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

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	18-10

Document Change History

Date	Release	Changed by	Description
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

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

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

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

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

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Scope of 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 Acronyms and abbreviations

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

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.
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`.
Capability	A capability is a property of an `Adaptive Application`. Access to an AUTOSAR resource is granted if a requesting AA possesses all capabilities that are necessary for that specific AUTOSAR Resource.

Table 2.1: Acronyms and Abbreviations

3 Requirements Tracing

The following table references the features specified in [2] 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_00003] [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]

4 Functional Overview

Identity and Access Management (IAM) provides services for Adaptive Applications 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.

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. E.g., if there is a port mapping modeled, this gives us the information that the corresponding service proxy is allowed to access the service stub. When the port mapping is not modeled, this must result in access not being granted.

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

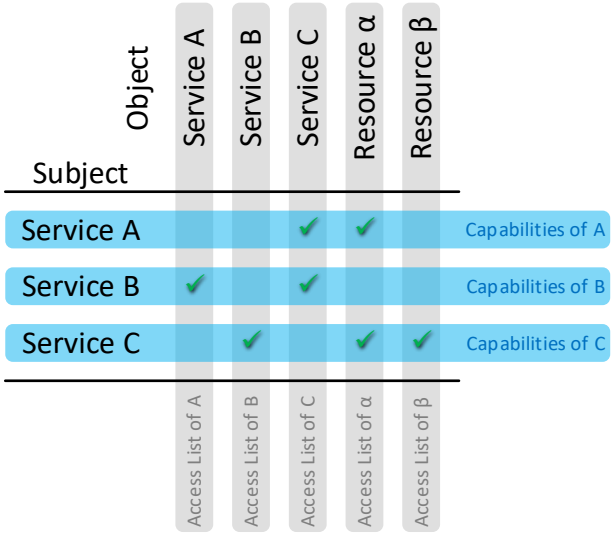

Figure 4.1: Access Matrix

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 *capabilities*–, 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 capabilities 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.

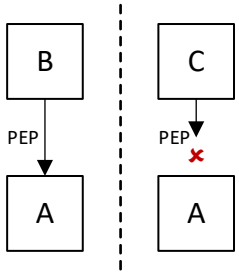

Figure 4.2: Access Control by Policy Enforcement

As previously declared in Fig. 4.1, service B is allowed to access service A – B has the capability to access A. However, service C does not have the capability to access A. A *Policy Enforcement Point* (PEP) must supervise the interaction and thus prevent the

access of C on A. The information, whether a capability 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 capabilities provided in a manifest must be authenticated. An attacker should not be able to change the capabilities 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.

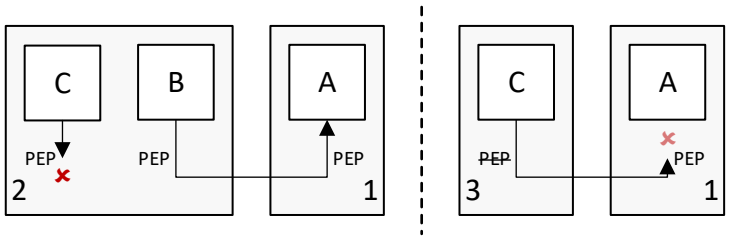

Figure 4.3: Double Access Control for Inter-Platform Communication

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 capability 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 capability, the platform instance need to exchange their capability information. 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.

5 Requirements Specification

5.1 Functional Requirements

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

Type:	draft
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 deploying App. Attacker controls App during runtime. Attacker gains no more permissions than App's initial permissions.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00002] Enforcement of access control shall happen within `Adaptive Platform Foundation` ⌈

Type:	draft
Description:	Access control to interfaces of the Adaptive Platform Foundation and Services shall be enforced by the `Adaptive Platform Foundation` and not by the application.
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 requests. The PEP shall be implemented and executed using a separate process not under control of the actual `Adaptive Application`. IPC must separate `Adaptive Application` from PEP. Access to Adaptive Platform Services are controlled by PEP in `Adaptive Platform Foundation`. For Functional Clusters that are executed in the process context of an `Adaptive Application` (e.g. Persisteny) the Operating System shall be incorporated for access control.
Dependencies:	–
Use Case:	App requests a method on Service Interface. Language binding library implements isolation of processes. IAM identifies App, App cannot spoof its identity. IAM enforces access restrictions.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00003] Applications shall be prevented from taking control over the AUTOSAR PEP ⌈

Type:	draft
Description:	A Policy Enforcement Point (PEP) must not run in the context of an `Adaptive Application` it shall enforce policies on. Therefore each PEP shall be part of the `Adaptive Platform Foundation`.
Rationale:	Enforcement of access restrictions to AUTOSAR Resources should happen within the `Adaptive Platform Foundation`.
Dependencies:	RS_SEC_5019, RS_SEC_5018
Use Case:	Attacker gains control over execution process of App. In the context of App the attacker requests access on resource. Access Control is enforced in context of `Adaptive Platform Foundation` and thus not under control of attacker.
Supporting Material:	Requirements on Security Management for Adaptive Platform[3]

