Digital ID (AGDIS) Data Standards 2024

I, Katy Gallagher, Minister for Finance, make the following instrument.

Dated

Katy Gallagher **DRAFT ONLY—NOT FOR SIGNATURE**
Minister for Finance

Contents

1 Name

This instrument is the *Digital ID (AGDIS) Data Standards 2024*.

2 Commencement

This instrument commence at the same time as the *Digital ID Act 2024* commences.

3 Authority

This instrument is made under section 99 of the *Digital ID Act 2024*.

4 Definitions

Note 1: A number of expressions used in this instrument are defined in section 9 of the Act, including the following:

(a) accredited attribute service provider;
(b) accredited identity exchange provider;
(c) attribute;
(d) Australian Government Digital ID System;
(e) digital ID;
(f) entity;
(g) participate;
(h) participating relying party;
(i) relying party;
(j) restricted attribute.

Note 2: A number of expressions used in this instrument are defined in [*rule 1.4*] of the Digital ID Rules, including the following:

(a) participating entity;
(b) pairwise identifier.

Note 3: A number of expressions used in this instrument are defined in [*subrule 1.4(2)*] of the Accreditation Rules, including the following:

(a) authenticated session;
(b) commencement of identity credential;
(c) DI data environment;
(d) identity proofing level;
(e) IP level;
(f) single logout;
(g) single sign on.

Note 4: A number of expressions used in this instrument are defined in [*subrule 5(2)*] of the Accreditation Data Standards, including the following:

(a) authentication level.

(1) Expressions defined in the Accreditation Rules, Accreditation Data Standards and the Digital ID Rules have the same meaning in this instrument.

(2) In this instrument:

access channel has the meaning in section 4.1.2 of Schedule 1 (AGDIS Onboarding Specifications).

Accreditation Data Standards means the *Digital ID (Accreditation) Data Standards 2024*.

Accreditation Rules means the *Digital ID (Accreditation) Rules 2024*.

Act means the *Digital ID Act 2024*.

attribute set means a group of attributes specified in any row of Table 4, Table 36 or Table 37 in Schedule 3 (AGDIS Attribute Profile).

Note: Attributes are not unique to a single attribute set. The same attribute may be used across multiple attribute sets. Further information can be found in Schedule 3 (AGDIS Attribute Profile).

attribute sharing policy means any policy mentioned in Chapter 2 or Chapter 3 of Schedule 3 (AGDIS Attribute Profile) that describes rules to be applied when sharing attributes with a participating relying party.

authentication context class reference value has the same meaning as in OpenID Connect Extended Authentication Profile (EAP) ACR Values 1.0.

computed attribute means an attribute that is dynamically derived from the attributes in an attribute set using an algorithm.

consumer history, in relation to an individual, means the history of all the individual's interactions with a participating accredited identity exchange provider.

digital ID identifier means a unique identifier specified in section 2.2.1.1 of Schedule 3 (AGDIS Attribute Profile).

Digital ID Rules means the *Digital ID Rules 2024.*

Document Verification Service has the same meaning as 'DVS' in section 15 of the *Identity Verification Services Act 2023*.

evidence of identity has the same meaning as the National Identity Proofing Guidelines, published by the Attorney-General's Department.

Note: At the time this instrument was made, located at: https://www.ag.gov.au/national-security/publications/national-identity-proofing-guidelines.

federation protocol: see section 5.

general purpose identifier, in relation to an individual, means a unique identifier assigned by a participating accredited identity service provider:

(a) to the individual; and

(b) independently from a participating relying party or a participating accredited identity exchange provider.

ITU E.164 means the standard for international public telecommunication structures, published by the International Telecommunication Union.

Note: At the time this instrument was made, located at: https://www.itu.int/rec/T-REC-E.164-201011-I/en.

JSON Web Key Set has the same meaning as in RFC 7517.

JSON Web Token has the same meaning as in RFC 7517.

JavaScript Object Notation has the same meaning as in RFC 7517 and RFC 8259.

levels of assurance has the meaning given in section 2.1.3 of Schedule 1 (AGDIS Onboarding Specifications).

MAY: see section 8.

MUST and ***MUST NOT***: see section 8.

OAuth 2.0 has the same meaning as in RFC 6749.

OpenID Connect Core 1.0 means the standard for an identity layer which operates on top of RFC 6749, published by the OpenID Foundation.

Note: At the time this instrument was made, located at: https://openid.net/specs/openid-connect-core-1_0.html.

OpenID Connect Extended Authentication Profile (EAP) ACR Values 1.0 means the standard used to request specific authentication context classes, published by the OpenID Foundation.

Note: At the time this instrument was made, located at: https://openid.net/specs/openid-connect-eap-acr-values-1_0.html.

OpenID Connect Core 1.0 provider means an entity which incorporates OpenID Connect Core 1.0 into its operations.

OPTIONAL: see section 8.

Privacy Act means the *Privacy Act 1988*.

proof key for code exchange has the same meaning as in RFC 7636.

RECOMMENDED and ***NOT RECOMMENDED***: see section 8.

REQUIRED: see section 8.

participating accredited attribute service provider means an accredited attribute service provider that is participating in the Australian Government Digital ID System.

participating accredited identity exchange provider means an accredited identity exchange provider that is participating in the Australian Government Digital ID System.

participating accredited identity service provider means an accredited identity service provider that is participating in the Australian Government Digital ID System.

relying party audit ID means a transaction audit identifier for transactions between participating identity exchange providers and participating relying parties.

Note: The relying party audit ID is never shared with a participating identity service provider. A separate audit ID is shared between a participating identity exchange provider and a participating identity service provider.

relying party profile means the profile in Schedule 2 (AGDIS Open ID Connect Profile).

RFC means a document published by the Internet Engineering Task Force that contains technical specifications regarding the internet.

Note: RFCs are freely available. For more information, see: https://www.ietf.org/process/rfcs/.

RFC 2119 means the RFC numbered 2119 and titled *Key Words for use in RFCs to Indicate Requirement Levels*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc2119.

RFC 2821 means the RFC numbered 2821 and titled *Simple Mail Transfer Protocol*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc2821.

RFC 3339 means the RFC numbered 3339 and titled *Date and Time on the Internet: Timestamps.*

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc3339.

RFC 3629 means the RFC numbered 3629 and titled *UTF-8, a transformation format of Unicode and ISO 10646*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc3629.

RFC 3986 means the RFC numbered 3986 and titled *Uniform Resource Identifier (URI): Generic Syntax*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc3986.

RFC 4122 means the RFC numbered 4122 and titled *A Universally Unique IDentifier (UUID) URN Namespace*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc4122.

RFC 5322 means the RFC numbered 5322 and titled *Internet Message Format*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc5322.

RFC 5646 means the RFC numbered 5646 and titled *Tags for Identifying Languages*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc5646.

RFC 6749 means the RF numbered 6749 and titled *The OAuth 2.0 Authorization Framework*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc6749.

RFC 6750 means the RFC numbered 6750 and titled *The OAuth 2.0 Authorization Framework: Bearer Token Usage*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc6750.

RFC 6819 means the RFC numbered 6819 and titled *OAuth 2.0 Threat Model and Security Considerations*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc6819.

RFC 7009 means the RFC numbered 7009 and titled *OAuth 2.0 Token Revocation*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc7009.

RFC 7517 means the RFC numbered 7517 and titled *JSON Web Key (JWK)*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc7517.

RFC 7591 means the RFC numbered 7591 and titled *OAuth 2.0 Dynamic Client Registration Protocol*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc7591.

RFC 7636 means the RFC numbered 7636 and titled *Proof Key for Code Exchange by OAuth Public Clients*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc7636.

RFC 7662 means the RFC numbered 7662 and titled *OAuth 2.0 Token Introspection*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc7662.

RFC 8141 means the RFC numbered 8141 and titled *Uniform Resource Names*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc8141.

RFC 8446 means the RFC numbered 8446 and titled *The Transport Layer Security (TLS) Protocol Version 1.3*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc8446.

RFC 8259 means the RFC numbered 8259 and titled *The JavaScript Object Notation (JSON) Data Interchange Format*.

Note: At the time this instrument was made, located at: https://datatracker.ietf.org/doc/html/rfc8259.

SHALL and ***SHALL NOT***: see section 8.

sector identifier has the same meaning as in OpenID Connect Core 1.0.

technical relying party: see section 6.

transport layer security has the same meaning as in RFC 8446.

uniform resource identifier has the same meaning as in RFC 3986.

uniform resource name has the same meaning as in RFC 8141.

UTF-8 means the standard for encoding electronic communications published by the Unicode Consortium.

Note: At the time this instrument was made, located at: https://www.unicode.org/versions/Unicode15.1.0/.

universally unique identifier has the same meaning as in RFC 4122.

5 Meaning of *federation protocol*

(1) A ***federation protocol*** means an open protocol that enables participating entities to communicate with each other and share attributes of individuals in a trusted manner.

(2) Schedule 2 (AGDIS OpenID Connect Profile) is the only federation protocol for the Australian Government Digital ID System.

6 Meaning of *technical relying party*

A ***technical relying party*** means:

(a) a participating accredited identity exchange provider's OpenID Connect Core 1.0 software used to co-ordinate the flow of data or information between entities participating in the Australian Government Digital ID System; or

(b) a participating relying party's OpenID Connect Core 1.0 software used to:

(i) authenticate the participating relying party with the participating accredited identity exchange; and

(ii) request and receive information or data from the participating accredited identity exchange; or

(c) a participating accredited attribute service provider's OpenID Connect Core 1.0 software used to:

(i) authenticate the participating accredited attribute service provider with the participating accredited identity exchange; and

(ii) request and receive information or data from the participating accredited identity exchange.

Note: To avoid doubt, a technical relying party is not a relying party as defined in section 9 of the Act.

7 Abbreviations

In this instrument, an abbreviation mentioned in column 1 of an item in the following table has the meaning set out in column 2 of that item.

Abbreviations

Item	**Column 1** **Abbreviation**	**Column 2** **Meaning**
1	ACR	authentication context class reference
2	AL	authentication level
3	AGDIS	Australian Government Digital ID System
4	ASP	participating accredited attribute service provider

Abbreviations

Item	Column 1	Column 2
	Abbreviation	**Meaning**
5	BDM	State and/or Territory Registry of Births, Deaths and Marriages
6	CoI credential	commencement of identity credential
7	DVS	Document Verification Service
8	EoI	evidence of identity
9	GPI	general purpose identifier
10	IP level	identity proofing level
11	IP#	identity proofing level number (for example, IP1, IP1 Plus, IP2, IP2 Plus, IP3, IP4)
12	ISP	participating accredited identity service provider
13	IXP	participating accredited identity exchange provider
14	JSON	JavaScript Object Notation
15	JWKS	JSON Web Key Set
16	JWT	JSON Web Token
17	OIDC provider	OpenID Connect Core 1.0 provider
18	PKCE	proof key for code exchange
19	PRP	participating relying party
20	RP audit ID	relying party audit ID
21	SLO	single logout
22	SSO	single sign on
23	TLS	transport layer security
24	TRP	technical relying party
25	URI	uniform resource identifier
26	URN	uniform resource name
27	UUID	universally unique identifier

8 Key terms

In this instrument a capitalised term mentioned in column 1 of an item in the following table has the meaning and effect, in relation to an entity, set out in column 2 of that item subject to any exception or requirement in column 3 of that item.

Key terms

Item	Column 1	Column 2	Column 3
	Term	**Meaning and effect**	**Exception/requirement**
1	**MAY**	optional—the entity has discretion in relation to the behaviour	except where it is necessary to: (a) interoperate with another implementation which does or does not include the option, in which case the entity **MUST** or **MUST NOT** implement the

Key terms			
Item	**Column 1**	**Column 2**	**Column 3**
	Term	**Meaning and effect**	**Exception/requirement**
			behaviour, as the case requires, to interoperate with the other implementation; or (b) comply with a condition imposed on the entity's approval to participate in the AGDIS under section 64 of the Act, in which case the entity **MUST** or **MUST NOT** implement the behaviour, as the case requires, to comply with that condition.
2	**MUST**	absolute requirement—the entity has no discretion in relation to the behaviour	
3	**MUST NOT**	absolute prohibition—the entity has no discretion in relation to the behaviour	
4	**NOT RECOMMENDED**	there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before the entity implements any behaviour to which this term applies	if the entity implements any behaviour to which this term applies, the entity is required to notify the Digital ID Regulator or System Administrator in accordance with the condition imposed on the entity's approval to participate in the AGDIS under [*rule 3.4*] of the Digital ID Rules.
5	**OPTIONAL**	same as item 1	
6	**RECOMMENDED**	there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before the entity implements a different behaviour to which this term applies	if the entity chooses a different course to which this term applies, the entity is required to notify the Digital ID Regulator in accordance with the condition imposed on the entity's approval to participate in the AGDIS under [*rule 3.4*] of the Digital ID Rules.
7	**REQUIRED**	same as item 2	
8	**SHALL**	same as item 2	
9	**SHALL NOT**	same as item 3	
10	**SHOULD**	same as item 6	same as item 6
11	**SHOULD NOT**	same as item 4	same as item 4

9 Incorporated instruments

If a provision of this instrument applies, adopts or incorporates, with or without modification, any matter contained in any other instrument or other writing (***incorporated instrument***), then, unless the contrary intention appears in the provision, the reference to the incorporated instrument is as in force at the commencement of this instrument.

10 Schedules

Each instrument that is specified in a Schedule to this instrument is amended or repealed as set out in the applicable items in the Schedule concerned, and any other provision (however described) in a Schedule to this instrument has effect according to its terms.

Schedule 1 – AGDIS Onboarding Specifications

[Draft Digital ID Data Standard v0.2]

Contents

List of Tables

1. Common requirements for participating accredited entities

This chapter outlines the role requirements, where applicable, that MUST be met by ASPs, ISPs and IXPs.

1.1 Security considerations

ASPs, ISPs and IXPs MUST comply with:

(a) section 10 (Security Considerations) of RFC 6749; and
(b) section 5 (Security Considerations) of RFC 6819.

1.2 Identity resolution

1.2.1 General purposes identifiers

An IXP MUST NOT issue a GPI.

An ISP MAY issue a GPI. Where an ISP chooses to issue a GPI, the ISP MUST:

(a) bind the GPI to only a single digital ID account;
(b) only use the GPI within the AGDIS;
(c) only use the GPI to map the individual's digital ID to a single IXP; and
(d) issue additional GPIs for each IXP an individual accesses.

A GPI MUST be unique within the DI data environment of the ISP that issues it.

1.2.2 Pairwise identifiers

An IXP MUST issue a pairwise identifier.

An ISP SHOULD issue a pairwise identifier. If an ISP issues a pairwise identifier, the ISP MUST NOT issue a GPI.

IXPs or ISPs that issue digital ID identifiers as pairwise identifiers MUST generate the identifiers in accordance with the algorithm outlined in section 8.1 (Pairwise Identifier Algorithm) of OpenID Connect Core 1.0.

1.2.3 Sector Identifiers

IXPs and ISPs SHOULD use Sector Identifiers.

The scope of a Sector Identifier MUST be limited as per the role specific role requirements outlined in the following specified sections of OpenID Connect Core 1.0.

If a Sector Identifier is used, it MUST conform to the definition outlined in the following specified sections of OpenID Connect Core 1.0.

For this standard, the following sections of OpenID Connect Core 1.0 are specified:

(a) section 1.2 (Terminology); and
(b) section 8.1 (Pairwise Identifier Algorithm).

1.3 Data sharing

Irrespective of the federation protocol used, ASPs, ISPs and IXPs MUST handle and transmit data in accordance with the data sharing policies outlined in Schedule 3 (AGDIS Attribute Profile).

1.3.1 Request processing

ASPs, ISPs and IXPs MUST NOT attempt to fulfill requests for unknown attributes or attribute sets.

Requests for known attributes or attribute sets that cannot be fulfilled, regardless of the reason, MUST NOT be returned with empty values unless explicitly permitted by an attribute sharing policy as outlined in Schedule 3 (AGDIS Attribute Profile).

2. Identity exchange provider

This chapter outlines the requirements that MUST be met by IXPs.

2.1 Technical integration requirements

2.1.1 Protocol requirements

An IXP MUST implement Schedule 2 (AGDIS OpenID Connect Profile) for an:

(a) IXP acting as an OIDC provider; and
(b) IXP acting as TRP to an ISP.

2.1.2 Audit requirements

An IXP MUST generate the RP Audit ID attribute as outlined in Schedule 3 (AGDIS Attribute Profile).

The IXP MUST provide the RP Audit ID attribute it generated in response to every logical transaction between itself (the IXP) and the PRP (including ASPs).

The IXP MUST include the RP Audit ID attribute it generated for the PRP's authentication request in every logical transaction between itself and each of the ASPs required to fulfill the attribute requirements of a PRP's authentication request.

The IXP MUST NOT send the RP Audit ID attribute it generated for the PRP's authentication request to its ISP.

2.1.3 Levels of assurance

Levels of assurance are used to defined the IP level and AL of an individual's authenticated session.

Levels of assurance are ranked from the lowest to the highest degree of confidence in the authentication process. The rankings of levels of assurance are specified in Table 1 below. Note that the URNs in Table 1 use a legacy acronym in their namespace.

A mapping of the new authentication levels to legacy credential levels, used in the URNs outlined in Table 1, is provided below in Table 2.

An IXP MUST allow a PRP to request a minimum level of assurance when making authentication requests.

An IXP MUST publish the levels of assurance supported by its connected ISPs as outlined in Table 1.

Table 1 Level of assurance combinations used in the AGDIS.

Identity proofing level	Authentication level	URN	Ranking (low to high)
IP1	AL1	`urn:id.gov.au:tdif:acr:ip1:cl1`	1

Identity proofing level	Authentication level	URN	Ranking (low to high)
	AL2	`urn:id.gov.au:tdif:acr:ip1:cl2`	2
	AL3	`urn:id.gov.au:tdif:acr:ip1:cl3`	3
IP1 Plus	AL1	`urn:id.gov.au:tdif:acr:ip1p:cl1`	4
	AL2	`urn:id.gov.au:tdif:acr:ip1p:cl2`	5
	AL3	`urn:id.gov.au:tdif:acr:ip1p:cl3`	6
IP2	AL2	`urn:id.gov.au:tdif:acr:ip2:cl2`	7
	AL3	`urn:id.gov.au:tdif:acr:ip2:cl3`	8
IP2 Plus	AL2	`urn:id.gov.au:tdif:acr:ip2p:cl2`	9
	AL3	`urn:id.gov.au:tdif:acr:ip2p:cl3`	10
IP3	AL2	`urn:id.gov.au:tdif:acr:ip3:cl2`	11
	AL3	`urn:id.gov.au:tdif:acr:ip2p:cl2`	12
IP4	AL3	`urn:id.gov.au:tdif:acr:ip4:cl3`	13

Table 2 Mapping of new authentication levels to legacy credential levels.

Authentication level	Credential level
AL1	CL1
AL2	CL2
AL3	CL3

2.1.4 Identity resolution

An IXP MUST generate pairwise identifiers as required in section 1.2.2 of this Schedule.

If an IXP does not use sector identifiers, the IXP MUST generate a pairwise identifier for every distinct service it connects, even if these services are operated by the same PRP.

