**1. RECOMMENDED ACTION: EFFECT OF EC VOTE TO ACCEPT RECOMMENDED ACTION:**

|  |  |  |  |
| --- | --- | --- | --- |
| X | Accept as requested | X | Change to Existing Practice |
|  | Accept as modified below |  | Status Quo |
|  | Decline |  |  |

**2. TYPE OF DEVELOPMENT/MAINTENANCE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Per Request:** | | **Per Recommendation:** | |
| X | Initiation | X | Initiation |
|  | Modification |  | Modification |
|  | Interpretation |  | Interpretation |
|  | Withdrawal |  | Withdrawal |
|  |  |  |  |
| X | Principle | X | Principle |
| X | Definition | X | Definition |
| X | Business Practice Standard | X | Business Practice Standard |
|  | Document |  | Document |
|  | Data Element |  | Data Element |
|  | Code Value |  | Code Value |
|  | X12 Implementation Guide |  | X12 Implementation Guide |
|  | Business Process Documentation |  | Business Process Documentation |

**3. RECOMMENDATION**

**SUMMARY:**

The UCAIug OpenADE Task Force submitted a request for the initiation of NAESB Model Business Practices on July 29, 2010 (R10008) to standardize the interface which allows for the exchange of energy usage information between designated parties. The UCAIug OpenADE Task Force provided the artifacts on which these Model Business Practices were based.

These Model Business Practices will build on the NAESB Energy Usage Information (EUI) Model and, subject to the Governing Documents and Applicable Regulatory Authority, will help enable Retail Customers to share energy usage information with Third Parties who have acquired the right to act in this role. This Energy Services Provider Interface will provide a consistent method for Retail Customers to authorize a Third Party to gain access to energy usage data. Doing so will help enable Retail Customers to choose Third Party products to assist them to better understand their energy usage and to make more economical decisions about their usage. This Energy Services Provider Interface will contribute to the development of an open and interoperable method for Third Party authorization and machine-to-machine exchange of Retail Customer usage information.

**Recommended Standards:**

# REQ.21 ENERGY SERVICES PROVIDER INTERFACE

## EXECUTIVE SUMMARY

This document establishes the Model Business Practices for the Energy Services Provider Interface. For Retail Customers to better realize the benefits of the Smart Grid, Retail Customer related data (e.g. usage information, etc.) should be made available in a timely manner to the Retail Customer and to the Authorized Third Parties chosen by the Retail Customer.

The Energy Services Provider Interface encompasses a variety of interactions between Retail Customers, Distribution Companies, and Third Parties. In a business environment where best practices are voluntary, Model Business Practices should be applied within the context of regulatory requirements and agreements. These Model Business Practices define an Energy Services Provider Interface, but any obligation to use it would be established by Governing Documents and Applicable Regulatory Authority rules and regulations not these Model Business Practices

## INTRODUCTION

The North American Energy Standards Board (NAESB) is a voluntary non-profit organization comprised of members from all aspects of the natural gas and electric industries. Within NAESB, the Retail Electric Quadrant (REQ) and the Retail Gas Quadrant (RGQ) focus on issues impacting the retail sale of energy to Retail Customers. REQ / RGQ Model Business Practices are intended to provide guidance to Distribution Companies, and other Market Participants involved in providing energy service to Retail Customers. The focus of these Model Business Practices is the Energy Service Provider Interface.

These Model Business Practices are voluntary and do not address policy issues that are the subject of state legislation or regulatory decisions. These Model Business Practices have been adopted with the realization that as the industry evolves, additional and amended Model Business Practices may be necessary. Any industry participant seeking additional or amended Model Business Practices (including principles, definitions, data elements, process descriptions, and technical implementation instructions) should submit a request to the NAESB office, detailing the change, so that the appropriate process may take place to amend the Model Business Practice.

## BUSINESS PROCESSES AND PRACTICES

## Overview

## REQ.21.1 Principles

**REQ.21.1.1** The processes for ESPI should minimize the complexity associated with authorizing Third Parties to access Retail Customers energy usage data.

**REQ.21.1.2** The processes associated with the ESPI should be consistent with any related requirements established by the Governing Documents and Applicable Regulatory Authority.

## REQ.21.2 Definitions

### REQ.21.2.A Business Definitions [Definitions identified with an asterisk are works in process to be modified based on the actors identified in the use cases]

TASK FORCE NOTE: The REQ working glossary can be found through the following link: <http://www.naesb.org/pdf3/req_rgq_glossary_defined_terms.xls>

Terms used from the glossary

**Applicable Regulatory Authority**: The state regulatory agency or other local governing body that provides oversight, policy guidance, and direction to any parties involved in the process of providing energy to Retail Customers through regulations and orders.

**Distribution Company**: A regulated Entity which provides distribution services and may provide energy and/or transmission/transportation services in a given area.

**Entity**: A person or organization with sufficient legal standing to enter into a contract or arrangement with another such person or organization (as such legal standing may be determined by those parties) for the purpose of conducting and/or coordinating energy transactions.

**Governing Documents**: Documents that determine the interactions among parties, including but not limited to: applicable law, regulatory documents (e.g., tariffs, rules, regulations), contractual agreements, Distribution Company Operational Manuals, and other relevant models and operational procedures.

**Retail Customer:** Any Entity that takes gas and/or electric service for its own consumption.

### REQ.21.2.B Technical Definitions

### *[Definitions identified with an asterisk are works in process to be modified based on the actors identified in the use cases]*

**REQ.21.2.B.x Authorizing Entity:** An Entity (e.g. PUC, Distribution Company, bonding agent, etc.) who approves a Third Party to utilize the Energy Services Provider Interface. (return to this and add specifics)

**REQ.21.2.B.x Third Party:** An Entity which provides some service to a Retail Customer based on information it does not have direct access to or direct authority over. A Third Party relies on a Data Custodian to provide access to Retail Customer information.

**REQ.21.2.B.x Authorized Third Party:** A Third Party that has been approved by the Authorizing Entity to utilize the Energy Services Provider Interface. Third Party must provide proof that they meet the governing documents’ requirements for data security and data privacy protection in order to become an Authorized Third Party..

**\*REQ.21.2.B.x Energy Service Provider Interface:** A standardized machine-to machine interface that permits a Data Custodian to share, at the Retail Customer’s request and under the Retail Customer’s direction, a broad set of that Retail Customer’s Data Custodian data with Authorized Third Parties.

**REQ.21.2.B.x Personally Identifiable Information:** any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual‘s identity, such as name, social security number, date and place of birth, mother‘s maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information**[[1]](#footnote-1)**.

**REQ.21.2.B.x**

**Data Custodian:** A Data Custodian holds Retail Customer resource information and will share this information with Third Parties only in accordance with the Governing Documents, Applicable Regulatory Authority and the direction of the Retail Customer. A Data Custodian typically has direct access to the pertinent information (e.g., by directly acquiring electricity usage data from a meter). A Data Custodian may be a Distribution Company.

**\*REQ.*21.2.B.x* Energy Usage Information*:*** Any information concerning a Retail Customer’s use of energy.

### REQ.21.2.C Acronyms

| **Abbreviation / Acronym** | **Meaning** |
| --- | --- |
| ADE | Automatic Data Exchange |
| ESPI | Energy Services Provider Interface |
| EUI | Energy Usage Information |
| NISTIR | National Institute of Standards and Technology Interagency Report |
| PII | Personally Identifiable Information |

## REQ.21.3 Model Business Practices

### REQ.21.3.1 General Practices for Energy Services Provider Interface (ESPI)

**REQ.21.3.1.1** To the extent required by the Applicable Regulatory Authority, Authorized Third Parties and Data Custodians should exchange Retail Customer’s EUI at the Retail Customer’s request pursuant to the requirements as set forth in NAESB REQ.21, subject to the Governing Documents.

**REQ.21.3.1.2** The ESPI relationship requires a set of agreements between Retail Customer-Authorized Third Party, Retail Customer-Data Custodian, and Authorized Third Party-Data Custodian to ensure that the appropriate information is provided as needed and other information access is restricted.

**REQ.21.3.1.3** A Third Party cannot access PII from a Data Custodian. PII may only be provided to a Third Party by the Retail Customer.

**REQ.21.3.1.4** Subject to the Governing Documents and Applicable Regulatory Authority, the ESPI should enable a Retail Customer to share EUI for such Retail Customer with Authorized Third Parties who have acquired the right to act in this role.

**REQ.21.3.1.5** The ESPI should allow exchange of usage information without requiring access to PII.

**REQ.21.3.1.6** All information exchanged by the ESPI should be secure in accordance with the security requirements stated herein. These may be further constrained by Governing Documents and Applicable Regulatory Authority.

**REQ.21.3.1.7** A Retail Customer should have the ability to authorize the Data Custodian to release EUI for such Retail Customer to an Authorized Third Party who has acquired the right to act in this role, subject to the Governing Documents and Applicable Regulatory Authority.

**REQ.21.3.1.8** Subject to the Governing Documents and Applicable Regulatory Authority, a Retail Customer should have the ability to authorize multiple Authorized Third Parties to have limited time based read only access to specified EUI or other types of information for such Retail Customer, with any default expiration for such access established by such Governing Documents or Applicable Regulatory Authority.

**REQ.21.3.1.9** Subject to the Governing Documents and Applicable Regulatory Authority, a Retail Customer should have the ability to designate a specific expiration date, extend any specific expiration date, or indicate an open-ended access timeframe other than the default access period.

**REQ.21.3.1.10** The ESPI should have the capability to support the Retail Customers’ ability to select / revoke which Authorized Third Parties are authorized for read-only access to EUI.

**REQ.21.3.1.11** The ESPI should have the capability to notify the relevant Authorized Third Parties, Data Custodian and Retail Customers when access has been granted, access has been changed, or access has been revoked for a UsagePoint.

