# Preliminary OC Design Specification

Version 0.1_published, 27.07.2018

## 1 Disclaimer

**OC-10040 -**

This document is a DRAFT version which is still under construction. Its content may change in the ongoing concept phase of SmartRail 4.0. The document is not completely verified and is not finalized by now. The document is published to enable an open discussion of the ongoing work of the SmartRail 4.0 program.

Links and references inside of this document may refer to other documents inside of the program SmartRail 4.0, that may not be published at this stage.

# Content

# List of Figures

## 2 Glossary

### OC-11305 -

| Term | Abbrev. | Description |
|---|---|---|
| **Clear Track Signalling Installation** | CTS | Trackside installation that indicates track clearance (occupation or non-occupation). |
| **Commercial off-the-shelf** | COTS | Commercial items, including services, available in the commercial marketplace that can be bought and used, without the need to commission custom-made, or bespoke, solutions. |
| **ETCS Interlocking** | EI | ETCS cab-signalling based interlocking comprising the Radio Block Center (RBC). Its dynamic, rule based and geometric safety logic controls all movements of the objects and all changes of the state of the Trackside Assets (TA) within the EIs effective range. All operational logic is moved to the higher-level systems. |
| **Level crossing** | LX | A level crossing is an intersection where a railway line crosses a road or path at the same level. |

| | | |
|---|---|---|
| **Object Controller** | OC | The Object Controller connects the ETCS Interlocking (EI) with the trackside assets (TA) by translating Commands/Messages between ETCS Interlocking and trackside asset (e.g. point motor). |
| **Rail Safe Transport Application** | RaSTA | It is a network protocol especially designed to meet the requirements of railway applications, but that can be also used in other areas with similar requirements. |
| **smartrail 4.0** | SR40 | A program with disruptive innovations for the processes and systems of the railway production. |

## 3 Overview

The underlying document describes a possible architecture and design for the Object Controller (OC). This stage of design can be used as preliminary specification for the OC.

## 4 Overall SR40 System Topology

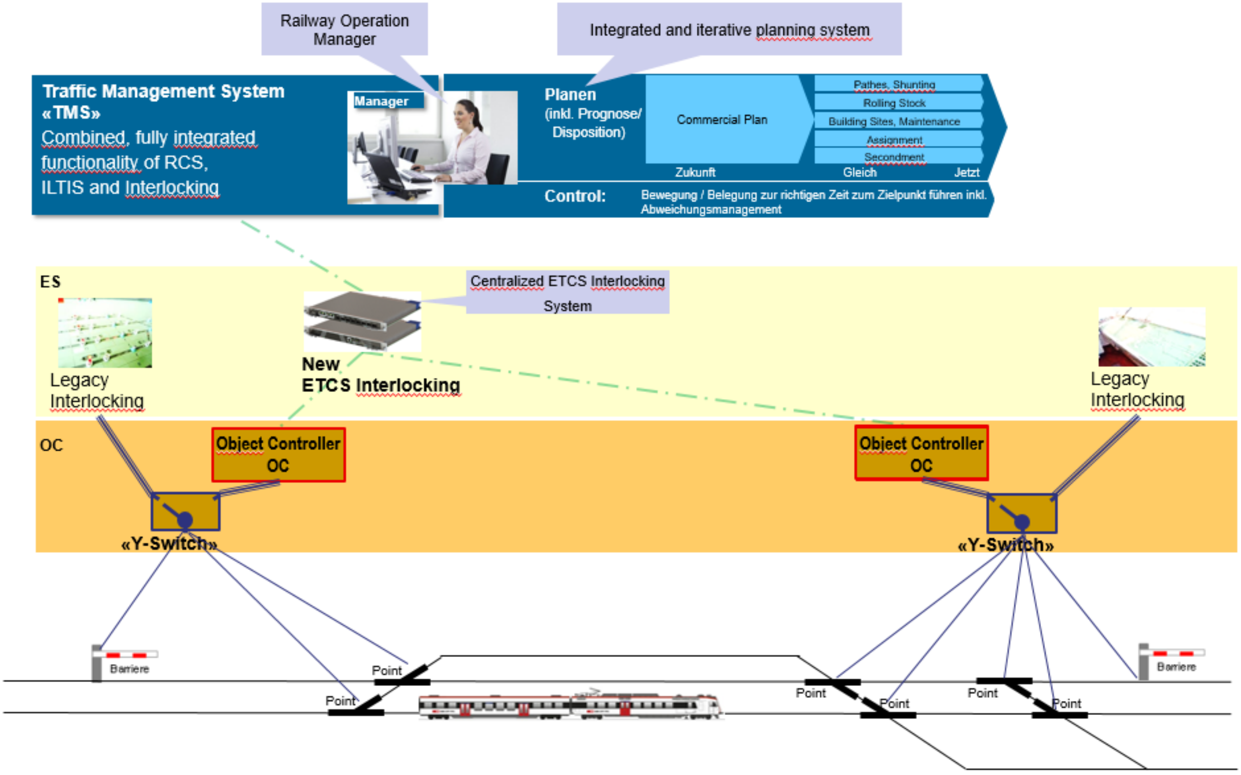
Railway Operation Manager
Integrated and iterative planning system
Traffic Management System «TMS»
Combined, fully integrated functionality of RCS, ILTIS and Interlocking
Manager
Planen (inkl. Prognose/ Disposition)
Commercial Plan
Pathes, Shunting
Rolling Stock
Building Sites, Maintenance
Assignment
Secondment
Zukunft
Gleich
Jetzt
Control:
Bewegung / Belegung zur richtigen Zeit zum Zielpunkt führen inkl. Abweichungsmanagement
ES
Centralized ETCS Interlocking System
Legacy Interlocking
New ETCS Interlocking
Legacy Interlocking
OC
Object Controller OC
Object Controller OC
«Y-Switch»
«Y-Switch»
Barriere
Point
Point
Point
Point
Point
Point
Barriere

*Figure 1 smartrail 4.0 overall functionality*

The illustration above shows the overall smartrail 4.0 system topology. Main instances for the overall functionality are the cetralized Traffic Management and ETCS Interlocking systems (TMS, EI). The OC acts as gateway between ETCS Interlocking and existing Trackside Assets.

The main functionality of the OC is to control Trackside Assets (TA) like points and level crossings. In this role the OC acts as subsystem of the ETCS interlocking system EI.

There will exist (at least logically spoken) one central ETCS interlocking system and thousands of Object Controllers. EI provides services for centralized interlocking and the OC for controlling and monitoring Trackside Assets. A peer-to-peer connection ensures the communication and therefore the interaction between the two parties.

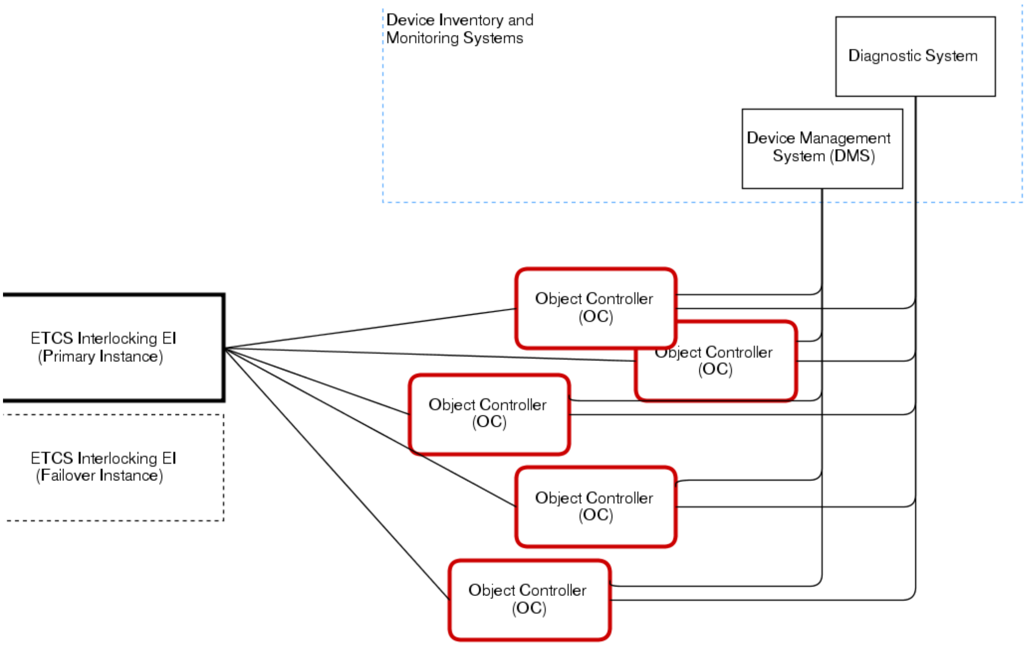

*Figure 2 1:n cardinality between EI and OC*

The OC will be also connected to smartrail 4.0 system Device Management and Diagnostic systems. Over this channels the OC will be provided with configuration data on one hand and be monitored by collecting runtime information data on the other.

