

Plugins configuration design

- Introduction
- Overview
 - Functional architecture
 - Configuration data categories
- Protocol translation
 - Pivot object definition
 - Use case 1: monitoring
 - Single Substation/RTU gateway
 - Multiple Substation/RTU gateway
 - Use case 2: monitoring and control
 - Single Scada/single protocol
 - Multiple Scadas/multiple protocols
 - Benefits and challenges of a pivot object
 - Benefits
 - Challenges
 - Global process from data acquisition to data exposition
 - Pivot object model
 - PIVOT TM model, representation of a tele measurement
 - Example of a pivot model instance representing a Tele Measurement:
 - PIVOT TS model, representation of a tele signal
 - Example of a pivot model instance representing a Tele Signal:
 - PIVOT TC model, representation of a tele command
 - Example of a pivot model instance representing a Tele Command:
 - Protocol translation process
- Exchanged data configuration
 - Attributes definition
 - Configuration JSON structure

Introduction

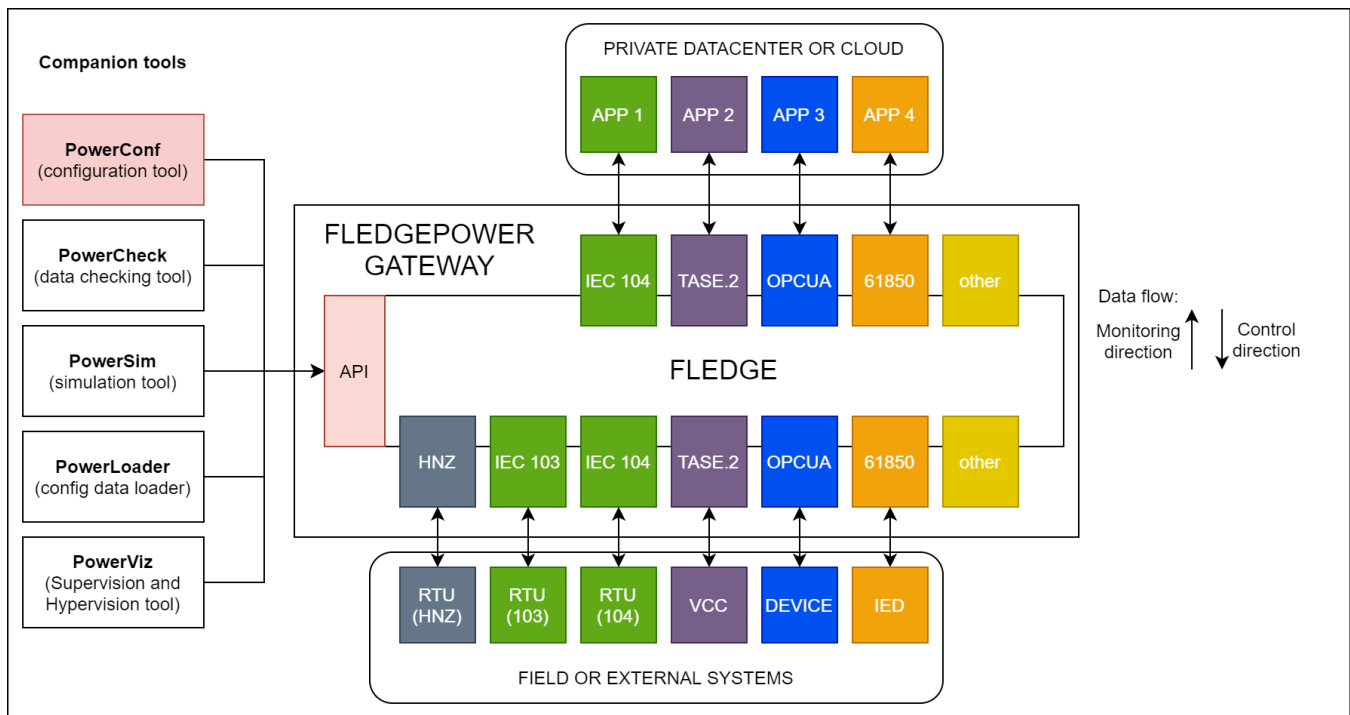
Being able to configure the way a protocol plugin will manage the communication with the remote servers or clients using the protocol stack parameters, whether communication is secured or not, how the messages attributes are translated from the input protocol data model to the output protocol data model and what kind of rules (filters, math, routing, etc..) to apply to the messages that are exchanges with the remote systems are key features of any protocol translation gateway.

In this design document we will be describing the implementation of the 3 main plugins configuration data models: **protocol stack configuration** (including application layer, transport layer and security layer), **protocols translation configuration**, **exchanged datapoints configuration** and **data processing/filtering rules**.

