# Smart Power Management Protocol

Draft Version 0.1

**Author(s)**: Mark W., Andrew D., Thomas C.
Clear Blue Technologies Inc,

Kashif A., and Vishal T.
Facebook Inc.

# Scope

This document specifies version 0.1 of the Smart Power Management Protocol (SPMP). The SPMP V0.1 introduces following feature to the power management for rural infrastructure:

- Bootstrap service for the solar charge controller (device)
- Management, service enablement and information reporting using object model for the device
- Smart load power management
- Inter-object conversion to support smart load communication
- Integration with public and private cloud service provider
- Integration with network management service.

# Introduction

Smart power management protocol (SPMP) is an application layer protocol between the smart charge controller, smart load and cloud infrastructure. SPMP enable bootstrapping of the smart controller and smart load, information reporting from controller, and ability to manage the devices using public/private cloud.

SPMP utilize client-server architecture and is composed of five major components:

- Smart charge controller
- Smart load
- Bootstrap server
- Device management and Information reporting server
- Public/Private cloud

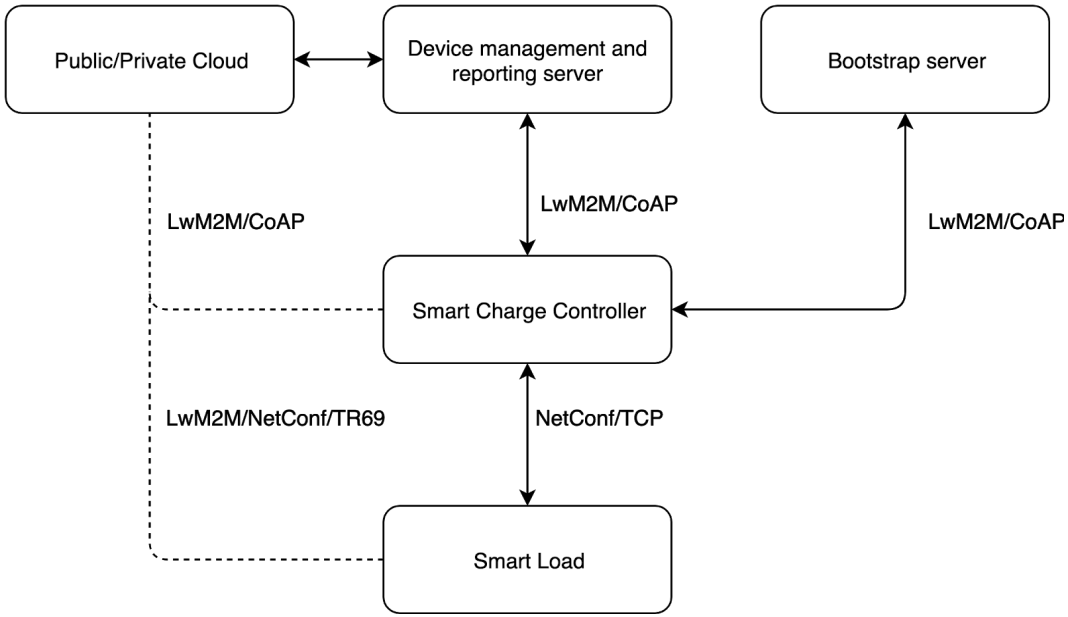



Figure 1: The overall architecture of the SPMP enabled system.

## Background

**Light-weight Machine to Machine:** LwM2M is a protocol from the Open Mobile Alliance for M2M and IoT device management. LwM2M is frequently utilized with CoAP (Constrained Application Protocol) specifically for resource constrained Internet devices, which solar controllers and related telemetry would certainly fit. CoAP provides throughput efficiency and utilizes UDP for transport. Further information on LwM2M can be found on the OMA Specworks website.

**NETCONF:** NETCONF is a standardized protocol adopted by the IETF in 2006. It provides mechanisms to manage configurations of network devices through a simple remote procedure call (RPC) layer. XML data is used to encode configuration data, which is then exchanged over a secure transport protocol. Further information on NETCONF can be found on the Cisco Tail-f website.

**SNMP:** Simple Network Management Protocol (SNMP) is an application-layer protocol used to manage and monitor network devices and their functions.

**ModBUS**: Modbus is a serial communications protocol for use with its programmable logic controllers (PLCs) with extensions to work over TCP/IP.

