Document Title	Requirements on Platform Health Management
Document Category	Requirement specification (RS)
Document UID	852
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	• Clarified allocation of Supervisions to threads and processes in RS_PHM_00105 and RS_PHM_00106 • Removed "Draft" status from all requirements
2025-11-27	R25-11	AUTOSAR Release Management	• Removed RS_PHM_00107 (Multiple instantiation of PHM) • Clarified and cleaned acronyms and abbreviations
2024-11-27	R24-11	AUTOSAR Release Management	• Removed obsolete Health Channel requirements RS_PHM_00102, RS_PHM_09255 and RS_PHM_09257 • Removed obsolete Daisy Chaining requirements RS_PHM_00108 and RS_PHM_00109 • Removed requirements RS_PHM_00001, RS_PHM_00002 and RS_PHM_00003 • Clarified requirement RS_PHM_00107
2023-11-23	R23-11	AUTOSAR Release Management	• Added requirements RS_PHM_00118 and RS_PHM_00119 • Removed RS_PHM_00103 • RS_PHM_00105: Supervised Entities replaced by Supervisions

▽

△

2022-11-24	R22-11	AUTOSAR Release Management	• Added RS_PHM_00114, RS_PHM_00115, RS_PHM_00116 and RS_PHM_00117 • Modified RS_PHM_00111 (Replaced Local Supervision with Elementary Supervision) • Cleanup of requirement trace
2021-11-25	R21-11	AUTOSAR Release Management	• Added RS_PHM_09255, RS_PHM_09257, RS_PHM_09240, RS_PHM_09241 (moved from FO) • Removed RS_PHM_00110 • Cleanup of requirement trace
2020-11-30	R20-11	AUTOSAR Release Management	• Marked Health Channel related items as obsolete • Added RS_PHM_00111 and RS_PHM_00112 for Mode Dependent Configuration • Modified description of Supervision Mode in RS_PHM_00104
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	• removed references to RS_Main_00330
2018-10-31	18-10	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes
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 6

2 Conventions to be used 7

3 Acronyms and abbreviations 8

4 Requirements Specification 10
- 4.1 Functional Overview . . . 10
- 4.2 Constraints and assumptions . . . 10
  - 4.2.1 Limitations . . . 10
  - 4.2.2 Applicability to car domains . . . 10
- 4.3 Functional Requirements . . . 11
  - 4.3.1 Supervision functions . . . 11
  - 4.3.2 Mapping of `Supervised Entity`s to threads and processes . . . 14
- 4.4 Non-Functional Requirements (Qualities) . . . 16

5 Requirements Tracing 17
- 5.1 Not applicable requirements . . . 17

6 References 18

A Change History of AUTOSAR traceable items 19
- A.1 Traceable item history of this document according to AUTOSAR Release R22-11 . . . 19
  - A.1.1 Added Requirements in R22-11 . . . 19
  - A.1.2 Changed Requirements in R22-11 . . . 19
  - A.1.3 Deleted Requirements in R22-11 . . . 19
- A.2 Traceable item history of this document according to AUTOSAR Release R23-11 . . . 20
  - A.2.1 Added Requirements in R23-11 . . . 20
  - A.2.2 Changed Requirements in R23-11 . . . 20
  - A.2.3 Deleted Requirements in R23-11 . . . 20
- A.3 Traceable item history of this document according to AUTOSAR Release R24-11 . . . 20
  - A.3.1 Added Requirements in R24-11 . . . 20
  - 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 R25-11 . . . 21
  - A.4.1 Added Requirements in R25-11 . . . 21
  - A.4.2 Changed Requirements in R25-11 . . . 21
  - A.4.3 Deleted Requirements in R25-11 . . . 22
- A.5 Traceable item history of this document according to AUTOSAR Release R26-11 . . . 22
  - A.5.1 Added Requirements in R26-11 . . . 22

A.5.2 Changed Requirements in R26-11 22
A.5.3 Deleted Requirements in R26-11 22

1 Scope of Document

This document specifies requirements on Platform Health Management. Platform Health Management implements the Platform Health Monitoring on the AUTOSAR Adaptive Platform.

2 Conventions to be used

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].

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to the specification or implementation of Health Monitoring that are not included in the [2, AUTOSAR glossary].