If an IXP makes use of sector identifiers:

(a) the pairwise identifiers MUST only have a one-to-one mapping with a sector identifier; and

(b) a sector identifier MUST only map to a single PRP, regardless of the number of connected services.

Pairwise identifiers MUST be used when presenting individuals to PRPs irrespective of the federation protocols being brokered by the IXP in a transaction.

2.1.5 Authenticated sessions

An authenticated session managed by the IXP MUST expire.

For each supported federation protocol, an IXP MUST provide PRPs with information regarding session expiration times.

An IXP MAY support features as specified in the relevant federation protocol to allow refreshing of an authenticated session before expiry.

2.1.6 Single sign on

An IXP MAY support a SSO scheme.

If an IXP supports SSO, it MUST support the ability for a PRP to request authentication for a particular individual, using all the methods outlined in the profile of their supported federation protocols.

An IXP MUST permit a PRP to request an individual's reauthentication even when a pre-existing session is active and valid, regardless of the PRP that created the session.

An IXP MUST ensure their implementation of SSO does not result in the disclosure of attributes including restricted attributes of an individual.

2.1.7 Single logout

If an IXP supports SSO, the IXP MUST implement SLO.

The IXP MUST ensure their implementation of SLO does not result in the disclosure of attributes including restricted attributes of an individual.

2.1.8 Attribute service provider integration

When an IXP receives an authentication request from a PRP that includes attributes supplied by an ASP, the IXP MUST facilitate gathering those attributes from the ASP.

An IXP MUST implement mechanisms to support the access channels it has agreed with an ASP as referenced in section 4.1.2 of this Schedule.

When accessing attributes directly from the ASP, an IXP MUST do so using the pairwise identifier it has issued to the ASP.

To facilitate direct communication between an ASP and a PRP, an IXP MAY share an individual's encrypted pairwise identifier:

(a) for the ASP with the PRP requesting authentication; and
(b) for the PRP with the ASP that can fulfill the attribute requests.

If an IXP chooses to share an encrypted identifier, the identifier MUST be:

(a) encrypted using a public key of the PRP or ASP it belongs to; and
(b) be packaged with a timestamp and a nonce to limit the opportunity for replay and the creation of de facto pairwise identifiers.

2.1.9 Federation protocol brokering

When accepting authentication requests from a PRP, an IXP MUST broker the federation protocol used by the PRP to the federation protocol of the ISP selected by the individual.

2.2 Identity provider selection

An IXP MUST provide a mechanism for the individual to choose an ISP.

The list of ISPs presented by an IXP MUST only display ISPs that satisfy the IP level and AL requested in the PRP's authentication request.

An IXP MAY provide a mechanism for the individual to remember their choice of ISP for a given PRP.

If an individual has remembered their ISP choice for a given PRP, an IXP MAY redirect them to the remembered ISP.

If an IXP provides a mechanism to remember ISP selection, the IXP MUST provide:

(a) a notice to ensure the individual understands the nature of the express consent they are providing;
(b) a notice outlining the duration of the remembered ISP selection and the limitations on how it is remembered (for example, if it is limited to the device or web browser from which the express consent is given); and
(c) a mechanism to revoke the remembered choice.

2.2.1 Blinding

An IXP MUST NOT broker any information about the PRP requesting authentication to any of its ISPs.

An IXP MAY inform the PRP which ISP the individual used to authenticate.

If an IXP informs the PRP which ISP the individual used to authenticate, the IXP MUST do so in accordance with the requirements outlined for the federation protocol used by the PRP.

2.3 User dashboard

An IXP MUST provide a user dashboard that allows an individual, for the digital ID used to create the individual's current authenticated session, to:

(a) view their consumer history;
(b) manage the express consent they have given; and
(c) manage their linked PRPs.

The user dashboard MUST only display information for services that match the level of assurance for the current authenticated session.

An IXP MAY re-authenticate the individual when accessed even if the individual has an active and valid session.

2.4 Data requirements

2.4.1 Attribute requirements

An IXP MUST support brokering all attributes as outlined in Schedule 3 (AGDIS Attribute Profile).

An IXP MUST support the disclosure of all IXP managed or generated attributes as outlined in Schedule 3 (AGDIS Attribute Profile).

An IXP MUST support requesting any ISP specific attributes outlined in Schedule 3 (AGDIS Attribute Profile) in an authentication request to an ISP.

2.4.2 Computed attributes

For each of their supported federation protocols, an IXP MUST implement the following requirements.

An IXP MUST support the disclosure of computed attributes as described in Schedule 3 (AGDIS Attribute Profile).

An IXP MAY source computed attributes from an ISP or an ASP.

An IXP MAY define support for additional computed attributes derived from attributes available from its ISPs and ASPs.

Any new computed attribute MUST NOT violate attribute sharing policies defined in Schedule 3 (AGDIS Attribute Profile).

2.4.3 Attribute sharing policy

The IXP MUST only disclose an attribute or attribute set with PRPs in accordance with that attribute or attribute set's attribute sharing policy as defined in Schedule 3 (AGDIS Attribute Profile).

2.4.4 Data representation

When disclosing IXP specific attributes, an IXP MUST use the data representation as prescribed in Schedule 3 (AGDIS Attribute Profile).

An IXP MAY use the data representation, defined in Schedule 3 (AGDIS Attribute Profile), to validate attribute payloads received from ISPs and ASPs.

An IXP MUST NOT alter payloads received from an ISP or ASP unless the relevant attribute sharing polices explicitly permit alteration to occur.

3. Identity service provider

This chapter outlines the requirements that MUST be met by ISPs.

3.1 Technical integration requirements

3.1.1 Protocol requirements

An ISP MUST implement the ISP requirements in Schedule 2 (AGDIS OpenID Connect Profile).

3.1.2 Identity resolution

An ISP MAY generate GPIs as prescribed in section 1.2.1 of this Schedule.

An ISP SHOULD generate pairwise identifiers as prescribed in section 1.2.2 of this Schedule.

An identifier linking a digital ID to an IXP MUST only be used to map that one-to-one relationship. The identifier MUST NOT be shared by the IXP with any other service connected to the ISP.

3.1.3 Single sign on

An ISP MAY support an SSO scheme operated by one or more of its connected IXPs.

Where an ISP chooses to support an IXP's SSO scheme, the ISP MAY choose to re-authenticate an individual at their discretion when servicing a SSO request.

If an ISP is using securely cached attributes for SSO and they receive an authentication request which cannot be fulfilled using the cached information the individual MUST be re-authenticated.

3.1.4 Single logout

The ISP MUST support the SLO scheme operated by its connected IXPs.

The ISP MUST support the SLO scheme (of its connected IXPs) regardless of whether the ISP has supported the SSO.

3.2 Data requirements

3.2.1 Attribute requirements

An ISP MUST support the disclosure of attributes based on support and fulfilment requirements outlined in Schedule 3 (AGDIS Attribute Profile).

For each of their supported federation protocols, an ISP MUST implement the specific attribute profile and attribute mapping.

3.2.2 Computed attributes

An ISP MUST support computed attributes as outlined in Schedule 3 (AGDIS Attribute Profile).

An ISP MAY define support for additional computed attributes. The new computed attribute MUST NOT violate attribute sharing policies defined in Schedule 3 (AGDIS Attribute Profile).

4. Attribute service provider

This chapter outlines the requirements that MUST be met by ASPs.

4.1 Technical integration requirements

4.1.1 Protocol requirements

An ASP, in addition to these requirements, MUST be a PRP.

The ASP MUST implement the relying party profile for one of the federation protocols supported by the IXP it is connecting to.

At the time this instrument was made, the only relying party profile for the AGDIS is in Schedule 2 (AGDIS OpenID Connect Profile).

4.1.2 Access channels

An access channel is any mechanism that allows ASP managed attributes to be provided either:

(a) directly to a PRP; or
(b) to a PRP via an IXP.

Examples include, but are not limited to, the use of OpenID Connect Core 1.0 distributed claims, event streaming services or application programming interfaces.

An ASP MAY make multiple access channels available for IXPs or PRPs to access the attributes they control.

For each access channel that an ASP makes available, the ASP MUST:

(a) provide documentation outlining the technical integration requirements for the IXP and/or the PRP, as the case may be;
(b) outline the attribute lifecycle features supported by the channel and how the IXP and/or PRP, as the case may be, can use them;
(c) document and demonstrate how an individual's consent is collected and managed;
(d) document and demonstrate how an individual can manage their consent; and
(e) document how the audit logging will be undertaken.

4.2 Audit logging

An ASP's audit log MUST include any individual's consent managed by the ASP that enables the sharing of attributes with PRPs.

An ASP's audit log MUST include the value of the RP Audit ID supplied by the IXP:

(a) when binding a digital ID brokered by the IXP to any of the attributes managed by the ASP;
(b) if the IXP directly retrieves the attributes; and

(c) if the ASP provides attributes indirectly and the RP Audit ID is available.

4.3 Attribute schema

An ASP MUST publish an attribute schema for any attributes it provides.

The attribute schema MAY support multiple data formats if required by the various access channels provided by the ASP to IXPs and PRPs to access the ASP's attributes.

A published attribute schema MUST outline data types and constraints for the fields that comprise the attribute sets managed by the ASP.

The published attribute schema MUST support a data format compatible with the federation protocol the ASP uses to connect to the IXP.

An ASP MUST establish procedures to publish updates to their attribute schema in accordance with Schedule 3 (AGDIS Attribute Profile).

Schedule 2 – AGDIS Open ID Connect Profile

[Draft Digital ID Data Standard v0.2]

Contents

List of Tables

List of Figures

1. Identity exchange

This chapter outlines the requirements that an IXP MUST satisfy to facilitate brokering access to ISPs using OpenID Connect Core 1.0 as their federation protocol.

Interoperability between IXPs is not within scope for this release of the *Digital ID (AGDIS) Data Standards 2024*.

1.1 Authorisation grant types

An IXP MUST NOT use the resource owner password credentials grant type defined in RFC 6749.

1.2 Client types

1.2.1 Full Client with User Delegation

An IXP MUST support the Full Client with User Delegation client type.

This client type applies to clients that act on behalf of a particular resource owner and require delegation of that user's authority to access the protected resource. Furthermore, these clients can interact with a separate web browser application to facilitate the resource owner's interaction with the authentication endpoint of the authorisation server.

All IXP clients of this type MUST use the authorisation code flow of OAuth 2.0 RFC 6749 by sending the resource owner to the authorisation endpoint to obtain authorisation.

An IXP MUST ensure that the individual authenticates to the authorisation endpoint.

The user's web browser is then redirected back to a URI hosted by the client service, from which the client can obtain an authorisation code passed as a query parameter. The client then presents that authorisation code along with its own credentials (`private_key_jwt`) to the authorisation server's Token Endpoint to obtain an access token.

An IXP MUST associate the clients with a unique public key as described in section 1.5.1 of this Schedule.

An IXP MAY issue this client type a refresh token if the security parameters of the access request allow for it.

1.2.2 Native Client with User Delegation

A Native Client with User Delegations is a client that acts on behalf of a particular resource owner, such as an application on a mobile platform, and requires delegation of that individual's authority to the protected resource. Furthermore, these clients can interact with a separate web browser application to facilitate the resource owner's interaction with the authentication endpoint of the authorisation server. Specifically, this client type runs natively on the resource owner's device, often leading to many identical instances of a piece of software operating in different environments and running simultaneously for different end users.

An IXP MAY support Native Clients with User Delegation.

If an IXP supports Native Clients with User Delegation, the IXP MUST implement the following requirements.

An IXP client MUST use the authorisation code flow of RFC 6749 by sending the resource owner to the authorisation endpoint to obtain authorisation.

An IXP MUST authenticate an individual to the authorisation endpoint. The user's web browser is then redirected back to a URI hosted by the client, from which the client can obtain an authorisation code passed as a query parameter. The client then presents that authorisation code along to the authorisation servers Token Endpoint to obtain an access token.

Native clients connecting to a IXP MUST either be:

(a) dynamically registered to obtain a separate client identifier for each instance; or
(b) act as Public Clients (as defined by section 2.1 of RFC 6749) by using a common client identifier and using PKCE (as per RFC 7636) to protect calls to the Token Endpoint.

An IXP supporting dynamic registration of native applications MUST support one of the following methods to register or exchange a unique public key value:

(a) the native application generates unique public and private keys on the device and registers the public key value with the IXP; or
(b) the IXP generates a unique public and private key pair, registering the public key with itself and securely transmitting the public and private key pair to the client. After transmission, the IXP MUST discard the private key.

An IXP MUST NOT permit sharing of client credentials among instances of client software.

All native applications registered with a IXP not registering a separate public key for each instance are considered Public Clients and MUST use PKCE with the S256 code challenge mechanism (as defined in RFC 7636).

An IXP MUST NOT permit Public Clients to authenticate to the Token Endpoint in any other way.

If an IXP supports Native Clients with User Delegation, the IXP MAY implement the following requirements.

An IXP MAY permit dynamically registered native applications to use PKCE.

An IXP MAY issue a refresh token to a Native Client with User Delegation if the IXP is satisfied that there are no security issues.

1.2.3 Direct Access Client

A Direct Access Client is a client that connects directly to protected resources and do not act on behalf of a particular resource owner, for example, machine to machine access.

An IXP MAY only implement this client type to support interactions between itself and other entities participating in the AGDIS.

If an IXP supports the Direct Access Client type, the IXP MUST support the confidential client type and the `client_credentials` grant types.

1.3 Client Registration

An IXP MUST support Client Registration by static configuration or dynamic configuration. An IXP MAY support both static and dynamic Client Registration.

All clients of an IXP MUST register with the authorisation server.

For client software that may be installed on multiple client instances, an IXP MUST issue each client instance a unique client identifier from the authorisation server.

An IXP MUST advise a PRP of the information that is required to be supplied when configuring its connection as a TRP of the IXP.

An IXP MUST require that a TRP seeking to register dynamically provides an initial access token (as defined in RFC 7591).

If an IXP supports dynamic registration of clients, the IXP MUST implement support for bearer tokens in the manner prescribed in RFC 6750.

1.4 Redirect URI

An IXP's clients that use the authorisation code grant type MUST register its full redirect URIs.

An IXP MUST as the authorisation server validate the redirect URI given by the client at the authorisation endpoint using strict string comparison.

An IXP MUST ensure that the redirect URI used by a client is one of the following:

(a) hosted on a website with TLS protection (HTTPS);
(b) hosted on a local domain of the client (for example: http://localhost/); or
(c) hosted on a client specific non-remote protocol URI scheme (for example: myapp:// or au.gov.app://).

If a client's redirect URI is either hosted on the local domain of the client or hosted on a client specific non-remote protocol URI schema, then a IXP MAY require that the TRP uses the PKCE extension to the authorisation code flow.

An IXP MUST NOT allow its ASPs to have URIs in more than 1 of the 3 categories outlined above.

An IXP SHOULD NOT allow ASP clients to have multiple redirects URIs on different domains.

1.4.1 Native Client Applications

The use of client specific non-remote protocol URI schemes SHOULD be phased out for native application.

An IXP MAY allow existing native application clients to continue using non-remote protocol URI schemes in their redirect URI.

When a new native client application is running on a platform that supports Claimed HTTPs Scheme URI redirection, an IXP SHOULD require these native applications to use that scheme in their redirect URI.

1.5 Connection to the authorisation server

1.5.1 Client keys

An IXP MUST require clients using the authorisation code grant type to have a public and private key pair type for use in authentication to the Token Endpoint.

An IXP MUST require each client to register its public keys in its client registration metadata by either sending the public key directly in the `jwks` field or by registering a `jwks_uri`.

If a client registers a `jwks_uri`, the IXP MUST require the URI to be reachable by the authorisation server.

An IXP SHOULD require the use of a `jwks_uri` as it allows for easier key rotation.

An IXP MUST reject a `jwks` field or the content available from a `jwks_uri` provided by the client if the content does not contain a public key in the format prescribed in RFC7517.

As the authorisation server, a IXP MUST verify the content available from a client's registered `jwks_uri` contains a valid JSON Web Key Set.

The example below is of a 2048-bit RSA key.

```
{

    "keys": [{
           "alg": "RS256",

           "e": "AQAB",

           "n":
"kAMYD62n_f2rUcR4awJX4uccDt0zcXRssq_mDch5aifcShx9aTtTVza23PTn3KaKrsBXwWcfioXR
6zQn5eYdZQVGNBfOR4rxF5i7t3hfb4WkS50EK1gBYk2lO9NSrQzxG9QsUsAnN6RHksXqsdOqvnxjL
exDfIJlgbcCN9h6TBC66ZXv7PVhl19gIYVifSU7liHkLe0l0fw7jUI6rHLHf4d96_neR1HrNIK_xs
sr99Xpv1EM_ubxpktX0T925qej9fMEpzzQ5HLmcNt1H2_VQ_Ww1JOLn9vRnH48FDj7TxlIT74XdTZ
gTv31w_GRPAOfyxEw_ZUmxhz5ZngTlQ",

           "kty": "RSA",

           "kid": "oauth-client"

    }]

}
```

Figure 1 JSON WEB Key Set example

An IXP MAY allow native client applications to omit their key during registration if they are a Public Client using PKCE.

1.6 Grant types

The grant type of `authorisation_code` MUST be supported by an IXP.

The Authorisation Code authentication flow implemented by an IXP MUST follow the steps outlined in the section 3.1 of OpenID Connect Core 1.0.

1.7 Relying party profile

In this section the terms client and technical relying party refer to the OpenID Connect Core 1.0 software operated by a PRP or an ASP.

1.7.1 Requests to the Authorisation Endpoint

IXP's clients making a request to the authorisation endpoint SHOULD use an unpredictable value for the state parameter with at least 128 bits of entropy.

IXP's clients MUST validate the value of the state parameter upon return to the redirect URI and MUST ensure that the state value is securely tied to the individual's current session, for example, by relating the state value to a session identifier issued by the client software to the browser.

IXP's clients MUST include their full redirect URI in the authorisation request.

To prevent open redirection and other injection attacks, an IXP MUST match the entire redirect URI using a direct string comparison against registered values and MUST reject requests with invalid or missing redirect URIs.

