OPENADE 1.0 SERVICE DEFINITION - REST EXTENSION

*VERSION: **DRAFT V0.85***

Release Date: 4/15/2010

Acknowledgements

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

- Chad Maglaque from Microsoft
- Charles Spirakis from Google
- Dave Mollerstuen from Tendril Networks
- Gerald Gray from CIMple Integrations
- Jeffrey Kenward from DTE Energy
- Jeremy McDonald from SCE
- Mark Ortiz from Consumers Energy
- Shawn Hu from Xtensible Solutions / SCE
- Steve Van Ausdall from Xtensible Solutions / SCE

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

Document History

Revision History

Date of this revision: Apr. 15, 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
0.85	4/15/10	Steve Van Ausdall	Changes from first review meeting	Y

Open Issues Log

Last updated: Mar. 1, 2010

Issue	Issue Date	Provided By	Summary of the Issue

Contents

List of Figures

List of Tables

1 INTRODUCTION

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

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

1.1 RIGHTS / MANAGEMENT / GOVERNANCE

1.1.1 INTELLECTUAL PROPERTY RIGHTS

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

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

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

1.1.2 CIM OBJECT MODELS

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

Information on the management of rights and governance can be found at the page below.
http://www.iec.ch/tctools/patent-guidelines.htm

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 SPECIFICATIONS

- [1] OpenADE B&UR 1.0 - http://osgug.ucaiug.org/sgsystems/OpenADE/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2fsgsystems%2fOpenADE%2fShared%20Documents%2fBusiness%20and%20User%20Requirements
- [2] OpenADE SRS 1.0 - http://osgug.ucaiug.org/sgsystems/OpenADE/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2fsgsystems%2fOpenADE%2fShared%20Documents%2fSRS
- [3] IEC CIM (TC 57 61968/61970) - http://tc57.iec.ch
- [4] OAuth - http://tools.ietf.org/html/draft-hammer-oauth-10
- [5] HTTP(S) – IETF RFC 2616 - http://www.ietf.org/rfc/rfc2616.txt
- [6] GData - http://code.google.com/apis/gdata/docs/2.0/reference.html
- [7] OData - http://www.odata.org/docs/[MC-APDSU].htm
- [8] PubSubHubbub - http://code.google.com/p/pubsubhubbub/
- [9] Atom Publishing Protocol (RFC 5023) – http://tools.ietf.org/html/rfc5023

1.3 REFERENCED GUIDANCE

- [G1] 3PDA – Security Profile for Third Party Data Access (ASAP-SG) http://osgug.ucaiug.org/utilisec/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2futilisec%2fShared%20Documents%2fThird%20Party%20Data%20Access%20Security%20Profile
- [G2] OpenSG OpenADE SD – Common http://osgug.ucaiug.org/sgsystems/OpenADE/Shared%20Documents/Service%20Definition/OpenADE%201.0%20Service%20Definition/OpenSG%20OpenADE%20SD%20-%20Common%20v0.8.doc

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/aoade
```

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).

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

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

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

- **ReadingType** – Represents a type of reading represented by MeterReading
    - **IntervalReading** – A durational measurement
    - **Reading** – An instantaneous measurement
- **MeterReading** – Represents a collection of readings associated with a specific ReadingType
- **ServiceSupplier** – The supplier of energy service
- **CustomerAuthorisation** – Represents the agreement to share data with the 3rd Party
- **ServiceDeliveryPoint** – The point at which the meter takes readings

2.1 URI FORMAT / SYNTAX

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

```
http://<baseURL>/<resource>?<query>
```

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

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

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

2.2 MESSAGE DOCUMENT FORMAT

Message documents shall use the extended version of the Atom Publishing Protocol ([9] AtomPub, which extends the Atom Syndication Protocol) defined in [6] GData to fulfill this need.

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

2.3 PAYLOAD ENTITIES

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

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

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

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

2.3.1 RESOURCES

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

Some examples are shown below.

Note that this is a preliminary draft for discussion purposes.

/MeterReading/fj2ofj8

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
    xmlns:m="http://osgug.ucaiug.org/ns/2010/aoade">
    <category term="m:MeterReading"/>
    <id>https://data.utility.com/rs/MeterReading/fj2ofj8</id>
    <m:mRID>fj2ofj8</m:mRID>
    <link rel=’ReadingType’ href=”https://data.utility.com/rs/ReadingType/7.6.7.1.0.12.0.0.0.3.72’>
    <link rel=’ServiceDeliveryPoint’ href=”https://data.utility.com/rs/ServiceDeliveryPoint/98374’>
</entry>
```

