

SURFACE VEHICLE RECOMMENDED PRACTICE

J2847-1

NOV2013

Issued 2010-06
Revised 2013-11

Superseding J2847-1 MAY2011

(R) Communication for Smart Charging of Plug-in Electric Vehicles
using Smart Energy Profile 2.0

RATIONALE

This document is being updated to address updates to the ZigBee Alliance and HomePlug Powerline Alliance Smart Energy Profile 2.0 (SEP 2.0) specification. The initial publication of SAE J2847/1 included the requirements from SAE J2836/1™ Use Cases and additional communication requirements are now mature enough to be included in this SAE document.

TABLE OF CONTENTS

1.	SCOPE.....	2
1.1	Purpose	2
2.	REFERENCES.....	2
2.1	Applicable Documents	2
2.2	Related Publications	3
3.	DEFINITIONS	3
4.	TECHNICAL REQUIREMENTS.....	5
4.1	System Definition	5
4.1.2	Mapping of Smart Energy Profile 2.0™ Function Sets to SAE Use Cases	6
4.2	Important Notes About Figures and Physical & Logical Devices.....	7
4.3	Application Access	7
4.3.1	Operation Sequences	8
4.4	Application of Messaging	9
4.4.1	Loss of Communications or Power	9
4.4.2	Pricing: Rates/Tariffs/Programs	9
4.4.3	U2 Direct Load Control Programs (Demand Response)	17
4.4.4	U3 Real Time Pricing (RTP)	24
4.4.5	U4 Critical Peak Pricing (CPP)	24
4.4.6	U5 Optimized Energy Transfer Programs.....	24
5.	NOTES.....	36
5.1	Marginal Indicia	36

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2013 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 877-606-7323 (inside USA and Canada)
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
http://www.sae.org

**SAE values your input. To provide feedback
on this Technical Report, please visit
http://www.sae.org/technical/standards/J2847/1_201311**

APPENDIX A	MAPPING OF SAE J2847/1-2011 MESSAGES TO SEP 2.0.....	37
APPENDIX B	DEFINITIONS FOR SAE J2847/1-2011 MESSAGES	41
FIGURE 1	FLOW RESERVATION PARAMETERS	26
TABLE 1	SMART PEV AND EVSE COMBINATIONS	6
TABLE 2	SEP2 FUNCTION SETS BY USE CASE.....	7
TABLE B-1	SMART PEV AND EVSE COMBINATIONS	42

1. SCOPE

This document describes the details of the Smart Energy Profile 2.0 (SEP2.0) communication used to implement the functionality described in the SAE J2836/1™ use cases. Each use case subsection includes a description of the function provided, client device requirements, and sequence diagrams with description of the steps. Implementers are encouraged to consult the SEP2.0 Schema and Application Specification for further details. Where relevant, this document notes, but does not formally specify, interactions between the vehicle and vehicle operator.

1.1 Purpose

The primary purpose of SAE J2847/1 is specifying the communications needed for *grid-optimized energy transfer for plug-in electric vehicles (PEV)* – that is, ensuring that vehicle operators have sufficient energy for driving while enabling the delivery of that energy to vehicles in ways that minimize stress upon the grid or reduce customer costs. This can be accomplished, for example, by vehicle owners' voluntary participation in a utility controlled-charging program in return for incentives, and the specification therefore supports information flows that enable such mechanisms.¹

This specification supports Forward Power Flow (FPF) energy transfer from the grid to the vehicle to charge the vehicle's rechargeable energy storage system (RESS). Implementation of SAE J2847/1 is encouraged for enabling utility- or premises Energy Management System-controlled charging load management and demand response incentive programs and vehicle communication with home area network (HAN) SEP2.0 communications-capable electrical devices.

Reverse Power Flow (RPF) is the transfer of power from the vehicle to the grid and is used to provide the utility sources of power to augment ancillary and regulatory services to control the reliability of the grid, such as to prevent grid outages. PEV to Utility Grid communications for RPF is specified in SAE J2847/3.

Beyond its primary purpose of energy transfer, SAE J2847/1 enables other applications between vehicles and the grid, such as vehicle participation in a utility-controlled charging plan (as noted above), or participation in a home-area network (HAN) of SEP2.0 communications-capable electrical devices.

2. REFERENCES

2.1 Applicable Documents

The following publications form a part of this specification to the extent specified herein. Unless otherwise indicated, the latest issue of SAE publications shall apply.

2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or 724-776-4970 (outside USA), www.sae.org.

J2836/1™ Use Cases for Communication Between Plug-in Vehicles and the Utility Grid

J1772™ SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler

¹ Such programs exist for other large electrical loads, such as air conditioners, and it is an explicit goal of the specification to support information flows that enable such mechanisms for plug-in vehicles. This will enable systems that distribute the economic and environmental benefits of plug-in vehicles to all market participants, thereby ensuring faster market development.

J2836/2™	Use Cases for Communication between Plug-in Vehicles and Off-Board DC Chargers
J2836/3™	Use Cases for Plug-in Vehicle Communication as a Distributed Energy Resource
J2836/5™	Use Cases for Communication between Plug-in Vehicles and their Customers
J2847/2	Communication between Plug-In Vehicles and Off-Board DC Chargers
J2847/3	Communication for Plug-in Vehicles as a Distributed Energy Resource
J2847/5	Communication between Plug-in Vehicles and their Customers
J2931/1	Digital Communications for Plug-in Electric Vehicles
J2931/4	Broadband PLC Communication for Plug-in Electric Vehicles

2.1.2 ZigBee Alliance Publication

Available from ZigBee Alliance, 2400 Camino Ramon, Suite 375, San Ramon, CA 94583
(<http://www.zigbee.org/Standards/Downloads.aspx>)

Smart Energy Profile 2.0 Application Protocol Specification

Smart Energy Profile 2.0 Web-Application Description Language

Smart Energy Profile 2.0 XML Schema Definition

Smart Energy Profile 2.0 Marketing Requirements Document

Smart Energy Profile 2.0 Technical Requirements Document

2.2 Related Publications

The following publications are provided for information purposes only and are not a required part of this SAE Technical Report.

2.2.1 UCA® International Users Group Publication

Available from UCA® International Users Group, 10604 Candler Falls Court, Raleigh, NC 27614 (www.ucaiug.org)

UCAIug Home Area Network System Requirements Specification

3. DEFINITIONS

3.1 ADVANCED METERING INFRASTRUCTURE (AMI)

AMI or Advanced Metering Infrastructure typically refers to the measurement and collection system that includes meters at the customer site, communication networks between the customer and a service provider, such as an electric, gas, or water utility, and data reception and management systems that make the information available to the service provider.

3.2 BATTERY ELECTRIC VEHICLE (BEV)

The BEV is a vehicle that receives its power solely from batteries, unlike a hybrid vehicle that may receive a portion of its power from a separately-fueled power source, such as an internal combustion engine (ICE).

3.3 CUSTOMER INFORMATION SYSTEM (CIS) – used in UTILITY definition

3.4 DEMAND RESPONSE LOAD CONTROL (DRLC) – used in UTILITY definition

3.5 ELECTRIC UTILITY POWER SYSTEM (UTILITY)

The system that generates and delivers commercial electrical power to a residential or commercial building or facility. It extends to and includes a billing apparatus (electric meter).

3.6 ELECTRIC UTILITY/LOCAL LOAD MANAGEMENT SYSTEM (LMS)

A system that is responsible to monitor and control the load on some portion of the Utility or local premises' feeder and branch circuits. The goal of control may be to prevent overload or to reduce the cost of energy based on a specific billing agreement.

3.7 ELECTRIC VEHICLE STORAGE BATTERY (BATTERY)

A group of electrochemical cells electrically connected in a series and/or parallel arrangement, the principal purpose of which is to provide DC electrical energy to propel the EV. Means a system that stores energy for delivery of electric energy and which is rechargeable. The terms Battery, ESS, and RESS are often used interchangeably.

3.8 ELECTRIC VEHICLE SUPPLY EQUIPMENT (EVSE)

The equipment from the branch circuit to, and including, the connector that couples to the electric vehicle inlet, the purpose of which is to transfer electric energy to an EV. This equipment is located off-board the vehicle.

3.9 END-USE-MEASUREMENT-DEVICE (EUMD)

The End-Use-Measurement-Device (EUMD) is a revenue-grade submeter responsible for directly measuring energy delivered to a PEV. If a utility or electricity vendor wishes to offer a rate-advantaged program that requires specific metering of energy to a PEV, an EUMD must be present in the appropriate circuit. The physical form, location and ownership of the EUMD may be unique for different applications.

3.10 ENERGY MANAGEMENT SYSTEM (EMS)

The Energy Management System refers to the device(s) or software that controls the energy transfer and use of smart loads throughout the home or premises. The EMS may take several different physical forms. For example, the utility itself may serve as the EMS, dispatching data and commands to loads in customers' homes. Conversely, the EMS may be a premises-installed device or software (ex. local management in a public building).

3.11 ENERGY PORTAL

Energy Portal is a 120 V standard outlet.

3.12 ENERGY TRANSFER

The process of flowing energy to the EV from the EVSE.

3.13 ESI

A device, with multiple network interfaces, which is a member of both the home smart 846 energy network and a service provider's private network. This is the primary mechanism for 847 the service provider to contribute data and directives into the smart energy network and to 848 receive responses from smart energy devices. - used in the utility definition

3.14 HOME AREA NETWORK (HAN)

A HAN is an energy related network, contained within a premises used for communicating with devices within the premises. HANs do not necessarily require connectivity outside the premises, but may be connected to one or more external communication networks (e.g., Utility AMI, internet, cell phone network, etc.) using gateways, bridges and interfaces.

3.15 HUMAN MACHINE INTERFACE (HMI)

HAN application characteristics that provide local user input and/or output. These are based constrained and based on the data type.

- User Input - Provides consumers with a means to input data into an Application (e.g., Touch screen, Keypad)
- User Output - Provides an Application with a means to output data to the consumer (e.g., In-Home Display, text message)

3.16 IN-HOME DISPLAY

Refers generically to a device positioned somewhere in the home or premises that is used to display information to the customer. Information displayed may include, but is not limited to any combination of the following: home/premises energy/power usage, device energy/power usage, energy cost, estimated charge times, estimated completion times, historical charge data, etc. The display device may be a stand-alone unit, integrated into another communications-enabled device (e.g., a display on a smart meter, EVSE, or even a PC).

3.17 RECHARGEABLE ENERGY STORAGE SYSTEM (RESS)

An energy storage system for delivery of electric energy and which is rechargeable. The terms Battery, ESS, and RESS are often used interchangeably.

3.18 REVERSE POWER FLOW (RPF)

Reverse Power Flow means the direction of energy is flowing from the vehicle back to the grid.

3.19 SMART CHARGING

Smart Charging is a communication system in which PEVs optimize the combination of charging schedule, customer preferences and grid capacity. Smart Charging can include participation in utility programs that include time of use rates, demand response, or regulation services.

3.20 UTILITY

Utility provides electrical energy and typically refers to a collection of systems that include the Customer Information System (CIS), the Advanced Metering Infrastructure (AMI), Rates and Revenue. The utility makes available to PEV through the ESI pricing tables, DRLC or discrete events. The utility also supplies information such as tariff rate, interval for metered kWhr consumption, and validation of PEV program for PEV ID, etc.

3.21 V2G Power Transfer

When vehicle power is fed into the electric grid, we refer to it as "Vehicle-to-Grid" power, or V2G.

4. TECHNICAL REQUIREMENTS

4.1 System Definition

The messages within this document apply when the customer enrolls in one or more the following Utility programs. The use cases that describe these are per SAE J2836/1™.

U1: Time-Of-Use (TOU) Rates / Tariffs / Programs (Load Shifting)

U2: Direct Load Control Programs (Demand Response)

U3: Real Time Pricing (RTP: Load Shifting / Demand Response) (Active Management)

U4: Critical Peak Pricing (CPP / Load Shifting)

U5: Optimized Energy Transfer Programs (Demand Response, Regulation Services, etc.)