The Authentication Request MUST contain the following REQUIRED parameters and MAY contain the following OPTIONAL parameter values:

- `client_id`
  - REQUIRED. OAuth 2.0 Client Identifier valid at the authorisation server.
- `response_type`
  - REQUIRED. Must be set to `code`.
- `scope`
  - REQUIRED. Indicates the attributes being requested. The `openid` scope MUST always be present. Additional scopes are defined in Schedule 3 (AGDIS Attribute Profile).
- `redirect_uri`
  - REQUIRED. Indicates a valid endpoint where the client will receive the authentication response. The URI MUST match exactly one of the Redirection URIs preregistered at the IXP. The URI MUST follow the schemes outlined in section 1.4 of this Schedule.
- `state`
  - REQUIRED. Un-guessable random string generated by the client used to protect against CSRF attacks. MUST contain sufficient entropy to avoid guessing and is returned to the Client in the authentication response.
- `prompt`
  - OPTIONAL. A space delimited, case sensitive list of string values that specifies if the authorisation server prompts for the End-User to re-authenticate or provide consent. Defined values that a IXP MUST support are:

        - `none`: The authorisation server MUST NOT display any authentication or consent user interface pages. An error is returned if the End-User is not already authenticated or not already provided consent for the requested claims or does not fulfil any other conditions for processing the request.
        - `login`: The authorisation server MAY prompt the End-User for reauthentication.
        - `consent`: The authorisation server MAY prompt the end user for consent before returning information to the client. Consent should be requested in accordance with the attribute sharing policies defined in Schedule 3 (AGDIS Attribute Profile).
        - `select_account`: The authorisation server MAY prompt the End-User to select a user account. This allows an end user with multiple accounts at the authorisation server to select amongst their accounts that currently have an active session at the authorisation server.
- `display`
    - OPTIONAL: A string value that specifies how the authorisation server displays the authentication and consent interface pages to the End-User.
        - `page`: The authorisation server SHOULD display the authentication and consent user interface consistent with a full User Agent page view. This is the default where the display parameter is not specified.
        - `popup`: The authorisation server MAY display the authentication and consent user interface consistent with a popup User Agent window. The popup User Agent window should be of an appropriate size for a login-focused dialog and should not obscure the entire window that it is popping up over.
        - `touch`: The authorisation server MAY display the authentication and consent user interface consistent with a device that leverages a touch interface.
        - `wap`: The authorisation server MAY display the authentication and consent user interface consistent with a "feature phone" type display.
- `nonce`
    - REQUIRED. Un-guessable random string generated by the client, used to protect against cross site request forgery attacks. MUST contain sufficient entropy to avoid guessing. Returned to the client in the ID Token.
- `acr_values`
    - OPTIONAL. A TRP may specify the required level(s) of assurance here. For permissible ACR values see section 1.9.3 of this Schedule.
- `code_challenge` and `code_challenge_method`
    - OPTIONAL. If an IXP supports PKCE as described in section 1.2.2 of this Schedule, they MUST support these parameters.
- `max_age`
    - OPTIONAL. Maximum Authentication Age. Specifies the allowable elapsed time in seconds since the last time the End-User was actively authenticated by the IXP. If the elapsed time is greater than this value, the IXP MUST attempt to actively re-authenticate the End-User.
- `ui_locales`
    - OPTIONAL. End-User's preferred languages and scripts for the user interface, represented as a space-separated list of language tag values as specified in RFC 5646, ordered by preference.
- `id_token_hint`
    - OPTIONAL. ID Token previously issued by the authorisation server being passed as a hint about the End-User's current or past authenticated session with the client. If the

End-User identified by the ID Token is logged in or is logged in by the request, then the Authorization Server returns a positive response; otherwise, it MAY return an error, such as login_required.

- `login_hint`
    - OPTIONAL. Hint to the authorisation server about the login identifier the End-User might use to log in (if necessary). This hint can be used by a TRP if it first asks the End-User for their e-mail address (or other identifier) and then wants to pass that value as a hint to the discovered authorisation service.
- `user_flow`
    - OPTIONAL. A string value that indicates the desired user flow for the individual. Defined values are:
        - `sign_in`: A TRP requests this flow when it expects the user to already have a digital identity and sign in at the ISP.
        - `sign_up`: A TRP requests this flow when it expects the user to need to create a digital identity at an ISP.
        - `mygov_link`: A TRP requests this when it expects a user to set up a linkage between their myGov account and their digital ID.
- `claims`
    - OPTIONAL. This parameter is used to request that specific Claims be returned. The value is a JSON object listing the requested Claims. This is made according to section 5.5 of OpenID Connect Core 1.0

A sample HTTP GET request may look like the following example:

```
https://idexchange.gov.au/oidc/authorization?response_type=code
&client_id=827937609728-m2mvqffo9bsefh4di90saus4n0diar2h
&scope=profile%20openid%20email
&redirect_uri=https%3A%2F%2Frp.agency.gov.au%2Foidc%2FloginResponse
&state=2ca3359dfbfd0 &prompt=select_account
&acr_values=urn%3Aid.gov.au%3Atdif%3Aacr%3Aip3%3Acl2
```

Figure 2 Example authorisation request

1.7.2 Requests to the Token Endpoint

Requests to the Token Endpoint uses client authentication. The client authentication mechanism is signed JWT as defined in section 1.8.1 of this Schedule.

The JWT assertion used in the client authentication MUST be signed by the client using the client's public key it has registered with the Authorisation server. Registration of keys should occur in accordance with process outlined section 1.5.1 of this Schedule.

For clients that are required to use PKCE as described in section 1.2.2 and section 1.4 of this Schedule, the following claims MUST be included in the request to the Token Endpoint.

- `code_verifier`
    - Code verifier generated by client to use PKCE with the S256 code challenge mechanism.

A TRP MUST include the following claims in the request to the Token Endpoint, when requesting authentication for an individual:

- `grant_type`
    - MUST be set to `authorization_code`
- `code`
    - The value of the code parameter returned in the authorisation response.
- `redirect_uri`
    - Value MUST be identical to the `redirect_uri` parameter that was included in the authorisation request as described above.
- `client_assertion_type`
    - MUST be set to `urn:ietf:params:oauth:client-assertion-type:jwt-bearer`
- `client_assertion`
    - The value of the signed client authentication JWT generated as described below in the ID Token section (section 1.7.3.1 of this Schedule). The Client MUST generate a new assertion JWT for each call to the Token Endpoint.

These would be sent to the Token Endpoint as illustrated in the HTTP POST example below.

```
POST /token HTTP/1.1 Content-Type: application/x-www-form-urlencoded Host:
idexchange.gov.au grant_type=authorization_code &code=sedaFh
&scope=openid+email+profile
&redirect_uri=https%3A%2F%2Frp.agency.gov.au%2Foidc%2FloginResponse
&client_id=55f9f559-2496-49d4-b6c3-351a586b7484
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclientassertion-
type%3Ajwt-bearer
&client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.ew0KICAgImlzcy
I6ICI1NWY5ZjU1OS0yNDk2LTQ5ZDQtYjZjMy0zNTFhNTg2Yjc0ODQiLA0KICAgInN1YiI
6ICI1NWY5ZjU1OS0yNDk2LTQ5ZDQtYjZjMy0zNTFhNTg2Yjc0ODQiLA0KICAgImF1ZCI6
ICJodHRwczovL2lkcC1wLmV4YW1wbGUuY29tL3Rva2VuIiwNCiAgICJpYXQiOiAxNDE4N
jk4Nzg4LA0KICAgImV4cCI6IDE0MTg2OTg4NDgsDQogICAianRpIjogIjE0MTg2OTg3OD
gvMTA3YzRkYTUxOTRkZjQ2M2U1MmI1Njg2NWM1YWYzNGU1NTk1Ig0KfQ.t_gX8JQGq3G2
OEc2kUCQ8zVj7pqff87Sua5nktLIHj28l5onO5VpsL4sRHIGOvrpo7XO6jgtPWy3iLXv3
NLyo1TWHbtErQEGpmf7nKiNxVCXlGYJXSDJB6shP3OfvdUc24urPJNUGBEDptIgT7Lhf6
BbwQNlMQubNeOPRFDqQoLWqe7UxuI06dKX3SEQRMqcxYSIAfP7CQZ4WLuKXb6oEbaqz6g
L4l6p83G7wKGDeLETOTHsztjKR38v4F_MnSrx8e0iIqgZwurW0RtetEWvynOCJXkp166T
7qZR45xuCxgOotXY6O3et4n77GtgspMgOEKj3b_WpCiuNEwQ
```

Figure 3 Sample Token Endpoint HTTP POST request

1.7.3 Token response

The token response includes an access token, which can be used to make a UserInfo request as per section 1.7.4 of this Schedule, and an ID token as per section 3.1.3.3 of OpenID Connect Core 1.0. It MAY also include a Refresh token.

1.7.3.1 ID Tokens

All clients MUST validate the signature of an ID Token before accepting it using the public key of the issuing server, published in JWK format.

An IXP MAY encrypt ID Tokens using the appropriate key of the requesting client.

An IXP's TRPs MUST verify the following in received ID tokens:

- `iss`
  - The Issuer field is the URL of the expected issuer.
- `aud`
  - The audience field contains the client ID of the client.
- `exp`, `iat`, `nbf`
  - The expiration, issued at and not before tokens are dates (integer number of seconds since 00:00:00Z 1st January 1970, i.e. Unix epoch) are within acceptable ranges.

1.7.4 Request to the UserInfo Endpoint

An IXP MUST be able to accept UserInfo Request from clients using either the HTTP GET or POST methods.

An IXP MUST only accept Access Tokens from its clients when presented as Bearer Tokens, as outlined in the RFC 6750.

1.7.5 Request Object

A client MAY send request to the authorisation end point using the request parameters as outlined in OpenID Connect Core 1.0.

Request objects MUST be signed by the client's public key.

A client MAY encrypt the request object with the authorisation server's public key.

1.7.6 Discovery

Clients and protected resources MAY cache an IXP's OpenID Connect metadata once the IXP has been discovered, as outlined in this Schedule.

1.8 Identity Exchange OpenID Connect provider Profile

1.8.1 Connecting to clients

1.8.1.1 Grant types

The `authorization_code` grant type is the only grant type that is supported under this Schedule. Accordingly, an IXP MUST only support the `authorization_code` grant type when fulfilling PRP authentication requests for individuals.

The authorisation code flow returns an authorisation code to the client. The client can then exchange this one-time code for an ID Token and an Access Token. This provides the benefit of not exposing any tokens to the User Agent and potentially malicious applications with access to the User Agent.

1.8.1.2 Client authentication

An authorisation server MUST enforce client authentication for access to the authorisation server's Token Endpoint.

An authorisation server MUST only authenticate clients using the `private_key_jwt` method as prescribed in the OpenID Connect Core 1.0.

An authorisation server MUST NOT authenticate clients using any other method.

The JWT used to authenticate the client MUST expire with a lifetime no longer than 5 minutes.

An authorisation server MUST reject an JWT with an expiry time passed.

An authorisation server SHOULD:

(a) allow for clock skew between system when assessing the expiry of a JWT; and
(b) reject any JWT with expiry that is unreasonably far into the future.

The JWT MUST contain the following REQUIRED claims and MAY contain the following OPTIONAL claims:

- iss
  - REQUIRED. Issuer. This MUST contain the client_id of the client creating the token.
- sub
  - REQUIRED. Subject. This MUST contain the client_id of the client creating the token.
- aud
  - REQUIRED. Audience. The value that identifies the authorisation server as an intended audience. The authorisation server MUST verify that it is an intended audience for the token. The Audience MAY be the URL of the authorisation server's Token Endpoint.
- jti
  - REQUIRED. JWT ID. A unique identifier for the token generated by the client, which can be used to prevent reuse of the token. This identifier MUST contain at least 128 bits of entropy and MUST NOT be re-used by any subsequent authentication token.
- exp
  - REQUIRED. Expiration time on or after which the ID Token MUST NOT be accepted for processing.
- iat
  - OPTIONAL. Time at which the JWT was issued.

The following is an example of the use of the REQUIRED claims for a client authentication JWT as defined in this Schedule. Noting that additional claims MAY be included in this set.

```
{
    "iss": "55f9f559-2496-49d4-b6c3-351a586b7484",
    "sub": "55f9f559-2496-49d4-b6c3-351a586b7484",
    "aud": "https://idexchange.gov.au/token",
    "iat": 1418698788,
    "exp": 1418698848,
    "jti": "1418698788/107c4da5194df463e52b56865c5af34e5595"
}
```

Figure 4 Client authentication JWT example

1.8.1.3 Dynamic registration

An IXP MAY support the dynamic registration of clients.

1.8.1.4 Discovery

Endpoints and parameters specified in the Discovery document below MAY be considered public information regardless of the existence of the discovery document.

An IXP MUST provide a well-known endpoint for its configuration as described in the OpenID Connect Discovery 1.0.

An IXP MUST secure the well-known endpoint MUST secured as outlined in OpenID Connect Discovery 1.0. An IXP MAY apply additional security controls if required by their business need.

The discovery document published by a IXP MUST as a minimum contain the following fields:

- `issuer`
    - REQUIRED. The fully qualified issuer URL of the server.
- `authorization_endpoint`
    - REQUIRED. The fully qualified URL of the server's authorisation endpoint defined by RFC 6749.
- `token_endpoint`
    - REQUIRED. The fully qualified URL of the server's Token Endpoint defined by RFC 6749.
- `introspection_endpoint`
    - OPTIONAL. The fully qualified URL of the server's introspection endpoint defined by the RFC 7662.
- `revocation_endpoint`
    - OPTIONAL. The fully qualified URL of the server's revocation endpoint as defined by RFC 7009.
- `jwks_uri`

  - REQUIRED. The fully qualified URI of the server's public key in JWK Set format as defined in RFC 7517.
- `scopes_supported`
  - REQUIRED. The list of scopes that MUST be made available by a IXP as defined in Schedule 3 (AGDIS Attribute Profile).
- `claims_supported`
  - REQUIRED. The list of claims that MUST be made available by an IXP as defined in Chapter 4 of Schedule 3 (AGDIS Attribute Profile).

The following is sample of the current AGDIS IXP's well-known endpoint response.

```
{

    "issuer": "https://auth.identity.gov.au",

    "authorization_endpoint": "https://auth.identity.gov.au/authorise",

    "token_endpoint":

        "https://auth.identity.gov.au/sso/sps/oauth/oauth20/token",

    "userinfo_endpoint":
            "https://auth.identity.gov.au/sso/sps/oauth/oauth20/userinfo",

    "jwks_uri":

            "https://auth.identity.gov.au/.well-known/jwks.json",

    "end_session_endpoint": "https://auth.identity.gov.au/logout",

    "response_types_supported": [

        "code"

    ],

    "grant_types_supported": [

        "authorization_code"

    ],

    "subject_types_supported": [

        "pairwise"

    ],

    "id_token_signing_alg_values_supported": [

        "RS256"

    ],

    "scopes_supported": [
        "openid",
        "profile",
        "email",
        "phone",
        "tdif_business_authorisations",
        "tdif_doc",
        "tdif_other_names"
    ],
    "claims_supported": [
        "tdif_business_authorisations",
        "tdif_doc",
        "acr"
    ],
    "user_flows_supported": [
        "sign_in",
        "sign_up",
        "mygov_link"
    ],
    "prompts_supported": [
        "none",
        "login"
    ],
    "display_values_supported": [
        "no_transitions"
    ],
    "acr_values_supported": [
        "urn:id.gov.au:tdif:acr:ip1:cl1",
        "urn:id.gov.au:tdif:acr:ip1:cl2",
        "urn:id.gov.au:tdif:acr:ip2:cl2",
        "urn:id.gov.au:tdif:acr:ip3:cl2"
    ],

    "frontchannel_logout_supported": true,

    "frontchannel_logout_session_supported": false

}
```

Figure 5 Normative example of the well-known configuration

1.8.1.5 PKCE

An authorisation server MUST support the PKCE extension to the authorisation code flow, including support for the S256 code exchange methods.

The authorisation server MUST NOT allow a client to use the plain code challenge method.

1.8.2 Response to Authorisation Requests

The authorisation code flow’s authorisation response MUST return the following fields in the response:

- `state`
    - The value of the state parameter passed in via the authentication request. This value MUST match exactly.
- `code`
    - The authorisation code, a random string issued by the OIDC provider to be used request to the Token Endpoint.

PKCE parameters MUST be associated with the “code” as defined in RFC 7636.

This response MUST be sent to the client via a HTTP redirect to the URI specified in the request.

1.8.2.1 Authentication Error Response

The Authentication Error Response is the message returned from an IXP’s Authorisation Endpoint in response to the Authorisation Request sent by the client.

If the End-User denies or cancels the request or the End-User authentication fails, the OIDC provider informs the TRP by using the error responses defined in either section 4.1.2.1 of RFC 6749 or the error codes defined in section 3.1.2.6 of the OpenID Connect Core 1.0. The additional authentication error responses defined by this Schedule are:

- `authentication_cancelled`
    - The End-User did not proceed with the authentication interaction.

1.8.2.2 Responding to Invalid Claims

A scope or claim is invalid if a IXP cannot source the underlying attributes from its ISPs or ASPs and from the IXP specific attributes outlined in Chapter 4 of Schedule 3 (AGDIS Attribute Profile).

If an IXP receives a request for a scope or claim it cannot fulfill, the IXP MUST ignore these scopes or claims.

An IXP MUST deny an authentication request with the `access_denied` error code (as described in RFC 6749) if its client requests a scope or claim that under the relevant attribute sharing policy it is not authorised to request. For example, if a client requests a restricted attribute set when they are not approved to an `access_denied` error code MUST be returned.

1.8.3 Token Response

All tokens issued by an IXP MUST be signed with the IXP's private key.

For clients using the Authorisation Code Grant type, access tokens MUST have a valid lifetime of no greater than one hour.

If an IXP issues refresh token they MUST have a lifespan of no longer than 24 hours.

Token lifespans MAY be shorter than these prescribed values to meet the requirements of an IXP's security risk assessment.

1.8.3.1 ID Token

An IXP MAY encrypt an ID Token or the ID Token's fields with the requesting clients public key when the ID Token contains attributes of the individual.

An IXP issued ID Token MUST:

(a) expire; and
(b) have a lifespan of no longer than 5 minutes.

An IXP SHOULD assign lifespans of shorter than 5 minutes to an ID Token.

An IXP issued ID Token MUST contain the following fields that are defined as REQUIRED.

An IXP issued ID Token MAY include the following fields that are defined as OPTIONAL.

ID token fields:

- `iss`
  - REQUIRED. The issuer field is the URL of the expected issuer.
- `aud`
  - REQUIRED. The audience field contains the client ID of the client.
- `sub`
  - REQUIRED. The identifier of the user. Note it MUST be a pairwise identifier and be unique to the client's sector.
- `acr`
  - REQUIRED. This is the level of assurance at which the user was authenticated at.
- `nonce`

  - If a nonce value was supplied with the authentication request, then this value is REQUIRED.
- `jti`
  - REQUIRED. A unique identifier for the token which can be used to prevent the reuse of the token.
- `exp`, `iat`, `nbf`
  - REQUIRED. The expiration, issued at, and not before timestamps for the tokens. They are dates presented as an integer representing the number of seconds since 1970-01-01T00:00:00Z UTC (Unix epoch) within acceptable ranges.

1.8.4 UserInfo Endpoint

An IXP MUST support returning claims via the UserInfo endpoint as prescribed by the requirements outlined in section 4.1.1 of Schedule 3 (AGDIS Attribute Profile).

When processing a UserInfo request, a IXP MUST only return the claims that are authorised within the authentication request associated with the presented access token.

An IXP MUST NOT return empty or null values for claims that cannot be fulfilled unless Chapter 2 and Chapter 3 of Schedule 3 (AGDIS Attribute Profile) specifically permits these values for a given attribute set.

The `sub` claim MUST always be present in the UserInfo response.

1.8.5 Request Objects

An IXP operating as OpenID Connect provider MUST accept requests containing a request object signed by the client's private key.

An IXP MUST validate the signature on request objects using the client's public key.

An IXP MUST implement support for receiving requests objects encrypted with one of its public keys.

1.8.6 Authentication Context

An IXP MUST provide an ACR in line with the claims outlined in section.1.3 of Schedule 3 (AGDIS Attribute Profile).

An IXP MUST return the `acr` attained by the individual during authentication at the ISP even if the `acr` was not marked as essential, the `acr_values` parameter was used, or no acr value was supplied in the TRP's authentication request.