⌋*(RS_Main_00514)*

[RS_IAM_00004] Circumvention of AUTOSAR PEP interfaces by Applications shall be prevented. ⌈

Type:	draft
Description:	`Adaptive Platform` shall prevent Applications from circumventing AUTOSAR PEPs 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:	Capabilites 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] `Adaptive Platform Foundation` shall enforce that only Applications that are configured accordingly are able to gain information about the permissions of other applications ⌈

Type:	draft
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] Access control policies shall be available to the PDP ⌈

Type:	draft
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 Applications's Capabilities 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 hsa to return binary decision, if identified App brings capabilities that fullfill policy.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00007] The `Adaptive Platform Foundation` shall provide access control decisions ⌈

Type:	draft
Description:	The `Adaptive Platform Foundation` shall provide access control decisions based on capabilities 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 capabilities. 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 capabilities of Application A. PDP returns access control decision to PEP.
Supporting Material:	–

⌋(*RS_Main_00514*)

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

Type:	draft
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] An `Adaptive Application` may provide access control decisions ⌈

Type:	draft
Description:	The adaptive `Adaptive Platform Foundation` shall provide an interface to `Adaptive Application` to facilitate access control decisions based on access control policies and capabilities 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 Capabilities are well-defined by AUTOSAR. OEM-IAM enables the adaptive integration of OEM-specific acces 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] Adaptive applications shall only be able to use AUTOSAR Resources when authorized ⌈

Type:	draft
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. In contrast to e.g. methods of Service Interfaces, access control on resources is more fine grained: differents kind of access must be distinguished. According Capabilites define the type of access an `Adaptive Application` may execute. In the example of key-slots it is possible to define an `Adaptive Application` as owner of a specific key-slot. This enables write-access while another role just allow read-access.
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] Policies shall be enforced by the local `Adaptive Platform Foundation` ⌈

Type:	draft
Description:	Policies shall be enforced by the local `Adaptive Platform`, i.e., that `Adaptive Platform` that runs on the machine of the requesting application.
Rationale:	Requests to remote machines shall be restricted on requesting side since Application's identity is available on local machine.
Dependencies:	–
Use Case:	Scenario 1: Application Aa on `Adaptive Platform` Pa requests access on Service Interface implemented by Application Ab on `Adaptive Platform` Pb. Application Aa does not bring necessary capabilities. PEP of Communicaton Management denies access with the help of PDP and does not forward request to Pb.

▽

▽

△

	△ Scenario 2: Application Aa on `Adaptive Platform` Pa requests access on Service Interface implemented by Application Ab on `Adaptive Platform` Pb. Application Aa does bring necessary capabilities. PEP of Communicaton Management accepts the request with the help of PDP and forwards the request to Pb. On Pb an additional check based on the platform's identity (Pa) is executed.
Supporting Material:	–

⌋(*RS_Main_00514*)

[RS_IAM_00014] Unique `Adaptive Application` ID ⌈

Type:	draft
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] Identity information shall be stored and handled tamper-proof throughout its lifecycle. ⌈

Type:	draft
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 Capabilities 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] Set of capabilities shall be provided in the corresponding manifest ⌈

Type:	draft
Description:	The set of capabilities of an `Adaptive Application` shall be provided in the corresponding manifest.
Rationale:	Capabilities defined for an `Adaptive Application` shall be determined by the corresponding manifest. If an `Adaptive Application` is compromised, we need the manifest with the capabilities to actually enforce the restrictions implied by the capabilities. 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 capabilites 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] Capabilities of an `Adaptive Application` shall be authentically linked to the manifest ⌈

Type:	draft
Description:	The set of capabilities 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 capabilities. It shall not be possible to extend or restrict this set except by signed updates. The `Adaptive Application` should always possess the same capabilities 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. Capabilities of App are still determined and privilege escalation is prevented.
Supporting Material:	–

⌋(*RS_Main_00514*, *RS_Main_00510*)

[RS_IAM_00020] `Adaptive Platform Foundation` must allow to specify a superset manifest file of capabilities ⌈

Type:	draft
Description:	`Adaptive Platform Foundation` shall allow to specify a superset manifest file of capabilities.
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 a capability 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 capability. 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 capability to access B. This information is provided by the superset manifest of P_A.
Supporting Material:	–

⌋(*RS_Main_00514*)

6 References

[1] Glossary
AUTOSAR_TR_Glossary

[2] Main Requirements
AUTOSAR_RS_Main

[3] Requirements on Security Management for Adaptive Platform
AUTOSAR_RS_SecurityManagement