Safety requirements have to be addressed by applying a specific safety architecture with redundancy and other adequate measures.

### 4.1 EI - OC Interaction

In interaction with EI the OC has two primary functions. First it must process commands from EI to change states of TAs. For this kind of interaction the EI acts as master and the OC as slave. Further, the OC must send Trackside Asset status data (e.g. occupancy information from Clear Track Signaling Installations CTS) in safe way to EI.

However the overall behavior of EI - OC interaction has an asynchronous, stateless character. Regarding the kind how EI controls Trackside Assets we notice a kind of order management.

TMS has to treat the railway production plan. It determines the routes and organizes the train movements. Using railway infrastructure and sending Movement Permissins to trains is done via ETCS interlocking system (EI). EI ensures the safety for all interactions between trains amgong tehmselves and among infrastructure.

Reference for the current state of the whole infrastructure and the complete rolling stock is the Operation State. All actions that are requested by TMS and should be granted by EI are checked against the Operation State.

The OC will not have to verify actions sent by EI against any overall safety criteria. They are granted and have to be treated by the OC in a safe way.

Collecting and sending states of Trackside assets to EI is at least as important as performing actions. Only safe and time close backporting of TA states guarantees that the Operation State is acutalized properly.

### 4.2 OC as host for many Logical Trackside Assets

In a long-term perspective it is planned that every Trackside Asset has its own integrated object controller functionality. This would mean that every TA can communicate with EI in a Internet Of Things (IOT) manner.

In the meantime a rack mounted device which is located in interlocking room will realize the communication between EI and the Trackside Assets.

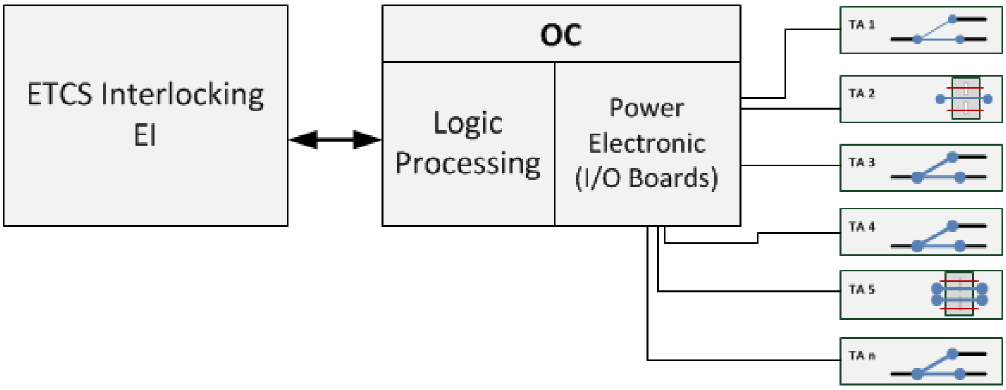

*Figure 3 One rack mounted OC connects multiple Trackside Assets*

Inside the OC one Software control unit called Base Module acts as communication endpoint towards EI. From a logical point of view the OC groups a set of Trackside Assets which are physically connected to that device. Trackside Assed functionality is built Software based. Therefore an abstracted physical Trackside Asset is called *Logical Trackside Asset*. Thus towards EI the OC acts as provider for a set of Logical Trackside Assets.

For controlling Trackside Assets the OC works fully transparently. EI communicates with Logical Trackside Assets and not with the OC device. The OC as device must be known by EI only for communication purposes. It must be ensured that commands to a certain TA are sent to the correct OC to which the TA is connected.

### 4.3 Communication Environment

The OC communicates with up to four pheripheral systems which are part of the operational environment of the smartrail 4.0 overall system.

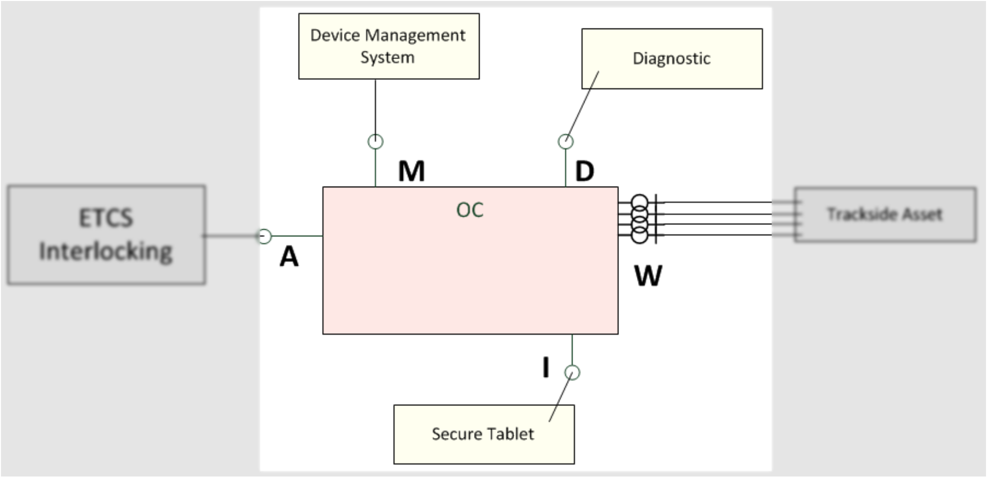

*Figure 4 Communication Interfaces of OC*

**Remark:** The W-Interface is a power electronic interface. Trackside assets are connected directly by wiring. Communication is done analogously and without any high-level protocol.

Connectivity and communication is manly provided by components of the Transfersystem (TS) framework. For serving the different parties, different kind of data contracts have to be implemented. Details can be found in Subconcept Transfer System.

### 4.3.1 A - Interface vs Controller Interface

In current OC concepts, certain reference points are defined for describing physical interfaces. Concerning data exchange with EI, the following setup was described:

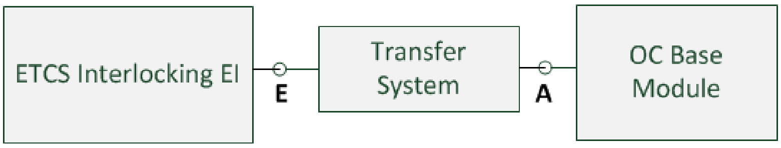

*Figure 5 Transfersystem and Reference Points*

**The Transfersystem provides logical, not physical interfaces**. In other words, the Transfersystem is the instance that implements the data transfer between EI and OC.

In smartrail 4.0 system architecture the corresponding EI - OC interface is called Controller Interface.