Abbreviation:	Description:
EM	see [2] AUTOSAR Glossary
PHM	see [2] AUTOSAR Glossary
SE	Supervised Entity
SM	see [2] AUTOSAR Glossary

Table 3.1: Abbreviations

Acronym:	Description:
Adaptive Application	see [2] AUTOSAR Glossary
Alive Supervision	Mechanism to check the timing constraints of cyclic Supervised Entityes to be within the configured min and max limits.
ara::com	Communication middleware for the AUTOSAR Adaptive Platform
AUTOSAR Adaptive Platform	see [2] AUTOSAR Glossary
Checkpoint	A point in the control flow of a Supervised Entity where the activity is reported.
Daisy chaining	Chaining multiple instances of Health Monitoring
Deadline End Checkpoint	A Checkpoint for which Deadline Supervision is configured and which is a ending point for a particular Transition. It is possible that a Checkpoint is both a Deadline Start Checkpoint and Deadline End Checkpoint - if Deadline Supervision is chained.
Deadline Start Checkpoint	A Checkpoint for which Deadline Supervision is configured and which is a starting point for a particular Transition.
Deadline Supervision	Mechanism to check that the timing constraints for execution of the transition from a Deadline Start Checkpoint to a corresponding Deadline End Checkpoint are within the configured min and max limits.
Elementary Supervision Status	Status that represents the current state of an Alive Supervision, Deadline Supervision or Logical Supervision, based on the evaluation (correct/incorrect) of the supervision.
Executable	see [2] AUTOSAR Glossary
Execution Management	The element of the AUTOSAR Adaptive Platform responsible for the orderly startup and shutdown of the AUTOSAR Adaptive Platform and the Adaptive Application.
Function Group	A Function Group is a set of coherent Processes, which need to be controlled consistently. Depending on the state of the Function Group, Processes are started or terminated.
Function Group State	The element of State Management that characterizes the current status of a set of (functionally coherent) user-level Adaptive Application. The set of Function Groups and their Function Group States is machine specific and are deployed as part of the Machine Manifest.
Functional Cluster	see [2] AUTOSAR Glossary
Global Supervision Status	Status that summarizes the Elementary Supervision Status of a set of supervisions within a Function Group.

Health Monitoring	Supervision of the software behaviour for correct timing and sequence.
Logical Supervision	Kind of online supervision of software that checks if the software (Supervised Entity or set of Supervised Entities) is executed in the sequence defined by the programmer (by the developed code).
Machine Manifest	Manifest file to configure a Machine.
Machine	see [2] AUTOSAR Glossary
Machine State	The element of the State Management which characterize the current status of the machine. It defines a set of active Adaptive Applications for any certain situation. The set of Machine States is machine specific and it will be deployed in the Machine Manifest. Machine States are mainly used to control machine lifecycle (startup/shut-down/restart) and platform-level processes.
Manifest	see [2] AUTOSAR Glossary
Platform Health Management	Health Monitoring for the AUTOSAR Adaptive Platform.
Process	See [2] AUTOSAR Glossary.
State Management	The element of the Execution Management defining modes of operation for AUTOSAR Adaptive Platform. It allows flexible definition of functions which are active on the platform at any given time.
Supervised Entity	A whole or part of a software component type which is included in the supervision. A Supervised Entity denotes a collection of Checkpoints within the corresponding software component type. A software component type can include zero, one or more Supervised Entities. A Supervised Entity may be instantiated multiple times, in which case each instance is independently supervised. Remark: Safety critical Adaptive Applications and services are considered to be supervised entities, and therefore are expected to be treated as supervised entities within the AUTOSAR Methodology and Architectural Design.
Supervision Mode	State of a machine or Function Group in which Supervised Entity instances are to be monitored with a specific set of configuration parameters. Supervision parameters differ from one mode to other as the behavior (timing or sequence) of Supervised entity changes from one mode to other. Modes are mutually exclusive. A mode can be "Normal", "Degradation".

Table 3.2: Acronyms

4 Requirements Specification

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

4.1 Functional Overview

See RS Health Monitoring [3] for the overview of the functionality.

This document specifies the requirements regarding the realization of the Health Monitoring on the AUTOSAR Adaptive Platform. This includes:

- Standardized interfaces
- Mapping of abstract functionalities/concepts defined in Foundation to entities in the AUTOSAR Adaptive Platform.

EM, PHM and SM 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, for instance [RS_SAF_21101] [4]. A safety argumentation for the AUTOSAR Adaptive Platform, describing functional safety measures and use-cases is provided through Explanation of Safety Overview [5].

4.2 Constraints and assumptions

4.2.1 Limitations

No known limitation.

4.2.2 Applicability to car domains

No restriction.

