VDA German Association of the Automotive Industry

Position

Automated Valet Parking Systems

Requirements for automated valet parking systems

Description of the document

Automated Valet Parking is a functionality, which maneuvers vehicles automatically in parking garages to their designated parking lot. Communication is necessary between the vehicle and the AVP infrastructure in the parking facility. This document describes – additionally to the standard ISO 23374 AVPS – a concept for communication at the corresponding signals to ensure the interoperability between vehicles of different manufacturers and different providers of AVP – infrastructure.

Disclaimer

The VDA Recommendations are recommendations that may be freely adopted by anyone. Users are responsible for correct implementation of the recommendations as required on a case-by-case basis.

The recommendations take into account the prevailing technology at the time of publication. Use of the VDA Recommendations does not absolve anyone from responsibility for his/her own actions, and all users act at their own risk. Liability of VDA for this recommendation is excluded.

Content

Normative references 6

List of figures 7

List of tables 8

List of requirements 9

1 Scope 12
1.1 Structure of the document 12

2 Introduction 14
2.1 Glossary 14
2.2 Definitions 16
2.3 Interface-Specification Version 28

3 System framework regarding ISO 23374 30
3.1 System configuration 30
3.2 Classification AVP Type 2 31
3.3 Traffic environment categories 32
3.3.1 Mixed traffic 32
3.3.2 Exclusive traffic 32
3.4 Human interaction 32
3.4.1 Service provider 32
3.4.2 System operator 33
3.4.3 Facility manager 33

4 Requirements for the overall AVP system 34
4.1 Requirements for management functions 34
4.1.1 General 34
4.1.2 Remote engagement 34
4.1.3 Operation stop 35
4.1.4 Remote assistance 35
4.1.5 Remote disengagement 35
4.1.6 Central control 36
4.1.7 Response to incapacitation of the operation functions 36
4.1.8 Maintaining environmental conditions 36
4.2 Requirements for the environment within parking facilities 37
4.2.1 Operation zone 37
4.2.2 Drop-off and pick-up area 37
4.2.3 Wireless communication 37
4.2.4 Operation stop device 37
4.2.5 Lighting 38
4.2.6 Detection capabilities 39
4.2.7 Digital maps 39
4.2.8 Traffic environment category dependent requirements 39
4.2.8.1 Mixed traffic 39
4.2.8.2 Exclusive traffic 40

4.3 Requirements for the overall system operation 40
4.3.1 Requirements for the communication interface 40
4.3.1.1 General requirements 40
4.3.1.2 Security goals 41
4.3.1.3 Security requirements 42
4.3.2 Information to the user 42
4.4 Development process and management 42

5 Subject Vehicle and Vehicle Backend Requirements 43
5.1 General Requirements 43
5.2 Safety Requirements 45
5.3 Security Requirements 48
5.4 User Communication 50

6 Remote Vehicle Operation and Operator Backend Requirements 52

7 The AVP Mission Process 54
7.1 Mission Initialization 57
7.1.1 New Mission 57
7.1.2 Wake-Up Subject Vehicle 58
7.1.3 Certificate and Connection Parameter Exchange 59
7.1.4 Wi-Fi Connection 61
7.1.5 TLS/DTLS Connection 64
7.1.5.1 Establish DTLS through TLS connection 66
7.1.5.2 Heartbeat 67
7.1.6 Vehicle Functional Clock Synchronization 68
7.1.7 Mission Confirmation 69
7.1.8 Safe Vehicle Identification 70
7.1.9 Vehicle Safety Clock Synchronization 74
7.1.10 Safe Vehicle Type Confirmation 79
7.1.11 Enter Safe Driving State 80
7.2 Safe Driving 83
7.2.1 General requirements for automated vehicle operation functions 84
7.2.1.1 Principles for performing automated vehicle operation 84
7.2.1.2 Operational design domain 84
7.2.1.3 Requirements for Dynamic Driving Task (DDT) 84
7.2.1.4 Requirements for emergency stopping 87
7.2.1.5 Requirements for destination assignment 87
7.2.1.6 Requirements for route planning 88
7.2.2 Requirements to sub-systems 88
7.2.3 Control: Path Interface (Type 2.1) 92
7.2.3.1 Definitions 93
7.2.3.2 Initial Path and Path Updates 95
7.2.3.3 Following the Path 97
7.2.3.4 Stopping at the end of a PathSnippet 99
7.2.3.5 Temporarily limit velocity 100
7.2.3.6 Temporarily stop (Comfort Stop) 101

7.2.4 Control: Acceleration/Curvature Interface (Type 2.2/ 2.3) 101
7.2.4.1 Definitions 102
7.2.4.2 Type 2.2 interface 106
7.2.4.3 Type 2.3 interface 107
7.2.5 Safety: Driving Permission 108
7.2.5.1 Driving Direction 110
7.2.5.2 Velocity 110
7.2.5.3 Curvature 111
7.2.5.4 Expiration Time 112
7.3 Wrap Up Mission and Shutdown Subject Vehicle 122
7.3.1 Abort Mission 124
7.4 Check-out 124
7.5 Safety Checksums 124

8 Infrastructure - Vehicle – Connection 128
8.1 Overview 128
8.1.1 Link and Network layer 128
8.1.1.1 Wi-Fi 128
8.1.1.2 Cellular Network 128
8.1.2 Transport to Presentation layer 129
8.2 AVP-Message Data Protocol 129
8.2.1 Header 130
8.2.2 Fundamental Types 131
8.2.3 Complex Types 132
8.2.3.1 String 132
8.2.3.2 Buffer 132
8.2.3.3 Vector 133

A Vehicle Type Identifier 134

B Vehicle Error Codes 136

C AVP Messages 138
C.1 AccessPointChangeRequest 139
C.2 DetectedVehiclePose 140
C.3 DriveCommand 141
C.4 DrivingPermission 142
C.5 DtlsInterfaceRequest 143
C.6 DtlsInterfaceResponse 144
C.7 FunctionalTimeSyncRequest 145
C.8 FunctionalTimeSyncResponse 146
C.9 Heartbeat 147
C.10 InterfaceSpecificationVersion 148
C.11 MissionConfirmation 149
C.12 PathSnippet 150
C.13 RecordedMessage 151
C.14 RecordedMessages 152
C.15 SafetyTimeSyncRequest 153

C.16 SafetyTimeSyncResponse 154
C.17 SafeVehicleTypeConfirmation 155
C.18 VehicleCapabilities 156
C.19 VehicleDebug 158
C.20 VehicleError 159
C.21 VehicleInfo 160
C.22 VehicleSafetyFeedback 161
C.23 VehicleState 162
C.24 VehicleTrajectoryCommand 163
C.25 VidRequest 164
C.26 VidResponse 165

D AVP Enums 166
D.1 ControlInterfaceType 166
D.2 DirectionIndicator 166
D.3 DriveCommandAction 167
D.4 DrivingDirection 167
D.5 DtlsInterfaceRequestState 168
D.6 DtlsInterfaceResponseState 168
D.7 RecordingLevel 169
D.8 RecordingState 169
D.9 SafetyStopReason 170
D.10 TerminateReason 171
D.11 VehicleOperationMode 172
D.12 VidRequestState 173
D.13 VidVehicleState 174

E Certificate Chain 175
E.1 AVP_Vehicle_Controller_CA 176
E.2 Parking Facility Certificates 177
E.3 Vehicle Certificate 178

F Complete Sequence of AVP Mission Process 179

G Subject Vehicle system states and transition diagram 183

H Additional Information 186
H.1 PathSnippet update frequency 186
H.2 PathSnippet minimum length 187
H.2.1 Emergency stop 188
H.2.2 Lateral control (comfort driving) 189
H.2.3 Summary 190
H.3 Trajectory minimum length 190
H.4 Minimum allowed curvature 191

Normative references

[1] ISO, ISO 23374-1:2021 (E): AVPS, 2021.

[2] ISO, ISO 21434, 2021.

List of figures

Figure 1 – AVPS Participants 16
Figure 2 – parking facility coordinate system 23
Figure 3 – Curvatures in the AVP system 25
Figure 4 – Vehicle Clocks 26
Figure 5 – System architecture [1] 30
Figure 6 – Relationship of the operation functions [1] 34
Figure 7 – Illuminance measurement on horizontal and vertical surfaces 38
Figure 8 – Mission sequence overview 55
Figure 9 – AVPS state machine 56
Figure 10 – New Mission 57
Figure 11 – Provisioning/ Certificate Exchange Sequence 59
Figure 12 – TLS and DTLS connection process 64
Figure 13 – FunctionalTimeSync mechanism 68
Figure 14 – Mission Confirmation Sequence 69
Figure 15 – Sequence vehicle-id process 71
Figure 16 – Indicator activation for an example code realized with variant a) 73
Figure 17 – SafetyTimeSync mechanism 75
Figure 18 – SafetyTimeSync Time Budget 78
Figure 19 – Enter Safe Driving State Sequence 80
Figure 20 – Separation of Function and Safety 83
Figure 21 – A typical Path with three segments 93
Figure 22 – Relation between path and Subject Vehicle when receiving a new path 95
Figure 23 – Points (red) at end of each PathSnippet on which the Subject vehicle shall stop on its own responsibility 99
Figure 24 – Temporarily limit velocity 100
Figure 25 – Temporarily stop (Comfort stop) 101
Figure 26 – Control loop type 2.2 and 2.3 105
Figure 27 – Trajectory element selector block 106
Figure 28 – Driving allowed: current curvature is within bounds of DrivingPermission 111
Figure 29 – Safety Violations: (left) current curvature is not within bounds. (right) velocityMaxForwards=0 112
Figure 30 – Nominal Time Budgets and Reaction Time 114
Figure 31 – Worst-Case Time Budgets and Reaction Time if Wi-Fi is interrupted 115
Figure 32 – Mission WrapUp Sequence 122
Figure 33 – Certificate Chain for Communication with Remote Vehicle Operation 175
Figure 34 – Complete Sequence of AVP Mission Process (Part 1) 180
Figure 35 – Complete Sequence of AVP Mission Process (Part 2) 181
Figure 36 – Complete Sequence of AVP Mission Process (Part 3) 182
Figure 37 – State transition diagram [1] 185
Figure 38 – Update times of planned path 186
Figure 39 – Behavior with further existing path 188
Figure 40 – Time distribution in critical situation 188
Figure 41 – Required path length for comfortable driving 190

List of tables

Table 1	– Addressee of Requirement	14
Table 2	– Glossary	14
Table 3	– Definitions [1]	16
Table 4	– Table of variables	28
Table 5	– Functional allocation [1]	31
Table 6	– Maximum braking distance in dependence on the initial velocity	46
Table 7	– Indicator states	72
Table 8	– Safety timing budgets in AVP-System	113
Table 9	– Fundamental Types	131
Table 10	– Set of parameters which defines a unique vehicle type identifier	135
Table 11	– Vehicle Error Codes to the reaction of AVP-Infrastructure	136
Table 12	– Fields of the AVP_Vehicle_Controller_CA intermediate certificates	176
Table 13	– Fields of the end entity certificates used by parking facilitie	177
Table 14	– Fields of the end entity certificates used by AVP vehicles	178
Table 15	– Description of Subject Vehicle State [1]	183
Table 16	– Example values for path update frequency	187

List of requirements

Req1: [General] NegotiateInterfaceSpecificationVersion 28
Req2: [General] ConfirmInterfaceSpecificationVersion 28
Req3: [Vehicle] VehicleRequiredCapabilities 43
Req4: [VehicleBackend] VehicleBackendWakesUpECUs 43
Req5: [Vehicle] VehicleChassisConfiguration 44
Req6: [Vehicle] VehicleSupportsFastWiFiRoaming 44
Req7: [Vehicle] VehicleMaintainsFunctionalClock 44
Req8: [VehicleBackend] VehicleBackendChecksAVPPreconditions 45
Req9: [Vehicle] VehicleHara 45
Req10: [Vehicle] VehicleFailureAnalysis 46
Req11: [Vehicle] VehicleBrakingPerformance 46
Req12: [Vehicle] VehiclePfh 47
Req13: [Vehicle] VehicleFTTI 47
Req14: [General] FreedomFromInterference 47
Req15: [Vehicle] VehicleMaintainsSafetyClock 48
Req16: [Vehicle] VehicleSecConceptforAVPIntegration 48
Req17: [Vehicle] VehicleEnsuresSufficientSecurity 49
Req18: [Vehicle] VehicleSecureOnboardCommunication 49
Req19: [Vehicle] VehicleInspectSecurityConcept 49
Req20: [VehicleBackend] VehicleBackendInspectSecurityConcept 49
Req21: [Vehicle] VehicleInspectPenTestResults 49
Req22: [VehicleBackend] VehicleBackendInspectPenTestResults 50
Req23: [Vehicle] VehicleInformSecurityBreach 50
Req24: [VehicleBackend] VehicleBackendInformSecurityBreach 50
Req25: [Vehicle] VehicleSynchronizeSecureClock 50
Req26: [App] AppGuideNoChangesOfVehicleOutline 50
Req27: [App] AppUserConfirmsEmptyVehicle 51
Req28: [Rvo] RvoHara 52
Req29: [Rvo] RvoPfh 52
Req30: [Rvo] RvoFailureAnalysis 52
Req31: [Rvo] RvoSynchronizeToWorldClock 52
Req32: [Rvo] RvoInspectSecurityConcept 53
Req33: [Rvo] RvoInspectPenTestResults 53
Req34: [Rvo] RvoInformSecurityBreach 53
Req35: [OperatorBackend] OperatorBackendCreateMissionId 57
Req36: [Vehicle] VehicleCreatePrivateKeyAndSigningRequest 60
Req37: [VehicleBackend] VehicleBackendCreateVehicleCertificate 60
Req38: [General] UseSeparateVehicleCertificates 61
Req39: [General] AgreeOnTrustAnchorExchangeAndRevocationProcess 61
Req40: [Vehicle] VehicleConnectToWifi 62
Req41: [Vehicle] VehicleChangeAccessPoint 63
Req42: [Vehicle] VehicleConnectTlsAndDtls 65
Req43: [Vehicle] VehicleSupportsInternetProtocols 65
Req44: [Rvo] RvoProcessTlsAndDtlsHandshake 66
Req45: [Vehicle] VehicleAmountOfDtlsConnectionAttempts 66
Req46: [Vehicle] VehicleDisableTcpNagleOption 67
Req47: [Vehicle] VehicleSendHeartbeatOnTlsAndDtls 67
Req48: [Vehicle] VehicleCloseOnMissingHeartbeat 67

Req49: [Vehicle] VehicleReconnectConditions 67
Req50: [Rvo] RvoDetermineVehicleFunctionalClock 68
Req51: [Vehicle] VehicleRespondToFunctionalTimeSyncRequest 69
Req52: [Rvo] RvoSendsMissionConfirmation 69
Req53: [Vehicle] VehicleVerifiesMissionId 70
Req54: [Vehicle] VehicleProvidesCapabilities 70
Req55: [Vehicle] VehicleVehIdReportReadyToStart 71
Req56: [Vehicle] VehicleVehIdFlashLights 72
Req57: [Vehicle] VehicleVehIdIndicatorStates 72
Req58: [Vehicle] VehicleVehIdSyncBit 72
Req59: [Vehicle] VehicleVehIdIndicatorTime 73
Req60: [Vehicle] VehicleVehIdReportCompleted 73
Req61: [Vehicle] VehicleVehIdRememberCode 74
Req62: [Vehicle] VehicleVehIdRepeat 74
Req63: [Rvo] RvoDetermineOffsetToVehicleSafetyClock 75
Req64: [Rvo] RvoDetermineSafetyTimeSyncRoundTripTime 76
Req65: [Rvo] RvoDetermineSafetyTimeSyncUncertainty 77
Req66: [Rvo] RvoDetermineCurrentVehicleSafetyClockTime 77
Req67: [Vehicle] VehicleRespondToSafetyTimeSyncRequest 77
Req68: [Rvo] SafetyTimeSyncRoundTripTimeBudget 78
Req69: [Vehicle] VehicleProvidesVehicleTypeIdentifier 79
Req70: [Rvo] RvoConfirmsSafeDrivingState 80
Req71: [Vehicle] VehicleEntersSafeDrivingState 81
Req72: [Vehicle] VehicleLeavesSafeDrivingState 81
Req73: [Vehicle] VehicleVerifiesPreconditionsBeforeEnteringSafeDrivingState 81
Req74: [Vehicle] VehiclePreventsAccessFromOutsideInSafeDrivingState 82
Req75: [Vehicle] VehicleAllowsDoorOpeningFromInsideInSafeDrivingState 82
Req76: [Vehicle] VehicleActivateActuators 83
Req77: [Vehicle] VehicleActivatesBrakeLight 88
Req78: [Vehicle] VehicleActivatesReverseLight 88
Req79: [Vehicle] VehicleActivatesIndicatorLightsOnRequest 89
Req80: [Vehicle] VehicleActivateHeadlights 89
Req81: [Vehicle] VehicleSuppressHighBeamLights 89
Req82: [Vehicle] VehicleDataLogging 89
Req83: [Vehicle] VehicleDataLoggingToInfrastructure 90
Req84: [Rvo] RvoDataLogging 90
Req85: [Vehicle] VehicleDetectsHumanIntervention 91
Req86: [Vehicle] VehicleLimitsSpeedInSafeDrivingState 92
Req87: [Vehicle] VehicleAbortsMissionOnCriticalVehicleFailure 92
Req88: [Vehicle] VehicleDoesNotUseRearAxleSteering 92
Req89: [General] AvpMessagePose 94
Req90: [General] AvpMessagePathPose 94
Req91: [Rvo] RvoSendsPath 95
Req92: [Vehicle] VehicleReceivesPath 96
Req93: [Rvo] RvoLimitSizeOfPathSnippet 96
Req94: [Rvo] RvoLimitFrequencyOfPathUpdates 97
Req95: [Rvo] RvoLimitCurvatureInPath 97
Req96: [Rvo] RvoSendsCurrentPose 97

Req97: [Vehicle] VehicleReceivesCurrentPose 98

Req98: [Vehicle] VehicleFollowsPath 98

Req99: [Vehicle] VehicleProvidesSpeedControl 99

Req100: [Vehicle] VehicleStopsAtEndOfPathSnippet 100

Req101: [General] AvpMessageVehicleTrajectory 102

Req102: [Vehicle] VehicleChoosesTrajectoryInterval 102

Req103: [Vehicle] VehicleChoosesTrajectoryDuration 103

Req104: [General] AvpMessageControlTrajectoryElement 104

Req105: [General] AvpMessageStateTrajectoryElement 104

Req106: [Vehicle] VehicleProvidesTrajectoryElementSelector 105

Req107: [Vehicle] VehicleExecutesTrajectoryCommand 106

Req108: [Vehicle] VehicleTransposesRequiredAcceleration 107

Req109: [Rvo] RvoInitiatesTemporaryStop 107

Req110: [Vehicle] VehicleStopsAtEndOfTrajectory 107

Req111: [Vehicle] VehicleEvaluateDrivingPermission 108

Req112: [Rvo] RvoPreventsCollisions 109

Req113: [Vehicle] VehicleStopsOnWrongDrivingDirection 110

Req114: [Vehicle] VehicleStopsOnVelocityViolation 110

Req115: [Vehicle] VehicleStopsOnCurvatureViolation 111

Req116: [Rvo] RvoStopOnUnknownVehicleSafetyClock 116

Req117: [Rvo] RvoDrivingPermissionExpirationTime 116

Req118: [Rvo] RvoDrivingPermissionTimeBudget 117

Req119: [Vehicle] VehicleDrivingPermissionTimeBudget 118

Req120: [Vehicle] VehicleStopsOnExpirationTimeViolation 119

Req121: [Vehicle] VehicleStopsOnExpirationTimeTooHigh 120

Req122: [Vehicle] VehicleAbortsMissionAfter10s 121

Req123: [Vehicle] VehicleSecuresOnFinishedMission 123

Req124: [Vehicle] VehicleMissionAbort 124

Req125: [Vehicle] VehicleVerifiesSafetyChecksums 125

Req126: [Rvo] RvoVerifiesSafetyChecksums 125

Req127: [General] CalculateSafetyChecksum 126

Req128: [General] CalculateSafetyChecksumDrivingPermission 127

Req129: [Vehicle] VehicleExternalCommunication 128

Req130: [Rvo] RvoPreventsCommunicationBetweenVehicles 128

Req131: [General] AvpMessageProtocol 129

Req132: [General] AvpMessageSerialization 130

Req133: [General] AvpMessageHeader 130

Req134: [General] AvpMessageFundamentalTypes 131

Req135: [General] AvpMessageByteOrder 132

Req136: [General] AvpMessageString 132

Req137: [General] AvpMessageBuffer 132

Req138: [General] AvpMessageVector 133

1 Scope

This document specifies the detailed communication protocol between the remote vehicle operation and the subject vehicle of an Automate Valet Parking System according to ISO 23374. It covers AVP type 2, which means that the infrastructure guides the vehicle through the garage. The described protocol allows safe and robust level 4 driving of the subject vehicle with up to 10 km/h.

It describes the signals, the communication sequences and the required reactions of the remote vehicle operation and the subject vehicle.

The protocol is designed for an IP based (point-to-point) communication between the remote vehicle operation and the subject vehicle and covers relevant security aspects. With some modifications the protocol should be extensible towards other radio technologies.

Three variants for vehicle motion control are considered, which are called type 2.1, 2.2 and 2.3. In the first variant the motion control is performed by the vehicle itself based on the path and current vehicle pose provided by the infrastructure. In type 2.2, the remote vehicle operation transmits acceleration/curvature values which are directly commanded to the vehicle actuators. Type 2.3 allows an improved path following accuracy by additionally transmitting target poses to the vehicle.

⚠ Please be aware that this document only refers to the document versions mentioned in Normative references. Details about e.g. messages or sequences can vary in other versions.

1.1 Structure of the document

The document begins in chapter 2 by describing used definitions and abbreviations. The system framework is defined in chapter 3. Also, a clear definition of AVP Type 2, mixed and exclusive traffic and occurring human interaction can be found there. Chapter 4 defines requirements for the overall AVP system. It addresses management functions, the physical environment, the overall system operation and in addition guidelines regarding the development process and management of a AVP Type 2 system. The chapters 5 and 6 define requirements regarding the sub-systems vehicle and vehicle backend respectively remote vehicle operation and operator backend.

Following, the complete AVP Process is described in chapter 7 by focusing on the interface between vehicle and remote vehicle operation. It starts by describing how a new mission is initiated. Afterwards, it explains the driving state with a distinction between two fundamentally different approaches on the one hand and the safety concept on the other hand. The chapter ends by defining the process of finishing a mission and an explanation of safety checksums in messages.

The connection between vehicle and infrastructure is described in chapter 8. Beside an overview of link, network, transport and presentation layer it also defines the used data protocol.

Subsequently, additional information is given in the appendix. Firstly, the definitions of the vehicle type identifier and possible vehicle error codes can be found (chapters A and B). Secondly it lists all the required messages and Enums that are exchanged between vehicle and infrastructure (chapters C and D). This is followed by the definition of used certificates in chapter E. The chapters F and G are showing graphically the sequence of the previously described AVP process respectively of the vehicle system states. The document ends with chapter H, where some additional information can be found.

2 Introduction

The chapter explains abbreviations and definitions on the one hand, and hints to the interface-specification version on the other hand.

For the overall AVP system to operate in a correct, safe and secure way, the subsystems Subject Vehicle and Remote Vehicle Operation need to interact in a specified way. This document describes the conditions of use that must be fulfilled by both the Subject Vehicle and the Remote Vehicle Operation, as well as the interface between them. Only in case both meet all these conditions, the AVP functionality will be safe and work as specified.

This document contains requirements for several addressees: Subject Vehicle, Vehicle Backend, User HMI and Remote Vehicle Operation. To improve readability the IDs of all requirements concerning a single addressee start with the addressee as detailed in the following table (Table 1):

Table 1 - Addressee of Requirement

ID	Addressee of Requirement
[App] App…	OEM User HMI and User Guide
[Rvo] Rvo…	Remote Vehicle Operation
[OperatorBackend] OperatorBackend…	Operator Backend
[Vehicle] Vehicle…	Subject Vehicle
[VehicleBackend] VehicleBackend…	Vehicle Backend
[General] …	General requirements

⚠ All requirements which are necessary for a functional communication between the Subject Vehicle and the Remote Vehicle Operation can be found in the associated requirement boxes, except the general requirements which characterize the whole AVPS.

2.1 Glossary

Table 2 - Glossary

Symbol	Definition	Source
AVP	Automated Valet Parking	Bosch
AVPS	Automated Valet Parking System	Bosch

D_des_fu	Designed distance to other facility users (e.g. vulnerable road users) and dynamic non-AVPS vehicles (a variable)	ISO 23374 [1]
D_des_gap	Designed longitudinal distance to the preceding vehicle during automated vehicle operation (a variable)	ISO 23374 [1]
D_des_ob	Designed distance to fixed structures or Objects other than facility users (e.g. parked vehicles) (a variable)	ISO 23374 [1]
DDT	Dynamic Driving Task	ISO 23374 [1]
DSRC	Dedicated Short Range Communication	ISO 23374 [1]
DTC	Diagnostic Trouble Code	Bosch
ECU	Electronic Control Unit	Bosch
FTTI	Fault Tolerant Time Interval	Bosch
HARA	Hazard and Risk Analysis	Bosch
HMI	Human Machine Interface	Bosch
ODD	Operational Design Domain	ISO 23374 [1]
OEDR	Object and Event Detection and Response	ISO 23374 [1]
OEM	Original Equipment Manufacturer	Bosch
P	Automated valet Parking facility management (sub-system)	ISO 23374 [1]
PFE	Automated valet parking facility equipment	ISO 23374 [1]
PFH	Probability for dangerous failures per hour	Bosch
RVO	Remote Vehicle Operation	Bosch
TARA	Threat assessment and risk analysis	ISO 21434 [2]
V	On-board Vehicle operation (sub-system)	ISO 23374 [1]
V_des_sv	Designed speed of a Subject Vehicle (a variable)	ISO 23374 [1]
V_max_sv	Maximum Designed speed of a Subject Vehicle (a fixed value)	ISO 23374 [1]
VID	Vehicle identification	Bosch
VRU	Vulnerable Road User	ISO 23374 [1]

2.2 Definitions

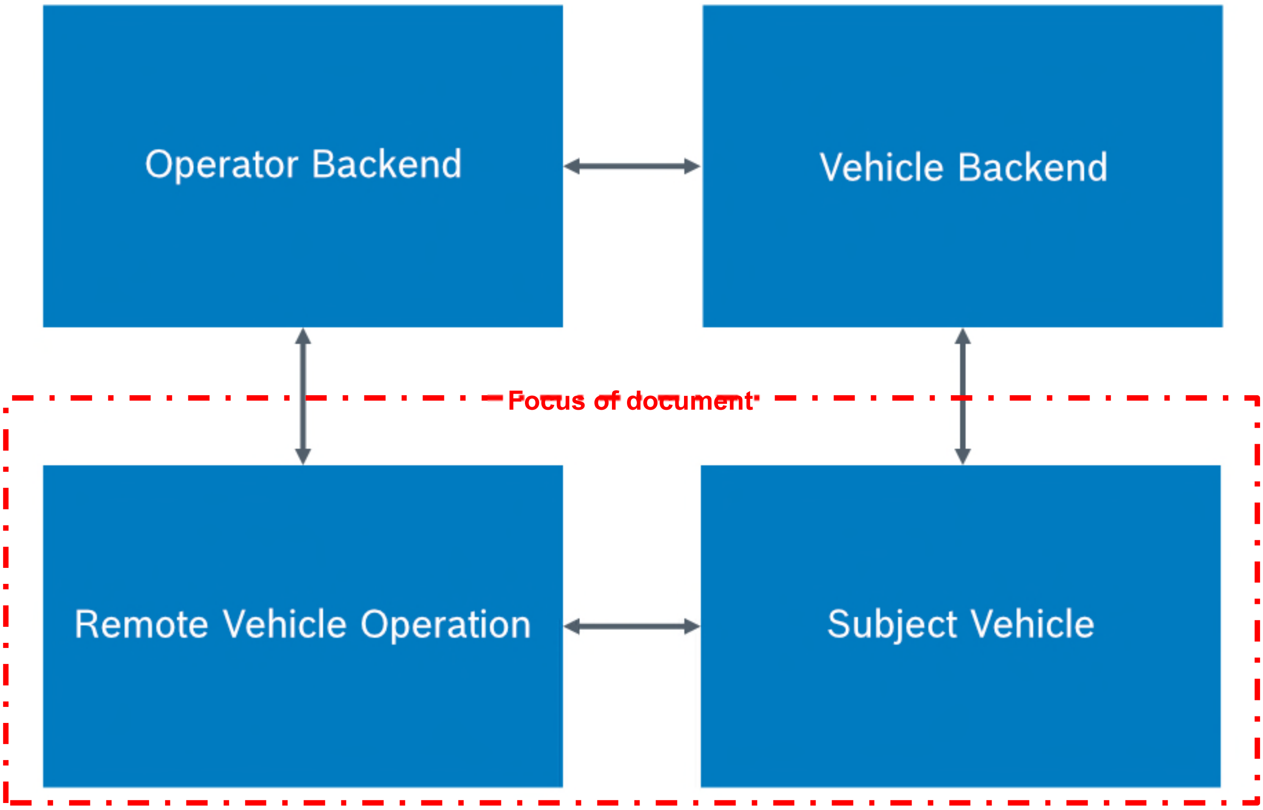


Figure 1 - AVPS Participants

Table 3 - Definitions [1]

Term	Definition
Service provider	Role of an organization that receives/ hands over Authority with Users through AVPS.
User (of AVPS)	Individual service recipient that hands over/ retrieves Authority to/from Service provider through AVPS. ⓘ The owner of a personal vehicle as well as a user of a car share service can both be an User of AVPS. ⓘ Within ISO/SAE PAS 22736, the term "user" is defined as the human role specifically in relation to driving automation systems. AVPS is a system that includes system participant management functions in

	addition to level 4 automated driving functions. Within this document, the term "system operator" authority is used as a role which performs dispatching and remote assistance in relation to the level 4 automated driving functions of AVPS. The term "user" is assigned to the individual service recipient, and not to the dispatcher or remote assistant.
Authority	Rights and ability to perform certain tasks on the Subject Vehicle.
Object	All physical entities (e.g. VRU, humans, obstacles) in the parking facility except the subject Vehicle.
Subject Vehicle	The Subject Vehicle is a light vehicle which is equipped with the on-board vehicle operation subsystem of AVPS.
Parking facility	Public or private car parking facility that is a AVPS. ⓘ The entire facility does not necessarily have to be capable of AVPS in being compliant to this document. For example, only a certain floor within a multi-story parking facility may be dedicated to AVPS. ⓘ Within ISO/TS 5206-1, the term is defined as a "Place" with related attributes.
Operation zone	Geographical area within a parking facility where automated driving can be performed. ⓘ Operation zone may contain information other than the two-dimensional geographical area, such as ceiling height or floor level information. ⓘ Within ISO/TS 5206-1, the term is defined as a "Place" using hierarchy elements.
Drop-off area	Location within the operation zone where the User leaves the Subject Vehicle and hands over Authority to the Service provider ⓘ The drop-off area may be for a single vehicle or a larger area (e.g. the entire operation zone). ⓘ Within ISO/TS 5206-1, the term is defined as a "Specific area" which is a sub-class of "Identified area".

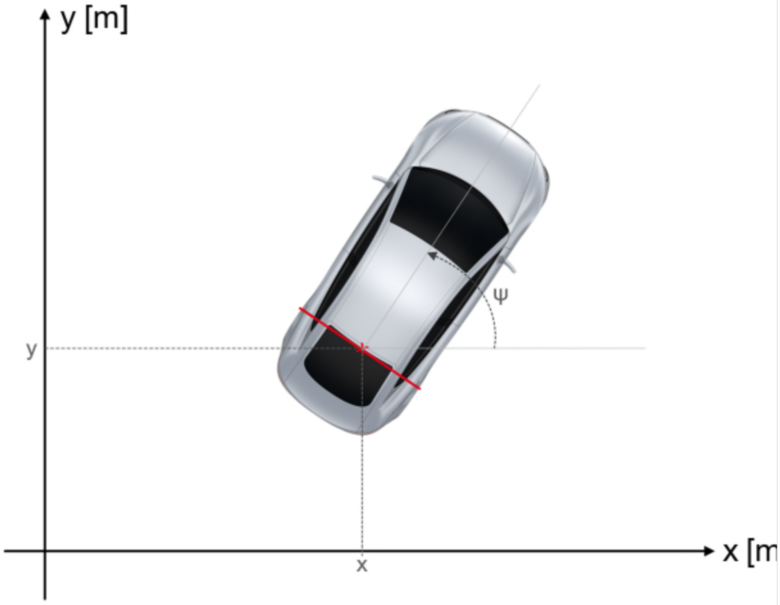
Pick-up area	Location within the operation zone where the Service provider sends the Subject Vehicle to the User for boarding, and hands over the Authority ℹ The pick-up area may be for a single vehicle or a larger area (e.g. the entire operation zone). ℹ Within ISO/TS 5206-1, the term is defined as a “Specific area” which is a sub-class of “Identified area”.
Parking spot	Area within the parking facility where a single vehicle can be parked ℹ Parking spots are typically delineated by line markers, curbstones, or other identification markings on the floor. ℹ Within ISO/TS 5206-1, the term is defined as a “Space”.
Parking area	Area within the operation zone consisting of multiple parking spots ℹ Within ISO/TS 5206-1, the term is defined as an “Identified area”.
Destination	Location within the operation zone to which the Subject Vehicle is transferred to ℹ The destination is determined by AVPS. Parking spots, service bays (e.g. location beside an electric vehicle charging station), and pick-up area are examples of a destination.
Route	Planned traversal of a Subject Vehicle from the point of origin to a destination ℹ When way point(s) are given, a route will be created to pass these way point(s).
Path	Planned sequence of way points for the Subject Vehicle to follow ℹ A path is determined based on the physical size and moving capabilities (e.g. turning radius) of the Subject Vehicle.

Trajectory	Planned Path with time information.
Automated valet parking facility equipment	Physical equipment installed in the parking facility for supporting AVPS. ⓘ For example, communication devices and detection sensors are PFEs.
Designed speed	Situation specific speed by design of a Subject Vehicle to travel under the given circumstances (e.g. traffic conditions, environmental conditions) determined by AVPS. ⓘ Designed speed is a variable and not a fixed value. AVPS will operate the Subject Vehicle based on the designed speed, resulting in dynamic changes of the actual speed of the Subject Vehicle during its operation. ⓘ Different manufacturers may provide different designed speeds under the same circumstances. ⓘ For example, AVPS will adjust the Subject Vehicle's operating speed when the Subject Vehicle is traveling towards a corner with limited visibility due to occlusion by a wall. The exact operating speed depends on the system design under this circumstance. Therefore, most of the test procedures in this document do not specify a specific value for the speed but only refer to the designed speed.
Designed distance	Situation specific physical distance by design from the Subject Vehicle to other facility users, Objects, or structures, which AVPS intends to maintain under the given circumstances while performing automated driving. ⓘ Different manufacturers may provide different designed distances towards the same Object.
Sub-system	Component of AVPS at a logical level which includes one or more functions.
Function	Ability of AVPS to process inputs to the system and contribute to conversion of the inputs into appropriate outputs.

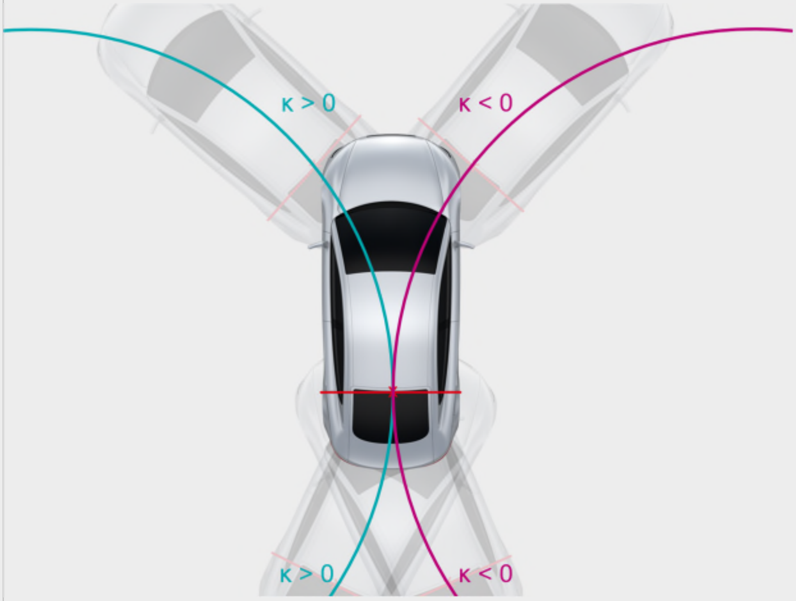
(Automated valet parking service) Reservation	Basic agreement between the User and the Service provider regarding the operation and management of the Subject Vehicle within a specific parking facility. ℹ Within this document, the term is shortened to “reservation”. ℹ A single reservation may be valid for a certain period of time or dedicated to a single Session.
Session	There may be more than one session during one valid reservation period, but multiple sessions are not carried out simultaneously for one Subject Vehicle. Typically, wireless connection is established on the operation interface during a session, leaving the possibility of suspending the connection during the sleep sub-state. [1] A session starts with the check-in and ends with the check-out of the Subject Vehicle to the AVPS. A typical session consists of multiple Missions, e.g. a Mission to drive the Subject Vehicle from the drop-off area to a parking spot and another Mission to drive the Subject Vehicle from the parking spot to the pick-up area. A session is started when the AVPS is requested to take over responsibility of the Subject Vehicle and it ends when the responsibility is given back to the User.
Mission	There may be more than one mission during one valid Session period, but multiple missions are not carried out simultaneously for one Subject Vehicle. [1] A mission is the task to move a Subject Vehicle from one position to another. A mission is started when the Remote Vehicle Operation starts preparations to move the Subject Vehicle e.g. from a dop-off area to a parking spot or from the parking spot to the pick-up area. A mission ends when the Remote Vehicle Operation detects the arrival of the Subject Vehicle at the destination or when the mission was aborted due to a failure (see chapter 7.3.1, “Abort Mission”)
System operator	The role of an organization which manages vehicle operation in the parking facility. This involves tasks which are either monitored while being performed automatically or performed manually by individuals from a remote location. (Also see chapter 3.4.2, “System operator”)
Facility manager	The role of an organization which includes tasks to be performed by individuals requiring physical access to Objects and events within the facility. (Also see chapter 3.4.3, “Facility manager”)

Remote Vehicle Operation	The Remote Vehicle Operation is the entity that guides the Subject Vehicle within the parking facility.
Operator Backend	The Operator Backend is a service that is connected to the Remote Vehicle Operation and provides an interface to the Vehicle Backend to manage reservations, Sessions and Missions.
Vehicle Backend	The Vehicle Backend is a service provided by the vehicle manufacturer, that is running outside the Subject Vehicle. The Vehicle Backend brokers between User, Subject Vehicle and Operator Backend, in order to manage reservations, initiate Sessions and Missions and provide feedback to the User. ⚠ In this document, the Vehicle Backend also includes tl tionality of the User Backend, described in ISO 23374
Standstill	Standstill means: • the Subject Vehicle's chassis is not moving physically in longitudinal direction (v_x= 0) ℹ The Subject Vehicle is kept in standstill actively. ℹ The body slip angle is negligible. Measurement accuracy is negligible. Pitching is allowed ⚠ Within the ISO, standstill is defined with a velocity below 0.01 m/s. [1]
Secure Standstill	Secure standstill means: • the Subject Vehicle's chassis is not moving physically in longitudinal direction (v_x= 0), and • Subject Vehicle is secured against rolling, even when the Subject Vehicle is powered down and even on ramps (up to 17 % inclination) ℹ The Subject Vehicle is kept in secure standstill passively ℹ Typical actuators for holding the Subject Vehicle when powered down are gear "P" or the automatic parking brake. Secure Standstill is the safe state.

Emergency Stop	Performing an emergency stop means braking with maximum available deceleration (full braking force) until any of the following conditions are true: • the reason / input for the emergency stop is withdrawn → the Subject Vehicle shall abort the emergency stop and resume driving • Subject Vehicle reaches Standstill → after the emergency stop, the Subject Vehicle shall stay in Standstill until the reason for the emergency stop is withdrawn The emergency stop is prioritized against other longitudinal control input (e.g. braking due to remaining distance). The ISO 23374 defines, that up to velocities of 10 km/h the AVPS shall bring the Subject Vehicle to a complete stop within 3s upon initiation of an operation stop command. [1] Note that a maximum of 1 s is estimated (including sufficient buffer) from the time that the operation stop device was activated until the V sub-system (see Table 5) receives the signal, giving the remaining time to actuate and bring the Subject Vehicle to a controlled stop. [1]
Valid Driving Permission	A DrivingPermission message is considered valid, if all the following conditions are true: • DrivingPermission.expirationTime is far enough in the future to allow driving (See Req120: [Vehicle] VehicleStopsOnExpirationTimeViolation) • DrivingPermission.expirationTime is not too far in the future (See Req121: [Vehicle] VehicleStopsOnExpirationTimeTooHigh) • DrivingPermission.checksum is correct (See Req125: [Vehicle] VehicleVerifiesSafetyChecksums) This means that the given boundaries (DrivingDirection, Curvature, Velocity) don’t need to allow movement for DrivingPermission message itself to be considered valid.
Safe Driving State	The safe driving state is the period within a Mission, in which the Remote Vehicle Operation safely operates the Subject Vehicle. While in safe driving state, the Remote Vehicle Operation is responsible that no damage is caused due to the Subject Vehicles movement, under the premise that the Subject Vehicle adheres to the requirements in this document.

