Document Title	Requirements on Execution Management
Document Category	Requirement specification (RS)
Document UID	720
Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R26-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2026-11-26	R26-11	AUTOSAR Release Management	• Removed: RS_EM_00014 • Minor changes, document clean up
2025-11-27	R25-11	AUTOSAR Release Management	• Minor changes, document clean up
2024-11-27	R24-11	AUTOSAR Release Management	• Requirements for deterministic execution are removed • Minor changes, document clean up
2023-11-23	R23-11	AUTOSAR Release Management	• Requirements for deterministic execution are set to obsolete • The right to create child processes can be configured by integrator • Added support for standardized trace points
2022-11-24	R22-11	AUTOSAR Release Management	• Added: RS_EM_00151 • Changed uptraces to RS_SAF requirements
2021-11-25	R21-11	AUTOSAR Release Management	• Added: RS_EM_00015
2020-11-30	R20-11	AUTOSAR Release Management	• Added: RS_EM_00150
2019-11-28	R19-11	AUTOSAR Release Management	• Updated: RS_EM_00009 and RS_EM_00103 • Changed Document Status from Final to published

△

2019-03-29	19-03	AUTOSAR Release Management	• Updated: RS_EM_00008 and RS_EM_00010
2018-10-31	18-10	AUTOSAR Release Management	• Removed: RS_EM_00003, RS_EM_00004, RS_EM_00110 and RS_EM_00111. • Added: RS_EM_00014.
2018-03-29	18-03	AUTOSAR Release Management	• Removed: RS_EM_00006, RS_EM_00007 and RS_EM_00012 • Minor changes and document clean up
2017-10-27	17-10	AUTOSAR Release Management	• Minor changes, document clean up
2017-03-31	17-03	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 6

2 Conventions to be used 7
2.1 Document Conventions . . . 7
2.2 Requirements Guidelines . . . 7
2.2.1 Requirements status . . . 7

3 Acronyms and abbreviations 8

4 Requirements Specification 10
4.1 Functional Overview . . . 10
4.2 Functional Requirements . . . 11
4.2.1 Startup and Shutdown of applications . . . 11
4.2.2 Execution . . . 14
4.2.3 State Management . . . 15
4.2.4 Error Handling . . . 16
4.2.5 Support for Diagnostics . . . 16
4.3 Non-Functional Requirements . . . 17

5 References 18

A Change history of AUTOSAR traceable items 19
A.1 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 19
A.1.1 Added Requirements in R26-11 . . . 19
A.1.2 Changed Requirements in R26-11 . . . 19
A.1.3 Deleted Requirements in R26-11 . . . 19
A.2 Traceable item history of this document according to AUTOSAR Release R25-11 . . . 20
A.2.1 Added Requirements in R25-11 . . . 20
A.2.2 Changed Requirements in R25-11 . . . 20
A.2.3 Deleted Requirements in R25-11 . . . 20
A.3 Traceable item history of this document according to AUTOSAR Release R24-11 . . . 21
A.3.1 Added Requirements in R24-11 . . . 21
A.3.2 Changed Requirements in R24-11 . . . 21
A.3.3 Deleted Requirements in R24-11 . . . 21
A.4 Traceable item history of this document according to AUTOSAR Release R23-11 . . . 22
A.4.1 Added Requirements in R23-11 . . . 22
A.4.2 Changed Requirements in R23-11 . . . 22
A.4.3 Deleted Requirements in R23-11 . . . 22
A.5 Traceable item history of this document according to AUTOSAR Release R22-11 . . . 23

