

<b>Document Title</b>	AUTOSAR Glossary
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	055
<b>Document Classification</b>	Auxiliary

<b>Document Version</b>	2.3.0
<b>Document Status</b>	Final
<b>Part of Release</b>	3.2
<b>Revision</b>	2

<b>Document Change History</b>			
<b>Date</b>	<b>Version</b>	<b>Changed by</b>	<b>Change Description</b>
23.05.2012	2.3.0	AUTOSAR Administration	<ul style="list-style-type: none"> <li>Extended Abbreviation</li> <li>Rework of several descriptions</li> </ul>
27.04.2011	2.2.0	AUTOSAR Administration	Following terms added: <ul style="list-style-type: none"> <li>Callback (3.27)</li> <li>Callout (3.28)</li> </ul>
23.06.2008	2.1.3	AUTOSAR Administration	Legal disclaimer revised
05.12.2007	2.1.2	AUTOSAR Administration	Following terms added: <ul style="list-style-type: none"> <li>Debugging (3.53)</li> <li>Implementation Conformance Statement (3.115)</li> </ul> Document meta information extended Small layout adaptations made
24.01.2007	2.1.1	AUTOSAR Administration	<ul style="list-style-type: none"> <li>“Advice for users” revised</li> <li>“Revision Information” added</li> </ul>

<b>Document Change History</b>			
<b>Date</b>	<b>Version</b>	<b>Changed by</b>	<b>Change Description</b>
06.12.2006	2.1.0	AUTOSAR Administration	Following terms added: <ul style="list-style-type: none"> <li>• FlexRay (3.81)</li> <li>• Vendor ID (3.201)</li> <li>• Callback (3.27)</li> <li>• Interrupt frames (3.120)</li> <li>• Interrupt vector table(3.122)</li> <li>• Accreditation (3.1)</li> <li>• Accreditation Body (3.2)</li> <li>• Conformance Test Agency (3.44)</li> <li>• Assessment (3.8)</li> <li>• Surveillance (3.188)</li> <li>• Attestation (3.11)</li> <li>• (Conformance) Declaration (3.54)</li> <li>• First party and (3.78)</li> <li>• Third party (3.196)</li> <li>• Safety (3.164)</li> <li>• ECU Configuration (3.58)</li> <li>• ECU Configuration Description (3.59)</li> <li>• Legal disclaimer revised</li> </ul>
28.04.2006	2.0.0	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• removed and added some terms</li> <li>• rework of several descriptions and some formal changes</li> </ul>
31.05.2005	1.0.0	AUTOSAR Administration	Initial Release

## Disclaimer

This specification 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 specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification 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 specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

## Advice for users

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

## Table of Contents

Abbreviations.....	9
1 Introduction.....	11
2 How to read this document.....	12
The title of the subchapters is identical to the term to be defined.....	12
2.1 <Definition> .....	12
3 Definitions.....	13
3.1 Accreditation .....	13
3.2 Accreditation Body .....	13
3.3 Application.....	13
3.4 Application Programming Interface (API) .....	14
3.5 Application Software Component .....	14
3.6 Architecture .....	14
3.7 Asserted Property.....	15
3.8 Assessment.....	15
3.9 Asset .....	15
3.10 Asynchronous Communication.....	16
3.11 Attestation .....	16
3.12 Atomic Software Component.....	16
3.13 AUTOSAR Authoring Tool.....	17
3.14 AUTOSAR Interface .....	17
3.15 AUTOSAR Metamodel .....	17
3.16 AUTOSAR Model .....	18
3.17 AUTOSAR Service .....	18
3.18 AUTOSAR Software-Component (SW-C) .....	18
3.19 AUTOSAR Virtual Functional Bus (VFB).....	18
3.20 AUTOSAR XML description .....	19
3.21 AUTOSAR XML Schema .....	19
3.22 Availability .....	20
3.23 Basic Software (BSW).....	20
3.24 Basic Software Module.....	20
3.25 Bulk Data.....	21
3.26 Calibration .....	21
3.27 Callback .....	21
3.28 Callout.....	22
3.29 Call Point.....	22
3.30 Causality of Transmission .....	22
3.31 Client .....	23
3.32 Client-Server Communication .....	23
3.33 Client-Server Interface .....	23
3.34 Cluster Signal.....	24
3.35 Code Variant Coding.....	24

3.36	Communication Attribute .....	24
3.37	Complex Device Driver (CDD).....	25
3.38	Component.....	26
3.39	Composition .....	26
3.40	Compositionality .....	26
3.41	Conditioned Signal .....	27
3.42	Configuration .....	27
3.43	Confirmation.....	27
3.44	Conformance Test Agency (CTA) .....	28
3.45	Conformance Test Suite (CTS) .....	28
3.46	Connector.....	28
3.47	Control Flow .....	29
3.48	Data.....	29
3.49	Data Element .....	29
3.50	Data Flow .....	30
3.51	Data Variant Coding .....	30
3.52	Deadline .....	30
3.53	Debugging.....	31
3.54	(Conformance) Declaration .....	31
3.55	Dependability .....	31
3.56	Dynamic Routing.....	32
3.57	ECU Abstraction Layer.....	32
3.58	ECU Configuration .....	32
3.59	ECU Configuration Description.....	33
3.60	ECU Firmware.....	33
3.61	g Electronic Control Unit (ECU).....	33
3.62	Electrical Signal.....	34
3.63	Entry Point.....	34
3.64	Error .....	34
3.65	Event.....	35
3.66	Execution Time.....	35
3.67	Exit Point .....	35
3.68	Fail-degraded .....	36
3.69	Fail-operational.....	36
3.70	Fail-safe .....	36
3.71	Fail-silent.....	37
3.72	Failure .....	37
3.73	Fault .....	37
3.74	Fault Detection .....	38
3.75	Fault Reaction .....	38
3.76	Fault Tolerance .....	38
3.77	Feature.....	39
3.78	First party .....	39
3.79	Flag .....	39
3.80	FlexRay Base Cycle .....	40
3.81	FlexRay Bus.....	40
3.82	FlexRay Cell.....	40
3.83	FlexRay Channel.....	41
3.84	FlexRay Cluster.....	41
3.85	FlexRay Cycle .....	42

3.86	FlexRay Cycle Number .....	42
3.87	FlexRay Cycle Offset.....	42
3.88	FlexRay Cycle Repetition .....	43
3.89	FlexRay Frame.....	43
3.90	FlexRay Global Time.....	43
3.91	FlexRay L-PDU .....	44
3.92	FlexRay L-PDU-Identifier .....	44
3.93	FlexRay L-SDU-Identifier .....	45
3.94	FlexRay Matrix .....	46
3.95	FlexRay Network.....	46
3.96	FlexRay Node .....	47
3.97	FlexRay Physical Communication Link .....	47
3.98	FlexRay Slot.....	47
3.99	FlexRay Slot Multiplexing .....	48
3.100	FlexRay Slot Number.....	49
3.101	FlexRay Star .....	49
3.102	Frame .....	49
3.103	Frame PDU.....	50
3.104	Function.....	50
3.105	Functional Network.....	50
3.106	Functional Unit.....	51
3.107	Functionality.....	51
3.108	Gateway .....	51
3.109	Gateway ECU.....	52
3.110	Hardware Connection .....	52
3.111	Hardware Element.....	52
3.112	Hardware Interrupt.....	53
3.113	Hardware Port.....	53
3.114	I-PDU.....	53
3.115	Implementation Conformance Statement .....	54
3.116	Indication .....	55
3.117	Integration.....	55
3.118	Interface.....	55
3.119	Interrupt .....	56
3.120	Interrupt frames .....	56
3.121	Interrupt Service Routine (ISR).....	56
3.122	Interrupt Vector Table .....	57
3.123	Invalid Flag .....	57
3.124	Invalid Value of Signal .....	57
3.125	Link time configuration.....	58
3.126	Mapping.....	58
3.127	MCAL Signal.....	58
3.128	Metadata.....	59
3.129	Microcontroller Abstraction Layer (MCAL) .....	59
3.130	Mistake .....	59
3.131	Multimedia Stream.....	60
3.132	Multiple Configuration Sets.....	60
3.133	Multiplexed PDU.....	61
3.134	Non-AUTOSAR Component .....	61
3.135	Notification.....	61

3.136	OS-Application.....	62
3.137	Partitioning.....	62
3.138	PCI.....	62
3.139	PDU.....	63
3.140	PDU Timeout.....	64
3.141	Peripheral Hardware.....	64
3.142	Personalization.....	64
3.143	Port.....	65
3.144	Port Interface.....	65
3.145	Post-build time configuration.....	65
3.146	Pre-Compile time configuration.....	66
3.147	Private Interface (API 3).....	66
3.148	Probability of failure.....	66
3.149	Procedure Call.....	67
3.150	Process.....	67
3.151	Provide Port.....	67
3.152	Redundancy.....	68
3.153	Reliability.....	68
3.154	Relocatability.....	68
3.155	Require Port.....	69
3.156	Required property.....	69
3.157	Resource.....	69
3.158	Resource-Management.....	70
3.159	Response Time.....	70
3.160	Risk.....	70
3.161	Robustness.....	71
3.162	RTE Event.....	71
3.163	Runnable Entity.....	71
3.164	Safety.....	72
3.165	Sample Application.....	72
3.166	Scalability.....	72
3.167	Scheduler.....	72
3.168	SDU.....	73
3.169	Security.....	73
3.170	Sender-Receiver Communication.....	73
3.171	Sender-Receiver Interface.....	74
3.172	Sensor/Actuator SW-Component.....	74
3.173	Server.....	74
3.174	Service.....	75
3.175	Service Port.....	75
3.176	Services Layer.....	76
3.177	Shipping.....	76
3.178	Software Configuration.....	76
3.179	Software Interrupt.....	77
3.180	Software Module.....	77
3.181	Software Signal.....	77
3.182	Special Periphery Access.....	77
3.183	Standard Periphery Access.....	78
3.184	Standardized AUTOSAR Interface.....	78
3.185	Standardized Interface.....	78

3.186	Standard Software .....	79
3.187	Static Configuration .....	79
3.188	Surveillance .....	79
3.189	Synchronous Communication .....	80
3.190	System.....	80
3.191	System Constraint .....	80
3.192	System Signal.....	81
3.193	Task.....	81
3.194	Technical Signal .....	81
3.195	Template.....	82
3.196	Third party .....	82
3.197	Timeout.....	82
3.198	Use Case .....	82
3.199	Validation .....	83
3.200	Variant Coding.....	83
3.201	Vendor ID .....	83
3.202	Verification .....	84
3.203	VFB View.....	84
3.204	Virtual Integration.....	84
3.205	Worst Case Execution Time .....	85
3.206	Worst Case Response Time .....	85
Annex 1: Literature .....		86



## Abbreviations

<b>Abbreviation</b>	<b>Description</b>
<b>ADC</b>	Analog Digital Converter
<b>API</b>	Application Programming Interface
<b>ASAM</b>	Association for Standardization of Automation and Measuring Systems
<b>AUTOSAR</b>	Automotive Open System Architecture
<b>BSW</b>	Basic SoftWare
<b>BswM</b>	Basic Software Mode Manager
<b>BSWMD</b>	Basic SoftWare Module Description
<b>CAN</b>	Controller Area Network
<b>CDD</b>	Complex Device Driver
<b>COM</b>	Communication
<b>CPU</b>	Central Processing Unit
<b>CTA</b>	Conformance Test Agency
<b>CTS</b>	Conformance Test Suite
<b>DAC</b>	Digital to Analog Converter
<b>DIO</b>	Digital Input/Output
<b>DTD</b>	Document Type Definition
<b>ECU</b>	Electronic Control Unit
<b>FIFO</b>	First In First Out
<b>FPU</b>	Floating Point Unit
<b>FW</b>	Fire Wire
<b>GSM</b>	Global System for Mobile Communication
<b>HIS</b>	Hersteller Initiative Software
<b>HW</b>	Hardware
<b>IEC</b>	International Electrotechnical Commission
<b>I-PDU</b>	Interaction Layer Protocol Data Unit
<b>ISR</b>	Interrupt Service Routine
<b>LIFO</b>	Last In First Out
<b>LIN</b>	Local Interconnected network
<b>LSB</b>	Least Significant Bit
<b>μC</b>	MicroController
<b>MCAL</b>	MicroController Abstraction Layer

<b>MIPS</b>	Million Instructions Per Second
<b>MMU</b>	Memory Management Unit
<b>MMI</b>	Man Machine Interface
<b>MOST</b>	Media Oriented Systems Transport
<b>μP</b>	MicroProcessor
<b>MPU</b>	Memory Protection Unit
<b>MSB</b>	Most Significant Bit
<b>NVRAM</b>	Non-Volatile Random Access Memory
<b>OEM</b>	Original Equipment Manufacturer
<b>OIL</b>	OSEK Implementation Language
<b>OS</b>	Operating System
<b>OSEK</b>	Open Systems and the Corresponding Interfaces for Automotive Electronics
<b>PCI</b>	Protocol Control Information
<b>PDU</b>	Protocol Data Unit
<b>PS</b>	Product Supplier
<b>PWM</b>	Pulse Width Modulation
<b>RfC</b>	Request for Change
<b>RTE</b>	Runtime Environment
<b>SDU</b>	Service Data Unit
<b>SIL</b>	Safety Integrity Level
<b>SW</b>	Software
<b>SW-C</b>	Software Component
<b>TTP</b>	Time Triggered Protocol
<b>USB</b>	Universal Serial Bus
<b>VFB</b>	Virtual Functional Bus
<b>WCET</b>	Worst Case Execution Time
<b>WCRT</b>	Worst Case Response time
<b>XML</b>	Extensible Markup Language

## 1 Introduction

This document is the overall glossary of AUTOSAR. It contains definitions of all major terms and notions used within AUTOSAR. It does not claim to be complete and please keep in mind that some WPs have more specific terms defined within their domain specific glossary.

The document is written in English, but to guarantee consistency in all languages AUTOSAR aims for translating all terms to different languages. Currently the translation has partly been done in the following languages:

- German
- French

In future more languages might follow.

## 2 How to read this document

The title of the subchapters is identical to the term to be defined.

