

<b>Document Title</b>	Requirements on SOME/IP Protocol
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	800

<b>Document Status</b>	published
<b>Part of AUTOSAR Standard</b>	Foundation
<b>Part of Standard Release</b>	R23-11

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2023-11-23	R23-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> </ul>
2022-11-24	R22-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> </ul>
2021-11-25	R21-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> </ul>
2020-11-30	R20-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> </ul>
2019-11-28	R19-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> <li>Changed Document Status from Final to published</li> </ul>
2019-03-29	1.5.1	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> </ul>
2018-10-31	1.5.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> </ul>
2018-03-29	1.4.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> </ul>
2017-12-08	1.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>No content changes</li> </ul>



△

2017-10-27	1.2.0	AUTOSAR Release Management	<ul style="list-style-type: none"><li>• Enhanced formal quality of requirements and requirements tracing</li></ul>
2017-03-31	1.1.0	AUTOSAR Release Management	<ul style="list-style-type: none"><li>• Added requirement for Serialization of Structured Datatypes and Arguments with Identifier and optional members</li></ul>
2016-11-30	1.0.0	AUTOSAR Release Management	<ul style="list-style-type: none"><li>• Initial release</li></ul>

## **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.

## Contents

1	Scope of this document	7
1.1	Document Conventions	8
2	Acronyms and Abbreviations	9
3	Requirements Tracing	10
4	Requirements	11
4.1	General	11
	[RS_SOMEIP_00002] SOME/IP protocol shall provide service-based communication	11
	[RS_SOMEIP_00041] SOME/IP protocol shall provide support of multiple versions of the protocol	11
	[RS_SOMEIP_00003] SOME/IP protocol shall provide support of multiple versions of a service interface	11
	[RS_SOMEIP_00004] SOME/IP protocol shall support event communication	12
	[RS_SOMEIP_00005] SOME/IP protocol shall support different strategies for event communication	12
	[RS_SOMEIP_00042] SOME/IP protocol shall support unicast and multicast based event communication	12
	[RS_SOMEIP_00006] SOME/IP protocol shall support uni-directional RPC communication	13
	[RS_SOMEIP_00007] SOME/IP protocol shall support bi-directional RPC communication	13
	[RS_SOMEIP_00008] SOME/IP protocol shall support error handling of RPC communication	14
	[RS_SOMEIP_00009] SOME/IP protocol shall support field communication	14
	[RS_SOMEIP_00010] SOME/IP protocol shall support different transport protocols underneath	14
	[RS_SOMEIP_00011] SOME/IP protocol shall support messages of different lengths	15
	[RS_SOMEIP_00012] SOME/IP protocol shall support session handling	15
	[RS_SOMEIP_00014] SOME/IP protocol shall support handling of protocol errors on receiver side	15
	[RS_SOMEIP_00051] SOME/IP protocol shall provide support for segmented transmission of large data	16
4.2	Services	16
	[RS_SOMEIP_00019] SOME/IP protocol shall identify services using unique identifiers	16
	[RS_SOMEIP_00015] SOME/IP protocol shall support multiple instances of a service	16
	[RS_SOMEIP_00016] SOME/IP protocol shall support combining multiple RPC methods, events and fields in one service	17

