Document Title	Requirements on Identity and Access Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	899

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R21-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2021-11-25	R21-11	AUTOSAR Release Management	• No content changes
2020-11-30	R20-11	AUTOSAR Release Management	• Capability renamed to Intent • Changes in architecture regarding process separation reflected in document
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	• Introduction of `Grant` concept
2018-10-31	18-10	AUTOSAR Release Management	• Functional Description of Capabilities • Functional Description of Access Control for Inter-Platform Communication • Requirement for Superset Manifests
2018-03-29	18-03	AUTOSAR Release Management	• Initial release

Disclaimer

Table of Contents

1 Scope of Document 4

2 Conventions to be used 4

2.1 Requirements Guidelines 4
2.1.1 Requirements quality 4
2.1.2 Requirements identification 4
2.1.3 Requirements status 4

3 Acronyms and abbreviations 4

4 Requirements Specification 5

4.1 Functional Overview 5
4.2 Functional Requirements 8
4.3 Non-Functional Requirements (Qualities) 14

5 Requirements Tracing 14

6 References 14

1 Scope of Document

This document specifies the requirements of Identity and Access Management to the AUTOSAR Adaptive Platform. The motivation is to provide standardized and portable security in Adaptive Applications.

2 Conventions to be used

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

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

2.1 Requirements Guidelines

2.1.1 Requirements quality

2.1.2 Requirements identification

2.1.3 Requirements status

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to Identity and Access Management that are not included in the AUTOSAR Glossary [2].

Term:	Description:
Identity and Access Management (IAM)	IAM is about managing access rights of an `Adaptive Application` to interfaces and resources of the Adaptive Platform Foundation and Services.
Policy Decision Point (PDP)	The PDP represents the logic in which the access control decision is made. It determines if the application is allowed to perform the requested task.
Policy Enforcement Point (PEP)	The PEP represents the logic in which the Access Control Decision is enforced. It communicates directly with the corresponding PDP to receive the Access Control Decision.
Access control Policy	Access Control Policies are bound to the targets of calls (i.e. Service interfaces) and are used to express what Identity Information are necessary to access those interfaces.
Access Control Decision	The Access Control Decision is a Boolean value indicating if the requested operation is permitted or not. It is based on the identity of the caller and the Access Control Policy.

Term:	Description:
Identity	Identity represents properties of an `Adaptive Application` the access control is decided / enforced upon.
AUTOSAR Resource	The term AUTOSAR Resource covers interfaces that are under the scope of IAM, i.e. Service Interfaces.
Application ID	Application ID is a unique identifier of an `Adaptive Application`. In the meta-model an `Adaptive Application` is represented by `Process`.
Intent	A Intent is a property of an `Adaptive Application`. Access to an AUTOSAR resource is granted if a requesting AA possesses all acknowledged intents that are necessary for that specific AUTOSAR Resource. Intents are assigned to `Adaptive Application`s 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.

Table 3.1: Acronyms and Abbreviations

4 Requirements Specification

4.1 Functional Overview

Identity and Access Management (IAM) provides services for `Adaptive Application`s and other clusters in the AUTOSAR Adaptive Platform. The goal of IAM is to prevent an erroneous or a compromised Adaptive Application to access a service or resource that was not intended to be accessed be the application's designer and the integrator.

We shortly discuss an example as motivation for IAM. We consider an infotainment application with internet access that has a high risk of being compromised. We assume that this application should never have access to a service allowing to brake the car, because the infotainment application is heavily exposed by its internet access. If somehow the infotainment gets compromised by an attacker, an AUTOSAR Adaptive Platform must prevent any access attempt of the infotainment application to the braking service.

For the AUTOSAR Adaptive Platform the concept is to derive access rights directly from the models. During the design phase of an `Adaptive Application` its intents are modelled and acknowledged by the integrator during deployment.

Another representation of access rights is an access matrix as shown in Fig. 4.1.

