

Document Title	Specification of Remote Persistency
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	1099

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	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	Introduction and Functional Overview	5
2	Definition of Acronyms and Abbreviations	6
2.1	Acronyms and Abbreviations	6
2.2	Definition of Terms	6
3	Related Documentation	7
3.1	Input Documents & Related Standards and Norms	7
3.2	Further Applicable Specifications	7
4	Constraints and Assumptions	8
4.1	Known Limitations	8
5	Dependencies to other Functional Clusters	9
5.1	Provided Interfaces	9
5.2	Required Interfaces	9
6	Requirements Tracing	11
7	Functional Specification	12
7.1	Architecture of Remote Persistency	12
7.1.1	Remote Persistency in the Manifest	12
7.1.1.1	Persistency Interfaces	13
7.1.1.2	Communication Interfaces	13
7.1.1.3	Access Control	14
7.2	Access to Remote Data Items	14
7.2.1	Access by Fields	14
7.2.2	Access by Methods	15
7.3	Update of Remote Data Items	17
7.4	Functional cluster life-cycle	17
7.4.1	Startup	17
7.4.2	Shutdown	18
7.4.3	Daemon Crash	18
7.5	Reporting	18
7.5.1	Security Events	18
7.5.2	Log Messages	19
7.5.3	Violation Messages	19
7.5.4	Production Errors	19
8	API Specification	20
9	Service Interfaces	21
9.1	Implementation Data Types	21
9.2	Provided Service Interfaces	22
9.2.1	Provided Ports	22

9.2.2	Sender-Receiver Interfaces	23
9.2.3	Client-Server Interfaces	23
9.3	Required Service Interfaces	27
9.4	Application Errors	27
10	Configuration	28
10.1	Default Values	28
10.2	Semantic Constraints	28
A	Mentioned Manifest Elements	29
B	Demands and Constraints on Base Software (normative)	45
C	Platform Extension Interfaces (normative)	46
D	Not Implemented Requirements	47
E	Change History of AUTOSAR Traceable Items	48
E.1	Traceable Item History of this Document According to AUTOSAR Re- lease R25-11	48
E.1.1	Added Specification Items in R25-11	48
E.1.2	Changed Specification Items in R25-11	48
E.1.3	Deleted Specification Items in R25-11	48
E.1.4	Added Constraints in R25-11	48
E.1.5	Changed Constraints in R25-11	48
E.1.6	Deleted Constraints in R25-11	49

1 Introduction and Functional Overview

This document is the software specification of the vehicle service `functional cluster Remote Persistency`.

`Remote Persistency` offers mechanisms to `Adaptive Applications` to store information in non-volatile memory, which may be located on a different ECU than the one the `Adaptive Application` is running on. The data is available over boot and ignition cycles, and may be accessed by multiple different `Adaptive Applications`.

The `Remote Persistency` will typically be implemented as a daemon which may run on any ECU, and exposes underlying services of `Persistency` through service interfaces, with access mediated by `IAM`. This daemon would be represented by a `GenericModuleInstantiation` with `category` equal to `REMOTE_PERSISTENCY`.

2 Definition of Acronyms and Abbreviations

2.1 Acronyms and Abbreviations

The table below contains all acronyms, abbreviations or initialisms that are component-specific and not available in the AUTOSAR glossary [1].

Acronym/ Abbreviation	Description
IAM	Identity and Access Management
RDI	Remote Data Item
UCM	Update and Configuration Management

Table 2.1: Acronyms and Abbreviations

2.2 Definition of Terms

The table below contains all definitions of terms that are used in this document.

Term	Description
Adaptive Application	Refers to the Adaptive Application defined in [1, FO TR Glossary].
Communication Management	Functional cluster that provides communication services to Adaptive Applications . See also [2, AP SWS Communication Management].
File Storage	A simple file system without support for nested directories.
Functional Cluster	Refers to the Functional Cluster defined in [1].
Identity and Access Management	A sub-component of the functional cluster Communication Management which restricts access to specific service interfaces.
Key-Value Pair	An element of a Key-Value Storage .
Key-Value Storage	A simple data base for storing Key-Value Pairs .
Persistency	Functional cluster that provides access to non-volatile storage as Key-Value Storage or File Storage . See also [3, AP SWS Persistency].
Remote Data Item	An item of data, which can contain any basic or configurable (potentially structured) C++ type. It is stored as a Key-Value Pair in a Key-Value Storage of Persistency and accessible via <code>ara::com</code> .
Remote Persistency	The software defined in this specification. It offers access to Key-Value Pairs of Key-Value Storages via Communication Management interfaces to the whole system.
Update and Configuration Management	Functional cluster that provides software package update services for Adaptive Applications and other parts of the system. See also [4, AP SWS Update and Configuration Management].

Table 2.2: Definition of Terms 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] Specification of Communication Management
AUTOSAR_AP_SWS_CommunicationManagement
- [3] Specification of Persistency
AUTOSAR_AP_SWS_Persistency
- [4] Specification of Update and Configuration Management
AUTOSAR_AP_SWS_UpdateAndConfigurationManagement
- [5] Specification of Adaptive Platform Core
AUTOSAR_AP_SWS_Core
- [6] Explanation of Adaptive Platform Software Architecture
AUTOSAR_AP_EXP_SWArchitecture
- [7] Requirements on Persistency
AUTOSAR_AP_RS_Persistency
- [8] General Requirements specific to Adaptive Platform
AUTOSAR_AP_RS_General
- [9] Specification of Manifest
AUTOSAR_AP_TPS_ManifestSpecification
- [10] Specification of Platform Types for Adaptive Platform
AUTOSAR_AP_SWS_PlatformTypes
- [11] Specification of Execution Management
AUTOSAR_AP_SWS_ExecutionManagement

3.2 Further Applicable Specifications

AUTOSAR provides a core specification [5] which is also applicable for this functional cluster. The chapter [5] 7.1 “*General requirements for all Functional Clusters*” shall be considered an additional and required specification for implementing this functional cluster.

4 Constraints and Assumptions

4.1 Known Limitations

- The service interface of [Remote Persistency](#) (see [Chapter 9](#)) is in a preliminary state. It is currently not available as a blueprint like the service interfaces of other [functional clusters](#).
- [Remote Persistency](#) does not support access to files. The feature of [File Storages](#) of the underlying [Persistency](#) is not supported.

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 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 between functional clusters 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 [6].

5.1 Provided Interfaces

This functional cluster does not provide any interfaces towards other functional clusters.

5.2 Required Interfaces

This section provides an overview of the public interfaces required by this functional cluster from other functional clusters.

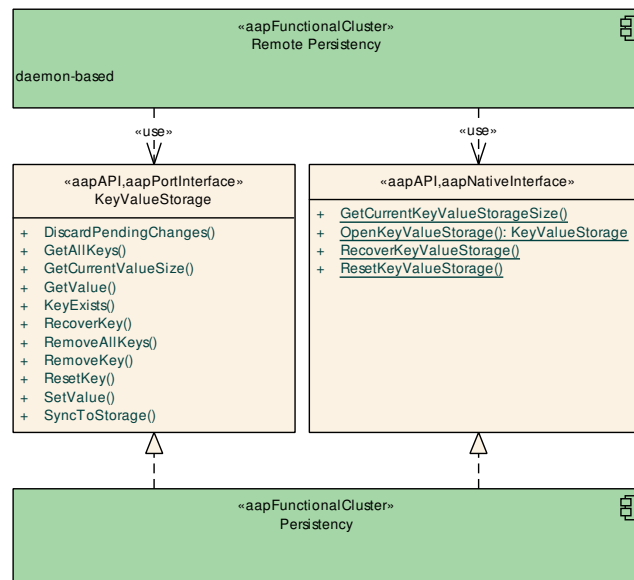


Figure 5.1: Interfaces required by Remote Persistency from other Functional Clusters

Figure 5.1 shows the interfaces required by Remote Persistency from other Functional Clusters within the AUTOSAR Adaptive Platform.

Table 5.1 lists the interfaces required from other Functional Clusters within the AUTOSAR Adaptive Platform and provides a rationale.

Functional Cluster	Interface	Purpose
Persistency	KeyValueStorageOperations	Used to access an underlying KeyValueStorage.
Persistency	KeyValueStorage	Used to access an underlying KeyValueStorage.

Table 5.1: Interfaces required from other Functional Clusters

6 Requirements Tracing

The following tables reference the requirements specified in [7, AUTOSAR RS Persistency] and the [8, AUTOSAR RS General], 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
[RS_PER_00020]	Configurable Layout of Remote Data Items	[AP_SWS_RPER_90001]
[RS_PER_00021]	Initialization and Shutdown	[AP_SWS_RPER_00001] [AP_SWS_RPER_00002] [AP_SWS_RPER_00003] [AP_SWS_RPER_00004]
[RS_PER_00022]	Access to Persisted Remote Data Items	[AP_SWS_RPER_00005] [AP_SWS_RPER_00006] [AP_SWS_RPER_00007] [AP_SWS_RPER_00008] [AP_SWS_RPER_00009] [AP_SWS_RPER_00010] [AP_SWS_RPER_00011] [AP_SWS_RPER_00012] [AP_SWS_RPER_90002] [AP_SWS_RPER_90003] [AP_SWS_RPER_90004] [AP_SWS_RPER_90005] [AP_SWS_RPER_90006] [AP_SWS_RPER_90008] [AP_SWS_RPER_90009] [AP_SWS_RPER_90010] [AP_SWS_RPER_90011] [AP_SWS_RPER_90012] [AP_SWS_RPER_90013] [AP_SWS_RPER_90014]
[RS_PER_00023]	Installation and Update of Remote Data Items	[AP_SWS_RPER_00006] [AP_SWS_RPER_00012]

Table 6.1: Requirements Tracing

7 Functional Specification

7.1 Architecture of Remote Persistency

Remote Persistency integrates Key-Value Storages and service interfaces to provide persistent storage of data items.

