TCCS SD1 - Data Model_00_Guide

1 Table of Contents

2 Terms

SPT2TS-2040 - The CCS/TMS Data Model defines the harmonised language to generate and transport the Domain Data at System Pillar interfaces. The Transversal CCS Subdomain 1 (SD1) is responsible for the specification of the CCS/TMS Data Model in collaboration with

- the System Pillar domains which apply the defined data structures in interface specifications
- the Innovation Pillar which proves the applicability of the data model by demonstrators.

[Content to be approved **]**

Engineering Data

The Engineering Data as part of the SPT2TS-2040 - CCS/TMS Data Model contains all the base data (i.e., track

topology, track geometry, track asset configuration) for compiling the next version(s) of use case-specific Domain Data. The standardized Engineering Data comprehensively covers the data needs for the SERA in the scope of the System Pillar (i.e. radio-based ETCS only). Specific data needed for migration is out of the scope of SPT2TS-2040 - CCS/TMS Data Model.

Domain Data

Domain Data as part of the SPT2TS-2040 - CCS/TMS Data Model is a use-case-specific representation of information following the specific needs of affected systems and their functionalities implementing the use case. Domain data are a tailored and potentially transformed subset of Engineering Data.

3 Motivation for CCS/TMS Data Model

SPT2TS-2037 - A harmonised digital CCS/TMS system requires a shared data language applied at all relevant interfaces with similar exchange items. Furthermore, it requires comprehensive engineering data for planning and building assets according to the System Pillar architecture (i.e. radio-based ETCS only). With the SPT2TS-2040 - CCS/TMS Data Model
the System Pillar provides a data structure for standardised engineering and to align a data structure for standardised interface specifications within CCS including CCS-TMS. This data structure shall be suitable across all relevant use cases of the System Pillar such as engineering and communication for traffic control or automated train operation.

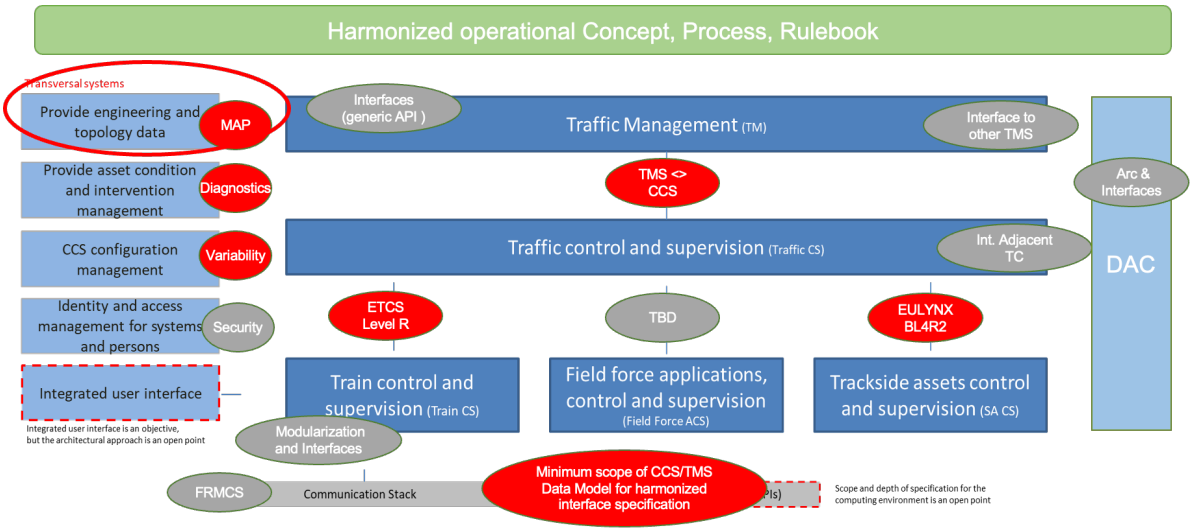


Figure 1 Interfaces between domains and minimum scope of CCS/TMS data model

The Data Model and interface specification must be comprehensive for SERA and therefore sufficiently detailed (not on a high conceptual level), at least on the Functional Interface Specifications (FIS) level within the System Pillar architecture. The development must be according to SEMP. The modelling approach is selected to support these goals.

While the TCCS domain is located in Task 2 of the System Pillar, it offers the inclusion of other tasks with the same and consistently applied CCS/TMS Data Model, as it is already applied for the interface between Task 2 and Task 3 TM.

Besides the static aspects of a data model, the dynamic aspects related to the life cycle of the data are

considered by TCCS. This includes the specification of an end-to-end process for data preparation and provisioning (configuration), starting from the required output from the engineering process (SPT2TS-2030 - Engineering Data) up to the compiling of use case-related data (SPT2TS-2031 - Domain Data) for distribution and configuration (e.g. functional use case of Maintenance/Diagnostics or ATO).