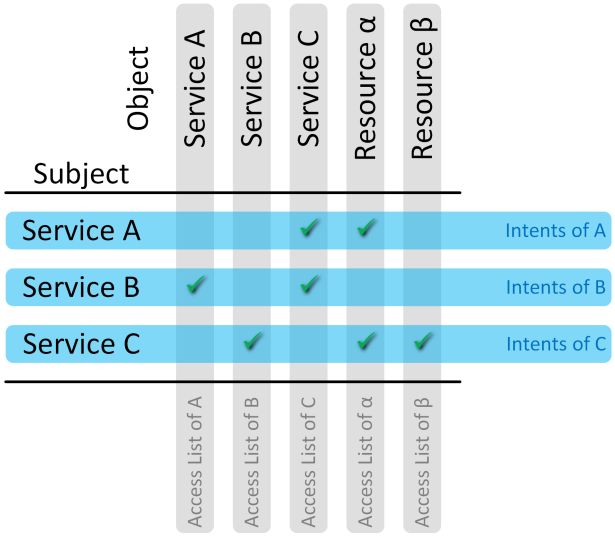

Figure 4.1: Access Matrix

The access matrix shows the access rights of subjects on objects. A *subject* is an artifact that wants to have access. Typically this is (part of) a process running on a system, but not a resource. An *object* is an artifact that access should be granted on. This can be either another (part of) a process or a resource.

The information about the access rights must deployed to the system using a *manifest*. There are two alternatives: For each service or application, provide an object list–its *Intents*–, i.e., the access rights that this service or application has as a subject. Or, for each service or resource, provide its *access list*, i.e., the list of all subjects having the right to access the service or resource as object.

For the AUTOSAR Adaptive Platform we deploy intents together with an Adaptive Application. For one subject this list of accessible objects typically does not change over time. For an object, however, an access list likely has to be updated with the deployment of a further application.

On a running platform instance access rights need to be enforced as shown by Fig. 4.2.

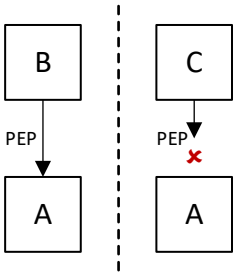

Figure 4.2: Access Control by Policy Enforcement

As previously declared in Fig. 4.1, service B is allowed to access service A – B has the intent to access A. However, service C does not have the intent to access A. A *Policy Enforcement Point* (PEP) must supervise the interaction and thus prevent the access of C on A. The information, whether an intent is present or not, is provided by a *Policy*

Decision Point (PDP). In order to provide this information, a PDP needs the *identity* of subject and object as well as further details on the kind of interaction between subject and object.

To provide a suitable level of security for this concept, there are some constraints that must be considered:

- The intents provided in a manifest must be authenticated. An attacker should not be able to change the intents of an application to gain more access.
- The Policy Enforcement must be implemented outside the application that is supervised. An application compromised by an attacker shall not be able to simply circumvent the PEP. The PEP may not be executed in the process-context of an application.

In case, the interaction between applications crosses platform boundaries, actually two PEPs have to be used as shown in Fig. 4.3.

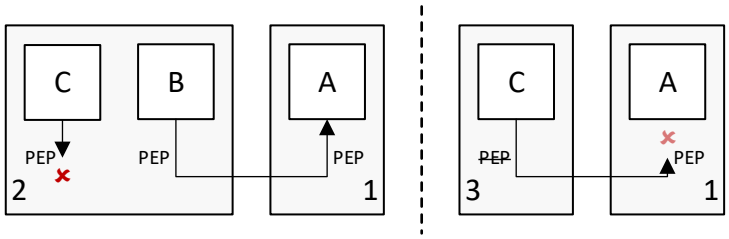

Figure 4.3: Double Access Control for Inter-Platform Communication

Ideally the first PEP on the side of subject already correctly enforces access rights like shown on the left side of Fig. 4.3. B is granted access, C is not. The check on the side of the object is redundant in this case, as only valid interaction is passed to the object side. The right side of Fig. 4.3 shows the case that the whole platform instance 3 of the subject has been compromised, i.e., the PEP is no longer effective. Then the object side 1 cannot rely on a correct enforcement by 3. Additionally 1 cannot rely on any identity information coming from 3.

If we assume that the channel between 3 and 1 is authentic, the second PEP on 1 at least can distinguish, whether any application on 3 has the intent to access A. If yes, access is granted; if not, access is denied for all requests from 3.