## SPMP Overview

This is an early draft specification (v0.1) of the related LwM2M and NETCONF objects designed to integrate:

- Solar charge controllers
- Smart loads (e.g., OpenCellular devices)
- Public / private cloud (e.g., Clear Blue's Illuminance cloud offering, or third party solution)
- Telco Network management system (NMS)
- Network device bootstrap services

The SPMP is a client/server based protocol that exposes various resources on remote devices as resources. Through LwM2M, these resources are accessed through uniform resource identifiers (URI's). Related resources are further grouped together into objects. Rather than utilize HTTP, CoAP is integrated as the transfer agent in order to mitigate cost issues associated with data transfer overhead normally found with HTTP.

Various objects already have been adopted by the LwM2M working group, and when possible, we have utilized these objects as a part of our solar controller specification (example: voltage and current objects.) We have inherited these objects, and included them within a larger set of objects that are unique to solar controllers.

## Interfaces

According to the SPMP architecture there are four interfaces: 1.) Bootstrap, 2.) Device Management and information reporting, 3.) Smart load, and 4.) Public/private cloud. SPMP is heavily derived from the Light-weigth Machine to Machine (LwM2M) tailored to smart power management. Most to the material presented here are from SPMP and re-stated for completion only. The operation of each interface is defined in section.

Bootstrap

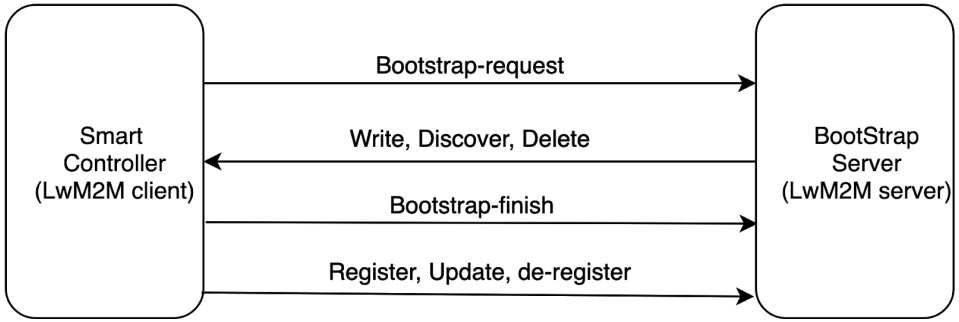



Fig.2 shows the operation model for interface “Bootstrap”, which also include device registration. For this interface, the operations are an uplink operation named “Bootstrap-Request”, “Register”, “Update” and “De-register” and downlink operations named “Discover”, “Write”, “Delete” and “Bootstrap-Finish”. These operations are used to initialize the needed Object(s) for the LwM2M-based smart controller to register with one or more LwM2M-enabled servers. Two mode exists for bootstrapping 1.) client initiated and 2.) server initiated. In the mode where the Server is addressing the Bootstrap Information to the charge controller, the Server MUST inform the controller when this transfer is over by sending a “Bootstrap-Finish” command. Each charge controller MUST register with each server it will update about its resources.
SPMP will use a device initiated bootstrap as defined in the LWM2M protocol. The charger will be factory provisioned with an address and credentials from a bootstrap server. If a charger only has bootstrap credentials when it powers up, it will contact the bootstrap server. The bootstrap server will configure the charger with the URL and credentials to communicate with the device management server. See section 5.2 in the LWM2M specification for more details.

When a charger that has been bootstrapped goes online, it will send a registration request to the Device Management Server. This informs the Device Management server that the charger is ready to receive commands. Metadata such as the LwM2M version, and the transport binding mode are also exchanged at this time. See section 5.3 of the LwM2M specification for more details. During this process, the server will also perform a discover operation on the client to determine its current configuration and state.