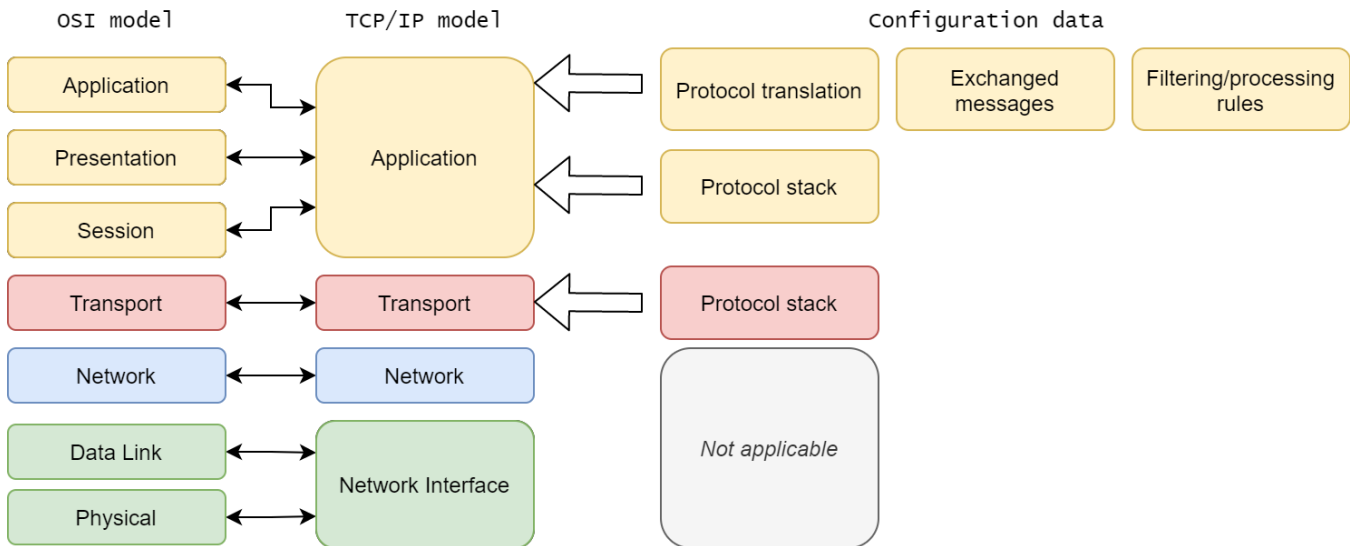
FledgePOWER being built on top of Fledge, all the configuration will be described using JSON data structures using nested objects and will be sent to the gateway using the provided HTTP API.

Overview

Functional architecture



Configuration data categories



Protocol translation

Pivot object definition

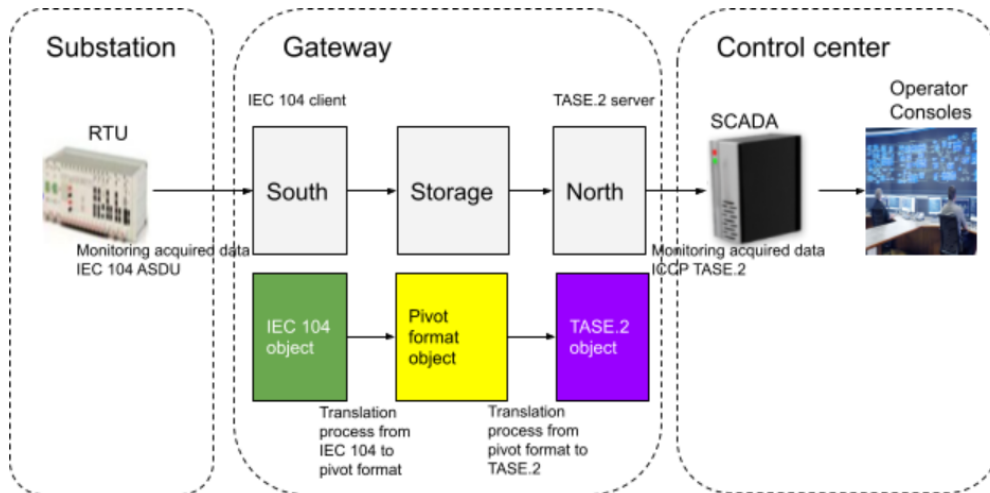
Before diving into the details of the specification, it is useful to illustrate some use cases. They should help demonstrate how the specification is expected to be used and to help understand the benefits of a pivot object.

The objective of a gateway is to allow communication between different systems by providing some translation mechanisms from one means of communication to another one.

All translation mechanisms introduce a limitation of functionality in the translated space compared to the original space. The major challenge is to provide some correspondences between the two systems, with a limited impact to the original scope of features.

