

**UCAIug OpenSG OpenADE Task Force**  
**OpenADE 1.0 Service Definition** - REST Extension

1 **OPENADE 1.0 SERVICE DEFINITION - REST EXTENSION**

2 *VERSION: DRAFT V0.8*

3 Release Date: 4/8/2010

1 **Acknowledgements**

2 The following individuals and their companies have contributed and/or provided support to the work of  
3 the OpenADE 1.0 Service Definition - REST Extension:

- 4 • Chad Maglaque from Microsoft
- 5 • Charles Spirakis from Google
- 6 • Dave Mollerstuen from Tendril Networks
- 7 • Gerald Gray from CIMple Integrations
- 8 • Jeremy McDonald from SCE
- 9 • Mark Ortiz from Consumers Energy
- 10 • Shawn Hu from Xtensible Solutions / SCE
- 11 • Steve Van Ausdall from Xtensible Solutions / SCE

12 The OpenADE Task Force wishes to thank all of the contributors to OpenADE, especially the above-  
13 mentioned individuals and their companies for their support of this important endeavor, as it sets a key  
14 foundation for an interoperable Smart Grid.

15

**UCAIug OpenSG OpenADE Task Force**  
**OpenADE 1.0 Service Definition - REST Extension**

---

1 **Document History**

---

2 **Revision History**

3 Date of this revision: Apr. 8, 2010

Revision Number	Revision Date	Revision By	Summary of Changes	Changes marked
0.5	2/25/10	Steve Van Ausdall	Initial draft discussion version.	N
0.6	3/1/10	Steve Van Ausdall	Additional details about defined resources	N
0.8	4/8/10	Steve Van Ausdall	Broke REST parts out of Core doc	Y

---

4 **Open Issues Log**

5 Last updated: Mar. 1, 2010

Issue	Issue Date	Provided By	Summary of the Issue

6

## Contents

1			
2	1	Introduction	6
3	1.1	Rights / Management / Governance	6
4	1.1.1	Intellectual Property Rights	6
5	1.1.2	CIM Object Models	6
6	1.1.3	Service Resource Definitions	7
7	1.2	Referenced Standards	7
8	1.3	Referenced Guidance	7
9	1.4	Namespaces	7
10	2	Resources	7
11	2.1	URI Format / syntax	8
12	2.2	Message document format	8
13	2.3	Payload entities	8
14	2.3.1	resources	9
15	3	Patterns	9
16	3.1	Creating, Updating, Deleting	9
17	3.2	Query, request and response (Retrieve) formats	10
18	3.2.1	Format	10
19	3.2.2	Category	10
20	3.2.3	Reference Expansion	10
21	3.2.4	Sorting	10
22	3.2.5	Filtering	11
23	3.2.6	Iteration	11
24	3.2.7	Conditional Retrieval	11
25	3.3	Event Notification (pub/sub)	11
26	3.4	Batch transfers	11

**UCAIug OpenSG OpenADE Task Force**  
**OpenADE 1.0 Service Definition** - REST Extension

1	4	Discovery	12
2	5	Metadata	12
3	6	Extensibility	12
4	7	Versioning	12
5	8	Concurrency	12
6	9	Functional Areas	13
7	9.1	Common	13
8	9.1.1	<b>Discover</b> Resource- (Sequence diagram)	13
9	9.2	Metering Consumption	14
10	9.2.1	<b>Consumption Request</b> - (Sequence diagram)	14
11	9.2.2	<b>Consumption Subscribe</b> - (Sequence diagram)	15
12	10	Resource Definitions	15
13	10.1	Resource Definition	16
14	10.2	Resource Details	17
15	10.2.1	Collection (Feed)	17
16	10.2.2	Category	17
17	10.2.3	Resources	17
18	10.2.4	Authorization	18
19	10.2.5	Access Token	18
20	10.2.6	Notification	18
21	11	Traceability	18
22	11.1	TBD	18

---

**List of Figures**

25	Figure 2: Discover Service Resources Sequence Diagram .....	13
26	Figure 6: Subscribe Sequence Diagram.....	15
27	Figure 7: Service Resource Interfaces .....	15
28	Figure 8: Use of CIM objects within feeds.....	16

29

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

1

### 1 INTRODUCTION

3 This document contains only the extensions necessary to the OpenADE Core specification to build an AtomPub  
4 resource representation syndication server. It is based heavily on GData, an open specification of AtomPub  
5 extensions required for general-purpose data synchronization. The "OpenSG OpenADE SD – Core" document  
6 should be thought of as the parent of this document, filling in sections not addressed in the Core specification.

