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

Previous version:

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

Latest version:

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

Paul Bruton (Paul.Bruton@tessella.com), Tessella Ltd.

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.
- *Minimal Management Interface Version 1.0*. Edited by David Snelling. Latest version: http://docs.oasis-open.org/coel/MMI/v1.0/MMI-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 the interface protocol for an Identity Authority (IDA). The IDA is a central web-based service, needed in any ecosystem that conforms to the ecosystem architecture, that statelessly provides unique, signed Pseudonymous Keys. These Pseudonymous Keys are used to register actors within the ecosystem and are then requested by an actor wishing to register an

individual person within the ecosystem (and thus enter into data exchanges about that person's Behavioural Atoms).

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-IDA-v1.0]

Identity Authority Interface Version 1.0. Edited by Paul Bruton. 13 October 2016. OASIS Committee Specification Draft 02 / Public Review Draft 01. http://docs.oasis-open.org/coel/IDA/v1.0/csprd01/IDA-v1.0-csprd01.html. Latest version: http://docs.oasis-open.org/coel/IDA/v1.0/IDA-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 5
1.1 Terminology 5
1.2 Normative References 5
1.3 Non-Normative References 5
2 The Identity Authority 6
3 Protocol Overview 7
4 API Overview 9
4.1 Introduction 9
4.2 Authentication and Authorisation 9
4.3 Identity Authority Information Request 9
4.4 PseudonymousKey endpoint 10
4.4.1 Response 10
4.5 PseudonymousKeyBatch endpoint 11
4.5.1 Request 12
4.5.2 Response 12
4.6 Validation endpoint 13
4.6.1 Request 13
4.6.2 Response 14
5 Conformance 15
Appendix A. Acknowledgments 16
Appendix B. Revision History 17

1 Introduction

[All text is normative unless otherwise labelled]

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.

[RFC4122] Leach, P., Mealling, M., Salz, R., “A Universally Unique Identifier (UUID) URN Namespace”, RFC 4122, July 2005. http://tools.ietf.org/html/rfc4122

[RFC3339] Klyne, G., Newman, C., “Date and Time on the Internet: Timestamps”, RFC 3339, July 2002. http://www.ietf.org/rfc/rfc3339.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

1.3 Non-Normative References

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

2 The Identity Authority

An Identity Authority (IDA) provides an Identity Authority Interface (the API) that generates and subsequently validates a digitally signed unique Pseudonymous Key to be used in signup to Data Engine services. The IDA does not require any input to generate the Pseudonymous Key.

Section 3 of this document describes how the API is used by Operators, Service Providers, and Data Engines to register Consumers or Devices. Section 4 gives details on the API itself.

The terms Pseudonymous Key, Data Engine, Operator, Hardware Developer and Service Provider are as defined in **[COEL-RPE-1.0]**.

3 Protocol Overview

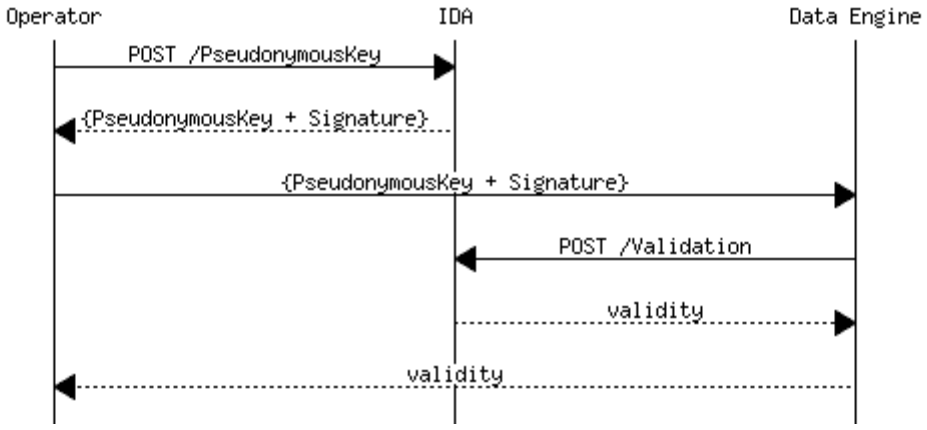

Figure 1: IDA/Data Engine signup sequence

Figure 1 shows the sequence an Operator MUST follow in signing up a new consumer: obtain a Pseudonymous Key from IDA and then use it to signup with the Data Engine. The Pseudonymous Key is used in all subsequent communications with the Data Engine such as sending data, requesting reports. For each new consumer, the Operator and Data Engine use a separate Pseudonymous Key. Applications that generate input (Behavioural Atoms) for the Data Engine also use the Pseudonymous Key.

