SEMP

Simple Energy Management Protocol

Specification

Specification | Edition 1.0.6 | SEMP-11:ZE3315

Revision History

History		Comment
1.0.0	2013-03-13	• First edition
1.0.1	2013-04-30	• XML namespace adjusted
1.0.2	2014-01-08	• Translated into English • LOCATION field cannot be omitted anymore for SSDP NOTIFY or M-Search Response → UPnP Device Description must be provided • Recommended base-path shortened to "/semp" • Sending DeviceControl message via HTTP GET removed • XML namespace adjusted • TimeframeId and LatestStart removed • semp:wsVersion added
1.0.3	2014-04-09	• Removed unsupported Timeframe.MaxPowerConsumption element from example
1.0.4	2014-11-24	• Example updated
1.0.5	2015-07-20	• Explanation of the SEMP device ID format added
1.0.6	2015-08-14	• Merged generic device documentation

Explanation of Symbols used in this Document

To enable optimal usage of this manual and safe operation of the device during installation, operation and maintenance routines, please note the following description of symbols:

➡ This symbol indicates information that is required for the optimal operation of the product. Read these sections carefully in order to ensure an optimal operation of the product and all its features.

⚠ This symbol indicates information that is essential for a trouble-free and safe operation of the product. Please read these sections carefully in order to avoid any damages of the equipment and for optimal personal protection.

📋 This symbol indicates an example.

Content

1 Overview 5

1.1 Scope 5

1.2 Protocol overview 6

1.3 Abbreviations 7

2 Addressing 8

3 Discovery (SSDP) 8

3.1 Advertisement 9

3.1.1 Registration 9

3.1.2 Deregistration 11

3.2 Search 11

3.2.1 M-Search Request 11

3.2.2 UPnP Device-Description 14

4 SEMP web service 17

4.1 Requests and data structures 19

4.2 Polling behavior of the EM 22

4.3 Device IDs 22

4.3.1 Device ID components 23

4.3.2 Representation of a device ID 25

4.4 Information delivered by a SEMP Gateway 26

4.4.1 Device information 27

4.4.2 Device status 32

4.4.3 Planning requests 36

4.5 Recommendations from the EM 43

5 Examples: 46

6 Referenced documents 48

1 Overview

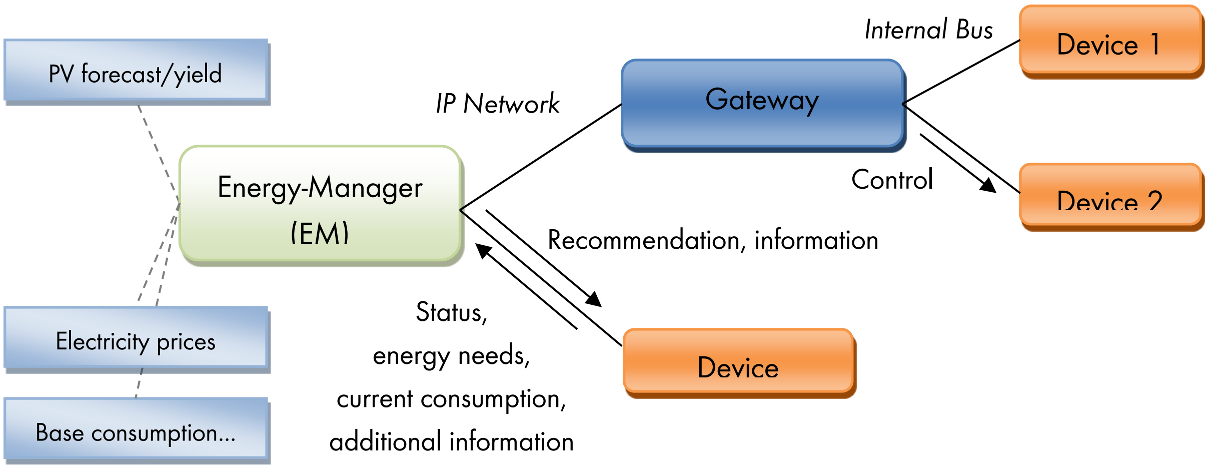
1.1 Scope

The Simple Energy Management Protocol (SEMP) describes the communication between an Energy-Manager (EM) and one or more energy management capable endpoint devices (gateways). The gateway informs the EM about current energy needs and receives recommendations for optimal switch-on times and power levels from the EM.

The device communicating with the EM does not necessarily have to be the energy managed device itself. Instead in many cases the EM communicates with a gateway, i.e. a controller or bus adapter that passes the recommendations of the EM to the controllable device or directly controls it.

For instance the gateway can be a control interface for washing machines, dryers or heat pumps. The devices managed by a gateway usually are connected to it by an internal bus like CAN that the EM cannot access directly. Another scenario would consist of a charging station for Electric Vehicles that manages multiple vehicles and accepts charge recommendations for them.

Keep in mind that if a device is connected to the EM via a gateway, the EM will only communicate to the gateway – not the device. If a message from the EM is targeted to a specific device, it will be addressed by a device ID provided in the message. The message is sent from the EM to the gateway and must be dispatched from the gateway to the target device.



As controllable devices are usually connected to a SEMP gateway, only this case will be considered in this specification. Direct communication between an EM and a controllable device equals the communication with a gateway that manages only one device, as the device additionally implements the functionality of a gateway.

1.2 Protocol overview

SEMP consists of two components:

- **Discovery:**
  UPnP SSDP based service to discover SEMP gateways in the local network
- **Webservice:**
  HTTP based service to exchange information between EM and SEMP gateway (status, energy-needs, ...)

Both of the components are based on the IP protocol. Figure 1.1 shows the components of the communication stack.

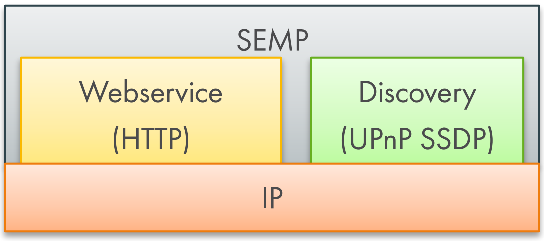


Figure 1.1SEMP Protocol stack

The basic addressing is explained in chapter 2, the discovery component in chapter 3 and the webservice in chapter 4.

1.3 Abbreviations

EM	(SEMP) Energy Manager
SHM	SMA Sunny Home Manager. SMA's EM which acts as a SEMP EM.
Sunny Portal	SMA visualization and configuration website for the Sunny Home Manager. Link: https://www.sunnyportal.com
(SEMP) Gateway	The entity the SEMP EM communicates with. Either the controllable device itself or a gateway that directly controls the device or passes recommendations to it. A gateway can manage multiple devices.
(SEMP) Device	A device controllable by a SEMP EM. A device does not have to communicate via SEMP directly. It is controlled by a Gateway implementing the SEMP protocol. It is possible that the device itself is a SEMP gateway (for this, it must be able to connect to an IP-based network, e.g. via a LAN interface).
GUI	Graphical User Interface. A device's configuration or visualization interface. In the case of the SHM the SMA Sunny Portal.
baseURL	SEMP base URL used as a prefix for requesting SEMP resources. It is specified by the SEMP gateway during discovery (via UPnP SSDP).

2 Addressing

SEMP is based on the IP protocol. Therefore EM and gateway must have a valid IP-address. How the IP addresses are assigned (DHCP, manual configuration ...) to the devices is not defined by this specification.

Only IPv4 is supported at the moment.

3 Discovery (SSDP)

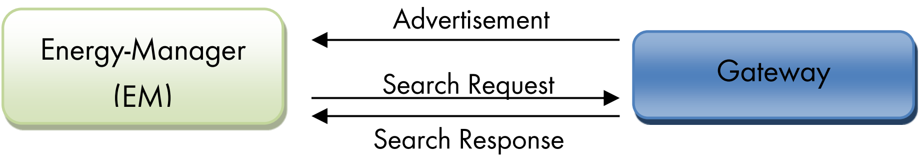
Before any communication between an EM and a gateway is possible, both entities have to know about each other. This is the task of a discovery-mechanism.

SEMP uses SSDP (Simple Service Discovery Protocol) which is specified in UPnP 1.0 (UPnP Forum, 2008) for discovery. See the UPnP specification for the general description.

UPnP (Universal Plug and Play) is a widespread standard and used to discover and control audio/video devices, routers, printers or house automation appliances over IP-based networks.

The SSDP component of UPnP specifies the discovery mechanism, consisting of the advertisement of devices and services of a specific type as well as a mechanism to search devices and services (Search Request/Response).

As communication in SEMP is unidirectional (a data transfer is always started by the EM), only the gateway has to send advertisement messages and the EM has to search for such a gateways. According to the UPnP spec the gateway corresponds to a Controlled Device and the EM to a Control Point.



The following sections shortly describe how to use SSDP for SEMP. It is not the intent of this document to specify SSDP. See the specification of UPnP 1.0 (UPnP Forum, 2008) as reference for SSDP instead.

3.1 Advertisement

Whenever an UPnP Controlled Device (SEMP gateway) is connected to or before it is disconnected from a network according to SSDP it has to notify other UPnP devices (notably the SEMP EM) of its availability. In SSDP this is referred to as advertisement. These advertisements have to be resend in regular intervals.

3.1.1 Registration

On connection a UDP datagram with a NOTIFY message of the type „ssdp:alive“ has to be sent to port 1900 of the SSDP multicast address (IPv4: 239.255.255.250).

According to SSDP a Controlled Device (here: SEMP gateway) has to send one NOTIFY message to announce the so-called root-device (the device itself as it might consist of sub-devices), one to announce its device ID and one message for every implemented device type and service. For example if a device implements a device type, it has to send at least three NOTIFY messages.