A.5.1 Added Requirements in R22-11 . . . 23
A.5.2 Changed Requirements in R22-11 . . . 23
A.5.3 Deleted Requirements in R22-11 . . . 23
A.6 Traceable item history of this document according to AUTOSAR Release R21-11 . . . 24
A.6.1 Added Requirements in R21-11 . . . 24
A.6.2 Changed Requirements in R21-11 . . . 24
A.6.3 Deleted Requirements in R21-11 . . . 24
A.7 Traceable item history of this document according to AUTOSAR Release R20-11 . . . 24
A.7.1 Added Requirements in R20-11 . . . 24
A.7.2 Changed Requirements in R20-11 . . . 24
A.7.3 Deleted Requirements in R20-11 . . . 24
A.8 Traceable item history of this document according to AUTOSAR Release R19-11 . . . 25
A.8.1 Added Requirements in 19-11 . . . 25
A.8.2 Changed Requirements in 19-11 . . . 25
A.8.3 Deleted Requirements in 19-11 . . . 25
A.9 Traceable item history of this document according to AUTOSAR Release 19-03 . . . 25
A.9.1 Added Requirements in 19-03 . . . 25
A.9.2 Changed Requirements in 19-03 . . . 25
A.9.3 Deleted Requirements in 19-03 . . . 25
A.10 Traceable item history of this document according to AUTOSAR Release 18-10 . . . 26
A.10.1 Added Requirements in 18-10 . . . 26
A.10.2 Changed Requirements in 18-10 . . . 26
A.10.3 Deleted Requirements in 18-10 . . . 26
A.11 Traceable item history of this document according to AUTOSAR Release 18-03 . . . 27
A.11.1 Added Requirements in 18-03 . . . 27
A.11.2 Changed Requirements in 18-03 . . . 27
A.11.3 Deleted Requirements in 18-03 . . . 27
A.12 Traceable item history of this document according to AUTOSAR Release 17-10 . . . 28
A.12.1 Added Requirements in 17-10 . . . 28
A.12.2 Changed Requirements in 17-10 . . . 28
A.12.3 Deleted Requirements in 17-10 . . . 28
A.13 Traceable item history of this document according to AUTOSAR Release 17-03 . . . 28
A.13.1 Added Requirements in 17-03 . . . 28
A.13.2 Changed Requirements in 17-03 . . . 29
A.13.3 Deleted Requirements in 17-03 . . . 29

1 Scope of Document

This document specifies requirements of the AUTOSAR Adaptive Platform on the Execution Management. The motivation is to provide a standardized way to start, stop and police applications platform wide.

2 Conventions to be used

2.1 Document Conventions

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

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

2.2 Requirements Guidelines

There are no special requirement guidelines for this document.

2.2.1 Requirements status

The following requirements are described within this document but not otherwise considered in this release:

- [RS_EM_00111] – Identification of Processes

The functionality described above is subject to modification and will be considered for inclusion in a future release of this document.

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to this Document.

Abbreviation / Acronym:	Description:
Modelled Process	A Modelled Process is an instance of an `Executable` to be executed on a `Machine` and has a 1:1 association with the ARXML/Meta-Model element Process.This document also uses the term `Process`(without the "modelled" prefix) to refer to the OS concept of a running process.
Execution Dependency	Dependencies between `Modelled Processes` instances can be configured to define a sequence for starting and terminating them.
Execution Management	The element of the `AUTOSAR Adaptive Platform` responsible for the ordered startup and shutdown of the `AUTOSAR Adaptive Platform` and `Adaptive Applications`.
Function Group	A `Function Group` is a set of `Modelled Processes` which need to be controlled consistently. Depending on the state of the `Function Group`,`Processes` (related to the `Modelled Processes`) are started or terminated. `Modelled Processes` can belong to more than one `Function Group State`, but at exactly one `Function Group` (if a `Modelled Process` were allowed to be running at the same point in time in more than one `Function Group` then potential contradictions in the logic of the `Function Group` state transitions could cause errors).`Function Groups` are sort of general purpose tools used to control `Processes` of user-level `Adaptive Applications`.
Function Group State	A `Function Group State` defines a set of `Modelled Processes` that shall be started during `Function Group State` transition. Please note that a `Function Groups` can have only one `Function Group State` active at any given point of time.
Machine State	A state of `Function Group` named "`MachineFG`".
MachineFG	MachineFG is a `Function Group` with a predefined name, which is mainly used to control Machine lifecycle and `Processes` of platform level application.
Process State	Lifecycle state of a `Modelled Process`

Table 3.1: Technical Terms

The following technical terms used in this document are defined in the corresponding document mentioned in the table below.

