Document Title	Specification of Identity and Access Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	900

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R20-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2020-11-30	R20-11	AUTOSAR Release Management	• Introduced policy enforcement in Adaptive Applications • Added IAM for Platform Health Management
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • Changed Document Status from Final to published
2019-03-29	19-03	AUTOSAR Release Management	• Reworked chapter 7 to incorporate new concept of `Grants` • Incorporation of several bug tickets
2018-10-31	18-10	AUTOSAR Release Management	• Reworked functional specification • Removed API specification for general rework
2018-03-29	18-03	AUTOSAR Release Management	• Initial release

Disclaimer

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction and functional overview

This specification describes the functionality, API and the configuration for the Identity and Access Management functional cluster of the AUTOSAR Adaptive Platform.
The Identity and Access Management offers applications a standardized interface to access management operations.

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Identity and Access Management module that are not included in the AUTOSAR glossary [1].

Abbreviation / Acronym:	Description:
PDP	Policy Decision Point
PEP	Policy Enforcement Point
IPC	Inter-Process Communication

3 Related documentation

3.1 Input documents & related standards and norms

[1] Glossary
AUTOSAR_TR_Glossary

[2] Requirements on Identity and Access Management
AUTOSAR_RS_IdentityAndAccessManagement

3.2 Further applicable specification

See chapter 3.1.

4 Constraints and assumptions

4.1 Known Limitations

- A detailed API will be added in a future release

- Currently limited to ara::com

- For other Functional Clusters, implementation on Policy Enforcement Points are envisaged for a future release.

4.2 Assumptions

The integrator can configure an authentic channel between `Policy Decision Points` and `Policy Enforcement Points`. This could be done through the operating system's access rights for example.

5 Dependencies to other modules

The implementation of a `Policy Enforcement Point` in a `Functional Cluster` is defined in the corresponding software specification.

The following implementations are defined:

- Communication Management (Section 7.5 Security)

- Platform Health Management

5.1 Protocol layer dependencies

Identity and Access Management (IAM) prevents unauthorized usage of APIs of Functional Clusters. IAM does not introduce additional dependencies from the perspective of Adaptive Applications. The implementation of IAM itself is not specified by AUTOSAR AP. Accordingly additional dependencies are not mandatory.

6 Requirements traceability

The following tables reference the requirements specified in [2] 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_IAM_00010]	Adaptive applications shall only be able to use AUTOSAR Resources when authorized	[SWS_IAM_01002]

7 Functional specification

The functional specifications provides a technical overview to elaborate the generic approach to modeling and implementing Identity and Access Management features in the Adaptive Platform.

The technical concept of IAM shall be discussed from a modeling and a runtime perspective. The modeling perspective shall elaborate the concept from the view of an application designer and an integrator. The runtime perspective shall outline the integration of the concept into the Adaptive Platform functional clusters.

7.1 Functional cluster life-cycle

IAM is specified by additional requirements for a specific `Functional Cluster`. Methods under access control shall not be executed without IAM being active. Still the life-cycle is specific to the affected `Functional Cluster`. The implementation of IAM itself is not specified by `AUTOSAR` and thus a common life-cycle is not specifiable.

7.1.1 Startup

Dependencies to interfaces and objects involved in the IAM implementation have to be fulfilled, i.e. processes and databases have to be available to PEP and PDP. Those services should be available early.

7.1.2 Shutdown

Dependencies to interfaces and objects involved in the IAM implementation have to be fulfilled, i.e. processes and databases have to be available to PEP and PDP. Those services should be shutdown last.

7.2 Terms

Before discussing concept of Identity and Access Management the employed terminology shall be elaborated.

- **Intent -** An Intent is a property of an Adaptive Application. Access to an AUTOSAR resource (e.g. `ServiceInterface` and its members `Method`, `Event` and `Field`) is granted only if a requesting Adaptive Application possesses all acknowledged intents that are mandatory for that specific resource. Intents are assigned to Adaptive Applications within their Application Manifest by

means of `ComSpecs` (e.g. `ClientComSpec`) and `GrantDesigns` (e.g. `ComFieldGrantDesign`).

- **Grant -** The integrator acknowledges an Adaptive Application's intent by transferring `GrantDesigns` to a `Grant` in deployment. `Grant` elements may be processed into access control lists for the PDP implementation.

- **Process -** A `Process` is the meta model's runtime instance of an Adaptive Application and represents its runtime identity. A `Process` may be identified during runtime by an uniquely assigned identifier (e.g. a Unix user).

- -

- **Policy Enforcement Point (PEP) -** The PEP is usually implemented within a functional cluster and will query a PDP for allowance to perform an operation and will block the operation if necessary.

- **Policy Decision Point (PDP) -** The PDP should be implemented in a single locations (e.g. an OEM-specific application or an application provided by the stack vendor) and serve the PEP with authorization decisions. Furthermore, the PDP manages the authorization database(s).

7.3 Modeling

The application designer shall model each interaction point of an application with the ARA API. Therefore, every functional cluster shall define `PortInterfaces` representing its ARA API features as well as a set of sensible `GrantDesigns`. The `GrantDesigns` shall be structured such that security critical portions of the ARA API can be restricted. For example the ara::com API uses the `ServiceInterface` to represent its interaction points. The `ServiceInterface` itself consists of `Method`, `Event` and `Field` entities. Each of these entities may be subject to fine-grained access control restrictions. Hence, there are `GrantDesigns` for each of these entities.

Using a functional cluster's available meta model design elements, the application designer shall create a model consisting of:

- `PortPrototypes` referencing a functional cluster's `PortInterfaces` to express the need for using an ARA API

- `GrantDesigns` to request access to specific elements within a functional cluster's `PortInterface` or to the `PortInterface` itself

The integrator shall accept each requested access by creating an explicit `Grant` entity in the deployment model. The `Grant` shall reference the application designer's `GrantDesign`, a `Process`, and the functional cluster's respective deployment model entities (e.g. `SomeipServiceInterfaceDeployment`). If an integrator does not accept a requested access, a valid model cannot be created and therefore the integrator and the application designer shall reconsider requesting the access or granting the access. This prevents using the IAM concept to achieve variant modeling.

Using the functional cluster's available meta model deployment elements, the integrator shall create a model consisting of:

- The application's runtime instances (`Process`)

- The functional cluster's deployment model

- The functional cluster's `Grants` linking the runtime instance to the protected asset

Given the deployment model the IAM-related entities can be transformed into a Processed Manifest for deployment to an ECU or for an integration into a software update package.

7.4 Runtime

There are several approaches to implementing the proposed Identity and Access Management concept. An exemplary approach will be outlined to provide a better understanding.

[SWS_IAM_01002]{DRAFT} **Identification of an application** ⌈An application's runtime instance shall be identifiable by the local PEP.⌋*(RS_IAM_00010)*

One mechanism for identifying a runtime instance of an application is to run the process as a distinct operating system user (e.g. Unix users). Given that example, Execution Management shall start application processes as distinct operating system users. Operating systems provide functionality to query peer credentials on an IPC channel (e.g. `getpeereid()` with Unix domain sockets). The Adaptive Platform implementation's IPC mechanism can encapsulate this functionality and transfer the information to the Policy Enforcement Point. The Policy Enforcement Point shall translate the actual runtime identity (i.e. the Unix user ID) to the model's representation (i.e. the Process). The Policy Decision Point shall use the reference to the model's runtime instance and the referencing Grants to allow or deny access.

8 API specification

No API specified for IAM.

9 Service Interfaces

No interfaces specified.

A   Mentioned Manifest Elements

B   Interfaces to other Functional Clusters

C   History of Constraints and Specification Items