1.9 Entity information

1.9.1 Claims supported

An authorisation server MUST return claims on best effort basis. An IXP asserting it can support specific claims does not guarantee it is available for all individuals.

An IXP MUST only return claims in accordance with data sharing requirements and data formats outlined in Chapter 4 of Schedule 3 (AGDIS Attribute Profile).

1.9.2 Scope profiles

An authorisation server MUST fulfill scopes on best effort basis.

An IXP MUST only return scopes in accordance with the data sharing policies and data formats outlined in Chapter 4 of Schedule 3 (AGDIS Attribute Profile).

1.9.3 Valid ACR Claims

Assurance levels are outlined in section 2.1.3 of Schedule 1 (AGDIS Onboarding Specifications).

An IXP MUST implement support to allow its relying parties to use either the `acr_values` or `acr` claim to request their required ACR.

An IXP MUST reject any request that include both the `acr_values` and `acr` claims.

When the `acr` claim is requested an IXP MUST support the `acr` claim being optionally marked as essential claim by the client. For example:

```
“claims”: {
    “id_token”: {
        “acr”: {
            “essential”: true,
            “values”: [“urn:id.gov.au:tdif:acr:ip2:cl3”]
        }
    }
}
```

Figure 6 Requesting authentication assurance level with claims

When the `acr` values are marked as an essential claim, the IXP MUST return a value that matches the requested values.

If the individual is unable to achieve the required level of assurance outlined in the request and the `acr` claim is marked as essential, then an IXP MUST respond with an authentication error.

If the `acr` claim is not marked as essential or no `acr` value was supplied in the authentication request, then an IXP MUST responds with the level of assurance the individual was able to achieve.

1.10 User consent

Given IXPs operate as central trusted entity in the AGDIS, IXPs MUST collect consent of the individual to whom the digital ID relates before redirecting the individual to client that originated the authentication request.

1.11 Privacy considerations

An IXP MUST adhere to all the attribute sharing policies set out in Chapter 2 and Chapter 3 of Schedule 3 (AGDIS Attribute Profile).

1.12 Security considerations

All clients of an IXP MUST conform to the applicable security considerations outlined in section 5 of RFC 6819.

2. Identity service provider

This chapter outlines the requirements for:

(a) IXPs in their role as a TRP to ISPs; and
(b) ISPs in their roles as OIDC providers to IXPs.

2.1 Client types

The resource owner password credential grant type as defined in RFC 6749 MUST NOT be used under this Schedule.

Given an IXP is only acting as a proxy, the Full Client with delegation is the only client available.

2.1.1 Full Client with User Delegation

This client type applies to clients that act on behalf of a particular resource owner and require delegation of that user's authority to access the protected resource. Furthermore, these clients can interact with a separate web browser application to facilitate the resource owner's interaction with the authentication endpoint of the authorisation server.

An ISP MUST only support clients of this type.

All clients of an ISP MUST use the authorisation code flow of RFC 6749 by sending the resource owner to the authorisation endpoint to obtain authorisation.

An ISP MUST ensure that the user authenticates to the authorisation endpoint.

The user's web browser is then redirected back to a URI hosted by the client service, from which the client can obtain an authorisation code passed as a query parameter. The client then presents that authorisation code along with its own credentials (`private_key_jwt`) to the authorisation server's Token Endpoint to obtain an access token.

An ISP MUST associate the clients with a unique public key as described in section 2.4 of this Schedule.

If an ISP issues a refresh token to this type of client, they MUST only do so if the security parameters of the request permit its issuance.

2.2 Client registration

An IXP MUST register with the authorisation server.

Each client IXP MUST receive a unique client identifier from the authorisation server.

Clients of the authorisation server MUST be statically configured.

An ISP MUST NOT support the dynamic registration of IXPs.

2.3 Redirect URI

An ISP MUST register an IXP’s full redirect URIs as required by the `authorization_code` grant type.

The authorisation server MUST validate the redirect URI passed to the authorisation end using strict string comparison.

An ISP MUST only permit TLS protected redirect URIs to be registered.

An ISP MUST NOT permit a IXP to have multiple redirect URIs on different domains.

An ISP MUST NOT forward values passed back to their redirect URIs to other arbitrary or user provided URIs.

An ISP MUST NOT permit open redirection.

2.4 Client keys

All connected IXPs acting as clients MUST have a public and private key that MUST be used when authenticating to the Token Endpoint.

As clients all IXPs MUST provide their public keys in their client registration metadata.

An ISP SHOULD support the use of `jwks` field, or registration of a `jwks_uri`.

If a IXP uses a `jwks_uri` to register its public key, the URI MUST be reachable by the authorisation server.

If an ISP supports the use of `jwks_uri:`

(a) the IXP SHOULD use a `jwks_uri` to simplify key rotation; and
(b) the ISP MUST validate the content of the clients registered `jwks_uri` document.

2.5 Grant types

An ISP MUST only support the `authorization_code` grant type when providing services to IXPs.

2.6 Technical Relying Party Profile

This section outlines the profile that an ISP MUST provide for the IXPs that are its TRPs.

2.6.1 Audit Logging

An IXP MUST log all interactions with its ISPs using a unique audit identifier it has generated for an authentication request from its TRP.

To enable a traceable audit trail for requests sent to an ISP, an IXP MUST implement a scheme to ensure that each request is uniquely identifiable at the ISP.

A recommended scheme is for an IXPs to transport a unique generated value using the `state` parameter. Note this unique value MUST NOT be the RP audit ID issued by an IXP to an PRP.

2.6.2 Request to the Authorisation Endpoint

An IXP making a request to the Authorisation Endpoint MUST use an unpredictable value for the `state` parameter with at least 128 bits of entropy.

An IXP MUST validate the `state` parameter upon return to the redirect URI.

An IXP MUST ensure that the `state` value is securely tied to the user's current IXP session.

An IXP MUST include their redirect URIs in the authorisation request.

An ISP MUST match the entire redirect URI using strict string comparison against registered values and reject request with missing or invalid redirect URIs.

The authentication request MUST contain the following REQUIRED parameters and MAY contain the following OPTIONAL parameters.

- `client_id`
  - REQUIRED. OAuth 2.0 Client Identifier assigned to a IXP by the ISP.
- `response_type`
  - REQUIRED. MUST be set to `code`.
- `scope`
  - REQUIRED. Indicates the attributes being requested. The `openid` scope MUST always be present.
- `redirect_uri`
  - REQUIRED. Indicates the valid endpoint where the client will receive the authorisation response. The URI MUST exactly match one of preregistered redirect URIs at the ISP.
- `state`
  - REQUIRED. An opaque value generated by the IXP used to protect against CSRF attacks. The value MUST contain sufficient entropy to avoid guessing and is returned to the IXP in the authentication response.
  - An IXP SHOULD use a unique value for each authorisation request.
- `nonce`
  - REQURIED. Un-guessable random string generated by the client, used to protect against cross site request forgery attacks. The string MUST contain sufficient entropy to avoid guessing. The value is returned to the IXP in the ID Token.
- `acr_values`
  - OPTIONAL. If the originating PRP requesting authentication has supplied `acr_values` a IXP MUST pass these values in accordance with protocol brokering requirements outlined in section 2.8.3 of this Schedule.
- `user_flow`
  - OPTIONAL. A string value that indicates the desired user flow for the user. Defined values are:
    - `sign_in`: An Exchange requests this when a TRP expects the user to already have a digital ID and sign in at the ISP.
    - `sign_up`: A IXP requests this when a TRP expects the individual to need to create a digital ID at an ISP
- `claims`
  - OPTIONAL. This parameter is used to request that specific Claims be returned. The value is a JSON object listing the requested Claims. This is made according to section 5.5 of OpenID Connect Core 1.0.

The values of the `claims`, `scope` and `acr_values` parameters are mapped from the original authentication request to the IXP from one of its TRPs. Additional OpenID Connect Core 1.0 parameters MAY also be mapped from the original request to the IXP that triggered the request from the IXP to the ISP. Mapping of these parameters are described Chapter 4 of this Schedule.

2.6.3 Request to the Token Endpoint

Request to the Token Endpoint require client authentication. The client authentication mechanism is a singed JWT and defined in the Identity Provider profile outlined in section 2.7 of this Schedule.

The claims that are included in the JWT are summarised below:

- `iss`
    - The client Id of the client creating the JWT.
- `sub`
    - The client Id of the client creating the JWT.
- `aud`
    - The URL of the authorisation server’s Token Endpoint.
- `iat`
    - The time that the token was created by the client.
- `exp`
    - The expiration time after which the token MUST be considered invalid.
- `jti`
    - A unique, random, identifier generated by the client for this authentication.

An IXP making the request MAY include additional claims in this set.

The following is an example of the required claims for a client authentication JWT as defined in this profile.

```
{
   "iss": "8428fea9-2815-82e9-a9f3-a32a9e9b723c",
   "sub": "8428fea9-2815-82e9-a9f3-a32a9e9b723c ",
   "aud": "https://idp.gov.au/token",
   "iat": 1516986988,
   "exp": 1516986929,
   "jti": "1516986988/c4da1075193e524df46b5e565c5a59568f34"
}
```

Figure 7 Token Endpoint private JWT example

The JWT assertion MUST be signed with the IXP’s private key from the key pair they have registered with the ISP.

The following claims MUST be included in a request to a Token Endpoint:

- `grant_type`
    - MUST be set to `authorization_code`.
- `code`
    - The value of the code parameter returned in the authorisation response.

- `redirect_uri`
    - value MUST be identical to the value of the `redirect_uri` parameters that was included in the authorisation request.
- `client_assertion_type`
    - MUST be set to `urn:ietf:params:oauth:client-assertion-type:jwt-bearer`.
- `client_assertion`
    - The value of the signed client authentication JWT generated as described below in the ID Tokens section (section 2.6.5 of this Schedule). The TRP MUST generate a new assertion JWT for each call to the Token Endpoint.

The following is an example of how these claims would be sent Token Endpoint:

```
POST /token HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Host: idp.gov.au

grant_type=authorization_code
&code=sedaFh
&redirect_uri=https%3A%2F%2Fidexchange.gov.au%2Foidc%2FloginResponse
&client_id=55f9f559-2496-49d4-b6c3-351a586b7484
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-
type%3Ajwt-bearer
&client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.ew0KICAgImlzcyI6ICI1NW
Y5ZjU1OS0yNDk2LTQ5ZDQtYjZjMy0zNTFhNTg2Yjc0ODQiLA0KICAgInN1YiI6ICI1NWY5ZjU1OS0
yNDk2LTQ5ZDQtYjZjMy0zNTFhNTg2Yjc0ODQiLA0KICAgImF1ZCI6ICJodHRwczovL2lkcC1wLmV4
YW1wbGUuY29tL3Rva2VuIiwNCiAgICJpYXQiOiAxNDE4Njk4Nzg4LA0KICAgImV4cCI6IDE0MTg2O
Tg4NDgsDQogICAianRpIjogIjE0MTg2OTg3ODgvMTA3YzRkYTUxOTRkZjQ2M2U1MmI1Njg2NWM1YW
YzNGU1NTk1Ig0KfQ.t_gX8JQGq3G2OEc2kUCQ8zVj7pqff87Sua5nktLIHj28l5onO5VpsL4sRHIG
Ovrpo7XO6jgtPWy3iLXv3-
NLyo1TWHbtErQEGpmf7nKiNxVCXlGYJXSDJB6shP3OfvdUc24urPJNUGBEDptIgT7Lhf6BbwQNlMQ
ubNeOPRFDqQoLWqe7UxuI06dKX3SEQRMqcxYSIAfP7CQZ4WLuKXb6oEbaqz6gL4l6p83G7wKGDeLE
TOTHsztZjKR38v4F_MnSrx8e0iIqgZwurW0RtetEWvynOCJXk-
p166T7qZR45xuCxgOotXY6O3et4n77GtgspMgOEKj3b_WpCiuNEwQ
```

Figure 8 UserInfo endpoint example request

2.6.4 Request to the UserInfo Endpoint

An ISP MUST implement support for an IXP to send a UserInfo request using either the HTTP GET or POST methods.

The Access Token obtained from authentication request MUST be sent as a Bearer Token, as described in RFC 6750.

2.6.5 ID Tokens

An IXP MUST validate the signature of an ID Token before accepting it by using the public key of the ISP that issued it.

An IXP MAY require an ISP to encrypt the ID Tokens it returns.

If an ISP supports encrypting ID Tokens, it MUST use the appropriate key of the requesting IXP.

An IXP MUST verify the following in received ID Tokens:

- `iss`
    - The issue field is the URL of the expected issuer.
- `aud`
    - The audience field contains the client ID of the IXP.
- `nonce`
    - String value used to associate the client session with the ID Token.
- `exp`, `iat`, `nbf`
    - The expiration, issued at and not before tokens are dates (integer number of seconds since 00:00:00Z 1st January 1970, i.e. Unix epoch) are within acceptable ranges.

2.6.6 Request Objects

An IXP MAY optionally send requests to the authorisation endpoint using the request parameter as defined in the OpenID Connect Core 1.0.

An IXP MUST sign the request object with its registered key.

An IXP MAY encrypt the request object using the ISP’s public key.

2.6.7 Discovery

An IXP MAY cache the ISP’s OIDC provider metadata once the ISP has been discovered and used by that IXP.

2.7 Identity Provider Profile

An ISP providing authentication services to an IXP using OpenID Connect Core 1.0 MUST implement this Schedule.

2.7.1 Audit Logging

An ISP MUST log all authentication requests and responses, including the values of the `client_id` and the `state` parameters associated with the request.

2.7.2 Connecting to clients

2.7.2.1 Grant types

The only supported grant type is `authorization_code`.

The authorisation code flow is the only authentication flow support under this Schedule. The authorisation code flow returns an Authorisation Code to the client that the client can then exchange for an ID Token and an Access Token. This provides the benefit of not exposing any tokens to the User Agent and potentially malicious applications with access to the User Agent.

The ISP MUST validate all redirect URIs for the `authorization_code` grant type.

2.7.2.2 Client authentication

An ISP MUST enforce client authentication for access to the authorisation server's Token Endpoint.

An ISP MUST only authenticate clients using the `private_jwt_key` method as prescribed in the OpenID Connect Core 1.0.

An ISP's authorisation server MUST NOT authenticate clients using any other method.

The JWT used to authenticate the client MUST expire and have a lifetime of no longer than 5 minutes.

An ISP's authorisation server MUST reject a JWT with an expiry time that has passed.

An ISP SHOULD:

(a) allow for clock skew between system when assessing the expiry of a JWT; and
(b) reject any JWT with expiry that is unreasonably far into the future.

The JWT MUST contain the following REQUIRED claims and MAY contain the following OPTIONAL claims:

- `iss`
  - REQUIRED. Issuer. This MUST contain the `client_id` of the client creating the token.
- `sub`
  - REQUIRED. Subject. This MUST contain the `client_id` of the client creating the token.
- `aud`
  - REQUIRED. Audience. The value that identifies the authorisation server as an intended audience. The authorisation server MUST verify that it is an intended audience for the token. The Audience MAY be the URL of the authorisation server's Token Endpoint.
- `jti`
  - REQUIRED. JWT ID. A unique identifier for the token generated by the client, which can be used to prevent reuse of the token. This identifier MUST contain at least 128 bits of entropy and MUST NOT be re-used by any subsequent authentication token.
- `exp`
  - REQUIRED. Expiration time on or after which the ID Token MUST NOT be accepted for processing.

- `iat`
  - OPTIONAL. Time at which the JWT was issued.

2.7.2.3 Dynamic registration

An ISP MUST NOT support dynamic registration of IXPs as clients.

2.7.2.4 Discovery

Endpoints and parameters specified in the Discovery document MAY be considered public information regardless of the existence of the discovery document.

An ISP MAY provide a well-known endpoint for its OpenID configuration as described in the OpenID Connect Discovery 1.0.

If an ISP provides a well-known endpoint, it MUST secure the well-known endpoint MUST secured as outlined in the OpenID Connect Discovery 1.0 specification. An IXP MAY apply additional security controls if required by their business need.

If an ISP publishes a discovery document, the document MUST as a minimum contain the following REQUIRED fields and MAY contain the OPTIONAL fields:

- `issuer`
  - REQUIRED. The fully qualified issuer URL of the server.
- `authorization_endpoint`
  - REQUIRED. The fully qualified URL of the server's authorisation endpoint defined by RFC6749.
- `token_endpoint`
  - REQUIRED. The fully qualified URL of the server's Token Endpoint defined by RFC6749.
- `introspection_endpoint`
  - OPTIONAL. The fully qualified URL of the server's introspection endpoint defined by RFC7662.
- `revocation_endpoint`
  - OPTIONAL. The fully qualified URL of the server's revocation endpoint as defined by RFC7009.
- `jwks_uri`
  - REQUIRED. The fully qualified URI of the server's public key in JWK Set format as defined in RFC7517.
- `scopes_supported`
  - REQUIRED. The list of scopes that MUST be made available by an ISP as defined in Schedule 3 (AGDIS Attribute Profile).
- `claims_supported`
  - REQUIRED. The list of claims that MUST be made available by an ISP as defined in Schedule 3 (AGDIS Attribute Profile).

2.7.3 Requests to the Authorisation Endpoint (Authentication Request)

An ISP MUST support all mechanisms for requesting a level of assurance as prescribed in section 2.8.3 of this Schedule.

2.7.4 User consent

As a central and trusted participant of the AGDIS all IXPs are responsible for collecting express consent from the individual in accordance with the attribute sharing policies outlined in Chapter 2 and Chapter 3 of Schedule 3 (AGDIS Attribute Profile).

An IXP MUST provide the mechanisms to capture express consent from the individual.

An ISP MAY gather consent from the individual to share attributes to the AGDIS but SHOULD take into consideration the end-to-end user experience impacts of additional consent gathering mechanisms.

2.7.5 Response to Authorisation Requests

The authorisation response to the authorisation code flow MUST return the following fields in the response:

- `state`
  - The value of the state parameter passed in via the authentication request. This value MUST match exactly.
- `code`
  - The authorisation code, a random string issued by the OIDC provider to be used request to the Token Endpoint.

The key requirements for these fields are described in section 4.1.2 of RFC 6749.

This response MUST be sent to the client via a HTTP redirect to the URI specified in the request.

2.7.5.1 Authentication Error Response

The Authentication Error Response is the message returned from an ISP's Authorisation Endpoint in response to the Authorisation Request sent by an IXP.

If the individual denies or cancels the request, or the individual fails to authenticate, the ISP MUST inform the IXP by using the error responses defined in either:

(a) section 4.1.2.1 of RFC 6749; or
(b) section 3.1.2.6 of OpenID Connect Core 1.0.

This profile defines an additional authentication error response:

- `authentication_cancelled`
  - The End-User did not proceed with the authentication interaction.

2.7.6 Token Response

A successful token response includes an access token, which can be used to make a UserInfo request, and as ID token (signed and optionally encrypted JWT) as per section 3.1.3.3 of the OpenID Connect Core 1.0.

An ISP MAY also include a Refresh Token in the Token Response.

2.7.6.1 ID Token

An ISP MAY encrypt ID Tokens using the appropriate key of the requesting IXP.

An ISP issued ID Token MUST expire with a lifespan of no longer than 5 minutes.

An ISP SHOULD assign lifespans shorter than 5 minutes to the ID Tokens it issues.

An ISP issued ID Token MUST contain the following fields that are defined as REQUIRED.

