Document Title	Explanation of Application Interfaces of the Powertrain Domain
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	269
Document Classification	Auxiliary

Document Version	3.1.0
Document Status	Final
Part of Release	4.1
Revision	2

Document Change History

Date	Version	Changed by	Change Description
31.10.2013	3.1.0	AUTOSAR Release Management	• Updated/Rework of Chapter "Sensor/Actuator Design Pattern" • Update of Chapter "Appendix: Mapping Ports to Display Names - Powertrain Domain" according to changes in application interfaces model or patterns
01.03.2013	3.0.0	AUTOSAR Administration	• New: chapter describing Sensor Actuator Design Pattern • Update w.r.t. to obsolete names etc. according to changes in AISpecification etc.
09.12.2011	2.2.0	AUTOSAR Administration	• Splitting of document: Topic of Measurement and Calibration moved to new document TR_AIMeasurementCalibrationDiagnostics_537 • Update w.r.t. to names etc. according to changes in AISpecification
22.10.2010	2.1.0	AUTOSAR Administration	• Display names made consistent to AISpecification • Rule MCM390 added: Suffix should not exceed 3 char
07.12.2009	2.0.0	AUTOSAR Administration	• Added: overview figures of different torque interfaces • New: chapter about modeling aspects and automatic generation of display names • New: Mapping Ports to Display Names

Document Change History			
Date	**Version**	**Changed by**	**Change Description**
			• Legal disclaimer revised
23.06.2008	1.0.1	AUTOSAR Administration	• Legal disclaimer revised
15.11.2007	1.0.0	AUTOSAR Administration	• Initial Release

Disclaimer

Advice for users

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

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

Contents

1 Purpose of this Document

This document explains design decisions that lead to the standardized applications interfaces relevant to the Powertrain Domain.

The sensor actuator pattern described in this document is not specific to the powertrain domain but can be applied to other domains too, e.g. the chassis domain.

NOTE: If any information in diagrams or text (or conclusions drawn from them) conflict with the information in [2] or [3] or [3b] and this is not explicitly mentioned the information in [2] or [3] or [3b], resp., should be regarded as definitive.

2 References

[1] SW-C and System Modeling Guide
AUTOSAR_TR_SW-CModelingGuide

[2] Table of Application Interfaces
AUTOSAR_MOD_AITable

[3] XML Specification of Application Interfaces
AUTOSAR_MOD_AISpecification

[3b] Application Interfaces Examples
AUTOSAR_MOD_AISpecificationExamples

[4] Explanation of Application Interfaces of the Chassis Domain
AUTOSAR_EXP_AIChassisExplanation

[5] Unique Names for Documentation, Measurement and Calibration: Modeling and Naming Aspects including Automatic Generation
AUTOSAR_TR_AIMeasurementCalibrationDiagnostics

[6] Software Component Template
AUTOSAR_TPS_SoftwareComponentTemplate

[7] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[8] ANTLR parser generator V3
http://www.antlr.org

[9] Virtual Function Bus
AUTOSAR_EXP_VFB

[10] Glossary
AUTOSAR_TR_Glossary

3 Description of Terms and Concepts

3.1 Abbreviations

For abbreviations used in this document please refer to the keyword list in [2] (as .xls) and in [3] (as .arxml).

Additionally please also refer to [10] for explanation of commonly used terms and abbreviations within AUTOSAR.

3.2 Terminology – Torque within the Powertrain Domain

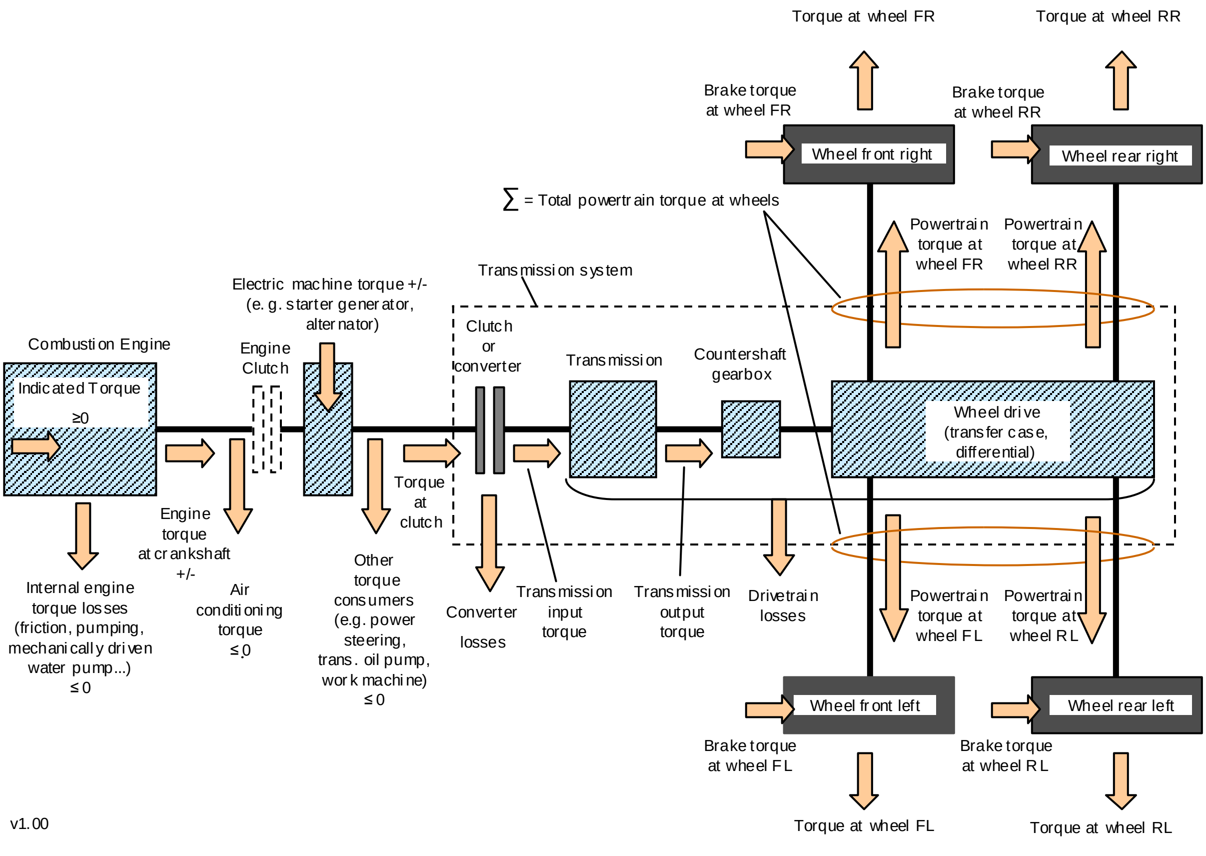

Figure 1: Powertrain Torque terminology

Sign definition for torque at clutch / torque at wheels:
Positive value means that torque is transmitted from the engine to the drivetrain / from the powertrain to the wheels.
Negative value means that torque is transmitted from the drivetrain to the engine / from the wheels to the powertrain.

Zero means that no torque is transmitted between engine and drivetrain / between wheels and powertrain.

Engine Clutch
For Hybrid Systems an additional clutch can be present between combustion engine and electric machine.

3.3 Terminology – Fast and Slow Torque Requests

Many torque request interfaces have the additional descriptors “Fast” or “Slow”.
These descriptors are relevant to gasoline spark ignition engines, whose torque output can be modified by means of throttle angle (and hence air mass) and ignition timing. In general, the torque output responds slowly to changes in throttle angle due to fluid dynamics in the manifold and cylinder head. The reaction to ignition timing changes is almost instantaneous, especially at higher engine speeds.
“Fast” refers to the “immediate” / “instant” torque request, typically achieved by ignition timing.
“Slow” refers to the longer term or “torque reserve” request, usually the input to throttle control.
Note that a gasoline engine running at optimum ignition timing cannot **increase** torque quickly as the throttle is the only means for the increase. However, pre-emptively opening the throttle and running with retarded ignition to maintain the the original (lower) torque allows the torque to be increased quickly by ignition a short time in the future. This operation is usually achieved by setting the “Slow” torque request to be greater than the “Fast” torque request to provide this “torque reserve”, allowing the torque to be rapidly increased by increasing the “Fast” request.

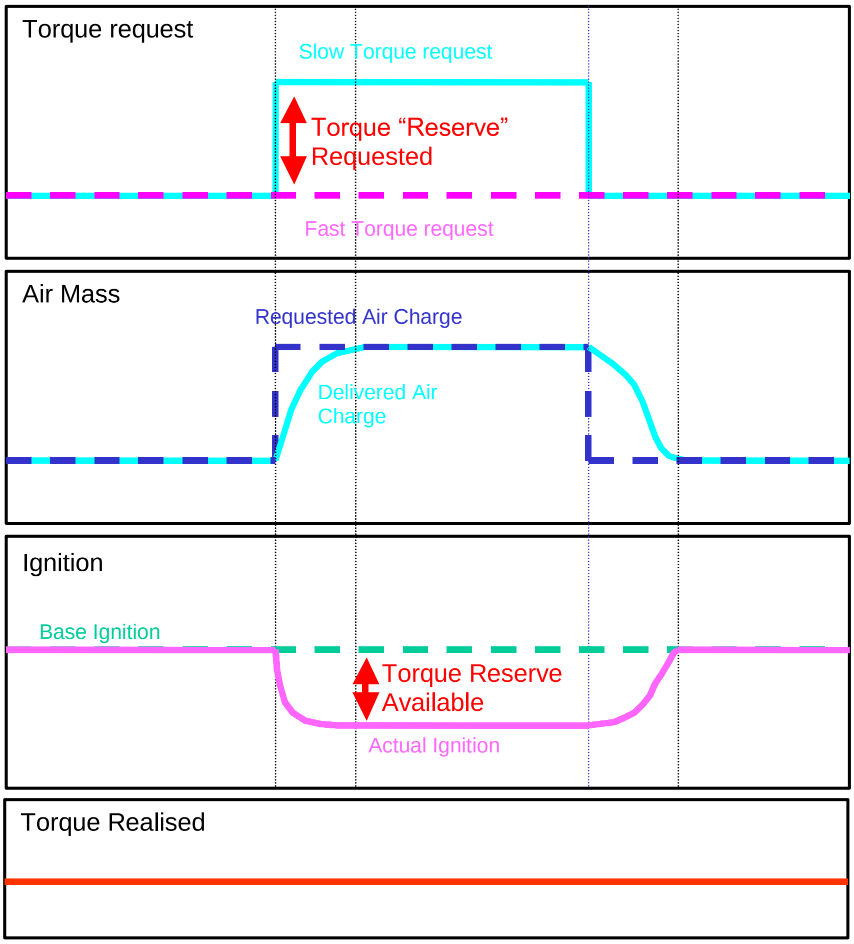

Figure 2: The Torque Reserve concept, with "Fast" and "Slow" torque requests

For conventional diesel engines only the fast torque interfaces are relevant. However, future diesel engines could have the possibility to use both fast and slow torque interfaces.

3.4 Overview of AUTOSAR torque application interfaces

Legend within Figure 3:
<ShortName of Powertrain Port>/ <ShortName of Chassis Port>]
<LongName of Port>
Note: Obsolete Interfaces from before AUTOSAR 4.1.1 are still included.

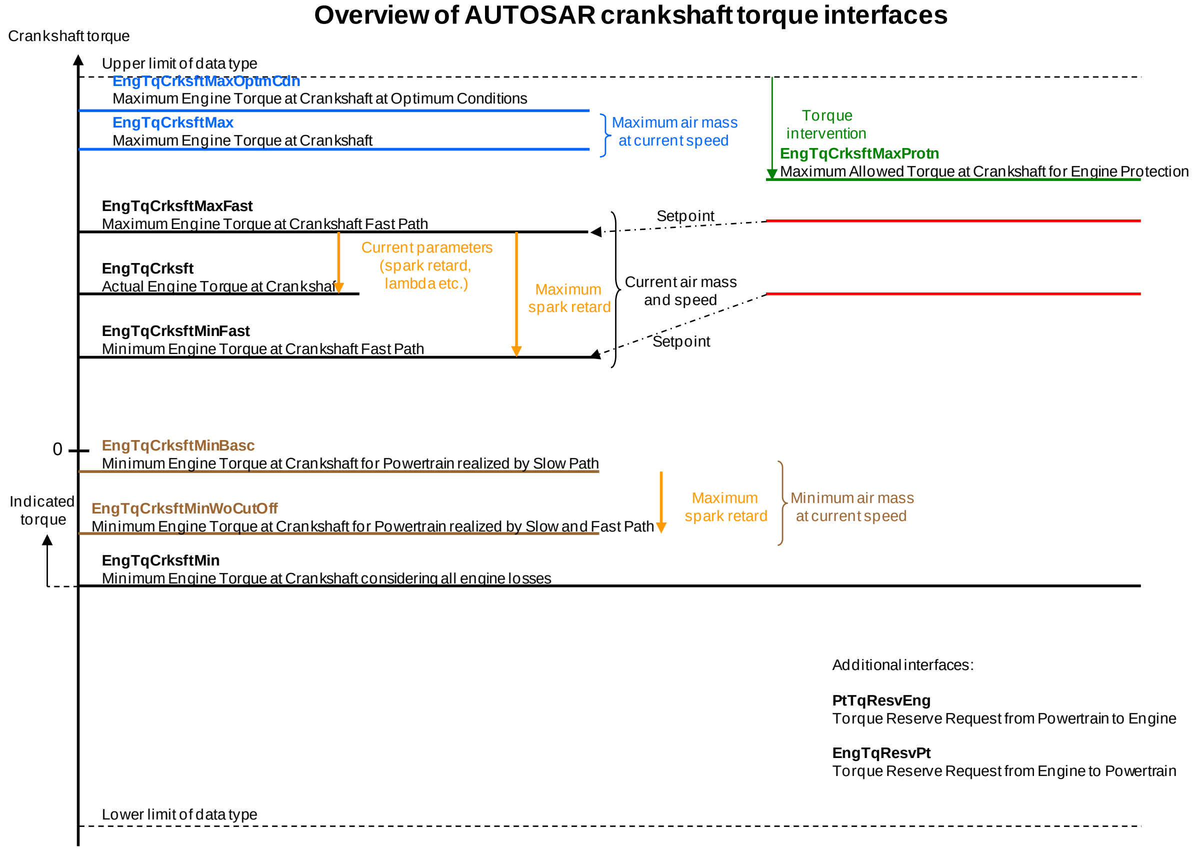

Figure 3: Overview of AUTOSAR crankshaft torque interfaces

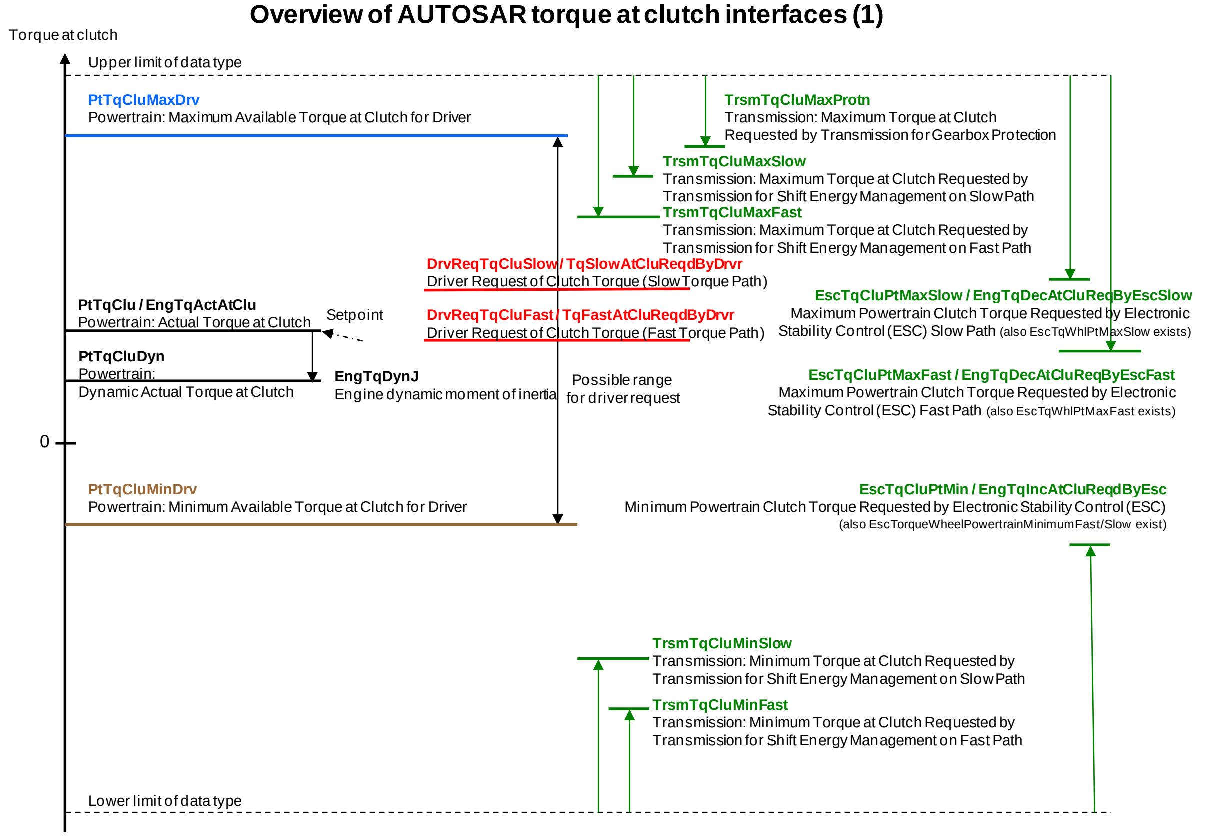

Figure 4: Overview of AUTOSAR torque at clutch interfaces (1)