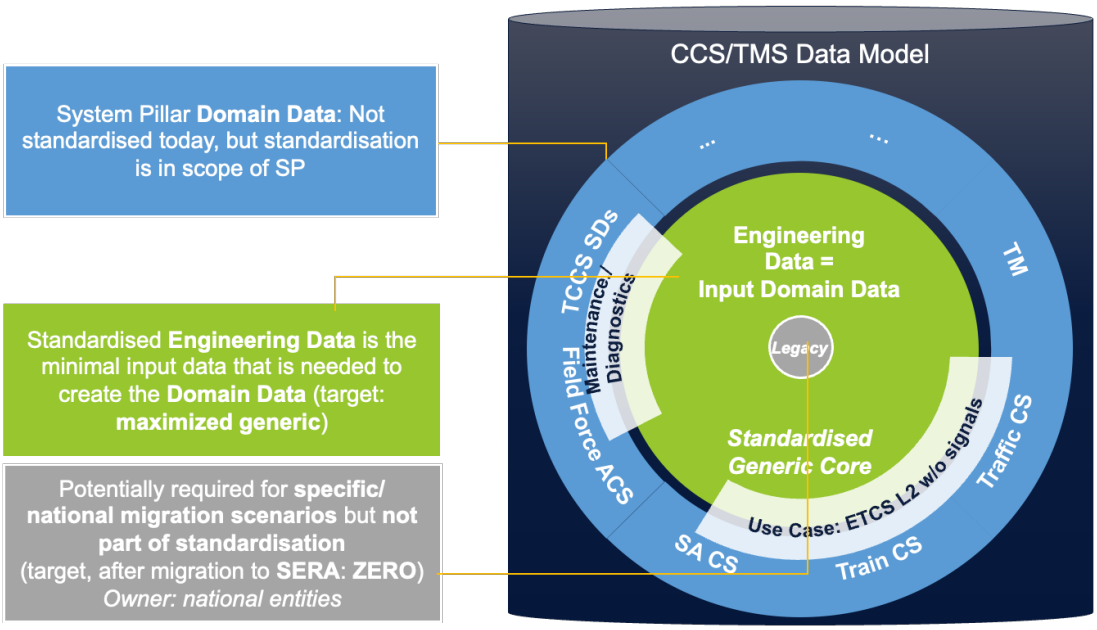


Figure 2 : Scope of harmonised CCS/TMS Data Model for the end-to-end process from preparation/distribution to application of data

[Content to be approved **]**

For more details, see CCS/TMS Data Model - Scope and Approach for Collaboration and Specification.

4 Development History and Next Steps

SPT2TS-122465 - CCS/TMS Data Model Revisions

In the following table, the revision history of the SPT2TS-2040 - CCS/TMS Data Model developments is documented, containing details about the release notes and the review activities. The feedback for one release leads to improvements within the next iteration of the data model and its documentation. Also, schemas are added or extended to cover more use cases of ERJU System Pillar. Besides the history, the planned releases are listed to give an overview of the next steps.

Data Model Version	Release Date	Release Notes	Publish and Review

Data Model Version	Release Date	Release Notes	Publish and Review
0.1	31.03.2023	By the following documentation, we would like to inform you about the defined scope and mode of work of SP TCCS SD1, i.e. regarding the derived SPT2TS-2040 - CCS/TMS Data Model and the intended collaboration with the other domains/FAs: TCCS SD1 - Scope and Approach for Collaboration and Specification Attachments: TCCS SD1 - Data Model TCCS SD1 - Data Model Schema We kindly ask you to take note and provide comments in Polarion (or exported word file). Some additional contexts regarding the SPT2TS-2040 - CCS/TMS Data Model: The data model initially focuses on the first layers (topology, geometry) and shall demonstrate the SD1 approach in a practical way. This paves the way and defines a base for further extensions in collaboration with you. Besides just providing clarification about the scope and approach of SD1, we also encourage (i.e. the FAs) to give feedback based on their implementation experiences.	The release (Polarion, PDF, JSON) has been shared via email and presentation with: • All System Pillar Domain Leads for further distribution to responsible domain members • Innovation Pillar FA1 (incl. CDM) • Innovation Pillar FA2 (WP27) The documentation was used for general scope & approach alignment rather than detailed model content discussion. The received comments (written form, discussions) are used to improve the explanations and structure of documents.

