

TCCS SD1 - Data Model - Schema


1 Table of Contents

1	Table of Contents	1
2	Description	1

2 Description

SPT2TS-124877 - The Schema can also be found in the ERTMS sharepoint.

Download Folder: TCCS SD1 - Data Model Schema: <https://eeigertms.sharepoint.com/:f:/r/sites/SPOpenShare/Gedeelde%20documenten/General/23-07-06%20TCCS%20SD1%20Data%20Model%20v0.2?csf=1&web=1&e=OLAhJ>

 Content to be approved]

SPT2TS-123924 -

```
{
  "$id": "https://ERJU",
  "$schema": "http://json-schema.org/draft-08/schema#",
  "definitions": {
    "attr": {
      "type": "object",
      "properties": {
        "intId": { "type": "integer", "minimum": 1, "description": "zero-based position" },
        "name": { "type": "string", "pattern": "^[a-z][a-zA-Z0-9_]{0,64}$",
          "description": "Expressive name of the attribute. Given in camelCase and must be accepted as an attribute name by well known programming languages."
        },
        "dataType": {
          "type": "string",
          "enum": [ "uint32", "int32", "uint64", "int64", "double", "float", "string", "bytes", "boolean", "timestamp" ],
          "description": "The data type of an attribute. Bytes are given as base64 string"
        },
        "composition": {
          "type": "string", "description": "Reference to an owned class. In format prefix.ClassName. Prefix is needed only if in a different namespace."
        },
        "reference": {
          "type": "string",
```

"description": "Reference to a shared class. In format prefix.ClassName, Prefix is needed only if in a different namespace."

},
"enumType": {
 "type": "string",
 "description": "Reference to an enumeration. In format prefix.EnumerationName, prefix is needed only if in a different namespace."

},
"key": {
 "type": "string", "default": "none",
 "description": "defines if the data-object is addressable by this attribute",
 "enum": ["none", "local", "global"]
},
"sameKeyAs": {
 "type": "string",
 "description": "if a big class is splitted into parts (layers), all parts share the same key representing original object. Here is the reference to other part-class e.g. infra.TrackEdge"

},
"multiplicity": {
 "type": "string",
 "description": "The range of occurrences, e.g. 2, 1..2, 1..*",
 "pattern": "(\\d\\.\\.\\.\\d+|\\|*)|\\d+|\\|*",
 "default": "1"
},
"sortedByKey": {
 "type": "boolean", "default": false,
 "description": "true, if the elements are sorted according to their key-attribute"
},
"unit": {
 "type": "string",
 "enum": ["m", "s", "minute", "km/h", "m/s", "m/s2", "m2", "m3", "kg", "kN", "Pa", "V", "W", "A", "Ohm", "K", "Hz", "degree", "rad", "1/m2", "W/m2", "permill", "percent"]