**REQ.21.3.1.12** Subject to the Governing Documents and Applicable Regulatory Authority, the ESPI should be consistent with the applicable guidelines around security and authorization for Third Party data access as set forth in the NISTIR 7628.

**REQ.21.3.1.13** The ESPI should provide the ability for any party to terminate any information sharing relationship in which that party is involved.

**REQ.21.3.1.14** Future versions of the ESPI should be backwards compatible, including provisions for exchanging versioning information and negotiating interface capabilities.

**REQ.21.3.1.15** Any Third Party wishing to access EUI via the ESPI must establish and maintain a trusted relationship with each Data Custodian who provides a ESPI. Subject to the Governing Documents and Applicable Regulatory Authority, both the Data Custodian and the Authorized Third Party should disallow requests from Entities who are not Authorized Third Parties.

**REQ.21.3.1.16** Subject to the Governing Documents and Applicable Regulatory Authority during communications of any information confidentiality will be maintained.

**REQ.21.3.1.17** Subject to the Governing Documents and Applicable Regulatory Authority, Third Parties must be authorized by the Authorizing Entity and/or the Data Custodian to be an Authorized Third Party and utilize the ESPI and must maintain their status as an Authorized Third Party.

**REQ.21.3.1.18** If an Authorizing Entity exists within a jurisdiction, Retail Customers should be able to view a list of Third Parties who have been authorized to use the ESPI.

**REQ.21.3.1.19** Subject to the Governing Documents and Applicable Regulatory Authority, EUI shouldbe made availableto Authorized Third Parties (as directed by the Retail Customer) in a reasonable and timely fashion.

**REQ.21.3.1.20** When any of the required authorized relationships described in this recommendation are terminated, access to the ESPI shall not be granted.

**REQ.21.3.1.21** It is necessary in this model business practice to identify participants and their relationships with globally unique identifiers. The identifiers should conform to IETF RFC 4122.

**REQ.21.3.1.22** The creation and dissolution of trusted relationships between any two parties are outside the scope of this Model Business Practice and are necessary preconditions for the use of the ESPI.

**REQ.21.3.1.23** Upon dissolution of any of the required trusted relationships any ESPI relationships must be terminated and parties notified.

**REQ.21.3.1.24** If and when the relationships or criteria change all effected parties should be notified.

**REQ.21.3.1.25** The most interoperable and widely supported technologies should be used to ensure adoption regardless of development and deployment platforms used.

**REQ.21.3.1.26** The technologies chosen should be well specified, with active communities and tools and/or frameworks available.

**REQ.21.3.1.27** Technologies chosen should be compatible and interoperable with technologies specified for access to HAN resources.

## REQ.21.4 Models

### REQ.21.4.1 Profile of REQ.18 Energy Usage Information Model

The following model represents the implementable profile of NAESB PAP10 EUI model for ESPI.



**Figure** 1: ESPI Usage



**Figure** 2: ESPI Usage Summary Classes



**Figure** 3: ESPI Inheritence



**Figure** 4: ESPI Publication



**Figure** 5: ESPI Types



**Figure** 6: ESPI Links

**Batch**

Contains a collection of Lists that contain object representations.

**BatchItem**

Includes elements that make it possible to include multiple transactions in a single (batch) request.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **operation** | *UInt8* | Specifies the operation requested of this item.  0=Create  1=Read  2=Update  3=Delete |
| **name** | *HexBinary16* | An identifier for this object that is only unique within the containing collection. |
| **statusCode** | *UInt16* | Indicates the status code of the associated transaction.  200 - Ok  201 - Created  204 - No Content  301 - Moved Permanently  302 - Redirect  304 - Not Modified  400 - Bad Request  401 - Unauthorized  403 - Forbidden  404 - Not Found  405 - Method Not Allowed  410 - Gone  500 - Internal Server Error |
| **statusReason** | *String32* | Indicates the reason for the indicated status code. |

**Resource**

Generalization of any data object that can be exchanged.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **published** | *TimeType* | Specifies the time at which the object was initially published. (As determined by the publisher) |
| **updated** | *TimeType* | Specifies the time at which the object was last updated. (As determined by the publisher) |

**ServiceStatus**

Contains the current status of the service.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **currentStatus** | *UInt8* | The current status of the service.  0 = Unavailable  1 = Normal, operational |

**Subscription**

Defines the parameters of a subscription between third party and data custodian

**IdentifiedObject**

This is a root class to provide common naming attributes for all classes needing naming attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **mRID** | *HexBinary128* | A Model Authority issues mRIDs. Given that each Model Authority has a unique id and this id is part of the mRID, then the mRID is globally unique.  For ESPI (and SEP 2.0), the Model Authority Unique Idenfier is the IANA PEN provider ID which shall be specified in the first 32 bits, and objects created by that provider shall be assigned unique IDs with the remaining 96 bits. |
| **name** | *HexBinary16* | An identifier for this object that is only unique within the containing collection. |
| **description** | *String32* | The description is a human readable text describing or naming the object. |
| **extension** | *anyType* |  |

**List**

Container to hold a collection of object instances or references.

**ReferenceableObject**

A resource is an addressable unit of information, either a collection (List) or instance of an object (identifiedObject, or simply, Resource)

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **href** | *anyURI* | A reference to the resource address (URI). Required in return from GET, ignored otherwise. (What is this referring to? need a bit more sustance to the definition) |

**ElectricPowerQualitySummaryListLink**

A Link to a List of IntervalBlock instances.

**ElectricPowerUsageSummaryListLink**

A Link to a List of IntervalBlock instances.

**IntervalBlockListLink**

A Link to a List of IntervalBlock instances.

**MeterReadingListLink**

A Link to a List of MeterReading instances.

**ReadingTypeLink**

A Link to a ReadingType.

**ElectricPowerQualitySummaryList**

A List element to hold UsagePoint objects.

**ElectricPowerUsageSummaryList**

A List element to hold UsagePoint objects.

**IntervalBlockList**

A List element to hold IntervalBlock objects.

**MeterReadingList**

A List element to hold MeterReading objects.

**UsagePointList**

A List element to hold UsagePoint objects.

**ServiceCategory**

Category of service provided to the customer.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **kind** | *ServiceKind* | Service classification  Examples are:  0 - electricity  1 - gas  The list of specific valid values per the standard are itemized in ServiceKind. |

**UsagePoint**

Logical point on a network at which consumption or production is either physically measured (e.g. metered) or estimated (e.g. unmetered street lights).

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **status** | *UInt8* | Specifies the current status of this usage point.  The only valid values are:  0 = off  1 = on |

**HexBinary128**  «XSDsimpleType»

A 128-bit field encoded as a hex string (32 characters / 16 octets)

**HexBinary16**  «XSDsimpleType»

A 16-bit field encoded as a hex string (4 characters / 2 octets)

**Int48**  «XSDsimpleType»

Signed integer, max inclusive 281474976710655 (2^48-1), restriction of xs:long

**String32**  «XSDsimpleType»

Character string of max length 32

**UInt16**  «XSDsimpleType»

Unsigned integer, max inclusive 65535 (2^16-1), same as xs:unsignedShort

**UInt32**  «XSDsimpleType»

Unsigned integer, max inclusive 4294967295 (2^32-1), same as xs:unsignedInt

**UInt48**  «XSDsimpleType»

Unsigned integer, max inclusive 281474976710655 (2^48-1), restriction of xs:unsignedLong

**UInt8**  «XSDsimpleType»

Unsigned integer, max inclusive 255 (2^8-1), same as xs:unsignedByte

**AccumulationBehaviourType**

The only valid values are:

0 = Not Applicable

1 = BulkQuantity

3 = Cumulative

4 = DeltaData

6 = Indicating

9 = Summation

12 = Instantaneous

**CommodityType**

The only valid values are:

0 = Not Applicable

1 = Electricity secondary metered value (a premise meter is typically a secondary meter)

2 = Electricity primary metered value

4 = Air

7 = NaturalGas

8 = Propane

9 = PotableWater

10 = Steam

11 = WasteWater

12 = HeatingFluid

13 = CoolingFluid

**ConsumptionTierType**

The only valid values are:

0 = Not Applicable

1 = Block Tier 1

2 = Block Tier 2

3 = Block Tier 3

4 = Block Tier 4

5 = Block Tier 5

6 = Block Tier 6

7 = Block Tier 7

8 = Block Tier 8

9 = Block Tier 9

10 = Block Tier 10

11 = Block Tier 11

12 = Block Tier 12

13 = Block Tier 13

14 = Block Tier 14

15 = Block Tier 15

16 = Block Tier 16

**CurrencyCode**

Follows codes defined in ISO 4217. Full list at tiny.cc/4217.