/MeterReading/fj2ofj8/IntervalReading/2001-12-17T09_30_47Z

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
    xmlns:m="http://osgug.ucaiug.org/ns/2010/aoade">
    <category term="m:IntervalReading"/>
    <m:timeStamp>2001-12-17T09:30:47Z</m:timeStamp>
    < m:endTimeStamp>2001-12-17T10:30:47Z</m:endTimeStamp>
    <m:value>3.1</m:value>
</entry>
```

ReadingType/7.6.7.1.0.12.0.0.0.3.72

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
    xmlns:m="http://osgug.ucaiug.org/ns/2010/aoade">
    <category term="m:ReadingType"/>
    <ID>https://data.utility.com/rs/ReadingType/7.6.7.1.0.12.0.0.0.3.72</id>
    <mRID>7.6.7.1.0.12.0.0.0.3.72</mRID>
    <name>Hourly Interval Delivered Energy (kWh)</name>
</entry>
```

ServiceDeliveryPoint/98374

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
    xmlns:m="http://osgug.ucaiug.org/ns/2010/aoade">
    <category term="m:ServiceDeliveryPoint"/>
    <id>https://data.utility.com/rs/ServiceDeliveryPoint/98374</id>
    <aliasName>My House</aliasName>
    <mRID>98374</mRID>
    <ServiceCategory>
        <kind>electricty</kind>
    </ServiceCategory>
</entry>
```

CustomerAuthorisation/23049857203

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
    xmlns:m="http://osgug.ucaiug.org/ns/2010/aoade">
    <category term="m:CustomerAuthorisation"/>
    <id>https://data.utility.com/rs/CustomerAuthorisation/23049857203</id>
    <mRID>23049857203</mRID>
    <signDate>2001-11-16T09:30:47Z</signDate>
    <validityInterval>
        <end>2002-11-17T09:30:47Z</end>
        <start>2000-11-17T09:30:47Z</start>
    </validityInterval>
</entry>
```

3 PATTERNS

This section contains guidance and decisions on how message exchanges flow for the general scenarios below. In general, the constructs and operations defined in [9] AtomPub shall be used, including requests for Services, Workspaces, Collections, Members, Categories, and Media Types. Extensions are generally refined subsets of the full specifications detailed in [6] GData, and full implementations should not break clients who only implement these recommendations.

3.1 CREATING, UPDATING, DELETING

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

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.

[7] 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.

[7] 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?

[6] 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)

[7] 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.

3.2.5 FILTERING

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

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

In [6] GData, “most queries are simply full text search queries”.

3.2.6 ITERATION

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

Query results may require additional terms for iteration.

[6] GData uses some openSearch terms.

3.2.7 CONDITIONAL RETRIEVAL

Both [7] OData and [6] GData use If-None-Match in HTTP header to retrieve entries only if they have changed.

3.3 EVENT NOTIFICATION (PUB/SUB)

The publish / subscribe pattern is incredibly useful, and is specified mostly in [9] AtomPub. However, there is no mechanism defined in AtomPub to notify subscribers of new feed entries. This requires them to "poll" for new data, and while this is sufficient in many cases, some business processes require ability to notify in order to achieve reduced latency in client updates.

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

3.4 BATCH TRANSFERS

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

[6] GData describes batch processing here http://code.google.com/apis/gdata/docs/batch.html.

[7] OData describes batch processing in section 2.2.7.6.

4 DISCOVERY

Discovery of available resources shall utilize the [9] AtomPub constructs defining services, workspaces, and collections. This is accomplished with a client request to GET the definition of all collections, followed by enrollment / authorization, and finally subscription to the appropriate feeds.

