

---

**FIP API**  
***Release 0.3***

**EcoGrid 2.0**

**Aug 29, 2019**

# CONTENTS

<b>1</b>	<b>Changelog</b>	<b>2</b>
<b>2</b>	<b>General information</b>	<b>3</b>
<b>3</b>	<b>Protocol</b>	<b>5</b>
<b>4</b>	<b>Greenwave Native Control Interface</b>	<b>10</b>
<b>5</b>	<b>Siemens Native Control Interface</b>	<b>12</b>
<b>6</b>	<b>Siemens Simple Control Interface</b>	<b>15</b>
<b>7</b>	<b>Glossary</b>	<b>17</b>
	<b>Index</b>	<b>18</b>

The FIP (Flexibility Interoperability Platform) is a system built for EcoGrid 2.0. It provides interoperability between aggregators and home energy management systems (also called distributed energy resources, DERs).

## CHANGELOG

### 1.1 0.3

- Renamed Greenwave and Siemens specifications to Greenwave Native Control Interface and Siemens Native Control interface
- Added description of Siemens Simple Control Interface
- Updated Greenwave specification to match Siemens Simple Control Interface

### 1.2 0.2

- Added Greenwave implementation

### 1.3 0.1

- Added Siemens implementation

## GENERAL INFORMATION

The FIP provides the infrastructure necessary to test interoperability in EcoGrid 2.0. It enables aggregators to talk to DERs using the EcoGrid 2.0 interoperability protocol. This document describes version 0.3 of the FIP.

### 2.1 Platform overview

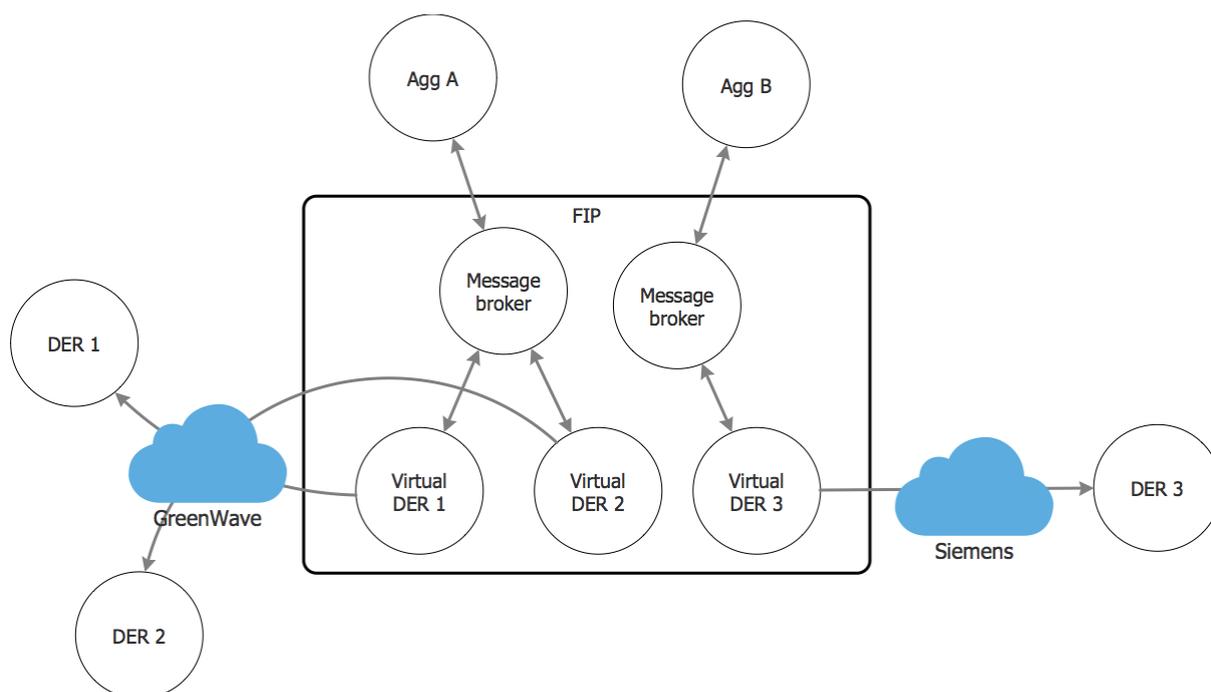


Fig. 1: Overview of the platform

The FIP provides two distinct features: A message broker (MQTT) for each aggregator, and a Virtual DER (VDER) for each physical DER. The point of the VDER is three-fold:

1. Bridge the connection between the MQTT broker and the manufacturer-specific cloud-based communication systems.

2. Translate messages from the interoperability protocol to the native protocol of each device.
3. Add any necessary control logic on top of the native controller in each device.

## 2.2 Connecting to the platform

Aggregators will interact with their DERs through the MQTT broker. The IBM broker is available at `fip.ecogrid.io:8883` while the Inero broker can be reached at `fip.ecogrid.io:8884`.

When connecting to the MQTT broker it is necessary to specify a username and password. Each aggregator should have received a username and password.

## 2.3 Mapping DERs to aggregators

Tracking which DERs are assigned to which aggregators is handled by the WP6 web portal. The table `eg_ecosphere_installations` has a row for each DER, and each row has `aggregator_id` and `dso_install_nbr` columns. The DSO number is used as `device_id`, which must be known to communicate with the DERs. This is described in *Topic structure*.

## 2.4 Controlling the DERs

While the VDERs ensure that the aggregators can control all devices with the same protocol, the actual capabilities of the device types differ, and they have different control characteristics. For this reason, the FIP exposes three different control interfaces:

- *Greenwave Native Control Interface* (GNCI)
- *Siemens Native Control Interface* (SNCI)
- *Siemens Simple Control Interface* (SSCI)

The native control interfaces expose a control interface that matches the capabilities of the device very closely. The SSCI is very similar to the GNCI, and is provided to make it easier for aggregators to control both device types in a uniform way.

## PROTOCOL

The EcoGrid 2.0 interoperability protocol is based on use cases outlined in [BC01 Contracting Flexibility Representation](#) and [BC04 Interoperability and Control of Flexibility during operation](#). The information model is based on [Woopsa](#), with some changes to accommodate bidirectional, asynchronous communication. It uses MQTT for message transport.

### 3.1 General

#### 3.1.1 Topic structure

All topics are structured according to the formula: `{device_id}/{service}/{channel}`. `device_id` uniquely identifies a DER while `service` references a service from BC01 or BC04 (see [Services](#) below). Each step of the service message flow has its own `channel`, which is unidirectional.

#### 3.1.2 Message format

The payload of each MQTT message must contain a JSON message with a header and body element. The header has 4 required fields. The body content varies: it contains parameters, and will be null if no parameters are needed. An example message:

```
{
  "Header": {
    "Verb": "meta",
    "CorrelationId": "db0fb1f-2934-4848-98e4-a1d55b1fad8",
    "SentAt": "2017-08-24T18:49:26",
    "Ttl": 300
  },
  "Body": {
    "Path": "/"
  }
}
```

- Verb: (JSON string) the Woopsa verb.

- **CorrelationId:** (JSON string) a UUID used to correlate messages belonging to the same service message flow.
- **SentAt:** (JSON string) a UTC timestamp in ISO8601 format, indicating when the message was sent.
- **Ttl:** (JSON number) an integer of seconds, indicating message time to live (after which the message will be discarded).

---

**Note:** The fields `CorrelationId`, `SentAt` and `Ttl` are left out for brevity in the examples below. They must always be present in the header.

---

### 3.1.3 Verbs

**meta** Used to describe an object (see *Object model* below). Use this to traverse the object tree of a DER.

**read** Used to read the value of a property.

**write** Used to write a value to a property.

### 3.1.4 Object model

The internals of a DER is mapped to an object model. The objects are arranged in a hierarchical tree, and the top of the tree is called “root” or “the root object”.

