Document Title	Explanation of Application Interfaces of the Body and Comfort Domain
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	268

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.4.0

Document Change History

Date	Release	Changed by	Change Description
2018-10-31	4.4.0	AUTOSAR Release Management	• Editorial changes
2017-12-08	4.3.1	AUTOSAR Release Management	• Editorial changes
2016-11-30	4.3.0	AUTOSAR Release Management	• Extension of number of ports of Body composition • Editorial changes
2014-10-31	4.2.1	AUTOSAR Release Management	• Editorial changes
2013-03-15	4.1.1	AUTOSAR Administration	• Update of “Antitheft Warning System” functionality Software Component Decomposition • Detailed “Operation Mode Manager” signal levels inside” Terminal Clamp Control” functionality • Updated description for “Convertible Control” functionality
2011-12-22	4.0.3	AUTOSAR Administration	• Inconsistencies fixed between Application Interfaces Master Table and Body Domain Explanatory Document : all functions impacted • Update of all functionality Software Components Decompositions for better readability • Removed ParkDistance Control functionality

Document Change History

Date	Release	Changed by	Change Description
2010-09-30	3.1.5	AUTOSAR Administration	• Update of all chapters according to the introduction of the concept "port blue print" : all functions impacted • Inconsistencies fixed between Application Interfaces Master Table and explanatory document for the body domain: all functions impacted
2010-02-02	3.1.4	AUTOSAR Administration	• Functions added : ExteriorLights, ATWS, Horn, ParkDistanceControl, DefrostControl, TerminalClamp Control, Immobilizer, SeatAdjustment, SeatClimatization, PassiveEntry, Mirror and RemoteKeylessEntry, ConvertibleControl, BodySensors, KeyPad • Functions maintenance on : Interior Light, WiperWasher, Central Locking • Legal disclaimer revised
2008-08-13	3.1.1	AUTOSAR Administration	• Legal disclaimer revised
2007-12-21	3.0.1	AUTOSAR Administration	• Initial Release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Purpose of this Document

This document explains all design decisions that lead to the AI contents.

In case of inconsistencies between pictures and explanations in this document and the AI, the information in the AI has to be considered as prior.

2 Description of Terms and Concepts of the Body- and Comfort Domain

2.1 Design Rationale

This design standard is the AUTOSAR-architecture (RTE-view) for Software Compositions described in chapter 4. It provides a decomposition into SW-Cs and a list of standardized interfaces related to each functionality (like Interior Lights).

This decomposition is limited in granularity to sensor components, adapter components, the core functionality and actuator components.

The intent is for the decomposition to get not to atomic SW-Cs but to "purchasable" SW-Cs (as long as these are not sensor/actuator SW-Cs they are realized as AUTOSAR compositions for formal reasons). These SW-Cs will be obtained as a unit so that all the internal (and hence not AUTOSAR standardized) interfaces are controlled by a single vendor, even if there are SW-Cs within the bought unit that reside on different ECUs. All interfaces between SW-Cs from different vendors should be standardized.

In addition, all ports are described, showing their AUTOSAR data qualities. Invalidation (where needed) is defined as in-band invalidation. Init values are specified where appropriate e.g. "off", "idle", "undefined", "unknown" … Where standardization is inappropriate, recommendations may be provided.

SW-C with a dashed line (in opposite to one with a simple line) means that this standardization does not describe it in a more precise level. During the implementation, this SW-C might be further decomposed

2.2 Abbreviations

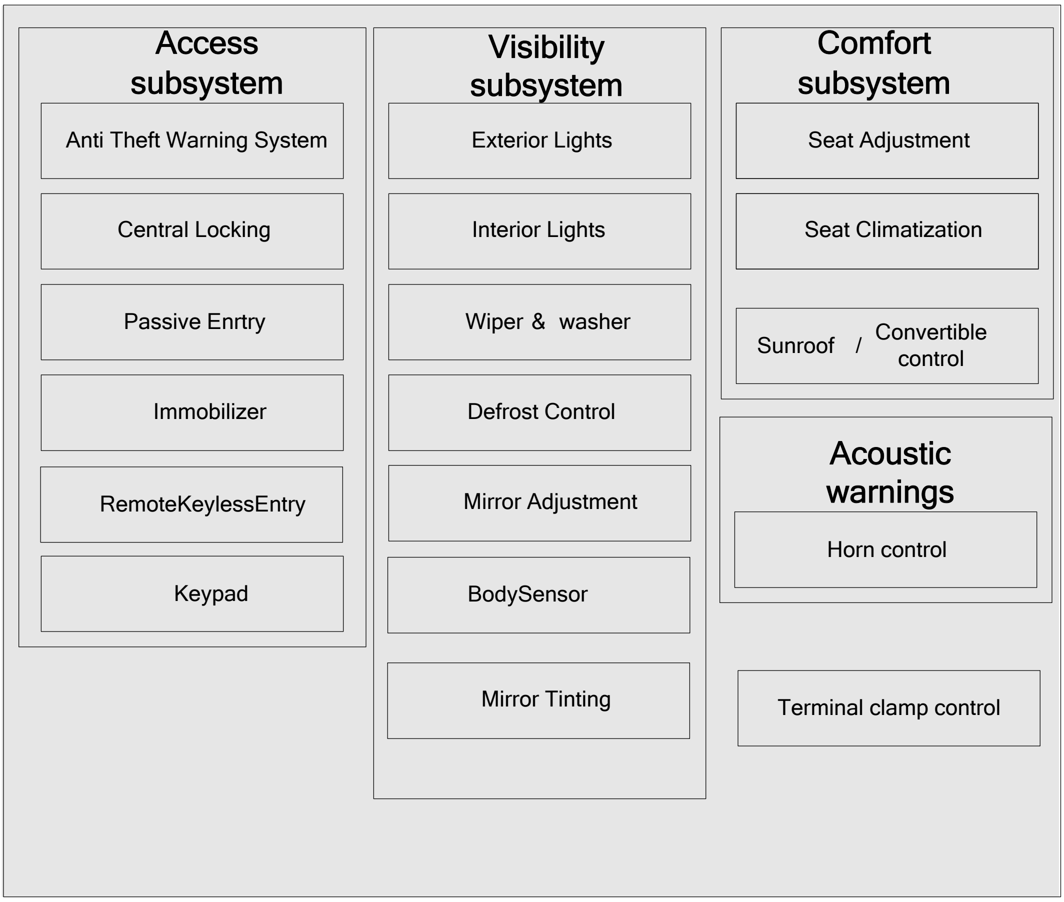
List of abbreviation used in this document:

Abbreviations	Meanning
ATWS	Anti-Theft Warning System
BBS	Battery Backed Sensor
CAN	Controller Area Network
CHLH	Coming Home/Leaving Home
CL	Central Locking
ECM	Engine Control Module
GBS	Glass Brake Sensor
HMI	Human Machine Interface
ID	Identity
IMMO	Immobilizer
INCL	Inclination sensor

ISC	Interior Scanner
LED	Light-Emitting Diode
LHFD	Left Hand Front Door
LHRD	Left Hand Rear Door
OEM	Original Equipment Manufacturer
PASE	Passive Entry
PATS	PATS: Passive Alarm Theft Sensor
RF	Radio Frequency
RHFD	Right Hand Front Door
RHRD	Right Hand Rear Door
RKE	Remote Keyless Entry
HVAC	Heating, Ventilation and Air Conditioning
DC	Defrost Control
SC	Seat Climatization
SA	Seat Adjustment
SW-C	Software Component

3 Decomposition Overview

This section gives an overview of the top level de-composition of the Body and Comfort Domain.



Decomposition overview Fig 1

The function components from Body and Comfort domain are identified according to the controlled group of actuators (motors, bulbs and other special devices) and type of operations (car access, lighting, comfort, acoustic signalling and other indirect operations like Terminal clamp control or battery monitoring (future function)). These function components can be grouped also in different ways considering other criterias.

The defined subsystems do not always mean a strong link between the included function components. The subsystems are also linked between them by a set of interfaces. For example the Access subsystem is strongly linked with a part of the Visibility subsystem.

There is a considerable complexity of interconnections inside body domain but also a set of interfaces are provided and required from other domains.

4 Description of Software Compositions and Components

The following Software Compositions and Components were identified in the body- and comfort domain.

4.1 Wiper/Washer

The WiperWasher software component group controls the wiper and washer functionality of a car. WiperWasher receives driver wishes, senses the environment and controls the behaviour of the wipers and washers of a car.

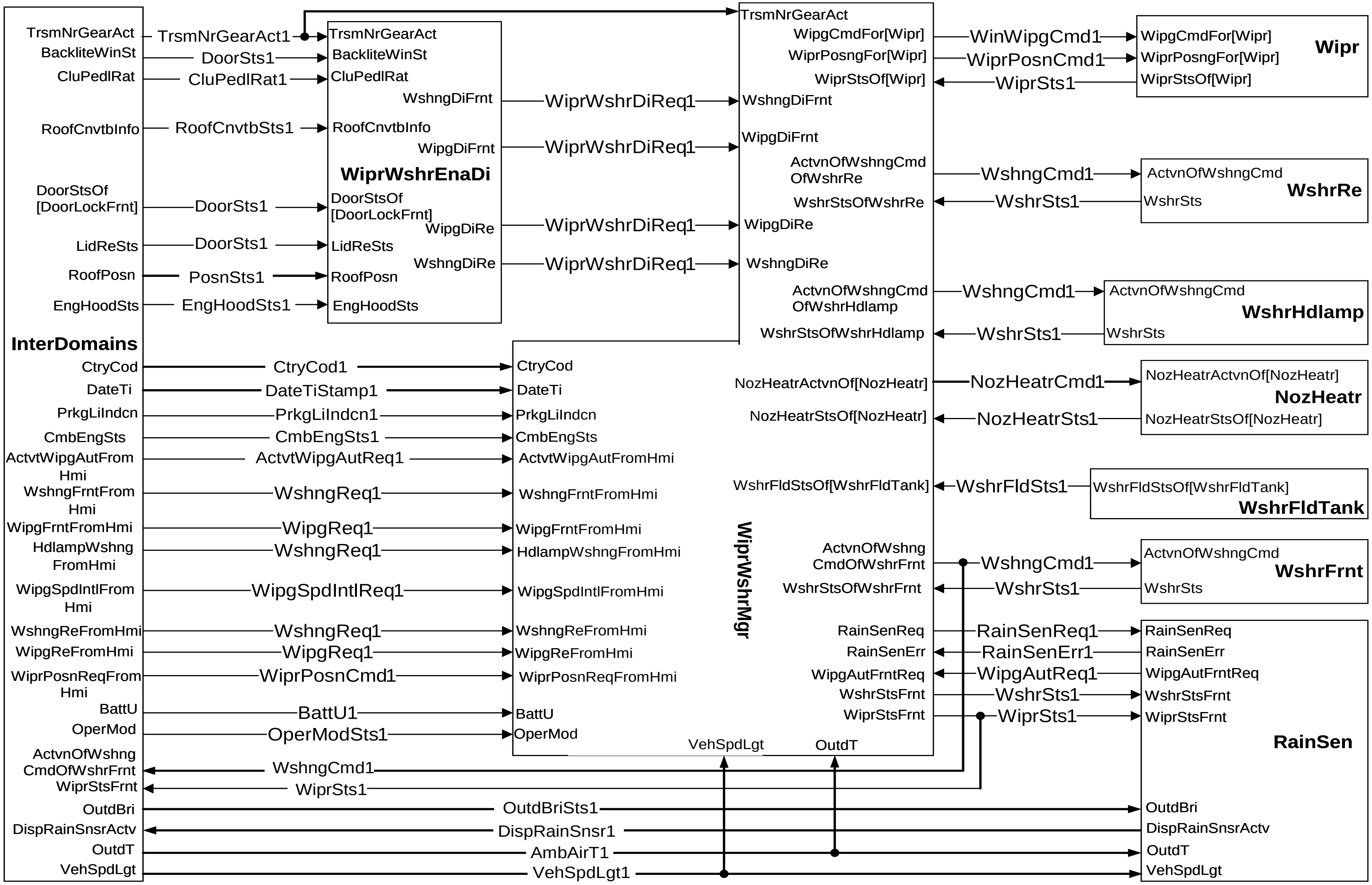
Components related to WiperWasher are described in the following sections.

ExteriorLight provides information about the outside brightness for the Rain Sensing component.
TerminalClampControl and BatteryMonitor (future function) provide information about the current state of the vehicle.
Rain Sensing, consists of the Rain Sensor and an adaptor component that calculates the wiping requests, considering output from ExteriorLight and other environmental and vehicle-related information.
WasherFluidTank senses the Level of the Washer Fluid container(s), it is instantiated two times to represent the front and rear washer tank.
EnableDisableWiperWasher is an adaptation component that manages the enabling of wiper washer functionality according to vehicle specific needs.
HMI contains all sensor (and switch) components that detect the drivers demands related to WiperWasher.
WiperWasherManager provides the core functionality.

Washer is an actuator component. It provides all the functionality needed for washing. It is reused to represent the front and rear washers, as well as the headlamp washer.
FrontWasher is an actuator component. It contains all funtionality for the washing of the windshield.
RearWasher is an actuator component. It contains all funtionality for the washing of the rear windshield.
WasherHeadLamp is an actuator component. It contains all funtionality for the washing of the Head Lamp.

Wiper is an actuator component. It provides all the functionality needed for the wiping of the window. It is instantiated to represent the front and rear washers.
FrontWiper is an actuator component. It provides all the functionality needed for the wiping of the windshield.
RearWiper is an actuator component. It provides all the functionality needed for the wiping of the rear windshield.

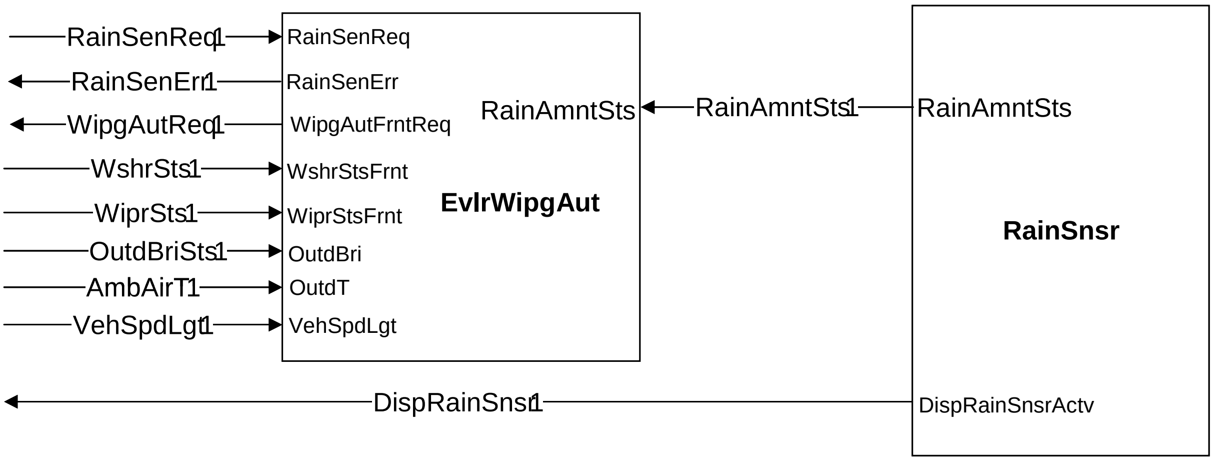
NozzleHeater is an actuator component. It contains all functionality to heat the headlamp nozzle, including tubes. This component type is instantiated several times to represent the necessary heater for front and rear washer and for the head lamp washer.
Here below is provided the Wiper Washer SWCs decomposition.
Data flows are indicated by arrows.



WiperWasher fig1

4.1.1 Rain sensing

This section explains the SW-component rain sensing. The following figure gives insight into the decomposition of the SW-composition rain sensing.



WiperWasher fig2

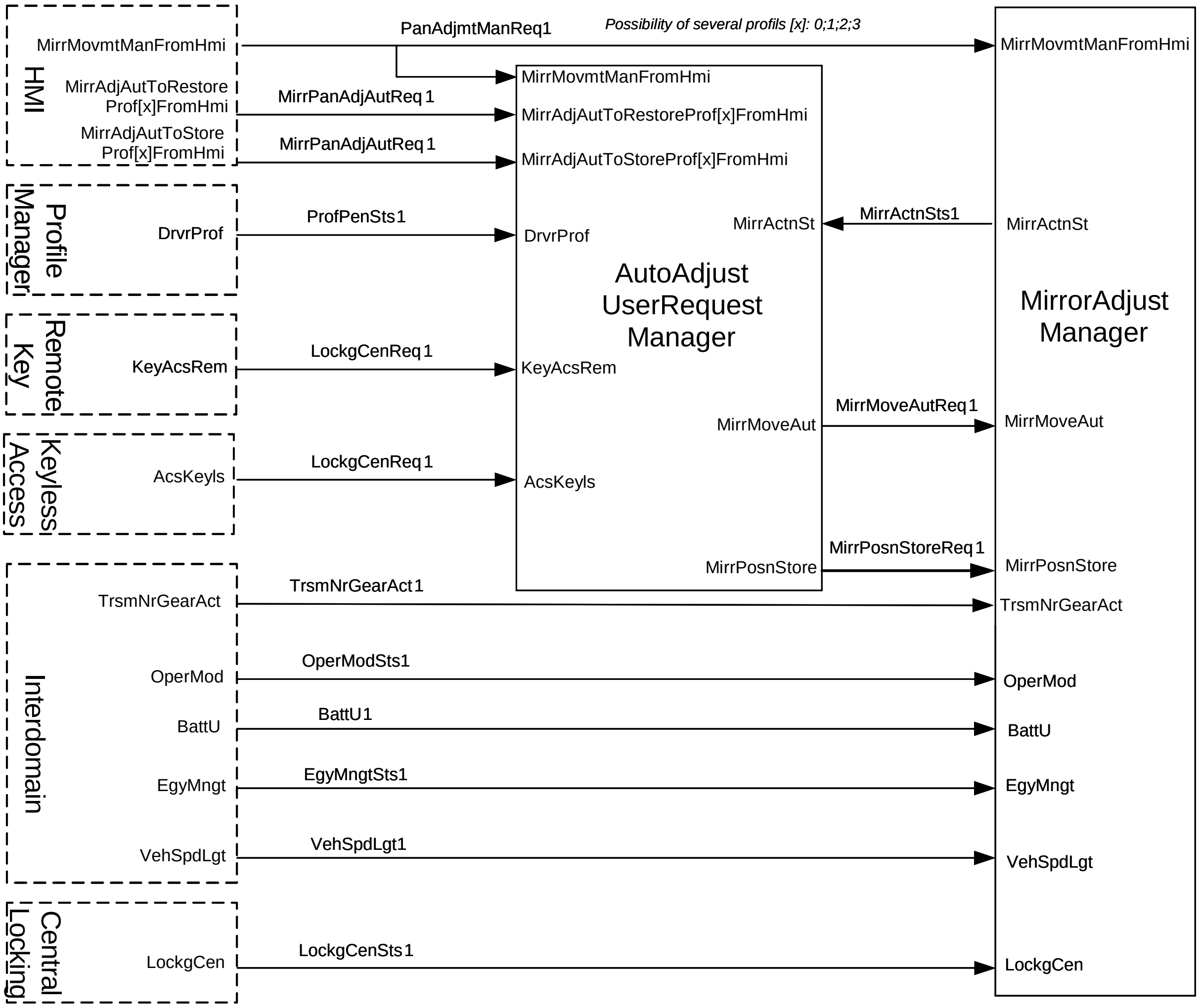
Rain sensing can be split up to the basic `RainSensor`, which delivers some basic information about the detected rain that can be used to calculate / evaluate the need for automatic front wiping.
In the above figure this information is exchanged via the port RainAmountStatus(`RainAmntSts)`. As this information is strongly dependent on the used technology, the type of the data-element is currently not binding.
`AutoWipingEvaluator` calculates/ evaluates the request for automatic front wiping. It takes several additional environmental information into account, including temperature, brightness, speed, wiper/washer information.

4.2 Mirror Adjustment & Tinting

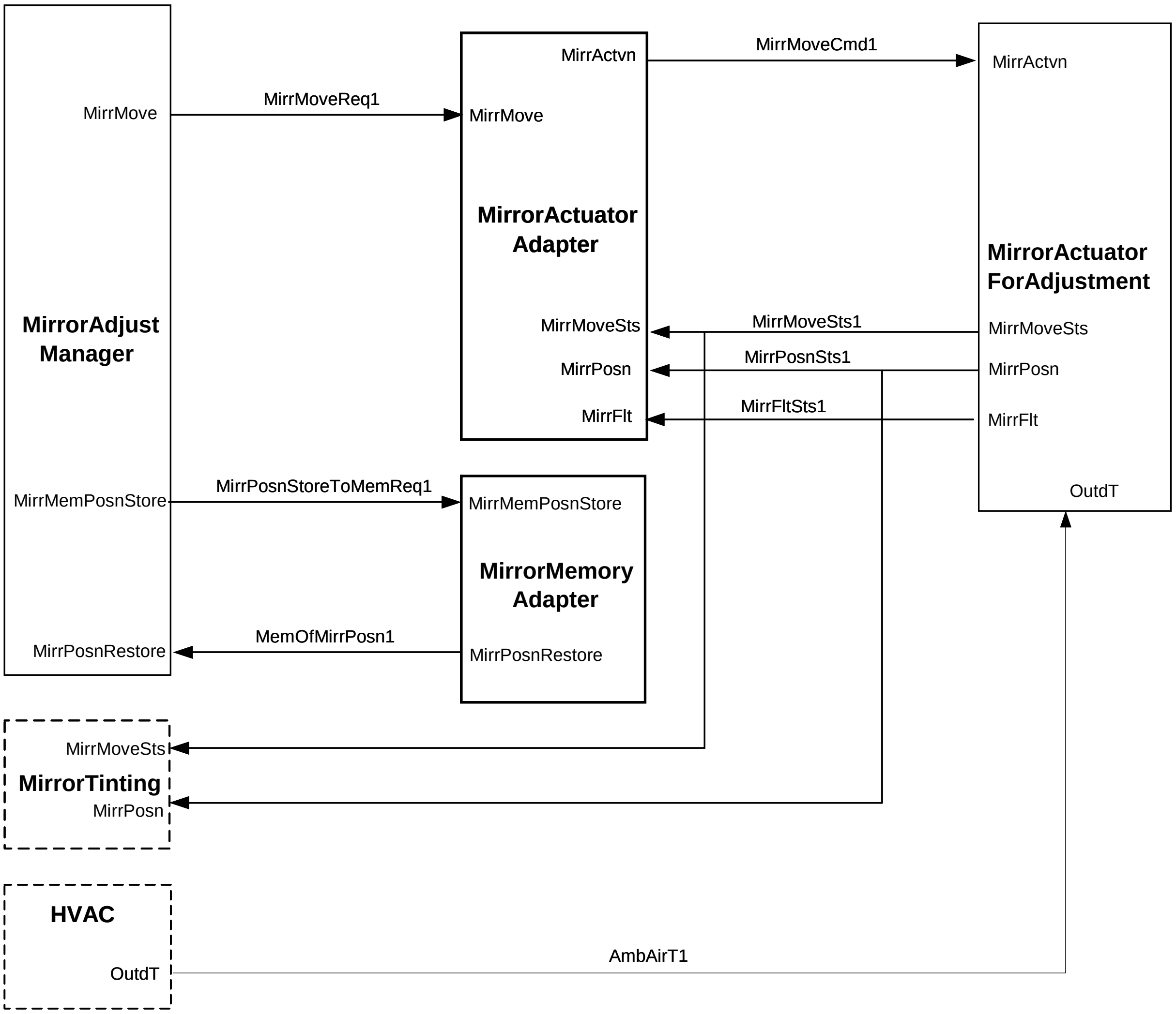
The mirror adjustment and tinting controls the manual and automatic adjustment of the exterior or interior mirrors of the car, respectively, the tinting level of the mirrors based on the outside brightness intensity.
The functionality has two independent decompositions, one for adjustment and one for tinting. The link between them is the mirror positions and it's movement status.
The functionality needs information from other body domain SW-Cs and also interdomain information.

4.2.1 Mirror Adjustment

Components related to MirrorAdjustment are bundled in several Blocks.



Mirror Adjustment & Tinting fig1



Mirror Adjustment & Tinting fig2

Inside MirrorAdjustManager the core functionality is implemented.

HMI contains the Mirror Manual Adjustment, detecting the user’s demands for manual adjustment and the Mirror Auto Adjustment, the panel with the memory buttons. It detects the user’s demands to store and recall mirror positions.

The Profile manager uses information from the key transponder, keyless access and remote key to generate the current driver profile identification.

The Remote key indicates a user request from the remote key and this may trigger the Auto adjust user request adapter to perform a store or recall.

The Keyless access indicates a user request from keyless access and this may trigger the Auto adjust user request adapter to perform a store or recall.

Inter domain encapsulates SW-Cs that exchange cross-domain information such as selected gear position, operational mode or energy management. Through Inter Domain Interfaces inter-domain status information is exchanged.

Central locking status is received from the Central Locking component.

The Auto adjust user request adapter combines and prioritizes all the information from above and requests mirror movements and storing of memory positions.

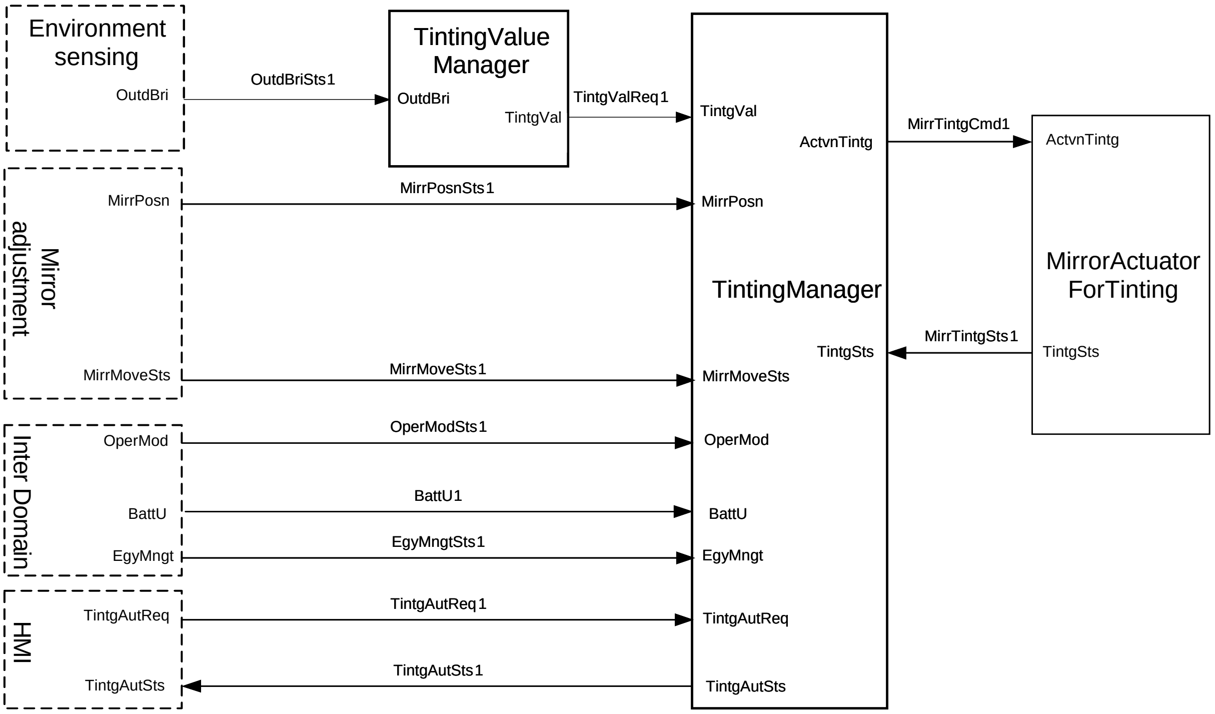
Mirror actuator adapter knows about the electrical and mechanical constraints of each single axis of the mirror. It converts requests from the Mirror adjust manager to commands to each single axis (actuators). It compares the current position of the mirror axis with the desired position.

Mirror memory adapter is responsible for the actual memory position storage and recall. This may be local or global storage depending on the personalization control strategy. External temperature information is used to compensate storage and re-storage of mirror positions.

Mirror Actuator For Adjustment is the software representation of a mechanical actuator. It is able to move one axis of the mirror to a certain direction. It also detects the position of the axis.