	- When all initialization steps are completed, the Remote Vehicle Operation accepts the Mission and starts to operate the Subject Vehicle by sending the first valid DrivingPermission to the Subject Vehicle and a confirmation to the Operator Backend. (See RvoConfirmsSafeDrivingState and Req71: [Vehicle] VehicleEntersSafeDrivingState). - At latest 10 seconds after the Remote Vehicle Operation sent the last valid DrivingPermission to the Subject Vehicle, the Mission ends and the Remote Vehicle Operation resigns from its Authority and stops guiding the Subject Vehicle. (See Req72: [Vehicle] VehicleLeavesSafeDrivingState) After leaving the safe driving state, responsibility of the Subject Vehicle either remains within the AVPS or is handed back to the User.
Coordinate System	All coordinates are given in a fixed parking facility coordinate system. The parking facility coordinate system is a cartesian coordinate system with fixed origin. Coordinates can have both positive and negative values.  *Figure 2 - parking facility coordinate system* Poses always point to the center of the rear axle of the Subject Vehicle. The orientation $\psi \in [0, 2\pi)$ defines the orientation of the Subject Vehicle relative to the x-axis. Positive values define a rotation counterclockwise.
Velocity	The sign of a velocity is defined as follows: - positive sign in case of driving forwards

	• negative sign in case a friving backwards ⚠ This does not apply to the given velocity in the message DrivingPermission, where the velocity is unsigned. The direction of travel is clearly specified in this message via DrivingDirection.
Acceleration	The acceleration is defined as the time derivative of the magnitude longitudinal Velocity at the rear axle centre (also see Req108: [Vehicle] VehicleTransposesRequiredAcceleration). The sign of an acceleration specified by the Remote Vehicle Operation is defined as follows: • positive sign in case of a required driving torque of the powertrain • negative sign in case of a required braking torque In terms of the direction of travel, this means for forward and backward driving (DrivingDirection = FORWARDS or DrivingDirection = BACKWARDS): • positive sign: acceleration • negative sign: braking ℹ This means that if an increasing Velocity is required during a backward movement in the plane, the Velocity has a negative sign, but a positive sign on the acceleration given.
Curvature	The curvature κ $[\frac{1}{m}]$ describes driving on a circle with radius $r = \frac{1}{\kappa}$. • κ = 0 corresponds to driving straight • κ > 0 corresponds to driving left • κ < 0 corresponds to driving right For slow speeds as used by AVP, the curvature κ is related to the steering angle δ using wheelbase l as defined by the Ackermann steering formula: $\delta = \kappa \cdot l$

	The steering angle δ equals the front axis steering angle δ_F. The rear axis steering angle is fixed: $\delta_R = 0$  $\kappa > 0$ $\kappa < 0$ $\kappa > 0$ $\kappa < 0$ *Figure 3 - Curvatures in the AVP system*

Clocks

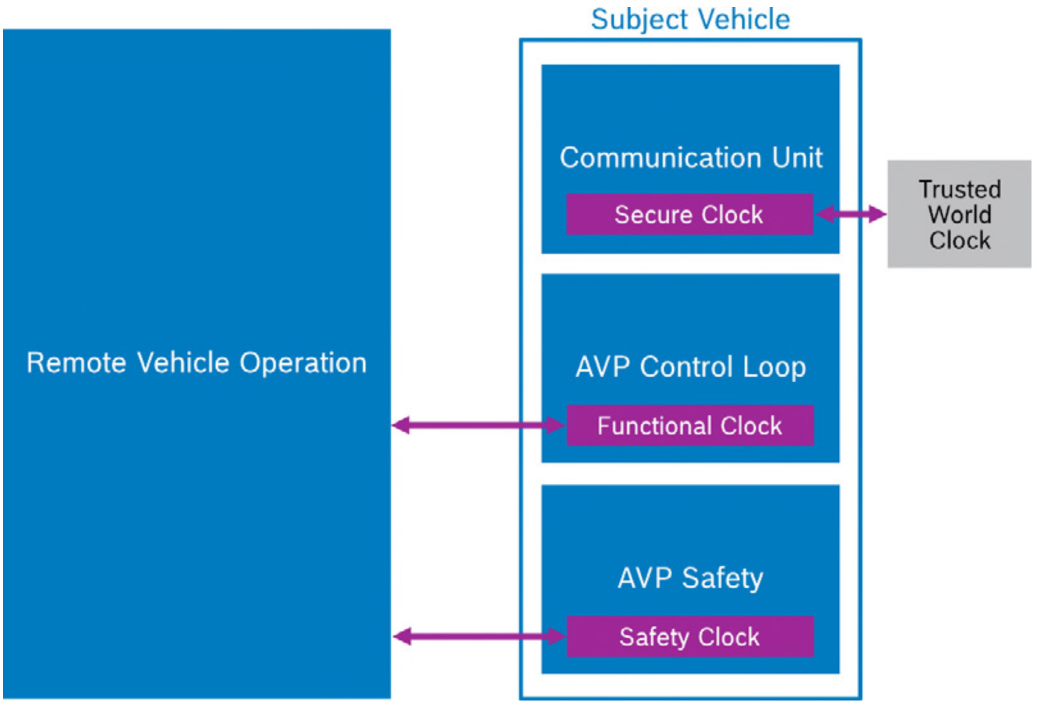


Figure 4 - Vehicle Clocks

The Subject Vehicle typically maintains the Vehicle Secure Clock, Vehicle Safety Clock and Vehicle Functional Clock. The actual number depends on the architecture of the Subject Vehicle. In the simplest case, all functions have access to the same hardware clock, in which case the three clocks are equal. For the communication between Subject Vehicle and Remote Vehicle Operation, timestamps are based on either Vehicle Safety Clock or Vehicle Functional Clock. In general, Vehicle Safety Clock is used for safety-related messages, exchanged with an ASIL capable ECU and Vehicle Functional Clock is used for all other communication. In chapter C, “AVP Messages”, the applicable reference clock is specified for each message.

World Clock	The Coordinated Universal Time (UTC) is the reference for all time operations.
Remote Vehicle Operation Clock	The Remote Vehicle Operation clock is synchronized with the World Clock (See Req31: [Rvo] RvoSynchronizeToWorldClock).
Vehicle Secure Clock	This clock is used to verify the validity of the certificates when connecting to the Remote Vehicle Operation. The Subject Vehicle synchronizes this clock with the World Clock (See Req25: [Vehicle] VehicleSynchronizeSecureClock).
Vehicle Functional Clock	This clock is mainly used by the control loop to estimate the vehicle movement since the Remote Vehicle Operation captured the last known vehicle Pose.

	Additionally, it is used for timestamps of non-safety related messages exchanged between Remote Vehicle Operation and Subject Vehicle. It does not need to be synchronized to a World Clock, but it must be steady and must not be adjusted during a Mission. The Remote Vehicle Operation starts synchronizing with this clock after the TLS and DTLS connections are established, using the messages FunctionalTimeSyncRequest and FunctionalTimeSyncResponse. See Req50: [Rvo] RvoDetermineVehicleFunctionalClock and Req51: [Vehicle] VehicleRespondToFunctionalTimeSyncRequest for details. This clock focuses on precision. Chapter 7.1.5, “TLS/DTLS Connection” shows how TLS and DTLS works.
Vehicle Safety Clock	This clock has an ASIL-B rating and is used to evaluate DrivingPermission messages. Additionally, it is used for timestamps of safety related messages exchanged between Remote Vehicle Operation and Subject Vehicle. It does not need to be synchronized to a World Clock, but it must be steady and must not be adjusted during a Mission. In the simplest case, it can be a counter that starts at 0 when the ECU is booted and is reliably increased e.g. every 10 ms. It can equal the Vehicle Functional Clock, but doesn’t need to. The Remote Vehicle Operation starts synchronizing with this clock after the vehicle identification process was finished successfully, using the messages SafetyTimeSyncRequest and SafetyTimeSyncResponse. See Req63: [Rvo] RvoDetermineOffsetToVehicleSafetyClock and Req67: [Vehicle] VehicleRespondToSafetyTimeSyncRequest for details. This clock focuses on safety, which means that uncertainties are considered in particular, e.g. the expected clock drift needs to be determined and configured in the Vehicle Type Identifier.

2.3 Interface-Specification Version

Subject Vehicle and Remote Vehicle Operation need to use the same version of the Interface Specification.

Req1: [General] NegotiateInterfaceSpecificationVersion

Operator Backend and Vehicle Backend shall negotiate the version of the Interface Specification for each Mission. They shall select the most recent version, supported by both the Remote Vehicle Operation and the Subject Vehicle.

Req2: [General] ConfirmInterfaceSpecificationVersion

Remote Vehicle Operation and Subject Vehicle shall confirm the agreed version of the Interface Specification to each other directly after the TLS connection is established, using InterfaceSpecificationVersion.version = VersionNumber (See Table 4)

If the confirmation is not received within 10 s after the TLS connection was established or if the received version does not equal the expected version, which was negotiated between the backends, both Subject Vehicle and Remote Vehicle Operation shall abort the Mission. (See chapter 7.3.1, “Abort Mission”)

The following table gives an overview of all relevant variables and its values defining the interface specifications in this document: 33

Table 4 - Table of variables

Variable	Value	Remark
VersionNumber	2.0	
TransformationConstant	0xAB54958A14FAFAD5	This constant is generated randomly.
IndicatorConstant	0xAFAD5	The constant consists of the 20 least significant bits (LSB) of the constant TransformationConstant.
AdditionalSafetyTransformationConstant	0x61767073	An additional constant to calculate the checksum DrivingPermission

⚠ Notes on Safety

The selected version of the Interface Specification is important for safety because it not only specifies the protocol between Subject Vehicle and Remote Vehicle Operation, but also the safety-related behavior of both Subject Vehicle and Remote Vehicle Operation.

Whenever any (A)SIL rated requirement or message changes between two released versions of the Interface Specification, the transformation constant (TransformationConstant for VersionNumber) which is applied to the vehicle identification seed for calculating safety checksums, will be changed.

By changing the aforementioned transformation, it is ensured that a wrongfully submitted or processed version information has no negative consequences on safety. See Req127: [General] CalculateSafetyChecksum for details.

ⓘ ISO 23374 provides a communication interface compliance check sequence (see [1], chapter A.4.1).

3 System framework regarding ISO 23374

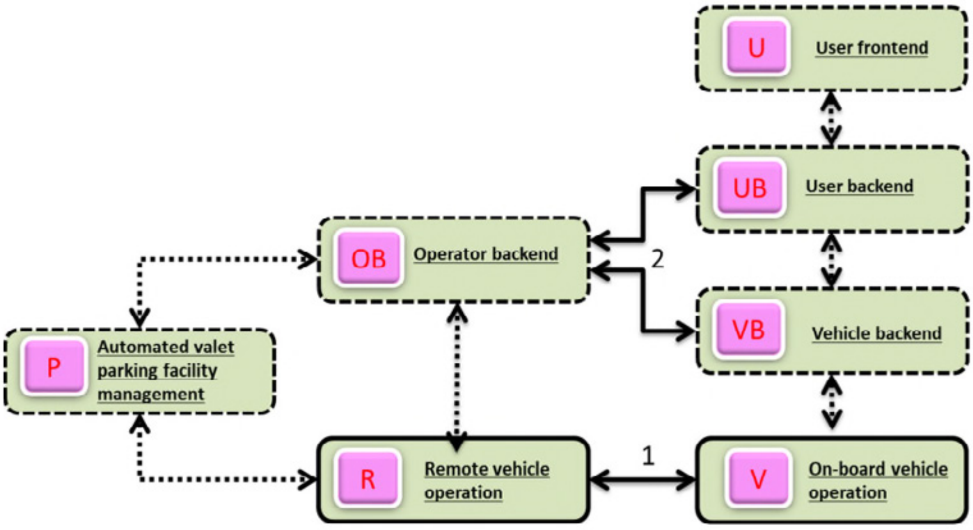
Besides explaining the system architecture this chapter classifies the AVP Type 2 system regarding ISO 23374 [1], the difference between mixed and exclusive traffic and occurring human interaction with the system.

3.1 System configuration

Figure 5 shows the logical architecture of AVPS. Implementation of the logical sub-systems to physical components is up to system design. In addition, sub-systems may be comprised of multiple physical components.

ISO 23374 specifies performance requirements for the functions allocated to the sub-systems represented by the boxes with solid lines. These two sub-systems mainly perform the operation functions.

ISO 23374 also describes the required functions of sub-systems represented by the boxes with dotted lines. These sub-systems mainly perform the management functions (4.1).



Key

1 Operation interface

2 Management interfaces

Figure 5 - System architecture [1]

Table 5 – Functional allocation [1]

ID	Sub-system	Role	Main functions
R	Remote Vehicle Operation	Performs automated vehicle operation	• Subject Vehicle identification • Destination assignment • Route planning • OEDR • Localization of Subject Vehicle • Path determination • Trajectory calculation • Emergency stopping
V	On-board vehicle operation	Performs automated vehicle operation	• Vehicle motion control • Trajectory calculation
U	User frontend	Interface to the user	• Sends user requests • Receives and informs vehicle status to user
UB	User backend	Manages the system participants	• User request processing
VB	Vehicle Backend	Manages the system participants	• Remote engagement/ disengagement
OB	Operator Backend	Manages the system participants	• Manages parking facility availability • Checks compatibility between Subject Vehicle and parking facility • Dispatches Subject Vehicles into driverless operation • Performs remote assistance
P	Automated valet parking facility management	Manages the system participants	• Manages environmental conditions • Responds to incapacitation of the operation functions

3.2 Classification AVP Type 2

The Remote Vehicle Operation sub-system carries out most of the operation functions. Means to perform the OEDR and localization functions are established by installing PFE (e.g. detection sensors, control units) in the parking facility. OEDR by Subject Vehicle on-board sensors (as part of the V sub-system) is not required.

3.3 Traffic environment categories

Parking facilities capable of handling AVPS are categorized into mixed traffic and exclusive traffic based on the traffic environment. They have different requirements for the DDT and environment.

3.3.1 Mixed traffic

Mixed traffic is a condition where the vehicles operated by AVPS share the same operation zone with other facility users, such as manually driven vehicles and VRUs. Refer to chapter 4.2.8.1, “Mixed traffic” for detailed information.

3.3.2 Exclusive traffic

Exclusive traffic is a condition where only vehicles managed by AVPS exist in the operation zone, and other facility users (e.g. vehicles not managed by AVPS, vulnerable road users VRU) are prohibited and cannot access to the operation zone. In such a traffic environment, level 4 automated driving is allowed only when there are no Objects which could negatively affect the automated vehicle operation. Refer to chapter 4.2.8.2, “Exclusive traffic” for detailed information.

3.4 Human interaction

The roles defined below can be assigned to one organization, or multiple organizations. For example, the System operator role and Facility manager role can be performed by the same organization. The tasks can also be assigned separately or to the same individual.

In general, human interaction is considered to be external to AVPS. However, this document assumes that some of the requirements allocated to the P sub-system may be designed to be performed by a human as stated.

3.4.1 Service provider

Service provider is the main organization which manages all the necessary information in order to provide services to the User through AVPS. Examples of the Service provider’s tasks are as follows:

- Coordinate the entities involved in AVPS and provide services to the User.
- Ensure that each sub-system fulfils the requirements stated in [1] and ensure that the cooperation of these sub-systems will comply with the system requirements defined in [1].

3.4.2 System operator

System operator interacts with AVPS through the Operator Backend sub-system. At a minimum, the following tasks shall be assigned:

- Dispatch the Subject Vehicle into driverless operation either manually or automatically.
- Perform remote assistance when requested by AVPS.
- Having the capability to terminate system operation when deemed necessary.

3.4.3 Facility manager

Facility manager assists the P sub-system. The requirements allocated to the P sub-system may be performed automatically or manually by the Facility manager. Following are examples of the tasks which may need to be performed by a human:

- Maintain the environment in the parking facility (e.g. illuminance, floor conditions, PFE operating conditions)
- React upon incapacitation of the automated vehicle operation (e.g. reboot the system, manually transfer incapacitated vehicles)
- Assist handover of the authority and safe start of the automated vehicle operation (e.g. ensure that all passengers have left the Subject Vehicle, and that other facility users are in an appropriate location when automated vehicles start).

4 Requirements for the overall AVP system

The following chapter focuses on defining requirements to the overall AVP system. It starts by showing the relationship of the operation functions and defining requirements to them. Following, requirements to the environment within parking facilities are described addressing e.g. Wireless communication, an operation stop device or the lighting in the facility. It ends by defining requirements to the overall system operation and the development process of AVPS.

4.1 Requirements for management functions

4.1.1 General

Figure 6 shows the relationship of the operation functions, and how some of the management functions influence the operation. In addition, an optional function for the situation where automated vehicle operation is simultaneously performed to multiple vehicles is defined in 4.1.6. Requirements for Path determination, trajectory calculation, Vehicle Motion Control, and OEDR are combined together as requirements for the DDT in chapter 7.2.1.3, “Requirements for Dynamic Driving Task (DDT)”. [1]

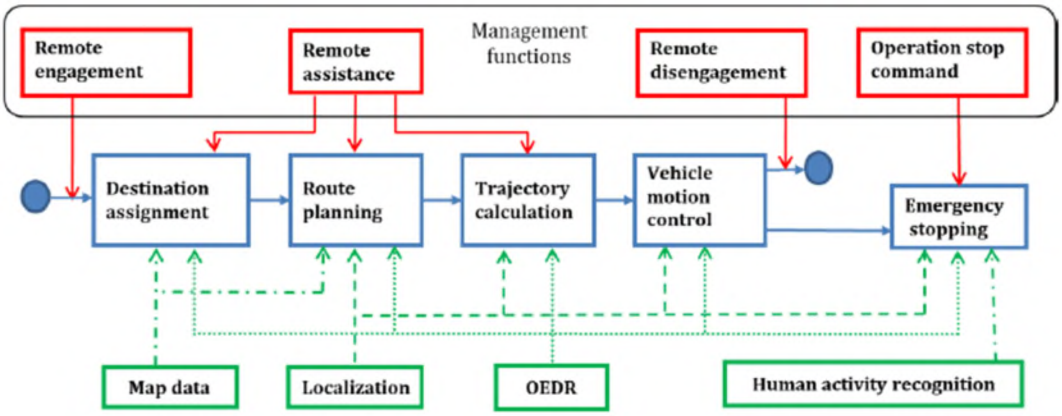


Figure 6 - Relationship of the operation functions [1]

Relationship between functions expressed in arrows with broken lines are informative and the specification is up to the system design. [1]

4.1.2 Remote engagement

The Vehicle Backend sub-system shall be capable of remotely engaging the V sub-system upon request from the Operator Backend sub-system to dispatch the Subject Vehicle into driverless operation. [1]

4.1.3 Operation stop

The Operator Backend sub-system shall be capable of commanding the automated vehicle operation to stop by issuing an operation stop command (see Emergency Stop for reaction by the operation functions).

This function also provides other facility users and Facility managers with the opportunity to immediately stop the automated vehicle operation (See chapter 4.2.4, “Operation stop device” and chapter G, “Subject Vehicle system states and transition diagram”). [1]

Further information to the messages between Operator Backend, Remote Vehicle Operation, V sub-system, Vehicle Backend can be found in [1], chapter A.4.2.

4.1.4 Remote assistance

The Operator Backend sub-system shall be capable of providing remote assistance when the automated vehicle operation functions cannot resolve certain situations on their own and process necessary commands to Remote Vehicle Operation and/or V sub-systems.

For example, two AVPS vehicles are paused facing each other at a bottleneck created by an unexpected Object on the corridor. In this case, the human operator may assign a new Route to one of the AVPS vehicles to resolve the situation.

This function, at a minimum, shall provide the System operator with an interface with the opportunity to submit the following commands:

- Remote assistance shall provide the System operator with the capability to change the assigned destination and planned Route.
- Remote assistance shall provide the System operator with the capability, at a minimum, to command the Subject Vehicle to “pause”, or “permission to proceed”.
  – In addition, commands such as “slow down” or “reverse” may be added.

In comparison to the operation stop command where the Subject Vehicle(s) performs an immediate stop, the System operator may prefer to stop the operation without blocking a certain location (e.g. emergency pathways, in front of fire extinguishers) under certain situations (e.g. earthquakes, fire). In such cases, the remote assistance function may be utilized to evacuate the Subject Vehicles to a nearby preferable location. [1]

4.1.5 Remote disengagement

The Vehicle Backend sub-system shall be capable of disengaging the V sub-system upon receiving report of reaching the destination (see Transition 4R in [1], chapter 9.3.3.8).

The Operator Backend sub-system shall be capable of disengaging the V sub-system by triggering a transition to suspend state (See Transition 8b in [1], chapter 9.3.3.18 and Req125: [Vehicle] VehicleVerifiesSafetyChecksums).

4.1.6 Central control

The Operator Backend sub-system may be equipped with a central control function. Central control is an optional function, which coordinates the operations of multiple vehicles.

By utilizing such a function, interference between AVPS vehicles can be avoided. Under an exclusive traffic environment, collisions between vehicles can be avoided, and traffic efficiency can be optimized. Under mixed traffic environment, potential conflict between two AVPS vehicles at certain location can be prevented, and right of way can be given to one of the two vehicles. [1]

4.1.7 Response to incapacitation of the operation functions

The P sub-system shall be capable of performing the following activities when automated vehicle operation becomes incapacitated.

- Determine if the system capabilities have been recovered and automated vehicle operation may be resumed.
  - Resume automated vehicle operation if the conditions are satisfied.
  - If the conditions are not satisfied, physically access to the Subject Vehicle and transfer the Subject Vehicle by means other than the automated vehicle operation. For example, by manually driving the Subject Vehicle or utilizing a tow truck. [1]

4.1.8 Maintaining environmental conditions

The P sub-system shall be capable of maintaining the environmental conditions within the parking facility as defined in chapter 4.2, “Requirements for the environment within parking facilities”. Therefore, the P sub-system should frequently check the environment within parking facilities, especially for the conditions which cannot be detected or recovered by other sub-systems. The frequency of the check depends on the characteristics of the parking facility.

For example, a parking facility located in the basement of a building is not likely to be affected by snow fall, thus compared to an outdoor facility, the frequency to check the road surface conditions may be relatively low.

When environmental conditions are not satisfied, the P sub-system shall limit the operation of AVPS as needed.

The operation zone may be limited, for example, excluding areas covered with snow, which are not suitable for automated vehicle operation. The P sub-system is expected to communicate such a situation and stop accepting new reservations, retrieval request, or check-ins via AVPS.

For another example, if an Object is blocking the driveway, the P sub-system is expected to remove this Object. [1]

4.2 Requirements for the environment within parking facilities

This sub-chapter defines ODD elements and PFE requirements to be established within the parking facilities capable of AVPS. [1]

4.2.1 Operation zone

The operation zone shall be predetermined. For example, the operation zone may be the entire facility with a mixed traffic environment, or it may be a certain floor within a multi-story facility for an exclusive traffic environment.

The boundaries of the operation zone shall be defined with a sufficient buffer between the operation zone and the area where automated vehicle operation is not allowed. [1]

4.2.2 Drop-off and pick-up area

Both drop-off and pick-up areas shall be located within the operation zone. Both areas should be large enough to provide comfortable space for boarding and exiting of the driver and passengers as well as for loading/ and unloading of luggage. It is recommended that these areas should be located in places where they do not interfere with the traffic flow. There may be more than one drop-off and more than one pick-up area within an operation zone. Drop-off and pick-up areas may share the same space (e.g. separated by time). [1]

4.2.3 Wireless communication

There shall be no communication blind spots within the operation zone. Sufficient bandwidth should be secured for system operation. Chapter 4.3.1 “Requirements for the communication interface” specifies the general requirements for the communication interface.

The communication media is not specified in [1], and it is up to the system design. Wireless communication may be realized by cellular network (e.g. 4G, 5G), wireless local area network (e.g. Wi-Fi, Dedicated Short Range Communication (DSRC)), the combination of the two, or other means suitable for the specific implementation. [1] More specific information are described in Req40: [Vehicle] VehicleConnectToWifi and chapter 8.1.1, “Link and Network layer”.

4.2.4 Operation stop device

One or more operation stop device(s), which can immediately stop the automated vehicle operation (see [1], chapter 6.3.3 and 7.1.3), shall be installed either in the facility or in a separate location where it can be manually operated, for example by other facility users, System operators, and Facility managers. A typical implementation of this device is similar to the “emergency stop button” installed on heavy machinery or at train stations.

The number of operation stop devices and applicable area vary depending on the system design and the floor space. For example, a multi-story facility may require such a device on each floor to stop the vehicles only for the particular floor. [1]

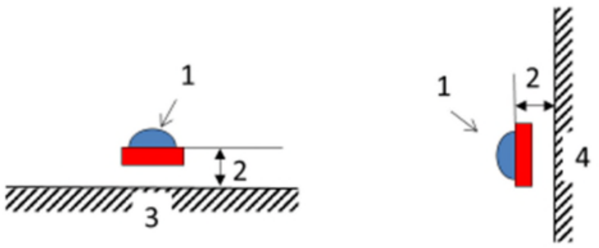
4.2.5 Lighting

The illuminance should be greater than 20 lx throughout the operation zone.
The illuminance value shall be ensured at the following segments. It may be lower at the borders of these segments.

- Along the centre line of the Subject Vehicle driveways
- Along the centre line of pedestrian walkways
- In the centre of all possible destinations
- If localization markers as described in [1] are used, at the location where a localization marker is placed.

An illuminance meter that meets ISO 19476:2014 shall be used for measurement. The illuminance meter shall be positioned less than 20 cm away from the surface of the above items to measure the illuminance in vertical direction (90° to the surface). See Figure 7 for illustration of the orientation of the illuminance meter.

⚠ Note that the illuminance requirements specified in ISO 20900 do not apply to AVPS.



a) measurement on horizontal surface

b) measurement on vertical surface

Key

1	Illuminance meter	**2**	< 20 cm
3	Horizontal surface (e.g. ground)	**4**	Vertical surface (e.g. pillar with a localization marker)

Figure 7 - Illuminance measurement on horizontal and vertical surfaces

4.2.6 Detection capabilities

Means to perform the OEDR and localization functions allocated to the Remote Vehicle Operation sub-system are typically established with detection sensors installed in the parking facility. For this reason, maintaining the necessary detection performance is classified as part of the environmental conditions.

The Remote Vehicle Operation sub-system shall be capable of performing the OEDR and localization functions necessary for automated vehicle operation. For this reason, the detection capability of the Remote Vehicle Operation sub-system shall cover the entire operation zone without blind spots. The Remote Vehicle Operation sub-system shall detect Objects, localize the Subject Vehicle, and detect occupancy of destinations.

Object detection capabilities are specified based on mixed traffic environment. Capabilities could be reduced for operation under an exclusive traffic environment. [1]

4.2.7 Digital maps

Type 2 vehicle operation type requires general information about the geometry of the parking facility and operation zone which may be provided by a digital map, a facility layout, or other suitable means. [1]

Further details to digital maps are described in [1], chapter 8.3.3.

4.2.8 Traffic environment category dependent requirements

4.2.8.1 Mixed traffic

For system installation, the parking facility structure shall comply with relevant local regulations/guidelines for pathway width, minimum Curvature, ramp angle, dimensions for parking spaces, etc. Also see information provided in [1], chapter F for an example of facility structure requirements in case of absence of such local regulations/guidelines.

- If speed bumps are placed in the operation zone, the maximum height shall be limited to 8,5 cm to avoid unintended detection results.
- If Objects are intentionally placed to prohibit entrance of vehicles to certain areas, the Object used to represent the border shall be larger than the size of the Object specified in [1], chapter B.1. For example, chains or ropes may not be classified as obstacles by AVPS and therefore should not be used.

Although [1] does not require a certain surface condition to parking facilities for mixed traffic environments, it should be noted that systems which only comply with the minimum requirements defined in [1] are based on operation on paved surfaces (see chapter 7.2.1.2, “Operational design domain”). [1]

4.2.8.2 Exclusive traffic

There shall be no Objects within the operation zone which could damage the Subject Vehicle upon physical contact (e.g. Objects larger than a golf ball, Objects with sharp edges).

The road surface within the operation zone shall be a flat pavement with no irregular bumps/holes which could negatively affect the automated vehicle operation.

Access control gates shall be installed at the border of the operation zone to allow entry of vehicles with a valid Session only and prevent other facility users from entering the operation zone. Visible signs are recommended to be installed as needed.

Dividing barriers shall be installed along the outer edge of the operation zone to restrict entrance of other facility users, especially humans. Visible signs are recommended to be installed as needed.

Means to prevent initiation of automated vehicle operation when humans (e.g. User, passenger) are present in the drop-off or pick-up areas shall be provided, for example, by surveillance or physical barriers. This is due to these areas temporarily becoming a mixed traffic environment to allow humans to board and exit an Subject Vehicle. [1]

4.3 Requirements for the overall system operation

AVPS is a cooperative system comprised of physically separated sub-systems, in many cases provided by different organizations. For this reason, the communication interface is crucial for the overall system operation, and communication among the different organizations is the key element for establishing interoperability.

General requirements for the communication interface are specified in chapter 4.3.1, “Requirements for the communication interface”.

The minimum set of data elements that need to be communicated within AVPS is described based on the logical architecture shown in Figure 5.

4.3.1 Requirements for the communication interface

4.3.1.1 General requirements

- AVPS shall be capable of processing the following requests from the User on demand. These requests are processed independently of an active Session. For example, after entering an AVPS compliant parking facility, the User may request for availability of a compatible parking facility near the User’s next destination.
  - Availability request
  - If applicable, payment requests

- AVPS shall be capable of processing on-demand Mission requests from the User when a valid reservation or Session exists. For example, the User may request an additional service while the Subject Vehicle is being parked.
    - At a minimum, AVPS shall be capable of processing a retrieval request when a valid Session exists.
    - AVPS may also process additional service-related requests when a valid reservation or Session exists.
- A reservation shall be established before a Session is created. Also, the User is expected to confirm the parking facility, Service provider, and terms and conditions of the service when a reservation is established. However, this document does not specify means to establish such reservation and User confirmation.
    - The three backend sub-systems shall be capable of performing a communication interface compliance check anytime a valid reservation exists.
- A Session shall be created upon successful handover of the Authority from the User to the Service provider at the latest. The created Session shall be maintained at least until successful handover of the Authority from the Service provider to the User, or the Service provider decides to revoke the Session.
    - AVPS shall be capable to support, at a minimum, event-based communication regarding the condition of the Subject Vehicle and status of the Session and/or Mission when state transition occurs between the system management states shown in Figure 37 (this does not include system states within the automated vehicle operation). The information shall be communicated to the Vehicle Backend sub-system via the Operator Backend sub-system while a valid Session exists.
- A Mission shall be assigned when dispatching an Subject Vehicle and maintained until disengaged.
    - The operation interface shall be capable of (re)establishing a valid time synchronization for any communication on that channel while a valid Mission exists (see chapter 7.1.9, “Vehicle Safety Clock Synchronization” and chapter 7.1.6, “Vehicle Functional Clock Synchronization”).
    - The V sub-system shall be capable of providing a periodic message regarding the condition of the Subject Vehicle while a valid Mission exists. [1]

4.3.1.2 Security goals

System designers shall perform threat assessment and risk analysis and implement mechanisms to avoid and protect AVPS from security threat. Following are examples of security threat specific to AVPS.

- Vehicle theft
    - For example, unauthorized command to unlock the Subject Vehicles doors and turning on the ignition transmitted by a malicious source may lead the Subject Vehicle to be manually transferred outside of the facility.
- Safety hazards
    - For example, unintended DDT relevant commands transmitted by a malicious source may lead the Subject Vehicle to collide with a facility user.
- Private information leakage

  - For example, malicious attacks to the backend sub-systems may lead to leakage of private information of the customer and vehicle.
- Availability degradation or loss
  - For example, denial of service attacks may lead to degradation or loss of the system performance. [1]

4.3.1.3 Security requirements

- AVPS shall be capable of (re)establishing secure channels among the sub-systems throughout a reservation in order to achieve the security goals.
- AVPS shall be capable of (re)establishing mutual authentication of the management interfaces when a valid Session exists in order to achieve the security goals.
- AVPS shall be capable of (re)establishing mutual authentication of the operation interfaces when a valid Mission exists in order to achieve the security goals. [1]

4.3.2 Information to the user

AVPS shall provide the User with the following information.

- Authority has been handed over between the User and Service provider
- Estimated time of arrival in response to a retrieval request

The system may provide the User with the following information:

- Location of the parked Subject Vehicle
- Suspend conditions
- Mission assignment status [1]

4.4 Development process and management

Service providers shall ensure that the sub-systems comply with the requirements specified in the following documents.

- Compliance of the Remote Vehicle Operation sub-system with the respective part(s) of IEC 61508
  - When communication is used to fulfil safety scenarios, countermeasures for failures of the communication process shall be considered according to ISO IEC 61508.
- Compliance of the V sub-system with the respective part(s) of ISO 26262 series (Functional safety).
- Compliance of the Remote Vehicle Operation and V sub-systems with ISO 21448 (SOTIF) and ISO/SAE 21434 (Cyber security engineering)
- Compliance of the Operator Backend/ Vehicle Backend sub-systems with respective part(s) of ISO/IEC 27000 family (Information security management system) [1]

5 Subject Vehicle and Vehicle Backend Requirements

To enable a Subject Vehicle for AVP operation and to ensure proper AVP operation, the Subject Vehicle needs to fulfill certain preconditions. These are listed in this chapter

5.1 General Requirements

Req3: *[Vehicle] VehicleRequiredCapabilities*

For AVP operation, the Subject Vehicle needs to have all the following capabilities:

- starting and switching off the engine, without a key being physically present
- longitudinal control (i.e. remaining driving distance; target driving velocity; forwards/ backwards etc.)
- steering the Subject Vehicle
- indicator actuation
  - for indicating the turning intention and
  - for Vehicle Identification (See chapter 7.1.8, “Safe Vehicle Identification”) where it is needed even before entering Safe Driving State.
  - for activating and deactivating the warning lights

Req4: *[VehicleBackend] VehicleBackendWakesUpECUs*

The Vehicle Backend shall be able to wake up the Subject Vehicle.

ⓘ A Subject Vehicle might be parked for several hours, days or even weeks. While being parked, a Subject Vehicle usually powers down most of its ECUs to reduce power consumption. In order to drive the Subject Vehicle e.g. back to the pick-up area, the Subject Vehicle needs to be woken up on request by the Vehicle Backend.

Req5: [Vehicle] VehicleChassisConfiguration

The Subject Vehicle shall have two axles and two tires per axle.

The Subject Vehicle shall not have any attachments or boarding aids which modify the vehicle shape defined in the corresponding Vehicle Type Identifier.

This means that permanent attachments shall be considered by assigning a dedicated Vehicle Type Identifier.

The Remote Vehicle Operation may decline the Mission if the actual shape and the shape defined in the corresponding Vehicle Type Identifier do not match.

The User is informed about temporary attachments in Req26: [App] AppGuideNoChangesOfVehicleOutline.

Req6: [Vehicle] VehicleSupportsFastWiFiRoaming

The Subject Vehicle shall support the Wi-Fi standards 802.11k, 802.11v, and 802.11r to allow for faster roaming between access points.

This requirement only applies to vehicles that connect to the Remote Vehicle Operation by using a Wi-Fi connection.

Req7: [Vehicle] VehicleMaintainsFunctionalClock[1]

The Subject Vehicle shall maintain a steady Vehicle Functional Clock on the ECU on which it runs the control loop.

[1] derived from Vehicle Functional Clock

Req8: [VehicleBackend] VehicleBackendChecksAVPPreconditions

The Session may only be established if all the following conditions are true:

- The User is eligible to use AVP
- Operator Backend has accepted the Subject Vehicle type for the requested parking facility

ⓘ The provider of the Remote Vehicle Operation and the OEM maintain a list of eligible vehicles, depending on the Vehicle Type Identifier (See chapter 7.1.10, “Safe Vehicle Type Confirmation”).

⚠ The Operator Backend/ Vehicle Backend sub-systems shall be capable of checking occupancy and compatibility of parking facilities with respect to the characteristics of the Subject Vehicle upon receiving User requests, and to communicate the necessary information to the User. Among other information the vehicle operation type and ODD definition between Remote Vehicle Operation and V sub-system shall be confirmed. [1]

5.2 Safety Requirements

Req9: [Vehicle] VehicleHara

The OEM shall perform an own Hazard and Risk Analysis (HARA) that covers potential vehicle failures inside and outside AVP situations.

The Automotive Safety Integrity Levels (ASIL) assigned in this document are a guideline and shall be evaluated individually by the OEM, based on its HARA.

The HARAs performed by the OEM and by the provider of the Remote Vehicle Operation shall be compared and differences shall be discussed and documented (also see Req28: [Rvo] RvoHara).

Req10: [Vehicle] VehicleFailureAnalysis

The OEM is responsible for performing a failure analysis (e.g. an FMEA) on the Subject Vehicle with the purpose of identifying safety-critical failures, e.g. sudden loss of braking capabilities.

Req87: [Vehicle] VehicleAbortsMissionOnCriticalVehicleFailure defines how to handle such an event.

Req11: [Vehicle] VehicleBrakingPerformance

The Remote Vehicle Operation shall consider the vehicles braking distance depending on the allowed maximum velocity. These values are associated with the Vehicle Type Identifier.

The braking distance $d_{braking}(v)$ shall be measured from the point in time $t_{brakingInitiated}$ when the signal for braking/deceleration is set until the vehicle reached standstill, in an environment with a friction coefficient $\mu \approx 1.0$ on a horizontal level, in dependence on the initial velocity v.

The point in time $t_{brakingInitiated}$ shall be chosen as close as possible to the point in time $t_{decelerationBegins}$, which describes when the actual deceleration begins. This period of time shall contain e.g. the time needed to establish braking force and essential signal delays within the braking system. This is illustrated in Figure 30 and Figure 31.

Table 6 gives a guideline for a reasonable braking performance, assuming maximum braking force is reached after $t_{decelerationBegins} - t_{brakingInitiated} = 250ms$:

Table 6 - Maximum braking distance in dependence on the initial velocity

Initial velocity v [km/h]	Braking distance $d_{braking}\ (v)$ [m]
10.0	0.69 m + 0.80 m = 1.49
9.0	0.63 m + 0.70 m = 1.33
8.0	0.55 m + 0.60 m = 1.15
7.0	0.49 m + 0.50 m = 0.99
6.0	0.42 m + 0.40 m = 0.82
5.0	0.35 m + 0.30 m = 0.65
4.0	0.28 m + 0.20 m = 0.48
3.0	0.21 m + 0.15 m = 0.36
2.0	0.14 m + 0.10 m = 0.24
1.0	0.07 m + 0.05 m = 0.12

Critical vehicle failures include the loss of the Subject Vehicle's capability to stop within the specified braking distance.

The time period $t_{safetyToBrakingInitiated}$ between evaluating the DrivingPermission and initiating the braking shall be known within the Subject Vehicle and is considered separately in Req119: [Vehicle] VehicleDrivingPermissionTimeBudget.

Req12: [Vehicle] VehiclePfh

The maximum acceptable probability of dangerous failure per hour (PFH) shall be derived from the hazard and risk analysis according to IEC 61508. The PFH must be split between Remote Vehicle Operation and Subject Vehicle. See also Req29: [Rvo] RvoPfh.

Req13: [Vehicle] VehicleFTTI

If a failure in AVP relevant vehicle component occurs in Safe Driving State, the Subject Vehicle shall go into Secure Standstill within 3 seconds (Fault Tolerant Time Interval, FTTI) and abort the Mission (see chapter 7.3.1, “Abort Mission”) if the Subject Vehicle cannot recover from the failure within 10 seconds.

Req14: [General] FreedomFromInterference

SIL=ASIL-B

The OEM ensures that the on-board vehicle operations (V) does not access the actuators when not in Safe Driving State (freedom from interference).

The Subject Vehicle can be in or outside of Safe Driving State. The transition between the two is illustrated in Figure 9.

Req15: [Vehicle] VehicleMaintainsSafetyClock[2]

The Subject Vehicle shall maintain a steady Vehicle Safety Clock on the ECU on which it evaluates DrivingPermission messages.