- `iss`
  - REQUIRED. The issuer field is the URL of the expected issuer.
- `aud`
  - REQUIRED. The audience field contains the client ID of the client.
- `sub`
  - REQUIRED. The identifier of the user. Note it MUST be a pairwise anonymous identifier and be unique to the client’s OpenID Sector.
- `acr`
  - REQUIRED. This is the level of assurance at which the user was authenticated at.
- `nonce`
  - If a nonce value was supplied with the authentication request, then this value is REQUIRED.
- `jti`
  - REQUIRED. A unique identifier for the token which can be used to prevent the reuse of the token.
- `exp`, `iat`, `nbf`
  - REQUIRED. The expiration, issued at, and not before timestamps for the tokens. They are dates presented as an integer representing the number of seconds since 1970-01-01T00:00:00Z UTC (Unix epoch) within acceptable ranges.

The following is an example of an ID token signed using the server’s RSA key.

```
eyJhbGciOiJSUzI1NiJ9.eyJhdXRoX3RpbWUiOjE0
MTg2OTg3ODIsImV4cCI6MTQxODY5OTQxMiwic3ViI
joiNldaUVBwblF4ViIsIm5vbmNlIjoiMTg4NjM3Yj
NhZjE0YSIsImF1ZCI6WyJjMWJjODRlNC00N2VlLTR
iNjQtYmI1Mi01Y2RhNmM4MWY3ODgiXSwiaXNzIjoi
aHR0cHM6XC9cL2lkcC1wLmV4YW1wbGUuY29tXC8iL
CJpYXQiOjE0MTg2OTg4MTJ9mQc0rtL56dnJ7_zO_f
x8-qObsQhXcn-qN-FC3JIDBuNmP8i11LRA_sgh_om
RRfQAUhZD5qTRPAKbLuCD451lf7ALAUwoGg8zAASI
5QNGXoBVVn7buxPd2SElbSnHxu0o8ZsUZZwNpircW
NUlYLje6APJf0kre9ztTj-5J1hRKFbbHodR2I1m5q
```

```
8zQR0ql-FoFlOfPhvfurXxCRGqP1xpvLLBUi0JAw3
F8hZt_i1RUYWMqLQZV4VU3eVNeIPAD38qD1fxTXGV
Ed2XDJpmlcxjrWxzJ8fGfJrbsiHCzmCjflhv34O22
zb0lJpC0d0VScqxXjNTa2-ULyCoehLcezmssg
```

Figure 9 Sample ID Token signature

Its claims are as follows:

```
{
    "auth_time": 1418698782,
    "exp": 1418699412,
    "sub": "6WZQPpnQxV",
    "nonce": "188637b3af14a",
    "aud": [
        "c1bc84e4-47ee-4b64-bb52-5cda6c81f788"
    ],
    "iss": "https://idp.gov.au/",
    "acr":"urn:id.gov.au:tdif:acr:ip3:cl2",
    "iat": 1418698812,
    "nbf": 1418698812
}
```

Figure 10 Claims used to the generate the prior signature

2.7.7 UserInfo Endpoint

An ISP MUST support returning claims via the UserInfo endpoint as prescribed by the requirements outlined in Schedule 3 (AGDIS Attribute Profile).

When processing a UserInfo request an ISP MUST only return the claims that are authorised within the authentication request associated with the presented access token.

An ISP MUST NOT return empty or null values for claims that cannot be fulfilled unless Schedule 3 (AGDIS Attribute Profile) specifically permits these values for a given attribute set.

The `sub` claim MUST always be present in the UserInfo response.

2.7.8 Request Object

An ISP MUST accept requests containing a request object signed by the requesting IXPs private key.

An ISP MUST validate the signature on request objects using the requesting IXP’s public key.

An ISP MUST implement support for receiving requests objects encrypted with one of its public keys.

2.7.9 Authentication context

An ISP MUST return the ACR value used for the authentication even if the `acr` claim was not marked as essential or the `acr_values` parameter was used.

2.8 Entity information

2.8.1 Claims supported

IXPs and ISPs MUST return claims on a best effort basis.

An ISP or its connected IXPs asserting it can provide a user claim does not imply that the data is available for all users.

An IXP MAY returns claims outside of the `claims_supported` list but MUST ensure that they do not violate the data sharing and privacy constraints prescribed under the Act.

2.8.2 Scope profiles

An ISP MUST implement the OpenID Connect support ISP specific scopes and claims outlined as in Schedule 3 (AGDIS Attribute Profile).

2.8.3 Valid ACR Claims

Assurance levels are outlined in Schedule 1 (AGDIS Onboarding Specifications).

An ISP MUST implement support to allow its relying parties to use either the `acr_values` or `acr` claim to request their required ACR.

An ISP MUST reject any request that include both the `acr_values` and `acr` claims.

When the `acr` claim is requested, an ISP MUST support the `acr` claim being optionally marked as essential claim by the client. For example:

```
“claims”: {
    “id_token”: {
         “acr”: {
             “essential”: true,
             “values”: [“urn:id.gov.au:tdif:acr:ip2:cl3”]
         }
    }
}
```

Figure 11 Sample assurance level requests using claims

When the `acr` values are marked as an essential claim, the ISP MUST return a value that matches the requested values.

If the individual is unable to achieve the required level of assurance outlined in the request and the `acr` claim is marked as essential, then an ISP MUST respond with an authentication error.

If the `acr` claim is not marked as essential or no `acr` value was supplied in the brokered authentication request, then an ISP MUST responds with the level of assurance the individual was able to achieve.

An IXP SHOULD ensure its clients understand their obligation to determine if a returned ACR meets the minimum requirement for the ACR that was requested.

2.9 Privacy Requirements

An ISP MUST adhere to all the ISP relevant attribute sharing policies set out in Chapter 2 and Chapter 3 of Schedule 3 (AGDIS Attribute Profile).

2.10 Security Considerations

An ISP MUST ensure all of clients conform to:

(a) section 10 (Security Considerations) of RFC 6749; and
(b) section 5 (Security Considerations) of RFC 6819.

3. Protocol brokering

An IXP MUST implement support to broker between the protocols used by its connected ISPs and PRPs.

The AGDIS only supports the OpenID Connect Core 1.0.

3.1 OIDC to OIDC brokering

When an IXP is accepting an authentication request from a TRP using OIDC, if the selected ISP implements this Schedule, an IXP MUST interact with the ISP using the ISP technical relying party profile as prescribed in this Schedule.

3.1.1 Mapping Claims and Scopes

If the `sub` claim is specified it MUST be handled as outlined in section 3.1.2 of this Schedule.

If the TRP's authentication request includes attributes that are fulfilled by itself or an ASP, an IXP SHOULD NOT forward theses attribute requests to the ISP unless it is necessary.

All other attributes included in the TRP's authentication request MUST be included in the IXPs authentication request to the ISP in accordance with the attribute sharing policies as prescribed in Chapter 4 of Schedule 3 (AGDIS Attribute Profile).

An IXP MAY expand the scopes defined in an authentication request into underlying claims when brokering an authentication request to its ISPs and ASPs.

Scopes and claims not outlined in Schedule 3 (AGDIS Attribute Profile) MUST be ignored by the IXP when brokering an authentication request. The IXP MUST NOT raise an error when scopes and claims are ignored.

3.1.2 Handling of Subject ID

An IXP MAY support the `sub` claim in authentication requests.

If an IXP implements support the `sub` claim in authentication requests:

(a) the IXP MUST resolve the pairwise identifier presented in the sub claim in the authentication request from the TRP to an existing pairwise identifier; and
(b) the IXP MAY return an error if the pairwise identifier for the user cannot be resolved to the target ISP.

3.1.3 Mapping assurance levels

If an IXP receives a single value for the `acr_values` or `acr` claim in an authentication request, the IXP MUST pass the set of ACR values that meet or exceed the requested value to the ISP in the brokered authentication request.

If the `acr` claim is marked as essential in the authentication request, an IXP MUST mark the `acr` claim as essential when sending the authentication request to the selected ISP.

An IXP MUST evaluate the ACR returned from the ISP and if the ACR meets or exceeds the originally requested value(s), return one of the originally requested values.

3.1.4 Prompt Parameter

An IXP MUST implement these processing rules to broker the OIDC `prompt` parameter.

Table 1 IXP prompt parameter brokering requirements

From Relying Party	To Identity Service Provider
`None`	`None`
`Login`	`Login`
`Consent`	Ignored. An IXP MUST implement consent for the release of attributes in accordance with the attribute sharing policies as defined in Schedule 3 (AGDIS Attribute Profile).

3.1.5 ID Token Hint Parameter

If an IXP has implemented support for the `id_token_hint` mechanism, the following processing rules apply:

(a) where the IXP receives an `id_token_hint` within an authentication request from a TRP, the IXP is REQUIRED to validate the Identity Token and extract the subject;

(b) the IXP MUST resolve the subject identifier at the ISP as per section 3.1.2 of this Schedule; and

(c) the IXP MUST include the resolved subject identifier when brokering the authentication request to the ISP using the `sub` claim as per section 5.5 of the OpenID Connect Core 1.0 and mark the `sub` claim as essential.

4. Attributes

Schedule 3 (AGDIS Attribute Profile) outlines the attributes and attribute sets that are available to PRPs.

4.1 Restricted attributes

Schedule 3 (AGDIS Attribute Profile) MAY define access restrictions for certain attributes and attribute sets.

An IXP MUST implement access control mechanism on a per client basis to ensure restricted attributes are only accessible to approved PRPs.

An ISP MUST implement access control mechanism on a per IXP basis to ensure restricted attributes are only accessible in accordance with access restrictions as defined by the Act.

4.2 OIDC Attribute Mapping

When an entity participating in the AGDIS is making or responding to a request using this Schedule, that entity MUST use the mapping of the attributes to the scopes and claims as defined in Schedule 3 (AGDIS Attribute Profile).

Schedule 3 – AGDIS Attribute Profile

[Draft Digital ID Data Standard v0.2]

List of Tables

Table of Figures

1. Operation of attributes

This chapter outlines how attributes operate within the context of the AGDIS and the components of an attribute sharing policy.

1.1 Attributes and attribute sets

Attributes, including those related to a transaction, a digital ID or the AGDIS consist of singular values or groups of values.

Singular values are attributes of an individual within the meaning of section 10 of the Act.

Groups of values are groups of attributes of an individual within the meaning of section 10 of the Act and referred to in the *Digital ID (AGDIS) Data Standards 2024* as an attribute set. The same value can be used as part of one or more attribute sets.

To support the objectives of the data minimisation principle, ASPs, ISPs and IXPs SHOULD, if possible, permit the elements of an attribute sets to be requested individually.

1.2 Attribute sharing policy

All attribute or attribute sets transmitted across the AGDIS MUST be subject to an attribute sharing policy.

All ASPs, ISPs and IXPs MUST comply with the attribute sharing policies in this Schedule that apply to them.

If an attribute or attribute set is not subject to an attribute sharing policy in this Schedule, ASPs, ISP and IXPs MUST NOT transmit that attribute or attribute set.

An attribute sharing policy MUST outline:

(a) the attribute or attribute set to which the policy is applied;
(b) the consent type applied to the attributes;
(c) the fulfillment requirements;
(d) the access policy; and
(e) the data representation.

1.2.1 Consent types

Consent types prescribe requirements for gathering of express consent from the individual by an accredited entity participating in the AGDIS.

The consent types outlined in this Schedule are found in Table 1 below.

An attribute sharing policy MUST include a consent type and assign the management of that consent to either or both:

(a) a specific entity (which is a named accredited entity such as myGov as an ASP); and
(b) a specific role (such as the ASP, ISP and IXP).

Table 1 Attribute profile consent models.

Consent type	Description
Not required	Express consent is not required for the attribute or attribute set. This consent type MUST only be applied to an attribute or attribute set when: • the attributes are explicitly exempt from the express consent requirements under the Act; • they are technical in nature and do not convey personal information on their own or when combined with other attributes; and • they are classified as identity system meta data.
Every use	Express consent is required every time the attribute or attribute set is shared. The consent MUST NOT be remembered or reused in subsequent requests for the attribute or attribute set.
Ongoing	Express consent is required at least the first time the attribute or attribute set is bound to the individual or shared. The consent MAY be remembered for a fixed duration as determined by the Act. The individual MUST: • be made aware of the use cases they are providing on-going consent to facilitate; • have the option for this consent to not be remembered; and • be provided with a mechanism to revoke consent if it was remembered. The individual SHOULD be notified: • if the attribute is an authorisation and it is revoked by a third party, for example, by the owner or creator of an authorisation; and • when their consent facilitated use of their attributes in the execution of automated use cases.

Consent type	Description
Every Change	Express consent for the attribute or attribute set is required the first time the attribute or attribute set is shared with the PRP, and every time it is modified. Accordingly, subsequent requests for express consent MUST occur when: • the attribute or attribute set has been modified; • the individual has revoked any remembered on-going consent; and • the duration of the remembered consent has expired. Attribute sharing policies SHOULD only apply this consent type when the underlying attributes support the detection of changes.

1.2.2 Fulfillment requirements

Not all attributes have guaranteed availability, and some may not be available for the IP level associated with the authenticated session.

The assertion of attributes in an individual's digital ID account is dependent on the documents available to them, the documents they chose to verify their identity, and any additional linkages an individual has with ASP managed attributes. Accordingly, the fulfillment of some attributes may not be practical however some attributes may still be required.

The necessity to meet an attributes request is referred to as a fulfillment requirement in this Schedule.

An attribute sharing policy MUST have a fulfillment policy to inform the behaviours of entities participating in the AGDIS when attribute requests cannot be fulfilled.

The two fulfillment requirements used in this Schedule are outlined in Table 2 below.

Table 2 Attribute fulfillment requirement classes.

Fulfillment requirement	Criteria and description
Best effort	The attribute or attribute set MUST be fulfilled if the individual has verified or asserted the requested attributes or attribute set and has provided express consent to share them. If the attributes request cannot be fulfilled or the individual has not provided express consent to share, an error SHOULD NOT be raised unless the attribute sharing policy requires it.
Required	The attribute or attribute set MUST be fulfilled. If the attribute or attribute set cannot be fulfilled or the individual did not provide express consent to share, an error MUST be raised.

1.2.3 Access policy

Attributes shared with PRPs via the AGDIS are either readily available to PRPs or require approval to access. The rules defining if or why a PRP can request attributes or attribute sets is referred to as an access policy in this Schedule.

An attribute sharing policy MUST outline the access policy that applies to its attributes or attribute sets.

An attribute sharing policy MAY extend the access policy for a given context provided the intent of the policy is not overridden.

The 4 access policies available under in this Schedule are outline in Table 3 below.

Table 3 Access policies that MUST be applied to attributes.

Policy	Details
Open	Any PRP MAY request an attribute or attribute set with this access control. If attributes are available, an ASP, ISP and IXP MUST fulfill requests for attributes subject to this access policy.
Not available	The attribute request MUST NOT be fulfilled. An ASP, ISP and IXP MAY respond with an error.

Policy	Details
Restricted	A PRP MUST be approved to request this attribute or attribute set. An IXP MUST only broker requests to the ASP, ISP and IXP that are permitted to fulfill the request.
Platform	A PRP MUST meet the platform specific requirements to have the attribute request fulfilled. An IXP MUST only broker these attribute requests to the ASP, ISP and IXP that can fulfill the requests. An IXP MAY respond with an error if the PRP is not a member of the attribute's related platform or approved by the platform to request the attribute or attribute set.

1.2.4 Data representation

An attribute sharing policy MUST be accompanied by a high-level specification of an attribute or attribute set's underlying values.

The specification MUST outline the implementation agnostic features of the attribute or attribute set including the data type and the range of acceptable values.

The default character encoding used for attributes being transmitted across the AGDIS is UTF-8.

If an attribute set for a given federation protocol uses a different character encoding the data representation MUST stipulate the character encoding used.

2. Core Attributes

This chapter outlines the attribute sharing policies for the Core Attributes available from the AGDIS.

2.1 Mutual attributes

An IXP MUST support brokering attribute requests for all mutual attributes.

An ISP MUST support processing attribute requests for all mutual attributes.

The available mutual attributes supported under this profile are outlined in Table 4 below.

Table 4 Mutal Attribute Sets

Attribute set	Attributes	Description
Core	Full Name Family Name Given Names Middle Names Preferred Name Date of Birth Core Attributes Last Updated	The core attributes that describe an individual – name and date of birth.
Validated Contact Details	Validated Email Validated Email Last Updated Validated Phone Number Validated Phone Number Last Updated	The validated email address and validated mobile phone number that is linked to a digital ID account at the ISP.
Verified Other Names	Verified Other Names Verified Other Names Last Updated	Other names an individual has verified during the initial or subsequent identity proofing processes at the ISP.
Verified Documents	Verified Documents	The verified attributes from the documents an individual used to prove their identity at an ISP during the initial or subsequent identity proofing processes. Access to this attribute set is restricted.

2.1.1 Core

Core Attributes are foundational elements used to identify an individual.

A summary of the attribute sharing policies, outlined here for Core Attributes, is provided in Table 5 below.

All Core Attributes are subject to an Open access policy as provided in Table 4 above.

An IXP MUST request express consent from the individual on every change to any Core Attribute.

An IXP MUST support brokering requests to its ISPs for all Core Attributes for requested IP Level of IP1 Plus or stronger.

An ISP MUST only fulfill attributes when the requested IP Level is IP1 Plus and the digital ID can attain IP1 Plus.

At IP1 Plus or stronger, an ISP MUST fulfill an attribute request for:

(a) Family Name;
(b) Date of Birth;
(c) Core Attributes Last Updated.

An ISP MUST return an error if it cannot fulfill a request for the above attributes.

At IP1 Plus or stronger, an ISP MUST fulfill requests for the following attributes if they are available:

(a) Given Names;
(b) Middle Names.

At IP1 Plus or stronger, an ISP MAY fulfill an attribute request for Full Name.

At an IP Level less than IP1 Plus, an ISP MUST NOT fulfill a request for Core Attributes other than Preferred Name.

An ISP MAY fulfill requests for Preferred Name at any IP Level. Noting that Preferred Name should be considered a self-asserted attribute as defined in section 2.3.1 of this Schedule.

An ISP MUST note every change made to the Core Attributes set by recording the date and time of the last change in the Core Attributes Last Updated attribute.

An IXP SHOULD use the Core Attributes Last Updated attribute to determine if the individual is required to provide consent before responding to the PRP's attribute request.

2.1.2 Validated Contact Details

Contact details like an email address and mobile phone number are sourced by the ISP during the identity proofing process or the individual's on-going use of their digital ID. Contact details are only referred to as validated due to ownership of an email address or phone number not being readily verifiable. A summary of the attribute sharing policies for these attributes is outlined in Table 6 below.

Validated Contact Details attributes are subject to an Open access policy.

An IXP MUST request express consent from the individual on every change to the Validated Contact Details attributes an ISP maintains.

An ISP SHOULD support one or both Validated Email Address and Validated Phone Number attributes.

If the individual has not validated a phone number or email address, an ISP MUST ignore a request for these attributes.

An IXP MUST support brokering attribute requests for Validated Contact Details.

An ISP MAY fulfill attribute requests for Validated Contact Details at all requestable assurance levels.

A Validated Email Address MUST be compliant with RFC 5322 and have a maximum length of 254 characters in compliance with RFC 2821.