	[RS_SOMEIP_00017] SOME/IP protocol shall support grouping events into eventgroups . . . . .	17
	[RS_SOMEIP_00018] SOME/IP protocol shall support grouping fields in eventgroups . . . . .	18
	[RS_SOMEIP_00021] SOME/IP protocol shall identify RPC methods of services using unique identifiers . . . . .	18
	[RS_SOMEIP_00022] SOME/IP protocol shall identify events of services using unique identifiers . . . . .	18
	[RS_SOMEIP_00023] SOME/IP protocol shall identify event groups of services using unique identifiers . . . . .	19
	[RS_SOMEIP_00024] SOME/IP protocol shall define reserved identifiers	19
	[RS_SOMEIP_00025] SOME/IP protocol shall support the identification of callers of an RPC using unique identifiers . . . . .	19
4.3	On Wire Representation . . . . .	20
	[RS_SOMEIP_00027] SOME/IP protocol shall define the header layout of messages . . . . .	20
	[RS_SOMEIP_00028] SOME/IP protocol shall specify the serialization algorithm for data . . . . .	20
	[RS_SOMEIP_00029] SOME/IP protocol shall specify how data in the payload are aligned . . . . .	20
	[RS_SOMEIP_00026] SOME/IP protocol shall define the endianness of header and payload . . . . .	21
	[RS_SOMEIP_00030] SOME/IP protocol shall support transporting integer data types . . . . .	21
	[RS_SOMEIP_00031] SOME/IP protocol shall support transporting boolean data type . . . . .	21
	[RS_SOMEIP_00032] SOME/IP protocol shall support transporting float data types . . . . .	22
	[RS_SOMEIP_00033] SOME/IP protocol shall support transporting structured data types . . . . .	22
	[RS_SOMEIP_00034] SOME/IP protocol shall support transporting union data types . . . . .	23
	[RS_SOMEIP_00035] SOME/IP protocol shall support transporting one-dimensional and multi-dimensional array data types . . . . .	23
	[RS_SOMEIP_00036] SOME/IP protocol shall support transporting array data types with a fixed length . . . . .	23
	[RS_SOMEIP_00037] SOME/IP protocol shall support transporting array data types with flexible length . . . . .	24
	[RS_SOMEIP_00038] SOME/IP protocol shall support transporting string types with a fixed length . . . . .	24
	[RS_SOMEIP_00039] SOME/IP protocol shall support transporting string data types with flexible length . . . . .	24
	[RS_SOMEIP_00040] SOME/IP protocol shall support providing the length of a serialized data element in the payload . . . . .	25
	[RS_SOMEIP_00050] SOME/IP protocol shall support serialization of extensible data structs . . . . .	25

- 5 References 26
- A Change history of AUTOSAR traceable items 27
  - A.1 Traceable item history of this document according to AUTOSAR Release R23-11 27
    - A.1.1 Added Requirements in R23-11 27
    - A.1.2 Changed Requirements in R23-11 27
    - A.1.3 Deleted Requirements in R23-11 27

# 1 Scope of this document

This document specifies the requirements of the **Scalable service-Oriented MiddlewarE over IP (SOME/IP)** protocol — an automotive/embedded protocol and the underlying serialization / wire format which can be used for request/response and event-based communication.

The basic motivation to specify "yet another RPC-Mechanism" instead of using an existing infrastructure/technology is the goal to have a technology that:

- Is compatible through as many use-cases and communication partners as possible
- Provides the features required by automotive use-cases
- Is scalable from tiny to large platforms
- Can be implemented on different operating system (i.e. AUTOSAR, GENIVI, and OSEK) and even embedded devices without operating system

## 1.1 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS\_STDT\_00078], see Standardization Template, chapter Support for Traceability ([1]).

The verbal forms for the expression of obligation specified in [TPS\_STDT\_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([1]).



## 2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the SOME/IP specification that are not included in the [2, AUTOSAR glossary].

Abbreviation / Acronym:	Description:
Event	A uni-directional data transmission that is only invoked on changes or cyclically and is sent from the producer of data to the consumers.
Eventgroup	A logical grouping of events and notification events of fields inside a service in order to allow subscription
Field	A field does represent a status and thus has an valid value at all times on which getter, setter and notifier act upon.
Getter	A Request/Response call that allows read access to a field.
Method	A method, procedure, function, or subroutine that is called/invoked.
Notification Event	An event message of the notifier of a field.
Notifier	Sends out event message with a new value on change of the value of the field.
Remote Procedure Call (RPC)	A method call from one ECU to another that is transmitted using messages
Service	A logical combination of zero or more methods, zero or more events, and zero or more fields.
Service Instance	Implementation of a service, which can exist more than once in the vehicle and more than once on an ECU
Setter	A Request/Response call that allows write access to a field.
SOME/IP	Scalable service-Oriented MiddlewarE over IP.
TCP	Transmission Control Protocol.
UDP	User Datagram Protocol.
Union	A data structure that dynamically assumes different data types.