4.3 Functional Requirements

4.3.1 Supervision functions

[RS_PHM_00101] `Platform Health Management` shall provide a standardized C++ interface for the reporting of `Checkpoints`. ⌈

Description:	`Platform Health Management` shall provide a standardized C++ interface for the reporting of `Checkpoints`.
Rationale:	`Checkpoints` are locations inside the code of `Supervised Entitys`. `Platform Health Management` checks that these locations are reached in correct time and order. Therefore `Platform Health Management` needs to be informed when a `Checkpoint` is reached.
Dependencies:	–
Use Case:	Reporting of reached code locations for `Alive Supervision`, `Deadline Supervision` and `Logical Supervision`.
Supporting Material:	–

⌋

[RS_PHM_09240] `Platform Health Management` shall support multiple occurrences of the same Supervised Entity. ⌈

Description:	`Platform Health Management` shall support multiple occurrences of the same `Supervised Entity`.
Rationale:	An `Adaptive Application` or component can be instantiated multiple times
Dependencies:	–
Use Case:	Multiple occurrences of the same software component or `Adaptive Application` launched multiple times, as separate processes or threads.
Supporting Material:	–

⌋

[RS_PHM_09241] Health Monitoring shall support multiple instances of Checkpoints in a Supervised Entity occurrence. ⌈

Description:	`Platform Health Management` shall support multiple instances of `Checkpoints` in a `Supervised Entity` occurrence, where the number of `Checkpoint` instances at runtime may be variable.
Rationale:	An `Adaptive Application` or component containing a `Checkpoint` can be instantiated multiple times
Dependencies:	–
Use Case:	Parallel/concurrent execution of the same worker threads that execute the same code.
Supporting Material:	–

⌋

[RS_PHM_00111] Platform Health Management shall determine Supervision status ⌈

Description:	Platform Health Management shall determine the Supervision status of Supervisions and Function Groups. i.e. it shall determine the following. • Elementary Supervision Status of Alive, Deadline and Logical Supervisions • Global Supervision Status of whole/part of a Function Group
Rationale:	Global Supervision Status is needed by State Management to trigger recovery action. Global Supervision Status will be an aggregation of Elementary Supervision Status of Supervisions corresponding to processes of a Function Group.
Dependencies:	–
Use Case:	Notification based on Global Supervision status to State Management.
Supporting Material:	–

⌋

[RS_PHM_00112] Platform Health Management shall provide configurable delays of error reactions. ⌈

Description:	Platform Health Management shall provide configurable delays of error reactions.
Rationale:	Giving the time to the whole software to prepare properly to the upcoming recovery actions, e.g. to the reset.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋

[RS_PHM_00115] If supervision of State Management fails then Platform Health Management shall trigger a watchdog reset.

Upstream requirements: RS_SAF_10006, RS_SAF_10030, RS_SAF_10005

⌈

Description:	If supervision of State Management fails then Platform Health Management shall trigger a watchdog reset.
Rationale:	State Management is a fundamental functional cluster of the Adaptive AUTOSAR, if it fails then Platform Health Management (which controls the watchdog) shall trigger a reset which is the only reasonable safety measure
Use Case:	SM is managing a safety critical Adaptive Application. Supervision of SM fails and is detected by PHM. PHM shall trigger a watchdog reset.
Dependencies:	SM

▽

△

Supporting Material:	–

⌋

[RS_PHM_00116] If supervision of Execution Management fails then Platform Health Management shall trigger a watchdog reset.

Upstream requirements: RS_SAF_10006, RS_SAF_10030, RS_SAF_10005

⌈

Description:	If supervision of Execution Management fails then Platform Health Management shall trigger a watchdog reset.
Rationale:	Execution Management is a fundamental functional cluster of the Adaptive AUTOSAR, if it fails then Platform Health Management (which controls the watchdog) shall trigger a reset which is the only reasonable safety measure
Use Case:	EM is managing safety critical `Adaptive Application`s and supervision of EM fails and is detected by PHM. PHM shall trigger a watchdog reset.
Dependencies:	EM
Supporting Material:	–

⌋

[RS_PHM_00117] Platform Health Management shall notify State Management in case an `AUTOSAR Adaptive Platform` functional cluster, `Adaptive Application` or service other than Execution Management and State Management fails.

Upstream requirements: RS_SAF_10005, RS_SAF_10006

⌈

Description:	Platform Health Management shall notify State Management in case an `AUTOSAR Adaptive Platform` functional cluster, `Adaptive Application` or service other than Execution Management and State Management fails.
Rationale:	Recovery actions are coordinated in SM, the failures shall be reported to SM except if SM or EM themselves fail.
Use Case:	PHM supervises a safety critical `Adaptive Application`. This application fails. PHM detects the issue and reports to SM.
Dependencies:	–
Supporting Material:	–

⌋

[RS_PHM_00118] PHM shall only process a checkpoint reported from corresponding processes.