Data Model Version	Release Date	Release Notes	Publish and Review
0.2	05.07.2023	Version 0.2 of the SPT2TS-2040 - CCS/TMS Data Model has been updated with incoming needs from first use cases like ETCS, ATO, and Train Protection. In addition, the document structure has been modified to provide data model separation between different domains with sufficient linking between the domains. The SPT2TS-2040 - CCS/TMS Data Model version 0.2 contains the following set of documents: TCCS SD1 - Scope and Approach for Collaboration and Specification (for additional information) TCCS SD1 - Data Model_00_Release Notes (this document) TCCS SD1 - Data Model_01_Introduction TCCS SD1 - Data Model_02_Schema TCCS SD1 - Data Model_10_INFRA TCCS SD1 - Data Model_11_ATO TCCS SD1 - Data Model_11_ENG TCCS SD1 - Data Model_11_TP We ask you to take note and provide comments in Polarion (or exported word file).	The release (Polarion, PDF, JSON, XML) has been shared with: • All System Pillar Domain Leads for further distribution to responsible domain members • Innovation Pillar FA1 (incl. CDM) • Innovation Pillar FA2 (i.e. WP27, WP13, WP44) Amongst others, the release with extended model content was used for the following collaborations: • Discussion and model improvement together with FA2 WP13 and WP27 experts regarding data model needs for Moving Block demonstrator (TP, INFRA schema) • Discussion of needs regarding the onboard system (e.g. localisation) data needs with FA2 WP27 experts (INFRA Schema) • Discussion with engineering and tool development experts regarding data and model needs, i.e. for ETCS, Interlocking, and Infrastructure; including practical feedback from tool development supporting the data model for import/export. All feedback comments are used for model improvement (ENG/INFRA schema) to ensure applicability for implementation. • Check of ATO schema against actual ATO configuration data (i.e. Segment Profiles) All relevant data model schemata have been commented on within Polarion (or by PDF comments).

Data Model Version	Release Date	Release Notes	Publish and Review
0.3	31.08.23	Correction release based on all comments of previous release. In addition, more clarification is given in the TCCS SD1 - Data Model_01_Introduction document regarding the structure and linking of data model schemas, identification and referencing (between data model parts and between instances). The SPT2TS-2040 - CCS/TMS Data Model version 0.3 contains the following set of documents: CCS/TMS Data Model - Scope and Approach for Collaboration and Specification: updated i.e. regarding top-down / bottom-up development process TCCS SD1 - Data Model_00_Release Notes: this document TCCS SD1 - Data Model_01_Introduction: more details regarding modelling language, structure, linking, identification TCCS SD1 - Data Model_02_Schema : error corrections TCCS SD1 - Data Model_10_INFRA : processed feedback from model usage and review TCCS SD1 - Data Model_11_ATO: processed feedback from model usage and review) TCCS SD1 - Data Model_11_ENG: processed feedback from model usage and review TCCS SD1 - Data Model_11_TP: processed feedback from model review In addition to the documentation, the data model for implementation in the formats JSON, XML and the JSON Schema will be published. The following use cases are in the focus of this release regarding test case scenarios (e.g. in collaboration with Innovation Pillar) - ETCS/Balise Engineering - ATO Engineering	The release was used for SP Steering Group information in Sep 23, which led to an official sharing of the data model to Innovation Pillar for specification and demonstration purposes. In addition, the release was shared and discussed with ERA to clarify the connection to ERA vocabulary.

Data Model Version	Release Date	Release Notes	Publish and Review
0.4	22.12.2023	The SPT2TS-2040 - CCS/TMS Data Model is now extended with Data Model for CCS - TMS Interface (SCI-OP) and semantic linking for ERA. In addition, corrections based on all comments of the previous release were performed. The SPT2TS-2040 - CCS/TMS Data Model version 0.4 contains the following set of documents: CCS/TMS Data Model - Scope and Approach for Collaboration and Specification: processed feedback from model review TCCS SD1 - Data Model_00_Release Notes:this document TCCS SD1 - Data Model_01_Introduction: processed feedback from model review TCCS SD1 - Data Model_02_Schema: processed feedback from model review TCCS SD1 - Data Model_10_INFRA: processed feedback from model usage and review; added ERA Linking TCCS SD1 - Data Model_11_ATO: processed feedback from model usage and review TCCS SD1 - Data Model_11_ENG: processed feedback from practical model usage, comparison against other engineering model, and review; added Fouling Point and other details required for engineering phase. TCCS SD1 - Data Model_11_TP: processed feedback from model review ; added Field Object Controller with first details as required by fist demonstrators TCCS SD1 - Data Model_11_OPP : new schema to cover needs of CCS-TMS interface. The current state is aligned with Concept_Interface_TMS_CCS_V1_2 and will be further developed in collaboaration with TMS (and Traffic CS) in 2024 TCCS SD1 - Data Model_11_MAP: Map data model (coordinates) has been extracted out of INFRA as a new domain. Not included: the connection to the diagnostic data model as developed by the responsible taskforce has not been included yet due to needed alignment with other model parts. The plan is to include it in the next release.	Besides the release of documents (Polarion and PDF in SP Open Share), the automatically created data model files (json, xml/xsd,..) are provided for further review and usage within specification and demonstration. Links to SP Open Share: • Data Model Files (json) • JSON Schemata • UML diagrams / UML pdf version • XSD Files In addition, based on the alignment process started with ERA, a semantic version (TTL files) of SPT2TS-2040 - CCS/TMS Data Model is provided for this release, including linking to ERA vocabulary: • CCS TMS Data Model Semantic (ttl) The model will further evolve during the further collaboration with ERA.