**Table 2.1: Acronyms and Abbreviations**

### 3 Requirements Tracing

The following table references the requirements fulfilled by this document:

Requirement	Description	Satisfied by
[RS_Main_00280]	Standardized Automotive Communication Protocols	[RS_SOMEIP_00002] [RS_SOMEIP_00003] [RS_SOMEIP_00004] [RS_SOMEIP_00005] [RS_SOMEIP_00006] [RS_SOMEIP_00007] [RS_SOMEIP_00008] [RS_SOMEIP_00009] [RS_SOMEIP_00010] [RS_SOMEIP_00011] [RS_SOMEIP_00012] [RS_SOMEIP_00014] [RS_SOMEIP_00015] [RS_SOMEIP_00016] [RS_SOMEIP_00017] [RS_SOMEIP_00018] [RS_SOMEIP_00019] [RS_SOMEIP_00021] [RS_SOMEIP_00022] [RS_SOMEIP_00023] [RS_SOMEIP_00024] [RS_SOMEIP_00025] [RS_SOMEIP_00026] [RS_SOMEIP_00027] [RS_SOMEIP_00028] [RS_SOMEIP_00029] [RS_SOMEIP_00030] [RS_SOMEIP_00031] [RS_SOMEIP_00032] [RS_SOMEIP_00033] [RS_SOMEIP_00034] [RS_SOMEIP_00035] [RS_SOMEIP_00036] [RS_SOMEIP_00037] [RS_SOMEIP_00038] [RS_SOMEIP_00039] [RS_SOMEIP_00040] [RS_SOMEIP_00041] [RS_SOMEIP_00042] [RS_SOMEIP_00050] [RS_SOMEIP_00051]

**Table 3.1: RequirementsTracing**

## 4 Requirements

### 4.1 General

**[RS\_SOMEIP\_00002] SOME/IP protocol shall provide service-based communication** [

<b>Description:</b>	SOME/IP protocol shall provide service-based communication where communication paths are established during runtime.
<b>Rationale:</b>	Service-based communication allows a communication which was not predefined during system design phase.
<b>Use Case:</b>	Communication between partners is not statically defined during system design phase.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00041] SOME/IP protocol shall provide support of multiple versions of the protocol** [

<b>Description:</b>	SOME/IP protocol shall provide support of multiple versions of the protocol to distinguish the version of the messages on the network.
<b>Rationale:</b>	A version is needed to be able to distinguish SOME/IP messages of different versions with different structure in header or payload.
<b>Use Case:</b>	Backward incompatible extension and modifications of SOME/IP where both the old and the new protocol are used within the same network.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00003] SOME/IP protocol shall provide support of multiple versions of a service interface** [

<b>Description:</b>	SOME/IP protocol shall provide support of multiple versions of a service interface to detect incompatible changes of the service interface.
<b>Rationale:</b>	The communication partners need to distinguish different versions of a service.
<b>Use Case:</b>	Backward-incompatible changes of a service interface.
<b>AppliesTo:</b>	FO





<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00280)

**[RS\_SOMEIP\_00004] SOME/IP protocol shall support event communication** [

<b>Description:</b>	SOME/IP protocol shall support event communication which is a uni-directional communication that is produced and sent by the service provider.
<b>Rationale:</b>	Event based communication needs to be considered in the communication over the network.
<b>Use Case:</b>	Communication of data which are produced on event-base, e.g gear-shifts.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00280)