5.3 Security Requirements

Since AVPS is offering a manner to remotely guide a Subject Vehicle, it must be ensured that the Subject Vehicle only trusts authenticated and authorized entities within a defined environment and period of time for a Mission.

Req16: [Vehicle] VehicleSecConceptforAVPIntegration

The integration of the AVP functionality into an AVP-enabled Subject Vehicle requires a threat and risk analysis of the architecture for integration in the Subject Vehicle, in order to enable a derivation of adequate security measures in a dedicated security concept.

The OEM shall derive a security concept for the AVP integration in the Subject Vehicle that entails at least the following baseline points:

- protection of the AVP related part of the firmware and cryptographic material from manipulation and unauthorized access (Secure Boot and Secure Storage)
- protection of AVP related messages from manipulation on the vehicle bus (secure, authentic and fresh communication, i.e. protection from e.g. replay attacks)
- isolation of the AVP functionality to prevent misuse through other functionality
- hardening of the processing entities, in particular the ECU receiving the external communication and forwarding the messages on the vehicle bus.
- vulnerability, incident and patch management to ensure timely delivery of security updates in an authentic manner
- protection of the AVP related ECUs from unauthorized access
- Threat assessment and risk analysis (TARA).

In general, the AVP messages are conveyed in an authenticity protected manner, which entails protective measures on the vehicle bus. The goal is to provide authenticity between the Remote Vehicle Operation and the actuators executing the Mission on the Subject Vehicle. A comprehensive E/E security architecture ensures that safety related messages are protected from Remote Vehicle Operation up to the actuators.

[2] derived from Vehicle Safety Clock

Req17: [Vehicle] VehicleEnsuresSufficientSecurity

The Subject Vehicle shall be secured against malicious activity, according to UNECE R 155.

This includes protection against cyberattacks and unauthorized physical access.

Req18: [Vehicle] VehicleSecureOnboardCommunication

All AVP related messages shall be protected from manipulation whilst being conveyed within the Subject Vehicle.

The Subject Vehicle shall use individual cryptographic keys per vehicle to protect messages conveyed within the Subject Vehicle to ensure that an attack on one Subject Vehicle does not scale, i.e. does affect only one individual Subject Vehicle.

Req19: [Vehicle] VehicleInspectSecurityConcept

The parts of the vehicle security concept relevant for AVP shall be made available to Remote Vehicle Operation providers for inspection upon request.

Req20: [VehicleBackend] VehicleBackendInspectSecurityConcept

The parts of the Vehicle Backend security concept relevant for AVP shall be made available to Remote Vehicle Operation providers for inspection upon request.

Req21: [Vehicle] VehicleInspectPenTestResults

The results of AVP related vehicle penetration tests shall be made available to Remote Vehicle Operation providers for inspection upon request.

Req22: [VehicleBackend] VehicleBackendInspectPenTestResults

The results of AVP related Vehicle Backend penetration tests shall be made available to Remote Vehicle Operation providers for inspection upon request.

Req23: [Vehicle] VehicleInformSecurityBreach

Any evidence and/or knowledge of a vehicle security breach relevant for AVP shall be communicated to Remote Vehicle Operation providers without undue delay.

Req24: [VehicleBackend] VehicleBackendInformSecurityBreach

Any evidence and/or knowledge of a Vehicle Backend security breach relevant for AVP shall be communicated to Remote Vehicle Operation providers without undue delay.

Req25: [Vehicle] VehicleSynchronizeSecureClock[3]

The Subject Vehicle shall maintain a Secure Clock and synchronize it to a trusted World Clock with a typical deviation of less than 60 s.

It shall not be possible to turn the clock back in time, to e.g. prevent an attacker to use expired or revoked certificates.

Purpose of this clock is to verify the validity of certificates when connecting to the Remote Vehicle Operation.

5.4 User Communication

Req26: [App] AppGuideNoChangesOfVehicleOutline

SIL=ASIL-QM
When the user hands over the authority to the AVPS, the User shall be informed that installations that temporarily or permanently change the outline of the Subject Vehicle are not allowed when using AVP.

[3] derived from Vehicle Secure Clock

Such installations could be trailer coupling, bike carriers, etc.

The Remote Vehicle Operation may decline the Mission if the actual shape and the shape defined in the corresponding Vehicle Type Identifier do not match, see Req5: [Vehicle] VehicleChassisConfiguration.

Req27: [App] AppUserConfirmsEmptyVehicle

When the user hands over the authority to the AVPS, the user shall be asked to confirm that no people or animals are inside the Subject Vehicle.

The confirmation shall be given latest before the Subject Vehicle can enter the safe driving state, otherwise both session and Mission shall be rejected.

The Users confirmation can be verified or even replaced with sufficient sensors in the Subject Vehicle, see Req73: [Vehicle] VehicleVerifiesPreconditionsBeforeEnteringSafeDrivingState

6 Remote Vehicle Operation and Operator Backend Requirements

In the following requirements for the infrastructure are described regarding Failure Analysis or clock synchronization. The appropriate requirements to the vehicle are described in chapter 5, “Subject Vehicle and Vehicle Backend Requirements”.

Req28: *[Rvo] RvoHara*

The Remote Vehicle Operation provider has to perform an own Hazard and Risk Analysis (HARA) that covers potential infrastructure failures.

The Safety Integrity Levels (SIL) assigned in this document are a guideline and need to be evaluated individually by the Remote Vehicle Operation provider, based on its HARA.

Req29: *[Rvo] RvoPfh*

The maximum acceptable probability of dangerous failure per hour (PFH) shall be derived from the hazard and risk analysis according to IEC 61508. The PFH must be split between Remote Vehicle Operation and Subject Vehicle. Also see Req12: [Vehicle] VehiclePfh.

Req30: *[Rvo] RvoFailureAnalysis*

The Remote Vehicle Operation provider is responsible for performing a failure analysis (e.g. an FMEA) on the Remote Vehicle Operation with the purpose of identifying safety-critical failures.

The Remote Vehicle Operation shall detect a failure and bring the AVPS to the safe state within 3 s. (in accordance to [1], also see Emergency Stop).

Req31: *[Rvo] RvoSynchronizeToWorldClock*

The Remote Vehicle Operation shall securely synchronize its clock to the Coordinated Universal Time (UTC) and maintain a deviation of less than 20 ms. The time shall not jump while vehicles are connected.

Req32: *[Rvo] RvoInspectSecurityConcept*

The Remote Vehicle Operation security concept shall be made available to OEMs for inspection upon request.

Req33: *[Rvo] RvoInspectPenTestResults*

The results of Remote Vehicle Operation penetration tests shall be made available to OEMs for inspection upon request.

Req34: *[Rvo] RvoInformSecurityBreach*

Any evidence and/or knowledge of a Remote Vehicle Operation security breach shall be communicated to OEMs without undue delay.

7 The AVP Mission Process

This chapter focuses on the execution of a single Mission. This is illustrated in Figure 8. It includes the safe driving state and the transitions On Hold → Safe Driving → On Hold. Furthermore, the content and requirements are described in detail in the following chapters. A detailed sequence diagram of the process can be found in chapter F, "Complete Sequence of AVP Mission Process". Figure 9 shows a top-level overview of the Subject Vehicle and Remote Vehicle Operation related states. A more detailed description of the Subject Vehicle states can be found in chapter G, "Subject Vehicle system states and transition diagram".

[1] formulates considerations when the User initially hands over the Authority to the Service provider as illustrated in Figure 8, this document presupposes that an agreement exists between the User and Service provider to perform the following tasks regarding the Subject Vehicle through AVPS, at a minimum.

- Dispatch the Subject Vehicle into driverless operation
- Perform level 4 automated driving on such a dispatched vehicle
- Turn off the Subject Vehicle at its parked location.

Additionally, the User may agree on certain tasks such as the following:

- Allow the Facility manager to enter the vehicle and manually operate the vehicle in the case that automated vehicle operation is not possible
- Allow opening and closing of the trunk of the Subject Vehicle for package delivery service
- Allow maintenance work

It is assumed that the User and Service provider agree on the terms and conditions when using AVPS, which would include the aspects above. However, the form and the contents of the agreement are not within the scope of this document.

It should be noted that the Facility manager is required to manually relocate the Subject Vehicle in the case of incapacitation of automated vehicle operation. If entering the vehicle cannot be agreed upon, alternative means shall be prepared in order to comply with the requirements of this document.

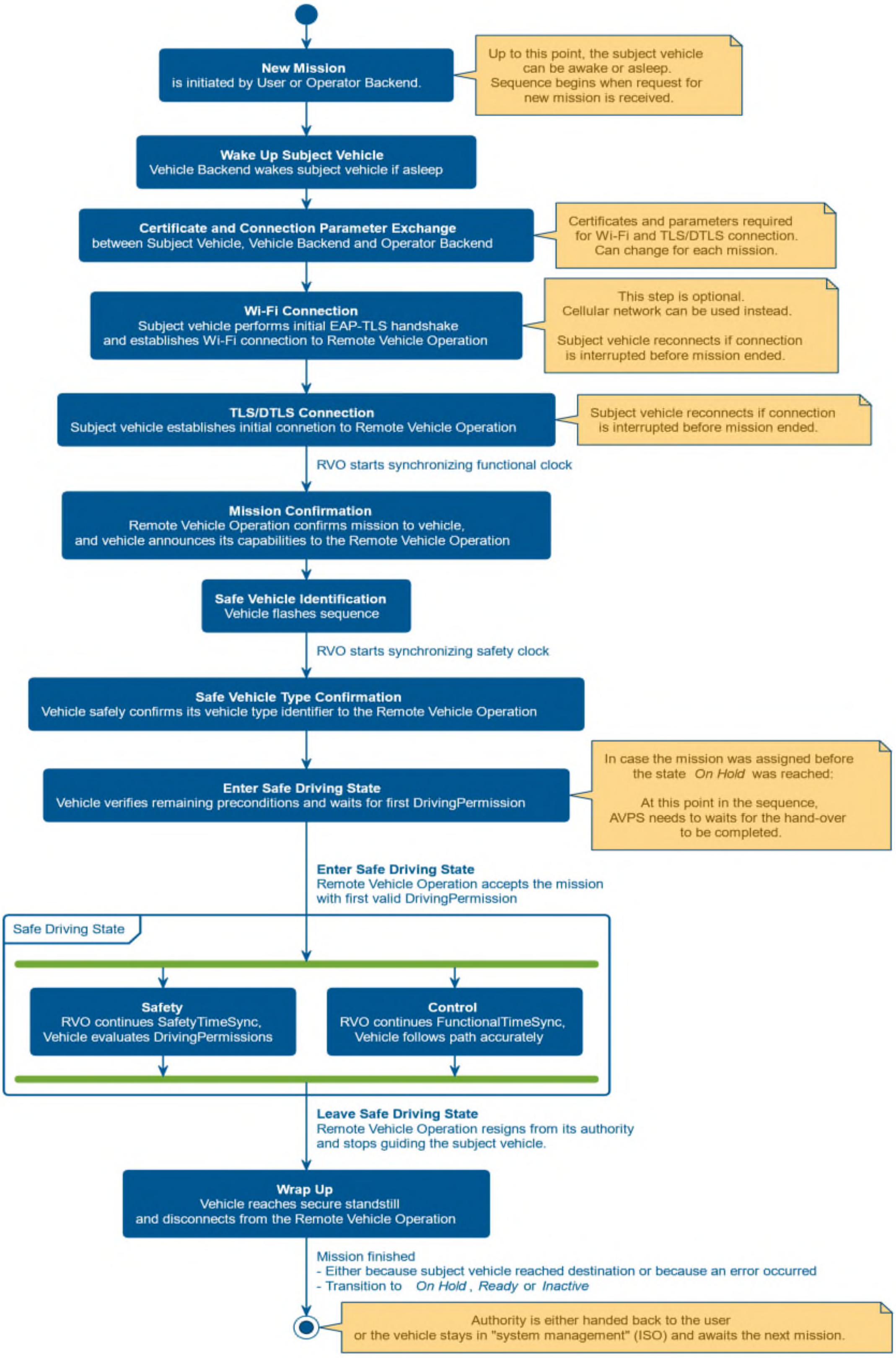

New Mission
is initiated by User or Operator Backend.
Up to this point, the subject vehicle can be awake or asleep. Sequence begins when request for new mission is received.
Wake Up Subject Vehicle
Vehicle Backend wakes subject vehicle if asleep
Certificate and Connection Parameter Exchange
between Subject Vehicle, Vehicle Backend and Operator Backend
Certificates and parameters required for Wi-Fi and TLS/DTLS connection. Can change for each mission.
Wi-Fi Connection
Subject vehicle performs initial EAP-TLS handshake and establishes Wi-Fi connection to Remote Vehicle Operation
This step is optional. Cellular network can be used instead.
Subject vehicle reconnects if connection is interrupted before mission ended.
TLS/DTLS Connection
Subject vehicle establishes initial connetion to Remote Vehicle Operation
Subject vehicle reconnects if connection is interrupted before mission ended.
RVO starts synchronizing functional clock
Mission Confirmation
Remote Vehicle Operation confirms mission to vehicle, and vehicle announces its capabilities to the Remote Vehicle Operation
Safe Vehicle Identification
Vehicle flashes sequence
RVO starts synchronizing safety clock
Safe Vehicle Type Confirmation
Vehicle safely confirms its vehicle type identifier to the Remote Vehicle Operation
Enter Safe Driving State
Vehicle verifies remaining preconditions and waits for first DrivingPermission
In case the mission was assigned before the state *On Hold* was reached:
At this point in the sequence, AVPS needs to waits for the hand-over to be completed.
Enter Safe Driving State
Remote Vehicle Operation accepts the mission with first valid DrivingPermission
Safe Driving State
Safety
RVO continues SafetyTimeSync, Vehicle evaluates DrivingPermissions
Control
RVO continues FunctionalTimeSync, Vehicle follows path accurately
Leave Safe Driving State
Remote Vehicle Operation resigns from its authority and stops guiding the subject vehicle.
Wrap Up
Vehicle reaches secure standstill and disconnects from the Remote Vehicle Operation
Mission finished
- Either because subject vehicle reached destination or because an error occurred
- Transition to *On Hold* , *Ready* or *Inactive*
Authority is either handed back to the user or the vehicle stays in "system management" (ISO) and awaits the next mission.

Figure 8 - Mission sequence overview

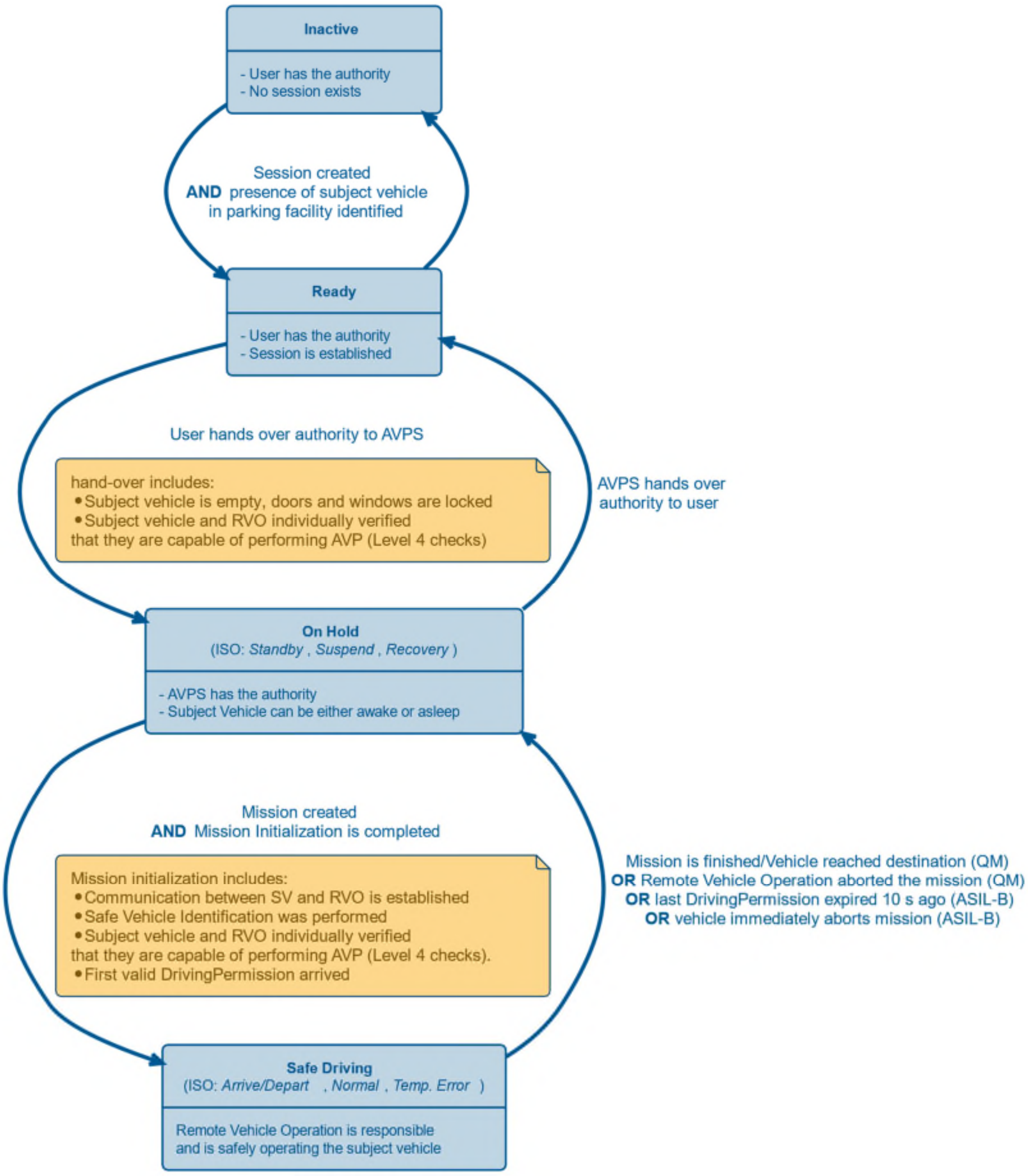

Inactive
- User has the authority
- No session exists
Session created
AND presence of subject vehicle
in parking facility identified
Ready
- User has the authority
- Session is established
User hands over authority to AVPS
hand-over includes:
•Subject vehicle is empty, doors and windows are locked
•Subject vehicle and RVO individually verified
that they are capable of performing AVP (Level 4 checks)
AVPS hands over
authority to user
On Hold
(ISO: Standby , Suspend , Recovery)
- AVPS has the authority
- Subject Vehicle can be either awake or asleep
Mission created
AND Mission Initialization is completed
Mission initialization includes:
•Communication between SV and RVO is established
•Safe Vehicle Identification was performed
•Subject vehicle and RVO individually verified
that they are capable of performing AVP (Level 4 checks).
•First valid DrivingPermission arrived
Mission is finished/Vehicle reached destination (QM)
OR Remote Vehicle Operation aborted the mission (QM)
OR last DrivingPermission expired 10 s ago (ASIL-B)
OR vehicle immediately aborts mission (ASIL-B)
Safe Driving
(ISO: Arrive/Depart , Normal , Temp. Error)
Remote Vehicle Operation is responsible
and is safely operating the subject vehicle

Figure 9 – AVPS state machine

The presence of the Subject Vehicle in the parking facility may be identified by the Subject Vehicle itself, by the Remote Vehicle Operation or can just be confirmed by the User.

For the transition Ready → On Hold, both Subject Vehicle and Remote Vehicle Operation verify individually, that they are capable of performing AVP ("Level 4 Checks"). In this stage, communication is handled through the backends. No direct communication between Subject Vehicle and Remote Vehicle Operation happens.

The transitions Ready → On Hold and On Hold → Safe Driving can be prepared in parallel, because Missions may already be assigned before the hand-over is completed.

By assigning a Mission in Ready, the Mission initialization process can already be performed while hand-over is still in progress. As soon as the hand-over is completed and the state On Hold is reached, the conditions for the transition to Safe Driving may already be fulfilled and the transition can therefore be taken immediately.

7.1 Mission Initialization

7.1.1 New Mission

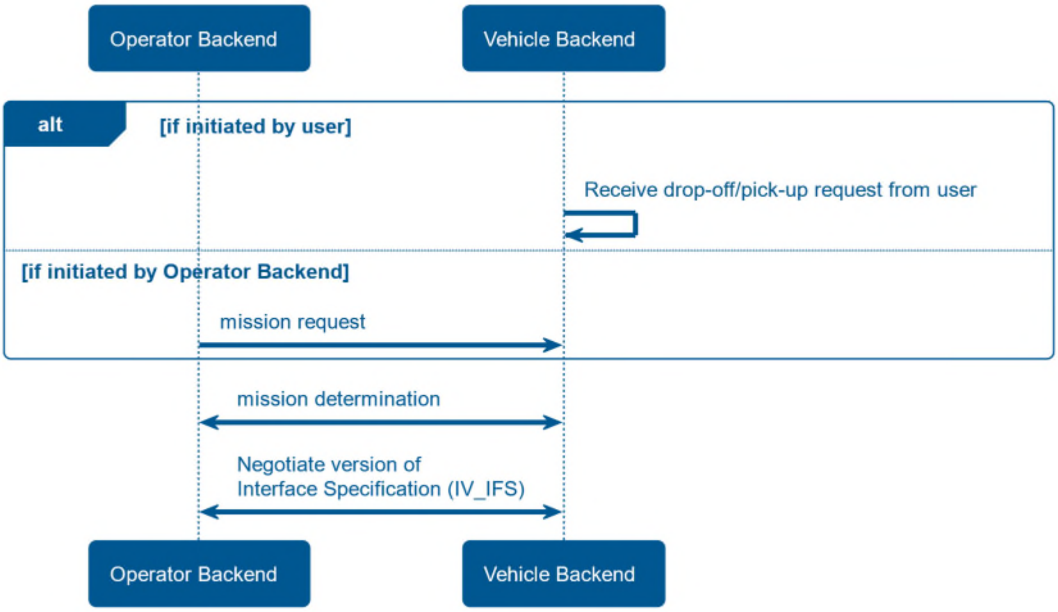


Figure 10 - New Mission

A Mission can either be started by a User through the Vehicle Backend or by the Operator Backend, e.g. to restart a previously aborted Mission or move the Subject Vehicle from one parking spot to another.

Req35: [OperatorBackend] OperatorBackendCreateMissionId

For each Mission, the Operator Backend shall create a unique alphanumerical Mission id randomly.

The mission id shall have a length of exactly 32 characters. To ensure uniqueness across all Remote Vehicle Operation providers, the first 3 characters shall uniquely identify the Remote Vehicle Operation provider, the remaining 29 characters shall be chosen randomly.

The Mission id may contain the characters a-z and 0-9. The mission id shall be case-insensitive.

7.1.2 Wake-Up Subject Vehicle

The Vehicle Backend typically wakes up the Subject Vehicle via cellular network.

For further information about the Wake-up sequence refer to [1], chapter A.2.6.

7.1.3 Certificate and Connection Parameter Exchange

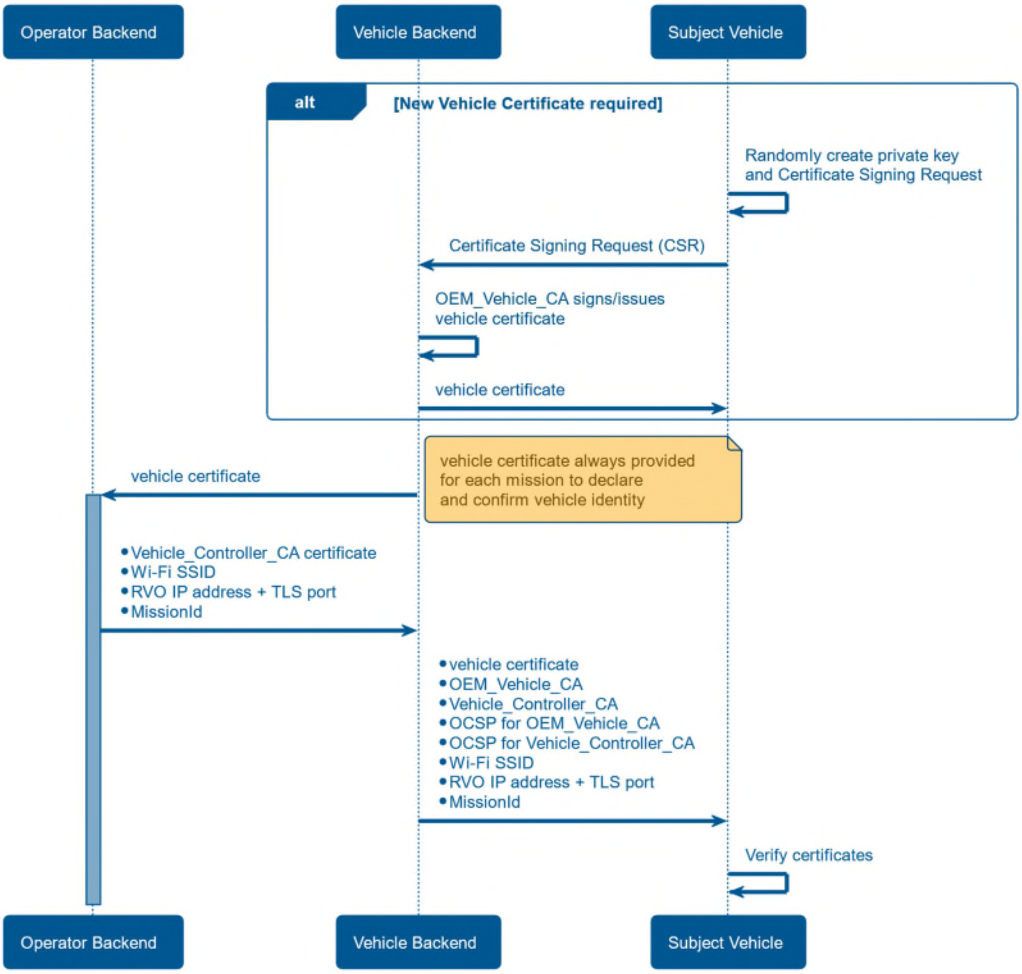

Operator Backend
Vehicle Backend
Subject Vehicle
alt
[New Vehicle Certificate required]
Randomly create private key and Certificate Signing Request
Certificate Signing Request (CSR)
OEM_Vehicle_CA signs/issues vehicle certificate
vehicle certificate
vehicle certificate always provided for each mission to declare and confirm vehicle identity
vehicle certificate
•Vehicle_Controller_CA certificate
•Wi-Fi SSID
•RVO IP address + TLS port
•MissionId
•vehicle certificate
•OEM_Vehicle_CA
•Vehicle_Controller_CA
•OCSP for OEM_Vehicle_CA
•OCSP for Vehicle_Controller_CA
•Wi-Fi SSID
•RVO IP address + TLS port
•MissionId
Verify certificates
Operator Backend
Vehicle Backend
Subject Vehicle

Figure 11 - Provisioning/ Certificate Exchange Sequence

Req36: [Vehicle] VehicleCreatePrivateKeyAndSigningRequest

When necessary, the Subject Vehicle shall generate a new ECDSA P-384 private key randomly and a certificate signing request (CSR) according to chapter E.3, "Vehicle Certificate" and send the CSR to the Vehicle Backend.

Before generating a private key, the Subject Vehicle shall ensure that there is enough entropy available to back the random generation process.

The Subject Vehicle shall only store the private key in a secure location. It shall not be possible to access the private key from outside the respective ECU in any way.

Validity

- There is no maximum time specified for which a private key can be used.
- The private key shall be valid for at least the anticipated duration of the Mission.
- It is recommended to use the same private key for the whole session, because that might have advantages in the future, like e.g. a faster initialization due to skipping the vehicle identification process in certain situations. These will be specified in future versions of the interface specification.
- To increase privacy, it is recommended to randomly create a new private key for each session.

Req37: [VehicleBackend] VehicleBackendCreateVehicleCertificate

When necessary, the OEM_Vehicle_CA authority shall sign/issue a Vehicle Certificate that complies with parameters given in Table 14 - Fields of the end entity certificates used by AVP vehicles. The Vehicle Certificate shall be valid for at least the anticipated duration of the Mission.

Req38: *[General] UseSeparateVehicleCertificates*[4]

If the Subject Vehicle establishes the Wi-Fi connection (i.e. performs the EAP-TLS handshake) and the TLS/DTLS connection from different ECUs, which do not have access to the same secure key storage, the Subject Vehicle shall create two separate private keys and certificate signing requests. One on each of the respective ECUs.

Rationale:

A private key shall always be created on the device that uses it, and it shall never be shared between multiple devices.

Req39: *[General] AgreeOnTrustAnchorExchangeAndRevocationProcess*

The OEM and the provider of the Remote Vehicle Operation shall agree on a process for:

- exchanging the OEM_Root and OEM_Vehicle_CA certificates
- signing the AVP_Vehicle_Controller_CA certificate
- revocation of individual vehicle certificates
- revocation of the AVP_VehicleController_CA certificate

7.1.4 Wi-Fi Connection

A Wi-Fi connection is not mandatory. If supported by the Subject Vehicle and the respective Remote Vehicle Operation, the Subject Vehicle can instead establish the TLS- and DTLS connections to the Remote Vehicle Operation using a cellular network.

[4] derived from [Vehicle] VehicleCreatePrivateKeyAndSigningRequest , [VehicleBackend] VehicleBackendCreateVehicleCertificate

Req40: [Vehicle] VehicleConnectToWifi

When requested by the Vehicle Backend, the Subject Vehicle shall establish a mutually authenticated Wi-Fi connection using EAP-TLS. The Wi-Fi connection shall only be established to the SSID received from the Vehicle Backend for the specific Mission.

The Subject Vehicle shall only accept a TLS handshake with TLS version 1.2 or 1.3 and use the cipher suite ECDHE-ECDSA-AES256-GCM-SHA384.

- The Subject Vehicle shall present the Vehicle Certificate received during the provisioning phase described in chapter 7.1.3, "Certificate and Connection Parameter Exchange ". When separate certificates are used for Wi-Fi and TLS/DTLS connection, the Subject Vehicle shall use the certificate with subject "ST=access" for the Wi-Fi connection.
- The Subject Vehicle shall verify that the certificate presented by the Remote Vehicle Operation is valid, signed by the Vehicle_Controller_CA authority (see AVP_Vehicle_Controller_CA) and contains the subject "ST=access".
- The Subject Vehicle shall verify that the Remote Vehicle Operation presented a valid OCSP response using OCSP stapling and verify that this OCSP response confirms that the presented certificate is valid.

If the Subject Vehicle is unable to connect with the Remote Vehicle Operation within a reasonable amount of time, it shall inform the Vehicle Backend. The Vehicle Backend shall then inform the Operator Backend and abort the Mission (See chapter 7.3.1, "Abort Mission").

The Remote Vehicle Operation shall use DHCP to assign an IPv4 address to the Subject Vehicle. DHCP shall not be used for anything beyond address assignment.

This Requirement is only applicable for Wi-Fi usage.

With respect to Req122: [Vehicle] VehicleAbortsMissionAfter10s, the reconnection must happen within 10 s, otherwise the Subject Vehicle has to abort the mission due to missing DrivingPermissions.

ST is a standard attribute of certificates that means "state or province name".

Req41: [Vehicle] VehicleChangeAccessPoint

The Subject Vehicle shall initially connect to the access point with the highest signal strength.

Vehicle-Controlled Mode

Unless requested otherwise by the Remote Vehicle Operation, the Subject Vehicle shall switch access points automatically, whenever necessary. The Subject Vehicle shall apply an algorithm that minimizes the number of access points switches. The Subject Vehicle shall preferably switch between access points while in Standstill.

Infrastructure-Controlled Mode

If the Subject Vehicle receives the message AccessPointChangeRequest from the Remote Vehicle Operation with a BSSID value other than ANY, the Subject Vehicle shall immediately switch to the access point with the given BSSID.

The Subject Vehicle shall stay connected to that access point, until it receives another AccessPointChangeRequest message or the connection to that access point is interrupted:

- If the Subject Vehicle receives AccessPointChangeRequest.bssid != ANY, it shall immediately switch to the access point with the given BSSID.
- If the Subject Vehicle receives AccessPointChangeRequest.bssid == ANY or if the connection to an access point is interrupted, the Subject Vehicle shall return to the Vehicle-Controlled Mode and automatically select and switch access points.

While in Infrastructure-Controlled Mode, the Subject Vehicle shall not scan for available networks.

⚠ This Requirement is only applicable for Wi-Fi usage.

⚠ The transition time from one access point to another must be minimized to ensure that the DrivingPermission message reaches the Subject Vehicle without avoidable delay. The Remote Vehicle Operation knows the Wi-Fi infrastructure and the desired Path and can therefore optimize the roaming behavior.

7.1.5 TLS/DTLS Connection

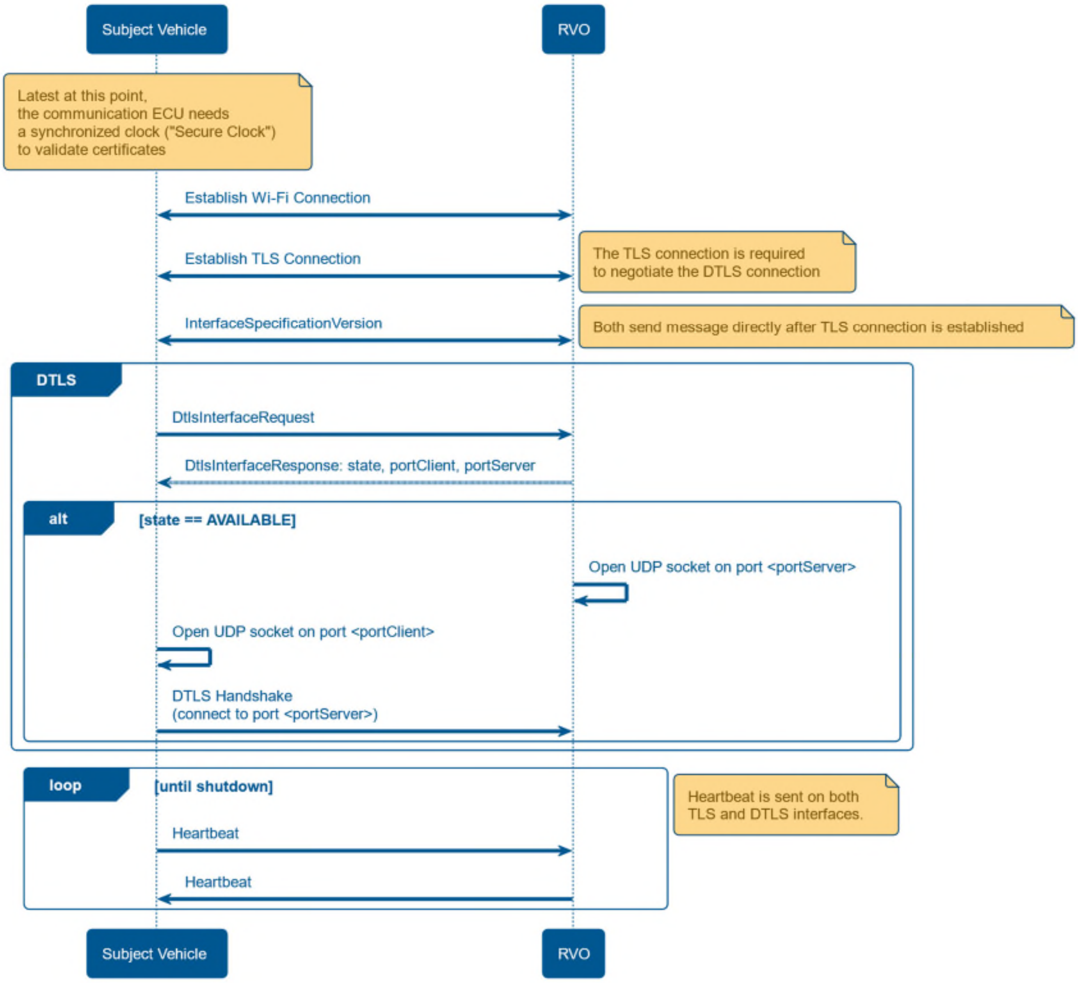

Subject Vehicle
RVO
Latest at this point,
the communication ECU needs
a synchronized clock ("Secure Clock")
to validate certificates
Establish Wi-Fi Connection
Establish TLS Connection
The TLS connection is required
to negotiate the DTLS connection
InterfaceSpecificationVersion
Both send message directly after TLS connection is established
DTLS
DtlsInterfaceRequest
DtlsInterfaceResponse: state, portClient, portServer
alt
[state == AVAILABLE]
Open UDP socket on port <portServer>
Open UDP socket on port <portClient>
DTLS Handshake
(connect to port <portServer>)
loop
[until shutdown]
Heartbeat
Heartbeat
Heartbeat is sent on both
TLS and DTLS interfaces.
Subject Vehicle
RVO

Figure 12 - TLS and DTLS connection process

Req42: [Vehicle] VehicleConnectTlsAndDtls

The Subject Vehicle shall establish both a mutually authenticated TLS and a mutually authenticated DTLS connection to the Remote Vehicle Operation. Both connections shall only be established to the server address and port received from the Vehicle Backend for the specific Mission.

The Subject Vehicle shall only accept a TLS or DTLS handshake with TLS version 1.2 or 1.3 and use the cipher suite ECDHE-ECDSA-AES256-GCM-SHA384.

- The Subject Vehicle shall present the vehicle certificate received during the provisioning phase described in chapter 7.1.3, “Certificate and Connection Parameter Exchange ”. When separate certificates are used for Wi-Fi and TLS/DTLS connection, the Subject Vehicle shall use the certificate with subject "ST=drive" for both TLS- and DTLS connection.
- The Subject Vehicle shall verify that the certificate presented by the Remote Vehicle Operation is valid, signed by the Vehicle_Controller_CA authority (see AVP_Vehicle_Controller_CA) and contains the subject ST=drive.
- The Subject Vehicle shall verify that the Remote Vehicle Operation presented a valid OCSP response using OCSP stapling and verify that this OCSP response confirms that the presented certificate is valid.

If the Subject Vehicle is unable to connect with the Remote Vehicle Operation within a reasonable amount of time, it shall inform the Vehicle Backend. The Vehicle Backend shall then inform the Operator Backend and abort the Mission. (See chapter 7.3.1, “Abort Mission”).

ⓘ Data is only exchanged between Subject Vehicle and Remote Vehicle Operation through the server-side TLS port announced through the Vehicle Backend and the server-side DTLS port announced by the Remote Vehicle Operation in DtlsInterfaceResponse. The Subject Vehicle can choose the client-side ports arbitrarily. Other ports and network services shall not be used.

Req43: [Vehicle] VehicleSupportsInternetProtocols[5]

The Subject Vehicle shall use and support IPv4 addresses assigned using DHCP when using a Wi-Fi connection.

[5] derived from [Vehicle] VehicleConnectTlsAndDtls

Req44: [Rvo] RvoProcessTlsAndDtlsHandshake

The Remote Vehicle Operation shall only accept a TLS or DTLS handshake if all the following conditions are true:

- TLS version 1.2 or 1.3 is used
- The cipher suite ECDHE-ECDSA-AES256-GCM-SHA384 is used
- The certificate presented by the Subject Vehicle is valid and equals the certificate received from the Vehicle Backend along the Mission

If the Remote Vehicle Operation denied an incoming TLS or DTLS handshake associated with an active Mission, it shall inform the Operator Backend and abort the Mission.

An incoming TLS or DTLS handshake can only be associated with a Mission if the certificate presented by the Subject Vehicle is valid, i.e. the Remote Vehicle Operation can't abort a Mission if certificate presented by the Subject Vehicle was not valid.

7.1.5.1 Establish DTLS through TLS connection

- First, the Subject Vehicle establishes a TLS connection to the given address and port.
- When the TLS connection is established, the Subject Vehicle uses this connection to send the message DtlsInterfaceRequest to the Remote Vehicle Operation.
- The Remote Vehicle Operation will open a DTLS socket and reply with the message DtlsInterfaceResponse.
- The DtlsInterfaceResponse contains the port of the DTLS socket to which the Subject Vehicle must then establish a connection.

Req45: [Vehicle] VehicleAmountOfDtlsConnectionAttempts

If the received DtlsInterfaceResponse.state is DENIED, the Subject Vehicle shall request the DTLS connection a second time. If it fails a second time, the Subject Vehicle shall abort the Mission (see chapter 7.3.1, “Abort Mission”).

Req46: [Vehicle] VehicleDisableTcpNagleOption

The Subject Vehicle shall disable Nagle's algorithm on the TCP socket, because the algorithm may cause the bundling of packets and thus delay sending small packets.

7.1.5.2 Heartbeat

Req47: [Vehicle] VehicleSendHeartbeatOnTlsAndDtls

The Subject Vehicle shall send the message Heartbeat to the Remote Vehicle Operation once per second through both the TLS and DTLS interface.