0 - Not Applicable

36 - Australian Dollar

124 - Canadian Dollar

840 - US Dollar

978 - Euro

**DataQualifierType**

The only valid values are:

0 = Not Applicable

2 = Average

8 = Maximum

9 = Minimum

12 = Normal

**DateTimeInterval**  «Compound»

Interval of date and time. End is not included because it can be derived from the start and the duration.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **start** | *TimeType* | Date and time that this interval started. |
| **duration** | *UInt32* | Duration of the interval, in seconds. |

**ElectricPowerQualitySummary**

A summary of power quality events. This information represents a summary of power quality information typically required by customer facility energy management systems. It is not intended to satisfy the detailed requirements of power quality monitoring. All values are as defined by measurementProtocol during the period. The standards typically also give ranges of allowed values; the information attributes are the raw measurements, not the "yes/no" determination by the various standards. See referenced standards for definition, measurement protocol and period.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **flickerPlt** | *Int48* | A measurement of long term Rapid Voltage Change in hundredths.  flickerPlt is derived from 2 hours of Pst values (12 values combined in cubic relationship). |
| **flickerPst** | *Int48* | flickerPst is a value measured over 10 minutes that characterizes the likelihood that the voltage fluctuations would result in perceptible light flicker. A value of 1.0 is designed to represent the level that 50% of people would perceive flicker in a 60 watt incandescent bulb.  The value reported is represented as an integer in hundredths. |
| **harmonicVoltage** | *Int48* | A measurement of the Harmonic Voltage during the period. For DC, distortion is with respect to a signal of zero Hz. |
| **longInterruptions** | *Int48* | A count of Long Interruption events (as defined by measurementProtocol) during the summary interval period. |
| **mainsVoltage** | *Int48* | A measurement of the Mains [Signaling] Voltage during the summary interval period in uV. |
| **measurementProtocol** | *UInt8* | A reference to the source standard used as the measurement protocol definition.  Examples are:  0 = "IEEE1519-2009"  1 = "EN50160" |
| **powerFrequency** | *Int48* | A measurement of the power frequency during the summary interval period in uHz. |
| **rapidVoltageChanges** | *Int48* | A count of Rapid Voltage Change events during the summary interval period |
| **shortInterruptions** | *Int48* | A count of Short Interruption events during the summary interval period |
| **summaryInterval** | *DateTimeInterval* | Interval of summary period |
| **supplyVoltageDips** | *Int48* | A count of Supply Voltage Dip events during the summary interval period |
| **supplyVoltageImbalance** | *Int48* | A count of Supply Voltage Imbalance events during the summary interval period |
| **supplyVoltageVariations** | *Int48* | A count of Supply Voltage Variations during the summary interval period |
| **tempOvervoltage** | *Int48* | A count of Temporary Overvoltage events (as defined by measurementProtocol) during the summary interval period |

**ElectricPowerUsageSummary**

Summary of usage for a billing period

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **billingPeriod** | *DateTimeInterval* | The billing period to which the included measurements apply |
| **billLastPeriod** | *Int48* | The amount of the bill for the previous period , in millionths of the currency specified in the ReadingType for this reading (e.g. 840 = USD, US dollar). |
| **billToDate** | *Int48* | The bill amount related to the billing period as of the date received, in millionths of the currency specified in the ReadingType for this reading. (e.g. 840 = USD, US dollar). |
| **costAdditionalLastPeriod** | *Int48* | Additional charges from the last billing period, in millionths of the currency specified in the ReadingType for this reading. (e.g. 840 = USD, US dollar). |
| **currency** | *CurrencyCode* | The ISO 4217 code indicating the currency applicable to the bill amounts in the summary. See list at http://www.unece.org/cefact/recommendations/rec09/rec09\_ecetrd203.pdf |
| **currentBillingPeriodOverAllConsumption** | *SummaryMeasurement* | The total consumption for the billing period |
| **currentDayLastYearNetConsumption** | *SummaryMeasurement* | The amount of energy consumed one year ago interpreted as same day of week same week of year (see ISO 8601). |
| **currentDayNetConsumption** | *SummaryMeasurement* | Net consumption for the current day (delivered - received) |
| **currentDayOverallConsumption** | *SummaryMeasurement* | Overall energy consumption for the current day |
| **peakDemand** | *SummaryMeasurement* | Peak demand recorded for the current period |
| **previousDayLastYearOverallConsumption** | *SummaryMeasurement* | The amount of energy consumed on the previous day one year ago interpreted as same day of week same week of year (see ISO 8601). |
| **previousDayNetConsumption** | *SummaryMeasurement* | Net consumption for the previous day |
| **previousDayOverallConsumption** | *SummaryMeasurement* | The total consumption for the previous day |
| **qualityOfReading** | *QualityOfReading* | Indication of the quality of the summary readings |
| **ratchetDemand** | *SummaryMeasurement* | The current ratchet demand value for the ratchet demand period |
| **ratchetDemandPeriod** | *DateTimeInterval* | The period over which the ratchet demand applies |
| **statusTimeStamp** | *TimeType* | Date/Time status of this UsageSummary |

**FlowDirectionType**

The only valid values are:

0 = Not Applicable

1 = Forward

19 = Reverse

**KindType**

The only valid values are:

0 = Not Applicable

3 = Currency

8 = Demand

12 = Energy

37 = Power

**PhaseCode**

The only valid values are:

0 = Not Applicable

129 = Phase AN

128 = Phase A

132 = Phase AB

64 = Phase BN

64 = Phase B

32 = Phase CN

32 = Phase C

224 = Phase ABC

66 = Phase BC

40 = Phase CA

**PowerOfTenMultiplierType**  «XSDsimpleType»

The only valid values are:

0 = None

1 = deca=x10

2 = hecto=x100

-3 = mili=x10-3

3 = kilo=x1000

6 = Mega=x106

-6 = micro=x10-3

9 = Giga=x109

**ServiceKind**

The only valid values are:

0 - electricity

1 - gas

2 - water

4 - pressure

5 - heat

6 - cold

7 - communication

8 - time

**SummaryMeasurement**

An aggregated summary measurement reading.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **powerOfTenMultiplier** | *PowerOfTenMultiplierType* | The multiplier part of the unit of measure, e.g. "kilo" (k) |
| **timeStamp** | *TimeType* | The date and time (if needed) of the summary measurement. |
| **uom** | *UomType* | The units of the reading, e.g. "Wh" |
| **value** | *UInt48* | The value of the summary measurement. |

**TOUType**

The only valid values are:

0 = NotApplicable

1 = TOU A

2 = TOU B

3 = TOU C

4 = TOU D

5 = TOU E

6 = TOU F

7 = TOU G

8 = TOU H

9 = TOU I

10 = TOU J

11 = TOU K

12 = TOU L

13 = TOU M

14 = TOU N

15 = TOU O

**TimeAttributeType**

The only valid values are:

0 = Not Applicable

1 = 10-minute

2 = 15-minute

4 = 24-hour

5 = 30-minute

7 = 60-minute

11 = Daily

13 = Monthly

15 = Present

16 = Previous

24 = Weekly

32 = ForTheSpecifiedPeriod

79 = Daily30minuteFixedBlock

**TimeType**  «XSDsimpleType»

Time is a signed 64 bit value representing the number of seconds since 0 hours, 0 minutes, 0 seconds, on the 1st of January, 1970.

**UomType**

The only valid values are:

0 = Not Applicable

5 = A (Current)

29 = Voltage

31 = J (Energy joule)

33 = Hz (Frequency)

38 = Real power (Watts)

42 = m3 (Cubic Meter)

61 = VA (Apparent power)

63 = VAr (Reactive power)

65 = Cos? (Power factor)

67 = V² (Volts squared)

69 = A² (Amp squared)

71 = VAh (Apparent energy)

72 = Real energy (Watt-hours)

73 = VArh (Reactive energy)

106 = Ah (Ampere-hours / Available Charge)

119 = ft3 (Cubic Feet)

122 = ft3/h (Cubic Feet per Hour)

125 = m3/h (Cubic Meter per Hour)

128 = US gl (US Gallons)

129 = US gl/h (US Gallons per Hour)

130 = IMP gl (Imperial Gallons)

131 = IMP gl/h (Imperial Gallons per Hour)

132 = BTU

133 = BTU/h

134 = Liter

137 = L/h (Liters per Hour)

140 = PA(gauge)

155 = PA(absolute)

169 = Therm

**IntervalBlock**

Time sequence of Readings of the same ReadingType.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **interval** | *DateTimeInterval* | Specifies the time period during which the contained readings were taken. |

**IntervalReading**

Specific value measured by a meter or other asset. Each Reading is associated with a specific ReadingType.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **cost** | *UInt48* | Specifies a cost associated with this reading, in millionths of the currency specified in the ReadingType for this reading. (e.g. 840 = USD, US dollar) |
| **timePeriod** | *DateTimeInterval* | The date time and duration of a reading. If not specified, readings for each "intervalLength" in ReadingType are present.  (this is a bit confusing as it only provides the length and NOT the where the first one starts). |
| **value** | *UInt48* | Value in units specified by ReadingType |

**MeterReading**

Set of values obtained from the meter.

**QualityOfReading**

List of codes indicating the quality of the reading, using specification:

<system> "." <categorization> "." <index>

where <system> is one of:

0 - not applicable

1 - end device

2 - metering collection system

3 - meter data management system

<categorization> "." <index> is one of:

0.0 - valid

0.1 - validated

7.0 - manually edited

8.0 - estimated

10.0 - questionable

11.0 - derived deterministically

11.1 - derived with inference

12.0 - projected

**ReadingQuality**

Quality of a specific reading value or interval reading value. Note that more than one Quality may be applicable to a given Reading. Typically not used unless problems or unusual conditions occur (i.e., quality for each Reading is assumed to be 'Good' (valid) unless stated otherwise in associated ReadingQuality).