Smart Energy Profile 2.0 (SEP2.0) offers a host of different communications structures and resources, and is meant to be applicable to a wide spectrum of devices throughout the Home Area Network (HAN) environment. This document refers to and makes use of only the portions of SEP2.0 relevant to the application of the five use cases described above. For full and complete information regarding SEP2.0 implementation, including requirements, schemas, and certification, please refer to the source SEP 2.0 documents (see References).

It should be noted that multiple Use Cases may be implemented at once. For example, it is possible to have a customer enrolled in both Time-of-Use (U1) and Load Control/Demand Response (U2) programs. In this case, all messaging and requirements defined in this section for each of those use cases must be supported simultaneously by the PEV. The messages within this document are also applicable when a PEV is not directly enrolled in a utility program (e.g., for customer charging management or facility demand management).

4.1.1.1 End Device Control

Implementations of electric charging with SAE J2847/1 messages support the control of the charging session (determination of charge times, user's preferences, etc.). This control may reside either in the EVSE or the PEV. In the case where both devices support SAE J2847/1 messages, and are capable of managing the session, it has been assumed that the PEV will be the default controlling device. There are various ways for this to be accomplished, all of which are determined by the functionality of the EVSE or preferences of the EVSE owner.

When an SEP 2.0 enabled PEV wishes to connect to an SEP 2.0 network, it must follow the same sequence to access SEP 2.0 resources that other devices do. This involves an out of band enrollment, certificate exchange, and device registration process (See 4.3 Application Access). This process is how SEP 2.0 devices join and 'advertise' their capabilities on an SEP 2.0 network (similar to exchanging a message saying "Smart PEV Present" or "SEP 2.0 Device Present"). However, where control of a session ultimately resides depends on the functionality of the EVSE and the preference of the EV owner. Whether the EVSE bridges SEP 2.0 messages from the PEV to the ESI/EMS, acts as the SEP 2.0 server or gateway (e.g., mirrors messages from the ESI/EMS), or retains control depends on the EVSE being used in the charging session. No matter how it is accomplished, when an EVSE and a PEV SEP are both SEP 2.0 devices, an EVSE shall be able to allow the PEV to manage the charging session.

The table below indicates the expected outcomes for the various scenarios. It is advisable for vehicle manufacturers to include an easily accessible option (Smart Charging OVERRIDE or CHARGE NOW function) for selecting immediate charging. The maximum PEV charging rate is limited by either the SAE J1772™ Control pilot pulse width or internal controls (i.e., battery management system), whichever is lower.

TABLE 1 – SMART PEV AND EVSE COMBINATIONS

Case	PEV Supports SAE J2847/1 Messaging	EVSE Supports SAE J2847/1 Messaging	Result
1	NO	NO	No "smart charging" support.
2	NO	YES	EVSE is "smart charging" enabled, and computes charge schedule. EVSE manipulates charging by controlling SAE J1772™ pilot.
3	YES	NO	PEV is "smart charging" enabled, and computes charge schedule.
4	YES	YES	Both devices are "smart charging" capable. Upon completion of Application Access processes (see section 4.3), and EVSE shall be able to relinquish control of the charging session to the PEV.

4.1.2 Mapping of Smart Energy Profile 2.0™ Function Sets to SAE Use Cases

The following table describes which SEP2.0 Function Sets are required to achieve the functionality as described in the SAE J2836-1 Use Cases. The fields marked in the table represent only the bare minimum Function Sets required to implement the functionality described in the Use Cases. It should be remembered that it is possible for multiple Use Cases to be implemented in parallel, or for additional functionality not discussed within this document to be applied in parallel to any Use Case functions described, which could result in additional Function Sets being employed. A simple example of this could be the Messaging Function Set - while not *required* to implement the functionality described in any of the Use Cases, it could be used in addition to the required Function Sets to provide messaging to the consumer.

TABLE 2 – SEP2 FUNCTION SETS BY USE CASE

Function Sets/Resources	Registration/	J2836-1					Comments
	Enrollment	U1	U2	U3	U4	U5	
1. Billing							Not addressed in the current revision of J2847/1
2. Demand Response / Load Control			X				
3. Support and Common Resources	X	X	X	X	X	X	Includes Self Device/End Device, (used for registration and enrollment), Time, Device Information, Power Status, Function Set Assignments, Subscription/Notification, Response, Network Status, Log Events
4. Distributed Energy Resource						X	This function set is employed in J2847-3 (Reverse Power Flow).
5. Software Download							
6. Messaging							
7. Metering/Meter Mirroring		X					The Metering Function Set may be employed in any of the Use Cases if the EUMD is within the PEV but is required for specific EV TOU programs
8. Prepayment							
9. Pricing		X		X	X		
10. Flow Reservation						X	
11. Response		X	X	X	X	X	

4.2 Important Notes About Figures and Physical & Logical Devices

The following sections of this document contain several figures which illustrate example message flows between devices. In most of these figures, the devices are defined by their logical function, with their *most likely* physical form listed in brackets. For example, “DRLC Client (PEV)” means that the logical function of the device or that is being performed is a DRLC Client, while the *most likely* physical form or location of this function is within the PEV. The references to physical devices are meant to illustrate only the simplest or most common application and are meant to improve the reader's understanding.

However, it is important to note that the physical device shown in brackets in which the logical function resides may be something other than what is described in each figure. For example, while in the simplest and perhaps most likely arrangement, most of the server/ESI functions described are likely to reside in a “smart meter” in a home, they may also reside in another device in other applications, such as perhaps an Energy Management System (EMS) in a public or commercial parking structure. Similarly, while the PEV is envisioned as the simplest and most likely physical form of the SEP2.0 device or client, these functions could also be implemented into an “intelligent” EVSE or other device.

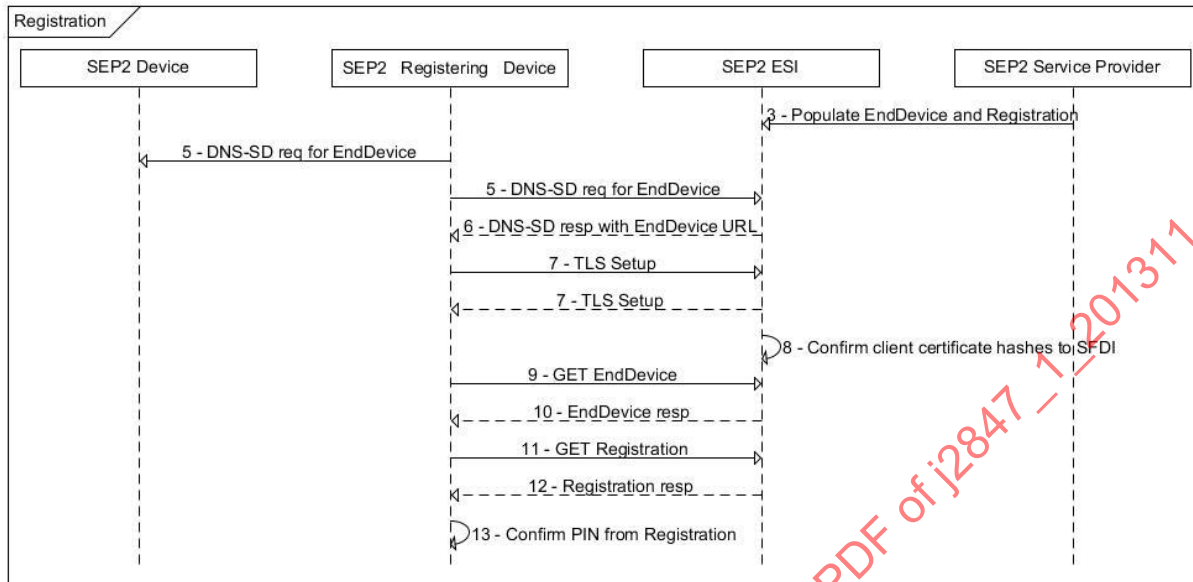
4.3 Application Access

Once authenticated and authorized through network access, a node can freely communicate in the network at the network layer. However, authentication and authorization at the application layer normally needs to occur to allow the access of resources as a client host. Registration with a utility or third party service provider may also be needed to provide explicit device and user authorization at the application layer. Further information related to application authentication and authorization can be found in the SEP 2.0 Application Specification.

4.3.1 Operation Sequences

This flow describes client device registration to the customer HAN and is assumed a prerequisite to implement each use case.

Sequence diagram:



NOTE: The Registering Device could be located in the same physical device as the ESI, it is logically represented separately in this diagram.

Step descriptions:

Step	Description
1	At time of device manufacture/packaging Short Form Device Identification (SFDI) and/or QR-Code of the client device certificate is printed on client device packaging. Registration PIN is printed on the client device packaging.
2	(Out of band) User establishes secured and authenticated session with desired service provider. This may be a password protected IVR system, web site, etc. SFDI and/or QR-Code is delivered to the service provider.
3	(Out of band) Service provider populates client device's EndDevice (containing SFDI) and Registration (containing PIN) resources to appropriate HAN registration server (typically the ESI).
	<i>Begin client boot into HAN</i>
4	Client device joins the HAN (layer 2)
5	Client issues DNS-SD request to locate its EndDevice (keyed by its SFDI)
6	ESI provides DNS-SD response with URL to client's EndDevice.
7	Clients connects to ESI, performing TLS and client authentication. Note that client certificate is sent in the clear and thus SFDI can be determined. <i>Client and server now have an encrypted connection. Each has determined it is talking to an authenticated SEP2 device, but have not confirmed they are talking to the correct specific SEP2 device.</i>
8	ESI verifies client certificate hashes to the client SFDI <i>ESI has determined it is talking to the correct client.</i>
9	Client GET's its EndDevice from ESI Client sends the following request: <full HTTP request example here>

Step	Description
10	ESI responds with the EndDevice resource. Server sends the following response: <full HTTP response example here>
11	Client GET's its Registration (containing its PIN) from the ESI. Client sends the following request: <full HTTP request example here>
12	ESI responds with the Registration resource. Note: the PIN is thus transmitted over a secure channel. Server sends the following response: <full HTTP response example here>
13	Client verifies its PIN versus that provided by the ESI. <i>Client has verified it is connected to the correct HAN</i>

4.4 Application of Messaging

This section describes the details of the messages required to implement the functionality described in the SAE J2836/1™ use cases using SEP2.0.

In the context of Source and Destination listed for each message below, it is important to note that 'Utility' does not necessarily refer literally to the generating company, but rather generically to a grouping of entities responsible for providing services to the premises, which may include one or more of the utility, transmission operator, aggregator, energy services companies, etc..

4.4.1 Loss of Communications or Power

In the event that reliable data communication between the PEV/EVSE and Energy Management System cannot be established or are interrupted during the session, but power remains, it is understood that the PEV/EVSE shall have the discretion to revert to a default state in which it charges at the maximum rate allowed by the SAE J1772™ Pilot signal.

In the event that power is lost during the charging session, the PEV/EVSE may resume charging upon restoration of power, subject to the requirements in SAE J2894. SAE J2894 requirements must be met to prevent large, abrupt load swings, and reduce stress on the electrical infrastructure.

4.4.2 Pricing: Rates/Tariffs/Programs

The set of Pricing messages are intended to allow the customer to send a request to a utility or ESI for current and future electricity pricing information, upon which the customer or vehicle may make a decision about when to charge the vehicle. Some utilities or energy providers may wish to offer different unit costs for energy at different times during the day, and this set of messages is used to communicate those periods and rates. It is understood that the customer shall be permitted the option to charge at any time irrespective of pricing, but that doing so may result in a higher electricity cost to the customer, and/or non-compliance with a utility rate program (if the customer is registered in one) that may result in an increased cost the customer.

The following device requirements and operational sequences are derived from the Smart Energy Profile 2.0 Application Specification. The device requirements section captures normative description of Pricing client device behavior, this is simply a summary, additional detail and application guidelines can be found within the application specification. The operational sequence section will cover three basic operations of Pricing functionality.

4.4.2.1 U1 Time of Use, U3 Real Time Pricing, U4 Critical Peak Pricing

These three SAE pricing use cases can be achieved using the SEP 2.0 Pricing Function Set, though only U1 Time of Use, is represented here. Access to certain prices/tariffs might be dependent on program enrollment. Pricing Implementation Guidelines can be found in section 12.4 and section 17 of the SEP 2.0 Application Specification. Note, objects will vary for different pricing programs, only an example of one will be shown below.