7 These extensions define a collection of resource feeds as a discoverable, stateless data service, using HTTPS to  
8 send and receive requests and information in XML. This resource-oriented architecture is proposed, similar to  
9 efforts elsewhere, such as web / internet of things, GData, and OData. This architecture provides secure access to  
10 scalable methods and data resources hosted by the provider, while maintaining concurrency and integrity. All data  
11 is secured at the user level, so that access to individual operations can be provided or revoked to external services,  
12 and other users' data will still be protected.

### 1.1 RIGHTS / MANAGEMENT / GOVERNANCE

#### 1.1.1 INTELLECTUAL PROPERTY RIGHTS

15 This document and the information contained herein is provided on an "AS IS" basis. UCAIug DISCLAIMS ALL  
16 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE  
17 INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF  
18 MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

19 UCAIug requests any party that believes it has a patent claim that would necessarily be infringed by  
20 implementations of this UCAIug work, to notify UCAIug immediately, so that fair and reasonable licensing terms  
21 can be negotiated. UCAIug invites any party aware of applicable undisclosed patent claims to contact the UCAIug.  
22 UCAIug may include such claims on its website, but disclaims any obligation to do so.

23 UCAIug takes no position regarding the validity or scope of any intellectual property or other rights that might be  
24 claimed to pertain to the implementation or use of the technology described in this document or the extent to  
25 which any license under such rights might or might not be available; neither does it represent that it has made any  
26 effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of  
27 licenses to be made available, or the result of an attempt made to obtain a general license or permission for the  
28 use of such proprietary rights by implementers or users of this UCAIug recommendation, can be obtained from the  
29 UCAIug. UCAIug makes no representation that any information or list of intellectual property rights will at any time  
30 be complete, or that any claims in such list are, in fact, Essential Claims.

#### 1.1.2 CIM OBJECT MODELS

32 The recommendations herein build on work owned by the IEC. Required extensions identified in this  
33 recommendation will be submitted to the IEC, and will be tracked for inclusion in the model.

34 Information on the management of rights and governance can be found at the page below.

35 <http://www.iec.ch/tctools/patent-guidelines.htm>

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

### 1.1.3 SERVICE RESOURCE DEFINITIONS

If necessary, UCAIug is willing to work with standards development organizations to incorporate additional aspects of this recommendation into standards, including the specification of how to use profiled (restricted) CIM objects within a RESTful HTTP environment, and possibly the resource definitions themselves.

### 1.2 REFERENCED STANDARDS

- [1] OpenADE B&UR 1.0 - <http://osgug.ucaiug.org/sgsubsystems/OpenADE/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2fsgsystems%2fOpenADE%2fShared%20Documents%2fBusiness%20and%20User%20Requirements>
- [2] OpenADE SRS 1.0 - <http://osgug.ucaiug.org/sgsubsystems/OpenADE/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2fsgsystems%2fOpenADE%2fShared%20Documents%2fSRS>
- [3] IEC CIM (TC 57 61968/61970) - <http://tc57.iec.ch>
- [4] Atom Publishing Protocol (RFC 5023) – <http://tools.ietf.org/html/rfc5023>
- [5] OAuth - <http://oauth.net/>
- [6] GData - <http://code.google.com/apis/gdata/docs/2.0/reference.html>
- [7] OData - [http://www.odata.org/docs/\[MC-APDSU\].htm](http://www.odata.org/docs/[MC-APDSU].htm)
- [8] PubSubHubbub - <http://code.google.com/p/pubsubhubbub/>
- [9] OpenSG OpenADE SD - Core

### 1.3 REFERENCED GUIDANCE

- [G1] 3PDA – Security Profile for Third Party Data Access (ASAP-SG)

### 1.4 NAMESPACES

The subject of namespaces is important, because the namespace identifies the domain managing the definitions of protocol resources and formats. OpenSG proposes to use a temporary namespace until the final destination is identified. In any case, namespaces already defined elsewhere and used directly within reference service definitions will remain where they are, and will reference the identified body.