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **quality** | *QualityOfReading* | Quality, to be specified if different than 'raw'.  The specific format is specified per the standard is defined in QualityOfReading.  (this is inconsistent with the "Quality of Reading" definition) 'raw' is not defined -- needs to be consistent withdefaultQuality too) |

**ReadingType**

Characteristics associated with all Readings included in a MeterReading.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **accumulationBehaviour** | *AccumulationBehaviourType* | Code indicating how value is accumulated over time for Readings of ReadingType. The list of valid values per the standard are defined in AccumulationBehaviorType.  Examples are:  0 = Not Applicable  1 = BulkQuantity  3 = Cumulative |
| **commodity** | *CommodityType* | Code for commodity classification of Readings of ReadingType. The valid values per the standard are defined in CommodityType.  Examples are:  0 = Not Applicable  1 = Electricity secondary metered value (a premise meter is typically a secondary meter)  2 = Electricity primary metered value  4 = Air  7 = NaturalGas |
| **consumptionTier** | *ConsumptionTierType* | Code for consumption tier associated with a Reading of ReadingType. The valid values are define in ConsumptionTierType.  Examples are:  0 = Not Applicable  1 = Block Tier 1  2 = Block Tier 2 |
| **currency** | *CurrencyCode* | Code for the currency for costs associated with this ReadingType. The valid values per the standard are defined in CurrencyCode.  Examples are:  0 - Not Applicable  36 - Australian Dollar  124 - Canadian Dollar  840 - US Dollar  978 - Euro |
| **dataQualifier** | *DataQualifierType* | Code describing a salient attribute of Readings of ReadingType. Valid values per the standard are defined in DataQualifierType.  Examples are:  0 = Not Applicable  2 = Average |
| **defaultQuality** | *QualityOfReading* | Default value to be used if no value of ReadingQuality.quality is provided.  Specific format and valid values per the standard are specified in QualityOfReading. |
| **flowDirection** | *FlowDirectionType* | Direction associated with current related Readings. valid values per the standard are defined in FlowDirectionType.  Examples are:  0 = Not Applicable  1 = Forward  19 = Reverse |
| **kind** | *KindType* | Code for general classification of a Reading of ReadingType. Valid values per the standard are defined in KindType.  Examples are:  0 = Not Applicable  3 = Currency  8 = Demand |
| **intervalLength** | *UInt32* | Default interval length specified in seconds for Readings of ReadingType. |
| **phase** | *PhaseCode* | Code for phase information associated with Readings of ReadingType. Valid values per the standard are defined in PhaseCode.  Examples are:  0 = Not Applicable  129 = Phase AN  128 = Phase A  132 = Phase AB |
| **powerOfTenMultiplier** | *PowerOfTenMultiplierType* | Code for the power of ten multiplier which, when used in combination with the uom, specifies the actual unit of measure for Readings of ReadingType. Valid values per the standard are defined in PowerOfTenMultiplierType.  Examples are:  0 = None  1 = deca=x10  2 = hecto=x100  -3 = mili=x10-3 |
| **timeAttribute** | *TimeAttributeType* | Code used to specify a particular type of time interval method for Readings of ReadingType. Valid values per the standard are defined in TimeAttributeType.  Examplesare:  0 = Not Applicable  1 = 10-minute  2 = 15-minute |
| **tou** | *TOUType* | Code for the TOU type of Readings of ReadingType. valid values per the standard are defined in TOUType.  Examples are:  0 = NotApplicable  1 = TOU A  2 = TOU B |
| **uom** | *UomType* | Code for the base unit of measure for Readings of ReadingType. Used in combination with the powerOfTenMultiplier to specify the actual unit of measure. Valid values per the standard are defined in UomType.  Examples are:  0 = Not Applicable  5 = A (Current)  29 = Voltage |











































### REQ.21.4.2 Additional Models in Support of Services

**REQ.21.4.2.2 Authorization**

An Authorization is a Customer grant of 3rd Party access to specific resources. The attributes of this object are listed below. The structure and format of these fields is defined by OAuth.

* Information consumer (Third Party) identifier (Consumer Key)
* Resource subject (Scope)  
  Specifies the resource to which access from the information provider is granted by the customer to the information consumer
* Authentication Token (Access Token)  
  One or more token / secret pairs proving the identity of the requester to be the Customer associated with the resource. Different methods may be defined against which token / secret pairs are created and verified.
* Authorized period
* Status (requested, valid, invalid, error, unavailable)

## REQ.21.5 Related Model Business Practices

### REQ.21.5.1 Conformance to REQ.18 Energy Usage Information Model

The ESPI services conform to REQ.18 Energy Usage Information Model (PAP10), due to the ability to directly transform between the models without loss of precision, and without external inputs. The ESPI model uses the same concept names and structures as NAESB PAP10 EUI.

## REQ.21.6 Technical Implementation

### REQ.21.6.1 Protocol Specifications

ESPI endpoints shall implement the following protocol aspects.

#### REQ.21.6.1.1 Security

Providers of ESPI services, including both data custodian and third party, are responsible for protecting their systems, networks, and interface endpoints against threats, as recommended in NIST-IR and Security Profile For Third Party Data Access.

#### REQ.21.6.1.1.1 Encryption

Establishment of mutually-authenticated encrypted channels shall be performed using HTTP/S, as documented in IETF RFC 2818, over which data may be securely transferred between data custodian and third party.

#### REQ.21.6.1.1.2 User Authorization

OAuth, as documented in IETF RFC 5849, shall be used for authorization grant and access by Retail Customers and Authorized Third Parties to shared Data Custodian resources. This protocol results in access tokens that can be used to subscribe to specific user data, or to request it immediately, if supported.

### REQ.21.6.2 Communication Specifications

This section defines the expected behavior of implementations using REST. This style uses HTTP methods as verbs, and URIs as nouns.

ESPI endpoints shall exhibit behavior described by Atom, IETF RFC 4287, with the following modifications.

* Representations shall be identified as media type “application/xml”
* ESPI namespace and types (<http://naesb.org/espi>) shall be used
* espi:IdentifiedObject and specializations implement atom:entry
* espi:List and specializations implement atom:feed
* espi:mRID implements atom:id
* espi:href implements atom:link (rel=“self”)
* espi:description implements atom:title
* espi:updated implements atom:updated
* espi:published implements atom:published
* Associated objects implement atom:link (rel=“related”)
* Status codes and other behavior related to the elements above are to be implemented as defined in Atom.
* Other definitions in Atom are not used.

ESPI endpoints shall use HTTP and/or HTTPS, IETF RFC 2616 and 2818, to expose ESPI resources using the following method conventions.

* POST to a List shall add the new object to the List
* PUT to an IdentifiedObject shall update it by replacement
* GET to a List shall return the most recently updated objects in it
* GET to an IdentifiedObject shall return its full representation
* DELETE to an IdentifiedObject shall delete the object

ESPI endpoints may choose to allow read-only access or read-write, and can allow users to specify this setting upon authorization.

URIs should be kept as short as possible, and must not exceed 255 bytes.

Relative URIs may be used when resources are on the same host. Additional definition regarding URIs and HTTP/S shall follow the IETF specifications.

A URI example is provided below.

* https://espi.datacustodian.com/{third\_party\_id}/Batch/{#}

Since all URIs are opaque references, and the current preferred reference can be obtained from an mRID, there is no mandated form. However, it may be useful to organize them hierarchically, in order to define URIs for the appropriate List containers (feeds), and to manage permissions. Typically, feeds (Lists) will have a named path segment, and entries (IdentifiedObjects) will have a numbered instance segment, represented by {#} above. Instance segments may use mRID, but may use name, since they only need to be unique within the scope of the containing URI. URIs should be as persistent as possible, but they may change. mRID, however, like atom:id, shall not change, even if the resource is moved or replicated. Clients accessing out-of-date URIs may be redirected, but if they are not, may need to request the current preferred resource location.

The following query parameters may be used to filter the resources returned by a feed. These use typical “?name=value[&…]” syntax.

* published-max, published-min
* updated-max, updated-min
* max-results
* start-index

In addition, mRID or name can be passed as a path element to a List URL, to retrieve that object from the collection.

Date and time values for the above parameters shall use RFC 3339 format.

Results from List and specializations are sorted by updated descending.

Clients and servers shall use the following guidance regarding required elements.

* espi:mRID, espi:updated, and espi:published shall be provided, except on creation using POST
* espi:href shall be provided when the resource is available via that URL, except on creation using POST
* Associated object references may be represented using espi:mRID inside the appropriate tags and no other elements or attributes.
* Associated objects may be represented using full representation in containment. The objects are still separate when represented this way.
* All other elements and attributes shall be provided when values exist.

#### REQ.21.6.3 Examples

The following examples show the creation, retrieval, update, and deletion of an object within a List.

Upon authorization of a resource, the OAuth “scope” attribute will contain the mRID or URI of the resource. With it, the client can request a subscription to it as in the example below. Note that this mRID is a reference to the target resource of the subscription. (The resource being subscribed to) The mRID of the Subscription is not specified, since it is assigned by the data custodian. Note also that signed OAuth parameters must be included in the header to prove authorization.

POST /Subscription HTTP/1.1

Host: espi.datacustodian.com

Content-Type: application/xml

Content-Length: 163

<?xml version="1.0" encoding="UTF-8"?>

<Subscription xmlns="http://naesb.org/espi">

<List>

<mRID>e5c056e0819611e0b2780800200c9a66</mRID>

</List>

</Subscription>

The server may refuse this request, if valid authorization was not provided, in which case result “401 Not Authorized” or similar, shall be returned.

If the request is accepted, the server shall respond with the full resource representation, as in the response example below.

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 335

<?xml version="1.0" encoding="UTF-8"?>

<Subscription

xmlns=”http://naesb.org/espi”

href=”/Subscription/7f23”>

<published>1305758077</published>

<updated>1305758077</updated>

<mRID>e69c4c2528854de0a3d8d29b5f823b79</mRID>

<name>7f23</name>

<List

href=”/9b6c7063”>

<mRID>e5c056e0819611e0b2780800200c9a66</mRID>

</List>

</Subscription>

Retrieval of the object is performed using GET. The example below shows the request – the response would be the same as the response to POST above.

GET /Subscription/7f23 HTTP/1.1

Host: espi.datacustodian.com

Deletion uses DELETE, as in the example below. Again, authorization parameters must also be included.

DELETE /Subscription/7f23 HTTP/1.1

Host: espi.datacustodian.com

Response is simply the status of the request, as below.

HTTP/1.1 200 OK

Batch processing involves inclusion of the “Batch” attributes with regular objects in a list, as in the example below. This example shows delivery of new objects.