The format of a NOTIFY message reads as follows:

```
NOTIFY * HTTP/1.1\r\n
HOST: 239.255.255.250:1900\r\n
CACHE-CONTROL: max-age = <seconds until advertisement expires>\r\n
LOCATION: <URL for UPnP description for root device>\r\n
NT: <search target>\r\n
NTS: ssdp:alive\r\n
SERVER: <OS>/<version> UPnP/1.0 <product>/<version>\r\n
USN: <advertisement UUID>\r\n
\r\n
```

Field	Description
HOST	SSDP multicast address and port (IPv4: 239.255.255.250:1900)
CACHE-CONTROL	Validity period (in seconds) of the message. If no advertisement is received during this time, it is assumed that the device is no longer available. The UPnP spec recommends a value of 1800 or above.
LOCATION	URL of the UPnP description for the device. *Must use an IP-Address as DNS name resolution might not be supported.*
NT (Notification Type)	Search-Target (a device ID, device type or service). As the SEMP device types are not standardized by the UPnP Forum the format is: „`urn:<domain-name>:device:<deviceType>:<v>`“ <v> specifies the version of the device type specification.
NTS	Notification Sub Type: here „`ssdp:alive`“
SERVER	Operating System and type of the SSDP server
USN	Unique ID of the device + „::“ + NT. As the SEMP device types are not standardized by

	the UPnP Forum the format is: „uuid:<device-UUID>::urn:<domain-name>:device:<deviceType>:<v>”

UPnP 1.0 (UPnP Forum, 2008) does not specify a format for UUIDs. Nevertheless UUIDs should be generated according to the procedure recommended in UPnP 1.1 (UPnP Forum, 2008). The procedure described in (The Open group, 1997) generates a UUID based on the device’s MAC address. A resulting UUID might look like this: “2fac1234-31f8-11b4-a222-08002b34c003”.

Depending on the validity period (defined by CACHE-CONTROL) the NOTIFY messages must be resent by the SEMP gateway on a regular base. For example if the gateway selects a validity period of 30 minutes (1800 seconds), it should resent the messages every 15 minutes. If no message is received during the validity period the EM assumes that the gateway was disconnected from the network (timeout).

The three NOTIFY messages of a SEMP gateway should be similar to the following messages:

1. NOTIFY for the root-device:

```
NOTIFY * HTTP/1.1\r\n
HOST: 239.255.255.250:1900\r\n
CACHE-CONTROL: max-age = 1800\r\n
SERVER: Linux/2.6.32 UPnP/1.0 SomeGateway SSDP Server/1.0.0\r\n
NTS: ssdp:alive\r\n
LOCATION: http://192.168.1.1:8080/SEMP/description.xml\r\n
NT: upnp:rootdevice\r\n
USN: uuid:2fac1234-31f8-11b4-a222-08002b34c003::upnp:rootdevice\r\n
\r\n
```

2. NOTIFY for the device’s UUID:

```
NOTIFY * HTTP/1.1\r\n
HOST: 239.255.255.250:1900\r\n
CACHE-CONTROL: max-age = 1800\r\n
SERVER: Linux/2.6.32 UPnP/1.0 SomeGateway SSDP Server/1.0.0\r\n
NTS: ssdp:alive\r\n
LOCATION: http://192.168.1.1:8080/SEMP/description.xml\r\n
NT: uuid:2fac1234-31f8-11b4-a222-08002b34c003\r\n
USN: uuid:2fac1234-31f8-11b4-a222-08002b34c003\r\n
\r\n
```

3. NOTIFY for SEMP gateway device type:

```
NOTIFY * HTTP/1.1\r\n
HOST: 239.255.255.250:1900\r\n
CACHE-CONTROL: max-age = 1800\r\n
SERVER: Linux/2.6.32 UPnP/1.0 SomeGateway SSDP Server/1.0.0\r\n
NTS: ssdp:alive\r\n
LOCATION: http://192.168.1.1:8080/SEMP/description.xml\r\n
NT: urn:schemas-simple-energy-management-protocol:device:Gateway:1\r\n
USN: uuid:2fac1234-31f8-11b4-a222-08002b34c003::urn:schemas-simple-energy-
management-protocol:device:Gateway:1\r\n
\r\n
```

A SEMP EM will discover gateways with the following device type set as Search-Target in the NT field:

```
urn:schemas-simple-energy-management-protocol:device:Gateway:1
```

If an EM receives such a NOTIFY message it will fetch the URL given in the LOCATION field. The URL has to link to an XML document called Device Description which contains information on the device itself and the services implemented by it.

The EM will use the UUID given in the USN field to identify the gateway.

3.1.2 Deregistration

A gateway should notify other devices before it disconnects from a network. This is only possible if the gateway triggered the disconnection itself, e.g. when the gateway or its network interface is shut-down.
In this case a UDP datagram with a NOTIFY message has to be sent to the SSDP multicast address at port 1900. The NTS field has to be set to „ssdp:byebye".

```
NOTIFY * HTTP/1.1\r\n
HOST: 239.255.255.250:1900\r\n
NT: <search target>\r\n
NTS: ssdp:byebye\r\n
USN: uuid:<advertisement UUID>\r\n
\r\n
```

For each search target announced in the registration process, one NOTIFY messages must be sent during deregistration. One for the root-device, one for the UUID and one for every device type and service implemented. Usually three messages (root-device, UUID, 1x device type) have to be sent.

3.2 Search

One way to detect devices is to passively listen to the multicast advertisement messages (NOTIFY) in the network. The interval between consecutive NOTIFY messages of a device is usually several minutes. If a gateway connects to a network after a device's NOTIFY message was sent, the gateway might have to wait a whole interval until it is able detect the device. For that reason an active search can be triggered by the gateway - for example on startup of the gateway. In UPnP this is done by sending a multicast M-Search request message. All devices that match the search request have to reply to the sender with a unicast M-Search response message.

3.2.1 M-Search Request

An SSDP search is started by sending an M-Search request UDP datagram to port 1900 of the SSDP multicast address. The format of an M-Search request reads as follows:

```
M-SEARCH * HTTP/1.1\r\n
HOST: 239.255.255.250:1900\r\n
MAN: "ssdp:discover"\r\n
MX: <seconds to delay response>\r\n
ST: <search target>\r\n
\r\n
```

Field	Description
HOST	SSDP multicast address and port (for IPv4 239.255.255.250:1900)
MAN	Must be set to „ssdp:discover“
MX	Maximum time in seconds until a reply must be delivered. The UPnP-specification recommends a value in the range between 1 and 120.
ST	Search target (a device ID, device type or service name).

The Search-Target (ST) determines what kind of devices should reply to the search request. A SEMP EM usually searches for devices implementing the SEMP Gateway device type.

Search by device type

For device types that are not specified by the UPnP Forum (as it is the case for the SEMP device types) the format of the Search-Target is:

```
urn:<domain-name>:device:<deviceType>:<v>
```

In SEMP a Gateway implements the following device type an EM searches for:

```
urn:schemas-simple-energy-management-protocol:device:Gateway:1
```

With this an M-Search request of an EM for SEMP Gateways will look like this:

```
M-SEARCH * HTTP/1.1\r\n
HOST: 239.255.255.250:1900\r\n
MAN: "ssdp:discover"\r\n
MX: 10\r\n
ST: urn:schemas-simple-energy-management-protocol:device:Gateway:1\r\n
\r\n
```

The value of MX can be selected by the EM according to constraints given above.

Search for all SSDP enabled devices

Additionally to the device type SSDP also defined a generic search target „ssdp:all“ („`ST: ssdp:all`“). All root-devices, device types, device IDs and services will be delivered by this. A device answers to such a request by returning an M-Search response for each search target announced during the registration process. A SEMP gateway should respond to the “ssdp:all” search target.

Search by UUID

Besides searches for a device type or "ssdp:all" a third possibility is to search for devices by their UUID („`ST: uuid:<device-UUID>`").

M-Search Response

Responses to M-Search requests are similar to NOTIFY message. An M-Search response is sent in a unicast UDP datagram to the device that started the search – in SEMP usually the EM. The message will be addressed to the source IP-address and port of the request message.

```
HTTP/1.1 200 OK\r\n
CACHE-CONTROL: max-age = <seconds until advertisement expires>\r\n
DATE: <when response was generated>\r\n
EXT:
LOCATION: <URL for UPnP description for root device>\r\n
SERVER: <OS>/<version> UPnP/1.0 <product>/<version>\r\n
ST: <search target>\r\n
USN: <advertisement UUID>\r\n
\r\n
```

The fields are similar to that of the NOTIFY messages with subtype „ssdp:alive". The NT field is replaced by the field ST. Except for the Search-Target „`ssdp:all`" the value used for the NT field in the response equals the value of the ST field of the request. For „`ssdp:all`" the value of the NT field is set to the corresponding root-device, device type, device id or service (note that multiple responses have to be sent here).
The optional field „DATE" – if used – has to be set to the date and time of the generation of the response message. The format is defined in RFC 2616.
The UUID of the USN field uniquely identifies the device (SEMP gateway).

The response to an M-Search Request with SEMP Gateway Search-Target will look like this:

```
HTTP/1.1 200 OK\r\n
CACHE-CONTROL: max-age = 1800\r\n
DATE: Mon Mar 11 11:50:15 2013\r\n
EXT: \r\n
LOCATION: http://192.168.1.1:8080/uuid:2fac1234-31f8-11b4-a222-
08002b34c003/description.xml\r\n
SERVER: Linux/2.6.32 UPnP/1.0 SomeGateway SSDP Server/1.0.0\r\n
ST: urn:schemas-simple-energy-management-protocol:device:Gateway:1\r\n
USN: uuid:2fac1234-31f8-11b4-a222-08002b34c003::urn:schemas-simple-energy-
management-protocol:device:Gateway:1\r\n
\r\n
```

If the EM receives such a message from a SEMP Gateway, it will fetch the URL of the LOCATION field which contains further information on the device and the SEMP protocol. According to the UPnP spec the

LOCATION field is mandatory. The URL has to point to a valid UPnP-Device-Description. The format of a Device-Description is defined in the UPnP spec.

3.2.2 UPnP Device-Description

SSDP NOTIFY messages with type "ssdp:alive" as well as M-Search response messages contain a LOCATION field which links to a UPnP Device-Description. The Device-Description contains a general description of the device and the device types and services implemented by it.

A Device-Description is an XML document described by the UPnP spec. The following example gives an overview of the mandatory (**bold**) and optional (*italic*) elements. Note that the semp:X_SEMPSERVICE element is not described by UPnP. It is a SEMP specific extension that describes the SEMP web service implemented by a gateway.