The signature is used so that the Data Engine can be assured that the Pseudonymous Key is genuine. Rather than using asymmetric key-pairs and distributing a public key and signing algorithm, the IDA provides the means for a receiver of a signed Pseudonymous Key to validate its signature. The Data Engine MUST use this validation mechanism.

It is assumed that this transaction is short – Operators only request Pseudonymous Keys when they need them and register them shortly afterwards (probably within minutes). The Identity Authority needs to be free to alter the means of signature (if for example it believes the mechanism used internally has been revealed). If this change happens during a transaction then validation will fail. This is an unlikely event, but parties in the transaction need to be able to manage it:

- Data Engines receiving a failed validation code MUST pass the failure back to the Operator.
- Operators receiving a failed validation code from the Data Engine MUST discard the Pseudonymous Key and request a new one from the IDA.
- If the second attempt also fails, the Operator SHOULD try once more after a short delay (1-2 seconds) before aborting the attempt to register.

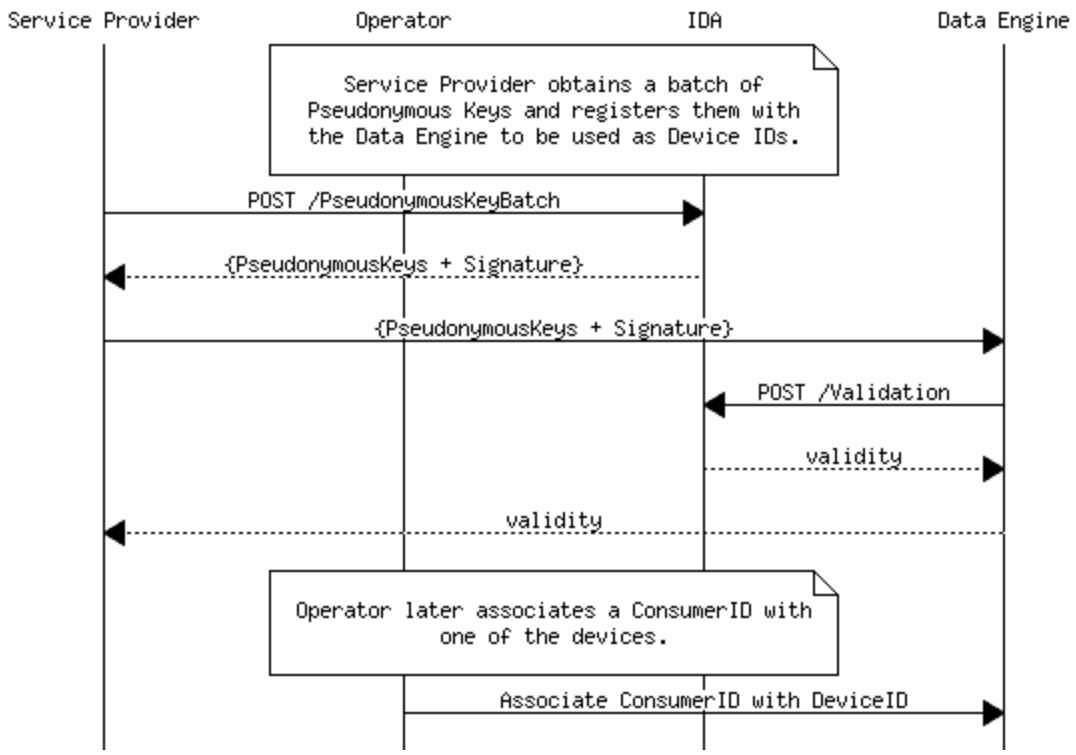

Figure 2: Service Provider registering a batch of DeviceIDs

The IDA also provides a means to generate a batch of up to 1000 Pseudonymous Keys in one request. The batch contains a single signature and the same protocol MUST be followed for validation: The Service Provider passes the batch to the Data Engine which MUST validate the batch with the IDA. It is expected that the Service Provider will then provide this batch of IDs to a Hardware Developer to be used in devices.

Pseudonymous Keys are primarily intended to represent a consumer in the ecosystem. However, Operators and Service Providers also require keys to identify themselves in their machine to machine interactions. IDA generated Pseudonymous Keys MAY be used for this purpose since they are devoid of DIPI and unique across the ecosystem.

4 API Overview

4.1 Introduction

The Identity Authority (IDA) API provides a means for Operators and/or Service Providers to generate unique Pseudonymous Keys for Consumers or Devices. A Pseudonymous Key is REQUIRED when an Operator or Service Provider registers a consumer or device with a Data Engine. Pseudonymous Keys are digitally signed so that Data Engines can validate them to ensure they were generated by the IDA and have not been altered.

4.2 Authentication and Authorisation

To access the IDA API, callers need API Credentials with two components:

- A userid to identify the caller.
- A password to authenticate the caller.