4.2.2 Mirror Tinting

Components related to Mirror tinting are bundled in several Blocks.



Mirror Adjustment & Tinting fig3

Inside Tinting Manager the core functionality is implemented.

Based on the information of the Environment Sensing Component, the Tinting Value Manager Component calculates the desired tinting value using the outside brightness information.

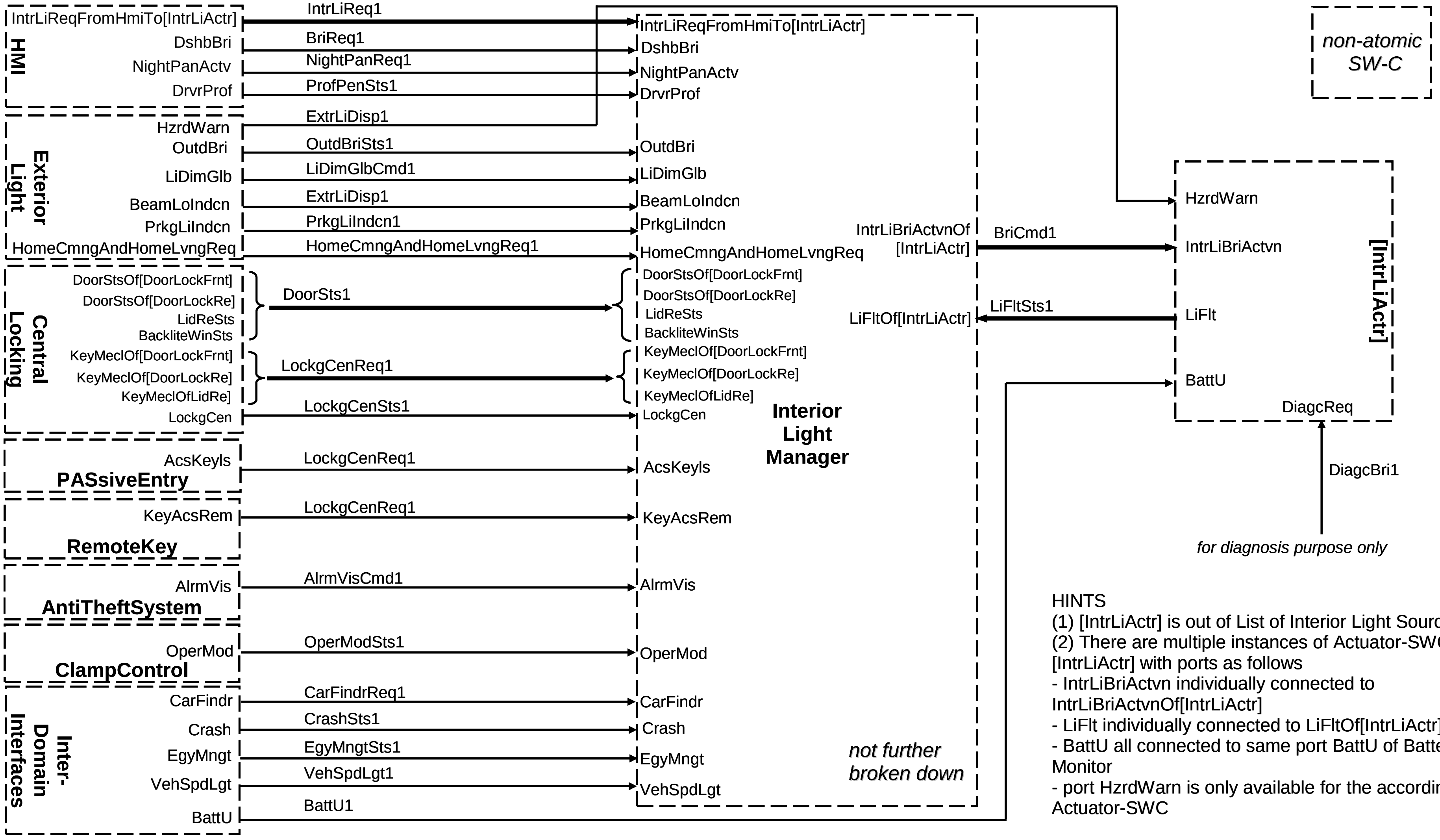
The Inter domain, HMI and Mirror Adjustment Component provide information about the necessity of automatic tinting.

The Mirror Tinting Actuator executes the demands of the Tinting Manager and delivers TintingStatus information back to the Manager.

4.3 Interior Light

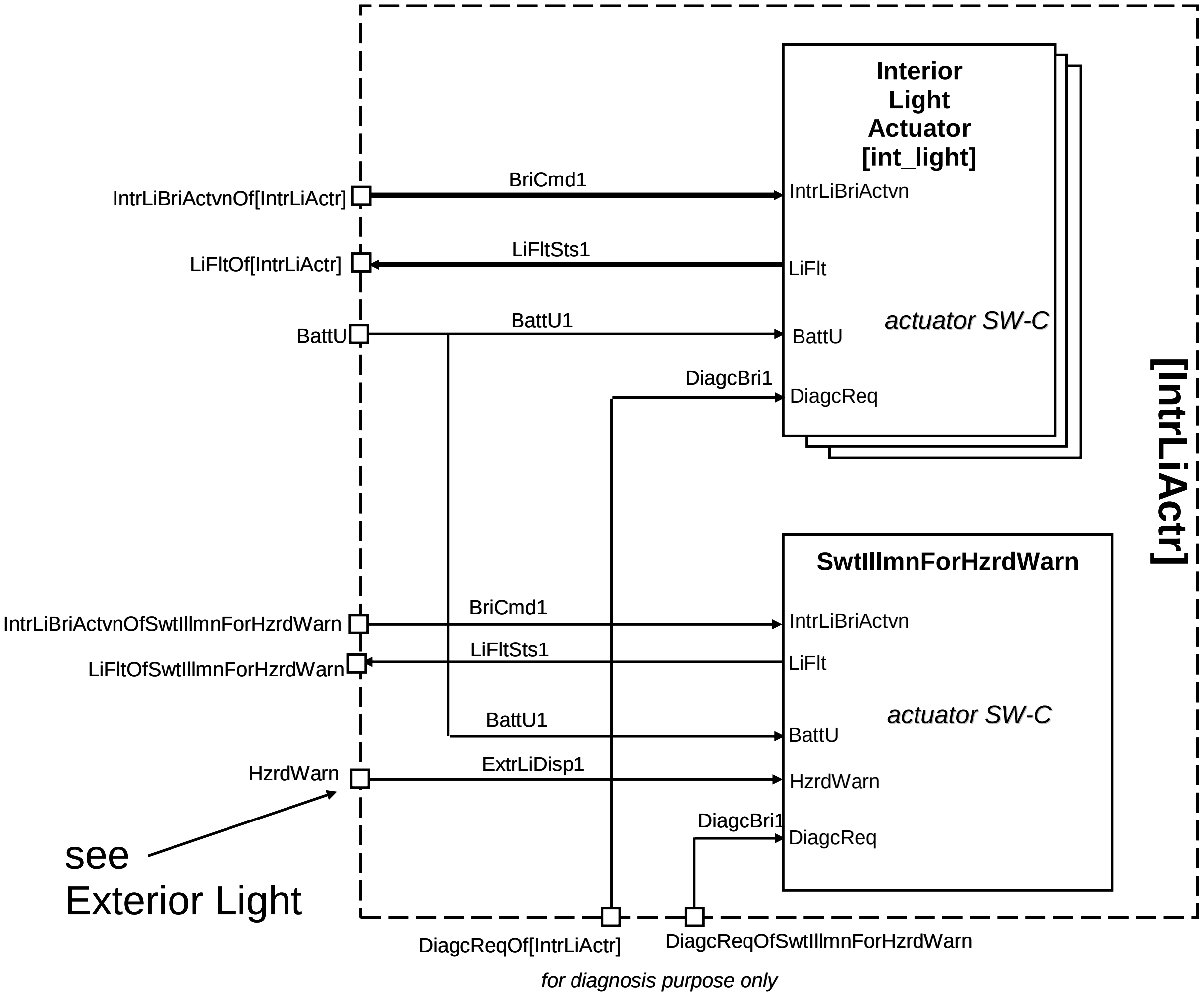
The `InteriorLight` software component group controls the interior light functionality of a car. `InteriorLight` receives driver wishes, senses the environment and controls the behavior of the interior light sources of a car.

Components related to InteriorLight are bundled in several Blocks.



HINTS
(1) [IntrLiActr] is out of List of Interior Light Sources
(2) There are multiple instances of Actuator-SWC [IntrLiActr] with ports as follows
- IntrLiBriActvn individually connected to IntrLiBriActvnOf[IntrLiActr]
- LiFlt individually connected to LiFltOf[IntrLiActr]
- BattU all connected to same port BattU of Battery Monitor
- port HzrdWarn is only available for the according Actuator-SWC

Interior Light fig.1



Interior Light fig.2

Fig. 1 and Fig. 2 shows the decomposition of Interior Light architecture. Arrows indicate data flows.
Remark: This decomposition is used for explanation of standardized interfaces

4.3.1 InteriorLightManager

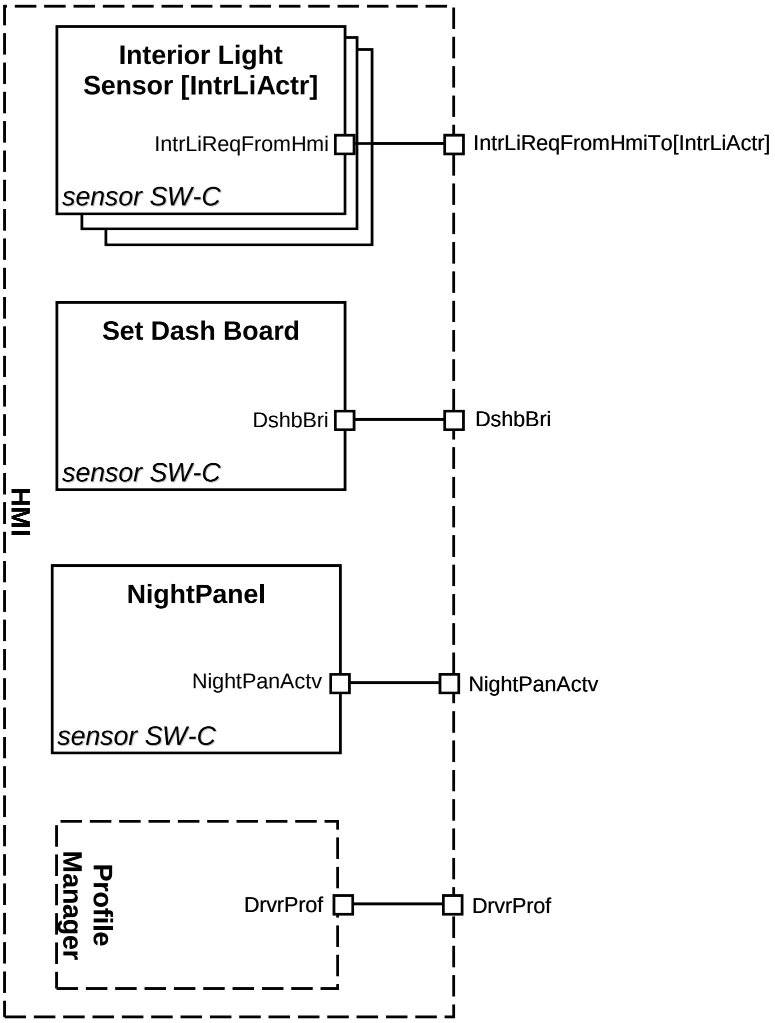
Inside InteriorLightManager the core functionality is implemented. Comments ‘not further broken down’ and “Non atomic SW-C” mean that this standardization does not describe it in a more precise level. During the implementation, this SW-C might be further decomposed

4.3.2 Body Sensors for Interior Light (HMI)

`HMI` contains all sensor components that detect the drivers demands related to interior light. Please note that “HMI” is used only for non-normative description.

For further decomposition see InteriorLight Fig3, whereas usually several SW-C InteriorLightSensor[IntrLiActr] exist. For Doorlocks please refer to CentralLocking Fig 2.

The purpose of the Sensor SW-C (HMI) is to ensure (as far as needed):
- filtering (debouncing) e.g. calculation of stable output values e.g. by debouncing
- conditioning e.g. calculation of average value
- plausibility check as basis for failure detection (implausible)
- failure detection (but no error-handling/ error-compensation) e.g. valid/not valid



InteriorLight Fig.3

Fig 3 shows the further Decomposition of HMI.
Note: The HMI SW-C listed in Fig 3 are just a proposition to help the understanding.

4.3.3 CentralLocking (including Doorlocks) / Remote Key / PASE

Out of `Doorlocks` the information is generated whether the corresponding door is open or closed as well as information on mechanical door lock(s). For further details see also separate chapter Central Locking.

Driver requests to lock or unlock the car might also be provided by `Remote Key`. Interior light uses only exterior requests (i.e. issued outside of the car) and not internal requests (such as emergency locking requests).
Also there is information received from `PASE` in order to be able to switch on (off) interior lights when a person with PaseIdCard is near the car (leaves the car).

Note, that interior light may also use the actual door locking status for certain use cases (auto-relocking event starts interior light activity). Reflecting the lock state to the driver is not an intended feature of this standardized interior light.

4.3.4 Exterior Light

Other blocks like `ExteriorLight` provide some information as well as Environment Sensing (e.g. Light Sensor) which provides information about the outside brightness.

For further details on the above mentioned functionalities see also separate chapter Exterior Light.

4.3.5 Anti-Theft-System

One interfaces is provided also by this body and comfort functionalitiy. For further details on Anti-Theft-System see according chapter.

4.3.6 ProfileManager

It is supposed only a single source coordinates profile Ids from certain personalization sources (remote key, keyless etc.). The `ProfileManager` is NOT in scope of this document.

4.3.7 TerminalClampControl

The status of ignition key is provided by Terminal`ClampControl`.

4.3.8 Inter-Domain Interfaces

Through `Inter-Domain-Interfaces` inter-domain status information is exchanged.

4.3.9 Battery Monitor (Future function)

Battery Monitor (futur function) provides information about Battery Voltage.

4.3.10 Actuators

[IntrLiActr] (including HazardWarningSwitchIllumination): All interior light actuators are described by the LightSource[IntrLiActr]components where [IntrLiActr] is a placeholder for individual light sources (e.g. CbnLiFrntCen as Cabinlight “Front-Middle”) or a group of light sources (e.g. LiForReadingGroup). See next chapter for a list of possible effective names.

It might be decomposed into a light actuator type independent and a light actuator type dependent component, where light actuator types include conventional bulbs, LEDs or other decompositions, which allow the switching of lights on a more “generic” level.

The actuator also covers low level functionalities (e.g. measures to prevent flickering due to voltage variations) as well as over-/under voltage compensation. Also hardware dependencies are considered (e.g. is PWM possible for corresponding actuator). Status feedback is generated (only operational/defective; no detailed diagnosis). Actuator SW-Cs do not consider system restrictions (e.g. restrictions due to mechanical reasons regarding activation of several actuators). This should be task of a separate Actuator-Adapter SW-C.

Restrictions to Interior Light Actuator: Actuator for IntrLiGroup0..15 is allowed only if all light sources within the group are
1) not part of any other light group
2) not directly used by another light actuator
Same applies for groups like CbnLiGroup, TrCmptmtGroup, FootwellGroup etc.

HINT: Please note that HazardWarningSwitchIllumination is also an InteriorLightActuator. There is just one additional port HzrdWarn fed by Exterior Light.

4.3.11 List of interior light sources – groups of light sources

The list below defines individual light sources (like CbnLiFrntRi) as well as functional groups (e.g. CbnLiGroup) as well as virtual groups (see IntrLiGroup0..15):

CabinLights: CbnLiFrntRi (front_right), CbnLiFrntLe (front_left), CbnLiFrntCen (front_center e.g. in the middle), CbnLiMidRi (middle_right), CbnLiMidLe (middle_left), CbnLiMidCen (middle_center), CbnLiReRi (rear_right), CbnLiReLe (rear_left), CbnLiReCen (rear_center)

TrunkCompartmentLights: TrCmptmtLiCen (light_middle), TrCmptmtLiLoRi (lower_right), TrCmptmtLiLoLe (lower_left), TrCmptmtLiUpRi (upper_right), TrCmptmtLiUpLe (upper_left)

FootwellLights: FootwellLiFrntLe (front_left), FootwellLiFrntCen (front_middle), FootwellLiFrntRi (front_right), FootwellLiMidLe (middle_left), FootwellLiMidCen (middle_middle), FootwellLiMidRi (middle_right), FootwellLiReLe (rear_left), FootwellLiReCen (rear_middle), FootwellLiReRi (rear_right)

AshtrayLights: AshtrayLiFrntLe (front_left), AshtrayLiFrntCen (front_middle), AshtrayLiFrntRi (front_right), AshtrayLiMidLe (middle_left), AshtrayLiMidCen (middle_middle), AshtrayLiMidRi (middle_right), AshtrayLiReLe (rear_left), AshtrayLiReCen (rear_middle), AshtrayLiReRi (rear_right)

ReadingLights: ReadingLiFrntLe (front_left), ReadingLiFrntCen (front_middle), ReadingLiFrntRi (front_right), ReadingLiReLe (rear_left), ReadingLiReCen (rear_middle), ReadingLiReRi (rear_right), ReadingLiThirdRowLe (ThirdRow_left), ReadingLThirdRowCen (ThirdRow_middle), ReadingLiThirdRowRi (ThirdRow_right)

Note: Front means first row, Rear means second row and at the end is the third row.

CurbLights: CurbLiFrntLe (front_left), CurbLiFrntRi (front_right), CurbLiMidLe (middle_left), CurbLiMidRi (middle_right), CurbLiReLe (rear_left), CurbLiReRi (rear_right)

VanityLights: VanityLiFrntLe (front_left), VanityLiFrntRi (front_right), VanityLiMidLe (middle_left), VanityLiMidRi (middle_right), VanityLiReLe (rear_left), VanityLiReRi (rear_right)

CigaretteLighterIlluminationLights: LiForCigaretteLighterIllmnFrntLe (front_left), LiForCigaretteLighterIllmnFrntCen (front_middle), LiForCigaretteLighterIllmnFrntRi (front_right), LiForCigaretteLighterIllmnMidLe (middle_left), LiForCigaretteLighterIllmnMidCen (middle_middle), LiForCigaretteLighterIllmnMidRi (middle_right), LiForCigaretteLighterIllmnReLe (rear_left), LiForCigaretteLighterIllmnReCen (rear_middle), LiForCigaretteLighterIllmnReRi (rear_right), LiForCigaretteLighterIllmnTrLe (trunk_left), LiForCigaretteLighterIllmnTrRi (trunk_right)

DoorSillIlluminationLights: LiForDoorSillIllmnFrntLe (front_left), LiForDoorSillIllmnFrntRi (front_right), LiForDoorSillIllmnMidLe (middle_left), LiForDoorSillIllmnMidRi (middle_right), LiForDoorSillIllmnReLe (rear_left), LiForDoorSillIllmnReRi (rear_right)

DoorOpenerIlluminationLights: LiForDoorOpenerIllmnFrntLe (front_left), LiForDoorOpenerIllmnFrntRi (front_right), LiForDoorOpenerIllmnMidLe (middle_left), LiForDoorOpenerIllmnMidRi (middle_right), LiForDoorOpenerIllmnReLe (rear_left), LiForDoorOpenerIllmnReRi (rear_right)

SafetyBeltLockIlluminationLights: LiForSftyBltLockIllmnFrntLe (front_left), LiForSftyBltLockIllmnFrntCen (front_middle), LiForSftyBltLockIllmnFrntRi (front_right), LiForSftyBltLockIllmnMidLe (middle_left), LiForSftyBltLockIllmnMidCen (middle_middle), LiForSftyBltLockIllmnMidRi (middle_right), LiForSftyBltLockIllmnReLe (rear_left), LiForSftyBltLockIllmnReCen (rear_middle), LiForSftyBltLockIllmnReRi (rear_right)

Others: GlvCmptmtLi (GloveCompartmentLight), EngCmptmtLi (EngineCompartmentLight), IgnSwtIllmn (IgnitionSwitchIllumination), SwtIllmnForHzrdWarn (HazardWarningSwitchIllumination), ExtrLiSwtIllmn (ExteriorLightSwitchIllumination)

Functional groups defined are: CbnLiGroup, TrCmptmtGroup, FootwellGroup, AshtrayLiGroup, LiForReadingGroup, CurbLiGroup, VanityLiGroup, LiForCigaretteLighterIllmnGroup, LiForDoorSillIllmnGroup, LiForDoorOpenerIllmnGroup, LiForSftyBltLockIllmnGroup

These groups of interior light interfaces are defined (virtual interior light interfaces): IntrLiGroup0, IntrLiGroup1, … IntrLiGroup15

Each group 0..15 can consist out of any number of above mentioned interior light sources. This is configurable per implementation.

As a further example for the application of an “IntrLiGroup” there is the „Battery-Saver Feature“ for deactivation of inadvertent loads of interior light (example glove box: lamp(s) might be hardwired activated by a switch. If left open unintentionally a separate actuator could be provided for deactivation of all these hardwired lamps after a certain timeout after ignition off).

The groups of light sources mentioned above (e.g. IntrLiGroup0..15) may be configured by means of “Configuration-Bitfields” (at implementation level) whereas each position corresponds to a certain interior light source. This means that this information might be stored in NVRAM and accessible through calibration and/or diagnostic tools. Also possible is providing this information as static input to Interior Light Manager.

4.3.12 Scope definition / possible future enhancements

The intension of this chapter is to clarify what’s considered in this standardization document. Also known deficiencies and/or possible future enhancements can be listed here.

4.3.12.1 Signal qualifiers:

There is only inband invalidation used (see according definitions in AI). Please note that safe communication frame (e.g. measures related to safe transmission of data) are considered as implementation related and not scope of this document.

HINT: “inband invalidation” means that there is a special value “invalid” used additional to the “regular” enum-values (where invalidation is necessary). This is in order to minimize the bitsize of the according interfaces. Contrary to that “outband invalidation” would be useless for the according interfaces because a data element is supposed to be either invalid or to have a “valid” value.

4.3.12.2 Timing:

Scope of this document is VFB level. Thus no timing requirements neither on vehicle function level nor system / implementation level are under consideration.

4.3.12.3 Variant handling:

Scope of this document is VFB level. Variant handling is regarded as project specific (e.g. implementation specific).

4.4 Seat Adjustment

The `SeatAdjustment` software component group controls the seat adjustment functionality of a car. `SeatAdjustment` receives driver wishes, senses the environment and controls the behavior of the `SeatAdjustment` actuators in a car.

4.4.1 Seat Configurations

The following table lists the allowed seat names that are defined for use in AUTOSAR systems. The table also shows the name text that should be used for constructing port and interface references in place of the place holder “[Seat]”.

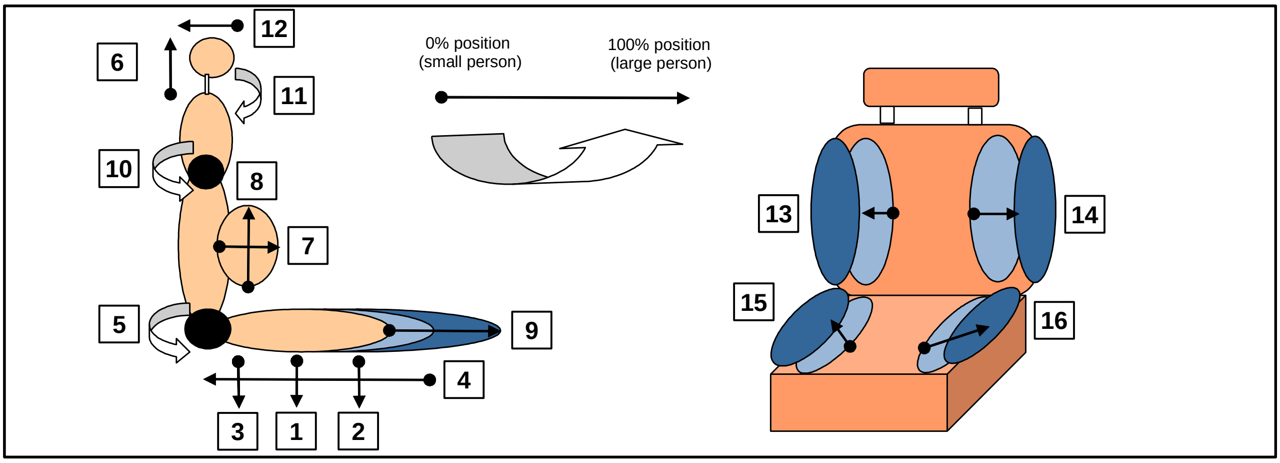
Seat Name	***Reference Text***
Driver Seat	DrvrSeat
Front Passenger Seat	PassSeat
Front Central Seat	SeatRowFirstMid
Left Middle Seat	SeatRowSecLe
Centre Middle Seat	SeatRowSecMid
Right Middle Seat	SeatRowSecRi
Left Rear Seat	SeatRowThrdLe
Centre Rear Seat	SeatRowThrdMid
Right Rear Seat	SeatRowThrdRi

The following table lists the allowed axis names that are defined for use in AUTOSAR systems. The table also shows the name text that should be used for constructing port and interface references in place of the place holder “[SeatAxis]”.

Diagram ID	***Axis Name***	***Reference Text***
1	Seat Height	AxisActrHei
2	Seat Front Height	AxisActrHeiFrnt
3	Seat Rear Height	AxisActrHeiRe
4	Seat Slide	AxisActrSld
5	Seat Back Inclination	AxisActrIncl
6	Headrest Height	AxisActrHdrestHei
7	Lumbar Extend	AxisActrLumExtn
8	Lumbar Height	AxisActrLumHei
9	Seat Cushion Extend	AxisActrCushExtn
10	Seat Back Top Inclination	AxisActrTopIncl
11	Headrest Fold	AxisActrHdrestFold
12	Headrest Tilt	AxisActrHdrestTilt

13	Seat Back Right Bolster	AxisActrBackBlstrRi
14	Seat Back Left Bolster	AxisActrBackBlstrLe
15	Seat Cushion Right Bolster	AxisActrCushBlstrRi
16	Seat Cushion Left Bolster	AxisActrCushBlstrLe

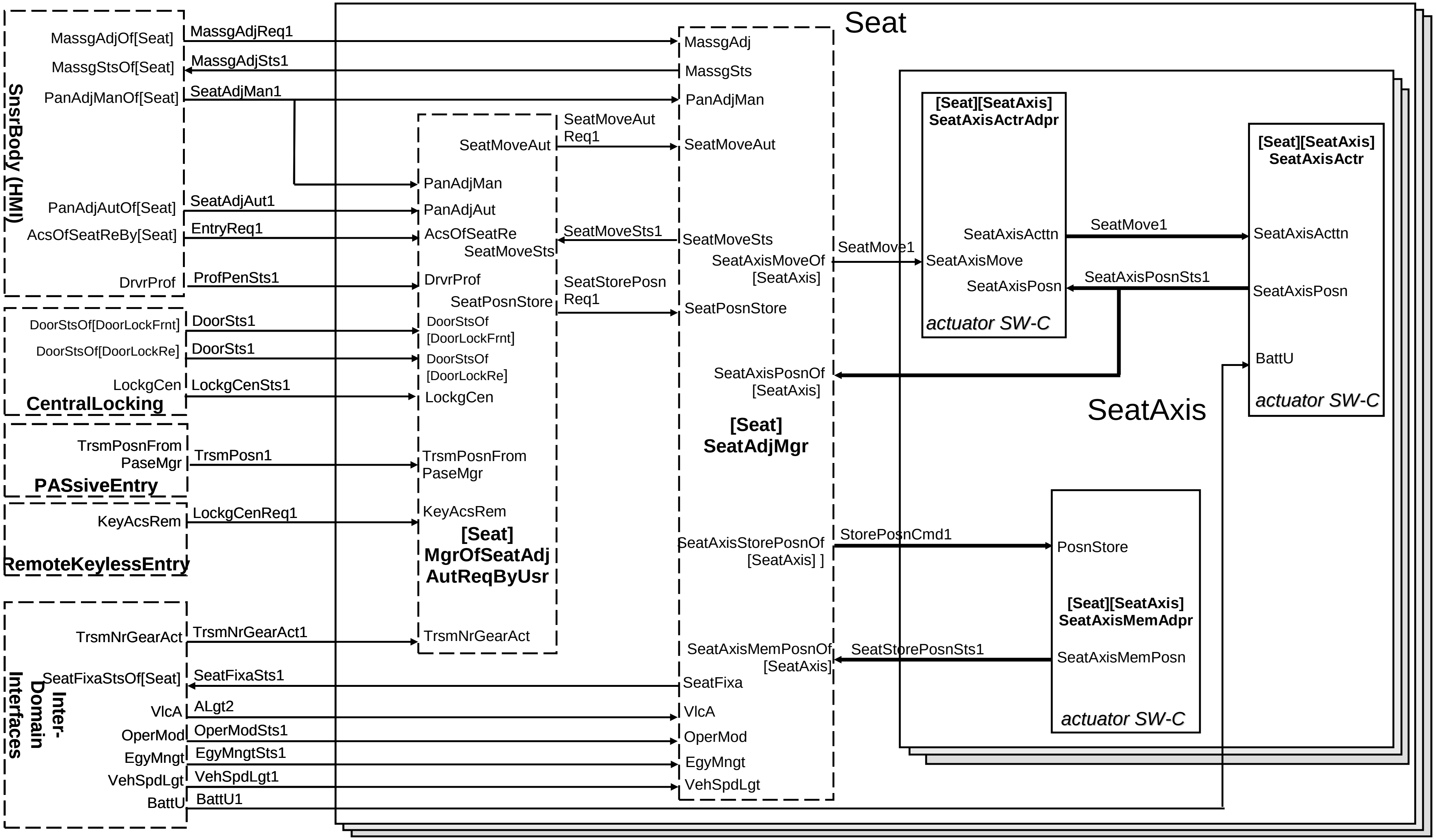
The axis are shown on the diagram below to show the sense of the position data.