### 2.1 <Definition>

<b>Definition</b>	<i>tbd - term to be defined</i>
<b>Initiator</b>	<i>tbd – person and/or group who introduced the term</i>
<b>Further Explanations</b>	<i>tbd – further explanation of the definition</i>
<b>Comment</b>	<i>tbd – comment or hints</i>
<b>Example</b>	<i>tbd – example of the term</i>
<b>Reference</b>	<i>tbd – reference of definition</i>
<b>German Term</b>	<i>tbd – German translation of the term</i>
<b>French Term</b>	<i>tbd – French translation of the term</i>

## 3 Definitions

### 3.1 Accreditation

<b>Definition</b>	Third-party (→ definition 0) attestation to an organization conveying formal demonstration of its competence to carry out AUTOSAR conformance test tasks.
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	Within the AUTOSAR environment two different scopes of Accreditation exist: - In reference to ISO/IEC 17025 for accreditation of organizations performing conformance testing within own laboratories. - In reference to ISO/IEC Guide 65 for accreditation of Conformance Test Agencies (CTA) (→ definition 3.44) performing 3rd party product conformance attestation.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[ISO/IEC 17000]
<b>German Term</b>	Akkreditierung
<b>French Term</b>	Accréditation

### 3.2 Accreditation Body

<b>Definition</b>	This is an organization, which employs or contracts individual assessors, who in turn perform the independent assessments (→ definition 3.8) as part of the accreditation (→ definition 3.1) process.
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	In the AUTOSAR context the Accreditation Bodies have the role to accredit the Conformance Test Agencies.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Akkreditierungskörperschaft
<b>French Term</b>	--

### 3.3 Application

<b>Definition</b>	A software (or program) that is specified to the solution of a problem of an end user requiring information processing for its solution. The software configuration (→ definition 3.178) of a software entity.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	To 1. of Definition: In AUTOSAR Application software is located above the AUTOSAR RTE (RunTimeEnvironment).
<b>Comment</b>	Definition 1 is the “by default” meaning for application in AUTOSAR. When definition 2 is meant, it has to be explicitly mentioned.
<b>Example</b>	--
<b>Reference</b>	[ISO 2382-20]
<b>German Term</b>	Applikation
<b>French Term</b>	--

### 3.4 Application Programming Interface (API)

<b>Definition</b>	An Application Programming Interface (API) is the prescribed method of a specific software part by which a programmer writing a program can make requests to that software part.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	OSEK OS API
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.5 Application Software Component

<b>Definition</b>	An Application Software Component is a specific AUTOSAR Software Component (→ definition 3.18) realizing a defined functionality of a set (one or more) of features.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Applikations Software Komponente
<b>French Term</b>	--

### 3.6 Architecture

<b>Definition</b>	The fundamental organization of a system embodied in its components, their static and dynamic relationships to each other, and to the environment, and the principles guiding its design and evolution.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	“Static and dynamic” added to EAST definition.
<b>Example</b>	--
<b>Reference</b>	[IEEE 1471], [EAST-Glossary]
<b>German Term</b>	Architektur
<b>French Term</b>	--

### 3.7 Asserted Property

<b>Definition</b>	A property or quality of a design entity (e.g. SW component or system) is asserted, if the design entity guarantees that this property or quality is fulfilled.
<b>Initiator</b>	WP10.1
<b>Further Explanations</b>	A property or quality of a design unit can be asserted by the design unit itself or in combination with another design unit.
<b>Comment</b>	--
<b>Example</b>	If the worst case execution time of a task (w.r.t. a certain CPU etc.) is asserted to be 3 ms, the execution time of this task will under any circumstances be less than or equal to 3 ms.
<b>Reference</b>	Compare required property (→ definition 3.156)
<b>German Term</b>	Zugesichert
<b>French Term</b>	--

### 3.8 Assessment

<b>Definition</b>	Activity conducted by an assessor to demonstrate that the assessed organization fulfills the set requirements.
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	Within the AUTOSAR context after the initial assessment a new full-set assessment will be repeated every 4 years. In between, surveillances (→ definition 3.188) take place once a year.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[ISO/IEC 17011]
<b>German Term</b>	Begutachtung
<b>French Term</b>	Evaluation

### 3.9 Asset

<b>Definition</b>	An item that has been designed for use in multiple contexts.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	An asset can be design, specifications, source code, documentation, test suits, manual procedures, etc..
<b>Reference</b>	[IEEE 1517], [EAST-Glossary]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.10 Asynchronous Communication

<b>Definition</b>	Asynchronous communication does not block the sending software entity. The sending software entity continues its operation without getting a response from the communication partner(s).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	There could be an acknowledgement by the communication system about the sending of the information. A later response to the sending software entity is possible.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.11 Attestation

<b>Definition</b>	Issue of a statement, based on a decision following a review, that fulfillment of specified requirements has been demonstrated.
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	The resulting statement, referred to in the International Standard as a "statement of conformity", conveys the assurance that the specified requirements have been fulfilled. Such an assurance does not, of itself, afford contractual or other legal guarantees.
<b>Comment</b>	--
<b>Example</b>	Attestation of product conformance given by a third party like a CTA (→ definition 3.44). Attestation of product conformance given by a first party like the Product Supplier (also called declaration).
<b>Reference</b>	[ISO/IEC 17000]
<b>German Term</b>	Bestätigung
<b>French Term</b>	Attestation

### 3.12 Atomic Software Component

<b>Definition</b>	Smallest non-dividable software entity connected to the AUTOSAR Virtual Functional Bus (→ definition 3.19).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	An Atomic Software Component might access HW or not, therefore not all Atomic SW-Cs are relocatable.
<b>Comment</b>	--
<b>Example</b>	AUTOSAR Software Component, Complex Device Driver
<b>Reference</b>	--
<b>German Term</b>	Atomare Software Komponente
<b>French Term</b>	--



### 3.13 AUTOSAR Authoring Tool

<b>Definition</b>	An AUTOSAR authoring tool is a software tool which supports interpreting, processing and creating of AUTOSAR XML descriptions (→ definition 3.20).
<b>Initiator</b>	WP1.2
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.14 AUTOSAR Interface

<b>Definition</b>	The AUTOSAR Interface of a component (→ definition 3.38) refers to the collection of all ports (→ definition 3.143) of that component through which it interacts with other components.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	Note that an AUTOSAR Interface is different from a Port Interface (→ definition 3.144). The latter characterizes one specific port of a component.
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus], Chapter “ Modeling of Communication, Graphical Notation”
<b>German Term</b>	--
<b>French Term</b>	--

### 3.15 AUTOSAR Metamodel

<b>Definition</b>	The AUTOSAR metamodel is a UML2.0 model that defines the language for describing AUTOSAR systems and related artifacts.
<b>Initiator</b>	MMT
<b>Further Explanations</b>	The AUTOSAR metamodel is a graphical representation of a template (→ definition 3.195). UML2.0 class diagrams are used to describe the attributes and their interrelationships. Stereotypes and OCL (object constraint language) are used for defining specific semantics and constraints.
<b>Comment</b>	The AUTOSAR XML Schema (→ definition 3.21) is derived from the AUTOSAR metamodel.
<b>Example</b>	--
<b>Reference</b>	[UML 2.0]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.16 AUTOSAR Model

<b>Definition</b>	An AUTOSAR model is an instance of the AUTOSAR metamodel (→ definition 3.15). The information contained in the AUTOSAR model (→ definition 3.16) can be anything that is representable according to the AUTOSAR metamodel. The AUTOSAR model can be stored in many different ways: it might be a set of files in a file system, an XML stream, a database or memory used by some running software tools, etc.
<b>Initiator</b>	--
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.17 AUTOSAR Service

<b>Definition</b>	An AUTOSAR Service is a logical entity of the basic software (→ definition 3.23) offering general functionality to be used by various AUTOSAR software components. The functionality is accessed via standardized AUTOSAR Interfaces (→ definition 3.14).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	Parts of the basic software required to realize AUTOSAR communication patterns and communication attributes are not called AUTOSAR services.
<b>Example</b>	Error memory for diagnosis. Timer service. ECU state manager.
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.18 AUTOSAR Software-Component (SW-C)

<b>Definition</b>	The AUTOSAR Software Components encapsulate an application which runs on the AUTOSAR infrastructure. The AUTOSAR Software Components have well-defined interfaces, which are described and standardized within AUTOSAR.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	An AUTOSAR Software Component has a formal description (→ definition 3.18), i.e. software component template (→ definition 3.195). AUTOSAR Software-Components are located "above" the AUTOSAR Runtime Environment.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	AUTOSAR Software Komponente
<b>French Term</b>	--

### 3.19 AUTOSAR Virtual Functional Bus (VFB)

<b>Definition</b>	The AUTOSAR Virtual Functional Bus is an abstraction of the communication between Atomic Software Components (→ definition 3.11) and AUTOSAR Services (→ definition 3.17). This abstraction is such that specification of the communication mechanisms is independent from the concrete technology chosen to realize the communication.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	After compilation and linking of software for a dedicated ECU (→ definition 3.61) the AUTOSAR Virtual Functional Bus interfaces are realized by the AUTOSAR Runtime Environment.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	AUTOSAR Virtueller Funktionsbus
<b>French Term</b>	--

### 3.20 AUTOSAR XML description

<b>Definition</b>	An AUTOSAR XML description describes the XML representation of an AUTOSAR model (→ definition 3.16). The AUTOSAR XML description can consist of several fragments (e.g. files). Each individual fragment must validate successfully against the AUTOSAR XML schema.
<b>Initiator</b>	--
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.21 AUTOSAR XML Schema

<b>Definition</b>	The AUTOSAR XML Schema is an XML language definition for exchanging AUTOSAR models (→ definition 3.16) and descriptions.
<b>Initiator</b>	WP1.2
<b>Further Explanations</b>	The AUTOSAR XML Schema is a W3C XML schema that defines the language for exchanging AUTOSAR models. This Schema is derived from the AUTOSAR metamodel (→ definition 3.15). The AUTOSAR XML Schema defines the AUTOSAR data exchange format.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.22 Availability

<b>Definition</b>	1. Probability that a system or functional unit is able to perform its normal operation under specified conditions at a specific time.  2. The property of data or resources being accessible and usable on demand by an authorized entity.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	The time can be used to model a decrease of the availability over time due to e.g. aging of components.
<b>Comment</b>	1. Degraded modes are covered by this definition (see example)
<b>Example</b>	1. Power Steering: if the support function fails it is not available while the steering as a base function has full availability.
<b>Reference</b>	based on [ISO 2382-14], [ISO 2382-8], Reliability (→ definition 3.153)
<b>German Term</b>	Verfügbarkeit
<b>French Term</b>	--

### 3.23 Basic Software (BSW)

<b>Definition</b>	The Basic Software provides the infrastructural (schematic dependent and schematic independent) functionalities of an ECU (→ definition 3.61). It consists of ECU Firmware (→ definition 3.60) and Standard Software (→ definition 3.186).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	MCAL, AUTOSAR services, communication layer
<b>Reference</b>	--
<b>German Term</b>	Basis-Software
<b>French Term</b>	--

### 3.24 Basic Software Module

<b>Definition</b>	A collection of software files (code and description) that define a certain basic software functionality present on an ECU.
<b>Initiator</b>	WP 4.1.1.2
<b>Further Explanations</b>	Standard software (→ definition 3.186) may be composed of several software modules (→ definition 3.180) that are developed independently. A software module may consist of ECU firmware (→ definition 3.60), and/or standard software (→ definition 3.186).
<b>Comment</b>	--
<b>Example</b>	A Digital IO Driver, Complex Device Driver, OS are examples of basic software modules.
<b>Reference</b>	--
<b>German Term</b>	Basis-Software-Modul
<b>French Term</b>	--

### 3.25 Bulk Data

<b>Definition</b>	“Bulk Data” is a set of data such big in size, that standard mechanisms used to handle smaller data sets become inconvenient. This implies that bulk data in a software system are modeled, stored, accessed and transported by different mechanisms than smaller data sets.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Bulk data are typically handled by adding a level of abstraction (e.g. files) which separates the containment of the data from the internal structure.
<b>Comment</b>	The critical size, above which data must be regarded as bulk data depends on the technical infrastructure (e.g. bus system) and the considered use case (transport, storage etc.).
<b>Example</b>	Data on a persistent medium which has a capacity of a few kBytes (e.g. EEPROM) can be directly accessed via memory addresses, address offsets can be mapped to symbols of a programming language: No bulk data mechanisms are needed. For media with bigger capacity this becomes inconvenient or even impossible, so that a file system is used: The data are treated as bulk data.
<b>Reference</b>	--
<b>German Term</b>	Massendaten
<b>French Term</b>	Masse de Données

### 3.26 Calibration

<b>Definition</b>	Calibration is the adjustment of parameters of SW-Components realizing the control functionality (namely parameters of AUTOSAR SWCs, ECU abstraction or Complex Device Drivers).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Only those software modules can be calibrated, which are above RTE and ECU Abstraction and CDD. Calibration is always done at post-build time. Used techniques to set calibration data include end-of-line programming, garage programming and adaptive calibration (e.g. in the case of anti-pinch protection for power window).
<b>Comment</b>	--
<b>Example</b>	The calibration of the engine control will take into account the production differences of the individual motor this system will control.
<b>Reference</b>	--
<b>German Term</b>	Kalibrierung
<b>French Term</b>	--

### 3.27 Callback

<b>Definition</b>	Functionality that is defined by an AUTOSAR module so that lower-level modules (i.e. lower in the Layered Software Architecture) can provide notification as required (e.g. when certain events occur or asynchronous processing completes).
<b>Initiator</b>	WP-1.1.1
<b>Further Explanations</b>	In AUTOSAR, modules usually provide a register mechanism for callback functions which is set through configuration.  A module provides callbacks so that other modules can initiate its processing while the module calls callouts (--> definition 3.xx) to execute functionality that could not be specified by AUTOSAR, i.e. integration code (--> 3.125)
<b>Comment</b>	--

<b>Example</b>	<p>(from the viewpoint of a particular SWS): The module being specified (Msws) should be informed about an event (--&gt; definition 3.68) in another module (Mexternal). In this example, Msws calls Mexternal to perform some processing and can only resume when Mexternal completes. Upon completion, Mexternal calls Msws's callback function. That is, the called module (Mexternal) CALLS the calling module (Msws) BACK when complete ==&gt; a callback.</p> <p>(from AUTOSAR): COM callback (OSEK-BD)</p>
<b>Reference</b>	--
<b>German term</b>	--
<b>French term</b>	--

### 3.28 Callout

<b>Definition</b>	Function stubs that the system designer can replace with code to add functionality to a module which could not be specified by AUTOSAR.
<b>Initiator</b>	WP-1.1.1 MMS
<b>Further Explanations</b>	<p>A module calls callouts (--&gt; definition 3.xx) to execute functionality that could not be specified by AUTOSAR, i.e. integration code (--&gt; 3.125) while the module provides callbacks so that other modules can initiate its processing.</p> <p>Callouts can be separated into two classes: 1) callouts that provide mandatory functionality and thus serve as a hardware abstraction layer 2) callouts that provide optional functionality</p>
<b>Comment</b>	--
<b>Example</b>	<p>In the EcuM: For class 1): EcuM_EnableWakeupSources For class 2): The Init Lists (EcuM_AL_DriverInitZero)</p>
<b>Reference</b>	--
<b>German term</b>	--
<b>French term</b>	--

### 3.29 Call Point

<b>Definition</b>	A point in a AUTOSAR Software-Component (→ definition 3.18) where the SW-C enforce an execution entity (Entry point → definition 3.63) in an other SW-C.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	<p>Request Service Send Information</p>
<b>Reference</b>	--
<b>German Term</b>	Aufrufpunkt
<b>French Term</b>	--

### 3.30 Causality of Transmission

<b>Definition</b>	Transmit order of PDUs with the same identifier (instances of PDUs) from a source network is preserved in the destination network.
<b>Initiator</b>	WP4.2.2.1.6 Gateway
<b>Further Explanations</b>	Transmission of PDUs (→ definition 3.139) with the same identifier has a particular temporal order in a given source network. After routing over a gateway the temporal order of transmission of PDUs in a destination network may be changed. Only in case that the temporal order is the same, causality is given. Otherwise causality is violated. Causality can be in contradiction to prioritization of PDUs.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Einhaltung der Übertragungsreihenfolge bei Botschaften mit demselben Identifier
<b>French Term</b>	--

### 3.31 Client

<b>Definition</b>	Software entity which uses services of a server (→ definition 3.173).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The client and the server might be located on one ECU (→ definition 3.61) or distributed on different calculation units (e.g. ECU, external diagnostic tester).
<b>Comment</b>	Adapted from Balzert.
<b>Example</b>	--
<b>Reference</b>	[Balzert99]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.32 Client-Server Communication

<b>Definition</b>	A specific form of communication in a possibly distributed system in which software entities act as clients (→ definition 3.31), servers (→ definition 3.173) or both, where 1...n clients are requesting services via a specific protocol from typically one server.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Client-server communication can be realized by synchronous or asynchronous communication. <ul style="list-style-type: none"> <li>• Client takes initiative: requesting that the server performs a service, e.g. client triggers action within server (server does not start action on its own)</li> <li>• Client is after service request blocked / non-blocked</li> <li>• Client expects response from server: data flow (+ control flow, if blocked)</li> </ul> One example for 1 client to n server communication (currently not supported) is a functional request by diagnosis. This has to be treated as a specific exception.
<b>Comment</b>	Adapted from Hyper Dictionary
<b>Example</b>	Internet (TCP/IP)
<b>Reference</b>	[Hyper Dictionary]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.33 Client-Server Interface

<b>Definition</b>	The client-server interface is a special kind of port-interface (→ definition 3.144) used for the case of client-server communication (→ definition 3.32). The client-server interface defines the operations that are provided by the server (→ definition 3.173) and that can be used by the client (→ definition 3.31).
<b>Initiator</b>	SystemTeam
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.34 Cluster Signal

<b>Definition</b>	A cluster signal represents the aggregating system signal on one specific communication cluster. Cluster signals can be defined independently of frames. This allows a development methodology where the signals are defined first, and are assigned to frames in a later stage.
<b>Initiator</b>	WP 2.1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.35 Code Variant Coding

<b>Definition</b>	Adaptation of SW by selection of functional alternatives according to external requirements
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Code Variant Coding might influences RTE (RuntimeEnvironment) and BSW modules (→ definition 3.24), not only the application software modules. Code Variant Coding is always done at pre-compile time or at link time. Code Variant Coding also includes vehicle-specific (not user-specific) SW adaptation due to end-customer wishes (e.g. deactivation of speed dependent automatic locking).
<b>Comment</b>	In case of the C language the #if or #ifdef directive can be used for creating code variants. Code Variant Coding is a design time concept.
<b>Example</b>	The same window lifter ECU is used for cars with 2 and 4 doors, however different code segments have to be used in both cases.
<b>Reference</b>	[AUTOSAR_WP2.1.1.1_SW-C-Attributes]
<b>German Term</b>	Variantenkodierung
<b>French Term</b>	--

### 3.36 Communication Attribute

<b>Definition</b>	Communication attributes define, according to the development phase, behavioral as well as implementation aspects of the AUTOSAR communication patterns.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The exact characteristics of the communication patterns provided by AUTOSAR (client-server and sender-receiver) can be specified more precisely by communication attributes.