Device Management and Information Reporting

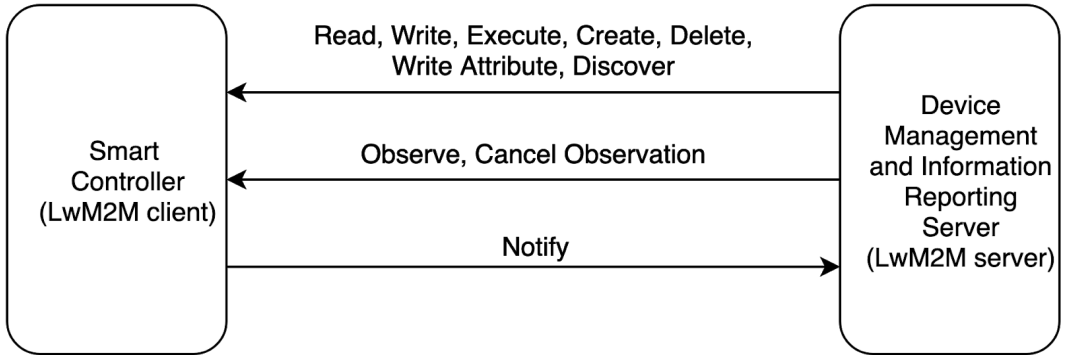
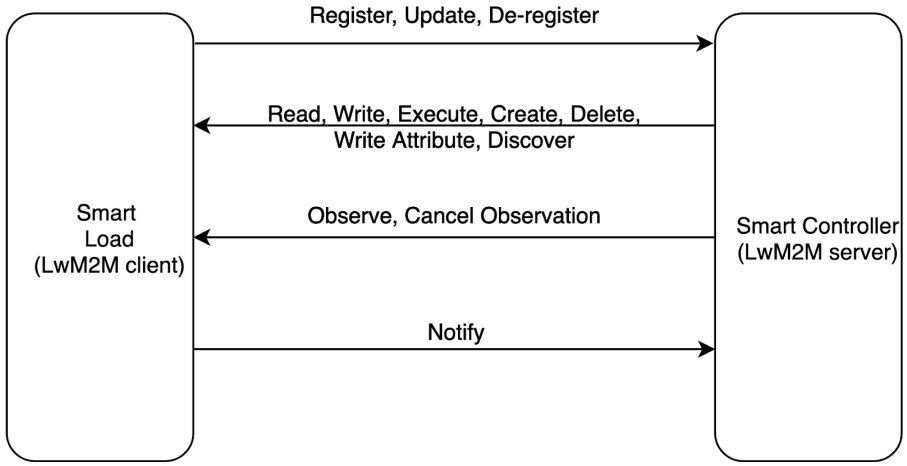



Fig.3 shows the operation model for interface “Device Management and Information Reporting”. For this interface, the operations are “Read”, “Write”, “Execute”, “Create”, “Delete”, “Write Attributes”, “Discover”, “Observe”, “Cancel Observation” and “Notify”. Subset of this operation model also applies to Public/Private cloud which acts as Device management and information reporting server. For some use cases, the bootstrapping server might also be co-located with device management and reporting server on the cloud.

Once the charger has registered, it can be managed by the server. The server can manage the device by instantiating new objects such as a Smart Load configuration for a port, by writing values to existing objects to control loads or change how data is collected, or subscribe to events and telemetry.
A custom LwM2M object will be used to transmit telemetry data from the charger to the server. One Port Status object per port will be initialized. The resources in the object will include voltage and amperage averages, the length of time in seconds over which the averages were obtained, the timestamp of the last update, and a link to an alerts object. The charger will collect this data from the sensors, and will update the values in the Port Status object periodically according to the length setting. The server may receive telemetry updates by subscribing (LwM2M Observe operation, see section 5.5.1 of LwM2M specification) to either individual fields each port status object and the corresponding alerts object, or the entire object. The server may also change the length of time over which the voltage and amperage are averaged by writing to the “length” field. The alerts object consists of many optional boolean resources that will be instantiated by the firmware of the client according to its capabilities. Each resource will correspond to a type of alert that may occur on that port, such as short circuit, over-voltage, and over-current.