SeatAdjustment Fig1

4.4.2 Seat Adjustment Decomposition

The Seat Adjustment sub-system is shown in Fig 2.
Software Components related to `SeatAdjustment` are described in the following text.



SeatAdjustment Fig.2

Fig 2 shows the SW-Cs mentioned above and the software components contained. Arrows indicate data flows.

The usage of placeholder in brackets […] for some port names, means that for these ports there can be multiple instantiation; in particular:
- [DoorLockFrnt] can be instantiated to Driver door and Passenger door;
- [DoorLockRe] can be instantiated to Rear Left and Right doors.

Please note:

(1) individual signals per seat - e.g. see interface SeatAdjMan1:
SW-C MmedTelmHmi (HMI) has one port per seat: PanAdjManOfDrvrSeat, PanAdjManOfPassSeat, ...
each SW-C [Seat]SeatAdjMgr has one port PanAdjMan
e.g. MmedTelmHmi/PanAdjManOfDrvrSeat -> DrvrSeat/SeatAdjMgr/PanAdjMan,
MmedTelmHmi/PanAdjManOfPassSeat -> PassSeat/SeatAdjMgr/PanAdjMan, ...

(2) individual signals per seat - e.g. see interface SeatMoveSts1:
each SW-C [Seat]SeatAdjMgr has one port SeatMoveSts
each SW-C [Seat]MgrOfSeatAdjAutReqByUsr has one port SeatMoveSts
ATTENTION: There is an one-to-one connection between SW-C's [Seat]SeatAdjMgr and [Seat]MgrOfSeatAdjAutReqByUsr of EACH SEAT
e.g. DrvrSeat/SeatAdjMgr/SeatMoveSts -> DrvrSeat/MgrOfDrvSeatAdjAutReqByUsr /SeatMoveSts,
PassSeat/SeatAdjMgr/SeatMoveSts -> PassSeat/MgrOfDrvSeatAdjAutReqByUsr/ SeatMoveSts, ...

(3) individual signals per seat & axis - e.g. see interface SeatAxisPosnSts1:
Taking into consideration DrvSeat SW-C, consider for example the following possible nested SWCs:
- DrvSeat/AxisActrHei/SeatAxisActr (example of Actuator)
- DrvSeat/AxisActrSld/SeatAxisActr (example of Actuator)
- DrvSeat/AxisActrIncl/SeatAxisActr (example of Actuator)
- …
and
- DrvSeat/AxisActrHei/SeatAxisActrAdpr (example of ActuatorAdapter)
- DrvSeat/AxisActrSld/SeatAxisActrAdpr (example of ActuatorAdapter)
- DrvSeat/AxisActrIncl/SeatAxisActrAdpr (example of ActuatorAdapter)
- …

each one of these SWC instances (both Actuator and ActuatorAdapter) has one port called SeatAxisPosn
The same situation applies for other SWCs related to different seats like for example, related to PassSeat SW-C:
PassSeat/AxisActrHei/SeatAxisActr (example of Actuator)
and
PassSeat/AxisActrHei/SeatAxisActrAdpr (example of ActuatorAdapter)

On the other side the SW-C DrvSeat/SeatAdjMgr has one port per axis:

- *SeatAxisPosnOfAxisActrHei*,
- *SeatAxisPosnOfAxisActrSld*,
- *SeatAxisPosnOfAxisActrIncl*,
- ...

Of course the same situation applies also for other SWCs related to different seats like for example PassSeat/SeatAdjMgr.

Collecting these indications, the correspondence between ports is, for example:
DrvSeat/AxisActrHei/SeatAxisActr/*SeatAxisPosn* ->
DrvSeat/AxisActrAdprHei/SeatAxisActrAdpr/*SeatAxisPosn* ->
DrvSeat/SeatAdjMgr/*SeatAxisPosnOfAxisActrHei*

4.4.2.1 Body Sensors for Seat Adjustement (HMI)

HMI contains all sensor components that detect the drivers demands related to seat adjustement. Please note that “HMI” is used only for non-normative description.

MassgAdjReq1 enables the user to choose one or more massage programs from a massage control panel (e.g. a switchpack or touch screen menu) and inform the [Seat]/SeatAdjustManager. There may be specific “massage actuators” in addition to the defined seat axis. These actuators are not named here.

MassgAdjSts1 provides status of the active massage functions back to the HMI to inform the user of the current mode of operation.

SeatAdjMan1 enables the user to activate the seat actuator using a manual movement control panel (e.g. a switchpack or touch screen menu) and inform the [Seat]/SeatAdjustManager. This information additionally goes to the [Seat]/ManagerOfSeatAdjustAutoRequestByUser so that manual operations can be used to over-ride memory adjustment operations.

SeatAdjAut1 indicates the user’s demands to store and recall seat positions.

EntryReq1 detects the users demand for access to the rear seats. This may result in a front seat moving forward. This feature may also be known as auto glide. Similarly, easy front entry requests are possible.

4.4.2.2 Doors, KeylessAccess and RemoteKey

The Central Locking component provides information about operation of vehicle accesses (doors and locks) that might indicate a need to prepare seats for occupation. The ProfileManager uses information from the key transponder, keyless access and remote key to generate the current driver profile identification ProfPenSts1.

If the trigger_source (that is an input from the user to do somenthing) inside the interface LockgCenSts1 indicates a user request from the remote key or keyless entry, this may trigger the [Seat]/ManagerOfSeatAdjustAutoRequestByUser to perform a store or recall.

The CentralLocking indicates a user request from central locking system and this may trigger the [Seat]/ManagerOfSeatAdjustAutoRequestByUser to perform a store or recall.

The DoorSts1 indicates whether the door associated with the seat has been opened or closed and this can be used to trigger seat movements or to prevent movement (Note: for R4.0.3 DoorStsOfDrvrDoorLock, DoorStsOfPassDoorLock, DoorStsOfDoorLockReLe and DoorStsOfDoorLockReRi are received for each seat also if in a real SWC implementation only the related DoorSts is really used).

4.4.2.3 Inter-DomainInterfaces

Inter-DomainInterfaces encapsulates interfaces to SW-Cs that exchange cross-domain information such as vehicle speed, vehicle longitudinal and lateral accelerations, operational modes or energy management status. Inter-domain status information is exchanged through Inter-Domain-Interfaces.

4.4.2.4 AutoAdjustUserRequestManager

There is a [Seat]/ManagerOfSeatAdjustAutoRequestByUser associated with each seat that requires seat adjustment.
The [Seat]/ManagerOfSeatAdjustAutoRequestByUser combines and prioritizes all the information from above and requests seat movements and storing of memory positions.

4.4.2.5 SeatAxisActuatorAdapter

There is a [Seat]/[SeatAxis]/SeatAxisActuatorAdapter associated with each axis in every seat that requires seat adjustment control. [Seat]/[SeatAxis]/SeatAxisActuatorAdapter knows about the electrical and mechanical constraints of a single axis of the seat. It converts requests from the [Seat]SeatAdjustManager into commands to a single axis (actuator). It compares the current position of the seat axis with the desired position.

4.4.2.6 SeatAxisActuator

There is a [Seat]/[SeatAxis]/SeatAxisActuator associated with each axis in every seat that requires seat adjustment control.
[Seat]/[SeatAxis]/SeatAxisActuator is the software representation of a mechanical actuator. It is able to move one axis of the seat in a certain direction. It also detects the position and speed of the axis.

4.4.2.7 SeatAxisMemoryAdapter

There is a `[Seat]/[SeatAxis]/SeatAxisMemoryAdapter` associated with each axis in every seat that requires seat adjustment control.
`[Seat]/[SeatAxis]/SeatAxisMemoryAdapter` is responsible for the actual memory position storage and recall. This may be local or global storage depending on the personalization control strategy.

4.4.2.8 SeatAdjustManager

Finally, the `[Seat]/SeatAdjustManager` is the core of the seat adjustment implementation. There is a `[Seat]/SeatAdjustManager` associated with each seat that requires seat adjustment. This component coordinates all the various requests and triggers into a specific control strategy for a seat. It also ensures that seat adjustment is carried out safely and in accordance with feature functional priorities and behaviors.

4.4.3 Personalization Issues

It is expected that seat adjustment can be personalized.

Known use cases for personalization of the seat adjustment functionality are, for example:

- Store the preferred seat position of the driver;
- Enable/disable certain seat adjustment functionality (e.g. easy entry).

4.4.4 Design Rationale

This design standard is the AUTOSAR-architecture for seat adjustment. It provides a decomposition into SW-Cs and a list of standardized interfaces related to seat adjustment.

The decomposition is limited in granularity to sensor components, adapter components, the core functionality and actuator components. It is not intended to focus only on event driven implementations and the standardization should be also open for “cyclic implementations” sending out information periodically (therefore some interfaces contain an “idle” value by intent).

Additionally, all ports are described together with AUTOSAR agreed data qualities. Invalidation (where needed) is defined as in-band invalidation. Init values are specified where appropriate e.g. “off”, “idle”, “undefined”, “unknown” … Where standardization is in-appropriate, recommendations may be provided.

The decomposition in this state is good enough to gather a common understanding of interfaces for standardization. A finer granularity would not provide greater ease of standardization, as noted during exterior light standardization.

By intent the decomposition does not get to atomic SW-Cs but to “purchasable” SW-Cs (although sensor/actuator SW-Cs are realized as AUTOSAR compositions for formal reasons). These SW-Cs will be obtained as a unit so that all the internal (and hence not AUTOSAR standardized) interfaces are controlled by a single vendor, even if there are SW-Cs within the bought unit that reside on different ECUs. All interfaces between SW-Cs from different vendors should be standardized.

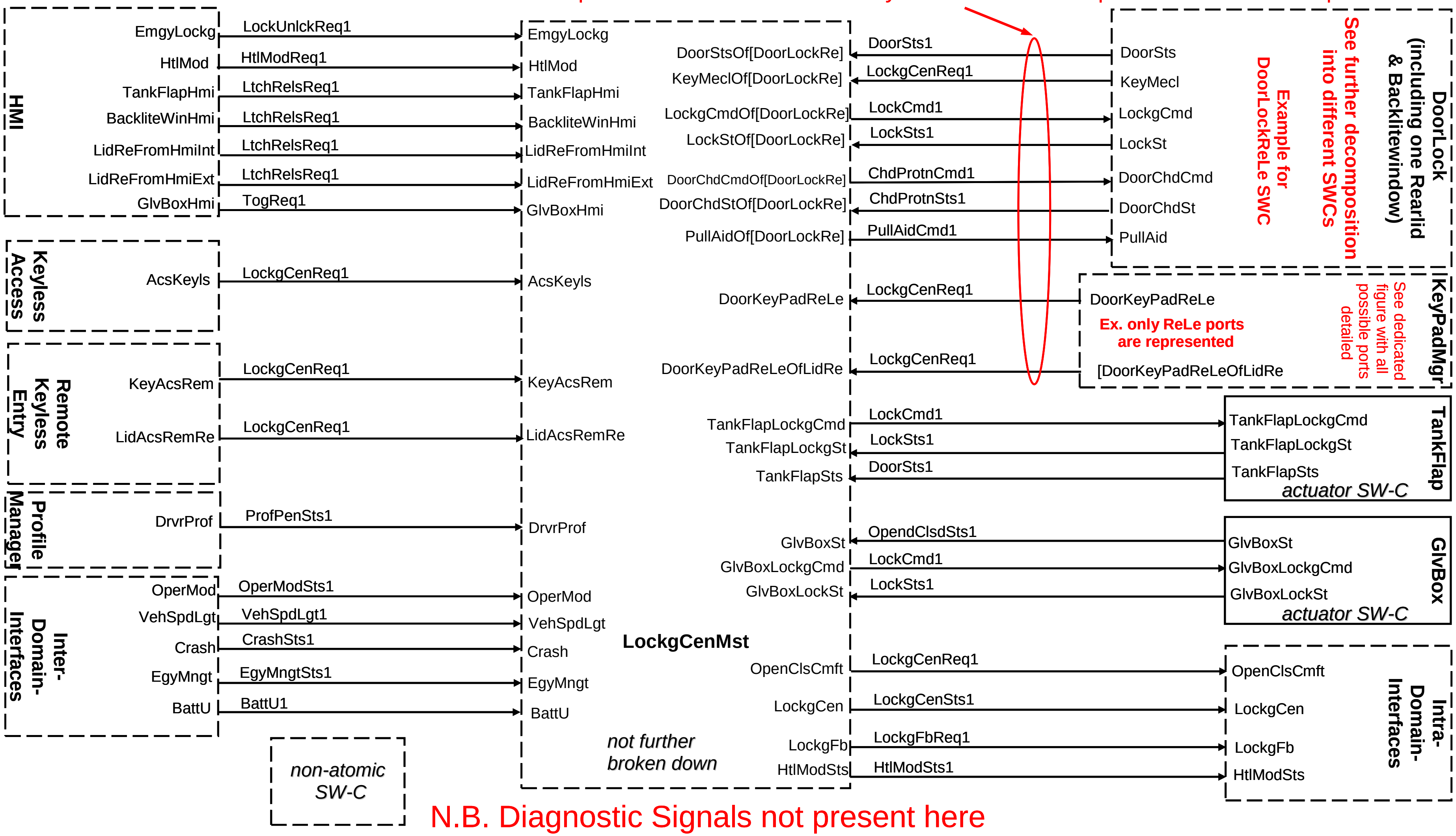
4.5 Central Locking

The `CentralLocking` software component group controls the central locking functionality of a car. `CentralLocking` receives driver wishes, senses the environment and controls the behavior of the central locking sources of a car.

4.5.1 Central Locking Decomposition main view

Components related to `CentralLocking` are bundled in several blocks:

N.B. Ports and interfaces with multiple instantiation have here only one instantiation represented as an example



N.B. Diagnostic Signals not present here

CentralLocking Fig 1

Notes related to CL decomposition:
The figure above gives only the main view of the decomposition. For better readability it has been chosen to provide in separate paragraphs the further decomposition of **DoorLock** SWCs and **KeyPadMgr** SWC detail of all the possible ports .

So, for a complete decomposition overview of this functionality, please refer also to the further decomposition of **DoorLock** SWCs and **KeyPadMgr** SWC ports details provided in the following paragraphs.

Please note that diagnostic interfaces are not part of the graphics.

The usage of placeholder in brackets […] for some port names, means that for these ports there can be multiple instantiation; in particular:
- [DoorLockFrnt] can be instantiated to Driver door and Passenger door;
- [DoorLockRe] can be instantiated to Rear Left and Right doors.

Note: Rear Lid and BackLite Window have dedicated SWCs with single instantiation.

For the detailed usage of these placeholders, please refer to the further decomposition of DoorLock SWC provided in the following paragraphs.

4.5.2 Explanation of SW-Cs

4.5.2.1 CentralLockingMaster

Inside CentralLockingMaster the core functionality is implemented (not further broken down).

4.5.2.2 HMI

HMI contains all other sensor components that detect the drivers demands related to central locking. Please note that “HMI” is used only as non-normative description.

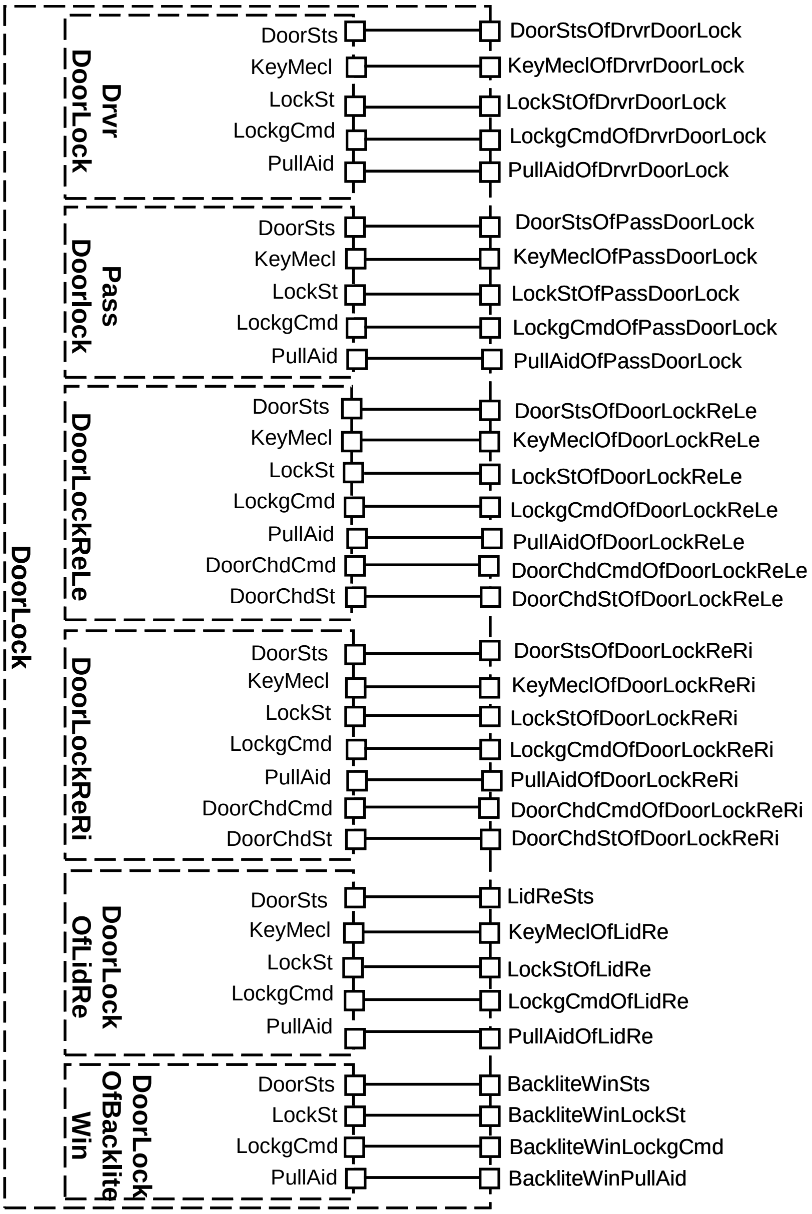
4.5.2.3 RemoteKey / Keyless Access

Driver requests to access the car might be provided by RemoteKey or Keyless Access System (including profile information for personalization).

4.5.2.4 DoorLock

Out of the doorlocks the information of the doorlock status is generated as well as information on mechanical keys.

Lock-commands as well as commands for pull aids and setting the child-protection-mechanism are also sent to the doors. Same applies for RearLid and BackliteWindow. See also “mechanical decomposition door” and “mechanical events vs. standardized interfaces” (see below).



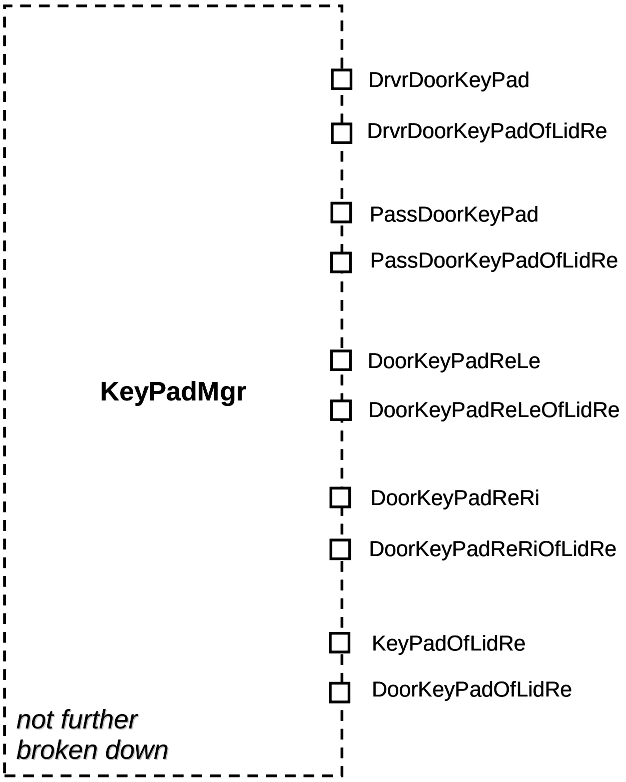
CentralLocking Fig 2

The above figure shows the further decomposition of DoorLock SWC.

4.5.2.5 KeyPad Manager

Out of the keypad the request to activate the CL-functionality for doors is generated as well as the request to separately activate rear lid.

KeyPad is a single SWC (that could be non-atomic) that provides dedicated ports for front doors, rear doors and rear lid (see figure below).



CentralLocking Fig 3

4.5.2.6 GloveBox / Tankflap

Other blocks like Glovebox and TankFlap provide status information and (partially) receive locking commands. There is no use case for engine-hood.

4.5.2.7 Intra-Domain-Interfaces

Through Intra-Domain-Interfaces intra-domain information is provided to other body and comfort systems (please note that scope of this section is only central locking).

Examples:
- Comfort open/close functionality is requested from windows as well as sunroof.
- Visual feedback is requested from Blink Master (see separate documentation for ExteriorLight).

4.5.2.8 Inter-Domain-Interfaces

Through Inter-Domain-Interfaces inter-domain status information is exchanged.

4.5.2.9 ProfileManager

It is supposed only a single source coordinates profile Ids from certain personalization sources (remote key, keyless etc.). The ProfileManager is NOT in scope of this section.

4.5.3 Further Issues concerning Central Locking

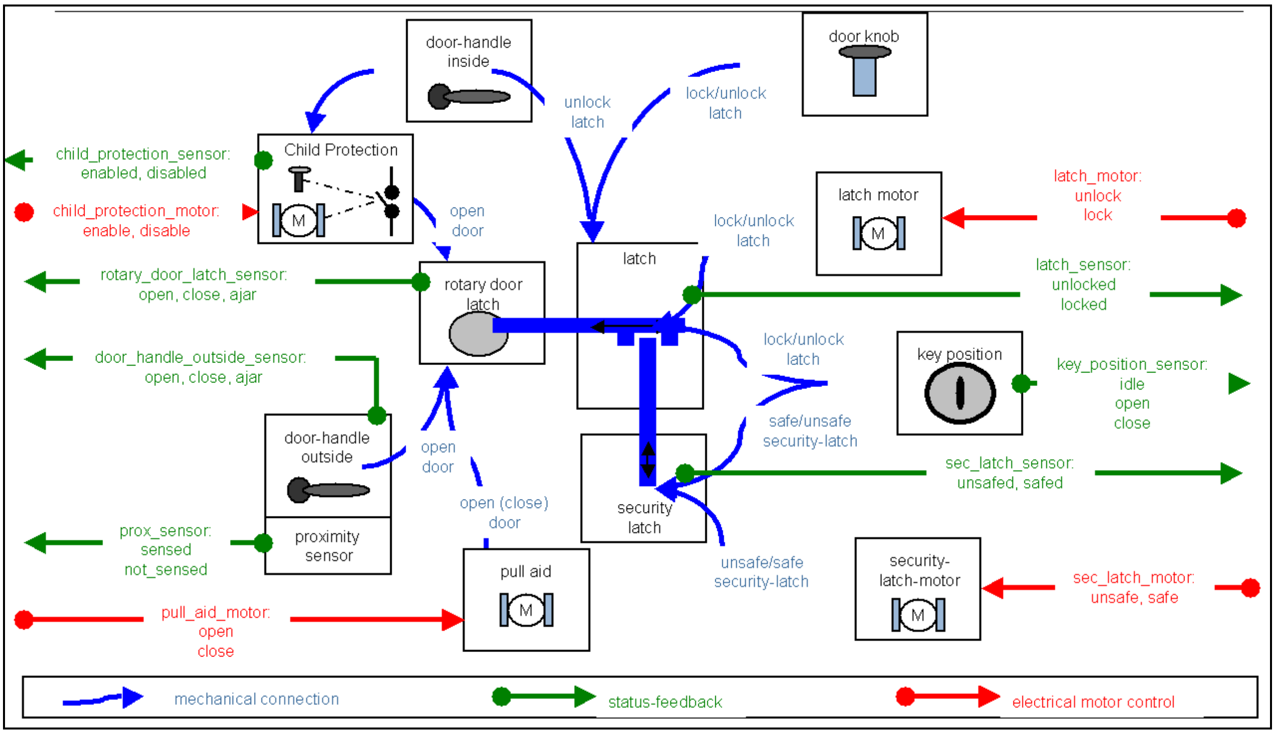
4.5.3.1 Personalization

It is supposed that the Central Locking Master is able to be personalizabled.

Known use cases for personalization of central locking functionality are for example:

- “Speedlock”: lock all doors at (or above) a certain vehicle speed.
- Selective door opening: open only driver door or open all doors.
- Enable/disable certain central locking functionality (e.g. visual feedback).

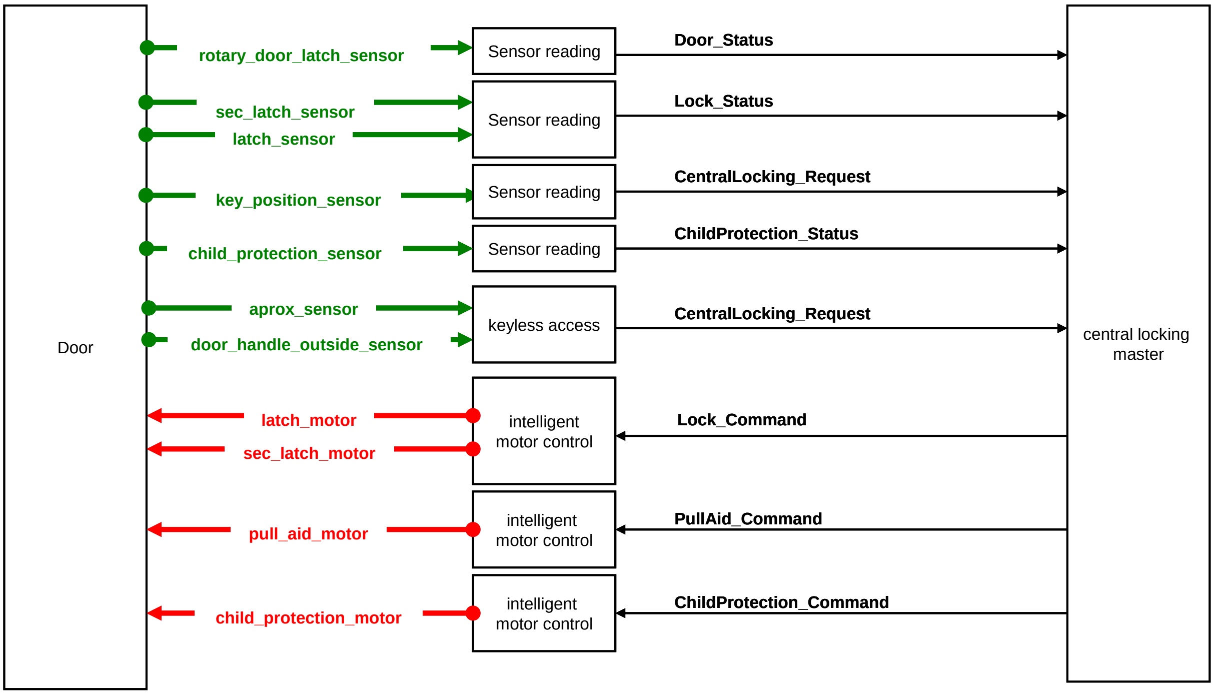
4.5.3.2 Mechanical decomposition door


door-handle inside
door knob
unlock latch
lock/unlock latch
child_protection_sensor: enabled, disabled
Child Protection
M
child_protection_motor: enable, disable
open door
lock/unlock latch
latch motor
M
latch_motor: unlock lock
latch
rotary_door_latch_sensor: open, close, ajar
rotary door latch
latch_sensor: unlocked locked
door_handle_outside_sensor: open, close, ajar
lock/unlock latch
key position
key_position_sensor: idle open close
safe/unsafe security-latch
door-handle outside
open door
open (close) door
security latch
sec_latch_sensor: unsafed, safed
prox_sensor: sensed not_sensed
proximity sensor
unsafe/safe security-latch
security-latch-motor
M
sec_latch_motor: unsafe, safe
pull aid
M
pull_aid_motor: open close
mechanical connection
status-feedback
electrical motor control

CentralLocking Fig 4

The figure above is intended for clarification of further mechanical relations.