In order to decide, whether any application of a different platform instance has a certain intent, the platform instance need to exchange their information on intents. Each platform instance shall create a *superset manifest* containing all the manifests of each application currently deployed on the platform instance.

The synchronization of superset manifests between platform instances is out of scope of this standard.

4.2 Functional Requirements

[RS_IAM_00001]{DRAFT} **Limit `Adaptive Application` access to the `Adaptive Platform Foundation` and Services.** ⌈

Description:	An Adaptive Platform Instance shall provide means to actively restrict access of an `Adaptive Application` to those interfaces and resources of the Adaptive Platform Foundation and Services that the `Adaptive Application` was originally designed to use.
Rationale:	Privilege Escalation in case of an attack shall be prevented. Integrators shall be enabled to retrace and control intended tasks of Applications.
Dependencies:	RS_IAM_00010
Use Case:	Designer of App declares intended usage of App. Integrator reviews set of requested actions and accepts by assigning `GRANT` elements to requested intents. Attacker controls App during runtime. Attacker gains no more permissions than App's initial permissions.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00002]{DRAFT} **Position of Policy Enforcement** ⌈

Description:	Access control to interfaces of the Adaptive Platform Foundation and Services shall be enforced by PEP(s) located in (a) the object's process, (b) the operating system, and/or (c) a process of the Adaptive Platform Foundation. A PEP that runs in the context of the subject `Adaptive Application` must not be used for enforcement of access control on requests by the subject itself.
Rationale:	Adaptive Applications are considered to potentially being compromised thus their access shall be restricted by IAM. An `Adaptive Application` shall not be able to control policy decisions restricting their own requests. The PEP that restricts the requesting `Adaptive Application` shall be implemented and executed using a separate process not under control of the requesting `Adaptive Application`. IPC and/or network communication must separate the subject `Adaptive Application` from the PEP.
Dependencies:	–
Use Case:	Application requests a method on Service Interface. IAM (located in, e.g., process of Adaptive Platform Foundation or process of the service provider) identifies app and enforces access restrictions. Due to process separation, application cannot spoof its identity or manipulate policy enforcement.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00004]{DRAFT} **Circumvention of AUTOSAR Policy Enforcement by Applications shall be prevented.** ⌈

Description:	`Adaptive Platform` shall prevent Applications from circumventing AUTOSAR policy enforcement by using other APIs than the AUTOSAR defined APIs.
Rationale:	The runtime environment of the `Adaptive Platform Foundation` shall ensure that an Adaptive Application may not circumvent PEPs by selecting alternative interfaces not under access control.
Dependencies:	RS_SEC_5019, RS_SEC_5018
Use Case:	Intents for access on Service Interface SIf provided by Service Instance SInst are not assigned to Application A. Communication Management exposes API to `Adaptive Application`s and forwards requests to local instances via IPC. A tries to open communication channel to SInst directly (implementation specific). Access control of runtime environment prevents direct access.
Supporting Material:	–

⌋(*RS_Main_00514*, *RS_Main_00060*)

[RS_IAM_00005]{DRAFT} **`Adaptive Platform Foundation` shall enforce that only Applications that are configured accordingly are able to gain information about the permissions of other applications** ⌈

Description:	The `Adaptive Platform Foundation` must prevent applications from gaining information about the permissions of other applications unless explicitly configured to be allowed to access this information, i.e. for implementing a PDP in this specific Application.
Rationale:	Information about the overall-system that might help attackers to analyze the system shall not be exposed by IAM.
Dependencies:	RS_SEC_5018
Use Case:	Application A implements PDP and provides according interface to PEPs. During a request A gains access on processed manifests of `Adaptive Platform Foundation` in order to provide the access control decision. Malicious Application B requests access on processed manifests. Since the application was not registered as PDP access on manifests is denied.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00006]{DRAFT} **Access control policies shall be available to the PDP** ⌈

Description:	Access control policies shall be available to the PDP. Policies are either modelled in implementation-specific ways or even represented by code. Policies are not part of the AUTOSAR meta-model.
Rationale:	The PDP shall provide actual decisions for access control. Those decisions are based on Application's Intents and Policies, so both shall be available to PDP.
Dependencies:	–