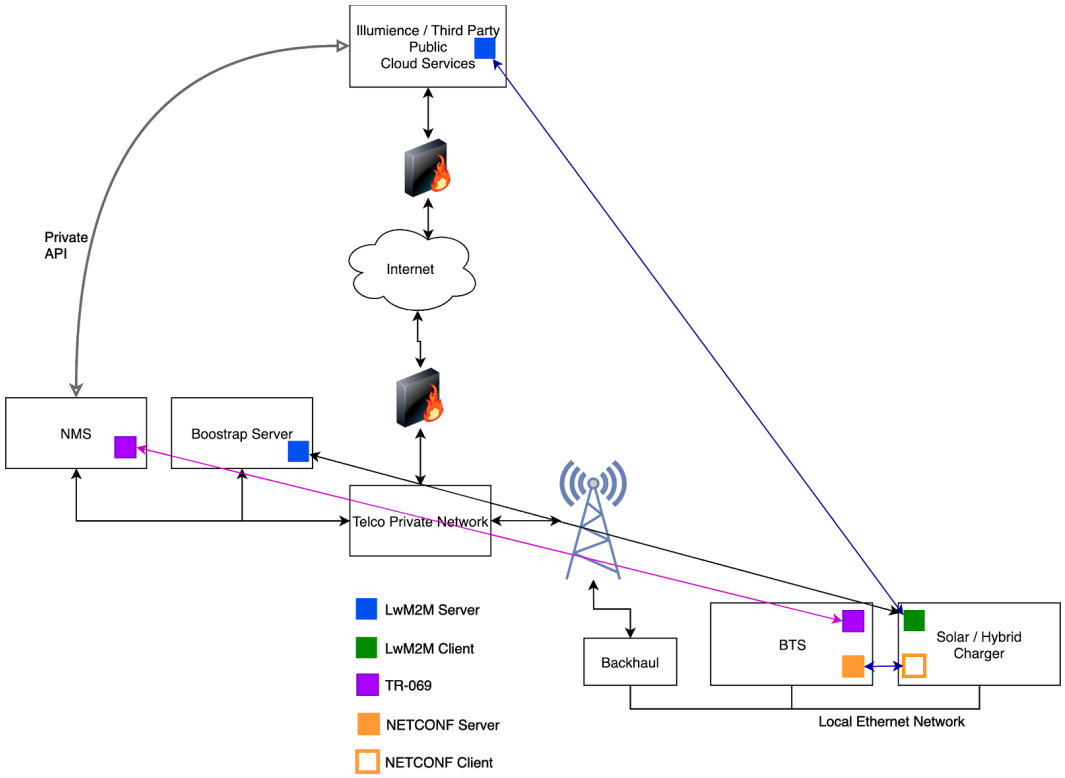
Smart Load

Fig 4. Shows the operation model for interface “Smart Load”. For this interface, the operations are “Register”, “Update”, “De-register”, “Read”, “Write”, “Execute”, “Create”, “Delete”, “Write Attributes”, “Discover”, “Observe”, “Cancel Observation” and “Notify”. For this version, the charge controller converts the object resources into respective NetConf data models.




## Example scenario

For the purposes of this example, the remote Telco location consists of a single backhaul, BTS and smart solar / hybrid charger (the “charger”). The backhaul communicates with the Telco over a private network, and on that private network is an outbound link to the Internet which provides access to a third party (or Clear Blue’s Illumience) cloud platform (the “public cloud”).




The SPMP protocol utilizes the IETF standardized LwM2M protocol stack, which uses the Constrained Application Protocol (CoAP) as the low level transfer protocol over UDP. Secure communications are implemented through Datagram Transport Layer Security (DTLS). DTLS utilizes public key technology to secure communications between client and server.

A bootstrap server on the Telco's private network provides the credentials necessary for the smart charger to communicate with the remote BTS, as well as credentials for a third party cloud platform.

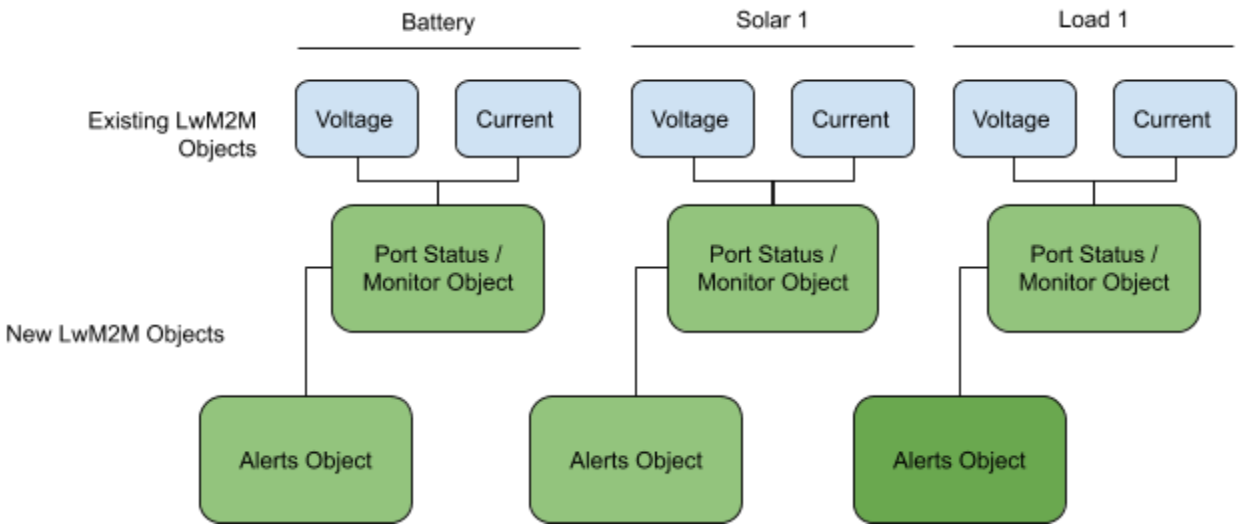
Credentials for the BTS will be used to bootstrap the local configuration of NETCONF on the solar / hybrid charger.