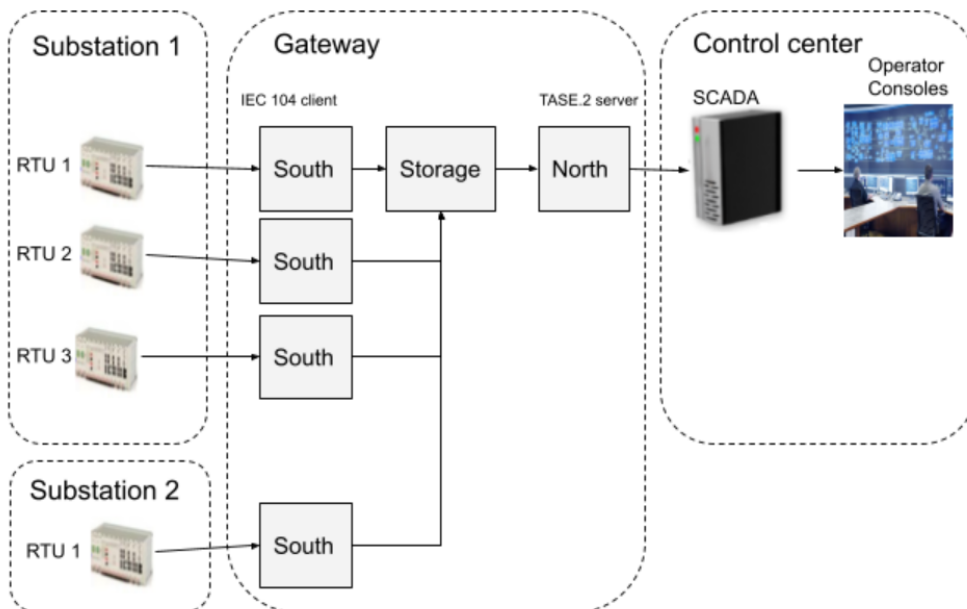
Use case 1: monitoring

Single Substation/RTU gateway



In this use case, monitoring data are acquired from a field device called RTU (Remote Terminal Unit) which acts as a IEC 104 server (slave) and are exposed to the central SCADA which acts as a ICCP TASE.2 client. Each incoming message from the RTU comes as an IEC 104 ASDU object. The south service plugin applies a first translation process which consists of (1) reading and checking the ASDU object using the IEC 104 data model and (2) mapping each data attribute to the corresponding pivot object data attribute. Once the message is converted to a pivot object, it can then be stored or consumed by the north or any other service. The north service plugin can then apply a second translation process which consists of (3) reading and checking the pivot object using the pivot data model and (4) mapping each data attribute to the corresponding TASE.2 data attribute. The resulting TASE.2 object can then be exposed to the TASE.2 client by the north service plugin.

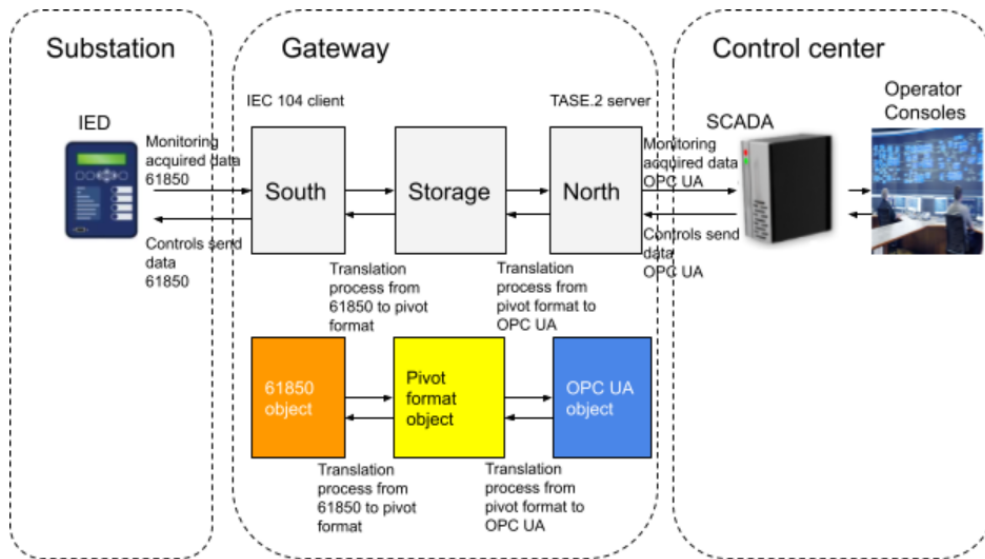
Multiple Substation/RTU gateway



In this alternative scenario, the gateway is connected to multiple RTUs located in multiple substations. Each South plugin (IEC 104 client) is connected to an RTU (IEC 104 server) to collect data. The collected data flow then through the Storage and the North plugin (TASE.2) to expose data to the SCADA.

Use case 2: monitoring and control

Single Scada/single protocol

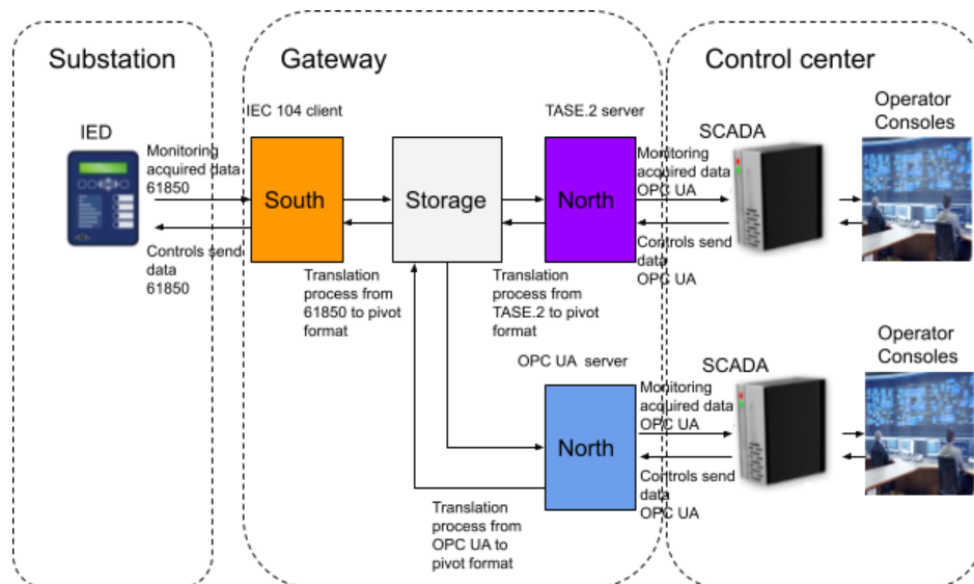


In this use case data flows in two directions: monitoring and control.