A userid MUST be assigned to one of the following two roles in the IDA:

- *Generator:* Allowing the userid to generate Pseudonymous Keys
- *Validator*: Allowing the userid to validate Pseudonymous Keys

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.

Example:

```
"9abf5386-2ac6-4e61-abc4-
6b809a85d6cb:JhmiDAlnpo1SBrlrN6H09RqQoerdLCyepbXgE7005OSzXzMeUsGCEXaVNA
MrKv8D"
```

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

If a request is made with a userid that is assigned a role that is not authorized to perform that action then the HTTP status code *403 Unauthorised* SHALL be returned.

4.3 Identity Authority Information Request

Every Identity Authority SHALL publish its Home URI. Performing a GET on this URI SHALL return general information about the Identity Authority as JSON object.

Method	**Request**	**Response Status**	**Response Content-Type**	**Response Body**
GET	None	200 (OK)	application/json	JSON object
POST	Any	405 (Method Not Allowed)	None	None

Format for the returned JSON Object:

Name	**Value**	**Description**	**REQUIRED**
IdentityAuthorityURI	String	The URI of the Identity Authority service encoded as a string.	Yes
ServerTime	Integer	Current server time in UTC as a Unix timestamp.	Yes

IdentityAuthorityStatus	String	The current status of the Identity Authority service encoded as a string. It MUST be one of “Up”, “Down”, or “Unknown”.	Yes

The JSON object of the response MAY contain additional fields with information about the Identity Authority.

Example request message:

```
GET /home
```

Example response message:

```
HTTP/1.1 200 OK

{“IdenityAuthorityURI”: “https://www.ida.com/api”,
 “IdentityAuthorityStatus”: “Up”,
 “ServerTime”: 1470822001}
```

4.4 PseudonymousKey endpoint

The IDA SHALL provide a PseudonymousKey end-point which provides the means to generate Pseudonymous Keys for users whose API Credentials have the Generator role.

API	Description
POST /PseudonymousKey	Generate a new signed Pseudonymous Key for an actor. The mechanism used to sign the response is periodically changed, so the response SHOULD be passed to the Data Engine shortly after generation or validation can fail.

4.4.1 Response

The response SHALL contain three parameters: The Pseudonymous Key; the timestamp at which the response was generated; and a signature that can be used for validation.

Parameter Name	Description	Type
PseudonymousKey	Unique key to be used to represent a consumer, Service Provider, Operator or device in the ecosystem.	**String**: Formatted as a UUID as defined in [RFC_4122, Section 3]
TimeStamp	Date and time at which the PseudonymousKey was generated.	**String:** Formatted as a date-time according to [RFC_3339].
Signature	ASCII encoded signature which the IDA will use for validation.	**String**

Media type:

application/json, text/json

Sample:

```
{
  "PseudonymousKey": "00000000-0000-0000-0000-000000000000",
  "TimeStamp": "2011-02-14T00:00:00",
  "Signature": "SGFDXCTVIVVIFUJUVUYBKYKJHBK=="
}
```

Status:

200: The operation was successful

401/403: The operation failed due to authentication or authorization failure. The caller should confirm its credentials and retry.

500: Internal error, the caller should retry

4.5 PseudonymousKeyBatch endpoint

The IDA SHALL provide a PseudonymousKeyBatch end-point which provides the means to generate a batch of Pseudonymous Keys in one response packet for users whose API Credentials have the Generator role.

API	Description
POST /PseudonymousKeyBatch	Generate a new signed batch of Pseudonymous Keys. The mechanism used to sign the response is periodically

API	Description
	changed, so the response SHOULD be passed to the Data Engine shortly after generation or validation can fail.

4.5.1 Request

The request body SHALL contain one parameter: Size.

Parameter Name	Description	Type
Size	The number of PseudonymousKeys to return in the batch.	**Integer:** 1 <= n <= 1000

Media type:

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

Sample:

```
{"Size": 1000}
```

4.5.2 Response

The response SHALL contain three parameters: An array of Pseudonymous Keys; the timestamp at which the response was generated; and a signature that can be used for validation.

Parameter Name	Description	Type
PseudonymousKeys	Array of unique keys to be used to represent a devices in the ecosystem.	**Array of String**: Each string formatted as a UUID as defined in [RFC_4122, Section 3]
TimeStamp	Date and time at which the PseudonymousKey was generated.	**String:** Formatted as a date-time according to [RFC_3339].
Signature	ASCII encoded signature which the IDA will use for validation.	**String**

Media type:

application/json, text/json

Sample:

```
{
  "PseudonymousKeys": [
      "00000000-0000-0000-0000-000000000000",
      "00000000-0000-0000-0000-000000000001",
      "00000000-0000-0000-0000-000000000002"]
  "TimeStamp": "2011-02-14T00:00:00",
  "Signature": "SGFDXCTVIVVIFUJUVUYBKYKJHBK=="
}
```

Status:

200: The operation was successful

400: The operation failed due to the request body being malformed or the size being out of range [1..1000]

401/403: The operation failed due to authentication or authorization failure. The caller should confirm its credentials and retry.