4.4.2.1.1 Client Device Requirements

The following requirements are derived from the Pricing Function Set (Section 12.4) of the SEP2.0 specification. These requirements capture the baseline functionality required for Pricing client devices as defined by SEP2.0, however, they can be modified as seen fit by the SAE community.

A SHALL below defines required functionality while a SHOULD defines optional functionality.

Client devices of the Pricing function set SHALL:

- Implement and support the Time function set as a client
- Implement and support the Response function set as a client
- Be able to internally store and support at least 2 unique Pricing Events per RateComponent
- Poll the Pricing Event list (TimeTariffIntervalList) when not subscribed, in order to stay updated on new or canceled events. It is RECOMMENDED that polling pricing client devices request updated information for pending TimeTariffInterval instances just prior to those TimeTariffInterval instances becoming active.
- Provide Response acknowledgments to the Pricing server for events requiring a response
- Read the Status of a Pricing Event (TimeTariffInterval) before execution of the event, if the event was scheduled in the past and the client is not subscribed to the event

Client devices of the DRLC function set SHOULD:

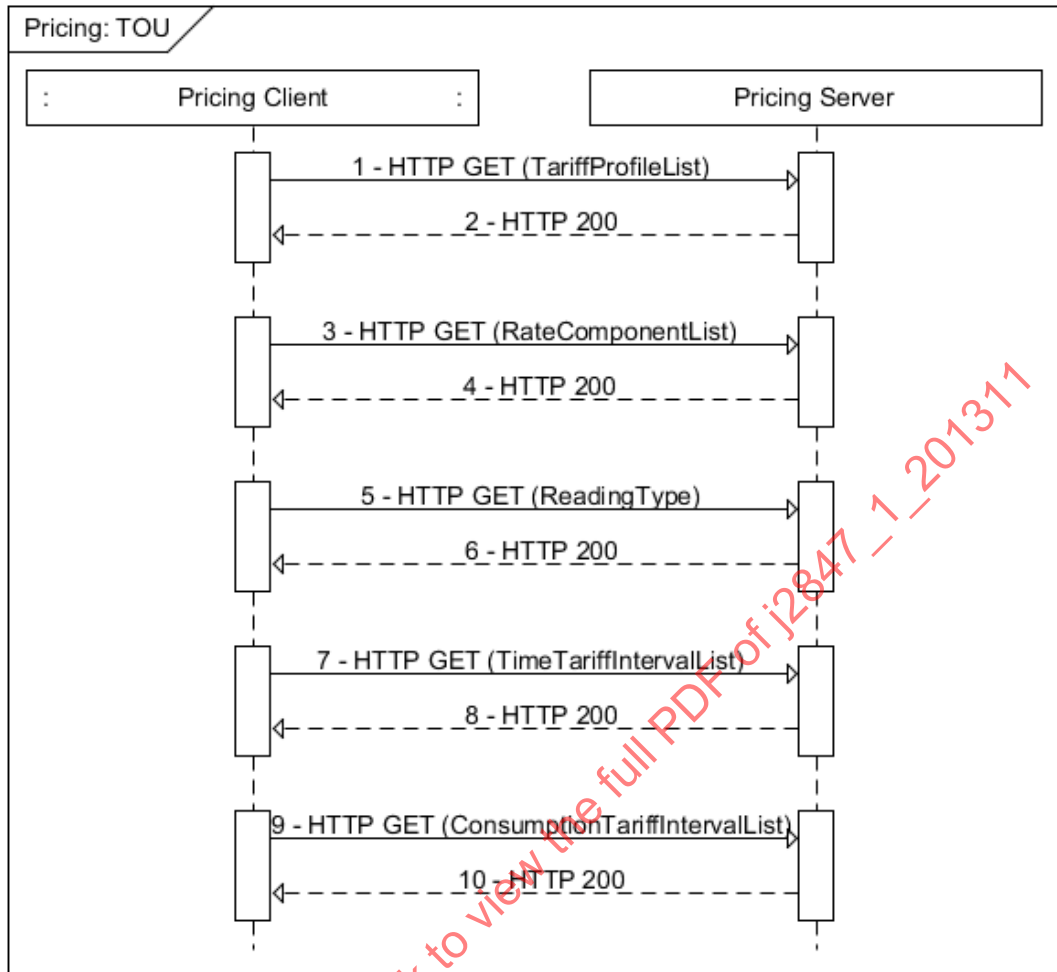
- Support resource subscription mechanisms
- Be able to internally store and support at least 5 unique Pricing Events per RateComponent
- Prioritize local storage of events and give preference to events with start times in the near future
- Poll the Pricing Event list at a minimum of every 15 minutes when not subscribed, in order to stay updated on new or canceled events

4.4.2.2 Operation Sequences

The following sequence diagrams use semantics from the SEP2.0 specification; therefore it is important to first provide a high-level definition of the resources and objects found in the Pricing function set of SEP2.0.

#	SEP2 Resource/Object	Description	Subscription
1	TariffProfileList URI: /tp	A list of TariffProfiles.	Yes
2	RateComponentList URI: /tp/{#}/rc	A list of RateComponents. A RateComponent allows a service provider to define the cost of a particular rate component, such as consumption or generation.	No
3	TariffProfile URI: /tp/{#}	A schedule of charges (a "rate") that allows definitions of ToU tiers. Contains the currency type.	No
4	TimeTariffIntervalList URI: /tp/{#}/rc/{#}/tti	A list of TimeTariffIntervals. Each TimeTariffInterval defines a specific pricing event, including the ToU tier, indicating the relative price. Multiple TimeTariffIntervals take effect at different times to define multiple ToU tiers (for example. Peak, Mid-Peak, and Off-Peak)	Yes
5	ConsumptionTariffIntervalList URI: /tp/{#}/rc/{#}/tti/{#}/cti	List of consumption blocks hosted by the server. Consumption blocks define thresholds at which aggregated usage across the billing period changes the actual price.	No
6	RateComponent URI: /tp/{#}/rc/{#}	Specifies the applicable charges for a single component of the rate	No

4.4.2.2.1 Time-of-Use Pricing Event with Confirmation Required



NOTE: In most cases, registration is required to obtain access to pricing, and the client is directed to a specific TariffProfile through a Link in the FunctionSetAssignments found in their EndDevice FunctionSetAssignmentsLink, or from a different function set such as Billing.

Step	Description
1	Client GETs the TariffProfile from the Pricing Server. Client sends the following request: <pre>GET /tp/3 HTTP/1.1 Host: {hostname} Accept: application/sep+xml</pre>
2	Pricing server responds with the TariffProfile. Server sends the following response: <pre>HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength}</pre> <pre><TariffProfile href="/tp/3" xmlns="http://zigbee.org/sep"> <mRID>799794f4620b17e00000e566</mRID> <description>PEV TOU Rate</description> <currency>840</currency> <pricePowerOfTenMultiplier>-6</pricePowerOfTenMultiplier> <primacy>0</primacy> <rateCode>TOU-D-PEV Baseline 6</rateCode> <RateComponentListLink all="1" href="/tp/3/rc"/> <serviceCategoryKind>0</serviceCategoryKind> </TariffProfile></pre>

Step	Description
3	Client GETs the RateComponentList from the Pricing Server. Client sends the following request: <pre>GET /tp/3/rc?l=1 HTTP/1.1 Host: {hostname} Accept: application/sep+xml</pre>
4	Pricing server responds with the RateComponentList. Server sends the following response: <pre>HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength}</pre> <pre><RateComponentList all="1" href="/tp/3/rc" results="1" xmlns="http://zigbee.org/sep"> <RateComponent href="/tp/3/rc/3"> <mRID>fc000b07143d24fc0000e566</mRID> <description>TOU-D-PEV</description> <ActiveTimeTariffIntervalListLink all="0" href="/tp/3/rc/3/acttti"/> <flowRateEndLimit> <multiplier>0</multiplier> <unit>38</unit> <value>400</value> </flowRateEndLimit> <flowRateStartLimit> <multiplier>0</multiplier> <unit>38</unit> <value>0</value> </flowRateStartLimit> <ReadingTypeLink href="/rt/1"/> <roleFlags>12</roleFlags> <TimeTariffIntervalListLink all="5" href="/tp/3/rc/3/tti"/> </RateComponent> </RateComponentList></pre>
5	Client GETs the ReadingType from the Pricing Server. Client sends the following request: <pre>GET /rt/1 HTTP/1.1 Host: {hostname} Accept: application/sep+xml</pre>
6	Pricing server responds with the ReadingType. Server sends the following response: <pre>HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength}</pre> <pre><ReadingType href="/rt/1" xmlns="http://zigbee.org/sep"> <accumulationBehaviour>4</accumulationBehaviour> <commodity>1</commodity> <dataQualifier>12</dataQualifier> <flowDirection>1</flowDirection> <intervalLength>3600</intervalLength> <kind>12</kind> <numberOfConsumptionBlocks>1</numberOfConsumptionBlocks> <numberOfTouTiers>3</numberOfTouTiers> <phase>0</phase> <powerOfTenMultiplier>3</powerOfTenMultiplier> <tieredConsumptionBlocks>false</tieredConsumptionBlocks> <uom>72</uom> </ReadingType></pre>
7	Client GETs the TimeTariffIntervalList from the Pricing Server. Client sends the following request: <pre>GET /tp/3/rc/3/tti?l=5 HTTP/1.1 Host: {hostname} Accept: application/sep+xml</pre>

Step	Description
8	Pricing server responds with the TimeTariffIntervalList. Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <TimeTariffIntervalList all="5" href="/tp/3/rc/3/tti" results="5" subscribable="1" xmlns="http://zigbee.org/sep"> <TimeTariffInterval href="/tp/3/rc/3/tti/5" subscribable="1"> <mRID>ef06fa23dc0a0f650000e566</mRID> <description>Off-Peak 1</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <duration>28800</duration> <start>1357516800</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/5/cti"/> <touTier>1</touTier> </TimeTariffInterval> <TimeTariffInterval href="/tp/3/rc/3/tti/6" subscribable="1"> <mRID>41fc7c07e16820770000e566</mRID> <description>Mid-Peak 1</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <duration>14400</duration> <start>1357545600</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/6/cti"/> <touTier>2</touTier> </TimeTariffInterval> <TimeTariffInterval href="/tp/3/rc/3/tti/7" subscribable="1"> <mRID>63eed7b30c1c87a40000e566</mRID> <description>On-Peak 1</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <duration>21600</duration> <start>1357552800</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/7/cti"/> <touTier>3</touTier> </TimeTariffInterval> <TimeTariffInterval href="/tp/3/rc/3/tti/8" subscribable="1"> <mRID>9b04f0713e9212d90000e566</mRID> <description>Mid-Peak 2</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> </pre>

Step	Description
	<pre> <duration>18000</duration> <start>1357574400</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/8/cti"/> <touTier>2</touTier> </TimeTariffInterval> <TimeTariffInterval href="/tp/3/rc/3/tti/9" subscribable="1"> <mRID>c13c8755dc39b5950000e566</mRID> <description>Off-Peak 2</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>false</potentiallySuperseded> </EventStatus> <interval> <duration>10800</duration> <start>1357592400</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/9/cti"/> <touTier>1</touTier> </TimeTariffInterval> </TimeTariffIntervalList> </pre>
9	Client GETs a ConsumptionTariffIntervalList from the Pricing Server (note that there is a ConsumptionTariffIntervalList for each TimeTariffInterval, but only one is shown below.) Client sends the following request: <pre> GET /tp/3/rc/3/tti/5/cti?l=1 HTTP/1.1 Host: {hostname} Accept: application/sep+xml </pre>
10	Pricing server responds with the ConsumptionTariffIntervalList. Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <ConsumptionTariffIntervalList all="1" href="/tp/3/rc/3/tti/5/cti" results="1" xmlns="http://zigbee.org/sep"> <ConsumptionTariffInterval href="/tp/3/rc/3/tti/5/cti/1"> <consumptionBlock>1</consumptionBlock> <price>113000</price> <startValue>0</startValue> </ConsumptionTariffInterval> </ConsumptionTariffIntervalList> </pre>
Step	Description
1	Client GETs the TariffProfile from the Pricing Server. Client sends the following request: <pre> GET /tp/3 HTTP/1.1 Host: {hostname} Accept: application/sep+xml </pre>
2	Pricing server responds with the TariffProfile. Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <TariffProfile href="/tp/3" xmlns="http://zigbee.org/sep"> <mRID>799794f4620b17e00000e566</mRID> <description>PEV TOU Rate</description> <currency>840</currency> <pricePowerOfTenMultiplier>-6</pricePowerOfTenMultiplier> </pre>