Req48: [Vehicle] VehicleCloseOnMissingHeartbeat

If no Heartbeat has been received from the Remote Vehicle Operation on an interface within 5 seconds, the Subject Vehicle shall close the respective connection.

Req49: [Vehicle] VehicleReconnectConditions

SIL = ASIL-QM

If one or more (TLS/DTLS) interfaces get disconnected after entering the safe driving state, but before the Remote Vehicle Operation signals DriveCommand.action = TERMINATE, the Subject Vehicle shall do all the following:

1. perform an Emergency Stop
2. switch on the warning lights
3. try to reconnect all interfaces.

If both interfaces can be successfully reconnected before the Mission needs to be aborted because of Req122: [Vehicle] VehicleAbortsMissionAfter10s, the Subject Vehicle shall switch off the warning lights and resume the Mission as requested by the Remote Vehicle Operation.

7.1.6 Vehicle Functional Clock Synchronization

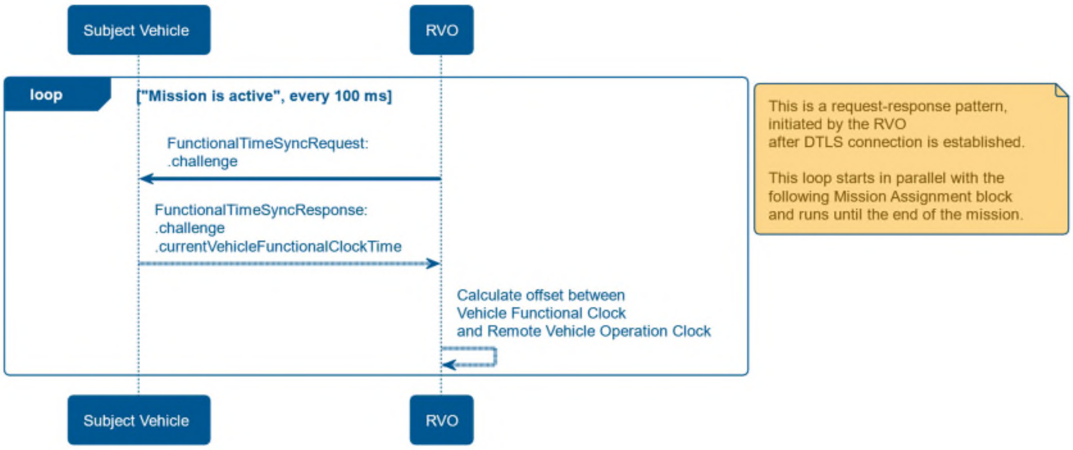

Subject Vehicle
RVO
loop
["Mission is active", every 100 ms]
FunctionalTimeSyncRequest:
.challenge
FunctionalTimeSyncResponse:
.challenge
.currentVehicleFunctionalClockTime
Calculate offset between
Vehicle Functional Clock
and Remote Vehicle Operation Clock
This is a request-response pattern,
initiated by the RVO
after DTLS connection is established.
This loop starts in parallel with the
following Mission Assignment block
and runs until the end of the mission.
Subject Vehicle
RVO

Figure 13 - FunctionalTimeSync mechanism

Req50: [Rvo] RvoDetermineVehicleFunctionalClock

The Remote Vehicle Operation shall determine the time of the Vehicle Functional Clock by sending FunctionalTimeSyncRequest messages to the vehicle and evaluating the corresponding responses.

For maximum accuracy, the Remote Vehicle Operation shall assume, that the times for sending and receiving the request and sending and receiving the response are equal.

The typical uncertainty shall be less than +/- 50 ms, i.e. the Remote Vehicle Operation and Subject Vehicle subsystems shall be designed in a way that the roundtrip time is typically less than 100 ms.

A higher uncertainty might lead to a worse Pose estimation (see Req97: [Vehicle] VehicleReceivesCurrentPose) which in turn might lead to a worse control performance.

If it is known, that the round trip time is not distributed 50:50 between request and response for a certain Remote Vehicle Operation/Subject Vehicle combination, the Remote Vehicle Operation shall use the known distribution.

Req51: [Vehicle] VehicleRespondToFunctionalTimeSyncRequest

When the Subject Vehicle receives the message FunctionalTimeSyncRequest it shall respond with the message FunctionalTimeSyncResponse as fast as possible.

The Remote Vehicle Operation and Subject Vehicle subsystems shall be designed in a way that the roundtrip time is typically less than 100 ms, including processing and transmission time

⚠

The Remote Vehicle Operation needs to send some messages like DtlsInterfaceResponse or FunctionalTimeSyncRequest before the functional time synchronization was performed. In these cases, the Remote Vehicle Operation can use a rough estimation, e.g. based on the timeSent value of the DtlsInterfaceRequest message.

The synchronization only needs to be precise for the signal DetectedVehicle-Pose.measurementTime.

7.1.7 Mission Confirmation

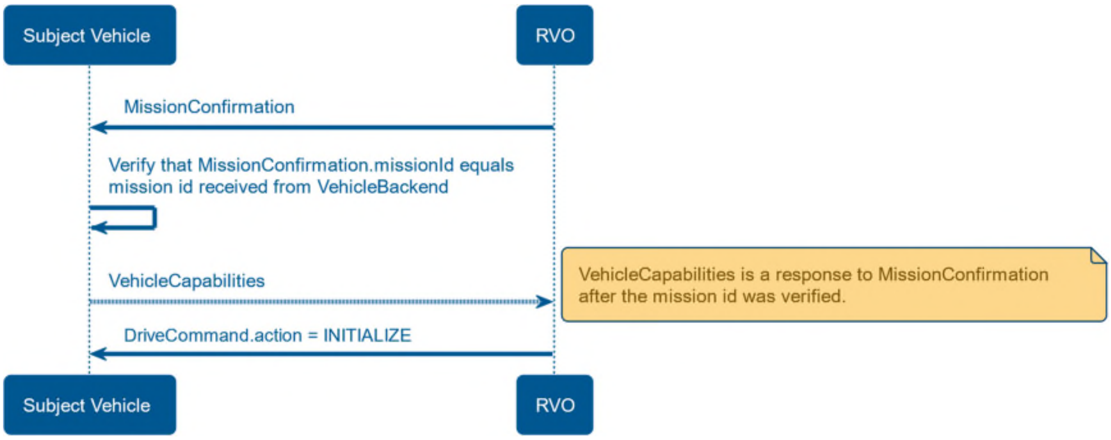

Subject Vehicle
RVO
MissionConfirmation
Verify that MissionConfirmation.missionId equals
mission id received from VehicleBackend
VehicleCapabilities
VehicleCapabilities is a response to MissionConfirmation
after the mission id was verified.
DriveCommand.action = INITIALIZE
Subject Vehicle
RVO

Figure 14 - Mission Confirmation Sequence

Req52: [Rvo] RvoSendsMissionConfirmation

When the TLS and DTLS connection to a new Subject Vehicle is established, the Remote Vehicle Operation shall send the message MissionConfirmation to the Subject Vehicle.

Req53: [Vehicle] VehicleVerifiesMissionId

When the Subject Vehicle receives the message MissionConfirmation, it shall verify that MissionConfirmation.missionId equals the Mission id it received from the Vehicle Backend earlier.

If the Mission ids do not match, the Subject Vehicle shall abort the Mission.

With this check, the vehicle verifies that the connection between Vehicle Backend, Operator Backend and Remote Vehicle Operation is established and that communication is recent.

Req35: [OperatorBackend] OperatorBackendCreateMissionId requires the Mission id to be unique and random.

Req54: [Vehicle] VehicleProvidesCapabilities

After the Subject Vehicle received the message MissionConfirmation, and verified the mission ids, it shall respond with the message VehicleCapabilities.

The message VehicleCapabilities contains non-safety related parameters of the Subject Vehicle.

7.1.8 Safe Vehicle Identification

When the Subject Vehicle reaches the assigned facility with a valid reservation, AVPS performs Subject Vehicle identification. Subject Vehicle identification may be performed at the drop-off area, anywhere inside the operation zone, or at the borders of the operation zone. [1]

The Remote Vehicle Operation/Operator Backend sub-systems shall be capable of identifying the (physical) Subject Vehicle against the V sub-system as the correct communication participant (refer to [1], chapter 8.2.3). Also [1] describes examples of means to perform Subject Vehicle identification (e.g. blink code transmission or license plate recognition).

The following requirements specify the vehicle identification process used in this document. The process is initiated by the Remote Vehicle Operation after the TLS/DTLS connection is established. During this process, the Subject Vehicle flashes a given code through its turn indicator lights.

This is a typical sequence for the vehicle-id process:

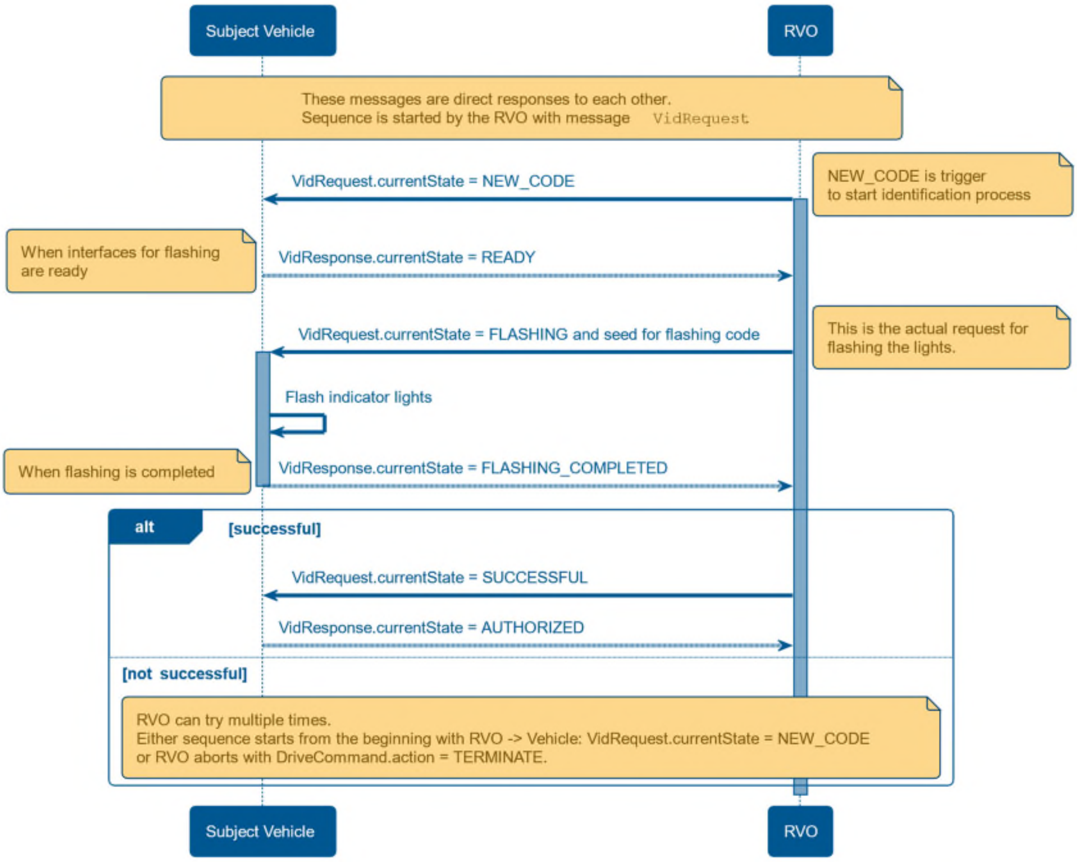

Subject Vehicle
RVO
These messages are direct responses to each other.
Sequence is started by the RVO with message VidRequest
VidRequest.currentState = NEW_CODE
NEW_CODE is trigger
to start identification process
When interfaces for flashing
are ready
VidResponse.currentState = READY
VidRequest.currentState = FLASHING and seed for flashing code
This is the actual request for
flashing the lights.
Flash indicator lights
When flashing is completed
VidResponse.currentState = FLASHING_COMPLETED
alt
[successful]
VidRequest.currentState = SUCCESSFUL
VidResponse.currentState = AUTHORIZED
[not successful]
RVO can try multiple times.
Either sequence starts from the beginning with RVO -> Vehicle: VidRequest.currentState = NEW_CODE
or RVO aborts with DriveCommand.action = TERMINATE.
Subject Vehicle
RVO

Figure 15 - Sequence vehicle-id process

Req55: *[Vehicle] VehicleVehIdReportReadyToStart*

The Subject Vehicle shall send the message VidResponse with state READY if all the following is true:

- the last received DriveCommand.action equals INITIALIZE
- the Subject Vehicle received the message VidRequest with state NEW_CODE
- all interfaces required for flashing the turn indicators are ready
- all turn indicators are turned off (i.e. no warning lights, no active turn indicator)

Req56: [Vehicle] VehicleVehIdFlashLights

The Subject Vehicle shall flash the indicator lights if all the following is true:

- the last received DriveCommand.action equals INITIALIZE
- the Subject Vehicle received the message VidRequest with state FLASHING

The code to be flashed (IndicatorCode) is calculated by applying the following static transformation to the seed (a 64-bit integer) given in the VidRequest message:

IndicatorCode = truncate(seed XOR IndicatorConstant, codeLength)

The integer is truncated to VidRequest.codeLength, representing $2^{codeLength}$ possible combinations, with $8 \leq codeLength \leq 20$.

The Subject Vehicle shall be able to dynamically flash between 8 Bits and 20 Bits, depending on the request from the Remote Vehicle Operation.

ⓘ The constant IndicatorConstant consists of the 20 least significant bits of the constant TransformationConstant, which is used when calculating safety checksums. (See Table 4 in chapter 2.3, “Interface-Specification Version”)

⚠ Whenever any (A)SIL rated requirement or message changes between two released versions of the Interface Specification, the transformation constant which is applied to the vehicle identification seed for calculating safety checksums, will be changed (also see Table 4 in chapter 2.3, “Interface-Specification Version”).

Req57: [Vehicle] VehicleVehIdIndicatorStates

For each code bit of IndicatorCode the following indicator states shall be used:

Table 7 - Indicator states

Code bit value	Left indicator	Right indicator
0	on	off
1	off	on

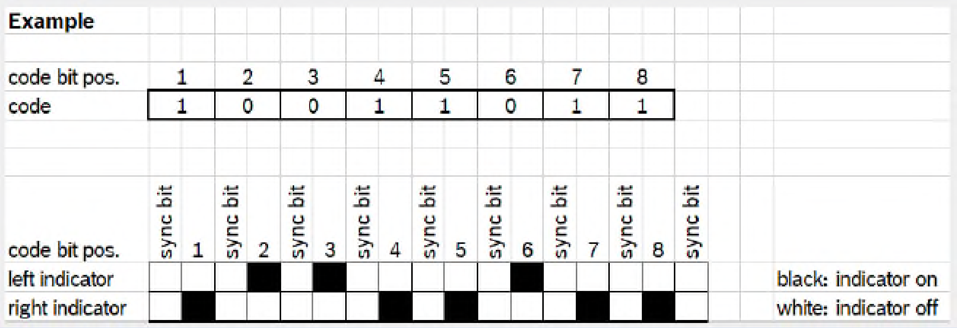
Req58: [Vehicle] VehicleVehIdSyncBit

A synchronization bit (sync bit) between all code bits of IndicatorCode shall be added. The sequence shall start and end with a sync bit.

The sync bit can be realized either as

a) left and right indicator off or

b) left and right indicator on



Example

code bit pos.	1		2		3		4		5		6		7		8		
code	1		0		0		1		1		0		1		1		
code bit pos.	sync bit	1	sync bit	2	sync bit	3	sync bit	4	sync bit	5	sync bit	6	sync bit	7	sync bit	8	sync bit
left indicator	□	□	□	■	□	■	□	□	□	□	□	■	□	□	□	□	□
right indicator	□	■	□	□	□	□	□	■	□	■	□	□	□	■	□	■	□

black: indicator on

white: indicator off

Figure 16 - Indicator activation for an example code realized with variant a)

Using a different indicator state for synchronization helps to avoid issues with capturing video frames with the same state and not being able to distinguish which bit of the message has been received.

Variant a) recommended because it’s expected to be realized with lower effort.

Req59: [Vehicle] VehicleVehIdIndicatorTime

During the flashing of the IndicatorCode the indicators shall change their state (on/ off) after min. 150 ms and max. 400 ms.

Recommendation

200 ms per state, because 8 bit result in a total of relatively short 3,4s. Longer times will lead to a higher duration of the authentification process.

Req60: [Vehicle] VehicleVehIdReportCompleted

The Subject Vehicle shall send the message VidResponse with state FLASHING_COMPLETED if it finished flashing the given code.

Req61: [Vehicle] VehicleVehIdRememberCode

The Subject Vehicle shall send the message VidResponse with state AUTHORIZED if it received the message VidRequest with state SUCCESSFUL.

ⓘ The transformed seed from which the blinking code is derived, is used in chapter 7.5, “Safety Checksums” of safety payloads as a unique identifier of the Subject Vehicle to ensure that the infrastructure communicates with the correct Subject Vehicle only.

From now on, safety payloads can be transmitted between Remote Vehicle Operation and Subject Vehicle.

Req62: [Vehicle] VehicleVehIdRepeat

The Subject Vehicle shall restart the vehicle identification process if it receives the message VidRequest with state NEW_CODE while DriveCommand.action is INITIALIZE.

ⓘ The Remote Vehicle Operation can try the vehicle identification multiple times. If the vehicle identification wasn’t successful multiple times, the Remote Vehicle Operation aborts the Mission by sending DriveCommand.action=TERMINATE (also see chapter 7.3.1, “Abort Mission”).

7.1.9 Vehicle Safety Clock Synchronization

To calculate the Unix timestamp given in DrivingPermission.expirationTime, the Remote Vehicle Operation needs to be able to continuously determine the current time in which the Subject Vehicle operates.

This means that the Remote Vehicle Operation determines the offset between the Remote Vehicle Operation Clock and the Vehicle Safety Clock. Based on this offset it is able to determine the time of the Subject Vehicle.

Clocks gives an overview of all the clocks which are relevant in the AVPS.

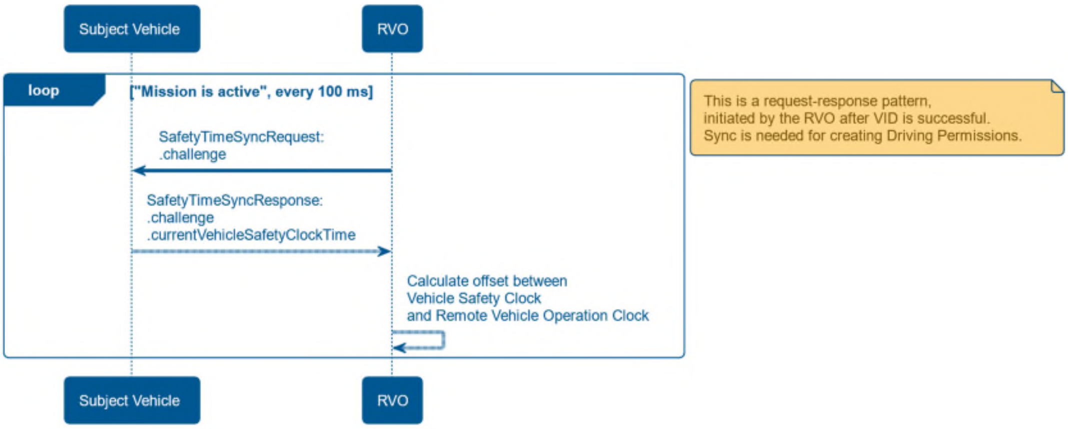

Subject Vehicle
RVO
loop
["Mission is active", every 100 ms]
SafetyTimeSyncRequest:
.challenge
SafetyTimeSyncResponse:
.challenge
.currentVehicleSafetyClockTime
Calculate offset between
Vehicle Safety Clock
and Remote Vehicle Operation Clock
This is a request-response pattern,
initiated by the RVO after VID is successful.
Sync is needed for creating Driving Permissions.
Subject Vehicle
RVO

Figure 17 - SafetyTimeSync mechanism

The process to determine the current Vehicle Safety Clock time is described with the following requirements:

Req63: [Rvo] RvoDetermineOffsetToVehicleSafetyClock

SIL=ASIL-B

The Remote Vehicle Operation shall cyclically determine the clock offset $\tau_{ClockOffset,VehicleSafety}$ between the Vehicle Safety Clock and the Remote Vehicle Operation Clock, by sending a SafetyTimeSyncRequest message every 100 ms to the Subject Vehicle and receiving a SafetyTimeSyncResponse message in response. The clock offset of the SafetyTimeSync with challenge k shall be calculated as follows (see SafetyTimeSyncRequest.challenge for details about the challenge):

$$\tau_{ClockOffset,VehicleSafety}[k] = t_{Vehicle,Response}[k] - t_{RVO,Request}[k]$$

$\boldsymbol{t_{Vehicle,Response}[k]}$ The time of the Vehicle Safety Clock, given in SafetyTi sponse.currentVehicleSafetyClockTime

$\boldsymbol{t_{RVO,Request}[k]}$ Time of the Remote Vehicle Operation Clock when Safe cRequest was sent

This calculation assumes that the trip of SafetyTimeSyncRequest from Remote Vehicle Operation to Subject Vehicle took 0% of the round-trip-time and SafetyTimeSyncResponse from Subject Vehicle to Remote Vehicle Operation took 100% of the round-trip-time. Since it is not possible to determine the exact split between request and response, this results in an uncertainty.

Using this offset only, without considering the uncertainty, the Remote Vehicle Operation would overestimate the time of the Subject Vehicle, i.e. estimate the time higher than it actually is. This is unsafe, because this way a DrivingPermission would expire later than intended (See Req120: [Vehicle] VehicleStopsOnExpirationTimeViolation).

Req65: [Rvo] RvoDetermineSafetyTimeSyncUncertainty and Req66: [Rvo] RvoDetermineCurrentVehicleSafetyClockTime specify how to consider this uncertainty and finally estimate the time of the Vehicle Safety Clock at any point in time.

ASIL-B due to the dependency on Req122: [Vehicle] VehicleAbortsMissionAfter10s due to the dependency on Req120: [Vehicle] VehicleStopsOnExpirationTimeViolation.

A SafetyTimeSyncResponse can be mapped to its request by means of the 16 bit challenge.

When handling safety payloads like DtlsInterfaceRequest and SafetyTimeSyncResponse, both Subject Vehicle and Remote Vehicle Operation always needs to consider and verify the safety checksums: See Req125: [Vehicle] VehicleVerifiesSafetyChecksums and Req126: [Rvo] RvoVerifiesSafetyChecksums.

Req64: [Rvo] RvoDetermineSafetyTimeSyncRoundTripTime

SIL=ASIL-B

The Remote Vehicle Operation shall calculate the round-trip-time $\tau_{RoundTripTime}[k]$ of a particular SafetyTimeSync with identifier k as follows:

$$\tau_{RoundTripTime}[k] = t_{RVO,Response}[k] - t_{RVO,Request}[k]$$

$\boldsymbol{t_{RVO,Request}[k]}$	Remote Vehicle Operation Clock time when SafetyTimeSyncRequest was sent
$\boldsymbol{t_{RVO,Response}[k]}$	Remote Vehicle Operation Clock time when SafetyTimeSyncResponse was received

Req65: [Rvo] RvoDetermineSafetyTimeSyncUncertainty

SIL=ASIL-B

The Remote Vehicle Operation shall determine the uncertainty $\tau_{safetyTimeSyncUncertainty}[k](t_{RVO})$ of a particular SafetyTimeSync with identifier k for time t_{RVO}, by calculating:

$$\tau_{safetyTimeSyncUncertainty}[k](t_{RVO}) = \tau_{RoundTripTime}[k] + \tau_{ClockDrift,Vehicle}[k](t_{RVO})$$

$\boldsymbol{t_{RoundTripTime}[k]}$

Duration between sending a SafetyTimeSyncRequest and receiving the corresponding SafetyTimeSyncResponse of SafetyTimeSync k.

$\boldsymbol{t_{ClockDrift,Vehicle}[k](\ t_{RVO})}$

The estimated clock drift of the Vehicle Safety Clock since the corresponding SafetyTimeSyncRequest of SafetyTimeSync k was sent.

Example: Assuming a clock drift of 10 %:

$$\tau_{ClockDrift,Vehicle}[k](t_{RVO}) = 0.1 \cdot (t_{RVO} - t_{RVO,Request}[k])$$

Req66: [Rvo] RvoDetermineCurrentVehicleSafetyClockTime

SIL=ASIL-B

The Remote Vehicle Operation shall estimate the current time of the Vehicle Safety Clock by calculating:

$$t_{VehicleSafety,now} = t_{RVO,now} + \tau_{ClockOffset,Vehicle}[k] - \tau_{safetyTimeSyncUncertainty}[k](t_{RVO,now})$$

For the calculation, the Remote Vehicle Operation shall select the k for which $\tau_{safetyTimeSyncUncertainty}[k](t_{RVO,now})$ is smallest, but only considering time synchronizations within the last 10 s.

Req67: [Vehicle] VehicleRespondToSafetyTimeSyncRequest

SIL=ASIL-B

When the Subject Vehicle receives the message SafetyTimeSyncRequest it shall verify the safety checksum (See Req125: [Vehicle] VehicleVerifiesSafetyChecksums) and respond with the message SafetyTimeSyncResponse as fast as possible.

Req68: [Rvo] SafetyTimeSyncRoundTripTimeBudget[6]

Subject Vehicle and Remote Vehicle Operation shall be designed in a way that a complete round trip of a SafetyTimeSync is possible in typically less than 100 ms.

The round-trip time is measured from when the Remote Vehicle Operation generates the message SafetyTimeSyncRequest in its internal safety component until the very same component receives the associated SafetyTimeSyncResponse.

The following nominal time budgets are assigned to the subsystems of the AVPS:

Remote Vehicle Operation (2* 10 ms)

The message forwarding and signal processing within the Remote Vehicle Operation shall take less than 10 ms in each direction

Wireless data transmission (2* 10 ms)

The transmission of the messages SafetyTimeSyncRequest and SafetyTimeSyncResponse shall typically take less than 10 ms in each direction.

Subject Vehicle (60 ms)

The processing of the message SafetyTimeSyncRequest and generation of SafetyTimeSyncResponse shall take less than 60 ms.

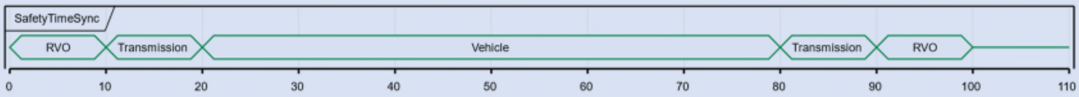


Figure 18 - SafetyTimeSync Time Budget

⚠ If the round-trip-time exceeds the budget rarely, it won’t influence the Subject Vehicle's movement, because only the best synchronization within the last 10 s is used (Req66: [Rvo] RvoDetermineCurrentVehicleSafetyClockTime). Since the round-trip-time is directly incorporated into the estimated time sync uncertainty (Req65: [Rvo] RvoDetermineSafetyTimeSyncUncertainty) and the time sync uncertainty reduces the time which the Subject Vehicle is allowed to drive (See Req118: [Rvo] RvoDrivingPermissionTimeBudget), it is crucial that Subject Vehicle and Remote Vehicle Operation adhere to this budget most of the time, because otherwise the Subject Vehicle won’t be able to drive.

[6] derived from [Rvo] RvoDetermineOffsetToVehicleSafetyClock, [Vehicle] VehicleDrivingPermissionTimeBudget

7.1.10 Safe Vehicle Type Confirmation

Req69: [Vehicle] VehicleProvidesVehicleTypeIdentifier

SIL=ASIL-A

After performing the vehicle identification process, the Subject Vehicle shall send the message SafeVehicleTypeConfirmation with its Vehicle Type Identifier to the Remote Vehicle Operation.

ⓘ The Vehicle Type Identifier is agreed upon between OEM and the Remote Vehicle Operation provider.

ⓘ From the Vehicle Type Identifier, the vehicle-specific properties such as wheelbase, front and rear overhangs etc. can be inferred.

ⓘ Operator Backend and Remote Vehicle Operation already received the Vehicle Type Identifier with QM rating from the Vehicle Backend when the Session was initiated.

The message SafeVehicleTypeConfirmation from the Subject Vehicle is an additional confirmation with ASIL-A rating.

7.1.11 Enter Safe Driving State

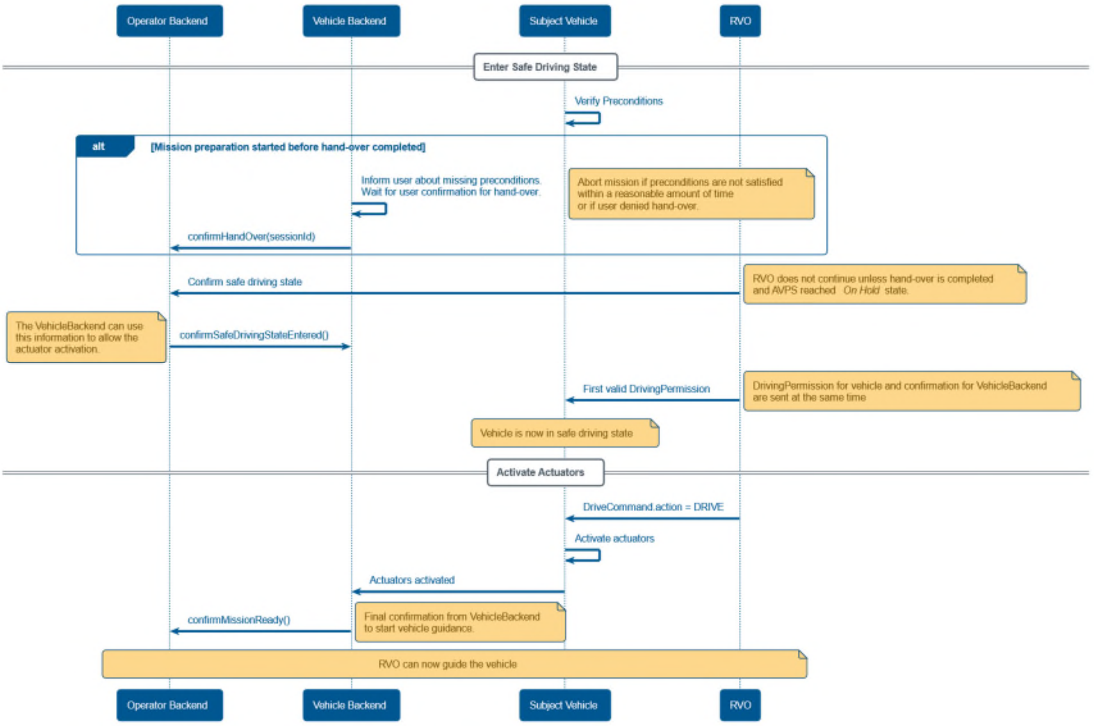

Operator Backend
Vehicle Backend
Subject Vehicle
RVO
Enter Safe Driving State
Verify Preconditions
alt
[Mission preparation started before hand-over completed]
Inform user about missing preconditions.
Wait for user confirmation for hand-over.
Abort mission if preconditions are not satisfied
within a reasonable amount of time
or if user denied hand-over.
confirmHandOver(sessionId)
Confirm safe driving state
RVO does not continue unless hand-over is completed
and AVPS reached *On Hold* state.
The VehicleBackend can use
this information to allow the
actuator activation.
confirmSafeDrivingStateEntered()
First valid DrivingPermission
DrivingPermission for vehicle and confirmation for VehicleBackend
are sent at the same time
Vehicle is now in safe driving state
Activate Actuators
DriveCommand.action = DRIVE
Activate actuators
Actuators activated
confirmMissionReady()
Final confirmation from VehicleBackend
to start vehicle guidance.
RVO can now guide the vehicle
Operator Backend
Vehicle Backend
Subject Vehicle
RVO

Figure 19 - Enter Safe Driving State Sequence

Req70: [Rvo] RvoConfirmsSafeDrivingState

The Remote Vehicle Operation shall confirm the safe driving state to the Operator Backend and send the first valid DrivingPermission to the Subject Vehicle when all the following is true:

- AVPS is in the on hold state, i.e. hand-over of the Subject Vehicle to AVPS is completed
- The Remote Vehicle Operation completed its initialization and is ready to start operating the Subject Vehicle

Req71: [Vehicle] VehicleEntersSafeDrivingState

When the Subject Vehicle receives the first valid DrivingPermission and all preconditions are satisfied (see Req73: [Vehicle] VehicleVerifiesPreconditionsBeforeEnteringSafeDrivingState), it shall enter the Safe Driving State.

ⓘ See Valid Driving Permission for the definition of "valid DrivingPermission".

⚠ Only after entering the Safe Driving State, the vehicle is allowed to activate the powertrain / start the engine (See Req76: [Vehicle] VehicleActivateActuators).

Req72: [Vehicle] VehicleLeavesSafeDrivingState

When any of the following conditions is true, the Subject Vehicle shall leave the Safe Driving State:

- The last DrivingPermission expired 10 s ago (See Req122: [Vehicle] VehicleAborts-MissionAfter10s)
- There is a reason for the vehicle to immediately abort the Mission (See Req87: [Vehicle] VehicleAbortsMissionOnCriticalVehicleFailure)
- The Mission is finished and the vehicle is requested to leave the Safe Driving State (See Req123: [Vehicle] VehicleSecuresOnFinishedMission)

Req73: [Vehicle] VehicleVerifiesPreconditionsBeforeEnteringSafeDrivingState

At the latest before entering the safe driving state, the Subject Vehicle shall verify that all the following conditions are satisfied:

- Subject Vehicle is in secure standstill (Secure Standstill)
- ignition is off
- the Subject Vehicle is locked
- all windows are closed
- all doors are closed
- the trunk is closed

- the Subject Vehicle has enough traction energy for a 30 min / 5 km drive (e.g. fuel, State of Charge, …)
- there are no critical failures (Req10: [Vehicle] VehicleFailureAnalysis)
- tire pressure control does not indicate a flat tire, if possible
- there is no trailer electrically connected to the Subject Vehicle
- the Subject Vehicle is not connected to an electric charger or petrol nozzle
- (in case of a convertible): the rooftop is closed
- There are no people or animals in the Subject Vehicle, if possible (Req27: [App] AppUserConfirmsEmptyVehicle)

If a precondition that can be resolved by the user is not satisfied and the hand-over is not completed yet, the user shall be informed and asked to resolve the respective precondition.

If the preconditions are not satisfied within a reasonable amount of time, the Subject Vehicle shall abort the Mission.

ⓘ These conditions are continuously monitored while in safe driving state. See Req85: [Vehicle] VehicleDetectsHumanIntervention and Req87: [Vehicle] VehicleAbortsMissionOnCriticalVehicleFailure.

Req74: [Vehicle] VehiclePreventsAccessFromOutsideInSafeDrivingState

While in Safe Driving State, doors, trunk and windows shall be closed and locked in a way, that access from outside is prevented.

ⓘ Hence, the vehicle shall not enter the Safe Driving State unless doors, trunk and windows are closed and locked.

Req75: [Vehicle] VehicleAllowsDoorOpeningFromInsideInSafeDrivingState

While in Safe Driving State, it shall be possible to open doors from the inside such that passengers can leave the Subject Vehicle if necessary. This does not apply if the door is equipped with an activated child safety lock.

ⓘ Passengers should not be in the Subject Vehicle during a Session; but in case they are, this requirement is important.

Req76: [Vehicle] VehicleActivateActuators

If the Subject Vehicle receives DriveCommand.action = DRIVE while in Safe Driving State, it shall perform all necessary tasks for driving (e.g. turn ignition on, activate actuators, activate power train).

The Subject Vehicle shall not activate actuators when it is not in Safe Driving State.

Such tasks are vehicle-specific but may include turning on ignition, deactivating the immobilizer, validating the car key etc.

7.2 Safe Driving

In the AVPS, the functional control loop and safety mechanisms are completely decoupled (see Figure 20). While the control loop decides how fast and in which direction the Subject Vehicle shall drive, the safety chain can only decide whether to stop with maximum deceleration (see Emergency Stop), abort the Mission (see chapter 7.3.1, “Abort Mission”) or keep driving.

More details about the driving modes (Type 2.1, 2.2, 2.3) can be found in the chapters 7.2.3 and 7.2.4. The safety concept is explained in chapter 7.2.5.

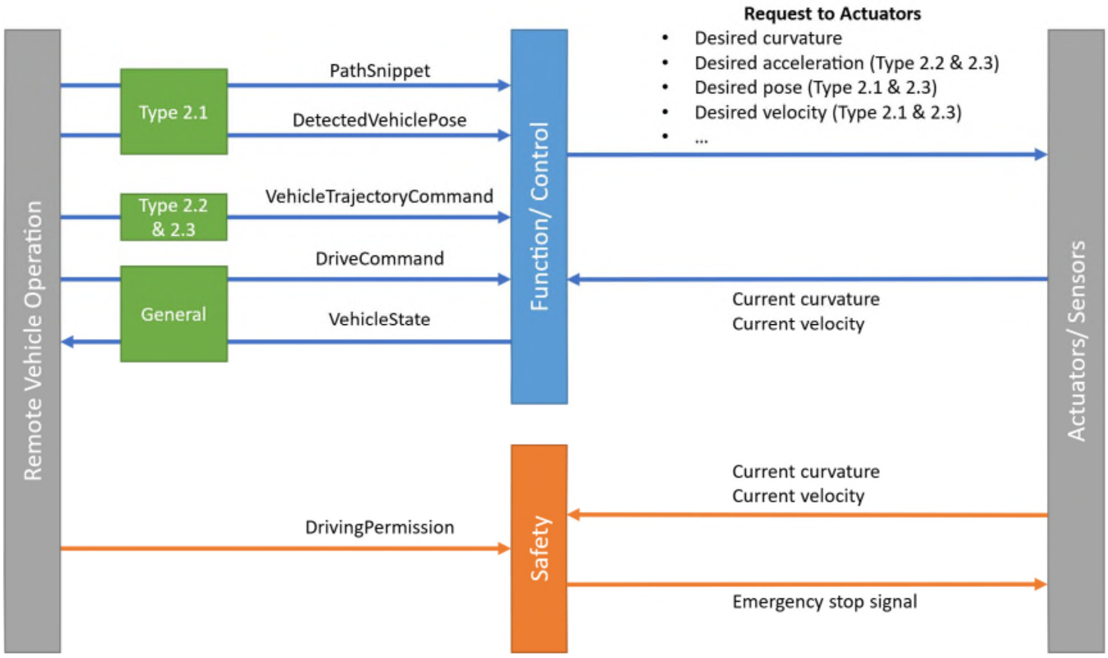


Figure 20 - Separation of Function and Safety

7.2.1 General requirements for automated vehicle operation functions

The relationship of the operation function is described in Figure 6.

7.2.1.1 Principles for performing automated vehicle operation

It should be recognized by system designers that safe operation of an unoccupied vehicle shall be given the highest priority among the overall objectives of AVPS.

The aim for the performance of the operation functions by AVPS is to be at least equal to or better than the driving capabilities of an experienced and attentive human driver (i.e. recognizing the environment, planning the reaction, and operating the vehicle). [1] Also see chapter 7.2.5.4, “Expiration Time “.

7.2.1.2 Operational design domain

Both Remote Vehicle Operation and V sub-systems shall predetermine their ODD. Automated vehicle operation is possible only when ODD conditions for both sub-systems are satisfied.

For example, Subject Vehicle on-board sensors and PFE used for Object detection may have different operational limits. Automated vehicle operation is not possible when the environmental condition fulfils the ODD requirements of one of the sub-systems but not the other.

At a minimum, AVPS shall be capable of performing automated vehicle operation within indoor parking facilities with a paved surface, where the effect of weather is minimized. Operation in outdoor facilities is optional. Therefore, requirements for operation on rough surfaces and/or under various weather conditions are not specified in this document. [1]

7.2.1.3 Requirements for Dynamic Driving Task (DDT)

General requirements for Dynamic Driving Task

System designers shall consider improving the performance beyond the minimum requirements during specific driving scenarios (e.g. to increase distances to other facility users under clear traffic conditions).

It is strongly recommended that OEDR be designed in a conservative and safety-oriented manner to avoid unnecessary risks and loss of social acceptance by commanding unnecessarily aggressive Vehicle Motion Control which may be perceived as threatening by other facility users. [1]

Basic performance requirements

The following requirements apply during normal operation in both mixed and exclusive traffic environment.