Abbreviation / Acronym:	Description:
State Management	see [2] Requirements of State Management
Identity and Access Management	see [3] Explanation of Identity and Access Management
Communication Management	see [4] Software Specification of Communication Management
Operating System	see [5] Requirements Specification of Operating System Interface
Platform Health Management	see [6] Specification of Platform Health Management
Execution Manifest	see [7] Methodology for Adaptive Platform
Machine Manifest	see [7] Methodology for Adaptive Platform
Service Instance Manifest	see [7] Methodology for Adaptive Platform
Process	see [8] AUTOSAR Glossary
Adaptive Application	see [8] AUTOSAR Glossary
AUTOSAR Adaptive Platform	see [8] AUTOSAR Glossary
Adaptive Platform Foundation	see [8] AUTOSAR Glossary
Manifest	see [8] AUTOSAR Glossary
Executable	see [8] AUTOSAR Glossary



△

Functional Cluster	see [8] AUTOSAR Glossary
Adaptive Platform Service	see [8] AUTOSAR Glossary
Machine	see [8] AUTOSAR Glossary
Service	see [8] AUTOSAR Glossary
Service Interface	see [8] AUTOSAR Glossary
Service Discovery	see [8] AUTOSAR Glossary
Trusted Platform	see [8] AUTOSAR Glossary

Table 3.2: Reference to Technical Terms

4 Requirements Specification

This chapter describes all requirements driving the work to define the Execution Management.

4.1 Functional Overview

The AUTOSAR Adaptive Platform provides services to influence the lifecycle of applications based on configuration. This document therefore includes requirements that determine the facilities provided by Execution Management to affect the machine-wide startup, shutdown and restart of an application based on configuration.

Execution Management is responsible for all aspects of platform lifecycle management and application lifecycle management, including:

- Machine startup and shutdown.
  - Execution Management is the initial (“boot”) process of the operating system.
- Required process hierarchy of started services, e.g., init and its child process.
  - after booting. The boot process in this case corresponds to machine init process.
- Provision of process isolation with each instance of an Executable managed as a single process.
- Startup and shutdown of applications.
  - Loading Executable based on a defined Execution Dependency.
  - Specific requirements until starting an Executable main function (i.e. entry point)
- Privileges and use of access control
  - description and semantics of access control in manifest files
- State management
  - Conditions for the execution of applications

Execution Management, Platform Health Management and State Management are the main safety relevant functional clusters of the AUTOSAR Adaptive Platform. Consequently, their development may require certain processes to be followed - as recommended in ISO26262. A safety argumentation for the AUTOSAR Adaptive Platform, describing functional safety measures and use-cases is provided through Explanation of Safety Overview [9].

4.2 Functional Requirements

4.2.1 Startup and Shutdown of applications

[RS_EM_00002] Execution Management shall set-up one process for the execution of each Modelled Process.

Upstream requirements: RS_SAF_10037

⌈

Description:	For each instance of an `Executable`, `Execution Management` shall allocate one POSIX process. Furthermore process specific properties (like priority, scheduling policy and access rights) shall be assigned based on the `Execution Manifest`.
Rationale:	Isolation of `Executable` instances from each other.
Dependencies:	–
Use Case:	Safety and security related applications require isolation.
Supporting Material:	–

⌋

[RS_EM_00015] Execution Management shall support integrity and authenticity monitoring. ⌈

Description:	`Execution Management` shall support configurable integrity and authenticity monitoring for all `Executables` and their corresponding `Execution Management` meta-data (i.e. processed Machine and Execution Manifests).
Rationale:	`Execution Management` takes over the responsibility from `Operating System` and/or boot loader for `AUTOSAR Adaptive Platform` startup and hence for keeping the platform trusted. `Execution Management` is the only `AUTOSAR Adaptive Platform` entity allowed to start `Executables` and therefore responsible for the continuation of trust for the `AUTOSAR Adaptive Platform`. However unsigned SW (or incorrectly signed SW) may at times be used and to allow this, `Execution Management` should optionally support execution. However the presence of such deployments should be noted.
Dependencies:	–
Use Case:	Support deployment of prototype (unsigned) software during system development.
Supporting Material:	–

⌋

[RS_EM_00005] Execution Management shall support the configuration of OS resource budgets for process and groups of processes.

Upstream requirements: RS_SAF_10008

⌈

Description:	Based on the `Execution Manifest`, `Execution Management` shall allocate OS resources to the `Process`. The allocation shall be possible for single `Process` and groups of `Processes`.
Rationale:	Real-time guarantees shall be defined
Dependencies:	–
Use Case:	Like `cgroups` (based on containers which contain one or more processes) and `ulimit`.
Supporting Material:	–

⌋

[RS_EM_00008] Execution Management shall support the binding of all threads of a given process to a specified set of processor cores.