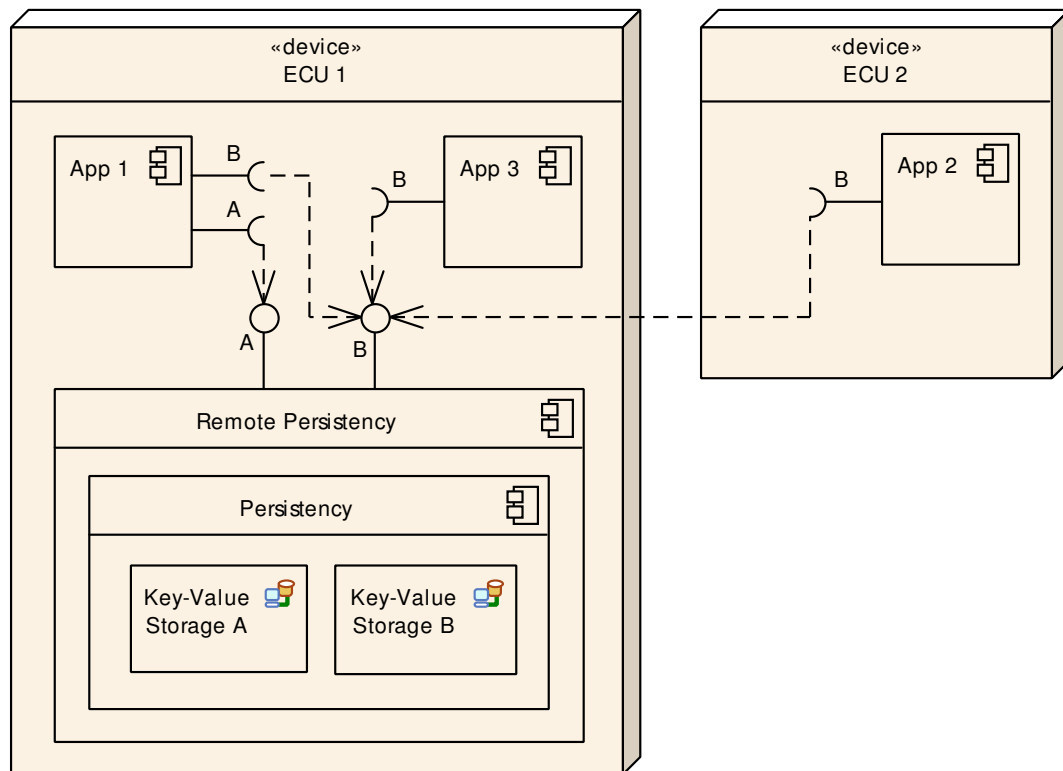


Figure 7.1: Architecture of the Remote Persistency

Remote Persistency uses an internal Key-Value Storage for each provided service interface. Each data item is represented by a field and uses a Key-Value Pair of the corresponding Key-Value Storage to store the data. The methods and fields of the provided service interface of Remote Persistency are described in Chapter 9, while the configuration is defined in Chapter 10.2. The remainder of this section describes the used model elements of the [9, TPS Manifest Specification].

7.1.1 Remote Persistency in the Manifest

Similar to functional clusters that are implemented as a daemon, Remote Persistency is represented by a Process that is linked to the Executable containing the code of Remote Persistency, and which is mapped to a Machine and a NonOsModuleInstantiation by the ProcessToMachineMapping. Remote Persistency uses the derived GenericModuleInstantiation, with category equal to REMOTE_PERSISTENCY.

Each [Key-Value Storage](#) of [Remote Persistency](#) is configured as an [RPort-Prototype](#) typed by a [PersistencyKeyValueStorageInterface](#), and is exposed via a corresponding [PPortPrototype](#) typed by a [ServiceInterface](#). The two interfaces need to be matching, which is ensured by the rules described in [Chapter 10.2](#).

7.1.1.1 Persistency Interfaces

The design time configuration of a [Key-Value Storage](#) of [Remote Persistency](#) is represented by the [PersistencyKeyValueStorageInterface](#), while the [data items](#) are represented by aggregated [PersistencyDataElements](#), which in turn are [AutosarDataPrototypes](#) typed by an [AutosarDataType](#). Initial values can be configured at design time using [PersistencyDataRequiredComSpecs](#) that are aggregated by the same [RPortPrototype](#). Each [PersistencyDataRequiredComSpec](#) refers to the corresponding [PersistencyDataElement](#) and aggregates a [ValueSpecification](#).

The deployment configuration of each designed [Key-Value Storage](#) is represented by a [PersistencyKeyValueStorage](#), which is mapped to the design time configured [RPortPrototype](#) and [Process](#) via a [PersistencyPortPrototypeToKeyValueStorageMapping](#). For single [data items](#), the deployment configuration consists of a [PersistencyKeyValuePair](#) which is aggregated by the [PersistencyKeyValueStorage](#). [PersistencyDataElements](#) and [PersistencyKeyValuePairs](#) are linked by their [shortName](#). Initial values can be configured (overwriting the design time configuration) by a [ValueSpecification](#) aggregated by the [PersistencyKeyValuePair](#). Configuring initial values at deployment time is especially useful for updates of the [Remote Persistency](#).

7.1.1.2 Communication Interfaces

The design time configuration of [Remote Persistency](#) contains a service interface for each [Key-Value Storage](#). The [ServiceInterface](#) needs to provide the fields and methods described in [Chapter 9](#).

Service interfaces are accessible as [RemotePersistencyPort](#) typed by [RemotePersistency](#).

Fields for pre-configured [PersistencyDataElement](#) are configured as [Field](#) where all options ([hasGetter](#), [hasSetter](#), [hasNotifier](#)) are enabled. The [Field](#) needs to be typed by the same [AutosarDataType](#) as the corresponding [PersistencyDataElement](#). They are accessible as [DataElement](#) of type [ElementDataType](#).

Methods are configured as [ClientServerOperations](#). The methods [GetValue](#) and [SetValue](#) need to be created for each [AutosarDataType](#) that is used to type a

`PersistencyDataElement`, where the parameters `value/value` are `ArgumentDataPrototypes` typed by these `AutosarDataType`.

7.1.1.3 Access Control

Access to the service interfaces of `Remote Persistency` is controlled via `IAM`, which is configured with the help of `ComGrants`, specifically `ComFieldGrants` and `ComMethodGrants`.

A `ComFieldGrant` references a specific field represented by a `ServiceFieldDeployment` via `ComFieldGrant.serviceDeployment`, and restricts the possible access via the `FieldAccessEnum` aggregated as `ComFieldGrant.role` to the remote entities identified by `ComFieldGrant.remoteSubject`.

Similarly, a `ComMethodGrant` references a specific method represented by a `ServiceMethodDeployment` via `ComMethodGrant.serviceDeployment`, and restricts the possible access to the remote entities identified by `ComMethodGrant.ComGrant.remoteSubject`.

7.2 Access to Remote Data Items

7.2.1 Access by Fields

The most straight-forward approach to access `data items` of `Remote Persistency` is through the field `DataElement` provided by the service interface `RemotePersistency` for each pre-configured `Key-Value Pair`.

[AP_SWS_RPER_00005] Reading a Field

Upstream requirements: [RS_PER_00022](#)

[When the getter of `DataElement` is called, the `Remote Persistency` shall call `ara::per::KeyValueStorage::GetValue` of the corresponding `Key-Value Storage` with the `shortName` of the corresponding `PersistencyDataElement`. When `ara::per::KeyValueStorage::GetValue` returns successfully, the getter shall return the received value, otherwise it shall return the value configured as `FieldSenderComSpec.initValue`.]

[AP_SWS_RPER_00006] Writing a Field

Upstream requirements: [RS_PER_00022](#), [RS_PER_00023](#)

[When the setter of `DataElement` is called, the `Remote Persistency` shall call `ara::per::KeyValueStorage::SetValue` of the corresponding `Key-Value Storage` with the `shortName` of the corresponding `PersistencyDataElement`, and then the `Key-Value Storage` shall be synchronized using `ara::per::KeyValueStorage::SyncToStorage`.

When `ara::per::KeyValueStorage::SetValue` or `ara::per::KeyValueStorage::SyncToStorage` return an error, the [Remote Persistency](#) shall call `ara::per::KeyValueStorage::DiscardPendingChanges` to return the [Persistency](#) to the previous state, and then update the field to the previous value.

If the value of the field was changed by the setter, the notifications for [DataElement](#) shall be triggered using the `<interface name>::skeleton::fields::<field name>::Update` method.

In any case, the final value of the field shall be returned by the setter.]

Sending a notification even for a failed change of the value can be used as a feedback by the application that initiated the change.

7.2.2 Access by Methods

Using the provided methods of the service interface [RemotePersistency](#), an application can exert more control over its [remote data items](#). [SetValue](#) and [GetValue](#) allow for accessing existing and creating new [data items](#), while the other methods allow for managing the storage.

[AP_SWS_RPER_00007] Getting a List of all Keys

Upstream requirements: [RS_PER_00022](#)

[When [GetAllKeys](#) is called, the [Remote Persistency](#) shall call `ara::per::KeyValueStorage::GetAllKeys` of the corresponding [Key-Value Storage](#) and shall return the list of keys returned by this call.

When `ara::per::KeyValueStorage::GetAllKeys` fails with any error, [GetAllKeys](#) shall return [kStorageFailure](#) instead.]