Step	Description
	<pre> <primacy>0</primacy> <rateCode>TOU-D-PEV Baseline 6</rateCode> <RateComponentListLink all="1" href="/tp/3/rc"/> <serviceCategoryKind>0</serviceCategoryKind> </TariffProfile> </pre>
3	Client GETs the RateComponentList from the Pricing Server. Client sends the following request: <pre> GET /tp/3/rc?l=1 HTTP/1.1 Host: {hostname} Accept: application/sep+xml </pre>
4	Pricing server responds with the RateComponentList. Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <RateComponentList all="1" href="/tp/3/rc" results="1" xmlns="http://zigbee.org/sep"> <RateComponent href="/tp/3/rc/3"> <mRID>fc000b07143d24fc0000e566</mRID> <description>TOU-D-PEV</description> <ActiveTimeTariffIntervalListLink all="0" href="/tp/3/rc/3/acttti"/> <flowRateEndLimit> <multiplier>0</multiplier> <unit>38</unit> <value>400</value> </flowRateEndLimit> <flowRateStartLimit> <multiplier>0</multiplier> <unit>38</unit> <value>0</value> </flowRateStartLimit> <ReadingTypeLink href="/rt/1"/> <roleFlags>12</roleFlags> <TimeTariffIntervalListLink all="5" href="/tp/3/rc/3/tti"/> </RateComponent> </RateComponentList> </pre>
5	Client GETs the ReadingType from the Pricing Server. Client sends the following request: <pre> GET /rt/1 HTTP/1.1 Host: {hostname} Accept: application/sep+xml </pre>
6	Pricing server responds with the ReadingType. Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <ReadingType href="/rt/1" xmlns="http://zigbee.org/sep"> <accumulationBehaviour>4</accumulationBehaviour> <commodity>1</commodity> <dataQualifier>12</dataQualifier> <flowDirection>1</flowDirection> <intervalLength>3600</intervalLength> <kind>12</kind> <numberOfConsumptionBlocks>1</numberOfConsumptionBlocks> <numberOfTouTiers>3</numberOfTouTiers> <phase>0</phase> <powerOfTenMultiplier>3</powerOfTenMultiplier> <tieredConsumptionBlocks>false</tieredConsumptionBlocks> <uom>72</uom> </ReadingType> </pre>

Step	Description
7	Client GETs the TimeTariffIntervalList from the Pricing Server. Client sends the following request: <pre>GET /tp/3/rc/3/tti?l=5 HTTP/1.1 Host: {hostname} Accept: application/sep+xml</pre>
8	Pricing server responds with the TimeTariffIntervalList. Server sends the following response: <pre>HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength}</pre> <pre><TimeTariffIntervalList all="5" href="/tp/3/rc/3/tti" results="5" subscribable="1" xmlns="http://zigbee.org/sep"> <TimeTariffInterval href="/tp/3/rc/3/tti/5" subscribable="1"> <mRID>ef06fa23dc0a0f650000e566</mRID> <description>Off-Peak 1</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <duration>28800</duration> <start>1357516800</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/5/cti"/> <touTier>1</touTier> </TimeTariffInterval> <TimeTariffInterval href="/tp/3/rc/3/tti/6" subscribable="1"> <mRID>41fc7c07e16820770000e566</mRID> <description>Mid-Peak 1</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <duration>14400</duration> <start>1357545600</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/6/cti"/> <touTier>2</touTier> </TimeTariffInterval> <TimeTariffInterval href="/tp/3/rc/3/tti/7" subscribable="1"> <mRID>63eed7b30c1c87a40000e566</mRID> <description>On-Peak</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <duration>21600</duration> <start>1357552800</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/7/cti"/> <touTier>3</touTier> </TimeTariffInterval> <TimeTariffInterval href="/tp/3/rc/3/tti/8" subscribable="1"> <mRID>9b04f0713e9212d90000e566</mRID> <description>Mid-Peak 2</description></pre>

Step	Description
	<pre> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <duration>18000</duration> <start>1357574400</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/8/cti"/> <touTier>2</touTier> </TimeTariffInterval> <TimeTariffInterval href="/tp/3/rc/3/tti/9" subscribable="1"> <mRID>c13c8755dc39b5950000e566</mRID> <description>Off-Peak 2</description> <creationTime>1357430400</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1357430400</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <duration>10800</duration> <start>1357592400</start> </interval> <randomizeDuration>300</randomizeDuration> <randomizeStart>300</randomizeStart> <ConsumptionTariffIntervalListLink all="1" href="/tp/3/rc/3/tti/9/cti"/> <touTier>1</touTier> </TimeTariffInterval> </TimeTariffIntervalList> </pre>
9	Client GETs a ConsumptionTariffIntervalList from the Pricing Server (note that there is a ConsumptionTariffIntervalList for each TimeTariffInterval, but only one is shown below.) Client sends the following request: <pre> GET /tp/3/rc/3/tti/5/cti?l=1 HTTP/1.1 Host: {hostname} Accept: application/sep+xml </pre>
10	Pricing server responds with the ConsumptionTariffIntervalList. Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <ConsumptionTariffIntervalList all="1" href="/tp/3/rc/3/tti/5/cti" results="1" xmlns="http://zigbee.org/sep"> <ConsumptionTariffInterval href="/tp/3/rc/3/tti/5/cti/1"> <consumptionBlock>1</consumptionBlock> <price>113000</price> <startValue>0</startValue> </ConsumptionTariffInterval> </ConsumptionTariffIntervalList> </pre>

4.4.3 U2 Direct Load Control Programs (Demand Response)

The set of Demand Response /Load Control (DRLC) messages are intended to allow the utility to send a request to an EVSE/PEV to reduce or shed its load (i.e., reduce the rate of charging or cease charging entirely) whenever the utility identifies a present or anticipated future need for a reduction in the load. It is understood that DRLC events would be a relatively infrequent occurrence and would be used only when the anticipated load on the utility approaches critical levels. It is further understood that the customer shall be permitted the option to decline to comply with DRLC requests, but that doing so may result in a non-compliance with a utility rate program (if the customer is registered in one), which may result in an increased cost to the customer.

The following device requirements and operational sequences are derived from the Smart Energy Profile 2.0 Application Specification. The device requirements section captures normative description of DRLC client device behavior, this is simply a summary, additional detail and application guidelines can be found within the application specification. The operational sequence section will cover three basic operations of DRLC functionality, these include:

- Load Control Event with confirmation required
- Canceled Load Control Event
- Subscription to Load Control Event list

Application Guidelines and further information related to the DRLC Function set can be found in section 12.2 of the SEP 2.0 Application Specification

4.4.3.1 DRLC Client Device Requirements

The following requirements are derived from the DRLC Function Set (Section 12.2) of the SEP2.0 specification. These requirements capture the baseline functionality required for DRLC client devices as defined by SEP2.0, however, they can be modified as seen fit by the SAE community.

A SHALL below defines required functionality while a SHOULD defines optional functionality.

Client devices of the DRLC function set SHALL:

- Implement and support the Time function set
- Implement and support the Response function set
- Be able to internally store and support at least 3 unique Load Control Events
- Poll the Load Control Event list at a minimum of every 15 minutes when not subscribed, in order to stay updated on new or canceled events
- Provide acknowledgments to the DRLC server for events requiring a response
- Read the Status of a Load Control Event before execution of the event, if the event was scheduled in the past and the client is not subscribed to the event

Client devices of the DRLC function set SHOULD:

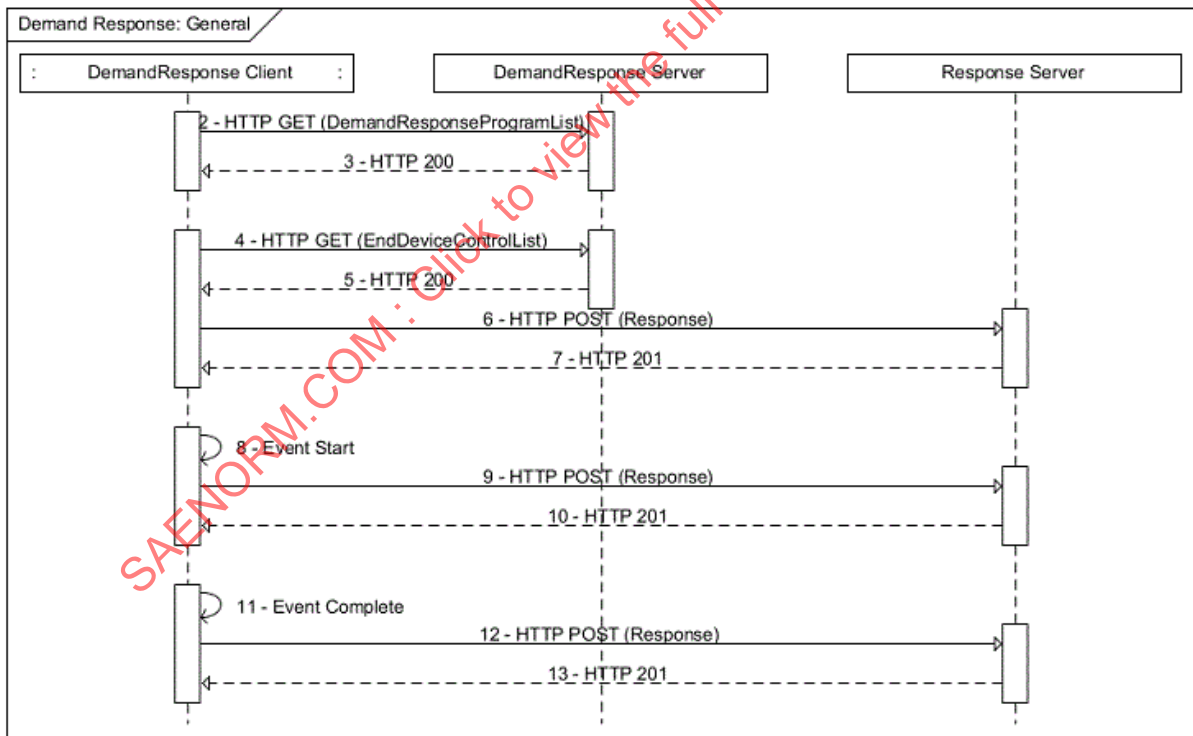
- Support resource subscription mechanisms
- Be able to internally store and support at least 5 unique Load Control Events
- Prioritize local storage of events and give preference to events with start times in the near future and events flagged as mandatory

4.4.3.2 Operation Sequences

The following sequence diagrams use semantics from the SEP2.0 specification; therefore it is important to first provide a high-level definition of the resources and objects found in the DRLC function set of SEP2.0.