Once configured, the charger will be able to submit telemetry data to the public cloud, as well as receive configuration parameters regarding smart load control. Smart load control will provide details for the charger to configure the load ports (provides power to the BTS). This same configuration will be used to adjust parameters on the BTS as power fluctuates within the local battery system. As available energy starts to decrease, the charger will be able to connect to the NETCONF server on the BTS to reconfigure sub-systems to lower power consumption. As the battery normalizes, the charger will connect to the NETCONF server to return to normal operations (increasing power consumption.)

The actual capabilities of each BTS will be defined by the BTS manufacturer through LwM2M object definitions. In the initial release, we envision the charger converting these objects to NETCONF for communication with the BTS. Ultimately, it will be possible to utilize a variety of RPC type integrations with the BTS (LwM2M, NETCONF, gRPC, etc.)

## SPMP Data Model and Objects

SPMP uses the LwM2M data model. The state and configuration of each client can be accessed, monitored, and updated through this data model. There are three levels: Objects, Object Instances, and Resources. Each LwM2M object represents a set of data that can be accessed through the LwM2M client. For example, in SPMP there are objects for control, monitoring, and alert management for ports. Some LwM2M objects are single instance, for example there can only be one Firmware Update object. Others can have multiple instances. An SPMP charger will have objects for each port. Each object defines a set of Resources, which is a set of related properties that the server can read, write, execute or subscribe to. The SPMP charger object will contain resources such as voltage and amperage averages. For more information on the LwM2M data model, see section 7 of the LwM2M specification.




Note: additional ports will have similar objects defined

## Telemetry Object (LwM2M Voltage Object)

| Name | Object ID | Object Version | LWM2M Version | | | | | |
|---|---|---|---|---|---|---|---|---|
| Voltage | 3316 | 1 | 1 | | | | | |
| Object URN | | Instances | Mandatory | | | | | |
| urn:oma:lwm2m:ext:3316 | | Multiple | Optional | | | | | |
| | | | | | | | | |
| ID | Name | Operations | Instances | Mandatory | Type | Range or Enumeration | Units | Description |
| 5700 | Sensor Value | Read | Single | Mandatory | Float | | | Last or Current Measured Value from the Sensor |
| 5701 | Sensor Units | Read | Single | Optional | String | | | Measurement Units Definition e.g. â€œCelâ€ for Temperature in Celsius |
| 5601 | Min Measured Value | Read | Single | Optional | Float | | | The minimum value measured by the sensor since power ON or reset |
| 5602 | Max Measured Value | Read | Single | Optional | Float | | | The maximum value measured by the sensor since power ON or reset |
| 5603 | Min Range Value | Read | Single | Optional | Float | | | The minimum value that can be measured by the sensor |
| 5604 | Max Range Value | Read | Single | Optional | Float | | | The maximum value that can be measured by the sensor |
| 5605 | Reset Min and Max Measured Values | E | Single | Optional | | | | Reset the Min and Max Measured Values to Current Value |
| 5821 | Current Calibration | Read/Write | Single | Optional | Float | | | Read or Write the current calibration coefficient |
| 5750 | Application Type | Read/Write | Single | Optional | String | | | The application type of the sensor or actuator as a string depending on the use case |