[AP_SWS_RPER_00008] Checking for the Existence of a Key

Upstream requirements: [RS_PER_00022](#)

[When [KeyExists](#) is called, the [Remote Persistency](#) shall call `ara::per::KeyValueStorage::KeyExists` of the corresponding [Key-Value Storage](#) and shall return the value returned by this call.

When `ara::per::KeyValueStorage::KeyExists` fails with any error, [KeyExists](#) shall return [kStorageFailure](#) instead.]

[AP_SWS_RPER_00009] Removing all Key-Value Pairs

Upstream requirements: [RS_PER_00022](#)

[When [RemoveAllKeys](#) is called, the [Remote Persistency](#) shall call `ara::per::KeyValueStorage::GetAllKeys` of the corresponding [Key-Value Storage](#) to obtain the list of currently available [Key-Value Pairs](#), and then call `ara::per::KeyValueStorage::RemoveKey` for each [Key-Value Pair](#) that is not available as a field.

When the call to `ara::per::KeyValueStorage::GetAllKeys` or any call to `ara::per::KeyValueStorage::RemoveKey` fails with any error apart from `ara::per::PerErrc::kKeyNotFound`, `RemoveAllKeys` shall return `kStorageFailure`.]

[AP_SWS_RPER_00010] Removing a Key-Value Pair

Upstream requirements: [RS_PER_00022](#)

[When `RemoveKey` is called, the `Remote Persistency` shall call `ara::per::KeyValueStorage::RemoveKey` of the corresponding `Key-Value Storage` if the key is not available as a field, and otherwise return `kKeyPreconfigured`.

When the call to `ara::per::KeyValueStorage::RemoveKey` fails with an error, `RemoveKey` shall return the following error:

- `kKeyNotFound` for `ara::per::PerErrc::kKeyNotFound`
- `kStorageFailure` for any other error

]

[AP_SWS_RPER_00011] Getting the Value of a Key-Value Pair

Upstream requirements: [RS_PER_00022](#)

[When `GetValue` is called, the `Remote Persistency` shall call `ara::per::KeyValueStorage::GetValue` of the corresponding `Key-Value Storage` and shall return the value returned by this call.

When the call to `ara::per::KeyValueStorage::GetValue` fails with an error, `GetValue` shall return the following error instead:

- `kDataTypeMismatch` for `ara::per::PerErrc::kDataTypeMismatch`
- `kKeyNotFound` for `ara::per::PerErrc::kKeyNotFound`
- `kStorageFailure` for any other error

]

[AP_SWS_RPER_00012] Setting the Value of a Key-Value Pair

Upstream requirements: [RS_PER_00022](#), [RS_PER_00023](#)

[When `SetValue` is called, the `Remote Persistency` shall call `ara::per::KeyValueStorage::SetValue` of the corresponding `Key-Value Storage`, followed by a call to `ara::per::KeyValueStorage::SyncToStorage`. If the provided key is available as a field, `Remote Persistency` shall additionally call the `<interface name>::skeleton::fields::<field name>::Update` method to trigger the notifications for the corresponding field `DataElement`.

When the call to `ara::per::KeyValueStorage::SetValue` or `ara::per::KeyValueStorage::SyncToStorage` fails with an error, the `Remote Persistency` shall call `ara::per::KeyValueStorage::DiscardPendingChanges`, and `SetValue` shall return the following error:

- `kDataTypeMismatch` for `ara::per::PerErrc::kDataTypeMismatch`
- `kOutOfStorageSpace` for `ara::per::PerErrc::kOutOfStorageSpace` and `ara::per::PerErrc::kQuotaExceeded`
- `kStorageFailure` for any other error

]

7.3 Update of Remote Data Items

`Remote Persistency` knows two entirely different ways how `remote data items` can be updated: Through selected applications or through `UCM`.

Updates by applications are done using the setters of fields (`DataElement`) and the `SetValue` methods of the service interfaces (`RemotePersistency`), while `Remote Persistency` uses `IAM` to configure protection for the write access to the service interfaces, so that only selected applications can change `remote data items`.

Updates through `UCM` are possible for the whole `SoftwarePackage` that contains the `Remote Persistency` or only for the manifests. Both approaches can update and introduce `remote data items`, but only the update of the whole `SoftwarePackage` can add or remove fields or data types. Please refer to chapter 7.2.5 of [3, AP SWS Persistency] and to [4, AP SWS Update and Configuration Management] (see [SWS_UCM_00342] for manifest updates) for details.

7.4 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].

7.4.1 Startup

Upon startup, `Remote Persistency` initializes the `functional clusters` and all fields, and starts offering the field's events for subscription.

[AP_SWS_RPER_00001] Initialize Functional Clusters

Upstream requirements: `RS_PER_00021`

[`Remote Persistency` shall initialize the used `functional clusters` by calling `ara::core::Initialize` at process startup before creating any threads.]

[AP_SWS_RPER_00002] Open Key-Value Storages, Initialize Fields, and Start Events

Upstream requirements: [RS_PER_00021](#)

[After fulfilling [\[AP_SWS_RPER_00001\]](#), [Remote Persistency](#) shall open all configured [Key-Value Storages](#) by calling `ara::per::OpenKeyValueStorage` with the corresponding `ara::core::InstanceSpecifier`, set all fields to the values stored in the configured [Key-Value Storages](#), register the setters and getters of all fields, and start offering the service for each [Key-Value Storage](#).

When `ara::per::OpenKeyValueStorage` returns an error for any [Key-Value Storage](#), [Remote Persistency](#) shall try to recover this [Key-Value Storage](#), and if that also fails, it shall not offer the service for this [Key-Value Storage](#).]

7.4.2 Shutdown

Upon shutdown, [Remote Persistency](#) de-initializes the [functional clusters](#) and stops offering the field's events.

[AP_SWS_RPER_00003] Stop Events and Close Key-Value Storages

Upstream requirements: [RS_PER_00021](#)

[When shutting down, [Remote Persistency](#) shall first stop offering the events of all fields and then close all [Key-Value Storages](#) by destructing the `ara::per::SharedHandles`.]

[AP_SWS_RPER_00004] Deinitialize Functional Clusters

Upstream requirements: [RS_PER_00021](#)

[When shutting down, [Remote Persistency](#) shall de-initialize the used [functional clusters](#) by calling `ara::core::Deinitialize` after fulfilling [\[AP_SWS_RPER_00003\]](#) and after joining all threads.]

7.4.3 Daemon Crash

When [Remote Persistency](#) is restarted after a crash (typically by PHM), it shall initialize itself again according to [\[AP_SWS_RPER_00001\]](#) and [\[AP_SWS_RPER_00002\]](#).

7.5 Reporting**7.5.1 Security Events**

This functional cluster does not define any security events.

7.5.2 Log Messages

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

7.5.3 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]).

7.5.4 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 APIs.

9 Service Interfaces

This chapter lists the service interfaces provided or required by this functional cluster.

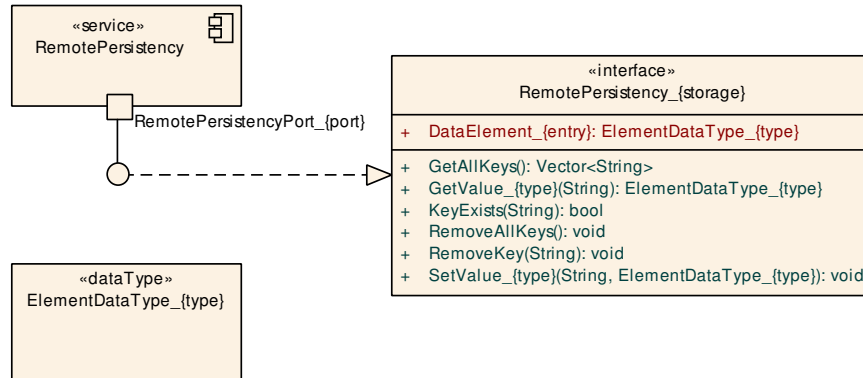


Figure 9.1: Service Interface of the **Remote Persistency**

9.1 Implementation Data Types

Remote Persistency does not define any local implementation data types, it only uses implementation data types defined in the manifest for the stored **Remote Data Items**. It is expected that the same implementation data types are referenced by the **PersistencyKeyValueStorageInterface** and the **ServiceInterface** (see [AP_SWS_RPER_CONSTR_00001]).

[AP_SWS_RPER_90003] Definition of ImplementationDataType ElementDataType_{type}

Upstream requirements: [RS_PER_00022](#)

[

Name	ElementDataType_{type}
Namespace	ara::rper
Kind	AutosarDataType
Derived from	-
Description	Any type {type} defined in the AP SWS Platform Types [10] or a type configured as CppImplementationDataType or ApplicationDataType .
Variation	{type} represents the shortName of an AutosarDataType that is represented by this implementation data type.

]

[AP_SWS_RPER_90009] Definition of ImplementationDataType KeyNameType*Upstream requirements:* [RS_PER_00022](#)

[

Name	KeyNameType
Namespace	ara::rper
Kind	STRING
Derived from	-
Description	Key name

]

[AP_SWS_RPER_90008] Definition of ImplementationDataType ListOfKeyNames Type*Upstream requirements:* [RS_PER_00022](#)

[

Name	ListOfKeyNamesType
Namespace	ara::rper
Kind	VECTOR < KeyNameType >
Derived from	-
Description	List of key names

]

9.2 Provided Service Interfaces

[Remote Persistency](#) provides one interface for each [Key-Value Storage](#) that is configured for the underlying [Persistency](#). This interface contains one field for each [Key-Value Pair](#) defined for the [Key-Value Storage](#), and a set of generic methods to access the storage.