#	SEP2 Resource/Object	Description	Subscription
1	EndDeviceControl URI: /dr/{#}/edc/{#}	Specific EndDeviceControl. This object represents the actual load control event. This will provide all required attributes for an event, such as timing, duration, etc.	Yes
2	EndDeviceControlList URI: /dr/{#}/edc	List of EndDeviceControls within one Demand Response Program	Yes
3	Active EndDeviceControlList URI: /dr/{#}/aedc	A list of current active EndDeviceControls. A server can host multiple active events as they can be targeted at different device types.	Yes
4	DemandResponseProgram URI: /dr/{#}	A specific Demand Response Program. This object is essentially a container for multiple EndDeviceControl objects. Servers/providers may use this object to create multiple lists of specific events, perhaps to separate by device type, region, etc. (for example, a provider may offer one set of events or DemandResponseProgram to which thermostats should respond, and another to which PEVs should respond)	No
5	DemandResponseProgramList URI: /dr	List of Demand Response Programs hosted by the server.	Yes

4.4.3.2.1 Load Control Event with Confirmation Required



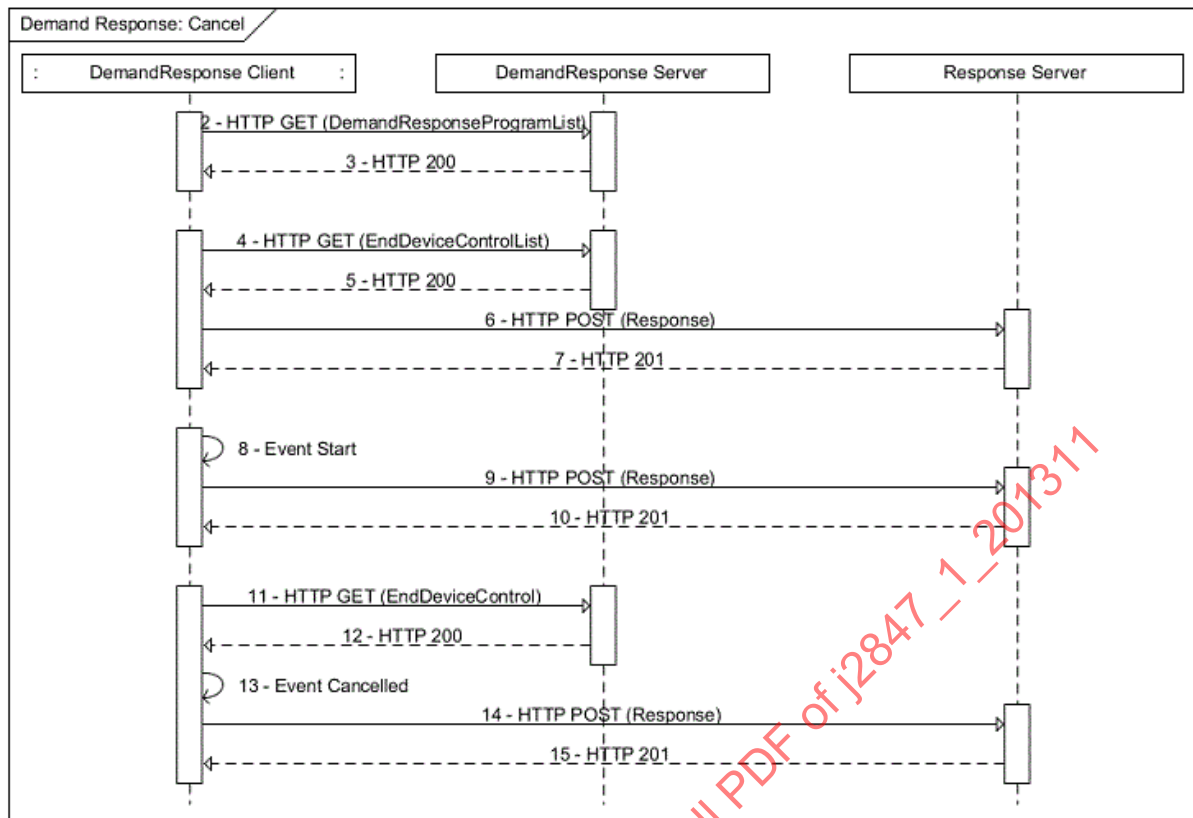
NOTE: The Response server could be located in the same physical device as the DRLC server, it is logically represented separately in this diagram.

4.4.3.2.2 Canceled Load Control Event

Step	Description
1	The client has discovered its EndDevice instance on the Server with a link to its FunctionSetAssignments. Within its FunctionSetAssignments, the client discovers it is part of a DemandResponseProgram. The enrollment is provided out of band.
2	Client GETs the list of DemandResponsePrograms from the DRLC Server. Client sends the following request: GET /drp?l=2 HTTP/1.1 Host: {hostname} Accept: application/sep+xml
3	DemandResponse server responds with the list of DemandResponsePrograms. Server sends the following response: HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <DemandResponseProgramList all="2" results="2" subscribable="0" xmlns="http://zigbee.org/sep"> <DemandResponseProgram href="/drp/1"> <mRID>0FB7</mRID> <description>Operation X</description> <ActiveEndDeviceControlListLink all="0" href="/drp/1/aedc"/> <EndDeviceControlListLink all="1" href="/drp/1/edc"/> <primacy>0</primacy> </DemandResponseProgram> <DemandResponseProgram href="/drp/2"> <mRID>80000001</mRID> <description>The Wackness</description> <ActiveEndDeviceControlListLink all="0" href="/drp/2/aedc"/> <EndDeviceControlListLink all="0" href="/drp/2/edc"/> <primacy>1</primacy> </DemandResponseProgram> </DemandResponseProgramList>
4	Client GETs the list of EndDeviceControls from the DRLC Server for the desired DemandResponseProgram. Client sends the following request: GET /drp/1/edc HTTP/1.1 Host: {hostname} Accept: application/sep+xml

Step	Description
5	DemandResponse server responds with the list of EndDeviceControls. Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <EndDeviceControlList all="1" results="1" subscribable="0" xmlns="http://zigbee.org/sep"> <EndDeviceControl href="/drp/1/edc" replyTo="{hostname}/rsp" responseRequired="01" subscribable="0"> <mRID>CAFEFEED</mRID> <description>Emergency One Hour Coffee Brew</description> <creationTime>1234556</creationTime> <EventStatus> <currentStatus>0</currentStatus> <dateTime>1234556</dateTime> <potentiallySuperseded>false</potentiallySuperseded> <reason>Need Caffeine Soon</reason> </EventStatus> <interval> <duration>360</duration> <start>1234900</start> </interval> <randomizeDuration>60</randomizeDuration> <randomizeStart>60</randomizeStart> <deviceCategory>08</deviceCategory> <drProgramMandatory>true</drProgramMandatory> <loadShiftForward>true</loadShiftForward> <SetPoint> <heatingSetpoint>10000</heatingSetpoint> </SetPoint> </EndDeviceControl> </EndDeviceControlList> </pre>
6	For each EndDeviceControl with ResponseRequired Bit 0, set the Client POSTs a Response with a Status of "Event Received" to the Response resource specified by the replyTo field in the EndDeviceControl. Client sends the following: <pre> POST /rsp HTTP/1.1 Host: {hostname} Content-Type: application/sep+xml Content-Length: {contentLength} <Response xmlns="http://zigbee.org/sep"> <createdDateTime>1234560</createdDateTime> <endDeviceLFDI>C0FFEE00</endDeviceLFDI> <status>1</status> <subject>CAFEFEED</subject> </Response> </pre>
7	Response Server replies with Status 201 Created: <pre> HTTP/1.1 201 Created </pre>
8	Client begins the event at the defined start Time.

Step	Description
9	Client POST a Response with a Status of "Event Started" to the Response resource specified by the replyTo field in the EndDeviceControl. Client sends the following: <pre>POST /rsp HTTP/1.1 Host: {hostname} Content-Type: application/sep+xml Content-Length: {contentLength} <Response xmlns="http://zigbee.org/sep"> <createdDateTime>1234900</createdDateTime> <endDeviceLFDI>C0FFEE00</endDeviceLFDI> <status>2</status> <subject>CAFEFEED</subject> </Response></pre>
10	Response Server Responds with Status 201 Created: HTTP/1.1 201 Created
11	Client completes the event.
12	Client POST a Response with a Status of "Event Completed" to the Response resource specified by the replyTo field in the EndDeviceControl. Client sends the following: <pre>POST /rsp HTTP/1.1 Host: {hostname} Content-Type: application/sep+xml Content-Length: {contentLength} <Response xmlns="http://zigbee.org/sep"> <createdDateTime>1235260</createdDateTime> <endDeviceLFDI>C0FFEE00</endDeviceLFDI> <status>3</status> <subject>CAFEFEED</subject> </Response></pre>
13	Response Server Responds with Status 201 Created: HTTP/1.1 201 Created



Step	Description
1	(Same as Demand Response: General)
2	(Same as Demand Response: General)
3	(Same as Demand Response: General)
4	(Same as Demand Response: General)
5	(Same as Demand Response: General)
6	(Same as Demand Response: General)
7	(Same as Demand Response: General)
8	(Same as Demand Response: General)
9	(Same as Demand Response: General)
10	(Same as Demand Response: General)
11	Client GETs a specific EndDeviceControl from the DRLC Server to check on the current Status of the event it's executing. This should happen on a periodic basis to check if the control is cancelled. Client sends the following request: <pre>GET /drp/1/edc/3 HTTP/1.1 Host: {hostname} Accept: application/sep+xml</pre>

Step	Description
12	DemandResponse Server indicates the event is cancelled: HTTP/1.1 200 OK Content-Type: application/sep+xml Content-Length: {contentLength} <pre><EndDeviceControl href="/drp/1/edc/3" replyTo="{hostname}/rsp" responseRequired="01" subscribable="0" xmlns="http://zigbee.org/sep"> <mRID>CAFEFEED</mRID> <description>Emergency One Hour Coffee Brew</description> <creationTime>1234556</creationTime> <EventStatus> <currentStatus>2</currentStatus> <dateTime>1234960</dateTime> <potentiallySuperseded>false</potentiallySuperseded> <reason>Caffeine Overload</reason> </EventStatus> <interval> <duration>360</duration> <start>1234900</start> </interval> <randomizeDuration>60</randomizeDuration> <randomizeStart>60</randomizeStart> <deviceCategory>0008</deviceCategory> <drProgramMandatory>true</drProgramMandatory> <loadShiftForward>true</loadShiftForward> <SetPoint> <heatingSetpoint>10000</heatingSetpoint> </SetPoint> </EndDeviceControl></pre>
13	Client cancels the event.
14	Client responds with the specific instance of the EndDeviceControl with an updated EventStatus of 'Canceled' Server sends the following response: POST /rsp HTTP/1.1 Host: {hostname} Content-Type: application/sep+xml Content-Length: {contentLength} <pre><Response xmlns="http://zigbee.org/sep"> <createdDateTime>1235100</createdDateTime> <endDeviceLFDI>C0FFEE00</endDeviceLFDI> <status>6</status> <subject>CAFEFEED</subject> </Response></pre>
15	Response Server replies with Status 201 Created: HTTP/1.1 201 Created

4.4.4 U3 Real Time Pricing (RTP)

See section 4.4.2

4.4.5 U4 Critical Peak Pricing (CPP)

See section 4.4.2

4.4.6 U5 Optimized Energy Transfer Programs

This function set provides an interface for exchange of energy flow (e.g., charge or discharge) reservation events. Server devices of this function set include ESIs, EVSEs, and EMSs. FlowReservations allow for the scheduling of high demand periods such as during fast-charging transactions, to make them run at different times and avoid high aggregated demand. Typically, energy rates have penalties, charges, or customer classes for different demand tiers, so it is usually least expensive to keep the maximum demand as low as possible. Distribution utilities may support this function set to minimize the maximum demand across the distribution system.

Servers accept FlowReservationRequests from client devices by exposing a FlowReservationRequestList for each EndDevice. Clients POST a request to transfer a certain amount of energy during a specific interval, at a specific rate. Servers create an associated FlowReservationResponse in the EndDevice's FlowReservationResponseList. Servers may create superseding events to modify the interval within the requested timeframe and update the status to affect client behavior and distribute load across multiple reservations. To do this, the server must have knowledge of multiple clients, but can simply approve all requests unchanged if there is no other information

Application Guidelines and further information related to the Flow Reservation function set can be found in section 12.2 of the SEP 2.0 Application Specification

4.4.6.1 Flow reservation Client Device Requirements

The following requirements are derived from the Energy Flow Reservation Function Set (Section 12.8) of the SEP2.0 specification. These requirements capture the baseline functionality required for Flow Reservation client devices as defined by SEP2.0, however, they can be modified as seen fit by the SAE community.

A client generates a FlowReservationRequest in order to trigger a FlowReservationResponse event from the server.