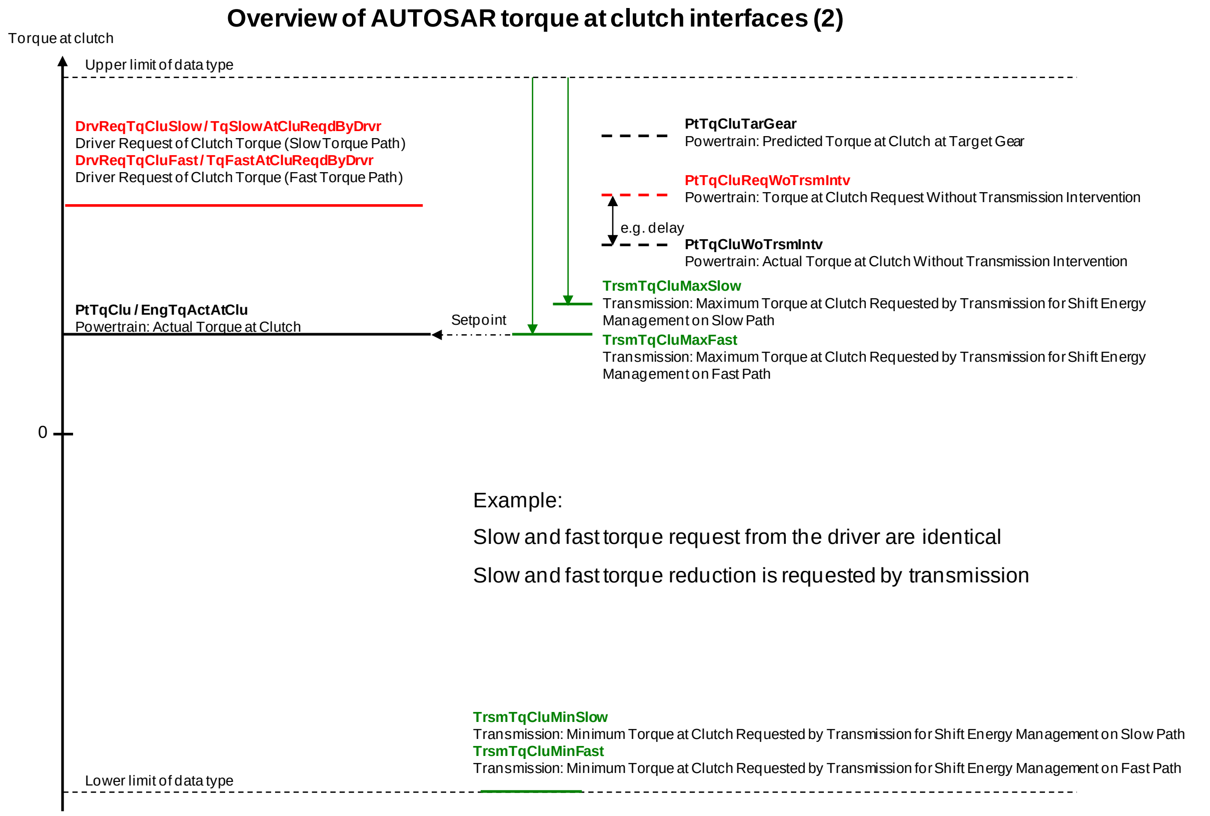

Figure 5: Overview of AUTOSAR torque at clutch interfaces (2)

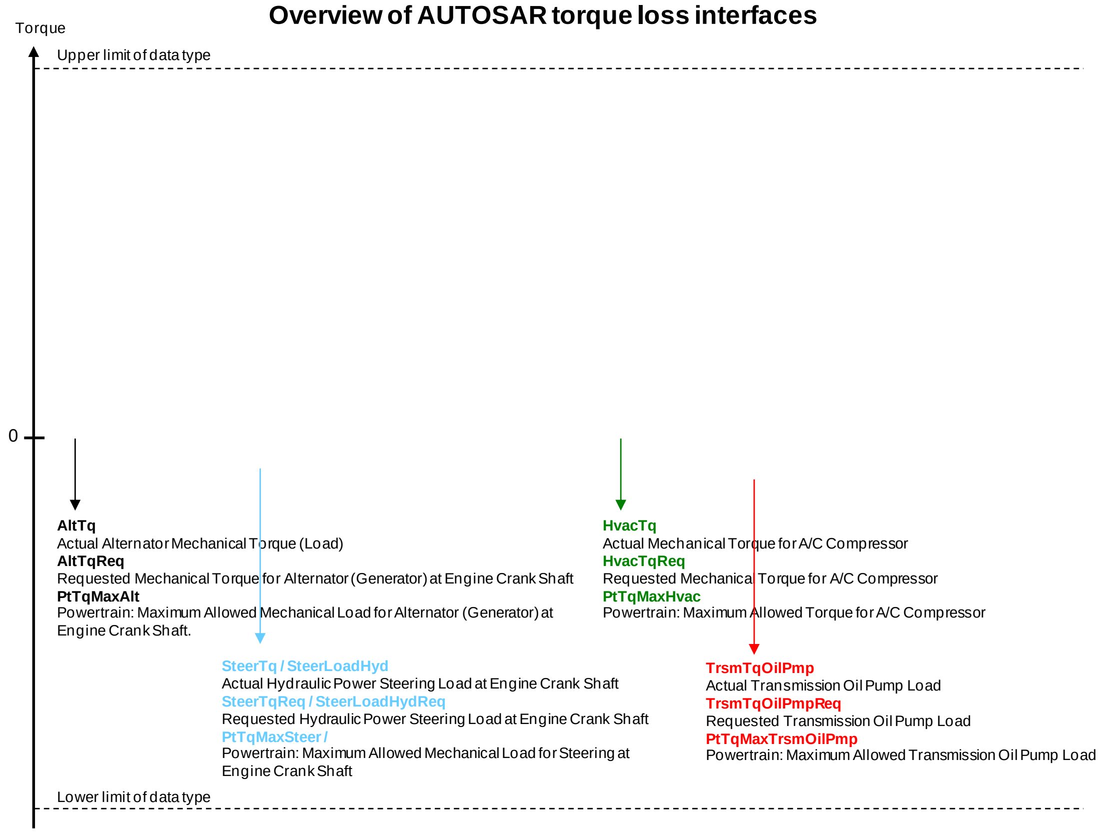

Figure 6: Overview of AUTOSAR torque loss interfaces

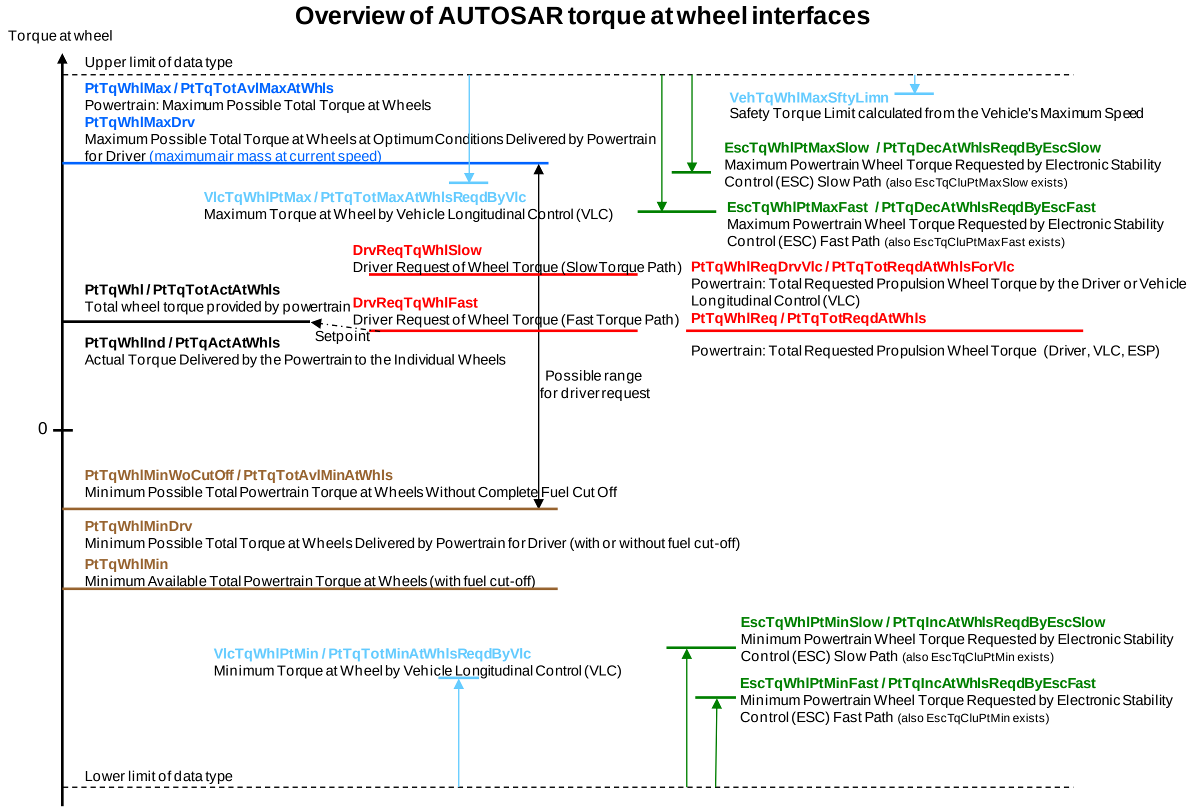

Figure 7: Overview of AUTOSAR torque at wheel interfaces

4 Architecture Overview

The following figures give an overview of the domain or functional architecture. They not necessarily give a complete picture but show the most relevant interconnections and components.

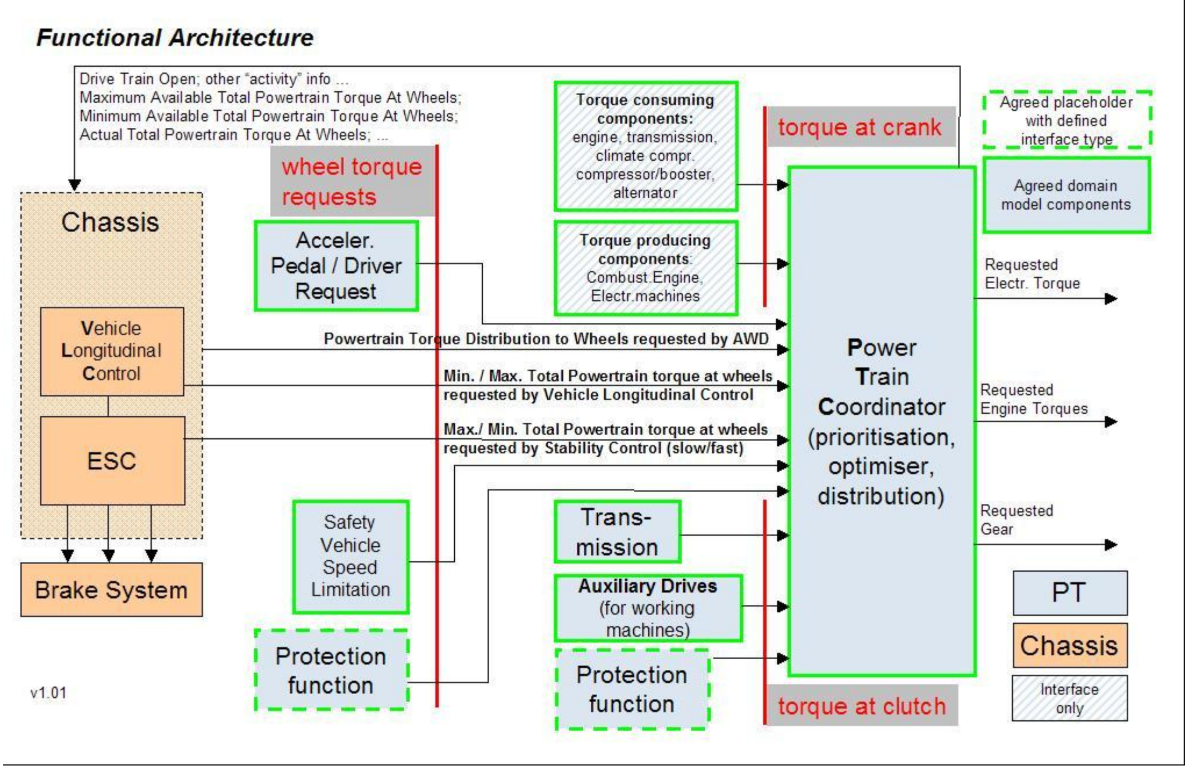

Figure 8: Overview of Functional Architecture

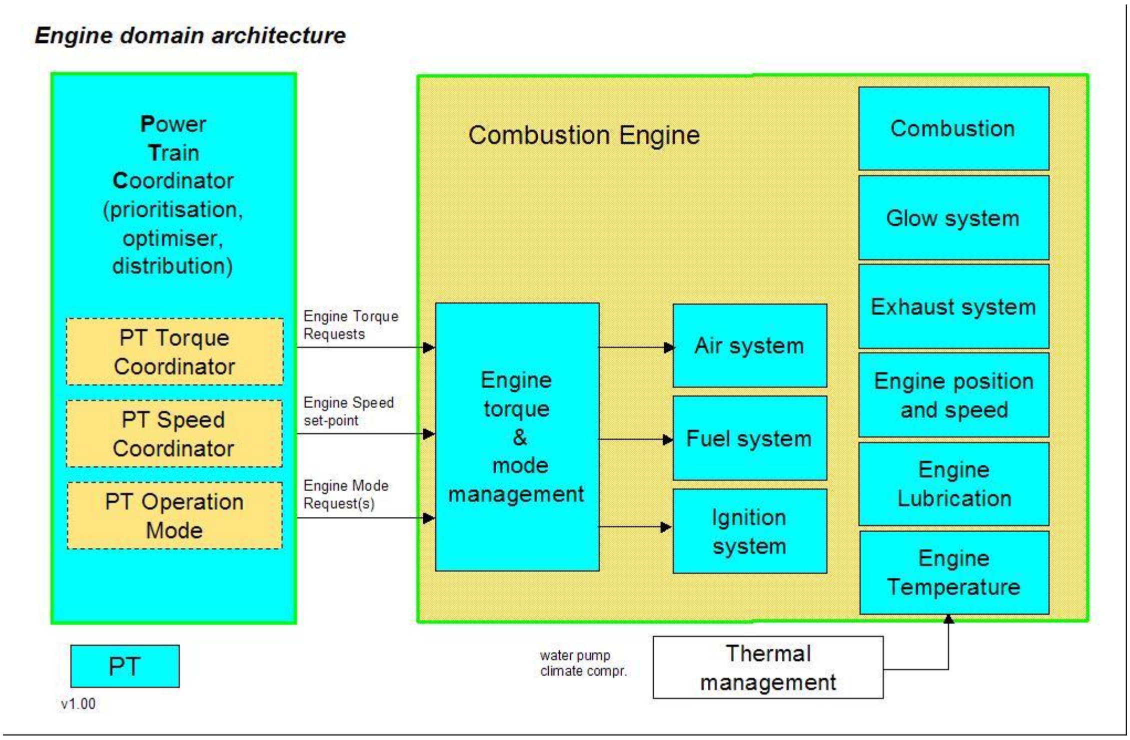

Figure 9: Detail – Combustion Engine Domain Architecture

Transmission System Architecture

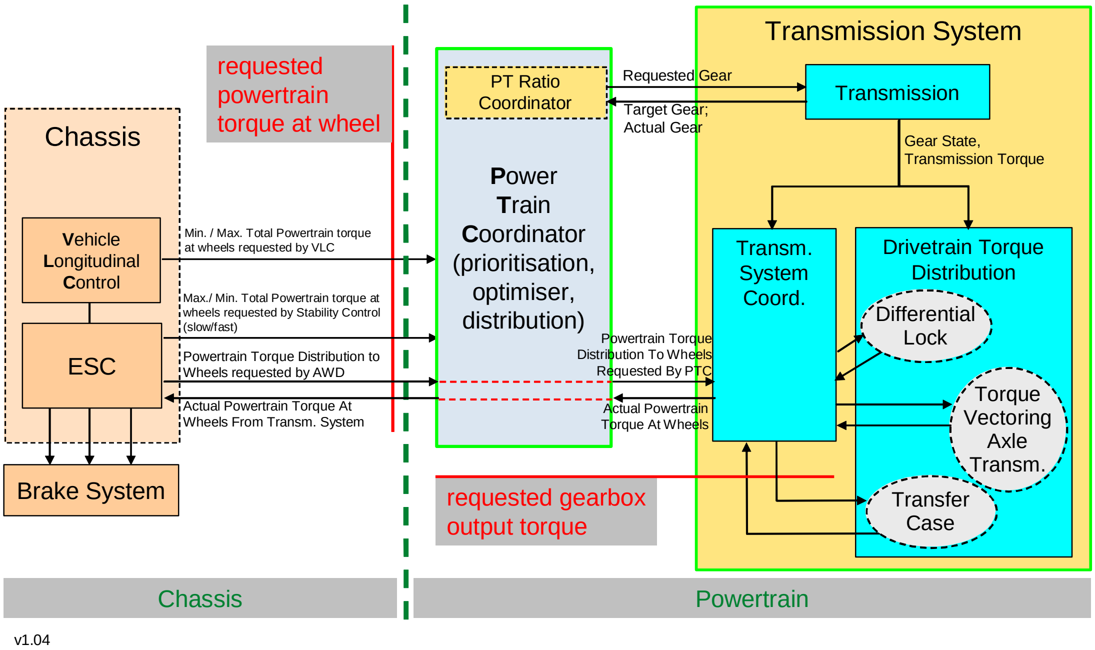

Figure 10: Detail –Transmission System Domain Architecture

5 Description of Exemplary Software Components

For being able to use and understand the standardized application interfaces a typical domain architecture was used as basis for demonstrating the signal flow. The components of this example domain architecture are described in the following.