**[RS\_SOMEIP\_00005] SOME/IP protocol shall support different strategies for event communication** [

<b>Description:</b>	SOME/IP protocol shall support different update strategies for event communication to enable communication on a cyclic base or when the value has changed
<b>Rationale:</b>	Different data need to be communicated on different conditions
<b>Use Case:</b>	Some event-based data need to be communicated only on changes, other are communicated in cyclic manner, e.g, to avoid errors if ECUs were restarted during runtime
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00280)

**[RS\_SOMEIP\_00042] SOME/IP protocol shall support unicast and multicast based event communication** [

<b>Description:</b>	SOME/IP protocol shall support unicast and multicast based event communication and automated switching between unicast and multicast based on configurable threshold.
---------------------	---



△

<b>Rationale:</b>	Depending on the number of receiver unicast or multicast communication is more efficient. Unicast messages are a mechanism to hide data for certain receivers that do not need to spend processing resources on reception of unwanted data. Multicast allows to save bandwidth if many receivers shall receive the same data.
<b>Use Case:</b>	Number of subscribers drastically changes during runtime.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00006] SOME/IP protocol shall support uni-directional RPC communication** [

<b>Description:</b>	SOME/IP protocol shall support uni-directional RPC communication which triggers the execution of an RPC without informing the caller about the result
<b>Rationale:</b>	If the caller does not need to be informed about the result of an RPC, uni-directional communication suffices.
<b>Use Case:</b>	Execution of RPCs where the caller is only interested in triggering the RPC but in no result
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00007] SOME/IP protocol shall support bi-directional RPC communication** [

<b>Description:</b>	SOME/IP protocol shall support bi-directional RPC communication and inform the caller about the result
<b>Rationale:</b>	If the caller needs to be informed about the result of an RPC, it has to be communicated back to the caller.
<b>Use Case:</b>	Execution of RPCs and transporting the result back to the caller.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00008] SOME/IP protocol shall support error handling of RPC communication [**

<b>Description:</b>	SOME/IP protocol shall support error handling of RPC communication and communicate errors on callee side back to the caller.
<b>Rationale:</b>	The caller needs to be able to react on remote errors of the callee.
<b>Use Case:</b>	An RPC fails and the failure is considered at the error handling routine of the caller.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00009] SOME/IP protocol shall support field communication [**

<b>Description:</b>	SOME/IP protocol shall support field communication with getters, setters and notification events.
<b>Rationale:</b>	For properties held at a central party, this communication pattern provides the needed access functionality.
<b>Use Case:</b>	One party in the vehicle holds a central property which is set and/or used by multiple other parties.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00010] SOME/IP protocol shall support different transport protocols underneath [**

<b>Description:</b>	SOME/IP protocol shall support different transport protocols underneath. SOME/IP protocol shall transport data based on TCP/IP and UDP/IP.
<b>Rationale:</b>	SOME/IP does not directly operate on IP but needs an transport protocol based on IP that provides basic transport services.
<b>Use Case:</b>	SOME/IP over UDP for time critical communication, SOME/IP over TCP for time-uncritical communication of large data
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00011] SOME/IP protocol shall support messages of different lengths [**

<b>Description:</b>	SOME/IP protocol shall support messages of different lengths. The PDU size shall be variable based on the size of the data to transmit.
<b>Rationale:</b>	Message lengths are application specific.
<b>Use Case:</b>	Some application transmit small data chunks, other transmit large data chunks. Length of data sent by the same application can vary during execution.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00012] SOME/IP protocol shall support session handling [**

<b>Description:</b>	SOME/IP protocol shall support session handling and provide a mechanism to distinguish multiple parallel communications between the same parties.
<b>Rationale:</b>	Session handling is used to distinguish multiple parallel communications between the same parties for the same service.
<b>Use Case:</b>	One client issues multiple RPCs which get handled separately and the responses are correctly assigned to the requests.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00014] SOME/IP protocol shall support handling of protocol errors on receiver side [**