<?xml version="1.0" encoding="UTF-8"?>

<Batch xmlns="http://naesb.org/espi">

<UsagePointList href="/9b6c7063/UsagePoint">

<published>1325484000</published>

<updated>1325484000</updated>

<mRID>299215f0831011e09d780800200c9a66</mRID>

<description>Collection 1</description>

<UsagePoint href="/9b6c7063/UsagePoint/01">

<published>1325484000</published>

<updated>1325484000</updated>

<mRID>c990b150832011e09d780800200c9a66</mRID>

<name>01</name>

<description>Elm St.</description>

<MeterReadingList href="/9b6c7063/UsagePoint/01/MeterReading">

<published>1325484000</published>

<updated>1325484000</updated>

<mRID>f2034e90832011e09d780800200c9a66</mRID>

<description>Elm St. Readings</description>

<MeterReading href="/9b6c7063/UsagePoint/01/MeterReading/01">

<published>1325484000</published>

<updated>1325484000</updated>

<mRID>f2034e91832011e09d780800200c9a66</mRID>

<name>01</name>

<description>Hourly Energy Consumption</description>

<IntervalBlockList href="/9b6c7063/UsagePoint/01/MeterReading/01/IntervalBlock">

<published>1325484000</published>

<updated>1325484000</updated>

<mRID>f2034e92832011e09d780800200c9a66</mRID>

<IntervalBlock href="/9b6c7063/UsagePoint/01/MeterReading/01/IntervalBlock/0173">

<published>1325484000</published>

<updated>1325484000</updated>

<mRID>f2034e93832011e09d780800200c9a66</mRID>

<name>0173</name>

<interval>

<duration>86400</duration>

<start>1325397600</start>

</interval>

<IntervalReading>

<cost>3000000</cost>

<timePeriod>

<duration>3600</duration>

<start>1325397600</start>

</timePeriod>

<value>383</value>

</IntervalReading>

<IntervalReading>

<cost>3000000</cost>

<timePeriod>

<duration>3600</duration>

<start>1325401200</start>

</timePeriod>

<value>427</value>

</IntervalReading>

</IntervalBlock>

</IntervalBlockList>

<ReadingType href="/ReadingType/07">

<published>1317445200</published>

<updated>1317445200</updated>

<mRID>2557def0832111e09d780800200c9a66</mRID>

<name>07</name>

<description>Energy Delivered (kWh)</description>

<accumulationBehaviour>4</accumulationBehaviour><!--DeltaData-->

<commodity>1</commodity><!--Electricity-->

<consumptionTier>0</consumptionTier><!--N/A-->

<currency>0</currency><!--N/A-->

<dataQualifier>12</dataQualifier><!--Normal-->

<flowDirection>1</flowDirection><!--Forward-->

<kind>12</kind><!--Energy-->

<phase>0</phase><!--N/A-->

<powerOfTenMultiplier>3</powerOfTenMultiplier><!--kilo-->

<timeAttribute>0</timeAttribute><!--N/A-->

<tou>0</tou><!--N/A-->

<uom>72</uom><!--Watt hours-->

</ReadingType>

</MeterReading>

</MeterReadingList>

<ServiceCategory>

<kind>0</kind><!--electricity-->

</ServiceCategory>

<status>1</status><!--on-->

</UsagePoint>

</UsagePointList>

</Batch>

#### REQ.21.6.4 Conformance

Conformant Data Custodian implementations must implement the following:

* Security
  + Server certificates and mutually authenticated HTTPS
  + Accept requests to OAuth endpoints
* Subscriptions
  + Accept POST to Subscription, Batch resource
  + Allow subscriptions to authorized resources
* Delivery
  + Accept GET to Batch resource, specific to each Authorized Third Party.
  + Optionally support POST to Authorized Third Party Notification resource
  + Optionally support POST to Authorized Third Party Batch resource
  + Optionally support GET of resources directly
* Content
  + No specific subset of domain information elements is required for conformance, but those that are made available must use the format defined.
  + Any additional information elements must be placed in extension elements, must be optional, must use a different namespace, and must be requested and refused for standardization in a future release.

Conformant Third Party implementations must implement the following:

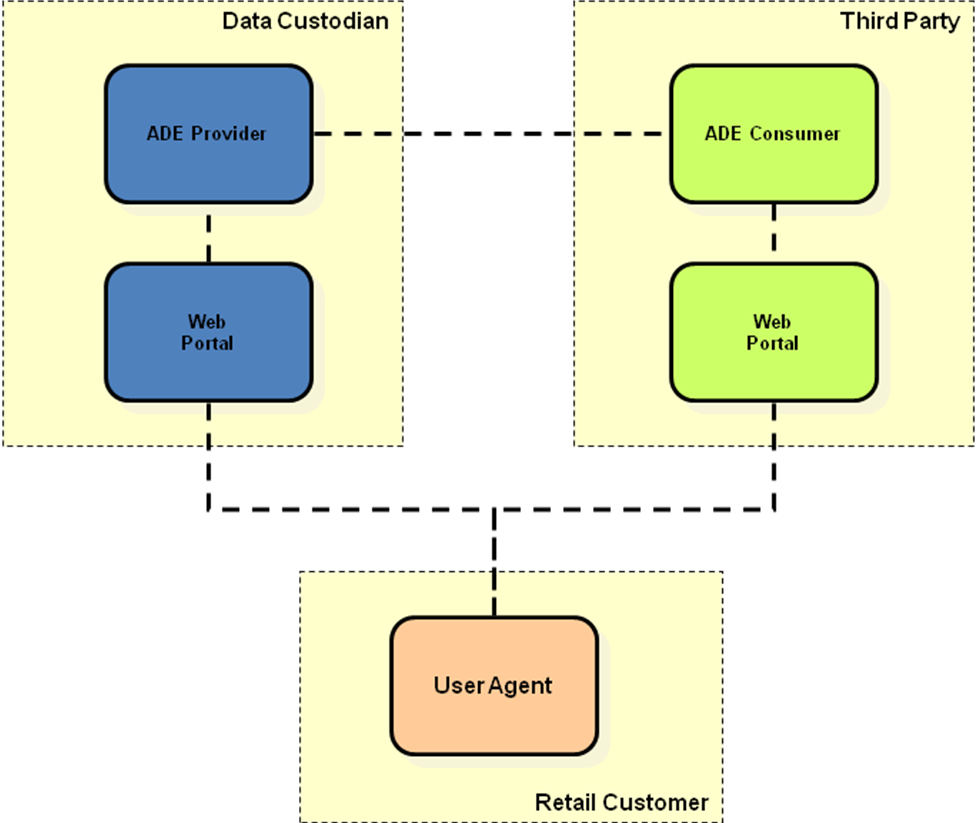
* Security
  + Server certificates and mutually authenticated HTTPS
  + Make requests to OAuth endpoints
* Subscriptions
  + Submit POST requests to Subscription, Batch resource
* Delivery
  + Submit GET to Batch resource
  + Optionally accept POST to Authorized Third Party Notification resource
  + Optionally accept POST to Authorized Third Party Batch resource
  + Optionally GET resources directly

# Appendices

This section contains informative descriptions and requirements derived by the committee for these service definitions.

# A. Overview

The scope of these Model Business Practices includes authorization by the Retail Customer and the Automatic Data Exchange of the EUI to the Authorized Third Party in accordance with parameters (e.g. term of access, type of data, quantity of data, etc.) determined by the Retail Customer subject to the Governing Documents and Applicable Regulatory Authority. The diagram below shows the logical components involved in this authorization and data exchange process. Note that while the authorization process shown in this figure is made using a web browser, the services provided by the Authorized Third Party are not required to use a web browser to deliver such services.



**Figure : Overview of Logical Components**

# B. Use Cases

This section presents a superset of the use cases that are informative of the third party data access relationship. Alternative use cases are presented for certain activities that can be achieved in different ways, such as the delivery of shared resource information by push or by pull.

The concept of a Shared Resource Key is used throughout these use cases. A Shared Resource Key is a token used to uniquely identify an instance of a Third Party data access relationship (i.e., each Retail Customer-Data Custodian-Third Party combination for a particular resource will have a unique Shared Resource Key). A Shared Resource Key contains no PII regarding the Retail Customer and so can be freely shared among all three roles without unnecessary disclosure of sensitive information. Once the relationship is established, inclusion of a Shared Resource Key in an interaction is sufficient to identify a specific Third Party data access relationship.

Each use case contains the following sections:

* Use Case Description: This is a summary of the use case, describing the overall purpose.
* Pre-Conditions: These are conditions that must be true for the use case to be successfully executed.
* Invariant Conditions: These are properties that will be true any time the use case is initiated, regardless of whether it terminates successfully.
* Post-Conditions:These are properties that will be true only if the use case terminates successfully. This requires that all preconditions and all condition checks (e.g., for validity of a request) be satisfied during execution of the use case.
* Basic Path Scenario: This defines the series of steps undertaken by each role during successful execution of the use case. The scenario is depicted graphically in a Unified Modeling Language (UML) sequence diagram and each step is summarized in text.

The following use cases are informative and not normative.



**Figure :** ESPI Use Case Diagram

**1: Third Party Establishes Relationship With Data Custodian**



**Figure :** Third Party Establishes Relationship With Data Custodian

**Description**

A Third Party service provider wants to register with a Data Custodian to provide services to Retail Customers with data stored at the Data Custodian.

**Pre-Condition:** Third Party had demonstrated that it meets security and privacy requirements as specified by governing documents.

**Invariant Constraint:** No resource data or personal data is provided to the Third Party by the Data Custodian as part of this interaction.

**Post-Condition:** A Shared Identity Key is generated to allow the Third Party to identify its identity to Data Custodian.

**Post-Condition:** The Third Party has permission to get specified resource data from the Data Custodian with permission of a Retail Customer.

