Minimal Management Interface Version 1.0

Committee Specification Draft 02 / Public Review Draft 01

13 October 2016

Specification URIs

This version:

http://docs.oasis-open.org/coel/MMI/v1.0/csprd01/MMI-v1.0-csprd01.docx (Authoritative)
http://docs.oasis-open.org/coel/MMI/v1.0/csprd01/MMI-v1.0-csprd01.html
http://docs.oasis-open.org/coel/MMI/v1.0/csprd01/MMI-v1.0-csprd01.pdf

Previous version:

http://docs.oasis-open.org/coel/MMI/v1.0/csd01/MMI-v1.0-csd01.docx (Authoritative)
http://docs.oasis-open.org/coel/MMI/v1.0/csd01/MMI-v1.0-csd01.html
http://docs.oasis-open.org/coel/MMI/v1.0/csd01/MMI-v1.0-csd01.pdf

Latest version:

http://docs.oasis-open.org/coel/MMI/v1.0/MMI-v1.0.docx (Authoritative)
http://docs.oasis-open.org/coel/MMI/v1.0/MMI-v1.0.html
http://docs.oasis-open.org/coel/MMI/v1.0/MMI-v1.0.pdf

Technical Committee:

OASIS Classification of Everyday Living (COEL) TC

Chairs:

David Snelling (David.Snelling@UK.Fujitsu.com), Fujitsu Limited
Joss Langford (joss@activinsights.co.uk), Activinsights Ltd

Editor:

David Snelling (David.Snelling@UK.Fujitsu.com), Fujitsu Limited

Related work:

This specification is related to:

- *Classification of Everyday Living Version 1.0.* Edited by Joss Langford. Latest version: http://docs.oasis-open.org/coel/COEL/v1.0/COEL-v1.0.html.
- *Roles, Principles, and Ecosystem Version 1.0*. Edited by Matthew Reed. Latest version: http://docs.oasis-open.org/coel/RPE/v1.0/RPE-v1.0.html.
- *Behavioural Atom Protocol Version 1.0*. Edited by Joss Langford. Latest version: http://docs.oasis-open.org/coel/BAP/v1.0/BAP-v1.0.html.
- *Identity Authority Interface Version 1.0*. Edited by Paul Bruton. Latest version: http://docs.oasis-open.org/coel/IDA/v1.0/IDA-v1.0.html.
- *Public Query Interface Version 1.0.* Edited by David Snelling. Latest version: http://docs.oasis-open.org/coel/PQI/v1.0/PQI-v1.0.html.

Abstract:

This document defines a minimal interface between the Data Engine and other actors in the ecosystem, namely the Service Provider and the Operator. The interface provides for registering and managing Operators, Devices, and Consumers within a Data Engine. This interface represents the minimal requirements of a Data Engine's management interface, but does not limit this interface to these capabilities.

Status:

This document was last revised or approved by the OASIS Classification of Everyday Living (COEL) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=coel#technical.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment button on the TC's web page at https://www.oasis-open.org/committees/coel/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/coel/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[COEL-MMI-v1.0]

Minimal Management Interface Version 1.0. Edited by David Snelling. 13 October 2016. OASIS Committee Specification Draft 02 / Public Review Draft 01. http://docs.oasis-open.org/coel/MMI/v1.0/csprd01/MMI-v1.0-csprd01.html. Latest version: http://docs.oasis-open.org/coel/MMI/v1.0/MMI-v1.0.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS 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.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS 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. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. 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 OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS 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.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

1 Introduction

This document defines the Minimal Management Interface (MMI) between the Data Engine and other actors in the ecosystem. It provides operation definitions on the Data Engine for use by a Service Provider to register a new Operator, to retrieve a list of existing Operators, to retrieve a list of Consumers associated with a given Operator, to suspend and resume Operators, register and unassign Devices and to assure a consumer is registered. It also provides operations definitions on the Data Engine for use by an Operator to register a Consumer, forget a Consumer and to associate a device with a consumer.

This interface represents the minimal requirements of a Data Engine's management interface, but does not limit this interface to these capabilities. High quality Data Engines may offer more comprehensive management services.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **[RFC2119]**.

1.2 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

[COEL_RPE-1.0] *Roles, Principles, and Ecosystem Version 1.0.* Latest version: http://docs.oasis-open.org/coel/RPE/v1.0/RPE-v1.0.docx.