<b>Description:</b>	SOME/IP protocol shall support handling of protocol errors on receiver side and define checks to be performed by the receiver.
<b>Rationale:</b>	Any mismatch between received and expected message on receiver side could lead to an erroneous behavior.
<b>Use Case:</b>	Errors which are a consequence of faulty configuration (either sender or receiver side) can be recognized on receiver side. Received data are not handed over from the communication stack to the receiving applications if they are not expected by an application.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00051] SOME/IP protocol shall provide support for segmented transmission of large data** [

<b>Description:</b>	SOME/IP protocol shall provide support for segmented transmission of large data. The sender shall be able to split one large message into multiple smaller segments. These are transported to the receiver. The receiver shall be able to reassemble the original message.
<b>Rationale:</b>	Time critical data can't be transmitted over TCP but UDP has to be used. UDP has a maximum packet length of 4KB. IP Fragmentation can only be used until 65KB. So, no existing technology provides suitable possibilities of segmentation. For these cases segmentation needs to be done by SOME/IP and not by the application itself.
<b>Use Case:</b>	Time critical transmission of data larger than 65KB
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

## 4.2 Services

**[RS\_SOMEIP\_00019] SOME/IP protocol shall identify services using unique identifiers** [

<b>Description:</b>	SOME/IP protocol shall identify services using vehicle-wide unique identifiers
<b>Rationale:</b>	It is necessary to distinguish services vehicle wide.
<b>Use Case:</b>	Discovery of a specific service.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00015] SOME/IP protocol shall support multiple instances of a service** [

<b>Description:</b>	SOME/IP protocol shall support multiple instances of a service for services which provide the same functionality.
<b>Rationale:</b>	One system design might provide one service with multiple instances, another system design might define multiple services with one instance each.





△

<b>Use Case:</b>	The same service is provided by multiple parties in the vehicle to enable fail-over-handling which enables the subscribers to switch to another service instance if the provider of one service instance suffers a failure.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00280)

**[RS\_SOMEIP\_00016] SOME/IP protocol shall support combining multiple RPC methods, events and fields in one service** [

<b>Description:</b>	SOME/IP protocol shall support combining multiple RPC methods, events and fields in one service. Functionality based on services (e.g service discovery) shall treat them together but the encapsulated RPC methods, events and fields themselves shall be usable on its own.
<b>Rationale:</b>	If functionality is often used together or other reasons for grouping exist from architectural point of view, it is not suitable to provide an own service per functionality.
<b>Use Case:</b>	Combination of functionality which is always used together.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00280)

**[RS\_SOMEIP\_00017] SOME/IP protocol shall support grouping events into event-groups** [

<b>Description:</b>	SOME/IP protocol shall support grouping events of the same provider into eventgroups to enable subscriptions to all events of a specific eventgroup in one step.
<b>Rationale:</b>	If events are provided by the same party, a grouping into an eventgroup enables more efficient service subscriptions.
<b>Use Case:</b>	A large number of related events is provided by the same party.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00280)

**[RS\_SOMEIP\_00018] SOME/IP protocol shall support grouping fields in event-groups [**

<b>Description:</b>	SOME/IP protocol shall support grouping fields of the same provider in eventgroups to enable subscriptions to all fields of a specific eventgroup in one step.
<b>Rationale:</b>	If fields are provided by the same party, a grouping into an eventgroup enables more efficient service subscriptions.
<b>Use Case:</b>	A large number of related fields is provided by the same party.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00021] SOME/IP protocol shall identify RPC methods of services using unique identifiers [**

<b>Description:</b>	SOME/IP protocol shall identify RPC methods, getters of fields and setters of fields using identifiers which are unique for that service.
<b>Rationale:</b>	Methods, getters and setters within one service needs to be distinguishable from other methods, getters or setter within the same service.
<b>Use Case:</b>	Services which consist of more than one method, getter or setter.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00022] SOME/IP protocol shall identify events of services using unique identifiers [**