## Telemetry Object (LwM2M Current Object)

| Name | Object ID | Object Version | LWM2M Version | | | | | |
|---|---|---|---|---|---|---|---|---|
| Current | 3317 | 1 | 1 | | | | | |
| Object URN | | Instances | Mandatory | | | | | |
| urn:oma:lwm2m:ext:3317 | | Multiple | Optional | | | | | |
| | | | | | | | | |
| ID | Name | Operations | Instances | Mandatory | Type | Range or Enumeration | Units | Description |
| 5700 | Sensor Value | Read | Single | Mandatory | Float | | | Last or Current Measured Value from the Sensor |
| 5701 | Sensor Units | Read | Single | Optional | String | | | Measurement Units Definition e.g. degrees for Temperature in Celsius |
| 5601 | Min Measured Value | Read | Single | Optional | Float | | | The minimum value measured by the sensor since power ON or reset |
| 5602 | Max Measured Value | Read | Single | Optional | Float | | | The maximum value measured by the sensor since power ON or reset |
| 5603 | Min Range Value | Read | Single | Optional | Float | | | The minimum value that can be measured by the sensor |
| 5604 | Max Range Value | Read | Single | Optional | Float | | | The maximum value that can be measured by the sensor |
| 5605 | Reset Min and Max Measured Values | E | Single | Optional | | | | Reset the Min and Max Measured Values to Current Value |
| 5821 | Current Calibration | Read/Write | Single | Optional | Float | | | Read or Write the current calibration coefficient |
| 5750 | Application Type | Read/Write | Single | Optional | String | | | The application type of the sensor or actuator as a string depending on the use case |

## SPMP Port Status Object

| **Port Status/Monitor Object** | | | | |
|---|---|---|---|---|
| | | | | |
| **Name** | **Type** | **Mandatory** | **Instances** | **Operations** |
| Name | String | Mandatory | Single | Read/Write |
| Voltage Average | Float | Optional | Single | Read |
| Amperage Average | Float | Optional | Single | Read |
| Length | Integer | Mandatory | Single | Read/Write |
| Timestamp | Timestamp | Mandatory | Single | Read |
| Alerts | Object Link | Optional | Single | Read |

## SPMP Alerts Object

| **Alerts Object** | | | | |
|---|---|---|---|---|
| | | | | |
| **Name** | **Type** | **Mandatory** | **Instances** | **Operations** |
| Short circuit | Boolean | Optional | Single | Read |
| Over-voltage | Boolean | Optional | Single | Read |
| Over-current | Boolean | Optional | Single | Read |
| Low Voltage Disconnect | Boolean | Optional | Single | Read |
| Low Voltage Reconnect | Boolean | Optional | Single | Read |
| (others TBD / Custom) | Boolean | Optional | Single | Read |

## Smart Load (BTS) Object

| **BTS Object** | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| **Name** | **Type** | **Mandatory** | **Instances** | **Operations** | **Description** |
| Logging enabled | Boolean | | Single | RW | Enable or disable logging |
| Transceiver chain enabled | Boolean | | Single | RW | Enable or disable transceiver chain |
| Channel Bandwidth | Float | | Single | RW | Channel bandwidth (might be a per antenna/LTE/3G setting) |
| User limit | Integer | | Single | RW | Number of serviceable users |
| Backhaul enabled | Boolean | | Single | RW | Switch-off backhaul - local switching only. |
| LTE data enabled | Boolean | | Single | RW | Switch off data (LTE) |
| Edge data enabled | Boolean | | Single | RW | Switch off data (Edge) |
| GPRS data enabled | Boolean | | Single | RW | Switch off data (GPRS) |
| Software updates enabled | Boolean | | Single | RW | Ability to disable software updates |
| Emergency calls only | Boolean | | Single | RW | State where only emergency calls are allowed |
| Duty cycle | Integer | | Single | RW | 0-100, duty cycle of BTS. Is the period of the duty cycle configurable, or fixed for a given BTS? |
| Sleep | Boolean | | Single | RW | Puts the BTS to sleep, allowing manual wakeup at a later time |
| RF power output | Integer or Float? | | Single | RW | Reduce service area. Reduce RF power output as stepping function. |
| | | | | | |

### SPMP Rules Object

| **Smart Load Rules Object** | | | | |
|---|---|---|---|---|
| | | | | |
| **Name** | **Type** | **Mandatory** | **Instances** | **Operations** |
| Voltage Threshold | Float | Yes | Single | RW |
| Object ID | Integer | Yes | Single | RW |
| Object Instance ID | Integer | Yes | Single | RW |
| Resource ID | Integer | Yes | Single | RW |
| Value | Opaque | Yes | Single | RW |
| Type | String | Yes | Single | RW |

## Calibration

Voltage and Current Sensors require calibration due to inherent errors in analog-to-digital converters (ADC). This calibration can be stored in the respective calibration resource in the voltage and current sensor objects. If the unit it calibrated at the factory and the calibration values are stored on the unit, it can be backed up on the SPMP server.