5.1 Powertrain Coordinator – PTC (PtCoorr)

This composition includes all functions that coordinate the operation of the Powertrain, including:
Powertrain operation mode – management of states of all actuators (e.g. combustion engine, clutch(es), transmission, electric motors, etc.), including engine start / stop management (conventional & hybrid Powertrains).
Powertrain torque coordination – Torque coordination at Powertrain (PT) level, torque prioritisation, torque distribution for realisation at at PT level, torque reserve request for the PTC, pre-coordination of driveability functions for hybrids, Powertrain driveability filters, determination of total Powertrain losses for torque calculation, wheel torque calculation (min, max, consolidated), torque at clutch calculation (min, max, consolidated), transformation of torque set point from wheel torque to torque at clutch, transformation of torque set point from torque at clutch to torque at crankshaft, control/coordination of auxiliary drivers/actuators.
Powertrain speed coordination – Maximum speed limitation coordination (for protection of all PT components from damage from over speed) and coordination of idle speed / engine speed set point requests from all sources, e.g. transmission.
Powertrain ratio coordination – all transmission ratio set point logic. Note that realisation of ratio set point is carried out by transmission system, not PTC.

5.2 Transmission System (Trsm)

This composition includes all functions of the transmission system, including:
Transmission system coordination – Determines the torque and speed ratio over transmission, converter and differential, including the calculation of torque losses in the transmission system. Coordinates mechanical protection of the Drivetrain (gearbox, driveshafts, etc.), including calculation of torque limitation.

For manual transmission, this function includes the determination of the current gear and clutch status.

Transmission **–** Management of particular states in the transmission, including shift transition, driving off situation, creeping mode etc.. In case of shift transition, this functionality calculates torque requests to optimise the transition.

Control of transmission actuators to adjust the gear to the target gear (or to adjust the gear ratio to the target gear ratio in case of CVT). Gear ratio means the theoretical / physical ratio belonging to each gear and not any actual measured value. Control of gearbox countershaft (low/higher range) actuators is not included.

Calculates the torque gain of a hydrodynamic converter and the torque required to the converter input side in idle, etc. and controls clutch or converter actuators.

All functionality related to the protection of the transmission, including calculation of torque limitation, measurement or calculation of gearbox oil temperature, etc., and calculation of requests to other systems.

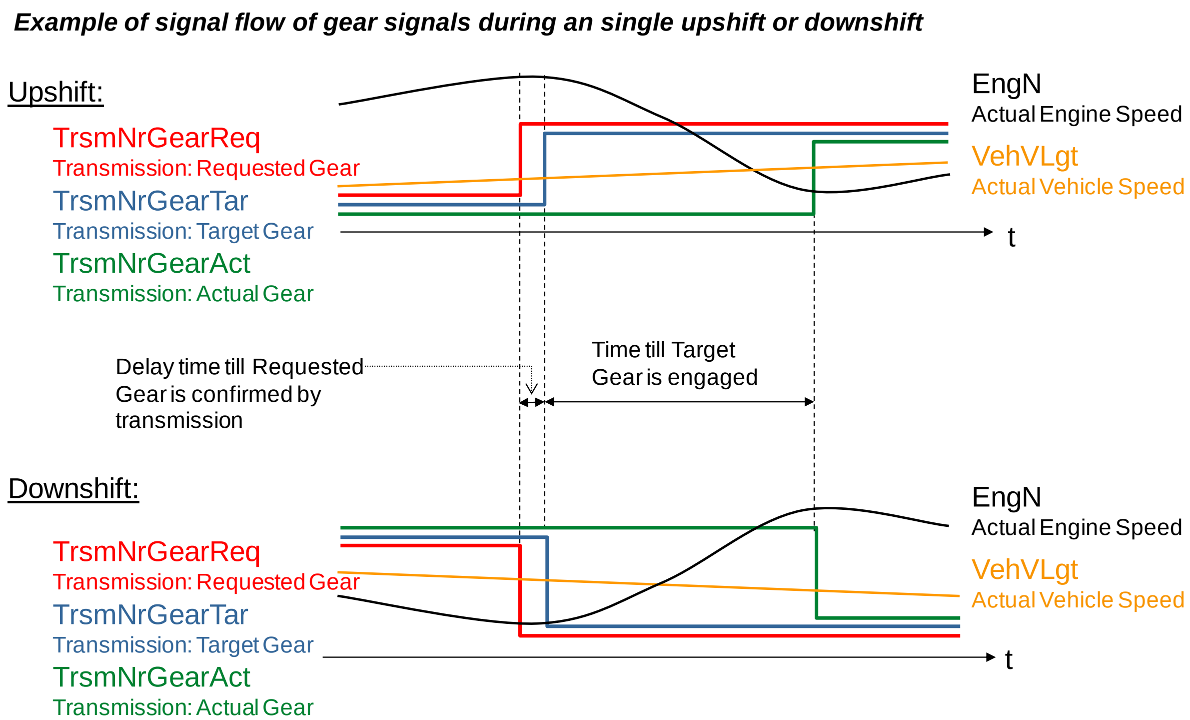

Figure 11: Example of signal flow of gear signals during a single upshift or downshift

Drivetrain Torque Distribution (DtTqDibtn) Differential Lock – All functionality related to the differential(s), which manage the torque distri-

bution between left and right wheels, for example locking of the differential. Does not include the calculation of the distribution set point.
Drivetrain Torque Distribution (DtTqDibtn) Transfer Case – All functionality related to the transfer case, which manages the torque distribution between front and rear wheels. Does not include the calculation of the distribution set point.
Drivetrain Torque Distribution (DtTqDibtn) Torque vectoring axle transmission – All functionality related to active distribution of powertrain torque to all four wheels individually. Does not include the calculation of the distribution set point.

For additional information on Drivetrain Torque Distribution (DtTqDibtn) please also refer to [4].

5.3 Combustion Engine (CmbEng)

This composition includes all functions directly related to the operation and control of the vehicle's combustion engine. The following sections, 5.3.1 to 5.3.3 inclusive, define the components as a result of Combustion Engine functionality decomposition agreed to date.

5.3.1 Engine Speed and Position (EngSpdAndPosn)

Functions that provide all parameters linked to engine shaft position and speed, including the synchronisation on between crankshaft and camshaft.
Crankshaft and camshaft signal acquisition.
Calculation of the engine position.
Calculation of the relative camshaft position for systems with variable valve timing and/or lift.
Related diagnosis and plausibility checks.

5.3.2 Engine Torque Mode Management (EngTqModMngt)

Includes calculation of engine torque set point, realisation of that set point (coordination of air / fuel / ignition, etc.), determination of consolidated engine torque, control of engine speed (idle / off-idle / limitation), and management of engine modes (including overall mode, modes for realisation of engine start & stop, and combustion modes).

5.3.3 Combustion Engine: Miscellaneous (CmbEngMisc)

Combustion Engine Misc gathers together miscellaneous engine interfaces. In general these are common data required for correct operation of the engine (engine temperature, ambient air pressure and battery voltage) or required for fail-safe actions (crash status). The way in which these interfaces are used is not standardised In future AUTOSAR releases, it is likely that these interfaces may be moved to different (more appropriate) provider or receiver components / compositions.

5.4 Vehicle Motion relevant for Powertrain (VehMtnForPt)

This composition includes Powertrain functions related to vehicle motion. The following sections, 5.4.1 to 5.4.3 inclusive, define the components that have so far been agreed as part of this composition.

5.4.1 Driver Request (DrvReq)

Driver-specific conversion of accelerator pedal position to requested torque: determines the driver request related to the motion of the vehicle. For longitudinal motion, this functionality interprets the driver request as a torque request.

5.4.2 Accelerator Pedal Position (AccrPedlPosn)

The component calculates a percentage from the acquired position of the sensor, and contains plausibility checks to ensure the information. Kick-down detection is included in this component.

5.4.3 Safety Vehicle Speed Limitation (VehSpdLimnForSfty)

Hard limitation of vehicle speed by engine torque reduction, without any comfort functionality.

5.4.4 Vehicle Motion (Powertrain): Miscellaneous (VehMtnForPtMisc)

VehMtnForPtMisc gathers together miscellaneous interfaces in the context of vehicle motion powertrain. The way in which these interfaces are used is not standardised. In future AUTOSAR releases, it is likely that these interfaces may be moved to different (more appropriate) provider

or receiver components / compositions. It is even not excluded that they are moved to components that already exist.
VehMtnForPtMisc e.g. is used to close open interfaces in the case that it is committed that some component within vehicle motion powertrain will request or provide it but it is not yet decided which component or the component is missing.

5.5 Powertrain: Miscellaneous (PtMisc)

PtMisc gathers together miscellaneous powertrain interfaces. The way in which these interfaces are used is not standardised. In future AUTOSAR releases, it is likely that these interfaces may be moved to different (more appropriate) provider or receiver components / compositions. It is even not excluded that they are moved to components that already exist.
PtMisc e.g. is used to close open interfaces in the case that it is committed that some component within powertrain will request or provide it but it is not yet decided which component or the component is missing.

6 Sensor Actuator Design Pattern

The Sensor Actuator Design Pattern describes how to handle software (SW) device drivers to control sensors or actuators that are connected to an ECU. It is not specific for the powertrain domain but can also be applied to any other domain, e.g. chassis.
Note: In AUTOSAR release 4.1, there are mismatches in the standardized application interfaces. E.g. chassis domain Explanatory Document [4], chapter 2.5.4.1. Internal state sensors shall be synchronized in a future release.

The pattern also includes naming pattern for ports and components. The pattern is described on two levels of abstraction:

“Blackbox”: Blackbox means in this context that the inner architectural splitting is not displayed. Any splitting within the Blackbox is possible, but the mandatory ports of the Blackbox have to be provided.
Note: Within the blackbox figures we used the icon for “Sensor/Actuator SW-C”, however, the way of implementing it, whether as a composition or an atomic is not normalized within the blackbox pattern.

“Whitebox”: Whitebox means in this context that the inner architectural splitting according to the descriptions below has to be followed.
Note: Although compositions are shown they are only to be seen as example compositions. I.e. it is not normalized whether there is one large device abstraction composition or a composition for each sensor or actuator resp. a.s.o.

Device Abstraction as defined in this document is part of the application software (ASW). The Device Abstraction is located above the RTE. It is a set of software components that abstracts from the sensors and actuators connected to a specific ECU.

The interfaces between Device Abstraction and ECU Abstraction as well as between Device Abstraction and other software components of the Application Software (ASW) are not part of the pattern. They are only included to show the embedding in the overall architecture. „A sensor-actuator software-component is written for a concrete sensor or actuator and uses the ECU abstraction interface [9].” The interfaces between sensor-actuator components and ECU abstraction components are client-server interfaces.

Note: The pattern does not yet cover diagnostics. To support diagnostics it might be usefull to split an atomic component into one composition containing several atomic components.

This pattern was derived based on the AUTOSAR method described in Figure 12.

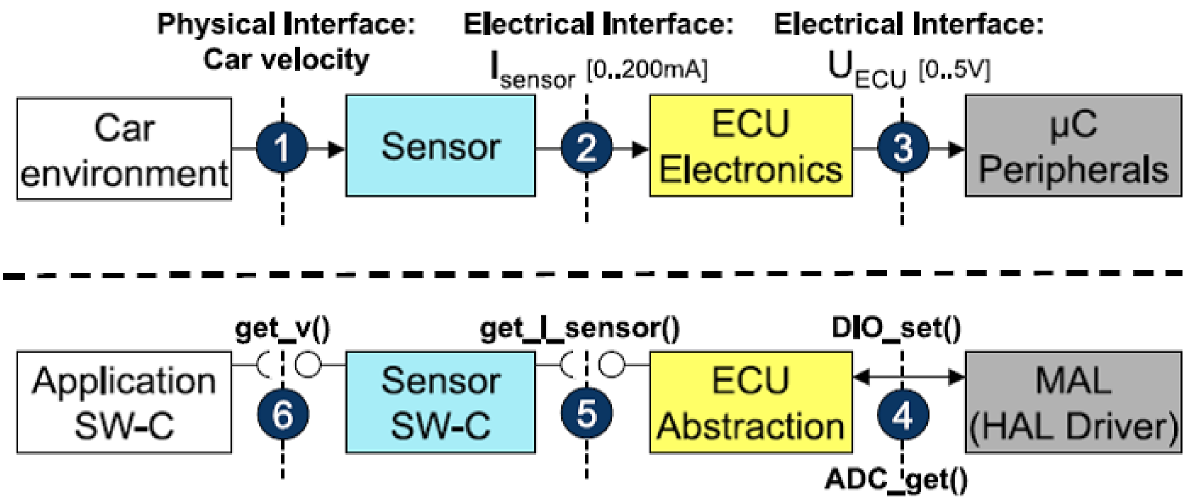

Figure 12: Interfaces between Hardware and Software [6]

The name pattern uses the following syntax defined according to ANTLR [8]. This syntax and some parts used (like 'keyword') are also used in [7], e.g. in [TPS_STDT_00055].

Semantic port prototype (blueprint) definition together with short name pattern and long name pattern:

<u>**({Prefix})ElecRaw**</u> : **Electrical Raw Value of {Prefix}** - Unfiltered electrical raw sensor value(s). Electrical signals can only be represented in voltage, current and time [10].

<u>**({Prefix})ElecBascFild**</u> **: Electrical Basic Value of {Prefix}** - Basic filtered electrical raw sensor value(s) (e.g. maximum allowed phase shift is one scheduling raster or maximum 360° crankshaft rotation if exhaust gas pulsation dependent). Electrical representation of a technical signal [10]. Electrical signals can only be represented in voltage, current and time.

<u>**({Prefix})Raw**</u> **: Raw Value of {Prefix}** - physical raw/base sensor value(s). Simple conversion of basic filtered electrical (ElecBascFild) to physical value(s).

<u>({Prefix})Measd</u> : {Prefix} (Measured) - Final filtered and offset corrected physical sensor value(s). Physical sensor value(s)/standard sensor value(s). The physical sensor value is the linerarized/filtered physical raw/base sensor value including offset. At this step a (significant) phase-shift could be possible.

<u>({Prefix}|Consold)</u> : {Prefix} - Consolidated physical value, either a measured value (Measd) or a modelled value (Estimd). Final filtered and offset corrected consolidated actuator value/physical sensor value(s). Virtual physical sensor value/fused sensor value that comes as close as possible to the technical signal. In case of inability to provide a physical sensor value (e.g. failure, implausibility or other reasons) a substitute value/default value or a frozen value is provided.

<u>({Prefix})Estimd</u> : {Prefix} (Estimated) - Final filtered and offset corrected physical sensor value(s) replacement model value(s) for physical sensor value(s)/standard sensor value(s).

<u>({Prefix})Outp</u> : Output of {Prefix} - Final controller output (closed loop or open loop). E.g. torque or engine speed. Typically expressed as percentage.

<u>({Prefix})Sp</u> : Setpoint {Prefix} - Final actuator setpoint. Typically expressed as percentage.

<u>({Prefix})Reqd</u> : Requested Setpoint {Prefix} - Final required physical setpoint. Typically expressed as percentage but could also be expressed e.g. as factor.