[COEL_IDA-1.0] *Identity Authority Interface Version 1.0.* Latest version: http://docs.oasis-open.org/coel/IDA/v1.0/IDA-v1.0.docx

[ISO/IEC 5218] Codes for the representation of human sexes, December 2004. http://www.iso.org/iso/catalogue_detail.htm?csnumber=36266

1.3 Non-Normative References

[Coelition] http://www.coelition.org

[Data to Life] Reed, M. & Langford, J. (2013). *Data to Life*. Coelition, London. ISBN 978-0957609402

2 Interface Specification

The Minimal Management Interface on the Data Engine is divided into sections depending on which actor and function in a COEL ecosystem is communicating with the Data Engine. The following sub-sections define these interfaces.

2.1 Authentication and Authorisation

To access all Service Provider functions of the Data Engine MMI API, Service Providers need access credentials with two components:

- A userid to identify the caller.
- A password for authentication.

HTTP basic authentication SHALL be used to authenticate calls to the API. Passwords SHOULD be 64 bytes in length and MUST be supplied as an ASCII string. This MUST be prefixed with the userid followed by a colon to form the token passed in the HTTP Authorisation Header.

Note that while Operators need to secure their connection to the Data Engine with TLS, they do not need to Authenticate or Authorise.

Example:

```
"9abf5386-2ac6-4e61-abc4-6b809a85d6cb:J1dOeWJJOkd3akhnSn4ma007M
DtUMVAxISgyOn9jI2U9NHNdRi4hfiw9c2I8PURcVltNMWQkamsrfGR4T24vKA=="
```

If the userid is unrecognized, or the wrong password is supplied a HTTP status code *401 Invalid username or password* SHALL be returned.

Note: All Operator functions do not require authentication or authorisation.

2.2 Service Provider: Create New Operator

Create a new Operator within the Data Engine and associate it with the requesting Service Provider. Completion of this operation allows the Operator to register new Consumers.

API	Description
POST service-provider/operator	Create an Operator identity within the Data Engine permitting that operator to create and register Consumers.

2.2.1 Request

Parameter Name	Description	Type
OperatorID	A Pseudonymous Key generated by an IDA and associated with the Operator being registered.	**String**: Format defined in [COEL_IDA-1.0].
TimeStamp	Time stamp of the OperatorID indicating when the IDA created this Pseudonymous Key.	**DateTimeString:** Format defined in [COEL_IDA-1.0].
Signature	Signature proving that an IDA created this OperatorID.	**String**: Format defined in [COEL_IDA-1.0].

Media type:

`application/json, text/json`

Sample:


```
{"OperatorID": "00000000-0000-0000-0000-000000000000",
 "TimeStamp": "2011-02-14T00:00:00",
 "Signature":
      "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="}
```


2.2.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned. If unsuccessful, an HTTP error code SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure.

If validation of the OperatorID fails, with a 410 (Gone) error from the IDA, an error 410 (Gone) should be returned.

Parameter Name	Description	Type
Reason	An optional description of why the registration failed.	**String:**

Media type:

`application/json, text/json`

Sample:

```
{"Reason":"Operator was not valid."}
```

2.3 Service Provider: Retrieve Operator List

A Service Provider uses this operation to retrieve a list of all registered Operators registered to the requesting Service Provider.

API	Description
GET service-provider/operators	Retrieve a list of all Operators associated with the requesting Service Provider.

2.3.1 Request

The request is empty.

2.3.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned along with an array of Pseudonymous Keys each associated with an Operator associated with the requesting Service Provider. If unsuccessful, an HTTP error code SHOULD be returned, in which case a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2.

Parameter Name	Description	Type
OperatorIDs	An array of Pseudonymous Keys one for each of the Operators associated with the requesting Service Provider.	**Array of String**: Format defined in [COEL_IDA-1.0].
Reason	An optional description of why the operation failed.	**String:**

Media type:

```
application/json, text/json
```

Sample:

```
{"OperatorIDs": [
      "00000000-0000-0000-0000-000000000000",
      "00000000-0000-0000-0000-000000000001",
      "00000000-0000-0000-0000-000000000002"]
}
```

2.4 Service Provider: Retrieve Consumer List

A Service Provider uses this operation to retrieve a list of all Consumers registered to a given Operator, which is in turn registered to the requesting Service Provider.