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <device>
    <deviceType>urn:domain-name:device:deviceType:ver</deviceType>
    <friendlyName>short user-friendly title</friendlyName>
    <manufacturer>manufacturer name</manufacturer>
    <modelName>model name</modelName>
    <UDN>uuid:UUID</UDN>
    <serviceList>
      <service>
        <serviceType>urn:domain-name:service:serviceType:v</serviceType>
        <serviceId>urn:domain-name:serviceId:serviceID</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
    </serviceList>
    <presentationURL>URL for presentation</presentationURL>
    <semp:X_SEMPSERVICE
         xmlns:semp="urn:schemas-simple-energy-management-protocol:service-1-0">
      <semp:server>http://server-addr[:port]</semp:server>
      <semp:basePath>Base-Path for requests, e.g. /SEMP</semp:basePath>
      <semp:transport>HTTP/Pull</semp:transport>
      <semp:exchangeFormat>XML</semp:exchangeFormat>
      <semp:wsVersion>x.y.z</semp:wsVersion>
    </semp:X_SEMPSERVICE>
  </device>
</root>
```

The element "deviceType" contains the device type corresponding to the search target of the NOTIFY or M-Search response. For SEMP Gateways the SEMP device type is used (see sections on Advertisement).

The element "UDN" contains the UUID equal to the first part (i.e. without the search target delimited by "::") of the USN field of NOTIFY and M-Search messages.

The element "presentationURL" can be used to specify a URL to a device's presentation, control or configuration interface. Some Operating Systems already use this information. On Windows a user can easily open a device's web interface by clicking on the device symbol in the network environment explorer.

Service List

The service list (element "serviceList") contains all UPnP services the device implements. UPnP services are based on a special UPnP specific RPC mechanism. The device type determines which services have to be implemented. For example a device with type „urn:schemas-upnp-org:device:InternetGatewayDevice" has to implement a service to grant internet access to a device.
The SEMP web service does not use the UPnP RPC mechanism. As a result it is not listed in the service list. No UPnP service has to be implemented for the SEMP gateway device type.

According to the UPnP 1.0 spec the "serviceList" element can be empty or omitted. However some Control-Points do not comply with this, e.g. Windows XP is not able to handle devices without services correctly. As a result a NULL-service (a dummy service) should be specified.

SEMP web service configuration

The element "semp:X_SEMPSERVICE" is an extension of the UPnP Device-Description specific to SEMP. It contains information on how to establish communication via the SEMP web service that must be implemented by a SEMP gateway. The element must be provided by all devices of the type "schemas-simple-energy-management-protocol:device:Gateway".

- "semp:server" contains the address and port (defaults to 80 if omitted) of the HTTP server that implements the SEMP web service. The address must be given as IP address as DNS name resolution might not be supported by the EM.
- "semp:basePath" contains the prefix for SEMP request URLs.
- "semp:transport" specifies the communication mechanism. It must be set to „HTTP/Pull".
- "semp:exchangeFormat" specifies the serialization mechanism. It must be set to „XML".
- "semp:wsVersion" specifies the version of the web service and its data-structures. Use the value specified in the SEMP XSD file. The format is "<major>.<minor>.<release>", e.g. "1.1.0".

BaseURL

The concatenation of the contents "semp:server" and "semp:basePath" ("<semp:server><semp:basePath>") results in the BaseURL. It is used by the EM as a prefix for the creation of request URLs (see description of the SEMP web service).

An example UPnP Device-Description of a SEMP Gateway:

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
   <specVersion>
      <major>1</major>
      <minor>0</minor>
   </specVersion>
   <device>
      <deviceType>
         urn:schemas-simple-energy-management-protocol:device:Gateway:1
      </deviceType>
      <friendlyName>Some Gateway</friendlyName>
      <manufacturer>Some Company</manufacturer>
      <manufacturerURL>http://www.somecompany.xy</manufacturerURL>
      <modelDescription>SEMP Gateway Demo</modelDescription>
      <modelName>SEMP GWD</modelName>
      <modelNumber>1.0.0</modelNumber>
      <modelURL>http://www.somecompany.xy/modelZ</modelURL>
      <serialNumber>53-4D-41-53-4D-41</serialNumber>
      <UDN>uuid:2fac1234-31f8-11b4-a222-08002b34c003</UDN>
      <serviceList>
         <service>
            <serviceType>
               urn:schemas-simple-energy-management-protocol:service:NULL:1:service:NULL:1
            </serviceType>
            <serviceId>
               urn:schemas-simple-energy-management-protocol:serviceId:NULL:serviceId:NULL
            </serviceId>
            <SCPDURL>/XD/NULL.xml</SCPDURL>
            <controlURL>/UD/?0</controlURL>
            <eventSubURL></eventSubURL>
         </service>
      </serviceList>
      <presentationURL>index.html</presentationURL>
      <semp:X_SEMPSERVICE
           xmlns:semp="urn:schemas-simple-energy-management-protocol:service-1-0">
         <semp:server>http://192.168.1.1:8080</semp:server>
         <semp:basePath>/semp</semp:basePath>
         <semp:transport>HTTP/Pull</semp:transport>
         <semp:exchangeFormat>XML</semp:exchangeFormat>
         <semp:wsVersion>1.1.0</semp:wsVersion>
      </semp:X_SEMPSERVICE>
   </device>
</root>
```

4 SEMP web service

Once the devices discovered each other the SEMP EM and the SEMP Gateway are ready to communicate with each other. This is done by an HTTP based web service.

The resources of a SEMP gateway (status, planning requests und further information) can be requested by performing an HTTP GET request on the SEMP gateway. The SEMP web service is unidirectional; i.e. the SEMP gateway implements an HTTP server and the EM an HTTP client. Requests and data transmissions in general are always triggered by the EM. Gateways cannot request data from the EM directly.

Requests for data structures are sent to a gateway as HTTP requests using the GET method. The gateway responds to a request with an HTTP response. The requested data structures are included in the body of the HTTP response – embedded into a Device2EM XML element. Data transmissions (e.g. used for recommendations) from EM to gateway are performed with an HTTP request using the POST method. The data structures are transferred in the message body – embedded into an EM2Device XML element.

The URL of a SEMP HTTP request from an EM to a gateway always starts with a specific prefix called Base-URL. A gateway provides its Base-URL during the discovery phase via its UPnP Device-Description.

The data structures transmitted by an EM or gateway are encoded in XML. The Content-Type for HTTP-POST and HTTP-Response should be set to „application/xml". The XML structures must be encoded in UTF-8.

The XML root element for SEMP data structures sent from the EM to the gateway is always <`EM2Device`>. In the direction from gateway to EM `<Device2EM>` is used as the XML root element.

The following HTTP status codes can be used in a gateway's HTTP responses:

- 200 (OK): the request was handled by the gateway
- 404 (Not Found): the requested URL is not supported
- 400 (Bad Request): a parameter of the requested URL or the data structure in the body is invalid

Examples:

- Request via HTTP-GET:

```
GET /semp/ HTTP/1.1\r\n
Host: 192.168.1.1\r\n
\r\n
```

- Gateway response to the request above:

```
HTTP/1.1 200 OK\r\n
Content-Length: 2682\r\n
Connection: close\r\n
Content-Type: application/xml\r\n
\r\n
<?xml version="1.0" encoding="UTF-8"?>
<Device2EM xmlns="http://www.sma.de/communication/schema/SEMP/v1"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
     <!-- data structures -->
     <...></...>
</Device2EM>
```

- Request via HTTP-POST:

```
POST /semp/ HTTP/1.1\r\n
Host: 192.168.1.1\r\n
Content-Type: application/xml\r\n
Content-Length: 364\r\n
\r\n
<?xml version="1.0" encoding="UTF-8"?>
<EM2Device xmlns="http://www.sma.de/communication/schema/SEMP/v1"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
     <!-- data structures -->
     <…></…>
</EM2Device>
```

4.1 Requests and data structures

Request URLs in SEMP have the following format:

```
<baseURL>/<resource>
```

The Base URL (`<baseURL>`) is provided by the gateway in the discovery process (see section 3.2.2). The resource (`<resource>`) is the name of the requested SEMP data structure. All available data structures are defined by an ***XML Schema (XSD)***. Requests can refer to all direct child-elements of the <Device2EM>-root element. This way the XSD file not only specifies the format of the data structures but also all possible request URLs.

If no resource is selected (i.e. "<baseURL>/" is requested), the gateway will respond with the data structures for *all available elements* (see examples below). By default the gateway delivers the selected resource for *all of the devices* managed by it.

Devices managed by a SEMP Gateway have a unique ID (see section 4.3). A device's ID is provided by the corresponding DeviceInfo data structure to the EM. A device ID can be specified in the query of the EM to request only the data of a single device managed by the queried gateway. This is done with the query parameter "`DeviceId=<DeviceID>`" added to the request URL.

Data structures are sent from EM to gateway with an HTTP request using the POST method. The name of the sent data structure(s) is not reflected in the URL of the request as the data (including the element's names) is contained in the request body. As a result all data structure transmissions from EM to gateway are directed to the URL „<baseURL>/".

Data requests

Following a list of combinations of resource and device selectors which might be requested by the EM:

- Request all information (DeviceInfo, DeviceStatus, PlanningRequests of all devices):
    - HTTP GET: <baseURL>/
- Request DeviceInfo of all devices:
    - HTTP GET: <baseURL>/DeviceInfo
- Request DeviceInfo of specific device:
    - HTTP GET: <baseURL>/DeviceInfo?DeviceId=<DeviceID>
- Request DeviceStatus of all devices:
    - HTTP GET: <baseURL>/DeviceStatus
- Request DeviceStatus of specific device:
    - HTTP GET: <baseURL>/DeviceStatus?DeviceId=<DeviceID>
- Request PlanningRequest of all devices:
    - HTTP GET: <baseURL>/PlanningRequest
- Request PlanningRequest of specific device:
    - HTTP GET: <baseURL>/PlanningRequest?DeviceId=<DeviceID>

Data transmissions

Data transmissions (e.g. used for recommendations) from an EM to a Gateway:

- HTTP POST: <baseURL>/

An EM regularly polls a gateway for DeviceInfo, DeviceStatus and PlanningRequests.