With

Prefix : ('keyword')+
used for signal naming
A typical prefix could look like this:
{**Comp**}('Stg'{'index'})('AtBnk'{'index'})
with
('Stg'{'index'})('AtBnk'{'index'})
meaning "{'index'} stage at bank {'index'}", the
two {'index'}-parts have a complete different semantic meaning
but are numbers:
is used if there are different sensors in usage and several sensors

of the same kind are used in different stages and banks
Example: TrboChrgrStg3rdAtBnk2nd

Suffix1 : ('Stg'{‘index’})('AtBnk'{‘index’})
used only as part of the component prototype names
(instance of
component types)
('Stg'{‘index’})('AtBnk'{‘index’})
means “{‘index’} stage at bank {‘index’}”, the
two {‘index’}-parts have a complete different semantic meaning
but are numbers:
is used if there are different sensors in usage and several sensors
of the same kind are used in different stages and banks
Example: TrboChrgrStg3rdAtBnk2nd

Suffix2 : ('keyword')+
used only as part of the component type names to distinguish
between
different HW components of the same type

Comp : ({**Senso**r} | {**Actuator**})
Sensor : ('keyword')+
Actuator : ('keyword')+
Comp/Sensor/Actuator –
name of the Actuator/Sensor (HW component)
or
name of a virtual actuator/sensor, i.e. there is not
necessarily a HW component available with this name
but the device abstraction provides a corresponding signal
as if there would be
or
a more generic description of a set of actuators/sensors
with common properties
or
HW implementation specific identifier
Examples: TrboChrgr / EngT a.s.o.

‘index’ in this context is one of the following keywords:
(1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th | 9th)

The semantics of ' Stg' and ' Bnk' is defined as follows:
'Stg' - Index for devices connected in a row.

'AtBnk' - Index for parallel-connected devices

Semantics of 'keyword' is defined in [7], [TPS_STDT_00004].

‘At’, 'Stg' and 'Bnk' correspond to standardized keywords.
Please note that not all keyword abbreviations used in names of examples are (yet) part of the standard.

Components together with name pattern

SwComponentPrototype Name and SwComponentType Name:
(Drvr{Device}Elec | DevDrvr{Device} | Dev{Device}Virt | DevCoorrVirt)('keyword')*
With
Device: (Snsr | Actr)

Typical derived names with long name snippets (these names are also used in the figures):
SwComponentPrototypeName :
DrvrSnsrElec(For{Sensor}({Suffix1})) | // Electrical sensor driver
DevDrvrSnsr(For{Sensor}({Suffix1})) | // Sensor device driver
DevSnsrVirt(For{Sensor}({Suffix1})) | // Virtual device driver
DrvrActrElec(For{Actuator}({Suffix1})) | // Electrical actuator driver
DevDrvrActr(For{Actuator}({Suffix1})) | // Actuator device driver
DevCoorrVirt(For{Actuator}({Suffix1})) | // Virtual device coordinator

SwComponentTypeName :
DrvrSnsrElec(For{Sensor}({Suffix2})) | // Electrical sensor driver
DevDrvrSnsr(For{Sensor}({Suffix2})) | // Sensor device driver
DevSnsrVirt(For{Sensor}({Suffix2})) | // Virtual device driver
DrvrActrElec(For{Actuator}({Suffix2})) | // Electrical actuator driver

DevDrvrActr(For{Actuator}({Suffix2})) | // Actuator device driver
DevCoorrVirt(For{Actuator}({Suffix2})) | // Virtual device coordinator

Note: // are remarks

Please be aware: The {Sensor} part of the component type and the component prototype name are not necessarily the same: within the type a more generic sensor/actuator description might be choosen whereas the prototype is very precise and denotes a specific sensor/actuator. Example: (Prototype : Type) *DevDrvrActrForTrboChrgrStg3rdAtBnk2nd : DevDrvrActrForStdPidCtrlr*.

For compositions instantiating the atomic component types typical names are

{Comp}({Suffix1}) : {Comp}({Suffix2})

Used colours in figures:
Black - mandatory
Red - recommended
Green - optional

Figure 16 presents the complete overall design pattern as it is used for a closed loop controlled actuator with position feedback.

Examples for applying the pattern are shown in Figure 17 (throttle valve) and Figure 18 (turbo charger). For turbo charger an extended example showing multiple instantiations within a 2 bank system with 3 stages is presented in Figure 19. Please note: In most cases it is not recommended to use company names in model names (like “HellaXYZ” used in the figures). Company names etc. are only used in the examples to show the difference between type and prototype and what is the reason for the difference. For general rules and recommendations how to deal with variants in models, as for example expressed by the company names in the examples, please refer to the modeling guides and templates.

Different tailored patterns derived from the overall pattern in Figure 16 for sub use cases are given:

Figure 21 presents the tailoring for a standard sensor; typical standard sensors are temperature, pressure or position sensors.
An example is given in Figure 23 (environment temperature sensor).
In Figure 22 the signal flow is shown.
Figure 20 presents the tailoring for a smart actuator.

The sensor actuator design pattern supports software sharing on different levels:
Development partner 1 might deliver the sensors together with the basic electrical driver software (DrvrSnsrElec), development partner 2 might deliver the sensor device driver software (DevDrvrSnsr) and the third partner might develop the substitute models together with the virtual device drivers (DevSnsrVirt).
There might be different suppliers for the same sensor / actuator or there might be different sensors / actuators used within one and the same system.

In some cases there might be parts of the implementation that can be reused for different sensors / actuators. Therefore the name pattern for the component type name is more generic and does not necessarily contain the sensor / actuator name.

In other cases the sensor / actuator names are not sufficient enough to make the component type names unique so an additional identifier can be added to the component type name.

In some cases, e.g. in more easy cooperation models or in full service supplier model (black box view), it might not be required to have the full structuring according to the overall pattern and some of the SW-C might be combined in a single SW-C. The full service supplier view for sensor actuator is shown in Figure 13. The same tailored views as for the overall pattern are possible as shown in Figure 14 (simple actuator) and Figure 15 (standard sensor). They only contain mandatory interfaces.

Please note: Setpoint requests from several different SW-Cs will be handled outside the Sensor Actuator design pattern. This is demonstrated as a design pattern in Figure 24.

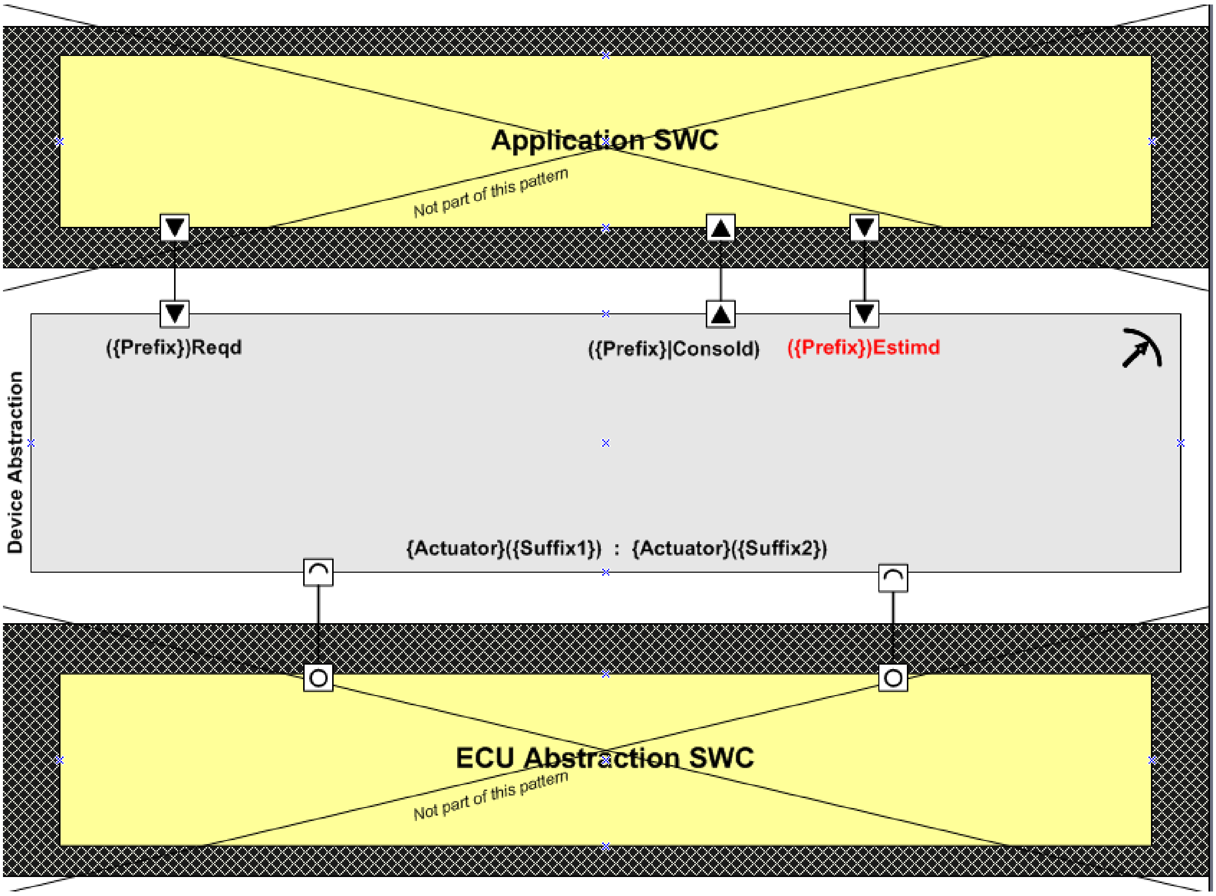

Figure 13: Closed loop controlled actuator and position feedback (black box view)

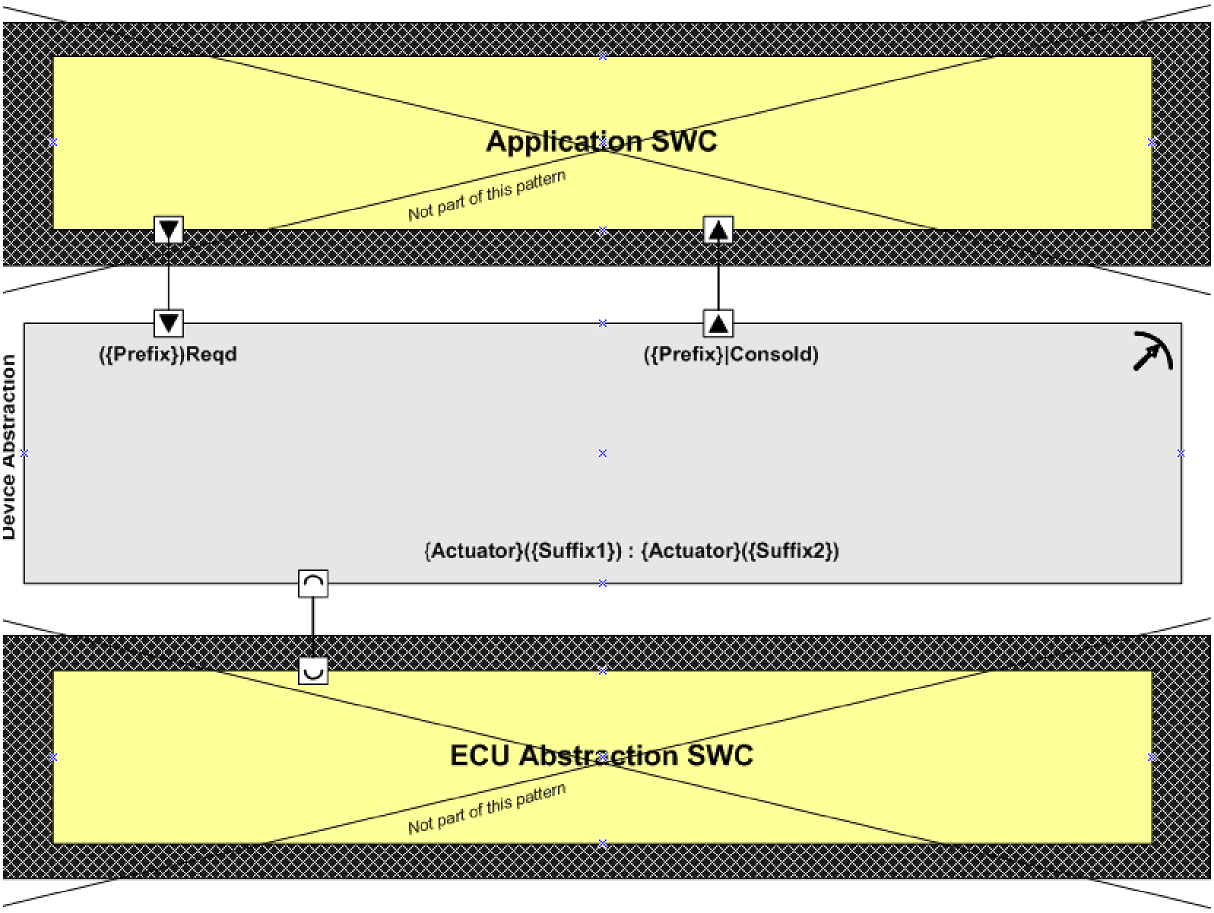

Figure 14: Design pattern for simple actuator (black box view)

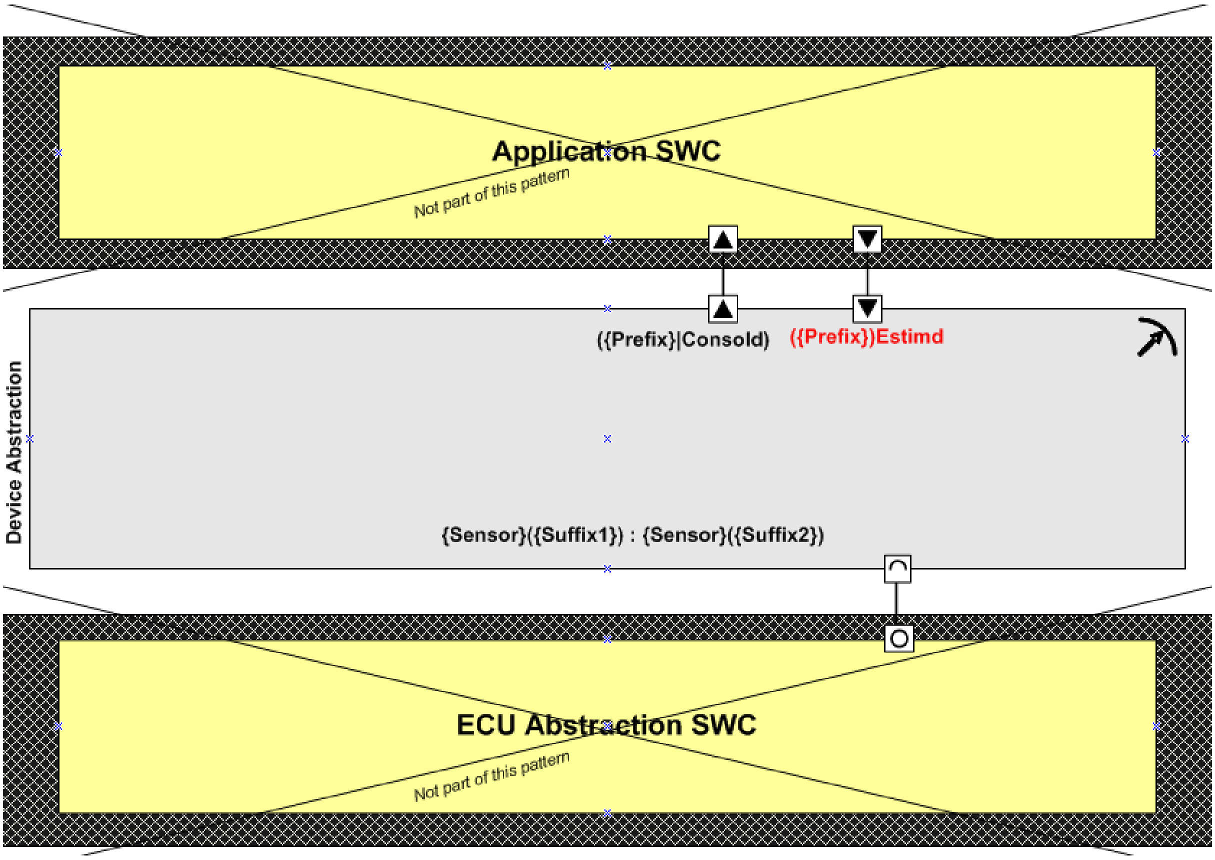

Figure 15: Design pattern for standard sensor (black box view)