Data Model Version	Release Date	Release Notes	Publish and Review
0.4.1	04.04.2024	This is a minor release of the SPT2TS-2040 - CCS/TMS Data Model with an update of the data model for CCS - TMS Interface as well as further improvements based on inputs from Innovation Pillar FA2 Demonstrators. Updated documents: TCCS SD1 - Data Model_10_INFRA: integrated feedback from IP FA2 Moving Block Demonstrator TCCS SD1 - Data Model_11_ATO: minor changes to the data model TCCS SD1 - Data Model_11_ENG: Replace mileage by Km-Signs. TCCS SD1 - Data Model_11_TP: integrated feedback from IP FA2 Moving Block Demonstrator TCCS SD1 - Data Model_11_OPP : integrated feedback from Task 3 CMS/TMS and added upstream data model objects.	Besides the release of documents (Polarion and PDF in SP Open Share), the automatically created data model files (json, xml/xsd,..) are provided for further review and usage within specifications and demonstrators (IP FA1, FA2, FA3, FA5, and FA6). Links to SP Open Share for XML, UML, JSON, and PlantUML exports: results
0.4.2	28.06.2024	This is a minor release of the SPT2TS-2040 - CCS/TMS Data Model with following changes: TCCS SD1 - Data Model_11_OPP : integrated feedback from Task 3 CMS/TMS and IP FA2 Moving Block Demonstrator TCCS SD1 - Data Model_11_TP: integrated feedback from IP FA2 Moving Block Demonstrator Amended certain naming conventions in the data model to make the data model compatible for ERA integration.	Besides the release of documents (Polarion and PDF in SP Open Share), the automatically created data model files (json, xml/xsd,..) are provided for further review and usage within specifications and demonstrators (IP FA1, FA2, FA3, FA5, and FA6). In addition the ontology version for ERA integration is published. Links to SP Open Share for Ontology, XML, UML, JSON, and PlantUML exports: https://eeigertms.sharepoint.com/sites/SPOpenShare/Gedeelde%20documenten/Forms/AllItems.aspx?FolderCTID=0x0120009759543C5D980E4786FE7B845BBBAF5A&id=%2Fsites%2FSPOpenShare%2FGedeelde%20documenten%2FGeneral%2F24%2D06%2D28%20TCCS%20CCS%20TMS%20Data%20Model%20v0%2E4%2E2&viewid=7d2094bc%2D4ed7%2D4bd4%2D8055%2Dcda6a753bac3

Data Model Version	Release Date	Release Notes	Publish and Review
1.0	04.09.2024	Final release of CCS/TMS Data Model for remit period SC2.3 • Improvements based on Innovation Pillar demonstrator feedback • Alignment with domains and already defined interfaces - including feedback from domain approval • Improvements after Mirror Group Approval • added schema TCCS SD1 - Data Model_20_Equipment, imported from already approved Equipment Model from EULYNX (partial transfer based on needs, to be extended by other aspects according user needs during next remit period, i.e. for OC configuration and diagnosis)	As usual, all model files are created for this release. The files are published on ERJU GitHub and Polarion.

[Open **]**

5 Structure of CCS/TMS Data Model

SPT2TS-122470 - The CCS/TMS data model follows a compositional tree structure, wherein the root of the model is a composition of several packages that constitute the CCS/TMS data model (e.g., Infrastructure, Restrictions, Rolling Stock, etc.), as shown in the figure below. Every package can be divided into sub-packages (e.g., TopologyArea, GeometryArea, etc.) depending on its needs. The figure below depicts the CCS/TMS model structure with the current included content:

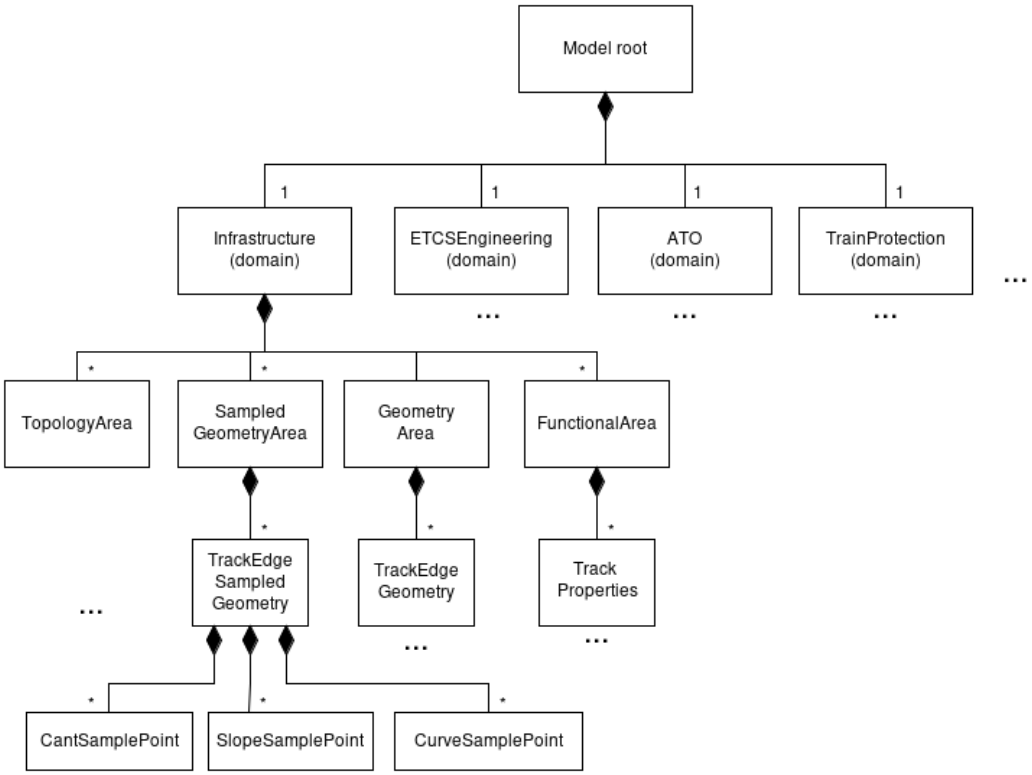


Figure 3 Overview of a compositional tree in data model.

[Content to be approved]

Rationale for compositional tree: SPT2TS-123593 - Discussion regarding selected model structure

SPT2TS-122471 - Constructing model packages should avoid cyclic dependencies, so the packages build a layer-hierarchy: upper layer-packages use (refer to) classes and enumerations specified in lower-layers-packages. The referencing is done via the point notation: `PackageId.class` or `PackageId.enumeration`. E.g. a DangerPoint in package Engineering refers to the TrackEdgePoint struct defined in package Infrastructure:

```
{
    "name": "DangerPoint",
    "attrs": [
      { "intId": 1, "name": "id",  "dataType": "string", "key": "local"},
      { "intId": 2, "name": "name", "dataType": "string" },
      { "intId": 3, "name": "trackEdgePoint", "composition": "infra.TrackEdgePoint" }
    ]
}
```
[Content to be approved]

Regarding more details and discussions it is referred to TCCS SD1 - Data Model_01_Introduction.

6 Application of the Data Model

This chapter gives guidance how to identify and reference objects as instances of the SPT2TS-2040 - CCS/TMS Data Model. After describing the general referencing/identification prinicples, the referencing between objects that carry information from different packages is explained.

Regarding the underlying discussions it is referred to TCCS SD1 - Data Model_01_Introduction

6.1 Identification & Referencing Objects

SPT2TS-122473 - Recommendation: Instead of referencing objects via data management containers (document in XML, table in SQL database, namespace in a single OPC-UA-Server) the referencing is done via the data model itself by providing the object-position inside of the objects-tree defined by composition relations:

trackEdge="/infra/topoArea[area51]/trackEdges[edge21]"

Besides an absolute path the object position can be also referred in relative way (e.g. "trackEdges[edge21]" only) if it is still unambigous due to pointing within the same area.

All the aspects in the address are parts of the data model itself and are agnostic to the data management technology:

```
{
 "prefix": "infra",
 "intId": 1,
 "structs": [
 {
  "name": "infrastructure", "moduleContainer": true,
  "attrs": [{"name": "topoAreas", "compostion": "TopoArea", "intId": 2, "multiplicity": "*", ...}]
 },
 {
  "name": "TopoArea",
  "attrs": [{"name": "trackEdges", "composition": "TrackEdge", "intId": 3, "multiplicity": "*", ...}]
 }
]}
```

Having a layered and flexible model structure requires fine-grained objects with MANY references. It is assumed, that references would consume a considerable amount of the bandwidth. To reduce them a compact reference representation is proposed, where the attribute-names are replaced by integer-IDs also specified inside of the data model in "intId"-attributes:

trackEdge="/**1**/**2**[area51]/**3**[edge21]"

The last open question is: "how to select an object from the array-attribute?"
Here we follow the XPath approach - it can be selected by one of two possibilities:

- By a zero-based-index: /1/2[#32]/3[#211]
- By the key-attribute-value: /1/2[area51]/3[edge21]. In this case the class to be selected must have one attribute marked as "key": true. Normally it is an "id"-attribute of type string. To allow separation of index-based vs. key-based referencing, the keys must not start with '#'-character.

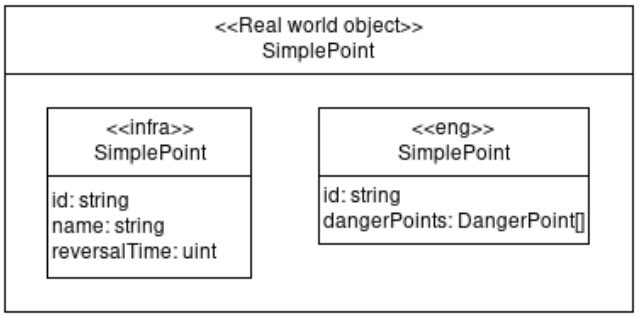
Both methods are applicable. For clear separation by the parser the '#' should only be used vor index based referenced and are forbidden in key values. In addition, the slash '/' is not allowed in references or keys, to allow speration of absolute paths from relative object positions or other ways of referencing (like GUID), which are also supported by the model, but not recommended due to the reasons mentioned above.