- AVPS shall be capable of using braking or evasive maneuvers in order to avoid reasonably foreseeable and preventable collisions which may cause damages to Object or Subject Vehicle.
  - AVPS shall be capable of providing smooth braking and acceleration and avoid sudden braking or acceleration of the Subject Vehicle under non-emergency/non-hazardous situations.
  - AVPS shall be capable of inducing full braking force of the Subject Vehicle in an attempt to avoid a collision in emergency/hazardous situations. (See Req11: [Vehicle] VehicleBrakingPerformance and Req112: [Rvo] RvoPreventsCollisions)
- AVPS shall be capable of continuously monitoring the driving environment, other facility users, system conditions, and behaviors of the Subject Vehicle.
- AVPS shall be capable of operating the Subject Vehicle to the assigned destination, following the planned Route.
  - AVPS shall be capable of physically maintaining the Subject Vehicle within the operation zone, and within the permitted boundaries if applicable.
  - AVPS shall be capable of controlling the Subject Vehicle in such a way that the Subject Vehicle observes the pre-determined traffic rules within the facility.
- AVPS shall be capable of responding to remote assistance commands (e.g. to pause at a location or to proceed to a way point), and shall determine when to start maneuvering within a reasonable amount of time.
  - AVPS shall prioritize remote assistant commands to stop, unless said command is determined by AVPS to potentially create a hazardous situation.
  - AVPS may pause the Subject Vehicle until the conditions are determined suitable to respond to remote assistance commands to proceed to a certain location.
- AVPS shall be capable of controlling the Subject Vehicle to perform all of the parking maneuvers specified in ISO 20900, including compliance to the requirements of the end position accuracy.
- AVPS shall be capable of positioning the Subject Vehicle within the pick-up area so as to facilitate smooth boarding of the User and passengers as well as the loading of goods when applicable. [1]
- Also consider the following requirements:
  - Req99: [Vehicle] VehicleProvidesSpeedControl
  - Req86: [Vehicle] VehicleLimitsSpeedInSafeDrivingState

Additional requirements for operation under mixed traffic environment

The following additional requirements apply when AVPS is operated in mixed traffic environment.

- AVPS shall be capable of avoiding collisions with other facility users, especially with VRUs.
    - AVPS shall be capable of controlling the Subject Vehicle so that the Subject Vehicle does not approach other facility users in the Subject Vehicle's direction of travel closer than 100 cm. This distance may be reduced to 40 cm when other facility users are not in the Subject Vehicle's direction of travel.
        - D_des_fu ≥ 100 cm, when other facility users are in the direction of travel.
        - D_des_fu ≥ 40 cm, when other facility users are not in the direction of travel.
    - This distance may not be achievable under reasonably unforeseeable situations. Examples of such limitations are provided in [1], chapter 6.2.3.3.
- AVPS shall be capable of maintaining a distance longer than 200 cm to preceding vehicles while traveling on a driveway under non-emergency/non-hazardous situations. Note, that this requirement does not apply to parking manoeuvres.
    - D_des_gap ≥ 200 cm
- AVPS shall be capable of controlling the Subject Vehicle so that the Subject Vehicle does not block the pathway of other facility users.
    - AVPS shall be capable of controlling the Subject Vehicle so that the Subject Vehicle does not approach Objects other than facility users (e.g. walls, pillars, parked vehicles) closer than 40 cm when traveling faster than 5 km/h.
        - D_des_ob ≥ 40 cm when V_des_sv ≥ 5 km/h (See Req86: [Vehicle] VehicleLimitsSpeedInSafeDrivingState)
- AVPS shall determine a suitable stop position of the Subject Vehicle relative to the destination so as not to disrupt the facility operation and/or interfere with other facility users. For example, when a parked vehicle is located in the area adjacent to the destination, a sufficient distance should be ensured so that the doors of the adjacent vehicle can be opened without contacting the Subject Vehicle.
- AVPS shall be able to avoid creating unnecessary traffic congestion. For example, by approaching too close to vehicles that are attempting to park, or by entering an intersection when the intended direction of travel is blocked.
- AVPS shall operate the Subject Vehicle in accordance with all local road and vehicle regulations. For example, to use the Subject Vehicle's turn signals, turn on head lamps. [1]

Examples of scenarios in mixed traffic environments are listed in [1], chapter 6.2.3.2.

Hazardous situations with moving Objects (e.g. pedestrians, bicyclists, and vehicles) that are not reasonably avoidable may occur in mixed traffic conditions. A comparison with the driving capability of an experienced and attentive driver should be considered as the threshold for such situation. These scenarios are covered in the test scenarios defined in [1], chapter 10.

Typical but not exhaustive situations where AVPS may not be able to avoid collisions are as follows:

- A bicyclist or pedestrian approaches from an occluded lateral position where time to collision (TTC) is less than the time required by a human driver to stop the vehicle by applying full braking force.

- A parked vehicle suddenly drives out of a parking spot close to the Subject Vehicle with high acceleration while the Subject Vehicle is already near the parked vehicle.
- A pedestrian who deliberately tests the system's capabilities and suddenly jumps into the Subject Vehicle pathway directly in front of the Subject Vehicle. [1]

7.2.1.4 Requirements for emergency stopping

AVPS shall be capable of bringing the Subject Vehicle to a controlled stop under the situations defined in the following sub-clauses, at a minimum.

- The applied braking force to stop the Subject Vehicle depends on the driving situation. Other facility users, including vehicles approaching from behind, shall also be considered when performing an Emergency Stop.
- As soon as an Emergency Stop is initiated, the hazard lights shall be turned on, unless otherwise regulated. They may be switched off when the situation is mitigated, and traffic is not obstructed (see Req124: [Vehicle] VehicleMissionAbort).
- Human interaction is needed to resume automated vehicle operation after emergency stopping has occurred (see chapter 4.1.7, "Response to incapacitation of the operation functions").

As DDT fallback AVPS shall be capable in bringing the Subject Vehicle to a stable stopped condition in response to the following events.

- Occurrence of a failure in any sub-system which may or has become safety relevant (the type of failure that triggers an Emergency Stop is design specific)
- Either or both Remote Vehicle Operation and V sub-systems are approaching ODD boundaries, or the driving environment changes such that the ODD is no longer fulfilled (e.g. blackouts, communication loss). [1]

7.2.1.5 Requirements for destination assignment

- The assigned destination shall be suitable in size for the Subject Vehicle to park, including any external attachments/ loads (if applicable).
  - In some cases, a destination may be assigned so that part of the vehicle or its external attachments/loads exceed the boundaries of the destination. In this case, the destination shall be assigned such that the portion of the vehicle exceeding the boundaries of the destination does not obstruct the pathway of other vehicles. For exclusive traffic environments, vehicles that exceed the boundaries of destinations should be allowed to enter the operation zone only when such a non-interfering assignment is possible[7]. For example, by assigning a corner location of a parking facility to oversized vehicles, so that the obstructed pathway is not used by other vehicles. As another example, if a pathway becomes obstructed, the pathway may be excluded from the drivable area.

[7] AVPS under exclusive traffic conditions are not required to detect irregular objects, including those protruding into the pathway for the Subject Vehicle to travel.

- If the destination becomes unavailable during operation, an alternative destination shall be assigned.
- When a new destination is assigned via remote assistance during automated vehicle operation, the change shall be accepted by AVPS. [1]

7.2.1.6 Requirements for route planning

The Remote Vehicle Operation sub-system shall plan the Route for the Subject Vehicle to reach the assigned destination. The following requirements shall apply.

- The Route shall be created to reach the destination without violating traffic and facility-specific rules (e.g. driving the wrong way in a one-way traffic zone).
  - If way points (intermediate destinations) are provided, the Route shall be created to reach the destination via these way points.
- The most efficient Route to the destination should be created
- AVPS shall initiate one of the following actions if blocked areas are recognized in the planned Route:
  - determine a new path within the Route for the Subject Vehicle to avoid the blockage without changing the Route, for example, to steer around a temporary construction zone.
  - plan a new Route or change the destination
  - communicate a request to the P sub-system to clear the blocked condition.
- When a new Route is assigned via remote assistance, the new Route shall be used accordingly. [1]

7.2.2 Requirements to sub-systems

Req77: [Vehicle] VehicleActivatesBrakeLight

The Subject Vehicle shall activate its brake lights according to the current driving condition.

Req78: [Vehicle] VehicleActivatesReverseLight

The Subject Vehicle shall activate its reverse lights according to the current driving condition.

Req79: [Vehicle] VehicleActivatesIndicatorLightsOnRequest

The Subject Vehicle shall control the indicator lights (off, left, right, warning) depending on the command DriveCommand.directionIndicator.

Req80: [Vehicle] VehicleActivateHeadlights

In Safe Driving State, the Subject Vehicle shall turn its headlights and taillights on, unless forbidden by local law.

Req81: [Vehicle] VehicleSuppressHighBeamLights

In Safe Driving State, the Subject Vehicle shall suppress the high beams (risk of blinding if accidentally left on by the driver).

Req82: [Vehicle] VehicleDataLogging

The Subject Vehicle shall record all data relevant for reconstructing its internal behavior in case of an incident. This data may e.g. include relevant bus signals, longitudinal and lateral control requests, actuator states, requests received from the Remote Vehicle Operation, etc.

With respect to [1] AVPS shall record and store the following data at a minimum:

- Video image of the surroundings of the Subject Vehicle while automated vehicle operation is being performed. Note that this may be done by a vehicle on-board camera or camera installed in the facility, and these cameras may also serve different purposes.
- Data log of the following events:
  - Change in system states
  - Communication within the system
  - Suspend condition codes

The granularity of the data (e.g. resolution of the video image) and its storage duration should be determined by the relevant stakeholders in the area of system implementation, as certain countries or regions may have specific requirements on privacy and ownership of such data. [1]

The Subject Vehicle shall either store this data internally, send it to the Vehicle Backend or to the Remote Vehicle Operation as RecordedMessages message.

ⓘ This is required by authorities and furthermore needed in order to allow proper analysis of incidents and answer liability related questions.

ⓘ The required storage duration is currently in discussion between data protection and liability experts but should be at least 1 week, starting when the Subject Vehicle was handed back to the User.

Req83: [Vehicle] VehicleDataLoggingToInfrastructure[8]

This requirement is applicable if the Subject Vehicle sends recordings to the Remote Vehicle Operation.

If the Subject Vehicle received MissionConfirmation.recordingLevel = NORMAL, the Subject Vehicle shall record basic data that allows reconstruction of the vehicles externally visible behavior (e.g. how fast did it drive, when and why did it stop etc.).

If the Subject Vehicle received MissionConfirmation.recordingLevel = VERBOSE, the Subject Vehicle shall record all previously mentioned data, as well as additional information that could be helpful for debugging internal behaviour.

To transfer recorded messages to the Remote Vehicle Operation, the Subject Vehicle shall serialize each message and embed it in a RecordedMessage message.

The Subject Vehicle shall accumulate up to 500 RecordedMessage messages in one RecordedMessages message.

The Subject Vehicle shall only send the next RecordedMessages message if the transport layer indicates that the previous one was sent successfully.

Req84: [Rvo] RvoDataLogging

The Remote Vehicle Operation shall record and store all data relevant for reconstructing its behaviour in case of an incident, as well as all data sent to and received from the Subject Vehicle.

The Remote Vehicle Operation shall store all data received from the Subject Vehicle in messages of type RecordedMessages.

ⓘ (1) This is required by authorities and furthermore needed in order to allow proper analysis of incidents and answer liability related questions.

[8] derived from [Vehicle] VehicleDataLogging

(2) The required storage duration is currently in discussion between data protection and liability experts, but should be at least 1 week, starting when the Subject Vehicle was handed back to the user.

Req85: [Vehicle] VehicleDetectsHumanIntervention

SIL=ASIL-QM

If the Subject Vehicle detects an AVP related human intervention while in Safe Driving State, the Subject Vehicle shall abort the Mission (See chapter 7.3.1, “Abort Mission”).

To do so the AVPS shall be capable of recognizing the User’s intent to interact with the Subject Vehicle. Also, the AVPS shall be capable of recognizing overriding activities while the system is operating the Subject Vehicle. For example, an occupant who did not leave the Subject Vehicle at the drop-off area may open the door, operate the steering wheel, accelerator/brake pedal, or other interfaces defined by the system designer. These activities are considered as misuse by the User. In addition, AVPS shall be capable of recognizing unexpected activities of other facility users from outside the Subject Vehicle. For example, other facility users may pull the door handle or try to open the trunk lid of the Subject Vehicle. [1], chapter 9.3.3.18

Only reliable sensors shall be considered for detecting human interventions, to prevent accidental aborts.

AVP relevant human interventions are any of the following:

- at least one door has been opened from the inside
- at least one door incl. trunk has been attempted to be opened from the outside
- the engine bonnet has been opened
- the vehicle key has been pressed
- the ignition switch has been pressed
- the gas pedal has been pressed
- the brake pedal has been pressed
- the steering wheel has been manually turned or held
- a manual gear change was attempted
- the parking brake switch was activated
- in-cabin surveillance has detected a person inside the Subject Vehicle

Req86: [Vehicle] VehicleLimitsSpeedInSafeDrivingState

SIL=ASIL-C

In Safe Driving State, the Subject Vehicle shall limit its velocity to 2.8 m/s. If the Subject Vehicle accelerates above this threshold, the Subject Vehicle shall abort the Mission (see chapter 7.3.1, “Abort Mission”).

> The cause for unintended acceleration in the parking facility could be the powertrain or rolling down a ramp.

Req87: [Vehicle] VehicleAbortsMissionOnCriticalVehicleFailure

SIL=ASIL-B

If the Subject Vehicle detects a critical failure (as identified in the failure analysis see Req10: [Vehicle] VehicleFailureAnalysis) it shall perform an Emergency Stop and abort the Mission (see chapter 7.3.1, “Abort Mission”).

> It is required that the OEM identifies which Subject Vehicle failures are critical (e.g. in an FMEA) and ensures that such failures are detected and safe state is reached within the specified FTTI.

Req88: [Vehicle] VehicleDoesNotUseRearAxleSteering

SIL=ASIL-B

The Subject Vehicle shall not use rear axle steering, i.e. it shall use a fixed steering angle $\delta_R = 0°$ for rear wheels.

> Rear axle steering will be supported in future interface specifications. Supporting rear axle steering requires changes in the path interface as well as the DrivingPermissions.

7.2.3 Control: Path Interface (Type 2.1)

This chapter describes the geometrical path interface. In the path interface, the Remote Vehicle Operation provides a Path and the current Pose to the Subject Vehicle and the Subject Vehicle follows that Path using its own controller.

The Subject Vehicle selects whether it uses this path interface or the acceleration/curvature Interface, by sending the message VehicleCapabilities.

7.2.3.1 Definitions

Paths and Poses

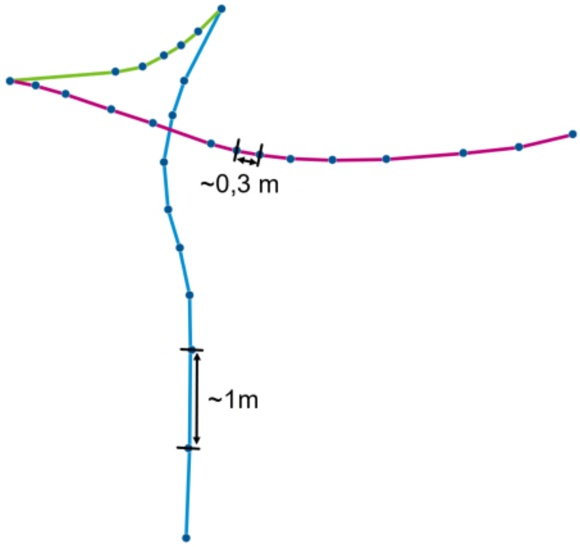


Figure 21 - A typical Path with three segments

- A Path usually consists of multiple segments, each segment consists of a series of PathPoses.
- The driving direction must not change within a segment, but it can, and usually will, change after a segment.
- The distance between two PathPoses is dynamically selected by the Remote Vehicle Operation, but typically ranges from between 0,3 m in curves and up to multiple meters on straight parts
- The Remote Vehicle Operation typically sends parts of one segment as PathSnippet message to the Subject Vehicle.

Pose

Req89: [General] AvpMessagePose

A Pose shall be defined as:

```
struct Pose {
        metre32                       x
        metre32                       y
        radian32                      psi
}
```

A pose shall be serialized by concatenating the serialization of the individual elements.

- psi has the following range: $\psi \in [0, 2\pi)$

PathPose

Req90: [General] AvpMessagePathPose

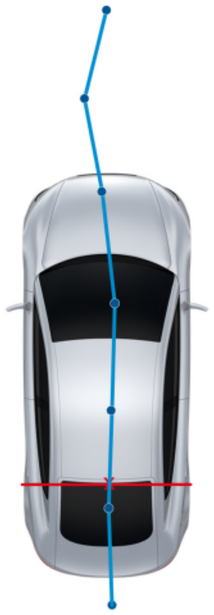
A PathPose shall be defined as

```
struct PathPose {
        metre32                       x
        metre32                       y
        radian32                      psi
        metrePerSecond32              velocity
        perMetre32                    curvature
}
```

- velocity defines the desired maximum Velocity at this point. See Req100: [Vehicle] VehicleStopsAtEndOfPathSnippet. The direction of travel is made clear by a corresponding plus/minus sign.
- curvature defines the desired Curvature at this point, assuming a vehicle model without rear-axle steering.

A PathPose shall be serialized by concatenating the serialization of the individual elements.

7.2.3.2 Initial Path and Path Updates



initial vehicle pose is on path

Figure 22 – Relation between path and Subject Vehicle when receiving a new path

Req91: [Rvo] RvoSendsPath

To ensure smooth driving, the Remote Vehicle Operation shall plan and send paths to the Subject Vehicle in a way that all the following conditions are true:

- the estimated vehicle Pose at the point in time when the new PathSnippet is applied by the Subject Vehicle (see VehicleCapabilities.pathSnippetTakeoverTime) is on or very close to the new PathSnippet. Very close means, that the lateral difference between this estimated vehicle pose and a newly created PathSnippet shall be less than 5 cm and the difference in orientation shall be less than 2°.
- the Velocity in the new PathSnippet at the estimated vehicle Pose (see above) is larger than or equal to the Velocity in the previous PathSnippet at the same Pose.

This means that the Subject Vehicle shall not be required to reduce its velocity uncomfortably due to an updated Path.

Req92: [Vehicle] VehicleReceivesPath

The Subject Vehicle shall receive path snippets in the message PathSnippet from the Remote Vehicle Operation.

- If the Subject Vehicle receives a new PathSnippet, it shall replace any previously received path snippets.
- The Subject Vehicle shall be prepared to receive a PathSnippet messages containing a Path with zero length, in which case the Subject Vehicle shall stop and wait for further instructions (e.g. new PathSnippet or DriveCommand.action = TERMINATE).
- The Subject Vehicle shall apply a newly received PathSnippet without interruption within VehicleCapabilities.pathSnippetTakeoverTime, i.e. the Subject Vehicle shall not stop when it receives a new PathSnippet, but keep driving and start following the new Path.
- If the Subject Vehicle can't process or follow the received PathSnippet, it shall send PATH_NOT_DRIVEABLE (also see Vehicle Error Codes).

ⓘ The frequency in which the Subject Vehicle is able to receive and process new PathSnippet messages as well as the maximum size of those messages is defined individually in the message VehicleCapabilities.

Because of Req91: [Rvo] RvoSendsPath, the Subject Vehicle can switch to a new PathSnippet smoothly, without any internal planning. This is true for both the first Path received by the Subject Vehicle and any updates while driving.

Req93: [Rvo] RvoLimitSizeOfPathSnippet

The Remote Vehicle Operation shall limit the size of a PathSnippet depending on VehicleCapabilities.maximumPathSnippetSize.

The distance between two PathPoses shall be greater or equal than VehicleCapabilities.minimumDistanceBetweenPathPoses and smaller or equal than VehicleCapabilities.maximumDistanceBetweenPathPoses.

If the Remote Vehicle Operation cannot fit all necessary PathPoints into one PathSnippet message, it shall send smaller parts at appropriate points in time.

ⓘ Because a new PathSnippet always replaces any previous PathSnippet messages, the Remote Vehicle Operation needs to wait until the Subject Vehicle passed a significant part of the current PathSnippet, before sending a new one.

Req94: [Rvo] RvoLimitFrequencyOfPathUpdates

The frequency with which the Remote Vehicle Operation can send a new Path to the Subject Vehicle depends on VehicleCapabilities.maximumPathSnippetFrequency.

Depending on this frequency, the Remote Vehicle Operation can perform more or less dynamic maneuvers.

Req95: [Rvo] RvoLimitCurvatureInPath

The Remote Vehicle Operation shall plan the path in a way that the given curvature bounds VehicleCapabilities.maximumDriveableCurvatureForwards and VehicleCapabilities.maximumDriveableCurvatureBackwards are not exceeded when following the path.

ⓘ These limits already contain a small buffer to allow for corrections by the controller within the Subject Vehicle.

7.2.3.3 Following the Path

Req96: [Rvo] RvoSendsCurrentPose

The Remote Vehicle Operation shall send an updated DetectedVehiclePose to the Subject Vehicle at least every 100 ms.

The Pose shall be no older than 500 ms at the point in time when it is passed to the wireless data transmission interface.

The Pose shall have an accuracy of:

- lateral: +/- 5 cm
- longitudinal: +/- 5 cm
- orientation: +/- 2°

Req97: [Vehicle] VehicleReceivesCurrentPose

The Subject Vehicle shall receive and use the vehicle Pose provided by the Remote Vehicle Operation in DetectedVehiclePose for following the Path.

ⓘ For details regarding the coordinate system used in Pose, see Coordinate System.

ⓘ The Remote Vehicle Operation sends an updated vehicle Pose about every 100 ms. Transmission and calculation delays cause the vehicle Pose to be out-of-date when arriving at the Subject Vehicle. Therefore, the Pose contains a timestamp with which the Subject Vehicle can precisely determinate the age of the Pose by comparing it to the Vehicle Functional Clock.

ⓘ Since the last known vehicle Pose received from the Remote Vehicle Operation can be more than 500 ms old, it is recommended that the Subject Vehicle uses this timestamp and its own odometry to extrapolate the driven distance since the Remote Vehicle Operation captured the Pose, in order to improve accuracy.

Req98: [Vehicle] VehicleFollowsPath

Given the predefined Path and the current vehicle Pose, the Subject Vehicle shall be actuated by the in-vehicle systems to drive along the Path.

Driving along the Path means, that the rear axle center is located on the Path and the orientation of the Subject Vehicle is tangential to Path. The Velocity given in each PathPose is a maximum Velocity, that shall be reached whenever possible.

The expected accuracy of the Path following control is:

- lateral: +/- 5 cm
- longitudinal: +/- 5 cm (at the end of a PathSnippet)
- orientation: +/- 2°

ⓘ Accuracy of the Path following control is measured by comparing the desired Path in PathSnippet with the actually driven Path as detected by the Remote Vehicle Operation in DetectedVehiclePose.

Due to the accuracy requirements defined in Req96: [Rvo] RvoSendsCurrentPose and Req98: [Vehicle] VehicleFollowsPath, the actual deviation of the vehicle from the nominal path to be expected in the worst case should be in the range of 10 cm respectively 4 °.

Req99: [Vehicle] VehicleProvidesSpeedControl

The Subject Vehicle shall offer speed control that allows

- drive of slopes up to ± 17 %
- within a speed range from 0.15 m/s and 2.8 m/s
- and a resolution of +/- 0.05 m/s.
- in a smooth way (no optically perceivable jerking)

7.2.3.4 Stopping at the end of a PathSnippet

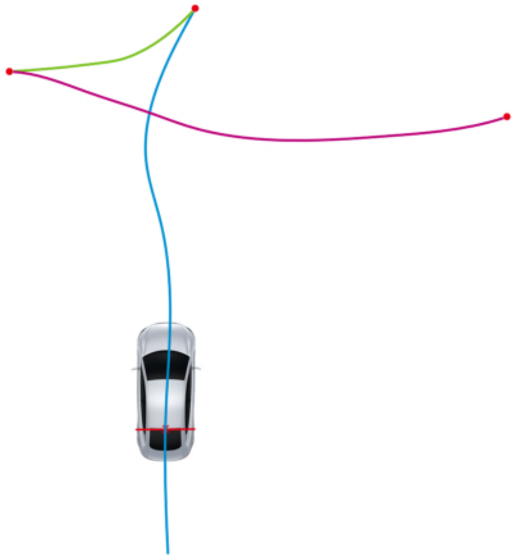


Figure 23 - Points (red) at end of each PathSnippet on which the Subject vehicle shall stop on its own responsibility

Req100: [Vehicle] VehicleStopsAtEndOfPathSnippet

If the Subject Vehicle does not receive a new PathSnippet early enough, it shall stop on its own at the end of a PathSnippet and wait for the Remote Vehicle Operation to send a new PathSnippet message.

The Subject Vehicle can determine its own Pose in relation to the end of the Path faster and more precise than the Remote Vehicle Operation would be able to, by estimating the distance driven since the last Pose received from the Remote Vehicle Operation based on odometry. Therefore, by handing over the responsibility to stop at the end of a PathSnippet to the Subject Vehicle, the overall accuracy is improved.

The Velocity given in each PathPose is a maximum Velocity. The Subject Vehicle shall reduce its Velocity towards the end of the current PathSnippet appropriately in a way that it reaches 0 m/s at the last PathPose of the current PathSnippet, i.e. come to a full stop.

The Velocity given in the last PathPose of a PathSnippet will therefore not equal 0 m/s, because it defines the maximum Velocity with which the Subject Vehicle is allowed to approach the end of that PathSnippet.

7.2.3.5 Temporarily limit velocity

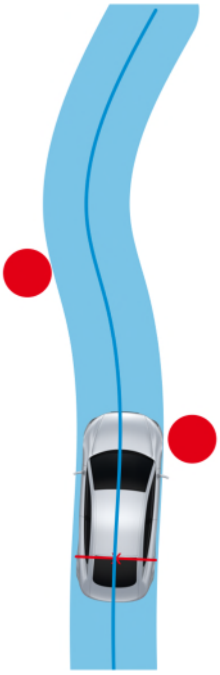


Figure 24 - Temporarily limit velocity

When there are Objects near the desired Path, the Remote Vehicle Operation may decide that it is temporarily necessary to drive slower than anticipated when the Path was originally planned.

In that case, the Remote Vehicle Operation can plan a new Path that contains lower velocities and send this new Path to the Subject Vehicle.

7.2.3.6 Temporarily stop (Comfort Stop)

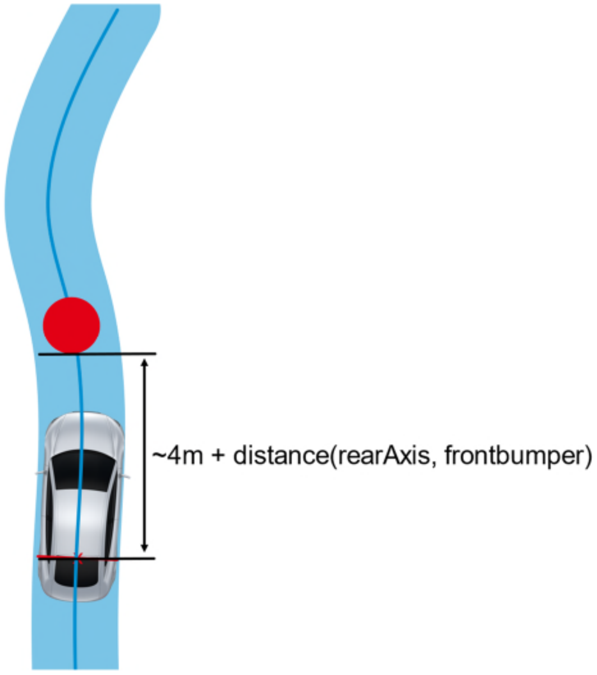


Figure 25 - Temporarily stop (Comfort stop)

When there is an obstacle on the desired Path, the Remote Vehicle Operation has two possibilities to avoid a collision comfortably:

1. Calculate a new Path that bypasses the obstacle and send it to the Subject Vehicle in a PathSnippet message
2. Shorten the current Path so that the current PathSnippet ends with an appropriate distance to the obstacle (~ 4 m distance between obstacle and front bumper or rear bumper, depending on the driving direction).

Regarding chapter “Basic performance requirements”, AVPS shall be capable of providing smooth braking and acceleration and avoid sudden braking or acceleration of the Subject Vehicle under non-emergency/non-hazardous situations.

7.2.4 Control: Acceleration/Curvature Interface (Type 2.2/ 2.3)

This chapter describes the acceleration/curvature interface. In the acceleration/ curvature interface, the Remote Vehicle Operation provides a series of Acceleration and Curvature commands to the Subject Vehicle.

The Subject Vehicle selects whether it uses the path interface or this acceleration/curvature interface, by sending the message VehicleCapabilities.

7.2.4.1 Definitions

The type 2.2 interface is intended to be used with an infrastructure that imposes minimal requirements to the vehicle. The infrastructure shall perform the safety time sync supervision. When the offset is considered to be too large, an Emergency Stop shall be requested. The vehicle will carry this out regardless of its time offset. Also see chapter 7.1.9, “Vehicle Safety Clock Synchronization”. See Figure 36 for the sequence of messages.

Regarding chapter “Basic performance requirements”, AVPS shall be capable of providing smooth braking and acceleration and avoid sudden braking or acceleration of the Subject Vehicle under non-emergency/non-hazardous situations.

VehicleTrajectory

A VehicleTrajectory consists of a vector of ControlTrajectoryElement’s (type 2.2 and 2.3) and StateTrajectoryElement’s (for type 2.3). It also contains a reference time. The reference time indicates the absolute Unix time given in Vehicle Functional Clock at which the first element of the ControlTrajectoryElement vector and StateTrajectoryElement vector is expected to be executed. The elements will have a constant temporal spacing, with the interval defined by vehicleTrajectoryInterval in VehicleCapabilities.

Req101: *[General] AvpMessageVehicleTrajectory*

```
struct VehicleTrajectory {
    millisecond_ui64                          timeReference
    vector ControlTrajectoryElement           controlTrajectory
    vector StateTrajectoryElement      stateTrajectory
}
```

Req102: *[Vehicle] VehicleChoosesTrajectoryInterval*

The discretization interval (VehicleCapabilities.vehicleTrajectoryInterval) and length (VehicleCapabilities.vehicleTrajectoryDuration) shall be defined by the vehicle at identification time. The recommended interval is 40ms. Optionally a *short* (20ms) or *long* (60ms) interval can be selected.

Req103: [Vehicle] VehicleChoosesTrajectoryDuration

VehicleCapabilities.vehicleTrajectoryDuration shall be such that it covers the moving phase for the valid duration of a DrivingPermission message and the subsequent stopping time. Referring to Req11: [Vehicle] VehicleBrakingPerformance and chapter H.3, “Trajectory minimum length”, this means that the Trajectory shall be at least 1.85 s long (47 elements) for a maximum velocity of 10 km/h.

The elements in the StateTrajectoryElement vector are considered as odometry target values. Errors are calculated relative to these values in a cartesian coordinate system based on the reference point of the Subject Vehicle (see Coordinate System). Therefore, the error effectively becomes 0 after a new VehicleTrajectory is received.

ControlTrajectoryElement

Req104: [General] AvpMessageControlTrajectoryElement

```
struct ControlTrajectoryElement {
        perMetre32                          curvature
        meterPerSecondSquared32             acceleration
}
```

StateTrajectoryElement

Req105: [General] AvpMessageStateTrajectoryElement

```
struct StateTrajectoryElement {
        metrePerSecond32                    velocity
        metre32                             vehiclePoseX
        metre32                             vehiclePoseY
        radian32                            vehiclePosePsi
}
```

Control Loops

The control loops for type 2.2 and 2.3 are illustrated in Figure 26. For type 2.2, the main loop leads through the infrastructure, which controls the vehicle by its Curvature and Acceleration trajectories. For this reason, a VehicleTrajectory update rate of 10Hz or higher is recommended.

Type 2.3 also uses the Curvature and Acceleration trajectories as feedforward commands. It features additional low-latency control loops on the vehicle, that track the state trajectory a higher rate.

⚠ Note that the feedback at the infrastructure arrives through two channels: the VehicleState message communicates the Curvature and Velocity from the vehicle's odometry. Additional infrastructure sensing provides absolute position measurements of the Subject Vehicle. The two sources are typically fused in a state estimator.

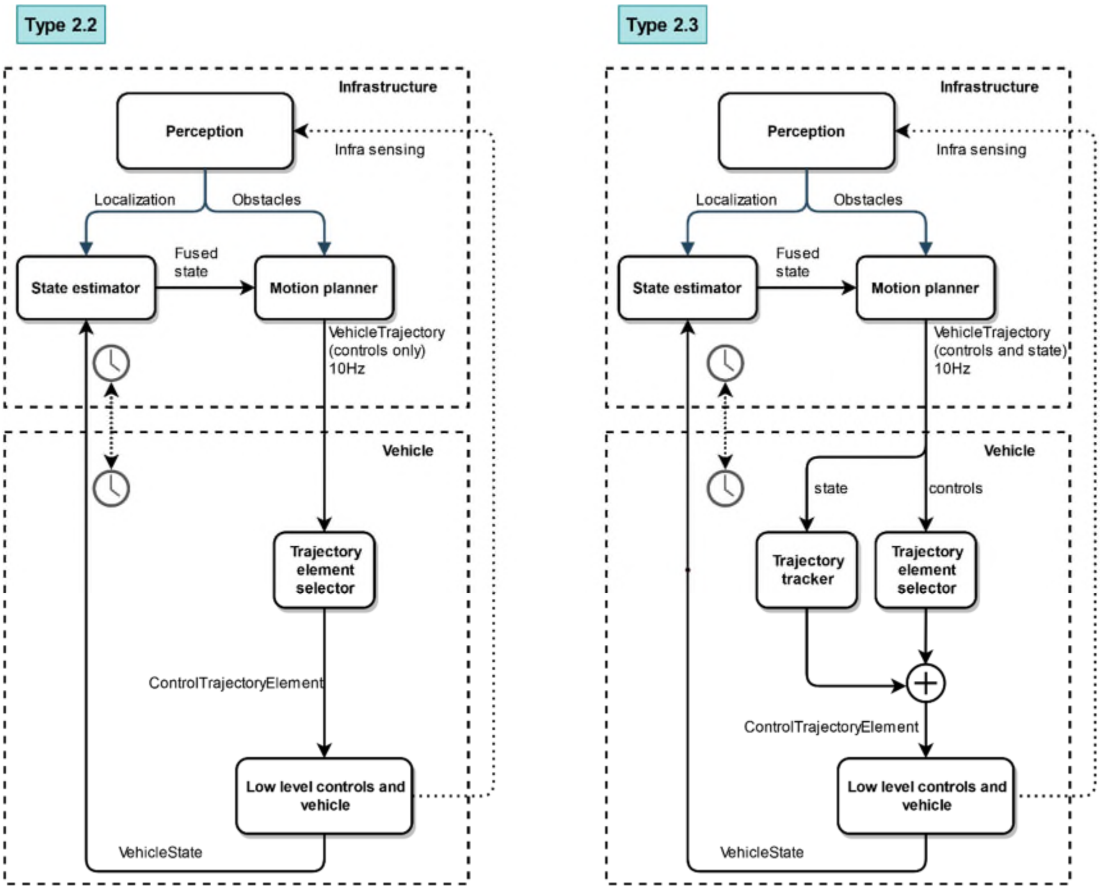


Figure 26 - Control loop type 2.2 and 2.3

Req106: [Vehicle] VehicleProvidesTrajectoryElementSelector

The function of the trajectory element selector block of interface type 2.2 and type 2.3 is illustrated in Figure 27. Using the current time, it selects the correct element from the time-based Trajectory. If the time is between elements, it is suggested to use linear rather than nearest neighbor interpolation. Additionally, this function can be used to compensate for known actuator delays, by selecting trajectory elements from the future.

Note that for this to work, the functional clock of the infrastructure and vehicle should be synced as defined in chapter 7.1.6, “Vehicle Functional Clock Synchronization”.

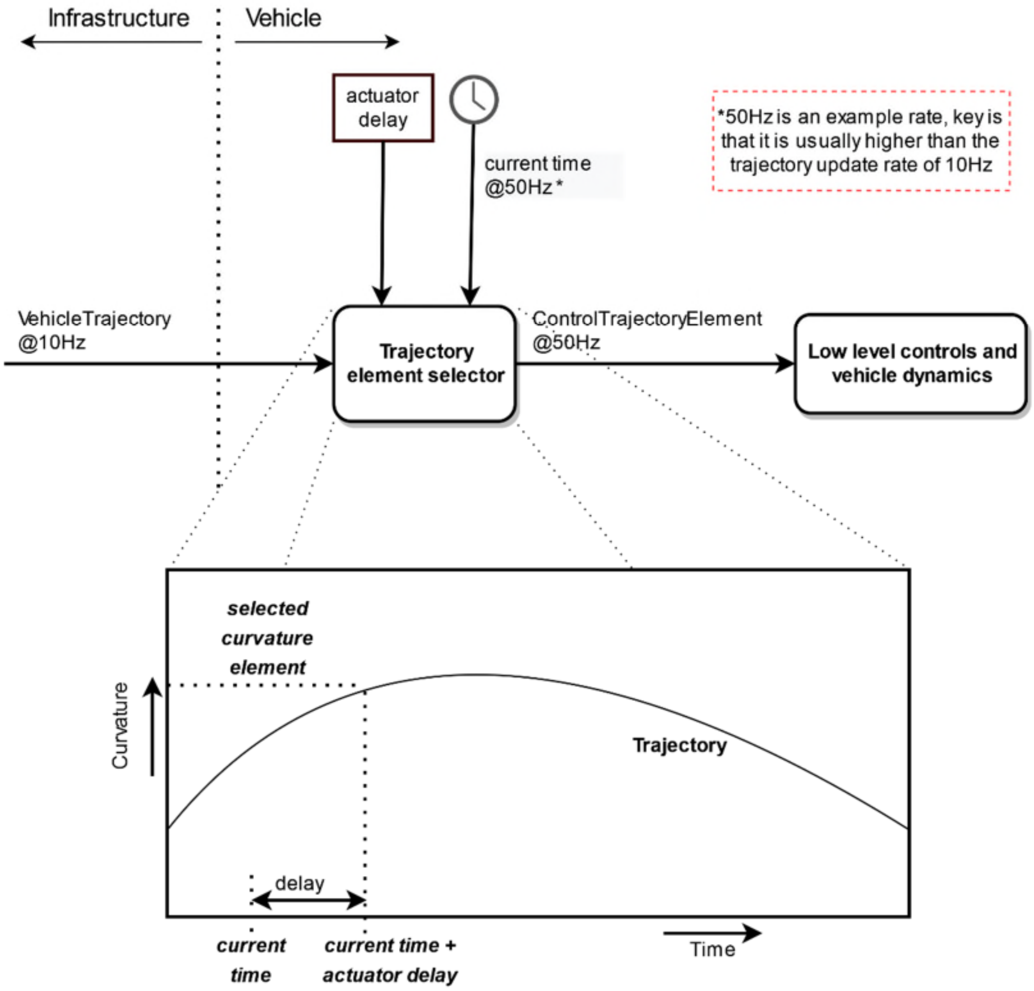

Infrastructure
Vehicle
actuator delay
current time @50Hz*
*50Hz is an example rate, key is that it is usually higher than the trajectory update rate of 10Hz
VehicleTrajectory @10Hz
Trajectory element selector
ControlTrajectoryElement @50Hz
Low level controls and vehicle dynamics
selected curvature element
Curvature
Trajectory
delay
current time
current time + actuator delay
Time

Figure 27 - Trajectory element selector block

7.2.4.2 Type 2.2 interface

For the type 2.2 interface the infrastructure plans the full state Trajectory at a high enough rate to control the vehicle. This means that no onboard Path tracking needs to take place.

> *Req107: [Vehicle] VehicleExecutesTrajectoryCommand*
>
> The vehicle is expected to execute the controls in the VehicleTrajectoryCommand message. This message consists of a list of controls (Curvature, Acceleration) and their timestamps. The timestamps make the system robust against latency jitter.
>
> Additional info about the DrivingDirection is included in the same message.

Req108: [Vehicle] VehicleTransposesRequiredAcceleration

The Acceleration is defined as the time derivative of the magnitude longitudinal Velocity at the rear axle centre. This means it is not the Acceleration as measured by an accelerometer in case of an inclined road. For example, at Standstill on a ramp, the commanded Acceleration will be 0. This also means that the vehicle will itself need to take care of providing the right actuator commands to reach the target Acceleration on inclined roads.

The Remote Vehicle Operation assumes that the vehicle executes the required Acceleration.

Req109: [Rvo] RvoInitiatesTemporaryStop

The motion trajectory planner in the infrastructure has full perception information. When obstacles are present on the desired Path, the trajectory planner has two ways to react depending on the situation:

1. The Trajectory will be planned around the obstacle. The corresponding Acceleration and Curvature commands will be provided with the next VehicleTrajectoryCommand message to the vehicle.
2. Stop comfortably at a safe distance from the obstacle. It will do so by sending a negative Acceleration trajectory until Standstill.

ⓘ Note that because the controls are timestamped, the trajectories can even handle moving obstacles.

