System status points

Module description

This plugin aims to manage system status points. A system status point is a specific type of status points used to transmit technical information on the operating status of the gateway or the equipment to which it is connected

In the current version, the plugin handles the following system status points :

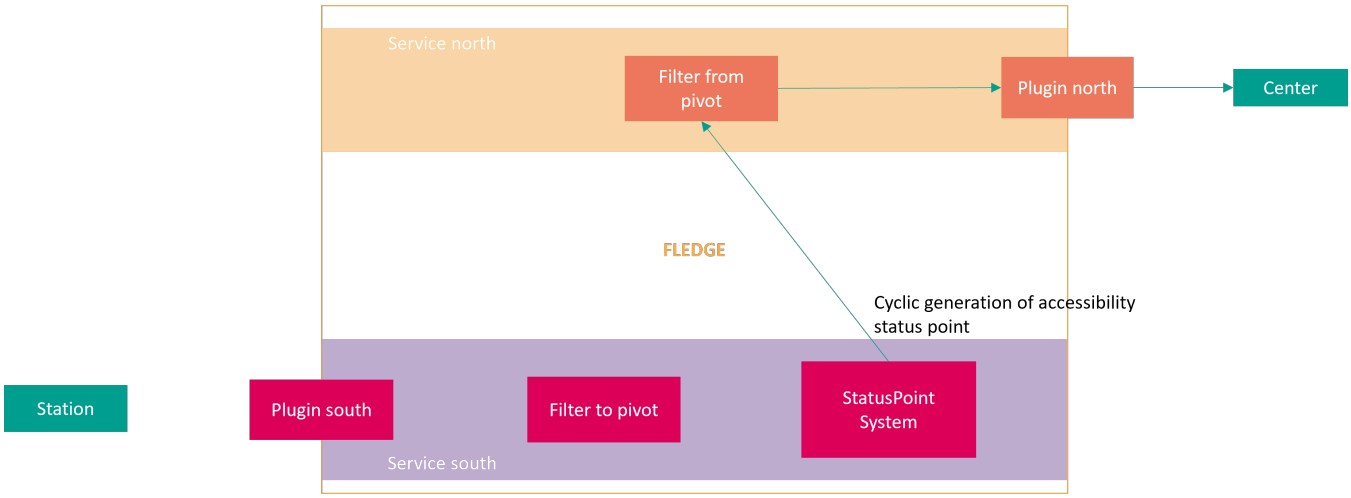
- ACCESS status point
- CONNEXION LOSS status point

The plugin can be extended to add more system status points in the future if needed.

ACCESS status point

The 104 centers must receive an accessibility status point indicating that the connected RTU is still functional.

This status message is generated and sent cyclically. The details of the sent message and its cycling are determined in the configuration.



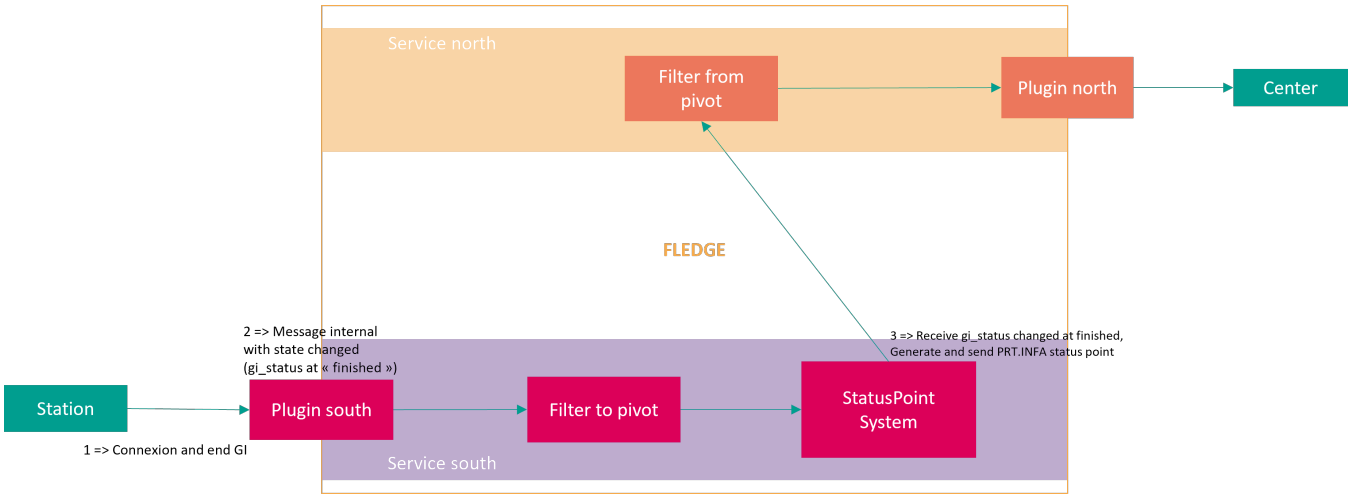
CONNEXION LOSS status point

In case of a loss of connection with the RTU, the idea is to inform the Center that the connection has been lost. This information is sent as follows:

- A 0 is sent when the RTU is disconnected.
- A 1 is sent to the Center after the end of the GI from the RTU, triggering a request for GI from the Center.

Therefore, the plugin must detect the disconnection of the RTU and the end of the GI issued by the South service to send the information to the Center. In this case, the plugin must create and send a unique and specific datapoint to the North, which will, in turn, transmit it to the Center. This datapoint should be present in the *Exchanged_data* configuration, tagged with the subtype *prt.info*.