9.2.1 Provided Ports

[AP_SWS_RPER_90001] Definition of Port RemotePersistencyPort_{port} provided by functional cluster RPer*Upstream requirements:* [RS_PER_00020](#)

[

Name	RemotePersistencyPort_{port}		
Kind	ProvidedPort	Interface	RemotePersistency
Description	Provided Port of Remote Persistency for a specific Key-Value Storage .		
Variation	{port} represents the shortName of an RPortPrototype that is typed by a PersistencyKeyValueStorageInterface .		

]

9.2.2 Sender-Receiver Interfaces

Remote Persistency does not define any sender-receiver interfaces.

9.2.3 Client-Server Interfaces

[AP_SWS_RPER_90004] Definition of ServiceInterface RemotePersistency_{storage}

Upstream requirements: [RS_PER_00022](#)

[

Name	RemotePersistency_{storage}
Namespace	ara::rper
Version	1.0
Fields	DataElement
Methods	• GetValue • SetValue • GetAllKeys • KeyExists • RemoveKey • RemoveAllKeys
Variation	{storage} represents the shortName of a PersistencyKeyValueStorageInterface .

]

[AP_SWS_RPER_90005] Definition of Field RemotePersistency.DataElement_{entry}

Upstream requirements: [RS_PER_00022](#)

[

Field	DataElement_{entry}
Description	Field for accessing the configured Key-Value Pair {entry} with a value of type ElementDataType .
Version	1.0
Type	ElementDataType
HasGetter	true
HasNotifier	true
HasSetter	true
Enclosing Service Interface	RemotePersistency
Variation	{entry} represents the shortName of a PersistencyDataElement of the PersistencyKeyValueStorageInterface associated with this service interface.

]

[AP_SWS_RPER_90013] Definition of Method RemotePersistency.GetValue_{type}

Upstream requirements: [RS_PER_00022](#)

[

Method	GetValue_{type}	
Description	Read the value of a data item of type ElementDataType .	
Version	1.0	
FireAndForget	false	
Parameter	key	
	Description	Key name.
	Type	KeyNameType
	Variation	
	Direction	IN
Parameter	value	
	Description	Value of the data item.
	Type	ElementDataType
	Variation	
	Direction	OUT
Application Errors	kDataTypeMismatch	The GetValue or SetValue method does not match the stored data type.
Application Errors	kKeyNotFound	The provided key cannot be not found in the Key-Value Storage .
Application Errors	kStorageFailure	An error occurred when accessing the Key-Value Storage , e.g. physical or logical corruption or redundancy or cryptographic errors.
Enclosing Service Interface	RemotePersistency	
Variation	{type} represents the shortName of the AutosarDataType that is accessible via this method.	

]

[AP_SWS_RPER_90014] Definition of Method RemotePersistency.SetValue_{type}

Upstream requirements: [RS_PER_00022](#)

[

Method	SetValue_{type}	
Description	Update the value of a data item of type {type}, or create the item if it does not exist.	
Version	1.0	
FireAndForget	false	
Parameter	key	
	Description	Key name.
	Type	KeyNameType
	Variation	
	Direction	IN
Parameter	value	
	Description	Value of the data item.

▽



	Type	ElementDataType
	Variation	
	Direction	IN
Application Errors	kDataTypeMismatch	The GetValue or SetValue method does not match the stored data type.
Application Errors	kOutOfStorageSpace	The available storage space was exceeded by the value provided via SetValue .
Application Errors	kStorageFailure	An error occurred when accessing the Key-Value Storage , e.g. physical or logical corruption or redundancy or cryptographic errors.
Enclosing Service Interface	RemotePersistency	
Variation	{type} represents the shortName of the AutosarDataType that is accessible via this method.	

]

[AP_SWS_RPER_90006] Definition of Method RemotePersistency.GetAllKeys

Upstream requirements: [RS_PER_00022](#)

[

Method	GetAllKeys	
Description	Get the names of all data items of this storage which have been configured in the manifest or were introduced at runtime.	
Version	1.0	
FireAndForget	false	
Parameter	keys	
	Description	List of key names.
	Type	ListOfKeyNamesType
	Variation	
	Direction	OUT
Application Errors	kStorageFailure	An error occurred when accessing the Key-Value Storage , e.g. physical or logical corruption or redundancy or cryptographic errors.
Enclosing Service Interface	RemotePersistency	

]

[AP_SWS_RPER_90010] Definition of Method RemotePersistency.KeyExists

Upstream requirements: [RS_PER_00022](#)

[

Method	KeyExists	
Description	Check if a data item has been configured in the manifest or was introduced at runtime.	
Version	1.0	
FireAndForget	false	
Parameter	key	
	Description	Key name.





	Type	KeyNameType
	Variation	
	Direction	IN
Parameter	exists	
	Description	True if the key exists, false otherwise.
	Type	bool
	Variation	
	Direction	OUT
Application Errors	kStorage-Failure	An error occurred when accessing the Key-Value Storage, e.g. physical or logical corruption or redundancy or cryptographic errors.
Enclosing Service Interface	RemotePersistency	

]

[AP_SWS_RPER_90012] Definition of Method RemotePersistency.RemoveKey

Upstream requirements: RS_PER_00022

[

Method	RemoveKey	
Description	Remove a data item from the storage that was created at run-time or by deployment.	
	Please note: Data items that are available as a field cannot be removed.	
Version	1.0	
FireAndForget	false	
Parameter	key	
	Description	Key name.
	Type	KeyNameType
	Variation	
	Direction	IN
Application Errors	kKeyPreconfigured	The provided key cannot be removed from the Key-Value Storage because it represents a field.
Application Errors	kStorage-Failure	An error occurred when accessing the Key-Value Storage, e.g. physical or logical corruption or redundancy or cryptographic errors.
Enclosing Service Interface	RemotePersistency	

]

[AP_SWS_RPER_90011] Definition of Method RemotePersistency.RemoveAllKeys

Upstream requirements: [RS_PER_00022](#)

Method	RemoveAllKeys	
Description	Remove all data items from this storage that were created at run-time or by deployment. Please note: Data items that are available as a field will not be removed.	
Version	1.0	
FireAndForget	false	
Application Errors	kStorage-Failure	An error occurred when accessing the Key-Value Storage , e.g. physical or logical corruption or redundancy or cryptographic errors.
Enclosing Service Interface	RemotePersistency	

9.3 Required Service Interfaces

[Remote Persistency](#) does not require service interfaces.

9.4 Application Errors

[AP_SWS_RPER_90002] Definition of Application Error Domain of functional cluster RPer

Upstream requirements: [RS_PER_00022](#)

Name	Code	Description
kDataTypeMismatch	1	The GetValue or SetValue method does not match the stored data type.
kKeyNotFound	2	The provided key cannot be not found in the Key-Value Storage .
kKeyPreconfigured	3	The provided key cannot be removed from the Key-Value Storage because it represents a field.
kOutOfStorageSpace	4	The available storage space was exceeded by the value provided via SetValue .
kStorageFailure	5	An error occurred when accessing the Key-Value Storage , e.g. physical or logical corruption or redundancy or cryptographic errors.

10 Configuration

The configuration model of this functional cluster is defined in [9]. This chapter defines the default values for attributes and semantic constraints for elements specified in [9] that are part of the configuration model of this functional cluster.

`Remote Persistency` is described by a `GenericModuleInstantiation` that relates to an `AdaptiveApplicationSwComponentType` with at least one `RPortPrototype` typed by a `PersistencyKeyValueStorageInterface` and at least one `PPortPrototype` typed by a `ServiceInterface`.

10.1 Default Values

This functional cluster does not define any default values for attributes specified in [9].

10.2 Semantic Constraints

This section defines semantic constraints for elements specified in [9] that are part of the configuration model of this functional cluster.

[AP_SWS_RPER_CONSTR_00001] Synchronicity of Interface [For every `Key-Value Storage` (represented by an `RPortPrototype` typed by a `PersistencyKeyValueStorageInterface` in the application design for the respective `AdaptiveApplicationSwComponentType`), there shall be a corresponding `ServiceInterface` (represented by an `PPortPrototype` in the application design for the respective `AdaptiveApplicationSwComponentType`), and vice versa.]

[AP_SWS_RPER_CONSTR_00002] Type of PersistencyInterface [The `access-Mode` of all configured `Key-Value Storages` shall be set to `readWrite`.]

[AP_SWS_RPER_CONSTR_00003] Synchronicity of Data Types [For every `PersistencyDataElement` aggregated as `dataElement` by `PersistencyKeyValueStorageInterface` that is typed by an `AutosarDataType`, there shall be a `Field` aggregated as `field` by the corresponding `ServiceInterface` that is typed by the same `AutosarDataType`, and vice versa.]

[AP_SWS_RPER_CONSTR_00004] Synchronicity of Init Values [The initial value configured for a `PersistencyDataElement` by `PersistencyDataRequiredComSpec.initValue` shall be identical to the initial value configured for the corresponding `Field` by `FieldSenderComSpec.initValue`.]

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	AbstractPersistencyRequireComSpec (abstract)			
Note	This meta-class acts as a bas-class of persistency-related ComSpec classes. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject, RPortComSpec			
Subclasses	PersistencyDataRequiredComSpec, PersistencyFileRequiredComSpec			
Aggregated by	AbstractRequiredPortPrototype.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	Type	Mult.	Kind	Note
accessMode	PersistencyAccess Enum	0..1	attr	This attribute controls how persistent data can be accessed.