The proposed temporary namespace for definitions to be submitted to standards is below.

`http://osgug.ucaiug.org/ns/2010/aoad`

## 2 RESOURCES

Resource Oriented Architecture is nothing new; in fact the web we are all familiar with today provides restful (browser) access to internet resources. When you specify the Address URL of a page, you are providing the address of that resource that you requested. This architecture provides similar operations for external data consumer applications to request data and methods. Generally, data is made available as a feed, which is an agreement about how to query, create, update, request, and delete entries (individual object records, which have a defined schema according to their type).

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

1 Since this document is the first to define the general-purpose conventions, several resources were identified to  
2 allow consumers to gain access to the resources they want. These extensions are listed below.

- 3 • **Resource** – To discover provided resources
- 4 • **Authorization** – To exchange tokens for authorizations
- 5 • **Subscription** – To register for notifications

6 In addition, the following “data” resources are currently in scope, as defined in [1] OADE-B&UR and [2] OADE-SRS.

- 7 • **ReadingType** – Represents a type of reading represented by MeterReading
- 8 • **MeterReading** – Represents a collection of readings associated with a specific ReadingType

9

### 10 2.1 URI FORMAT / SYNTAX

11 The URIs of the resources defined in OpenADE take the general form below. The <baseUrl> does not need to be  
12 the same across different implementations, since resource addresses include the entire string.

13 `http://<baseUrl>/<resource>?<query>`

14 Resource requests require inputs of user, key, and resource object. Additional path elements may be required for  
15 some resources. Resources return a list (feed) or an individual entry.

16 Requests for protected resources require https, and require authorization token in HTTP header.

17 The query section contains additional inputs that can be specified to affect processing, passed as a list of  
18 `name=value` pairs.

19

### 20 2.2 MESSAGE DOCUMENT FORMAT

21 Message documents shall use the extended version of the Atom Publishing Protocol (AtomPub, which extends the  
22 Atom Syndication Protocol) defined in GData to fulfill this need.

23 In addition to the recommended format, it is possible to support additional representations. An input can be  
24 accepted to provide RSS or other formats, but these additional formats are all optional, and will only be considered  
25 for this specification if needed.

### 26 2.3 PAYLOAD ENTITIES

27 Payload entities will be specializations (subclasses) of the message document entry. They will therefore inherit all  
28 elements defined in the message document entry, as well as implement an XML schema representation of a CIM  
29 class defining additional schema elements.

30 Feed payloads will contain a list of references to resource entries that match the request query criteria.



# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

1 The batch payload defined in the Core document allows a number of documents or resource requests to be  
2 included in a single request. Possibly a flag could be passed as an input parameter to feed resources to specify to  
3 include full representations instead of just references.

4 [8] PubSubHubbub leaves it up to the publisher (provider) of subscribed feeds whether to deliver full  
5 representations or references to modified entities. It may be possible to allow clients to specify their preference  
6 when subscribing.

---

### 7 2.3.1 RESOURCES

8 Domain data objects build on the IEC CIM model. In general, resources will be named using the CIM class as the  
9 resource part of the URI. Collections are returned if no specific entry ID is specified in the resource. For listings of  
10 fields, see the details for each resource, defined in Section 10.