## Control

The charger will instantiate a control object for each port present. The object will be an LwM2M Digital Output. Ports may be enabled or disabled by the server by performing a write operation on the “Digital Output State” resource of the object.

| **Port Object** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |

| Name | Type | Mandatory | Instances | Operations | | | | |
|---|---|---|---|---|---|---|---|---|
| Name | String | Yes | Single | R | | | | |
| Control | Object Link | Yes | Single | R | Link to Control Object | | | |
| Status | Object Link | Yes | Single | R | Link to Status Object | | | |
| Smart Load Configuration | Object Link | No | Single | RW | Link to smart load object | | | |
| Smart Load Status | Object Link | No | Single | RW | Link to lwm2m object mapping | | | |
| Voltage | Object Link | No | Single | R | Link to raw voltage object | | | |

## Authentication

Communications between the Device Management Server and the SPMP Device are secured using Datagram Transport Layer Security (DTLS). DTLS is a protocol that modified TLS to run over UDP. LWM2M supports using PreShared Keys, raw Public/Private keypairs, or X.509 Certificates.

## Frame format, Transport layer and encoding

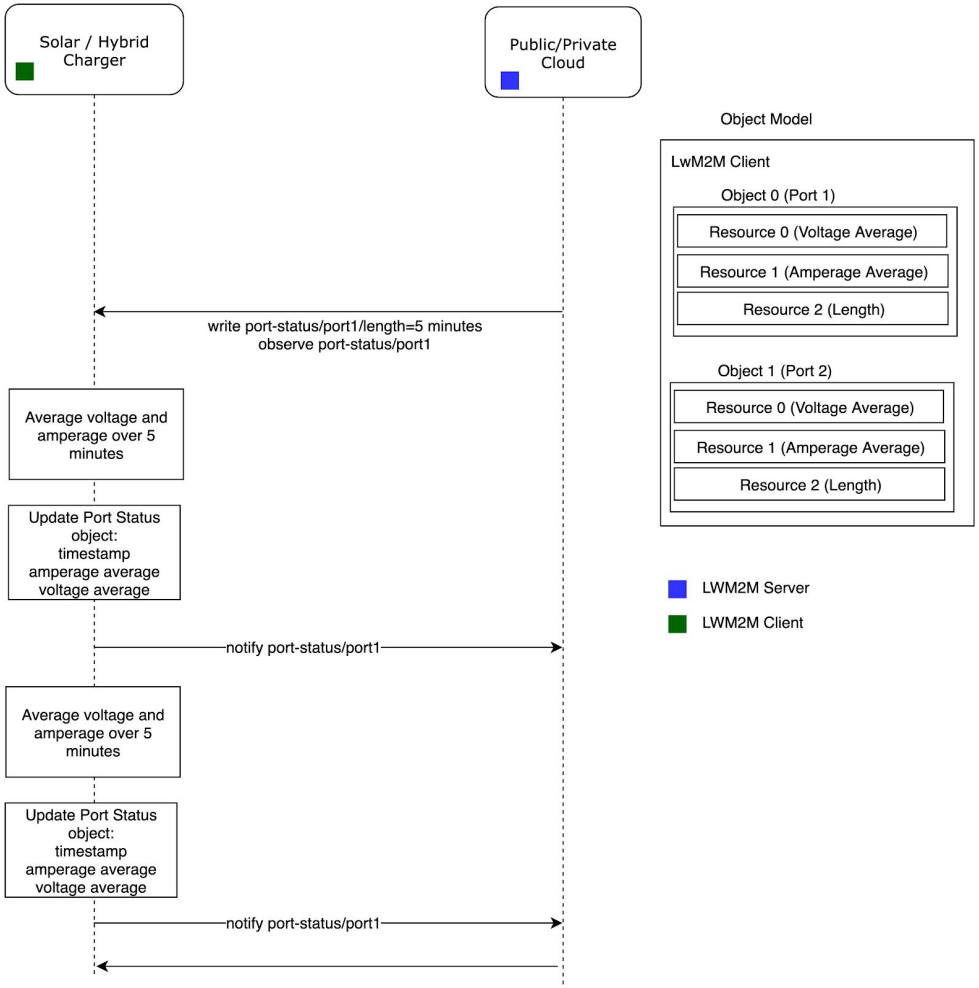
SPMP utilizes the Constrained Application Protocol (CoAP). CoAP is a specialized web transfer protocol that was designed specifically for M2M applications such as smart energy.