**Scenario:** Basic Path

1. The Third Party wishes to provide value added services to Retail Customers with data stored by the Data Custodian.
2. Third Party requests that the Data Custodian establish relationship.
3. Third Party provides proof that they meet the governing documents’ requirements for data security and privacy protection.
4. Third Party provides description of the services it wishes provide for Retail Customers.
5. The Data Custodian generates an Identity Key for the Third Party. Third Party will use this key to identify itself during Use Cases 2 through 12.
6. The Data Custodian adds the Third Party to its list of available services it presents to Retail Customers in Use Case 2.
7. Third Party adds Data Custodian to its list of Data Custodians it presents in Use Case 2.
8. Third Party persists the Identity Key.
9. As needed, Third Party checks their ability to connect to the service, and obtains the current status of the service.

**2: Retail Customer Authorizes Third Party Resource Access via Data Custodian**



**Figure :** Retail Customer Authorizes Third Party Resource Access via Data Custodian

**Description**

A Retail Customer wants to grant permission for a Data Custodian to share a particular data resource with a Third Party. The Retail Customer initiates the process through the Data Custodian.

**Pre-Condition:** Retail Customer has established accounts with Data Custodian and Third Party.

**Pre-Condition:** Third Party has an established account with Data Custodian.

**Pre-Condition:** Data Custodian and Third Party have published and made Retail Customer aware of their privacy policy related to collection and handling of customer information.

**Invariant Constraint:** No resource data or personal data is provided to the Third Party by the Data Custodian as part of this interaction.

**Post-Condition:** A Shared Resource Key is generated to allow all three roles to refer to the same shared resource without disclosing PII. This key is known to all three roles.

**Post-Condition:** The Third Party has the Retail Customer's permission to get the specified resource data from the Data Custodian.

**Post-Condition:** The Data Custodian sends the Retail Customer confirmation of establishment of the Third Party data access relationship.

**Scenario:** Basic Path

1. The Retail Customer decides to grant permission for the Data Custodian to share their resource data with the Third Party
2. (Optional) Retail Customer finds their appropriate Data Custodian from Third Party, and navigates to the appropriate place to begin establishment of sharing relationship.
3. Retail Customer requests that the Data Custodian establish a new data access relationship.
4. Data Custodian presents the Retail Customer with a list of resources that can be shared with Third Parties. Any additional attributes (e.g., duration for which permission should be granted) that can be selected are also presented.
5. Retail Customer selects a resource to share, sets any available attributes for the relationship, and specifies a Third Party that is known to the Data Custodian. Selecting these parameters and completing the interaction indicates permission for the Data Custodian to grant the specified Third Party access to the specified shared resource.
6. The relationship will only be created if the Data Custodian accepts the selections for the Third Party (e.g., a Data Custodian may constrain access to certain resource attributes depending on resource sensitivity).
7. Data Custodian generates a Shared Resource Key (Request Token) to begin creation of this relationship and provides it to the Third Party. Each Shared Resource Key is unique to the relationship between a Retail Customer, Data Custodian, Third Party, and specific data resource.
8. Third Party requests authorization of the token by the Retail Customer, via the Data Custodian.
9. Retail Customer authenticates with Data Custodian and authorizes the Request Token.
10. Third Party exchanges the authorized Request Token for an Access Token from the Data Custodian.
11. Third Party and Data Custodian persist the Authorization, associating it with its identity of the Retail Customer.

**3: Retail Customer Modifies Resource Authorization**



**Figure :** Retail Customer Modifies Resource Authorization

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party and wants to either extend or restrict the permissions associated with that relationship.

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship.

**Invariant Constraint:** No resource data or personal data is provided to the Third Party by the Data Custodian as part of this interaction.

**Post-Condition:** Future interactions between the Data Custodian and the Third Party with respect to the specified resource are governed by the modified permissions.

**Post-Condition:** The Third Party handles any data not allowed by the termination of the relationship in the manner specified in any service agreements among the parties in the relationship (e.g., all instances of the data in control of the Third Party are deleted within**Post-Condition:** The Data Custodian sends the Retail Customer confirmation of modification of the permissions of the Third Party data access relationship.

**Scenario:** Basic Path

1. Retail Customer chooses to modify relationship permissions with the Data Custodian.
2. Data Custodian presents the Retail Customer with a list of resources that are shared with Third Parties. If the Retail Customer may only grant access to one resource, S2 and S3 may be skipped.
3. Retail Customer chooses particular resource whose permissions he/she wishes to modify.
4. Data Custodian provides available resource attributes and current settings to Retail Customer.
5. Retail Customer chooses new settings.
6. The new permissions governing the relationship will apply only if the Data Custodian accepts the selections for the Third Party (e.g., a Data Custodian may constrain access to certain resource attributes depending on resource sensitivity).
7. Data Custodian persists the new permissions, which will be used from this point forward to govern the relationship (until further changed or the relationship is terminated).
8. Data Custodian notifies Third Party that permissions have changed (identifying the resource by its Shared Resource Key). This notification may be immediate or deferred (e.g., as part of a resource push from Use Case 8, perhaps as part of a header).
9. Data Custodian notifies Retail Customer that permissions have been changed.
10. The Third Party handles any data not allowed by the modification of the resource authorization in the manner specified in any service agreements among the parties in the relationship.

**4: Retail Customer Revokes Resource Authorization**



**Figure :** Retail Customer Revokes Resource Authorization

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party and wants to terminate that relationship

**Pre-Condition:** Third Party has an established account with Data Custodian.

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship.

**Invariant Constraint:** No resource data or personal data is provided to the Third Party by the Data Custodian as part of this interaction.

**Post-Condition:** Both the Third Party and the Data Custodian delete the Shared Resource Key for the relationship and no future interactions are permitted for that relationship.

**Post-Condition:** The Third Party handles any data not allowed by the termination of the relationship in the manner specified in any service agreements among the parties in the relationship (e.g., all instances of the data in control of the Third Party are deleted within

**Post-Condition:** The Data Custodian sends the Retail Customer confirmation of termination of the Third Party data access relationship.

**Scenario:** Basic Path

1. Retail Customer requests that Data Custodian terminate the data access relationship.
2. Data Custodian presents the Retail Customer with a list of resources for which there are valid relationships with Third Parties. If the Retail Customer only has one valid relationship, S2 and S3 may be skipped.
3. Retail Customer chooses a resource whose relationship is to be terminated.
4. Data Custodian terminates the relationship, deleting the appropriate Shared Resource Key from its list of valid relationships.
5. Data Custodian notifies Third Party that the relationship has been terminated (identifying the relationship by its Shared Resource Key).
6. Data Custodian notifies Retail Customer that the relationship has been terminated.
7. The Third Party handles any data not allowed by the termination of the relationship, in the manner specified in any service agreements among the parties in the relationship.

**5: Data Custodian Revokes Resource Authorization**



**Figure :** Data Custodian Revokes Resource Authorization

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party. The Data Custodian wants to terminate the relationship (for whatever reason).

**Pre-Condition:** Third Party has an established account with Data Custodian.

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship.

**Invariant Constraint:** No resource data or personal data is provided to the Third Party by the Data Custodian as part of this interaction.

**Post-Condition:** Both the Third Party and the Data Custodian delete the Shared Resource Key for the relationship and no future interactions are permitted for that relationship.

**Post-Condition:** The Third Party handles any data not allowed by the termination of the relationship, in the manner specified in any service agreements among the parties in the relationship (e.g., all instances of the data in control of the Third Party are deleted within

**Post-Condition:** The Data Custodian sends the Retail Customer notification of termination of the Third Party data access relationship.

**Scenario:** Basic Path

1. Data Custodian decides to terminate relationship with Third Party.
2. Data Custodian notifies Retail Customer of termination decision; no acknowledgement or confirmation is required.
3. Data Custodian notifies Third Party of termination of the relationship, identifying the relationship by a Shared Resource Key.
4. The Third Party handles any data not allowed by the termination of the relationship, in the manner specified in any service agreements among the parties in the relationship.

**6: Third Party Terminates Relationship**



**Figure :** Third Party Terminates Relationship

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party. The Third Party determines that it no longer wants to provide services to the Retail Customer and terminates the relationship.

**Pre-Condition:** Third Party has an established account with Data Custodian.

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship.

**Invariant Constraint:** No resource data or personal data is provided to the Third Party by the Data Custodian as part of this interaction.

**Post-Condition:** Both the Third Party and the Data Custodian delete the Shared Resource Key for the relationship and no future interactions are permitted for that relationship.

**Post-Condition:** The Third Party handles any data not allowed by the termination of the relationship, in the manner specified in any service agreements among the parties in the relationship (e.g., all instances of the data in control of the Third Party are deleted within

**Post-Condition:** The Data Custodian sends the Retail Customer notification of termination of the Third Party data access relationship.

**Scenario:** Basic Path

1. Third Party decides to terminate a third party data access relationship.
2. Third Party notifies Data Custodian of termination of relationship, identifying the relationship by the Shared Resource Key.
3. An invalid request (e.g., specification of a Shared Resource Key not associated with the Third Party) will not be accepted.
4. Data Custodian deletes Shared Resource Key, terminating the relationship.
5. Data Custodian notifies the Retail Customer of termination of the relationship. No acknowledgement or confirmation is required.
6. The Third Party handles any data not allowed by the termination of the relationship, in the manner specified in any service agreements among the parties in the relationship.

**7: Third Party Establishes Subscription with Data Custodian - Asynchronous**



**Figure :** Third Party Establishes Subscription with Data Custodian - Asynchronous

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party. The Third Party establishes a *subscription* indicating the circumstances (i.e., an agreed-upon schedule and/or specification of special events) under which the Data Custodian should provide the Third Party with the relevant resource data.

Depending on the services offered by a Data Custodian, the subscription may indicate the circumstances under which the Data Custodian will send resource data or only notification that resource data is available (i.e., whether the Data Custodian supports a push or pull model). Subscriptions may be parameterized, if supported by the Data Custodian, to define preferred delivery criteria (e.g., new data whenever available or only once per day).