},
"info": {"type": "string"},
"isDefinedBy": {"type": "string", "description": "Reference URL"},
"deprecated": {"type": "boolean", "default": false},
"exp": {"type": "integer", "description": "factor to use int for fixed point e.g. 23e-3=0.023"},
"range": {"type": "string", "pattern": "\\d+\\.\\.\\.\\d+", "description": "for integer values if known"},
},
"required": ["intId", "name"],
"oneOf": [
 {"required": ["dataType"]},
 {"required": ["composition"]},
 {"required": ["reference"]},
 {"required": ["enumType"]}
],
"additionalProperties": false,
},
"EnumLiteral": {

```

"type": "object",
"properties": {
  "name": {"type": "string", "pattern": "^[a-zA-Z][a-zA-Z0-9_]{0,120}$",
    "description": "Name of the enumerator (aka field). Accepted in well known programming languages.
Preferably in capitals"
  },
  "intId": {"type": "integer",
    "description": "integer value to which this enumerator maps. Typically zero-based. The most commonly used
enumerator should evaluate to 0"
  },
  "info": {"type": "string", "description": "=description"},
  "see": {"type": "string", "description": "URL pointing to authoratative documentation, anywhere on the web"}
},
"required": ["name", "intId"],
"additionalProperties": false
},
"struct": {
  "type": "object",
  "properties": {
    "name": {"type": "string", "pattern": "^[A-Z][a-zA-Z0-9_]{0,64}$",
      "description": "Expressive name of the struct, written in PascalCase and accepted in well known programming
languages."
    },
    "see": {
      "type": "string", "description": "reference to an ontology defining the term"
    }
  },
  "extends": {"type": "string",
    "description": "a reference to another struct written as prefix.StructName where prefix is an abbreviation of the
namespace URL"
  },
  "attrs": {"type": "array", "items": {"$ref": "#/definitions/attr"}},
  "info": {"type": "string", "description": "Semantics of the class; unequivocal and clear description intended for
subject matter experts"},
  "see": {"type": "string", "description": "URL pointing to authoratative documentation, anywhere on the web"},
  "deprecated": {"type": "boolean", "default": false},
  "belongsToSubPackage": {"type": "string", "description": "Reference to a package that the class belongs to, this
expresses that this class is in a subject area inside the present namespace"},
  "union": {"type": "boolean", "default": false, "info": "only one attribute is allowed. Valid for dataTypes,
enumTypes, compositions and references in mixture."}
},
"required": ["name"],
"additionalProperties": false
},
"enum": {
  "type": "object",
  "properties": {
    "name": {"type": "string", "pattern": "^[A-Z][a-zA-Z0-9_]{0,64}$",
      "description": "Expressive name of the enumeration, written in PascalCase and accepted in well known
programming languages."
    }
  }
}

```

```

},
"see": {
  "type": "string", "description": "reference to an ontology defining the term"
}
"enumLiterals": {"type": "array", "items": {"$ref": "#/definitions/EnumLiteral"}},
"info": {"type": "string"},
"isDefinedBy": {"type": "string", "description": "Reference url"},
"deprecated": {"type": "boolean", "default": false},
"belongsToSubPackage": {"type": "string", "description": "belongs to package listed in the same namespace in
subPackages-property."}
},
"required": ["name", "enumLiterals"],
"additionalProperties": false
},
"package": {
  "type": "object",
  "properties": {
    "name": {"type": "string",
      "description": "groups classes of a similar concern within the namespace for the purpose of documentation"
    }
  }
},
"required": ["name"],
"additionalProperties": false
}
},
"type": "object",
"description": "Definition of the namespace (module)",
"properties": {
  "info": {"type": "string"},
  "$schema": {"type": "string", "description": "reference to this schema in each module"},
  "version": {"type": "string", "description": "E.g. 1.02a or 2021-10-07"},
  "isDefinedBy": {"type": "string", "description": "Reference url, e.g. http://example.eu/domain/topology"},
  "containerStruct": {
    "type": "string",
    "description": "name of the struct representing the main container of the package. It is used for creating
references /nameOfContainerStruct/nameOfAttr/nameOfAttr/nameOfAttr"
  },
  "deprecated": {"type": "boolean", "default": false, "description": "true when this namespace will be deleted from
future releases"},
  "name": {"type": "string",
    "description": "Name of this namespace that expresses the discipline, e.g. Power Supply. Typically the same
as, or similar to, the last part of the namespace"
  },
  "prefix": {
    "type": "string", "description": "namespace abbreviation used for name resolution. E.g. TP.TrackEdge is found
inside the namespace given by isDefinedBy."
  },
  "intId": {"type": "integer", "minimum": 1,
    "description": "

```

use-case1: referencing in Data (Timetable.Trip = /30/7) - main use-case,
use-case2: for binary protocols replacing attr-names Protobuf/OPC-UA,
use-case3: referencing in Model (**TP**.TrackEdge) - *use prefix.localName instead*

```
},  
"enums": {"type": "array", "items": {"$ref": "#/definitions/enum"}},  
"structs": {"type": "array", "items": {"$ref": "#/definitions/struct"}},  
"subPackages": {"type": "array", "items": {"$ref": "#/definitions/package"},  
  "description": "list of packages"  
}  
},  
"required": ["intId", "prefix", "name", "isDefinedBy"],  
"additionalProperties": false  
} [🔍🔒 Content to be approved ]
```