Upstream requirements: RS_SAF_10008

⌈

Description:	`Execution Management` shall allow the binding of threads to specific set of processor cores based on configuration in the `Execution Manifest`. The binding granularity shall be at process level.
Rationale:	Mechanism to influence load balancing, reaction times, and latencies.
Dependencies:	–
Use Case:	A `Process` can be assigned to designated cores to limit thread migration between cores available on the `Machine`.
Supporting Material:	–

⌋

[RS_EM_00009] Execution Management shall control the right to create child process for each process it starts.

Upstream requirements: RS_SAF_10001, RS_SAF_10008

⌈

Description:	`Execution Management` is responsible for starting child `Processes` and shall prevent such child `Processes` from directly starting other `Processes`, unless configured otherwise.

▽

△

Rationale:	Execution Management needs full control of starting applications to ensure required isolation of temporal and spatial properties. However, existing software may require rights to create child Processes and it can be unpractical to modify it for use with AUTOSAR Adaptive Platform. For this reason, Execution Management allows selected Processes to create child Processes, but this must be configured by integrator and is not a right that is granted by default.
Dependencies:	–
Use Case:	Segregation between applications with different safety and/or security properties.
Supporting Material:	–

⌋

[RS_EM_00010] Execution Management shall support multiple instances of Executables. ⌈

Description:	It shall be possible to start more than one Modelled Process from a single Executable. Instance specific information is described in Modelled Process startup configuration.
Rationale:	Avoid code duplication.
Dependencies:	–
Use Case:	Redundancy of an Executable by parallel execution of two instances.
Supporting Material:	–

⌋

[RS_EM_00011] Execution Management shall support self-initiated graceful shutdown of processes. ⌈

Description:	Execution Management shall support self-initiated graceful shutdown of Processes.
Rationale:	Self-initiated graceful shutdown enables a Process to free allocated dedicated resources and inform other interacting entities about its shutdown (e.g. de-registering a service) to create a consistent state within the Machine/vehicle. Self-initiated Process shutdown is, by definition, only be initiated by the Process itself.
Dependencies:	–
Use Case:	The process of an Executable instance is finished and shuts down itself.
Supporting Material:	–

⌋

[RS_EM_00100] Execution Management shall support the ordered startup and shutdown of processes. ⌈

Description:	Execution Management shall support the ordered startup and shutdown of Executable instances.
Rationale:	Ensure that startup and shutdown dependencies between Executable instances are respected, if an execution dependency is specified in the Execution Manifest of an Executable instance. If no execution dependency is specified between Executable instances, they can be started and stopped in an arbitrary order.
Dependencies:	–
Use Case:	An Executable needs a specific functional cluster to be up and running before it can be started.
Supporting Material:	–

⌋

4.2.2 Execution

[RS_EM_00111] Execution Management shall assist identification of processes during Machine runtime. ⌈

Description:	Adaptive Applications shall be identifiable, for example by Identity and Access Management, during runtime so that access restrictions can be enforced. Execution Management spawns runtime processes based on Execution Manifest. Execution Management is qualified to assist AUTOSAR Adaptive Platform software, such as Identity and Access Management, by providing information about the link between runtime representation and Modelled Process.
Rationale:	Adaptive Applications shall be identifiable by Identity and Access Management on the basis of their runtime representation as spawned by Execution Management.
Dependencies:	–
Use Case:	App A requests access on Service Interface. Identity and Access Management is able to retrieve runtime information of App A, e.g. POSIX pid or cryptographic token. Execution Management assists Identity and Access Management by resolving this runtime information to the Adaptive Application.
Supporting Material:	–

⌋

[RS_EM_00152] Execution Management shall support standardized trace points throughout the state transitions. ⌈

Description:	Execution Management shall support standardized trace points throughout the state transitions.
Rationale:	Providing standardized trace points in a Functional Cluster allows comparison of timing in different implementations upon state changes, such as upon application process creation, initialization, and termination.
Dependencies:	ara::log
Use Case:	Tracing and timing analysis of different applications and Functional Clusters behavior.
Supporting Material:	–