Upstream requirements: RS_SAF_10030

⌈

Description:	PHM shall only process a checkpoint reported from corresponding processes.
Rationale:	The checkpoint can only be considered valid if it was reported from the corresponding configured process.
Use Case:	–
AppliesTo:	AP
Dependencies:	RS_IAM_00002, RS_IAM_00010
Supporting Material:	–

⌋

[RS_PHM_00119] A security event shall be raised if a checkpoint is reported from a non-corresponding process.

Upstream requirements: RS_SAF_10030

⌈

Description:	A security event shall be raised if a checkpoint is reported from a non-corresponding process.
Rationale:	A malicious software might try to enforce a false positive or a false negative by reporting checkpoints corresponding to other processes.
Use Case:	–
AppliesTo:	AP
Dependencies:	RS_IAM_00002, RS_IAM_00010, RS_Ids_00810
Supporting Material:	–

⌋

4.3.2 Mapping of `Supervised Entitys` to threads and processes

[RS_PHM_00104] `Platform Health Management` shall derive the Supervision Mode from Function Group State(s). ⌈

Description:	`Platform Health Management` shall derive the Supervision Mode from Function Group State(s).
Rationale:	Depending on Function Group State, the behavior of process can differ (e.g. other execution path, other timing). Hence, it should be possible to change Supervision configuration based on Function Group State.

▽

△

Dependencies:	RS_HM_09253
Use Case:	The program flow of a Sensor driver could differ between "Normal mode" and "Sensor Learning mode". Logical Supervision configuration will have to be changed between the corresponding Function Group States.
Supporting Material:	–

⌋

[RS_PHM_00105] `Platform Health Management` shall support different allocations/distributions of a Supervision through threads and processes. ⌈

Description:	`Platform Health Management` shall support the following Supervision: • A Supervision belonging to one thread • A Supervision spread across several threads or processes
Rationale:	Algorithms can be executed in one thread, multiple threads or processes. It must be possible to supervise a whole algorithm.
Dependencies:	–
Use Case:	Supervision of the global flow of algorithms distributed to multiple threads or processes.
Supporting Material:	–

⌋

[RS_PHM_00106] `Platform Health Management` shall support allocating of multiple `Supervised Entitys` to the same process. ⌈

Description:	`Platform Health Management` shall support allocating of multiple `Supervised Entitys` to the same process
Rationale:	It shall be possible to define separate `Supervised Entitys` for subfunctions within the same process e.g. threads of that process.
Dependencies:	–
Use Case:	Separate `Supervised Entitys` for different threads within the same process.
Supporting Material:	–

⌋

4.4 Non-Functional Requirements (Qualities)

[RS_PHM_00114] `Platform Health Management` at highest safety integrity level

Upstream requirements: RS_SAF_10001

⌈

Description:	`Platform Health Management` shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.
Rationale:	`Platform Health Management` is responsible for ensuring part of the safe execution of safety relevant processes/applications, it should at least be developed with the highest ASIL as the process/application that is being executed.
Use Case:	An ASIL C, B and QM application is running on the `AUTOSAR Adaptive Platform`. PHM shall supervise the ASIL C and B application, therefore PHM shall be implemented with an ASIL C.
Dependencies:	–
Supporting Material:	–

⌋

5 Requirements Tracing

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

Requirement	Description	Satisfied by
[RS_SAF_10001]	AUTOSAR shall provide mechanisms to support safe initialization of application software.	[RS_PHM_00114]
[RS_SAF_10005]	AUTOSAR shall provide mechanisms to support safe shutdown and termination of application software and embedded middleware.	[RS_PHM_00115] [RS_PHM_00116] [RS_PHM_00117]
[RS_SAF_10006]	AUTOSAR shall provide mechanisms to support safe transition of states in embedded middleware or service life cycle.	[RS_PHM_00115] [RS_PHM_00116] [RS_PHM_00117]
[RS_SAF_10030]	AUTOSAR shall provide mechanisms to support safe program execution.	[RS_PHM_00115] [RS_PHM_00116] [RS_PHM_00118] [RS_PHM_00119]

Table 5.1: Requirements Tracing

5.1 Not applicable requirements

[RS_PHM_NA] ⌈These requirements are not applicable as they are not within the scope of this release.⌋

6 References

[1] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_FO_TR_Glossary

[3] Requirements on Health Monitoring
AUTOSAR_FO_RS_HealthMonitoring

[4] Safety Requirements for AUTOSAR Adaptive Platform and AUTOSAR Classic Platform
AUTOSAR_FO_RS_Safety

[5] 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 specification items that have been removed from the specification in a later version. These constraints and specification items do not appear as hyperlinks in the document.