11 An example is shown below.

12 *Note that this is a preliminary draft for discussion purposes.*

```
13 <?xml version="1.0" encoding="UTF-8"?>  
14 <entry xmlns="http://www.w3.org/2005/Atom"  
15     xmlns:m="http://osgug.ucaiug.org/ns/2010/aoad"   
16     xml:lang="en">  
17     <id>https://data.utility.com/rs/MeterReading/fj2ofj8</id>  
18     <category term="m:MeterReading"/>  
19     <m:mRID>fj2ofj8</m:mRID>  
20     <m:IntervalReading>  
21         <m:beginTimeStamp>2001-12-17T09:30:47Z</m:beginTimeStamp>  
22         <m:endTimeStamp>2001-12-17T10:30:47Z</m:endTimeStamp>  
23         <m:value>3.14159E0</m:value>  
24     </m:IntervalReading>  
25     <m:IntervalReading>  
26         <m:beginTimeStamp>2001-12-17T10:30:47Z</m:beginTimeStamp>  
27         <m:endTimeStamp>2001-12-17T11:30:47Z</m:endTimeStamp>  
28         <m:value>3.14159E0</m:value>  
29     </m:IntervalReading>  
30     <link rel='ReadingType' href="https://data.utility.com/rs/ReadingType/7.6.7.1.0.12.0.0.3.72">  
31     <link rel='ServiceDeliveryPoint' href="https://data.utility.com/rs/ServiceDeliveryPoint/98374">  
32 </entry>  
33
```

## 34 3 PATTERNS

35 This section contains guidance and decisions on how message exchanges flow for the general scenarios below. In  
36 general, the constructs and operations defined in [4] AtomPub shall be used, including requests for Services,  
37 Workspaces, Collections, Members, Categories, and Media Types. Extensions are generally refined subsets of the  
38 full specifications detailed in GData, and full implementations should not break clients who only implement these  
39 recommendations.

### 40 3.1 CREATING, UPDATING, DELETING

41 The POST method is to be used for creation of new entries, PUT is to be used for updates to existing entries, and  
42 DELETE is to be used to delete an entry.

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

### 3.2 QUERY, REQUEST AND RESPONSE (RETRIEVE) FORMATS

This section specifies the input parameters that can be passed to GET method operations for format, category, reference expansion, sorting, filtering, and iteration through list items.

OData defines several useful query constructs for consideration in addition to those below, in Section 2.2.3.6 Query Options.

---

#### 3.2.1 FORMAT

The default, and only required format, will be CIM-extended AtomPub feed / entry XML.

Additional formats to be considered include RSS 2.0 and JSON.

OData defines several additional capabilities for definition of formats, including 2.2.1 Abstract Data Model and Conceptual Schema Definition Language.

TODO: Need to determine platform and tool support for these extensions. Could XSD be used to define the structure of entries of different category / types?

GData does not appear to link resources to a specific type, but includes all defined elements in a domain-specific namespace. (For example, PowerMeter does define elements for measurement quantities, etc.)

---

#### 3.2.2 CATEGORY

Specification of the category of entries is accomplished using the Atom element "term", and if needed could be supported as a qualifier in queries by accepting category terms as inputs. In general, each CIM object class will become a category of entry, so that the representations of entries can be specified with a schema.

---

#### 3.2.3 REFERENCE EXPANSION

By default, feed queries will return a list of resource links. If the reference expansion flag is set, entries returned will be expanded to contain their full representation.

(Need to determine if nested expansion is necessary / possible, and if so, how to specify to what level)

OData uses an `m:inline` extension to the `atom:link` element for this purpose.

---

#### 3.2.4 SORTING

Ability to specify the sort order of resulting query / request entries is not necessary - subsequent processing of received data can display or rearrange data however desired. However, ordering of entries shall remain consistent across requests, so that an iterator can be used to page through results.

## UCAIug OpenSG OpenADE Task Force

### OpenADE 1.0 Service Definition - REST Extension

---

#### 1 3.2.5 FILTERING

2 Filtering requires inputs that allow the specification of the resource name and/or path, as well as a range of  
3 publication or update date/times. Properties of the entry element (defined by its category type) could be defined  
4 to be acceptable by default as filter terms for the associated resource. Need to determine if it is feasible to  
5 implement all, or if identification is necessary of only the filter terms required for specific use cases. Possibly usage  
6 patterns could determine the need for indexing, etc.