4.5.3.3 Mechanical actions vs. SWCs interfaces (door only)



CentralLocking Fig 5

The figure shows the relation between mechanical actions at the door and related SWCs interfaces.
The purpose is just to give an idea of relationship between a mechanical action made by the user (on one door) and its “codification” into corresponding signals towards/from the CL Manager passing through the “translation” made by Sensors/Actuators.

Note: This figure does not have punctual correspondence with the detailed decomposition of the functionality already explained in previous paragraphs, so also the Naming Convention is not applied: the names used have only the goal of giving easy comprehension of the information.

4.5.4 Known Defects

Diagnostics: Diagnostics Ports are not part of this explanatory description. Nevertheless, diagnostic interfaces/ports are defined (see AI).

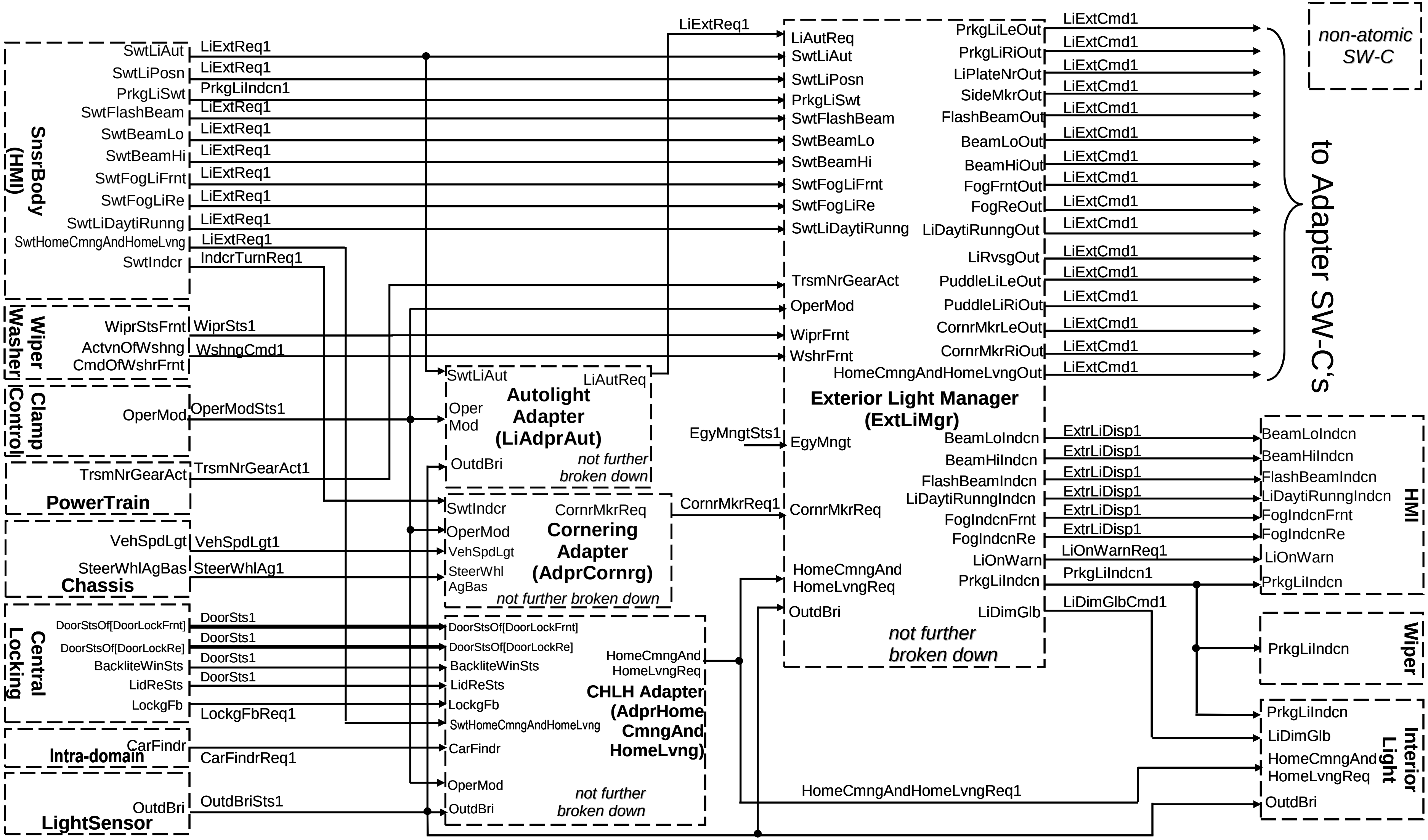
4.6 Exterior Light

The `ExteriorLight` software component group controls the exterior light functionality of a car. `ExteriorLight` receives driver wishes, senses the environment and controls the behavior of the exterior light sources of a car.

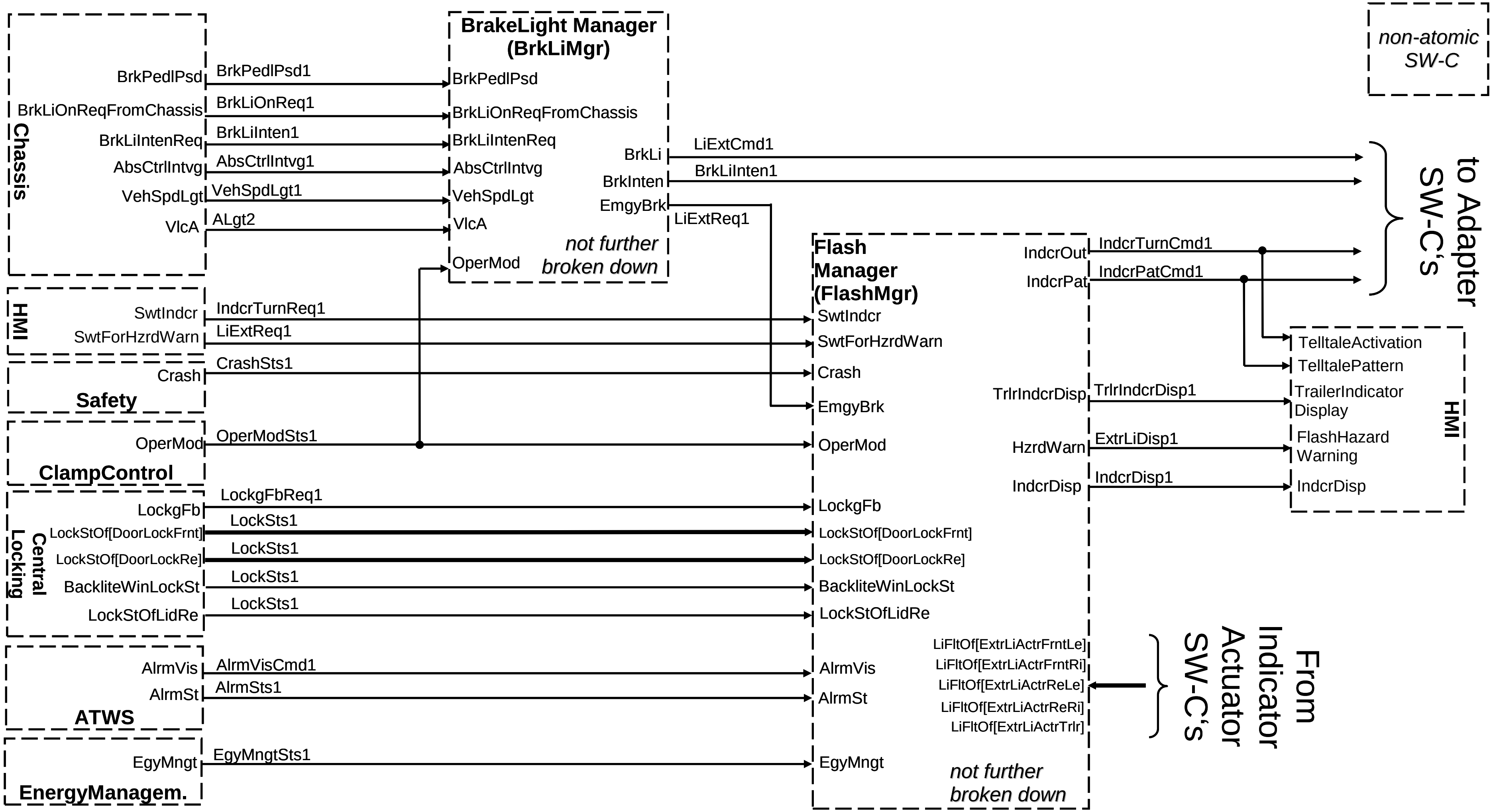
Despite its early introduction to automobiles already more than 100 years ago the exterior light remains a considerably complex functionality with many variations between different manufacturers. Therefore, the driving force behind the chosen decomposition of the exterior light functionality was to find components which can easily be adapted to different light strategies and their implementation and mapped to different vehicle network topologies.

All described components come with a superset of provided and requested ports. This should ensure that any variant is supplied with all necessary data.

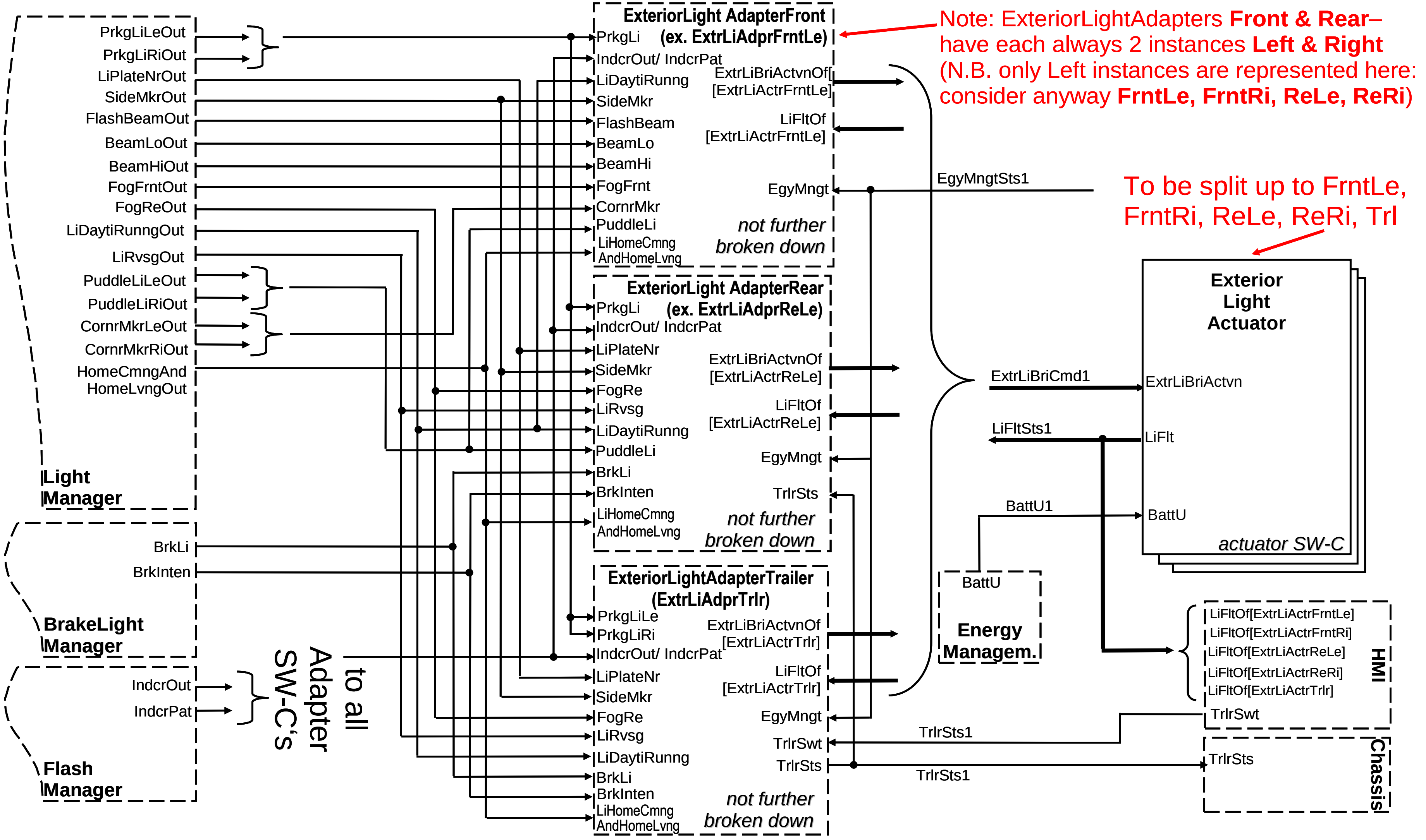
Components related to `ExteriorLight` are bundled in several Blocks.



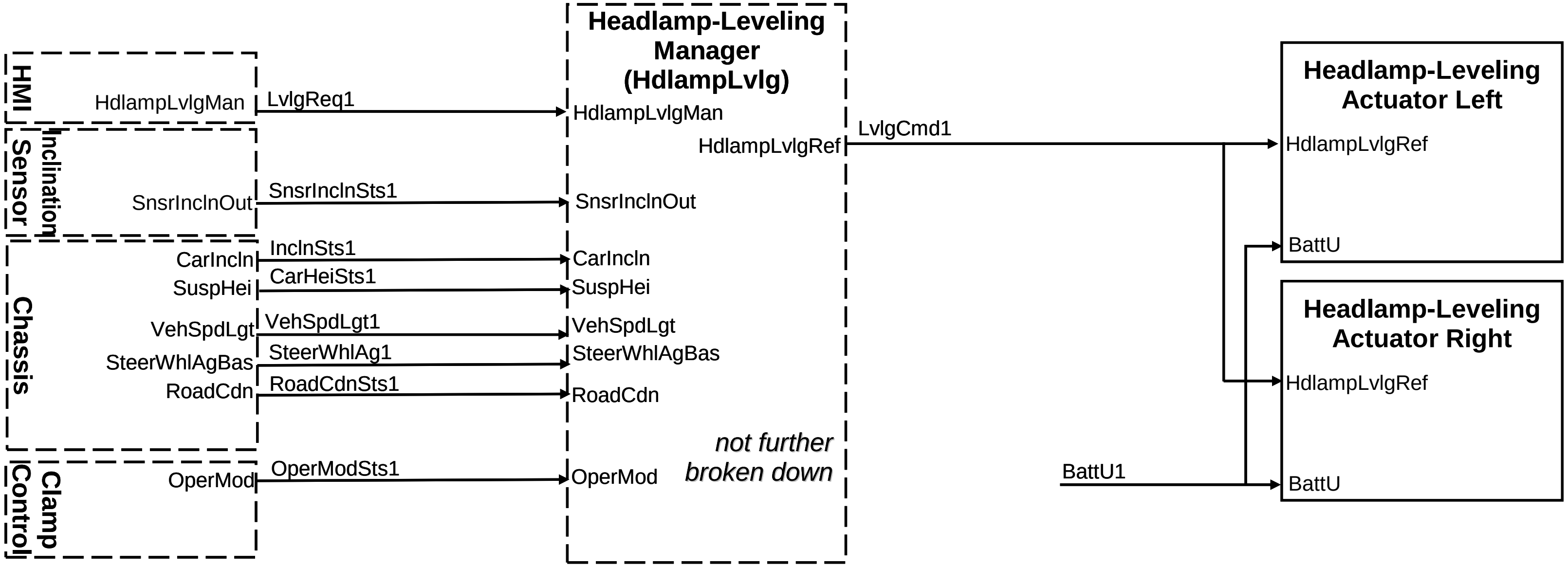
ExteriorLight Fig 1



ExteriorLight Fig 2



ExteriorLight Fig 3



ExteriorLight Fig 4

Figures 1 to 4 shows the decomposition of exterior light architecture. Arrows indicate data flows.

Remark: This decomposition is used for explanation of standardized interfaces

4.6.1 Overview on Exterior Light

4.6.1.1 ExteriorLightManager

Inside ExteriorLightManager, BrakeLightManager, FlashManager the relevant core functionality is implemented (not further broken down). These Manager components are considered to be configurable. This means, that in principle a standardized implementation should be feasible and its adaptation to OEM specific scenarios should be possible by configuration.

4.6.1.2 ExteriorLightAdapter

ExteriorLightAdapter components act as a “translator” between OEM specific interfaces which are likely to change. A common variation point between OEMs is the handling of light substitution in case of errors of the different light sources. This task is designated to the ExteriorLightAdapter components. It should be possible to construct this component in such a manner, that enables an exchange of its failure and replacement strategy by a simple exchange of some configuration tables. Please also note that ExteriorLightAdapter components front and rear are considered to be a component of which multiple instances can co-exist in a vehicle. This is required as it should be possible to run them in a physical front module and another one in a rear module or any other configuration

4.6.1.3 ExteriorLightActuator

ExteriorLightActuator finally addresses all light source components. Please note that ExtrLiActrFrntLe, ExtrLiActrFrntRi, ExtrLiActrReLe, ExtrLiActrReRi, ExtrLiActrTrlr are basically identical instances of a generic SWC ExtrLiActr. These 5 different SWC’s had been created in order to work around limitations of the AI table

4.6.1.4 Body Sensors for Exterior Light (HMI)

HMI contains software components responsible for collecting the user requests for various light modes such as fog light or high beam. Raw switch inputs are debounced and further abstraction from the switch hardware are provided. Also plausibility and availablitiy check of input signals are made. Please note that “HMI” is used only for non-normative description.

4.6.1.5 Other Functionalities

Other blocks like Chassis, Powertrain, TerminalClampControl exchange certain information with Exterior Light. For further details on these functionalities above shown see according documentation.

4.6.2 ExteriorLightManager

The mode masters for light, brake and flash functionality are defined as separate components. Also the ExteriorLightManager component encapsulates the prioritization strategy in case of simultaneous user and/or system requests and might be OEM specific due to company specific prioritization strategy but shall adhere to the interfaces defined in this document.

Its decision is based on engine and clamp status, vehicle speed and a number of (possibly competing) requests for exterior light modes or a subset of these inputs. Safety relevant aspects of exterior light are considered exclusively by the ExteriorLightManager; for example: loaddump during cranking or US legal restriction to maximum 2 active light sources (see remarks below) .

Apart from fog light and flash beam the main light functionality handled by the ExteriorLightManager includes low beam, high beam, day time running light, parking light, coming home, leaving home and carfinder. Also the decision on automatic light is taken by the ExteriorLightManager based upon user request and the precondition set by AutolightAdapter.

The ExteriorLightManager checks the admissibility of the requested light modes e.g. it only sets the “commands” for low beam if OperationMode indicates the ignition is on. It possibly modifies the requested light functionality which would mean in the latter example that a requested low beam results park lights to be activated (due to ignition is off).

The light functionality(ies) with the highest priority of all requested admissible functionalities is selected and published via its outputs.

4.6.2.1 Remarks

The decision on flash beam is included in the ExteriorLightManager as there are interconnections with other light modes. In US the front beams are limited to 2 at a time by legal restriction. There is a use-case that when foglight and high beam are activated, low beam must be switched off. In other requirements, when high beam and low beam are active, fog light must be switched off.

In order to obtain a ExteriorLightManager component free from competitive functionalities, the computation of the coming home/leaving home (CHLH) and the automatic light preconditions is transferred to other components. However, the final decision e.g. on CHLH modes is part of the ExteriorLightManager in order to avoid lights-off-glitches in the transition from e.g. Lowbeam to ComingHome functionality.

Day time running lights are turned on automatically when the ignition is started and are overridden when regular headlights are activated. Day time running lights typically are high-beam headlamps at reduced intensity or low-beam headlamps at full or reduced power. Tail lamps and/or turn signals may also be lit.

In case there is a legal requirement to switch on lights when it is raining the current wiping request has to be taken into account.

LiDimGlbCmd1 is the command of ExteriorLightManager to give a global exterior and interior light dimming information based on the outside brightness level. It is used e. g. to dim the exterior light depending on outside brightness.

ExteriorLightManager delivers functional state to be displayed by e.g. the MMI. Even with a temporarily switched off functionality, the display should still show the functionality. USE-CASE: a temporarily switched off fog light during flash beam (US) must still be displayed on the MMI. The same applies for light functionality that must be switched off during cranking.

4.6.3 BrakeLight Manager

A separate BrakeLightManager is established due to the fact that the brake light functionality has hardly no or only few functional interaction with the other light and flash modes like:

- Hazard Warning on Emergency Brake

Decision on Emergency Brake Situation is considered to be part of Exterior Lighting (other domains like Chassis will provide input information). In case of braking-force-dependent brake light the component calculates the percentage information of the brake light intensity. Information required might be:

- car velocity or change of car velocity (to determine a sudden change in car speed due to the braking),
- adaptive cruise control braking requests (e.g. in automated stop-and-go traffic),
- actual (de-)acceleration (HINT: brake pedal angle or change of brake pedal angle is supposed not to be required as input information).

With this information it should be possible to evaluate the brake forces in a brake event and therefore to request certain light effects to be visualized with the aid of the corresponding light sources.

In case of conventional brake light the binary input from the brake light switch is used one-to-one to zero or full intensity without any further conversion. Please note that BrakeLightsOnRequest is provided by chassis system and indicates system generated braking (e.g. anti-blocking system).

The physical brake lights may be adaptive, that is they can be activated with different brightness levels or different number (segments) of light sources that are illuminated while braking. That way, different levels of braking emergencies and therefore brake intensities can be indicated to following traffic.

4.6.4 Flash Manager

The flashing functionality is implemented in FlashManager which encapsulates the prioritization strategy in case of simultaneous user and/or system requests. Due to different prioritization strategies it might be necessary that each OEM establishes his own component, however it shall adhere to the interfaces defined in this document.

The FlashManager software component has to perform the following:

- generate the current blink mode from the various requests
- generate normal and special blink dynamics for normal operation and driver feedback from
  - blink mode,
  - clamp status,
  - blink mode dynamics description.

The incoming requests are evaluated based on the currently executed blink mode. If necessary, the currently executed mode will be changed to the appropriate mode. The current blink dynamics information will be updated as well depending on the blink configuration.

Flash Manager has to set the contents of the IndcrTurnCmd1. Different timing and sequencing of on/off phases of the blinkers must be possible (see below: Timing/Synchronization).

Therefore IndcrTurnCmd1 contains a record with five fields:

- IndcrTrig: the trigger for the blink functionality (stop, left, right, both)
- IndcrTelltl: (not) activate telltale
- IndcrNrPat provides the pattern handle which has to be used during the current blink sequence
- IndcrCycForStop: hardstop/softstop on ON or OFF phase of current cycle
- IndcrCycNotStop: indicator_cycle is number of remaining cycles to be executed without re- triggering

FlashManager may also set the contents of the IndcrPatCmd1 (e.g. as long as there are “static” indicator patterns used there is no need for usage of this signal). IndcrPatCmd1 is a record of

- IndcrNrPat
- TiOn
- TiOff

It is supposed that no complex bit pattern is to be performed by ExteriorLightActuator.

4.6.5 Use case descriptions

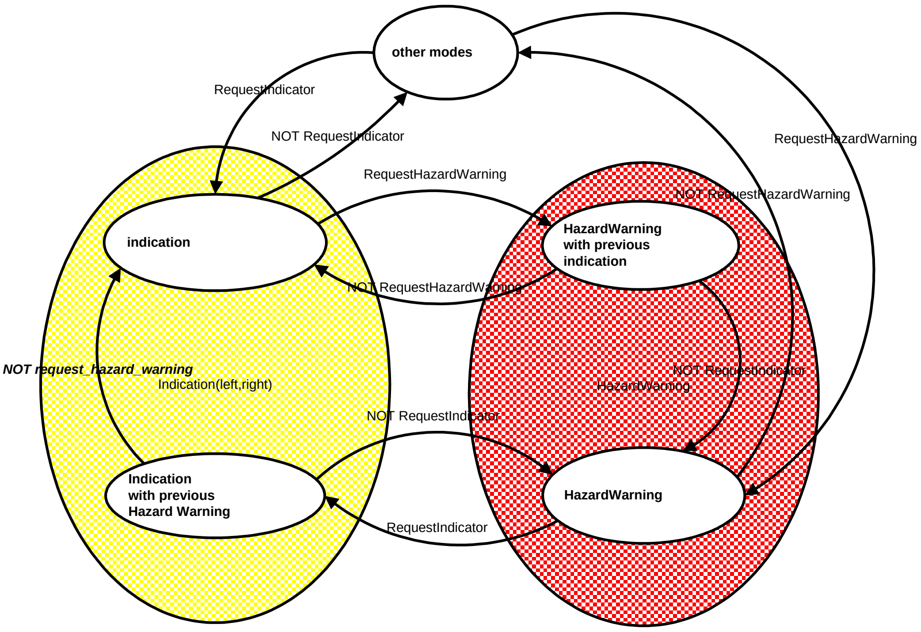
4.6.5.1 Blink mode transitions

The blink mode transition table excerpts below (Figure 5: sample transition table) shows the behavior of one sample implementation of a `FlashManager` software component.

current blink mode	Request	new blink mode
`LockingFeedback`	`HazardWarning`	`HazardWarning`
	`CrashWarning`	`CrashWarning`
	`IndicatorLeft`	`IndicatorLeft`
	`IndicatorRight`	`IndicatorRight`
	`Alarm`	`Alarm`
	`PanicAlarm`	`PanicAlarm`
	`LockingFeedback`	`LockingFeedback`
	`ArmingFeedback`	`LockingFeedback`
	`KeyLearn`	`LockingFeedback`
	`None`	`None`

ExteriorLight Fig 5

Fig 5 shows a sample transition table



ExteriorLight Fig 6

ExteriorLight Fig 6 shows an example of blink mode transition diagram

Above diagram shows one example of the blink mode transitions, reduced to the known hazard_warning (right) and indication (left) modes.

Transitions take place from “other mode” to the “simple” modes (hazard warning and indication blinking), depending on the requests.

Transitions from the simple modes take place when there are no particular requests for that mode or when the other request becomes active. Provided the currently executed state is `Indication(left,right)`, then if the `RequestIndicator` is removed, a transition to “other modes” will be executed.

When `hazard_warning` is requested, while `IndicationBlinking` is active, `HazardWarning` will be executed until `RequestHazardWarning` is removed. Then, `IndicationBlinking` will be executed again, if `RequestIndicator` is still active.

4.6.5.2 Timing/Synchronization

4.6.5.2.1 Timing information of a blink sequence:

For different blink requests or other circumstances, different timing and sequencing of on/off phases of the blinkers must be possible.

- lower duty-cycle for hazard-warning at ignition off
- higher frequency when an indicator light is broken
- more than one on-off-cycle with different duty-cycles

4.6.5.2.2 Trigger a limited number of sequences:

It must be possible to trigger a blink sequence of one or more cycles of a certain pattern.

- trigger a sequence of 3 on/off cycles for alarm blinking
- trigger only one on/off cycle

4.6.5.2.3 Trigger a continuous blink sequence:

It must be possible to start and re-trigger a continuous blink sequence by the `Flash Manager` directly.