⌋

4.2.3 State Management

[RS_EM_00101] Execution Management shall support State Management functionality. ⌈

Description:	Execution Management shall provide an interface to State Management to request a change in Function Group State.
Rationale:	To support the starting and stopping of Processes based on declared Function Group State dependencies, Execution Management provides an interface to request Function Group State (including Machine State) changes by the State Management fuctional cluster. In response to state change requests, Execution Management ensures that only the required set of application Processes are running in any given operation conditions and therefore platform resources are saved for relevant Processes.
Dependencies:	–
Use Case:	Provide a mechanism to define modes of operation of the Machine.
Supporting Material:	–

⌋

[RS_EM_00103] Execution Management shall support process lifecycle management. ⌈

Description:	The lifecycle of a Process consists of its initialization, running and terminating (shutdown) phases. As well as supporting transitions between these phases of the Process lifecycle, Execution Management should ensure that phases, e.g. the startup and shutdown, of Processes can be coordinated between groups of Processes which shall run in the same Machine State or Function Group State. Coordination and tracking of lifecycle phases enables Execution Management to ensure that Executable's Processes are fully established and running before other Processes which depend on their functionality can be started.

▽

△

Rationale:	Coordination and tracking of lifecycle phases enables `Execution Management` to ensure that Executable `Processes` are fully established and running before other executable `Processes` which depend on their functionality can be started.
Dependencies:	–
Use Case:	
Supporting Material:	–

⌋

4.2.4 Error Handling

[RS_EM_00150] Error Handling. ⌈

Description:	`Execution Management` shall support error handling including unrecoverable errors.
Rationale:	`Execution Management` may face conditions where it has no mechanism to recover the system. These situations are typically expected to result from a misconfigured system and therefore a suitable response might be to halt startup so that the misconfiguration can be resolved.
Dependencies:	–
Use Case:	`Execution Management` can not start `Platform Health Management` or `State Management` and hence the platform as a whole cannot be started, it is not possible to recover from this situation hence `Execution Management` must halt startup.
Supporting Material:	–

⌋

4.2.5 Support for Diagnostics

Support for Diagnostics is handled by `State Management` and therefore the requirements are replaced by the ones from [2].

4.3 Non-Functional Requirements

[RS_EM_00151] Execution Management shall be implemented at least according to the highest safety integrity level from any process that is supported on the platform.

Upstream requirements: RS_SAF_10001

⌈

Description:	`Execution Management` shall be implemented at least according to the highest safety integrity level from any `Process` that is supported on the platform.
Rationale:	`Execution Management` manages `Process` instantiation and termination of all the `Processes` and therefore needs to be developed and executed according to the same safety standards as the highest rated safety application managed by `Execution Management` in the system.
Use Case:	An ASIL C, B and QM application is running on the AUTOSAR Adaptive Platform. Execution Management shall execute the ASIL C, B and the QM application, therefore Execution Management shall be implemented with an ASIL C.
Dependencies:	–
Supporting Material:	–

⌋

5 References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] Requirements of State Management
AUTOSAR_AP_RS_StateManagement

[3] Explanation of Identity and Access Management
AUTOSAR_AP_EXP_IdentityAndAccessManagement

[4] Specification of Communication Management
AUTOSAR_AP_SWS_CommunicationManagement

[5] Requirements on Operating System Interface
AUTOSAR_AP_RS_OperatingSystemInterface

[6] Specification of Platform Health Management
AUTOSAR_AP_SWS_PlatformHealthManagement

[7] Methodology for Adaptive Platform
AUTOSAR_AP_TR_Methodology

[8] Glossary
AUTOSAR_FO_TR_Glossary

[9] Explanation of Safety Overview
AUTOSAR_FO_EXP_SafetyOverview

A Change history of AUTOSAR traceable items

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

A.1 Traceable item history of this document according to AUTOSAR Release R26-11

A.1.1 Added Requirements in R26-11

none

A.1.2 Changed Requirements in R26-11

Number	Heading
[RS_EM_00002]	Execution Management shall set-up one process for the execution of each Modelled Process.
[RS_EM_00005]	Execution Management shall support the configuration of OS resource budgets for process and groups of processes.
[RS_EM_00008]	Execution Management shall support the binding of all threads of a given process to a specified set of processor cores.
[RS_EM_00009]	Execution Management shall control the right to create child process for each process it starts.
[RS_EM_00015]	Execution Management shall support integrity and authenticity monitoring.
[RS_EM_00101]	Execution Management shall support State Management functionality.
[RS_EM_00111]	Execution Management shall assist identification of processes during Machine runtime.
[RS_EM_00150]	Error Handling.
[RS_EM_00151]	Execution Management shall be implemented at least according to the highest safety integrity level from any process that is supported on the platform.