7 If a specific entry ID is specified, that entry is returned.

8 In GData, “most queries are simply full text search queries”.

---

#### 9 3.2.6 ITERATION

10 Iteration inputs allow consumers to request a subset of entries or references at a time, and then page through  
11 them for processing. Inputs include the starting entry index, and number of entries per page.

12 Query results may require additional terms for iteration.

13 GData uses some openSearch terms.

---

#### 14 3.2.7 CONDITIONAL RETRIEVAL

15 Both OData and GData use If-None-Match in HTTP header to retrieve entries only if they have changed.

#### 16 3.3 EVENT NOTIFICATION (PUB/SUB)

17 The publish / subscribe pattern is incredibly useful, and is specified mostly in AtomPub. However, there is no  
18 mechanism defined in AtomPub to notify subscribers of new feed entries. This requires them to “poll” for new  
19 data, and while this is sufficient in many cases, some business processes require ability to notify in order to achieve  
20 reduced latency in client updates.

21 [8] PubSubHubbub defines a mechanism for this purpose, and is worth consideration for use in this specification.

#### 22 3.4 BATCH TRANSFERS

23 If desired, a feed for each data service consumer could be provided through which all subscribed content would be  
24 returned in a single request (or series of large chunks). This mechanism should allow any resource type to be  
25 included within a single feed. It may be possible to implement this as a regular feed request with reference  
26 expansion specified.

27 GData describes batch processing here <http://code.google.com/apis/gdata/docs/batch.html>.

28 OData describes batch processing in [section 2.2.7.6](#).

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

### 1 4 DISCOVERY

2 Discovery of available resources shall utilize the AtomPub constructs defining services, workspaces, and  
3 collections. This is accomplished with a client request to GET the definition of all collections, followed by  
4 enrollment / authorization, and finally subscription to the appropriate feeds.

5 Discovery of available services and resources is specified in OData, “limited capability negotiation” using  
6 DataServiceVersion (section 2.2.5.3) and MaxDataServiceVersion (section [2.2.5.7](#)).

### 7 5 METADATA

8 A “Resource” resource shall allow retrieval of the representation of all available resources, and the currently  
9 implemented version of each.

10 [4] AtomPub defines a “workspace collection” for this.

11 OData describes “Data Service Metadata” in [section 2.2.3.7](#).

### 12 6 EXTENSIBILITY

13 AtomPub is specified to be extensible, and implementations should be able to function even with different  
14 versions of client or server. In addition, section 6.2 in AtomPub provides recommended behavior.

15 Extensions to the CIM objects will be associated with specific versions of the namespace, specified in the version  
16 attribute of the schema element.

### 17 7 VERSIONING

18 As additional capabilities are added to the interface definition, a specification of the version of the definition will  
19 be needed to help in service discovery negotiation. This should not change the namespace of any definitions.

### 20 8 CONCURRENCY

21 In order to ensure data integrity, clients may only update resources if they are updating the current version of the  
22 resource. If an update request fails due to conflict (not current version), the client must request the latest version,  
23 apply changes to that representation, and retry the update.

24 GData and OData both use [ETags](#) for versioning / concurrency management.

25 See GData Resource Versioning <http://code.google.com/apis/gdata/docs/2.0/reference.html#ResourceVersioning>  
26 for more information.

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

### 9 FUNCTIONAL AREAS

#### 9.1 COMMON

The flows in this section represent general-purpose functions that are needed for all protected resource publications.