**Pre-Condition:** Third Party has an established account with Data Custodian.

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship.

**Invariant Constraint:** No resource data or personal data is provided to the Third Party by the Data Custodian as part of this interaction.

**Post-Condition:** The Data Custodian records a valid subscription on behalf of the Third Party. Future data availability triggers satisfying the subscription will result in the appropriate information being sent to the Third Party.

**Post-Condition:** Data Custodian sends the Third Party confirmation of its subscription request

**Post-Condition:** Data Custodian sends the Retail Customer notification of the Third Party's subscription request

**Scenario:** Basic Path

1. Third Party requests that the Data Custodian establish a new subscription.
2. Third Party provides Data Custodian with information defining the subscription request. At a minimum, this information includes a Shared Resource Key identifying the resource whose data is to be shared. The information may include additional subscription parameters, as supported by the Data Custodian.
3. The subscription will not be accepted if the Shared Resource Key is invalid.
4. The Data Custodian saves the subscription information, associating the subscription with the Shared Resource Key and the Third Party.
5. The Data Custodian notifies the Third Party that the subscription request was successful. No acknowledgement or confirmation is required.
6. The Data Custodian notifies the Retail Customer that the Third Party has completed a subscription for their resource data. No confirmation is required, as the Third Party already has permissions as indicated by the valid Shared Resource Key. If the subscription is not acceptable to the Retail Customer, Use Case 3 can be exercised to modify permissions for the Third Party.

**8: Third Party Requests Data from Data Custodian - Asynchronous**



**Figure :** Third Party Requests Data from Data Custodian - Asynchronous

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party. The Third Party requests specific resource data to be delivered with next transfer.

**Pre-Condition:** Third Party has an established account with Data Custodian

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship

**Invariant Constraint:** No resource data or personal data is provided to the Third Party by the Data Custodian as part of this interaction.

**Post-Condition:** The Data Custodian records the request on behalf of the Third Party. Future data availability triggers will result in the appropriate information being sent to the Third Party.

**Post-Condition:** Data Custodian sends the Third Party confirmation of its data request.

**Scenario:** Basic Path

1. Third Party decides to request resource data from the Data Custodian.
2. Data Custodian Checks validity of request.
3. Data Custodian queues request for next asynchronous transfer.
4. Data Custodian sends confirmation to Third Party.

**9: Data Custodian Sends (Pushes) Data to Third Party - Asynchronous**



**Figure :** Data Custodian Sends (Pushes) Data to Third Party - Asynchronous

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party. The Third Party has established a subscription for receiving the relevant resource data from the Data Custodian. This resource data is sent (pushed) to the subscribed Third Party by the Data Custodian when an event triggers indicates a need to push new resource data.

Conditions observable to the Data Custodian change, causing a data availability trigger to be checked to see if there is a need to push resource data to the Third Party. Such triggers can be caused by any of the following observable changes

* New resource data is received by the Data Custodian
* A new subscription is received by the Data Custodian
* A pre-defined interval has elapsed
* A request for resource data has been received from a Third Party

**Pre-Condition:** Third Party has an established account with Data Custodian.

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship.

**Pre-Condition:** A subscription by the Third Party to receive resource data from the Data Custodian has been established.

**Invariant Constraint:** No personal information is provided to the Third Party by the Data Custodian.

**Post-Condition:** The Data Custodian sends resource data to the subscribed Third Party.

**Post-Condition:** Only data specifically requested or modified and in a subscription is sent to the Third Party.

**Scenario:** Basic Path

1. A data availability trigger is received by the Data Custodian.
2. Data Custodian determines the Shared Resource Keys associated with the data availability trigger. It then determines if there are any subscriptions associated with the Shared Resource Key and whether the conditions of the subscription are satisfied (i.e., if it is time to send out resource data). If so, it proceeds to S3.
3. Data Custodian determines the Third Party associated with the subscriptions. This includes a check that the Third Party is still in a valid relationship with the Data Custodian and any other relevant checks prior to releasing resource data to that Third Party.
4. Data Custodian provides data resources to Third Party.
5. Third party persists data for the period specified by data retention requirements.

**10: Data Custodian Notifies Third Party of Data Availability - Asynchronous**



**Figure :** Data Custodian Notifies Third Party of Data Availability - Asynchronous

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party. The Third Party has established a subscription for receiving the relevant resource data from the Data Custodian. A Third Party is notified when new data satisfying its subscription parameters is available.

Conditions observable to the Data Custodian change, causing a data availability trigger to be checked to see if there is a need to notify a Third Party of resource data availability. Such triggers can be caused by any of the following observable changes

* New resource data is received by the Data Custodian
* A new subscription is received by the Data Custodian
* A pre-defined interval has elapsed
* A request for resource data has been received from a Third Party

**Pre-Condition:** Third Party has an established account with Data Custodian

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship.

**Pre-Condition:** Data Custodian has resource data relevant to the Third Party

**Invariant Constraint:** No personal information is provided to the Third Party by the Data Custodian

**Post-Condition:** The Data Custodian has resource data (e.g., electricity usage data) that is available for access by the Third Party

**Post-Condition:** The Data Custodian sends the Third Party notification of availability of resource data

**Scenario:** Basic Path

1. A data availability trigger event is received by the Data Custodian.
2. Data Custodian determines the Shared Resource Keys associated with the data availability trigger. The Data Custodian then determines if there are any subscriptions associated with the Shared Resource Key and whether the conditions of the subscription are satisfied (i.e., if it is time to notify a Third Party). If so, it proceeds to S3.
3. Data Custodian determines the Third Party associated with subscriptions. This includes a check that the Third Party is still in a valid relationship with the Data Custodian and any other relevant checks prior to determining that it is appropriate to send resource data to that Third Party
4. Data Custodian notifies the Third Party of the availability of resource data associated with the Shared Resource Key. Note that notification can take different forms. Notification could be sent asynchronously as soon as the trigger is evaluated. Notification for several resources could be bundled for delivery to a common Third Party. Notification could be queued, awaiting the next scheduled interaction with the Third Party (e.g., as part of a response to a regular pull from the Third Party). The essence of the use case is that a notification is prepared and delivered at some point; the specific mechanism and timing is not restricted.

**11: Third Party Receives (Pulls) Requested Data from Data Custodian - Asynchronous**



**Figure :** Third Party Receives (Pulls) Requested Data from Data Custodian - Asynchronous

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party. The Third Party requests the relevant subscribed and requested resource data from the Data Custodian, who replies with the data if the request is valid.

**Pre-Condition:** Third Party has an established account with Data Custodian

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship

**Pre-Condition:** Data Custodian has resource data relevant to the Third Party

**Invariant Constraint:** No personal data is provided to Third Parties by the Data Custodian.

**Post-Condition:** The Data Custodian replies with the requested data

**Post-Condition:** Only the requested resource data is provided by the Data Custodian

**Scenario:** Basic Path

1. Third Party receives notification or periodically attempts to pull resource data from the Data Custodian.
2. Data Custodian checks validity of request.
3. Data Custodian replies with requested and subscribed resource data to Third Party.
4. Third Party persists resource data for use in performing services for Retail Customer.

**12: Third Party Requests Data from Data Custodian - Synchronous**



**Figure :** Third Party Requests Data from Data Custodian - Synchronous

**Description**

The Retail Customer has an existing third party data access relationship with a particular Data Custodian and Third Party. The Third Party directly requests specific resource data from the Data Custodian, who replies with the requested data synchronously if the request is valid.

**Pre-Condition:** Third Party has an established account with Data Custodian

**Pre-Condition:** Retail Customer has established a Third Party data access relationship with the Data Custodian and the Third Party with respect to a particular resource, resulting in a unique Shared Resource Key identifying the relationship

**Pre-Condition:** Third Party requests authorized resource data

**Invariant Constraint:** No personal data is provided to Third Parties by the Data Custodian.

**Post-Condition:** The Data Custodian replies with the requested data.

**Post-Condition:** Only the requested resource data is provided by the Data Custodian.

**Scenario:** Basic Path

1. Third Party decides to pull resource data from the Data Custodian.
2. Third Party specifies the resource data being requested. The request must contain the Shared Resource Key. It may also contain parameters (e.g., the period over which the specified data is requested), if permitted by Data Custodian.
3. Data Custodian checks validity of request (e.g., Shared Resource Key is still valid and registered with this Third Party or validity of any additional parameters).
4. Data Custodian sends requested resource data to Third Party.
5. Third Party persists resource data for use in performing services for Retail Customer.

# C. ESPI Abstract Services

This section provides definition of the abstract services used in the use cases. These are the services that will be specified fully for the ESPI model business practice specification. The services are named using the following conventions, since not all are intended to be fully standardized.

* Underscore before the method name means “must be done, but is not standardized”
* Underscore after the method name means “optional”



**Figure:**  *-* Logical Service Interfaces

**DataCustodian**

The DataCustodian service interface contains methods to be called by Third Party in order to authorize and receive data.