Monitoring data are acquired from a field device called IED (Intelligent Electronic Device) which acts as a 61850 MMS server and are exposed to the central SCADA which acts as a OPC UA client. Each incoming message from the IED comes as a 61850 MMS object. The south service plugin applies a first translation process which consists of (1) reading and checking the 61850 MMS object using the 61850 MMS data model and (2) mapping each data attribute to the corresponding pivot object data attribute. Once the message is converted to a pivot object, it can then be stored or consumed by the north or any other service. The north service plugin can then apply a second translation process which consists of (3) reading and checking the pivot object using the pivot object data model and (4) mapping each data attribute to the corresponding OPC UA data attribute. The resulting OPC UA object can then be exposed to the OPC UA client by the north service plugin.

Control data are received from the control center SCADA which acts as a OPC UA client and are sent to the IED which acts as a 61850 server. Each incoming message from the control center SCADA comes as an OPC UA object. The north service plugin applies a first translation process which consists of (1) reading and checking the OPC UA object using the OPC UA data model and (2) mapping each data attribute to the corresponding pivot object data attribute. Once the message is converted to a pivot object, it can then be stored or consumed by the south or any other service. The south service plugin can then apply a second translation process which consists of (3) reading and checking the pivot object using the pivot object data model and (4) mapping each data attribute to the corresponding 61850 MMS data attribute. The resulting 61850 MMS object can then be exposed to the 61850 client by the south service plugin.

Multiple Scadas/multiple protocols



In this alternative scenario, the gateway is connected to multiple SCADAs using two different protocols. The North plugin (TASE.2) exposes data to the TASE.2 SCADA and the North plugin OPC UA exposes data to the OPC UA SCADA. Instead of having a one to one translation process (61850 to TASE.2, 61850 to OPC UA, TASE.2 to 61850, OPC UA to 61850), we use the pivot object to transmit data from 61850 to both TASE.2 and OPC UA. The translation process is done only once for each protocol.

Benefits and challenges of a pivot object

Benefits

The main benefit of the pivot object is it allows decoupling the protocols data model from each other. Each south or north plugin deals with the complexity and the specifics of a given protocol without interfering with the core of the gateway or with another south/north plugin. Adding new protocol plugins or maintaining existing ones is then made much easier. This contributes in maintaining the whole gateway system at a low level of complexity thus minimizing the costs of new developments or maintenance.

Challenges

In order to reach the goal of simplicity we want to achieve, the main challenge is to be able to build a pivot object that is technically independent from the protocols. The data model should also be neutral regarding protocols data models. The protocol translation should not imply a loss or degradation of the source information, especially in the case where the input and output protocols are identical.

Global process from data acquisition to data exposition

This is an illustration with IEC 104 as input protocol and IEC 104, TASE.2 and OPC UA as output protocols.

We have defined here 3 data processing pipelines (see Fledge documentation about [Data processing](#) for more details.)