A.1 Traceable item history of this document according to AUTOSAR Release R22-11

A.1.1 Added Requirements in R22-11

Number	Heading
[RS_PHM_00114]	Platform Health Management shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.
[RS_PHM_00115]	If supervision of State Management fails then Platform Health Management shall trigger a watchdog reset.
[RS_PHM_00116]	If supervision of Execution Management fails then Platform Health Management shall trigger a watchdog reset.
[RS_PHM_00117]	Platform Health Management shall notify State Management in case an AUTOSAR Adaptive Platform functional cluster, application or service other than Execution Management and State Management fails.

Table A.1: Added Requirements in R22-11

A.1.2 Changed Requirements in R22-11

Number	Heading
[RS_PHM_00101]	`Platform Health Management` shall provide a standardized C++ interface for the reporting of `Checkpoints`.
[RS_PHM_00102]	`Platform Health Management` shall provide a standardized C++ interface for the reporting of `Health Channel`.
[RS_PHM_00111]	`Platform Health Management` shall determine Supervision status
[RS_PHM_09241]	Health Monitoring shall support multiple instances of Checkpoints in a Supervised Entity occurrence.

Table A.2: Changed Requirements in R22-11

A.1.3 Deleted Requirements in R22-11

none

A.2 Traceable item history of this document according to AUTOSAR Release R23-11

A.2.1 Added Requirements in R23-11

Number	Heading
[RS_PHM_00118]	PHM shall only process a checkpoint reported from corresponding processes.
[RS_PHM_00119]	A security event shall be raised if a checkpoint is reported from a non-corresponding process.

Table A.3: Added Requirements in R23-11

A.2.2 Changed Requirements in R23-11

Number	Heading
[RS_PHM_00105]	`Platform Health Management` shall support different allocations/distributions of a Supervision through threads and processes.

Table A.4: Changed Requirements in R23-11

A.2.3 Deleted Requirements in R23-11

Number	Heading
[RS_PHM_00103]	`Platform Health Management` functionality shall be available within the same process and as a separate one.

Table A.5: Deleted Requirements in R23-11

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_PHM_00107]	`Platform Health Management` shall support multiple instantiation on different platforms.

Table A.6: Changed Requirements in R24-11

A.3.3 Deleted Requirements in R24-11

Number	Heading
[RS_PHM_00001]	The `Platform Health Management` shall provide a standardized header file structure for each service.
[RS_PHM_00002]	The service header files shall define the namespace for the respective service.
[RS_PHM_00003]	The `Platform Health Management` shall define how language specific data types are derived from modeled data types.
[RS_PHM_00102]	`Platform Health Management` shall provide a standardized C++ interface for the reporting of `Health Channel`.
[RS_PHM_00108]	`Platform Health Management` shall provide a standardized interface between `Platform Health Management` components used in a daisy chain.
[RS_PHM_00109]	`Platform Health Management` shall provide the `Daisy chaining` interface over `ara::com`.
[RS_PHM_09255]	`Platform Health Management` shall provide an interface to receive Health Channel supervision status
[RS_PHM_09257]	`Platform Health Management` shall provide an interface to Supervised Entities to report their health status.

Table A.7: Deleted Requirements in R24-11

A.4 Traceable item history of this document according to AUTOSAR Release R25-11

A.4.1 Added Requirements in R25-11

none

A.4.2 Changed Requirements in R25-11

none

A.4.3 Deleted Requirements in R25-11

Number	Heading
[RS_PHM_00107]	`Platform Health Management` shall support multiple instantiation on different platforms.

Table A.8: Deleted Requirements in R25-11

A.5 Traceable item history of this document according to AUTOSAR Release R26-11

A.5.1 Added Requirements in R26-11

none

A.5.2 Changed Requirements in R26-11

Number	Heading
[RS_PHM_00105]	`Platform Health Management` shall support different allocations/ distributions of a Supervision through threads and processes.
[RS_PHM_00106]	`Platform Health Management` shall support allocating of multiple `Supervised Entitys` to the same process.

Table A.9: Changed Requirements in R26-11

A.5.3 Deleted Requirements in R26-11

none