Since this information is configured as transient, it is not necessary to reset it to 0. This implies that the Status Point plugin must be placed **before** the transient plugin in the sequence of south plugins.



Configuration

ACCESS status point

The configuration of a status point cyclic is present in the *Exchanged_data* configuration.

An attribute called "ts_syst_cycle" is added to allow the configuration of a status point as cyclic (in seconds).

Example :

```
{
        "label" :"GAMBS_0_TS-SYST_ACCES",
        "pivot_id" :"xxx",
        "pivot_type" :"SpsTyp",
        "pivot_subtypes" : ["acces"],
        "ts_syst_cycle" :30
        "protocols" :[
           {
                  "name" :"hnzip",
                  "typeid":"yyy",
                  "address":"zzz"
           }
        ]
}
```

CONNEXION LOSS status point

The configuration of the status point in *Exchanged_data* includes the subtype of the status point. To configure a status point as PRT.INFO, the *pivot_subtypes* attribute is set to *prt.inf*.

```
{
        "label" :"GAMBS_0_TS-SYST_PRT.INFA",
        "pivot_id" :"xxx",
        "pivot_type" :"SpsTyp",
        "pivot_subtypes" : ["prt.inf"],
        "protocols" :[
           {
                  "name" :"hnzip",
                  "typeid":"yyy",
                  "address":"zzz"
           }
        ]
}
```

Also, a new attribute is configured for PRT.INFO. This attribute allows naming the status reading of the monitored South service.

The format of this attribute is a "string":

"asset": "CONSTAT-1"

Where:

- "asset" is the name of the status transmission asset.

Filtering rules

ACCESS status point

The readings are transmitted without any check or modification.

CONNEXION LOSS status point

The readings are transmitted without any check or modification.

Data processing

ACCESS status point

Input

The cycling of the status point is not conditioned by the reception of data by the module.

Ouput

For each status point configured to be sent cyclically, the generated data is as follows:

For a Simple type data:

- PIVOT.GTIS.Cause.stVal = 3
- PIVOT.GTIS.Identifier = Identifier from the configuration
- PIVOT.GTIS.<TypeDataInExchangedData>.stVal = True
- PIVOT.GTIS.<TypeDataInExchangedData>.t.SecondSinceEpoch (Timestamp of the gateway)
- PIVOT.GTIS.<TypeDataInExchangedData>.t.FractionOfSecond (Timestamp of the gateway)
- PIVOT.GTIS.<TypeDataInExchangedData>.q.Source = substituted
- PIVOT.GTIS.TmOrg.stVal = substituted

The type of the status point is read directly from the *exchanged_data* configuration.

CONNEXION LOSS status point

Input

The plugin listens to the status reading of the associated South service.

If the "connx_status" is in the "not connected" state, the plugin must generate a status point configured with the value 0.

If the "gi_status" is in the "finished" state, the plugin must generate a status piont configured with the value 1.

Output

The plugin generates the status point in its output as follows:

- For a Simple type data:
  - PIVOT.GTIS.Cause.stVal = 3
  - PIVOT.GTIS.Identifier = Identifier from the configuration
  - PIVOT.GTIS.<TypeDataInExchangedData>.stVal = True
  - PIVOT.GTIS.<TypeDataInExchangedData>.t.SecondSinceEpoch (gateway timestamp)
  - PIVOT.GTIS.<TypeDataInExchangedData>.t.FractionOfSecond (gateway timestamp)
  - PIVOT.GTIS.<TypeExchnagedData>.q.Source = substituted

- PIVOT.GTIS.TmOrg.stVal = substituted

The type of the status point is read directly from the *exchanged_data* configuration.

Implementation

Fledge plugins

In order to implement the features above, a plugin with the following features needs to be created:

- Ability to read the Reading data sent by the south plugin.
- Ability to create and send new Readings in the system at will (eg: when a timer expires).

Currently the plugin types available in Fledge (**Filter**, **Notification Rule** and **Notification Delivery**) do not offer these functionalities in a single plugin:

Filter plugins allow to read, edit, add and/or remove Readings passing through Fledge, but only when at least one Reading was published by a south plugin.

Notification Rule plugins allow to send Notification messages (not Readings) when an asset was modified and a predefined set of rules is matched by this asset (so no free Reading creation).

Notification Delivery plugins allow to receive Notifications and act accordingly by producing external API calls to other services (logging, DB, etc...) or internal Fledge API calls (including the creation of Readings).

Design choices

In order to achieve the features needed here, two plugins are created:

- A **Notification Rule** plugin (fledgepower-rule-systemsp) responsible of scanning the Readings flowing through Fledge and generate a Notification when a relevant event occured.
- A **Notification Delivery** plugin (fledgepower-notify-systemsp) responsible of:
  - Receiving the Notifications from the other plugin, create new Readings based on the data they contain if necessary, and send those through Fledge.
  - Run cyclic task that will send a Reading through Fledge at regular intervals.

Here the **Notification Rule** plugin is used to detect changes in readings of type "south_event" and send a notification when a Connection Loss Status Point should be created containing the type of event it represents ("connected" / "connection lost").

The **Notification Delivery** plugin is used to send Access Status Point Readings periodically based on the exchanged_data configuration and to send Connection Loss Status Point Readings when receiving a notification from the **Notification Rule** plugin.

Notification structure:

```
{
	"asset": "prt.inf",
	"reason": "connected" // or "connection lost"
}
```