▽

△

Use Case:	App requests access on resource. PEP identifies App and forwards request to PDP. PDP has to return binary decision, if identified App brings the required intents that fullfill the policy.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00007]{DRAFT} **The `Adaptive Platform Foundation` shall provide access control decisions** ⌈

Description:	The `Adaptive Platform Foundation` shall provide access control decisions based on intents that are stored in the corresponding manifests and policies specific to `Functional Cluster`.
Rationale:	Policies used by PDP implemented in `Adaptive Platform Foundation` are well-defined by AUTOSAR.
Dependencies:	–
Use Case:	Application A requests access on public interface of `Functional Cluster` (FC). The manifest of Application A defines its intents. PEP forwards description of request to PDP via inter-functional-cluster interface. Policies used by PDP are predefined by AUTOSAR. The representation of policies is implementation-specific and may even be hard-coded. PDP checks processed manifests for intents of Application A. PDP returns access control decision to PEP.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00008]{DRAFT} **Access shall be denied by the PEP if the corresponding PDP is not available** ⌈

Description:	Access shall be denied by the PEP if the corresponding PDP is not available. Applications that depend on access control during startup have to be covered by IAM. Therefore IAM should be available as soon as possible.
Rationale:	Attackers shall not gain access on resources by DoS-attacks on the PDP.
Dependencies:	–
Use Case:	Attacker requests access on resource. During the request the attacker exhausts RAM which leads to a time-out of the communication between PEP and PDP. The PEP blocks access on resource.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00009]{DRAFT} **An `Adaptive Application` may provide access control decisions** ⌈

Description:	The adaptive `Adaptive Platform Foundation` shall provide an interface to `Adaptive Application` to facilitate access control decisions based on access control policies and intents that are stored in the corresponding manifests. `Adaptive Application`s implementing a PDP are used for OEM-specific IAM. This interface is used at runtime during a operation restricted by access control. The specific PEP calls an OEM-specific PDP in order to block or allow a current operation usage.
Rationale:	Policies and Intents are well-defined by AUTOSAR. OEM-IAM enables the adaptive integration of OEM-specific access control.
Dependencies:	–
Use Case:	Access on Service Interface I depends on the vehicle state. This vehicle state is gathered by App A via Communication Management. App A provides Policy Decision based on vehicle state.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00010]{DRAFT} **Adaptive applications shall only be able to use AUTOSAR Resources when authorized** ⌈

Description:	The `Adaptive Platform Foundation` must ensure that adaptive applications shall only be able to use an AUTOSAR Resource if an existing policy authorizes them to do so.
Rationale:	Fine grained modelling of types of access on resources shall be enabled.
Dependencies:	–
Use Case:	App A uses a method derivateKey(sourceKey, targetkey). App A is defined as user of sourceKey and owner of targetKey. This prevents App A from writing to sourceKey.
Supporting Material:	–

⌋(*RS_Main_00060, RS_Main_00514*)

[RS_IAM_00011]{DRAFT} **Multiple PEPs** ⌈

Description:	IAM should support policy enforcement by multiple PEPs for one single request by an adaptive application
Rationale:	If multiple PEPs enforce a policy, all PEPs have to be compromised or circumvented for a successful attack.
Dependencies:	–

▽

△

Use Case:	If access control cannot be enforced at the object's ECUs (e.g., because it is a legacy ECU or because its PEP has been compromised), an uncompromised PEP on the subject's ECU can prevent unauthorized access. If access control cannot be enforced at the subject's ECUs (e.g., because it is a legacy ECU or because its PEP has been compromised), an uncompromised PEP at the object's ECU can prevent unauthorized access.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00014]{DRAFT} **Unique `Adaptive Application` ID** ⌈

Description:	An `Adaptive Application` ID shall be unique regarding the local machine.
Rationale:	Adaptive Applications shall be linked to and held responsible for their actions.
Dependencies:	–
Use Case:	The IAM framework uses the application ID of `Adaptive Application`s to verify requests and grant access to certain AUTOSAR Resources based on the defined polices.
Supporting Material:	–

⌋(*RS_Main_00510, RS_Main_00514*)