Table A.1: AbstractPersistencyRequireComSpec

Class	AdaptiveApplicationSwComponentType			
Note	This meta-class represents the ability to support the formal modeling of application software on the AUTOSAR adaptive platform. Consequently, it shall only be used on the AUTOSAR adaptive platform. Tags: atp.recommendedPackage=AdaptiveApplicationSwComponentTypes This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, SwComponentType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
internalBehavior	AdaptiveSwcInternal Behavior	0..1	aggr	This aggregation represents the internal behavior of the AdaptiveApplicationSwComponentType for the AUTOSAR adaptive platform. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=internalBehavior.shortName, internal Behavior.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.2: AdaptiveApplicationSwComponentType

Class	ApplicationDataType (abstract)			
Note	ApplicationDataType defines a data type from the application point of view. Especially it should be used whenever something "physical" is at stake. An ApplicationDataType represents a set of values as seen in the application model, such as measurement units. It does not consider implementation details such as bit-size, endianness, etc. It should be possible to model the application level aspects of a VFB system by using ApplicationDataTypes only.			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	ApplicationCompositeDataType, ApplicationPrimitiveDataType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
—	—	—	—	—

Table A.3: ApplicationDataType

Class	ArgumentDataPrototype			
Note	An argument of an operation, carries direction and implementation information.			
Base	<i>ARObject</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , AutosarDataPrototype , <i>DataPrototype</i> , Identifiable , <i>MultilanguageReferrable</i> , Referrable			
Aggregated by	<i>AtpClassifier.atpFeature</i> , ClientServerOperation.argument			
Attribute	Type	Mult.	Kind	Note
direction	ArgumentDirectionEnum	0..1	attr	This attribute specifies the direction of the argument.
serverArgumentImplPolicy	ServerArgumentImplPolicyEnum	0..1	attr	This defines how the argument type of the servers <code>RunnableEntity</code> is implemented. If the attribute is not defined this has the same semantics as if the attribute is set to the value <code>useArgumentType</code> for primitive arguments and structures.

Table A.4: ArgumentDataPrototype

Class	AutosarDataPrototype (abstract)			
Note	Base class for prototypical roles of an AutosarDataType .			
Base	<i>ARObject</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , <i>DataPrototype</i> , Identifiable , <i>MultilanguageReferrable</i> , Referrable			
Subclasses	ArgumentDataPrototype , Field , <i>ParameterDataPrototype</i> , PersistencyDataElement , <i>VariableDataPrototype</i>			
Aggregated by	<i>AtpClassifier.atpFeature</i>			
Attribute	Type	Mult.	Kind	Note
type	AutosarDataType	0..1	tref	This represents the corresponding data type. Stereotypes: <code>isOfType</code>

Table A.5: AutosarDataPrototype

Class	AutosarDataType (abstract)			
Note	Abstract base class for user defined AUTOSAR data types for software.			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>AtpClassifier</i> , <i>AtpType</i> , <i>CollectableElement</i> , Identifiable , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , Referrable			
Subclasses	<i>AbstractImplementationDataType</i> , ApplicationDataType			
Aggregated by	<i>ARPackage.element</i>			
Attribute	Type	Mult.	Kind	Note
swDataDefProps	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType. Stereotypes: <code>atpSplitable</code> Tags: <code>atp.Splitkey=swDataDefProps</code>

Table A.6: AutosarDataType

Class	ClientServerOperation			
Note	An operation declared within the scope of a client/server interface.			
Base	<i>ARObject</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , Identifiable , <i>MultilanguageReferrable</i> , Referrable			
Aggregated by	<i>ApplicationInterface.command</i> , <i>AtpClassifier.atpFeature</i> , <i>ClientServerInterface.operation</i> , <i>DiagnosticDataElementInterface.read</i> , <i>DiagnosticDataIdentifierInterface.read</i> , <i>DiagnosticDataIdentifierInterface.write</i> , <i>DiagnosticExtendedDataRecordInterface.provide</i> , <i>DiagnosticRoutineInterface.requestResult</i> , <i>DiagnosticRoutineInterface.start</i> , <i>DiagnosticRoutineInterface.stop</i> , <i>PhmRecoveryActionInterface.recovery</i> , ServiceInterface.method			
Attribute	Type	Mult.	Kind	Note





Class	ClientServerOperation			
argument (ordered)	ArgumentDataPrototype	*	aggr	An argument of this <code>ClientServerOperation</code> . Stereotypes: <code>atpSplittable</code> ; <code>atpVariation</code> Tags: <code>atp.Splitkey=argument.shortName</code> , <code>argument.variationPoint.shortLabel</code> <code>vh.latestBindingTime=blueprintDerivationTime</code>
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	<code>ApApplicationError</code>	*	ref	This reference identifies <code>AdaptivePlatformApplicationErrors</code> as a possible error raised by the enclosing <code>ClientServerOperation</code> . This Attribute is only used by the AUTOSAR Adaptive Platform.
possibleApError Set	<code>ApApplicationErrorSet</code>	*	ref	This reference represents the ability to refer to an entire group of <code>ApApplicationErrors</code> as one model element instead of having to refer to all the represented <code>ApApplicationErrors</code> separately. This Attribute is only used by the AUTOSAR Adaptive Platform.

Table A.7: ClientServerOperation

Class	ComFieldGrant			
Note	This meta-class represents the ability to grant access to a <code>ServiceInterface.field</code> . Tags: <code>atp.Status=candidate</code> <code>atp.recommendedPackage=Grants</code> This Class is only used by the AUTOSAR Adaptive Platform.			
Base	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , ComGrant , <i>Grant</i> , Identifiable , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , Referrable , <i>UploadableDeploymentElement</i> , <i>UploadablePackageElement</i>			
Aggregated by	<code>ARPackage.element</code>			
Attribute	Type	Mult.	Kind	Note
design	<code>ComFieldGrantDesign</code>	0..1	ref	This reference identifies the <code>ComFieldGrantDesign</code> that the enclosing <code>ComFieldGrant</code> was created from. Stereotypes: <code>atpUriDef</code> Tags: <code>atp.Status=candidate</code>
role	FieldAccessEnum	0..1	attr	This attribute provides the ability to further specify the access to the <code>ServiceInterface.field</code> . Tags: <code>atp.Status=candidate</code>
service Deployment	ServiceField Deployment	0..1	ref	This reference identifies the applicable deployment within the context of an <code>AdaptivePlatformServiceInstance</code> for which the grant applies. Tags: <code>atp.Status=candidate</code>

Table A.8: ComFieldGrant

Class	<i>ComGrant</i> (abstract)
Note	This meta-class serves as the abstract base class for defining specific <code>ComGrants</code> Tags: <code>atp.Status=candidate</code> This Class is only used by the AUTOSAR Adaptive Platform.
Base	<i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>Grant</i> , Identifiable , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , Referrable , <i>UploadableDeploymentElement</i> , <i>UploadablePackageElement</i>
Subclasses	<code>ComEventGrant</code> , ComFieldGrant , ComMethodGrant , <code>ComTriggerGrant</code>





Class	ComGrant (abstract)			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
remoteSubject	AbstractIamRemoteSubject	*	ref	This optional reference defines the remoteSubject that is allowed to access the defined Object via the Grant. Tags: atp.Status=candidate
serviceInstance	AdaptivePlatformServiceInstance	0..1	ref	This reference identifies the applicable AdaptivePlatformServiceInstance for which the grant applies. Tags: atp.Status=candidate

Table A.9: ComGrant

Class	ComMethodGrant			
Note	This meta-class represents the ability to grant access to a ServiceInterface.method. Tags: atp.Status=candidate atp.recommendedPackage=Grants This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, CollectableElement, ComGrant , Grant, Identifiable , MultilanguageReferrable, PackageableElement, Referrable , UploadableDeploymentElement, UploadablePackageElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
design	ComMethodGrantDesign	0..1	ref	This reference identifies the ComMethodGrantDesign that the enclosing ComMethodGrant was created from. Stereotypes: atp.UriDef Tags: atp.Status=candidate
serviceDeployment	ServiceMethodDeployment	0..1	ref	This reference identifies the applicable deployment within the context of an AdaptivePlatformServiceInstance for which the grant applies. Tags: atp.Status=candidate

Table A.10: ComMethodGrant

Class	CplusplusImplementationDataType (abstract)			
Note	This meta-class represents the way to specify a reusable data type definition taken as a the basis for a C++ language binding This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, AbstractImplementationDataType, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType , CollectableElement, CplusplusImplementationDataTypeContextTarget, Identifiable , MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	CustomCplusplusImplementationDataType, StdCplusplusImplementationDataType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
arraySize	PositiveInteger	0..1	attr	This attribute can be used to specify the array size if the enclosing CplusplusImplementationDataType has array semantics. Stereotypes: atp.Variation Tags: vh.latestBindingTime=preCompileTime
headerFile	String	0..1	attr	Configuration of the Header File with the custom class declaration.
namespace (ordered)	SymbolProps	*	aggr	This aggregation allows for the definition an own namespace for the enclosing CplusplusImplementationDataType.





Class	<i>CppImplementationDataType</i> (abstract)			
subElement (ordered)	CppImplementationDataTypeElement	*	aggr	This represents the collection of sub-elements of the enclosing CppImplementationDataType
template Argument (ordered)	CppTemplateArgument	*	aggr	This aggregation allows for the specification of properties of template arguments
typeEmitter	NameToken	0..1	attr	This attribute can be taken to control how the respective CppImplementationDataType is contributed to the language binding.
typeReference	CppImplementationDataType	0..1	ref	This reference shall be defined to define a type reference (a.k.a. typedef).