<b>Comment</b>	See chapter 4.1.6 in Specification of the Virtual Functional Bus
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	Kommunikationsattribut
<b>French Term</b>	Attribut de communication

### 3.37 Complex Device Driver (CDD)

<b>Definition</b>	An Atomic Software Component (→ definition 3.11) that on one side interfaces with the AUTOSAR VFB (→ definition 3.19) and on the other side directly accesses Peripheral Hardware (→ definition 3.141) and/or ECU-Abstraction (→ definition 3.57) and/or MCAL (→ definition 3.129). Complex Device Driver can be accessed via AUTOSAR Interfaces and/or directly by Basic Software Modules.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	SW situated below the AUTOSAR RTE. This software in general is not relocatable.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.38 Component

<b>Definition</b>	A Component encapsulates a complete (automotive) functionality or a piece of it. It is described on the basis of a formal specification, that allows the seamless integration within the VFB (→ definition 3.19). A Component has well defined Ports (→ definition 3.143), through which the Component can interact with other Components.
<b>Initiator</b>	Bertrand Delord
<b>Further Explanations</b>	When modeling a system with AUTOSAR, a logical interconnection of components can be packaged as a component. Such a component is called a “composition”. In contrast to the Atomic Software Components (→ definition 3.11), the components inside a composition can be distributed over several ECUs.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Technical Overview]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.39 Composition

<b>Definition</b>	An AUTOSAR Composition encapsulates a collaboration of Components (→ definition 3.38), thereby hiding detail and allowing the creation of higher abstraction levels. Through Delegation Connectors (→ definition 3.46) a Composition (→ definition 3.39) explicitly specifies, which Ports (→ definition 3.143) of the internal components are visible from the outside. AUTOSAR Compositions are a type of Components, e.g. they can be part of further compositions.
<b>Initiator</b>	WP 2.1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	See Virtual Functional Bus Specification, Chapter “VFB View, Meta-Model
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	Komposition
<b>French Term</b>	--

### 3.40 Compositionality

<b>Definition</b>	Compositionality is given when the behavior of a software component or subsystem of a system is independent of the overall system load and configuration.
<b>Initiator</b>	BMW
<b>Further Explanations</b>	Compositionality is an important property of deterministic systems. This property leads to a complete decoupling of systems. Smooth subsystem integration without backlashes is then easily achievable.
<b>Comment</b>	--
<b>Example</b>	A new component or a subsystem can be added to a system without changing the behavior of the original components.
<b>Reference</b>	--
<b>German Term</b>	Zusammensetzbarkeit
<b>French Term</b>	--

### 3.41 Conditioned Signal

<b>Definition</b>	The conditioned signal is the internal electrical representation of the electrical signal within the ECU. It is delivered to the processor and represented in voltage and time (or, in case of logical signals, by high or low level).
<b>Initiator</b>	--
<b>Further Explanations</b>	The Electrical Signal (→ definition 3.62) usually can not be processed by the peripherals directly, but has to be adopted. This includes amplification and limitation, conversion from a current into a voltage and so on. This conversion is performed by some electrical devices in the ECU and the result of the conversion is called the Conditioned Signal. The description means for the Conditioned Signal can also be the same as for Technical Signals (→ definition 3.194) and Electrical Signals, but limited to electrical voltage
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.42 Configuration

<b>Definition</b>	The arrangement of hardware and/or software elements in a system.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A configuration in general takes place before runtime.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AST-Glossary], [ISO 61511-1]
<b>German Term</b>	Konfiguration
<b>French Term</b>	--

### 3.43 Confirmation

<b>Definition</b>	Service primitive defined in the ISO/OSI Reference model (ISO 7498). With the 'confirmation' service primitive a service provider informs a service user about the result of a preceding service request of the service user [OSEK BD]
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A confirmation is e.g. a specific notification generated by the OSEK underlying layer to inform about a Message Transmission Error.
<b>Comment</b>	--
<b>Example</b>	OSEK Com notification class 2 and 4
<b>Reference</b>	[SEK BD], [OSEK Com]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.44 Conformance Test Agency (CTA)

<b>Definition</b>	This is an organization performing a role with specific tasks in the conformance test process. Two possibilities are distinguished, a third party (→ definition 0) CTA or a first party (→ definition 3.78) CTA (which is also a product supplier).
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	The tasks/services provided by a CTA is: - Provision of a CTS - Test execution - Attestation of product conformance  For a third party CTA the focus is on the attestation and for a first party the focus is on the test execution.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.45 Conformance Test Suite (CTS)

<b>Definition</b>	is a test implementation used in the context of Conformance Testing. Typically, multiple test implementations from different vendors (e.g. CTA) will exist, each of which implements the standardized Conformance Test specifications.
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	ISO 9646 distinguishes between Abstract Test Suites and Executable Test Suites. For AUTOSAR the earlier relates to the Conformance Test Specifications, whereas the latter to the test implementations or Conformance Test Suites.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[ISO 9646, Parts 1,2 and 4]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.46 Connector

<b>Definition</b>	A connector connects ports (→ definition 3.143) of components (→ definition 3.38) and represents the flow of information between those ports.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	For more information see AUTOSAR Specification of VFB
<b>Example</b>	AssemblyConnector, DelegationConnector
<b>Reference</b>	[AUTOSAR Specification of Virtual Function Bus]
<b>German Term</b>	Konnektor
<b>French Term</b>	--

### 3.47 Control Flow

<b>Definition</b>	The directed transmission of information between multiple entities, directly resulting in a state change of the receiving entity.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A state change could result in an activation of a schedulable entity.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Kontrollfluss
<b>French Term</b>	--

### 3.48 Data

<b>Definition</b>	A reinterpretable representation of information in a formalized manner suitable for communication, interpretation or processing.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Flag, Notification, etc.
<b>Reference</b>	[ISO 2382-1]
<b>German Term</b>	Daten
<b>French Term</b>	--

### 3.49 Data Element

<b>Definition</b>	Data elements are declared within the context of a "Sender-Receiver Interface" (→ definition 3.171). They serve as the data units that are exchanged between sender and receiver.
<b>Initiator</b>	Stefaan Sonck Thiebaut
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR SoftwareComponentTemplate]
<b>German Term</b>	Datenelement
<b>French Term</b>	--

### 3.50 Data Flow

<b>Definition</b>	The directed transmission of data (→ definition 3.48) between multiple entities. The transmitted data are not directly related to a state change at the receiver side.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Asynchronous communication.
<b>Reference</b>	--
<b>German Term</b>	Datenfluss
<b>French Term</b>	--

### 3.51 Data Variant Coding

<b>Definition</b>	Adaptation of SW by setup of certain characteristic data according to external requirements.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Data Variant Coding might influence RTE (RunTimeEnvironment) and BSW modules (→ definition 3.24) not only the application software modules. (Multiple configuration parameter sets are needed.) Variant Coding is always done at post-build time. Variant Coding also includes vehicle-specific (not user-specific) SW adaptation due to end-customer wishes (e.g. deactivation of speed dependent automatic locking). Used techniques to select variants include end-of-line programming and garage programming.
<b>Comment</b>	The major difference with calibration is that this later doesn't aim to adapt the SW functionality itself but only aims to adjust the characteristic data of the SW to the HW/SW environment. Characteristic data in the source code of a software function have a significant impact on the functionality of the software.
<b>Example</b>	- Steering wheel controller adaptation to the left or right side can be done with Variant Coding. (Selection of the configuration.) - Country related adaptation of MMI with respect to speed and/or temperature unit (km/h vs. mph, °C vs. F).
<b>Reference</b>	[AUTOSAR_WP2.1.1.1_SW-C-Attributes]
<b>German Term</b>	Variantenkodierung
<b>French Term</b>	--

### 3.52 Deadline

<b>Definition</b>	The point in time when an execution of an entity must be finished.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A deadline is calculated dependent on its local reference system.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[OSEKtime, OS Specification]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.53 Debugging

<b>Definition</b>	Debugging is the process of gathering information in case of a software problem. The information is used to analyze the software problem.
<b>Initiator</b>	WP11 1.1.3
<b>Further Explanations</b>	To analyze and later fix a software problem, in many cases more information than the one provided by the software API is necessary. This can be for example the state of internal variables of the software or a trace of the communication. The information can be collected by different means, e.g. an emulator or a tracing tool for the communication bus.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Debugging/Fehlersuche
<b>French Term</b>	Debugging/recherche d'erreurs

### 3.54 (Conformance) Declaration

<b>Definition</b>	First-party (→ definition 3.78) attestation
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Attestation of product conformance given by a first party, i.e. the Product Supplier.
<b>Reference</b>	[ISO/IEC 17000]; [ISO/IEC 17050]
<b>German Term</b>	Erklärung
<b>French Term</b>	Déclaration

### 3.55 Dependability

<b>Definition</b>	Dependability is defined as the trustworthiness of a computer system such that reliance can justifiable be placed on the service it delivers.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[EAST-Glossary]
<b>German term</b>	Verlässlichkeit
<b>French term</b>	--

### 3.56 Dynamic Routing

<b>Definition</b>	The routing of signals or PDUs (→ definition 3.139) in a gateway can be changed throughout operation without change of the operation mode of the gateway.
<b>Initiator</b>	WP4.2.2.1.6 Gateway
<b>Further Explanations</b>	Dynamic routing requires the change of routing tables during operation. It is not intended to use dynamic routing in the gateway.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[EAST-Glossary]
<b>German term</b>	Verlässlichkeit
<b>French term</b>	--

### 3.57 ECU Abstraction Layer

<b>Definition</b>	The ECU Abstraction Layer is located above the Microcontroller Abstraction Layer (→ definition 3.129) and abstracts from the ECU schematic. It is implemented for a specific ECU and offers an API for access to peripherals and devices regardless of their location (onchip/offchip) and their connection to the microcontroller (port pins, type of interface). Task: make higher software layers independent of the ECU hardware layout.
<b>Initiator</b>	WP 1.1.2
<b>Further Explanations</b>	The ECU Abstraction Layer consists of the following parts: <ul style="list-style-type: none"> <li>• I/O Hardware Abstraction</li> <li>• Communication Hardware Abstraction</li> <li>• Memory Hardware Abstraction</li> <li>• Onboard Device Abstraction</li> </ul> Properties: <ul style="list-style-type: none"> <li>• Implementation: <math>\mu</math>C independent, ECU hardware dependent</li> <li>• Upper Interface (API): <math>\mu</math>C and ECU hardware independent, dependent on signal type</li> </ul>
<b>Comment</b>	--
<b>Example</b>	See WP1.1.2, Layered Software Architecture
<b>Reference</b>	[AUTOSAR SoftwareArchitecture]
<b>German Term</b>	Steuergeräte-Abstraktions-Schicht
<b>French Term</b>	--

### 3.58 ECU Configuration

<b>Definition</b>	Activity of integrating and configuring one ECU's software.
<b>Initiator</b>	WP 4.1.1.2
<b>Further Explanations</b>	Further Explanations: ECU Configuration denotes the activity when one ECU's software is set up for a specific usage inside the ECU. In AUTOSAR the ECU Configuration activity is divided into "Pre-compile time", "Link time" and "Post-build time" configuration.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	ECU Configuration Description (→ definition 3.59), Pre-compile time configuration (→ definition 3.146), Link time configuration (→ definition 3.125), Post-build time configuration (→ definition 3.145), Multiple Configuration Sets (→ definition 3.132).
<b>German Term</b>	--
<b>French Term</b>	--



### 3.59 ECU Configuration Description

<b>Definition</b>	Output of the ECU Configuration activity containing the values of configuration parameters and references.
<b>Initiator</b>	WP 4.1.1.2
<b>Further Explanations</b>	ECU Configuration Description holds the configuration parameter values and references to other module's configurations which have been defined in the ECU Configuration activity.
<b>Comment</b>	ECU Configuration Description may contain the whole ECU Configuration information or only the parts relevant for a specific configuration step (e.g. Pre-compile time).
<b>Example</b>	--
<b>Reference</b>	ECU Configuration Description (→ definition 3.59), Pre-compile time configuration (→ definition 3.146), Link time configuration (→ definition 3.125), Post-build time configuration (→ definition 3.145), Multiple Configuration Sets (→ definition 3.132).
<b>German Term</b>	--
<b>French Term</b>	--

### 3.60 ECU Firmware

<b>Definition</b>	ECU firmware is ECU schematic dependent software located below the AUTOSAR RTE (RunTimeEnvironment).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	ECU Abstraction, Complex Device Driver
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.61 g Electronic Control Unit (ECU)

<b>Definition</b>	Embedded computer system consisting out of at least one CPU and corresponding periphery which is placed in one housing.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	"Small" deleted from EAST definition The term ECU is problematic using LIN busses: An ECU can carry at least one AUTOSAR-SW-Component. Sensors and Actuators which are according to this definition not an ECU are related to an ECU which capsules the sensors or actuators functionality -new
<b>Example</b>	Head Unit (telematics domain).
<b>Reference</b>	[EAST-Glossary]
<b>German Term</b>	Steuergerät
<b>French Term</b>	--

### 3.62 Electrical Signal

<b>Definition</b>	The electrical signal is the electrical representation of technical signals (→ definition 3.194). Electrical signals can only be represented in voltage, current and time
<b>Initiator</b>	WP 2.1.1.2
<b>Further Explanations</b>	When a sensor processes the Technical Signal it is converted into an Electrical Signal. The information can be provided in the current, the voltage or in the timely change of the signal (e.g. a pulse width modulation).
<b>Comment</b>	To describe the Electrical Signal the same means as for the Technical Signal can be used, limited to electrical current and voltage.
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Elektrisches Signal
<b>French Term</b>	--