7.2.4.3 Type 2.3 interface

In addition to type 2.2 the type 2.3 interface adds StateTrajectoryElement's to the VehicleTrajectoryCommand message, as described in VehicleTrajectory. These include Velocity and Pose. On-vehicle low-latency feedback control can potentially improve the control accuracy.

These feedback loops make use of onboard measurements only. The Pose is defined in local vehicle coordinates, so it is 0 at the beginning of every VehicleTrajectory (also see Coordinate System). Odometry pose shall be reset to 0 at the time corresponding to VehicleTrajectory.timeReference (which will be in the past when the message arrives at the vehicle).

Req110: [Vehicle] VehicleStopsAtEndOfTrajectory

With type 2.3 the Pose trajectory is provided. The vehicle shall recognize when the Trajectory stops in the future. In that case, it shall stop accurately at the stopping Pose.

7.2.5 Safety: Driving Permission

The DrivingPermission is the core of the safety concept described in Figure 20. The DrivingPermission tells the Subject Vehicle the boundaries for driving direction, time, Velocity and Curvature within which it is allowed to move. As long as the Subject Vehicle moves within these boundaries or stops with maximum deceleration (see Emergency Stop) when the bounds are violated, functional safety is ensured.

When handling safety messages like DrivingPermission, the Subject Vehicle always needs to consider and verify the safety checksums (see Req125: [Vehicle] VehicleVerifiesSafetyChecksums).

⚠ During the Emergency Stop the vehicle shall keep following the Curvature commands as defined by the most recent VehicleTrajectoryCommand message, provided that it is within the Curvature bounds of the latest known DrivingPermission message.

Req111: [Vehicle] VehicleEvaluateDrivingPermission

SIL=ASIL-B

The Subject Vehicle shall evaluate its current state using the most recent known valid DrivingPermission at least every $\tau_{SafetyCycleTime} = 20ms$.

The DrivingPermission with the largest expiration time is considered to be the most recent.

After each evaluation, the Subject Vehicle shall:

- set VehicleSafetyFeedback.drivingAllowed to the result of the evaluation
- set VehicleSafetyFeedback.remainingTimeToDrive to $\tau_{RemainingTimeToDrive} = t_{DrivingPermission,expiration} - \tau_{SafetyCycleTime} - \tau_{SafetyToBrakingInitiated} - t_{VehicleSafety,now}$

Also see Req120: [Vehicle] VehicleStopsOnExpirationTimeViolation.

ⓘ See Valid Driving Permission for the definition of "valid DrivingPermission".

Req112: [Rvo] RvoPreventsCollisions

SIL=SIL-2

The Remote Vehicle Operation shall set the values in DrivingPermission in a way that prevents collisions with persons and avoids damage to the Subject Vehicle.

When checking for potential collisions, the Remote Vehicle Operation shall consider the braking distance $d_{braking}(v)$, defined in the respective Vehicle Type Identifier.

When driving in an environment with friction coefficient $\mu < 1.0$, the Remote Vehicle Operation shall take into account the increased braking distances.

ⓘ See Req11: [Vehicle] VehicleBrakingPerformance for the corresponding vehicle requirement.

7.2.5.1 Driving Direction

Each DrivingPermission only allows driving into a specific direction.

Req113: [Vehicle] VehicleStopsOnWrongDrivingDirection[9]

SIL=ASIL-A

The Subject Vehicle shall only allow movement into the direction given in DrivingPermission.drivingDirection.

The Subject Vehicle shall perform an Emergency Stop if it moves into a different direction than DrivingPermission.drivingDirection.

While the Subject Vehicle is not allowed to move because of one of these reasons, it shall add DRIVING_DIRECTION_VIOLATION to VehicleSafetyFeedback.safetyViolations.

⚠ The Subject Vehicle shall be designed in a way that prevents rolling in the unwanted direction when moving off from Standstill.

7.2.5.2 Velocity

Req114: [Vehicle] VehicleStopsOnVelocityViolation[10]

SIL=ASIL-B

The Subject Vehicle shall perform an Emergency Stop if the current Velocity exceeds the given allowed Velocity DrivingPermission.maximumVelocity and add VELOCITY_VIOLATION to VehicleSafetyFeedback.safetyViolations.

The Subject Vehicle shall consider uncertainties when measuring its current velocity and use a worst-case estimation to compare with DrivingPermission.maximumVelocity.

[9] derived from [Vehicle] VehicleEvaluateDrivingPermission
[10] derived from [Vehicle] VehicleEvaluateDrivingPermission

7.2.5.3Curvature

Req115: [Vehicle] VehicleStopsOnCurvatureViolation[11]

SIL=ASIL-B

The DrivingPermission contains a maximum and a minimum allowed Curvature.

The Subject Vehicle shall perform an Emergency Stop if the current Curvature is below the minimum allowed Curvature DrivingPermission.curvatureMin or if the current Curvature is above the maximum allowed Curvature DrivingPermission.curvatureMax.

While the Subject Vehicle is not allowed to move because of one of these reasons, it shall add CURVATURE_MIN_VIOLATION and/or CURVATURE_MAX_VIOLATION to VehicleSafetyFeedback.safetyViolations.

The Subject Vehicle shall consider uncertainties when measuring its current Curvature and use a worst-case estimation to compare with the DrivingPermission.

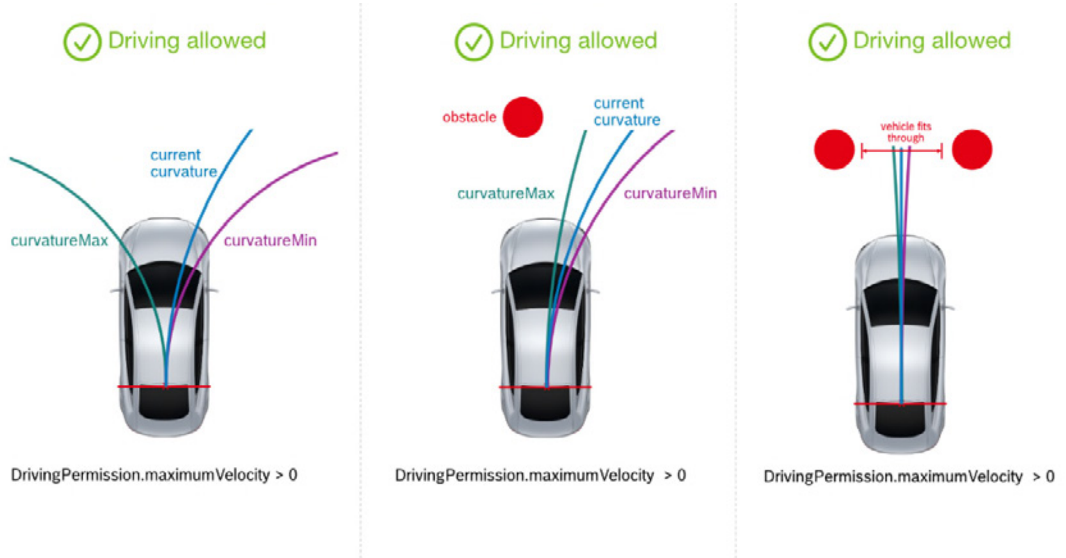


Figure 28 - Driving allowed: current curvature is within bounds of DrivingPermission

[11] derived from [Vehicle] VehicleEvaluateDrivingPermission

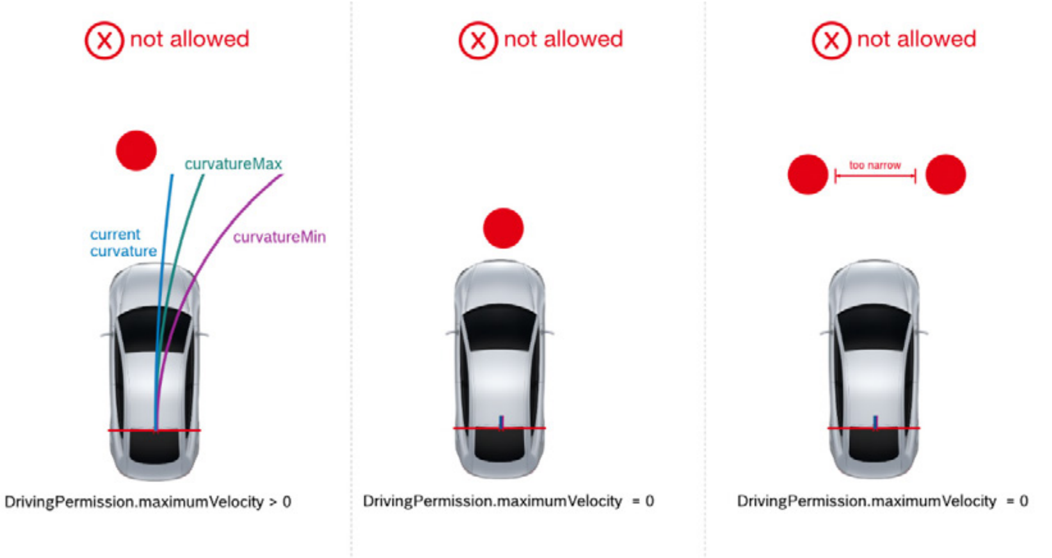

not allowed
curvatureMax
current curvature
curvatureMin
DrivingPermission.maximumVelocity > 0
not allowed
DrivingPermission.maximumVelocity = 0
not allowed
too narrow
DrivingPermission.maximumVelocity = 0

Figure 29 - Safety Violations: (left) current curvature is not within bounds. (right) velocityMaxForwards=0

7.2.5.4 Expiration Time

For evaluating Velocity and Curvature bounds, it is crucial that the Subject Vehicle can verify that the latest received DrivingPermission is recent. Since DrivingPermission are transmitted over an unreliable network, latencies, constant delays, repeatedly arriving messages etc. need to be taken into account.

⚠ The expiration time must not be changed by any instance after it was sent.

All these concerns are addressed by the signal DrivingPermission.expirationTime. The expirationTime is a Unix timestamp, given based on the Vehicle Safety Clock, that defines the point in time up to which the respective DrivingPermission is valid.

The following budgets for processing a DrivingPermission are assigned to the components in the AVP-System:

Table 8 - Safety timing budgets in AVP-System

Instance	Time budget
Remote Vehicle Operation	$\tau_{infrastructure} = 650\ ms$ Sensing the environment and computation of the DrivingPermission is performed in less than 650 ms. Clock uncertainties within the Remote Vehicle Operation as well as uncertainties in the estimation of the Vehicle Safety Clock need to be considered. For details, see Req118: [Rvo] RvoDrivingPermissionTimeBudget
Wireless data transmission	$\tau_{transmission} = 30\ ms$ The transmission of the DrivingPermission to the Subject Vehicle takes less than 30 ms.
Subject Vehicle	$\tau_{Vehicle} = 200\ ms$ Reception and processing of the DrivingPermission in the Subject Vehicle until braking can be initiated, takes less than 200 ms. After that, the configured braking distance $d_{braking}(v)$ (See Req11: [Vehicle] VehicleBrakingPerformance) applies. ⚠ A smaller $\tau_{Vehicle}$ can either improve robustness, help compensate for larger values of $d_{braking}(v)$ or $\pm roveoverallreactiontimeofthesystem.$ For details, see Req119: [Vehicle] VehicleDrivingPermissionTimeBudget

This budget results in the time $\tau_{driving}$ in which the Subject Vehicle can move with a single DrivingPermission:

$$\tau_{driving} = \tau_{reaction} - \tau_{infrastructure} - \tau_{transmission} - \tau_{Vehicle}$$
$$= 1000\text{ms} - 650\text{ms} - 30\text{ms} - 200\text{ms} = 120\text{ms}$$

The following two figures illustrate the timing behaviour for six consecutive DrivingPermission messages. In each case, obstacles enter the scene and require an immediate reaction of the vehicle.

In case of Figure 30 the wireless communication works as intended and therefore the Remote Vehicle Operation can send a DrivingPermission that forbids driving to the Subject Vehicle, i.e. the vehicle stops because of an explicit command.

Because of that, the $\tau_{SafetyTimeSyncUncertainty}$ is not relevant for the reaction time. Since the uncertainty is included in the budget $\tau_{infrastructure}$, the actual reaction time of the infrastructure needs to be derived:

$$\tau_{reaction,infrastructure} = \tau_{infrastructure} - \tau_{SafetyTimeSyncUncertainty}$$
$$= 650ms - 100ms = 550ms$$

Additionally, the time span between two consecutive DrivingPermission messages needs to be considered. In the best case, an obstacle enters the scene right before a measurement is taken. In the worst case, an obstacle enters the scene right after a measurement and therefore will only be considered in the next measurement. From this, the dead time $\tau_{objectAppearance} \in [0, \tau_{DrivingPermissionCycle}] = [0, 100ms]$ follows.

A minimum and a maximum nominal reaction time $\tau_{reaction,nominal}$ of the overall system can be calculated as follows:

$$\tau_{reaction,nominal,min} = \tau_{reaction,infrastructure} + \tau_{transmission} + \tau_{vehicle}$$
$$= 550ms + 30ms + 200ms = 770ms$$

$$\tau_{reaction,nominal,max} = \tau_{eaction,nominal,min} + \tau_{objectAppearance}$$
$$= 550ms + 30ms + 200ms + 100ms = 880ms$$

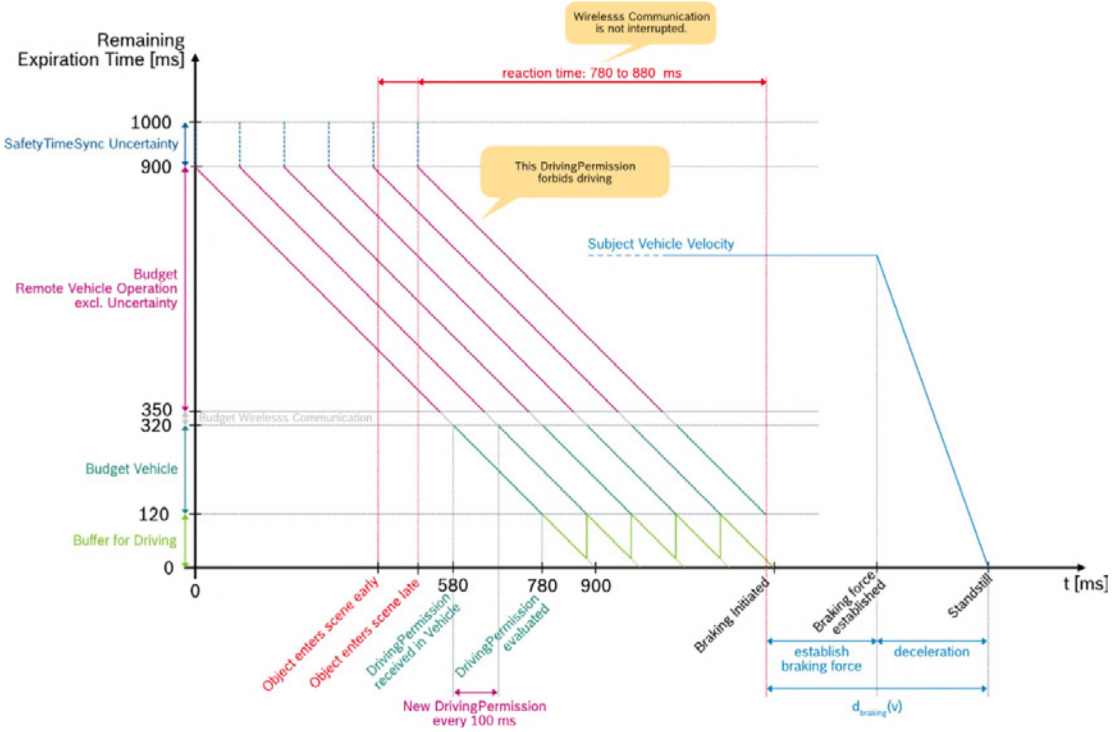

Wirelesss Communication is not interrupted.
reaction time: 780 to 880 ms
Remaining Expiration Time [ms]
1000
SafetyTimeSync Uncertainty
900
This DrivingPermission forbids driving
Subject Vehicle Velocity
Budget Remote Vehicle Operation excl. Uncertainty
350
320
Budget Wirelesss Communication
Budget Vehicle
120
Buffer for Driving
0
0
580
780
900
t [ms]
Object enters scene early
Object enters scene late
DrivingPermission received in Vehicle
DrivingPermission evaluated
New DrivingPermission every 100 ms
Braking Initiated
Braking force established
Standstill
establish braking force
deceleration
d_braking(v)

Figure 30 - Nominal Time Budgets and Reaction Time

In case of Figure 31 the Remote Vehicle Operation cannot send a DrivingPermission that forbids driving to the Subject Vehicle, i.e. the vehicle drives until the last received DrivingPermission expires. In this case, $\tau_{SafetyTimeSyncUncertainty}$ needs to be considered in the reaction time:

$$\tau_{reaction,WiFiInterruption} = \tau_{reaction} \leq 1000ms$$

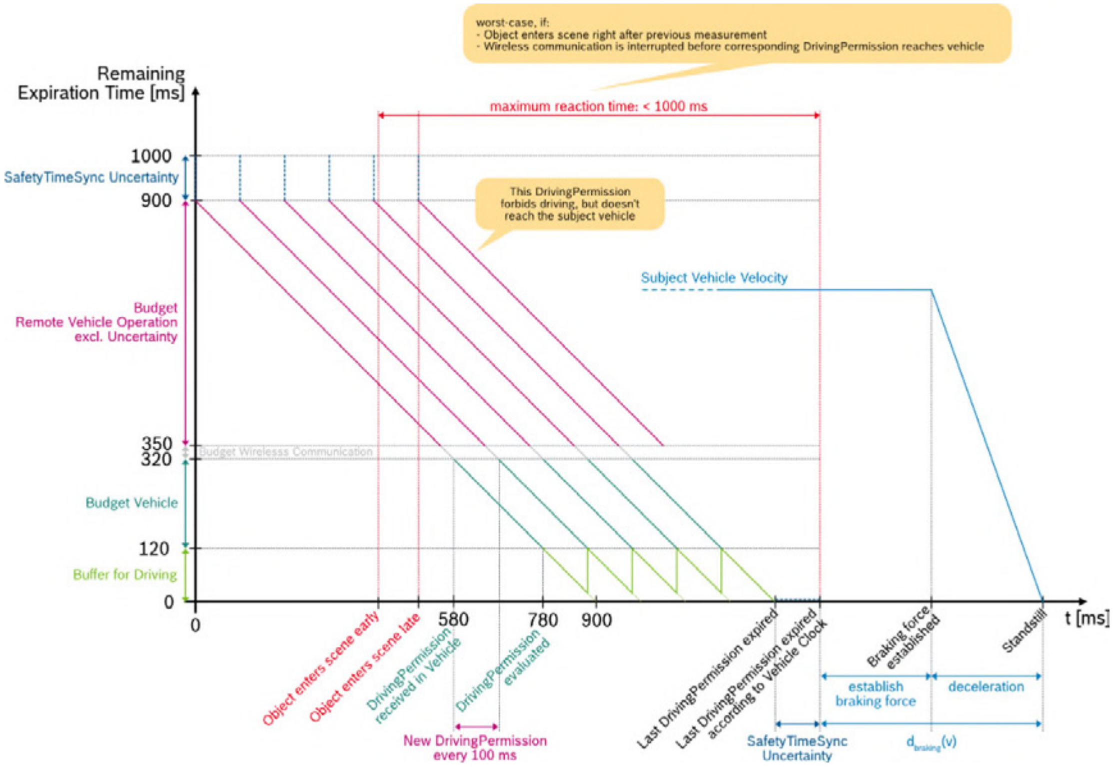


Figure 31 - Worst-Case Time Budgets and Reaction Time if Wi-Fi is interrupted

⚠ In case of an vehicle failure the Subject Vehicle shall abort the mission immidiately (see Req87: [Vehicle] VehicleAbortsMissionOnCriticalVehicleFailure).

Remote Vehicle Operation

For the Remote Vehicle Operation the following requirements are derived:

Req116: [Rvo] RvoStopOnUnknownVehicleSafetyClock[12]

SIL=ASIL-B

The Remote Vehicle Operation shall not send DrivingPermission messages if it can't estimate $t_{VehicleSafety,now}$ according to Req66: [Rvo] RvoDetermineCurrentVehicleSafetyClock-Time.

Req117: [Rvo] RvoDrivingPermissionExpirationTime

SIL=SIL-2

The Remote Vehicle Operation shall compute the expiration time $t_{DrivingPermission,expiration}$ as

$$t_{DrivingPermission,expiration} = t_{VehicleSafety,now} + \tau_{measurement} + \tau_{reaction}$$

$\boldsymbol{t_{VehicleSafety,now}}$

The current Vehicle Safety Clock time. Usually, this point in time is not known exactly, therefore a worst-case estimation considering all uncertainties shall be used:

$$t_{VehicleSafety,now} = t_{RVO,now} + \tau_{ClockOffset,vehicle} - \tau_{safetyTimeSyncUncertainty}$$

See Req66: [Rvo] RvoDetermineCurrentVehicleSafetyClockTime for details.

$\boldsymbol{\tau_{measurement}}$

The time between when the Remote Vehicle Operation started the measurement, i.e. started sensing the environment and now.

$$\tau_{measurement} = t_{RVO,now} - t_{RVO\_measurement}$$

If $t_{RVO\_measurement}$ is not known exactly, a worst-case estimation considering all uncertainties shall be used.

$\boldsymbol{\tau_{reaction}}$

The time between the Remote Vehicle Operation sensing the environment until the Subject Vehicle needs to start braking.

[12] derived from [Rvo] RvoDrivingPermissionExpirationTime

The exact time needs to be negotiated between OEM and Remote Vehicle Operation provider. Though, the reaction time shall be less than 1 s to be equal to or better than the driving capabilities of an experienced and attentive human driver (also see chapter 7.2.1.3, “Requirements for Dynamic Driving Task (DDT)”).

> This is the worst-case reaction time in case an obstacle appears and the communication between Remote Vehicle Operation and Subject Vehicle is interrupted at the same time.

Req118: [Rvo] RvoDrivingPermissionTimeBudget

SIL=SIL-QM

To ensure smooth driving, the processing time $\tau_{infrastructure}$ within Remote Vehicle Operation shall be less than 650 ms, i.e.

$$\tau_{infrastructure} = t_{sent} - t_{measurement} + \tau_{safetyTimeSyncUncertainty} < 650ms$$

$\boldsymbol{t_{sent}}$ The time, when the infrastructure sent the DrivingPermission on the wireless data transmission interface to the Subject Vehicle.

$\boldsymbol{t_{measurement}}$ The time when the infrastructure started the measurement. If this point in time is not known exactly, a worst-case estimation considering all uncertainties shall be used.

$\boldsymbol{\tau_{safetyTimeSyncUncertainty}}$ The current uncertainty in estimating the Vehicle Safety Clock:

$$= \tau_{RoundTripTime} + \tau_{ClockDrift,vehicle}$$

See Req65: [Rvo] RvoDetermineSafetyTimeSyncUncertainty for details.

> According to Req68: [Rvo] SafetyTimeSyncRoundTripTimeBudget:
>
> $\tau_{RoundTripTime} \leq 100ms$!

The Remote Vehicle Operation shall send a new DrivingPermission every 100 ms, i.e.

$$\tau_{DrivingPermissionCycle} = t_{sent,i} - t_{sent,i-1} \leq 100ms$$

⚠ It is crucial that the Remote Vehicle Operation adheres to this budget. Individual exceeding will lead to sudden stops and if the Remote Vehicle Operation exceeds the limits permanently, the Subject Vehicle won't be able to drive at all.

Subject Vehicle

For the Subject Vehicle the following requirements are derived:

Req119: [Vehicle] VehicleDrivingPermissionTimeBudget

SIL=ASIL-QM

To ensure smooth driving, the accumulated processing time $\tau_{Vehicle}$ within the Subject Vehicle shall be less than 200 ms, i.e.

$$\tau_{Vehicle} = \tau_{comToSafety} + \tau_{safetyToBrakingInitiated} < 200ms$$

$\boldsymbol{\tau_{comToSafety}}$ The processing time and signal delays within the Subject Vehicle from when the DrivingPermission is received by the wireless data transmission interface until it reached and evaluated by the safety component.

$\boldsymbol{\tau_{safetyToBrakingInitiated}}$ The time between the safety component evaluating the DrivingPermission until the signal for braking/ deceleration is set. This is the point in time, from which the configured braking distance $d_{braking}(v)$ (See Req11: [Vehicle] VehicleBrakingPerformance) applies.

⚠ It is crucial that the Subject Vehicle adheres to this budget. Individual exceeding will lead to sudden stops and if the Subject Vehicle exceeds the limits permanently, it won't be able to drive at all.

Req120: [Vehicle] VehicleStopsOnExpirationTimeViolation[13]

SIL=ASIL-B

The Subject Vehicle shall perform an Emergency Stop if the most recent DrivingPermission.expirationTime is reached or will be reached before the next evaluation cycle, i.e. the Subject Vehicle shall stop when:

$$t_{VehicleSafety,now} \geq t_{DrivingPermission,expiration} - \tau_{SafetyCycleTime} - \tau_{safetyToBrakingInitiated}$$

$\boldsymbol{t_{VehicleSafety,now}}$ The current Vehicle Safety Clock time

$\boldsymbol{t_{DrivingPermission,expiration}}$ equals DrivingPermission.expirationTime

$\boldsymbol{\tau_{SafetyCycleTime}}$ The cycle time in which the functional safety component evaluates the current Subject Vehicle state. See Req111: [Vehicle] VehicleEvaluateDrivingPermission.

$\boldsymbol{\tau_{safetyToBrakingInitiated}}$ The time between the safety component evaluating the DrivingPermission until the signal for braking/ deceleration is set. This is the point in time, from which the configured braking distance (see Req11: [Vehicle] VehicleBrakingPerformance) applies.

While the Subject Vehicle is not allowed to move because of this reason, it shall add EXPIRATION_TIME_VIOLATION to VehicleSafetyFeedback.safetyViolations.

This means that the Subject Vehicle can only start driving if a DrivingPermission is received with an expiration time far enough in the future.

[13] derived from [Vehicle] VehicleEvaluateDrivingPermission

Req121: [Vehicle] VehicleStopsOnExpirationTimeTooHigh[14]

SIL=ASIL-A

The Subject Vehicle shall perform an Emergency Stop if it receives a DrivingPermission with an expiration time more than 1 s in the future:

$$t_{DrivingPermission,expiration} > t_{VehicleSafety,now} + 1\ s$$

$\boldsymbol{t_{VehicleSafety,now}}$	The current Vehicle Safety Clock time
$\boldsymbol{t_{DrivingPermission,expiration}}$	equals DrivingPermission.expirationTime

The Subject Vehicle shall discard this DrivingPermission.

While the Subject Vehicle is not allowed to move because of this reason, it shall add EXPIRATION_TIME_TOO_HIGH to VehicleSafetyFeedback.safetyViolations.

This is an additional validation mechanism of the safety time synchronisation.

[14] derived from [Vehicle] VehicleEvaluateDrivingPermission

Req122: [Vehicle] VehicleAbortsMissionAfter10s[15]

SIL=ASIL-B

The Subject Vehicle shall abort the Mission (see chapter 7.3.1, “Abort Mission”) when the most recent DrivingPermission expired more than 10 seconds ago:

$$t_{VehicleSafety,now} > t_{DrivingPermission,expiration} + 10\ s$$

$\boldsymbol{t_{VehicleSafety,now}}$	The current Vehicle Safety Clock time
$\boldsymbol{t_{DrivingPermission,expiration}}$	equals DrivingPermission.expirationTime

If the Subject Vehicle aborts the Mission because of this reason, it shall add LAST_DRIVING_PERMISSION_TOO_OLD to VehicleSafetyFeedback.safetyViolations.

This is required to ensure with ASIL-B that under no circumstances the Subject Vehicle leaves the parking facility in Safe Driving State.

This condition can only be evaluated if the Subject Vehicle received at least one DrivingPermission.

[15] derived from [Vehicle] VehicleEvaluateDrivingPermission

7.3 Wrap Up Mission and Shutdown Subject Vehicle

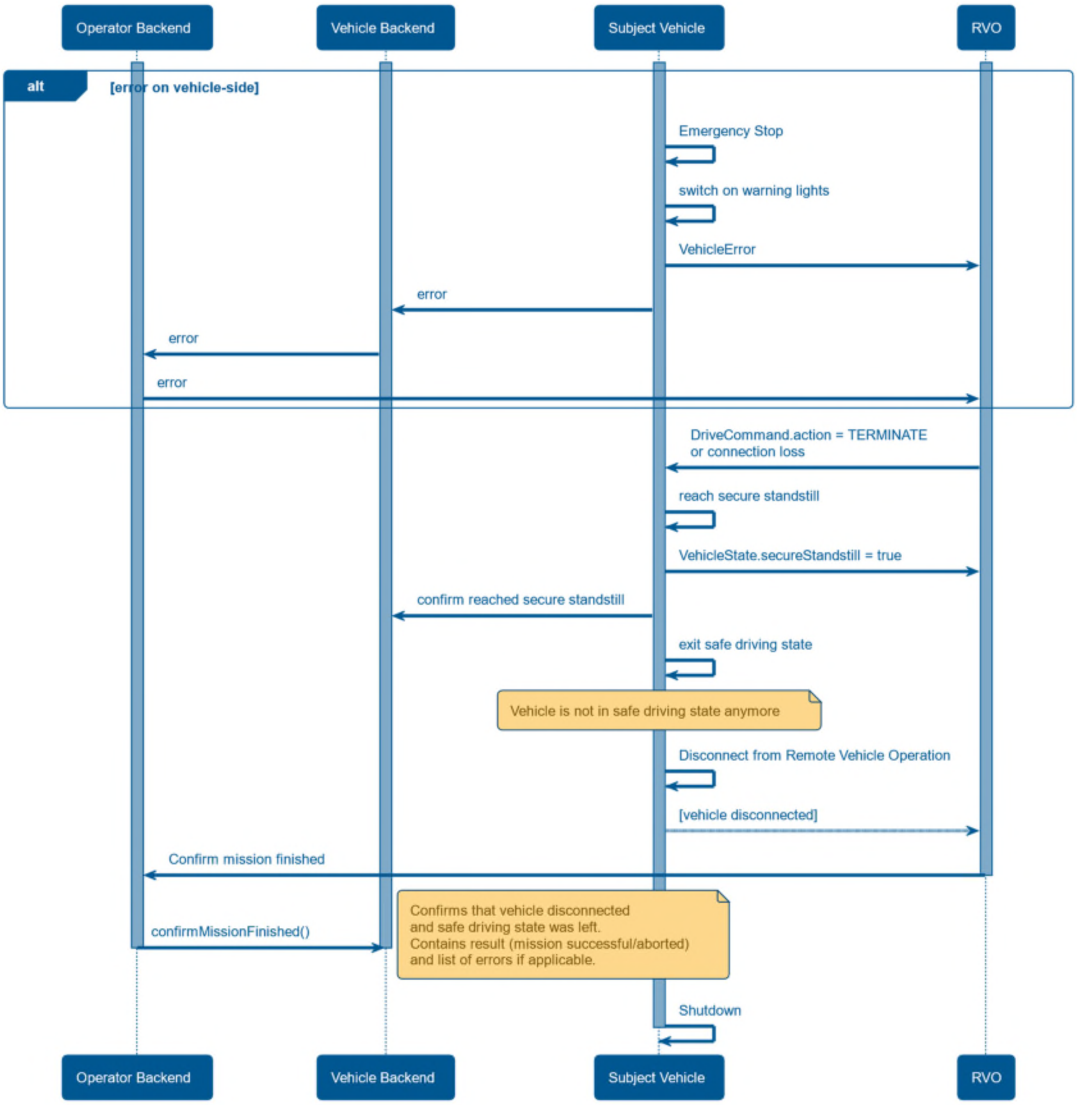


Figure 32 – Mission WrapUp Sequence

Req123: [Vehicle] VehicleSecuresOnFinishedMission

If any of the following conditions are true:

- The Subject Vehicle receives DriveCommand.action=TERMINATE from the Remote Vehicle Operation
- The Subject Vehicle aborted the Mission because it lost connection to the Remote Vehicle Operation

the Subject Vehicle shall:

1) ensure that it reaches Secure Standstill
2) after it reached Secure Standstill
    - send the message VehicleState .secureStandstill = TRUE to the Remote Vehicle Operation (if possible)
    - confirm Secure Standstill to the Vehicle Backend (if possible)
3) exits Safe Driving State
4) disconnect TLS, DTLS and Wi-Fi
5) performs all necessary tasks to shut down the Subject Vehicle permanently (e.g. disable actuator interfaces, turn of ignition, disable powertrain…)

In the event of an error, the higher-level system management must ensure that either a new mission is started to drive the vehicle on to the next parking space, or other measures are taken (e.g. tow truck).

7.3.1 Abort Mission

Req124: [Vehicle] VehicleMissionAbort

SIL=ASIL-B

Aborting a Mission means that the Subject Vehicle does all the following:

1. perform an Emergency Stop (ASIL-A)
2. switch on the warning lights (they remain on even when the engine is off) (QM)
3. report all errors to both the Vehicle Backend and the Remote Vehicle Operation using the message VehicleError (QM)
4. proceed with a regular wrap up as defined in Req123: [Vehicle] VehicleSecuresOnFinishedMission, i.e. wait for Remote Vehicle Operation to send DriveCommand.action=TERMINATE or continue directly with the defined actions if the Mission was aborted because the connection to the Remote Vehicle Operation was lost.

7.4 Check-out

ⓘ For further information about the retrieval request, handback sequence and check-out sequence which also includes a check-out report to the User, see [1], chapter A.5.2, A.2.4 and A.2.2.

7.5 Safety Checksums

Messages relevant for safety are:

- DrivingPermission
- SafetyTimeSyncRequest
- SafetyTimeSyncResponse
- SafeVehicleTypeConfirmation

These messages contain an additional checksum to ensure that the payload was not modified accidentally and was generated for that specific Subject Vehicle.

⚠ Since the vehicle identification code needs to be known for calculating the checksum, safety relevant payloads cannot be exchanged unless the vehicle identification process was successful.

Req125: [Vehicle] VehicleVerifiesSafetyChecksums

SIL=ASIL-B

The Subject Vehicle shall verify the checksum for all incoming safety-related messages and abort the Mission (See chapter 7.3.1, “Abort Mission”) if a checksum is not correct.

If the checksum of DrivingPermission is not correct, the Subject Vehicle shall add CRC_VIOLATION_DRIVING_PERMISSION to VehicleSafetyFeedback.safetyViolations.

If the checksum of SafetyTimeSyncRequest is not correct, the Subject Vehicle shall add CRC_VIOLATION_CLOCK_SYNC_RESPONSE to VehicleSafetyFeedback.safetyViolations.

Req126: [Rvo] RvoVerifiesSafetyChecksums

SIL=ASIL-B

The Remote Vehicle Operation shall verify the checksum for all incoming safety-related messages and abort the Mission (See chapter 7.3.1, “Abort Mission”) if a checksum is not correct.

Req127: [General] CalculateSafetyChecksum

SIL=ASIL-B

The following algorithm shall be used to calculate the checksum for the messages SafetyTimeSyncRequest, SafetyTimeSyncResponse and SafeVehicleTypeConfirmation:

checksum = CRC32K9 (serializedData + (seed XOR TransformationConstant))

Algorithm in Detail:

1. Serialize the message payload as specified in chapter 8.2, “AVP-Message Data Protocol”, by concatenating the byte-representation of all signals (except the checksum signal).
2. Append the byte-representation of the transformed vehicle-id seed, which is calculated as

   transformedVehicleIdSeed = seed XOR TransformationConstant
3. Calculate the CRC-32K/9 checksum over the data generated in step 1. and 2.

ⓘ seed refers to VidRequest.seed

ⓘ The whole 64 bit of the transformation constant TransformationConstant shall only be known to components that are compliant to ASIL-A. (See Table 4 in chapter 2.3, “Interface-Specification Version”)

⚠ Whenever any (A)SIL rated requirement or message changes between two released versions of the Interface Specification, the transformation constant which is applied to the vehicle identification seed for calculating safety checksums, will be changed (also see Table 4 in chapter 2.3, “Interface-Specification Version”).

Req128: [General] CalculateSafetyChecksumDrivingPermission

SIL=ASIL-B

The following algorithm shall be used to calculate the checksum for the message DrivingPermission:

```
checksumDrivingPermission = CRC32K9(serializedData + (seed XOR TransformationConstant))
XOR AdditionalSafetyTransformationConstant
```

Algorithm in Detail:

This algorithm equals the checksum calculation described in Req127: [General] CalculateSafetyChecksum with an additional XOR transformation applied to the result.

ⓘ seed refers to VidRequest.seed

⚠ Whenever any (A)SIL rated requirement or message changes between two released versions of the Interface Specification, the transformation constant which is applied to the vehicle identification seed for calculating safety checksums, will be changed (also see Table 4 in chapter 2.3, “Interface-Specification Version”).

8 Infrastructure - Vehicle – Connection

This chapter defines requirements to some layers of the OSI model and also explains the used AVP-message data protocol with a focus on the message header and used types.

8.1 Overview

8.1.1 Link and Network layer

AVP works in the current version with wireless IP based networks. The usage of C-V2X is possible in the future.

Req129: *[Vehicle] VehicleExternalCommunication*

For communicating with the Remote Vehicle Operation, the Subject Vehicle shall be able to connect to either an external Wi-Fi infrastructure or a cellular network.

8.1.1.1 Wi-Fi

The currently preferred radio technology is IEEE 802.11 n using the 2.4 GHz band, because it is widely available and has a better attenuation characteristic than Wi-Fi in the 5 GHz band. Also refer to Req6: [Vehicle] VehicleSupportsFastWiFiRoaming.

Req130: *[Rvo] RvoPreventsCommunicationBetweenVehicles*

The Wi-Fi network shall not allow communication between vehicles.

8.1.1.2 Cellular Network

The currently preferred technology is 5G, because of low latencies.

The Subject Vehicle receives the necessary connection information (e.g. Wi-Fi SSID, Remote Vehicle Operation IP address, Remote Vehicle Operation certificates) for each Mission from the Vehicle Backend. The chapters 7.1.3, “Certificate and Connection Parameter Exchange ” and 7.1.4, “Wi-Fi Connection” cover the authentication process.

8.1.2 Transport to Presentation layer

This AVP interface uses both a mutually authenticated TLS (encrypted TCP/IP) and DTLS (encrypted UDP/IP) point-to-point connection for exchanging data between the Subject Vehicle and the Remote Vehicle Operation:

- One-off messages, for which delivery shall be guaranteed by the protocol, use the TLS channel.
- Messages which are either sent cyclically, for which only the most recent message of a type is relevant, and for which dropped packets are preferable to retransmission, the DTLS channel is used.

Using an encrypted and authenticated point-to-point connection ensures that each Subject Vehicle receives only relevant messages from a trusted Remote Vehicle Operation.

The chapter 7.1.5, “TLS/DTLS Connection” covers the process to establish and authenticate both a TLS and DTLS connection.

8.2 AVP-Message Data Protocol

The AVP-Message data protocol describes how to de-/serialize data exchanged between Remote Vehicle Operation and Subject Vehicle. Furthermore, it defines the byte order and how to handle the packing of messages.

Req131: *[General] AvpMessageProtocol*

Subject Vehicle and Remote Vehicle Operation shall use the AVP-Message data protocol for communicating. The generic format is described in this chapter, the content of the individual messages is described in chapter C, “AVP Messages”.

Req132: [General] AvpMessageSerialization

- All data exchanged between Subject Vehicle and Remote Vehicle Operation shall be serialized using the AVP-Message data protocol.
- Each AVP-Message shall consist of a header with fixed size, followed by a payload with variable size.
- A single value or Object within an AVP-Message is called signal.
- An AVP-Message shall be serialized by concatenating the byte-representations of each signal one after the other. Complex signals like vectors or Poses may use a special serialization algorithm which are also described in this chapter.

8.2.1 Header

Req133: [General] AvpMessageHeader

The header shall be defined as

```
struct Header {
        uint32                      typeFingerprint
        float64                     timeSent in [s]
        uint16                      payloadLength in [byte]
}
```

- The size of the header is exactly 14 bytes.
- typeFingerprint indicates the message type as defined in chapter C, “AVP Messages”
- timeSent is a Unix timestamp with sub-second precision indicating the point in time when the message was created or modified. The time is given either in Vehicle Functional Clock or Vehicle Safety Clock. The appropriate clock is specified individually for each message.
- payloadLength shall contain the number of bytes of the serialized payload that will follow the header.

8.2.2 Fundamental Types

Req134: [General] AvpMessageFundamentalTypes

Fundamental types shall be serialized using their binary representation.