[Deleted **]**

6.2 Referencing between split objects

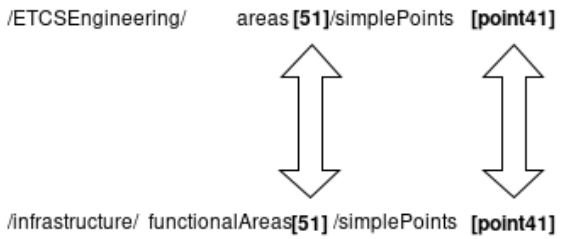
SPT2TS-122469 - The data model is structured in packages/layers covering specific aspects of the reality. In this situation one real-world-object is splitted in several data-objects representing various aspects of it, e. g.

- There is a "real-world-object" SimplePoint,
- Infra-package defines a functional data-object infra.SimplePoint covering name and transversal time important for operations,
- Engineering-package defines also a data-object eng.SimplePoint, adding danger-points to it, important for interlocking programming.



A typical Use-Case for a reading application is to combine several aspects of the physical object in memory. For this purpose it would need to retrieve several data-objects from the data storage and **link** them together. **[**Content to be approved **]**

SPT2TS-122476 - Splitted objects are linked to each other by **double-link-id-notation.** In case of the double-link-id-convention the real-world-object can be split in many parts and be constructed very effectively, as the reading application can deduce other parts from the current data-object automatically, e.g. knowing the reference to eng.SimplePoint it is easy to construct a reference to infra.SimplePoint:
/ETCSEngineering/areas[51]/simplePoints[point41] -> /infrastructure/functionalAreas[51]/simplePoints[point41]
The same is valid for the opposite direction:



This kind of dependencies are annotated in the model inside of key-attributes (here is a simplified notation from two different packages), e.g.

```
[
 {
   "name": "infra.SimplePoint",
```

```
    "attrs": [
      {"name": "id", "dataType": "string", "key": "global", "sameKeyAs": "eng.SimplePoint"}
    ]
  },
{
   "name": "eng.SimplePoint",
    "attrs": [
      {"name": "id", "dataType": "string", "key": "global", "sameKeyAs": "infra.SimplePoint"}
    ]
}
]
```


The "sameKeyAs"-dependency is transitive: a developer can follow the dependency-chain to identify all aspects of a real-world-object inside of the data-model. **[**Content to be approved **]**

7 Linking to other Data Models with Ontology

SPT2TS-2052 - The CCS/TMS Data Model is a specific Data Model which is directly used for data and interface specification. Specific data models describe far fewer data objects (ignoring all unneeded information) but are precise and optimized for specific functions. Central-specific data models try to find “the common precise data objects” of all functional input/output to avoid many conversions and redundant data creation between systems. They try to reduce the scope to the real needs and purely focus on the function.

While the CCS/TMS data model will be the base for platform-specific schemas to comply with all the restrictions (transmission/interface "on-the-wire"/"airgap", existing regulations, safety standards, System Pillar system needs,...), an integration into the ERA Vocabulary is pursued for the purpose of legislation and semantical linking to further data models (e.g. BIM/IFC). Besides reusing semantic definitions and obtaining consistent use of terms within the railway domain, this approach also offers the possibility to benefit from existing links of other data models which already are or will be connected with ERA vocabulary (e.g. railML, BIM IFC).

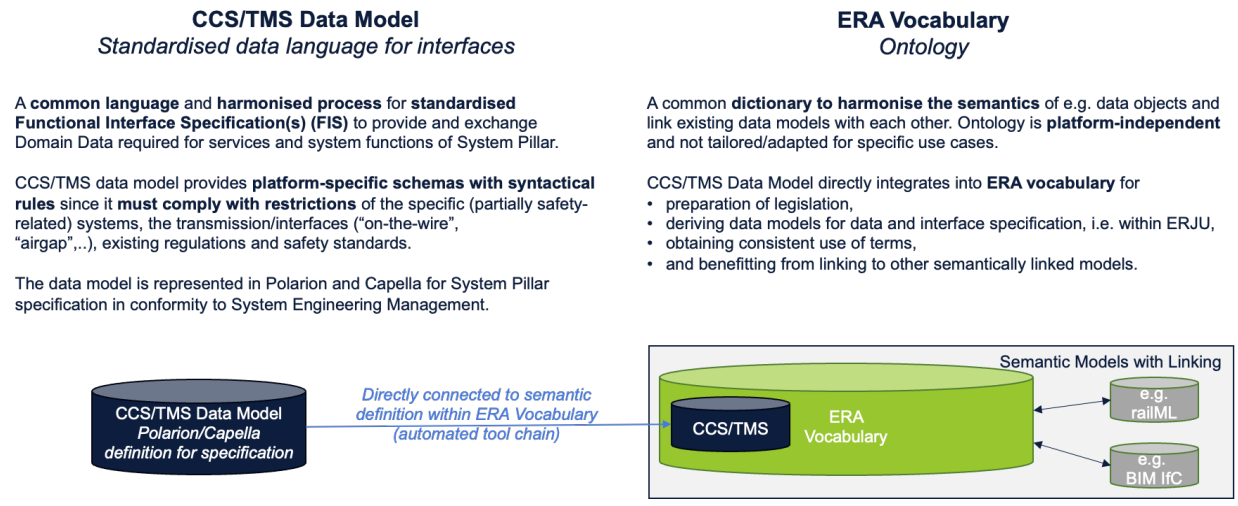


Figure 4 CCS/TMS Data Model bidirectionally linked to ERA vocabulary

In the first step of the integration into ERA vocabulary, an initial linking has been performed by identifying the