### 3.63 Entry Point

<b>Definition</b>	A point in a AUTOSAR Software-Component (→ definition 3.18) where an execution entity of the SW-C begins.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	<ul style="list-style-type: none"> <li>• Service of the Server in Client/Server Communication</li> <li>• Reaction after receive Information (Notification)</li> </ul>
<b>Reference</b>	--
<b>German Term</b>	Einsprungpunkt
<b>French Term</b>	--

### 3.64 Error

<b>Definition</b>	Discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[IEC 61508, Part 4]
<b>German Term</b>	Abweichung
<b>French Term</b>	--

### 3.65 Event

<b>Definition</b>	State change of a hardware and/or software entity.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Ereignis
<b>French Term</b>	--

### 3.66 Execution Time

<b>Definition</b>	The time during which a program is actually executing, or more precisely during which a certain thread of execution is active.
<b>Initiator</b>	WP 2.1.1.1
<b>Further Explanations</b>	The execution time of software is the time during which the CPU is executing its instructions. The time the CPU spends on task switches or on the execution of other pieces of software is not considered here. See also: response time, worst case execution time, worst case response time.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Ausführungszeit
<b>French Term</b>	--

### 3.67 Exit Point

<b>Definition</b>	A point in an AUTOSAR Software-Component (→ definition 3.18) where an execution entity of the SW-C ends.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Return point.
<b>Reference</b>	--
<b>German Term</b>	Ausstiegspunkt
<b>French Term</b>	--

### 3.68 Fail-degraded

<b>Definition</b>	Property of a system or functional unit. Describes the ability of a system to continue with intended degraded operation at its output interfaces despite the presence of hardware or software faults.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	<ol style="list-style-type: none"> <li>1. Safety means are not regarded as a part of the normal functionality respectively operation.</li> <li>2. Also known as: Fail-reduced, Fail-soft</li> </ol>
<b>Example</b>	"Limp home" functionality for ECU (reduce torque to assure an arrival at home or service station)
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.69 Fail-operational

<b>Definition</b>	Property of a system or functional unit. Describes the ability of a system or functional unit to continue normal operation at its output interfaces despite the presence of hardware or software faults.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	<ol style="list-style-type: none"> <li>1. Typically, a fail-operational system or functional unit has no safe state.</li> <li>2. Safety means are not regarded as a part of the normal functionality respectively operation.</li> </ol>
<b>Example</b>	Braking system
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.70 Fail-safe

<b>Definition</b>	Property of a system or functional unit. In case of a fault the system or functional unit transits to a safe state.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.71 Fail-silent

<b>Definition</b>	Property of a system or functional unit. In case of a fault the output interfaces are disabled in a way that no further outputs are made.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	Fail-silent is a special case of the fail-safe property.
<b>Comment</b>	--
<b>Example</b>	The fail-silent property can be used to avoid that “babbling idiots” disturb the overall communication.
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.72 Failure

<b>Definition</b>	Termination of the ability of a system or functional unit to perform a required function.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	The cause of a failure is a fault.
<b>Example</b>	--
<b>Reference</b>	[IEC 61508, Part 4]
<b>German Term</b>	Ausfall, Fehlfunktion
<b>French Term</b>	--

### 3.73 Fault

<b>Definition</b>	Abnormal condition that may cause a reduction in, or loss of, the capability of a system or functional unit to perform a required function.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	A fault of a functional unit may cause a failure of this unit.
<b>Example</b>	Physical defect
<b>Reference</b>	[IEC 61508, Part 4]
<b>German Term</b>	Störung
<b>French Term</b>	--

### 3.74 Fault Detection

<b>Definition</b>	The action of monitoring errors and setting fault states to specific values is called fault detection.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The different states are called “not detected”/ “present”/ “intermittent or maturing”/... The names of the fault states are following the ISO/SAE norms; however there is a coordination step in between the states of the DTCs (Diagnostic Trouble Code → see definition in ISO 15765/ ISO14229) and the states of the faults. The SW-C’s Fault Detection is executed decentralized, e.g. each SW-C sets the state of a fault according to the defined fault qualification (SW-C Template). Therefore the Fault Detection is implemented in the SW-C (SW-C could be either Application SW Component or Basic SW Component). There are exceptions; these will be pointed out individually for each fault. The SW-C’s developer will define the conditions (=fault qualification), when these conditions are fulfilled the SW-C notifies a fault to the Diagnostic Memory Management.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[ISO 15765], [ISO14229] [AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	Fehlerdetektion
<b>French Term</b>	Détection de fautes

### 3.75 Fault Reaction

<b>Definition</b>	In case of a Failure of a SW-C there is a specific action to be carried out. This action is called “Fault Reaction”.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Fault Reactions can be implemented decentralized in the SW-C. There might also be the need of coordinating the fault reactions since there are reactions excluding each other. This will be done by a central fault reaction manager.
<b>Reference</b>	--
<b>German Term</b>	Fehlerbehandlung
<b>French Term</b>	Traitement des fautes

### 3.76 Fault Tolerance

<b>Definition</b>	Property of a system or functional unit. In case of n faults the system or functional unit continues with full functionality (n>0).
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Fehlertoleranz
<b>French Term</b>	--

### 3.77 Feature

<b>Definition</b>	The term feature is commonly used in the software tool community to describe characteristics (functionality) of the software.
<b>Initiator</b>	WP 1.2
<b>Further Explanations</b>	In AUTOSAR a feature is represented by one or many metaclasses and their attributes in the AUTOSAR meta-model. Features are used to implement use cases such that a single use case requires one or more features for implementation.
<b>Comment</b>	--
<b>Example</b>	Automatic windshield wiper
<b>Reference</b>	[EAST-Glossary]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.78 First party

<b>Definition</b>	An organization that provides automotive products, which are subject to AUTOSAR conformance testing.
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	Première partie

### 3.79 Flag

<b>Definition</b>	A piece of data that can take on one of two values indicating whether a logical condition is true or false.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Notification flag
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.80 FlexRay Base Cycle

<b>Definition</b>	<p>One operand of the equation used to calculate the Cycle Numbers (→ definition 3.86) of the FlexRay Cells (→ definition 3.82) being used for periodic transmission of FlexRay Frames (→ definition 3.89) in a given FlexRay Slot (→ definition 3.98). Equation: <b>Cycle Number = <math>(B + n * 2^R)_{\text{mod}64}</math></b> Where:</p> <ul style="list-style-type: none"> <li>• Base Cycle <b>B</b> = 0 ... 63</li> <li>• Cycle Repetition <math>2^R = 2^0 \dots 2^6 = 1, 2, 4, 8, \dots 64</math></li> <li>• Variable <b>n</b> = 0 ... 64</li> <li>• <b>B</b> &lt; <math>2^R</math></li> </ul> <p>(See also graphic in FlexRay L-SDU-Identifier → definition 3.93)</p>
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	Synonym: "Cycle Offset", "Cycle Counter Offset"
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	„Basiszyklus“, „Zyklusoffset“, „Zykluszähleroffset“
<b>French Term</b>	--

### 3.81 FlexRay Bus

<b>Definition</b>	A communication system <b>topology</b> in which Nodes (→ definition 3.96) are directly connected to a single, common communication media (as opposed to connection through Stars (→ definition 3.101), gateways, etc.). The term "bus" is also used to refer to <b>the media itself</b> .
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	The term "FlexRay Bus" is not to be confused with the term "FlexRay Cluster" (→ definition 3.84) or "FlexRay Network" (→ definition 3.95).
<b>Comment</b>	Synonym: "FlexRay Communication Bus"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Bus“, „FlexRay Kommunikationsbus“
<b>French Term</b>	--

### 3.82 FlexRay Cell

<b>Definition</b>	<p>One element in a FlexRay Matrix (→ definition 3.94) unequivocally defined by a combination of exactly one FlexRay Slot (or FlexRay Slot Number) (→ definition 3.98) and exactly one FlexRay Cycle (or FlexRay Cycle Number) (→ definition 3.85). In other words: a FlexRay Cell is defined by the tuple &lt;Slot Number, Cycle Number&gt;.</p> <p>Each FlexRay Cell represents one (possible) transmission time interval for at most one FlexRay Frame (→ definition 3.89). If a FlexRay Network (→ definition 3.95) consists of two Channels (→ definition 3.83), there is one FlexRay Matrix per Channel, so there are also two FlexRay Cells defined by the same tuple &lt;Slot Number, Cycle Number&gt;, one for "Channel A" and one for "Channel B".</p>
<b>Initiator</b>	WP4.2.2.1.5
<b>Further</b>	In order to achieve periodic transmission of FlexRay Frames in a given FlexRay



<b>Explanations</b>	Slot, the Cycle Numbers of the FlexRay Cells being used for transmission have to fulfill the following equation: Equation: $\text{Cycle Number} = (\mathbf{B} + \mathbf{n} * 2^{\mathbf{R}})_{\text{mod}64}$ Where: <ul style="list-style-type: none"> <li>• Base Cycle <math>\mathbf{B} = 0 \dots 63</math></li> <li>• Cycle Repetition <math>2^{\mathbf{R}} = 2^0 \dots 2^6 = 1, 2, 4, 8, \dots 64</math></li> <li>• Variable <math>\mathbf{n} = 0 \dots 64</math></li> <li>• <math>\mathbf{B} &lt; 2^{\mathbf{R}}</math></li> </ul>
<b>Comment</b>	Synonym: "FlexRay Matrix Cell"
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	„FlexRay Zelle“, „FlexRay Matrixzelle“, „FlexRay Kommunikationsmatrixzelle“
<b>French Term</b>	--

### 3.83 FlexRay Channel

<b>Definition</b>	The inter-Node ( $\rightarrow$ definition 3.96) connection through which signals are conveyed for the purpose of communication. The communication channel abstracts both the network topology, i.e., Bus ( $\rightarrow$ definition 3.81) or Star ( $\rightarrow$ definition 3.101), as well as the physical transmission medium, i.e. electrical or optical.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	According to the FlexRay Protocol Specification, the two possible Channels of a FlexRay Network ( $\rightarrow$ definition 3.95) are named "Channel A" and "Channel B".
<b>Comment</b>	Synonym: "FlexRay Communication Channel"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Kanal“, „FlexRay Kommunikationskanal“
<b>French Term</b>	--

### 3.84 FlexRay Cluster

<b>Definition</b>	A communication system of multiple Nodes ( $\rightarrow$ definition 3.96) connected directly (Bus topology) or by Star Couplers (Star topology) ( $\rightarrow$ definition 3.101) via a Communication Network consisting of at least one Communication Channel.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	The term "FlexRay Cluster" is not to be confused with the term "FlexRay Bus" ( $\rightarrow$ definition 3.81) which describes a communication system topology. A FlexRay Cluster consists of a FlexRay Network ( $\rightarrow$ definition 3.95) and several FlexRay Nodes.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.85 FlexRay Cycle

<b>Definition</b>	One complete instance of the communication structure that is periodically repeated to comprise the media access method of the FlexRay system. The Communication Cycle consists of a Static Segment, an optional Dynamic Segment, an optional Symbol Window, and a Network Idle Time. The FlexRay Cycles are unequivocally numbered by the FlexRay Cycle Number (→ definition 3.86) ranging from 0 to 63. Even if a FlexRay Network (→ definition 3.95) consists of two Channels, the FlexRay Cycle is always a common quantity of both Channels, irrespective of the data transmission schedule possibly being different for the two Channels.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	Synonym: "FlexRay Communication Cycle"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Zyklus“, „FlexRay Kommunikationszyklus“
<b>French Term</b>	--

### 3.86 FlexRay Cycle Number

<b>Definition</b>	An unequivocal number of a FlexRay Cycle (→ definition 3.85), ranging from 0 to 63.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	Synonym: "FlexRay Communication Cycle Number"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Zyklusnummer“, „FlexRay Kommunikationszyklusnummer“
<b>French Term</b>	--

### 3.87 FlexRay Cycle Offset

<b>Definition</b>	See definition of Base Cycle (→ definition 3.80).
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	This term is mentioned here to simplify finding it via this document's table of contents.  Synonym: "Cycle Counter Offset", "Base Cycle"
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	„Zyklusoffset“
<b>French Term</b>	--

### 3.88 FlexRay Cycle Repetition

<b>Definition</b>	<p>One operand of the equation used to calculate the Cycle Numbers (→ definition 3.86) of the FlexRay Cells (→ definition 3.82) being used for periodic transmission of FlexRay Frames (→ definition 3.89) in a given FlexRay Slot (→ definition 3.98). Equation: <b>Cycle Number = <math>(B + n * 2^R)_{\text{mod}64}</math></b> Where:</p> <ul style="list-style-type: none"> <li>• Base Cycle <b>B</b> = 0 ... 63</li> <li>• Cycle Repetition <math>2^R = 2^0 \dots 2^6 = 1, 2, 4, 8, \dots 64</math></li> <li>• Variable <b>n</b> = 0 ... 64</li> <li>• <b>B</b> &lt; <math>2^R</math></li> </ul> <p>(See also graphic in FlexRay L-SDU-Identifier → definition 3.93)</p>
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	Synonym: "Cycle Counter Repetition"
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	„Zykluswiederholung“, „Zykluszählerwiederholung“
<b>French Term</b>	--

### 3.89 FlexRay Frame

<b>Definition</b>	A structure used by the communication system to exchange information within the system. A FlexRay Frame consists of a header segment, a payload segment and a trailer segment. The payload segment is used to convey application data.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	A FlexRay Frame is a "data package" that may be transmitted within a FlexRay Cell (→ definition 3.82).
<b>Comment</b>	Synonym: "FlexRay L-PDU"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.90 FlexRay Global Time

<b>Definition</b>	A tuple consisting of one specific value of the FlexRay Cycle Counter and one specific value of the FlexRay Cycle Time in Macroticks.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay globale Zeit“
<b>French Term</b>	--

### 3.91 FlexRay L-PDU

<b>Definition</b>	See definition of FlexRay Frame (→ definition 3.89).
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	This term is mentioned here to simplify finding it via this document's table of contents.
<b>Example</b>	--
<b>Reference</b>	Synonym: "FlexRay Frame"
<b>German Term</b>	--
<b>French Term</b>	--

### 3.92 FlexRay L-PDU-Identifier

<b>Definition</b>	<p>A unequivocal identifier of a set of FlexRay Cells (→ definition 3.82) used for periodic transmission of FlexRay Frames (→ definition 3.89) over one or both FlexRay Channels (→ definition 3.83) in a specific FlexRay Slot (→ definition 3.98), which fulfill the equation:</p> $\text{Cycle Number} = (\mathbf{B} + \mathbf{n} * 2^{\mathbf{R}})_{\text{mod}64}$ <p>with <b>exactly one tuple</b> of values for <b>B</b> and <b>2<sup>R</sup></b>. (See also graphics below.) In other words: a FlexRay L-PDU-ID comprises the 4 parameters:</p> <ul style="list-style-type: none"> <li>• Slot Number = 1 ... MaxSlotNumber (≤ 2047)</li> <li>• Base Cycle <b>B</b> = 0 ... 63</li> <li>• Cycle Repetition <b>2<sup>R</sup></b> = 2<sup>0</sup> ... 2<sup>6</sup> = 1, 2, 4, 8, ... 64</li> <li>• Channel = "A", "B", "A and B"</li> </ul> <p>In order to prevent collisions of FlexRay Frames on the Bus (→ definition 3.81), the FlexRay Cells of different FlexRay L-PDU-Identifiers used for transmission shall be disjunctive.</p>
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	<p style="text-align: center;"><b>FlexRay Communication Matrix with L-PDU-Identifier</b></p> <p>Legend:</p> <ul style="list-style-type: none"> <li><span style="color: red;">■</span> L-PDU-Identifier: Slot: 4 Channel: A Base Cycle: 0 Cycle Repetition: 2<sup>1</sup></li> <li><span style="color: cyan;">■</span> L-PDU-Identifier: Slot: 6 Channel: A&amp;B Base Cycle: 1 Cycle Repetition: 2<sup>2</sup></li> </ul>
<b>Comment</b>	Usually, on one specific FlexRay Node (→ definition 3.96), <b>one</b> FlexRay L-PDU-Identifier has <b>one</b> configuration of a FlexRay Communication Controller buffer

	assigned to it. Synonym: "FlexRay L-PDU-ID"
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.93 FlexRay L-SDU-Identifier