Table A.1: Changed Requirements in R26-11

A.1.3 Deleted Requirements in R26-11

Number	Heading
[RS_EM_00014]	Execution Management shall support a Trusted Platform.

Table A.2: Deleted Requirements in R26-11

A.2 Traceable item history of this document according to AUTOSAR Release R25-11

A.2.1 Added Requirements in R25-11

none

A.2.2 Changed Requirements in R25-11

Number	Heading
[RS_EM_00002]	Execution Management shall set-up one process for the execution of each Modelled Process.
[RS_EM_00005]	Execution Management shall support the configuration of OS resource budgets for process and groups of processes.
[RS_EM_00008]	Execution Management shall support the binding of all threads of a given process to a specified set of processor cores.
[RS_EM_00009]	Execution Management shall control the right to create child process for each process it starts.
[RS_EM_00010]	Execution Management shall support multiple instances of Executables.
[RS_EM_00011]	Execution Management shall support self-initiated graceful shutdown of processes.
[RS_EM_00014]	Execution Management shall support a Trusted Platform.
[RS_EM_00015]	Execution Management shall support integrity and authenticity monitoring.
[RS_EM_00100]	Execution Management shall support the ordered startup and shutdown of processes.
[RS_EM_00101]	Execution Management shall support State Management functionality.
[RS_EM_00103]	Execution Management shall support process lifecycle management.
[RS_EM_00111]	Execution Management shall assist identification of processes during Machine runtime.
[RS_EM_00150]	Error Handling.
[RS_EM_00151]	Execution Management shall be implemented at least according to the highest safety integrity level from any process that is supported on the platform.
[RS_EM_00152]	Execution Management shall support standardized trace points throughout the state transitions.

Table A.3: Changed Requirements in R25-11

A.2.3 Deleted Requirements in R25-11

none

A.3 Traceable item history of this document according to AUTOSAR Release R24-11

A.3.1 Added Requirements in R24-11

none

A.3.2 Changed Requirements in R24-11

Number	Heading
[RS_EM_00002]	Execution Management shall set-up one process for the execution of each Modelled Process.
[RS_EM_00005]	Execution Management shall support the configuration of OS resource budgets for process and groups of processes.
[RS_EM_00008]	Execution Management shall support the binding of all threads of a given process to a specified set of processor cores.
[RS_EM_00009]	Execution Management shall control the right to create child process for each process it starts.
[RS_EM_00011]	Execution Management shall support self-initiated graceful shutdown of processes.
[RS_EM_00101]	Execution Management shall support State Management functionality.
[RS_EM_00103]	Execution Management shall support process lifecycle management.
[RS_EM_00150]	Error Handling.
[RS_EM_00151]	Execution Management shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.
[RS_EM_00152]	Execution Management shall support standardized trace points throughout the state transitions.

Table A.4: Changed Requirements in R24-11

A.3.3 Deleted Requirements in R24-11

Number	Heading
[RS_EM_00050]	`Execution Management` shall perform `Machine`-wide coordination of process.
[RS_EM_00051]	`Execution Management` shall provide APIs to the process for configuring external trigger conditions for its activities.
[RS_EM_00052]	`Execution Management` shall provide APIs to the process for configuring cyclic triggering of its activities.
[RS_EM_00053]	`Execution Management` shall provide APIs to the process to support deterministic redundant execution of process.

▽

△

Number	Heading
[RS_EM_00113]	`Execution Management` shall support time-triggered execution.

Table A.5: Deleted Requirements in R24-11

A.4 Traceable item history of this document according to AUTOSAR Release R23-11

A.4.1 Added Requirements in R23-11

Number	Heading
[RS_EM_00152]	`Execution Management` shall support standardized trace points throughout the state transitions.