## 5 Elements of the OC

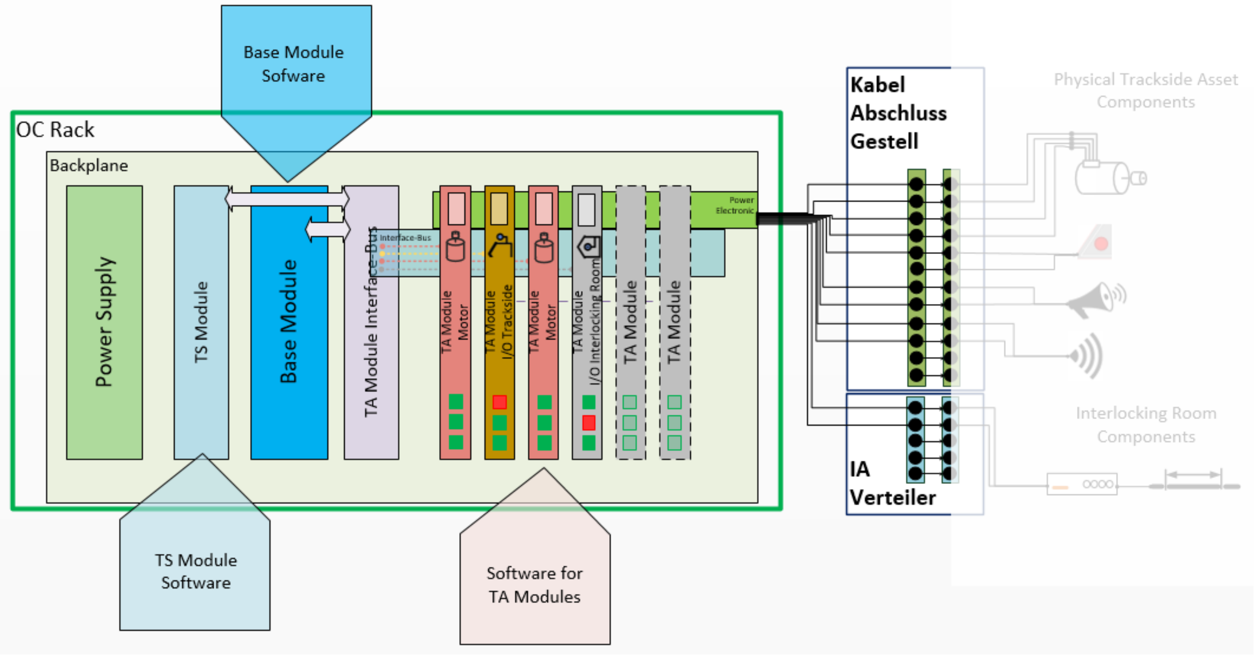

*Figure 6 Hard- and Software Elements of the OC*

The OC is a rack mounted device. It consits of several hardware parts, most of them equipped with microcontrollers and Software. Physical base element forms the backplane which is mounted into a 19" rack. The hardware components, especially the TA Modules, are plugged into the slots of the backplane.

Core Hardware component is the Base Module which forms the cetral controlling unit of the OC. Communication to pluggable Sub-Modules takes place via Module Interface Bus.

The components Base Module, TA Module and TS Module are intelligent in the sense that they are all equipped with microcontrollers and software.

### 5.1 OC Functional Blocks

The OC consist of certain hardware modules every one equipped with various software modules as depicted below.

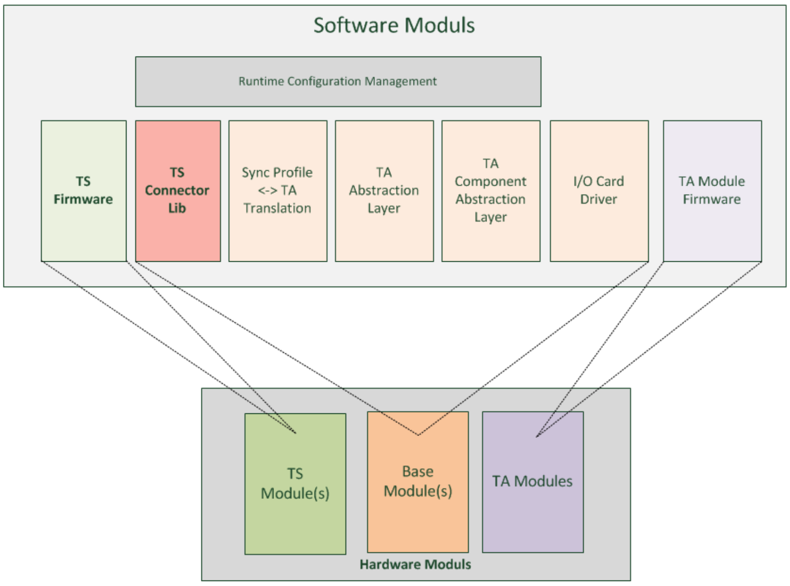

*Figure 7 Hardware Software Modules and their relationship*

## 6 Hardware Design

### 6.1 General Aspects

The OC shall be modular. It's built up from a combination of components, some of them available as Commercial off-the-shelf (COTS) and other which are special developments for the OC product.

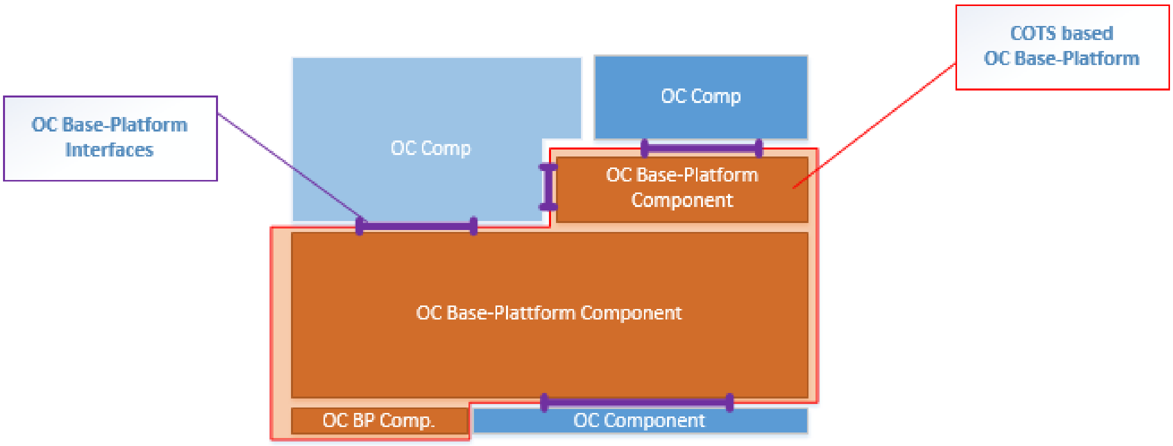
COTS based OC Base-Platform
OC Base-Platform Interfaces
OC Comp
OC Comp
OC Base-Platform Component
OC Base-Plattform Component
OC BP Comp.
OC Component

*Figure 8 OC as a combination of COTS and specific products*

The OC is a safety relevant component and thus must support Safety Integrity Level 4 (SIL-4). This requirement has a big impact on the (safety) achchitecture of the OC Hard- and Software.

### 6.2 Hardware Components

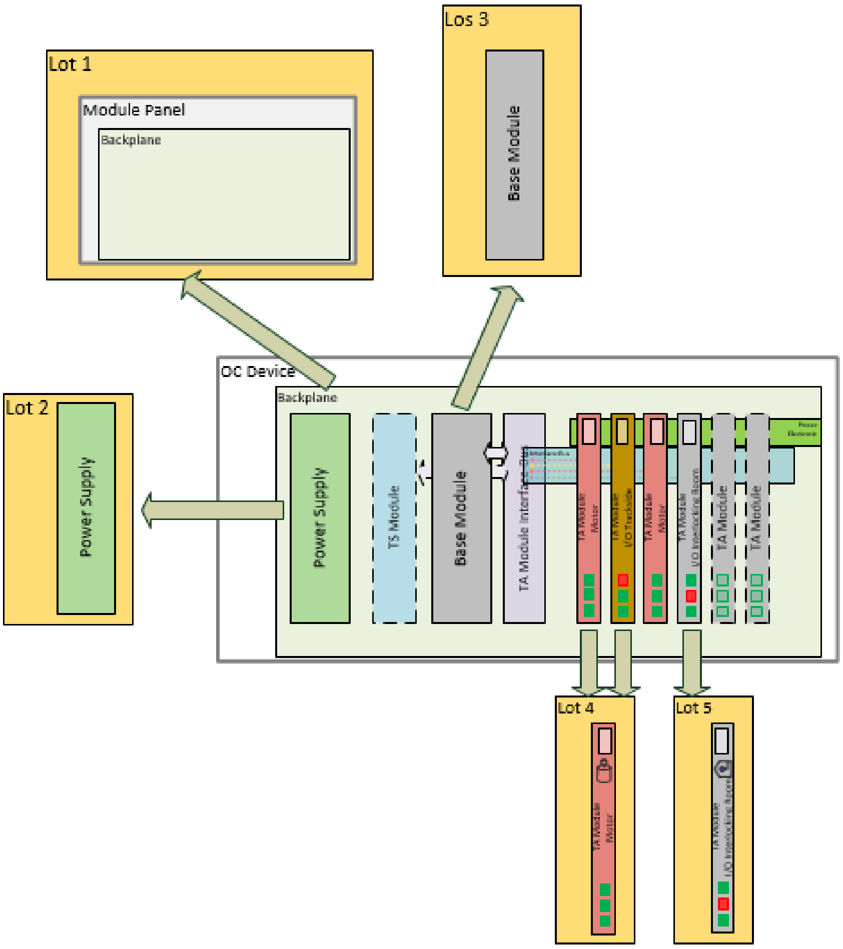

*Figure 9 Hardware Components*

#### 6.2.1 Base Module

The Base Module forms the cetral controlling unit of the OC. Main functionality is safe communication with centralized systems mainly EI and also Monitoring and Device Management Systems on one hand and with the Trackside Asset and Transfer System Modules on other.