- start continuous hazard blinking, it must not be stopped by e.g energy management or mode management .
- Start continuous alarm blinking, it may be stopped by certain circumstances, e.g energy management

4.6.5.2.4 Trigger on/off cycles by the Flash Manager directly:

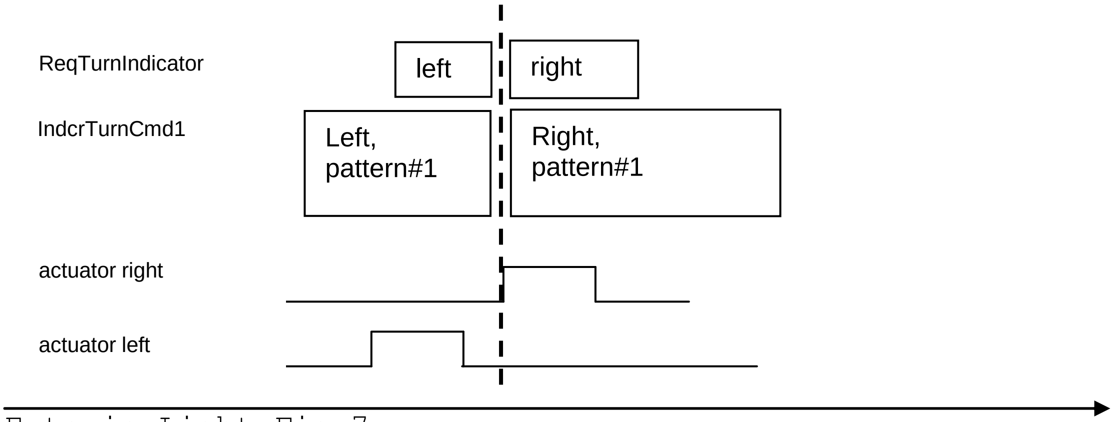
It must be possible to trigger on/off cycles by the `FlashManager` directly.

- submit the on/off commands continuously.
- submit on and off triggers

4.6.5.2.4.1 Example Timing Use Case

A change from one blink mode (including off) to another blink mode could always start with the on-phase of the output. The requested indicator pattern must be up to date, when a new blink mode is sent.

A change from right to left indicator blinking and vice versa could always trigger a new cycle, beginning with the on-phase. (see Figure below).



ExteriorLight Fig 7

Fig 7 shows an immediate mode change from indicator_left to indicator_right

4.6.5.3 Miscellaneous

FlashManager may also set Arming Feedback. (e.g. a LED that shows the activation of the arming system).

EgyMngtSts1 may be used for a change of the duty cycle/level in order to save energy.

TrlrSts1 might be used for the modification of blinking frequency for driver feedback in case of a failure of a trailer indicator. Usage of TrlrSts1 as indication for presence of trailer should be carefully considered as there are different possibilities to generate this signal:

- in case of a HW switch the presence of trailer can be detected by reading the switch and providing the status to any SW-C who is interested in.
- in case of missing switch the flasher(blinker) still detects the existance of trailer by measuring the current through the bulbs (this is safety requirement). In this case the provided information to other SW-Cs might lead to errors.

Remark: the evaluation of the "TrailerStatus" is planned for future extensions.

4.6.6 Autolight Adapter

The Autolight Adapter encapsulates the decision on automatic switch on/off the low beams (including other lamps required due to legal regulations like tail lamps, …).

4.6.7 Cornering Adapter

Inside Cornering Adapter the algorithm is located which decides –depending on various input information- on the (de-)activation of static cornering lights.

4.6.8 CHLH Adapter

The Coming Home Leaving Home (CHLH) functionality provides a special light service to the driver. A use case for general CHLH is to illuminate the way home after the driver has turned off the engine and left the car. Therefore e.g the front lights stay activated for a predefined or personalized time.

The CHLH functionality mainly consists of logic to determine the desired CHLH light request from light switches, general vehicle state information (e.g. clamps) and environmental conditions (e.g. brightness). Usually a timer is spawned to turn the lights off after a certain period of time. The time can be set through configuration by either vehicle coding or personalization.

The CHLH module may use inputs from:

- door status information including passenger/driver door and rear lid
- clamp information stating the current KL15 condition
- surrounding light characteristics provided by a light sensor (outside brightness)
- locking feedback as an additional path for the control of CHLH
- remote requests for leaving home functionalities e.g. carfinder (Use case for remote leaving home functionality might be activation of leaving home via remote key before one leaves the house at nighttime to illuminate the way to his car)
- driver requests for general activation of CHLH functionality e.g. through a switch (manual activation of CHLH functionality if further conditions as e.g. engine off are fulfilled).

The output of CHLH module is defined as a precondition for the ExteriorLightManager. Note, that CHLH might also be activated or deactivated via personalization.

4.6.9 ExteriorLight AdapterFront/Rear/Trailer

A common variation point between OEMs is the handling of light substitution in case of errors of the different light sources. This task is designated to the ExteriorLightAdapter components. It should be possible to construct these components in such a manner, that enables an exchange of its failure and replacement strategy by a simple exchange of some configuration tables. Please also note that ExteriorLightAdapter‘s are considered to be a component of which multiple instances co-exist in a vehicle. This is required as it should be possible to run e.g. one ExteriorLightAdapterFront in a physical front left ECU and another one in a front right ECU.

The task of the ExteriorLightAdapters is to drive the exterior light actuators (represented by the components in ExteriorLightActuator) via output

BriCmd1. The setting of the actuator brightness is mainly based on the light modes. The component receives the light modes as input from the ExteriorLightManager and the blink modes from the FlashManager (via IndcrTurnCmd1). Additionally it receives commands from the BrakeLightManager for the brake lights.

In addition to the requests mentioned above, the decision, which lights need to be active at which brightness in a mode depends on:

- the country coding and related light specific variant coding,
- whether a trailer is connected,
- whether lights are broken and have to be replaced by other lights.

For each light that is controlled by ExteriorLightAdapter, the software component calculates a brightness value (in percentage relative to the full brightness of the light source), by taking into account all the relevant input values that affect that particular light. An example table is shown in the table below.

Command for parkinglights	Command for daytime running light	Command for lowbeam	LiFltSts1 of positionlight front_left	BriCmd1 for lowbeam (left)
Left	Off	Off	ok	0%
Left	Off	Off	failure	40% replace broken position light
Off	On	Off	<irrelevant>	92% implement daytime running light
Off	Off	On	<irrelevant>	100% low beam

ExteriorLight Fig 8
Fig 8 shows an example calculation table for low beam light front left

The ExteriorLightAdapter software component may be instantiated once or several times in a car. Depending on the physical network architecture, there might be only a single instance on a central ECU or several instances distributed on several ECU's. An example with several instances is a car with one ECU that controls the back lights with one instance of ExteriorLightAdapter and two ECUs for the two front light systems, with one instance each

There are several command interfaces used between ExteriorLightManager and ExteriorLightAdapter. This was done since the main light functionalities shall be activated independently from each other. Thus changes to one light function (e.g. fog light switched on or off) do not affect the state of the other light functionalities (e.g. low beam or high beam). Note, however, that not all combinations might be valid – there might even be country specific limitations on the combination of light functionalities.

There are two main advantages of splitting functionality to different interfaces, compared to a solution where all modes are supported by the same interface using a record structure:

- as explained before, there might be multiple instances of ExteriorLightAdapter that control different lights of the car. Some instances might not need all the mode interfaces as input.

- a record is sent completely when a single entry changes. It might be more efficient to have independent data on independent interfaces to save communication bandwidth.

In the following paragraphs, all requested ports are listed. The description why the data is needed is intended as a hint for the reader and may not provide all kinds of usage of the input in the software component.

The input EgyMngtSts1 may be used in ExteriorLightAdapter to dim certain lights if energy management requests so. A complete deactivation of a function should already be taken into account only by ExteriorLightManager or FlashManager. Note also that the output from ExteriorLightAdapter is a brightness value in percentage relative to the full brightness of the light. Thus ExteriorLightAdapter does not perform dimming due to over-voltage from power supply. This dimming is light type specific and has to be performed by the respective ExteriorLightActuator components.

For each light that is controlled by ExteriorLightAdapter, the software component calculates a brightness value (in percentage relative to full brightness of the light). Thus, a value of 0% means that the light is switched off, and 100% means that the light has to be lit with full brightness. Brightness values have been chosen since they are independent of the light source implementation (bulb or neon or LED or LED array or xenon or …). Furthermore, even lamps of the same type may have different voltages used for full brightness and different response curves for different input voltages.

For light sources that do not support dimming (such as some gas emitter lamps) or that support a limited set of brightness values (such as LED array that switch on more or less LEDs), a useful translation from the different dimming factors to the levels that can be set has to be defined and set by the actuator component.

4.6.10 Exterior Light Actuator

All exterior light actuators are described by the ExtrLiActr[ext_light] components (see SWC ExtrLiActrFrntLe, ExtrLiActrFrntRi, ExtrLiActrReLe, ExtrLiActrReRi, ExtrLiActrTrlr in AI table) where [ext_light] is a placeholder for individual light sources e.g. parking light rear left. It might be decomposed into a light actuator type independent and a light actuator type dependent component, where light actuator types include conventional bulbs, LEDs, neon-, halogen- and xenon-lights, or other decompositions which allow the switching of lights on a more “generic” level.

A SW-C that encapsulates more than one light source with a combination of several single actuators is also possible (example: combined brake-taillamp). In this case the combined actuator component has more than one ExtrLiBriActvn port and more than one LiFlt port.

The operational/defective feedback of ExteriorLightActuator is sufficient (e.g. for doubling of blink frequency in case of defective bulb). No use cases are known for different behavior of ExteriorLightManager, Flash Manager, BrakeLight Manager depending on type of failure (open load, short circuit). E.g. to be distinguished only for diagnostic purposes.

4.6.11 Headlamp-Leveling Manager

The `Headlamp-Leveling Manager` controls the levelling of the front beams. It's not necessary to distinguish leveling value for left and right headlight.

The requests and commands for leveling are supposed to be of type percentage (as well as information on car's height inclination.

4.6.12 External interfaces

The `AntiTheftWarningSystem` encapsulates the theft protection system of a car and provides the necessary interfaces towards the exterior light. Typically, the arming status of the `AntiTheftWarningSystem` is given by a visual feedback of the exterior light and, in case of a theft incident, the `AntiTheftWarningSystem` also requests exterior lighting to guard the car visually (i.e. turning on indicator lamps or flashing them; no other use cases are known besides using turn indicators). Note that `AntiTheftWarningSystem` only requests an alarm; the visual representation is up to exterior light. Same applies for the visible feedback request of `CentralLocking`.

TerminalClampControl provides basic information regarding the driver's key position.

The vehicles crash detection system is encapsulated in Safety. Note, that this component may hide multiple crash sensors and provides only a single signal.

Out of `Chassis` the information is provided whether a trailer is connected. Same applies for the brake pedal switch (see BrakePedalPressed).

Information on whether reverse gear is selected will be provided by `Powertrain.`

4.6.13 ports to AUTOSAR-Services

For diagnostic ports and further ports to AUTOSAR-Services see according remarks inside chapter interior light.

4.6.14 Terminology Definitions

Exterior Light Sources
Note: the terms Beam, Bulb, Lamp, Light, Lighting, Illumination and Signal (or Beams, Bulbs, Lamps, Lights, Lighting, Illumination and Signals) are assumed interchangeable unless specifically stated otherwise. E.g. Stop Lamps are the same as Stop Illumination.

Note: These terms are presented as a guide to help comprehension of the many different names used for similar exterior lighting functions. The guide must not be used as a reference source for legislated lighting requirements.

Note: The lamp colours described below are generic terms. “Yellow” may also be known to some territories as “Orange” or “Amber”. Reference should be made to legal requirements documentation in order to determine the actual photometric requirements for a colour in any given territory.

ID	Name and Description	No. of lights	Colour	Location
1	**AFS** See *Advanced Front Lighting System.*	-	-	-
2	**Advanced Front Lighting System** Advanced Front Lighting Systems modify headlight output so that the beam pattern is optimized for different driving conditions such as vehicle direction and speed. AFS usually operates on *Low Beam* and *High Beam* lighting elements. They may be implemented using dynamic and static systems incorporating Halogen, High Intensity Discharge (HID)/Xenon and Light Emitting Diode (LED) sources	2 or 4	White	Front
3	**Backup Lamp(s)** See *Reverse Lighting.*	-	-	-
4	**Beam Steering** See *Advanced Front Lighting System.*	-	-	-
5	**Brake Lights** Bulbs illuminate when vehicle braking is applied (not handbrake) to warn following vehicles.	2	Red	Rear - symmetrical - as far apart as practicable
6	**Centre High Mounted Stop Lamp** Single *Brake Light* mounted at the back of a vehicle higher than the other two *Brake Lights.*	1	Red	Rear - Centreline
7	**CHMSL** See *Centre High Mounted Stop Lamp.*	-	-	-
8	**Cornering Lamps** Also known as *Static Bending Lamps*. Bulbs on front corners of the vehicle that light as the vehicle turns a corner junction in order to illuminate potential obstacles around that corner. There is no movement of reflector or light source.	2	White	Front
9	**Daytime Running Lamps** Combinations of *Sidelights* and *Low Beam* automatically turned on when the vehicle is operated, even if it is daylight. Two common variations exist: Scandinavian and Canadi-	Min. 2	White or Yellow	Front - symmetrical - as far apart as practicable

ID	Name and Description	No. of lights	Colour	Location
	an. Indicate in-use vehicle. "Daytime Running Lamps " may exist as separate outputs too besides Low beam and High beam. Also on some cars (Nissan or VW) the fog lamps may be used for Daytime Running Lamps			
10	**Dipped Beam** See *Low Beam*	-	-	-
11	**Direction Indicators** These lamps are located at the corners of the vehicle. Repeaters may also be used on the side, forward of the front doors. Direction Indicators are used in groups (all left lamps or all right lamps) to indicate turning of the vehicle. Additionally, all lamps may be flashed simultaneously to act as *Hazard Warning Lights*. Direction indicators may also be called *Front*, *Left*, *Right* or *Rear Turn Indicators*.	Min. 4	Yellow	Front and Rear – symmetrical – as far apart as possible
12	**Driving Lights** Spot lights added to the front of the vehicle to improve forward visibility of ground objects. Should be adjusted so as not to dazzle on-coming traffic.	-	White	-
13	**DRL** See *Daytime Running Lamps.*	-	-	-
14	**Dynamic Cornering Lamps** Modified *Low Beam* lighting that dynamically steers the beam pattern into turns in the road. This increases effective range of visibility. These systems typically use steering angle and vehicle speed to calculate the beam steer angle.	-	White	-
15	**Front Fog Lights** Additional front lighting, usually mounted below the High Beam and Dipped Beam sources. These lights are aimed low to improve visibility during fogging conditions and reduce the amount of back scatter to the driver compared to Low Beam.	-	-	-
16	**Front Side Marker Lamps** Front and Rear side marker lamps indicate vehicle's presence and length	Min. 2	Yellow	Side at front - as far forward as possible
17	**Front Side Reflex Reflectors** Front and Rear side reflex reflectors indicate vehicle's presence and length	Min. 2	Yellow	Side at front - as far forward as possible

ID	Name and Description	No. of lights	Colour	Location
				- facing sideways
18	**Front Turn Signals** Common term for Left and Right *Direction Indicators* at the front of the vehicle. Indicate direction of turn. Front turn signals are usually also used for front hazard warning lights.	Min. 2	Yellow	Front - symmetrical - as far apart as practicable
19	**Hazard Warning Lights** Simultaneous and synchronous operation of left and right direction indicators, front and rear, to identify a vehicle as disabled and presenting a hazard to other traffic.	4+	Yellow	All
20	**Headlamps** Common term covering *Low Beam* and *Main Beam.*	4	White	Front
21	**HIDs** See *High Intensity Discharge Lamps.*	-	-	-
22	**High Beam** Forward road illumination used for long distance visibility. Should be turned off when traffic is on-coming to prevent dazzling. Alternatively used as shutter control in order to create low/high beam in some cars (ex. VW). Shutter (kind of relay control) is used to separate the low-high beam activities having only one bulb output.	Min. 2	White	Front – symmetrical (If 4 lamp system, inboard or below *Low Beams*)
23	**High Intensity Discharge Lamps** A light source technology that produces high output illumination suitable for use as Low Beam and High Beam. Usually needs automatic levelling to prevent dazzling of oncoming drivers.	-	-	-
24	**High Mounted Stop Lamp** See *Centre High Mounted Stop Lamp.*	-	-	-
25	**HMSL** See *Centre High Mounted Stop Lamp.*	-	-	-
26	**Left Turn Indicators** Common term for Left *Direction Indicators* at the front, side and rear of the vehicle. Indicate direction of turn (to the left).	Min. 2	Yellow	Front and rear – symmetrical with right indicators
27	**License Plate Lamp(s)** Illumination of vehicle external license plates on rear and front (where fitted).	Min. 1	White	On rear – above or at side of

ID	Name and Description	No. of lights	Colour	Location
				plate
28	**Low Beam** Forward road illumination used when driving at night. Maybe used in traffic with on-coming vehicles.	Min. 2	White	Front - symmetrical - as far apart as practicable (If 4 lamp system - outboard or above *High Beams*)
29	**Main Beam** See *High Beam*.	-	-	-
30	**Parking Lamps** Indicate vehicle presence and width during driving. May be used when the vehicle is parked to make it visible to other road users. Low power consumption is required to prevent battery discharge. Some systems allow left or right parking lamps to be selected independently to indicate only the traffic-side of parked vehicle.	Min. 2	White or Yellow	Front - symmetrical - as far apart as practicable
32	**Rear Fog Lights** Additional bright Tail Lamps used in foggy conditions to indicate the vehicle presence and warn following vehicles.	1 or 2	Red	Rear - symmetrical - as far apart as practicable
37	**Rear Reflex Reflectors** These passive reflectors indicate the vehicle presence and width to vehicles behind.	Min. 2	Red	Rear - symmetrical - as far apart as practicable
38	**Rear Side Marker Lamps** Front and Rear side marker lamps indicate vehicle's presence and length	Min. 2	Red	Side at rear - as far back as possible
39	**Rear Side Reflex Reflectors** Front and Rear side reflex reflectors indicate vehicle's presence and length	Min. 2	Red	Side at rear - as far back as possible - facing sideways
40	**Rear Turn Signals** Common term for Left and Right *Direction Indicators* at the rear of the vehicle. Indicate direction of turn. Rear turn signals are usu-	Min. 2	Yellow	Rear - symmetrical - as far apart as

ID	Name and Description	No. of lights	Colour	Location
	ally also used for rear *Hazard Warning Lights*.			practicable
41	**Reverse Lighting** Lighting to improve visibility of obstacles on the ground behind the vehicle when reversing. Also, alerts vehicles approaching from rear.	1 or 2	White	Rear
42	**Right Turn Indicators** Common term for Right *Direction Indicators* at the front, side and rear of the vehicle. Indicate direction of turn (to the right).	Min. 2	Yellow	Front and rear – symmetrical with left indicators
44	**Side Repeaters** Additional *Direction Indicators* located on the side of the vehicle in front of the front doors. These flash as part of the left and right *Turn Signal* groups, and when *Hazard Warning Lights* are operated. They allow vehicles approaching a junction to see the direction indicators of a vehicle from the side.	2	Yellow	Side of vehicle – symmetrical – facing sideways
45	**Sidelights** See *Parking Lights*.			
46	**Spot Lights** See *Driving Lamps*.	-	-	-
47	**Static Bending Lamps** See *Cornering Lamps*.	-	-	-
48	**Static Cornering Lamps** See *Cornering Lamps*.	-	-	-
49	**Stop Lamps** See *Brake Lights*.	-	-	-
50	**Tail Lamps** Indicate vehicle's presence and width to vehicles behind.	Min. 2	Red	Rear - symmetrical - as far apart as practicable
51	**Turn Lights** See *Direction Indicators*.	-	-	-

Source: SVN1, filename: Terminology Definitions for Body 0.2.doc, Author: John Elvidge, Visteon UK Ltd.

4.6.15 List of light sources

`LightSource[ext_light]` encapsulates different light sources of a vehicle and provides access to these light sources via an implementation independent interface. Implementations which use physical light sources incapable of emitting dimmed light (e.g. Xenon) may define a percentage value, above which the light source is turned on respectively below which the light source is turned off.

Additional actuators, which are required to put the light source in its operational state (e.g. the fold-up motor of a lamp) are also encapsulated by this component.

`LightSource[ext_light]` is – for the sake of documentation reduction – a "template component": it is only documented once and its instances are listed in the next paragraph, resulting in many components, where [ext_light] is replaced by the name given in the next section (e.g. `BrkLiLe`).

`LightSource[ext_light]` has to be replaced by one the following instances (please note that ExtrLiActrFrntLe, ExtrLiActrFrntRi, ExtrLiActrReLe, ExtrLiActrReRi, ExtrLiActrTrlr are basically identical instances of a generic SWC ExtrLiActr. These 5 different SWC's had been created in order to work around limitations of the AI table. Naturally only the appropriate instances of `LightSource[ext_light]` might be used with ExtrLiActrFrntLe, ExtrLiActrFrntRi, ExtrLiActrReLe, ExtrLiActrReRi, ExtrLiActrTrlr):

BrakeLights: BrkLiLe (left), LampForStopOfBrkLiCenHiMntd, BrkLiRi (right), BrkLiTrlrLe, BrkLiTrlrRi

Indicators: IndcrFrntLe (front_left), IndcrFrntRi, IndcrSideLe, IndcrSideRi, IndcrReLe, IndcrReRi (rear_right), IndcrTrlrLe, IndcrTrlrRi

FogLights: FogLiFrntLe, FogLiFrntRi, FogLiReLe, FogLiReRi, FogLiTrlrLe, FogLiTrlrRi

Headlight: BeamHiLe, BeamHiRi, BeamLoLe, BeamLoRi

Daytime running lights: LiDaytiRunngFrntLe, LiDaytiRunngFrntRi

LicensePlateIllumination: LiVehPlateNr (Vehicle), LiTrlrPlateNr (Trailer)

ParkingLights: PrkgLiFrntLe, PrkgLiFrntRi, PrkgLiReLe, PrkgLiReRi, PrkgLiTrlrLe, PrkgLiTrlrRi

PuddleLights: PuddleLiFrntLe, PuddleLiFrntRi, PuddleLiReLe, PuddleLiReRi

ReverseLights: LiLeForRvsg, LiRiForRvsg, TrlrLiLeForRvsg, TrlrLiRiForRvsg

SideMarkers: SideMkrFrntLe, SideMkrFrntRi, SideMkrCeLe, SideMkrCeRi, SideMkrReLe, SideMkrReRi

CornerMarkers: CornrMkrFrntLe, CornrMkrFrntRi, CornrMkrCeLe, CornrMkrCeRi, CornrMkrReLe, CornrMkrReRi

BatterySaver: separate “Actuator” switched from ExteriorLightManager (use case: Lowbeam, … hardwired directly from switch)

Curb lights are not considered for exterior light as they are strongly correlated with interior light functionality. They will be included in interior light refinement.

4.6.16 Known Deficiencies

Other open Documentation Issues:

- personalization parameters and interfaces
- variant coding also lifecycle dependant modes (like for activation / deactivation of certain blinking functionalities depending on the life-cycle mode e.g disable arming feedback during transportation; same applies for special interfaces needed for factory usage and other than normal operation vehicle modes).
- diagnostic coding/interface
- decomposition of sensor and actuator components (e.g. in light-type-dependent and light-type-independent)
- Adaptive Frontlighting System (AFS) is not standardized (only static cornering lights as part of exterior light standardization is supposed to be sufficient)

4.7 Antitheft Warning System

ATWS functionality represents an anti-theft warning system which can warn about an unauthorized car access attempt by means of optical and acoustical signals and, in addition, can prevent car’s engine from starting. All these decisions are taken based on status of car’s openings and based on readings from certain detection means, like car interior scanning sensor or car inclination sensor.

ATWS functionality is split into the following **subsystems**:
anti-theft alarm: implements the basic function of ATWS, i. e. warning an unauthorized intrusion into the car
panic alarm: provides an alarm warning at driver request (by remote control), independent of the anti-theft alarm system status

4.7.1 Alarm triggers

The alarm system can be triggered by a number of inputs as doors (driver door could benefit of a special treatment), hood, trunk, rear window, glass break detection, radio, car inclination, car interior scanner, battery backed sounder, ignition, battery voltage level, fuel lid etc. Any of these sensors will trigger an alarm if armed before. Some sensors, like interior movement detector, inclination modification detector, battery backed sounder and so on, can be involved in a more complex communication with the alarm system.

4.7.2 Alarm system status indications

- change of status from disarmed to armed and vice-versa (arming or disarming acknowledge) by flashing the lamps and/or chirping the horn;
- armed status of ATWS (sensors are monitored to detect intrusions) by flashing a LED;
- indication of alarms occurred in the current armed period of time when the user disarms the system by flashing the lamps or chirping the horn

4.7.3 Alarm States

<u>Armed</u>
- armed by different Central Locking requests sources, auto-arming (ex.rearming if remote unlock but no door opened, etc.)
<u>Disarmed</u>
- disarming by different Central Unlocking request sources or valid ignition key, etc.
<u>Prearmed</u>
- requirement dependent and it is different from sensor to sensor, time and trigger dependent
<u>Alarm</u>

- different sensors activation during armed state leading to visual and acoustical warning.

Sensors can be temporary excluded by user intervention in disarmed state.
Sensors can be in different states at arming and disarming depending on different preconditions leading to independent or grouped treatment of them.

Anti theft system needs supplier dependent implementations for communication with complex sensors (eg. inclination sensor, interior scanner, battery backed sensor, etc.) → complex drivers with command and status interfaces.

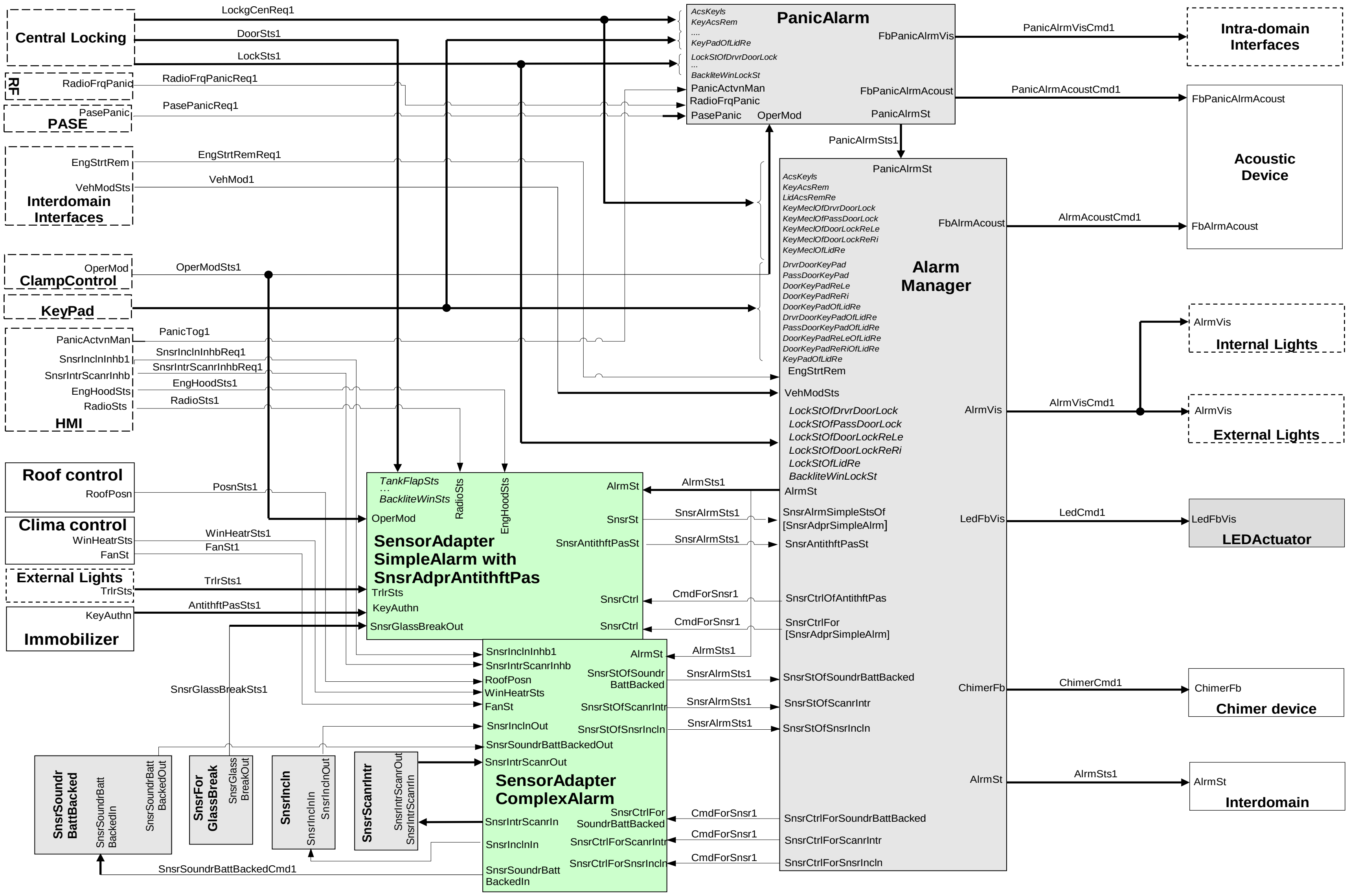
- Anti theft system features implementation is subject of international regulations.