Table A.6: Added Requirements in R23-11

A.4.2 Changed Requirements in R23-11

Number	Heading
[RS_EM_00008]	`Execution Management` shall support the binding of all threads of a given process to a specified set of processor cores.
[RS_EM_00009]	`Execution Management` shall control the right to create child process for each process it starts.
[RS_EM_00053]	`Execution Management` shall provide APIs to the process to support deterministic redundant execution of process.
[RS_EM_00151]	Execution Management shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.

Table A.7: Changed Requirements in R23-11

A.4.3 Deleted Requirements in R23-11

none

A.5 Traceable item history of this document according to AUTOSAR Release R22-11

A.5.1 Added Requirements in R22-11

Number	Heading
[RS_EM_00151]	Execution Management shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.

Table A.8: Added Requirements in R22-11

A.5.2 Changed Requirements in R22-11

Number	Heading
[RS_EM_00002]	`Execution Management` shall set-up one process for the execution of each `Modelled Process`.
[RS_EM_00005]	`Execution Management` shall support the configuration of OS resource budgets for process and groups of process.
[RS_EM_00008]	`Execution Management` shall support the binding of all threads of a given process to a specified set of processor cores.
[RS_EM_00009]	`Execution Management` shall ensure it is the sole entity starting process.
[RS_EM_00050]	`Execution Management` shall perform `Machine`-wide coordination of process.
[RS_EM_00053]	`Execution Management` shall provide APIs to the process to support deterministic redundant execution of process.
[RS_EM_00103]	`Execution Management` shall support process lifecycle management.
[RS_EM_00113]	`Execution Management` shall support time-triggered execution.
[RS_EM_NA]	

Table A.9: Changed Requirements in R22-11

A.5.3 Deleted Requirements in R22-11

none

A.6 Traceable item history of this document according to AUTOSAR Release R21-11

A.6.1 Added Requirements in R21-11

Number	Heading
[RS_EM_00015]	Execution Management shall support integrity and authenticity monitoring

Table A.10: Added Requirements in R21-11

A.6.2 Changed Requirements in R21-11

none

A.6.3 Deleted Requirements in R21-11

none

A.7 Traceable item history of this document according to AUTOSAR Release R20-11

A.7.1 Added Requirements in R20-11

Number	Heading
[RS_EM_00113]	Execution Management shall support time-triggered execution
[RS_EM_00150]	Error Handling

Table A.11: Added Requirements in R20-11

A.7.2 Changed Requirements in R20-11

none

A.7.3 Deleted Requirements in R20-11

Number	Heading
[RS_EM_00013]	Execution Management shall support configurable recovery actions

Table A.12: Deleted Requirements in R20-11

A.8 Traceable item history of this document according to AUTOSAR Release R19-11

A.8.1 Added Requirements in 19-11

none

A.8.2 Changed Requirements in 19-11

Number	Heading
[RS_EM_00009]	Only Execution Management shall start Executables

Table A.13: Changed Requirements in 19-11

A.8.3 Deleted Requirements in 19-11

none

A.9 Traceable item history of this document according to AUTOSAR Release 19-03

A.9.1 Added Requirements in 19-03

none

A.9.2 Changed Requirements in 19-03

Number	Heading
[RS_EM_00008]	The Execution Management shall support the binding of Executable threads to a specified set of processor cores.

Table A.14: Changed Requirements in 19-03

A.9.3 Deleted Requirements in 19-03

none

A.10 Traceable item history of this document according to AUTOSAR Release 18-10

A.10.1 Added Requirements in 18-10

Number	Heading
[RS_EM_00014]	Execution Management shall support a Trusted Platform
[RS_EM_00111]	Execution Management shall assist identification of Processes during Machine runtime

Table A.15: Added Requirements in 18-10

A.10.2 Changed Requirements in 18-10

Number	Heading
[RS_EM_00002]	The Execution Management shall set-up one process for the execution of each Executable instance
[RS_EM_00005]	The Execution Management shall support the configuration of OS resource budgets for Executable and groups of Executables
[RS_EM_00011]	Execution Management shall support self-initiated graceful shutdown of Executable instances
[RS_EM_00013]	Execution Management shall support configurable recovery actions
[RS_EM_00101]	The Execution Management shall provide Machine State Management functionality

Table A.16: Changed Requirements in 18-10

A.10.3 Deleted Requirements in 18-10

Number	Heading
[RS_EM_00003]	The Execution Management shall support the checking of the integrity of Executables at startup of Executable
[RS_EM_00004]	The Execution Management shall support the authentication and authorization of Executables at startup of Executable

Table A.17: Deleted Requirements in 18-10

A.11 Traceable item history of this document according to AUTOSAR Release 18-03

A.11.1 Added Requirements in 18-03

Number	Heading
[RS_EM_00151]	Execution Management shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.