<b>Definition</b>	<p>A unequivocal identifier of the <b>payload</b> contained in one or multiple FlexRay Frames (→ definition 3.89) assigned to the same FlexRay L-PDU-Identifier (→ definition 3.92) and therefore periodically transmitted over one or both FlexRay Channels (→ definition 3.83) in one or multiple FlexRay Cells (→ definition 3.82) in a specific FlexRay Slot (→ definition 3.98), and where the equation:</p> $\text{Cycle Number} = (\mathbf{B} + \mathbf{n} * 2^{\mathbf{R}})_{\text{mod}64}$ <p>is fulfilled with <b>exactly one tuple</b> of values for <b>B</b> and <math>2^{\mathbf{R}}</math>. (See also graphics below.)</p>
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	<p style="text-align: center;"><b>FlexRay Communication Matrix with L-SDU-Identifier</b></p> <p style="text-align: center;"> <span style="display: inline-block; width: 10px; height: 10px; background-color: #f08080; border: 1px solid black; margin-right: 5px;"></span> L-SDU-Identifier: Slot: 4 Channel: <b>A</b> Base Cycle: <b>0</b> Cycle Repetition: <math>2^1</math>  <span style="display: inline-block; width: 10px; height: 10px; background-color: #00b0f0; border: 1px solid black; margin-right: 5px;"></span> L-SDU-Identifier: Slot: 6 Channel: <b>A&amp;B</b> Base Cycle: <b>1</b> Cycle Repetition: <math>2^2</math> </p>
<b>Comment</b>	Synonym: "FlexRay L-SDU-ID"
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.94 FlexRay Matrix

<b>Definition</b>	<p>A two-dimensional array with a width of the number of FlexRay Slots (→ definition 3.98) within one FlexRay Cycle (→ definition 3.85) and a height of 64 FlexRay Cycles, numbered 0 ... 63. (See also graphics below.)</p> <p>This array is being used to describe the (possible) transmission time intervals on a FlexRay Channel (→ definition 3.83).</p> <p>If a FlexRay Network (→ definition 3.95) consists of two Channels, there is one FlexRay Matrix per Channel (resulting in a total of two Matrixes), since the data transmission schedule may be different for the two FlexRay Channels.</p>
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	<p style="text-align: center;"><b>FlexRay Communication Matrix</b></p> <p style="text-align: center;"> <span style="display: inline-block; width: 10px; height: 10px; background-color: #f080f0; border: 1px solid black; margin-right: 5px;"></span> FlexRay Slot No. 2  <span style="display: inline-block; width: 10px; height: 10px; background-color: #00ced1; border: 1px solid black; margin-right: 5px;"></span> FlexRay Cycle No. 5  <span style="display: inline-block; width: 10px; height: 10px; background-color: #008000; border: 1px solid black; margin-right: 5px;"></span> FlexRay Cell ( 2   5 )         </p>
<b>Comment</b>	Synonym: "FlexRay Communication Matrix"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Matrix“, „FlexRay Kommunikationsmatrix“
<b>French Term</b>	--

### 3.95 FlexRay Network

<b>Definition</b>	<p>The combination of the (up to two) FlexRay Communication Channels that connect the FlexRay Nodes (→ definition 3.96) of a FlexRay Cluster (→ definition 3.84).</p>
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	<p>The term "FlexRay Network" is not to be confused with the term "FlexRay Cluster" or "FlexRay Bus" (→ definition 3.81).</p>
<b>Comment</b>	Synonym: "FlexRay Communication Network"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Netzwerk“, „FlexRay Kommunikationsnetzwerk“
<b>French Term</b>	--

### 3.96 FlexRay Node

<b>Definition</b>	A logical entity connected to the FlexRay Network (→ definition 3.95) that is capable of sending and/or receiving frames.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Knoten“
<b>French Term</b>	--

### 3.97 FlexRay Physical Communication Link

<b>Definition</b>	An inter-Node (→ definition 3.96) connection through which signals are conveyed for the purpose of communication. All Nodes connected to a given Physical Communication Link share the same electrical or optical signals (i.e., they are not connected through repeaters, Stars (→ definition 3.101), gateways, etc.). Examples of a Physical Communication Link include a Bus (→ definition 3.81) Network or a point-to-point connection between a Node and a Star. A Communication Channel may be constructed by combining one or more Physical Communication Links together using Stars.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.98 FlexRay Slot

<b>Definition</b>	An interval of time during which access to a Communication Channel is granted exclusively (at least in the static segment) to a specific Node (→ definition 3.96) for the transmission of a Frame (→ definition 3.89) with a frame ID corresponding to the Slot Number (→ definition 3.100) of that Slot. FlexRay distinguishes between Static Communication Slots and Dynamic Communication Slots. The FlexRay Slots are unequivocally numbered by the FlexRay Slot Number ranging from 1 to a configurable maximum number $\leq 2047$ . If a FlexRay Network (→ definition 3.95) consists of two Channels (→ definition 3.83), the Static Slots of “Channel A” and the Static Slots of “Channel B” occur concurrently, since all Static FlexRay Slots have the same length irrespective of the data transmission schedule. However, the Dynamic Slots of “Channel A” are independent from the Dynamic Slots of “Channel B”, since the data transmission schedule may be different for the two FlexRay Channels.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	In the dynamic segment, Slot Multiplexing between multiple Nodes is allowed. In the static segment each Slot (→ definition 3.98) on a Channel is owned by exactly one Node (i.e., Slot Multiplexing is <b>not</b> allowed in the static segment). Slot Multiplexing (i.e., different FlexRay Nodes owning a Slot in different Cycles (→ definition 3.85) for data transmission) is allowed in the dynamic segment, and it is up to the application to ensure that in any given Cycle no two Nodes transmit in the same Slot on the same Channel.
<b>Comment</b>	Synonym: “FlexRay Communication Slot”
<b>Example</b>	--

<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Slot“, „FlexRay Kommunikationslot“
<b>French Term</b>	--

### 3.99 FlexRay Slot Multiplexing

<p><b>Definition</b></p>	<p>A method used to fill a FlexRay Slot (→ definition 3.98) on a Channel (→ definition 3.83) more efficiently by alternating the Frames being sent in this Slot from Cycle (→ definition 3.85) to Cycle. In order to achieve periodic transmission of FlexRay Frames (→ definition 3.89) in a given FlexRay Slot, the Cycle Numbers (→ definition 3.86) of the FlexRay Cells (→ definition 3.82) being used for transmission have to fulfill the equation:</p> $\text{Cycle Number} = (\mathbf{B} + \mathbf{n} * 2^{\mathbf{R}})_{\text{mod}64}$ <p>Where:</p> <ul style="list-style-type: none"> <li>• Base Cycle <b>B</b> = 0 ... 63</li> <li>• Cycle Repetition <math>2^{\mathbf{R}} = 2^0 \dots 2^6 = 1, 2, 4, 8, \dots 64</math></li> <li>• Variable <b>n</b> = 0 ... 64</li> <li>• <b>B</b> &lt; <math>2^{\mathbf{R}}</math></li> </ul> <p>In the static segment, each Slot on a Channel is owned by exactly one Node (→ definition 3.96). Therefore, in the static segment Slot Multiplexing is only allowed amongst semantically different Frames sent by the same Node, but <b>not</b> amongst different Nodes of a FlexRay Cluster (→ definition 3.84). Thus, this form of Multiplexing is called "<b>Single Sender Slot Multiplexing</b>".</p> <p>In the dynamic segment, Slot Multiplexing is also allowed amongst different Nodes of a FlexRay Cluster, i.e. different Nodes may send in the same dynamic Slot on the same Channel in different Cycles, hence with different FlexRay L-PDU-Identifier (→ definition 3.92) defining disjunctive FlexRay Cells. Thus, this form of Multiplexing is called "<b>Multiple Sender Slot Multiplexing</b>".</p> <p>In any case, it is up to the software to prevent concurrent sending attempts (of different Nodes or applications) in the same Cell.</p> <p>(See also graphics below.)</p>
<p><b>Initiator</b></p>	<p>WP4.2.2.1.5</p>
<p><b>Further Explanations</b></p>	<p style="text-align: center;"><b>FlexRay Communication Matrix with Slot Multiplexing</b></p> <p style="text-align: center;"> <span style="display: inline-block; width: 10px; height: 10px; background-color: red; border: 1px solid black; margin-right: 5px;"></span> L-PDU-Identifier: Slot: 5 Channel: A Base Cycle: 0 Cycle Repetition: <math>2^1</math>  <span style="display: inline-block; width: 10px; height: 10px; background-color: blue; border: 1px solid black; margin-right: 5px;"></span> L-PDU-Identifier: Slot: 5 Channel: A&amp;B Base Cycle: 3 Cycle Repetition: <math>2^2</math> </p>
<p><b>Comment</b></p>	<p>--</p>



<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.100 FlexRay Slot Number

<b>Definition</b>	An unequivocal number of a FlexRay Slot (→ definition 3.98), ranging from 1 to a configurable maximum number ≤ 2047.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	Synonym: "FlexRay Slot Identifier", "FlexRay Slot ID"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Slotnummer“, „FlexRay Kommunikationslotnummer“
<b>French Term</b>	--

### 3.101 FlexRay Star

<b>Definition</b>	A device that allows information to be transferred from one Physical Communication Link (→ definition 3.97) to one or more other Physical Communication Links. A star duplicates information present on one of its links to the other links connected to the star. A star can be either passive or active.
<b>Initiator</b>	WP4.2.2.1.5
<b>Further Explanations</b>	--
<b>Comment</b>	Synonym: "Star", "Star Couplers"
<b>Example</b>	--
<b>Reference</b>	[FR_PROTOCOL]
<b>German Term</b>	„FlexRay Stern“, „Stern“, „Sternkoppler“
<b>French Term</b>	--

### 3.102 Frame

<b>Definition</b>	Data unit according to the data link protocol specifying the arrangement and meaning of bits or bit fields in the sequence of transfer across the transfer medium .
<b>Initiator</b>	WP1.1.2
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	A CAN frame consists of up to 8 bytes of payload data and additional protocol specific bits / bit fields (e.g. CAN-Identifier).
<b>Reference</b>	[ISO OSEK, Glossary]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.103 Frame PDU

<b>Definition</b>	A PDU that fits into 1 frame instance. e.g. it does not need to be fragmented across more than 1 frame for transmission over a network.
<b>Initiator</b>	WP1.1.2
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.104 Function

<b>Definition</b>	<ol style="list-style-type: none"> <li>1. A task, action or activity that must be accomplished to achieve a desired outcome.</li> <li>2. A part of programming code that is invoked by other parts of the program to fulfill a desired purpose.</li> <li>3. In mathematics, a function is an association between two sets of values in which each element of one set has one assigned element in the other set so that any element selected becomes the independent variable and its associated element is the dependent variable.</li> </ol>
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	Due to the different meanings in texts using the term application the appropriate meaning should be explained in detail or referenced.
<b>Example</b>	<ol style="list-style-type: none"> <li>2. C-Code Function</li> <li>3. <math>Y=f(x)</math></li> </ol>
<b>Reference</b>	[IEEE12331], [EAST-Glossary]
<b>German Term</b>	Funktion
<b>French Term</b>	--

### 3.105 Functional Network

<b>Definition</b>	A logical structure of interconnections between defined functional parts of features (→ definition 3.77).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Funktionales Netzwerk
<b>French Term</b>	--

### 3.106 Functional Unit

<b>Definition</b>	An entity of software or hardware, or both, capable of accomplishing a specified purpose.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	ECU, AUTOSAR Software Component, ...
<b>Reference</b>	[ISO 2382-1]
<b>German Term</b>	Funktionale Einheit
<b>French Term</b>	--

### 3.107 Functionality

<b>Definition</b>	Functionality comprises User-visible and User-non-visible functional aspects of a system.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	EAST glossary not applicable, due to use of function.
<b>Example</b>	Functionality of a communication system is a user-non-visible aspect.
<b>Reference</b>	--
<b>German Term</b>	Funktionalität
<b>French Term</b>	--

### 3.108 Gateway

<b>Definition</b>	A gateway is functionality within an ECU that performs a frame or signal mapping function between two communication systems. Communication system in this context means e.g. a CAN system or one channel of a FlexRay system.
<b>Initiator</b>	WP 2.1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	Gateway ECU 0
<b>German Term</b>	--
<b>French Term</b>	--

### 3.109 Gateway ECU

<b>Definition</b>	A gateway ECU is an ECU (→ definition 3.61) that is connected to two or more communication channels, and performs gateway functionality.
<b>Initiator</b>	WP 2.1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	Gateway 3.108
<b>German Term</b>	--
<b>French Term</b>	--

### 3.110 Hardware Connection

<b>Definition</b>	HW Connections are used to describe the connection of HW elements (→ definition 3.111) among each other. It defines/characterizes the interrelationship among HW Elements (for abstract modelling). The HW Ports (→ definition 3.113) of the HW Elements serve as connection points for this purpose.
<b>Initiator</b>	WP 2.1.1.2
<b>Further Explanations</b>	In AUTOSAR are 2 kinds of HW Connections defined: Assembly HW Connection Delegation HW Connection
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of ECU Resource Template]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.111 Hardware Element

<b>Definition</b>	The HW Element is the main describing element of an ECU (→ definition 3.61). It provide HW ports (→ definition 3.113) for being interconnected among each others. A generic HW Element specifies definitions valid for all specific HW Elements.
<b>Initiator</b>	WP 2.1.1.2
<b>Further Explanations</b>	A HW Element is the piece or a part of the piece to be described with the ECU Resource Template. It uses other elements as primitive: This means HW elements can be nested (through HW Containers, a hierarchical structur of HW Elements). At the lowest level a HW Element only uses primitives
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of ECU Resource Template]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.112 Hardware Interrupt

<b>Definition</b>	Interrupt triggered by HW event
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	2 sorts of HW events <ul style="list-style-type: none"> <li>• Processor-intern: events as for example division by zero, arithmetical overflow, non-implemented instruction</li> <li>• Processor-extern: events as for example response of peripheral device (e.g. PWM), memory error, timer</li> </ul>
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	Translation/Adaptation from [VDI Lexikon]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.113 Hardware Port

<b>Definition</b>	The HW port exposes functionality to the exterior of the HW element (→ definition 3.111). HW elements can be connected via HW Connections (→ definition 3.110). It defines a connection Endpoint for the HW Element.
<b>Initiator</b>	SystemTeam
<b>Further Explanations</b>	HW elements provide HW ports for being interconnected among each others. Each HW port has a name which is unique within the HW element it is located in.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of ECU Resource Template]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.114 I-PDU

<b>Definition</b>	Interaction Layer Protocol Data Unit Collection of messages for transfer between nodes in a network. At the sending node the Interaction Layer ( <i>IL</i> ) is responsible for packing messages into an I-PDU and then sending it to the Data Link Layer ( <i>DLL</i> ) for transmission. At the receiving node the DLL passes each I-PDU to the IL which then unpacks the messages sending their contents to the application.
<b>Initiator</b>	WP1.1.2
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	OSEK COM specifies an Interaction Layer and works on I-PDUs
<b>Reference</b>	[ISO OSEK, Glossary]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.115 Implementation Conformance Statement