Under consideration of the standardisation aspect mentioned in OC-11175 - General Aspects it's obvious that the Base Module shall form the base platform. Therefore it's suitable to consider a COTS component that forms the Base Module.

In general two different technologies can be used for building the Base Module.

- "Speicher Programmierte Steuerung (SPS)" based controllers
- High-level language programmable multi controller platforms

#### General Requirements

Many of the requirement concerning Base Module technology, development and approval process are valid independently of which target platform is used. These requirements are:

- SIL-4 certified platform
- Supporting RaSTA implementation according DIN VDE V 0831-200
- Programmable according IEC 61131-3

Requirements that are highly recommended

- CENELEC EN 50128 approved development environment
- Security certification according IEC 62443

### 6.2.2 SPS Controllers

Many of industrial controllers are built up with SPS logic. For evaluating a SPS based platform following requirement has to be considered in addition to the general requirements:

- FUP programmable according IEC 61131-3

See below examples of candidates that may fulfill these requirements:

| Product | Characteristic | Remark |
|---|---|---|
| HIMA<br>www.hima.com | Products: HIMATRIX and HIMAX supporting SIL4 standards in accordance with CENELEC. | |
| PILZ<br>www.pilz.com | CENELEC/TÜV certified automation system PSS 4000;<br>PAS4000 Development environment for function block based programming. | |

An important criteria is that programming of business logic of the controller can be done by use of Function Block Programming ("Funktionsplan FUP") method. Because availability of FUP programming environment is mandatory, main development tasks for programming the Base Module Software are concerned to FUP programming and therefore on a high abstraction level.

### 6.2.3 Multi-Controller Unit Platforms

This kind of platform is less standardized than SPS based platforms. This implies one hand that there's more flexibilty for designing the Software but on the other hand there exist fewer standards and prepared certifications for this kind of platforms.

Common characteristic of these platforms is that programming will set up directly on Operating System functionality.

There are mainly two kinds of platforms.

1. Hardware based solutions. This solutions contains safety certified Hardware providing the base for also safety certified Operating Systems or proprietary Hardware abstraction layers. Some SPS suppliers do not deliver their own certified Operating System and therefore compatible products have to be evaluated

and used separately, intergration work is not to underestimate!

2. Realtime Operating Systems with deep integration of safety functions. This platforms provide certificated Software functionalites based on specific compatible hardware components. Because of a wide variaty of possible Hardware and Software combinations, the integration work is even more expensive than with the hardware based MCU platforms.

# 7 Software Design

## 7.1 Configurability

### 7.1.1 Configuration Handling

The OC is a safety relevant, configurable component. Therefore an integer handling of configuration data is an elementary functionality.

The content of an OC configuration addresses two main domains.

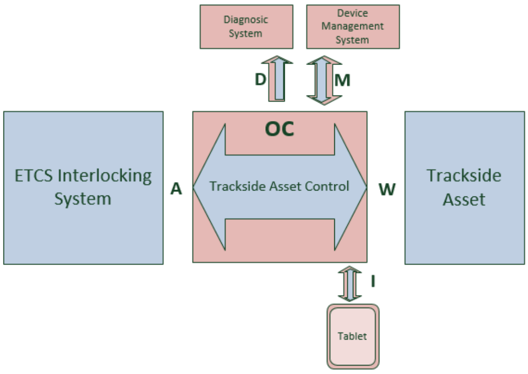

*Figure 10 Two general kinds of configuration*

One domain concerns to the OC as host for multiple Trackside Assets and as main communication endpoint for peer systems, especially the ETCS Interlocking system. It's the red part in above graphic. The basic data of this configuration part concerns the communication functionality of the OC.

The other domain is the dynamic instantiation of Logical Trackside Assets. It's the blue part in above graphic. The configuration determines the behaviour of such an instance. The Logical Trackside Asset confguration is created by a provisioning system through templates. In fact, the LTA configuration forms the real business functionality. Without a Logical TA configuration the OC is useless.

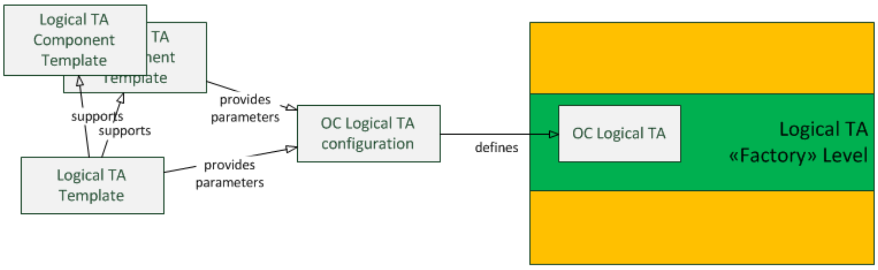

*Figure 11 Using templates for creating OC runtime configurations*

In some documents, the Logical TA configuration is called Infrastructure Object Configuration. Infrastructure Objects are the generic representation of Logical Trackisde Assets in EI. Therefore in EI documentations the term Infrastructure (IO) is used. Have a look to the explanation of corresponding term in OC Software Development Cost Estimation Chapter OC-11188 Documents and Terms

### 7.1.2 Functions and Parameters

The concept of building software based Trackside Assets and Trackside Asset components is based on the principle of using functions and parameters. The aim of separation between functions and its parameters is to simplify the approval processes.

**Example:**

Function: on / off

Parameters: Voltage level

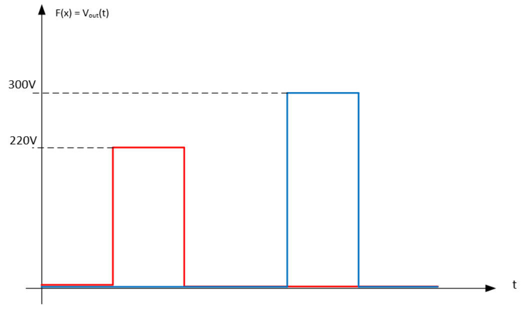

*Figure 12 Principle of using functions and parameters*

The goal is to achieve with a certain set of functions and parameters the needed compatibility to adapt a wide range of different types of Trackside Assets and Trackside Asset Components.

### 7.2 Base Module Software

As seen in OC-9704 - Elements of the OC multiple parts of the OC are equipped with Software. Because the main part of the business logic is located at Base Module Software, this part is considered in detail.

The graphic below depicts a visualisation of the functionality of the Base Module Software. The puzzles are configurable building blocks used for instantianting software based representations of Trackside Assets called Logical Trackside Assets.