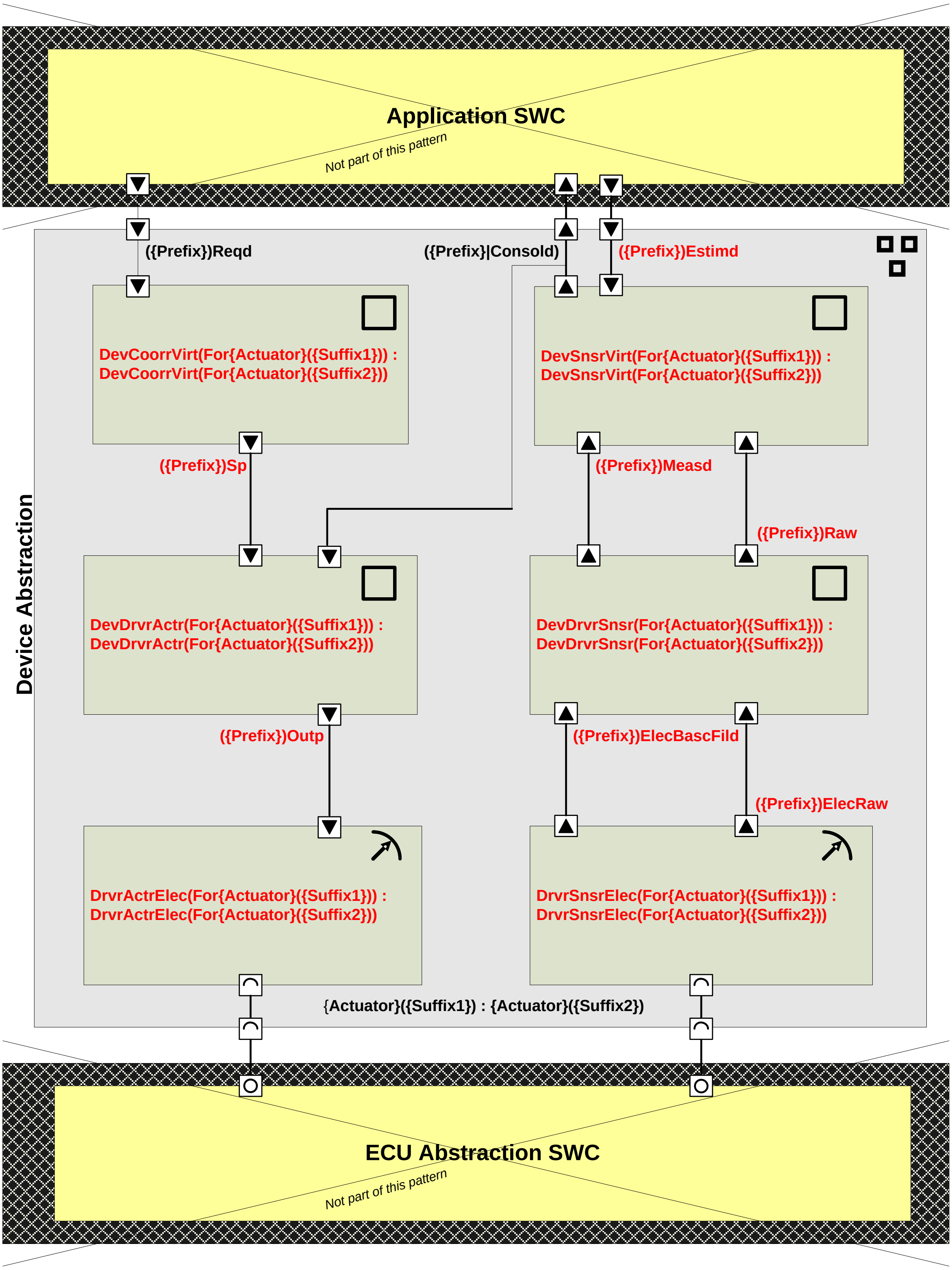

Figure 16: Closed loop controlled actuator and position feedback

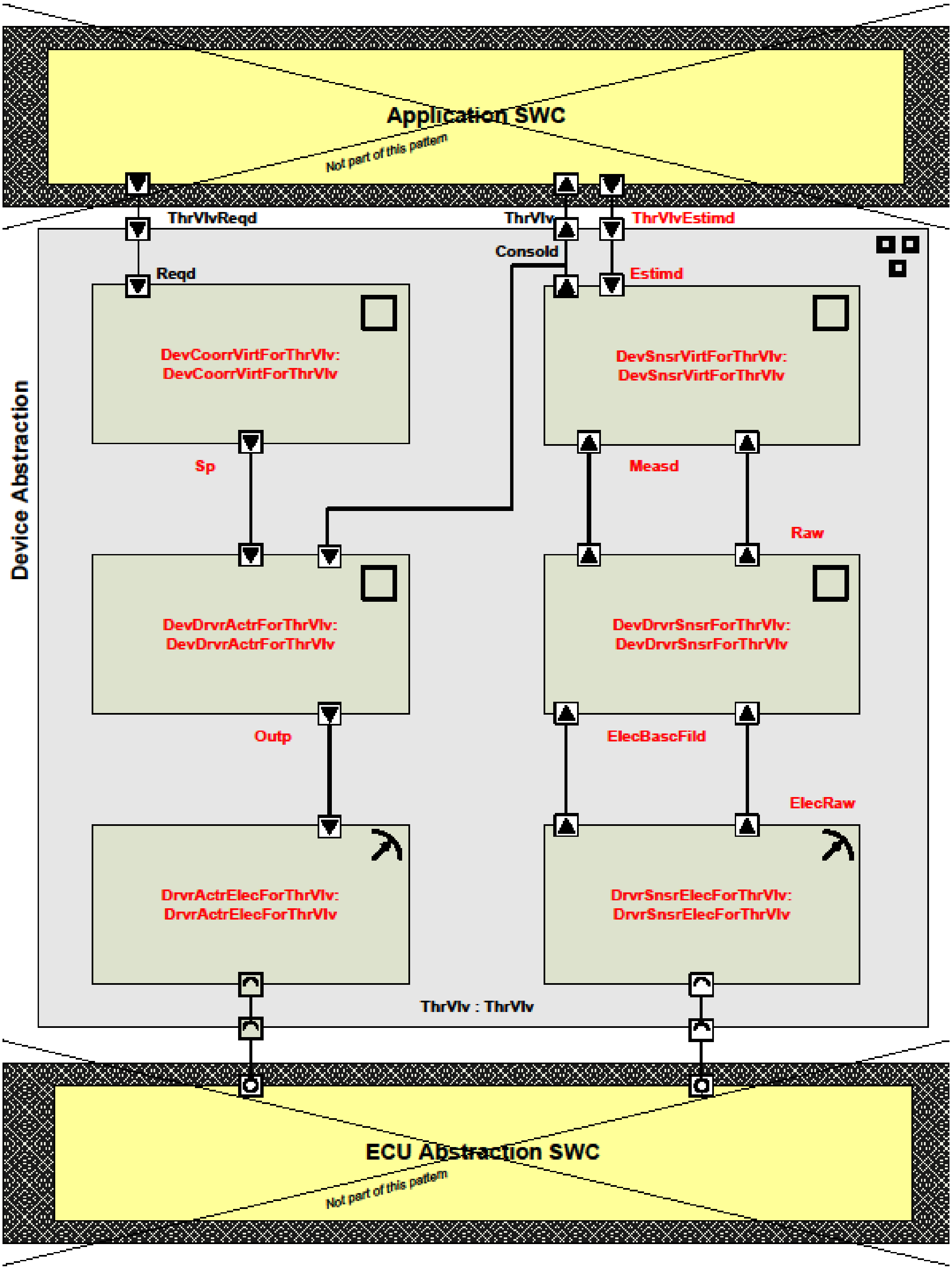

Figure 17: Example: Closed loop controlled throttle and position feedback

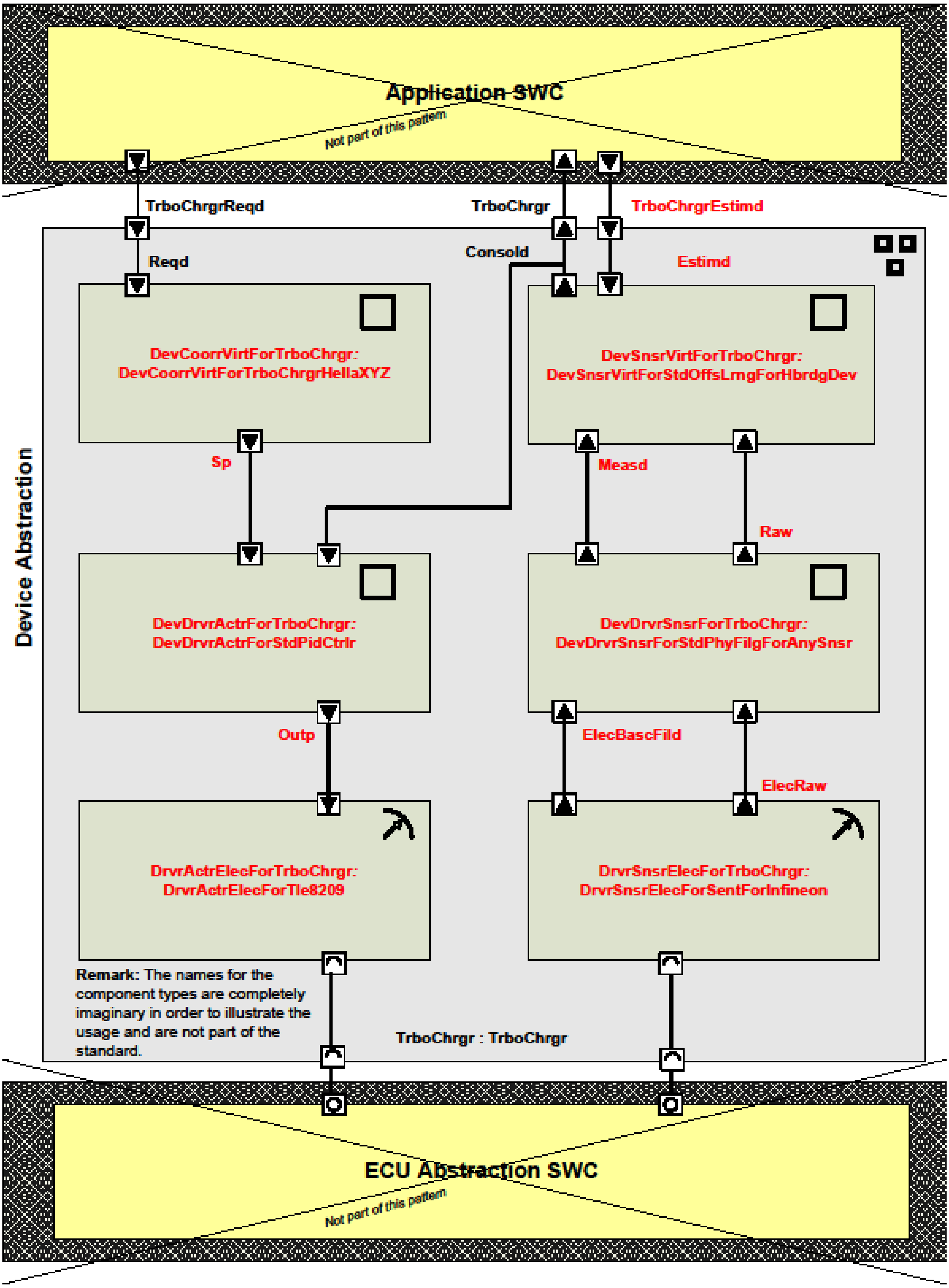

Figure 18: Example: closed loop controlled turbo charger and position feedback

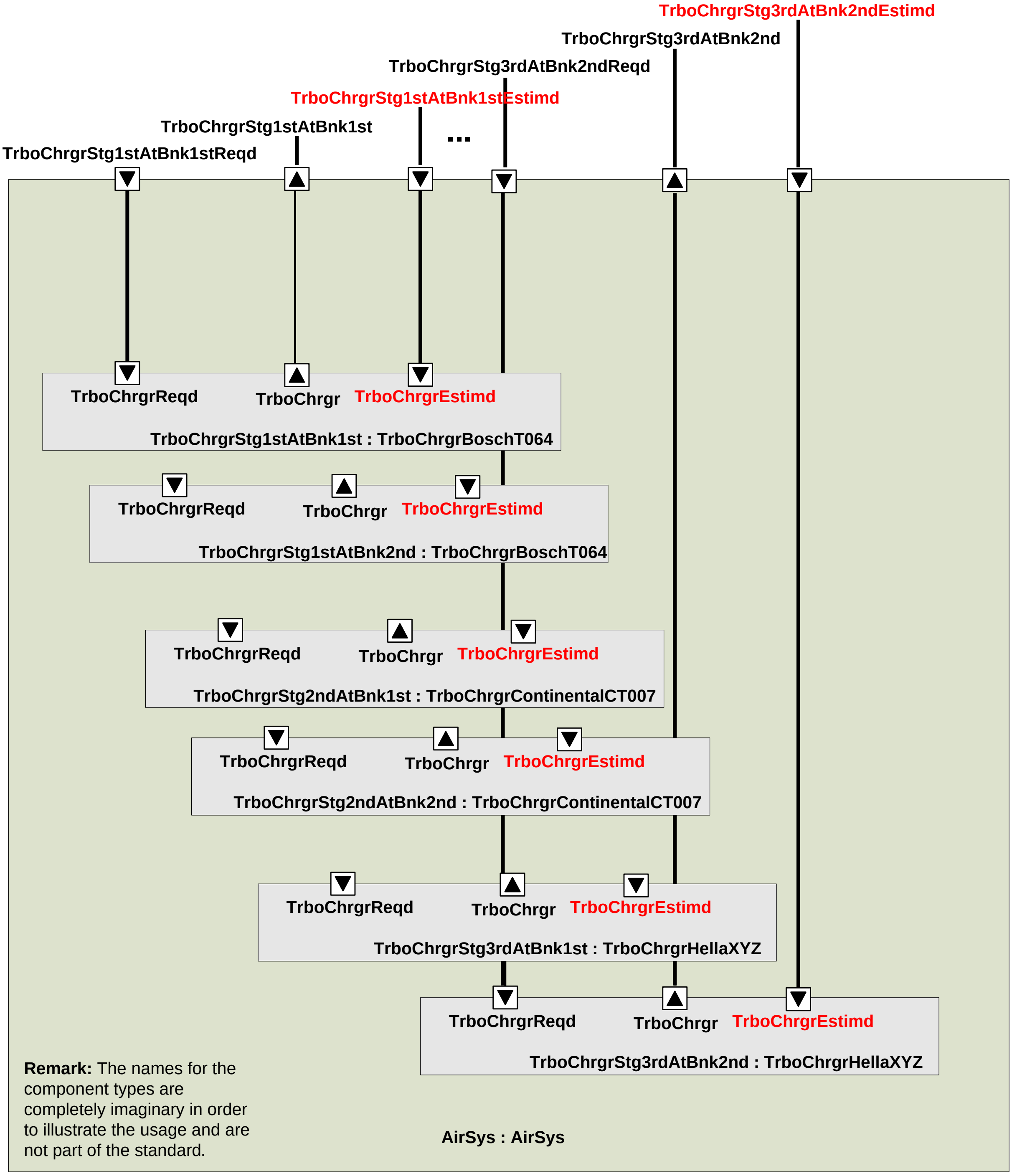

Figure 19: Example: Project system configuration for turbo charger with several stages and banks – delegation ports