##### 9.1.1 **DISCOVER** RESOURCE- (SEQUENCE DIAGRAM)

Addresses OpenSG OpenADE 1.0 SRS 3.2.1, bullet 1.2, 1.3

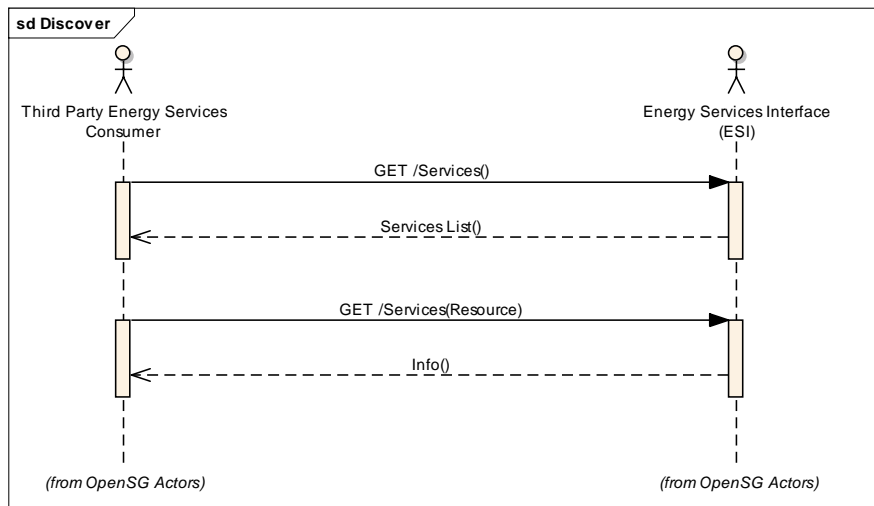


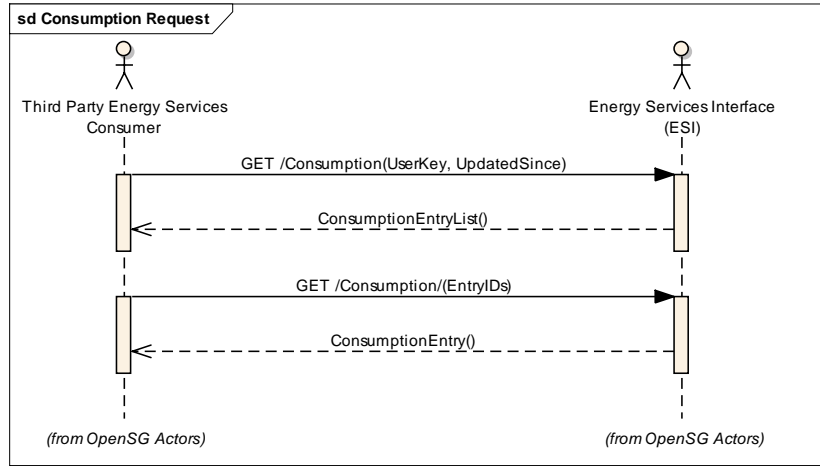
Figure 1: Discover Service Resources Sequence Diagram

**UCAIug OpenSG OpenADE Task Force**  
**OpenADE 1.0 Service Definition - REST Extension**

1 9.2 METERING CONSUMPTION

2 9.2.1 **CONSUMPTION REQUEST** - (SEQUENCE DIAGRAM)

3 Addresses OpenSG OpenADE 1.0 SRS 3.2.1, bullet 3.1, 3.2



4

5

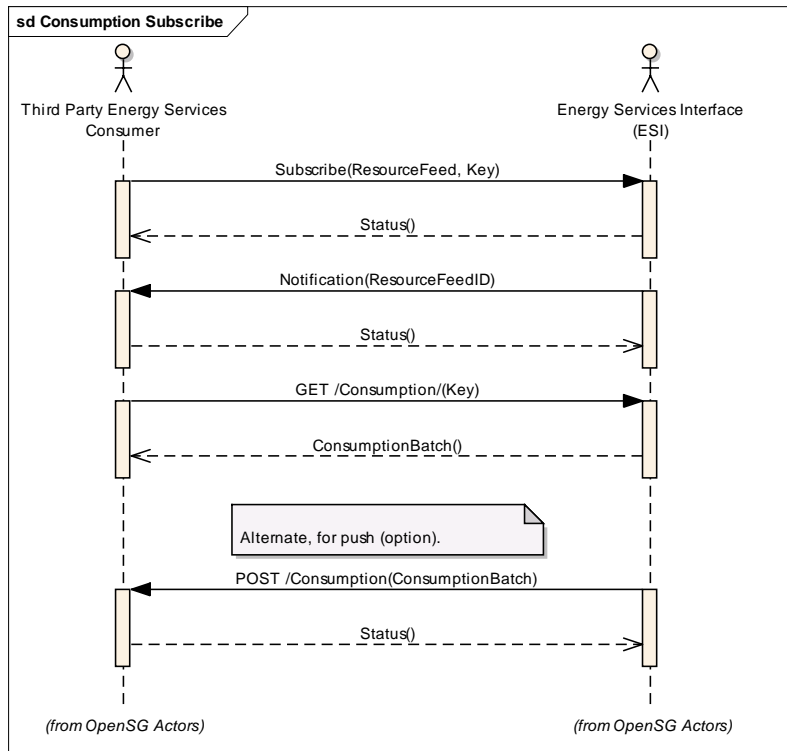
**Figure 2: MeterReading Request Sequence Diagram**