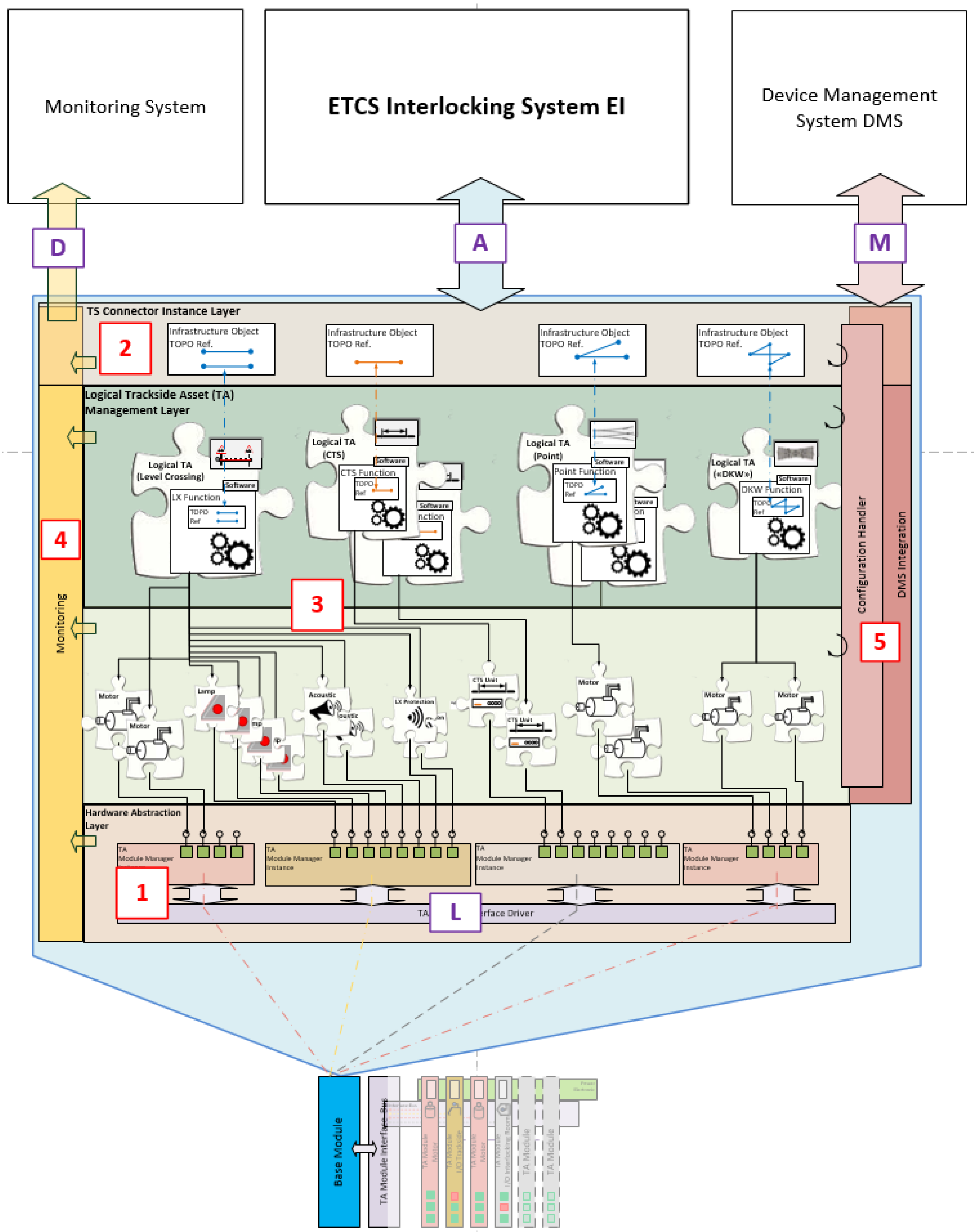
Monitoring System
ETCS Interlocking System EI
Device Management System DMS
D
A
M
TS Connector Instance Layer
2
Infrastructure Object TOPO Ref.
Infrastructure Object TOPO Ref.
Infrastructure Object TOPO Ref.
Infrastructure Object TOPO Ref.
Logical Trackside Asset (TA) Management Layer
Logical TA (Level Crossing)
Software
LX Function
TOPO Ref
Logical TA (CTS)
Software
CTS Function
TOPO Ref
Logical TA (Point)
Software
Point Function
TOPO Ref
Logical TA («DKW»)
Software
DKW Function
TOPO Ref
3
4
Monitoring
Motor
Motor
Lamp
Acoustic
LX Protection
CTS Unit
CTS Unit
Motor
Motor
Motor
Hardware Abstraction Layer
TA Module Manager Instance
TA Module Manager Instance
TA Module Manager Instance
TA Module Manager Instance
1
L
TA Interface Driver
Configuration Handler
DMS Integration
5
Base Module
TA Module Interface
TA Module Motor
TA Module I/O Trackside
TA Module Motor
TA Module I/O Interlocking Room
TA Module
TA Module

*Figure 13 Overview of OC Base Module Software functionality*

**Main Parts:**

[1] TA Module Hardware Abstraction

[2] Transfersystem Connector

[3] Software based forming of Trackside Assets and Trackside Asset Components

[4] Providing interability into Monotoring system

[5] Providing integration into Device Management System

In chapter OC-11152 - Software Functionality the different parts are described in detail.

**Main Interfaces:**

[D] Interface to monitoring and diagnostic system

[A] Controller Interface. Interface to ETCS Interlocking. Main interface for Trackside Asset functionality.

[M] Interface to Device Management System

[L] On-Site Bus Interface. Interface to TA- and TS Module Cards

### 7.3 Software Functionality

#### 7.3.1 Booting and running OC

##### 7.3.1.1 Run Level Handling

> The subconcept Subconcept Modes of Operation and Configuration describes two general runmodes. The main goal was to isolate levels with different safety requirements.
>
> This concept has to be optimized in a way that the OC shall run as often as possible in productive mode and a maintenance state shall only be applied to single instances of Logical Trackside Assets. Specifications described in this document are based on the new, optimized handling with Logical Trackside Asset as entity for supporting different modes.
>
> However, for installation and maintenance purpose during rollout phase, the OC may run in a general maintenance mode with lower requirements to the safety functionalities like watch dogs and other supervision tasks.

#### 7.3.2 Device Supervision and Logging

As the OC is a safety critical component that must support SIL 4, supervision of runtime behavior as well as detailed diagnostic and logging functionalities have to be realized.

### 7.3.3 Initialisation of Transfer System

The Transfersystem is a framework that encapsulates functionalities for safe and secure data exchange. The interface for using this framework is formed by the TS Connector component. This is a standardized implementation of data exchange functionalities which will run in same context as the application that uses it.

**TS Connector Instance Layer**

The TS Connector Instance Layer implements the communication functionality towards peer systems. One instance is responsible for communication of Logical TA abstraction with EI. According to the Transfersystem concepts the instance is an Transfersystem Connector instance. Because instantiating of Logical TAs can be done dynamically, a registration process is used for binding Logical TA functionality with TS Connector services.

| | |
|---|---|
| Status | draft |
| Linked Work Items | has parent: OC-11184 - Initialisation of Transfer System |

**Initializing communication infrastructure and establishing connectivity**

Initially, the OC have to find the Service Lookup Directory of the Tansfersystem. This is ensured by applying initial network configuration data which are part of the OC Base Module configuration.

With help of Service Lookup Directory the communication endpoints of the ETCS Interlocking, Diagnostic and Device Management Systems are determined. As parameters for finding right endpoints are the values data exchange Contract ID and QoS Parameters used.

The final TS channel configuration which is used for establishing direct communication between the peer hosts is delivered also by Service Lookup Directory.

After establishing all configured communication channels, the communication infrastrucutre of the OC is ready for being used by the sub services like Logical Trackside Asset instances.

For further details concerning Transfer System see Subconcept Transfer System .

### 7.3.4 Connecting Monitoring System

The whole smartrail 4.0 overall system will be monitored by a centralized monitoring system. While starting of OC, it will connect to this system and send runtime information about the OC and its elements.

**Very important requirement: Collecting of informations for diagnostic purposes must be done without any impact (Rückwirkungsfrei) to the safety relevant functionality.**

For having human readable monitoring information like an informal name or a type descirption of a point available, configuration of Logical TA is extended with data elements that are not used in a functional manner but are used for diagnostic purposes.

Every layer of the Base Module Software stack must support diagnostic functionality and broadcast important actions.

### 7.3.5 Connecting Device Management System

The DMS provides the OC mainly with Software and configuration data. During installation phase, initial Software and configuration is loaded. Later-on, updates can be deployed over this path. During boot-up process the OC will use factory preset configuration values for connecting the centralized Device Management System. If connection is established, checks for updating firmware and runtime configuration are done. If new versions are available, they will be loaded and applied.

### 7.3.6 Runtime Configuration Handler

The OC will have to fulfill very high availability requirements. This means that the uptime must be maximal. On the other hand, Trackside Assets will be maintained and modified. Changing of parameters or functionality of a Trackside Asset will result in a configuration change.

Therefore the Runtime Configuration Handler is a very important Software component. It must ensure that Logical Trackside Asset instances can be re-instantiated at runtime!

Every Logical Trackside Asset contains several states, at least a safety state and an operating state. Both of them are used by EI for evaluation whether a TA can be controlled in safe manner or not.