API	Description
POST service-provider/consumers	Retrieve a list of all Consumers associated with a given Operator, which is in turn associated with the requesting Service Provider.

2.4.1 Request

Parameter Name	Description	Type
OperatorID	A Pseudonymous Key generated by an IDA and associated with an Operator registered with the Data Engine.	**String**: Format defined in [COEL_IDA-1.0].

Media type:

```
application/json, text/json
```

Sample:

```
{"OperatorID": "00000000-0000-0000-0000-000000000000"}
```

2.4.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned along with an array of Pseudonymous Keys each associated with a Consumer registered with the given Operator which is in turn associated with the requesting Service Provider. If unsuccessful, an HTTP error code SHOULD be returned, in which case a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2.

Parameter Name	Description	Type
ConsumerIDs	An array of Pseudonymous Keys one for each of the Consumers associated with given Operator.	**Array of String**: Format defined in [COEL_IDA-1.0].

Parameter Name	Description	Type
Reason	An optional description of why the operation failed.	**String:**

Media type:

```
application/json, text/json
```

Sample:

```
{"ConsumerIDs": [
      "00000000-0000-0000-0000-000000000000",
      "00000000-0000-0000-0000-000000000001",
      "00000000-0000-0000-0000-000000000002"]
}
```

2.5 Service Provider: Suspend Operator

Suspend the given Operator's ability to create new Consumers and assign devices. This operation has no effect on data stored for existing Consumers. The Operator will still be permitted to execute a Forget Consumer operation.

API	Description
POST service-provider/suspendOperator	Suspend the given Operator's ability to register new Consumers and assign devices.

2.5.1 Request

Parameter Name	Description	Type
OperatorID	A Pseudonymous Key generated by an IDA and associated with the Operator to be suspended.	**String**: Format defined in [COEL_IDA-1.0].

Media type:

```
application/json, text/json
```

Sample:

```
{"OperatorID": "00000000-0000-0000-0000-000000000000"}
```

2.5.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned. If unsuccessful, an HTTP error code SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure.

Parameter Name	Description	Type
Reason	An optional description of why the Operator suspension failed.	**String:**

Media type:

```
application/json, text/json
```

Sample:

```
{"Reason":"Operator does not exist."}
```

2.6 Service Provider: Resume Operator

Resume the given Operator's ability to create new Consumers and assign devices.

API	Description
POST service-provider/resumeOperator	Resume the given Operator's ability to register new Consumers and assign devices.

2.6.1 Request

Parameter Name	Description	Type
OperatorID	A Pseudonymous Key generated by an IDA and associated with the Operator to be resumed.	**String**: Format defined in [COEL_IDA-1.0].

Media type:

```
application/json, text/json
```

Sample:

```
{"OperatorID": "00000000-0000-0000-0000-000000000000"}
```

2.6.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned. If unsuccessful, an HTTP error code SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure.

Parameter Name	Description	Type
Reason	An optional description of why the Operator resumption failed.	**String:**

Media type:

```
application/json, text/json
```

Sample:

```
{"Reason":"Operator does not exist."}
```

2.7 Service Provider: Register Devices

All devices associated with a Service Provider are registered in advance of being assigned to a Consumer (see Section 2.12). Register Devices associates one or more Devices with Service Provider, assigns it a device type (Personal or IoT), and validates the Pseudonymous Keys of the device. A Device SHALL be registered only once. Only Operators associated with the Registering Service Provider MAY Assign the Device to a Consumer.

API	Description
POST service-provider/registerDevices	Register one or more devices with the Data Engine and associate the Devices' Pseudonymous Keys and device types with the calling Service Provider.

2.7.1 Request

The request body is a JSON array containing the following JSON elements.

Parameter Name	Description	Type
DeviceIDs	An array of Pseudonymous Keys associated with the Devices and generated by an IDA.	**Array of String**: Format defined in [COEL_IDA-1.0].
TimeStamp	Time stamp indicating when the IDA created these Pseudonymous Keys.	**DateTimeString:** Format defined in [COEL_IDA-1.0].
Signature	Signature proving that an IDA created these Pseudonymous Keys.	**String**: Format defined in [COEL_IDA-1.0].
DeviceType	A string indicating that the devices are personal devices that MAY be assigned to exactly one Consumer each or IoT devices that MAY be assigned to multiple Consumers.	**String**: Either “Personal” or “IoT”.