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

### 1 9.2.2 CONSUMPTION SUBSCRIBE - (SEQUENCE DIAGRAM)

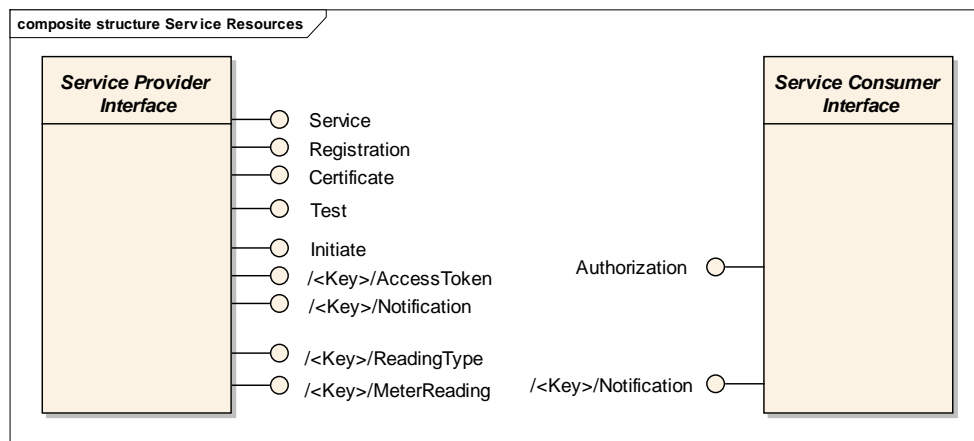
2 Addresses OpenSG OpenADE 1.0 SRS 3.2.1, bullet 3.1



3  
4 **Figure 3: Subscribe Sequence Diagram**

## 5 10 RESOURCE DEFINITIONS

6 The following diagram provides an overview of the service resources defined. Of course, the service consumer also  
7 has to implement client requests for required interfaces, in order to access the resources provided by the provider.



8  
9 **Figure 4: Service Resource Interfaces**

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

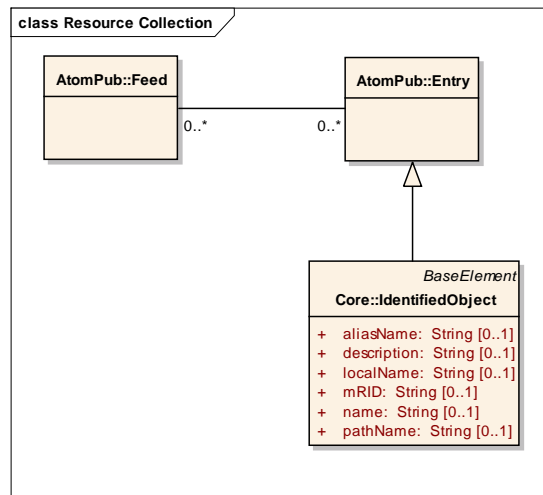
1 The following table lists the resources defined for OpenADE.