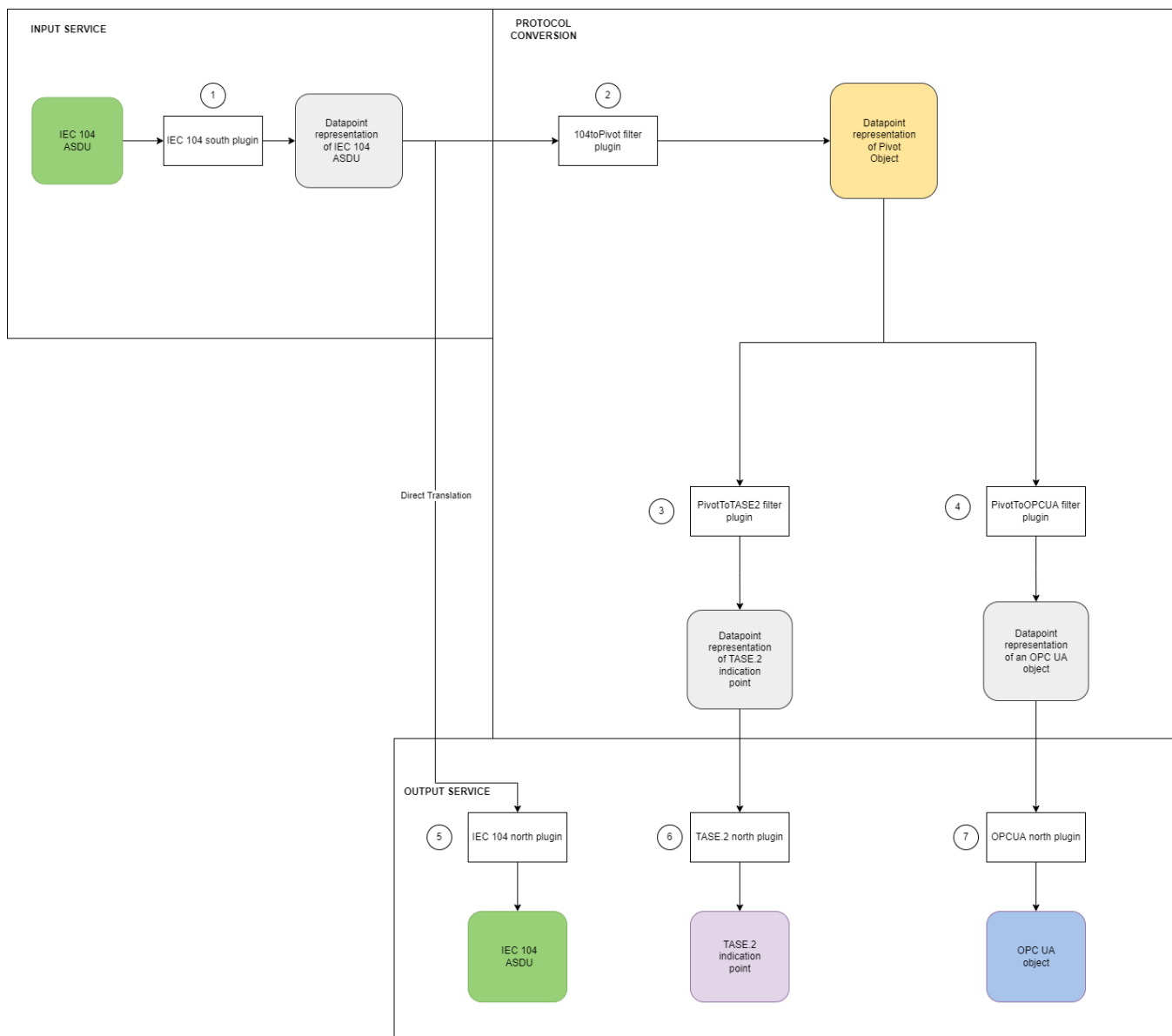
IEC 104 to IEC 104 direct translation : (1) => (5)

IEC 104 to TASE.2 protocols conversion : (1) => (2) => (3) => (6)

IEC 104 to OPCUA protocols conversion: (1) => (2) => (4) => (7)

Process steps:

- (1) IEC 104 ASDU acquisition using the iec104 south plugin, the output is a Datapoint representing the input message.
- (2) From IEC 104 to Pivot conversion using the 104toPivot filter plugin, the output is a Datapoint representing a Pivot object of a specific datatype.
- (3) Pivot To Tase2 conversion using the PivotToTase2 filter plugin, the output is a Datapoint representing a TASE.2 object of a specific datatype.
- (4) Pivot To Opcua conversion using the PivotToOpcua filter plugin, the output is a Datapoint representing a OPCUA object of a specific datatype.
- (5) IEC 104 ASDU exposition using the iec104 north plugin, the output is a IEC 104 ASDU.
- (6) TASE.2 exposition using the tase2 north plugin, the output is a TASE.2 indication point.
- (7) OPCUA exposition using the opcua north plugin, the output is a OPCUA object.



Pivot object model

The pivot object model is based on 61850 semantic.

PIVOT TM model, representation of a tele measurement

Attribute	Description	Expected values	Mandatory
PIVOT.GTIM.Beh.stVal	?		
PIVOT.GTIM.Cause.stVal	Cause of transmission	COT Cause of Transmission - FledgePower - LF Energy	Yes
PIVOT.GTIM.ChgValCnt.stVal	?		
PIVOT.GTIM.ComingFrom	Source protocol name	IEC103, IEC104, TASE.2, OPCUA, 61850, etc...	Yes
PIVOT.GTIM.Confirmation.stVal	Indicates positive or negative confirmation of activation	true, false, DEFAULT=false	
PIVOT.GTIM.Identifier	Unique identifier of the pivot object	Any non-empty string	Yes
PIVOT.GTIM.{CDC}	Common Data Class representing the information object	MVTyp Common Data Classes - FledgePower - LF Energy	Yes
PIVOT.GTIM.NormalSrc.stVal	?		
PIVOT.GTIM.NormalVal.stVal	?		

PIVOT.GTIM.TmOrg.stVal	Timestamp origin, genuine = source otherwise substituted	genuine, substituted, DEFAULT = genuine	Yes
PIVOT.GTIM.TmValidity.stVal	Timestamp validity	valid, invalid, DEFAULT = valid	Yes

Example of a pivot model instance representing a Tele Measurement:

```
{
  "@xmlns:xs": "http://www.w3.org/2001/XMLSchema",
  "PIVOT": {
    "GTIM": {
      "Beh": {
        "stVal": "on"
      },
      "Cause": {
        "stVal": 1
      },
      "ChgValCnt": {
        "stVal": 1
      },
      "ComingFrom": "String",
      "Confirmation": {
        "stVal": true
      },
      "Identifier": "String",
      "MvTyp": {
        "mag": {
          "f": 0.1,
          "i": 1
        },
        "q": {
          "DetailQuality": {
            "badReference": true,
            "failure": true,
            "inconsistent": true,
            "innacurate": true,
            "oldData": true,
            "oscillatory": true,
            "outOfRange": true,
            "overflow": true
          },
          "Source": "process",
          "Validity": "good",
          "operatorBlocked": true,
          "test": true
        },
        "t": {
          "FractionOfSecond": 1,
          "SecondSinceEpoch": 1,
          "TimeQuality": {
            "clockFailure": true,
            "clockNotSynchronized": true,
            "leapSecondKnown": true,
            "timeAccuracy": 1
          }
        }
      },
      "NormalSrc": {
        "stVal": "TELEMETERED"
      },
      "NormalVal": {
        "stVal": "NORMAL"
      },
      "Origin": {
        "q": {
          "DetailQuality": {
            "badReference": true,
            "failure": true,
            "inconsistent": true,
            "innacurate": true,
            "oldData": true,