Each object in the tree has a name and a path associated with it. The path of the root object is `/`. If the root object has an inner object named `Zones`, the path of `Zones` is `/Zones`.

In addition to inner objects, an object may have properties. The path to a property of an object is constructed the same way as the path to an inner object. The path to the property `Comfort` on an object with path `/Zones/Room1/Setpoints` is `/Zones/Room1/Setpoints/Comfort`.

Properties have a name and value type and may be read-only. These attributes are displayed when traversing the object tree.

### 3.1.5 Value types

**Null** Used to represent absence data — for example, if the body of a message is empty. Serialized as JSON null.

**Logical** Boolean true/false value. Serialized as JSON boolean.

**Real** Real number, floating-point representation. Serialized as JSON number.

**Integer** 64-bit integer value. Serialized as JSON number.

**Text** A chain of characters representing a string. Serialized as JSON string.

### 3.1.6 Errors

If an error happens after receiving a message, the FIP will reply with an error message, which is formatted as a **Woopsa error**. Example:

```
{
  "Error": true,
  "Message": string,
  "Type": string
}
```

At the moment, there are three types of errors; `WoopsaException`, `WoopsaNotFoundException` and `WoopsaInvalidOperationException`.

## 3.2 Services

In order to enable the message flows in BC01 and BC04, the interoperability platform exposes five different services: Information, Representation, Control, Coordination and State. Currently, Control and State are implemented.

### 3.2.1 State

Described in “State information exchange triggered by aggregator request” and “State information exchanged triggered by DER event” of BC04.

Allowed verbs: `meta` and `read`.

**device\_id/state/request** AGG to DER. Used to request information about the current state of the device. Example:

```
{
  "Header": {
    "Verb": "read"
  },
  "Body": {
    "Path": "/Zones/Bad/Operating mode selected"
  }
}
```

**device\_id/state/message** DER to AGG. Used to send state information from device to aggregator. Example:

```
{
  "Header": {
    "Verb": "meta",
  },
  "Body": {
```

(continues on next page)

(continued from previous page)

```

    "Name": "/",
    "Items": [
      "Zones",
      "Heat source"
    ],
    "Properties": []
  }
}

```

### 3.2.2 Control

Described in “Real-time control” of BC04.

Allowed verbs: write.

**device\_id/control/message** AGG to DER. Used by aggregator to send a real-time control command. Example:

```

{
  "Header": {
    "Verb": "write"
  },
  "Body": {
    "Path": "/Zones/Bad/Setpoints/Comfort",
    "Value": 22.5
  }
}

```

**device\_id/control/receipt** DER to AGG. Sent immediately after receiving a control message. Example:

```

{
  "Header": {
    "Verb": "write"
  },
  "Body": null
}

```

**device\_id/control/report** DER to AGG. Sent after executing the write operation. Example:

```

{
  "Header": {
    "Verb": "write"
  },
  "Body": {
    "Type": "Real",
    "Value": 22.5,

```

(continues on next page)

(continued from previous page)

```
"TimeStamp": "2017-08-24T18:49:27"  
}  
}
```

## GREENWAVE NATIVE CONTROL INTERFACE

The Greenwave devices provide a very simple tool for performing control. In houses with Greenwave equipment, a single relay is connected to a central heat source (either heat pump or heat panel). This relay can be switched on or off remotely, and is used to throttle the heat source. The heat source can be disabled by throttling it. When the heat source is not throttled it regulates itself. This means the Greenwave system can only control whether or not the heat source is allowed to heat.

Throttling of the heat source is controlled through the `Operating mode` property on the `HeatSource` object. Setting `Operating mode` to `Auto` unthrottles the heat source, while setting it to `Off` will throttle the heat source.

