port	AbstractRequiredPort Prototype	0..1	ref	This represents the required outer delegation Port Prototype of the PassThroughSwConnector.
serviceInterface Element Mapping	ServiceInterface ElementMapping	*	ref	Reference to a ServiceInterfaceElementMapping specifying the mapping of unequal named Service Interface elements of the two different ServiceInterfaces typing the two PortPrototypes which are referenced by the PassThroughSwConnector. This Attribute is only used by the AUTOSAR Adaptive Platform.

Table A.4: PassThroughSwConnector

Class	**ServiceInterface**			
Note	This represents the ability to define a PortInterface that consists of a heterogeneous collection of methods, events and fields. **Tags:** atp.recommendedPackage=ServiceInterfaces This Class is only used by the AUTOSAR Adaptive Platform.			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
event	VariableDataPrototype	*	aggr	This represents the collection of events defined in the context of a ServiceInterface. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=event.shortName, event.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30

▽

△

Class	ServiceInterface			
field	Field	*	aggr	This represents the collection of fields defined in the context of a ServiceInterface. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=field.shortName, field.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=40
majorVersion	PositiveInteger	0..1	attr	Major version of the service contract. **Tags:** xml.sequenceOffset=10
method	ClientServerOperation	*	aggr	This represents the collection of methods defined in the context of a ServiceInterface. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=method.shortName, method.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=50
minorVersion	PositiveInteger	0..1	attr	Minor version of the service contract. **Tags:** xml.sequenceOffset=20
trigger	Trigger	*	aggr	This represents the collection of triggers defined in the context of a ServiceInterface. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=trigger.shortName, trigger.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=60

Table A.5: ServiceInterface

B Demands and constraints on Base Software (normative)

This functional cluster defines no demands or constraints for the Base Software on which the AUTOSAR Adaptive Platform is running on (usually a POSIX-compatible operating system).

C Platform Extension Interfaces (normative)

This functional cluster does not specify any Platform Extension Interfaces.

D Not implemented requirements

This functional cluster implements all functional requirements specified in the corresponding requirement specifications.

E History of Constraints and Specification Items

Please note that the lists in this chapter also include constraints and specification items that have been removed from the specification in a later version. These constraints and specification items do not appear as hyperlinks in the document.

E.1 Constraint and Specification Item Changes between AUTOSAR Release R24-11 and R25-11

E.1.1 Added Specification Items in R25-11

none

E.1.2 Changed Specification Items in R25-11

Number	Heading
[AP_SWS_AAG_03450]	Network Binding Failure Reaction
[AP_SWS_AAG_03451]	Network Binding Failure Reaction for Subscriptions

Table E.1: Changed Specification Items in R25-11

E.1.3 Deleted Specification Items in R25-11

none

E.2 Constraint and Specification Item Changes between AUTOSAR Release R23-11 and R24-11

E.2.1 Added Specification Items in R24-11

Number	Heading
[AP_SWS_AAG_00101]	VSS Datatype Restrictions
[AP_SWS_AAG_00102]	Reject Update Requests Violating Datatype Restrictions
[AP_SWS_AAG_00103]	Read Values Violating Datatype Restrictions

▽

△

Number	Heading
[AP_SWS_AAG_01001]	Model Import of the VSS Catalog
[AP_SWS_AAG_02401]	Reject Splitting of VSS Struct
[AP_SWS_AAG_03301]	Declarative Mapping
[AP_SWS_AAG_03302]	Invalid and Unauthorized VISS Requests
[AP_SWS_AAG_03310]	VSS Leaf Read Corresponds To Field Get
[AP_SWS_AAG_03311]	VSS Branch Read Correponds to Single Reads
[AP_SWS_AAG_03319]	Initial Value of a Subscription
[AP_SWS_AAG_03320]	Mapping of a Subscription Request
[AP_SWS_AAG_03321]	Mapping of an Update Request
[AP_SWS_AAG_03322]	Cannot Update or Subscribe to VSS Branches
[AP_SWS_AAG_03323]	Conditions for Sending a VISS Subscription Event Message
[AP_SWS_AAG_03324]	Timestamp of a VISS Read Response and Subscription Event Message
[AP_SWS_AAG_03331]	Mapping of an Unsubscribe Request
[AP_SWS_AAG_03332]	Mapping of the Termination of a Subscription
[AP_SWS_AAG_03381]	Mapping of a Timebased Subscription Request
[AP_SWS_AAG_03382]	Mapping of the Termination of a Timebased Subscription
[AP_SWS_AAG_03383]	Conditions for Sending a VISS Subscription Event Message for a Timebased subscription
[AP_SWS_AAG_03401]	Field Get Error
[AP_SWS_AAG_03403]	Request to a Service that is not Offered
[AP_SWS_AAG_03404]	Reaction to an Update Request Function Error
[AP_SWS_AAG_03444]	Reaction to Subscribed Fields Becoming Unavailable
[AP_SWS_AAG_03450]	Network Binding Failure Reaction

▽

△

Number	Heading
[AP_SWS_AAG_03451]	Network Binding Failure Reaction for Subscriptions
[AP_SWS_AAG_03452]	Update Request with Invalid Data
[AP_SWS_AAG_04002]	Support VISS Server Capabilities
[AP_SWS_AAG_07001]	Subscription Timeout
[AP_SWS_AAG_09001]	C++ Binding
[AP_SWS_AAG_09003]	Binding of a Field Read
[AP_SWS_AAG_09004]	Binding of a ClientServerOperation Execution
[AP_SWS_AAG_09005]	Binding of Field Update
[AP_SWS_AAG_20001]	VISS Definition
[AP_SWS_AAG_20003]	Authorization
[AP_SWS_AAG_20005]	Rejection of Unsupported Filters
[AP_SWS_AAG_20010]	Malformed Subscription Request
[AP_SWS_AAG_20011]	Denied Subscription
[AP_SWS_AAG_20012]	Subscription after Token Expiriation
[AP_SWS_AAG_20013]	Subscription Removal of the Invalid Subscription
[AP_SWS_AAG_20014]	Reject Requests Addressing Non-Existent VSS Nodes
[AP_SWS_AAG_20015]	Malformed Subscription Request
[AP_SWS_AAG_20017]	Read Request with Filters
[AP_SWS_AAG_20018]	Change Filter Requests
[AP_SWS_AAG_20023]	Malformed Time-based Filter Request
[AP_SWS_AAG_20025]	Invalid Dynamic-filter Request

Table E.2: Added Specification Items in R24-11

E.2.2 Changed Specification Items in R24-11

none

E.2.3 Deleted Specification Items in R24-11

none