This state is used if a re-configuration takes place. The operation state of affected LTA is set to "remote managed".

### 7.3.7 On-Site Management via direct Link

In case that connection establishing via network is not possible, the OC must be configurable and maintainable via link cable that is connected to a Laptop. The whole functionality that is available via DMS must also be available by use of direct link.

### 7.3.8 TA Module Management Layer

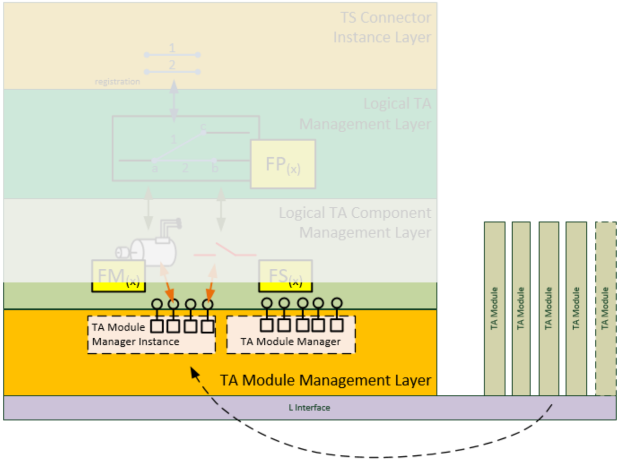

*Figure 14 TA Module Management Layer*

The TA Module Management Layer forms the Hardware abstraction layer for the OC business application. Logical TA Module Ports are Software based representations of electrotechnical functonality provided by the TA modules.

**TA Module Management Layer**

The TA Module Manager Layer acts as provider service for the Logical TA layer. It encapsulates the communication with the TA Modules. Different TA Modules provides port functionalities that can be used by Software Trackside Asset Components.

| | |
|---|---|
| Status | draft |
| Linked Work Items | has parent: OC-11163 - TA Module Management Layer |

**Transparent handling of TA Modules**

For the Logical TA Layer, the functionality of TA Module Manager is transparent. The relevant part for the interface is formed by logical ports of TA Components "Logical TA Component Port". These Ports are a logical representations of either a physical port or a certain encapsulated TA functionality provided by a TA Module.

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-11163 - TA Module Management Layer |

The Logical TA ports usable for the Logical TA component Layer are created by TA Module Handler instance depending on what kind of TA Module is recognized.

Main functionality of TA Module Management Layer is:

- Monitoring backplane slots about TA Module presence
- Recognition of TA Module presence and TA Module Type
- Forming of Logical TA Module Ports depending on TA Module Type and optionally additional parameters read-out from TA Module.

### 7.3.8.1 Communication with TA Modules

Communication with TA Modules have to be done by a RaSTA protocol implementation. The protocol will be a simple communication protocol for asynchronous use of Logical TA Module Port functions.

### 7.3.9 Logical TA Component Management Layer

Logical TA compponents are Software abstractions of physical TA components like motors, sensors, flashing lights etc. The goal of defining such elements is to be able to build up Logical TAs by use of standardized modular components.

Logical TA Components use the Logical TA Module Ports for controlling the electronic ports that are wired with the physical component.

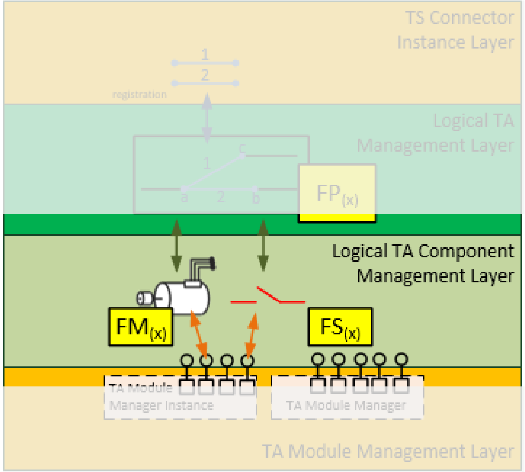

*Figure 15 Logical TA Component Management Layer*

### 7.3.10 Logical Trackside Asset Management Layer

#### Logical TA Management Layer

The Logical TA Management Layer is the level where the Software based abstractions of Trackside Assets instantiated and managed.

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-11165 - Logical Trackside Asset Management Layer |

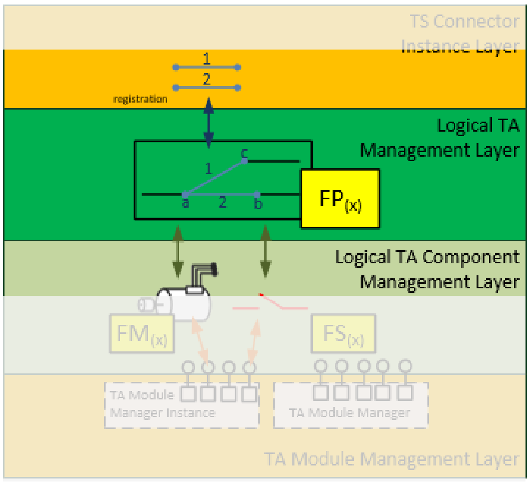

*Figure 16 Logical Trackside Asset Management Layer*

#### Dynamic management of Logical Trackside Asset instances

The OC must be able to manage life-cycle of Logical Trackside Assets at run-time and in safe manner. This means that creation, activation, deactivation and change of Logical Trackside Assets have to be done while OC is up and running and without disturbing functionality of other .

| | |
|---|---|
| Status | awaiting approvals |

| | |
|---|---|
| Linked Work Items | has parent: OC-11165 - Logical Trackside Asset Management Layer |

**Content of Logical TA configuration**

The configuration that is used by Logical TA Management Layer for instantiating Logical TAs is created by a centralized provissioning system. The Logical Trackside Asset configuration is created by use of templates. It contains mainly following information:

- the combination of components (assembly information)
- the parameter set that have applied for every component
- the Topo representation and reference
- the parameter set concerning the Logical Trackside Asset as a whole
- meta data for data integrity (hashes) and version info for release managemen

| | |
|---|---|
| Status | draft |
| Linked Work Items | has parent: OC-11165 - Logical Trackside Asset Management Layer |

In fact, the configuration does only parametrize functionalities that are hardcoded and present as Software library function in the OC Base Module. Therefore, every Logical TA Component / Logical TA has a type information and a version. These informations are used for assigning configuration values to the correct Software library function.

### 7.3.10.1 Topo References of Logical Trackside Assets

Every Logical Trackside Asset containts configured references to the (safe) topology TOPO. These elements form the base for the communication between OC and EI in an abstracted manner. Trafficability of points, protection of level crossing and so on is established by EI by using capabilites of TOPO references assigned to a logical TA.

E.g. a point contains two TOPO references which abstract the trafficability over the two legs. In the EI world, these elements are called Infrastructure Object Elements.

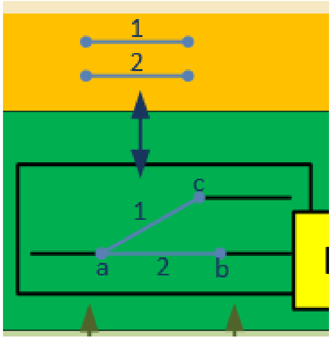

*Figure 17 Navigability elements of a point*

A level crossing contains as many TOPO references (protection sections) as the number of tracks which are protected by it.

### 7.3.10.2 Example: Sequence for switching a point

The example of switching a point shows exemplary how commands on different abstraction levels could look like.

Topology of the point:

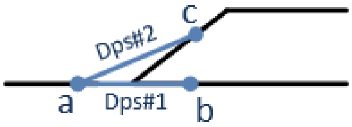

*Figure 18 TOPO reference of a point*

Sequences to switch point into diverging position and back.