### 4.1 Object tree

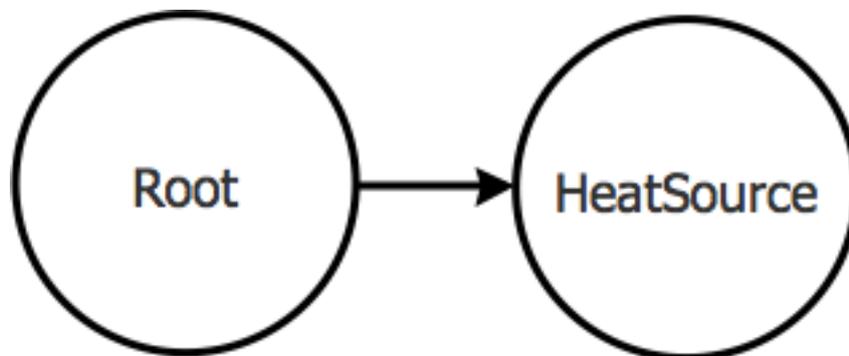


Fig. 1: GNCI Object Tree

The root object has a single inner `HeatSource` object.

#### 4.1.1 HeatSource

Changed in version 0.3: Replaced `Relay state` property with `Operating mode`

**Path:** `/HeatSource`

**Properties**

**Operating mode** controls state of the relay, either “Auto” or “Off”

## SIEMENS NATIVE CONTROL INTERFACE

The Siemens installations consist of a webserver (OZW772.01), central apartment unit (QAX913) and a heating circuit controller (RRV912 or RRV918). In addition, each room has a room adjustment unit (QAW910). The devices communicate through wireless KNX.

All Siemens households are heated with electric heating panels. The output of each heating circuit controller is connected to a contactor, and the contactor controls the state of the heating panel (on/off). The output of the heating circuit controller can be between 0% (off) and 100% (on). It is uncertain at which percentage the contactor changes state.

It is not possible to turn the electric heating panels on or off, nor to check whether the panels are on or off. The only indication is the output of the heating circuit controller. This output is referred to as “valve position” in the central apartment unit.

The heat panels in Siemens installations are on a per-room basis. It is possible to select one of five operating modes: “Auto”, “Comfort”, “Precomfort”, “Economy” or “Protection”. If “Auto” is selected, the current operating mode depends on an internal calendar schedule. Selecting any of the other modes will make that mode the current operating mode, with one exception: If the installation is in “Absent” mode, the current operating mode will depend on which operating mode has been selected for “Absent” mode.

Each operating mode has an associated setpoint. The operating mode setpoints must be between 5 and 35 degrees.

The actual setpoint is a sum of the setpoint of the current operating mode, and the room adjustment setpoint. The room adjustment setpoint can be between -3 and 3 degrees.

### 5.1 Object tree

Below the root object is a Zones object, which has an inner Zone object for each separately controlled heat panel in the house (also called a room). These inner Zone objects are named according to the room, e.g. “Bathroom”. Below each Zone object is a Setpoints object.