An ISP MUST note every change made to the Validated Email Address attribute by recording the date and time of the last change in the Validated Email Address Last Updated attribute.

A Validated Phone Number MUST be compliant with ITU E.164.

An ISP MUST note every change made to the Validated Phone attribute by recording the date and time of the last change in the Validated Phone Last Updated attribute.

2.1.3 Verified Other Names

Additional names an individual is or was known by are verified by ISPs from the EoI documents presented during the initial or subsequent identity proofing process. A summary of the attribute sharing policy for Verified Other Names is outlined in Table 7 below.

The Verified Other Names attribute set is subject to an Open access policy (as defined above) at IP2 and stronger.

An IXP MUST NOT broker requests for Verified Other Names for IP levels weaker than IP2.

An IXP MAY ignore attribute requests or raise an error if the requested assurance level is less than IP2.

An IXP MUST request express consent from the individual on every change to the Verified Other Names attribute set.

An ISP MUST support attribute requests for Verified Other Names.

An ISP MAY only fulfill attribute requests for Verified Other Names when the requested IP level is IP2 or stronger and appropriate documents have been verified.

An IXP MUST support brokering attribute requests for Verified Other Names.

The Verified Other Name attribute set is represented by a set of verified other names, or a collection verified other name sets.

The attribute response MUST include a Verified Other Names set for each document the verified other names have been source from.

A Verified Other Names collection MUST NOT be empty.

Each Verified Other Name attribute MUST follow the rules outlined in Table 6 below for Family Name, Middle Names and Given Names.

If the individual has not verified any additional names an ISP MUST ignore a request for this attribute set.

An ISP MUST note every change made to the Verified Other Names attribute set by recording the date and time of the last change in the Verified Other Names Last Updated attribute.

Table 51 Core Attribute sharing policies

Attribute	Description	Access policy	IP Level	Fulfill-ment	Proven-ance	Consent	Data representation
Full Name	Individual's Full Name that MUST be equal to a space separated concatenation of the given names, middle names, and family name attributes – in that order.	Open	IP1 Plus and stronger	Required	Verified	Every change	MUST be a non-empty string with a maximum length determined by possible constituent values. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
Family Name	Individual's family name. If the individual only has one name verified it MUST be presented in this attribute.	Open	IP1 Plus and stronger	Required	Verified	Every change	MUST be a string non-empty string with a maximum length of 100 characters. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
Given Names	Individual's given names. There may be zero or more names stored in this attribute.	Open	IP1 Plus and stronger	Best effort	Verified	Every change	If available MUST be a string non-empty string with a maximum length of 100 characters. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.

Attribute	Description	Access policy	IP Level	Fulfill-ment	Proven-ance	Consent	Data representation
Middle Names	Individual's middle names. There may be zero or more names stored in this attribute.	Open	IP1 Plus and stronger	Best effort	Verified	Every change	If available MUST be a string non-empty string with a maximum length of 100 characters. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
Preferred Name	Individual's preferred name. This is an attribute the individual MAY self-assert to indicate the name they prefer to be known by at the PRP.	Open	IP1 and stronger	Best effort	Self-asserted	Every change	If available MUST be a string non-empty string with a maximum length of 100 characters. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
Date of Birth	Individual's date of birth.	Open	IP1 Plus and stronger	Required	Verified	Every change	RFC 3339 compliant time and date string of the form YYYY, YYYY-MM, or YYYY-MM-DD.
Core Attributes Last Updated	Date and time of when the Core Attributes for this digital identity were updated.	Open	IP1 and stronger	Required	ISP managed	Every change	A date and time representation specified in the attribute profile for the implementer's federation protocol.

Table 6 Validated contact details attribute sharing policies

Attribute	Description	Access policy	IP Level	Fulfill-ment	Proven-ance	Consent	Data representation
Validated Email	The email address an individual has validated at the ISP.	Open	IP1 and stronger	Best effort	Validated	Every change	Email address conforming to RFC 5322 address syntax. With a maximum length of 254 characters in compliance with RFC 2821.
Validated Email Last Updated	The date and time at which the individual last validated the email address.	Open	IP1 and stronger	Best effort	Validated	Every change	A date and time representation specified in the attribute profile of each federation protocol.
Validated Phone Number	The mobile phone number an individual has validated at the ISP.	Open	IP1 and stronger	Best effort	Validated	Every change	A mobile number in E.164 format.
Validated Phone Number Updated	The date and time at which the individual last validated the mobile phone number.	Open	IP1 and stronger	Best effort	Validated	Every change	A date and time representation specified in the attribute profile for the implementer's federation protocol.

Table 7 Verified other names attribute sharing policies

Attribute	Description	Access policy	IP level	Fulfill-ment	Proven-ance	Consent	Data representation
Verified Other Names	A collection of Family Name, Middle Names and Given Names for each of the person's other verified names. The requirements Family Name, Middle Names and Given Names attributes are as specified in the Core Attributes set.	Open	IP2 and stronger	Best effort	Verified	Every change	A collection of verified other name sets. Each verified other name set MUST contain labelled elements for at least one of Family Name, Middle Names and Given Names. The data representation for each name MUST be compliant with the relevant data representation outlined for the Core Attribute set.
Verified Other Names Last Updated	Individual's family name. If the individual only has single it MUST be transmitted in this claim.	Open	IP2 and stronger	Best effort	ISP managed	Every change	A date and time representation determined aligned to data types used by the federation protocol.

Table 8 Verified documents attribute sharing policy

Attribute	Description	Access policy	IP level	Fulfillment	Provenance	Consent	Data representation
Verified Documents	Collection of verified documents including document metadata, document identifiers, document names, date of birth, and additional attributes specific to a document type.	Restricted	IP1 Plus and stronger	Required	Verified	Every use	A collection of distinct verified document types where each element of the collection is conformant to the schema for its document type.

2.1.4 Verified Documents

The verified fields from the documents an individual submitted during the identity proofing process at their chosen ISP are available as the Verified Documents attribute set. The attributes in the Verified Documents attribute sets are sourced from:

(a) Commencement of Identity Documents;
(b) Linking Documents;
(c) Use in the Community Documents;
(d) Photo ID Documents.

Not all fields from these documents can be readily verified, hence only a subset of a document's fields may be available. A summary of the attribute sharing policy for verified documents is presented in Table 8 above.

An IXP MUST request express consent from the individual on every use of the Verified Documents attribute set.

An IXP MUST NOT broker requests for Verified Documents for assurance levels weaker than IP1 Plus.

An IXP MUST support brokering of attribute requests for a single or multiple Verified Documents.

An IXP MUST only broker attribute requests for Verified Documents from approved PRPs.

An ISP MUST support attribute request for single or multiple Verified Documents.

An ISP MUST only make available attributes for the most recent instances of each Verified Document type.

When processing generic requests for Verified Documents, an ISP MUST prepare a response that conforms to the IP Level based rules outlined in Table 10 below.

When processing requests for specific Verified Documents, an ISP MUST only respond to request for valid document types and for documents the individual has verified. Valid document types are outlined in Table 11 below.

The data representation for a Verified Document is set of named simple and complex attributes. Each set representing an instance of a Verified Document MUST have the following members:

(a) Document Type Code with a value that MUST be one of the URNs outlined in Table 11 below;
(b) Document Verification Method that MUST be one of the enumerated values in Table 12 below;
(c) Document Verification Date MUST be valid a date and time value;
(d) Document Date of Birth MUST be a valid date extracted from the document;
(e) Document Identifiers is a collection that MUST contain one or more document identifiers outlined in the document type's schema;
(f) Document Names is a collection that MUST contain the required document names for the document type's schema;
(g) Document Attributes is a collection that MUST contain the required document attributes outlined in the document type's schema.

A more detailed overview of the fields is provided in Table 9 below.

An ISP’s attribute response for each document MUST be conformant to the schema for that document’s type. Schemas for each document type make use of the enumerations outlined in the following tables and the schemas for each document type are outlined in the following subsections.

Table 9 Verified document structure

Field	Description	Data representation
Document Type Code	A URN representing the type of document. To be a valid the URN MUST be specified in this profile.	The valid URNs are enumerated in Table 11 below.
Document Verification Method	The verification method that was used to verify the document during the identity proofing process. The Verification Method in the response MUST match one the values outlined in this profile.	A single character string enumerating the document verification methods. The enumeration is outlined in Table 12 below of these values for the AGDIS.
Document Verification Date	The date and time the document was verified.	A string representing the RFC 3339 date time coordinated to coordinated universal time.
Document Names	A collection of key-values pairs representing the names, by-parts or full, extracted from the document and verified during the identity proofing process. All name names values marked as required in the document type’s schema MUST be present.	The keys denoting entries in the set MUST be non-empty strings with values matching the enumerated name values for the document type. The names values MUST be non-empty strings compliant with constraints outlined for the document type.
Document Date of Birth	The date of birth as submitted and verified during the identity proofing process.	An RFC 3339 date string of the format YYYY-MM-DD.

Field	Description	Data representation
Document Identifiers	A collection of identifiers extracted from the document and verified during the identity proofing process. The elements of the collection MUST each have a type and value field. The identifier attributes marked as required in document type’s schema type MUST be present.	The type field MUST be a string with a value that matches one of the enumerated document identifier attributes for the given document type. The data types of the value field MUST match the data type defined in the schema of the document type.
Document Attributes	A collection of attributes extracted from the document and verified during the identity proofing process. The elements of the collection MUST each have a type and value field. The attributes marked as required in the document type’s schema MUST be present.	The type field MUST be a string with a value that matches one of the enumerated document attributes for the given document type. The data types of the value field MUST match the data type defined in the document types schema for a given document attribute.

Table 10 Default Verified Document attribute responses rules

IP level	Default attribute response
IP1	Nil response.
IP1 Plus	The attribute response MUST only contain the most recently verified documents that satisfy the IP1 Plus requirements.
IP 2	The attribute response MUST be the 2 most recently verified documents that satisfy IP2 requirements. The documents MUST NOT be the of the same type.
IP2 Plus	The 2 most recently verified documents used to satisfy IP2 Plus requirements. One of the returned verified documents MUST be the document used to meet the biometric binding requirements of IP2 Plus. The documents MUST NOT be of the same type.
IP3	If a CoI credential was used for biometric binding and identity verification, then the 2 most recently verified documents MUST be returned. Otherwise, the 3 most recently verified documents used to satisfy the IP3 requirements MUST be returned. One of the returned verified documents MUST be the document used to meet the biometric binding requirements of IP3. The documents MUST NOT be of the same type.
IP4	The 4 most recently verified documents used to satisfy the IP4 requirements. One of the returned verified documents MUST be the document used to meet the biometric binding requirements of IP4. The documents MUST NOT be of the same type.

Table 11 Document Type Code URNs

Document Type	Verification authority	Verification authority document type code	AGDIS document type code URN
Birth Certificate	DVS	BC	urn:id.gov.au:tdif:doc:type_code:BC
Change of Name Certificate	DVS	NC	urn:id.gov.au:tdif:doc:type_code:NC
Marriage Certificate	DVS	MC	urn:id.gov.au:tdif:doc:type_code:MC
Citizenship Certificate	DVS	CC	urn:id.gov.au:tdif:doc:type_code:CC
Registration by Descent	DVS	RD	urn:id.gov.au:tdif:doc:type_code:RD
ImmiCard	DVS	IM	urn:id.gov.au:tdif:doc:type_code:IM
Visa	DVS	VI	urn:id.gov.au:tdif:doc:type_code:VI
Australian Driver License	DVS	DL	urn:id.gov.au:tdif:doc:type_code:DL
Medicare Card	DVS	MD	urn:id.gov.au:tdif:doc:type_code:MD
Australian Travel Document	DVS	PP	urn:id.gov.au:tdif:doc:type_code:PP
Centrelink Concession Card	DVS	CO	urn:id.gov.au:tdif:doc:type_code:CO

Table 12 Document verification method enumeration

Document verification method	Enumeration
Technical Verification	T
Source Verification	S
Visual Verification	V

Table 13 State and Territory name enumeration

State/Territory	Enumeration
Australian Capital Territory	`ACT`
New South Wales	`NSW`
Northern Territory	`NT`
Queensland	`QLD`
South Australian	`SA`
Tasmania	`TAS`
Victoria	`VIC`
Western Australia	`WA`

Table 14 Medicare Card type enumeration

Medicare Card type	Enumeration
Blue	`B`
Green	`G`
Yellow	`Y`

Table 15 Medicare Card type expiry patterns

Medicare Card Type	Expiry Pattern	DVS Pattern
Blue	`YYYY-MM-DD`	`^[0-9]{4}\-[0-9]{2}\-[0-9]{2}`
Green	`YYYY-MM`	`^[0-9]{4}\-[0-9]{2}`
Yellow	`YYYY-MM-DD`	`^[0-9]{4}\-[0-9]{2}\-[0-9]{2}`

Table 16 Travel Document gender enumeration

Gender	Enumeration
Male	`M`
Female	`F`
Other	`X`

Table 17 Centrelink Concession Card enumeration

Card Type	Enumeration
Health Care Card	`HCC`
Pensioner Concession Card	`PCC`
Commonwealth Seniors Health Card	`SHC`

2.1.4.1 Verified Birth Certificate schema

Table 18 Verified Birth Certificate schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be "`urn:id.gov.au:tdif:doc:type_code:BC`".
Document Verification Method	-	Y	String	Only source verification is permitted for Birth Certificates. See Table 12 above for accepted values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Family Name Given Name	Y	String	For NSW, VIC or WA issued Birth Certificates: • Given Name MAY be left blank if Family Name has a value; • Family Name MAY be left blank if Given Name has a value. All other state and territory BDM issued certificates the string MUST be non-empty and have a maximum length of 80 characters. Values MUST only be composed of alpha characters, apostrophes, hyphens, and spaces.

Section	Field Name(s)	Required	Type	Constraints
Document Date of Birth	Date of Birth	Y	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY, YYYY-MM, or YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Identifiers	Registration Number	N	String	If available MUST be a string of numeric values with a maximum length of 10 characters. Only available for certificates issued by BDM registries from NSW, WA, TAS, NT, ACT and VIC.
	Certificate Number	N	String	If available MUST be a string of numeric values with a maximum length of 12 characters. Only available for certificates issued by BDM registries from NSW, TAS, ACT, NT, SA, or VIC.
	Registration Date Registration Year	N	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY or YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Attributes	Registration State	Y	Enumeration	MUST use the abbreviations outlined in Table 13 above.

2.1.4.2 Verified change of name schema

Table 19 Verified Change of Name Certificate schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be “`urn:id.gov.au:tdif:doc:type_code:NC`”.
Document Verification Method	-	Y	String	Only source verification is permitted for Change of Name Certificates. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Family Name Given Name	Y	String	For NSW, VIC or WA issued Birth Certificates: • Given Name MAY be left blank if Family Name has a value; • Family Name MAY be left blank if Given Name has a value. All other state and territory BDM issued certificates the string MUST be non-empty and have a maximum length of 80 characters. Values MUST only be composed of alpha characters, apostrophes, hyphens, and spaces.
Document Date of Birth	Date of Birth	N	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY or YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.

Section	Field Name(s)	Required	Type	Constraints
Document Identifiers	Registration Number	N	String	If available MUST be a string of numeric values with a maximum length of 10 characters. Only available for certificates issued by BDM registries from NSW, WA, TAS, NT, ACT and VIC.
	Certificate Number	N	String	If available, the value MUST be a string of numeric values with a maximum length of 12 characters. Only available for certificates issued by BDM registries from NSW, TAS, ACT, NT, SA, or VIC.
	Registration Date Registration Year	N	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY or YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Attributes	Registration State	Y	Enumeration	MUST use the abbreviations outlined in Table 13 above.

2.1.4.3 Verified Marriage Certificate schema

Table 20 Verified Marriage Certificate schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be “`urn:id.gov.au:tdif:doc:type_code:MC`”.
Document Verification Method	-	Y	String	Only source verification is permitted for Marriage Certificates. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Family Name Given Name Family Name 2 Given Name 2	Y	String	For NSW, VIC or WA issued Birth Certificates: • Given Name MAY be left blank if Family Name has a value; • Family Name MAY be left blank if Given Name has a value. All other state and territory BDM issued certificates the string MUST be non-empty and have a maximum length of 80 characters. Values MUST only be composed of alpha characters, apostrophes, hyphens, and spaces.
Document Date of Birth	Date of Birth	N	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY or YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.

Section	Field Name(s)	Required	Type	Constraints
Document Identifiers	Registration Number	N	String	String of numeric values with a maximum length of 10 characters. Only available for Birth Certificates issued by BDM registries from NSW, WA, TAS, NT, ACT and VIC.
	Certificate Number	N	String	If available, the value MUST be a string of numeric values with a maximum length of 12 characters. Only available for certificates issued by BDM registries from NSW, TAS, ACT, NT, SA, or VIC.
	Registration Date Registration Year	N	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY or YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000
Document Attributes	Registration State	Y	Enumeration	MUST use the abbreviations outlined in Table 13 above.

2.1.4.4 Verified Citizenship Certificate schema

Table 21 Verified Citizenship Certificate and Registration by Descent

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be "`urn:id.gov.au:tdif:doc:type_code:RD`".
Document Verification Method	-	Y	String	Only source verification is permitted for Certificates of Registration by Descent. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Family Name	Y	String	MUST only be comprised of alpha characters (including spaces, hyphens, and apostrophes) and MUST be a non-empty string with a maximum length of 100 characters. MUST only be composed of alpha characters, apostrophes, hyphens, and spaces.
	Given Name	N	String	If provided MUST only be comprised of alpha characters (including spaces, hyphens, and apostrophes) and MUST be a non-empty string with a maximum length of 100 characters. MUST only be composed of alpha characters, apostrophes, hyphens, and spaces.

Section	Field Name(s)	Required	Type	Constraints
Document Date of Birth	Date of Birth	Y	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY or YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Identifiers	Stock Number	Y	String	MUST be a non-empty string with a maximum length of 11 characters.
Document Attributes	Acquisition date	Y	String	MUST be a RFC 3339 formatted date value of the format YYYY-MM-DD.

2.1.4.5 Verified Immigration Card schema

Table 22 Verified ImmiCard schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be "`urn:id.gov.au:tdif:doc:type_code:IM`".
Document Verification Method	-	Y	String	Only source verification is permitted for ImmiCards. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Family Name Given Name	Y	String	Values MUST be a non-empty string with a maximum of 49 characters in length. Values MUST only be composed of alpha characters, apostrophes, hyphens, and spaces.
Document Date of Birth	Date of Birth	Y	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Identifiers	ImmiCard Number	Y	String	MUST be a string of characters in length 9. The string MUST be composed of 3 leading alpha characters, followed by 6 numeric characters, e.g., ABC123456.

Section	Field Name(s)	Required	Type	Constraints
Document Attributes	-	-	-	No attributes for the document.

2.1.4.6 Verified Visa Request schema

Table 23 Verified Visa Request schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be "`urn:id.gov.au:tdif:doc:type_code:VI`".
Document Verification Method	-	Y	String	Only source verification is permitted for Visa Requests. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Family Name Given Name	Y	String	MUST be a non-empty string with a maximum length of 49 characters. MUST only be composed of alpha characters, apostrophes, hyphens, and spaces.
Document Date of Birth	Date of Birth	Y	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Identifiers	Passport Number	Y	String	MUST be non-empty string with a length of 14 upper and lower case alphanumeric characters, in any order.