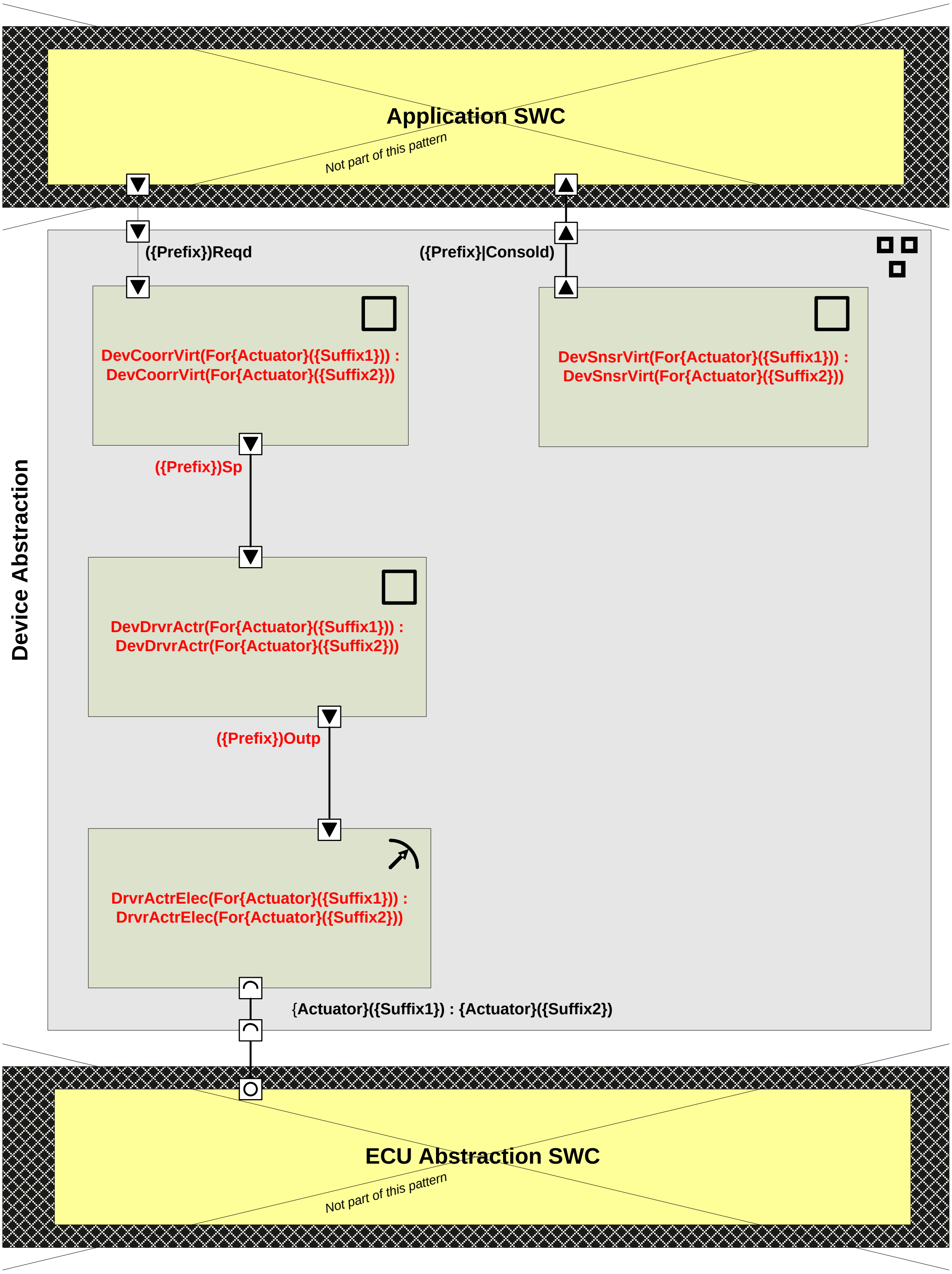

Figure 20: Open loop controlled actuator or smart actuator

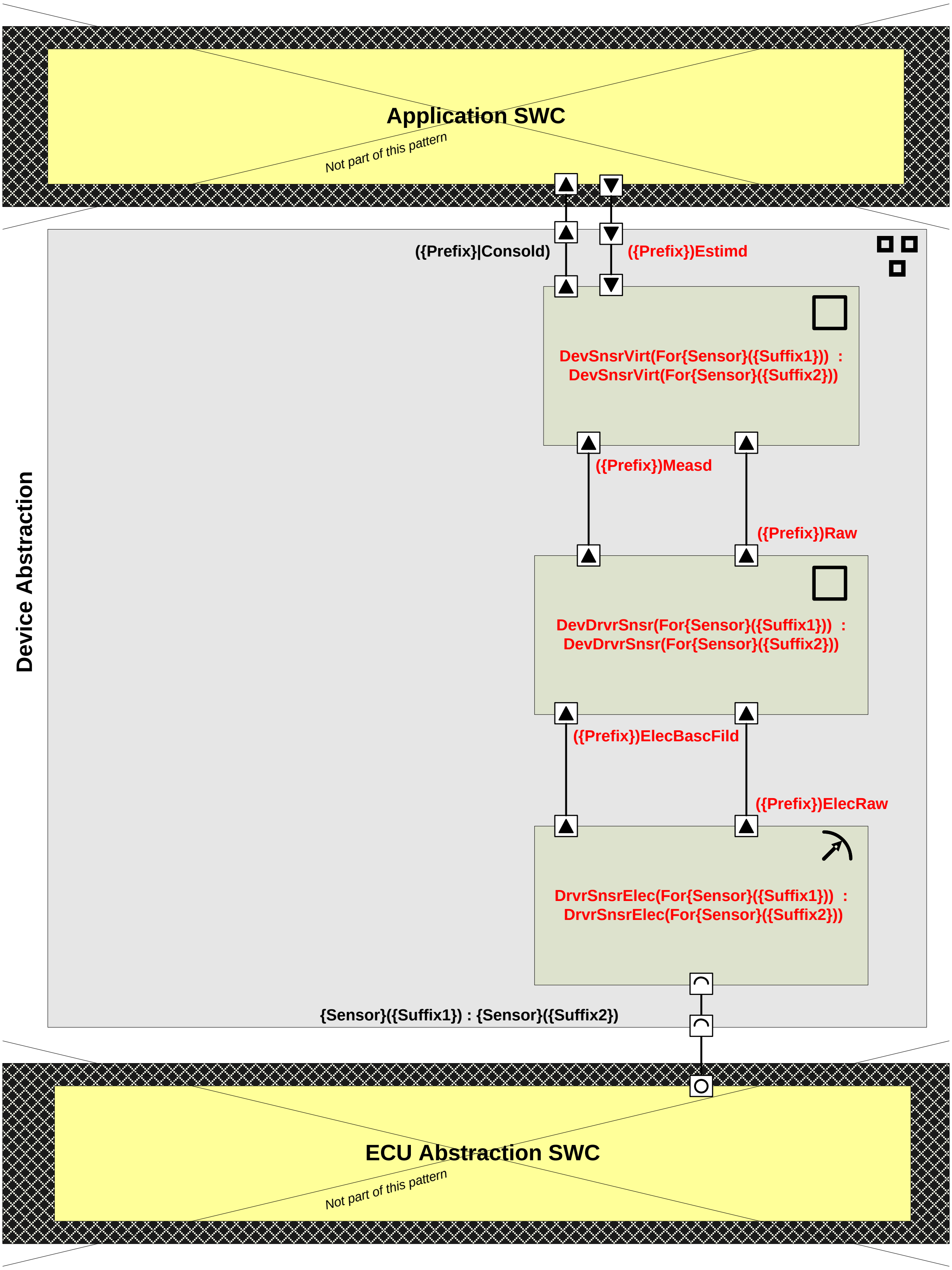

Figure 21: Design pattern of blueprint components for standard sensor

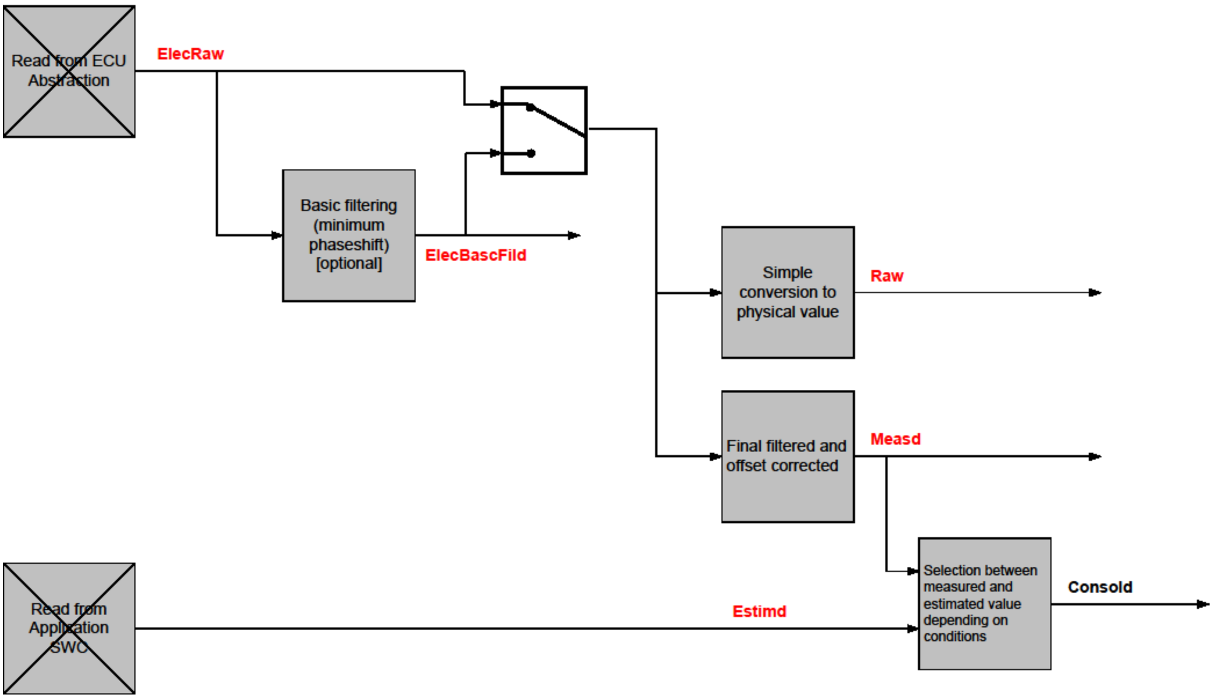

Figure 22: Signal flow within a sensor SW-C composition

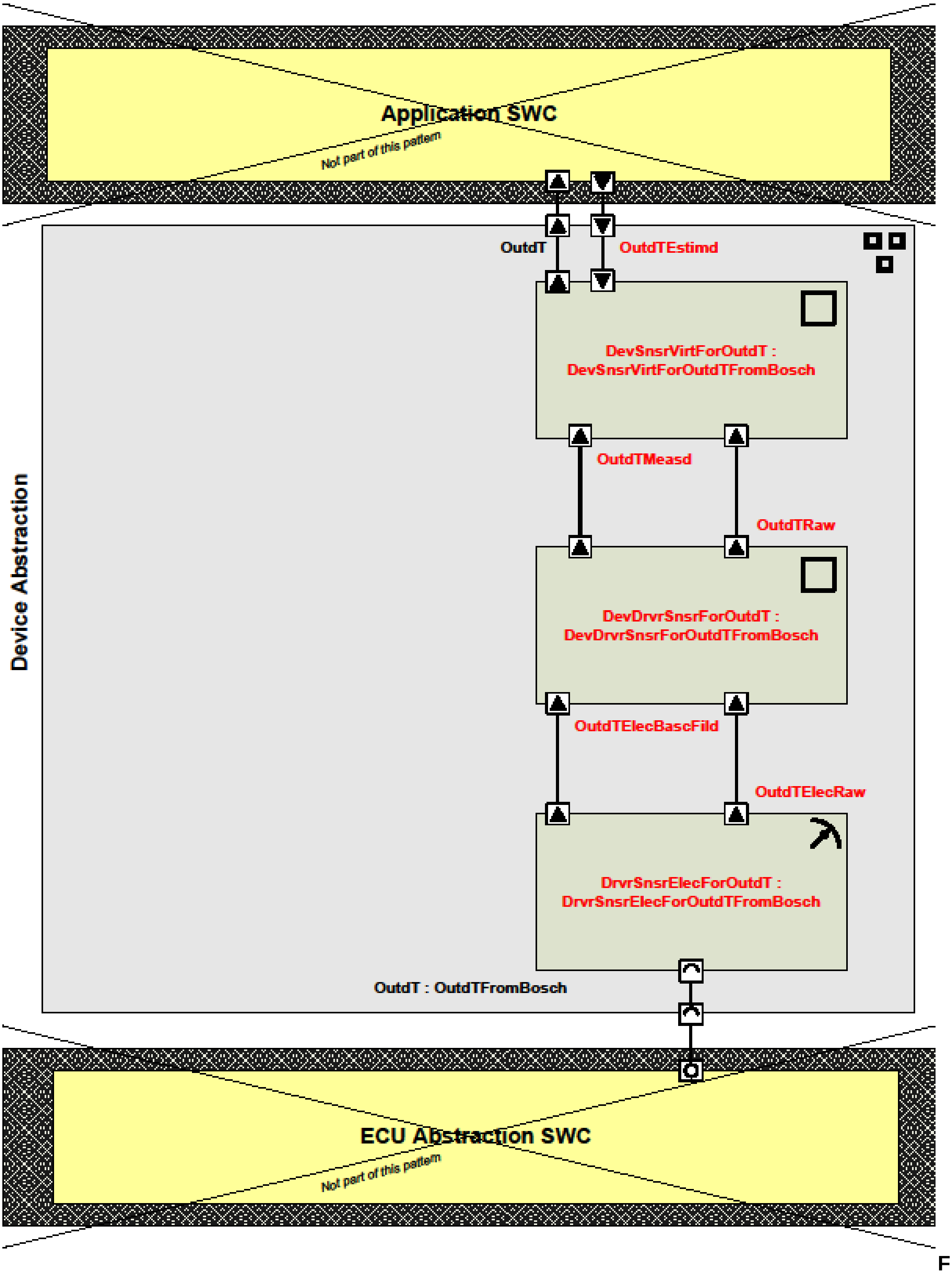

Figure 23: Example: Standard sensor for environment temperature

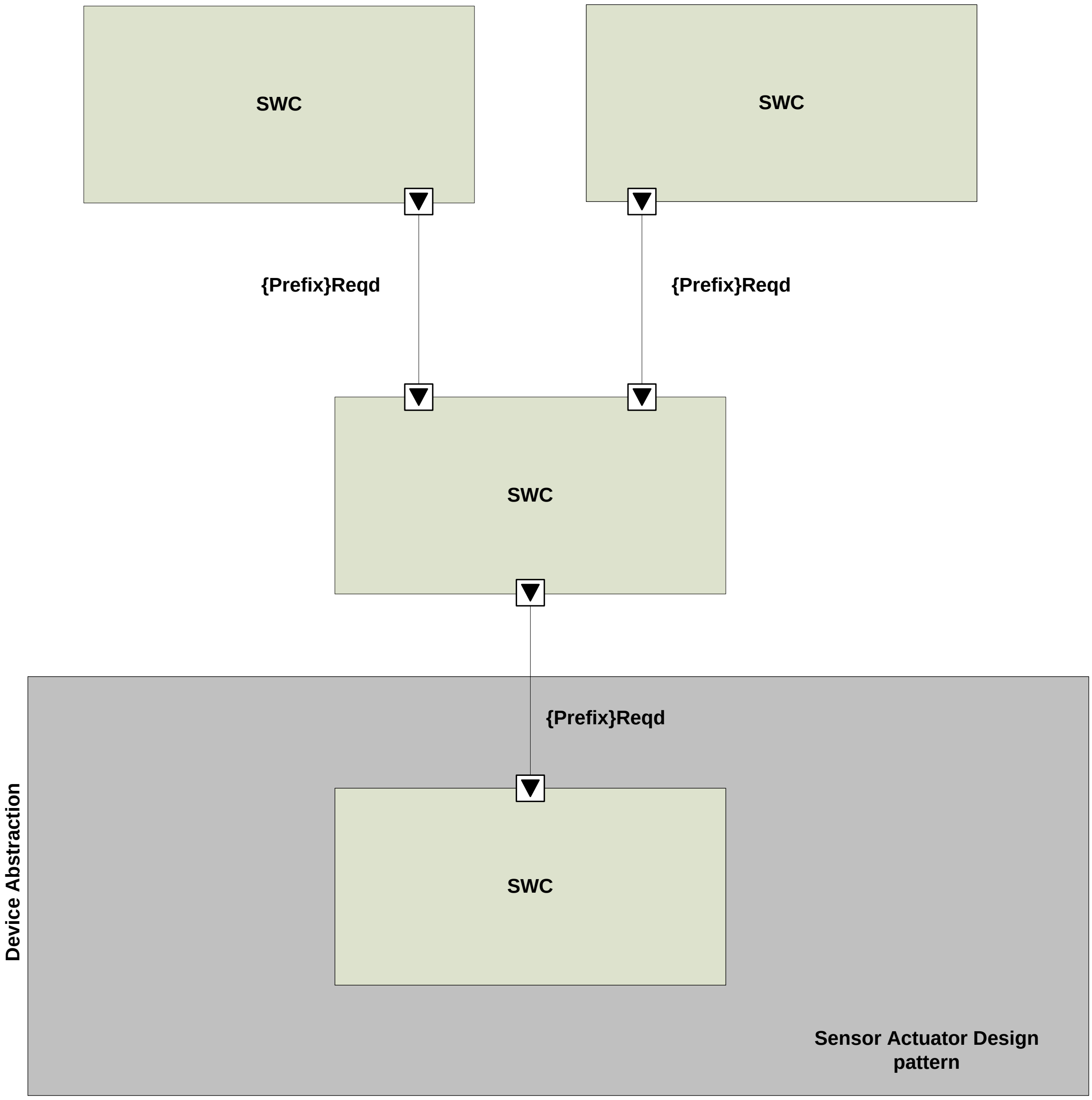

Figure 24: Design Pattern for arbitration between several different setpoint requesters

7 Additional Information

7.1 Differences between SW-Cs and ECUs

The SW components defined in chapter 4 are not to be confused with an ECU's functionalities.
For example, a combustion engine control ECU may contain the Combustion Engine SW-C plus other SW-Cs.

7.2 Functional safety

Many Powertrain signals are safety-relevant, therefore

- The AUTOSAR RTE will provide reliable communication for these signals at the low level, and
- Diagnostics and safety concepts for these signals must be applied at the higher, functional level.

AUTOSAR does not provide a Safety Concept for Powertrain systems. This must be done at the project level. This means that the specified interfaces must be checked to fulfill the safety requirements on each specific project.

7.3 Powertrain Application Interfaces - Decisions / Assumptions

7.3.1 Scope

In this document only passenger cars are considered.