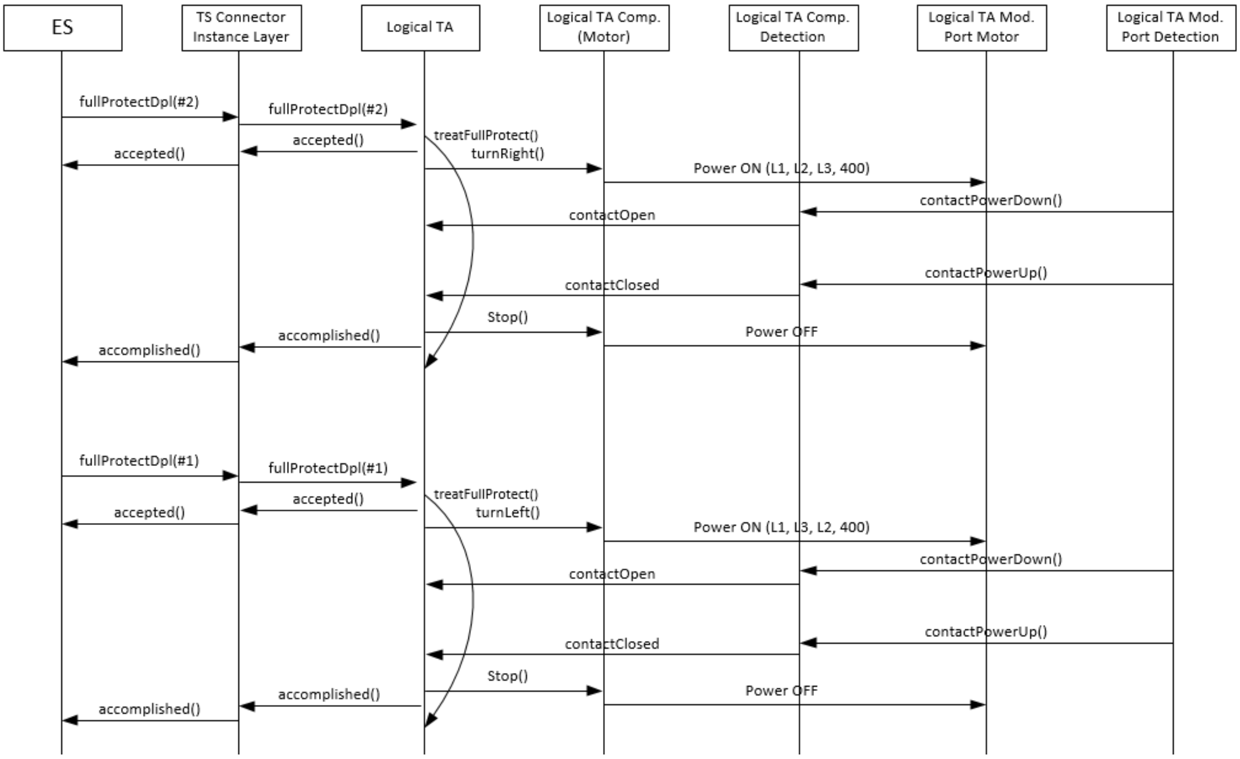

*Figure 19 Exemplary sequence for switching a point*

### 7.3.11 Binding TA Modules <--> Logical TA

**Binding of physical and logical world**

As described in previous chapters, there are two different parts that have to be combined.

The physical world is given by the presence of TA modules. Depending on the equipment, different electrical functionalities are possible. These possibilities are encapsulated and provided to the "Logical TA Component Management Layer" in form of Logical TA Module Ports.

The Logical TA Module Port functionalities are consumed by Software TA Components used for building Logical Trackside Assets.

For guaranteeing the compatibility between Hardware and Software parts, Logical TA Module Ports must be typified and must support version handling. The type declares the kind of port. The version is used for managing different releases of a certain type.

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10161 - Binding TA Modules <--> Logical TA |

### Port assignment as safety critical issue

To ensure safe functionality the assignment of virtual ports to Software Trackside Asset Components is a very important functionality. Various variants for doing binding and binding verification exist. The most apparent has to be determined.

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10161 - Binding TA Modules <--> Logical TA |

### Fix addressable numbered Slot- and Port Positions

While doing planning in AMP relative Slot- and Port Positions are used. TA Module slots in every backplane are numbered sequentially. Also every Port on any TA Module carries a number. All work with physical goods like plugging cards and cabling during installation in interlocking room is done by using these numbers as reference.

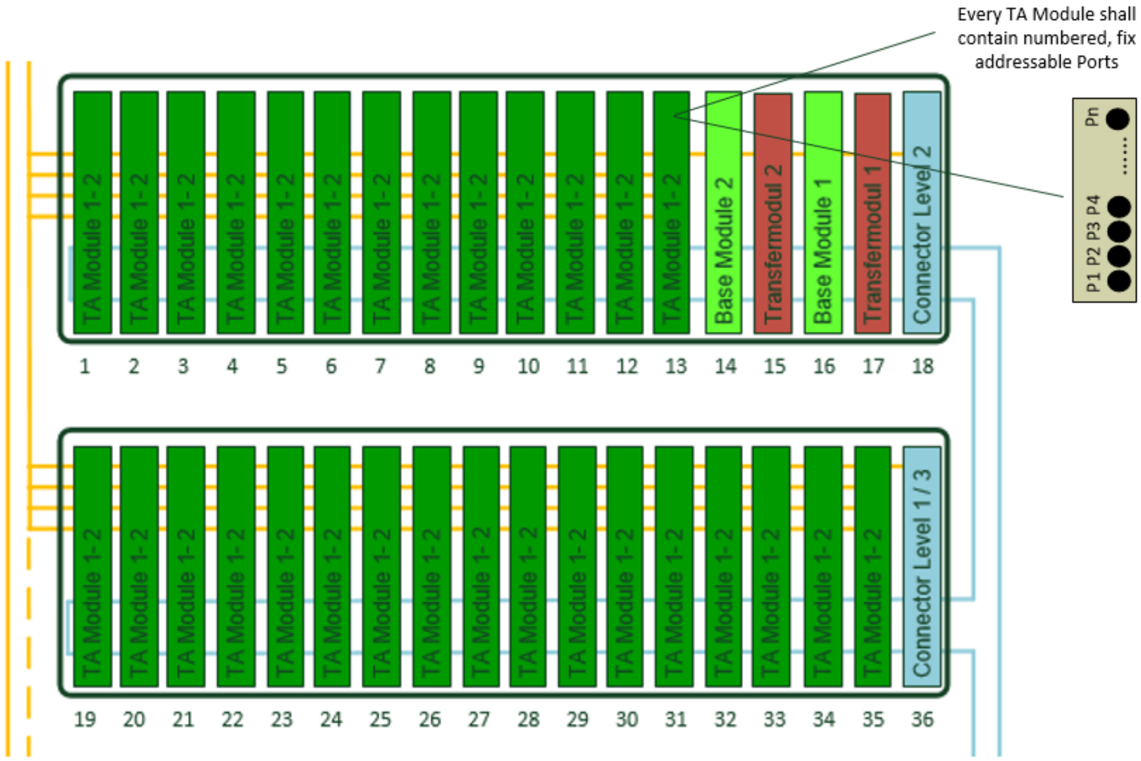

*Figure 20 Numbered and fix addressed TA Module slots and ports*

Slots for different TA Modules are physically identical for every TA Module type. Therefore while planning installation, backplanes can become filled up according to the needed number of physical connectors. If e.g. 26 points are installed and every TA Module 1-2 type supports two point machines, 13 (Pos 1 - 13) TA Modules must be plugged. Further slots can be used for TA Modules of other type for example for connecting level crossings.

Every TA Module type contains a certain number of Ports. While doing wiring, the combination of Slot- and Port Position is used for defining cabling between backplane connector and pin on Cable Termination Frame (CTF).
Example: Ports 1 to 4 of generic I/O TA Module from slot 20 are named 20.1, 20.2, 20.3 and 20.4;
Using this schema, the whole plant can be designed at any time in advance of installation time.

| | |
|---|---|
| Status | draft |
| Linked Work Items | has parent: OC-10161 - Binding TA Modules <--> Logical TA |

## 8 Appendix A: Templates

### Templates as building blocks for creation of Logical Trackside Asset configurations

Configurations for Software TA Components and Logical Elements are created with help of templates. Templates contain the parameters and rules needed for creating software based instances. Configuration is used at OC provisioning system for creating Infrastructure Object configuration.