Section	Field Name(s)	Required	Type	Constraints
Document Attributes	Country of Issue	Y	String	The issuing country of the associated the passport linked to this Visa. The value MUST be a non-empty string with a maximum length of 14 characters.

2.1.4.7 Verified Driver License schema

Table 24 Verified Driver License schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be "`urn:id.gov.au:tdif:doc:type_code:DL`".
Document Verification Method	-	Y	String	Only source verification is permitted for Driver Licenses. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Family Name	Y	String	MUST be a non-empty string and only be comprised of alpha characters (including spaces, hyphens, and apostrophes), and have a maximum length of 40 characters.
	Given Name	Y	String	MUST be a non-empty string and only be comprised of alpha characters (including spaces, hyphens, and apostrophes), and have a maximum length of 20 characters.
	Middle Name	N	String	If available it MUST be a non-empty string and only be comprised of alpha characters (including spaces, hyphens, and apostrophes), and have a maximum length of 20 characters.

Section	Field Name(s)	Required	Type	Constraints
Document Date of Birth	Date of Birth	Y	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Identifiers	License Number	Y	String	MUST be a non-empty alphanumeric string to with maximum length of 10 characters.
	Card Number	N	String	If available, MUST be a non-empty alphanumeric string to with maximum length of 10 characters.
Document Attributes	State of Issue	Y	Enumeration	Follow the enumeration outline in Table 13 above.

2.1.4.8 Verified Medicare Card schema

Table 25 Verified Medicare Card schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be "`urn:id.gov.au:tdif:doc:type_code:MD`".
Document Verification Method	-	Y	String	Only source verification is permitted for Medicare Cards. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Full Name 1	Y	String	Name line 1 from the card. MUST be a non-empty string with a maximum length of 27 characters. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
	Full Name 2	N	String	Name line 2 from the card. If available, MUST be a non-empty string with a maximum length of 25 characters. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.

Section	Field Name(s)	Required	Type	Constraints
	Full Name 3	N	String	Name line 3 from the card. If available, MUST be a non-empty string with a maximum length of 23 characters. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
	Full Name 4	N	Sting	Name line 4 from the card. If available, MUST be a non-empty string with a maximum length of 21 characters. It MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
Document Date of Birth	Date of Birth	Y	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Identifiers	Card Number	Y	String	MUST be a string of 10 numerical characters.
	Individual Ref Number	Y	Number	MUST be an integer value between 1 and 9.
Document Attributes	Card Expiry	Y	Date	MUST meet the data format for the Card Type. See Table 16 for expiry date formats date.
	Card Type	Y	Enumeration	MUST be one of the single character values outlined in Table 14 above.

2.1.4.9 Verified Passport schema

Table 26 Verified Passport schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be "`urn:id.gov.au:tdif:doc:type_code:PP`".
Document Verification Method	-	Y	String	Only source verification is permitted for Passports and Australian Travel Documents. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be a RFC 3339 formatted date time value in coordinated universal time.
Document Names	Family Name	Y	String	MUST be a non-empty string with a maximum length of 31 characters. MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
	Given Name	N	String	If available, MUST be a non-empty string with a maximum length of 31 characters. MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
Document Date of Birth	Date of Birth	Y	String	MUST be an RFC 3339 formatted date value of the format YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.

Section	Field Name(s)	Required	Type	Constraints
Document Identifiers	Travel Document Number	Y	String	A string of 10 numerical characters.
Document Attributes	Gender	Y	Enumeration	MUST use the abbreviated Gender enumeration in Table 16 above.

2.1.4.10 Verified Centrelink Concession Card schema

Table 27 Verified Centrelink Concession Card schema

Section	Field Name(s)	Required	Type	Constraints
Document Type Code	-	Y	URN	Value MUST be “`urn:id.gov.au:tdif:doc:type_code:CO`”.
Document Verification Method	-	Y	String	Only source verification is permitted for Centrelink Concession Cards. See Table 12 above for legitimate values.
Document Verification Date	-	Y	String	MUST be an RFC 3339 formatted date time value in coordinated universal time.
Document Names	Name	Y	String	MUST be a non-empty string with a maximum length of 32 characters. MUST only be composed of alpha characters, hyphens, apostrophes, and spaces.
Document Date of Birth	Date of Birth	N	String	MUST be an RFC 3339 formatted date value or year value of the format YYYY, YYYY-MM, or YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
Document Identifiers	CRN	Y	String	MUST be a string of 10 characters. The first 9 characters MUST be numerical values and the last character MUST be an alpha character, e.g., 123456789A

Section	Field Name(s)	Required	Type	Constraints
Document Attributes	Card Expiry	Y	Date	MUST be an RFC 3339 formatted date value of the format YYYY-MM-DD. Where the year value MUST be in the range 1753 to 3000.
	Card Type	Y	Enumeration	MUST be one of the values enumerated in Table 17 above.

2.2 Identity System Metadata

Identity System Metadata attributes defined in this section are values that do not convey any personal information about an individual but facilitate core AGDIS functionality.

Unless explicitly stated, all Identity System Metadata attributes:

(a) do not require express consent to be requested from the individual when sharing;
(b) are subject to an Open access policy; and
(c) MAY be fulfilled at all assurance levels.

A summary of the sharing policies outlined for these attributes can be found in Table 31 below.

Table 28 Identity System Metadata

Metadata Set	Attributes	Description
Common	Digital ID user identifier Authentication Time Identity Proofing Level Authentication Level Authentication Method Last Updated	Common attributes that support the use of a digital ID across the AGDIS. Implementers should review attribute sharing policy for each attribute to determine if they need to support its generation, management, or use.
Audit	RP Audit ID	An identifier unique to every logical interaction between an IXP and a PRP. The RP Audit identifier MUST not be shared with ISPs.

2.2.1 Common

2.2.1.1 Digital ID Identifier

The Digital ID Identifier is a unique identifier for an individual.

The Digital ID Identifier:

(a) For PRPs, is the pairwise identifier assigned to an individual by an IXP; and
(b) for IXPs, the GPI or pairwise identifier assigned to an individual by an ISP.

The generation and assignment of these unique identifiers MUST be compliant with the relevant requirements outlined in section 1.2.1 and section 1.2.2 of Schedule 1 (AGDIS Onboarding Specifications).

The Digital ID identifiers MUST be compliant with data representation prescribed in section 2.4 of Schedule 1 (AGDIS Onboarding Specifications).

2.2.1.2 Authentication Time

Authentication Time is the date and time when the individual successfully authenticated at the ISP.

An ISP MUST support this metadata value.

An IXP MUST support brokering requests for this metadata value.

The data representation for the date-time value is dependent upon the federation protocols being brokered. An IXP MAY need to translate values between types when brokering between different federation protocols in a transaction.

2.2.1.3 Assurance level

The assurance level of the authenticated digital ID outlines the IP level and authentication level attained during authentication at the ISP.

An IXP MUST broker requests for this value as requested from a PRP to the ISPs in accordance with the requirements in section 2.1.9 of Schedule 1 (AGDIS Onboarding Specifications) and requirements outlined by the federation protocols being used.

An ISP MUST process and respond to request for the value in accordance with the requirements in section 2.1.3 of Schedule 1 (AGDIS Onboarding Specifications) and requirements outlined in the relevant federation protocols.

The data representation for this value is specified in Schedule 1 (AGDIS Onboarding Specifications) and illustrated in Table 29 below.

2.2.1.4 Authentication Method

The Authentication Method value communicates to the PRP which ISP the individual to authenticate themselves. The Authentication Method is an IXP managed value.

An IXP MUST provide the Authentication Method based the ISP the individual has successfully authenticated themselves.

An IXP MAY cache the attribute to support subsequent SSO requests.

An IXP MUST NOT return any information about the Authentication Method if the authentication fails or is cancelled by the individual.

The data representation for the Authentication Method is a URN presented in Table 30 below.

2.2.1.5 Last Updated

Last Updated is the date and time that any of the individual's attributes were last updated.

An IXP MUST support brokering attribute request for this value.

An ISP MUST implement support for this value.

The data representation for this attribute is dependent upon the federation protocols in use.

2.2.2 Audit

2.2.2.1 RP audit Identifier

The RP audit ID requirements are outlined in section 2.1.2 of Schedule 1 (AGDIS Onboarding Specifications).

The RP audit ID is an IXP specific value that ISPs do not need to support.

An RP audit ID is a unique audit identifier generated for every logical interaction between a PRP and an IXP to enable an audit trail.

An IXP MUST NOT share the RP audit ID it has generated for a PRP with ISPs.

To ensure uniqueness of the RP audit ID within the AGDIS a UUID is used. The UUID generation MUST conform to RFC 4122.

Table 29 Authentication assurance levels

IP level	Authenticator level	URN
IP1	AL1	`urn:id.gov.au:tdif:acr:ip1:cl1`
	AL2	`urn:id.gov.au:tdif:acr:ip1:cl2`
	AL3	`urn:id.gov.au:tdif:acr:ip1:cl3`
IP1 PLUS	AL1	`urn:id.gov.au:tdif:acr:ip1p:cl1`
	AL2	`urn:id.gov.au:tdif:acr:ip1p:cl2`
	AL3	`urn:id.gov.au:tdif:acr:ip1p:cl3`
IP2	AL2	`urn:id.gov.au:tdif:acr:ip2:cl2`
	AL3	`urn:id.gov.au:tdif:acr:ip2:cl3`
IP2 PLUS	AL2	`urn:id.gov.au:tdif:acr:ip2p:cl2`
	AL3	`urn:id.gov.au:tdif:acr:ip2p:cl3`
IP3	AL2	`urn:id.gov.au:tdif:acr:ip3:cl2`
	AL3	`urn:id.gov.au:tdif:acr:ip2p:cl2`
IP4	AL3	`urn:id.gov.au:tdif:acr:ip4:cl3`

Table 30 Authentication Method URNs

ISP	Abbreviation	URN
myGovID	myGovID	`urn:id.gov.au:idp:mygovid`
-	-	-

Table 31 Identity System Metadata attributes

Attribute	Description	Access policy	IP level	Fulfill	Provenance	Consent Type	Data representation
Digital Pairwise Identifier	The Digital ID Identifier is a unique identifier for an individual. The Digital ID Identifier: (a) For PRPs, is the pairwise identifier assigned by an IXP; and (b) for IXPs, the GPI or pairwise identifier assigned by an ISP.	Open	IP1 and stronger	Require d	Provider managed	Not required	Regardless of the federation protocol used this attribute MUST be generate pairwise identifiers in accordance with algorithm outlined in section 8.1 of the OpenID Connect Core 1.0 specification.
Authentication Time	The date and time when the individual successfully authenticated at the ISP.	Open	IP1 and stronger	Require d	ISP managed	Not required	A date and time representation aligned to data types used by the federation protocol.
Assurance Level	The assurance level of the authenticated Digital ID outlines the IP level and authentication level attained during authentication at the ISP.	Open	IP1 and stronger	Require d	ISP managed	Not required	The assurance level MUST be one of the assurance level URNs outlined in Table 29 above.
RP Audit Identifier	The RP Audit Identifier requirements are outlined in Schedule 1 (AGDIS Onboarding	Open	IP1 and stronger	Require d	IXP managed	Not required	To ensure uniqueness the RP Audit Identifiers within the AGDIS a UUID is used. The

Attribute	Description	Access policy	IP level	Fulfill	Provenance	Consent Type	Data representation
	Specifications).						UUID generation MUST conform to RFC4122.

2.3 Assumed Self-Asserted Attributes

Assumed Self-Asserted Attributes are attributes provided by an individual that can assist with service delivery and enhance the overall user experience for example by prefilling online forms.

An ISP SHOULD support the Self-Asserted Attributes outlined in this section.

An IXP SHOULD support brokering request for these attributes.

Self-asserted attributes are subject to an Open access policy and attribute requests MAY be fulfilled at any assurance level unless an attribute's sharing policy explicitly states otherwise.

An IXP MUST request express consent from the individual on every change to any Self-Asserted Attributes.

2.3.1 Preferred Name

An individual may nominate a name they prefer to be known as at the PRPs they access. A Preferred Name can reflect an individual's personal preference or may be culturally significant. Ideally an individual should be able to nominate a preferred name they wish to use on the AGDIS.

Given an individual may use the attribute at their ISP to satisfy cultural requirements the preferred name SHOULD be easy to update.

Although it is a Core Attribute a Preferred Name MUST not be used as an authoritative attribute in same manner as a PRP would use any of the verified name attributes (Family Name, Give Names or Middle Names).

The data representation for preferred names follows the requirements for the verified name attributes, and the data representation of the federation protocol(s) used to fulfill an attribute request.

2.3.2 Addresses

An individual MAY self-assert a residential, postal, business, or other addresses.

Addresses MAY be collected during the identity proofing process or added by the individual during the normal use and management of their digital ID.

The attribute response for addresses MAY be a single address or a collection of addresses.

The data representation for an Address attribute MUST contain the following fields:

(a) the address type;
(b) street address;
(c) locality;
(d) region;
(e) post code; and
(f) validated.

The data representation MAY also contain the following fields:

(a) country; and
(b) formatted.

A list valid address type URNs is outlined in Table 33 below.

The field locality maps to suburb, city, administrative area, or place name.

The field region maps to the larger administrative areas like state, territory, province, or prefecture.

The field formatted, a print-ready string of the address, MAY also be included in the address object.

A detailed description of the fields in an address are outlined in Table 32 below.

An ISP fulfilling an address attribute request where the type of address is not specified SHOULD return a default value nominated by the individual.

2.3.3 Other Email Addresses

An individual MAY self-assert additional email addresses for use in different contexts, for example a when accessing business services with their personal digital ID.

An ISP MUST NOT permit an individual to add an email address that is currently claimed as the validated email address for the individual's or other active digital ID accounts on their system.

If an ISP uses the Validated Email Address as the account identifier, the ISP MAY use any of the other email addresses asserted by an individual as an ad hoc replacement for the individual's account identifier.

An ISP MUST validate the individual has control over all other email addresses they assert if it is used as a replacement for the account identifier.

The attribute response for other email addresses MAY be a single email address or a collection of email addresses.

Each Other Email Address MUST contain:

- the email address;
- a flag indicating if the email address has been validated; and
- a contact type label.

The data representation for each other email address string MUST conform to the requirements outlined above for Validated Email Address in section 2.1.2 of this Schedule.

The contact type value MUST be one the contact type URNs enumerated in Table 35 below.

An ISP fulfilling an attribute request for other email address where the contact type of address being requested is not specified SHOULD return a default value nominated by the individual.

2.3.4 Other Phone Numbers

An individual may self-assert additional phone numbers to use with their digital ID, for example an additional phone number may be a personal landline number or business phone number.

If an ISP uses the validated phone number as an account identifier, the ISP MUST NOT use any of the other phone numbers asserted by an individual as an ad hoc replacement for the individual account identifier.

An ISP SHOULD validate the individual has control over any other asserted phone numbers.

The attribute response for other phone numbers MAY be a single phone number object or a collection of phone number objects.

The data representation for each Other Phone Number object MUST contain:

- the phone number;
- a telephony type;
- a flag indicating if the phone number has been validated; and
- a contact type label.

The data representation for each Other Phone Number string MUST conform to the requirements outlined for the validated phone number in section 2.1.2 of this Schedule.

The telephony type MUST be one of the URNs enumerated in Table 34 below.

The contact type value MUST be one the contact type URNs enumerated in Table 35 below.

An ISP fulfilling an attribute request for Other Phone Numbers where one of contact type or telephony type is not specified SHOULD return a default value nominated by the individual.

Table 32 Self-asserted address attribute fields

Field	Required	Description
Address Type	Y	One of the enumerated values in Table 33 below.
Formatted	N	A print ready formatted string representation of the address.
Street address	Y	Full street address string, which MAY include house number, street name, Post Office Box, and multi-line extended street address information. The field MAY contain multiple lines separated by newline delimiters. The new line can be delimited by a carriage return (‘\n’) or a carriage return and line feed character (‘\r\n’).
Locality	Y	A string representing the city, suburb, or placename of the address.
Region	N	A string representing the state, territory, province, prefecture or region.
Post Code	Y	A string representing the postal or zip code.
Country	N	A string representing the country of the address.
Validated	Y	A Boolean value indicating if the legitimacy of the address has been validated.

Table 33 Address types for a collections

Address Type	URN	Notes
Default	`urn:id.gov.au:agdis:address:default`	Used in requests only. Selects the default address type nominated by the individual.
Residential	`urn:id.gov.au:agdis:address:residential`	-
Postal	`urn:id.gov.au:agdis:address:postal`	-
Business	`urn:id.gov.au:agdis:address:business`	-
Work	`urn:id.gov.au:agdis:address:work`	-
Other	`urn:id.gov.au:agdis:address:other`	-

Table 34 The telephony type for the other phone number attribute

Phone Type	URN	Notes
Default	`urn:id.gov.au:agdis:tel:default`	Used in requests only. Selects the default address type nominated by the individual.
Text	`urn:id.gov.au:agdis:tel:text`	-
Voice	`urn:id.gov.au:agdis:tel:voice`	-
Fax	`urn:id.gov.au:agdis:tel:fax`	-
Cell	`urn:id.gov.au:agdis:tel:cell`	-
Mobile	`urn:id.gov.au:agdis:tel:mobile`	-
Video	`urn:id.gov.au:agdis:tel:video`	-
Pager	`urn:id.gov.au:agdis:tel:pager`	-
Textphone	`urn:id.gov.au:agdis:tel:textphone`	-
Other	`urn:id.gov.au:agdis:tel:other`	-

Table 35 The category of contact and email or phone represents

Contact Type	URN	Notes
Default	`urn:id.gov.au:agdis:contact:default`	Used in requests only. Selects the default contact type nominated by the individual
Personal	`urn:id.gov.au:agdis:contact:personal`	
Business	`urn:id.gov.au:agdis:contact:business`	
Work	`urn:id.gov.au:agdis:contact:work`	
Other	`urn:id.gov.au:agdis:contact:other`	

2.3.5 Place of Birth

An individual can self-assert their place of birth. For individuals holding an Australian Birth Certificate the Place of Birth field is not verified during the identity proofing process and presently there are no readily available mechanisms to validate foreign birth certificates and their attributes.

If an ISP supports this self-asserted attribute, they SHOULD ensure the location supplied by the individual is real if practical to do so.

The data representation for this attribute is a string with a maximum length of 255 characters.

2.3.6 Personal Titles

Qualification, honorific or gender-based titles can asserted by the individual to tailor user experiences at a PRP.

An ISP MAY allow an individual to specify their preferred title.

An ISP SHOULD appropriately limit the types of titles an individual may choose.

The data representation for this attribute is string with a maximum length of 100 characters.

2.4 Computed Attributes

There are no Computed Attributes outlined in this Schedule or presently shared in the AGDIS.

Computed Attributes MUST be used to help the AGDIS achieve the requirements of the data minimisation principle, and limit the data transmitted across its digital ID data environment.

Computed Attributes are dynamically derived from existing attributes and attribute sets using a defined algorithm and SHOULD assert a statement about an individual without necessarily containing any identity information.

Attributes sharing policies for Computed Attributes MUST align with the underlying attributes the computed value is derived from. Specifically, a Computed Attribute derived from attributes subject to a Restricted access policy SHOULD also be a restricted attribute.