Table A.18: Added Requirements in 18-03

A.11.2 Changed Requirements in 18-03

Number	Heading
[RS_EM_00009]	Only Execution Management shall start Executables
[RS_EM_00101]	The Execution Management shall provide Machine State Management functionality
[RS_EM_00103]	Execution Management shall support application lifecycle management
[RS_EM_00110]	Execution Management shall support diagnostic reset cause

Table A.19: Changed Requirements in 18-03

A.11.3 Deleted Requirements in 18-03

Number	Heading
[RS_EM_00006]	The Execution Management shall support the analysis of available and required OS resource budgets for Executables and groups of Executables during installation and run-time
[RS_EM_00007]	The Execution Management shall support of the allocation of dedicated resources for the Executable (e.g GPU)
[RS_EM_00012]	Application Manifest shall support unambiguous identification of Executable instances

Table A.20: Deleted Requirements in 18-03

A.12 Traceable item history of this document according to AUTOSAR Release 17-10

A.12.1 Added Requirements in 17-10

Number	Heading
[RS_EM_00053]	The Execution Management shall provide functions to support redundant execution of Executables
[RS_EM_00110]	Execution Management shall support diagnostic reset cause

Table A.21: Added Requirements in 17-10

A.12.2 Changed Requirements in 17-10

none

A.12.3 Deleted Requirements in 17-10

Number	Heading
[RS_EM_00001]	The Execution Management shall load Executables
[RS_EM_00103]	Execution Management shall support application lifecycle management

Table A.22: Deleted Requirements in 17-10

A.13 Traceable item history of this document according to AUTOSAR Release 17-03

A.13.1 Added Requirements in 17-03

Number	Heading
[RS_EM_00001]	The Execution Management shall load Executables
[RS_EM_00002]	The Execution Management shall set-up one process for the execution of each Executable instance
[RS_EM_00003]	The Execution Management shall support the checking of the integrity of Executables at startup of Executable
[RS_EM_00004]	The Execution Management shall support the authentication and authorization of Executables at startup of Executable
[RS_EM_00005]	The Execution Management shall support the configuration of OS resource budgets for Executable and groups of Executables

▽

△

Number	Heading
[RS_EM_00006]	The Execution Management shall support the analysis of available and required OS resource budgets for Executables and groups of Executables during installation and run-time
[RS_EM_00007]	The Execution Management shall support of the allocation of dedicated resources for the Executable (e.g GPU)
[RS_EM_00008]	The Execution Management shall support the binding of Executable threads to a specified set of processor cores.
[RS_EM_00009]	Only Execution Management shall start Executables
[RS_EM_00010]	The Execution Management shall support multiple instantiation of Executables
[RS_EM_00011]	Execution Management shall support self-initiated graceful shutdown of Executable instances
[RS_EM_00012]	Application Manifest shall support unambiguous identification of Executable instances
[RS_EM_00013]	Execution Management shall support configurable recovery actions
[RS_EM_00100]	The Execution Management shall support the ordered startup and shutdown of Executables
[RS_EM_00050]	The Execution Management shall do a system-wide coordination of activities.
[RS_EM_00051]	The Execution Management shall provide functions to the Executable for configuring external trigger conditions for its activities
[RS_EM_00052]	The Execution Management shall provide functions to the Executable for configuring cyclic triggering of its activities
[RS_EM_00101]	The Execution Management shall provide Machine State Management functionality
[RS_EM_00103]	Execution Management shall support application lifecycle management

Table A.23: Added Requirements in 17-03

A.13.2 Changed Requirements in 17-03

none

A.13.3 Deleted Requirements in 17-03

none