Table A.11: CppImplementationDataType

Class	Executable			
Note	This meta-class represents an executable program. Tags: atp.recommendedPackage=Executables This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, AtpClassifier, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, UploadableDesignElement, UploadablePackageElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
implementation Props	ExecutableImplementationProps	*	aggr	This aggregation contains the collection of implementation-specific properties necessary to properly build the enclosing Executable.
minimumTimer Granularity	TimeValue	0..1	attr	This attribute describes the minimum timer resolution (TimeValue of one tick) that is required by the Executable.
reporting Behavior	ExecutionStateReportingBehaviorEnum	0..1	attr	this attribute controls the execution state reporting behavior of the enclosing Executable.
rootSw Component Prototype	RootSwComponentPrototype	0..1	aggr	This represents the root SwCompositionPrototype of the Executable. This aggregation is required (in contrast to a direct reference of a SwComponentType) in order to support the definition of instanceRefs in Executable context.
suspendToRam Awareness	SuspendToRamAwarenessEnum	0..1	attr	This attribute describes the type of awareness of the enclosing Executable to suspend-to-RAM functionality. Tags: atp.Status=candidate
version	StrongRevisionLabelString	0..1	attr	Version of the executable.

Table A.12: Executable

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.





Class	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.13: Field

Enumeration	FieldAccessEnum
Note	This meta-class provides values that qualify access to a field. This Enumeration is only used by the AUTOSAR Adaptive Platform.
Aggregated by	ComFieldGrant.role , ComFieldGrantDesign.role
Literal	Description
getter	Access to the getter of the Field. Tags: atp.EnumerationLiteralIndex=0
getterSetter	Access to getter and setter of the field Tags: atp.EnumerationLiteralIndex=2
setter	Access to the setter of the Field. Tags: atp.EnumerationLiteralIndex=1

Table A.14: FieldAccessEnum

Class	FieldSenderComSpec			
Note	Port specific communication attributes for a Field that is defined in a ServiceInterface. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject , PPortComSpec , SenderComSpec			
Aggregated by	AbstractProvidedPortPrototype.providedComSpec , PortPrototypeBlueprint.providedComSpec			
Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	Initial value for a Field that is set before the Service Interface is offered.

Table A.15: FieldSenderComSpec

Class	GenericModuleInstantiation			
Note	This meta-class defines the attributes for the generic module configuration on a specific machine. Different modules are distinguishable by the category attribute. This element can also be used to describe modules that are not standardized by AUTOSAR. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject , AdaptiveModuleInstantiation , AtpClassifier , AtpFeature , AtpStructureElement , Identifiable , MultilanguageReferrable , NonOsModuleInstantiation , Referrable			
Aggregated by	AtpClassifier.atpFeature , Machine.moduleInstantiation			
Attribute	Type	Mult.	Kind	Note
—	—	—	—	—

Table A.16: GenericModuleInstantiation

Class	Identifiable (abstract)
Note	Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables.





Class	Identifiable (abstract)			
Base	ARObject, MultilanguageReferrable, Referrable			
Subclasses	ARPackage, AbstractDolpLogicAddressProps, AbstractEvent, AbstractFunctionalClusterDesign, AbstractImplementationDataTypeElement, AbstractSecurityEventFilter, AbstractSecurityIdsmInstanceFilter, AbstractServiceInstance, AbstractSignalBasedToSignalTriggeringMapping, AdaptiveSwcInternalBehavior, ApApplicationEndpoint, ApmcAbstractDefinition, ApmcConfigurationElementDef, ApmcContainerElementValue, ApmcContainerValue, ApmcEnumerationLiteralDef, ApplicationEndpoint, ApplicationError, AppliedStandard, ArtifactChecksum, ArtifactLocator, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpFeature, AutosarOperationArgumentInstance, AutosarVariableInstance, BuildActionEntity, BuildActionEnvironment, Chapter, CheckpointTransition, ClientIdDefinition, ClientServerOperation , Code, CollectableElement, ComManagementMapping, CommConnectorPort, CommunicationConnector, CommunicationController, Compiler, ConsistencyNeeds, ConsumedEventGroup, CouplingPort, CouplingPortAbstractShaper, CouplingPortStructuralElement, CryptoCertificate, CryptoCertificateGroup, CryptoKeySlot, CryptoKeySlotDesign, CryptoKeySlotUsageDesign, CryptoProvider, CryptoServiceMapping, DataPrototypeGroup, DataPrototypeTransformationPropsIdent, DataTransformation, DdsAbstractServiceInstanceElementCp, DdsCpDomain, DdsCpPartition, DdsCpQosProfile, DdsCpTopic, DdsDomainRange, DependencyOnArtifact, DiagEventDebounceAlgorithm, DiagnosticAbstractSovdContent, DiagnosticAuthTransmitCertificateEvaluation, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticDebounceAlgorithmProps, DiagnosticExtendedDataRecordElement, DiagnosticFunctionInhibitSource, DiagnosticParameterElement, DiagnosticRoutineSubfunction , DiagnosticSovdMethodPrimitive, DltApplication, DltArgument, DltArgumentProps, DltMessage, DolpInterface, DolpLogicAddress, DolpLogicalAddress, DolpNetworkConfigurationDesign, DolpRoutingActivation, E2EProfileConfiguration, End2EndEventProtectionProps, End2EndMethodProtectionProps, EthernetWakeupSleepOnDataLineConfig, EventHandler, EventMapping, ExclusiveArea, ExecutableEntity, ExecutionTime, FMAttributeDef, FMFeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeatureRelation, FMFeatureRestriction, FMFeatureSelection, FieldMapping, FireAndForgetMethodMapping, FlexrayArTpNode, FlexrayTpPduPool, FrameTriggering , GeneralParameter, GlobalSupervision, GlobalTimeGateway, GlobalTimeMaster , GlobalTimeSlave , HealthChannel , HeapUsage , HwAttributeDef, HwAttributeLiteralDef, HwPin, HwPinGroup, IEEE1722TpAcfBus , IEEE1722TpAcfBusPart , IPSecRule, IPv6ExtHeaderFilterList, ISignalToIPduMapping, ISignalTriggering, IdentCaption , ImpositionTime, InternalTriggeringPoint, Keyword, LifecycleState, Linker, MacAddressVlanMembership, MacMulticastGroup, MacSecKayParticipant, McDataInstance, MemorySection, MemoryUsage, MethodMapping, ModeDeclaration, ModeDeclarationMapping, ModeSwitchPoint, ModeSwitchSenderComSpecProps, NetworkEndpoint, NmCluster , NmNode , PackageableElement , ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, PerInstanceMemory, PersistencyDeploymentElement , PersistencyInterfaceElement , PhmSupervision , PhysicalChannel , PortGroup, PortInterfaceMapping , ProcessToMachineMapping , Processor, ProcessorCore, PskIdentityToKeySlotMapping, QueuedReceiverComSpecProps, ResourceConsumption, ResourceGroup, RootSwClusterDesignComponentPrototype, RootSwComponentPrototype, RootSwCompositionPrototype, RptComponent, RptContainer, RptExecutableEntity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptServicePoint, RunnableEntityGroup, SdgAttribute , SdgClass, SecOcJobMapping, SecOcJobRequirement, SecureCommunicationAuthenticationProps, SecureCommunicationDeployment , SecureCommunicationFreshnessProps, SecurityEventContextDataElement, SecurityEventContextProps, ServerComSpecProps, ServiceInterfaceDeploymentElement , ServiceInterfaceElementSecureComConfig, ServiceNeeds , SignalServiceTranslationEventProps, SignalServiceTranslationProps, SocketAddress, SoftwarePackageStep, SomeipEventGroup, SomeipProvidedEventGroup, SomeipTpChannel, StackUsage , StateManagementActionItem , StateManagementActionList, StateManagementStateNotification, StateManagementStateRequest , StaticSocketConnection, StructuredReq, SupervisionCheckpoint, SupervisionMode, SupervisionModeCondition, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SystemMapping, TimeBaseResource , TimingClock , TimingClockSyncAccuracy, TimingCondition, TimingConstraint , TimingDescription , TimingExtensionResource, TimingModelInstance, TlsCryptoCipherSuite, TlsCryptoCipherSuiteProps, TlsJobMapping, Topic1, TpAddress, TraceableTable, TraceableText, TracedFailure , TransformationISignalPropsIdent, TransformationProps , TransformationTechnology, Trigger, UcmDescription, UcmRetryStrategy, UcmStep, UriDescription, VariableAccess, VariationPointProxy, VehicleRolloutStep, ViewMap, VlanConfig, WaitPoint			
Attribute	Type	Mult.	Kind	Note
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. Stereotypes: atpSplitable Tags: atp.Splitkey=adminData xml.sequenceOffset=-40





Class	Identifiable (abstract)			
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes. Tags: xml.sequenceOffset=-25
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints. Tags: xml.sequenceOffset=-50
desc	MultiLanguageOverviewParagraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question. More elaborate documentation, (in particular how the object is built or used) should go to "introduction". Tags: xml.sequenceOffset=-60
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. Tags: xml.sequenceOffset=-30
uuid	String	0..1	attr	The purpose of this attribute is to provide a globally unique identifier for an instance of a meta-class. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is "DCE:2fac1234-31f8-11b4-a222-08002b34c003". The uuid attribute has no semantic meaning for an AUTOSAR model and there is no requirement for AUTOSAR tools to manage the timestamp. Tags: xml.attribute=true

Table A.17: Identifiable

Class	Machine			
Note	Machine that represents an Adaptive Autosar Software Stack. Tags: atp.recommendedPackage=Machines This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, AtpClassifier, AtpFeature, AtpStructureElement, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, UploadableDeploymentElement, UploadablePackageElement			
Aggregated by	ARPackage.element, AtpClassifier.atpFeature			
Attribute	Type	Mult.	Kind	Note
default Application Timeout	EnterExitTimeout	0..1	aggr	This aggregation defines a default timeout in the context of a given Machine with respect to the launching and termination of applications.