<b>Description:</b>	SOME/IP protocol shall identify events and notifiers of fields using identifiers which are unique for that service.
<b>Rationale:</b>	An event within one service needs to be distinguishable from the other events within the same service.
<b>Use Case:</b>	Services which consist of more than one event.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00023] SOME/IP protocol shall identify event groups of services using unique identifiers [**

<b>Description:</b>	SOME/IP protocol shall identify event groups of services using identifiers which are unique for that service.
<b>Rationale:</b>	An event group within one service needs to be distinguishable from the other event groups within the same service.
<b>Use Case:</b>	Services which consist of more than one event group.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00024] SOME/IP protocol shall define reserved identifiers [**

<b>Description:</b>	SOME/IP protocol shall define reserved identifiers for special services. These identifiers cannot be used for normal services.
<b>Rationale:</b>	For compatibility reasons a range of reserved IDs needs to be defined. This can be used later on by OEMs without interfering with the rest of the services.
<b>Use Case:</b>	Service discovery of non-SOME/IP services
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00025] SOME/IP protocol shall support the identification of callers of an RPC using unique identifiers [**

<b>Description:</b>	SOME/IP protocol shall support the identification of callers of an RPC using identifiers which are unique for the RPC in the system.
<b>Rationale:</b>	Callers need to be identified if multiple callers on the same ECU which are using the same socket connection use the same RPC method to be able to assign the response correctly to the request.
<b>Use Case:</b>	Calling an RPC method from multiple callers which are located on one or multiple remote machines.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

### 4.3 On Wire Representation

**[RS\_SOMEIP\_00027] SOME/IP protocol shall define the header layout of messages** [

<b>Description:</b>	SOME/IP protocol shall define the header layout of messages for a respective version.
<b>Rationale:</b>	The layout of the header needs to be understood by the communication partners.
<b>Use Case:</b>	Communication between different parties.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00028] SOME/IP protocol shall specify the serialization algorithm for data** [

<b>Description:</b>	SOME/IP protocol shall specify the serialization algorithm for data to convert the machine-specific memory representation to a common network-representation.
<b>Rationale:</b>	The memory representation of data heavily depends on processor architecture and compiler settings which differ between the machine in the vehicle.
<b>Use Case:</b>	Communication of payload between communication partners using different processor architectures.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00029] SOME/IP protocol shall specify how data in the payload are aligned** [

<b>Description:</b>	SOME/IP protocol shall specify how data in the payload are aligned. This includes the start of primitive data types and gaps between them.
<b>Rationale:</b>	Some architectures allow a faster access to data which are aligned in memory. If speed matters, it is useful to be able to influence the alignment and optimize the accesses.
<b>Use Case:</b>	Fast access to serialized representation of structured data types.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–





<b>Supporting Material:</b>	–
-----------------------------	---

](RS\_Main\_00280)

**[RS\_SOMEIP\_00026] SOME/IP protocol shall define the endianness of header and payload** [

<b>Description:</b>	SOME/IP protocol shall define the endianness of data representation in header and payload. The communication partners shall use that endianness on the network.
<b>Rationale:</b>	If machines with different endianness communicate with each other, the format on the network needs to be defined.
<b>Use Case:</b>	A little-endian machine communicates with a big-endian machine.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00280)

**[RS\_SOMEIP\_00030] SOME/IP protocol shall support transporting integer data types** [

<b>Description:</b>	SOME/IP protocol shall support transporting integer data types, signed and unsigned of 8, 16 and 32 bit length.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of basic data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Communication of c-data types
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

](RS\_Main\_00280)

**[RS\_SOMEIP\_00031] SOME/IP protocol shall support transporting boolean data type** [

<b>Description:</b>	SOME/IP protocol shall support transporting boolean data type.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of basic data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Communication of boolean values.