4.7.4 Panic Alarm

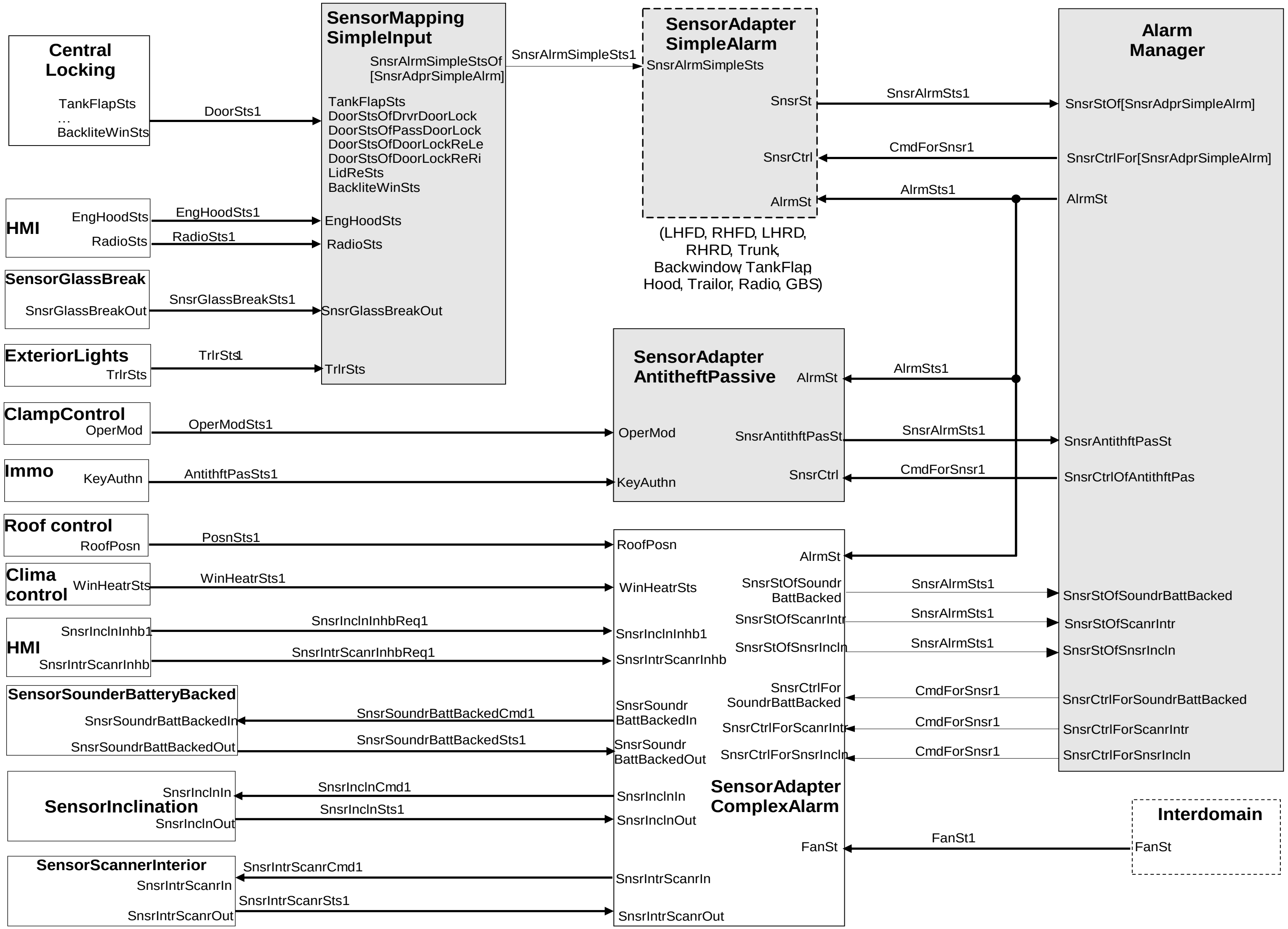
Panic alarm allows the user to activate the acoustical and visual warning devices installed on the vehicle independent of the ATWS state (arm, disarm, alarm). It is also possible for the vehicle user to switch off the panic alarm.

4.7.5 ATWS decomposition

In the following figure (Fig.1) is provided the main ATWS SWCs decomposition.
In the subsequent figure (Fig.2) are highlighted ATWS sensors and their connection to the Alarm manager.



AntitheftWarningSystem Fig.1



AntitheftWarningSystem Fig.2

The Alarm manager has to communicate with an important number of sensors through Sensor Adapter components. These adapter components can be simple one where the sensor itself is a switch type sensor. The complex adapter component deals with sensors where the sensor itself can be controlled using a certain communication line. In this case implementation of a Complex Driver is required.
Sensors can be excluded or inhibited by the alarm manager or the user.

The ATWS can be split on multiple ECUs taking into account the timing constraints of OEMs.

Simple sensor adapters can be instantiated according to the number of sensors of this type using the generic interfaces defined instantiating also the ports.

4.8 Horn

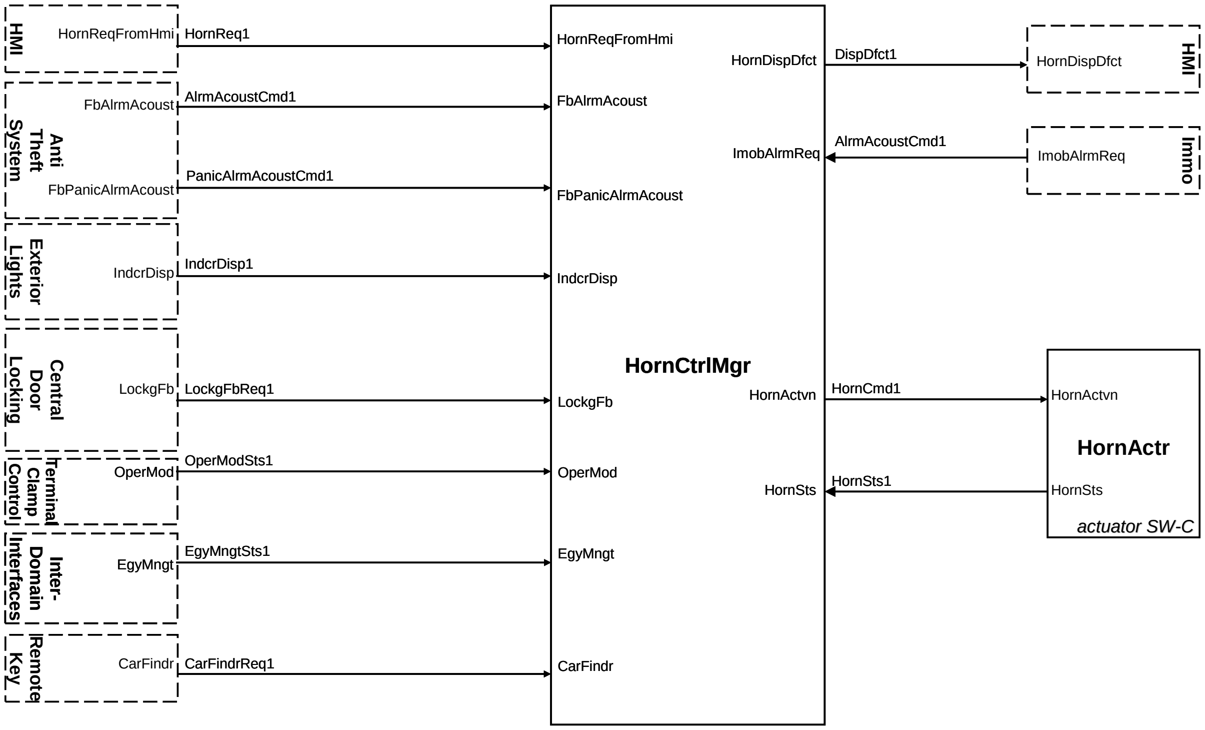
Horn functionality represents the feature to produce a sound, for warning other cars, pedestrians, or for giving information about the car status.

Only one SWC Manager composes the horn decomposition.

The different inputs of the decomposition are the following:
- HMI, with a switch
- ATWS, as acoustical signals for preventing theft
- Exterior lights
- Central Door Locking, for information about locking / unlocking status
- TerminalClampControl
- InterDomain
- Remote Key, to use the horn for finding the car.
- Status of Horn actuator

The outputs are:
- An indication of a possible defect of the horn, to the HMI
- The horn command to the actuator.

4.8.1 Horn decomposition



Horn Fig1

4.9 Defrost Control

4.9.1 Introduction

This chapter describes a basic architecture and gives an overview of the Defrost Control (DC) functionality.
The `DefrostControl` software component group controls the defrost control functionality of a car. `DefrostControl` receives driver wishes, senses the environment and controls the behavior of the defrost control surfaces of a car.
It decomposes Defrost Control functionality into the Defrost Control Manager component, Sensors components (that collect all the possible inputs to stimulate the functionality) and Actuators components (which perform DC Manager processed actions).

4.9.2 Sensors and actuators overview

Stimulus to DC Manager arrives from the following sources:

- HMI (which may consist of switches, touch screens or other input / output devices)
- HVAC (because defrosting action might also be initiated by the CabinClimatisation)
- TerminalClampControl

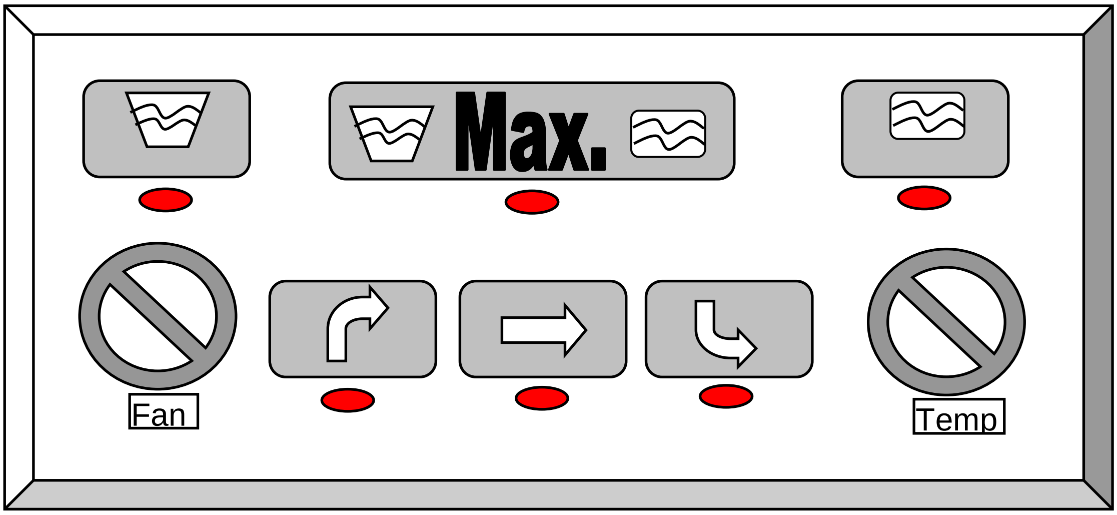
- Other Inter-domain interfaces (related to EnergyManagement conditions and BatteryVoltage level)

From Actuators side, there are two main types considered here:

- [Target]DefrostActuators: they perform the real activation of defrosting action
- DefrostTelltaleActuator: it performs only a visual indication.

To clarify some sensors and actuators presence, in particular the usage of a dedicated DefrostTelltaleActuator, it is useful to make some additional consideration, highlighted here in the following.

Regarding input from **HVAC**, please note that only the relevant interfaces regarding defrost functionality (including visual feedback for cabin climatization) are considered. Climate control functionality not directly related to defrosting, and particularly fan control, will be part of the decomposition of cabin climatization .
Because defrost functionality may often be implemented closely coupled to cabin climate control, also the **HMI** may be combined. In these cases, the cabin climate control may have some form of override on **defrost telltales**.
If a “Maximum” function is selected, the defrost telltales may be illuminated even if the system actually performs screen clearing only with blowers and without use of defrost heaters.



DefrostControl Fig 1

Another example is for **mirror defrost**. Exterior mirror defrosting may be automatically selected whenever rear screen defrost is selected but the interfaces allow for explicit independent mirror defrost control.

Additionally, defrost functions may be timed and the telltale automatically extinguished at the end of a defined period or the function may be continuous once initiated.
In general the **telltale** is always controlled via DC Manager, but initial request causing its activation may arrive from the HMI (that is through a user request of defrosting activation) or from HVAC. In either case, the telltale may report the actual status of the function or it may indicate the “selected” status, keeping the telltale illuminated

even if defrosting is suspended for power management reasons (e.g. **Start & Stop** functionality action).
In fact, there are in the recent car configurations, use-cases where the telltale indication on cluster regarding defrost status, is different than the real defroster actuator status. For example, during car stops at traffic lights when car engine might be automatically stopped (and started at acceleration) the telltale indication regarding defrost activation must be kept on even if the defrost actuator is stopped for that period in order to avoid driver confusion.

For this specific purpose is present among the Inter-domain Interfaces the **EnergyManagement** input, which takes into account also the **AlternatorStatus** to know if the alternator is recharging the battery or not.

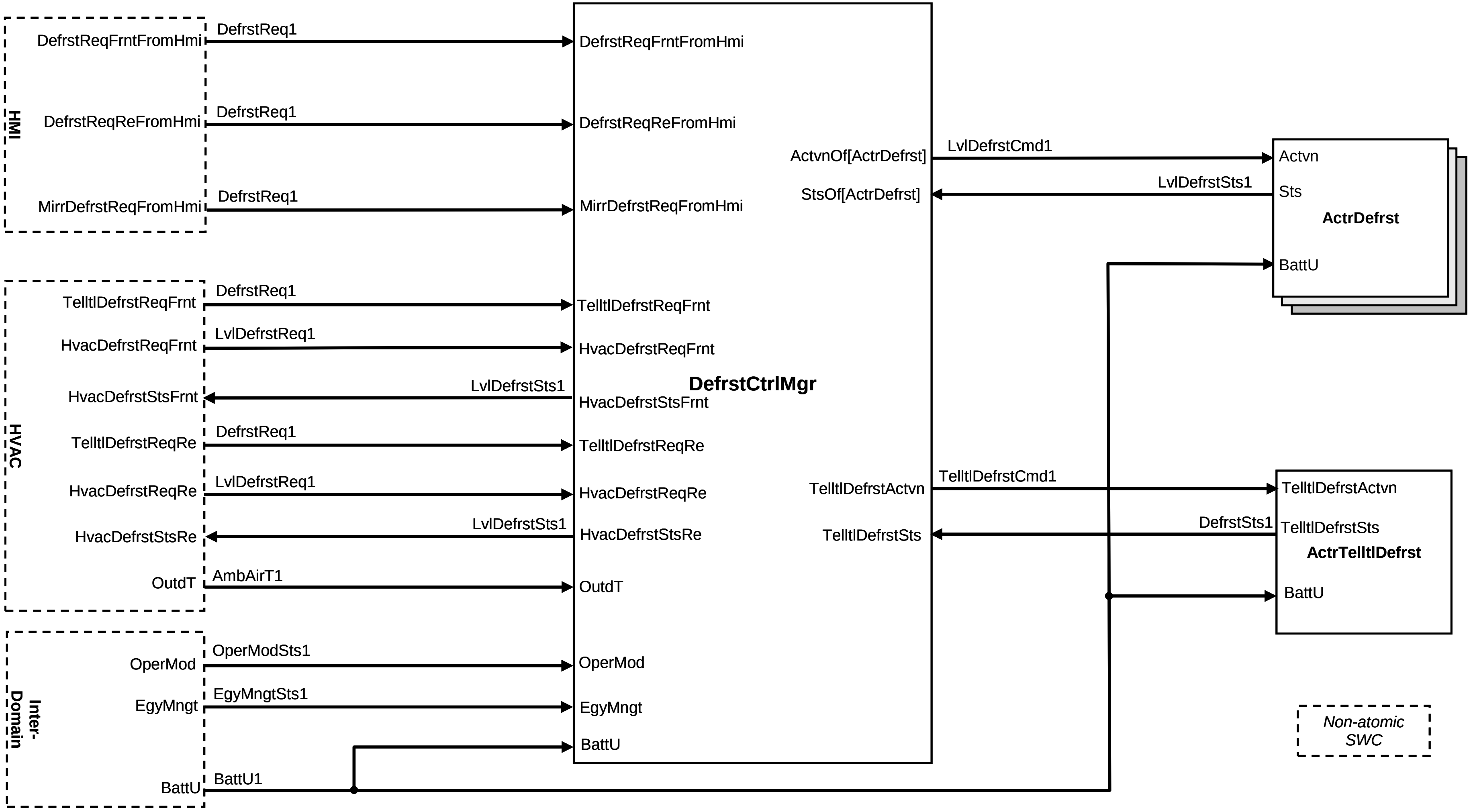
The same result applies, as described above, for short voltage drops or peaks, where the telltale must stay on if defrost was activated but without defrost actuator on and this reason explains the necessity of **Voltage** signal in Defrost functionality.

Considering both use cases above, has been introduced among the actuators, a **Defrost Telltale Actuator** instantiable SW-C.

Other Actuator SW-Cs provide the real activation of defrosting action and are related to Front, Rear and Mirror Defroster.
For all this Actuators, Voltage input signal is provided, because, in case of short voltage drops or peaks, actuators may be switched off independently from the decision of DC Manager, for reducing or saving loads.

4.9.3 Defrost Control Decomposition

In this section is provided the hierarchical decomposition of the Defrost Control functionality divided in Defrost Control Manager, sensor and actuator components.
Components related to `DefrostControl` are bundled in several blocks.
Arrows indicate data flows.

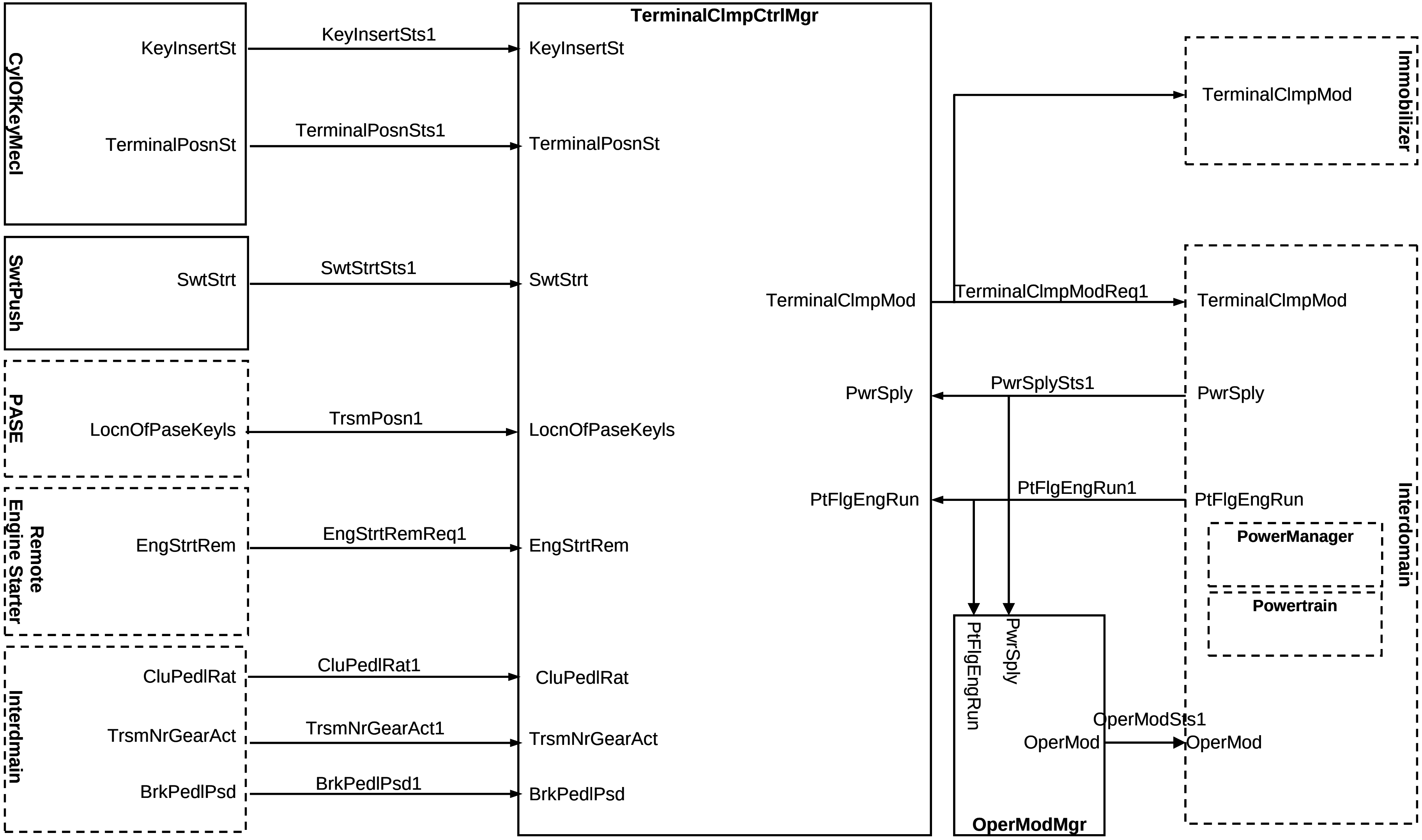


DefrostControl Fig 2

4.10 Terminal Clamp Control

The `terminal clamp control` software component group controls the mode change of electric power supply and driving motion power based on direct driver's operation and/or remote operation which is made by driver or driver's equipments. Power management system and powertrain system shall control operation mode compliant with the request generated by terminal clamp control function.

Components related to `Terminal Clamp control` are bundled in several blocks:



TerminalClampControl Fig1

4.10.1 Explanation of SW-Cs

Terminal Clamp Manager:
Terminal Clamp Manager arbitrates and controls the change requests of power supply and driving motion power from driver and/or some systems, and generates power supply request which is independent of real request devices such as mechanical key or Push Start.

Mechanical Key Cylinder
It provides the key position in mechanical cylinder which driver changed to, the Terminal Clamp Manager collects this driver's request to change power supply and driving motion power.

SwitchPush
It detects driver's request to change the operation mode such as "Start" or "Stop", and provides the driver's request to Terminal Clamp Manager.

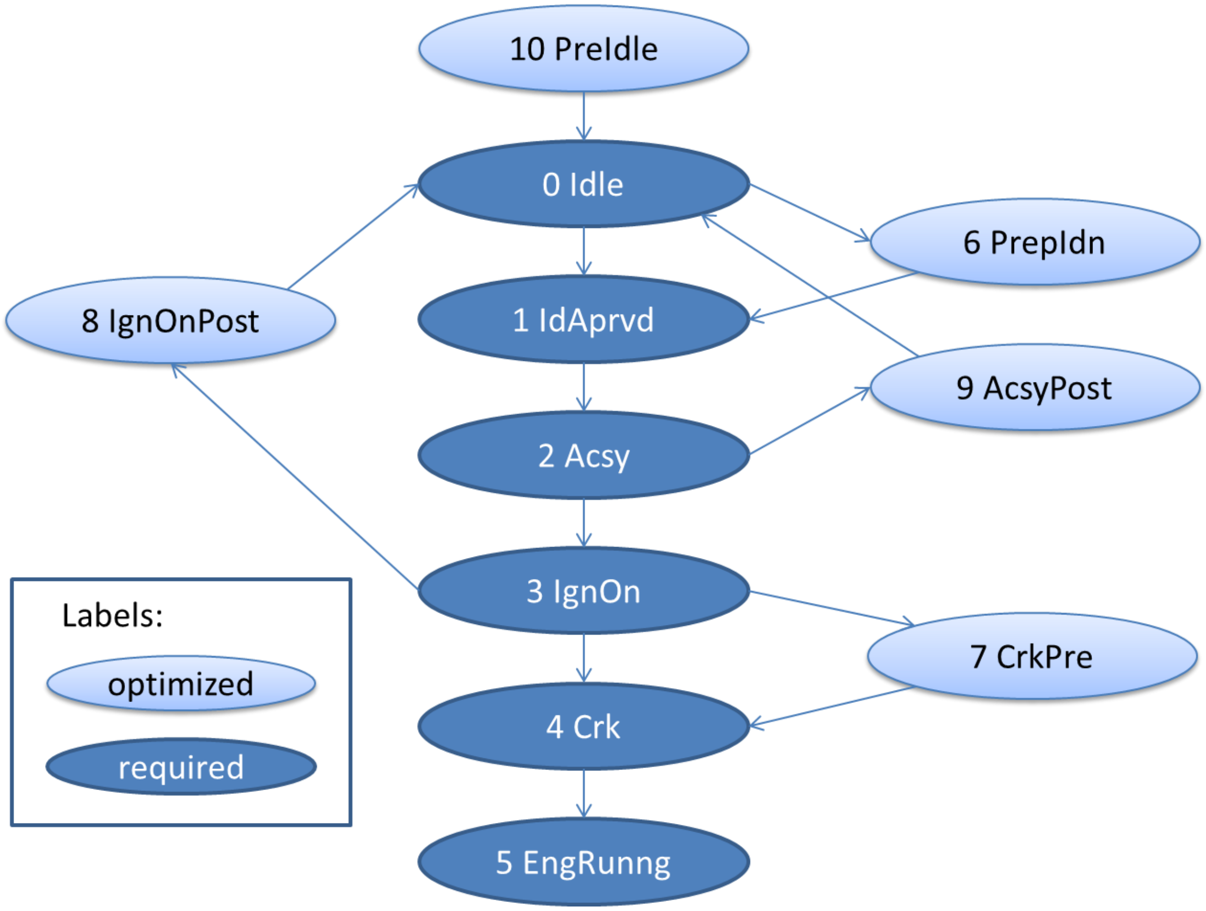
PASE
It provides also the position of driver with remote if he/she is in a car or not.

Remote Engine Starter
It provides request on Engine start/stop by owner.

Interdomain Power Manager, Powertrain
Power Manager supplies optimally power to whole vehicle systems based on diverse reqesests from other systems related to energy managements in case not only normal driving but also shipping, exhibition display, further more low energy and so on.
The `TerminalClmpModReq1` is the final target mode of power supply and driving motion power generated by Terminal Clamp Manager from view of driver's requests including remote engine starter. Power Manager and powertrain shall control power supply mode and driving motion power mode by taking into account it.
The `PwrSplySts1` is defined as current power supply mode in Power Manager. It's used by Terminal Clamp Manager to observe the current status of power supply mode.
The `PtFlgEngRun1` is status whether engine is running or not. It is used by Terminal Clamp Manager to observe the current status of driving motion power mode.

Power Manager and Powertrain shall be informed each mode status of operation in `PwrSplySts1` and `PtFlgEngRun1` to `Terminal Clamp Manager` as feedback data.