FlowReservation client devices SHALL:

- Be capable of internally storing and supporting at least one FlowReservationRequest instance and one FlowReservationResponse instance.
- NOT modify a FlowReservationRequest except to update the associated RequestStatus.
- Update the associated RequestStatus to Cancelled for any FlowReservationRequest that they want a server to subsequently disregard.
- If a FlowReservationRequest is removed from the server, clients and servers SHALL NOT assume the FlowReservationRequest has been cancelled. Servers MAY remove a request as required (e.g., after the request has been fulfilled or space is needed). It is the server's responsibility to manage FlowReservationRequests. A server SHALL return a 400 ("Bad Request") response code if it receives a PUT of a FlowReservationRequest that contains changes other than RequestStatus.

FlowReservation client devices SHOULD:

- Be capable of internally storing and supporting at least three unique FlowReservationRequest instances.

4.4.6.2 Vehicle Considerations for Flow Reservation

The SEP2 Model defines the objects and attributes associated with the Flow Reservation Request and the Flow Reservation Response. The Flow Reservation Response object is also an Event and therefore inherits attributes from the Event object. Figure 1 shows how some of the elements of information in the Flow Reservation function are defined. While the elements shown in figure 1 are used for communication, these are not necessarily directly related to the data entry required by the driver to set up the vehicle for participating in a charging session that uses Flow Reservation. The vehicle must provide the controls and displays to perform this setup function along with the software required to produce the information for the Flow Reservation Request and use the information received by the vehicle in the Flow Reservation Response. This will be explained in this section. It is important to understand the meaning of the information that is exchanged in order to understand how it is created and used by the vehicle.

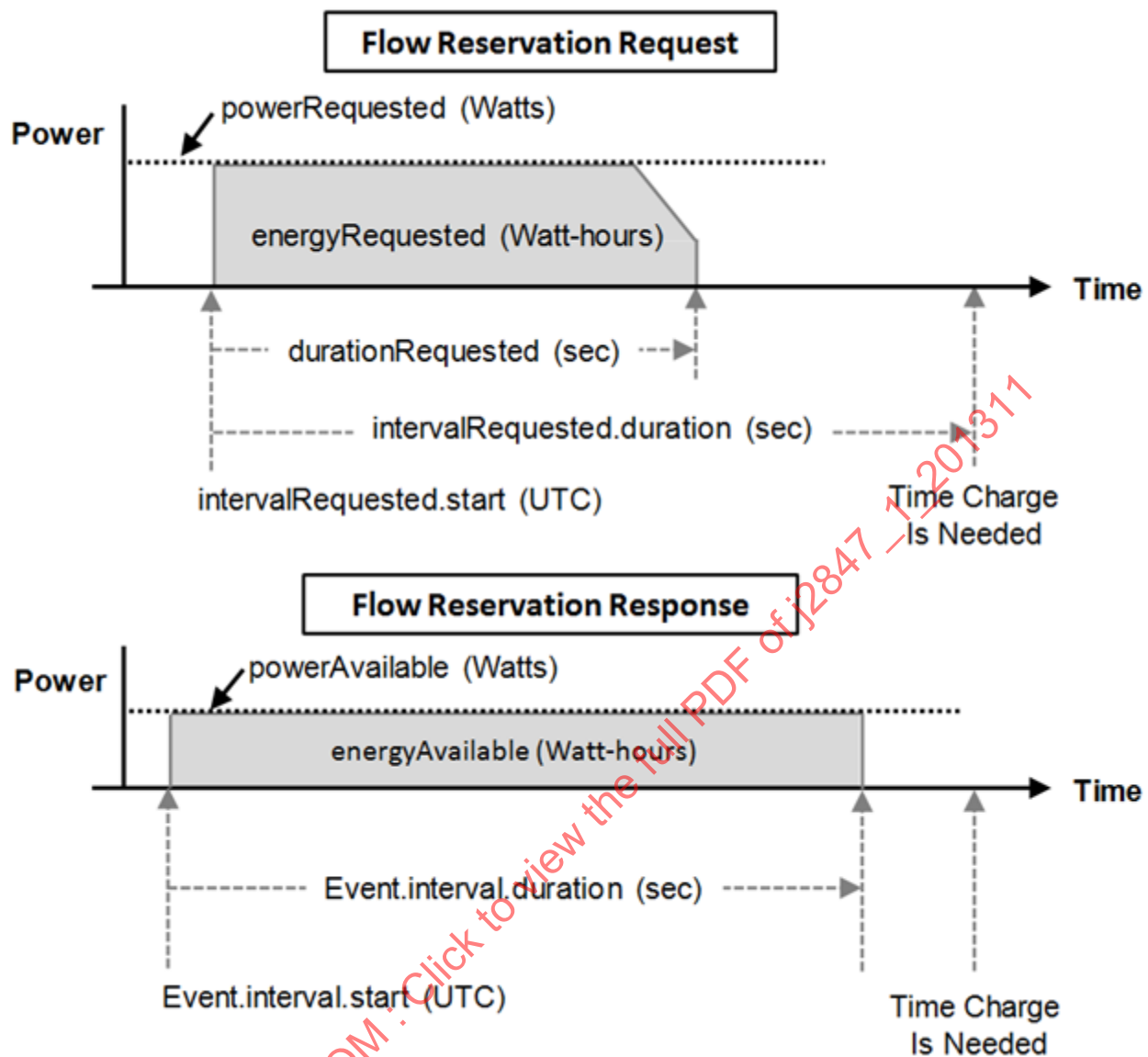


FIGURE 1 - FLOW RESERVATION PARAMETERS

4.4.6.2.1 Setting the Values of the Request

For charging sessions, the value of `powerRequested` should be the maximum sustained power in Watts that the vehicle charger could consume over most of its charging cycle. This value may be lower than the rated value of the charger as determined by battery condition, ambient temperature, and other factors. This must also be the lower of the above derated charger capability and any limitations imposed by the EVSE SAE J1772™ control pilot. For charging sessions the value of `powerRequested` should be computed by vehicle software and should not be provided by driver data entry. Allowing the `powerRequested` value to be set below the capability of the vehicle and the EVSE would be an unnecessary constraint on the reservation controller's ability to schedule power for the charging session.

The value of `energyRequested` indicates the total amount of energy, in Watt-Hours, requested to be transferred between the electric power system and the vehicle. Positive values indicate charging. While the vehicle could be designed to optionally allow for direct entry of the value, one method should be for the driver to only enter a target State of Charge for the session and have the vehicle calculate the required energy transfer. The default value of target SOC should be 100%. The calculated value of `energyRequested` would consider the target SOC, the current SOC, the efficiency of the charger, the usable capacity of the battery, parasitic loads, and other factors. These algorithms will be unique for each vehicle model. Direct entry of a value of `energyRequested` by the driver is equivalent to specifying the number of gallons of gasoline to transfer at a pump. Use of 100% SOC is equivalent to filling the tank with gasoline.

There are two values associated with `intervalRequested`: the start time and the duration. The driver data entry at connection could define a specific time for charging to start or could authorize use of a price schedule. The vehicle software would then use the appropriate value in the `intervalRequested.start`. At connection the driver also enters the Time Charge Is Needed (TCIN) for the charging session. This value is not directly used in the Flow Reservation Request, although it is available through the Power Status resource. The `intervalRequested.duration` is the time difference between the TCIN and the start time. This should be computed by the vehicle software and it is an indirect method of providing the reservation server with the TCIN.

The value of the `durationRequested` defines the minimum duration, in seconds, that it will take to complete the actual charging, including any ramp times and conditioning times, if applicable. This should be calculated by the vehicle software based on the values of `energyRequested` and `powerRequested` and with a full understanding of the charging cycle characteristics. This should not be a data entry because it is not independently derived.

4.4.6.2.2 Interpreting the Response

The Flow Reservation server uses the five request parameters from each vehicle's request to perform the power allocation for each vehicle in its aggregated fleet of vehicles. These algorithms are highly dependent on the specific application and well beyond the scope of the SAE documents. The controlling entity provides a Flow Reservation Response for the request which authorizes the start time and the power available, both of which the vehicle can directly control. The response should never provide a power available that is greater than the power requested and if a higher value is provided the vehicle should use the requested value. If during a transfer the charger limit is reduced by changes in the control pilot, the lower value of the power available and the limit should be used by the charger.

The response also provides the energy available. It should normally be expected that the energy available will equal the energy requested. While the charging profile shape looks different in figure 1, the areas (energy) in the request and the response are the same. This should be used as one confirmation that the charging session can be successfully completed even though the power available may be less than the power requested or start time is delayed. If the energy available is less than the energy requested, the vehicle must decide whether to proceed with or to disengage from flow reservation. If the driver entered a specific value of Watt-hours to transfer, charging should stop when the transfer has been completed. However, if the driver entered a target SOC, charging should continue until the target SOC has been achieved (and the available and requested energy values should not be used to terminate charging).

The response provides a value for duration for the charging event which will most likely always be incorrect because the controlling entity would not have a way of authoritatively calculating the actual duration required to transfer the available energy at the authorized maximum power available. This is not simply the ratio of available energy over available power because the actual transfer time requires a complete understanding of the vehicle's charging cycle. The vehicle should assume that this value is only indicative, but incorrect, and ignore it. The vehicle should estimate the transfer time based on the charging solution using the power available. If the start time plus the vehicle calculated transfer duration exceeds the TCIN, the charging response is also not viable, and the vehicle will need to decide whether to proceed with Flow Reservation.

The vehicle software needs to decide how to deal with conflicts between achieving vehicle charging objectives and staying within limitations imposed by the controlling entity. In the example in figure 1, the controller response designates an earlier start time than requested (which while unusual it could happen) and a lower `powerAvailable` than the `powerRequested`. The `energyAvailable` is equal to the `energyRequested` and therefore the charging solution is acceptable to the vehicle. The `interval.duration` (while approximate) fits within the TCIN constraint. This should be validated by the vehicle based on the reduced power level than requested.

The charging requirements continually change during the session and the original request and response at connection may deviate from what is required during the session. If the actual charger efficiency is lower than that used to calculate the energy request needed to achieve a 100% SOC, more total energy transfer will be needed during the session than the original request. The vehicle could cancel the original request and provide a new request if there is a significant deviation during the session. If the driver needed to disconnect from the EVSE earlier than time entered at connection, the driver could use telematics to update TCIN, it would be appropriate for the vehicle to automatically cancel and provide a revised Flow Reservation Request.

4.4.6.2.3 Using Power Status with Flow Reservation

The Power Status resource in SEP2 contains a PEVInfo object. The PEVInfo object was added to SEP2 to support the Distributed Energy Resource (DER) function set when used with a vehicle. The object presents the vehicle charging requirements to the DER controller. It is a dynamic version of the Flow Reservation Request. The object includes attributes for chargingPowerNow, energyRequestNow, maxForwardPower, minimumChargingDuration, targetStateOfCharge, and timeChargelsNeeded. This is always used to designate the charging requirements so energyRequestNow must always be a positive value based on achieving the target SOC from the current value of SOC. As the vehicle charges or discharges, this value continuously changes as SOC changes during a session.