```

```

        "oscillatory": true,
        "outOfRange": true,
        "overflow": true
    },
    "Source": "process",
    "Validity": "good",
    "operatorBlocked": true,
    "test": true
},
"stVal": "String",
"t": {
    "FractionOfSecond": 1,
    "SecondSinceEpoch": 1,
    "TimeQuality": {
        "clockFailure": true,
        "clockNotSynchronized": true,
        "leapSecondKnown": true,
        "timeAccuracy": 1
    }
}
},
"tmOrg": {
    "stVal": "genuine"
},
"tmValidity": {
    "stVal": "VALID"
}
}
}
}
}

```

PIVOT TS model, representation of a tele signal

Attribute	Description	Expected values	Mandatory
PIVOT.GTIS.Beh.stVal	?		
PIVOT.GTIS.Cause.stVal	Cause of transmission	COT Cause of Transmission - FledgePower - LF Energy	Yes
PIVOT.GTIS.ChgValCnt.stVal	?		
PIVOT.GTIS.ComingFrom	Source protocol name	IEC103, IEC104, TASE.2, OPCUA, 61850, etc...	Yes
PIVOT.GTIS.Confirmation.stVal	Indicates positive or negative confirmation of activation	true, false, DEFAULT=false	
PIVOT.GTIS.Identifier	Unique identifier of the pivot object	Any non-empty string	Yes
PIVOT.GTIS.{CDC}	Common Data Class representing the information object	SPSTyp, DPSTyp, BSCTyp Common Data Classes - FledgePower - LF Energy	Yes
PIVOT.GTIS.NormalSrc.stVal	?		
PIVOT.GTIS.NormalVal.stVal	?		
PIVOT.GTIS.TmOrg.stVal	Timestamp origin, genuine = source otherwise substituted	genuine, substituted, DEFAULT = genuine	Yes
PIVOT.GTIS.TmValidity.stVal	Timestamp validity	valid, invalid, DEFAULT = valid	Yes

Example of a pivot model instance representing a Tele Signal:

```

{
  "@xmlns:xs": "http://www.w3.org/2001/XMLSchema",
  "PIVOT": {
    "GTIS": {
      "Beh": {
        "stVal": "on"
      },
      "Cause": {
        "stVal": 1
      },
      "ChgValCnt": {
        "stVal": 1
      },
    },
  },
}

```

```

"ComingFrom": "String",
"Confirmation": {
    "stVal": true
},
"DpsTyp": {
    "q": {
        "DetailQuality": {
            "badReference": true,
            "failure": true,
            "inconsistent": true,
            "innacurate": true,
            "oldData": true,
            "oscillatory": true,
            "outOfRange": true,
            "overflow": true
        },
        "Source": "process",
        "Validity": "good",
        "operatorBlocked": true,
        "test": true
    },
    "stVal": "intermediate-state",
    "t": {
        "FractionOfSecond": 1,
        "SecondSinceEpoch": 1,
        "TimeQuality": {
            "clockFailure": true,
            "clockNotSynchronized": true,
            "leapSecondKnown": true,
            "timeAccuracy": 1
        }
    }
},
"Identifier": "String",
"NormalSrc": {
    "stVal": "TELEMETERED"
},
"NormalVal": {
    "stVal": "NORMAL"
},
"Origin": {
    "q": {
        "DetailQuality": {
            "badReference": true,
            "failure": true,
            "inconsistent": true,
            "innacurate": true,
            "oldData": true,
            "oscillatory": true,
            "outOfRange": true,
            "overflow": true
        },
        "Source": "process",
        "Validity": "good",
        "operatorBlocked": true,
        "test": true
    },
    "stVal": "String",
    "t": {
        "FractionOfSecond": 1,
        "SecondSinceEpoch": 1,
        "TimeQuality": {
            "clockFailure": true,
            "clockNotSynchronized": true,
            "leapSecondKnown": true,
            "timeAccuracy": 1
        }
    }
},
"SpsTyp": {
    "q": {