Operation Mode Manager
It provides a car operation mode related to power supply and traction drive. It could be seen as abstracted car operation mode.
In the following picture are detailed all the possible levels of vehicle operation states ("OperModeSts1") and the possible transition:



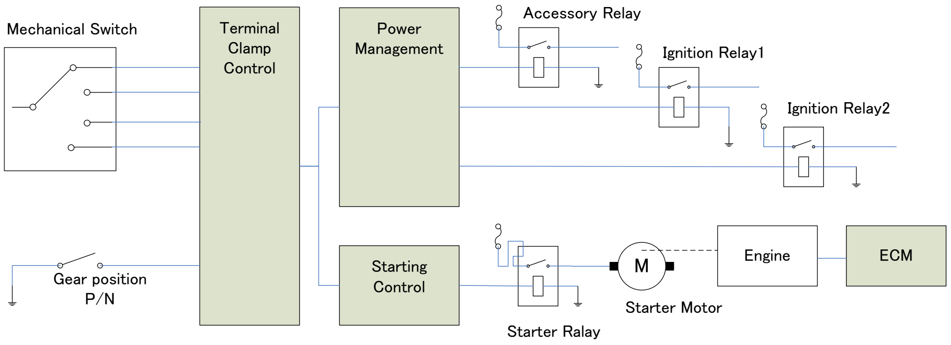
TerminalClampControl Fig.2 : **OperModeSts1** levels and transition

4.10.2 Use case descriptions

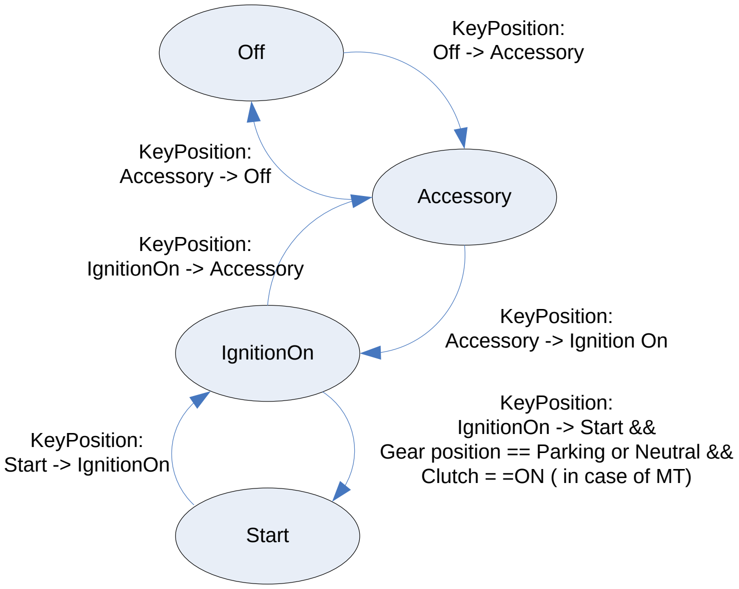
To understand more detail, 4 types of use cases are shown below.

4.10.2.1 Mechanical Cylinder with Accessory

This is an example in case of mechanical cylinder with Accessory position. In this use case, `TerminalClmpModReq1` states changes as transition of keypositions. To change from “IgnitionOn” to “Start”, gear position and clutch pedal (in case manual transmission) shall be taken care.



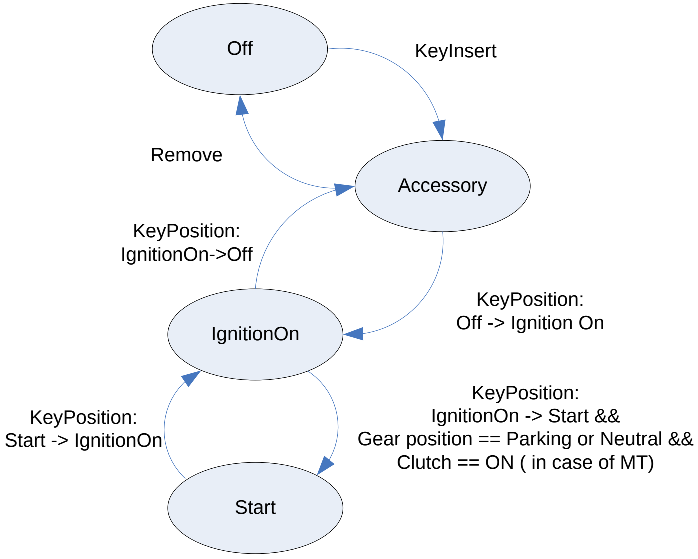
TerminalClampControl Fig.3



TerminalClampControl Fig.4

4.10.2.2 Mechanical Cylinder without Accessory

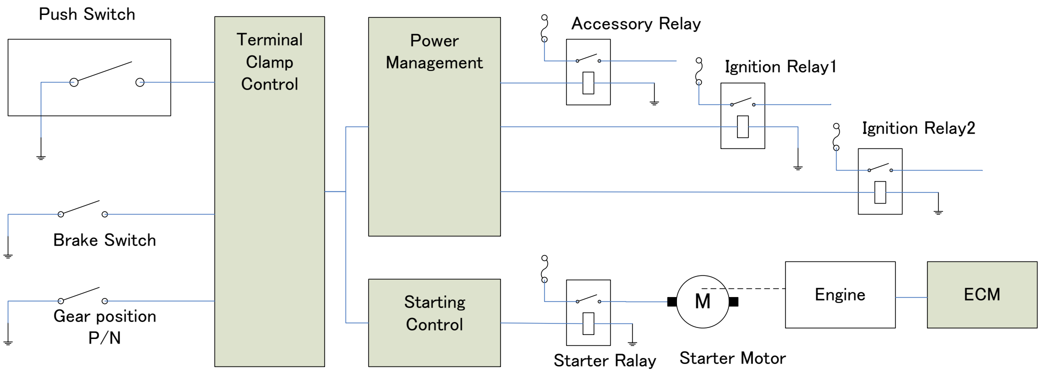
This is an example in case of mechanical cylinder without Accessory position. In this use case, `TerminalClmpModReq1` changes to “Accesory” when a key is inserted, and after a key removed, state changes to “Off” Others are same as in case of Mechanical Cylinder with Accessory.



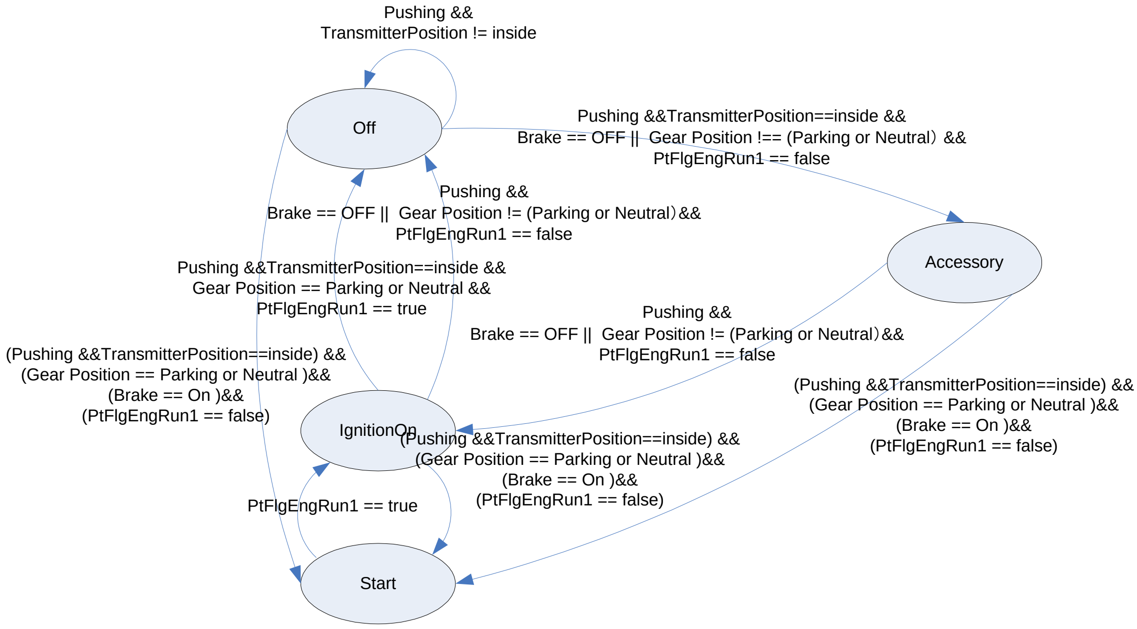
TerminalClampControl Fig.5

4.10.2.3 Push Start System

This is an example in case of Push Start System. In this use case every transitions of `TerminalClmpModReq1` are done by event caused by pushing start/stop. Transmitter Position that means driver has the transmitter shall be taken care because of antitheft issues. Only transmitter position is in vehicle, `TerminalClmpModReq1` states changes from"Off" to other states.
To change from "Off" or "Accessory" to "IgnitionOn", it is required to confirm if GearPosition is Parking or Neutral and braking pedal switch is ON It changes to Start directly as soon as safety confirmed.In case of others, it changes to "Accessory".



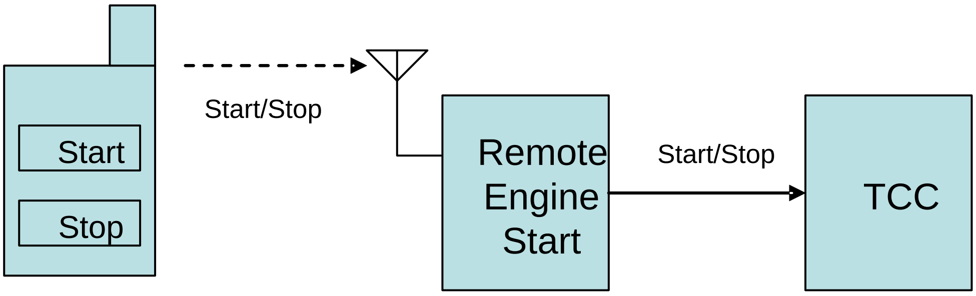
TerminalClampControl Fig.6


Pushing &&
TransmitterPosition != inside
Off
Pushing &&TransmitterPosition==inside &&
Brake == OFF || Gear Position !== (Parking or Neutral) &&
PtFlgEngRun1 == false
Accessory
Pushing &&
Brake == OFF || Gear Position != (Parking or Neutral)&&
PtFlgEngRun1 == false
Pushing &&TransmitterPosition==inside &&
Gear Position == Parking or Neutral &&
PtFlgEngRun1 == true
Pushing &&
Brake == OFF || Gear Position != (Parking or Neutral)&&
PtFlgEngRun1 == false
(Pushing &&TransmitterPosition==inside) &&
(Gear Position == Parking or Neutral)&&
(Brake == On)&&
(PtFlgEngRun1 == false)
IgnitionOn
(Pushing &&TransmitterPosition==inside) &&
(Gear Position == Parking or Neutral)&&
(Brake == On)&&
(PtFlgEngRun1 == false)
(Pushing &&TransmitterPosition==inside) &&
(Gear Position == Parking or Neutral)&&
(Brake == On)&&
(PtFlgEngRun1 == false)
PtFlgEngRun1 == true
Start

TerminalClampControl Fig.7

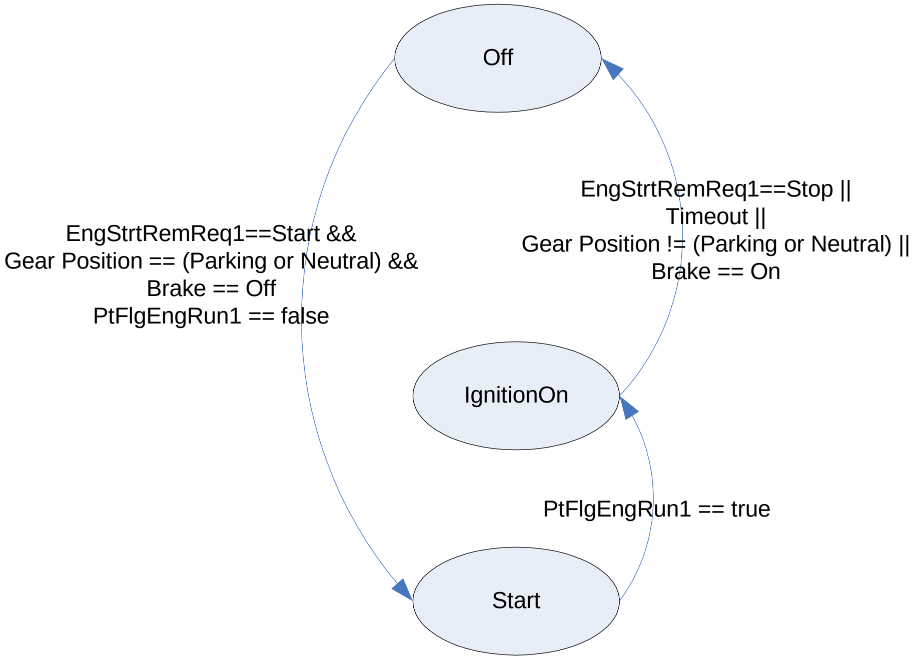
4.10.2.4 Remote Engine Start System

This is an example in case of Remote Engine Start System. This system enables to start engine from remote place using wireless communication.



TerminalClampControl Fig.8

Remote Engine Start System has to take care of security and antitheft issue before start engine and during engine running. In case everything is satisfied with security and antitheft issue, Remote Engine Start System requests Terminal Clamp Control to change state to Start. After starting engine system, it changes to “Off" in case Remote Engine Start System request to change to “Off”, or timeout defined by system is fired, or safety situation is changed by such as driver’s operation unexpected.



TerminalClampControl Fig.9

4.11 Immobilizer

Immobilizer represents an anti-theft system, which can prevent the vehicle’s engine from starting. Immobilizer receives the driver’s intentions, senses the environment and controls the behavior of immobilizer sources in the vehicle.

In case of any attempt for unauthorized engine start, immobilizer requests for visual and acoustical alarms to be generated.

All these decisions are taken based on the status of user provided key authorization.

Following input interfaces of the Immobilizer decomposition exists:

- PaSE, with key status
- RKE, with key status
- Terminal Clamp Control, with operation mode status
- Steering Column, with lock state information
- Central locking, with status of central locking request
- Inter-domain Interfaces

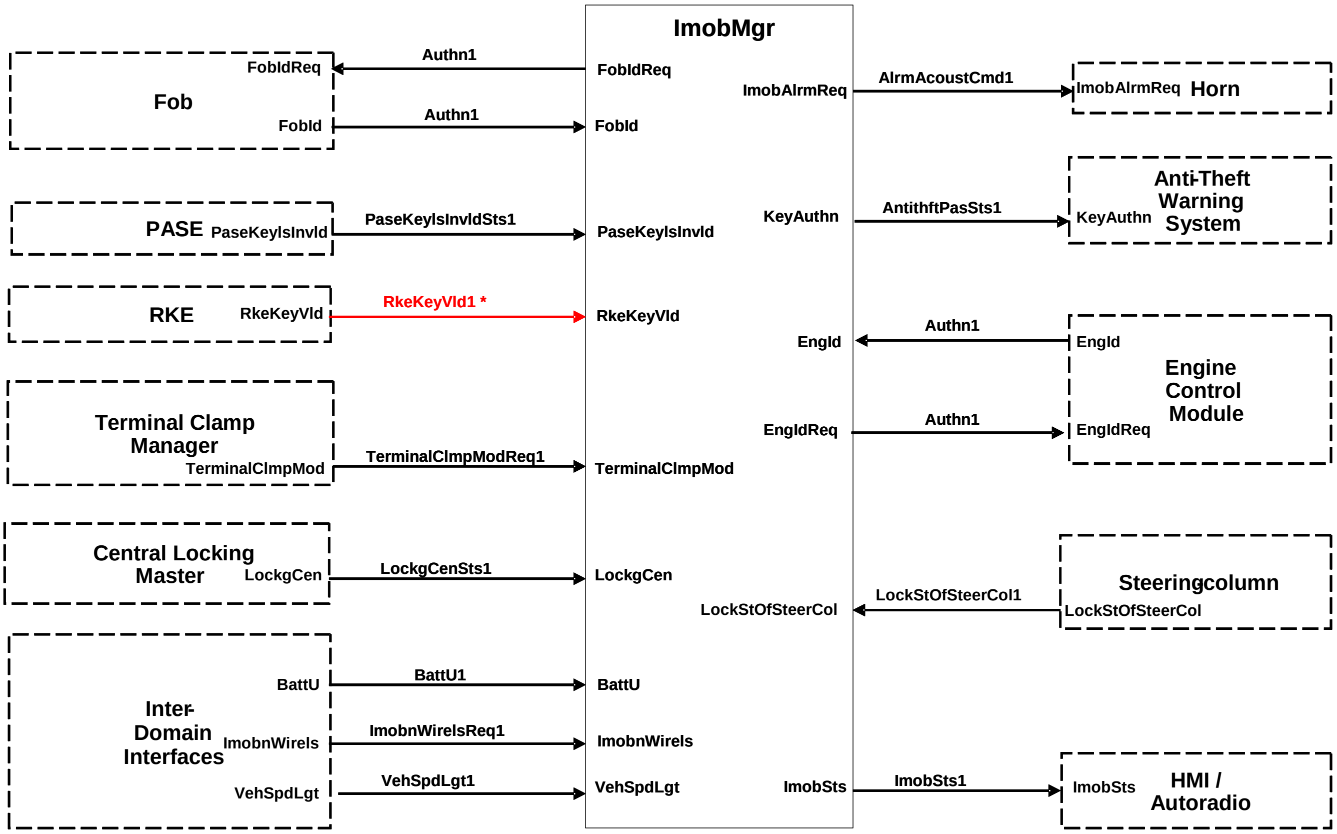
Following outputs interfaces of the Immobilizer decomposition exists:

- Horn, for acoustic alarm operation
- ATWS, to provide key authentication status operation
- HMI, for Immobilizer status display

Following interfaces of the Immobilizer decomposition exists:

- Immobilizer with Fob (with transponder),for getting key code and Fob status
- Immobilizer with ECM (Engine Control Module), for key learning and verification operation

Immobilizer Decomposition



***Note:** * - This interface is not fully defined for R4.0 Rev3*

Immobilizer Fig 1

The core functionality of Immobilizer is implemented within the Immobilizer Manager. It incorporates the basic functionalities like key learning, learnt key verification, Immobilizer diagnostics etc…

In normal mode operation, Immobilizer Manager shall verify the key fob depending on the status of dependent modules and Immobilizer's environment.

The Immobilizer status will be provided to

- Horn, to arm/diarm the acoustic alarm indication.
- ATWS, to process user provided key authentication and exterior light for warning indications.
- HMI, to display Immobilizer status.

The data shared between Fob, Immobilizer and ECM shall be encrypted to ensure security operations.

ATWS shall use the interface *"AntithftPasSts1"* to generate a visual alarm with Exterior Light (AlmVisCmd1) module.

The interface *"AlrmAcoustCmd1"* shall be used to generate an acoustic alarm command with Horn Control, if the vehicle has no ATWS.

4.11.1 Constraints/Limitations

1. The ports/interfaces between Fob, Immobilizer Manager and ECM use an array type data structure to provide flexibility and ensure signal security (e.g. key learning, key verification).
2. Exchange of encrypted data between Fob, Immobilizer and ECM is possible using the array structure. E.g. Possibility of vehicle theft by replacement of immobilizer, Fob and ECM shall be reduced.
3. Decomposition of Fob (key with transponder) is not part of this immobilizer decomposition.
4. Timing constraints for the Immobilizer system are implementation specific, like
   - Inter bit / byte timing for commands.
   - Timeout implementation for command/Request and the Status.
   - End of Immobilizer request and start of Fob response.
   - End of Immobilizer response and start of ECM new request.
5. Transponder type used has no impact on Immobilizer decomposition architecture.
6. Detailed synchronization of Immobilizer SW-C specification with Remote Keyless Entry, Passive Access System SW-C specification is to be completed.

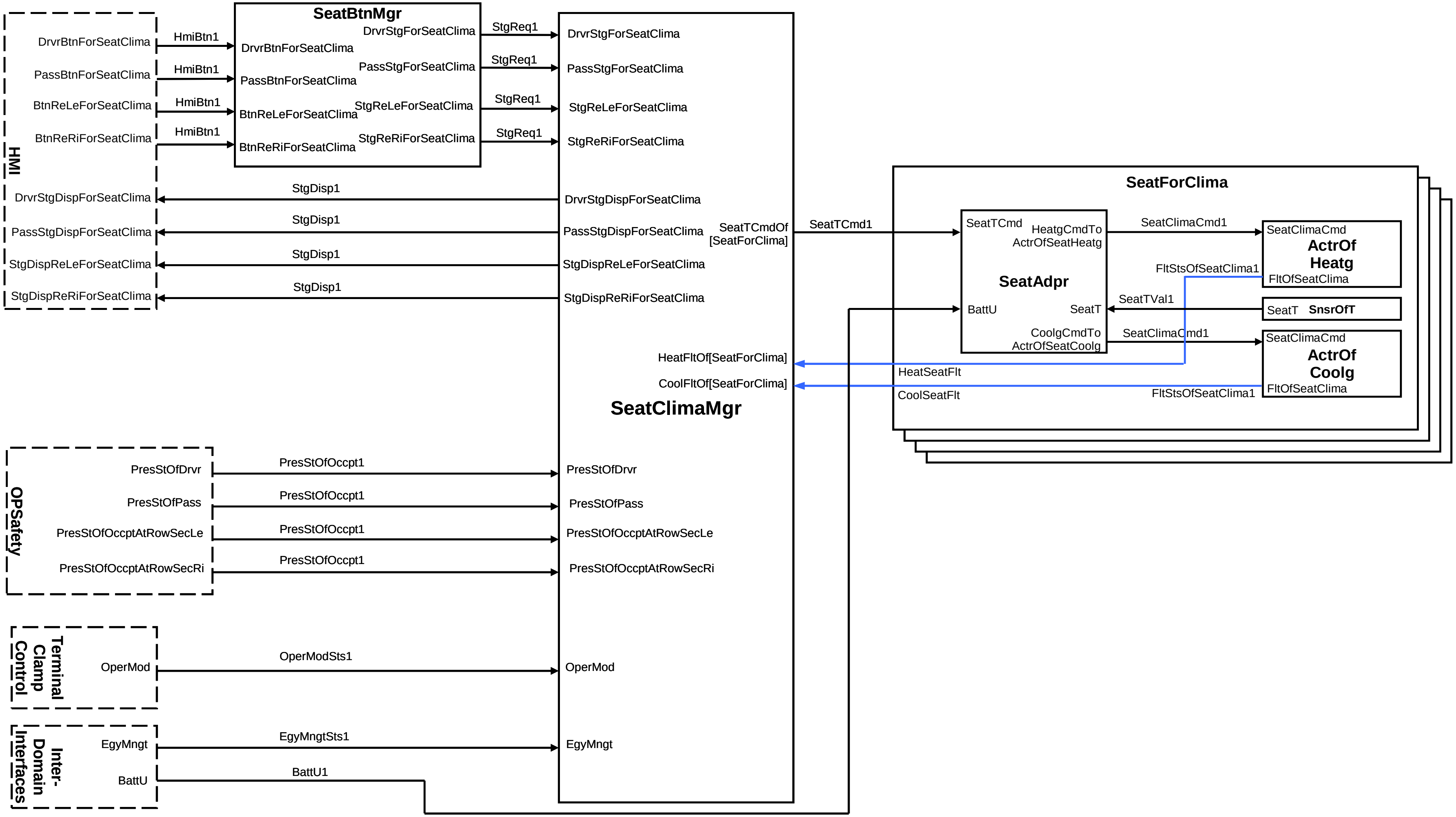
4.12 Seat Climatization

4.12.1 Introduction

The Seat Climatization functionality provides the possibility of regulating the temperature of the seats inside the car.
Regulations for both heating and cooling of the seats are included.

4.12.2 Seat Climatization Decomposition

In this section is provided the hierarchical decomposition of the Seat Climatization functionality divided into Manager SW-C, Sensor and Actuator SW-Cs.
Arrows indicate data flows.
In the decomposition are present also feedback signals from actuator indicating the status/fault of driver.
This decomposition is taking car only of the seat climatization of main four seats in a car (driver, front passenger, left and right rear passengers).



SeatClimatization Fig 1

DESCRIPTION

- **SeatButtonManager**
  - ➢ Converts the switch status (on/off) into requests (stage 1,2,…). → Heating or cooling

- **SeatClimatizationManager**
  - ➢ Receives the requests from the user (ReqStage).
  - ➢ Generates the required temperature to the adapter ([OFF, AdapterTemperatureLevel1, AdapterTemperatureLevel2,…]) , depending on the user request, the occupation of the seat, and the car state (energy management and operation mode).
  - ➢ Provides a status of the Seat Climatization to the HMI display.

- **SeatForClimatization**
  - ➢ Controls the Seat Climatization and is furtherly decomposed into Adapter SWCs and Actuator SWCs.

- **Adapter**
  - ➢ Controls the cooling/heating element(s) in order to reach the requested temperature, measured from one or several sensors in the seat.

- **Actuator**
  - ■ Cooling (e.g. fan) or heating (e.g. mat) element.
  - ■ Reports any fault detected (short circuit, open load,…).
  - ■ Each actuator can control several physical elements (for seat, backrest,…).

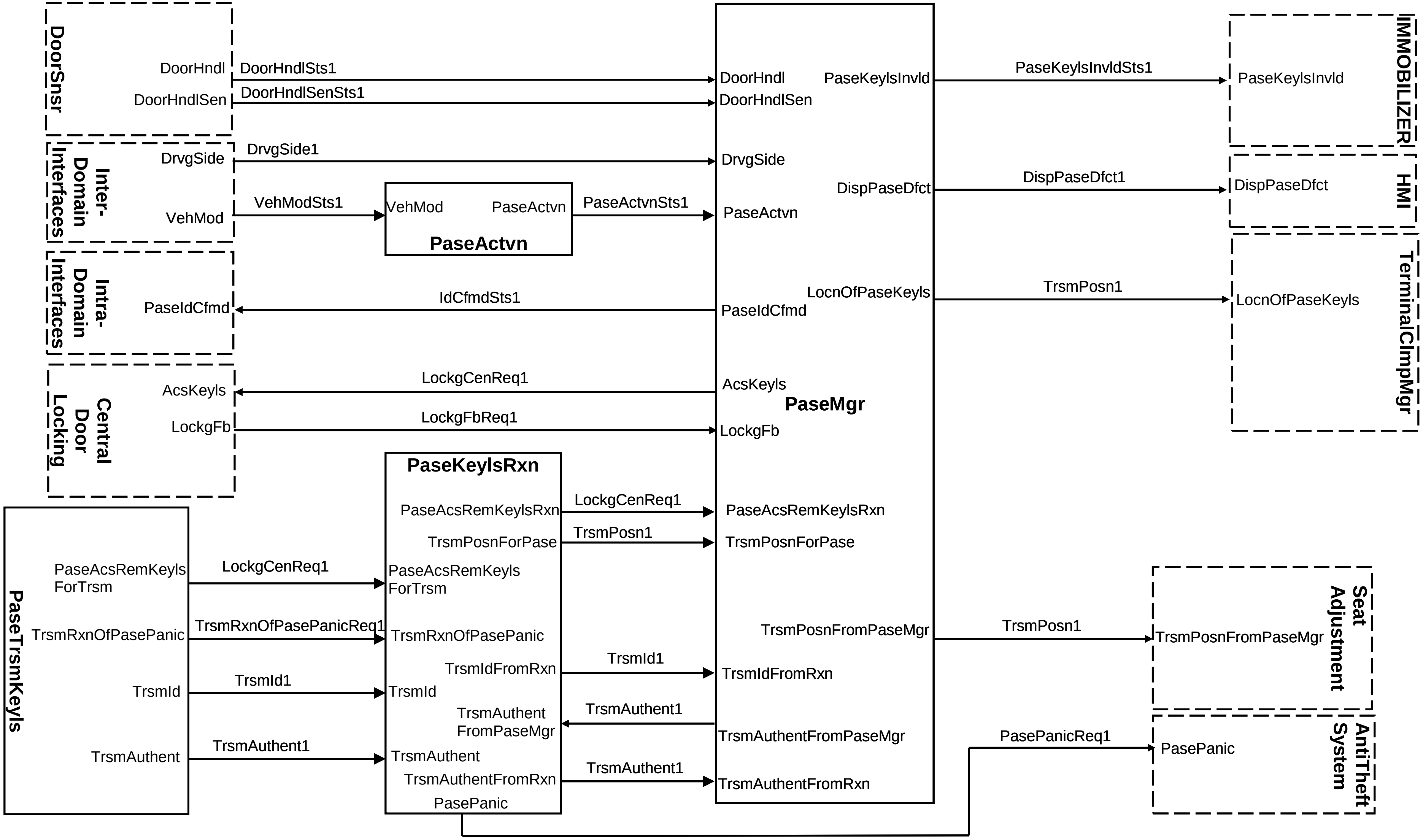
4.13 PASE (PASsive Entry)

4.13.1 Introduction

The Passive entry functionality represents a locking system which can help the user to enter his car without using his key physically. It requests CentralLockingMaster to lock or unlock the car based on Remote Card Id information and Doors sensors information.