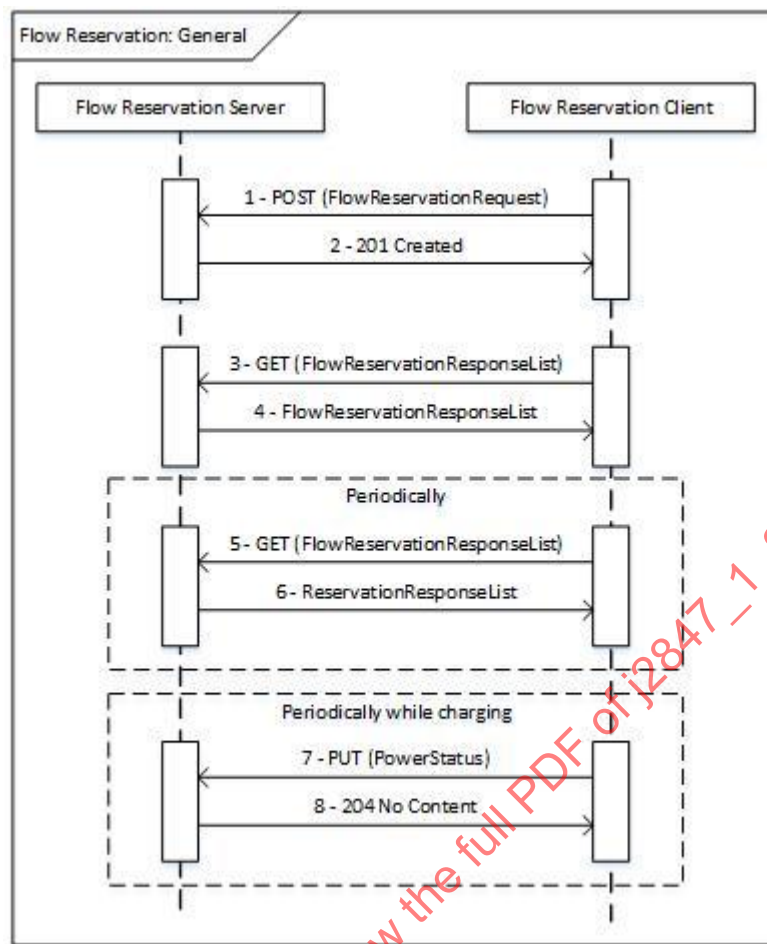
The reservation server could regularly monitor PowerStatus to track the status of vehicle charging during a session. This could be useful for applications where the vehicle makes a single request at connection and the controlling entity plans to issue a sequence of changes to powerAvailable during the session. It is not required that there can only be one response per request. The controller could use PowerStatus to track the “to go” solution and make adjustments as needed to successfully complete vehicle charging. There is no requirement for a flow reservation controller to use PowerStatus. The XML examples for flow reservation show that PowerStatus is provided during the session.

The value of energyRequestNow is continuously changing, but the controlling entity would only need it at a rate consistent with the requirements of its application. SEP2 does not provide a mechanism for the controller to define the desired update rate. If the vehicle was a discoverable server, it could refresh the value internally every second and the controller could use a GET to request it as needed. But when the vehicle hosts its resources on an end-device server, it must PUT the PowerStatus resource on the server at an update rate that supports the applications that need the information. The resource has a timestamp which allows the controlling entity to adjust the values for changes that may have happened. The energyRequestNow can be corrected to current time by using the chargingPowerNow over the time skew if this correction is important to the application. For use with Flow Reservation it should be acceptable to update PowerStatus every 15 minutes. Higher rates may be required for DER applications. The vehicle software will need to establish the update rates appropriate for PowerStatus for different functions or applications.

4.4.6.3 Operation Sequences

4.4.6.3.1 Flow Reservation: Request

The following sequence diagrams use semantics from the SEP2.0 specification; therefore it is important to first provide a high-level definition of the resources and objects found in the Flow Reservation function set of SEP2.0.



The following is a summary of the example:

Step 1 - Client (PEV) creates a FlowReservationRequest at 5:00 PM for charging between midnight and 8:00 AM, 12 kWh energy requested at a power level of 7 kW, 7371 seconds duration requested.

- Subsequently, the Server creates a FlowReservationResponse with a charge interval between 1:00 AM and 5:20 AM at 3 kW.

Step 3 - Client requests the FlowReservationResponseList to find the response matching the request.

Step 4 – Server responds with the FlowReservationResponseList.

Step 5 - Client periodically requests the FlowReservationResponse to look for changes.

Step 7 - Client updates PowerStatus periodically during charging.

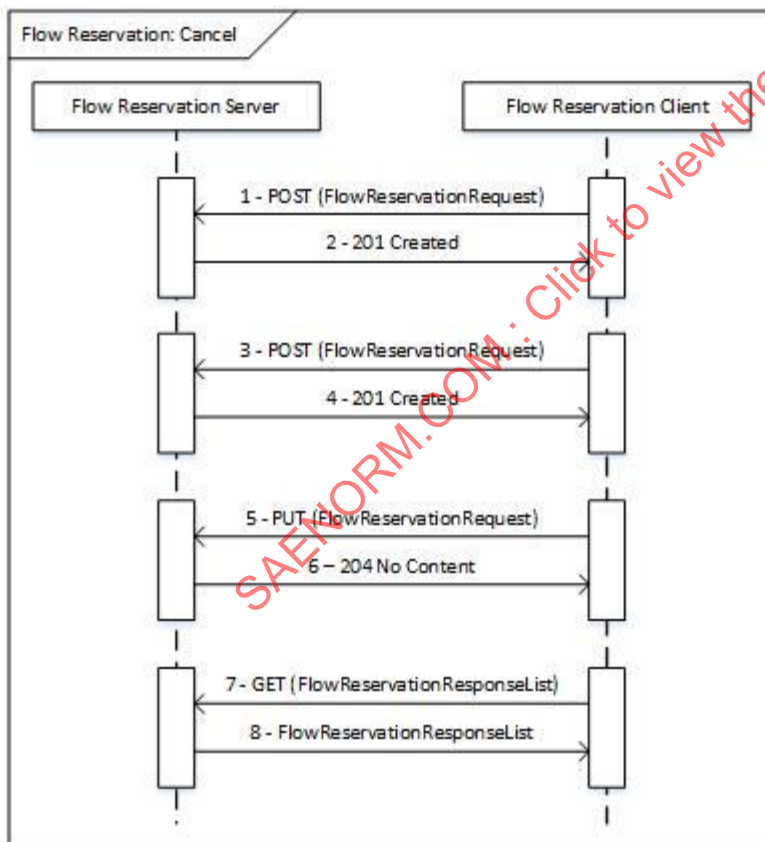
NOTE: In most cases, registration is required to obtain access to request flow reservations

Step	Description
1	Client POSTs a FlowReservationRequest to the Flow Reservation Server at 9/22/2013 5:00 PM. Client sends the following request: <pre> POST /edev/3/frq HTTP/1.1 Host: {hostname} <FlowReservationRequest xmlns="http://zigbee.org/sep"> <mRID>68512866203db3b10000e566</mRID> <description>Charge between 12AM and 8AM</description> <!-- 9/22/2013 5:00 PM GMT --> <creationTime>1379869200</creationTime> <!-- 6171sec charging + 1200sec conditioning --> <durationRequested>7371</durationRequested> <energyRequested> <!-- 12 kWh --> <multiplier>3</multiplier> <value>12</value> </energyRequested> <intervalRequested> <!-- 8 hours --> <duration>28800</duration> <!-- 9/23/2013 12:00 AM GMT --> <start>1379894400</start> </intervalRequested> <powerRequested> <!-- 7 kW --> <multiplier>3</multiplier> <value>7</value> </powerRequested> <RequestStatus> <dateTime>1379869200</dateTime> <!-- Requested --> <requestStatus>0</requestStatus> </RequestStatus> </FlowReservationRequest> </pre>
2	Flow Reservation server responds with the FlowReservationRequest location. Server sends the following response: <pre> HTTP/1.1 201 Created Location: /edev/3/frq/1 </pre>
3	Client GETs the FlowReservationResponseList from the Flow Reservation Server to look for a response to the request. Client sends the following request: <pre> GET /edev/3/frp HTTP/1.1 Host: {hostname} </pre>
4	Flow Reservation server responds with the FlowReservationResponseList Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml <FlowReservationResponseList all="1" href="/edev/3/frp" results="1" subscribable="1" xmlns="http://zigbee.org/sep"> <FlowReservationResponse href="/edev/3/frp/1" subscribable="1"> <mRID>f8afa6fde40db98d0000ea75</mRID> <description>Charge between 1AM and 5:20AM</description> <!-- 9/22/2013 5:01 PM GMT --> <creationTime>1379869260</creationTime> <EventStatus> <!-- Scheduled --> <currentStatus>0</currentStatus> <dateTime>1379869260</dateTime> <potentiallySuperseded>false</potentiallySuperseded> </EventStatus> </FlowReservationResponse> </FlowReservationResponseList> </pre>

Step	Description
	<pre> </EventStatus> <interval> <!-- 4 hours 20 minutes --> <duration>15600</duration> <!-- 9/23/2013 1:00 AM GMT --> <start>1379898000</start> </interval> <energyAvailable> <multiplier>0</multiplier> <value>12000</value> </energyAvailable> <powerAvailable> <multiplier>0</multiplier> <value>3000</value> </powerAvailable> <subject>68512866203db3b10000e566</subject> </FlowReservationResponse> </FlowReservationResponseList> </pre>
5	Client GETs the FlowReservationResponseList periodically (or subscribes) from the Flow Reservation Server. Client sends the following request: <pre> GET /edev/3/frp HTTP/1.1 Host: {hostname} </pre>
6	Flow Reservation server responds with the FlowReservationResponseList. Server sends the following response: <pre> HTTP/1.1 200 OK Content-Type: application/sep+xml <FlowReservationResponseList all="1" href="/edev/3/frp" results="1" subscribable="1" xmlns="http://zigbee.org/sep"> <FlowReservationResponse href="/edev/3/frp/1" subscribable="1"> <mRID>f8afa6fde40db98d0000ea75</mRID> <description>Charge between 1AM and 5:20AM</description> <!-- 9/22/2013 5:01 PM GMT --> <creationTime>1379869260</creationTime> <EventStatus> <!-- Scheduled --> <currentStatus>0</currentStatus> <dateTime>1379869260</dateTime> <potentiallySuperseded>>false</potentiallySuperseded> </EventStatus> <interval> <!-- 4 hours 20 minutes --> <duration>15600</duration> <!-- 9/23/2013 1:00 AM GMT --> <start>1379898000</start> </interval> <energyAvailable> <multiplier>0</multiplier> <value>12000</value> </energyAvailable> <powerAvailable> <multiplier>0</multiplier> <value>3000</value> </powerAvailable> <subject>68512866203db3b10000e566</subject> </FlowReservationResponse> </FlowReservationResponseList> </pre>
7	From 1:00 AM to 5:20 AM while charging the client periodically updates its PowerStatus. Client sends the following request: <pre> PUT /edev/3/ps HTTP/1.1 Host: {hostname} <PowerStatus xmlns="http://zigbee.org/sep"> <!-- more than LowChargeThreshold remaining --> <batteryStatus>1</batteryStatus> <!-- 9/23/2013 3:00 AM GMT --> <changedTime>1379905200</changedTime> <!-- mains --> </pre>

Step	Description
	<pre> <currentPowerSource>1</currentPowerSource> <estimatedChargeRemaining>7000</estimatedChargeRemaining> <PEVInfo> <chargingPowerNow> <multiplier>0</multiplier> <value>3000</value> </chargingPowerNow> <energyRequestNow> <multiplier>0</multiplier> <value>6100</value> </energyRequestNow> <maxForwardPower> <multiplier>3</multiplier> <value>7</value> </maxForwardPower> <!-- 3600sec * 6100Wh/7000W + 1200sec conditioning --> <minimumChargingDuration>4337</minimumChargingDuration> <targetStateOfCharge>10000</targetStateOfCharge> <!-- 9/23/2013 8:00 AM GMT --> <timeChargeIsNeeded>1379923200</timeChargeIsNeeded> <timeChargingStatusPEV>1379905200</timeChargingStatusPEV> </PEVInfo> </PowerStatus> </pre>
8	Flow Reservation server responds: HTTP/1.1 204 No Content

4.4.6.3.2 Flow Reservation: Cancel



The following is a summary of the example:

Step 1 - Client (PEV) creates a FlowReservationRequest at 5:00 PM for charging between midnight and 8:00 AM.

- Subsequently, the Server creates a FlowReservationResponse with a charge interval between 1:00 AM and 5:20 AM at 3 kW.

Step 3 – At 1:00 AM the Client wants to change the time the charging is needed and therefore creates a new FlowReservationRequest for charging between 1:00 AM and 5:00 AM.

- Subsequently, the Server creates a FlowReservationResponse with a charge interval between 1:02 AM and 4:22 AM at 4 kW.

Step 5 - Client cancels the original FlowReservationRequest.

- Subsequently, the Server cancels the original FlowReservationResponse.