Logical Resource Name	Consumer Operation	Implementer	Description
Resource	GET /rs/Resource	Utility	Get supported service resources and extensions
Auth Access Token	GET /rs/<Key>/AccessToken	Utility	Get the authorized request token
Reading Type	GET /rs/<Key>/ReadingType	Utility	Get meter reading types defining readings units and lengths
Meter Reading	GET /rs/<Key>/MeterReading	Utility	Get meter readings – may be interval or instantaneous
Notification	POST /rs/<Key>/Notification	Both	Get notifications such as user modified authorization

### 2 10.1 RESOURCE DEFINITION

3 All resources with beginning with /rs/ support the patterns in Section 3, returning a (possibly expanded) collection  
 4 of data resource stream entries. Individual entries can be managed using POST, UPDATE, and DELETE. Permissions  
 5 may be set according to policy, but guidance is provided regarding the typical configuration in resource details  
 6 following this table.

7 The UML diagram below shows a proposed method of linking CIM object types to a syndication collection  
 8 container such as AtomPub.



9  
10 **Figure 5: Use of CIM objects within feeds**

11 IdentifiedObject is the top-most generalization (superclass) of most CIM classes. By generalizing this with the feed  
 12 “Entry” element, all CIM IdentifiedObjects become valid Entry elements. In addition to the use of IdentifiedObject  
 13 as a specialization of an Entry, CIM classes used as resources shall also have category terms defined for them



# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

1 within service workspace collections, so that entries can use the term element to denote their type and link to  
2 schema.

### 3 10.2 RESOURCE DETAILS

4 Many of the resources below are necessary to support initial setup and authorization. Implementations shall  
5 conform to referenced specifications for details on these interfaces. Clarifications and refinements made to  
6 support these service resources are denoted where necessary.

7 All resources are to be implemented as collections, with the elements listed in the sections below. Additional  
8 allowed values may be specified, and will be included here as necessary.

---

#### 9 10.2.1 COLLECTION (FEED)

10 Collections consist of feeds, and exhibit behavior as defined in AtomPub.

---

#### 11 10.2.2 CATEGORY

12 Categories shall be specified for CIM identifiedObject entry classes using the atom constructs shown in the  
13 example below.

```
14     <?xml version="1.0" ?>  
15     <app:categories  
16         xmlns:app="http://www.w3.org/2007/app"  
17         xmlns:atom="http://www.w3.org/2005/Atom"  
18         fixed="yes" scheme="http://example.com/cats/big3">  
19         <atom:category term="animal" />  
20         <atom:category term="vegetable" />  
21         <atom:category term="mineral" />  
22     </app:categories>
```

---

#### 25 10.2.3 RESOURCES

26 Resource is used to discover service resources available via the addressed endpoint.

Schema	Use	Element
Resource	GET Output	ResourceURI
Resource	GET Output	Name
Resource	GET Output	Version
Resource	GET Output	Categories
Resource	GET Output	Acceptable Types

# UCAIug OpenSG OpenADE Task Force

## OpenADE 1.0 Service Definition - REST Extension

1 \_\_\_\_\_  
2 \_\_\_\_\_

### 3 10.2.4 AUTHORIZATION

4 This resource is used to post the signed authorization for the associated token to the 3<sup>rd</sup> Party.

Schema	Use	Element
Authorization	Input	oauth_token
Authorization	Input	oauth_verifier

### 5 10.2.5 ACCESS TOKEN

6 This resource allows the 3<sup>rd</sup> Party to get the authorized request token. A different key is created for each  
7 authorized resource, so in the case of Meter Readings, individual service point channels would have separate keys.

Schema	Use	Element
AccessToken	Output	realm
AccessToken	Output	oauth_consumer_key
AccessToken	Output	oauth_token
AccessToken	Output	oauth_signature_method
AccessToken	Output	oauth_timestamp
AccessToken	Output	oauth_nonce
AccessToken	Output	oauth_verifier
AccessToken	Output	oauth_signature

8 \_\_\_\_\_  
9 \_\_\_\_\_

### 10 10.2.6 NOTIFICATION

11 Notifications are to be used to announce the creation or modification of resources. Based on the design pattern  
12 chosen for each information exchange, notification may or may not be required.

Schema	Use	Element
Notification	GET Output	Resource List

13 \_\_\_\_\_

## 14 11 TRACEABILITY

### 15 11.1 TBD