Additional details about this protocol can be found on the CoAP website.

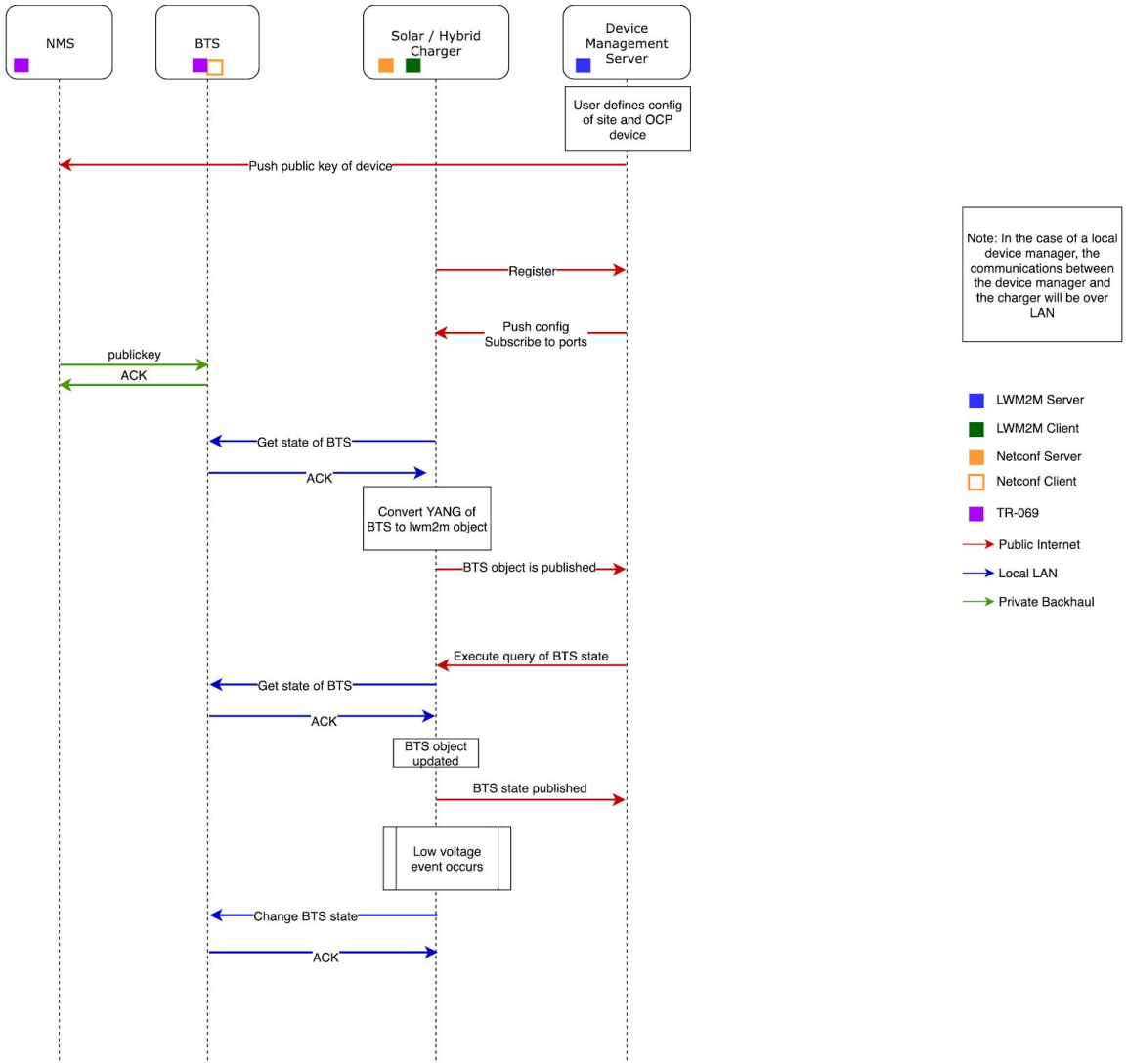
## Security and Threat Model

Details forthcoming in a future release

# Appendix




Sequence diagram for smart charge controller telemetry data to the public/private cloud




Sequence diagram between smart controller and smart load