7.3.2 PTC Composition (PtCoorr)

The PTC is not an atomic AUTOSAR SW-Component. In fact its functionalities should be separated, into several sub-components. These sub-components will communicate with each other and with AUTOSAR SW-Components outside the PTC. The interfaces between the sub-components are not in the current scope, which is restricted to the definition of main interfaces between the non-PTC components and the PTC sub-components.

7.3.3 Definition of overboost

Overboost is a state in which the maximum torque which the combustion engine can deliver is increased for a limited period of time. Depending on the engine type, this could be realised, for example, as an increase in boost pressure on a turbocharged engine.

7.3.4 Coordination at the vehicle level

Coordination of vehicle energy (mechanical / electrical / thermal), vehicle operation modes, vehicle personalisation, etc., should be done at the vehicle level. This is not in the scope of the Powertrain Application Interfaces.

The composition VehMtnForPtMisc was added to [2] as an interim solution for some vehicle level issues relevant to the powertrain domain.

7.3.5 PTC Arbitration between Driver and Chassis torque requests

Figure 25 and 26, unterhalb, shows how the VLC and Stability Control torques requests could be arbitrated with the Driver Request. This is just an example to illustrate the concept behind the powertrain torque request interfaces defined in [2], it is not intended to standardise the arbitration behavior in the PTC.

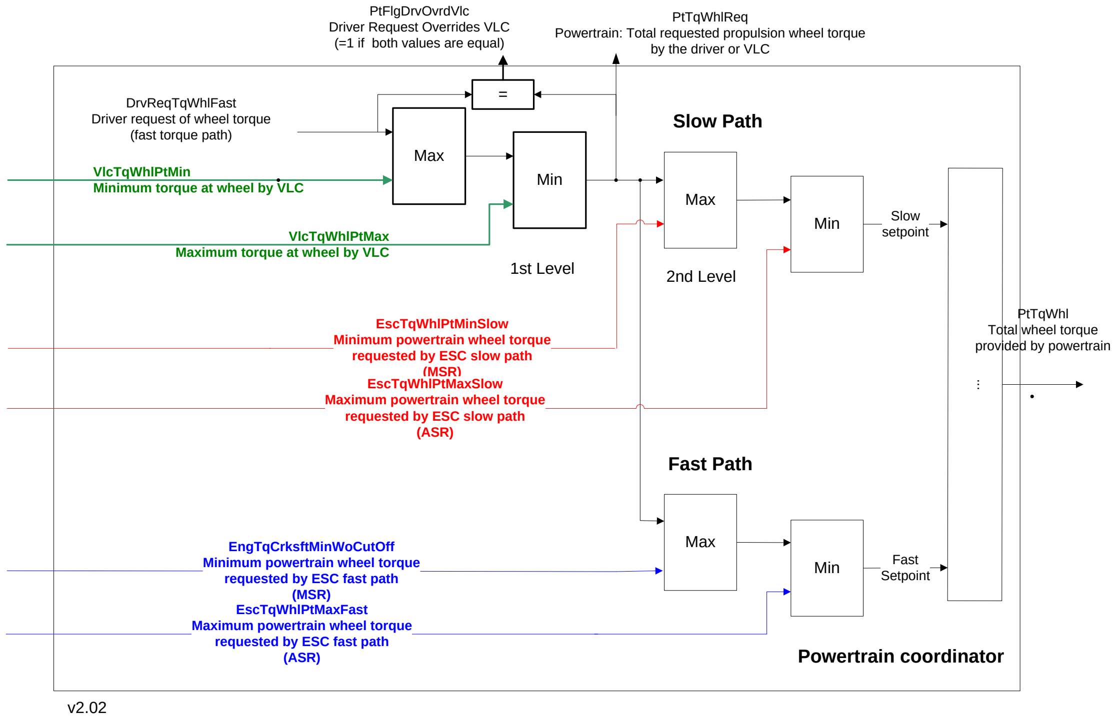

Figure 25: Example of possible PTC arbitration between Driver and Chassis torque requests (request based on wheel torque)

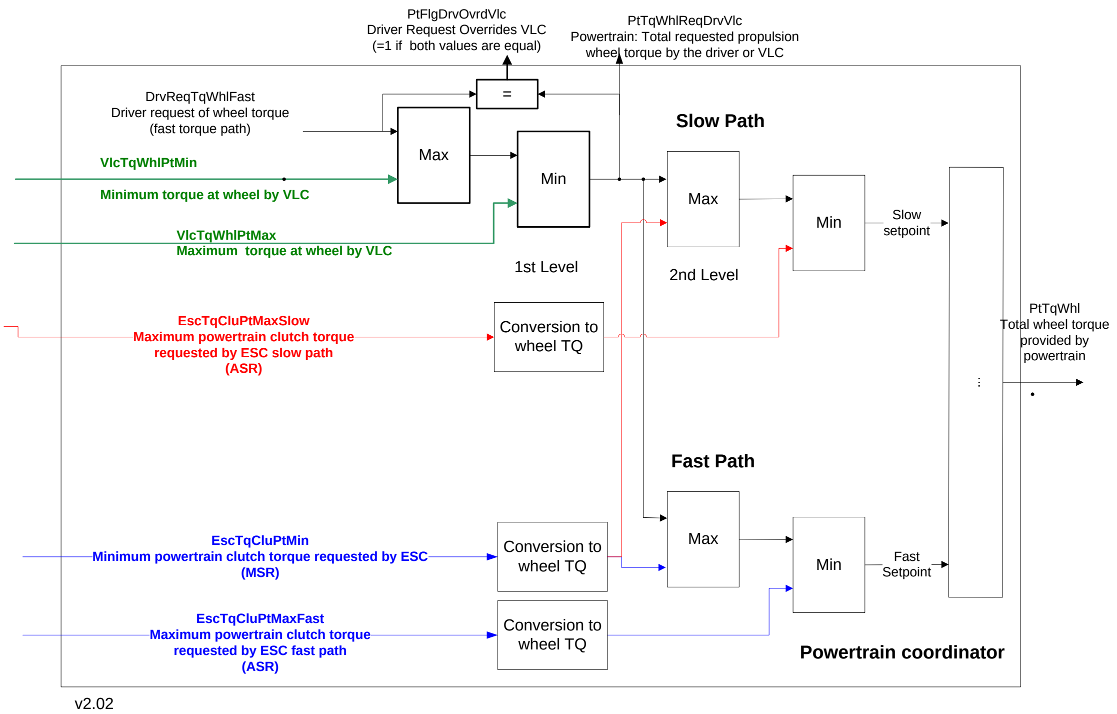

Figure 26: Example of possible PTC arbitration between Driver and Chassis torque requests (request based on clutch torque)

7.3.6 Assumptions on modeling style and naming aspects specific for powertrain domain

AUTOSAR provides a guideline for modeling and naming of model elements ([1]).

There are architectural design patterns like the sensor actuator design pattern described in chapter 6 that also include modeling and naming aspects.

In this section only additional patterns and modeling styles followed are explained to get an overall understanding of the signals standardized for the powertrain domain.

Please note: Here standardized ports or port interfaces mean standardized port prototype blueprints or port interface blueprints [7].

Kind of Modeling in general applied within powertrain domain, especially if the system modeling guideline [1] gives some freedom**.**

Kind of modeling or assumptions	Rationale
All *SenderReceiverInterfaces* standardized are assumed to be measurable.	In earlier versions of the standard [2] the standard did not contain information about calibration and measurement. Since R4.0.3 all data types allow measurement by default (see generated .arxml [3]). So our implicit assumption that all signals are measurable is fulfilled.
All ports are assumed to be optional.	Within our example components all ports are assumed to be optional. The ports are derived from the port prototype blueprints with the same name. It is optional per default that port prototype blueprints are allowed to be used but not necessarily used in every project. In previous releases without blueprints this assumption was very important because there was no variant handling done in [2]. So within the powertrain domain it was assumed that all ports are optional. Since only ports but no components are standardized this was even more important: it means that a supplier or OEM may create a single SW-C (Software Component) and use only the standardized ports that are relevant for this SW-C in his sw architecture.
Port interfaces are not designed to be reused: there is a 1:1 relationship between port and port interface. The port interface has the same name as the port + an additional index as required by the System Modeling Guideline [1]. Exception: If powertrain is not the provider then the rules of	Ports are attached to SW-C. Since SW-C are not standardized only port interfaces were really subject of usage in projects up to Release 3.1. With Release 4.0 the standardization of ports is supported by using so-called *PortPrototypeBlueprints* in the meta model. However, in practice older versions of the AUTOSAR meta model are still in use and the existing tools do not yet fully support *PortPrototypeBlueprints*. To be backward compatible and to enable the easy introduction of the standardized ap-

Kind of Modeling in general applied within powertrain domain, especially if the system modeling guideline [1] gives some freedom**.**	
Kind of modeling or assumptions	**Rationale**
other domains are respected.	plication interfaces within the powertrain domain not all features of the meta model (like e.g. connectiong of compatible interfaces with different port interface short names) therefore were yet fully exploited. A second reason was that [2] does not support connecting of compatible interfaces with different port interface short names.
If a port interface contains exactly one data prototype its name is identical to the port interface excluding the trailing index.	There were only two alternatives: - using full name - using name "Val" for Value Disadvantage of solution 2) would have been that many ports would have been assumed to be compatible by tools because identical data prototype names (with compatible interfaces) allow an automatic connection. Within specification tools like e.g. ASCET-SD or MATLAB/Simulink it might be possible to only show the data prototype name and to not display the port interface or port name.
The reuse of data types was explicit goal within the powertrain domain.	Within the powertrain domain it was an important design goal to use as few data types as possible.
Computation methods were not subject to reuse yet.	The [2] does not support the reuse of computation methods, only of data types. But reuse of computation methods should be considered for the next release.
In most cases only one data prototype was defined per port interface.	This kind of modeling allows the biggest flexibility in implementing the standard. On assembly level e.g. it is not allowed that the r-port has more data prototypes than the p-ports ([6], figure 6.3). Therefore in such cases, when several data prototypes are part of one r-port the data prototypes themselves cannot be assumed to be optional, only the complete r-port.

Kind of Modeling in general applied within powertrain domain, especially if the system modeling guideline [1] gives some freedom**.**	
Kind of modeling or assumptions	**Rationale**
	In older versions of the AUTOSAR standard is was not allowed that a sub-component provides only part of the port, i.e. if several data prototypes were part of a port interface it was implicitly standardized that there is one SW-C providing it. When splitting the information on several port interfaces each data prototype might be provided by a different SW-C.
In most cases no records were used to define port interfaces.	The rationale is similar to the one stated for the assumption to only define one *DataPrototype* per *PortInterface*. Since within the powertrain domain *PortInterfaces* with multiple *DataPrototypes* are seldomly used, there is no necessity to use records and it is assumed that timing related aspects of the data prototypes are to be handled separately, that there are other more flexible possibilities to do so.

Specific Naming Assumptions for Powertrain Domain	
Assumption	**Rationale**
The names were chosen such that automatic generation of display names is possible.	In Powertrain ECUs there are thousands of calibration relevant data. So it is important to apply the System Modeling Guideline [1] in a way that automatic generation of display names is possible. See [5] for details.
In general the keyword abbreviation “Consold” for “Consolidated” as well as the abbreviation “Act” for “Actual” was suppressed.	Reason for doing so is to have short names whenever possible (see [TR_MCM_70020] in [5]) Example: *PtTqClu* for “Torque at Clutch “
In case a port is only for Fast Path or Slow Path “Fast” and “Slow” is added. In the other cases “SlowFast” is suppressed.	Reason for doing so is to have short names whenever possible (see [TR_MCM_70020] in [5]) Example: *PtTqWhlMinWoCutOff* affects Slow and

Specific Naming Assumptions for Powertrain Domain	
Assumption	**Rationale**
	Fast path *EscTqWhlPtMaxSlow* or *EscTqWhlPtMaxFast* is for a specific path For more details on fast and slow paths see chapter 3.3
"State" is used as abbreviation "St" for "State" and "Status".	In most cases it is very difficult to explain the difference between State and Status. So for sake of simplicity and consistency within the powertrain domain only "St" is used.
TqWhl means sum of Torque Wheels whereas TqWhlInd means the torque of an individual wheel.	Within Powertrain the sum of torque wheels is more often used. So information ("Ind") is added in the case that individual wheel torques are meant. Example: *PtTqWhl* for "Total Wheel Torque Provided by Powertrain" *PtTqWhlInd* for "Torque Delivered by the Powertrain to the Individual Wheels"
Prepositions are only used in the short name if it really helps understanding.	In most cases the preposition does not really add information and makes short names unnecessarily long (see [TR_MCM_70020] in [5]). Examples: *PtTqWhlMinWoCutOff* for Minimum Possible Total Powertrain Torque at Wheels Without Complete Fuel Cut Off using the preposition 'Wo' (without) *PtTqWhlReqDrvVlc* for Powertrain: Total Requested Propulsion Wheel Torque by the Driver or Vehicle Longitudinal Control (VLC) instead of *PtTqAtWhlReqByDrvForVlc* as recommended by the system modeling guideline [1].

8 Appendix: Mapping Ports to Display Names - Powertrain Domain

In the following display names for the standardized port prototype blueprints are defined. It is recommended to use the display name without name space identifier “AR_” in case no naming conflicts are expected within the system or for the ECU. In all other cases it is recommended to use the display name with name space identifier.

The rules of [5] for generating display names are followed.
There are only the following exceptions in which a name was choosen manually:
EscVWhlInd: Instead of numbers for the index element a meaningful name part was added.
EscSts: We always use „st“ according to original Keywords Phys/Log (P/L-List) for Powertrain Domain.

The virtual name space for ports is the top-level package AUTOSAR abbreviated with AR_ (see [TR_MCM_70040] in [5]). Virtual name spaces are described in [5].
Port prototype blueprint names are unique within top-level package “AUTOSAR”.
The sub packages of the top-level package AUTOSAR partly define virtual name spaces (e.g. for data types and port interfaces) but none of these name spaces are relevant for the generation of display names.
Basis for the definition of the display names is [2], sheets „0502*“ for Powertrain.

Note: Long Names might differ in [2] (in content but not in meaning) since not all long names were yet made consistent with new sensor/actuator pattern. Consolidation with Chassis application interfaces might also lead to additional long name changes.

DisplayName w/o Name Space. With Name Space add AR_ before name, e.g. AR_Abs_flgActv	Shortname of Port / additional information if needed	Longname of Displayname (= PortPrototypeBlueprint name extended in case of multiple data prototypes or arrays)
Abs_flgActv	AbsFlgActv	Antilock Braking System (ABS) Con-

		trol Active
AccrPedl_rat	AccrPedlRat	Accelerator Pedal Ratio
AccrPedl_ratFild	AccrPedlRatFild	Filtered Accelerator Pedal Ratio
AccrPedl_ratGrdt	AccrPedlRatGrdt	Accelerator Pedal Ratio Gradient
Alt_tq	AltTq	Alternator Mechanical Torque (Load)
Alt_tqReq	AltTqReq	Requested Mechanical Torque for Alternator (Generator) at Engine Crank Shaft
AxleFrntCoorr_st	AxleFrntCoorrSt	Status of the Front Axle Coordinator
AxleReCoorr_st	AxleReCoorrSt	Status of the Rear Axle Coordinator
Batt_u	BattU	Battery Voltage
BrkPedl_flgPsd	BrkPedlFlgPsd	Brake Pedal Pressed
BrkPedl_rat	BrkPedlRat	Brake Pedal Position
CluPedl_rat	CluPedlRat	Clutch Pedal Ratio
Drv_flgGearShiftDwnReq	DrvFlgGearShiftDwnReq	Gear Shift Down Request by Driver
Drv_flgGearShiftUpReq	DrvFlgGearShiftUpReq	Gear Shift Up Request by Driver
Drv_flgKdDetd	DrvFlgKdDetd	Driver: Kickdown Detected
DrvReq_ratVirtAccrPedl	DrvReqRatVirtAccrPedl	Driver Request: Virtual Accelerator Pedal Ratio
DrvReq_tqCluFast	DrvReqTqCluFast	Driver Request of Clutch Torque (Fast Torque Path)
DrvReq_tqCluSlow	DrvReqTqCluSlow	Driver Request of Clutch Torque (Slow Torque Path)
DrvReq_tqWhlFast	DrvReqTqWhlFast	Driver Request of Wheel Torque (Fast Torque Path)
DrvReq_tqWhlSlow	DrvReqTqWhlSlow	Driver Request of Wheel Torque (Slow