matching classes and properties/attributes of ERA vocabulary and the CCS/TMS data model and applying some changes to the CCS/TMS data model for optimal integration. The result is automatically exported into ontology language (ttl files), which are used for technical integration of CCS/TMS data model classes into ERA vocabulary. After the integration process has been finished, all further developments in the System Pillar regarding data models are reflected in ERA vocabulary. Official releases of the data model will always be derived from ERA vocabulary by an automatized process. To support the specification process, the representation of the CCS/TMS data model in the toolchain of System Pillar (Polarion and Capella) is maintained by an automated toolchain, which keeps the connection to ERA vocabulary. **[**Content to be approved **]**

7.1 Example BIM IFC

7.1.1 SD1 Linking to external BIM model

SPT2TS-122316 - One important example regarding linking to external models is the connection to BIM (IFC).

Many Ifc objects relate to railway business and are likely to *partially* overlap objects that SD1 will define. This is business as usual because Ifc focus is building objects whilst SD1 is about functional object information with CCS focus (incl. CCS/TMS interface). This said SD1 and Ifc must avoid evolving in silos. The aim is to minimise redundant overlap and avoid potentially conflicting information.
Best practice is to expose object (type) information, such that users can freely navigate and retrieve needed data, e.g. to use alignment information structured in ifc as input to Engineering or Domain Data as defined by SPT2TS-2040 - CCS/TMS Data Model. The premise for finding and accessing information defined in different models is that objects and their definitions are identified, findable and linked. This is typically done by exposing the model in a way that users can navigate to the definitions. When given an object the user can navigate to the definition. This makes accidental model overlaps unlikely and optimises the reusability of data between SD1 and Ifc.
Exemplary linkings between SD1 and ifc are:

SD1	ifc
SPT2TS-64107 - Track Edge Geometry e.g. Horizontal Segment	Ifc alignment segment with a set of IfcAlignment Parameters
SPT2TS-49087 - Applicable direction	Ifc Signal with railway signal types
SPT2TS-63866 - Track Vacancy Proving Section (TVPS)	Axle counting equipment
...	

The actual linking of structures from the SPT2TS-2040 - CCS/TMS Data Model to the IFC data model will be applied as a joined initiative with FP1 WP30 (IFC is already included in the CDM of FP1). **[**Content to be approved **]**

7.1.2 SD1 Linking to external BIM data

SPT2TS-122318 - In addition to semantic linking of data models (SPT2TS-122316), the linking between SD1 and ifc after instantiation of objects during a real project shall be defined as well.

Ifc objects all inherit from IfcRoot and by this virtue have a IfcGloballyUniqueId. The easiest solution is to equate the SD1 identity to this IfcGloballyUniqueId. However, this could create dependence during track design (alignment) and subsequent phases of the lifecycle. Uncoupling the processes and model can be achieved by defining references with keys.
The linking on the object instance level requires further elaboration and definition during the semantic linking process. **[**Content to be approved **]**

8 Methodolgy

SPT2TS-2043 - To provide a model that is unambiguous and very close to implementation, the SD1 decided to use a formal, textual specification including meta-information such as semantics and linking to source data models. Based on the experiences in former modelling or implementation projects, a JSON notation has been selected and described as schema TCCS SD1 - Data Model_02_Schema. The data model itself is explained and documented in TCCS SD1 - Data Model_01_Introduction. The definitions of this document are automatically exported into technical schema files (e.g. XML/XSD, json,..) that can be used for the first implementation, demonstration and testing purposes to support a high grade of maturity even in early development stages. **[**Content to be approved **]**

The following excerpt from TCCS SD1 - Data Model_01_Introduction gives an impression of the chosen JSON notation (Example: Topology):