Example 1: Response of a gateway to the HTTP GET request “<baseURL>/“(fetch all information)

```
<?xml version="1.0" encoding="UTF-8"?>
<Device2EM xmlns="http://www.sma.de/communication/schema/SEMP/v1">
  <DeviceInfo>
    <Identification>
      <DeviceId>F-11223344-112233445566-00</DeviceId>
      <DeviceName>Some Device</DeviceName>
      <DeviceType>HeatPump</DeviceType>
      <DeviceSerial>2332532423</DeviceSerial>
      <DeviceVendor>Some Vendor</DeviceVendor>
    </Identification>
    <Characteristics>
      <MaxPowerConsumption>1500</MaxPowerConsumption>
    </Characteristics>
    <Capabilities>
      <CurrentPower><Method>Measurement</Method></CurrentPower>
      <Timestamps><AbsoluteTimestamps>false</AbsoluteTimestamps></Timestamps>
      <Interruptions>
        <InterruptionsAllowed>true</InterruptionsAllowed>
      </Interruptions>
      <Requests><OptionalEnergy>false</OptionalEnergy></Requests>
    </Capabilities>
  </DeviceInfo>
  <DeviceStatus>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EMSignalsAccepted>true</EMSignalsAccepted>
    <Status>On</Status>
    <PowerConsumption>
      <PowerInfo>
        <AveragePower>1000</AveragePower>
        <Timestamp>0</Timestamp>
        <AveragingInterval>60</AveragingInterval>
      </PowerInfo>
    </PowerConsumption>
  </DeviceStatus>
  <PlanningRequest>
    <Timeframe>
      <DeviceId>F-11223344-112233445566-00</DeviceId>
      <!-- earliest start of the device possible in 1 hour -->
      <EarliestStart>3600</EarliestStart>
      <!-- finish in the next 3 hours -->
      <LatestEnd>10800</LatestEnd>
      <!-- runtime: 0.5h -->
      <MinRunningTime>1800</MinRunningTime>
      <MaxRunningTime>1800</MaxRunningTime>
    </Timeframe>
  </PlanningRequest>
</Device2EM>
```

Example 2: Switch-on recommendation for a device managed by the gateway:

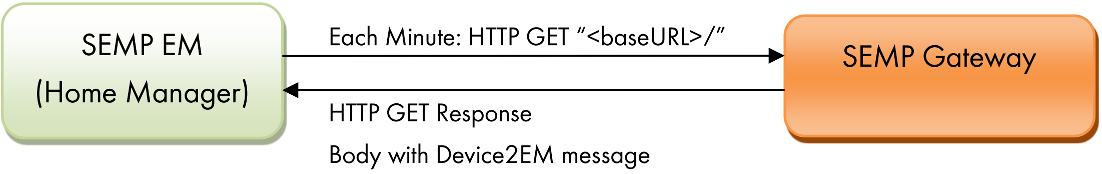
HTTP POST with: „<baseURL>/“ and body:

```
<?xml version="1.0" encoding="UTF-8"?>
<EM2Device xmlns="http://www.sma.de/communication/schema/SEMP/v1">
  <DeviceControl>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <On>true</On>
    <Timestamp>0</Timestamp><!-- Timestamp of this message -->
  </DeviceControl>
</EM2Device>
```

4.2 Polling behavior of the EM

The EM will periodically poll the SEMP gateway for new device information, device status information and planning requests.

With the current SEMP EM implementation of the Sunny Home Manager (SHM) the behavior is as follows: all of the available information of all SEMP devices connected to the gateway will be polled *once every minute* by the EM with the "<baseURL>/" URL.


SEMP EM
(Home Manager)
Each Minute: HTTP GET "<baseURL>/"
SEMP Gateway
HTTP GET Response
Body with Device2EM message

4.3 Device IDs

Each SEMP device is identified by a globally unique SEMP device ID.

SEMP devices found by an EM are usually not managed by the EM by default. The customer has to explicitly enable energy management for a SEMP device. For the SHM the SEMP device has to be assigned to the plant of the SHM via the Sunny Portal to enable energy management for the device.

It is not possible to assign one SEMP device to multiple plants. This is why every SEMP device ID has to be globally unique. If a SEMP device ID is already in use by a device, Sunny Portal rejects to add another device with the same device ID to a plant, even if the plants are different.

The device ID is represented by a 92 bit integer which consists of the following components:

Device ID (92 bits)			
Vendor (36 bits)		**Serial-number** (48 bits)	**Sub-device ID** (8 bits)
Vendor ID type (4 bits)	**Vendor ID** (32 bits)		

Do not confuse the SEMP device ID with the SEMP gateway ID. A SEMP gateway is identified by an UPnP UUID which is only used for discovery. The SEMP gateway manages one or more SEMP devices with each device having a unique SEMP device ID.

4.3.1 Device ID components

A) Vendor: the vendor part identifies the vendor of a SEMP device. The vendor part must be globally unique for each vendor and serves as an address space for vendor specific serial numbers. IANA PEN or IEEE OUI (known from MAC addresses) numbers can be used to specify the vendor.

If neither PEN nor OUI is available, the vendor ID type (0xF) can be used for demonstration models, prototypes or private self-build projects. Global uniqueness cannot be guaranteed for SEMP device IDs with this vendor ID type. ***The vendor ID type 0xF must not be used for products***.

- **A.1) Vendor ID type** (VID type): defines the type of the vendor ID. Two widespread industry standards (IANA PEN and IEEE OUI) are supported. See Table 4.1 for a description of the available types.
- **A.2) Vendor ID** (**VID**): Unique vendor ID according to the selected vendor ID type. The format of the vendor ID depends on the vendor ID type. See Table 4.1 for the formats of the supported vendor ID types.

Vendor ID type	Description	Vendor ID format
0x0	IANA PEN	IANA PEN number, padded with 0: 0xXXXXXXXX
0x1	IEEE OUI	24 bit ID, padded with 0: 0x00XXXXXX
0x2 - 0xE	Reserved for future use	
0xF	Local scope, for prototypes. **Do not use for products.**	0xXXXXXXXX

Table 4.1 Vendor ID types and formats

Following a short description of each vendor ID type:

- **IANA PEN** (Private Enterprise Number) uniquely identifies organizations and is the preferred vendor ID type in SEMP. A PEN can be requested for an organization for free from the IANA (Internet Assigned Numbers Authority). A list of already assigned numbers can be found in (IANA, 2015).

An organization with the IANA PEN number 100 000 (hex: 0x186A0) uses SEMP device IDs with vendor ID type 0x0 and 32-bit vendor ID 0x000186A0, which is combined to the SEMP device ID prefix 0-000186A0.

- **IEEE OUI** (Organizationally Unique Identifier) also uniquely identifies organizations. It is used for example in the 24-bit vendor-part of MAC-addresses assigned to network interface cards. An OUI shall only be used if the OUI is owned by the company. Hence, the MAC-address vendor-part of a SEMP device's network interface must not be used, as the SEMP vendor ID should identify the vendor of the SEMP device and not the vendor of the network card.

An organization with the 24 bit IEEE OUI 0x5CCD12 uses SEMP device IDs with vendor ID type 0x1 and the zero-padded 32-bit vendor ID 0x005CCD12, which is combined to the SEMP device ID prefix 1-005CCD12.

- **Local scope** for demonstration models, prototypes and private development projects. Vendor IDs with this type are not necessarily globally unique. The vendor ID can be selected arbitrarily but should be reused for all further prototypes and similar devices. If the SEMP device has a network interface of its own (i.e. does not share it with other devices managed by a SEMP gateway) the SEMP serial number of the device ID should be set to its 48 bit MAC-address.

A vendor ID for a private self-build project needs to be generated - no PEN or OUI is present so far. The vendor ID type is set to 0xF, the vendor ID is arbitrarily selected as 0x11223344. This is combined to the vendor part F-11223344 of the SEMP device ID.

B) Serial-number: the serial number can be freely defined by the vendor as the vendor part of the device ID defines a vendor-specific address space for serial-numbers. The serial-number must be unique for each of the vendor's devices. For SEMP device IDs using the local vendor scope (vendor type 0xF), the MAC-address of the device (if available) should be used as the SEMP serial-number to make the SEMP device IDs as unique as possible.

C) Sub-device ID: a SEMP device can consist of up to 256 sub-devices. Sub-devices can be used if a physical device consists of multiple virtual devices which should be presented as one device to the user. A device with ***sub-device ID 0x0***, i.e. the main device, ***must always be present***.

The sub-devices must be numbered (hexadecimal) in consecutive order. The main device must have sub-device ID 0x0, the first non-main sub-device must have sub-device ID 0x1, the second one ID 0x2 and so on. Do not leave gaps between the sub-device IDs.

A SEMP device consists of two heating elements. Although the EM should optimize both devices individually they should be presented as a single heating unit to the user.
The SEMP device ID of the first heating element is selected as F-11223344-112233445566-**00**, the ID of the second heating element is selected as F-11223344-112233445566-**01**. The first heating element becomes the main device as the sub-device ID 0 was assigned to it.

A SEMP device without any virtual devices is represented by a main-device. The sub-device ID assigned to it has to be 0x0, e.g. F-11223344-112233445566-**00**.

4.3.2 Representation of a device ID

The **device ID** combines the previously mentioned components. It is represented by a string of the hexadecimal coded components separated by hyphens (ASCII: 0x2D).

The string format of a device ID is as follows:

```
[VID Type:4bit]-[VID:32bit]-[Serial:48bit]-[SubDev-ID:8bit]
```

Each component is represented by hexadecimal digits ("X" stands for one 4-bit hex-digit):

```
X-XXXXXXXX-XXXXXXXXXXXX-XX
```

A company has the PEN number 100 000, so the vendor part of the device-ID is set to 0-000186A0. The vendor assigns consecutive serial numbers to each product unit, so the first product units will have the SEMP device IDs 0-000186A0-000000000000-00, 0-000186A0-000000000001-00, 0-000186A0-000000000002-00, ...

The network chip of a prototype has the MAC-address `00-80-41-ae-fd-7e`. A SEMP vendor ID 0x11223344 is arbitrarily chosen for the prototype. This results in the SEMP device ID F-11223344-008041AEFD7E-00.