**Operations**

| **Method** | **Notes** | **Parameters** |
| --- | --- | --- |
| **\_CreateThirdPartyId()** ThirdPartyId |  | ApplicationInformation [] ApplicationInformation |
| **ReadServiceStatus()** ServiceStatus |  |  |
| **CreateRequestToken()** RequestToken |  | Authorization [] authRequest |
| **Authorize()** | Provides ability for Retail Customer to authenticate and verify desire to authorize a Third Party request token. This results in a verifier to be used with CreateAccessToken. | Authorization [] authorization |
| **CreateAccessToken()** AccessToken |  | Authorization [] authRequest |
| **\_ReadAuthorizationList()** AuthorizationList |  | RetailCustomerId [] customerID |
| **UpdateAuthorization()** boolean |  | Authorization [] authorization |
| **CreateSubscription()** boolean |  | Authorization [] authorization |
| **DeleteSubscription()** boolean |  | Authorization [] authorization |
| **RequestData()** boolean |  | Authorization [] authorization |
| **ReadData()** |  | BatchLocation [] batch |
| **ReadData\_()** DataResource |  | Authorization [] authorization  DateTimeInterval [] requestedInterval |

**RetailCustomer**

**Operations**

| **Method** | **Notes** | **Parameters** |
| --- | --- | --- |
| **\_UpdateAuthorizationNotification()** boolean |  | Authorization [] authorization |
| **\_RequestAuthorization()** boolean |  | Authorization [] authorization |

**ThirdParty**

**Operations**

| **Method** | **Notes** | **Parameters** |
| --- | --- | --- |
| **\_ReadDataCustodianList()** DataCustodianList |  | RetailCustomerId [] reatilCustomerID |
| **\_ProvideAuthorization()** boolean |  | Authorization [] authorization |
| **NotifyUpdateAuthorization\_()** boolean |  | Authorization [] authorization |
| **NotifyData\_()** boolean | This optional method can be implemented in order to avoid having to poll for new data. It is called by the Data Custodian to indicate that requested authorized data is available via ReadData. | BatchList [] batchList |
| **UpdateData\_()** boolean |  | Batch [] data |

## Logical Information Model

This section contains descriptions of the data elements used in the abstract services.



**Figure** : ESPI Logical Information Model

**AccessToken**

AccessToken is a shared key representing the relationship between a RetailCustomer, DataCustodian, and ThirdParty for a particular data resource.

**ApplicationInformation**

Includes (non-standardized) information about the Third Party Application requesting access to the DataCustodian services. Information requested may include items such as Organization Name, Website, Contact Info, Application Name, Description, Icon, Type, default Notification and Callback endpoints, and may also include agreement with terms of service.

**Authorization**

Description of an authorization. Includes the information constraining and defining access to the Data. May include additional security elements, such as signature, timestamp, nonce, etc. as well as callback to allow redirection of the user agent.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **thirdPartyID** | *ThirdPartyId* |  |
| **requestToken** | *Token* |  |
| **accessToken** | *Token* |  |
| **data** | *DataResource* |  |
| **validityInterval** | *DateTimeInterval* |  |

**Batch**

**BatchList**

BatchList is a container to refer to a multiple batches of data.

**BatchLocation**

Specifies the location of a specific batch of data.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **reference** | *String* |  |

**CurrentStatus**

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **normal** |  |  |
| **unavailable** |  |  |
| **terminated** |  |  |

**DataCustodianId**

An identifier for the Data Custodian.

**DataResource**

Generalization of any data object that can be exchanged.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **operation** | *byte* |  |

**RequestToken**

RequestToken is the first step toward obtaining an authorized AccessToken.

**RequestorID**

Generic superclass for identifiers.

**RetailCustomerId**

An identifier for the Retail Customer.

**ServiceStatus**

Contains the current status of the service.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **currentStatus** | *CurrentStatus* |  |

**Subscription**

**ThirdPartyId**

An identifier for the Third Party.

**Token**

**UsagePointList**

Container to hold multiple UsagePoint objects.

# D. Goals for the Capability of the Standard

### Sample Business Process Guidelines

* ESPI Discovery
  + Authorized Third Party and Data Custodian agree to service level agreements (SLAs) and configure ESPIs services (REQ 21.1.3, REQ 21.1.16, REQ 21.1.18)
  + Authorized Third Party retrieves listing of supported operations with extensions and versions (REQ 21.1.14)
  + Data Custodian retrieves listing of supported operations with extensions and versions (REQ 21.1.14)
  + Authorized Third Party and Data Custodian subscribe to notifications
* Retail Customer Authorization
  + Retail Customer Grants Permission (REQ.21.1.4, REQ.21.1.7, REQ.21.1.8, REQ.21.1.9, REQ.21.1.10, REQ.21.1.15)
  + Retail Customer Extends Permission (REQ.21.1.9)
  + Retail Customer Terminates Permission (REQ.21.1.10)
* ADE Publication
  + Authorized Third Party Requests or Subscribes to Retail Customer EUI
  + Data Custodian Provides Retail Customer EUI to Authorized Third Party (REQ.21.1.20)

E. Cyber Security Requirements

*[These will be provided by CSWG/ASAP as contribution assignee: MJB Coordinator]*

F. Model Conformance Information

The following table provides information about the elements included in ESPI and their relation to the NAESB PAP10 Energy Usage Information Model as well as the IEC TC57 CIM Model. Harmonization across these models is a goal of this recommendation, as is aligning with other usage information interfaces, including Smart Energy Profile 2.0.

| **PAP10 EUI Model Element** | **Type** | **ESPI Model Element** | **Type** | **CIM Notes** |
| --- | --- | --- | --- | --- |
| CustomerAuthorisation.name | String | (OAuth) access\_token |  | N/A |
| CustomerAuthorisation.validityInterval | DateTimeInterval | (OAuth 2.0) expires\_in |  | N/A |
| UsagePoint.name | String | UsagePoint.mRID | HexBinary128 | Same |
| UsagePoint.description | String | UsagePoint.description | String32 | Same |
|  |  | UsagePoint.status | UInt8 | connectionState |
| ServiceCategory.kind | ServiceKind | ServiceCategory.kind | ServiceKind | Same |
| ServiceKind.electricity |  | ServiceKind 0 |  | (encoded) |
| ServiceKind.gas |  | ServiceKind 1 |  | (encoded) |
| ServiceKind.water |  | ServiceKind 2 |  | (encoded) |
| MeterReading.name | String | MeterReading.mRID | HexBinary128 | Same |
|  |  | MeterReading.description | String32 | Same |
| ReadingType.name | String | ReadingType.mRID | HexBinary128 | Same |
|  |  | ReadingType.description | String32 | Same |
| ReadingType.defaultQuality | QualityOfReading | ReadingType.defaultQuality | QualityOfReading | Recommended extension |
| ReadingType.direction | ReadingDirection | ReadingType.flowDirection | FlowDirectionType | Same |
| ReadingType.intervalLength | Duration | ReadingType.intervalLength | UInt32 | Recommended extension |
| ReadingType.kind | ReadingKind | ReadingType.kind | KindType | measurementKind |
| ReadingType.multiplier | UnitMultiplier | ReadingType.powerOfTenMultiplier | PowerOfTenMultiplierType | Recommended extension |
| ReadingType.unit | UnitSymbol | ReadingType.uom | UomType | unit |
|  |  | ReadingType.accumulationBehaviour | AccumulationBehaviourType | accumulation |
|  |  | ReadingType.dataQualifier | DataQualifierType | Recommended extension |
|  |  | ReadingType.tou | TOUType | Same |
|  |  | ReadingType.currency | CurrencyCode | Same |
|  |  | ReadingType.commodity | CommodityType | Same |
|  |  | ReadingType.consumptionTier | ConsumptionTierType | Same |
|  |  | ReadingType.phase | PhaseCode | phases |
|  |  | IntervalBlock.mRID | HexBinary128 | Recommended extension |
|  |  | IntervalBlock.description | String32 | Recommended extension |
|  |  | IntervalBlock.interval | DateTimeInterval | Recommended extension |
| Reading.cost | Float | Reading.cost | UInt48 | Recommended extension |
| Reading.timeStamp | AbsoluteDateTime | Reading.timePeriod | DateTimeInterval | Same |
| Reading.value | Float | Reading.value | UInt48 | Same |
| ReadingQuality.quality | QualityOfReading | ReadingQuality.quality | QualityOfReading | Recommended extension |
| DateTimeInterval.start | AbsoluteDateTime | DateTimeInterval.start | TimeType | Same |
| DateTimeInterval.duration | Duration | DateTimeInterval.duration | UInt32 | Uses "end" instead of "duration" |
| QualityOfReading.estimated |  | QualityOfReading \*.8.0 |  | (encoded) |
| QualityOfReading.raw |  | QualityOfReading 1.\* |  | (encoded) |
| QualityOfReading.validated |  | QualityOfReading \*.0.1 |  | (encoded) |
| ReadingDirection.forward |  | FlowDirectionType 1 |  | (encoded) |
| ReadingDirection.reverse |  | FlowDirectionType 19 |  | (encoded) |
| ReadingKind.energy |  | FlowDirectionType 12 |  | (encoded) |
| ReadingKind.power |  | FlowDirectionType 37 |  | (encoded) |
| ReadingKind.demand |  | FlowDirectionType 8 |  | (encoded) |
| UnitMultiplier.micro |  | PowerOfTenMultiplierType -6 |  | (encoded) |
| UnitMultiplier.m |  | PowerOfTenMultiplierType -3 |  | (encoded) |
| UnitMultiplier.c |  | PowerOfTenMultiplierType -2 |  | (encoded) |
| UnitMultiplier.d |  | PowerOfTenMultiplierType -1 |  | (encoded) |
| UnitMultiplier.k |  | PowerOfTenMultiplierType 3 |  | (encoded) |
| UnitMultiplier.M |  | PowerOfTenMultiplierType 6 |  | (encoded) |
| UnitMultiplier.G |  | PowerOfTenMultiplierType 9 |  | (encoded) |
| UnitMultiplier.T |  | PowerOfTenMultiplierType 12 |  | (encoded) |
| UnitMultiplier.none |  | PowerOfTenMultiplierType 0 |  | (encoded) |

**4. SUPPORTING DOCUMENTATION**

**a. Description of Request:**

**b. Description of Recommendation:**

**c. Business Purpose:**

**d. Commentary/Rationale of Subcommittee(s)/Task Force(s):**

1. NIST Special Publication 800-122, Guide to Protecting the Confidentiality of Personally Identifiable Information (PII) April 2010, page 2-1. [↑](#footnote-ref-1)