<b>Definition</b>	Implementation Conformance Statement (ICS) describes the actual implementation of a BSW module or a BSW cluster in terms of (1) the actually supported values or value ranges of parameters that enable the configuration of the functionality of a module/cluster implementation and (2) the signatures of the operations actually provided and required at the interface of a module/cluster implementation
<b>Initiator</b>	WP11-2.2
<b>Further Explanations</b>	(1) An SWS of a module/cluster specifies the configurable functionality and configurable operations of all SWS-conformant module/cluster implementations (BSW implementations). The ICS restricts the generally configurable functionality and operations according to what a concrete module/cluster actually implements according to the specification (SWS). (2) The ICS is used to derive values of configuration parameters (CP) valid for the given implementation of a module/cluster and, e.g., to select and parameterize conformance test suites. (3) A BSW implementation can consist of variable parts, i.e., operations and functionality. A BSW implementation can be adjusted with CPs within a certain range of permitted parameter values. An ICS describes which parts a BSW implementation actually supports and to what extend. (4) For any given CP : a. The absolute range is limited by the type of the CP. b. The permitted range is limited by the SWS. c. The actual range is limited by the implementation of the BSW module or BSW cluster. A test suite or the integration of the BSW module or BSW cluster into a target platform limits the actual range to a single value.
<b>Comment</b>	An ICS is a view on a BSW Module Description (BSWMD). An ICS can be extracted from a BSWMD.
<b>Example</b>	(1) An ICS describes which operations a BSW module/cluster actually provides (2) The number of entries for the standard job queue of the NVRAM Manager (NvM) is defined with CP NVM_SIZE_STANDARD_JOB_QUEUE. a. The type of the CP is unit8. Thus the absolute range is 0..255. b. The SWS limits the permitted range to 1..255. c. A specific NvM implementation states to support the full range of permitted values 1..255. Tests run the specific NvM implementation with job queues of 1, 8 and 173 entries. During integration the number of queue entries is limited to 64 entries because of known application-specific usage constraints.
<b>Reference</b>	The definition is partially based on ITU-T Recommendation X.296 [1]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.116 Indication

<b>Definition</b>	Service primitive defined in the ISO/OSI Reference Model (ISO 7498). With the service primitive 'indication' a service provider informs a service user about the occurrence of either an internal event or a service request issued by another service user. [OSEK BD]
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	An indication is e.g. a specific notification generated by the OSEK underlying layer to inform about a Message Reception Error.
<b>Comment</b>	--
<b>Example</b>	OSEK Com notification class 1 and 3.
<b>Reference</b>	[OSEK BD], [OSEK Com]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.117 Integration

<b>Definition</b>	The progressive assembling of system components into the whole system.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[ISO 2382-20]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.118 Interface

<b>Definition</b>	A shared boundary between two functional units (→ definition 3.106) defined by various characteristics pertaining to the functions, physical interconnections, signal exchanges, and other characteristics, as appropriate.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	In AUTOSAR the interface has specific meanings: See Standardized AUTOSAR Interface (→ definition 3.184) and Standardized Interface (→ definition 3.185).
<b>Comment</b>	--
<b>Example</b>	Diagnosis Service
<b>Reference</b>	[ISO 2382-1]
<b>German Term</b>	Schnittstelle
<b>French Term</b>	--

### 3.119 Interrupt

<b>Definition</b>	Event that enforces the processor to change its state. This interruption causes the normal sequence of instructions to be stopped. Once an interrupt occurred, the running software entity is suspended and an interrupt service routine (→ definition 0) (the one dedicated to this interrupt) is called.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Two sorts of interrupts exists: HW and SW interrupts (→ definition 3.112 and definition 3.179)
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	Translation/Adaptation from [VDI Lexikon]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.120 Interrupt frames

<b>Definition</b>	The interrupt frame is the (normally compiler specific) definition of the interrupt service routine (→ definition 0) (e.g. with a keyword <code>_interrupt</code> ) This definition may affect the layout of the interrupt vector table (→ definition 0). The service routine (ISR) is called inside the interrupt frame. This is a requirement from WP1.1.2 to separate the compiler dependent part.
<b>Initiator</b>	WP4.2.2.2.
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.121 Interrupt Service Routine (ISR)

<b>Definition</b>	A software routine called in case of an interrupt (→ definition 3.119)
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	ISRs have normally higher priority than normal processes and can only be suspended by another ISR which presents a higher priority than the one running.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[VDI Lexikon]
<b>German Term</b>	--
<b>French Term</b>	--



### 3.122 Interrupt Vector Table

<b>Definition</b>	An interrupt vector table is a table of interrupt vectors that associates the interrupt service routines (→ definition 0) with the corresponding interrupt request (typically by an array of jumps or similar mechanisms).
<b>Initiator</b>	WP4.2.2.2.
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.123 Invalid Flag

<b>Definition</b>	For a signal in a PDU an optional invalid flag can be added to the PDU payload layout. This flag indicates the validity of other signals in the payload. In case the invalid flag of a signal is set to true in a PDU instance, the respective signal in the payload of the PDU instance does not contain a valid signal value.
<b>Initiator</b>	WP4.2.2.1.6 Gateway
<b>Further Explanations</b>	This mechanism may be used in gateways to indicate that parts of an PDU do not contain valid data.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Ungültigkeitsflag
<b>French Term</b>	--

### 3.124 Invalid Value of Signal

<b>Definition</b>	For a signal in a PDU an optional invalid value can be defined.
<b>Initiator</b>	WP4.2.2.1.6 Gateway
<b>Further Explanations</b>	The invalid value is element of the signal value range that can be represented and transported by the signal. The invalid value is the value that is used in all situations where the receiver should be notified that the value in a signal is not valid.
<b>Comment</b>	--
<b>Example</b>	In case a PDU for a destination network of a gateway is composed from two PDUs of two different source networks, the failure to receive one PDU can be indicated as invalid values in the respective signals of the transmitted PDU in the destination network.
<b>Reference</b>	--
<b>German Term</b>	Ungültigkeitswert
<b>French Term</b>	--

### 3.125 Link time configuration

<b>Definition</b>	The configuration of the SW module is done during link time.
<b>Initiator</b>	WP4.1.1.2
<b>Further Explanations</b>	The object code of the SW modules receives parts of its configuration from another object code file or it is defined by linker options.
<b>Comment</b>	--
<b>Example</b>	Initial value of a signal.
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.126 Mapping

<b>Definition</b>	Mapping designates the distribution of elements in the logical view to elements in the physical view.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	In general several entities may be allocated to one container but an entity may be allocated to only one container.
<b>Comment</b>	--
<b>Example</b>	a) Mapping of AUTOSAR Signals onto Frames (for inter-ECU communication). b) Mapping of SW-C onto ECUs (Distribution of the SW-Components to the ECUs).
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.127 MCAL Signal

<b>Definition</b>	The MCAL signal is the software representation of the conditioned signal (→ definition 3.41). It is provided by the microcontroller abstraction layer (MCAL) and is further processed by the ECU abstraction.
<b>Initiator</b>	WP 2.1.1.2
<b>Further Explanations</b>	The processing unit is accessing the Conditioned Signal through some peripheral device that typically digitises the Conditioned Signal into a software representation. The transformation from the Conditioned Signal to the MCAL Signal has to take the digitalisation error into account in order to provide information about the quality loss between the Technical Signal and the MCAL Signal.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.128 Metadata

<b>Definition</b>	Metadata is data about data
<b>Initiator</b>	WP1.2
<b>Further Explanations</b>	Metadata includes pertinent information about data, including information about the authorship, versioning, access-rights, timestamps etc..
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.129 Microcontroller Abstraction Layer (MCAL)

<b>Definition</b>	Software layer containing drivers to enable the access of onchip peripheral devices of a microcontroller and offchip memory mapped peripheral devices by a defined API (→ definition 3.4). Task: make higher software layers independent of the microcontroller.
<b>Initiator</b>	WP 1.1.2
<b>Further Explanations</b>	The Microcontroller Abstraction Layer is the lowest software layer of the Basic Software. The Microcontroller Abstraction Layer consists of the following parts: <ul style="list-style-type: none"> <li>• I/O Drivers</li> <li>• Communication Drivers</li> <li>• Memory Drivers</li> <li>• Microcontroller Drivers</li> </ul> Properties: <ul style="list-style-type: none"> <li>• Implementation: <math>\mu</math>C dependent</li> <li>• Upper Interface (API): standardizable and <math>\mu</math>C independent</li> </ul>
<b>Comment</b>	--
<b>Example</b>	Examples of drivers located in the Microcontroller Abstraction Layer are: <ul style="list-style-type: none"> <li>• onchip eeprom driver</li> <li>• onchip adc driver</li> <li>• offchip flash driver</li> </ul>
<b>Reference</b>	[AUTOSAR Software Architecture]
<b>German Term</b>	Mikrocontroller-Abstraktions-Schicht
<b>French Term</b>	Couche d'abstraction du microcontrôleur

### 3.130 Mistake

<b>Definition</b>	Human error
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[DIN 40041]
<b>German Term</b>	menschliches Versagen
<b>French Term</b>	erreur humaine

### 3.131 Multimedia Stream

<b>Definition</b>	A consistent sequence of digital data versus time which is suited as input for devices which transfer these data into a continuous visible or audible impression to humans. When transferred over a physical link, multimedia stream data typically are produced at the same rate (by the data source), as they are consumed (by the data sinks).
<b>Initiator</b>	--
<b>Further Explanations</b>	<p>A multimedia stream usually follows a certain standard (e.g. MPEG-x). When transferred over a physical link, a multimedia stream needs a certain minimum bandwidth (in terms of bits/second) in order to allow continuous impressions.</p> <p>A multimedia stream in a car typically exists for several seconds (a warning signal, a navigation hint) up to several hours (a video film, a phone call, playing a radio program). Resources (e.g. bus system channels) needed by the stream have to be allocated continuously over this lifetime (this is a difference to e.g. file transfer, which may be split into several chunks of data).</p> <p>The source of a multimedia stream typically is a specialized device and/or software program (a tuner, a microphone, a text-to-speech engine, etc.). The same holds for the sinks (an audio amplifier or mixer, a voice recognition software, an MPEG decoder, etc.).</p>
<b>Comment</b>	The term "visible or audible impression to humans" should not be taken too literally, because streams can also be used to transfer machine readable data (e.g. modem, encrypted signals). But it is this condition, which defines the standards and technology used in multimedia streams.
<b>Example</b>	<p>Audio stream as output of or input to a telephone (mono, low bandwidth)</p> <p>Audio stream as output of a radio tuner (stereo, high bandwidth)</p> <p>Video stream as output of a television tuner</p> <p>An example for the physical implementation on a multimedia bus is the Firewire isochronous stream. see reference</p>
<b>Reference</b>	[IEEE 1394]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.132 Multiple Configuration Sets

<b>Definition</b>	A SW module has more than one alternative configuration (parameter) set, which can be selected according to external requirements. The set can ONLY be selected during start-up and it is not allowed to switch the set during runtime.
<b>Initiator</b>	WP4.1.1.2
<b>Further Explanations</b>	Multiple configuration sets reside in the ECU non-volatile memory at the same time, the active configuration is selected at the start-up of the ECU. Only BSW modules can have multiple configuration sets
<b>Comment</b>	Multiple configuration is a kind of data variant coding
<b>Example</b>	The same ECU can be used for the left and the right window lifter, the actual pin setting determines which configuration set will be used.
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.133 Multiplexed PDU

<b>Definition</b>	A multiplexed PDU is a PDU with a configurable number of different payload layouts.
<b>Initiator</b>	WP4.2.2.1.6 Gateway
<b>Further Explanations</b>	Each instance of a multiplexed PDU has a distinct layout. The set of possible layouts is statically defined. A selector signal defines which layout is used in a PDU instance. The selector signal must reside at the same position in all layouts. Each layout is identified by a unique selector value. The length of each instance of a multiplexed PDU is fixed.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.134 Non-AUTOSAR Component

<b>Definition</b>	A Non-AUTOSAR Component is a Component (→ definition 3.5 and definition 3.38) not implemented according to AUTOSAR guidelines and processes and which is not using the AUTOSAR Runtime Environment to communicate with its environment. A Non-AUTOSAR Component is located on a separate (non AUTOSAR) hardware entity which communicates with at least one Atomic Software Component.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	All interfaces have to be adapted to AUTOSAR Interfaces (→ definition 3.14) by the AUTOSAR Runtime Environment and/or Basic Software (→ definition 3.23) of ECUs containing Atomic Software Components (→ definition 3.11) communicating with Non-AUTOSAR Components.
<b>Comment</b>	A Non-AUTOSAR Component is a representation on the VFB of a non-AUTOSAR entity, such as a non-AUTOSAR ECU.
<b>Example</b>	ECUs developed not according to the AUTOSAR process and with no AUTOSAR Software inside. Smart Sensors
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.135 Notification

<b>Definition</b>	Informing a software entity about a state change of a hardware and/or software entity which has occurred.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The informing about a state change can be done by an activation of a software part or by setting a flag (→ definition 3.78).
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[OSEK Com]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.136 OS-Application

<b>Definition</b>	A block of software including tasks, interrupts, hooks and user services that form a cohesive functional unit.
<b>Initiator</b>	Robert Rimkus
<b>Further Explanations</b>	Trusted: An OS-Application that is executed in privileged mode and has unrestricted access to the API and hardware resources. Non-trusted: An OS-Application that is executed in non-privileged mode has restricted access to the API and hardware resources.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of OS]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.137 Partitioning

<b>Definition</b>	Decomposition, the separation of the whole system into functional units and further into software components.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	Partitioning has to be done before the mapping in order to identify the components that are to be mapped.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	Based on [IEEE Std.610.12-1990]
<b>German Term</b>	Partitionierung
<b>French Term</b>	--

### 3.138 PCI

<b>Definition</b>	PCI is the abbreviation of "Protocol Control Information". This Information is needed to pass a SDU (→ definition 3.168) from one instance of a specific protocol layer to another instance. E.g. it contains source and target information.
<b>Initiator</b>	WP 1.1.2
<b>Further Explanations</b>	The PCI is added by a protocol layer on the transmission side and is removed again on the receiving side.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

**3.139 PDU**

<b>Definition</b>	PDU is the abbreviation of "Protocol Data Unit". The PDU contains SDU (→ definition 3.168) and PCI (→ definition 3.138).
<b>Initiator</b>	WP1.1.2
<b>Further Explanations</b>	<p>On the transmission side the PDU is passed from the upper layer to the lower layer, which interprets this PDU as its SDU.</p>
<b>Comment</b>	--
<b>Example</b>	OSEK COM specifies an Interaction Layer
<b>Reference</b>	[ISO OSEK Glossary]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.140 PDU Timeout

<b>Definition</b>	Maximum time between the receptions of two instances of one PDU is exceeded.
<b>Initiator</b>	WP4.2.2.1.6 Gateway
<b>Further Explanations</b>	This timeout indicates that the last reception of a PDU instance is too long in the past. As a consequence it can be concluded that the data in the last PDU instance is outdated.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.141 Peripheral Hardware

<b>Definition</b>	Hardware devices integrated in micro-controller architecture to interact with the environment.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Memory, CAN-Controller, ADC, DIO, etc.
<b>Reference</b>	--
<b>German Term</b>	Peripherie
<b>French Term</b>	--

### 3.142 Personalization

<b>Definition</b>	User-specific and memorized adjustment of SW data or selection of functional alternatives.
<b>Initiator</b>	WP 10.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Seat parameters (position, activation status of drive-dynamic seat) can be stored in correlation to a user ID. For a given user ID the seat can be adjusted according to the stored position parameters and the drive-dynamic seat can be activated or deactivated.
<b>Reference</b>	--
<b>German Term</b>	Personalisierung
<b>French Term</b>	--



### 3.143 Port

<b>Definition</b>	A port belongs to a component (→ definition 3.38) and is the interaction point between the component and other components. The interaction between specific ports of specific components is modeled using connectors (→ definition 3.46). A port can either be a p-port (→ definition 3.151) or an r-port (→ definition 3.155).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	For more information see AUTOSAR Specification of VFB
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.144 Port Interface