[RS_IAM_00017]{DRAFT} **Identity information shall be stored and handled tamper-proof throughout its lifecycle.** ⌈

Description:	The generation, transmission and storage of the Application Identity shall be handled tamper-proof throughout the life cycle.
Rationale:	Application identity integrity is a fundamental component for enforcing access controls.
Dependencies:	–
Use Case:	Application Designer defined Intents in manifest. The manifest is cryptographically signed. During deployment the manifest is authenticated and checked for integrity.
Supporting Material:	–

⌋(*RS_Main_00510, RS_Main_00514*)

[RS_IAM_00018]{DRAFT} **Set of intents shall be provided in the corresponding manifest** ⌈

Description:	The set of intents of an `Adaptive Application` shall be provided in the corresponding manifest.
Rationale:	Intents defined for an `Adaptive Application` shall be determined by the corresponding manifest. If an `Adaptive Application` is compromised, we need the manifest with the intents to actually enforce the restrictions implied by the intents. We cannot solely rely on the correct behavior of each `Adaptive Application`. `Adaptive Platform Foundation` shall not provide any interface that allows applications to change its intents defined in the manifest during runtime.
Dependencies:	–
Use Case:	The Application Designer defines the actions the Application will request. The Integrator checks plausibility. The Integrator does not need to define permissions.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00019]{DRAFT} **Intents of an `Adaptive Application` shall be authentically linked to the manifest** ⌈

Description:	The set of intents of an `Adaptive Application` shall be authentically linked to the `Adaptive Application` in the corresponding manifest.
Rationale:	An `Adaptive Application` is provided with a set of intents. It shall not be possible to extend or restrict this set except by signed updates. The `Adaptive Application` should always possess the same intents as defined by signed manifests.
Dependencies:	–
Use Case:	Application designer cryptographically signes the corresponding manifest. The manifest is deployed. A) Attacker provides malicious update for Application. Authenticity-check prevents deployment. B) Attacker gains control of App during runtime. Intents of an App are still determined and privilege escalation is prevented.
Supporting Material:	–

⌋(*RS_Main_00514, RS_Main_00510*)

[RS_IAM_00020]{DRAFT} **`Adaptive Platform Foundation` must allow to specify a superset manifest file of intents** ⌈

Description:	`Adaptive Platform Foundation` shall allow to specify a superset manifest file of intents.
Rationale:	An `Adaptive Platform Foundation` must provide a collection of all its current manifests in one single superset manifest for exchange with a second `Adaptive Platform Foundation`. The second `Adaptive Platform Foundation` may want to confirm an intent of the first `Adaptive Platform`.

▽

△

Dependencies:	–
Use Case:	A service A on an `Adaptive Platform` P_A may want to access service B on an `Adaptive Platform` P_B. Normally, the identity and access management on P_A will prevent access from A on B, if it does not have the corresponding intent. However, in case P_A is compromised, P_B cannot rely on correct decisions by P_A. Therefore identity and access management on P_B has to check, whether any service on P_A has the intent to access B. This information is provided by the superset manifest of P_A.
Supporting Material:	–

⌋(*RS_Main_00514*)

4.3 Non-Functional Requirements (Qualities)

5 Requirements Tracing

The following table references the features specified in [3] and links to the fulfillments of these.

Feature	Description	Satisfied by
[RS_Main_00060]	AUTOSAR shall provide a standardized software interface for communication between Applications	[RS_IAM_00004] [RS_IAM_00010]
[RS_Main_00510]	AUTOSAR shall support secure onboard communication	[RS_IAM_00014] [RS_IAM_00017] [RS_IAM_00019]
[RS_Main_00514]	AUTOSAR shall support the development of secure systems	[RS_IAM_00001] [RS_IAM_00002] [RS_IAM_00004] [RS_IAM_00005] [RS_IAM_00006] [RS_IAM_00007] [RS_IAM_00008] [RS_IAM_00009] [RS_IAM_00010] [RS_IAM_00011] [RS_IAM_00014] [RS_IAM_00017] [RS_IAM_00018] [RS_IAM_00019] [RS_IAM_00020]

6 References

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_TR_Glossary

[3] Main Requirements
AUTOSAR_RS_Main