4.13.2 PASE Decomposition

Here in the following is provided the SWCs decomposition for PASE functionality.



PassiveEntry Fig1

SW component emitter	Port emitter	Interface	Port receiver	SW component receiver	Comments/justification
Central-Door-Locking	LockgFb (LockingFeedback)	LockgFbReq1 (LockingFeedbackRequest)	LockgFb (LockingFeedback)	PaseMgr (PassiveEntry siveEntryManager)	Information needed to know for the following reasons: 1/ needed to know if the order sent by PASE has been taken into account 2/ needed to know if another order of locking or unlocking has been sent (e.g. synchronization of PASE with mechanical key)
DoorSnsr (Door sensor)	DoorHndl (DoorHandle)	DoorHndlSts1 (DoorHandleStatus)	DoorHndl (DoorHandle)	PaseMgr (PassiveEntry siveEntryManager)	In case of a pull detector implemented on the door handle
DoorSnsr (Door sensor)	DoorHndlSen (DoorHandleSensing)	DoorHndlSenSts1 (DoorHandleSensingStatus)	DoorHndlSen (DoorHandleSensing)	PaseMgr (PassiveEntry siveEntryManager)	In case of a touched sensor implemented on the door handle
Inter-Domain-Interfaces	VehModSts (VehicleModeStatus)	VehMod1 (VehicleMode)	VehModSts (VehicleModeStatus)	PaseActvn (PassiveEntryActivation)	This information is needed to know if the vehicle is in factory mode, show room mode, customer mode
PassiveEntryActivation	PaseActvn (PaseActivation)	PaseActvnSts1 (PaseActivationStatus)	PaseActvn (PaseActivation)	PaseMgr (PassiveEntry siveEntryManager)	Activate or deactivate considering vehicle mode information
PaseMgr (PassiveEntryManager)	AcsKeyls (AccessKeyless)	LockgCenReq1 (LockingCentralRequest)	AcsKeyls (AccessKeyless)	Inter-Domain-Interfaces	Needed to know if a command of locking or unlocking is coming from the Keyless and if it shall be taken into account by PASE

SW component emitter	Port emitter	Interface	Port receiver	SW component receiver	Comments/justification
PaseKeylsRxn (PassiveEntryKeylessReceiption)	TrsmPosnForPase (TransmissionPositionForPassiveEntry)	TrsmPosn1 (TransmissionPosition)	TrsmPosnForPase (TransmissionPositionForPassiveEntry)	PaseMgr (PassiveEntry siveEntryManager)	Needed to know the position of all transmitters such as - Near or far from the car - rear or front of the car - right or left side of the car - Inside or Outside the car
PaseTrsmKeyls (PassiveEntryTransmissionKeyless)	TrsmId (TransmissionIdentifier)	TrsmId1 (TransmissionIdentifier)	TrsmId (TransmissionIdentifier)	PaseKeylsRxn (PassiveEntryKeylessReception)	Needed to know the ID of all transmitters
PaseMgr (PassiveEntryManager)	DispPaseDfct (DefectDisplayPassiveEntry)	DispPaseDfct1 (DisplayPassiveEntryDefect)	DispPaseDfct (DefectDisplayplayPassiveEntry)	HMI	Display of PASE defects
PaseMgr (PassiveEntryManager)	LocnOfPaseKeyls (LocationOfPaseKeyless)	LocnOfPaseKeylsSts1 (LocationOfPaseKeylessStatus)	LocnOfPaseKeyls (LocationOfPaseKeyless	Inter-Domain-Interfaces	Needed to know if the Keyless is inside or outside the car
PaseTrsmKeyls (PassiveEntryTransmissionKeyless)	PaseAcsRemKeylsForTrsm (PaseAccessRecessRemoteKeylessForTransmission)	LockgCenReq1 (CentralLockingRequest)	PaseAcsRemKeylsForTrsm (PaseAccessRemoteKeylessForTransmission)	PaseKeylsRxn (PassiveEntryKeylessReception)	Request of locking or unlocking from the transmitter of PASE

SW component emitter	Port emitter	Interface	Port receiver	SW component receiver	Comments/justification
PaseTrsmKeyls (PassiveEntryTransmissionKeyless)	TrsmAuthent (TransmissionAuthentication)	TrsmAuthent1 (TransmissionAuthentication)	TrsmAuthent (TransmissionAuthentication)	PaseKeylsRxn (PassiveEntryKeylessReceiption)	Authentification process between transitter and receiver (device inside the car): Request from the transmission device (Password sent)
PaseKeylsRxn (PassiveEntryKeylessReceiption)	PaseAcsRemKeylsRxn (AccessRemotePaseKeylessReception)	LockgCenReq1 (CentralLockingRequest)	PaseAcsRemKeylsRxn (AccessRemotePaseKeylessReception)	PaseMgr (PassiveEntry siveEntryManager)	Request of locking or unlocking from the transmitter to receiver of PASE
PaseKeylsRxn (PassiveEntryKeylessReceiption)	TrsmIdFromRxn (TransmissionIdenitifierFromReception)	TrsmId1 (TransmissionIdenitifier)	TrsmIdFromRxn (TransmissionIdenitifierFromReception)	PaseMgr (PassiveEntry siveEntryManager)	The ID of the Pase keyless device is provided to the receiver and from the receiver to the PASE
PaseKeylsRxn (PassiveEntryKeylessReceiption)	TrsmAuthentFromRxn (TransmissionAuthenticationFromReception)	TrsmAuthent1 (TransmissionAuthentication)	TrsmAuthentFromRxn (TransmissionAuthenticationFromReception)	PaseMgr (PassiveEntry siveEntryManager)	Authentification process between receiver and PASE manager: The request originally from the transmission device is sent by the receiption device to PaseMgr (Password sent)

SW component emitter	Port emitter	Interface	Port receiver	SW component receiver	Comments/justification
PaseMgr (PassiveEntryManager)	TrsmAuthentFromPaseMgr (TransmissionAuthenticationFromPaseManager)	TrsmAuthent1 (TransmissionAuthentication)	TrsmAuthentFromPaseMgr (TransmissionAuthenticationFromPaseManager)	PaseKeylsRxn (PassiveEntryKeylessReception)	Authentification process between receiver and PASE manager This information contains - a staus of the process: Ongoing, finished - a status of the authentification process (succeful or failed) - if successful, an answer ciphered to the request of authentification
PaseMgr (PassiveEntryManager)	PaseIdCfmd (PaseIdentifierComfirmed)	IdCfmdSts1 (IdentifierComfirmedStatus)	PaseIdCfmd (PaseIdentifierComfirmed)	Inter-Domain-Interfaces	This information is needed to Confirm to other functionalities that the PASE emmitter is known
PaseMgr (PassiveEntryManager)	PaseKeylsInvld (PaseKeylessInvalid)	PaseKeylsInvldSts1 (PaseKeylessInvalidStatus)	PaseKeylsInvld (PaseKeylessInvalid)	Inter-Domain-Interfaces	Needed to know if the pase Key is valid or not
PaseKeylsRxn (PassiveEntryKeylessReception)	PasePanic (PasePanic)	PasePanicReq1 (PasePanicRequest)	PasePanic (PasePanic)	Inter-Domain-Interfaces	Needed to transfer the request of the customer from PASE manager to interdomain to have a Panic Request

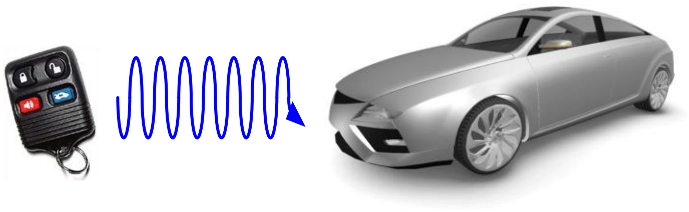
SW component emitter	Port emitter	Interface	Port receiver	SW component receiver	Comments/justification
PaseTrsmKeyls (PassiveEntry-Transmission-Keyless)	TrsmRxnOfPasePanic (TransmissionReceptionOfPasePanic)	TrsmRxnOfPasePanicReq1 (TransmissionReceptionOfPasePanicRequest)	TrsmRxnOfPasePanic (TransmissionReceptionOfPasePanic)	PaseKeylsRxn (PassiveEntryKeylessReceiption)	Needed to receive the panic request of the customer delivered by PASE transmitter
PaseMgr (PassiveEntryManager)	TrsmPosnFromPaseMgr (Transmission-PositionFrom-PaseManager)	TrsmPosn1 (TransmissionPosition)	TrsmPosnFromPaseMgr (TransmissionPositionFromPaseManager)	Inter-Domain-Interfaces + Seat Adjustment	Needed to know the position of all transmitters such as - Near or far from the car - rear or front of the car - right or left side of the car This interface is used to communicate the information through the system
Inter-Domain-Interfaces	DrvgSide (DrivingSide)	DrvgSide1 (DrivingSide)	DrvgSide (DrivingSide)	PaseMgr (PassiveEntry siveEntry-Manager)	

4.14 RKE (Remote Keyless Entry)

4.14.1 Introduction

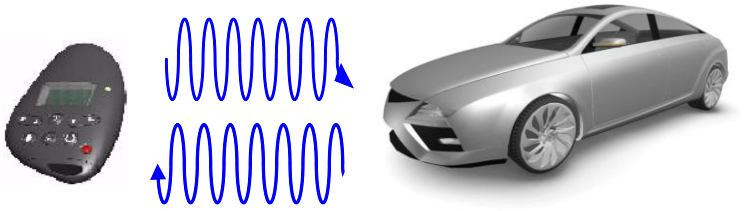
Remote keyless entry (RKE) system is a system designed to remotely lock or unlock access to automobiles.

- RKE transmission requires two components - a transmitter and a receiver.
  - Transmitter - RKE key fob, other ID device with RKE integrated
  - Receiver - Body Control ECU, other ECU with integrated RKE
- RKE operates by broadcasting radio waves on a particular frequency unidirectionally.
- RKE systems implement encryption and rolling code algorithms to prevent car thieves from intercepting and spoofing the telegrams.
- Typical RKE functions are:
  - lock
  - unlock
  - alarm activation/deactivation
  - approach light
  - comfort Open (open all windows)
  - comfort Close (close all windows)



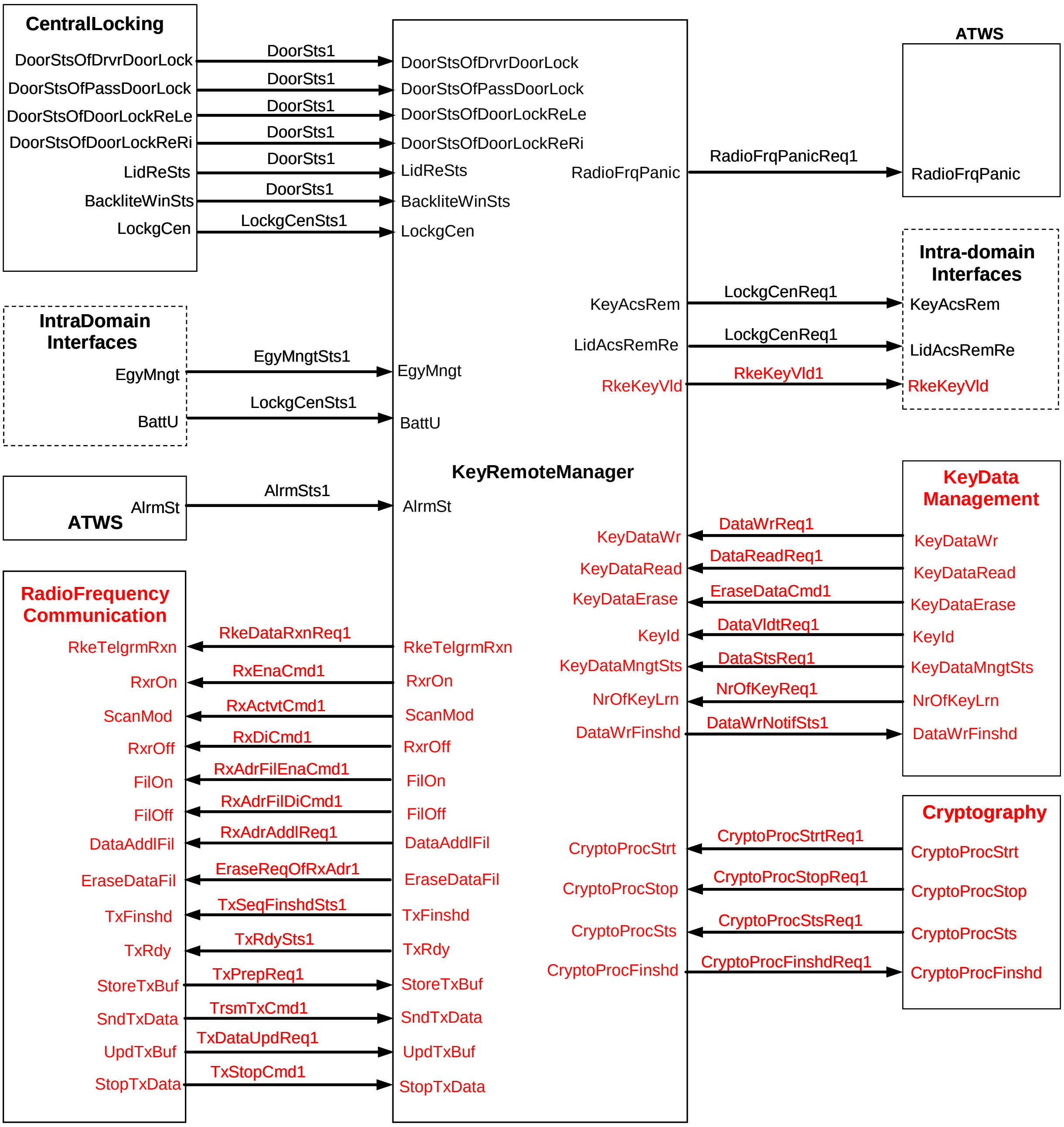
RemoteKeylessEntry Fig 1

- A BiCom system is the extention of the unidirectional RKE to bidirectional RKE system.
    - keyfob -> car (like a unidirectional RKE system)
    - car -> keyfob (car status information for displaying on the keyfob by LED or display)
- Typical BiCom status information transmitted to the keyfob are:
    - door status
        - window status
        - alarm status
        - service information
            - car battery info
            - fuel status
            - oil status



RemoteKeylessEntry Fig 2

4.14.2 RKE Decomposition



RemoteKeylessEntry Fig 3

KeyRemoteManager:

RKE Manager can be theoretically splitted into two main sections (not highlighted in the decomposition):

- protocol evaluation module

  Mainly provide exchanging information with Cryptography and KeyData-Management SWCs.

  This part will be enhanced in a future version

- command evaluation module

- comfort handling
- command execution precondition check
- command execution trigger (e.g. CL, CAN, ...)
- remark: locking feedback will be requested by CL (flasher, ...)
- id filtering control
- resynchronization
- learning
- learning mode detection
- learning procedure
- diagnosis
- store diagnosis relevant data
- provide diagnosis relevant functions

Justification for this architectural design is, to make a change to a other RKE system easier. All protocol and RKE system relevant parts are located in the protocol evaluation module. The command evaluation module is getting the extracted data in a abstracted form. For changing a RKE system (protocol, cryptograpy, telegram definition) but to keeping the command behavior, only the protocol evaluation module shall be replaced.
The AUTOSAR RTE could be located between both RKE modules.

Cryptography:
For encryption/decryption of RKE/Bicom telegram, following functions will be consumed:

- Cryptography Process Start
- Cryptography Process Stop
- Cryptography Process Status
- Cryptography Process Finished

KeyDataManagement :
A separate module is managing the key data. This means it controls writing in a non volatile memory and provides some service functions like:

- status of the key data management module
- write key data (notification callback must be provided by RKE)
- read key data
- erase key data
- validate the key ID
- number of learned keys

Radio Frequency Communication:
In this module are collected the following parts of the functionality:

- Receiver mode
- Filtering operation
- Transmissioi buffer activity

4.15 Convertible Control

4.15.1 Introduction

Convertible control represents the feature to open or close a roof for a convertible car. The roof can be a hard or soft roof, with no disctinction

Only one Manager composes the convertible control decomposition.

The different inputs of the decomposition are the following:

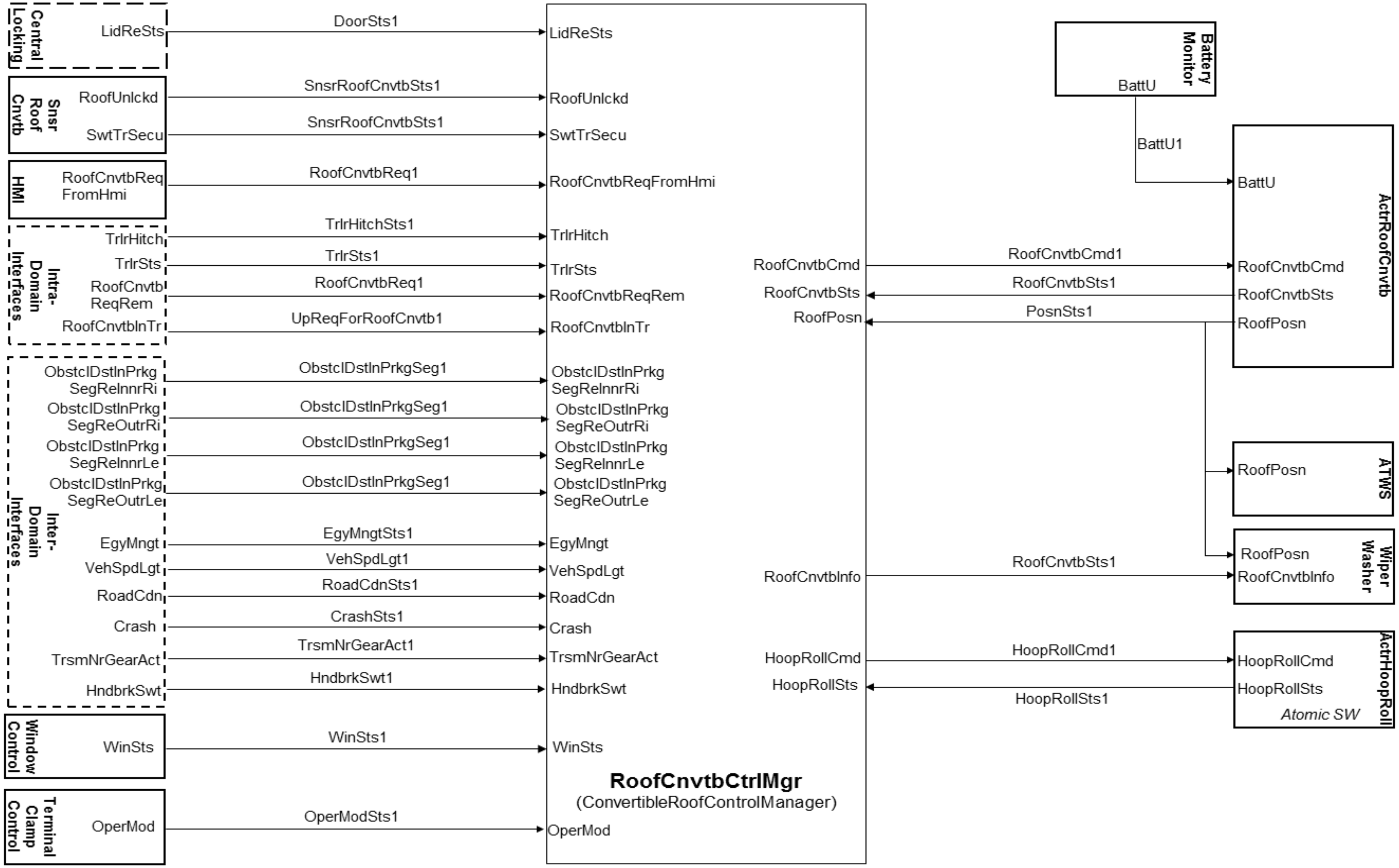
- Central door locking, to check if the door / roof can be opened
- Convertible sensor, to check if the roof is unlocked (in case of manual locking system), and if the roof switch is also locked
- HMI, to deal with convertible command and the wish to up / down the roof in the trunk
- Remote key, to open the roof outside of the car
- Parking Aid, to check if an obstacle is near the trunk. In this case, it will not be always possible to open the roof
- Intradomain, with trailer hitch
- Interdomain, for Energy management, vehicule speed condition, Road condition and crash status
- Powertrain, to check which gear is engaged, and if the handbrake is locked
- Terminal Clamp control, for operation mode
- Rool hoop status
- Window control, for Window status
- Convertible actuator
- Battery monitor (Future function)

The outputs are :

- Roof convertible command, to command the roof
- Wiper Washer and ATWS, to provide a status of the operation progress.

4.15.2 Convertible Control decomposition

Here in the following is provided the SWCs decomposition for Convertible Control functionality.



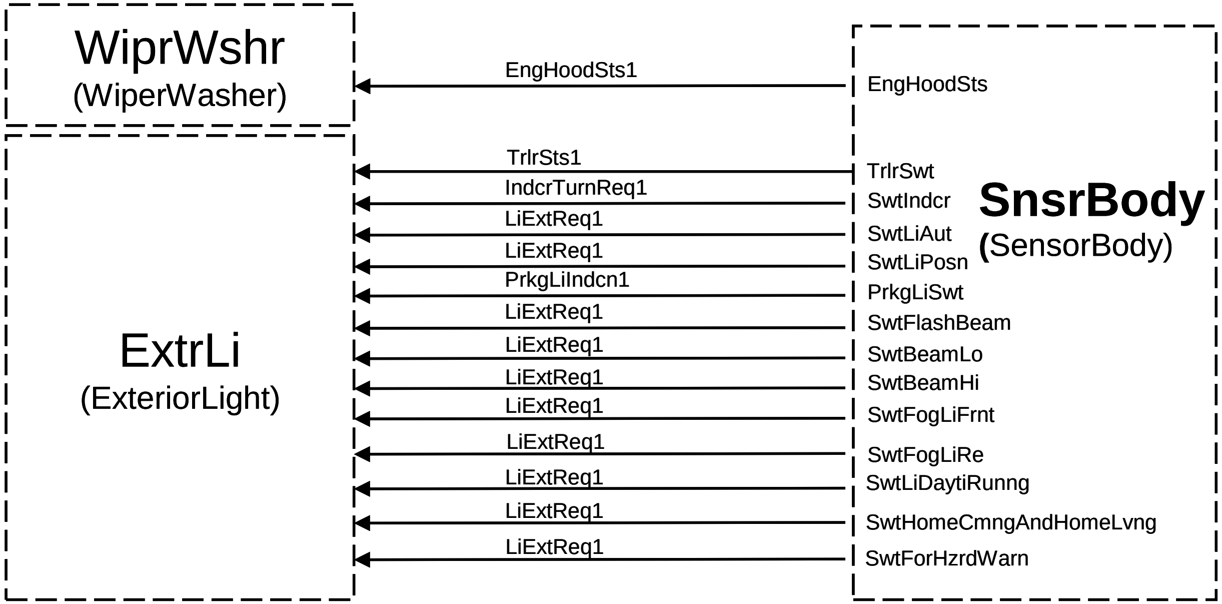
ConvertibleControl Fig 1

4.16 BodySensors

4.16.1 Introduction

The BodySensors functionality represents the ability of the system to sense the environment.

4.16.2 BodySensors decomposition



BodySensors Fig 1

SW component emitter	Port emitter	Interface	Port receiver	SW component receiver	Comments/justification
SnsrBody (SensorBody)	EngHoodSts (EngineHoodStatus)	EngHoodSts1 (EngineHoodState1)	EngHoodSts (EngineHoodStatus)	WiprWshr (WiperWasher)	This interface represents the status of the Engine Hood: Open or closed
SnsrBody (SensorBody)	TrlrSwt (TrailerSwitch)	TrlrSts1 (TrailerStatus1)	TrlrSwt (TrailerSwitch)	ExtrLi (ExteriorLight)	Detection of presence of a trailer by switch type detection. Information: Trailer attached 0 means 'not present' 1 means 'present'
SnsrBody (SensorBody)	SwtIndcr (Turn- (TurnSig- SignalLeverSwitch)	IndcrTurnReq1 (IndicatorTurnRequest1)	SwtIndcr SignalLeverSwitch)	ExtrLi (ExteriorLight)	Demand to activate/ deactivate left or right indicator blinking or temporary blinking left / right ("Autobahnblinken"). As long as direction indication is requested, the corresponding value should be set.
SnsrBody (SensorBody)	SwtLiAut (AutomaticlightSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtLiAut (AutomaticlightSwitch)	ExtrLi (ExteriorLight)	exterior lights request from an automatic mode
SnsrBody (SensorBody)	SwtLiPosn (PositionlightSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtLiPosn (PositionlightSwitch)	ExtrLi (ExteriorLight)	exterior lights request on position light
SnsrBody (SensorBody)	PrkgLiSwt (ParkingLightSwitch)	PrkgLiExtReq1 (ParkingLightExteriorRequest)	PrkgLiSwt (ParkingLightSwitch)	ExtrLi (ExteriorLight)	exterior lights request on parking light
SnsrBody (SensorBody)	SwtFlashBeam (FlashBeamSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtFlashBeam (FlashBeamSwitch)	ExtrLi (ExteriorLight)	exterior lights request on flash beam
SnsrBody (SensorBody)	SwtBeamLo (LowBeamSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtBeamLo (LowBeamSwitch)	ExtrLi (ExteriorLight)	exterior lights request : low beam
SnsrBody (SensorBody)	SwtBeamHi (HighBeamSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtBeamHi (HighBeamSwitch)	ExtrLi (ExteriorLight)	exterior lights request : high beam
SnsrBody (SensorBody)	SwtFogLiFrnt (FrontFoglightSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtFogLiFrnt (FrontFoglightSwitch)	ExtrLi (ExteriorLight)	exterior lights request : Front Fog Light
SnsrBody	SwtFogLiRe	LiExtReq1 (LightExte-	SwtFogLiRe	ExtrLi	exterior lights request : Rear Fog Light

SW component emitter	Port emitter	Interface	Port receiver	SW component receiver	Comments/justification
(SensorBody)	(RearFoglightSwitch)	riorRequest1)	(RearFoglightSwitch)	(ExteriorLight)	
SnsrBody (SensorBody)	SwtLiDaytiRunng (DaytimeRunninglightSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtLiDaytiRunng (DaytimeRunninglightSwitch)	ExtrLi (ExteriorLight)	exterior lights request : Day time Light
SnsrBody (SensorBody)	SwtHomeCmngAndHomeLvng (ComingHomeLeavingHomeSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtHomeCmngAndHomeLvng (ComingHomeLeavingHomeSwitch)	ExtrLi (ExteriorLight)	exterior lights request : Coming home and leaving home light
SnsrBody (SensorBody)	SwtForHzrdWarn (HazardWarningSwitch)	LiExtReq1 (LightExteriorRequest1)	SwtForHzrdWarn (HazardWarningSwitch)	ExtrLi (ExteriorLight)	exterior lights request :Hazard Warning light

4.17 KeyPad

See chapter 4.5 Central Locking

5 Additional Information (Optional)

This chapter shall contain information that doesn’t fit into the previous chapters.
Proper sub-titles needed.

This design standard is the AUTOSAR-architecture for Body domain functionalities. It provides a decomposition into SW-Cs and a list of standardized interfaces related to them.
Each decomposition is limited in granularity to sensor components, adapter components, the core functionality and actuator components. The intent is for the decomposition to get not to atomic SW-Cs but to “purchasable” SW-Cs. These SW-Cs will be obtained as a unit so that all the internal (and hence not AUTOSAR standardized) interfaces are controlled by a single vendor, even if there are SW-Cs within the bought unit that reside on different ECUs. All interfaces between SW-Cs from different vendors should be standardized.
In addition, all ports are described, showing their AUTOSAR data qualities. Invalidation (where needed) is defined as in-band invalidation. Init values are specified where appropriate e.g. “off”, “idle”, “undefined”, “unknown” … Where standardization is inappropriate, recommendations may be provided.