<b>Definition</b>	A Port Interface characterizes the information provided or required by a port (→ definition 3.143) of a component (→ definition 3.38).
<b>Initiator</b>	Wp 1.1.1
<b>Further Explanations</b>	A Port Interface is either a Client-Server Interface (→ definition 3.33) in case client-server communication (→ definition 3.32) is chosen or a sender-receiver Interface (→ definition 3.171) in case sender-receiver communication (→ definition 3.170) is used.
<b>Comment</b>	For more information see: AUTOSAR Specification of VFB
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.145 Post-build time configuration

<b>Definition</b>	The configuration of the SW module is possible after building the SW module.
<b>Initiator</b>	WP4.1.1.2
<b>Further Explanations</b>	The SW may either receive elements of its configuration during the download of the complete ECU software resulting from the linkage of the code, or it may receive its configuration file that can be downloaded to the ECU separately, avoiding a re-compilation and re-build of the ECU SW modules. In order to make the post-build time re-configuration possible, the re-configurable elements shall be stored at a known position in the ECU storage area
<b>Comment</b>	--
<b>Example</b>	Identifiers of the CAN frames
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.146 Pre-Compile time configuration

<b>Definition</b>	The configuration of the SW module is done at source code level and will be effective after compile time.
<b>Initiator</b>	WP4.1.1.2
<b>Further Explanations</b>	The source code contains all the ECU configuration data and when compiled together, it produces the given SW.
<b>Comment</b>	--
<b>Example</b>	Preprocessor switch for enabling the development error detection and reporting
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.147 Private Interface (API 3)

<b>Definition</b>	A private interface is an interface within the Basic Software (→ definition 3.23) of AUTOSAR which is neither standardized nor defined within AUTOSAR.
<b>Initiator</b>	WP 1.1.2
<b>Further Explanations</b>	The goal of the private interface is to enable a more efficient implementation of basic software modules. Basic software modules sharing a private interface have to be distributed as one package. This package has to behave exactly the same as separate modules would. It must provide the same standardized interfaces to the rest of the basic software and/or RTE as separate modules would. It has to be configured exactly the same as separate modules would be configured.
<b>Comment</b>	Private interfaces contradict the goal of exchangeability of standard software modules and should be avoided.
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.148 Probability of failure

<b>Definition</b>	Probability of the occurrence of a failure in a system or functional unit.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Ausfallwahrscheinlichkeit
<b>French Term</b>	--

### 3.149 Procedure Call

<b>Definition</b>	A simple statement that provides the actual parameters for and invokes the execution of a procedure (software function).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A synchronous communication mechanism can be implemented by a procedure call.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[ISO 2382-15]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.150 Process

<b>Definition</b>	An executable unit managed by an operating system scheduler that has its own name space and resources (including memory) protected against use from other processes.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A process consists of n Task (n>=1)
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Prozess
<b>French Term</b>	--

### 3.151 Provide Port

<b>Definition</b>	Specific Port (→ definition 3.143) providing data (→ definition 3.48) or providing a service of a server (→ definition 3.173).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The Provide Port is sometimes abbreviated as PPort or P-Port.
<b>Comment</b>	--
<b>Example</b>	<ul style="list-style-type: none"> <li>• Server Port</li> <li>• Sender Port</li> </ul>
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.152 Redundancy

<b>Definition</b>	Existence of means, in addition to the means which would be sufficient for a system or functional unit to perform a required function.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[IEC 61508, Part 4]
<b>German Term</b>	Redundanz
<b>French Term</b>	--

### 3.153 Reliability

<b>Definition</b>	Probability of a system or functional unit to perform as expected under specified conditions within a time interval.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Zuverlässigkeit
<b>French Term</b>	--

### 3.154 Relocatability

<b>Definition</b>	Capability of a software part being executed on different hardware environments without changing the code of the software part.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Verschiebbarkeit
<b>French Term</b>	--

### 3.155 Require Port

<b>Definition</b>	Specific Port (→ definition 3.143) requiring data (→ definition 3.48) or requiring a service of a server.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The Require Port is sometimes abbreviated as RPort or R-Port.
<b>Comment</b>	--
<b>Example</b>	<ul style="list-style-type: none"> <li>• Client Port</li> <li>• Receiver Port</li> </ul>
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.156 Required property

<b>Definition</b>	A <i>required</i> property or quality of a design entity (e.g. SW component or system) is a property or quality which has to be fulfilled by the environment of this design entity.
<b>Initiator</b>	WP10.1
<b>Further Explanations</b>	A property or quality can be required by a stakeholder (e.g. customer) or another design entity.
<b>Comment</b>	--
<b>Example</b>	<p>1) In order to meet its functionality, a SW component A requires a minimum temporal resolution of a signal (information on a required port) which has to be fulfilled by SW component B.</p> <p>2) SW component requires to be activated by the runtime environment every 100ms with a jitter of 10ms.</p>
<b>Reference</b>	Compare term asserted property (→ definition 3.7)
<b>German Term</b>	Erforderliche Eigenschaft
<b>French Term</b>	--

### 3.157 Resource

<b>Definition</b>	A resource is a required but limited hardware entity of an ECU (→ definition 3.61), which in general can be accessed concurrently, but not simultaneously, by multiple software entities.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	The OSEK definition [OSEK BD] cannot be used, due to the specific usage in OSEK OS.
<b>Example</b>	CPU-load, interrupts (mechanism itself and the resulting CPU-load), memory, peripheral hardware, communication, ...
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.158 Resource-Management

<b>Definition</b>	Entity which controls the use of resources (→ definition 3.157).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The main functionality of resource management is the control of simultaneous use of a single resource by several entities, e.g. scheduling of requests, multiple access protection.
<b>Comment</b>	--
<b>Example</b>	OS-scheduler (CPU-load management)
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.159 Response Time

<b>Definition</b>	Time between receiving a stimulus and delivering an appropriate response or reaction.
<b>Initiator</b>	WP 2.1.1.1
<b>Further Explanations</b>	The response time describes the time between a stimulus like e.g. the state change of hardware or software entity and the expected reaction of the system (e.g. response, actuator activation). Synonym: reaction time See also: execution time, worst case execution time and worst case response time.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Antwortzeit
<b>French Term</b>	--

### 3.160 Risk

<b>Definition</b>	The product of the probability of failure and the severity of outcome.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Risiko
<b>French Term</b>	--

### 3.161 Robustness

<b>Definition</b>	Ability of a system or functional unit to perform as expected also under unexpected conditions.
<b>Initiator</b>	WP 1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Robustheit
<b>French Term</b>	--

### 3.162 RTE Event

<b>Definition</b>	An RTEEvent encompasses all possible situations that can trigger execution of a runnable entity (→ definition 3.163) by the RTE. Thus they can address timing, data sending and receiving, invoking operations, call server returning, mode switching, or external events. RTE-Events can either activate a runnable entity or wakeup a runnable entity at it's waitpoints.
<b>Initiator</b>	Stefaan Sonck Thiebaut
<b>Further Explanations</b>	Note 'event' in this context is not necessarily synonymous with 'RTEEvent' as defined in the VFB specification. In particular, RTEEvents that result from communication are handled by communication-triggered runnable entities.
<b>Comment</b>	Events can have a variety of sources including time.
<b>Example</b>	Scheduling of runnable entities from angular position, e.g. a crankshaft, that are used to trigger an interrupt and hence an RTE notification. A software component needs to perform a regular interval, e.g. flash an LED, reset a watchdog, etc.
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.163 Runnable Entity

<b>Definition</b>	A Runnable Entity is a part of an Atomic Software-Component (→ definition 3.11) which can be executed and scheduled independently from the other Runnable Entities of this Atomic Software-Component. It is described by a sequence of instructions that can be started by the RTE (definition→ 3.162). Each runnable entity is associated with exactly one Entry Point (definition→ 3.63).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A Runnable Entity contains at least two points for the Scheduler (→ definition 3.167): 1 Entry Point (→ definition 3.63) and 1 Exit Point (→ definition 3.67). Due to the reason that an Atomic Software Component is not dividable, all its Runnable Entities are executed on the same ECU.
<b>Comment</b>	In general a task in the runtime system consists out of n Runnable Entities of m Atomic Software-Components.
<b>Example</b>	Server function of an AUTOSAR-Software Component.
<b>Reference</b>	--
<b>German Term</b>	Ausführbare Einheit
<b>French Term</b>	entité executable

### 3.164 Safety

<b>Definition</b>	Freedom from unacceptable risk to persons and goods.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	Based on [IEC 61508, Part 4]
<b>German Term</b>	Sicherheit
<b>French Term</b>	--

### 3.165 Sample Application

<b>Definition</b>	Defined system used for evaluation purposes.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The application may be simplified for better understanding within the evaluation phase.
<b>Comment</b>	--
<b>Example</b>	Diagnosis Application Exterior Light Management
<b>Reference</b>	--
<b>German Term</b>	Beispielapplikation
<b>French Term</b>	--

### 3.166 Scalability

<b>Definition</b>	The degree to which assets can be adapted to specific target environments for various defined measures.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	Target environment introduced compared to EAST-Glossary.
<b>Example</b>	--
<b>Reference</b>	[EAST-Glossary]
<b>German Term</b>	Skalierbarkeit
<b>French Term</b>	--

### 3.167 Scheduler

<b>Definition</b>	The scheduler handles the scheduling of the tasks/runnable entities (definition → 3.193 / 3.163) according to the priority and scheduling policy (pre-defined or configurable). It has the responsibility to decide during run-time when which task can run on on the CPU of the ECU.
<b>Initiator</b>	System Team
<b>Further Explanations</b>	There are many strategies (priority-based, time-triggered, round-robin, ...) a scheduler can use, depending of the selected and/or implemented algorithms
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	--
<b>French Term</b>	--



### 3.168 SDU

<b>Definition</b>	SDU is the abbreviation of "Service Data Unit". It is the data passed by an upper layer, with the request to transmit the data. It is as well the data, which is extracted after reception by the lower layer and passed to the upper layer.
<b>Initiator</b>	WP 1.1.2
<b>Further Explanations</b>	A SDU is part of a PDU (→ definition 3.139).
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.169 Security

<b>Definition</b>	Protection of data, software entities or resources from accidental or malicious acts.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	Slightly adapted norm.
<b>Example</b>	--
<b>Reference</b>	[ISO 2382-8]
<b>German Term</b>	Sicherheit
<b>French Term</b>	--

### 3.170 Sender-Receiver Communication

<b>Definition</b>	A communication pattern which offers asynchronous distribution of information where a sender communicates information to one or more receivers, or a receiver receives information from one or several senders.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The process of sending data does not block the sender and the sender usually gets no response from the receivers
<b>Comment</b>	Often used for data or event distribution
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	Sender-Empfänger Kommunikation
<b>French Term</b>	--

### 3.171 Sender-Receiver Interface

<b>Definition</b>	A sender-receiver interface is a special kind of port-interface (→ definition 3.144) used for the case of sender-receiver communication (→ definition 3.170). The sender-receiver interface defines the data-elements which are sent by a sending component (which has a p-port providing the sender-receiver interface) or received by a receiving component (which has an r-port requiring the sender-receiver interface).
<b>Initiator</b>	Stefaan Sonck Thiebaut
<b>Further Explanations</b>	--
<b>Comment</b>	A special kind of Port-Interface
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	Sender-Empfänger Schnittstelle
<b>French Term</b>	--

### 3.172 Sensor/Actuator SW-Component

<b>Definition</b>	AUTOSAR SW-Component (→ definition 3.18) dedicated to the control of a sensor or actuator.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	There will be several Sensor/ Actuator SW-Cs in each ECU. In general there will be one Sensor/Actuator SW-C for each sensor and one for each actuator (=> number of Sensor/Actuator SW-C = number of sensors + number of actuators).
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Sensor/Aktuator Software Komponente
<b>French Term</b>	Composant Logiciel de traitement des Capteurs/Actionneurs

### 3.173 Server

<b>Definition</b>	Software entity which provides services for clients (→ definition 3.31).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The server (→ definition 3.173) and the clients using its service might be located on one ECU or distributed on different calculation units (e.g. ECU).
<b>Comment</b>	Adapted from Balzert.
<b>Example</b>	--
<b>Reference</b>	[Balzert99]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.174 Service

<b>Definition</b>	A service is a type of operation that has a published specification of interface and behavior, involving a contract between the provider of the capability and the potential clients.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Diagnosis service, ...
<b>Reference</b>	[EAST-Glossary]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.175 Service Port

<b>Definition</b>	A Service Port is a Port (→ definition 3.143) of an AUTOSAR SW-C (→ definition 3.18), Complex Device Driver (→ definition 3.37) and/or ECU Abstraction (→ definition 3.57) connected to an AUTOSAR Service (→ definition 3.17).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The interface of a Service Port has to be a Standardized AUTOSAR Interface (→ definition 3.14 and 3.184). A Service Port does not need to be connected to another Port in the VFB View (→ definition 3.203).
<b>Comment</b>	If a service is provided by the ECU where a specific Atomic Software Component is located the VFB View is sufficient. If a service is provided by another ECU the connection of the service call to the service has to be done explicitly during the mapping step.
<b>Example</b>	Write data to non volatile memory.
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.176 Services Layer

<b>Definition</b>	The Services Layer is the highest layer of the Basic Software which also applies for its relevance for the application software: while access to I/O signals is covered by the Hardware Abstraction Layer, the Services Layer offers Operating system services Vehicle network communication and management services Memory services (NVRAM management) Diagnosis Services (including KWP2000 interface and error memory) ECU state management  Task: Provide basic services for application and basic software modules
<b>Initiator</b>	WP 1.1.2
<b>Further Explanations</b>	The Services Layer consists of the following parts: Communication Services Memory Services System Services
<b>Comment</b>	--
<b>Example</b>	Network Management, NVRAM Manager, ECU State Manager
<b>Reference</b>	[AUTOSAR Software Architecture]
<b>German Term</b>	Dienste-Schicht
<b>French Term</b>	--

### 3.177 Shipping

<b>Definition</b>	Component shipment refers to the action of a supplier releasing a software component (also a composition) to the system integrator. The integrator will gather all shipments of the components that make up the whole system and then map them to ECUs.
<b>Initiator</b>	WP 2.1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Auslieferung
<b>French Term</b>	--

### 3.178 Software Configuration

<b>Definition</b>	The arrangement of software elements in a SW system.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A software element is a clearly definable software part. A software configuration is a selection version of software modules, AUTOSAR software components, parameters and generator configurations. Calibration and Variant Coding (→ definition 3.200) can be regarded as subset of Software Configuration.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[EAST-Glossary]
<b>German Term</b>	Software Konfiguration
<b>French Term</b>	--

### 3.179 Software Interrupt

<b>Definition</b>	Interrupt triggered by SW event.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	SW events are for example calling an operating system service, starting a process with higher priority.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	Translation/Adaptation from [VDI Lexikon]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.180 Software Module

<b>Definition</b>	A collection of software files (code and description) that define a certain software functionality present on an ECU.
<b>Initiator</b>	WP 4.1.1.2
<b>Further Explanations</b>	A software module may be an Atomic SW-C (→ definition 3.11), or a Basic Software module (→ definition 3.24), or the RTE (RunTimeEnvironment).
<b>Comment</b>	This term shall be used if both basic software modules and Application software Components are addressed.
<b>Example</b>	A Digital IO Driver, Complex Device Driver, OS are examples of software modules.
<b>Reference</b>	--
<b>German Term</b>	Software-Module
<b>French Term</b>	--