		Torque Path)
Dtd_ratTqDistbnReqWhlFrntLe	DtdRatTqDistbnReq / Front Left Wheel	Requested Drivetrain Torque Distribution - Front Left Wheel
Dtd_ratTqDistbnReqWhlFrntRi	DtdRatTqDistbnReq / Front Right Wheel	Requested Drivetrain Torque Distribution - Front Right Wheel
Dtd_ratTqDistbnReqWhlReLe	DtdRatTqDistbnReq / Rear Left Wheel	Requested Drivetrain Torque Distribution - Rear Left Wheel
Dtd_ratTqDistbnReqWhlReRi	DtdRatTqDistbnReq / Rear Right Wheel	Requested Drivetrain Torque Distribution - Rear Right Wheel
Dtd_tqDftlAxleFrntReq	DtdTqDftlAxleFrntReq	Drivetrain Torque Distribution: Requested Differential Torque at Front Axle Actuator
Dtd_tqDftlAxleReReq	DtdTqDftlAxleReReq	Drivetrain Torque Distribution: Requested Differential Torque at Rear Axle Actuator
Dtd_tqDftlTrfReq	DtdTqDftlTrfReq	Drivetrain Torque Distribution: Requested Differential Torque at Transfer Case
EgyMngt_st	EgyMngtSt	State of Energy Management
Eng_n	EngN	Engine Speed
Eng_nGearTar	EngNGearTar	Engine Speed at Target Gear
Eng_nGrdt	EngNGrdt	Engine Speed Gradient
Eng_nMax	EngNMax	Maximum Allowed Engine Speed
Eng_nMin	EngNMin	Minimum Allowed

		Engine Speed
Eng_pAmbAir	EngPAmbAir	Engine Ambient Air Pressure
Eng_t	EngT	Engine Temperature
Eng_tqCrksft	EngTqCrksft	Engine Torque at Crankshaft
Eng_tqCrksftMax	EngTqCrksftMax	Maximum Engine Torque at Crank-shaft
Eng_tqCrksftMaxFast	EngTqCrksftMaxFast	Maximum Engine Torque at Crank-shaft Fast Path
Eng_tqCrksftMaxOptmCdn	EngTqCrksftMaxOptmCdn	Maximum Engine Torque at Crank-shaft at Optimum Conditions
Eng_tqCrksftMaxProtn	EngTqCrksftMaxProtn	Maximum Allowed Torque at Crank-shaft for Engine Pro-tection
Eng_tqCrksftMin	EngTqCrksftMin	Minimum Engine Torque at Crank-shaft considering all engine losses
Eng_tqCrksftMinBasc	EngTqCrksftMinBasc	Minimum Engine Torque at Crank-shaft for Powertrain realized by Slow Path
Eng_tqCrksftMinFast	EngTqCrksftMinFast	Minimum Engine Torque at Crank-shaft Fast Path
Eng_tqCrksftMinWoCutOff	EngTqCrksftMinWoCutOff	Minimum Engine Torque at Crank-shaft for Powertrain realized by Slow and Fast Path
Eng_tqDynJ	EngTqDynJ	Engine Dynamic Moment of Inertia
Eng_tqResvPt	EngTqResvPt	Torque Reserve Re-quest from Engine to Powertrain

Esc_flgNoFuCutOff	EscFlgNoFuCutOff	Request "No Fuel Cut Off" by Electronic Stability Control (ESC)
Esc_st	EscSts	Electronic Stability Control (ESC) Status
Esc_stShiftPrevnStaby	EscStShiftPrevnStaby	Electronic Stability Control (ESC): State of Shift Prevention for Stability Reasons
Esc_tqCluPtMaxFast	EscTqCluPtMaxFast	Maximum Powertrain Clutch Torque Requested by Electronic Stability Control (ESC) Fast Path
Esc_tqCluPtMaxSlow	EscTqCluPtMaxSlow	Maximum Powertrain Clutch Torque Requested by Electronic Stability Control (ESC) Slow Path
Esc_tqCluPtMin	EscTqCluPtMin	Minimum Powertrain Clutch Torque Requested by Electronic Stability Control (ESC)
Esc_tqWhlPtMaxFast	EscTqWhlPtMaxFast	Maximum Powertrain Wheel Torque Requested by Electronic Stability Control (ESC) Fast Path
Esc_tqWhlPtMaxSlow	EscTqWhlPtMaxSlow	Maximum Powertrain Wheel Torque Requested by Electronic Stability Control (ESC) Slow Path
Esc_tqWhlPtMinFast	EscTqWhlPtMinFast	Minimum Powertrain Wheel Torque Requested by Electronic Stability Control (ESC) Fast Path
Esc_tqWhlPtMinSlow	EscTqWhlPtMinSlow	Minimum Powertrain Wheel Torque Re-

		quested by Electronic Stability Control (ESC) Slow Path
Esc_vMax	EscVMax	Maximum Vehicle Speed due to Electronic Stability Control (ESC)
Esc_vVhlIndFrntLe	EscVWhlInd / Index 0	Electronic Stability Control (ESC): Vector of Individual Speed of Wheels - Front Left
Esc_vVhlIndFrntRi	EscVWhlInd / Index 1	Electronic Stability Control (ESC): Vector of Individual Speed of Wheels - Front Right
Esc_vVhlIndReLe	EscVWhlInd / Index 2	Electronic Stability Control (ESC): Vector of Individual Speed of Wheels - Rear Left
Esc_vVhlIndReRi	EscVWhlInd / Index 3	Electronic Stability Control (ESC): Vector of Individual Speed of Wheels - Rear Right
Hvac_tq	HvacTq	Mechanical Torque for A/C Compressor
Hvac_tqReq	HvacTqReq	Requested Mechanical Torque for A/C Compressor
Outd_t	OutdT	Outdoor Temperature
Pt_flgAltDeactvt	PtFlgAltDeactvt	Powertrain: Request to Deactivate Alternator (Generator)
Pt_flgDrvOvrdVlc	PtFlgDrvOvrdVlc	Powertrain: Driver Request Overrides Vehicle Longitudinal Control (VLC)
Pt_flgEngRun	PtFlgEngRun	Powertrain: Engine

		is Running
Pt_flgEngStop	PtFlgEngStop	Powertrain: Engine is Stopped
Pt_flgEngStopReq	PtFlgEngStopReq	Powertrain: Engine Stop Request
Pt_flgEngStopReqAllwd	PtFlgEngStopReqAllwd	Powertrain: Request to Stop Engine is Allowed
Pt_flgEngStrtReq	PtFlgEngStrtReq	Powertrain: Engine Start Request
Pt_flgEngStrtReqAllwd	PtFlgEngStrtReqAllwd	Powertrain: Request to Start Engine is Allowed
Pt_flgHvacDeactvt	PtFlgHvacDeactvt	Powertrain: Request to Deactivate Air Conditioner (A/C)
Pt_flgNoTqWhlReq	PtFlgNoTqWhlReq	Powertrain: No Torque Request for Wheel Torque
Pt_flgTqDecPsbl	PtFlgTqDecPsbl	Powertrain: Torque Decrease Possible
Pt_flgTqIncPsbl	PtFlgTqIncPsbl	Powertrain: Torque Increase Possible
Pt_nClu	PtNClu	Powertrain: Speed at Clutch
Pt_nEngSp	PtNEngSp	Powertrain: Engine Speed Setpoint
Pt_ratTqDistbnReqFrntLe	PtRatTqDistbnReq	Powertrain: Requested Percental Distribution of Torque to Wheels - Front Left Wheel
Pt_ratTqDistbnReqFrntRi	PtRatTqDistbnReq	Powertrain: Requested Percental Distribution of Torque to Wheels - Front Right Wheel
Pt_ratTqDistbnReqReLe	PtRatTqDistbnReq	Powertrain: Requested Percental Distribution of Torque to Wheels - Rear Left Wheel

Pt_ratTqDistbnReqReRi	PtRatTqDistbnReq	Powertrain: Requested Percental Distribution of Torque to Wheels - Rear Right Wheel
Pt_tqClu	PtTqClu	Powertrain: Torque at Clutch
Pt_tqCluDyn	PtTqCluDyn	Powertrain: Dynamic Torque at Clutch
Pt_tqCluMaxDrv	PtTqCluMaxDrv	Powertrain: Maximum Available Torque at Clutch for Driver
Pt_tqCluMinDrv	PtTqCluMinDrv	Powertrain: Minimum Available Torque at Clutch for Driver
Pt_tqCluReqWoTrsmIntv	PtTqCluReqWoTrsmIntv	Powertrain: Torque at Clutch Request Without Transmission Intervention
Pt_tqCluTarGear	PtTqCluTarGear	Powertrain: Predicted Torque at Clutch at Target Gear
Pt_tqCluWoTrsmIntv	PtTqCluWoTrsmIntv	Powertrain: Torque at Clutch Without Transmission Intervention
Pt_tqCrksftReqFast	PtTqCrksftReqFast	Crankshaft Torque Request to be realized by the Fast Path of Powertrain
Pt_tqCrksftReqSlow	PtTqCrksftReqSlow	Crankshaft Torque Request to be realized by the Slow Path of Powertrain
Pt_tqMaxAlt	PtTqMaxAlt	Powertrain: Maximum Allowed Mechanical Load for Alternator (Generator) at Engine Crank Shaft.

Pt_tqMaxHvac	PtTqMaxHvac	Powertrain: Maximum Allowed Torque for A/C Compressor
Pt_tqMaxSteer	PtTqMaxSteer	Powertrain: Maximum Allowed Mechanical Load for Steering at Engine Crank Shaft
Pt_tqMaxTrsmOilPmp	PtTqMaxTrsmOilPmp	Maximum Allowed Transmission Oil Pump Load
Pt_tqResvEng	PtTqResvEng	Torque Reserve Request from Powertrain to Engine
Pt_tqWhl	PtTqWhl	Total Wheel Torque Provided by Powertrain
Pt_tqWhlIndFrntLe	PtTqWhlInd	Torque Delivered by the Powertrain to the Individual Wheels - Front Left Wheel
Pt_tqWhlIndFrntRi	PtTqWhlInd	Torque Delivered by the Powertrain to the Individual Wheels - Front Right Wheel
Pt_tqWhlIndReLe	PtTqWhlInd	Torque Delivered by the Powertrain to the Individual Wheels - Rear Left Wheel
Pt_tqWhlIndReRi	PtTqWhlInd	Torque Delivered by the Powertrain to the Individual Wheels - Rear Right Wheel
Pt_tqWhlMax	PtTqWhlMax	Powertrain: Maximum Possible Total Torque at Wheels
Pt_tqWhlMaxDrv	PtTqWhlMaxDrv	Maximum Possible Total Torque at Wheels at Optimum Conditions Delivered by Powertrain for

		Driver
Pt_tqWhlMin	PtTqWhlMin	Minimum Available Total Powertrain Torque at Wheels
Pt_tqWhlMinDrv	PtTqWhlMinDrv	Minimum Possible Total Torque at Wheels Delivered by Powertrain for Driver
Pt_tqWhlMinWoCutOff	PtTqWhlMinWoCutOff	Minimum Possible Total Powertrain Torque at Wheels Without Complete Fuel Cut Off
Pt_tqWhlReq	PtTqWhlReq	Powertrain: Total Requested Propulsion Wheel Torque
Pt_tqWhlReqDrvVlc	PtTqWhlReqDrvVlc	Powertrain: Total Requested Propulsion Wheel Torque by the Driver or Vehicle Longitudinal Control (VLC)
Ssm_flgGearParkReq	SsmFlgGearParkReq	Standstill Manager: Request to Engage the Parking Lock
Steer_tq	SteerTq	Hydraulic Power Steering Load at Engine Crank Shaft
Steer_tqReq	SteerTqReq	Requested Hydraulic Power Steering Load at Engine Crank Shaft
TrfCaseCoorr_st	TrfCaseCoorrSt	Status of the Transfer Case Coordinator
TrsmClu_stAct	TrsmCluStAct	Transmission: Actual Clutch State
TrsmClu_stTar	TrsmCluStTar	Transmission: Target State of the Clutch
Trsm_flgCtrsftActv	TrsmFlgCtrsftActv	Transmission: Countershaft Active
Trsm_flgDtOpen	TrsmFlgDtOpen	Transmission:

		Drivetrain Opened
Trsm_flgParkLockEngd	TrsmFlgParkLockEngd	Transmission: Park Lock Engaged
Trsm_flgShiftProgs	TrsmFlgShiftProgs	Transmission: Flag indicates that a Gear Shift is In Progress
Trsm_flgSptMod	TrsmFlgSptMod	Transmission: Sport Mode Request by Driver
Trsm_flgWntrMod	TrsmFlgWntrMod	Transmission: Winter Mode Request by Driver
Trsm_nInp	TrsmNInp	Transmission: Speed at Input
Trsm_nrGearAct	TrsmNrGearAct	Transmission: Actual Gear
Trsm_nrGearReq	TrsmNrGearReq	Transmission: Requested Gear
Trsm_nrGearTar	TrsmNrGearTar	Transmission: Target Gear
Trsm_nrTyp	TrsmNrTyp	Transmission Type
Trsm_ratGear	TrsmRatGear	Transmission: Get the Gear Ratio of the Gear of Interest (C/S)
Trsm_ratGearAct	TrsmRatGearAct	Transmission: The Actual Gear Ratio being Currently Engaged in the Gear Box
Trsm_ratGearReq	TrsmRatGearReq	Transmission: Requested Gear Ratio
Trsm_ratGearTar	TrsmRatGearTar	Transmission: The Gear Ratio which will be Engaged in the Gear Box when Target Gear is Reached
Trsm_ratTqPtAct	TrsmRatTqPtAct	Transmission: Actual Powertrain Torque Ratio
Trsm_stAxelFrntActr	TrsmStAxleFrntActr	Transmission: Status of the Front Axle Ac-

		tuator
Trsm_stAxelReActr	TrsmStAxleReActr	Transmission: Status of the Rear Axle Actuator
Trsm_stGearLvr	TrsmStGearLvr	Transmission: Actual Gear Lever Position
Trms_stTrfCaseDftl	TrsmStTrfCaseDftl	Transmission: Status of the Transfer Case Differential
Trsm_tOil	TrsmTOil	Transmission: Oil Temperature
Trsm_tqCluMaxFast	TrsmTqCluMaxFast	Transmission: Maximum Torque at Clutch Requested by Transmission for Shift Energy Management on Fast Path
Trsm_tqCluMaxProtn	TrsmTqCluMaxProtn	Transmission: Maximum Torque at Clutch Requested by Transmission for Gearbox Protection
Trsm_tqCluMaxSlow	TrsmTqCluMaxSlow	Transmission: Maximum Torque at Clutch Requested by Transmission for Shift Energy Management on Slow Path
Trsm_tqCluMinFast	TrsmTqCluMinFast	Transmission: Minimum Torque at Clutch Requested by Transmission for Shift Energy Management on Fast Path
Trsm_tqCluMinSlow	TrsmTqCluMinSlow	Transmission: Minimum Torque at Clutch Requested by Transmission for Shift Energy Man-

		agement on Slow Path
Trsm_tqDftlAxleFrntAct	TrsmTqDftlAxleFrntAct	Transmission: Actual Differential Torque at Front Axle
Trsm_tqDftlAxleFrntMax	TrsmTqDftlAxleFrntMax	Transmission: Maximum Differential Torque at Front Axle
Trsm_tqDftlAxleReAct	TrsmTqDftlAxleReAct	Transmission: Actual Differential Torque at Rear Axle
Trsm_tqDftlAxleReMax	TrsmTqDftlAxleReMax	Transmission: Maximum Differential Torque at Rear Axle
Trsm_tqDftlTrfAct	TrsmTqDftlTrfAct	Transmission: Actual Differential Transfer Torque
Trsm_tqDftlTrfMax	TrsmTqDftlTrfMax	Transmission: Maximum Differential Transfer Torque
Trsm_tqOilPmp	TrsmTqOilPmp	Transmission Oil Pump Load
Trsm_tqOilPmpReq	TrsmTqOilPmpReq	Requested Transmission Oil Pump Load
Trsm_tqWhlIndDistbnFrntLe	TrsmTqWhlIndDistbn	Individual Torque at Wheel Distribution as realized by Transmission System - Front Left Wheel
Trsm_tqWhlIndDistbnFrntRi	TrsmTqWhlIndDistbn	Individual Torque at Wheel Distribution as realized by Transmission System - Front Right Wheel
Trsm_tqWhlIndDistbnReLe	TrsmTqWhlIndDistbn	Individual Torque at Wheel Distribution as realized by Transmission System - Rear Left

		Wheel
Trsm_tqWhlIndDistbnReRi	TrsmTqWhlIndDistbn	Individual Torque at Wheel Distribution as realized by Transmission System - Rear Right Wheel
Veh_stOper	VehStOper	Vehicle Operating State
Veh_tqWhlMaxSftyLimn	VehTqWhlMaxSftyLimn	Safety Torque Limit calculated from the Vehicle's Maximum Speed
Veh_vLgt	VehVLgt	Vehicle Speed (Longitudinal)
Vlc_stShiftPrevnCmft	VlcStShiftPrevnCmft	Vehicle Longitudinal Control (VLC): State of Shift Prevention for Comfort Reasons
Vlc_tqWhlPtMax	VlcTqWhlPtMax	Maximum Torque at Wheel by Vehicle Longitudinal Control (VLC)
Vlc_tqWhlPtMin	VlcTqWhlPtMin	Minimum Torque at Wheel by Vehicle Longitudinal Control (VLC)