SPT2TS-1697 - Formal specification "Topology":


```
{
  "structs": [
  {
    "name": "TrackEdge",
    "info": "A track edge is an uninterrupted stretch of railway track, without divergence or convergence.",
    "belongsToSubPackage": "topology",
    "see": "http://ontorail.org/rsm12/Common/Topology/LinearElementWithLength",
    "attrs": [
       {"intId": 1, "name": "id", "dataType": "string", "key": true, "info": "Identity of the track edge, needed for referencing"},
       {"intId": 2, "name": "name", "dataType": "string", "info": "User-friendly name. Empty string, if equal to the id"},
       {"intId": 3, "name": "length", "dataType": "uint32", "units": "m", "exp": -3,
         "info": "Distance along the TrackEdge's 3D-alignment. Use zero if not defined."}
    ]
  },
  {
    "name": "TrackEdgeLink",
    "info": "Defines a relation between two track edges along which a train can run.",
    "see": "http://ontorail.org/rsm12/Common/Topology/PositionedRelation",
    "belongsToSubPackage": "topology",
    "attrs": [
     { "intId": 1, "name": "id", "dataType": "string", "key": true, "info": "Identity for referencing, e.g. by points and crossings"},
     { "intId": 2, "name": "trackEdgeA", "reference": "TrackEdge", "info":"Connects to track edge A"},
     { "intId": 3, "name": "trackEdgeB", "reference": "TrackEdge", "info":"Connects to track edge B"},
     { "intId": 4, "name": "startOnA", "dataType": "boolean", "info": "True when linked to the start of track edge A, false when to the end"},
     { "intId": 5, "name": "startOnB", "dataType": "boolean", "info": "True when linked to the start of track edge B, false when to the end"}
    ]
  },
```

```
 {
   "name": "TopoArea",
   "info": "Defines a container for rail network topology",
   "belongsToSubPackage": "topology",
   "attrs": [
     {"intId": 1, "name": "id", "dataType": "string", "key": true},
     {"intId": 2, "name": "trackEdges", "composition": "TrackEdge", "multiplicity": "0..*", "sorted": true},
     {"intId": 3, "name": "trackEdgeLinks", "composition": "TrackEdgeLink", "multiplicity": "0..*", "sorted": true}
   ]
  }]
```

} **[** Deleted **]**

SPT2TS-2044 - The formal representation also allows automatic transformation into specific schemata for different applications (protobuf, xml/xsd, or further development/documentation (e.g. UML). The tools for transformations and resulting schemata will be provided as part of the next releases. **[**Content to be approved **]**

SPT2TS-1657 - The model is primarily documented, versioned and stored in Polarion. The model is exported with each release and parsed into the mentioned data schemata. These schemata allow unambiguous standardisation and fast integration into development tools or demonstrators with test data, i.e. Innovation Pillar. This practical usage of the model creates a short feedback loop so that a high level of maturity is quickly attained.

In addition, a translation into UML (e.g. plantUML, XMI) can be used for overall model visualisation, which is automatically created and always consistent with the model itself. The bridge to Capella (if available to Polarion) will be used to achieve a synchronised model view with the System Pillar architecture and all other domain stakeholders working with Capella.

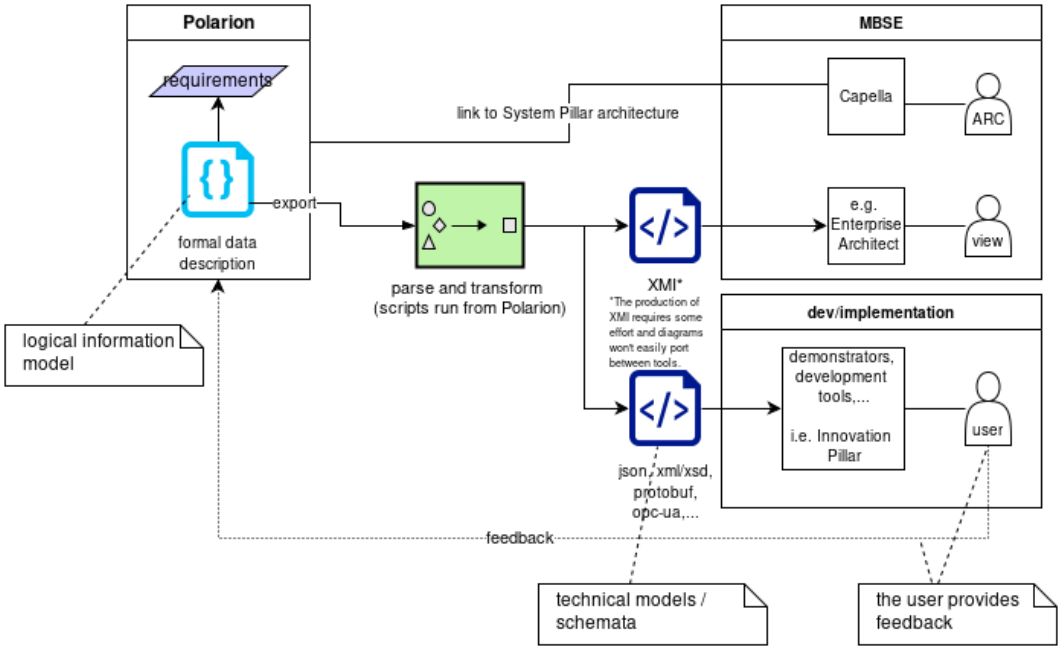


Figure 5: Handling the data model. The data model is a formal description of the object types and their relations. The ensuing technical information models are the schemata for transporting actual data over-the-wire or as files.

[Content to be approved **]**