4.4 Information delivered by a SEMP Gateway

Whenever the EM requests information from the gateway via “<baseURL>/” the gateway returns a Device2EM data-structure in the response body with the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<Device2EM xmlns=…>
  <DeviceInfo>…</DeviceInfo>
  <!-- further DeviceInfo elements if gateway manages multiple devices -->
  <DeviceStatus>…</DeviceStatus>
  <!-- further DeviceStatus elements if gateway manages multiple devices -->
  <PlanningRequest>…</PlanningRequest>
</Device2EM>
```

The content of each component is explained in the following sections. Note that the SEMP XML namespace has been replaced by dots here (xmlns=...) for readability. See section 4.1 for the correct namespace URL.

It is possible that one SEMP gateway handles multiple SEMP devices. As a consequence each of the three components can occur multiple times to describe each managed device separately.

At least DeviceInfo and DeviceStatus must be returned for each requested device (or all devices if “<baseURL>/” was used). The PlanningRequest element must be omitted for SEMP devices that do not have any pending energy demands.

Device List

If an EM requests the DeviceInfo data structure (“<baseURL>/” or “<baseURL>/DeviceInfo”) for all devices (i.e. the “DeviceId” parameter is not used in the query), the gateway has to list all devices currently managed by it. If a device that was announced by the gateway earlier is missing within the list, the EM assumes that the device has been disconnected and discards all plans generated for the device.

If a device is temporarily unavailable (e.g. due to a network disconnect) but its gateway is still reachable, the gateway should still provide a DeviceInfo element for the device but set the value of the status field in the corresponding DeviceStatus element to “Offline”:

```
<?xml version="1.0" encoding="UTF-8"?>
<Device2EM xmlns=…>
  <DeviceInfo>
    <Identification>
      <DeviceId>F-11223344-112233445566-00</DeviceId>…<Identification>
    …
  </DeviceInfo>
  <DeviceInfo>
    <DeviceId>F-11223344-112233445567-00</DeviceId>
    …
  </DeviceInfo>
  <DeviceStatus>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <Status>Offline</Status><!-- disconnected/unreachable -->
    …
  </DeviceStatus>
  <DeviceStatus>
    <DeviceId>F-11223344-112233445567-00</DeviceId>
    <Status>Off</Status><!-- connected, but switched off -->
    …
  </DeviceStatus>
</Device2EM>
```

4.4.1 Device information

The device information data structure contains static (i.e. non-changing) information about a device. The data should not change during runtime. It is split into three sections: “Identification”, “Characteristics” and “Capabilities”.

```
<DeviceInfo>
  <Identification>…</Identification>
  <Characteristics>…</Characteristics>
  <Capabilities>…</Capabilities>
</DeviceInfo>
```

The sub-elements Identification, Characteristics and Capabilities are described in the following sections.

Identification

The “Identification” part can be used by the EM in its GUI to help the user to identify a device. Hence ID, name, type and vendor are mandatory.

In addition to the SEMP device ID a vendor specific serial number has to be specified. The serial number helps to identify a device, as serial numbers are typically printed on the device’s case.

A DeviceURL can be specified. If present, it can be used in the GUI of the EM to provide a link to a device’s configuration interface.

Fields used for generic devices are explained below. For a full list of available fields and their datatypes see the SEMP XSD. In some specific application scenarios additional fields might be required. These exceptions to the generic usage are described in separate Application Notes (AN).

Field	**Description**
DeviceId	SEMP Device-ID of the device to which this information relates (see section 4.3).
DeviceName	Name of the device.
DeviceType	Type of device. One of the values predefined in the SEMP XSD should be used (Heater, Dryer, WashingMachine, Charger, ...).
DeviceSerial	Vendor specific serial number. This is not the serial number part of the SEMP device ID (see section 4.3). Instead use a serial number that is known by the user so that he is able to identify the device. If available use the vendor specific serial number which is printed on the device.
DeviceVendor	Name of the device's vendor.
DeviceURL (opt.)	Optional URL to configuration interface (global or local address). This can be used in the EM GUI. If possible, provide a URL which is accessible from the internet, so that the URL also works outside of the user's network (e.g. if presented on an internet based configuration website of the EM).

Example:

```
<DeviceInfo>
  <Identification>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <DeviceName>Name of the first device</DeviceName>
    <DeviceType>[Pre-defined value from SEMP XSD, e.g. Heater]</DeviceType>
    <DeviceSerial>ZYXVU342432</DeviceSerial>
    <DeviceVendor>ABC Ltd</DeviceVendor>
    <DeviceURL>http://abc.com/config?id=6767</DeviceURL>
  </Identification>
  <Characteristics>…</Characteristics>
  <Capabilities>…</Capabilities>
</DeviceInfo>
```

Characteristics

The "Characteristics" part specifies energy management related characteristics of the device that do not change during operation.

The specification of the maximum power consumption (MaxPowerConsumption) is required. The EM uses this value when it creates the first schedule for the device. The EM then learns the power consumption during each run. For the creation of the subsequent schedules, the EM uses a combination of the learned power consumption and the specified maximum power.

The value of the field MaxPowerConsumption is static and should not be changed during the device’s runtime. Changes in this field might not be evaluated by the EM.

If a device can be paused during runtime (which is called “interruption” in SEMP) the device can provide constraints to prevent inefficient use of interruptions or to prevent recommendations that could have a negative impact on the devices lifetime.

- If the device has to be left in “on” state after turning it on for a given time or if the device has to stay in “off” state after pausing it, these constraints can be defined by MinOnTime respectively MinOffTime.

  Note that the device or gateway is responsible not to accept a recommendation that could cause any damage. Even if MinOn/OffTime device characteristics were provided, the device should check that recommendations comply with the device´s constraints before applying it.

Constraints will influence the decision of the EM whether a device will be paused or not.

Fields used for generic devices are explained below. For a full list of available fields and their datatypes see the SEMP XSD. In some specific application scenarios additional fields might be required. These exceptions to the generic usage are described in separate Application Notes (AN).

Field	**Description**
MaxPowerConsumption	Maximum power consumption of the device (in W).
MinOnTime (opt.)	When switched on (or un-paused), the device has to remain on for at least the given amount of seconds.
MinOffTime (opt.)	When switched off (or paused), the device has to remain off for at least the given amount of seconds.

Example:

```
<DeviceInfo>
  <Identification>…</Identification>
  <Characteristics>
    <MaxPowerConsumption>22000</MaxPowerConsumption><!-- 22kW -->
    <MinOnTime>60</MinOnTime><!— 1 minute -->
    <MinOffTime>60</MinOffTime><!-- 1 minute -->
  </Characteristics>
  <Capabilities>…</Capabilities>
</DeviceInfo>
```

Capabilities

Defines the SEMP specific capabilities of a device.

Fields used for generic devices are explained below. For a full list of available fields and their datatypes see the SEMP XSD. In some specific application scenarios additional fields might be required. These exceptions to the generic usage are described in separate Application Notes (AN).

Field	**Description**
CurrentPower.Method	Specifies how the current power in the device status element is determined. A device that determines the current power by a power meter should select "Measurement". If it estimates the power consumption by look-up tables or other mechanisms select "Estimation". This information provides a hint about the quality of the power values provided in the device status section. It can be used to determine whether the power data can be used for learning device profiles or if it is suitable to be displayed in a GUI. As a rule of thumb, the method "Measurement" should only be used if the accuracy complies with both of the following constraints: • Relative deviation from actual value not more than 25% of the actual value • Absolute deviation not more than 100W. **For the EM to be able to provide a good energy management, it is important that the power information is reliable and accurate. For example other devices might be blocked if the device reports a higher or lower power than is actually consumed. It is strongly encouraged to provide measurements instead of estimations.**
Timestamps.AbsoluteTimestamps	SEMP supports two types of timestamps: • Relative timestamps with the number of seconds since now (might be negative if the corresponding event was in the past) • Absolute timestamps according to the UNIX timestamp format (number of seconds elapsed since 01.01.1970 00:00 UTC). The device has to decide whether it wants to send and receive absolute or relative timestamps. The EM evaluates this option to determine which format to use when sending messages to the device. Because of that the value of this option must not change for a device. Although a default value (see XSD file) is used if not specified, this option should always be provided by a device. Devices that do not have a synchronized clock (with time server protocols like NTP or radio control like DCF77) or do not have a reliable absolute time source should use relative timestamps. Note that most examples in this document refer to relative timestamps.

	Should be set to “false” (usage of relative timestamps) if absolute timestamps are not explicitly needed.
Interruptions.InterruptionsAllowed	Specifies whether the device can be interrupted (paused) during runtime. This allows a more flexible energy management for the device. For instance the EM can interrupt a device in case of unpredictable bad weather conditions or when the user switches on a device with conflicting energy needs and restart the interrupted device afterwards. Should be set to “true”. Set to “false” only if the device operation cannot be paused, e.g. for some program based devices.
Requests.OptionalEnergy	Specifies whether the device supports the consumption of optional energy, i.e. energy that can be assigned to the device but is not essentially needed by the device to operate correctly. For example heat-pumps can increase the water-temperature in the storage to store additional energy. If this option is set to true, additional configuration options (for constraints, etc.) can be displayed in the EM’s GUI. If a SEMP device uses either the MinRunningTime or MinEnergy fields of the PlanningRequest.Timeframe structure to define an optional energy demand, the value of this field must be set to “true”, otherwise “false”.

Example:

```
<DeviceInfo>
  <Identification>…</Identification>
  <Characteristics>…</Characteristics>
  <Capabilities>
    <CurrentPower>
      <Method>Measurement</Method>
    </CurrentPower>
    <Timestamps>
      <AbsoluteTimestamps>false</AbsoluteTimestamps>
    </Timestamps>
    <Interruptions>
      <InterruptionsAllowed>true</InterruptionsAllowed>
    </Interruptions>
    <Requests>
      <OptionalEnergy>true</OptionalEnergy>
    </Requests>
  </Capabilities>
</DeviceInfo>
```

4.4.2 Device status

The device status data structure contains information about the current status of the devices managed by the gateway.

Device Power

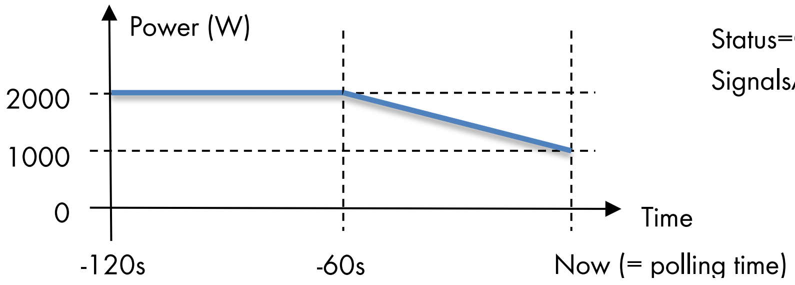
The most important section of the device status is the PowerConsumption element. It contains the device's power consumption over the past time specified by PowerInfo elements.

At least the Power (AveragePower), the measuring interval (AveragingInterval) and the time of the end of the measurement (Timestamp) have to be provided whenever the device is switched on (Status=On). If the device is switched off (Status=Off) it should communicate a power of 0W.

The measuring interval (AveragingInterval) **must always be set to one minute (60s)**. The power (AveragePower) is the averaged power over the measuring interval (one minute). The device should not use the current power at the end of the measuring interval instead.

The timestamp of the measurement of the previous minute is 0 in relative time. **The measurement of the previous minute must always be provided** if the device is on. It is possible to provide the measurements of up to the last 10 minutes (relative timestamps must be set to 0, -60, -120, -180, -240, -300, -360, -420, -480, -540). In contrast to the previous minute, the other nine measurements are optional and can be used by the EM if some measurements got lost, for example because of network errors. If multiple measurements are provided, there must not be a time gap between the measurements.

Given the following device power:



Status=On
SignalsAccepted=true/false

At the time of polling the device would return the following information:

- Timestamp: 0 (relative time)
- AveragingInterval: 60s
- AveragePower: 1500W (not the current power of 1000W)

The power of the last minute (now to -60s) must be provided. The power of the minute before (-60s to -120s) can optionally be provided.