#### 5.1.1 Zone

**Path:** /Zones/[Zone name]

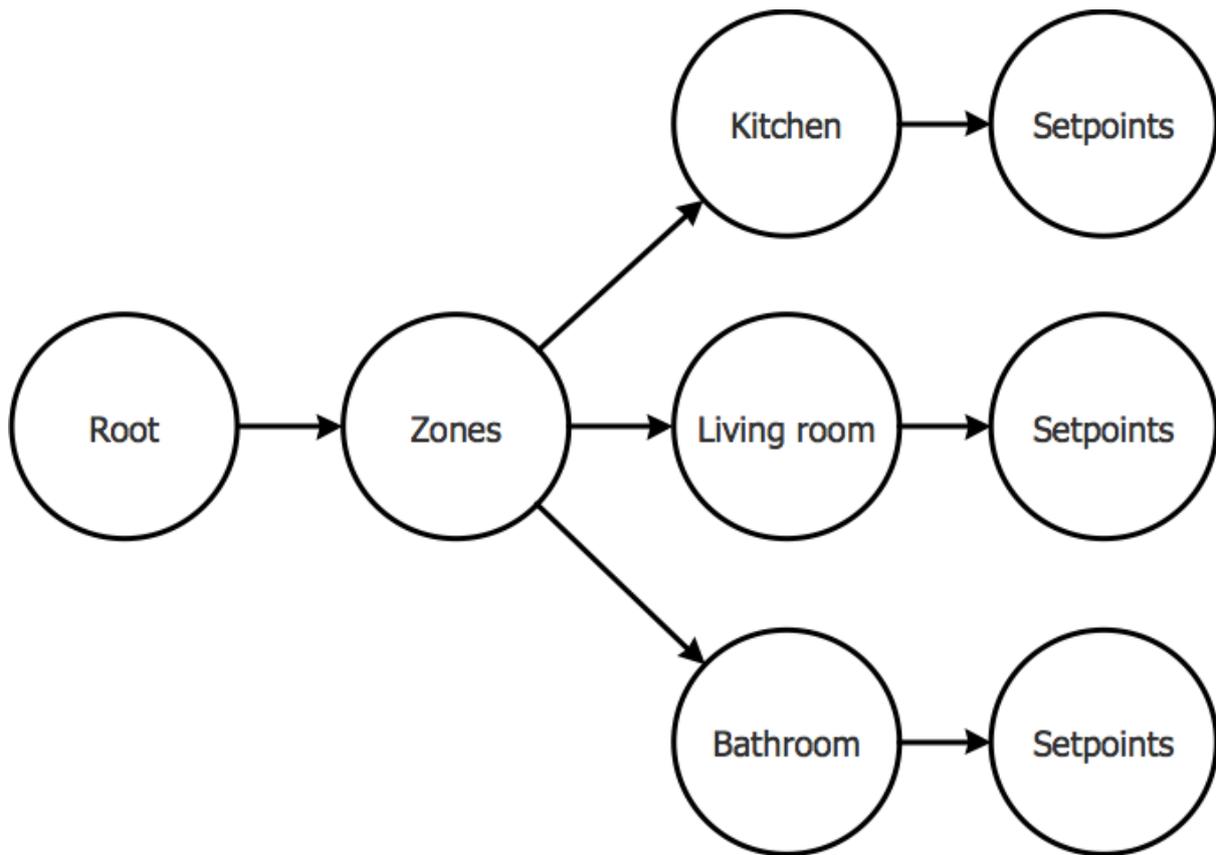


Fig. 1: Siemens Object Tree

## Properties

- Operating mode selected** the operating mode that has been manually selected
- Operating mode absence** the operating mode for the zone if the installation is in Absent mode
- Operating mode cause (ReadOnly)** the reason for the current operating mode
- Current operating mode (ReadOnly)** the current operating mode
- Actual temperature (ReadOnly)** the temperature as reported from the room adjustment unit
- Actual setpoint (ReadOnly)** the actual setpoint that the room adjustment unit is using for control
- Valve position (ReadOnly)** a measure of the state of the contactor

### 5.1.2 Setpoints

**Path:** /Zones/[Zone name]/Setpoints

## Properties

**Comfort** the setpoint temperature for Comfort operating mode (must be between Precomfort and 35)

**Precomfort** the setpoint temperature for Precomfort operating mode (must be between Economy and Comfort)

**Economy** the setpoint temperature for Economy operating mode (must be between Protection and Precomfort)

**Protection** the setpoint temperature for Protection operating mode (must be between 5 and Economy)

**Adjustment** the setpoint temperature for the room adjustment unit (must be between -3 and 3)

## **SIEMENS SIMPLE CONTROL INTERFACE**

The SSCI is almost identical to the GNCI. It allows an aggregator to treat the Siemens system of a house as a single, virtual heat source, through the `HeatSource` object. As with the GNCI, this `HeatSource` exposes the `Operating mode` property, but here the `Operating mode` has three different values: `On`, `Auto` and `Off`.