Step 7 - Client requests the FlowReservationResponseList.

Step 8 – Server responds with the FlowReservationResponseList, the first FlowReservationResponse is canceled and the second FlowReservationResponse is Active.

Step	Description
1	Client POSTs a FlowReservationRequest to the Flow Reservation Server at 9/22/2013 5:00 PM. Client sends the following request: <pre>POST /edev/3/frq HTTP/1.1 Host: {hostname} <FlowReservationRequest xmlns="http://zigbee.org/sep"> <mRID>68512866203db3b10000e566</mRID> <description>Charge between 12AM and 8AM</description> <!-- 9/22/2013 5:00 PM GMT --> <creationTime>1379869200</creationTime> <!-- 6171sec charging + 1200sec conditioning --> <durationRequested>7371</durationRequested> <energyRequested> <!-- 12 kWh --> <multiplier>3</multiplier> <value>12</value> </energyRequested> <intervalRequested> <!-- 8 hours --> <duration>28800</duration> <!-- 9/23/2013 12:00 AM GMT --> <start>1379894400</start> </intervalRequested> <powerRequested> <!-- 7 kW --> <multiplier>3</multiplier> <value>7</value> </powerRequested> <RequestStatus> <dateTime>1379869200</dateTime> <!-- Requested --> <requestStatus>0</requestStatus> </RequestStatus> </FlowReservationRequest></pre>
2	Flow Reservation server responds with the FlowReservationRequest location. Server sends the following response: <pre>HTTP/1.1 201 Created Location: /edev/3/frq/1</pre>

Step	Description
3	Due to a change in when the vehicle is needed, at 9/23/2013 1:00 AM the client creates a second FlowReservationRequest for charging between 1:00 AM and 5:00 AM. Client sends the following request: <pre>POST /edev/3/frq HTTP/1.1 Host: {hostname} <FlowReservationRequest xmlns="http://zigbee.org/sep"> <mRID>68512866203db3b10000e577</mRID> <description>Charge between 1AM and 5AM</description> <!-- 9/23/2013 1:00 AM GMT --> <creationTime>1379898000</creationTime> <!-- 6171sec charging + 1200sec conditioning --> <durationRequested>7371</durationRequested> <energyRequested> <!-- 12 kWh --> <multiplier>3</multiplier> <value>12</value> </energyRequested> <intervalRequested> <!-- 4 hours --> <duration>14400</duration> <!-- 9/23/2013 1:00 AM GMT --> <start>1379898000</start> </intervalRequested> <powerRequested> <!-- 7 kW --> <multiplier>3</multiplier> <value>7</value> </powerRequested> <RequestStatus> <dateTime>1379898000</dateTime> <!-- Requested --> <requestStatus>0</requestStatus> </RequestStatus> </FlowReservationRequest></pre>
4	Flow Reservation server responds with the FlowReservationRequest location. Server sends the following response: <pre>HTTP/1.1 201 Created Location: /edev/3/frq/2</pre>
5	Client PUTs a canceled status to the first FlowReservationRequest. Client sends the following request: <pre>PUT /edev/3/frq/1 HTTP/1.1 Host: {hostname} <FlowReservationRequest xmlns="http://zigbee.org/sep"> <mRID>68512866203db3b10000e566</mRID> <description>Charge between 12AM and 8AM</description> <!-- 9/22/2013 5:00 PM --> <creationTime>1379869200</creationTime> <durationRequested>7371</durationRequested> <energyRequested> <multiplier>3</multiplier> <value>12</value> </energyRequested> <intervalRequested> <duration>28800</duration> <start>1379894400</start> </intervalRequested> <powerRequested> <multiplier>3</multiplier> <value>7</value> </powerRequested> <RequestStatus> <!-- 9/23/2013 1:01 AM GMT --> <dateTime>1379898060</dateTime></pre>

Step	Description
	<pre><!-- Canceled --> <requestStatus>1</requestStatus> </RequestStatus> </FlowReservationRequest></pre>
6	Flow Reservation server sends the following response: HTTP/1.1 204 No Content
7	Client GETs the FlowReservationResponseList periodically (or subscribes) from the Flow Reservation Server. Client sends the following request: <pre>GET /edev/3/frp?l=2 HTTP/1.1 Host: {hostname}</pre>
8	Flow Reservation server responds with the FlowReservationResponseList. The first FlowReservationResponse is Canceled, the second is Active. Server sends the following response: HTTP/1.1 200 OK Content-Type: application/sep+xml <pre><FlowReservationResponseList all="2" href="/edev/3/frp" results="2" subscribable="1" xmlns="http://zigbee.org/sep"> <FlowReservationResponse href="/edev/3/frp/1" subscribable="1"> <mRID>f8afa6fde40db98d0000ea75</mRID> <description>Charge between 1AM and 5:20AM</description> <!-- 9/22/2013 5:01 AM GMT --> <creationTime>1379869260</creationTime> <EventStatus> <!-- Canceled --> <currentStatus>2</currentStatus> <dateTime>1379898060</dateTime> <potentiallySuperseded>true</potentiallySuperseded> </EventStatus> <interval> <!-- 4 hours 20 minutes --> <duration>15600</duration> <!-- 9/23/2013 1:00 AM GMT --> <start>1379898000</start> </interval> <energyAvailable> <multiplier>0</multiplier> <value>12000</value> </energyAvailable> <powerAvailable> <multiplier>0</multiplier> <value>3000</value> </powerAvailable> <subject>68512866203db3b10000e566</subject> </FlowReservationResponse> <FlowReservationResponse href="/edev/3/frp/1" subscribable="1"> <mRID>f8afa6fde40db98d0000ea76</mRID> <description>Charge between 1:02AM and 4:22AM</description> <!-- 9/23/2013 1:02 AM GMT --> <creationTime>1379898120</creationTime> <EventStatus> <!-- Active --> <currentStatus>1</currentStatus> <dateTime>1379898120</dateTime> <potentiallySuperseded>false</potentiallySuperseded> </EventStatus> <interval> <!-- 3 hours 20 minutes --> <duration>12000</duration> <!-- 9/23/2013 1:02 AM GMT --> <start>1379898120</start> </interval> <energyAvailable> <multiplier>0</multiplier> <value>12000</value> </energyAvailable></pre>

Step	Description
	<pre><powerAvailable> <multiplier>0</multiplier> <value>4000</value> </powerAvailable> <subject>68512866203db3b10000e577</subject> </FlowReservationResponse> </FlowReservationResponseList></pre>

5. NOTES

5.1 Marginal Indicia

A change bar (I) located in the left margin is for the convenience of the user in locating areas where technical revisions, not editorial changes, have been made to the previous issue of this document. An (R) symbol to the left of the document title indicates a complete revision of the document, including technical revisions. Change bars and (R) are not used in original publications, nor in documents that contain editorial changes only.

PREPARED BY THE SAE HYBRID TECHNICAL COMMITTEE

SAENORM.COM : Click to view the full PDF of J2847-1_201311

APPENDIX A
MAPPING OF SAE J2847/1-2011 MESSAGES TO SEP 2.0

Message	Primary Source	Need ed by PEV						SEP 2.0 Mapping ID	Comments
			Use Case Requirements						
			S = Standard, required to achieve functionality described by Use Case						
			U 1	U 2	U 3	U 4	U5		
Identifications									
Vehicle ID	PEV		S	S	S	S	S	AbstractDevice.sFDI	SelfDevice Class
Customer ID and/or PIN	PEV							Registration.pIN	
EUMD ID	EUMD		S		S	S	S	AbstractDevice.sFDI	
Communications Authenticated	EMS	YES	S	S	S	S	S	N/A see Comments	Communications are authenticated through registration and MDNS/DNS-SD process when edev (end device) resource created on server through authentication processes above.
Smart PEV Present	PEV		S	S	S	S	S	N/A see Comments	All SEP 2.0 devices (e.g., Smart PEV) identified from the registration certificate exchange process above.
EVSE Override Request	EVSE	YES	S	S	S	S	S	N/A see Comments	EVSE Manufacturer specific (provide section number)
EVSE ID	EVSE							AbstractDevice.sFDI	
Premise ID	METER							AbstractDevice.sFDI or CustomerAgreement.serviceAccount	

Energy Requests									
Energy Request (amount)	PEV						S	FlowReservationRequest.energyRequested	
Power Request (rate)	PEV						S	FlowReservationRequest.powerRequested	
Energy Available (amount)	EMS	YES					S	FlowReservationResponse.energyAvailable	
Power Available (rate)	EMS	YES					S	FlowReservationResponse.powerAvailable	
Power Schedule	EMS	YES					S	Via Sequence of FRR to one FRQ	Partial but acceptable support
Energy Delivered (charge kWh)	EUMD		S		S	S	S	ReadingType	Reading done through the Metering function set
Timing Information									
Time Charging to Start - Estimated	PEV		S	S	S	S		N/A see Comments	PEV starts based on TOU or other criterion and no estimate of start time is provided to utility
Time Charging to Start - Requested	PEV						S	FRQ: intervalRequested.start	
Time Charging to Start - Authorized	EMS	YES					S	Event.interval.start	Authorized Start by EMS
Time Charging to End - Estimated	PEV		S	S	S	S		N/A see Comments	PEV does not provide an estimate of when charging will be completed to utility
Time Charging to End - Requested	PEV						S	FRQ: intervalRequested.start + intervalRequested.duration	Method for providing Time Charge Is Needed for U5.
Time Charging to End - Authorized	EMS	YES					S	FRR: Event.interval.start + Event.interval.duration	Authorized Start + Duration
Anticipated Charge Duration	PEV						S	FlowReservationRequest.durationRequested	PowerStatus.minimumChargingDuration (For use with DER Function - optional for FR)

Time Charge is Needed	PEV		S	S	S	S	S	intervalRequested.start + intervalRequested.duration	PowerStatus.timeCharges Needed (For use with DER Function - optional elsewhere)
Charging Profile	PEV						S	PowerStatus.chargingPowerNow and PowerStatus.timeChargingStatusPEV	
Actual Charge Start Time	EUMD		S		S	S	S	N/A see Comments	Not directly provided
Pricing									
Request Scheduled Prices	PEV				S		S	TimeTarrifInterval	Various types of tariffs are supported
Publish Price	EMS	YES			S		S	TimeTarrifInterval	Various types of tariffs are supported
Define Rate Time Period	EMS	YES	S			S	S	TimeTarrifInterval	Various types of tariffs are supported
Rate Time Period Status Hash	EMS	YES	S			S	S	TimeTarrifInterval	Various types of tariffs are supported
Request Rate Time Period Info	PEV		S			S	S	TimeTarrifInterval	Various types of tariffs are supported
Price for Rate Time Period	EMS	YES	S			S	S	TimeTarrifInterval	Various types of tariffs are supported
Load Control									
Load Control	EMS	YES		S				EndDeviceControl	
Cancel Load Control	EMS	YES		S				EndDeviceControl.Event Status	
Report Event Status Request	EMS	YES		S				EventStatus.CurrentStatus	
Report Event Status Response	PEV			S				Response.Status	
Request Scheduled Events	PEV			S				EndDeviceControlList	

Vehicle Info / Status										
Time at Connection	EMS	YES	S	S	S	S	S	Time.currentTime and Time.quality	PEV to synchronize to EMS at connection - value can be saved at time of connection.	
Battery SOC Start	PEV						O	N/A see Comments	User must save actual SOC at start. NOT SUPPORTED - NOT NEEDED BY FR	
Battery SOC End	PEV						O	PowerStatus.targetStateOfCharge	Not Used by FR	
Battery SOC Actual	PEV						O	PowerStatus.estimatedChargeRemaining	DERStatus.stateOfChargeStatus (alternate for DER)	
Vehicle Type	PEV						O	N/A see Comments	NOT NEEDED	
Usable Battery Energy	PEV						O	DERCapability.rngWh (for DER function)	NOT NEEDED FOR FR (U5)	
Customer Mode Preference	PEV						S	N/A see Comments	Not needed here. To be dealt in conflict resolution (J2836/5)	

SAENORM.COM : Click to view the full PDF of J2847-1 Revised NOV2013