```

PIVOT TC model, representation of a tele command

Attribute	Description	Expected values	Mandatory
PIVOT.GTIC.Beh.stVal	defines different behaviours of a control object: - Direct control with normal security: dct-ctl-wns - Direct control with enhanced security: dct-ctl-wes - SBO (Select Before Operate) control with normal security: sbo-ctl-wns - SBO (Select Before Operate) control with enhanced security: sbo-ctl-wes	enum: dct-ctl-wns, dct-ctl-wes, sbo-ctl-wns, sbo-ctl-wes, DEFAULT = dct-ctl-wes	No
PIVOT.GTIC.Cause.stVal	Cause of transmission	COT Cause of Transmission - FledgePower - LF Energy	No
PIVOT.GTIC.ComingFrom	Source protocol name	IEC103, IEC104, TASE.2, OPCUA, 61850, etc...	Yes
PIVOT.GTIC.DevId.name	?		No
PIVOT.GTIC.{CDC}	Common Data Class representing the information object	SPCTyp, DPCTyp, INCTyp, APCTyp, BSCTyp Common Data Classes - FledgePower - LF Energy	Yes
PIVOT.GTIC.Identifier	Unique identifier of the pivot object	Any non-empty string	Yes
PIVOT.GTIC.Qu.stVal	?		No
PIVOT.GTIC.ServiceType.stVal	?		No
PIVOT.GTIC.Tag.stVal	?		No
PIVOT.GTIC.TagClass	?		No
PIVOT.GTIC.TaggedReason	?		No

PIVOT.GTIC.TmOrg	Timestamp origin, genuine = source otherwise substituted	genuine, substituted, DEFAULT = genuine	No
PIVOT.GTIC.TmStamp	Timestamp		No

Example of a pivot model instance representing a Tele Command:

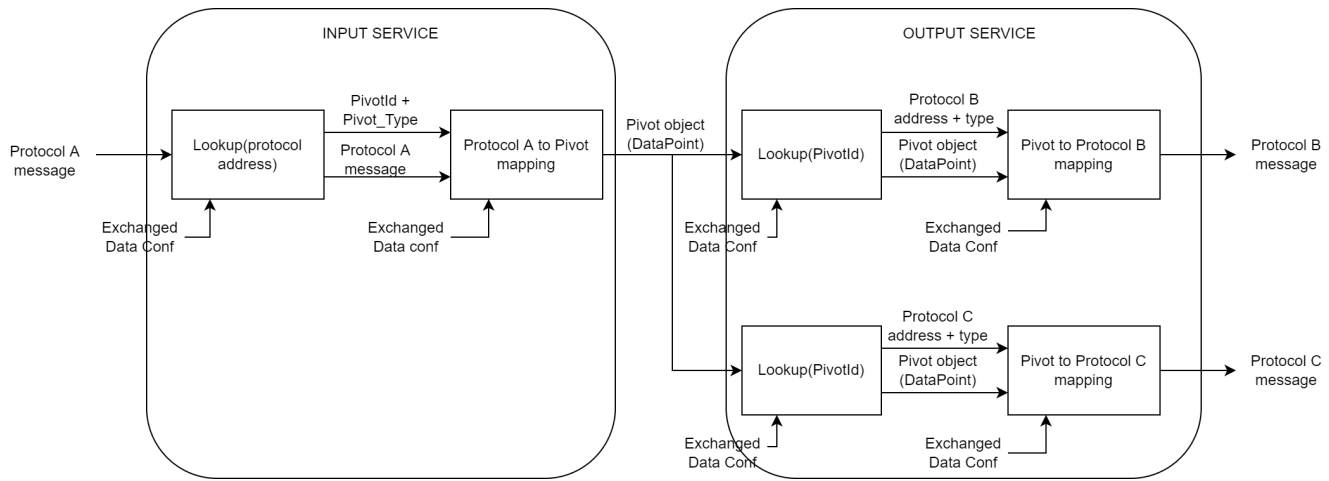
```
{
  "@xmlns:xs": "http://www.w3.org/2001/XMLSchema",
  "PIVOT": {
    "GTIC": {
      "Beh": {
        "stVal": "dct-ctl-wes"
      },
      "Cause": {
        "stVal": 1
      },
      "ComingFrom": "String",
      "DevId": {
        "name": "String"
      },
      "DevSt": {
        "q": {
          "DetailQuality": {
            "badReference": true,
            "failure": true,
            "inconsistent": true,
            "innacurate": true,
            "oldData": true,
            "oscillatory": true,
            "outOfRange": true,
            "overflow": true
          },
          "Source": "process",
          "Validity": "good",
          "operatorBlocked": true,
          "test": true
        },
        "stVal": true,
        "t": {
          "FractionOfSecond": 1,
          "SecondSinceEpoch": 1,
          "TimeQuality": {
            "clockFailure": true,
            "clockNotSynchronized": true,
            "leapSecondKnown": true,
            "timeAccuracy": 1
          }
        }
      },
      "DpcTyp": {
        "ctlVal": true,
        "q": {
          "DetailQuality": {
            "badReference": true,
            "failure": true,
            "inconsistent": true,
            "innacurate": true,
            "oldData": true,
            "oscillatory": true,
            "outOfRange": true,
            "overflow": true
          },
          "Source": "process",
          "Validity": "good",
          "operatorBlocked": true,
          "test": true
        },
        "stVal": "intermediate-state",
        "t": {
          "FractionOfSecond": 1,