`Auto` represents normal state. When the operating mode is changed to either `On` or `Off`, the FIP will change the actual setpoint of every room to either 35 or 5 degrees, respectively. Changing the operating mode back to `Auto` will restore the setpoints back to the values they had when the operating mode was last changed away from `Auto`.

Because each change of `HeatSource` operating mode requires multiple calls to the Siemens equipment (two calls per room), this interface faces an increased risk of failure. This is handled in the following way:

1. When a control signal to change `HeatSource` operating mode to `Off` is received, the VDER enters a transaction. The VDER will not respond to any other signals until the transaction has either been committed or rolled back.
2. If the current operating mode is `Auto`, the VDER first snapshots the current device settings.
3. The VDER then starts changing the setpoint of `Protection` to 5 degrees for each room.
  - If any call fails, it will be retried 3 times, with exponential backoff and jitter, for up to a minute. If no retry succeeds, the VDER will report failure to the aggregator, and the VDER will start to roll the transaction back.
4. When all setpoints have been changed, the VDER starts changing the operating mode of each room to `Protection`.
  - Calls will be retried as above.
5. When all operating modes have been changed, the VDER will report success to the aggregator, and the VDER commits the transaction.

If the VDER enters transaction rollback, it will start changing any changed settings back to the snapshot value (`Auto`). These calls may fail as well, but will be retried an infinite amount of times, with exponential backoff and jitter. The backoff will have an upper bound of 10 minutes. In addition, the FIP operators will be notified of the failure. The operators may decide to handle the situation manually, and they can cancel the transaction rollback.

Once transaction rollback completes, the VDER clears the transaction, and is ready to respond to signals again. If the rollback is resolved manually, the operators must also clear the VDER transaction manually.

The same procedure applies when changing the operating mode to `On` and `Auto`.

---

**Note:** When changing from `On` or `Off` to `Auto`, an eventual rollback procedure will always attempt to restore settings from the last `Auto` snapshot, instead of the previous operating mode. This is because `Auto` is the only “safe” mode of operation.

---

## 6.1 Object tree

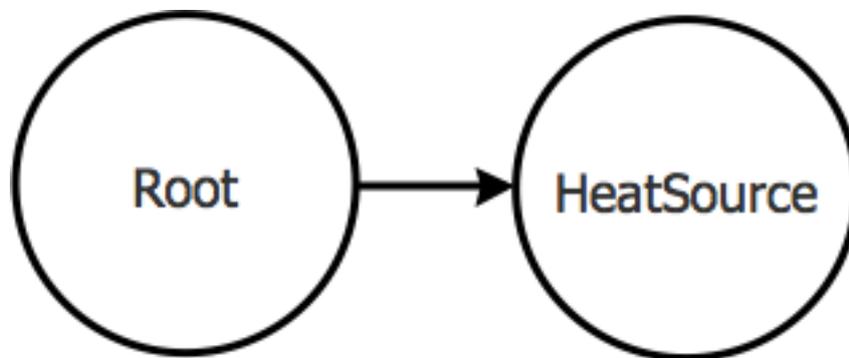


Fig. 1: SSCI Object Tree

The root object has a single inner `HeatSource` object.

### 6.1.1 HeatSource

New in version 0.3.

**Path:** `/HeatSource`

#### Properties

**Operating mode** controls setpoints of all rooms in the house. Either “On”, “Auto” or “Off”

## GLOSSARY

**AGG** aggregator

**DER** distributed energy resource

**GNCI** Greenwave Native Control Interface

**SNCI** Siemens Native Control Interface

**SSCI** Siemens Simple Control Interface

**VDER** virtual distributed energy resource, FIP-specific abstraction

# INDEX

## A

AGG, [17](#)

## D

DER, [17](#)

## G

GNCI, [17](#)

## S

SNCI, [17](#)

SSCI, [17](#)

## V

VDER, [17](#)