Media type:

```
application/json, text/json
```

Sample:


```
{“DeviceIDs”: [“00000000-0000-0000-0000-000000000001”,
               “00000000-0000-0000-0000-000000000002”,
               “00000000-0000-0000-0000-000000000003”],
 “TimeStamp”: “2011-02-14T00:00:00”,
 “Signature”:
      “AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=”,
“DeviceType”: “Personal”
}
```


2.7.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned. If unsuccessful, an HTTP error code SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2.

If validation of the OperatorID fails, with a 410 (Gone) error from the IDA, an error 410 (Gone) should be returned.

Parameter Name	Description	Type
Reason	An optional description of why the operation failed.	**String:**

Media type:

```
application/json, text/json
```

Sample:

```
{"Reason": "DeviceIDs failed to validate with the IDA."}
```

2.8 Service Provider: Unassign Device

Remove all the assignments of the Device from Consumers to which it has been assigned. Note: for IoT devices all assigned Consumers will be unassigned and the Operator might need to reassign some Consumers if for example the Operator wished to remove only one Consumer.

API	Description
DELETE service-provider/unassignDevice	Remove the assignment of the device identified by a Pseudonymous Key from all Consumers associated with the Device.

2.8.1 Request

Parameter Name	Description	Type
DeviceID	A Pseudonymous Key associated with the Device and generated by an IDA.	**String**: Format defined in [COEL_IDA-1.0].

Media type:

```
application/json, text/json
```

Sample:

```
{"DeviceID": "00000000-0000-0000-0000-000000000000"}
```

2.8.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned. If unsuccessful, an HTTP error code SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2.

Parameter Name	Description	Type
Reason	An optional description of why the operation failed.	**String:**

Media type:

```
application/json, text/json
```

Sample:

```
{"Reason": "Device not registered by this Service Provider."}
```

2.9 Service Provider: Assure

This operation provides assurance that a given Consumer is associated to a given Operator and that both are associated with the requesting Service Provider.

API	Description
POST service-provider/assure	Assure that the given Consumer and Operator are associated with each other and with the requesting Service Provider.

2.9.1 Request

Parameter Name	Description	Type
ConsumerID	A Pseudonymous Key associated with the Consumer and generated by an IDA.	**String**: Format defined in [COEL_IDA-1.0].
OperatorID	A Pseudonymous Key generated by an IDA and associated with the Operator.	**String**: Format defined in [COEL_IDA-1.0].

Media type:

application/json, text/json

Sample:

```
{
   "ConsumerID": "00000000-0000-0000-0000-000000000001",
   "OperatorID": "00000000-0000-0000-0000-000000000002"
}
```

2.9.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned along with a JSON object indicating if assurance was achieved or not. If unsuccessful, an HTTP error code SHOULD be returned, in which case a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2.

Parameter Name	Description	Type
Assured	A Boolean value that is true if the given Consumer and Operator are associated with each other and with the requesting Service Provider and false otherwise.	**Boolean**:

Media type:

application/json, text/json

Sample:

```
{"Assured": true }
```

2.10 Operator: Forget Consumer

Request that a Consumer associated with this Operator be forgotten by the Data Engine. This operation MAY not proceed synchronously, as the Data Engine MUST confirm the request with the Service Provider associated with the requesting Operator. The mechanism for confirmation is out of scope of this specification, e.g. email confirmation. The Data Engine MAY either delete all data associated with the Consumer or render that data non-personal.

The Data Engine SHOULD keep a record of which consumers have been forgotten (for audit purposes).

API	Description
POST operator/forgetConsumer	Delete or render non-personal all data associated with the given Consumer.

2.10.1 Request

Parameter Name	Description	Type
ConsumerID	A Pseudonymous Key associated with the Consumer and generated by an IDA.	**String**: Format defined in [COEL_IDA-1.0].

Media type:

application/json, text/json

Sample:

```
{"ConsumerID": "00000000-0000-0000-0000-000000000000"}
```

2.10.2 Response

If successful, an HTTP status code of 201 Accepted MUST be returned. If unsuccessful, an HTTP error code SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2.

Parameter Name	Description	Type
Reason	An optional description of why the operation failed.	**String:**

Media type:

application/json, text/json

Sample:

```
{"Reason": "Internal error."}
```

2.11 Operator: Create New Consumer

Create a new Consumer within the Data Engine and associate it with the given Operator. Completion of this operation allows Behavioural Atoms to be posted anonymously to the Data Engine and be associated with the given Consumer. This function does not require authentication or Authorization. This operation is not permitted when an operator is suspended.