Discovery of available services and resources is specified in [7] OData, "limited capability negotiation" using DataServiceVersion (section 2.2.5.3) and MaxDataServiceVersion (section 2.2.5.7).

5 METADATA

A "Resource" resource shall allow retrieval of the representation of all available resources, and the currently implemented version of each.

[9] AtomPub defines a "workspace collection" for this, as in the example below.

```
<?xml version="1.0" encoding='utf-8'?>
   <service xmlns="http://www.w3.org/2007/app"
            xmlns:atom="http://www.w3.org/2005/Atom">
     <workspace>
       <atom:title>Main Site</atom:title>
       <collection
           href="http://example.org/blog/main" >
         <atom:title>My Blog Entries</atom:title>
         <categories
            href="http://example.com/cats/forMain.cats" />
       </collection>
       <collection
           href="http://example.org/blog/pic" >
         <atom:title>Pictures</atom:title>
         <accept>image/png</accept>
         <accept>image/jpeg</accept>
         <accept>image/gif</accept>
       </collection>
     </workspace>
```

[7] OData describes “Data Service Metadata” in section 2.2.3.7.

6 EXTENSIBILITY

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

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

7 VERSIONING

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

8 CONCURRENCY

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

[6] GData and [7] OData both use ETags for versioning / concurrency management.

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

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

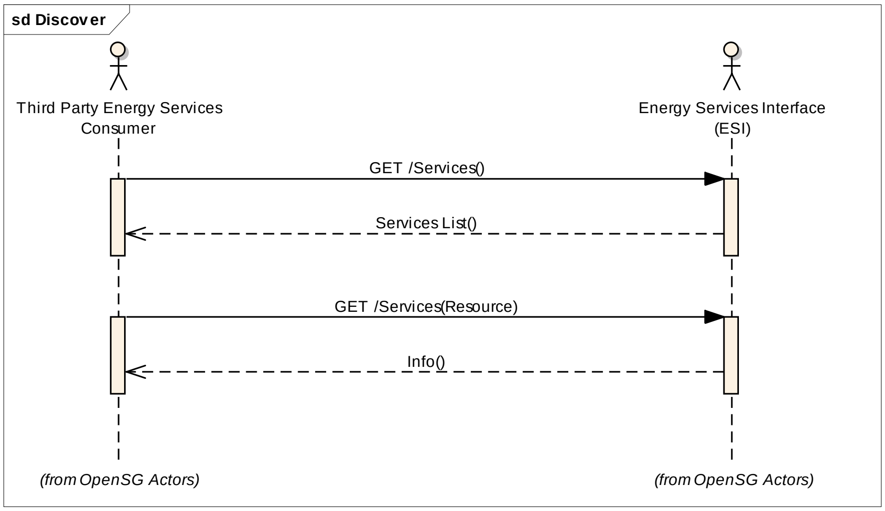

Figure 1: Discover Service Resources Sequence Diagram

9.2 METERING CONSUMPTION

9.2.1 CONSUMPTION REQUEST - (SEQUENCE DIAGRAM)

Addresses OpenSG OpenADE 1.0 SRS 3.2.1, bullet 3.1, 3.2

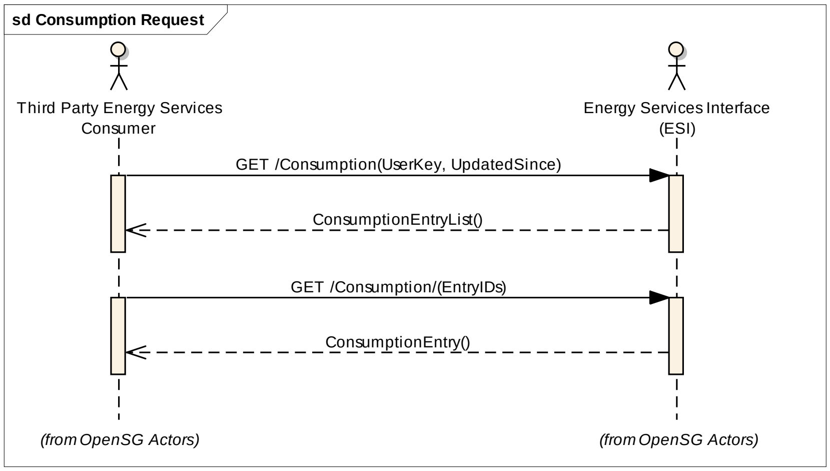
sd Consumption Request
Third Party Energy Services Consumer
Energy Services Interface (ESI)
GET /Consumption(UserKey, UpdatedSince)
ConsumptionEntryList()
GET /Consumption/(EntryIDs)
ConsumptionEntry()
(from OpenSG Actors)
(from OpenSG Actors)