Table 9 - Fundamental Types

AVP Type	C++ Type	Size in Bytes	Unit
bool	bool	1	-
uint8	uint8	1	-
uint16	uint16	2	-
uint32	uint32	4	-
uint64	uint64	8	-
int32	int32	4	-
second64	float64	8	[s]
metrePerSecond32	float32	4	[m/s]
perMetre32	float32	4	[1/m]
metre32	float32	4	[m]
radian32	float32	4	[rad]
millimetre_ui16	uint16	2	[mm]
millimetre_u32	uint32	4	[mm]
millisecond_ui16	uint16	2	[ms]
millisecond_i16	int16	2	[ms]
millisecond_i32	int32	4	[ms]
millisecond_i64	int64	8	[ms]
millisecond_ui64	uint64	8	[ms]
millimetrePerSecond_ui16	uint16	2	[mm/s]
perKilometre_i16	int16	2	[1/km]

Req135: [General] AvpMessageByteOrder

- Integer types shall be encoded using little endian byte order.
- Floating-point types shall be encoded according to IEEE 754 with little endian byte order.

8.2.3 Complex Types

8.2.3.1 String

Req136: [General] AvpMessageString

A string shall be serialized by concatenating the length of the string in bytes represented as uint16 followed by the string's ASCII encoded characters. Each ASCII character shall have a size of 1 byte.

```
String {
        uint16                    length n in [byte]
        uint8[n]                  ASCII characters
}
```

8.2.3.2 Buffer

Req137: [General] AvpMessageBuffer

A buffer represents arbitrary binary data. A buffer shall be serialized by concatenating the size of the buffer in bytes represented as uint16 followed by the buffer's binary data.

```
Buffer {
        uint16                    size n in [byte]
        uint8[n]                  bytes
}
```

8.2.3.3 Vector

Req138: [General] AvpMessageVector

A vector is a list of elements of a well-defined data type. A vector is serialized by concatenating the number of elements in the vector represented as uint16 followed by the serialization of the individual vector elements.

```
Vector<ElementType> {
        uint16                          size n in [byte]
        ElementType[n]                  Concatenated serializations of ElementType
}
```

A Vehicle Type Identifier

A Remote Vehicle Operation supports a fleet of AVP-enabled vehicles that differ in physical properties such as size, turning radius, braking performance etc. as well as capabilities like the frequency in which they can process certain messages.

For some of these properties, it is important that the Remote Vehicle Operation knows them, so that it can:

- decide whether the car physically fits into the parking facility (drivability)
- lay the safety zone as close as possible around the car (separability)
- adapt the behavior to the Subject Vehicle's capabilities

The vehicle type identifier (SIL=ASIL-B) shall be a string of at most 32 ASCII characters that is unique across the AVP ecosystem. The string shall start with the *world manufacturer identifier*, followed by a dash, followed by an arbitrary sequence of ASCII characters defined by the respective manufacturer, e.g. "XYZ-MODELA-VARIANTB-VERSIONC"

> ⓘ Theoretically, the arbitrary sequence of ASCII characters can be defined by the OEM in any way. Nevertheless, a systematic way like described before is recommendable.

The following set of parameters defines a unique vehicle type identifier. If any of these parameters change between different vehicle types or variants of a vehicle type, a new identifier shall be issued.

The parameters in Table 10 are safety critical.

> ⚠ Parameters and capabilities that are not relevant for safety are either sent directly from the Subject Vehicle to the Remote Vehicle Operation in the message VehicleCapabilities or are exchanged between Vehicle Backend and Operator Backend as part of the check-in process as specified in ISO 23374.

Table 10 - Set of parameters which defines a unique vehicle type identifier

Description	Unit	Tolerance
Vehicle height	[m]	+/- 0.01 m
Vehicle width, without mirrors	[m]	+/- 0.01 m
Wheelbase	[m]	+/- 0.01 m
Front excess	[m]	+/- 0.01 m
Rear excess	[m]	+/- 0.01 m
Shape of front bumper **List of points forming a closed polygon, coordinates given in [m]**	[m]	+/- 0.01 m
Shape of rear bumper **List of points forming a closed polygon, coordinates given in [m]**	[m]	+/- 0.01 m
Maximum velocity forwards in safe driving state	[m/s]	+/- 0.1 m/s
Maximum velocity backwards in safe driving state	[m/s]	+/- 0.1 m/s
Braking performance (Req11: [Vehicle] VehicleBrakingPerformance)	Table [m/s → m]	+/- 0.02 m
Expected Clock Drift of Vehicle Safety Clock e.g. 10 % equals 100 ms per second	[%]	+/- 1 %

B Vehicle Error Codes

The Subject Vehicle sends the message VehicleError to the Remote Vehicle Operation whenever an error occurred that prevents the Subject Vehicle from executing the given commands.

Depending on the given error, the Remote Vehicle Operation either tries to resolve the issue or aborts the Mission, saves this message for logging and forwards its content to the backend.

In addition to that, the Subject Vehicle can use the message VehicleInfo at any time, to communicate information that has no influence on the current Mission to the Remote Vehicle Operation. The information will be forwarded to the Vehicle Backend.

Table 11 specifies possible values for VehicleError.code.

The codes refer to the suspend condition codes of Layer 1 category A: Subject Vehicle sub-system fault as defined in ISO 23374 [1].

Table 11 - Vehicle Error Codes to the reaction of AVP-Infrastructure

Code ISO Layer 2	Description	Remote Vehicle Operation Reaction
00	**UNSPECIFIED** Any kind of error that is not specified otherwise	Remote Vehicle Operation aborts the Mission.
01	**Low fuel/battery**	Remote Vehicle Operation drives Subject Vehicle to the closest parking spot
02	**Internal overriding activity** E.g. steering wheel or gear level was touched or operated	Remote Vehicle Operation aborts the Mission.
03	**External human activity** E.g. door handle or trunk used	Remote Vehicle Operation aborts the Mission.
04	**Uneven road surface** *not applicable for AVP Type 2*	
05	**Perception disturbance** *not applicable for AVP Type 2*	
06	**Marker detection unsuccessful** *not applicable for AVP Type 2*	
07	**No signal from Remote Vehicle Operation** Wi-Fi, TLS or DTLS disconnect or no data received.	Remote Vehicle Operation aborts the Mission if the connection can't be re-established within a certain amount of time. See Req49: [Vehicle] VehicleReconnectConditions
08	**No signal from Vehicle Backend**	Remote Vehicle Operation continues the Mission until the Subject Vehicle arrived at the destination.

09	**Mission time-out**	Remote Vehicle Operation aborts the Mission.
10	**Invalid destination** *not applicable for AVP Type 2*	
11	**Invalid route** The Subject Vehicle can’t follow the given PathSnippet, based on the given DetectedVehiclePose.	Remote Vehicle Operation tries to plan a different path or aborts the Mission otherwise.
12	**Collision detected**	Remote Vehicle Operation aborts the Mission.
13	**Unavoidable obstacle in pathway** *not applicable for AVP Type 2*	
14	**Unsuitable road surface** *not applicable for AVP Type 2*	
15	**Invalid map data** *not applicable for AVP Type 2*	
16	**User confirmation time-out**	
17	**Retrieval time-out** *not applicable for AVP Type 2*	Remote Vehicle Operation aborts the Mission.
18	**User intervention** E.g. remote key operation	Remote Vehicle Operation aborts the Mission.

C AVP Messages

The following chapter describes the messages exchanged between the Remote Vehicle Operation and Subject Vehicle. In addition, the need for each of the three Type 2 modes is highlighted by three symbols. Different colors mean different dependencies. The symbols used are the followings:

○	No function
🟢	Optional (Not mandatory)
🟡	Required for general function
🔴	Required for safety

⚠ As mentioned in chapter 1, „Scope“ this document’s messages base on the linked documents. Therefore, details such as the fingerprint may differ between this document and various published versions of the basic documents.

C.1 AccessPointChangeRequest

With this message, the Remote Vehicle Operation can ask the Subject Vehicle to connect to a specific access point to optimize the roaming behavior and trigger a switch to another access point at a proper location and point in time.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x2825B52E
Size	2 to 14 bytes
Interface	TLS, on demand
Required	for vehicles with Wi-Fi connection, for AVP-Infrastructures with more than one access point when needed
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
bssid	string	2 to 14 bytes	The BSSID of the access point to which the Subject Vehicle shall connect. Given in lower case without delimiters, e.g. "00e201b434a2". Or "ANY" if the Subject Vehicle shall roam freely.

C.2 DetectedVehiclePose

This message contains the most recent Pose of the Subject Vehicle in the fixed parking facility coordinate system (see Co-ordinate System).

Message necessity

●	○	○
2.1	2.2	2.3

In the ISO standard [1] the Pose is part of the DrivingCommand.

Fingerprint	0x85D19CCD
Size	20 bytes
Interface	DTLS, 100 ms cycle time
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
x	metre32	4 bytes	Current Subject Vehicle Pose estimated by the Remote Vehicle Operation. x coordinate in parking facility coordinate system (see Coordinate System).
y	metre32	4 bytes	Current Subject Vehicle Pose estimated by the Remote Vehicle Operation. y coordinate in parking facility coordinate system (see Coordinate System).
psi	radian32	4 bytes	Current Subject Vehicle Pose estimated by the Remote Vehicle Operation. Psi in parking facility coordinate system (see Coordinate System).
measurementTime	second64	8 bytes	Measurement time of given Subject Vehicle Pose. Given based on Vehicle Functional Clock.

C.3 DriveCommand

This message contains most of the required information for a Mission.

Message necessity

●	●	●
2.1	2.2	2.3

In [1] the DriveCommand includes more details and summarizes multiple messages in one message. The separation ensures that the DriveCommand can be used for Type 2.1, 2.2 and 2.3.

Fingerprint	0x024FC135
Size	3 bytes
Interface	TLS, when content changes
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
action	enum DriveCommandAction	1 byte	Current driving command
terminateReason	enum TerminateReason	1 byte	If action = TERMINATE, indicates whether a TERMINATE is requested because the Subject Vehicle reached the destination or because of an error.
directionIndicator	enum DirectionIndicator	1 byte	The currently requested direction indicator

C.4 DrivingPermission

The driving permission defines the bounds within which the Subject Vehicle is allowed to drive.

- The Remote Vehicle Operation will not send DrivingPermissions unless vehicle identification was successful.
- The Subject Vehicle shall be in Secure Standstill until it receives the first DrivingPermission that allows driving
- The Subject Vehicle shall abort the Mission if the latest known expirationTime is older than 10 s

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0xFF4FADE9
Size	19 bytes
Interface	DTLS, 100 ms cycle time
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Safety Clock

Name	Type	Size	Description
expirationTime	millisecond_ui64	8 bytes	Time given based on Vehicle Safety Clock.
drivingDirection	enum DrivingDirection	1 byte	Allowed driving direction
maximumVelocity	millimetrePerSecond_ui16	2 bytes	Maximum allowed Subject Vehicle Velocity when driving in the given direction. Given without sign.
curvatureMin	perKilometre_i16	2 bytes	Right Curvature bounds when driving in the given direction
curvatureMax	perKilometre_i16	2 bytes	Left Curvature bounds when driving in the given direction
checksum	uint32	4 bytes	See Req128: [General] CalculateSafetyChecksumDrivingPermission

C.5 DtlsInterfaceRequest

Low level message in communication API. Message can be sent through a TLS connection to a DualInterface server to request a DTLS connection. The server will reply with DtlsInterfaceResponse.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0xF8FC844D
Size	3 bytes
Interface	TLS
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
state	enum DtlsInterfaceRequestState	1 byte	Set to START to request a new DTLS connection
portClient	uint16	2 bytes	Set to arbitrary value if client wants to choose the port. Otherwise set to 0 then the server will assign a port.

C.6 DtlsInterfaceResponse

Low level message in communication API. Response to a DtlsInterfaceRequest.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x3E29DFCD
Size	5 bytes
Interface	TLS
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
state	enum DtlsInterfaceResponseState	1 byte	Result of the interface request
portClient	uint16	2 bytes	Port to be used by client for DTLS connection
portServer	uint16	2 bytes	Port to be used by server for the DTLS connection

C.7 FunctionalTimeSyncRequest

This message can be used for a precise time synchronization though it cannot be used for safety-related functions.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x312F0A8A
Size	2 bytes
Interface	DTLS
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
challenge	uint16	2 bytes	Challenge chosen by the Remote Vehicle Operation. The Remote Vehicle Operation shall use this challenge to relate the response to this request.

C.8 FunctionalTimeSyncResponse

The message contains the response to the non-safe FunctionalTimeSyncRequest message.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x85D36187
Size	10 bytes
Interface	DTLS
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
challenge	uint16	2 bytes	Challenge received in the FunctionalTimeSyncRequest message.
currentVehicleFunctionalClockTime	second64	8 bytes	Time of the Vehicle Functional Clock when FunctionalTimeSyncRequest was received.

C.9 Heartbeat

Low level message in communication API to keep connection alive.
The Heartbeat message shall be sent once per second through all interfaces.
If no Heartbeat is received for 5 seconds, the connection shall be closed.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x59C599ED
Size	1 byte
Interface	TLS + DTLS
Required	yes
Sender	Both Remote Vehicle Operation + Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
alive	bool	1 byte	Always true

C.10InterfaceSpecificationVersion

With this message, both Subject Vehicle and Remote Vehicle Operation confirm the version of the interface specification which the backends agreed on, to each other.

Message necessity

● 2.1 ● 2.2 ● 2.3

Fingerprint	0x4DAC88AD
Size	4 to 36 bytes
Interface	TLS
Required	yes
Sender	Both Remote Vehicle Operation + Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
version	string	4 to 36 bytes	Version of the agreed Infrastructure-Vehicle-Interface Specification

C.11 MissionConfirmation

The message is sent to the Subject Vehicle when a Mission starts, which is usually right after the Subject Vehicle connected to the Remote Vehicle Operation.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x474ED63E
Size	71 to 103 bytes
Interface	TLS
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
parkingFacilityIdentifier	string	2 to 34 bytes	Identifier of the parking facility
sessionId	string	2 to 34 bytes	Unique identifier for management of one Subject Vehicle from the time of check-in until check-out
missionId	string	2 to 34 bytes	Unique Mission identifier created by Operator Backend
recordingLevel	enum RecordingLevel	1 byte	Amount of requested recordings (See Req82: [Vehicle] VehicleDataLogging)

C.12 PathSnippet

This message contains the current snippet which the Subject Vehicle is supposed to follow.

The maximum size and frequency depend on the settings in VehicleCapabilities.

The Remote Vehicle Operation ensures that there is sufficient overlap between consecutive PathSnippet and that the driving direction does not change within a PathSnippet.

A new PathSnippet replaces any previous PathSnippet.

Message necessity

●	○	○
2.1	2.2	2.3

In [1] the PathSnippet is part of the Drive Command.

Fingerprint	0x27737B3E
Size	6 bytes to 58 KB
Interface	TLS
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
identifier	uint32	4 bytes	
poses	vector PathPose	2 bytes to 58 KB	

C.13 RecordedMessage

This message encapsulates another AVP message which has been recorded in the Subject Vehicle for the transfer to the Remote Vehicle Operation (see Req82: [Vehicle] VehicleDataLogging).

Message neces-

●	●	●
2.1	2.2	2.3

Fingerprint	0x00B814DE
Size	10 bytes to 64 KB
Interface	TLS
Required	no
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
recordTime	second64	8 bytes	Time when this message has been recorded
buffer	Buffer	2 bytes to 64 KB	Serialized AVP message

C.14 RecordedMessages

Multiple recorded messages. It is more efficient to transfer multiple messages at once instead of one by one (see Req82: [Vehicle] VehicleDataLogging for details).

⚠ Size of payload cannot exceed 65535 bytes.

Message necessity

🟢	🟢	🟢
2.1	2.2	2.3

Fingerprint	0x0F4B4C51
Size	2 bytes to 64 KB
Interface	TLS
Required	no
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
messages	vector msg RecordedMessage	2 bytes to 64 KB	

C.15 SafetyTimeSyncRequest

This message is used for safe time synchronization. It is required for calculating a DrivingPermission's expiration time.
The Remote Vehicle Operation shall send this message to the Subject Vehicle in accordance with the respective safety requirements.
The Remote Vehicle Operation shall not send this message unless the vehicle identification through flashing lights was successful.

Message necessity

●	●	●
2.1	2.2	2.3

[1] does not have a checksum.

Fingerprint	0x5C0C1A89
Size	6 bytes
Interface	DTLS, 100 ms
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	not applicable

Name	Type	Size	Description
challenge	uint16	2 bytes	Challenge chosen by the Remote Vehicle Operation in accordance with the safety requirements. The Remote Vehicle Operation shall use this challenge to relate the response to this request. The Remote Vehicle Operation shall not use a challenge twice and abort a Mission if all challenges were used.
checksum	uint32	4 bytes	See Req127: [General] CalculateSafetyChecksum

C.16 SafetyTimeSyncResponse

The message contains the response to the safe time sync SafetyTimeSyncRequest.
The Subject Vehicle will not send this message unless the vehicle identification through flashing lights was successful.

Message necessity

●	●	●
2.1	2.2	2.3

[1] does not have a checksum. Furthermore, the unit for time differs.

Fingerprint	0x373C955D
Size	14 bytes
Interface	DTLS, on demand
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Safety Clock

Name	Type	Size	Description
challenge	uint16	2 bytes	Challenge received in the SafetyTimeSyncRequest message
currentVehicleSafetyClockTime	millisecond_ui64	8 bytes	Time of the Vehicle Safety Clock when SafetyTimeSyncRequest was received (shall equal time when SafetyTimeSyncResonse was sent)
checksum	uint32	4 bytes	See Req127: [General] CalculateSafetyChecksum

C.17 SafeVehicleTypeConfirmation

This safety-relevant message contains the Vehicle Type Identifier. Though the Vehicle Type Identifier was already sent to the Remote Vehicle Operation by the backends, this is a safe confirmation by the Subject Vehicle.
The Subject Vehicle shall send the SafeVehicleTypeConfirmation message once as soon as the vehicle identification was successful.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0xF7BB17E4
Size	6 to 38 bytes
Interface	TLS
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Safety Clock

Name	Type	Size	Description
vehicleType	string	2 to 34 bytes	Type identifier as defined in chapter A, “Vehicle Type Identifier”
checksum	uint32	4 bytes	See Req127: [General] CalculateSafetyChecksum

C.18 VehicleCapabilities

The message VelicleCapabilities contains non-safety related parameters of the Subject Vehicle.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x78A71FA6
Size	37 bytes
Interface	TLS, once as response to MissionConfirmation
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
controlInterfaceType	enum ControlInterfaceType	1 byte	The control interface requested by the Subject Vehicle.
maximumDriveableCurvatureForwards	perMetre32	4 bytes	Maximum driveable curvature when driving forwards (without rear-axle steering). Shall include a small buffer for the controller.
maximumDriveableCurvatureBackwards	perMetre32	4 bytes	Maximum driveable curvature when driving backwards (without rear-axle steering). Shall include a small buffer for the controller.
maximumPathSnippetSize	uint32	4 bytes	Maximum size of a PathSnippet message in [byte]. Each PathPose has a size of 20 bytes. Shall be large enough to allow at least 30 PathPoses (also see chapter H.2, “PathSnippet minimum length”). More is recommended, e.g. > 100 PathPoses. Only for ControlInterfaceType.PATH.
maximumPathSnippetFrequency	uint32	4 bytes	Maximum allowed frequency of PathSnippet messages in

			[Hz]. For a fast, yet comfortable reaction, 10 Hz is recommended (also see chapter H.1, “PathSnippet update frequency”). Only for ControlInterfaceType.PATH.
minimumDistanceBetween-PathPoses	metre32	4 bytes	Minimum allowed distance between two PathPoses. Shall be ⇐ 0.2 m. Only for ControlInterfaceType.PATH.
maximumDistanceBetween-PathPoses	metre32	4 bytes	Maximum allowed distance between two PathPoses. Shall be >= 1.0 m. Only for ControlInterfaceType.PATH.
pathSnippetTakeoverTime	millisecond_ui32	4 bytes	The time which is usually needed by the Subject Vehicle in order to apply a newly received PathSnippet. For a fast reaction, ⇐ 200 ms is recommended. Only for ControlInterfaceType.PATH.
vehicleTrajectoryDuration	millisecond_ui32	4 bytes	The time difference between first and last element in ControlTrajectoryElement/ StateTrajectoryElement in a VehicleTrajectory. Only for ControlInterface-Type.TRAJECTORY_CONTROLS and ControlInterfaceType.TRAJECTORY_FULL.
vehicleTrajectoryInterval	millisecond_ui32	4 bytes	The time difference between two consecutive elements in ControlTrajectoryElement/ StateTrajectoryElement in a VehicleTrajectory. Only for ControlInterface-Type.TRAJECTORY_CONTROLS and ControlInterfaceType.TRAJECTORY_FULL.

C.19 VehicleDebug

This message contains optional information from the Subject Vehicle that might help debugging and analysis.
The Subject Vehicle shall send this message cyclically while connected to the infrastructure.

Message necessity

● 2.1 ● 2.2 ● 2.3

Fingerprint	0xD057D5CD
Size	22 bytes
Interface	DTLS, about 50 ms cycle time
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
recordingState	enum RecordingState	1 byte	Current recording-state
recordedMessages	Uint32	4 bytes	Number of recorded messages in memory which were not sent to the Remote Vehicle Operation yet.
recordingSize	Uint32	4 bytes	Size of the recorded messages in memory which were not sent to the Remote Vehicle Operation yet, in byte.
requestedVelocity	metrePerSecond32	4 bytes	The currently requested Velocity sent to the Subject Vehicle. (DrivingPermissions shall be considered afterwards). Negative when driving backwards.
requestedRemainingDistance	metre32	4 bytes	The currently requested remaining distance sent to the Subject Vehicle. (DrivingPermissions shall be considered afterwards). Always positive.
requestedCurvature	perMetre32	4 bytes	The currently requested Curvature sent to the Subject Vehicle
powertrainActive	bool	1 byte	True if the power train is active, e.g. the engine is running

C.20 VehicleError

The Subject Vehicle shall send this message whenever an error occurred that prevents the Subject Vehicle from executing the given commands. Depending on the given error, the Remote Vehicle Operation either tries to resolve the issue or aborts the Mission, saves this message for logging and forwards its content to the backend.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0xB3961A8E
Size	18 bytes to 64 KB
Interface	TLS, on demand
Required	only in case of errors
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
time	second64	8 bytes	Timestamp of when the error occurred.
code	uint32	4 bytes	The error category, specified in chapter B, “Vehicle Error Codes”
ecuCode	uint32	4 bytes	Customer specific error code. The Remote Vehicle Operation will not interpret this value.
description	string	2 bytes to 64 KB	Optional description of the error with further details.

C.21 VehicleInfo

The Subject Vehicle can send this message whenever it has information that needs to be logged. The Remote Vehicle Operation will save this message for logging and forward its content to the backend.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x21DAF59D
Size	14 bytes to 64 KB
Interface	TLS, on demand
Required	no
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
time	second64	8 bytes	Timestamp of when the event happened
code	uint32	4 bytes	Customer specific code. The Remote Vehicle Operation will not interpret this value.
description	string	2 bytes to 64 KB	Optional description of the event with further details.

C.22 VehicleSafetyFeedback

This message is essential for monitoring, debugging and liability logging of the vehicle's functional safety actions.
It may be used by the Remote Vehicle Operation to adjust e.g. velocity or reaction time to improve driving performance.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0x713890B2
Size	5 to 16 bytes
Interface	TLS, about 10 ms cycle
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Safety Clock

Name	Type	Size	Description
drivingAllowed	bool	1 byte	true if the Subject Vehicle is currently allowed to drive, false if a safety stop happened (See Req111: [Vehicle] VehicleEvaluateDrivingPermission)
remainingTimeToDrive	millisecond_i16	2 bytes	The time which the Subject Vehicle is allowed to keep driving until brakes must be engaged. This is the difference between expiration time of most recent DrivingPermission and the current safe estimation of the server time, considering time sync uncertainties and signal delays between the safety component and the braking system. Shall be a signed value. (See Req111: [Vehicle] VehicleEvaluateDrivingPermission)
safetyViolations	vector enum SafetyStopReason	2 to 13 bytes	List of violations which currently lead to a SafetyStop.

C.23 VehicleState

This message contains the relevant information from the Subject Vehicle for performing a Mission.
The Subject Vehicle shall send this message cyclically while connected to the infrastructure.

Message necessity

●	●	●
2.1	2.2	2.3

In [1] sent information about current Curvature, Velocity, yaw rate, driving direction and shift position are included in the message “Vehicle state”. Additionally, it includes an information about the current PathSnippet ID, which can be found in VehicleState .

Therefore, the sent information’s are quite equally but without the yaw rate and shift position.

Fingerprint	0x201D70E3
Size	14 bytes
Interface	DTLS, about 50 ms cycle time
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
pathSnippetIdentifier	uint32	4 bytes	The identifier of the PathSnippet which the Subject Vehicle currently follows. 0 if not applicable.
operationMode	enum VehicleOperationMode	1 byte	The current operation mode or state of the Subject Vehicle
currentCurvature	perMetre32	4 bytes	The current Subject Vehicle Curvature
currentVelocity	metrePerSecond32	4 bytes	The current Subject Vehicle Velocity. Negative when driving backwards.
secureStandstill	bool	1 byte	True if the Subject Vehicle is currently in Secure Standstill (Secure Standstill), false otherwise

C.24 VehicleTrajectoryCommand

This message contains the time-based control trajectory that the vehicle has to execute, and the additional information required to drive.
The maximum size and frequency depend on the settings in VehicleCapabilities. Typically, 2 kBytes are needed for type 2.3.

- If the vehicle receives a new Trajectory, it shall entirely replace the previous Trajectory by it.
- The Trajectory contains controls (and vehicle states for type 2.3) for a time duration which is set by the vehicle at initialization time (as described in chapter 7.2.4.1, “Definitions”). The time discretization interval is constant and defined by the vehicle at initialization time.

Message necessity

○	●	●
2.1	2.2	2.3

Fingerprint	0xA8C9C5A4
Size	15 byte to 65 kByte
Interface	DTLS, about 100 ms cycle time
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	tbd

Name	Type	Size	Description
vehicleTrajectory	VehicleTrajectory	12 bytes to 65 kBytes	Controls and state Trajectory for the vehicle to execute.
drivingDirection	enum DrivingDirection	1 byte	The direction in which the vehicle must drive.

C.25VidRequest

This message is part of the vehicle identification process where the Subject Vehicle's position is verified through flashing the lights. It contains the current authorization state/ request and a seed necessary for calculating the code to be flashed and the code to be added to safety-related checksums.

Message necessity

● 2.1 ● 2.2 ● 2.3

Fingerprint	0xE2E4C5E4
Size	10 bytes
Interface	TLS
Required	yes
Sender	Remote Vehicle Operation
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
currentState	enum VidRequestState	1 byte	The current identification state/request of the Remote Vehicle Operation.
seed	uint64	8 bytes	Seed for calculating flashing codes and checksums.
codeLength	uint8	1 byte	Value that indicates how many bits from the seed shall be used for blinking. Will be between 8 and 20.

C.26 VidResponse

This message is part of the vehicle identification process where the Subject Vehicle's position is verified through flashing the lights.
The Subject Vehicle shall send this message to the Remote Vehicle Operation whenever the authorization state changed.

Message necessity

●	●	●
2.1	2.2	2.3

Fingerprint	0xDC76C02F
Size	1 byte
Interface	TLS
Required	yes
Sender	Subject Vehicle
Reference Clock	Vehicle Functional Clock

Name	Type	Size	Description
current-State	enum VidVehicleState	1 byte	The current identification state of the Subject Vehicle.

D AVP Enums

D.1 ControlInterfaceType

Selection of possible control interface types

Value uint8	Name	Description
0	PATH	AVP Type 2.1
1	TRAJECTORY_CONTROLS	AVP Type 2.2
2	TRAJECTORY_FULL	AVP Type 2.3

D.2 DirectionIndicator

Direction indicator signal aka turn signal

Value uint8	Name	Description
0	OFF	Do not flash lights.
1	RIGHT	Flash right
2	LEFT	Flash left
3	WARNING	Flash warning lights

D.3 DriveCommandAction

Value uint8	Name	Description
0	UNKNOWN	No command given by Remote Vehicle Operation. The Subject Vehicle shall stay in standby and wait for further instructions.
2	INITIALIZE	The Subject Vehicle shall initialize and prepare for an AVP Mission.
3	DRIVE	The Subject Vehicle shall actively drive and follow the path or trajectory commands.
4	TERMINATE	The Subject Vehicle shall disable all interfaces and shutdown as soon as possible.

D.4 DrivingDirection

Value uint8	Name	Description
0	UNKNOWN	Driving direction is unknown.
1	FORWARDS	Subject Vehicle is driving forwards.
2	BACKWARDS	Subject Vehicle is driving backwards.
3	STANDSTILL	Subject Vehicle is in Secure Standstill, e.g. in PARK gear.

D.5 DtlsInterfaceRequestState

DTLS Interface Request

Value uint8	Name	Description
0	UNKNOWN	Default value
1	START	Request new DTLS interface

D.6 DtlsInterfaceResponseState

DTLS Interface Reply

Value uint8	Name	Description
0	UNKNOWN	Default value
1	AVAILABLE	Request is successful. Interface available.
2	DENIED	Request failed. No interface available.

D.7 RecordingLevel

This enum describes the possible levels for the data logging and recording (See Req82: [Vehicle] VehicleDataLogging)

Value uint8	Name	Description
0	NORMAL	The Subject Vehicle shall record basic data that allows reconstruction of the vehicles externally visible behavior (e.g. how fast did it drive, when and why did it stop etc.)
1	VERBOSE	The Subject Vehicle shall additionally record information that could be helpful for debugging internal behavior.

D.8 RecordingState

Message recording state

Value uint8	Name	Description
0	STOPPED	Messages are not recorded.
1	RECORDING	Messages are being recorded.
2	TRANSFERRING	No messages are being recorded at the moment, but previously recorded messages are being transferred to the Remote Vehicle Operation.
3	FAILURE	Currently messages can't be recorded or transferred because conditions are not fulfilled unexpectedly.

D.9 SafetyStopReason

Possible reasons for a Safety Stop (functional safety)

Value uint8	Name	Description
1	NO_DRIVING_PERMISSION_RECEIVED	The Subject Vehicle did not receive any DrivingPermission yet.
2	LAST_DRIVING_PERMISSION_TOO_OLD	The most recent DrivingPermission expired more than 10 s ago (See Req122: [Vehicle] VehicleAbortsMissionAfter10s)
3	CRC_VIOLATION_CLOCK_SYNC_RESPONSE	The checksum of one or more SafetyTimeSyncResponse was not valid (See Req125: [Vehicle] VehicleVerifiesSafetyChecksums)
4	CRC_VIOLATION_DRIVING_PERMISSION	The checksum of one or more DrivingPermission was not valid (See Req125: [Vehicle] VehicleVerifiesSafetyChecksums)
5	EXPIRATION_TIME_VIOLATION	The expiration time of the most recent DrivingPermission has passed. (See Req120: [Vehicle] VehicleStopsOnExpirationTimeViolation)
6	DRIVING_DIRECTION_VIOLATION	The estimated driving direction doesn’t match the actual driving direction. (See Req113: [Vehicle] VehicleStopsOnWrongDrivingDirection)
7	VELOCITY_VIOLATION	The Subject Vehicle was driving faster than allowed in the most recent DrivingPermission. (See Req114: [Vehicle] VehicleStopsOnVelocityViolation)
8	CURVATURE_MIN_VIOLATION	The current Subject Vehicle Curvature was below the allowed minimum Curvature in the most recent DrivingPermission (See Req115: [Vehicle] VehicleStopsOnCurvatureViolation)
9	CURVATURE_MAX_VIOLATION	The current Subject Vehicle Curvature was above the allowed maximum Curvature in the most recent DrivingPermission (See Req115: [Vehicle] VehicleStopsOnCurvatureViolation)
10	EXPIRATION_TIME_TOO_HIGH	The expiration time in the most recent DrivingPermission is not valid, because it is too far in the future (See Req121: [Vehicle] VehicleStopsOnExpirationTimeTooHigh)
11	MONITORING	A monitoring detected a fault.

D.10 TerminateReason

Reasons for terminating a Mission

Value uint8	Name	Description
0	PROCEED	Everything is okay. No termination.
1	DESTINATION_REACHED	Termination because the destination was reached.
2	INFRASTRUCTURE_ERROR	Termination triggered by error in Remote Vehicle Operation.
3	VEHICLE_ERROR	Termination triggered by error in Subject Vehicle.

D.11 VehicleOperationMode

Possible states of the Subject Vehicle

Value uint8	Name	Description
0	UNKNOWN	
1	INITIALIZING	The Subject Vehicle is preparing for the Mission but hasn't entered Safe Driving State yet.
2	SAFE_DRIVING_STATE_STANDBY	The Subject Vehicle is in safe driving state, but currently doesn't follow a PathSnippet or trajectory actively.
3	SAFE_DRIVING_STATE_DRIVING	The Subject Vehicle is in safe driving state and either actively follows a given PathSnippet and hasn't reached the end of it yet or is executing the trajectory commands. Also applies if the Subject Vehicle stopped temporarily because of a SafetyStop.
4	TERMINATING	The Subject Vehicle left Safe Driving State and is terminating AVP related functions.

D.12 VidRequestState

All states/requests from the Remote Vehicle Operation for the vehicle identification process

Value uint8	Name	Description
0	UNDEFINED	Default value
1	FLASHING	Infrastructure is prepared and waiting for the Subject Vehicle to flash the code.
2	SUCCESSFUL	The Subject Vehicle was recognized correctly and the identification is completed
3	NEW_CODE	A new Subject Vehicle identification cycle was started.

D.13 VidVehicleState

All states/requests from the Subject Vehicle for the vehicle identification process on Subject Vehicle.

Value uint8	Name	Description
0	UNDEFINED	Default value
1	READY	Subject Vehicle is ready to get flashing code.
2	FLASHING_COMPLETED	Flashing is finished.
3	AUTHORIZED	Subject Vehicle identification was successful, and Subject Vehicle has switched its state.

E Certificate Chain

There are no further descriptions regarding certificates in the ISO 23374 norm.

Figure 33 shows the certificate chain involved in the mutual authentication process between Subject Vehicle and Remote Vehicle Operation.

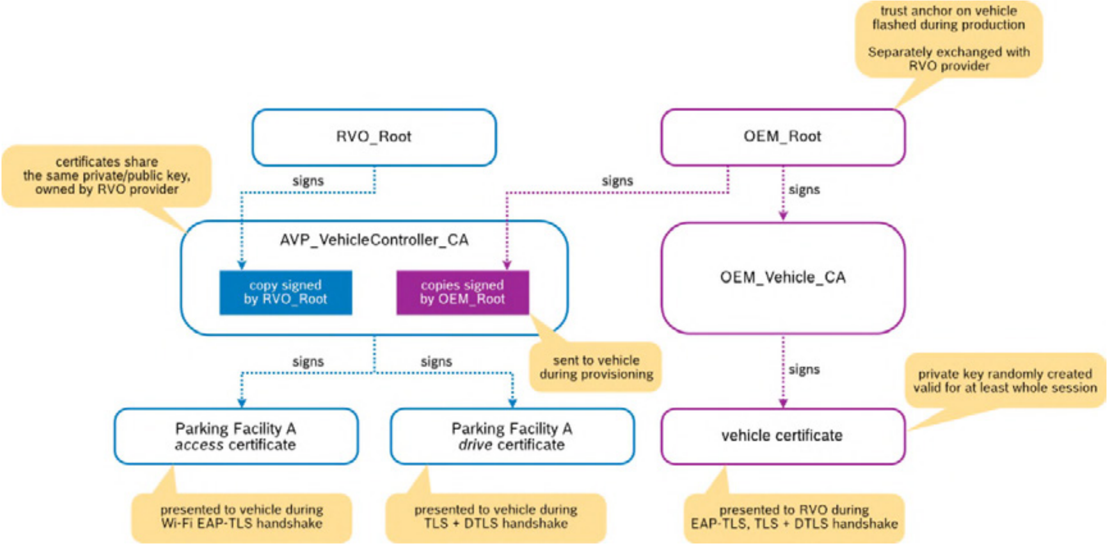

trust anchor on vehicle flashed during production
Separately exchanged with RVO provider
RVO_Root
OEM_Root
certificates share the same private/public key, owned by RVO provider
signs
signs
signs
AVP_VehicleController_CA
copy signed by RVO_Root
copies signed by OEM_Root
OEM_Vehicle_CA
signs
signs
sent to vehicle during provisioning
signs
private key randomly created valid for at least whole session
Parking Facility A access certificate
Parking Facility A drive certificate
vehicle certificate
presented to vehicle during Wi-Fi EAP-TLS handshake
presented to vehicle during TLS + DTLS handshake
presented to RVO during EAP-TLS, TLS + DTLS handshake

Figure 33 - Certificate Chain for Communication with Remote Vehicle Operation

E.1 AVP_Vehicle_Controller_CA

The certificate structure of the AVP_Vehicle_Controller_CA intermediate certificate which is used to issue parking facility certificates shall comply to the following definition:

Table 12 - Fields of the AVP_Vehicle_Controller_CA intermediate certificates

Field	Value	Comment
Version	3	
Signature Algorithm	sha384ECDSA	
Signature Hash Algorithm	sha384	
Issuer	RVO_Root or OEM_Root	
Valid to	maximum 2 years	

Subject	CN=AVP_Vehicle_Controller_CA
	O=Name of Remote Vehicle Operation provider	

Public Key	ECC (384 bits)	
Public Key Parameters	ECDSA_P384	
Authority Key Identifier	Hash of the RVO_Root or OEM_Root public key	Generated according to rfc5280 4.2.1.2.(1) "composed of the 160- bit SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)."
Subject key Identifier	Hash of the certificates public key	see above
Enhanced Key Usage: critical	Server Authentication (1.3.6.1.5.5.7.3.1)	
Basic Constraints: critical	Subject Type=CA, Path Length Constraint=0	
Key Usage: critical	Certificate Signing, Off-line CRL Signing, CRL Signing (06)	

E.2 Parking Facility Certificates

The structure of the end entity certificates on Remote Vehicle Operation side shall comply to the following definition:

Table 13 - Fields of the end entity certificates used by parking facilities

Field	Value	Comment
Version	3	
Signature Algorithm	sha384ECDSA	
Signature Hash Algorithm	sha384	
Issuer	AVP_Vehicle_Controller_CA	
Valid to	maximum 6 months	

Subject	CN=<ParkingFacilityIdentifier>
	ST=access/drive
	O=Name of Remote Vehicle Operation provider
	Using the state field (ST), we differentiate between Wi-Fi (= access) and TLS/DTLS (= drive)	

Public Key	ECC (384 bits)	
Public Key Parameters	ECDSA_P384	
Authority Key Identifier	Hash of the AVP_Vehicle_Controller_CA public key	Generated according to rfc5280 4.2.1.2.(1) "composed of the 160- bit SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)."
Subject key Identifier	Hash of the certificates public key	see above
Enhanced Key Usage: critical	Server Authentication (1.3.6.1.5.5.7.3.1)	Enables mutual authentication
Basic Constraints: critical	Subject Type=End Entity, Path Length Constraint=None	
Key Usage: critical	Digital Signature, Key Encipherment (a0), Key Agreement	

E.3 Vehicle Certificate

The certificate structure of the vehicle certificate shall comply to the following definition:

Table 14 - Fields of the end entity certificates used by AVP vehicles

Field	Value	Comment
Version	3	X509v3
Signature Algorithm	sha384ECDSA	
Signature Hash Algorithm	sha384	
Issuer	OEM_Vehicle_CA	
Valid to	At least anticipated duration of Mission. Recommended: anticipated duration of a session.	

Subject	CN=AVP_Vehicle
	ST=access/drive/access+drive
	• The value of ST depends on whether the certificate is used for WiFi (= access), TLS/DTLS (= drive) or both (= access+drive)	

Public Key	ECC (384 bits)	
Public Key Parameters	ECDSA_P384	
Authority Key Identifier		given by issuer (OEM_Vehicle_CA)
Subject key Identifier		given by issuer (OEM_Vehicle_CA)
Enhanced Key Usage: critical	Client Authentication (1.3.6.1.5.5.7.3.2)	
Basic Constraints: critical	Subject Type=End Entity, Path Length Constraint=None	
Key Usage: critical	Digital Signature, Key Encipherment (a0), Key Agreement	

F Complete Sequence of AVP Mission Process

The following graphs show the complete AVP sequence between Operator Backend and Vehicle Backend, as well as Subject Vehicle and Remote Vehicle Operation. The steps are explained in detail in the respective chapters.

For simplification, the sequence does not include the complete communication between Operator Backend and Remote Vehicle Operation, because this part is solely within the scope of the respective Remote Vehicle Operation provider.

Communication between Vehicle Backend and Subject Vehicle is completely within the scope of the respective OEM.