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10152 - Appendix A: Templates |

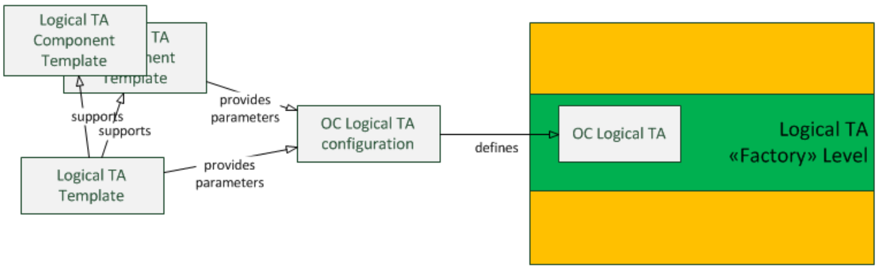

*Figure 21 Use of templates*

### Implementation of component functionality

Another important aspect is the use of Software libraries. Control characteristics of TA components are programmed. Various libraries provides functionalities for different kind of components, mainly motors, lights, and acoustic signals. Parameters are used for adapting functionality to the different electrical requirements of different devices.

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10152 - Appendix A: Templates |

### Main parts of Logical Trackside Asset

A Logical Trackside Asset contains following main parts:

Assembly information:
According to the Type of TA and therefore type of Logical TA, one to multiple logical components are combined for providing TA functionality as a whole. Depending on the hardware that will be connected, a valid set of parameters is applied to every used component.

Topo Reference:
An abstracted representation of Trackside Asset which ensures the use of Topo related interlocking functionality on EI. The very generic kind of communication with EI is done by providing an item that has on one hand a link to the railway topology and provides on other hand capabilities for calling actions. Such items contain also states which represent the current physical state of the corresponding Trackside Asset. Many actions provided by capabilities result in changes of states. These items represent Trackside Assets or parts of them and can be either a vector, an edge, a section or another Topo related item that contains capabilities implemented by the OC and that are usable by EI.

TA Parameters:
Every Trackside Asset contains properties and parameters related to the TA as a whole. The closing time for a level crossing for example depends on the combination of components and its physical nature and temporal behavior. Because these parameters are dependent on the combination of components they are technically assigned to the Compatibility Set of a Logical TA.

Compatibility Set
The Compatibility Set is a data structure that contains all compatible compositions of Software TA Components and Parameters that can be used for instantianting Logical Elements. For approval process, every set will reqire a generic approval.

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10152 - Appendix A: Templates |

### Logical Element Template

A Logical TA Template provides possible values for every main part of a Logical TA. It provides a superset of every approved combination of components and component parameters for a certain TA type.

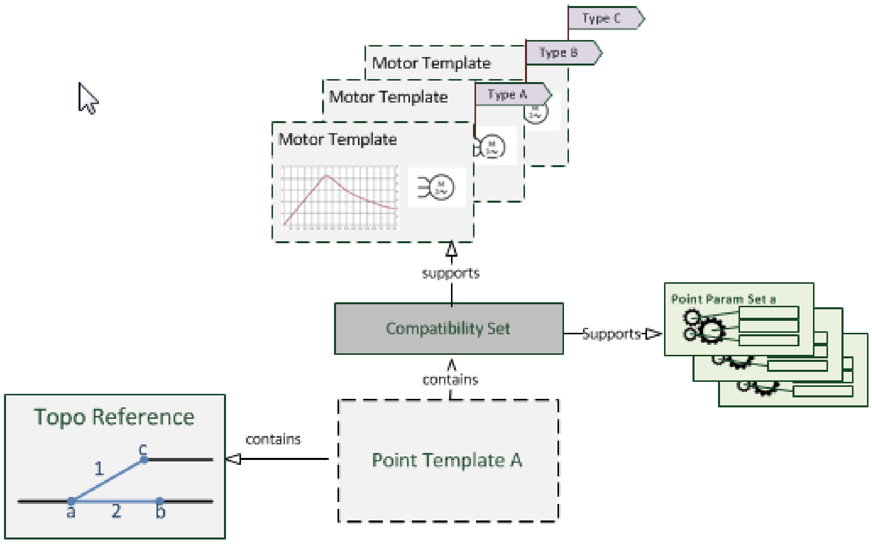

*Figure 22 Logical Element Template*

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10152 - Appendix A: Templates |

### Software TA Component Template

While creating Logical Element instances, required component instances are instantiated by use of Software TA Component templates. SWTAC templates contains references to Software libraries providing the functionality for the components. They further specifies valid sets of parameters that can be applied to these functionalities.

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10152 - Appendix A: Templates |

### Use of Virtual Ports

Additionally to the parameterisation of the components power and sensor electronic, the required type of virtual port is defined. Thus the virtual port acts as binding element between the logical and the representation of the physical world. The type of the virtual port on TA Module Management Layer is given by the physic of the plugged TA Module.

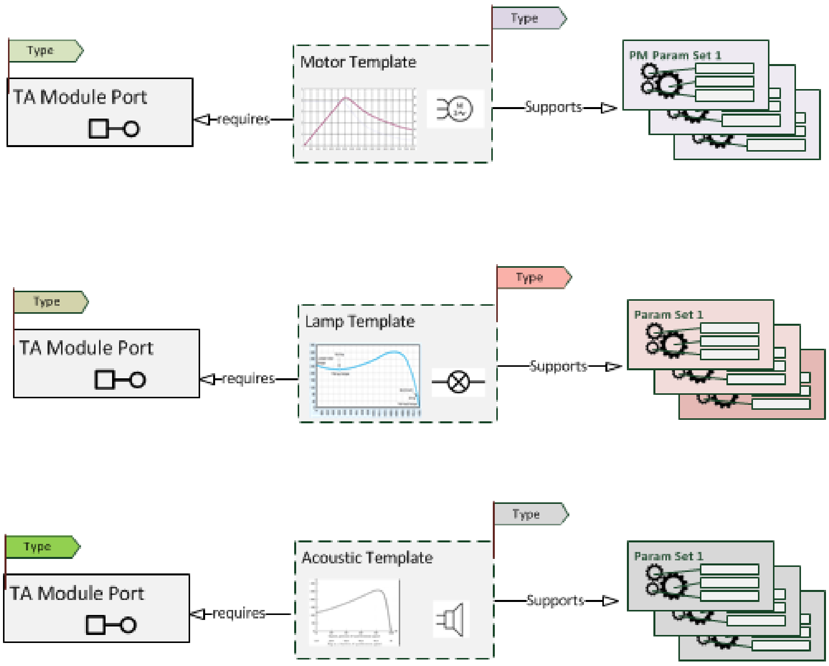

*Figure 23 Component templates*

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10152 - Appendix A: Templates |

**Template catalogue for Provisioning Systems**

Templates are used at stage where configuration is defined. Templates defines all possible, valid an approved combinations of components and parameter sets. The available set of templates forms a kind of catalogue for building all possible variants of software based representations of physical Trackside Assets.

The template catalogue can be used by a provisioning system for creating integer configurations. By help of templates, the compatibility between functionality provided by code libraries of OC base module and the approved parameters is ensured.

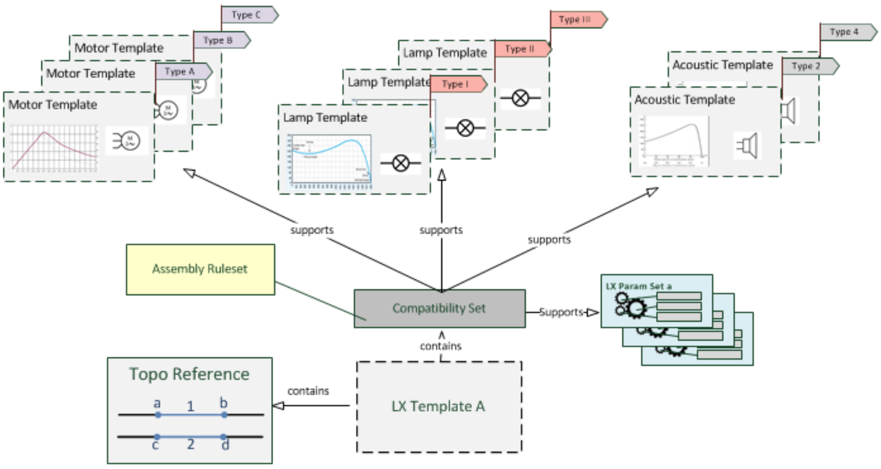

*Figure 24 Template catalogue*

| | |
|---|---|
| Status | awaiting approvals |
| Linked Work Items | has parent: OC-10152 - Appendix A: Templates |