Figure 2: MeterReading Request Sequence Diagram

9.2.2 CONSUMPTION SUBSCRIBE - (SEQUENCE DIAGRAM)

Addresses OpenSG OpenADE 1.0 SRS 3.2.1, bullet 3.1

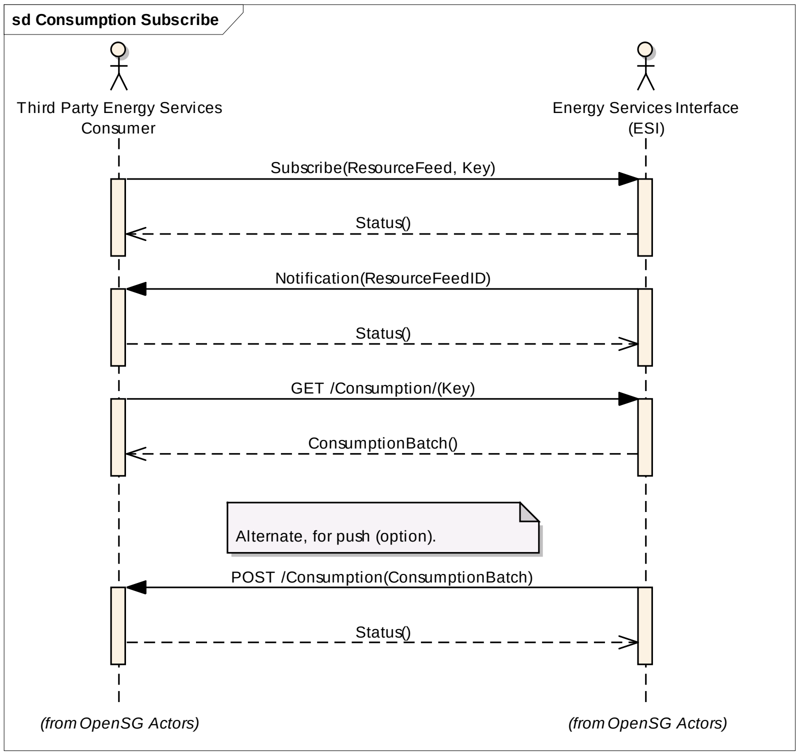
sd Consumption Subscribe
Third Party Energy Services Consumer
Energy Services Interface (ESI)
Subscribe(ResourceFeed, Key)
Status()
Notification(ResourceFeedID)
Status()
GET /Consumption/(Key)
ConsumptionBatch()
Alternate, for push (option).
POST /Consumption(ConsumptionBatch)
Status()
(from OpenSG Actors)
(from OpenSG Actors)

Figure 3: Subscribe Sequence Diagram

10 RESOURCE DEFINITIONS

The following diagram provides an overview of the service resources defined. Of course, the service consumer also has to implement client requests for required interfaces, in order to access the resources provided by the provider. The <Key> shown below may be an access token associated with a specific user, or with a group.