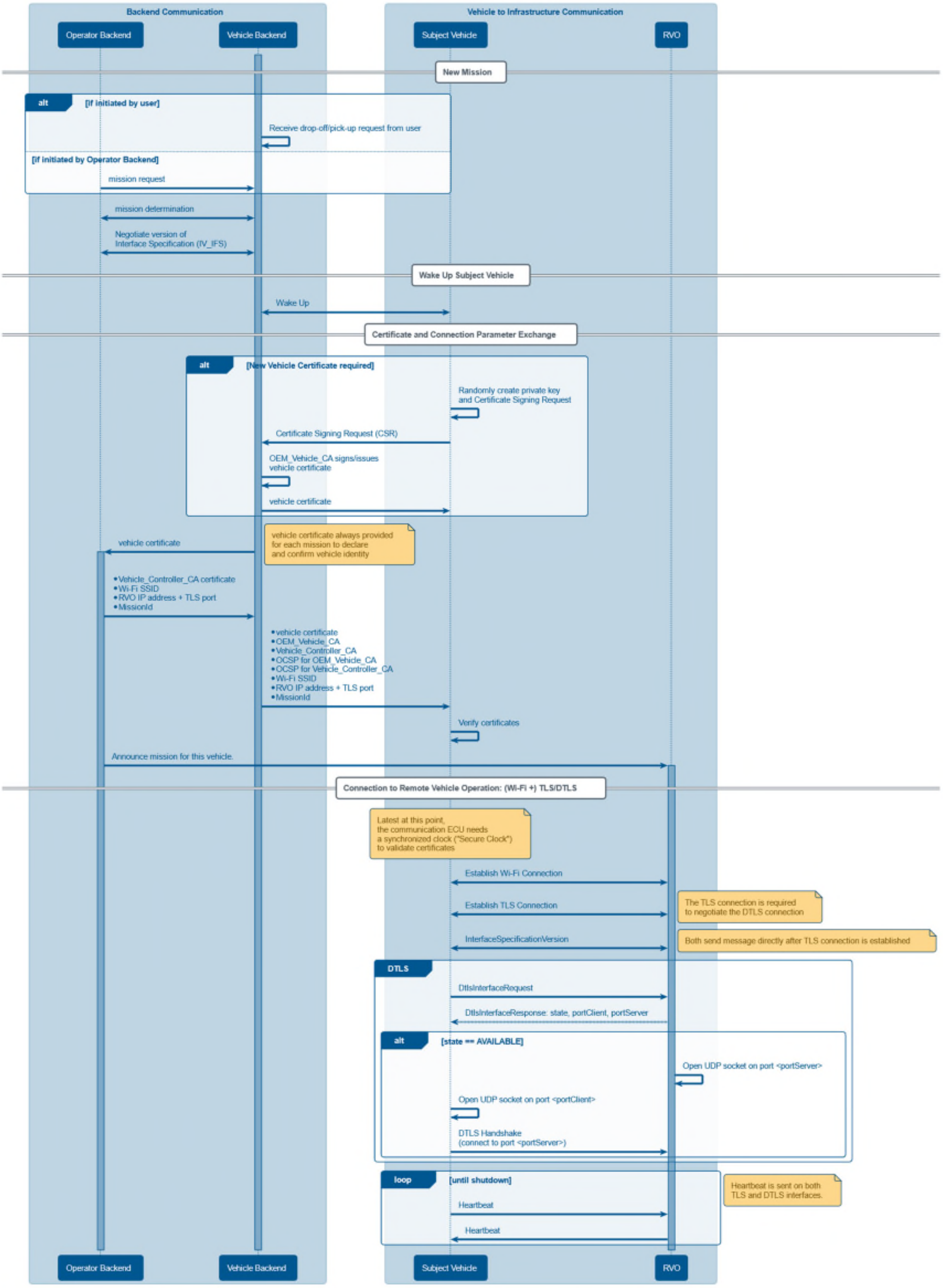
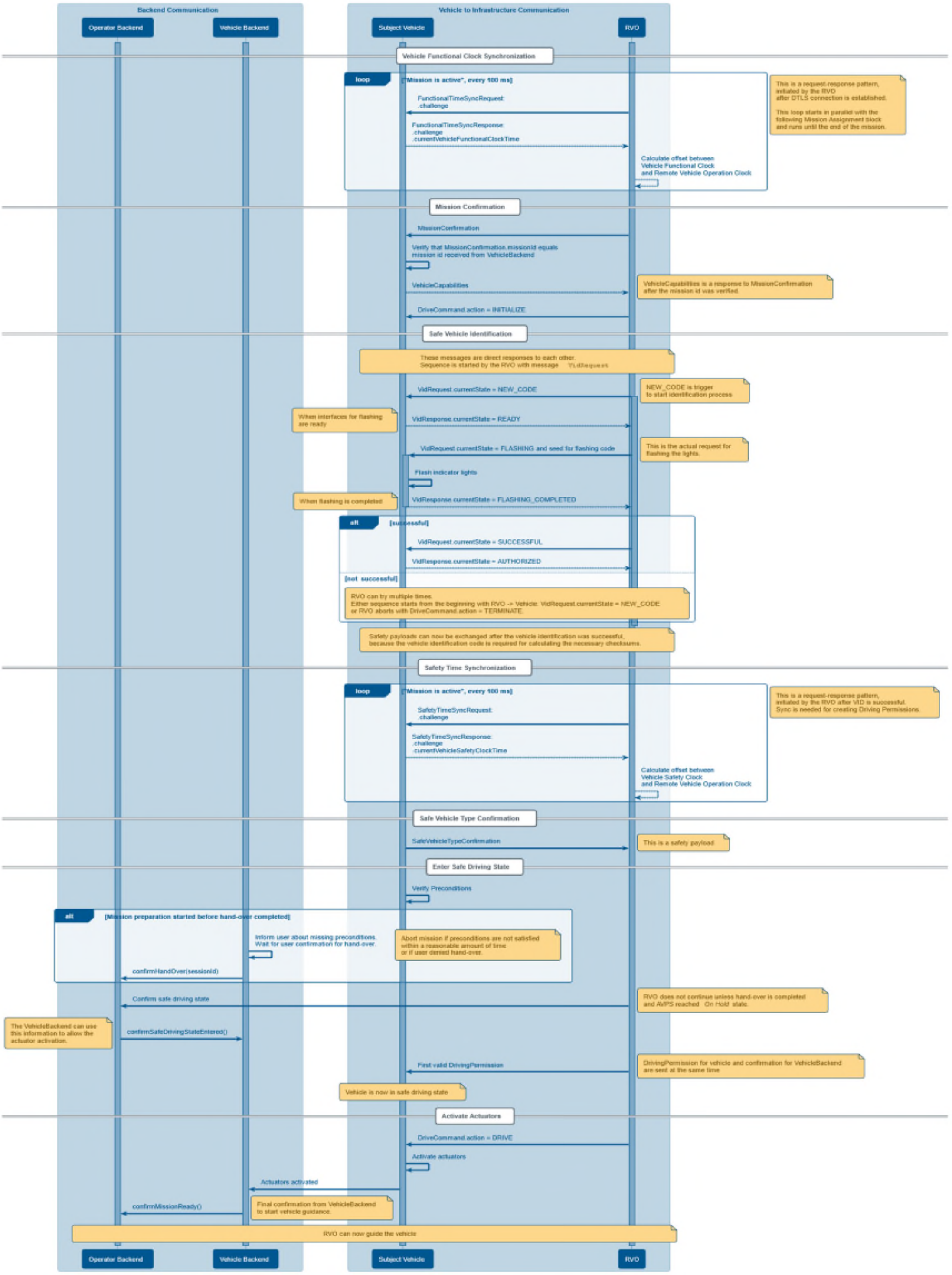

Backend Communication
Operator Backend
Vehicle Backend
Vehicle to Infrastructure Communication
Subject Vehicle
RVO
New Mission
alt
[If initiated by user]
Receive drop-off/pick-up request from user
[If initiated by Operator Backend]
mission request
mission determination
Negotiate version of
Interface Specification (IV_IFS)
Wake Up Subject Vehicle
Wake Up
Certificate and Connection Parameter Exchange
alt
[New Vehicle Certificate required]
Randomly create private key
and Certificate Signing Request
Certificate Signing Request (CSR)
OEM_Vehicle_CA signs/issues
vehicle certificate
vehicle certificate
vehicle certificate
vehicle certificate always provided
for each mission to declare
and confirm vehicle identity
•Vehicle_Controller_CA certificate
•Wi-Fi SSID
•RVO IP address + TLS port
•MissionId
•vehicle certificate
•OEM_Vehicle_CA
•Vehicle_Controller_CA
•OCSP for OEM_Vehicle_CA
•OCSP for Vehicle_Controller_CA
•Wi-Fi SSID
•RVO IP address + TLS port
•MissionId
Verify certificates
Announce mission for this vehicle.
Connection to Remote Vehicle Operation: (Wi-Fi +) TLS/DTLS
Latest at this point,
the communication ECU needs
a synchronized clock ("Secure Clock")
to validate certificates
Establish Wi-Fi Connection
Establish TLS Connection
The TLS connection is required
to negotiate the DTLS connection
InterfaceSpecificationVersion
Both send message directly after TLS connection is established
DTLS
DtlsInterfaceRequest
DtlsInterfaceResponse: state, portClient, portServer
alt
[state == AVAILABLE]
Open UDP socket on port <portServer>
Open UDP socket on port <portClient>
DTLS Handshake
(connect to port <portServer>)
loop
[until shutdown]
Heartbeat
Heartbeat
Heartbeat is sent on both
TLS and DTLS interfaces.
Operator Backend
Vehicle Backend
Subject Vehicle
RVO

Figure 34 - Complete Sequence of AVP Mission Process (Part 1)


Backend Communication
Vehicle to Infrastructure Communication
Operator Backend
Vehicle Backend
Subject Vehicle
RVO
Vehicle Functional Clock Synchronization
loop
["Mission is active", every 100 ms]
FunctionalTimeSyncRequest:
.challenge
FunctionalTimeSyncResponse:
.challenge
.currentVehicleFunctionalClockTime
Calculate offset between
Vehicle Functional Clock
and Remote Vehicle Operation Clock
Mission Confirmation
MissionConfirmation
Verify that MissionConfirmation.missionId equals
mission id received from VehicleBackend
VehicleCapabilities
DriveCommand.action = INITIALIZE
Safe Vehicle Identification
VidRequest.currentState = NEW_CODE
VidResponse.currentState = READY
VidRequest.currentState = FLASHING and seed for flashing code
Flash indicator lights
VidResponse.currentState = FLASHING_COMPLETED
alt
[successful]
VidRequest.currentState = SUCCESSFUL
VidResponse.currentState = AUTHORIZED
[not successful]
Safety Time Synchronization
loop
["Mission is active", every 100 ms]
SafetyTimeSyncRequest:
.challenge
SafetyTimeSyncResponse:
.challenge
.currentVehicleSafetyClockTime
Calculate offset between
Vehicle Safety Clock
and Remote Vehicle Operation Clock
Safe Vehicle Type Confirmation
SafeVehicleTypeConfirmation
Enter Safe Driving State
Verify Preconditions
alt
[Mission preparation started before hand-over completed]
Inform user about missing preconditions.
Wait for user confirmation for hand-over.
confirmHandOver(sessionId)
Confirm safe driving state
confirmSafeDrivingStateEntered()
First valid DrivingPermission
Activate Actuators
DriveCommand.action = DRIVE
Activate actuators
Actuators activated
confirmMissionReady()
Operator Backend
Vehicle Backend
Subject Vehicle
RVO

This is a request-response pattern, initiated by the RVO after DTLS connection is established.

This loop starts in parallel with the following Mission Assignment block and runs until the end of the mission.

VehicleCapabilities is a response to MissionConfirmation after the mission id was verified.

These messages are direct responses to each other. Sequence is started by the RVO with message `VidRequest`

NEW_CODE is trigger to start identification process

When interfaces for flashing are ready

This is the actual request for flashing the lights.

When flashing is completed

RVO can try multiple times. Either sequence starts from the beginning with RVO -> Vehicle: VidRequest.currentState = NEW_CODE or RVO aborts with DriveCommand.action = TERMINATE.

Safety payloads can now be exchanged after the vehicle identification was successful, because the vehicle identification code is required for calculating the necessary checksums.

This is a request-response pattern, initiated by the RVO after VID is successful. Sync is needed for creating Driving Permissions.

This is a safety payload

Abort mission if preconditions are not satisfied within a reasonable amount of time or if user denied hand-over.

RVO does not continue unless hand-over is completed and AVPS reached On Hold state.

The VehicleBackend can use this information to allow the actuator activation.

DrivingPermission for vehicle and confirmation for VehicleBackend are sent at the same time

Vehicle is now in safe driving state

Final confirmation from VehicleBackend to start vehicle guidance.

RVO can now guide the vehicle

Figure 35 - Complete Sequence of AVP Mission Process (Part 2)
]

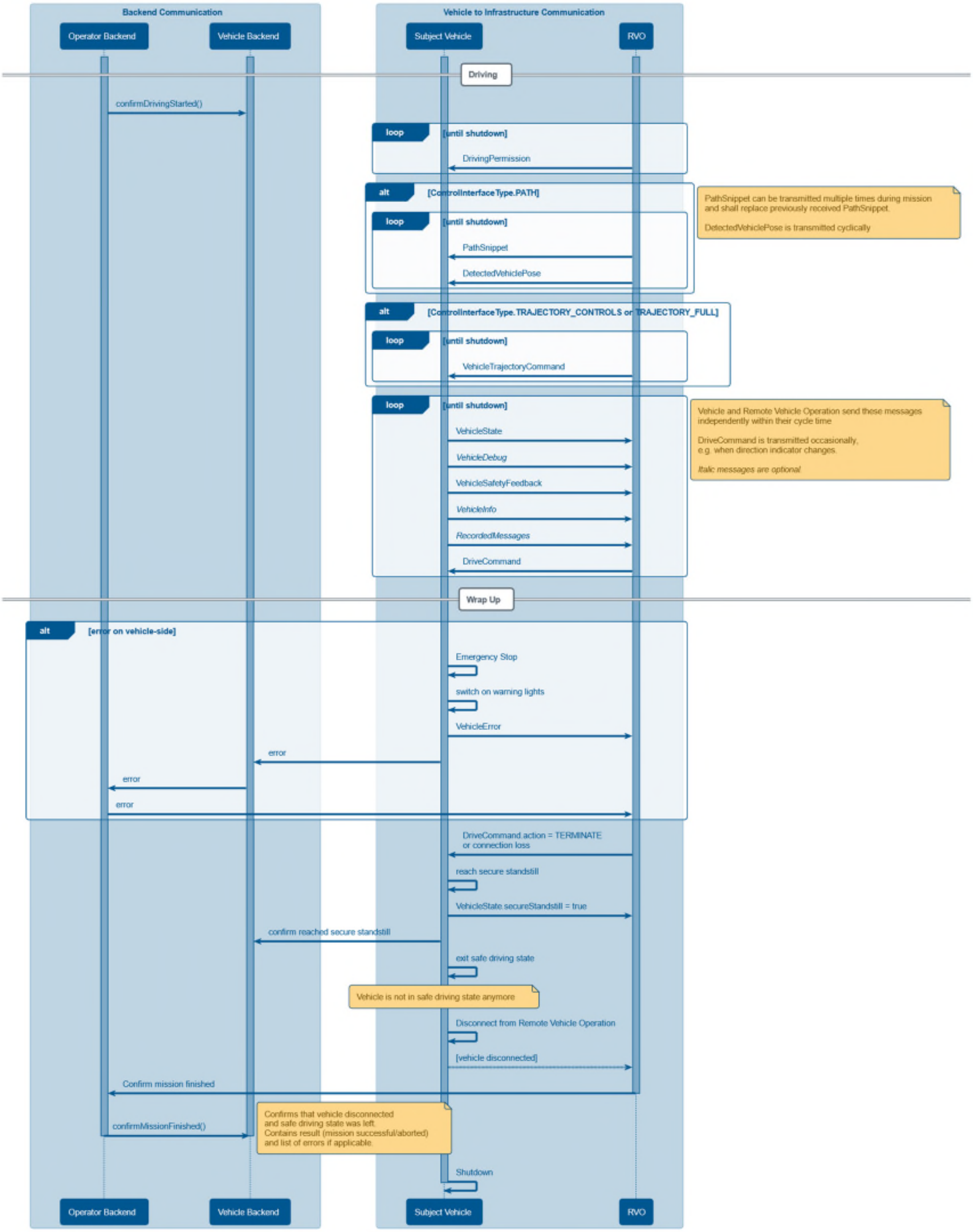

Backend Communication
Vehicle to Infrastructure Communication
Operator Backend
Vehicle Backend
Subject Vehicle
RVO
Driving
confirmDrivingStarted()
loop
[until shutdown]
DrivingPermission
alt
[ControlInterfaceType.PATH]
loop
[until shutdown]
PathSnippet
DetectedVehiclePose
PathSnippet can be transmitted multiple times during mission
and shall replace previously received PathSnippet.
DetectedVehiclePose is transmitted cyclically
alt
[ControlInterfaceType.TRAJECTORY_CONTROLS or TRAJECTORY_FULL]
loop
[until shutdown]
VehicleTrajectoryCommand
loop
[until shutdown]
VehicleState
VehicleDebug
VehicleSafetyFeedback
VehicleInfo
RecordedMessages
DriveCommand
Vehicle and Remote Vehicle Operation send these messages
independently within their cycle time
DriveCommand is transmitted occasionally,
e.g. when direction indicator changes.
Italic messages are optional
Wrap Up
alt
[error on vehicle-side]
Emergency Stop
switch on warning lights
VehicleError
error
error
error
DriveCommand.action = TERMINATE
or connection loss
reach secure standstill
VehicleState.secureStandstill = true
confirm reached secure standstill
exit safe driving state
Vehicle is not in safe driving state anymore
Disconnect from Remote Vehicle Operation
[vehicle disconnected]
Confirm mission finished
confirmMissionFinished()
Confirms that vehicle disconnected
and safe driving state was left.
Contains result (mission successful/aborted)
and list of errors if applicable.
Shutdown
Operator Backend
Vehicle Backend
Subject Vehicle
RVO

Figure 36 - Complete Sequence of AVP Mission Process (Part 3)

G Subject Vehicle system states and transition diagram

Figure 37 illustrates the state transition based on the condition of a single Subject Vehicle. Therefore, AVPS manages multiple states when simultaneously managing multiple Subject Vehicles. AVPS shall comply with the contents specified herein. Further implementation beyond the following specifications is up to system design.

ⓘ The inner rectangular box with a dotted line labelled “Automated vehicle operation” indicates that a Mission has been assigned, and the Subject Vehicle is automatically operated by AVPS through the states within the box.

ⓘ The outer rectangular box with a solid line labelled “System management” indicates that a Session has been established and the Subject Vehicle is managed by AVPS in the states within the box.

⚠ More in-depth information regarding the states and trasitions can be found in [1], chapter 9.3.

Table 15 - Description of Subject Vehicle State [1]

State	Description
Inactive	A valid session does not exist. The Subject Vehicle may be located within the operation zone.
Ready	Subject Vehicle identification is complete, session is established, and the User has the Authority.
Standby	Subject Vehicle is managed by AVPS and waiting for further commands or User retrieval. The standby state consists of two sub-states, “wait” and “sleep”. **Wait sub-state:** Subject Vehicle shall be capable of immediate transition to the Depart/ Arrive state. **Sleep sub-state:** Subject Vehicle is in a low energy consumption condition. During this state, the Subject Vehicle may carry out maintenance (e.g. online updates). During such maintenance, wake-up/mission requests may not be processed.
Depart/ Arrive	AVPS determines the destination and Route or waits for further commands after reaching the destination.
Normal	AVPS operates the Subject Vehicle in this state. The planned destination and Route may be changed during this state. The normal state consists of two sub-states, “drive” and “pause”. **Drive sub-state:** The Subject Vehicle is moving. **Pause sub-state:**

	The Subject Vehicle is stationary. This sub-state is important to be recorded in order to distinguish the state from other vehicle stationary conditions, such as temporary error or suspend states.
Temporary Error	AVPS recognizes occurrence of a temporary error. Depending on the type of the error, AVPS may continue automated vehicle operation.
Suspend	AVPS becomes incapacitated to perform or continue automated vehicle operation without sufficient measures taken by the P subsystem.
Recovery	The Subject Vehicle or PFE is physically accessed by the P subsystem in this state. Duration to be within the recovery state is not limited and may be applied as long as necessary.

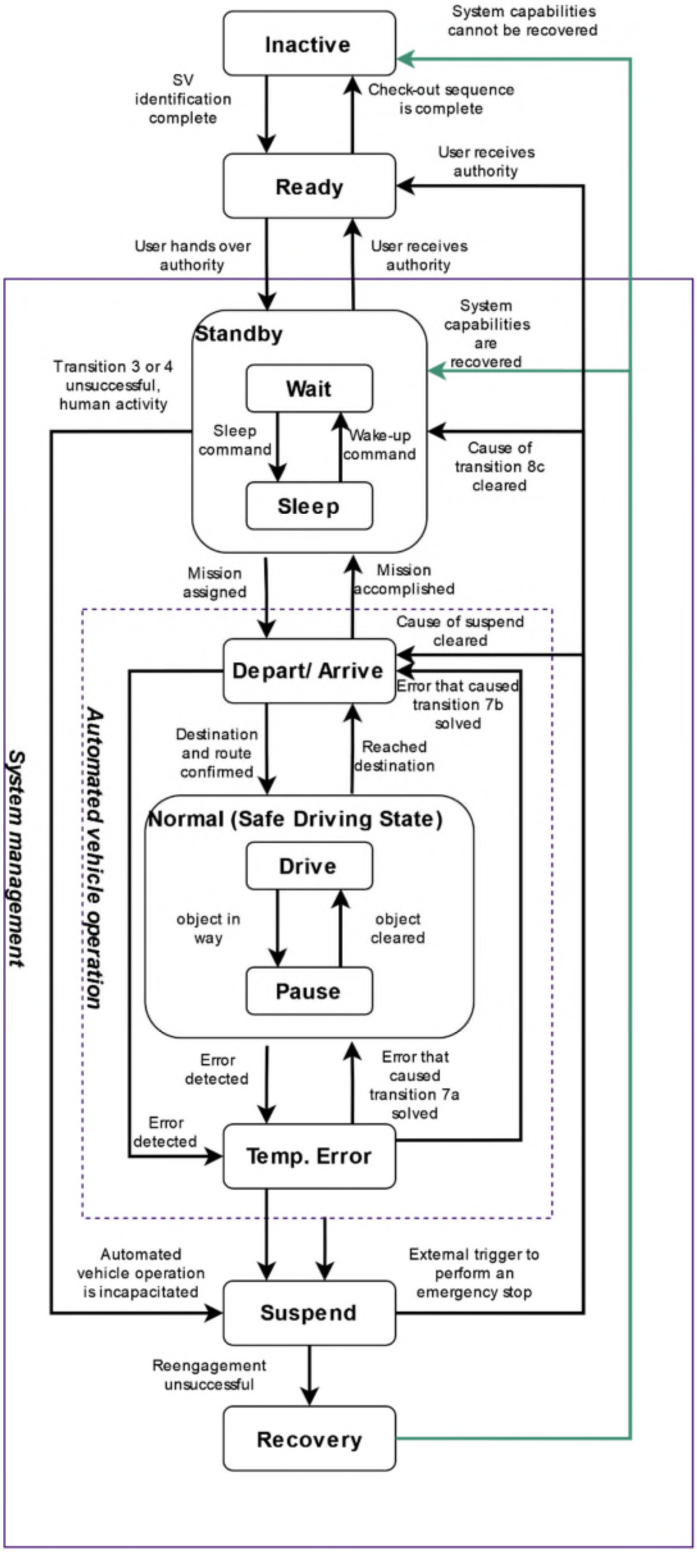

Inactive
System capabilities cannot be recovered
SV identification complete
Check-out sequence is complete
Ready
User receives authority
User hands over authority
User receives authority
Standby
System capabilities are recovered
Transition 3 or 4 unsuccessful, human activity
Wait
Sleep command
Wake-up command
Cause of transition 8c cleared
Sleep
Mission assigned
Mission accomplished
Cause of suspend cleared
Depart/ Arrive
Error that caused transition 7b solved
Destination and route confirmed
Reached destination
Normal (Safe Driving State)
Drive
object in way
object cleared
Pause
Error detected
Error that caused transition 7a solved
Error detected
Temp. Error
Automated vehicle operation
System management
Automated vehicle operation is incapacitated
External trigger to perform an emergency stop
Suspend
Reengagement unsuccessful
Recovery

Figure 37 - State transition diagram [1]

H Additional Information

This chapter describes the approach to derive recommended values regarding the PathSnippet update frequency, the PathSnippet length, VehicleTrajectory length and a maximum corner curvature.

⚠ It should be noted that the exact values or calculation approaches are negotiated between the infrastructure provider and the OEM and may therefore differ from the values or approaches described here.

H.1 PathSnippet update frequency

In AVP Type 2 the infrastructure is responsible for the Path planning. Once an oncoming vehicle or another dynamic Object is recognized by the Remote Vehicle Operation, the calculation of a new Path is provoked. In the case of Type 2.1, the Remote Vehicle Operation can hand over a new PathSnippet for an evasion maneuver instead of an Emergency Stop.

The time distribution that must be considered to perform an evasion maneuver by the AVPS by sending a new PathSnippet is shown in Figure 38.

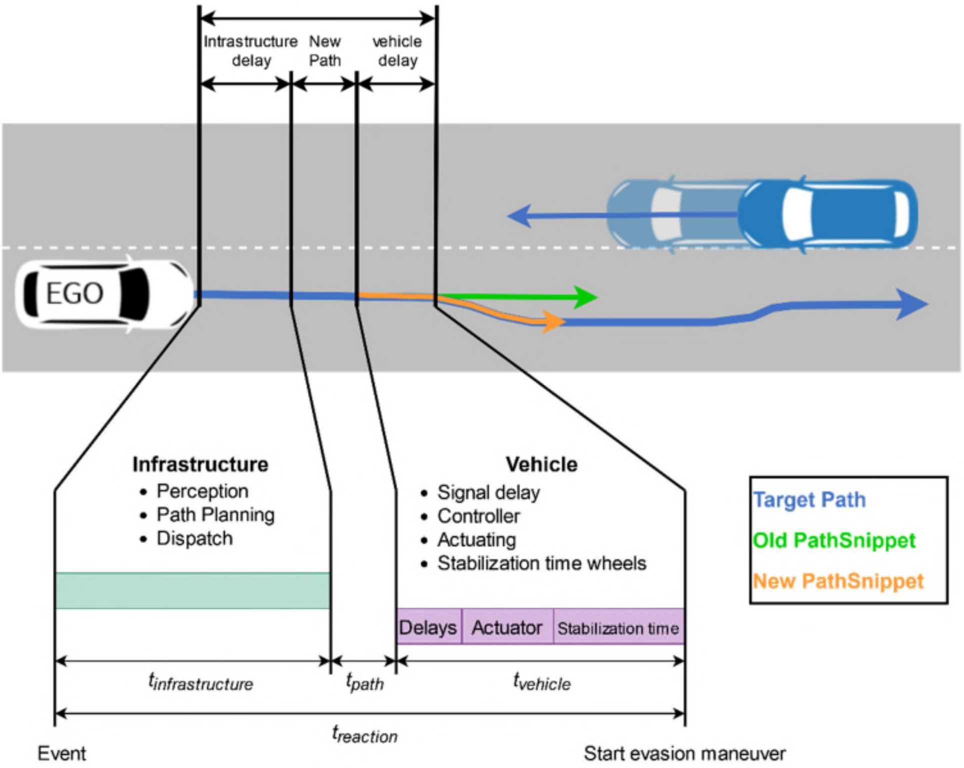


Figure 38 - Update times of planned path

The time $t_{reaction}$ defines the total time available for the AVP system to react. A reasonable assumption for the reaction time of an AVPS is $t_{reaction} = 1s$, because regarding chapter 7.2.1.3, “Requirements for Dynamic Driving Task (DDT)” the system needs to perform equal or better than an experienced and attentive driver. The response time from an event to the arrival at the vehicle is denoted with $t_{infrastructure} + t_{sending}$ and includes the time of Remote Vehicle Operation as well as Wireless data transmission. $t_{vehicle}$ is the effective dead time between the signal reception in the vehicle and motion conversion.

To ensure the system reacts in the defined reaction time $t_{reaction}$, the update rate of the PathSnippet must be within t_{path} by considering $t_{reaction} > t_{infrastructure} + t_{sending} + t_{vehicle}$.

To avoid unnecessary data which cannot be used by the Subject Vehicle, the required update frequency of the PathSnippet should be equal to or smaller than the frequency of the lateral controller.

This is described by the following equation:

$f_{controller} \geq f_{path} \geq \frac{1}{t_{path}}$ with $t_{path} = t_{reaction} - t_{infrastructure} - t_{sending} - t_{vehicle}$

In the following, example values to predict a reasonable update frequency are listed. They relate the values named in chapter 7.2.5.4, “Expiration Time “ without considering Clock uncertainties.

Table 16 - Example values for path update frequency

Time	Assumed values
$t_{reaction}$	1 s
$t_{infrastructure}$	0.5s
$t_{sending}$	0.03 s
$t_{vehicle}$	0.3 s
Resulting f_{path}	**8.3 Hz**

To ensure an equal reaction time to an experienced and attentive driver, the PathSnippet update frequency shall be at approximately 10 Hz.

H.2 PathSnippet minimum length

For the calculation of the required minimum length of the PathSnippet, two dependencies must be considered:

- Braking distance
- Needs and performance of the lateral controller

H.2.1 Emergency stop

The vehicle requirement for the minimum PathSnippet length is based on the vehicle's reaction time and the length of the braking distance, which is required for a controlled braking procedure until Standstill.

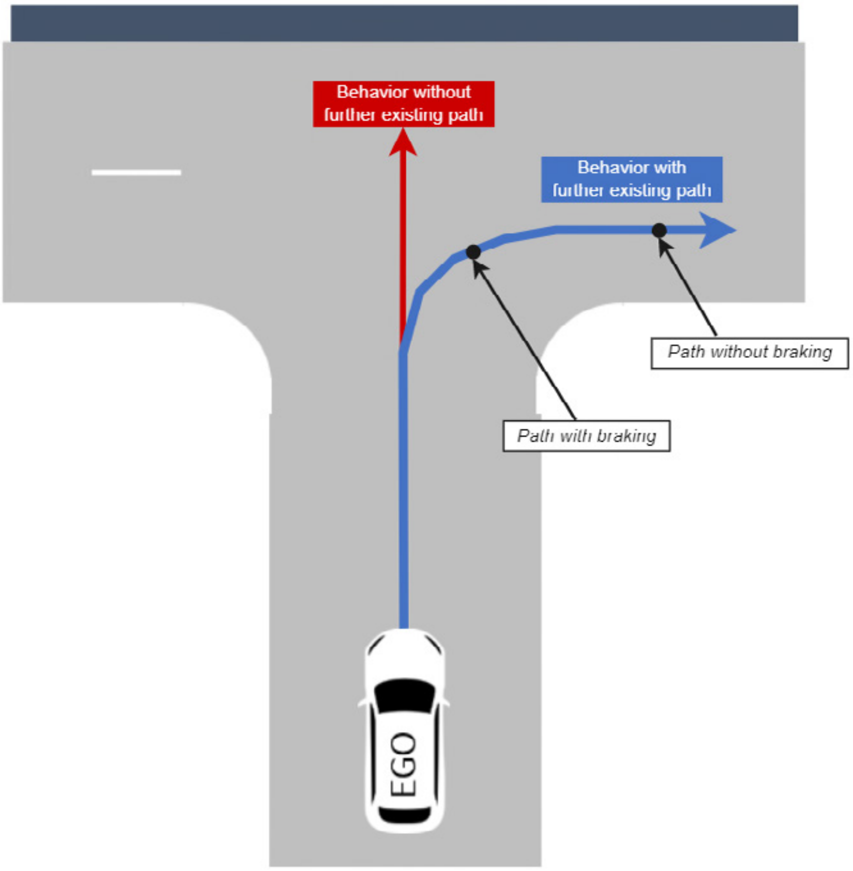

Behavior without further existing path
Behavior with further existing path
Path without braking
Path with braking
EGO

Figure 39 - Behavior with further existing path

The available time to answer a critical event is distributed between different modules. A simplified presentation of the time division is shown in Figure 40:

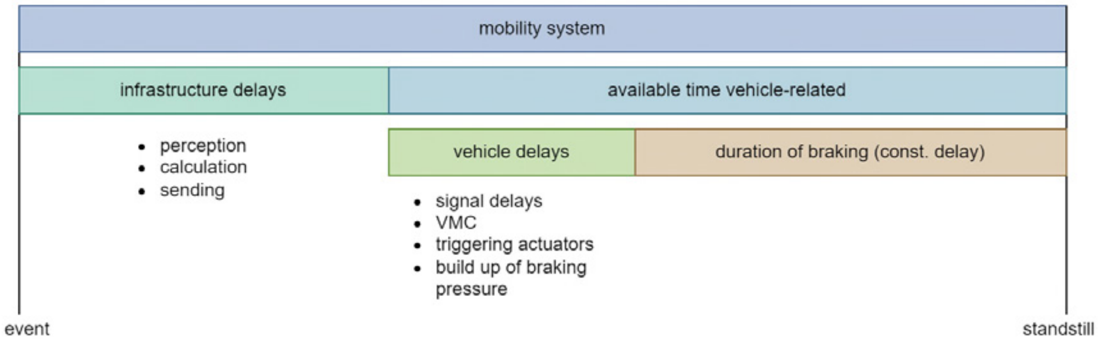


Figure 40 - Time distribution in critical situation

The available time for the mobility system is differently defined in [1] (also see Emergency Stop). The following consideration are based on the smaller time budget at 10 km/h, which are 1000 ms + time for defined braking distance (see Req11: [Vehicle] VehicleBrakingPerformance). The braking distance also includes a time of $\mathrm{t}_{decelerationBegins} - \mathrm{t}_{brakingInitiated} = 250\ ms$ until the maximum deceleration is applied. In this approach the simplification is made, that the total braking force is applied immediately. Regarding Table 6 the maximum braking distance $s_{braking}$ is at 1.49 m when driving 10 km/h.

The Path needs to be long enough to describe the vehicle's position within the braking process until Standstill when driving maximum velocity of 10 km/h. Regarding chapter 7.2.5.4, “Expiration Time ” the DrivingPermission needs approximately 680 ms to get to the vehicle. Thus, the available time on the vehicle’s side without braking is at around 320 ms.

Furthermore, the update frequency of the DrivingPermission must be considered to avoid complications due to asynchronous message reception. The minimum required PathSnippet can be calculated as follows:

$$s_{path} = v_{max} \cdot [t_{available} + t_{cycle,DrivingPermission}] + s_{braking}$$

For $v_{max} = 10\frac{km}{h}$, $t_{available} = 320\ \mathrm{ms}$ and $t_{cycle,DrivingPermission} = 100\ \mathrm{ms}$ it results in a minimum PathSnippet length of $s_{path} = 2.7\ m$.

⚠ The minimum length of a PathSnippet should be around 3 m to ensure a safe Emergency Stop. For comfort stopping it should be longer.

H.2.2 Lateral control (comfort driving)

A feedforward-steering-control is one possibility to control a vehicle in the lateral direction. The minimum required length of the PathSnippet is dependent of three times. A so-called lookahead-time can be used to identify the upcoming curvature of the corner as shown in Figure 41. In addition, a part that considers the needed update time of the PathSnippet is required (also see chapter H.1, “PathSnippet ”. Also, a time buffer shall be taken into account to ensure considering vehicle processing times or a missing message.

The minimum required PathSnippet length can be calculated the following way:

$$s_{path}(v) = v \cdot \left(t_{lookahead} + \frac{1}{f_{path}} + t_{buffer}\right)$$

Regarding chapter H.1, “PathSnippet ”, an update frequency of 10 Hz is recommended. With a lookahead-time $t_{lookahead} = 1\ s$ and a buffer time $t_{buffer} = 1\ s$ the recommended minimum length results in $s_{path} = 5.9\ m$

⚠ The minimum length of a PathSnippet should be around 6 m to ensure comfort driving.

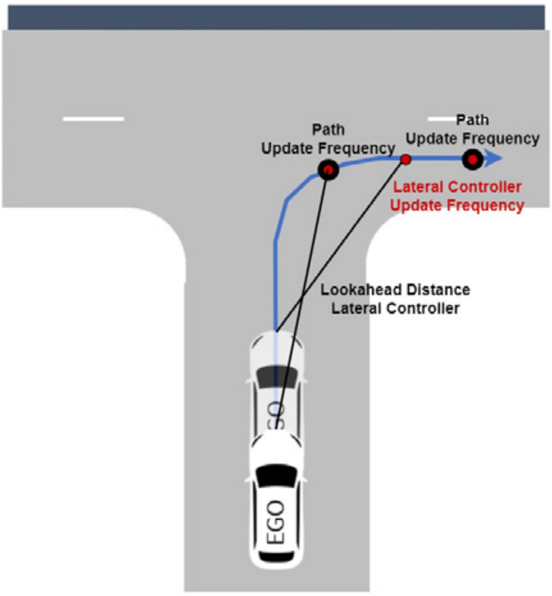


Figure 41 - Required path length for comfortable driving

H.2.3 Summary

Summarized the following can be defined for the considered examples:

- A minimum Path length of about **3 m** is necessary for an Emergency Stop.
- For comfortable driving and an update frequency of 10 Hz a PathSnippet length of about **6 m** is required.

> With a minimum PathSnippet resolution of 0.2 m (see VehicleCapabilities.minimumDistanceBetweenPathPoses), the minimum amount of path poses result in $\frac{6\ m}{0.2\frac{m}{pose}} = 30\ poses.$

H.3 Trajectory minimum length

The recommended minimum required Trajectory time-length (VehicleCapabilities.vehicleTrajectoryDuration) for the type 2.2 and 2.3 depends on the expiration time defined in chapter 7.2.5.4, “Expiration Time ” as well as the braking performance of the vehicle defined in Req11: [Vehicle] VehicleBrakingPerformance.

The defined maximum distance is split in a distance, that considers e.g. the time needed to establish the braking force, and the distance while the target braking force is applied. The first distance is defined by the driven velocity and a time of $250\ ms$.

To meet the specified braking distance of $s_{braking} = 0.8\ m$, a mean deceleration $\overline{d}$ of

$$\overline{d} = \frac{v^2}{2 \cdot s_{braking}} = 4.9\ \frac{m}{s^2}$$

must be applied at a velocity $v = 10 \frac{km}{h}$.The needed time for the braking process is at $t_{braking} = \frac{v}{\overline{d}} = 0.58\ s \approx 0.6\ s$.

ⓘ Thus, the recommended total time length of the Trajectory is at $1.85\ s$.

⚠ To ensure the demanded mean deceleration, a minimum friction between the tires and the road surface of $\mu \approx 0.5$ is required in the plane and of $\mu \approx 0.65$ for a ramp with $17\ \%$ slope.

H.4 Minimum allowed curvature

To ensure the steering capability of a vehicle, the maximum possible drivable curvature must be considered. Assuming a wet road surface with a friction coefficient of $\mu \approx 0.6$, a minimum allowed corner radius can be calculated. It is based on Kamm's circle, which includes a simplified relation between the longitudinal and lateral force potential of a tire. It is assumed, that the maximum side force in longitudinal direction is similar to the lateral direction. Regarding chapter H.3, “Trajectory minimum length”, the mean longitudinal deceleration is $4.9 \frac{m}{s^2}$.

$$a_{max} = \mu \cdot g = \sqrt{(d_x^2 + a_y^2)}$$

$$a_{max}(\mu = 0.6) = 5.9 \frac{m}{s^2}$$

$$a_{y,max} = \sqrt{(a_{max}^2 - d_x^2)} = 3.3 \frac{m}{s^2}$$

$$R_{min=} \frac{v^2}{a_{y,max}} = 2.4\ m$$

$$\kappa_{max} = \frac{1}{R_{min}} = 0.4 \frac{1}{m}$$

Thus, the minimum corner radius referred to the vehicles center of gravity must not be greater than $2.4\ m$. To ensure a safe driving in every corner by a maximum velocity of $10 \frac{km}{h}$, the maximum allowed bend of a corner is at $0.4 \frac{1}{m}$.

For a vehicle with a wheelbase of $5\ m$, this results in a maximum Ackermann-steering of $63.5\ °$. A maximum Ackermann steering of $45\ °$ results in a minimum wheelbase of $2.5\ m$.

Contact persons

Dr. Marcus Bollig
Managing Director
Products & Value Creation | Automotive Technologies & Ecosystems
marcus.bollig@vda.de

Henry Kuhle
Head of Coordination Unit for Connected & Automated Driving
Department Automotive Technologies and Eco-systems
henry.kuhle@vda.de

If you notice any errors, omissions or ambiguities in these recommendations, please contact VDA without delay so that these errors can be rectified.

Publisher
German Association of the Automotive Industry
Behrenstraße 35, 10117 Berlin
www.vda.de

Registered representative R001243
EU Transparency No. 95574664768-90

Version
Version 1.0, March 2023

@VDA_online
Verband der Automobilindustrie