Class	Machine			
environment Variable	TagWithOptionalValue	*	aggr	This aggregation represents the collection of environment variables that shall be added to the environment defined on the level of the enclosing Machine. Stereotypes: atpSplitable Tags: atp.Splitkey=environmentVariable
machineDesign	MachineDesign	0..1	ref	Reference to the MachineDesign this Machine is implementing.
module Instantiation	AdaptiveModule Instantiation	*	aggr	Configuration of Adaptive Autosar module instances that are running on the machine. Stereotypes: atpSplitable Tags: atp.Splitkey=moduleInstantiation.shortName
processor	Processor	*	aggr	This represents the collection of processors owned by the enclosing machine.
secure Communication Deployment	SecureCommunication Deployment	*	aggr	Target-configuration of secure communication protocol configuration settings to crypto module entities. Stereotypes: atpSplitable Tags: atp.Splitkey=secureCommunication Deployment.shortName
trustedPlatform Executable LaunchBehavior	TrustedPlatform ExecutableLaunch BehaviorEnum	0..1	attr	This attribute controls the behavior of how authentication affects the ability to launch for each Executable.

Table A.18: Machine

Class	NonOsModuleInstantiation (abstract)			
Note	This meta-class defines the abstract attributes for the configuration of an adaptive autosar module other than the OS module. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject, AdaptiveModuleInstantiation, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , Referrable			
Subclasses	AdaptiveFirewallModuleInstantiation, CmModuleInstantiation, CryptoModuleInstantiation, Dolp Instantiation, GenericDiagnosticTransportInstantiation, GenericModuleInstantiation , IdsPlatform Instantiation , LogAndTraceInstantiation, NmInstantiation, SovdModuleInstantiation, StateManagement ModuleInstantiation, SuspendToRamModuleInstantiation, TimeSyncModuleInstantiation, UcmModule Instantiation			
Aggregated by	AtpClassifier.atpFeature, Machine.moduleInstantiation			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.19: NonOsModuleInstantiation

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.20: PPortPrototype

Enumeration	PersistencyAccessEnum
Note	This enumeration provides standardized ways of how persistent data can be accessed.
Aggregated by	AbstractPersistencyRequireComSpec.accessMode
Literal	Description
read	Access to persistent data is read-only. Tags: atp.EnumerationLiteralIndex=0 This EnumerationLiteral is only used by the AUTOSAR Adaptive Platform.
readWrite	Persistent data can both be written and read. Tags: atp.EnumerationLiteralIndex=2 This EnumerationLiteral is only used by the AUTOSAR Adaptive Platform.
write	Access to persistent data is write-only. Tags: atp.EnumerationLiteralIndex=1 This EnumerationLiteral is only used by the AUTOSAR Adaptive Platform.
writeOnlyOnce	The elements of this storage are prevented from being changed after first initialization. Tags: atp.EnumerationLiteralIndex=3 This EnumerationLiteral is only used by the AUTOSAR Adaptive Platform.

Table A.21: PersistencyAccessEnum

Class	PersistencyDataElement			
Note	This meta-class represents the ability to formally specify a piece of data that is subject to persistency in the context of the enclosing PersistencyKeyValueStorageInterface. PersistencyDataElement represents also a key-value pair of the deployed PersistencyKeyValueStorage and provides an initial value. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject, AtpFeature, AtpPrototype, AutosarDataPrototype, DataPrototype, Identifiable, Multilanguage Referrable, PersistencyInterfaceElement, Referrable			
Aggregated by	AtpClassifier.atpFeature, PersistencyKeyValueStorageInterface.dataElement			
Attribute	Type	Mult.	Kind	Note
—	—	—	—	—

Table A.22: PersistencyDataElement

Class	PersistencyDataRequiredComSpec			
Note	This meta-class represents the ability to define port-specific attributes for supporting use cases of data persistency on the required side. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject, AbstractPersistencyRequireComSpec , RPortComSpec			
Aggregated by	AbstractRequiredPortPrototype.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	Type	Mult.	Kind	Note
dataElement	PersistencyDataElement	0..1	ref	This reference represents the PersistencyDataElement for which the PersistencyDataRequiredComSpec applies. Stereotypes: atpIdentityContributor
initValue	ValueSpecification	0..1	aggr	This aggregation represents the definition of an initial value for the PersistencyDataElement referenced by the enclosing PersistencyDataRequiredComSpec

Table A.23: PersistencyDataRequiredComSpec

Class	PersistencyKeyValuePair
Note	This meta-class represents the ability to formally model a key-value pair in the context of the target-configuration of persistency. This Class is only used by the AUTOSAR Adaptive Platform.





Class	PersistencyKeyValuePair			
Base	ARObject, Identifiable , MultilanguageReferrable , PersistencyDeploymentElement , Referrable			
Aggregated by	PersistencyKeyValueStorage.keyValuePair			
Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	This aggregation represents the ability to define an initial value for the value side of the key-value pair. Please note that it does not make sense to configure an initial value if the PersistencyDeploymentElement.updateStrategy is set to the value delete.
valueDataType	AbstractImplementation DataType	0..1	ref	This reference represents the data type applicable for the value of the key-value pair.

Table A.24: PersistencyKeyValuePair

Class	PersistencyKeyValueStorage			
Note	This meta-class represents the ability to model a key-value storage on target-configuration level. Tags: atp.recommendedPackage=PersistencyKeyValueStorages This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , PersistencyDeployment , Referrable , UploadableDeploymentElement , UploadableExclusivePackageElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
keyValuePair	PersistencyKeyValuePair	*	aggr	This aggregation represents the key-value-pairs owned by the enclosing PersistencyKeyValueStorage .

Table A.25: PersistencyKeyValueStorage

Class	PersistencyKeyValueStorageInterface			
Note	This meta-class provides the ability to implement a PortInterface for supporting persistency use cases for data. Tags: atp.recommendedPackage=PersistencyKeyValueStorageInterfaces This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpType , CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , PersistencyInterface , PortInterface , Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
dataElement	PersistencyDataElement	*	aggr	This aggregation represents the collection of PersistencyDataElement s in the context of the enclosing PersistencyKeyValueStorageInterface .
dataTypeFor Serialization	AbstractImplementation DataType	*	ref	This reference identifies the AbstractImplementationDataType s that shall be supported for storing in a key-value storage in addition to the types already determined from the aggregation of PersistencyDataElement .
dataType Mapping	PersistencyKeyValue DataTypeMapping	0..1	aggr	This aggregation provides a collection of replacement rules for data types used in the context of the enclosing PersistencyKeyValueStorageInterface .

Table A.26: PersistencyKeyValueStorageInterface

Class	PersistencyPortPrototypeToKeyValueStorageMapping			
Note	This meta-class represents the ability to define a mapping between a PortPrototype and a key-value storage. Tags: atp.recommendedPackage=PersistencyPortPrototypeToKeyValueStorageMappings This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, CollectableElement, Identifiable , MultilanguageReferrable, PackageableElement, PersistencyPortPrototypeToDeploymentMapping, Referrable , UploadableDeploymentElement, UploadableExclusivePackageElement, UploadablePackageElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
keyValueStorage	PersistencyKeyValueStorage	0..1	ref	This reference represents the mapped key-value storage.

Table A.27: PersistencyPortPrototypeToKeyValueStorageMapping

Class	Process			
Note	This meta-class provides information required to execute the referenced Executable . Tags: atp.recommendedPackage=Processes This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, AbstractExecutionContext, AtpClassifier, CollectableElement, Identifiable , MultilanguageReferrable, PackageableElement, Referrable , UploadableDeploymentElement, UploadablePackageElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
design	ProcessDesign	0..1	ref	This reference represents the identification of the design-time representation for the Process that owns the reference.
executable	Executable	*	ref	Reference to executable that is executed in the process. Stereotypes: atp.UriDef
functionClusterAffiliation	String	0..1	attr	This attribute specifies which functional cluster the Process is affiliated with.
numberOfRestartAttempts	PositiveInteger	0..1	attr	This attribute defines how often a process shall be restarted if the start fails. numberOfRestartAttempts = "0" OR Attribute not existing, start once numberOfRestartAttempts = "1", start a second time
preMapping	Boolean	0..1	attr	This attribute describes whether the executable is preloaded into the memory.
processStateMachine	ModeDeclarationGroupPrototype	0..1	aggr	Set of Process States that are defined for the process. This attribute is used to support the modeling of execution dependencies that utilize the condition of process state. Please note that the process states may not be modeled arbitrarily at any stage of the AUTOSAR workflow because the supported states are standardized in the context of the SWS Execution Management [11].
stateDependentStartupConfig	StateDependentStartupConfig	*	aggr	Applicable startup configurations.

Table A.28: Process

Class	ProcessToMachineMapping			
Note	This meta-class has the ability to associate a Process with a Machine. This relation involves the definition of further properties, e.g. timeouts. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject, Identifiable , MultilanguageReferrable, Referrable			
Aggregated by	ProcessToMachineMappingSet.processToMachineMapping			





Class	ProcessToMachineMapping			
Attribute	Type	Mult.	Kind	Note
design	ProcessDesignToMachineDesignMapping	0..1	ref	This reference represents the identification of the design-time representation for the ProcessToMachineMapping that owns the reference.
machine	Machine	0..1	ref	This reference identifies the Machine in the context of the ProcessToMachineMapping.
nonOsModuleInstantiation	NonOsModuleInstantiation	0..1	ref	This supports the optional case that the process represents a platform module.
persistencyCentralStorageURI	UriString	0..1	attr	This attribute identifies a central place for the mapped Process to store the list of available storages and version information.
process	Process	0..1	ref	This reference identifies the Process in the context of the ProcessToMachineMapping.
shallNotRunOn	ProcessorCore	*	ref	This reference indicates a collection of cores onto which the mapped process shall not be executing.
shallRunOn	ProcessorCore	*	ref	This reference indicates a collection of cores onto which the mapped process shall be executing.