```

```

        "SecondSinceEpoch": 1,
        "TimeQuality": {
            "clockFailure": true,
            "clockNotSynchronized": true,
            "leapSecondKnown": true,
            "timeAccuracy": 1
        }
    },
    "Identifier": "String",
    "Qu": {
        "stVal": 1
    },
    "ServiceType": {
        "stVal": "select"
    },
    "SpcTyp": {
        "ctlVal": true,
        "q": {
            "DetailQuality": {
                "badReference": true,
                "failure": true,
                "inconsistent": true,
                "innacurate": true,
                "oldData": true,
                "oscillatory": true,
                "outOfRange": true,
                "overflow": true
            },
            "Source": "process",
            "Validity": "good",
            "operatorBlocked": true,
            "test": true
        },
        "stVal": true,
        "t": {
            "FractionOfSecond": 1,
            "SecondSinceEpoch": 1,
            "TimeQuality": {
                "clockFailure": true,
                "clockNotSynchronized": true,
                "leapSecondKnown": true,
                "timeAccuracy": 1
            }
        }
    },
    "Tag": {
        "stVal": "NO-TAG"
    },
    "TagClass": {
        "q": {
            "DetailQuality": {
                "badReference": true,
                "failure": true,
                "inconsistent": true,
                "innacurate": true,
                "oldData": true,
                "oscillatory": true,
                "outOfRange": true,
                "overflow": true
            },
            "Source": "process",
            "Validity": "good",
            "operatorBlocked": true,
            "test": true
        },
        "stVal": true,
        "t": {
            "FractionOfSecond": 1,
            "SecondSinceEpoch": 1,
            "TimeQuality": {

```

Exchanged data configuration

In this chapter we will describe the configuration of the exchanged data. This configuration allow to specify a list of supported data objects. The protocol plugin is expected to make some controls against each entry of the configuration to check:

- if protocol message address or reference is known otherwise throw an error message.
- if protocol type of message is known otherwise throw an error message.
- The label attribute shall be used to populate the Fledge's DataPoint Asset Name attribute.

Attributes definition

Attribute	Description	Expected values	Mandatory
name	this identifies the exchanged data configuration		Yes
version	this is the version number of the configuration	x.y where x represents a major version and y a minor change	Yes
datapoints	array of datapoints that needs to be managed by the instance of the gateway		Yes
datapoints.label	label of the datapoint	any non empty string	Yes
datapoints.pivot_id	unique identifier of the datapoint, this is used to create a pivot object	any non empty string	Yes
datapoints.pivot_type	type of Common Data Class (CDC), this is used to create a pivot object	enum SpsTyp DpsTyp MvTyp SpcTyp DpcTyp ...	Yes
datapoints.pivot_subtypes	array of additional types of the pivot object	[transient, ...]	No
datapoints.tfId	identifies the math function to transform the input value	enum "normal" "square_root" "quadratic" "transparent"	No
datapoints.params	array of parameters for the math function	array of float 32 [a, b, c, ...]	No
datapoints.deadband	band of input values where the output is zero	array of 2 float 32 [min, max]	No
datapoints.protocols	array of protocols that needs to be managed for a datapoint		Yes
datapoints.protocols.name	name of the protocol	enum "iec104" "iec103" "tase2" "hnz" "61850" "opcua" ...	Yes
datapoints.protocols.address	address of the datapoint in the given protocol	any non empty string	Yes
datapoints.protocols.typeid	type id of the datapoint in the given protocol	any non empty string representing a valid protocol type id	Yes
datapoints.protocols.gi_groups	GI request status for north plugins, if "station" then datapoint is send for south plugins, it determines whether a datapoint is expected in the GI response	enum station 1 2 ...	No
datapoints.protocols.alternate_mapping_rule	Alternate mapping rule to convert datapoint from protocol model to pivot or pivot to protocol model, if not specified then default conversion rule is applied	any non empty string representing a function name	No

Configuration JSON structure

--

```

{
  "exchanged_data":{
    "name":"SAMPLE",
    "version":"1.0",
    "datapoints":[
      {
        "label":"TS1",
        "pivot_id":"ID114562",
        "pivot_type":"SpsTyp",
        "pivot_subtypes":[
          "transient"
        ],
        "protocols":[
          {
            "name":"iec104",
            "address":"45-672",
            "typeid":"M_SP_TB_1",
            "gi_groups":"station",
            "alternate_mapping_rule":"func_name_1"
          },
          {
            "name":"tase2",
            "address":"S_114562",
            "typeid":"Data_StateQTimeTagExtended"
          },
          {
            "name":"opcua",
            "address":"ID114562",
            "typeid":"opcua_sps"
          },
          {
            "name":"iec61850",
            "address":"simpleIOGenericIO/GGI01.Ind1",
            "typeid":"SPS"
          }
        ]
      },
      {
        "label":"TM1",
        "pivot_id":"ID99876",
        "pivot_type":"MVTyp",
        "tfid":"square_root",
        "params":[
          2.0,
          49.5
        ],
        "deadband":[
          -0.02,
          0.02
        ],
        "protocols":[
          {
            "name":"iec104",
            "address":"45-984",
            "typeid":"M_ME_NA_1"
          },
          {
            "name":"tase2",
            "address":"S_114562",
            "typeid":"Data_RealQ"
          },
          {
            "name":"opcua",
            "address":"ID99876",
            "typeid":"opcua_mvf"
          },
          {
            "name":"iec61850",
            "address":"simpleIOGenericIO/GGI01.AnIn1",
            "typeid":"MV"
          }
        ]
      }
    ]
  }
}

```

```
}  
  }  
    ]  
      }  
        ]
```



Do not use the name "IEC104Command" as a datapoint name in the exchanged data configuration. Doing so could potentially lead to conflicts when trying to read readings from both the South and Dispatcher components.