500: Internal error, the caller should retry

The IDA SHALL be capable of generating a batch of up to 1000 PseudonymousKeys.

4.6 Validation endpoint

The IDA SHALL provide a Validation end-point which provides the means to validate a signed PseudonymousKey or a signed batch of PseudonymousKeys for users whose API Credentials have the Validator role.

API	Description
POST /Validation	Validates a single signed Pseudonymous Key or a signed batch of Pseudonymous Keys to ensure that they were generated by IDA. The mechanism used for signing is periodically changed. If validation fails, the caller MUST request that the Operator requests a new Pseudonymous Key.

4.6.1 Request

The request body format SHALL conform to the specification of EITHER the /PseudonymousKey response packet OR the /PseudonymousKeyBatch response packet.

Media type:

application/json, text/json

Sample 1 (Single input):

```
{
  "PseudonymousKey": "00000000-0000-0000-0000-000000000000",
  "TimeStamp": "2011-02-14T00:00:00",
```

```
  "Signature": "SGFDXCTVIVVIFUJUVUYBKYKJHBK=="
}
```

Sample 2 (Batch input):

```
{
  "PseudonymousKeys": [
      "00000000-0000-0000-0000-000000000000",
      "00000000-0000-0000-0000-000000000001",
      "00000000-0000-0000-0000-000000000002"]
  "TimeStamp": "2011-02-14T00:00:00",
  "Signature": "SGFDXCTVIVVIFUJUVUYBKYKJHBK=="
}
```

4.6.2 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 operation failed.	**String:**

Media type:

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

Sample:

```
{"Reason": "The input was missing mandatory elements"}
```

Status:

200: The operation was successful. The Pseudonymous Key or batch of Pseudonymous Keys is valid.

410: The operation was successful (the request was properly formed and authorized) but the Pseudonymous Key or batch of Pseudonymous Keys is no longer valid.

400: The operation failed due to the request body being malformed.

401/403: The operation failed due to authentication or authorization failure. The caller should confirm its credentials and retry.

500: Internal error, the caller should retry

5 Conformance

An implementation is a conforming Identity Authority Interface if the implementation meets the conditions in Section 4 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	27/07/2015	Paul Bruton	Initial version for OASIS submission
2	15/09/2015	Paul Bruton	Header page, references, acknowledgements, conformance sections finalized. Body text still needs a review so that it clearly uses normative language in section 4, with sections 2 and 3 being descriptive text only.
3	21/09/2015	Paul Bruton	Clarification of terms by replacing ‘access credentials’ with ‘API Credentials’ to signify that the userid/password combination is used only for API access.
4	22/09/2015	Paul Bruton	Action #0016: Removed ‘api/’ prefix from methods in API.
5	23/09/2015	Paul Bruton	Added normative references for UUID and TimeStamp formats. Corrected sequence diagrams following removal of api/. Section 4 altered to use appropriate normative terms. Added definitions of request and response (action #0014). Added references to related works
6	25/09/2015	Joss Langford	Review with minor changes to clarify meanings and correct spelling.
7	05/10/2015	Paul Bruton	Previous changes accepted. Added reference to conformance criteria in RPE and clarified the name of the conformance target: Identity Authority Interface, referred to as the API.
8	19/10/2015	David Snelling	A few tweaks and fixed the COEL link.
9	20/10/2015	Paul Bruton	Accepted changes, Resolved COEL-39 (data engine must validate) and COEL-40 (removed unnecessary text about combining data engine data)
10	21/10/2015	Paul Bruton	Minor changes for consistent style
11	31/10/2015	Joss Langford	Accept all changes, track changes off, check references and style consistency.
12	02/11/2015	David Snelling	Final data change.
13	03/11/2015	Paul Bruton	Spelling correction following review.
14	25/11/2015	David Snelling	Set date for final CD publication.
15	07/01/2016	Paul Bruton	COEL-42 clarification of response codes and updated to WD02

16	20/01/2016	Paul Bruton	COEL-42 implementation of 410 status code in event of validation failure. Also fixed broken hyperlinks and removed hidden hyperlinks.
17	10/08/2016	David Snelling	Added status field to IDA information request, COEL-68.
18	14/08/2016	Joss Langford	Checked and changes accepted.
19	23/09/2016	Paul Bruton	Corrected role of Hardware Developer (COEL-83)
20	27/09/2016	David Snelling	Final review: Rebuild the ToC, corrected spelling of artefact, labelled, added Service Providers to list of IDA users in section 2 & 4.1, and accepted all changes.