Table A.29: ProcessToMachineMapping

Class	RPortPrototype			
Note	Component port requiring a certain port interface.			
Base	<i>ARObject</i> , <i>AbstractRequiredPortPrototype</i> , <i>AtpBlueprintable</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , Identifiable , <i>MultilanguageReferrable</i> , <i>PortPrototype</i> , Referrable			
Aggregated by	<i>AtpClassifier.atpFeature</i> , <i>SwComponentType.port</i>			
Attribute	Type	Mult.	Kind	Note
requiredInterface	PortInterface	0..1	ref	The interface that this port requires. Stereotypes: isOfType

Table A.30: RPortPrototype

Class	<i>Referrable</i> (abstract)			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
Base	<i>ARObject</i>			
Subclasses	<i>AtpDefinition</i> , <i>BswDistinguishedPartition</i> , <i>BswModuleCallPoint</i> , <i>BswModuleClientServerEntry</i> , <i>BswVariableAccess</i> , <i>CouplingPortTrafficClassAssignment</i> , <i>CppImplementationData TypeContextTarget</i> , <i>DiagnosticEnvModeElement</i> , <i>EthernetPriorityRegeneration</i> , <i>ExclusiveAreaNestingOrder</i> , <i>HwDescriptionEntity</i> , <i>ImplementationProps</i> , <i>ModeTransition</i> , <i>MultilanguageReferrable</i> , <i>NmNetworkHandle</i> , <i>PncMappingIdent</i> , <i>SingleLanguageReferrable</i> , <i>SoConIPdulIdentifier</i> , <i>SomeipRequiredEventGroup</i> , <i>TpConnectionIdent</i>			
Attribute	Type	Mult.	Kind	Note
shortName	Identifier	1	attr	This specifies an identifying shortName for the object. It needs to be unique within its context and is intended for humans but even more for technical reference. Stereotypes: atpIdentityContributor Tags: xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortNameFragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. Tags: xml.sequenceOffset=-90

Table A.31: Referrable

Class	ServiceFieldDeployment (abstract)			
Note	This abstract meta-class represents the ability to specify a deployment of a Field to a middleware transport layer. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject, Identifiable , MultilanguageReferrable, Referrable , ServiceInterfaceDeploymentElement			
Subclasses	DdsFieldDeployment, SomeipFieldDeployment, UserDefinedFieldDeployment			
Aggregated by	ServiceInterfaceDeployment.fieldDeployment			
Attribute	Type	Mult.	Kind	Note
field	Field	0..1	ref	Reference to a Field that is deployed to a middleware transport layer. Stereotypes: atpUriDef

Table A.32: ServiceFieldDeployment

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
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





Class	ServiceInterface			
trigger	Trigger	*	aggr	This represents the collection of triggers defined in the context of a ServiceInterface. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=trigger.shortName, trigger.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=60

Table A.33: ServiceInterface

Class	ServiceMethodDeployment (abstract)			
Note	This abstract meta-class represents the ability to specify a deployment of a Method to a middleware transport layer. This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARObject, Identifiable , MultilanguageReferrable , Referrable , ServiceInterfaceDeploymentElement			
Subclasses	SomeipMethodDeployment, UserDefinedMethodDeployment			
Aggregated by	ServiceInterfaceDeployment .methodDeployment			
Attribute	Type	Mult.	Kind	Note
method	ClientServerOperation	0..1	ref	Reference to a method that is deployed to a middleware transport layer. Stereotypes: atpUriDef

Table A.34: ServiceMethodDeployment

Class	SoftwarePackage			
Note	This meta-class represents the ability to formalize the content of a software package. Tags: atp.recommendedPackage=SoftwarePackages This Class is only used by the AUTOSAR Adaptive Platform.			
Base	ARElement, ARObject, CollectableElement , Identifiable , MultilanguageReferrable , PackageableElement , Referrable , UploadableDeploymentElement , UploadablePackageElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
actionType	SoftwarePackageActionTypeEnum	0..1	attr	This attribute defines the action to be taken in the step of processing the enclosing SoftwarePackage.
activationAction	SoftwarePackageActivationActionEnum	0..1	attr	This attribute governs the action to be taken after the installation of the SoftwareCluster completed.
artifactLocator	ArtifactLocator	0..1	aggr	This attribute identifies the software package at configuration time, out of the context of an AUTOSAR model.
compressed Software PackageSize	PositiveInteger	0..1	attr	This size represents the size of the compressed Software Package.
deltaPackage Applicable Version	StrongRevisionLabelString	0..1	attr	This attribute identifies the version of the included SoftwareCluster for which the enclosing SoftwarePackage can be used as a delta update
estimated DurationOf Operation	TimeValue	0..1	attr	This attribute provides an estimation about how long the operation of the SoftwarePackage is going to take for its transfer, processing and activation when updated standalone (not within an update campaign)
minimum SupportedUcm Version	RevisionLabelString	0..1	attr	This attribute identifies the minimum supported version of the UCM for this SoftwarePackage.





Class	SoftwarePackage			
packagerId	PositiveInteger	0..1	attr	This attribute identifies Id of the organization that provides the packager generating the SoftwarePackage.
packagerSignature	CryptoServiceCertificate	0..1	ref	This reference identifies the certificate that represents the packager's signature.
primaryDownloadUri	UriDescription	0..1	aggr	This attribute represents the description of the primary download location. Stereotypes: atpSplitable Tags: atp.Splitkey=primaryDownloadUri.shortName
purposeOfUpdate	Documentation	0..1	ref	The referenced Documentation is supposed to provide a description of the purpose of the update.
secondaryDownloadUri	UriDescription	*	aggr	This attribute represents the description of the secondary download locations. Stereotypes: atpSplitable Tags: atp.Splitkey=secondaryDownloadUri.shortName
softwareCluster	SoftwareCluster	0..1	ref	This reference identifies the SoftwareCluster that belongs to the SoftwarePackage. The nature of this relation is actually more like an aggregation than a reference. But the relation is still modelled as a reference because two ARElements cannot aggregate each other.
uncompressedSoftwareClusterSize	PositiveInteger	0..1	attr	This attribute gives an indication about the storage that has to be available on the target.

Table A.35: SoftwarePackage

Class	ValueSpecification (abstract)			
Note	Base class for expressions leading to a value which can be used to initialize a data object.			
Base	ARObject			
Subclasses	AbstractRuleBasedValueSpecification, ApplicationValueSpecification, CompositeValueSpecification, ConstantReference, NotAvailableValueSpecification, NumericalValueSpecification, ReferenceValueSpecification, TextValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, DiagnosticEnvSovdDataCondition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.receptionDefaultValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireComSpec.initValue, PersistenceDataRequiredComSpec.initValue, PersistenceKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, SomeipEventDeployment.eventReceptionDefaultValue, StateManagementCompareCondition.compareValue, SwDataDefProps.invalidValue, UserDefinedEventDeployment.eventReceptionDefaultValue, VariableDataPrototype.initValue			
Attribute	Type	Mult.	Kind	Note
shortLabel	Identifier	0..1	attr	This can be used to identify particular value specifications for human readers, for example elements of a record type.

Table A.36: ValueSpecification

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 chapter lists all functional requirements specified in the corresponding requirement specifications that are not implemented or violated by this specification and provides a rationale.

[AP_SWS_RPER_NA_00001] Requirements Not Applicable to this Specification

Upstream requirements: RS_AP_00170, RS_AP_00171, RS_AP_00172, RS_AP_00173, RS - AP_00177

[These requirements are not applicable to this specification.]

[AP_SWS_RPER_NA_00002] Access Control Requirement Not Applicable to this Specification

Upstream requirements: RS_PER_00024

[This requirement needs to be fulfilled by the configuration of [IAM](#).]

For details on the configuration of [IAM](#) see [Chapter 7.1.1.3](#).

E Change History of AUTOSAR Traceable Items

This chapter provides an overview of the 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 Traceable Item History of this Document According to AUTOSAR Release R25-11

E.1.1 Added Specification Items in R25-11

[\[AP_SWS_RPER_00001\]](#) [\[AP_SWS_RPER_00002\]](#) [\[AP_SWS_RPER_00003\]](#) [\[AP_SWS_RPER_00004\]](#) [\[AP_SWS_RPER_00005\]](#) [\[AP_SWS_RPER_00006\]](#) [\[AP_SWS_RPER_00007\]](#) [\[AP_SWS_RPER_00008\]](#) [\[AP_SWS_RPER_00009\]](#) [\[AP_SWS_RPER_00010\]](#) [\[AP_SWS_RPER_00011\]](#) [\[AP_SWS_RPER_00012\]](#) [\[AP_SWS_RPER_90001\]](#) [\[AP_SWS_RPER_90002\]](#) [\[AP_SWS_RPER_90003\]](#) [\[AP_SWS_RPER_90004\]](#) [\[AP_SWS_RPER_90005\]](#) [\[AP_SWS_RPER_90006\]](#) [\[AP_SWS_RPER_90008\]](#) [\[AP_SWS_RPER_90009\]](#) [\[AP_SWS_RPER_90010\]](#) [\[AP_SWS_RPER_90011\]](#) [\[AP_SWS_RPER_90012\]](#) [\[AP_SWS_RPER_90013\]](#) [\[AP_SWS_RPER_90014\]](#)

E.1.2 Changed Specification Items in R25-11

none

E.1.3 Deleted Specification Items in R25-11

none

E.1.4 Added Constraints in R25-11

[\[AP_SWS_RPER_CONSTR_00001\]](#) [\[AP_SWS_RPER_CONSTR_00002\]](#) [\[AP_SWS_RPER_CONSTR_00003\]](#) [\[AP_SWS_RPER_CONSTR_00004\]](#)

E.1.5 Changed Constraints in R25-11

none

E.1.6 Deleted Constraints in R25-11

none