API	Description
POST operator/consumer	Create a Consumer identity within the Data Engine associated with the given Operator.

2.11.1 Request

Parameter Name	Description	Type
OperatorID	A Pseudonymous Key associated with the Operator and generated by an IDA.	**String**: Format defined in [COEL_IDA-1.0].
ConsumerID	A Pseudonymous Key associated with the Consumer and generated by an IDA.	**String**: Format defined in [COEL_IDA-1.0].
TimeStamp	Time stamp of the ConsumerID indicating when the IDA created this Pseudonymous Key.	**DateTimeString:** Format defined in [COEL_IDA-1.0].
Signature	Signature proving that an IDA created this ConsumerID.	**String**: Format defined in [COEL_IDA-1.0].
SegmentData	An OPTIONAL object containing (OPTIONALLY) residential time zone and latitude, gender, and year of birth.	**Object:** Composed of ResidentTimeZone, ResidentLatitude, Gender, and YearOfBirth.
ResidentTimeZone	The time zone in which the Consumer generally resides.	**TimeZoneString:** As +/-hh:mm from UTC.
ResidentLatitude	The latitude (rounded to an integer) at which the Consumer generally resides.	**Integer:** Representing latitude rounded to an integer.
Gender	The gender of the Consumer.	**Integer 0-99:** 0 not known

Parameter Name	Description	Type
		1 male 2 female 9 not applicable
YearOfBirth	Year in which the Consumer was born.	**Integer:** Representing year of birth.

The Gender parameter SHALL have enumerated fields reserved for compliance with **[ISO/IEC 5218]**.

Media type:

`application/json, text/json`

Sample:


```
{"OperatorID": "00000000-0000-0000-0000-000000000000",
 "ConsumerID": "00000000-0000-0000-0000-000000000000",
 "TimeStamp": "2011-02-14T00:00:00",
 "Signature":
      "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=",
  "SegmentData":
      {"ResidentTimeZone": "+03:00",
       "ResidentLatitude": 51,
       "Gender": 2,
       "YearOfBirth": 1993
      }
}
```


2.11.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned. If unsuccessful due to errors in the request content, an HTTP error code 400 (Bad Request) SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2.

If validation of the Consumer ID fails, with a 410 (Gone) error from the IDA, an error 410 (Gone) should be returned. A JSON object MAY be returned providing some explanation of the failure, see section 2.2.2

Parameter Name	Description	Type
Reason	An optional description of why the operation failed.	**String:**

Media type:

```
application/json, text/json
```

Sample:

```
{"Reason": "Invalid Latitude: must be in range -90..+90 ."}
```

2.12 Operator: Assign a Device to a Consumer

Assign a Pseudonymous Key representing a device to a Consumer associated with the requesting Operator. All Atoms posted with this Pseudonymous Key will be associated with the corresponding Consumer. Once assigned to a Consumer, a Personal Device MUST not be reassigned to another Consumer, without first being Unassigned from all Consumers (see Section 2.8). An Operator MAY assign an IoT Device to multiple Consumers. This function does not require authentication or authorization. This operation is not permitted when an operator is suspended. The Device, the Operator, and the Consumer MUST already be registered with the Data Engine and associated with the same Service Provider.

API	Description
POST operator/device	Associate a device, identified by a Pseudonymous Key, to a registered Consumer associated with the requesting Operator.

2.12.1 Request

Parameter Name	Description	Type
DeviceID	A Pseudonymous Key associated with the Device and generated by an IDA.	**String**: Format defined in [COEL_IDA-1.0].
OperatorID	A Pseudonymous Key of the Operator to which the Consumer is associated.	**String**: Format defined in [COEL_IDA-1.0].
ConsumerID	A Pseudonymous Key of the user to which the device is to be associated. The user MUST already be associated with the requesting Operator.	**String**: Format defined in [COEL_IDA-1.0].

Media type:

application/json, text/json

Sample:

```
{“DeviceID”: “00000000-0000-0000-0000-000000000000”,
 “OperatorID”: “00000000-0000-0000-0000-000000000001”,
 “ConsumerID”: “00000000-0000-0000-0000-000000000002”
}
```

2.12.2 Response

If successful, an HTTP status code of 200 *OK* MUST be returned. If unsuccessful, an HTTP error code SHOULD be returned and a JSON object MAY be returned providing some explanation of the failure, see section 2.2.2.