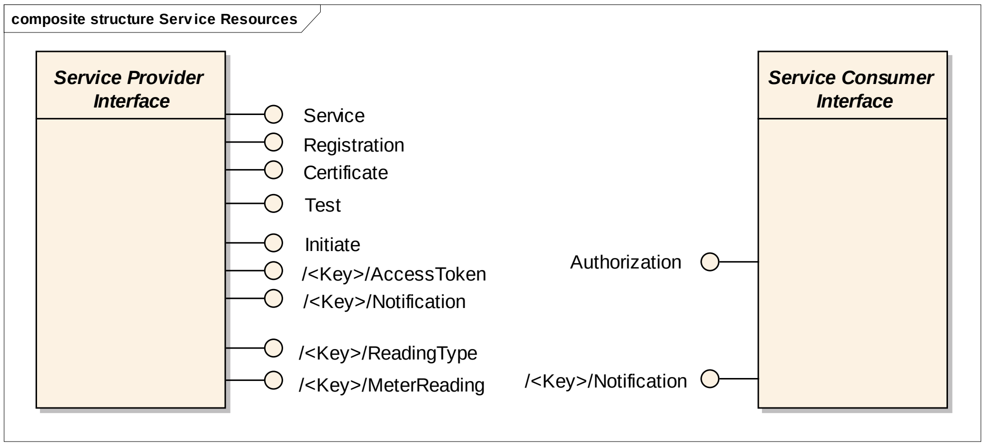

Figure 4: Service Resource Interfaces

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

Table 1: Resource Operations

10.1 RESOURCE DEFINITION

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

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

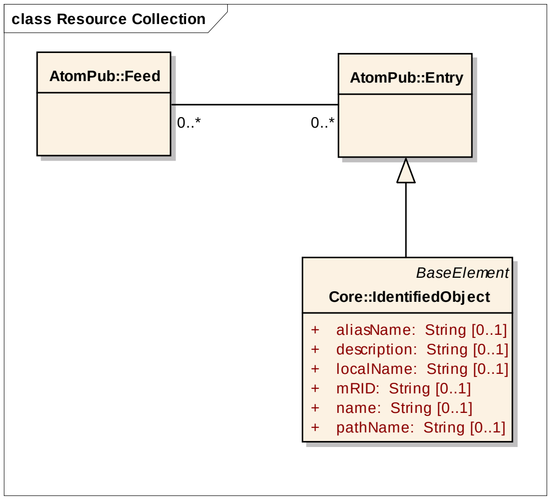
class Resource Collection
AtomPub::Feed
0..*
0..*
AtomPub::Entry
BaseElement
Core::IdentifiedObject
+ aliasName: String [0..1]
+ description: String [0..1]
+ localName: String [0..1]
+ mRID: String [0..1]
+ name: String [0..1]
+ pathName: String [0..1]

Figure 5: Use of CIM objects within feeds

IdentifiedObject is the top-most generalization (superclass) of most CIM classes. By generalizing this with the feed "Entry" element, all CIM IdentifiedObjects become valid Entry elements. In addition to the use of IdentifiedObject as a specialization of an Entry, CIM classes used as resources shall also have category terms defined for them within service workspace collections, so that entries can use the term element to denote their type and link to schema.

10.2 RESOURCE DETAILS

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

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

10.2.1 COLLECTION (FEED)

Collections consist of feeds, and exhibit behavior as defined in [9] AtomPub.

10.2.2 CATEGORY

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

```xml
<?xml version="1.0"?>
<app:categories
    xmlns:app="http://www.w3.org/2007/app"
    xmlns:atom="http://www.w3.org/2005/Atom" fixed="yes"
    scheme="http://http://osgug.ucaiug.org/ns/2010/aoade">
    <atom:category term="MeterReading"/>
    <atom:category term="Reading"/>
    <atom:category term="IntervalReading"/>
    <atom:category term="ReadingType"/>
    <atom:category term="CustomerAuthorisation"/>
    <atom:category term="ServiceSupplier"/>
    <atom:category term="ServiceDeliveryPoint"/>
</app:categories>
```

10.2.3 RESOURCES

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

10.2.4 AUTHORIZATION

This resource is used to post the signed authorization for the associated token to the 3rd Party.

Schema	Use	Element

Authorization	Input	oauth_token
Authorization	Input	oauth_verifier

10.2.5 ACCESS TOKEN

This resource allows the 3rd Party to get the authorized request token. A different key is created for each 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

10.2.6 NOTIFICATION

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

Schema	Use	Element
Notification	GET Output	Resource List