△

<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00032] SOME/IP protocol shall support transporting float data types** [

<b>Description:</b>	SOME/IP protocol shall support transporting float data types known as <i>single</i> and <i>double</i> (defined in IEEE 754 [3] as <i>binary32</i> and <i>binary64</i> )
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of basic data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Communication of c-data types
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00033] SOME/IP protocol shall support transporting structured data types** [

<b>Description:</b>	SOME/IP protocol shall support transporting structured data types which can consist of one or multiple structured data types of fixed or variable size and/or primitive data types.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Consistent communication of more than one primitive data type.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00034] SOME/IP protocol shall support transporting union data types [**

<b>Description:</b>	SOME/IP protocol shall support transporting union data types where the concrete data type is determined during execution.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Communication of data with the possibility to choose representation dynamically.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00035] SOME/IP protocol shall support transporting one-dimensional and multi-dimensional array data types [**

<b>Description:</b>	SOME/IP protocol shall support transporting one-dimensional and multi-dimensional array data types.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Consistent communication of more than one data element of the same type.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00036] SOME/IP protocol shall support transporting array data types with a fixed length [**

<b>Description:</b>	SOME/IP protocol shall support transporting array data types with a fixed length which represent the fixed number of array elements.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Consistent communication of a specific number of data elements of the same type.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

]([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00037] SOME/IP protocol shall support transporting array data types with flexible length** [

<b>Description:</b>	SOME/IP protocol shall support transporting one-dimensional and multi-dimensional array data types with flexible number of data elements. A maximum number of data elements shall be provided.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Consistent communication of a variable number of data elements of the same type.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00038] SOME/IP protocol shall support transporting string types with a fixed length** [

<b>Description:</b>	SOME/IP protocol shall support transporting string types with a fixed length.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Communication of strings.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00039] SOME/IP protocol shall support transporting string data types with flexible length** [

<b>Description:</b>	SOME/IP protocol shall support transporting string data types with flexible length. A maximum length shall be provided.
<b>Rationale:</b>	For the specification of a serialization algorithm of SOME/IP, a set of data types needs to be supported by SOME/IP.
<b>Use Case:</b>	Communication of strings.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))



**[RS\_SOMEIP\_00040] SOME/IP protocol shall support providing the length of a serialized data element in the payload** [

<b>Description:</b>	SOME/IP protocol shall support providing the overall length of a serialized data element in the payload without the need of parsing the content of the element.
<b>Rationale:</b>	The length information can be used to determine the start of the following serialized data element without parsing data or knowing the internal structure.
<b>Use Case:</b>	Backward compatible extension of structs by adding additional elements at the end.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

**[RS\_SOMEIP\_00050]{DRAFT} SOME/IP protocol shall support serialization of extensible data structs** [

<b>Description:</b>	SOME/IP shall support serialization of structs with tagged data elements in order to support optional elements and easy extensibility of an interface definition. The tags serve as identifiers for the individual element. An additional length field allows that during deserialization unknown elements can be skipped.
<b>Rationale:</b>	Allow compact transmission of complex data structures with optional elements, allow compatible extension of existing interfaces without recompilation of existing code.
<b>Use Case:</b>	Serialization of Structured Datatypes with Identifier and optional Members.
<b>AppliesTo:</b>	FO
<b>Dependencies:</b>	–
<b>Supporting Material:</b>	–

] ([RS\\_Main\\_00280](#))

## 5 References

- [1] Standardization Template  
AUTOSAR\_FO\_TPS\_StandardizationTemplate
- [2] Glossary  
AUTOSAR\_FO\_TR\_Glossary
- [3] IEEE Standard for Floating-Point Arithmetic (IEEE Std 754-2008)

## **A Change history of AUTOSAR traceable items**

### **A.1 Traceable item history of this document according to AUTOSAR Release R23-11**

#### **A.1.1 Added Requirements in R23-11**

none

#### **A.1.2 Changed Requirements in R23-11**

none

#### **A.1.3 Deleted Requirements in R23-11**

none