Parameter Name	Description	Type
Reason	An optional description of why the operation failed.	**String:**

Media type:

application/json, text/json

Sample:

```
{“Reason”: “DeviceID is already associated with a consumer.”}
```

3 Conformance

An implementation is a conforming Minimal Management Interface if the implementation meets the conditions set out in in Section 2 of this document AND the conformance criteria in **[COEL_RPE-1.0]**

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Paul Bruton, Individual Member
Joss Langford, Activinsights
Matthew Reed, Coelition
David Snelling, Fujitsu

Appendix B. Revision History

Revision	Date	Editor	Changes Made
1	21/08/2015	David Snelling	A few minor changes to test the revision process in Kavi.
2	21/09/2015	David Snelling	First complete version, based on submitted material.
3	25/09/2015	Paul Bruton	Added review comments
4	25/09/2015	Joss Langford	Review, spell correction and change of ‘sex’ to ‘gender’ in section 2.4
5	11/10/2015	David Snelling	Edits for issues: COEL-10 (Segment data), COEL-17 (Location of security), COEL-23 (Forget operation)
6	11/10/2015	David Snelling	Removed tracking
7	13/10/2015	Paul Bruton	Conformance includes reference to RPE document.
8	19/10/2015	David Snelling	COEL-13 and a few style and consistence issues.
9	23/10/2015	David Snelling	Adding OperatorID to New Consumer request.
10	30/10/2015	David Snelling	Removed text allowing reassignment of Devices by Operator.
11	31/102015	Joss Langford	Accept all changes, track changes off, check references and style consistency.
12	02/11/2015	David Snelling	Final date change
13	03/11/2015	Paul Bruton	Corrected authorization and authentication description; Spelling correction; Corrected description of TimeStamp and Signature parameters in operator/device, also added OperatorID parameter since there will be no authorization header in this request.
14	03/11/2015	Paul Bruton	Minor spelling correction.
15	25/11/2015	David Snelling	Fixed 45, 47, & 52.
16	25/11/2015	David Snelling	Fixed Revision History.
17	25/11/2015	Joss Langford	Changes accepted and track changes switched off.
18	25/11/2015	David Snelling	Set date for final publication.
19	07/01/2016	David Snelling	Update to WD02 and changed error code management in line with issue COEL-42.

20	14/01/2016	Paul Bruton	Made “Reason” codes in response body explicit and added comment in 2.6.2 about how to identify an invalid Consumer ID.
21	21/01/2016	David Snelling	Edits to change operator/forget to operator/forgetConsumer. Edits to pass on error 410 Gone from the IDA to the Operator. Text for call back mechanism for operator/forgetConsumer added.
22	12/02/2016	Paul Bruton	Accepted previous edits. Spelling correction in ‘forgetConsumer’ text; Change to 201 Accepted in forgetConsumer response
23	02/03/2016	Paul Bruton	COEL-58 Added Service Provider: Unassign Device and sequence diagram to illustrate usage
24	08/03/2016	Paul Bruton	COEL-58 Following discussion, proposing to implement this as a DELETE operation.
25	21/03/2016	Dave Snelling	Corrected spelling, updated ToC, and accepted changes for versions 22-24.
26	16/06/2016	Dave Snelling	COEL-15: Added suspension and resumption of operators. Moved Unassign Device next to other service provider operations.
27	17/06/2016	Dave Snelling	Typos and removed change tracking.
28	21/08/2016	Joss Langford	Gender field of segment data updated (COEL-74).
29	24/08/2016	David Snelling	Device assignment and unassignment and shared devices added. COEL-61.
30	24/08/2016	David Snelling	Added operation to assure the association between Consumer and Operator. COEL-66
31	26/08/2016	David Snelling	Fixed quotes in gender and batched DeviceIDs.
32	02/09/2016	David Snelling	Fixed number problems in the Register Devices Operation.
33	16/09/2016	Joss Langford	Reference correction COEL-81
34	23/09/2016	Paul Bruton	Reference to 410 (gone) added to sections that require IDA validation calls. Preamble updated. Question about parameters in Forget Consumer (COEL-82)
35	23/09/2016	Paul Bruton	Removed comment after resolution of COEL-82
36	10/10/2016	Joss Langford	Revision numbers corrected & changes accepted.