### 3.181 Software Signal

<b>Definition</b>	A Software Signal is an asynchronous event transmitted between one process and another.
<b>Initiator</b>	WP1.1.1
<b>Further Explanations</b>	A SW Signal is the software implementation of an (control-) information. Additionally it may have attributes (e.g. freshness, data type, ...). It is exchanged between SW-Components.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.182 Special Periphery Access

<b>Definition</b>	Special functions to standard peripheral devices or special peripherals.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Is only used when, because of technical issues, no standard periphery access can be used
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[HIS API IO Driver version 2.1] Interaction with [Hardware v0.3]
<b>German Term</b>	Spezielle Peripherie
<b>French Term</b>	--

### 3.183 Standard Periphery Access

<b>Definition</b>	Standard functions to typical standard peripheral devices that is available on an ECU (most microcontroller integrated) used in automotive embedded applications.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	Digital Input/Output, Analog/Digital Converter, Pulse Width (De)Modulator, EEPROM, FLASH, Capture Compare Unit, Watchdog Timer
<b>Reference</b>	[HIS API IO Driver version 2.1] Interaction with Hardware v0.3
<b>German Term</b>	Standard Peripherie
<b>French Term</b>	--

### 3.184 Standardized AUTOSAR Interface

<b>Definition</b>	This is an AUTOSAR Interface which is standardized within the AUTOSAR project.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	AUTOSAR Services interact with other components through a Standardized AUTOSAR Interface.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.185 Standardized Interface

<b>Definition</b>	A software interface is called Standardized Interface if a concrete standardized API exists.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Modules in the Basic Software interact which each other through Standardized Interfaces.
<b>Comment</b>	--
<b>Example</b>	OSEK COM Interface
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.186 Standard Software

<b>Definition</b>	Standard Software is software which provides schematic independent infrastructural functionalities on an ECU. It contains only Standardized Interfaces (→ definition 3.185), Standardized AUTOSAR Interfaces (→ definition 3.184) and/or Private Interfaces (→ definition 3.147).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	OSEK COM, MCAL, Services, OSEK OS
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.187 Static Configuration

<b>Definition</b>	A setup where the routing configuration cannot be changed during normal operation of the gateway.
<b>Initiator</b>	WP4.2.2.1.6 Gateway
<b>Further Explanations</b>	Static configuration doesn't allow reconfiguration of the routing during normal operation e.g. during driving. Static configuration does not restrict the update of the configuration in specific maintenance operation modes (e.g. programming mode).
<b>Comment</b>	--
<b>Example</b>	A software update may change a routing configuration such that a PDU is routed into two instead of one destination networks.
<b>Reference</b>	--
<b>German Term</b>	Statische Konfiguration
<b>French Term</b>	--

### 3.188 Surveillance

<b>Definition</b>	Systematic iteration of a reduced full-set assessment (→ definition 3.8) as a basis for maintaining the validity of the full-set assessment.
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	Within AUTOSAR context, reduced set surveillances of the accredited party are scheduled yearly between the full-set assessments.
<b>Comment</b>	Supplementary full-set assessments are scheduled.
<b>Example</b>	--
<b>Reference</b>	[ISO/IEC 17000]
<b>German Term</b>	Überwachung
<b>French Term</b>	Surveillance

### 3.189 Synchronous Communication

<b>Definition</b>	A communication is synchronous when the calling software entity is blocked until the called operation is evaluated. The calling software entity continues its operation by getting the result.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	Synchronous communication between distributed functional units has to be implemented as remote procedure call.
<b>Comment</b>	Are further mechanisms possible?
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.190 System

<b>Definition</b>	An integrated composite that consists of one or more of the processes, hardware, software, facilities and people, that provides a capability to satisfy a stated need or objective.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	ITEA EAST uses IEEE 14407 standard. Here not applicable because of problem with the definition of function. One correct interpretation is: - it might be a composition of one or more ECUs
<b>Example</b>	Braking system
<b>Reference</b>	[ISO 12207]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.191 System Constraint

<b>Definition</b>	Boundary conditions that restrict the Design-Freedom of the (cars E/E-) System.
<b>Initiator</b>	WP 2.1.1.3
<b>Further Explanations</b>	The design of ECU Networks and the distribution of functionalities to ECUs are limited by several constrains. These constraints result mostly by the communication matrix and safety requirements
<b>Comment</b>	--
<b>Example</b>	An existing communication matrix that restricts the distribution of signals to frames is a system constraint. Another system constraint is a safety requirement that does not allow to map a specified Software component to specific ECU.
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--



### 3.192 System Signal

<b>Definition</b>	The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with (at least) one system signal defined for each data element sent or received by a SW component instance. If data has to be sent over gateways, there is still only one system signal representing this data. The representation of the data on the individual communication systems is done by the cluster signals.
<b>Initiator</b>	WP2.1.1.3
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.193 Task

<b>Definition</b>	A Task is the smallest scheduleable unit managed by the OS. The OS decides when which task can run on the CPU of the ECU.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	A runnable entity (→ definition 3.163) of a software component runs in the context of a task. Also the Basic Software Modules runs in the context of a task.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.194 Technical Signal

<b>Definition</b>	The technical signal is the physical value of an external event coupled to an AUTOSAR system. Technical signals are represented in SI units (e.g. pressure in PA).
<b>Initiator</b>	WP 2.1.1.2
<b>Further Explanations</b>	The term Technical Signal is used when we are referring to the "real world" signal that is under consideration. So typical Technical Signals are temperature, velocity, torque, force, electrical current and voltage, etc.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.195 Template

<b>Definition</b>	A template is a structured collection of attributes that are required to formally describe AUTOSAR artifacts like e.g. software components or configurations of ECUs.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The term "Template" stresses the fact that the collected attributes still need to have actual values assigned in order to describe a particular artifact. Those values are collected in a Description. Templates are independent of the technology used for serialization of their respective descriptions. Possible serializations include XML, databases tables and so on.
<b>Comment</b>	--
<b>Example</b>	The templates defined by AUTOSAR are represented as an UML2.0 model (→ definition 3.15) and an W3C XML Schema (→ definition 3.21). Models and descriptions created according to the templates can be exchanged using the language defined in the AUTOSAR XML Schema).
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.196 Third party

<b>Definition</b>	An organization that is independent of the organization that provides automotive products, which are subject to AUTOSAR conformance testing.
<b>Initiator</b>	WP 20
<b>Further Explanations</b>	Criteria for the independence of conformity assessment (→ definition 3.8) bodies and accreditation (→ definition 3.1) bodies are provided in the International Standard and Guides applicable to their activities.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	ISO/IEC 17000]
<b>German Term</b>	
<b>French Term</b>	Tierce partie

### 3.197 Timeout

<b>Definition</b>	Notification with respect to deadline violation of an event or task (e.g.while working on/with information: receiving, sending, processing,...).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.198 Use Case

<b>Definition</b>	A model of the usage by the user of a system in order to realize a certain functional feature of the system.
<b>Initiator</b>	WP 1.1.1

<b>Further Explanations</b>	--
<b>Comment</b>	Added certain compared to EAST-glossary.
<b>Example</b>	--
<b>Reference</b>	[EAST-Glossary]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.199 Validation

<b>Definition</b>	Confirmation by examination and provision of objective evidence that the particular requirements of a specific intended use are fulfilled
<b>Initiator</b>	System Team
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[IEC 61508]
<b>German Term</b>	Validierung
<b>French Term</b>	--

### 3.200 Variant Coding

<b>Definition</b>	Adaptation of SW by selection of functional alternatives according to external requirements (e.g. country-dependent or legal restrictions).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The major difference with calibration is that this later doesn't aim to adapt the SW functionality itself but only aims to adjust the SW to the HW/SW environment, e.g. the calibration of engine control SW that is adjusted to the physical parameters of every engine. Variant Coding also includes vehicle-specific (not user-specific) SW adaptation due to end-customer wishes (e.g. deactivation of speed-dependent automatic locking). Variant Coding is always done after compile time. Used techniques to select variants include end-of-line programming and garage programming.
<b>Comment</b>	--
<b>Example</b>	Country related adaptation of MMI with respect to speed and/or temperature unit (km/h vs. mph, °C vs. F).
<b>Reference</b>	--
<b>German Term</b>	Variantenkodierung
<b>French Term</b>	--

### 3.201 Vendor ID

<b>Definition</b>	A vendor ID is a unique identification of the vendor of a software component. All basic software modules (→definition 3.24) conformant to the AUTOSAR standard shall provide a readable vendor ID.
<b>Initiator</b>	WP1.1.2
<b>Further Explanations</b>	AUTOSAR Vendor IDs are used to determine vendors of basic software modules before and during runtime. The mechanism is used to improve bug handling. AUTOSAR currently only provides Vendor IDs to members of the AUTOSAR partnership.
<b>Comment</b>	To apply for an AUTOSAR vendor ID the possible member has to send an E-Mail to request@autosar.org. Within the request name of the company, company address and contact person should be listed.

	In order to keep administrative overhead low, AUTOSAR and the HIS initiative run a joint list of vendor IDs.
<b>Example</b>	Vendor ID for EEPROM driver is called: EEP_VENDOR_ID
<b>Reference</b>	BSW00374
<b>German Term</b>	Lieferanten Identifikationsnummer
<b>French Term</b>	Numéro d'identification de fournisseur

### 3.202 Verification

<b>Definition</b>	Confirmation by examination and provision of objective evidence that the requirements have been fulfilled.
<b>Initiator</b>	System Team
<b>Further Explanations</b>	--
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[IEC 61508]
<b>German Term</b>	Verifikation / Überprüfung
<b>French Term</b>	--

### 3.203 VFB View

<b>Definition</b>	The VFB View describes systems or subsystems in the car independently of these resources; in other words, independently of: <ul style="list-style-type: none"> <li>• what kind of and how many ECUs are present in the car</li> <li>• on what ECUs the entities in the VFB-View run</li> <li>• how the ECUs are interconnected: what kind of network technology (CAN, LIN,...) and what kind of topology (presence of gateways) is used</li> </ul>
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	In the VFB-View, the system or subsystem under consideration is a Composition which consists out of Connectors and Components.
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	[AUTOSAR Specification of Virtual Functional Bus]
<b>German Term</b>	--
<b>French Term</b>	--

### 3.204 Virtual Integration

<b>Definition</b>	The simulated, modeled and/or calculated (not real) combination of software entities forming a system (→ definition 3.190).
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	By virtual integration several constraints and/or requirements are checked without the need of real hardware units, like needed CPU load, needed memory, completeness of interfaces, fulfillment of timing requirements etc.).
<b>Comment</b>	--
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	Virtuelle Integration
<b>French Term</b>	--

### 3.205 Worst Case Execution Time

<b>Definition</b>	Maximum possible time during which a program is actually executing
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The worst case execution time of a piece of software is the maximum possible time during which the CPU is executing instructions which belong to this piece. The worst case execution time is often identified by analytical methods. It is required to determine if a schedule meets the overall timing requirements. Abbreviation: WCET See also: response time, execution time, worst case response time
<b>Comment</b>	This definition has been extended by 2.1.1.1
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

### 3.206 Worst Case Response Time

<b>Definition</b>	Maximum possible time between receiving a stimulus and delivering an appropriate response or reaction.
<b>Initiator</b>	WP 1.1.1
<b>Further Explanations</b>	The worst case response time describes the maximum possible time between a stimulus like e.g. the state change of hardware or software entity and the expected reaction of the system (e.g. response, actuator activation). Typically: worst-case execution-time + infrastructure-overhead + scheduling-policy = worst-case reaction time Synonym: worst case reaction time See also: response time, execution time, worst case execution time
<b>Comment</b>	Worst case reaction time was renamed to worst case response time because response time is the more common terminology. This definition has been extended by 2.1.1.1.
<b>Example</b>	--
<b>Reference</b>	--
<b>German Term</b>	--
<b>French Term</b>	--

## Annex 1: Literature

**[Balzert99]**, Balzert, H. "Lehrbuch Grundlagen der Informatik"  
Spektrum Verlag, Heidelberg, 1999

**[DCE-IDL]**, Remote Procedure Call  
<http://www.opengroup.org/onlinepubs/9629399/chap1.htm>

**[EAST-Glossary]**, ITEA Project 00009 EAST-EEA Embedded Electronic Architecture  
"Glossary", Version 6.1, 2003

**[EN 50126]**, European Norm "EN 50126 – part 103, March 2000  
Railway applications – The specification and demonstration of Reliability, Availability,  
Maintainability and Safety (RAMS)"

**[Hyper Dictionary]**, Hyper Dictionary, 2003  
[www.HyperDictionary.com](http://www.HyperDictionary.com)

**[IEEE 1471]**, Institute of Electrical and Electronics Engineers, Inc. "IEEE 1471-2000:  
IEEE Recommended Practice for Architectural Description for Software- Intensive  
Systems", 2001

**[IEEE 1517]**, Institute of Electrical and Electronics Engineers, Inc. "IEEE 1517-1999:  
IEEE Standard for Information Technology – Software Life Cycle Processes – Reuse  
Processes", 2000

**[ISO 12207]**, International Standardization Organization "ISO/IEC 12207 Information  
technology – Software life cycle process", first edition, Geneva, 1995

**[ISO 2382-1]**, International Standardization Organization "ISO/IEC 2382 Part 1  
Information technology – Vocabulary – Fundamental Terms", Third Edition, Geneva,  
1993

**[ISO 2382-14]**, International Standardization Organization "ISO/IEC 2382 Part 14  
Information technology – Vocabulary – Reliability, maintainability and availability",  
Second Edition, Geneva, 1997

**[ISO 2382-15]**, International Standardization Organization "ISO/IEC 2382 Part 15  
Information technology – Vocabulary – Programming Languages", First Edition,  
Geneva, 1999

**[ISO 2382-20]**, International Standardization Organization "ISO/IEC 2382 Part 20  
Information technology – Vocabulary – System Development", First Edition, Geneva,  
1990

**[ISO 2382-8]**, International Standardization Organization "ISO/IEC 2382 Part 1  
Information technology – Vocabulary – Security", Second Edition, Geneva, 1998

**[ISO 61511-1]**, International Standardization Organization “ISO/IEC 61511 Part 1 Information technology – Software life cycle process”, First Edition, Geneva, 1995

**[OMG-IDL]**,  
[http://www.omg.org/technology/documents/formal/corba\\_2.htm](http://www.omg.org/technology/documents/formal/corba_2.htm)

**[OSEK BD]**, OSEK/VDX “Binding Specification”, Version 1.4.1, January 2003

**[OSEK Com]**, OSEK/VDX “Communication”, Version 3.0.1, January 2003

**[IEC 61508, Part 4]**, Functional safety of electrical/electronic/programmable electronic safety-related systems; Part 4: Definitions and abbreviations

**[ISO 7498]**, Information processing systems -- Open Systems Interconnection -- Basic Reference Model

**[DIN 40041]**, DIN 40041 Ausgabe:1990-12 Zuverlässigkeit; Begriffe  
Deutsche Industrie Norm

**[VDI Lexikon]**, Translation/Adaptation from VDI Lexikon Informatik und Kommunikationstechnik,  
Springer Verlag, Berlin 1999,

**[HIS API IO Driver V 2.1]**,  
<http://www.automotive-his.de/download>  
API\_IODriver\_2\_1\_3.pdf

**[IEEE.610.12-1990]**, IEEE Standard Glossary of Software Engineering Terminology;  
ISBN 1-55937-067-X, SH13748

**[UML 2.0]** Unified Modeling Language  
Superstructure, Version 2.0, OMG Available Specification, ptc/05-07-04.  
<http://www.omg.org/cgi-bin/apps/doc?formal/05-07-04.pdf>

**[IEEE 1394]** Firewire, see “isochronous stream”  
<http://www.1394ta.org/Technology/Specifications/specifications.htm>

**[FR\_PROTOCOL]** FlexRay Communications System Protocol Specification V2.1  
<http://www.flexray.com/>