```
<PowerConsumption>
  <PowerInfo>
    <AveragePower>1500</AveragePower>
    <Timestamp>0</Timestamp>
    <AveragingInterval>60</AveragingInterval>
  </PowerInfo>
  <PowerInfo><!-- optional -->
    <AveragePower>2000</AveragePower>
    <Timestamp>-60</Timestamp>
    <AveragingInterval>60</AveragingInterval>
  </PowerInfo>
</PowerConsumption>
```

The following examples demonstrate **invalid** usages of the PowerConsumption element.

Missing PowerInfo for last minute:

```
<PowerConsumption>
  <PowerInfo>
    <AveragePower>1500</AveragePower>
    <Timestamp>-60</Timestamp>
    <AveragingInterval>60</AveragingInterval>
  </PowerInfo>
</PowerConsumption>
```

AveragingInterval ≠ 60:

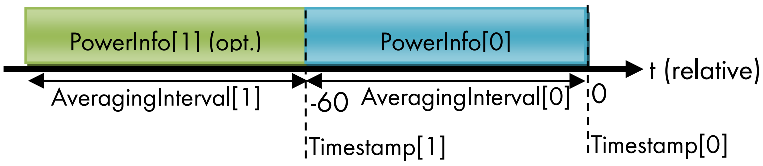
```
<PowerConsumption>
  <PowerInfo>
    <AveragePower>1500</AveragePower>
    <Timestamp>0</Timestamp>
    <AveragingInterval>30</AveragingInterval>
  </PowerInfo>
</PowerConsumption>
```

Non-consecutive PowerInfo elements:

```
<PowerConsumption>
  <PowerInfo>
    <AveragePower>1500</AveragePower>
    <Timestamp>0</Timestamp>
    <AveragingInterval>60</AveragingInterval>
  </PowerInfo>
  <!-- PowerInfo with timestamp=-60 is missing -->
  <PowerInfo>
    <AveragePower>1500</AveragePower>
    <Timestamp>-120</Timestamp>
    <AveragingInterval>60</AveragingInterval>
  </PowerInfo>
</PowerConsumption>
```

Fields used for generic devices are explained below. For a full list of available fields and their datatypes see the SEMP XSD. In some specific application scenarios additional fields might be required. These exceptions to the generic usage are described in separate Application Notes (AN).

Field	**Description**
DeviceId	Device ID of the device as specified in the device's DeviceInfo.
EMSignalsAccepted	Specifies if recommendations from the EM (see DeviceControl) are accepted at the moment. If "true" recommendations will be accepted. By setting this to "false" this can be used to inform the EM that the device is currently in manual mode and will not be able to follow recommendations. In this case the EM should not send DeviceControl messages to the device. The value should be set to "false" if there is no timeframe in the planning request section for the device at the moment. If it is set to "false" and timeframes for the device are given in a planning request section, the EM will not use the information for planning.
Status	Must be set to "On", "Off" or "Offline". A device being in status "On" means that the component controlled by this SEMP device is switched on. It is "Off" if the controlled component is switched off. This is even the case if the device itself is still operating or in standby mode. If the SEMP gateway is not the controllable device itself and the communication to the controllable device is lost, the status can be set to "Offline" to signal a communication timeout.
ErrorCode (opt.)	Can be used to notify the EM of an error state of the device. If this element is not present or a value of "0" is provided, no error is pending. Error codes are vendor specific and not interpreted by the EM. They might be displayed in the EM GUI to notify the user of error states.
PowerConsumption.PowerInfo	Whenever the device is switched on - no matter if recommended by the EM or on its own - it should provide the current power consumption. At least the average power, the time interval in which the power was measured and the timestamp of the end of the averaging interval have to be provided. Additionally the range of power consumption during the interval can be given with the Min-/MaxPower elements. If available the standard deviation and skewness of the measured values can be provided by StdDevPower and SkewPower. This is helpful if multiple measurements were merged into one value.

	The timestamp marks the time of the end of measurement. If relative timestamps are used (see device capabilities), this will normally be "0".  PowerInfo[1] (opt.) PowerInfo[0] t (relative) AveragingInterval[1] -60 AveragingInterval[0] 0 Timestamp[1] Timestamp[0] **When the device is in state On=true, it must provide a PowerInfo with the average power of the last minute: Timestamp=0 (relative), AveragingInterval=60.** Optionally, in addition to the PowerInfo of the last minute, it is possible to send up to nine more PowerInfo elements – i.e. 10 PowerInfos in total. Each of these PowerInfo elements provides the power of one of the previous ten minutes. The AveragingInterval for each of these PowerInfo elements has to be set to 60. The Timestamp for the i-th (i=0..9) PowerInfo has to be set to i*-60 (relative). Power values are interpreted by the EM by taking the CurrentPower.Method capability (see DeviceInfo) into account.

Example:

```
<DeviceStatus>
  <DeviceId>F-11223344-112233445566-00</DeviceId>
  <EMSignalsAccepted>true</EMSignalsAccepted><!-- true: ready to be con-
trolled by EM (e.g. not in manual operation mode) -->
  <Status>On</Status><!-- On: running, Off: otherwise -->
  <PowerConsumption>
    <PowerInfo>
      <AveragePower>7100</AveragePower><!-- in W -->
      <Timestamp>0</Timestamp><!-- end of averaging interval -->
      <AveragingInterval>60</AveragingInterval>
    </PowerInfo>
  </PowerConsumption>
</DeviceStatus>
```

4.4.3 Planning requests

The planning request data structure is the core component of SEMP. It is used to specify the current energy needs of a SEMP device. The EM uses this information to schedule the managed devices.

Timeframes

Each time the EM requests the PlanningRequest data structure of a device (once for each polling interval) the gateway provides a list of pending energy demands for each managed device.

Each demand is represented by a timeframe in the planning request. A timeframe specifies the range of time in which the demand should be fulfilled and the amount of energy (in terms of runtime) that is required during the given time range.

The time range is specified by the earliest start time of operation (EarliestStart) and the time at which the device operation must be finished at the latest (LatestEnd).

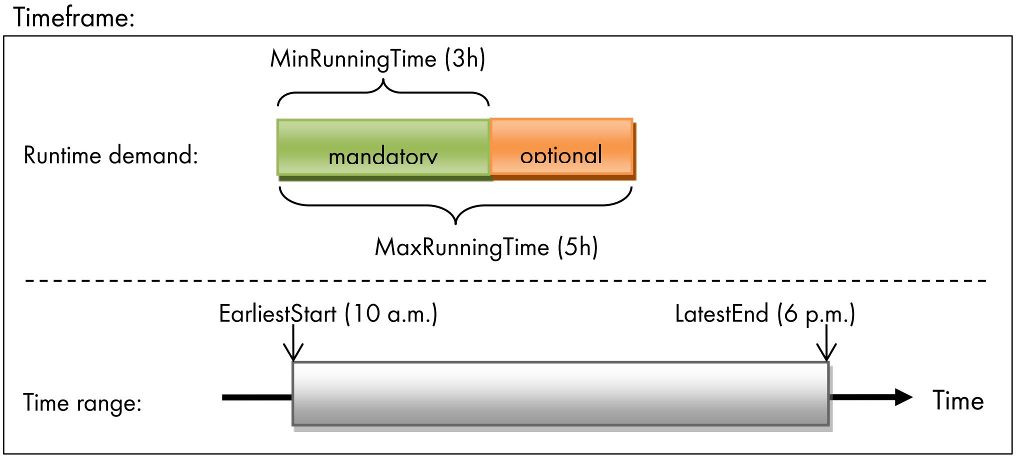
The amount of the required energy is specified by a minimum operation runtime (MinRunningTime) and a maximum operation runtime (MaxRunningTime). The difference between maximum and minimum runtime (MaxRunningTime - MinRunningTime) specifies the optional amount of energy (in terms of runtime) that a device is able to consume during the given time range. Here are some examples where optional energy demands might be requested:

- if a storage device (heater, cooler, battery) can store additional energy (e.g. surplus solar energy)
- if the optional operation is desirable but not necessary (e.g. eco-mode).

If only a mandatory and no optional energy demand is pending, only a maximum runtime (MaxRunningTime) has to be defined and the minimum runtime can be omitted (as it is equal to MaxRunningTime in this case).

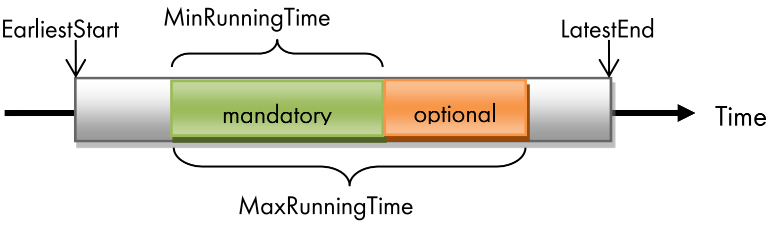
The EM has to schedule the device in such a way that at the time given by LatestEnd, the device has operated for the minimum requested runtime (MinRunningTime). The EM will only schedule the optional runtime (determined by MaxRunningTime) if the operation of the device meets certain conditions that the user can define by a GUI. Typically the condition is that no or only a user defined amount of grid energy is consumed by the device during the optional runtime.

A SEMP device requests a minimum runtime of 3 hours and a maximum runtime of 5 hours. The time range is from 10 a.m. 6 p.m.

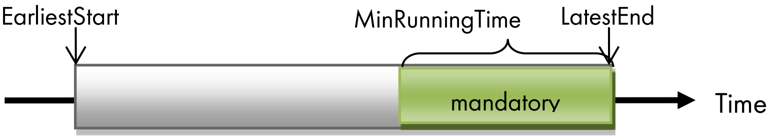


The EM can now schedule the runtime demand in the boundaries of the given time range. Below three examples for the schedules that might be created by the EM:

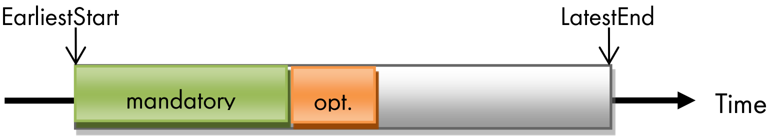
A) Maximum runtime allocated:



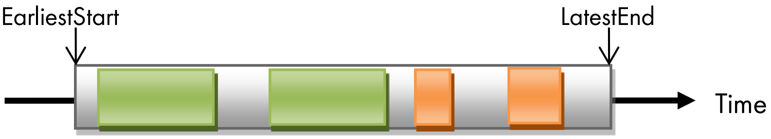
B) No optional runtime allocated:



C) Optional runtime reduced:

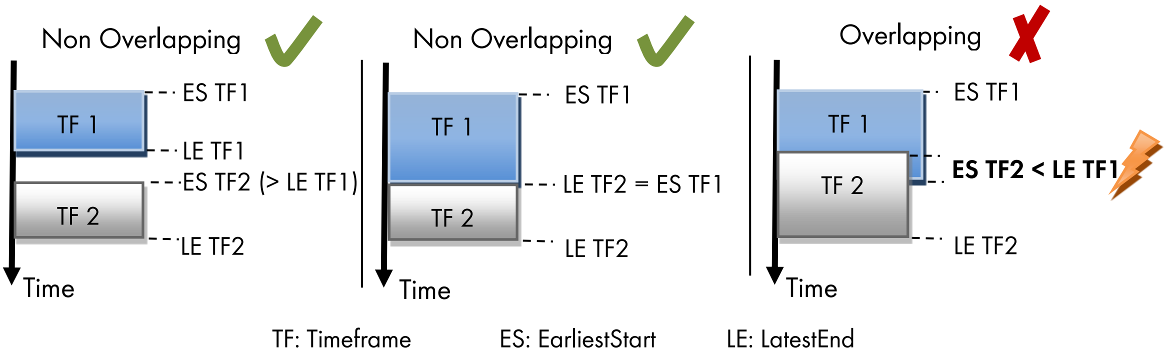


If the device is interruptible, the runtime can also be split into segments:



Timeframe overlap

Timeframes defined for one device must not overlap. Overlapping occurs if the time ranges defined by EarliestStart and LatestEnd of two timeframes of a device intersect.

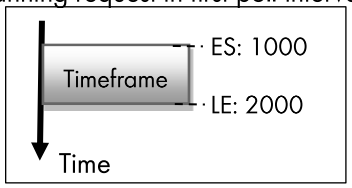


Updating planning requests

The planning request ***must contain all current energy demands*** (timeframes) for devices selected by the EM (with “<baseURL>/” all devices managed by the gateway are selected). Timeframes defined during previous poll intervals are discarded and only the timeframes defined in the current PlanningRequest are used to generate plans for the devices. If a previously defined timeframe is missing now, it is assumed that the energy need is canceled.

Example:

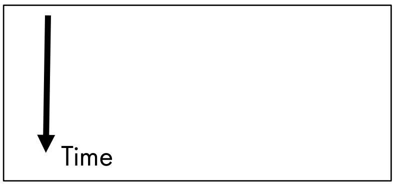
Planning request in first poll interval



- One demand in list

```
<PlanningRequest>
  <Timeframe>
    <DeviceId>…</DeviceId>
    <EarliestStart>1000</EarliestStart>
    …
  </Timeframe>
</PlanningRequest>
```

Planning request in second poll interval



- No demand in list ➔ no demand
  → Missing TF will not be planned

```
<!--PlanningRequest element omitted-->
```

If no energy demand is pending, the PlanningRequest element must be omitted. Returning an empty PlanningRequest is invalid as a PlanningRequest must contain at least one timeframe.

OK:

```
<!--PlanningRequest element omitted-->
```

Invalid (do not use):

```
<PlanningRequest>
</PlanningRequest>
```

If a device has already received a switch on recommendation for a timeframe, the timeframe must still be listed in the planning request until it is completed (requested runtime or energy consumed). If the device was switched on by the EM and the timeframe is removed while the device is still running, the EM will recommend to switch off if signals are accepted by the device (EMSignalsAccepted=true in DeviceStatus).

If energy demands of a device change, those changes should be reflected in the planning request. In each poll interval the energy demands can (and should) be adjusted to the current needs. This applies to changes in EarliestStart/LatestEnd as well as changes in the running time or energy need. This way the predictions of the expected energy need can be defined more precisely over time. Although a planning request can be provided that significantly differs from the previous one, usually the updated requests should only differ slightly.

The device should track its current runtime need. This implies that Min/MaxRunningTime is always set to the remaining runtime need, i.e. Min-/MaxRunningTime has to be adjusted by the amount of runtime that was already consumed. This in turn allows the EM to perform a better tracking of the remaining runtime need. There are four cases that have to be distinguished in order to adjust EarliestStart, LatestEnd and Min-/MaxRunningTime. In the following examples we assume that the device has an initial energy request of at least RT_{min0} and maximum RT_{max0}. The amount of time the device is already running after its operation was started is denoted by RT_{run}. Further we assume that times are given relative to now:

1. Device has not started operation:
   a. Device is not ready to be started by the EM:
      EarliestStart > 0, MinRunningTime = RT_{min0}, MaxRunningTime = RT_{max0}
   b. Device is ready to be started by the EM now:
      EarliestStart = 0, MinRunningTime = RT_{min0}, MaxRunningTime = RT_{max0}
2. Device has started operation:
   EarliestStart = 0, MinRunningTime = max(0, RT_{min0} - RT_{run}), MaxRunningTime = max(0, RT_{max0} - RT_{run})

This is under the assumption that the initial runtime need did not change. If there is a change in the runtime need, simply set Min-/MaxRunningTime according to the remaining need.

At 10 a.m. a device (in switched off state) has an initial runtime demand of 2 hours in a time range of 5 hours with the earliest possible start in one hour:

```
<PlanningRequest>
  <Timeframe>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EarliestStart>3600</EarliestStart>
    <LatestEnd>21600</LatestEnd>
    <MaxRunningTime>7200</MaxRunningTime>
  </Timeframe>
</PlanningRequest>
```

Five minutes later (10:05 a.m.) the device is still switched off as no switch-on recommendation was sent by the EM so far. As a result the runtime demand did not change. At this time the planning request will look like this:

```
<PlanningRequest>
  <Timeframe>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EarliestStart>3300</EarliestStart>
    <LatestEnd>21300</LatestEnd>
    <MaxRunningTime>7200</MaxRunningTime>
  </Timeframe>
</PlanningRequest>
```

Again five minutes later (10:10 a.m.) the device is still switched off. But the energy demand changed as the device is now able to provide a more precise prediction.

```
<PlanningRequest>
  <Timeframe>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EarliestStart>3000</EarliestStart>
    <LatestEnd>21000</LatestEnd>
    <MaxRunningTime>6000</MaxRunningTime>
  </Timeframe>
</PlanningRequest>
```

Two hours later (12:10 p.m.) the EM recommends to switch the device on. The device follows the recommendation):

```
<PlanningRequest>
  <Timeframe>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EarliestStart>0</EarliestStart>
    <LatestEnd>13800</LatestEnd>
    <MaxRunningTime>6000</MaxRunningTime>
  </Timeframe>
</PlanningRequest>
```

Five minutes later (12:15 p.m.) the runtime demand is reduced:

```
<PlanningRequest>
  <Timeframe>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EarliestStart>0</EarliestStart>
    <LatestEnd>13500</LatestEnd>
    <MaxRunningTime>5700</MaxRunningTime>
  </Timeframe>
</PlanningRequest>
```

After five more minutes(12:20 p.m.) the device adjusts the energy demand prediction again. Half an hour less than originally predicted will be needed to perform the task (as five minutes passed the runtime demand is decreased by additional five minutes):

```
<PlanningRequest>
  <Timeframe>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EarliestStart>0</EarliestStart>
    <LatestEnd>13200</LatestEnd>
    <MaxRunningTime>3600</MaxRunningTime>
  </Timeframe>
</PlanningRequest>
```

The device stays on for another hour and switches off as it has reached the maximum running time. The PlanningRequest element is omitted now as no further energy demand is pending:

```
<!-- PlanningRequest omitted -->
```

Fields used for generic devices are explained below. For a full list of available fields and their datatypes see the SEMP XSD. In some specific application scenarios additional fields might be required. These exceptions to the generic usage are described in separate Application Notes (AN).

Field	**Description**
DeviceId	Device ID of the device as specified in the device's DeviceInfo.
EarliestStart	Specifies the earliest start time of the device. Note that timestamps in the past must not be used. Instead • set EarliestStart=0 (if rel. time used or to current time for abs. time) if the device can be started immediately. • set EarliestStart =0 (if rel. time used or to current time for abs. time) when the device has already been started and is still running.
LatestEnd	Specifies the end of the planning range. This is the latest possible time the device operation has to be finished.
MinRunningTime	The minimum amount of time the device needs to run in the time range specified by EarliestStart and LatestEnd. It will be assigned to device by the EM even if there is no PV-production so that grid-energy has to be used to power the device.

	Note: If set to "0" all of the energy is optional and should only be allocated to the device if certain conditions are fulfilled (e.g. cheap PV-energy available).
MaxRunningTime	The maximum amount of time the device needs to run in the time range specified by EarliestStart and LatestEnd. The difference between MaxRunningTime and MinRunningTime is the optional runtime that can be assigned to the device by the EM if cheap energy (e.g. excess PV-energy) is available. As described in the previous section, after the device has been switched on, adjust this value according to the remaining runtime demand.

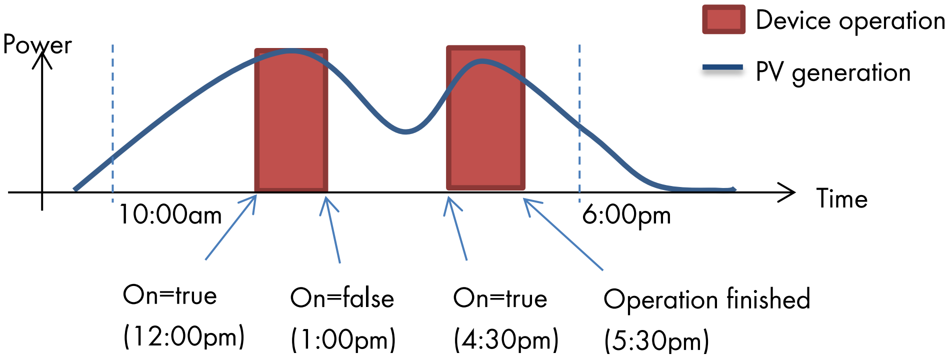
Example:

```
<PlanningRequest>
  <Timeframe>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EarliestStart>0</EarliestStart><!-- start possible now  -->
    <LatestEnd>21600</LatestEnd><!-- range: must be finished in 6h -->
    <MinRunningTime>7200</MinRunningTime><!-- exactly 2h -->
    <MaxRunningTime>7200</MaxRunningTime><!-- exactly 2h -->
  </Timeframe>
  <!-- more timeframes if available -->
</PlanningRequest>
```

4.5 Recommendations from the EM

Whenever the gateway provides planning requests to the EM, the EM starts to create new plans with optimized switch-on and switch-off times for the devices. The plan itself will not be provided to the device. Instead the EM will send recommendations via the DeviceControl structure whenever the device should be switched on or off.

An interruptible SEMP device requests a runtime of exactly two hours with earliestStart at 10 a.m. and latestEnd at 6 p.m. The EM splits the runtime into two parts to align the device operation with the predicted PV generation and creates the following plan:



According to the plan the EM will send a switch-on recommendation to the device at 12 p.m. Followed by a switch-off recommendation one hour later and again a switch-off recommendation at 4:30 p.m. As the device is able to detect the end of the total runtime by itself, the EM does not have to send a switch-off recommendation at 5:30 p.m.

The EM does not provide the plan to the SEMP device. In addition the plans might change over time because of changed PV forecasts or energy demands.

Recommendations (switch on/off) should be followed by the gateway/device whenever possible. Recommendations in DeviceControl messages always apply to the current time and should be followed immediately.

As an EM manages multiple devices, ignoring a recommendation might conflict with plans for other devices. If a device does not follow the recommendations and switches on when another device is already running, this might result in high energy prices.

If complying to a recommendation would lead to damages, inefficient behavior or customer inconvenience, the device must either ignore or modify the recommendation to meet the constraints. It is the device's responsibility to assure proper operation.

Switch-on recommendation

When a gateway receives a switch-on recommendation it should make sure that the specified device starts to run or resumes its operation as soon as possible.

```
<?xml version="1.0" encoding="UTF-8"?>
<EM2Device xmlns=…>
  <DeviceControl>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <On>true</On>
    <Timestamp>0</Timestamp>
  </DeviceControl>
</EM2Device>
```

Field	Description
DeviceId	SEMP Device-ID of the target device for this control message.
On	True if the device should switch on (or continue). False if the device should switch off. True in this case.
Timestamp	Timestamp (relative or absolute) of the generation of the message. Indicates the time when the message was generated, not the switch-on time.

Due to network problems or other errors it is possible that a device does not receive a switch-on recommendation from the EM even for timeframes with mandatory runtime demands. In this case the device should switch-on manually at the latest possible time if necessary.

Switch-off recommendation

When a gateway receives a switch-off recommendation for a device it should make sure that the specified device stops its operation as soon as possible.

If the device consumed the maximum requested energy (or runtime) the SEMP device has to switch off by itself. Although the EM might send a switch-off recommendation to signal the end of operation it does not have to.

If a device is interruptible a switch-off recommendation can either signal an interruption (which will be resumed later) or the termination of the operation. A termination only occurs if at least the requested minimum energy was consumed by the device. There is no way to distinguish both signals (interruption or termination) as an interruption might turn into a termination. For example if the PV generation drops unexpectedly during the interruption due to bad weather and the remaining energy demand is optional.

```
<?xml version="1.0" encoding="UTF-8"?>
<EM2Device xmlns=…>
  <DeviceControl>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <On>false</On>
    <Timestamp>0</Timestamp>
  </DeviceControl>
</EM2Device>
```

Field	Description
DeviceId	SEMP Device-ID of the target device for this control message.
On	True if the device should switch on (or continue). False if the device should switch off. False in this case.
Timestamp	Timestamp (relative or absolute) of the generation of the message. Indicates the time when the message was generated, not the switch-off time.

⚠ Due to network problems or other errors it is possible that a device does not receive a switch-off recommendation from the EM. In this case the EM should switch-off manually when the runtime demand is satisfied.

5 Examples:

1. SEMP-gateway responds to EM’s poll request to the URL “<baseURL>/” with all information:

```
<?xml version="1.0" encoding="utf-8"?>
<Device2EM xmlns="http://www.sma.de/communication/schema/SEMP/v1">
  <DeviceInfo>
    <DeviceInfo>
    <Identification>
      <DeviceId>F-11223344-112233445566-00</DeviceId>
      <DeviceName>Name of the first device</DeviceName>
      <DeviceType>Heater</DeviceType>
      <DeviceSerial>ZYXVU342432</DeviceSerial>
      <DeviceVendor>ABC Ltd</DeviceVendor>
      <DeviceURL>http://abc.com/config?id=6767</DeviceURL>
    </Identification>
    <Characteristics>
      <MaxPowerConsumption>1500</MaxPowerConsumption>
      <MinOnTime>60</MinOnTime>
      <MinOffTime>60</MinOffTime>
    </Characteristics>
    <Capabilities>
      <CurrentPower><Method>Measurement</Method></CurrentPower>
      <Timestamps><AbsoluteTimestamps>false</AbsoluteTimestamps></Timestamps>
      <Interruptions>
        <InterruptionsAllowed>true</InterruptionsAllowed>
      </Interruptions>
      <Requests><OptionalEnergy>true</OptionalEnergy></Requests>
    </Capabilities>
  </DeviceInfo>
  <DeviceStatus>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <EMSignalsAccepted>true</EMSignalsAccepted>
    <Status>On</Status>
    <PowerConsumption>
      <PowerInfo>
        <AveragePower>1000</AveragePower><!-- (in Watts) -->
        <Timestamp>0</Timestamp>
        <AveragingInterval>60</AveragingInterval>
      </PowerInfo>
    </PowerConsumption>
  </DeviceStatus>
  <PlanningRequest>
    <Timeframe>
      <DeviceId>F-11223344-112233445566-00</DeviceId>
      <EarliestStart>0</EarliestStart><!-- already running -->
      <LatestEnd>10800</LatestEnd><!—latest end in 3h -->
      <!-- mandatory demand of 0.5h (MinRunningTime = MaxRunningTime) -->
      <MinRunningTime>1800</MinRunningTime>
      <MaxRunningTime>1800</MaxRunningTime>
      <PreferenceIndifferentAreas>Late</PreferenceIndifferentAreas>
    </Timeframe>
    <Timeframe>
      <DeviceId>F-11223344-112233445566-00</DeviceId>
      <EarliestStart>18000</EarliestStart><!-- earliest start in 5h -->
      <LatestEnd>25200</LatestEnd><!-- latest end in 7h -->
      <!-- optional runtime of 30min (MinRunningTime=0) -->
      <MinRunningTime>0</MinRunningTime>
      <MaxRunningTime>1800</MaxRunningTime>
    </Timeframe>
  </PlanningRequest>
</Device2EM>
```

2. EM sends a switch-on recommendation for a device to the SEMP gateway:

```
<?xml version="1.0" encoding="utf-8"?>
<EM2Device xmlns="http://www.sma.de/communication/schema/SEMP/v1">
  <DeviceControl>
    <DeviceId>F-11223344-112233445566-00</DeviceId>
    <On>true</On>
    <Timestamp>0</Timestamp>
  </DeviceControl>
</EM2Device>
```

6 Referenced documents

IANA. (2015). *PEN (Private Enterprise Numbers)*. Retrieved 07 20, 2015, from http://www.iana.org/assignments/enterprise-numbers/enterprise-numbers

IEEE. (2015). *Assigned OUI numbers*. Retrieved 07 20, 2015, from http://standards-oui.ieee.org/oui.txt

RFC 2279: UTF-8, a transformation format of ISO 10646 (character encoding). (1998, January). Retrieved from http://www.ietf.org/rfc/rfc2279.txt

RFC 2616: HTTP: Hypertext Transfer Protocol 1.1. (1999, June). Retrieved from http://www.ietf.org/rfc/rfc2616.txt

The Open group. (1997). *DCE variant of Universal Unique Identifiers (UUIDs)*. Retrieved from http://www.opengroup.org/onlinepubs/9629399/apdxa.htm

UPnP Forum. (2002, August 08). *UPnP Device Architecture v1.0 Annex A - IPv6*. Retrieved March 08, 2013, from http://upnp.org/specs/arch/UPnP-arch-AnnexAIPv6-v1.pdf

UPnP Forum. (2008, October 15). *UPnP Device Architecture 1.0*. Retrieved March 08, 2013, from http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20080424.pdf

UPnP Forum. (2008, October 15). *UPnP Device Architecture 1.1*. Retrieved March 11, 2013, from http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf

XML: Extensible Markup Language. W3C recommendation. (n.d.). Retrieved from http://www.w3.org/TR/2000/REC-xml-20001006