Table 36 Self-Asserted Attributes

Attribute	Description	Access policy	IP level	Fulfill	Proven-ance	Consent	Data representation
Preferred Name	A name or names asserted by the individual that may not be verifiable.	Open	IP1 and stronger	Best effort	ISP managed	Every change	A non-zero length string or a collection of non-zero length strings.
Address	A residential, postal, or other address that can be gathered from the individual during the identity proofing process or the normal management of their digital ID account at their ISP.	Open	IP1 and stronger	Best effort	ISP managed	Every change	A single address or collection of addresses. Each address MUST adhere to the data representation outlined in section 2.3.2 of this Schedule.
Other Email Addresses	Addition email address.		IP1 and stronger				
Other Phone Numbers	Additional phone numbers provided by the individual during the identity proofing process or the normal management of their digital ID account at their ISP.	Open	IP1 and stronger	Best effort	ISP managed	Every change	A phone object or collection phone numbers. Each phone number MUST adhere to the data representation outlined in section 2.3.4 of this Schedule.
Personal Titles	A free text attribute allowing individuals to specify their qualification, honorific or gender-based titles.	Open	IP1 and stronger	Best effort	ISP managed	Every change	A non-zero length string.

3. Attribute Service Provider (ASP) Profiles

This chapter provides the attribute sharing policies for the attributes and attribute sets provided by ASPs.

An IXP MUST implement support to broker access to these attributes in accordance with this chapter's attribute sharing policies.

An IXP MUST not broker requests for ASP managed attributes or attribute sets to ISPs to ensure AGDIS privacy requirements are maintained.

Table 37 ASP managed attributes and attribute sets

Attribute Set	Attributes	Description
Business Authorisations	Authorisation Schemas Unique Relationship ID Entity ID Entity Type Entity Name Contact Emails Relationship Type Relationship Start Time Relationship End Time Roles Entitlements Attributes Last Updated	The Australian Taxation Office's Relationship Authorisation Manager (RAM) manages the authorisation for a person to act on behalf of a business entity that is registered with the Australian Business Register (ABR) and issued with an Australian Business Number (ABN).
myGov Link	myGov Link ID	A pairwise identifier used by the myGov platform. This is attribute is subject to restrictions.

3.1 Business Authorisations

Business Authorisations are an attribute set managed by the Australian Taxation Office's Relationship Authorisation Manager (RAM). A Business Authorisation represents a relationship between an individual, an Australian business, and a government service provider.

Business Authorisations have the following 4 key components:

(a) the individual or subject of the authorisation;
(b) the business on behalf of which the authorisation allows the individual to act, represented by an Australian Business Number (ABN);
(c) the service provider the individual's business authorisation is for; and
(d) the creator of the authorisation, a principal authority or administrator.

A summary of the attribute sharing policy for business documents is presented below.

Business Authorisations are subject to an open access sharing policy. With the caveat that a PRP SHOULD be established in RAM as a service provider before making an attribute request for Business Authorisations.

Acting in its capacity as an ASP, RAM:

(a) MUST request express consent from the individual when they accept a Business Authorisation;
(b) MUST inform the individual of the terms of the ongoing consent and the use cases it covers, and how it can be revoked;
(c) MAY notify a service provider when an authorisation has been issued, revoked, or expired; and
(d) SHOULD notify the subject of a Business Authorisation when the authorisation has been revoked or expired.

An IXP brokering attribute request to RAM MUST facilitate the mapping of the IXP's issued PRP client identifier(s) to the identifier RAM has assigned to a service provider.

The data representation for Business Authorisation is a collection of business authorisations, where each business authorisation contains the values outlined below.

3.2 myGov Link Identifier

The myGov Link Identifier is a pairwise identifier assigned by myGov to map the relationship between an individual and myGov's member service. It serves a similar purpose to digital ID pairwise identifiers outlined in section 2.2.1.1 of this Schedule.

The myGov Link Identifier is subject to the Platform access policy.

myGov MUST request express consent on every change to the myGov Link Identifier.

An IXP MUST NOT broker requests for the myGov Link Identifier to an ISP.

An IXP MUST only broker requests for the myGov Link Identifier to PRPs that are myGov member service.

The data representation for the myGov Link Identifier is managed by myGov and shared directly with its member services.

Table 38 AGDIS ASPs Attribute sharing policy summary

Attribute	Description	Access policy	Fulfillment	Provenance	Consent	Data representation
Business authorisations	A collection of business authorisations that permit and individual to act on behalf of a business at a government service.	Open	Best effort	ASP managed	Ongoing	Each entry in the collection MUST follow the data representation outlined in Table 40 below.
myGov Link ID	Specific to the myGov platform, the myGov link ID is a pairwise ID used to map relationships between myGov and its member services.	Platform	Best effort	ASP managed	Every change	Determined by myGov.

Table 39 Business Authorisation schema URNs

Version	URN	Notes
$VERSION	`urn:id.gov.au:tdif:authorisatons:business:$VERSION`	The generic URN format for business authorisations.
1.0	`urn:id.gov.au:tdif:authorisatons:business:1.0`	The legacy version for business authorisations.

Table 40 Schema definition for Business Authorisations

Field	Description	Data representation
Schemas	A URN or collection of URNs denoting the version of the business authorisation payload.	The collection MUST contain at least one URN denoting the version of data representation for this business authorisation. The format for the mandatory URNs is prescribed in Table 39 above.
Unique Entity Relationship ID	Unique identifier for the relationship between the individual and the entity issued by RAM.	A string representation of a unique identifier.
Entity ID	The unique identifier for the entity. In practice this is the ABN of the entity and in the future may represent additional identifiers.	A string with constraints determined by the Entity Type field.
Entity Type	The type of entity identifier.	A string that MUST be one of the enumerated values defined in Table 41 below.
Entity Name	The name of the entity. Information about the entity may be separately available from ABR using the Entity ID.	A string with a maximum length of 200 characters.
Contact Details	The email address for the individual may not match the individual's validated email address.	Email address conforming to RFC 5322 address syntax. With a maximum length of 254 characters in compliance with RFC 2821.
Relationship Type	The type of relationship the individual has with the entity.	A non-empty string.
Relationship Start Time	The date and time from which the authorisation is becomes valid.	RFC 3339 date and time in coordinated universal timeformat.

Field	Description	Data representation
Relationship End Time	The date and time after which the authorisation is no longer valid.	RFC 3339 date and time in coordinated universal time.format.
Attributes	A collection of key-value pairs describing the business authorisation.	A collection of objects with only 2 fields – name and value. An enumeration of these values is defined by RAM.
Roles	A list of literal values that define the roles an individual can assume on behalf of the entity.	A collection of literals defined by RAM.
Entitlements	Additional privileges or permissions the individual may have when acting on behalf of the entity. The entitlements may be specific to a given PRP context.	A collection of literals defined by RAM or the service provider.
Attributes Last updated	Date and time of when the authorisation was last modified.	RFC 3339 date and time in coordinated universal time format.

Table 41 Business Authorisation entity type enumeraiton

Entity Type	Description	Entity ID constraints
ABN	Australian Business Number	A numeric string of 11 characters in length.

4. OpenID Connect Attribute Profile

To enable the use of attributes and attribute sets across the AGDIS an idiomatic mapping from this profile to Schedule 2 (AGDIS OpenID Connect Profile) is outlined in this chapter.

The mapping outlines requirements to ensure interoperability amongst entities participating in the AGDIS making and fulfilling attribute requests. The data types assigned to the mapped OIDC scopes and claims, for each attribute, are also defined here along with their representation.

4.1 Attribute mapping

This Schedule outlines 2 attribute mappings:

(a) IXP to PRP; and
(b) ISP to IXP.

An IXP MUST implement support for the IXP relying party mapping (section 4.1.1 of this Schedule) for its PRPs.

PRPs using OpenID Connect Core 1.0 as their federation protocol SHOULD use the IXP relying party mapping (section 4.1.1 of this Schedule) when configuring their service.

An ISP MUST implement support for the ISP relying party mapping (section 4.1.2 of this Schedule) for its connected AGDIS IXPs.

An IXP MUST support brokering attribute requests from the IXP relying party mapping to the ISP relying party mapping to support ISPs and PRPs using OpenID Connect Core 1.0 as their federation protocol (as outlined in Schedule 2 (AGDIS Open ID Connect Profile)).

4.1.1 Identity Exchange Provider Relying Party Mapping

The scopes an IXP MUST make available for a PRP to request are:

- openid
- profile
- email
- phone
- tdif_other_names
- tdif_docs
- tdif_business_authorisations

The attribute sets mapped to these OpenID Connect Core 1.0 scopes along with the relevant sets of claims is outlined in Table 42 below. A detailed attribute mapping to claims, along with data types, is presented in Table 46 below.

Additional scopes for Self-Asserted Attributes that an IXP SHOULD support brokering attribute requests for are:

- adgis_address
- agdis_other_email
- agdis_other_phone
- agdis_birth_place
- agdis_personal_title

The Self-Asserted Attributes are mapped to OIDC scopes and claims are outlined in Table 43 below, with detailed mapping from attributes to claims provided in Table 46 below.

An IXP SHOULD support brokering of attributes requests for each claim outlined in Table 42 below and Table 43 below.

An IXP MAY fulfill attributes request via either the Token or UserInfo Endpoints.

An IXP MUST encrypt all scopes and claims, except for the common attributes set, when fulfilling attribute requests via the Token Endpoint.

4.1.2 Identity provider scopes and claims

The scopes an ISP MUST make available for its IXPs to request are:

- `openid`
- `tdif_core`
- `tdif_email`
- `tdif_phone`
- `tdif_other_names`
- `tdif_docs`

The attribute sets mapped to these OpenID Connect Core 1.0 scopes along with the relevant sets of claims are outlined in Table 44. A detailed attribute mapping to claims, along with data types, is presented in Table 46.

Additional scopes for Self-Asserted Attributes an ISP SHOULD make available for its IXPs to request are:

- `adgis_address`
- `agdis_other_email`
- `agdis_other_phone`
- `agdis_birth_place`
- `agdis_personal_title`

The self-asserted attribute sets are mapped to OpenID Connect Core 1.0 scopes and claims are outlined in Table 45, with detailed mapping from attributes to claims in Table 46.

An ISP SHOULD support attribute requests for each individual claim outlined in Table 44 and Table 45.

An ISP MAY fulfill attributes request via either the Token or UserInfo Endpoints.

An ISP SHOULD encrypt all scopes and claims, except for the common attributes set, when fulfilling and attribute requests via the Token Endpoint.

Table 42 OIDC Attribute profile for IXP relying parties

Attribute Set	Scope	Claims	Endpoint	Notes

Common	`openid`	`sub`
		`auth_time`
		`acr`
		`amr`
		`tdif_audit_id`
			Token
			UserInfo
				An IXP SHOULD ignore or respond with an error to attribute requests if this scope is missing.
				All common claims MUST be present in the response.

Core	`profile`	`name`
		`family_name`
		`given_name`
		`middle_name`
		`preferred_username`
		`birthdate`
		`updated_at`
			Token
			UserInfo
				Preferred name is a self-asserted attribute, see section 2.3.1 of this Schedule .

Attribute Set	Scope	Claims	Endpoint	Notes
Validated Email	email	email email_verified tdif_email_updated_at	Token UserInfo	An IXP MUST return all claims when fulfilling the scope.
Validated Phone	phone	phone phone_number_verified tdif_phone_updated_at	Token UserInfo	An IXP MUST return all claims when fulfilling the scope.
Verified Other Names	tdif_other_names	tdif_other_names tdif_other_names_updated	Token UserInfo	An IXP MUST return all claims when fulfilling the scope.
Verified documents	tdif_doc	tdif_doc	UserInfo	Access to these scopes and claims is restricted.
Business authorisations	tdif_business_authoristions	tdif_business_authoristions	UserInfo	Supplied by the Australian Taxation Office’s Relationship Authorisation Manager (RAM).

Table 43 Self-asserted attribute scopes and claims for IXP relying parties

Attribute Set	Scope	Claims	Endpoint	Notes
Preferred name	`profile`	`preferred_username`	Token UserInfo	This is the same attribute from Table 42 above an IXP MUST support brokering this claim.
Addresses	`adgis_address`	`adgis_address`	Token UserInfo	A PRP may specify values for the desired address type using the address type URNs outlined in Table 33 above when requesting this attribute via the claim parameter.
Other Email Address	`agdis_other_email`	`agdis_other_email`	Token UserInfo	A PRP may specify values for the desired email contact type using the contact type URNs outlined in Table 35 above when requesting this attribute via the claim parameter.
Other Phone Numbers	`agdis_other_phone`	`agdis_other_phone`	Token UserInfo	A PRP may specify the desired contact category or telephony type using the URNs outlined in Table 34 and Table 35 above when requesting this attribute via the claim parameter.

Attribute Set	Scope	Claims	Endpoint	Notes
Place of Birth	`agdis_birth_place`	`agdis_birth_place`	Token UserInfo	
Personal Title	`agdis_personal_title`	`agdis_personal_title`	Token UserInfo	

Table 44 OIDC Attribute profile for ISP relying parties

Attribute Set	Scope	Claims	Endpoint	Notes

Common	`openid`	`sub`
		`auth_time`
		`acr`
			Token
			UserInfo
				As per OpenID Connect Core 1.0 this scope MUST be present.

Core	`tdif_core`	`name`
		`family_name`
		`given_name`
		`middle_name`
		`preferred_username`
		`birthdate`
		`updated_at`
		`tdif_core_updated_at`
			Token
			UserInfo	

Validated Email	`tdif_email`	`email`
		`email_verified`
		`tdif_email_updated_at`
			Token
			UserInfo	

Attribute Set	Scope	Claims	Endpoint	Notes

Validated Phone	`tdif_phone`	`phone`
		`phone_number_verified`
		`tdif_phone_updated_at`
			Token
			UserInfo	

Verified Other Names	`tdif_other_names`	`tdif_other_names` `tdif_other_names_updated`	Token UserInfo	
Verified documents	`tdif_doc`	`tdif_doc`	UserInfo	An IXP MUST only request this attribute set when brokering attribute request for approved PRPs. The scope payload is defined in section 4.3.4 of this Schedule.

Table 45 Self-asserted attribute scopes and claims for ISP relying parties

Attribute Set	Scope	Claims	Endpoint	Notes
Preferred name	`tdif_core`	`preferred_username`	Token UserInfo	This is the same attribute from Table 44 above an ISP MUST support brokering this claim.
Addresses	`adgis_address`	`adgis_address`	Token UserInfo	When brokering request for a specific address type an IXP MUST ensure the requested address type(s) conform to the address type URNs outlined in Table 33 above.
Other Email Address	`agdis_other_email`	`agdis_other_email`	Token UserInfo	When brokering requests for a specific category of email address an IXP MUST ensure the requested contact type conforms to the URNs outlined in Table 35 above.
Other Phone Numbers	`agdis_other_phone`	`agdis_other_phone`	Token UserInfo	When brokering requests for a specific type of phone number an IXP MUST ensure the requested telephony type or contact type conform to the URNs specified in Table 34 and Table 35 above.

Attribute Set	Scope	Claims	Endpoint	Notes
Place of Birth	`agdis_birth_place`	`agdis_birth_place`	Token UserInfo	
Personal Title	`agdis_personal_title`	`agdis_personal_title`	Token UserInfo	

Table 46 OpenID Connect attribute mapping

Attribute	OIDC claim	JSON type	AGDIS type	Requestable from	Reference
Digital Identity (user identifier)	`sub`	string	-	ISP, IXP	OIDC Core 1.0, section 2 OIDC Core 1.0, section 8
Full Name	`name`	string	RequiredNameString	ISP, IXP	OIDC Core 1.0, section 5.1
Family Name	`family_name`	string	RequiredNameString	ISP, IXP	OIDC Core 1.0, section 5.1
Middle Names	`middle_name`	string	NameString	ISP, IXP	OIDC Core 1.0, section 5.1
Given Names	`given_name`	string	NameString	ISP, IXP	OIDC Core 1.0, section 5.1
Preferred Name	`preferred_username`	string	NameString	ISP, IXP	OIDC Core 1.0, section 5.1
Date of birth	`birthdate`	string	Date	ISP, IXP	OIDC Core 1.0, section 5.1
Core Attributes Last Updated	`tdif_core_upadated_at`	number	Unix timestamp	ISP, IXP	Section 4.2.2 of this Schedule
Validated Email	`email`	string	Email	ISP, IXP	OIDC Core 1.0, section 5.1
Email Validated Indicator	`email_verified` Note: Under this profile this claim MUST always be True	literal	Boolean	ISP, IXP	OIDC Core 1.0, section 5.1

Attribute	OIDC claim	JSON type	AGDIS type	Requestable from	Reference
Validated Email Last Updated	`tdif_email_updated_at` Note: Under this profile this claim MUST always be True	number	Unix timestamp	ISP, IXP	Section 2.1.2 section 4.2.2 of this Schedule
Validated Mobile Phone Number	`phone_number`	string	Phone	ISP, IXP	OIDC Core 1.0, section 5.1
Mobile Phone Number Validated Indicator	`phone_number_verified` Note: Under this profile this claim MUST always be True	literal	Boolean	ISP, IXP	OIDC Core 1.0, section 5.1
Validated Mobile Phone Last Updated	`tdif_phone_number_updated_at`	number	Unix timestamp	ISP, IXP	Section 2.1.2, section 4.2.2 of this Schedule
Verified Other Names	`tdif_other_names`	complex type	Other Names Object	ISP, IXP	Section 2.1.3, section 4.2.3.1 of this Schedule
Other Verified Names Last Updated	`tdif_other_names_updated_at`	number	Unix timestamp	ISP, IXP	Section 2.1.3, section 4.2.2 of this Schedule
Verified Documents	`tdif_doc`	complex type	Verified Documents	ISP, IXP	Section 2.1.4, section 4.3.4of this Schedule

Attribute	OIDC claim	JSON type	AGDIS type	Requestable from	Reference
Assurance level	`acr`	string	URN	ISP, IXP	OIDC Core 1.0, section 2 Section 2.2.1.3 of this Schedule
Authentication Method	`amr`	string	URN	IXP	OIDC Core 1.0, section 2 Section 2.2.1.4 of this Schedule
Authentication Time	`auth_time`	number	Unix timestamp	ISP, IXP	OIDC Core 1.0. section 2
RP Audit ID	`tdif_audit_id`	string	UUID	IXP	Schedule 1 (AGDIS Onboarding Specifications)
myGov Link Identifier	`mygov_link_id`	string	-	IXP	myGov defined payload
Business authorisations	`tdif_business_authorisations`	complex type	Business Authorisation	IXP, ASP	Section 3.1 of this Schedule
Last Updated	`update_at`	number	Unix Timestamp	ISP, IXP	OIDC Core 1.0, section 5.1

Table 47 OpenID Connect Mapping for Self-Asserted Attributes

Attribute	OIDC claim	JSON type	AGDIS type	Requestable from	Reference
Preferred Name	`preferred_username`	string	NameString	ISP, IXP	Section 2.1.1 of this Schedule Section 2.3.1 of this Schedule
Address	`agdis_address`	complex type	Address Object or Address Object Collection	ISP, IXP	Section 2.3.2 of this Schedule
Other Email Addresses	`agdis_other_email`	complex type	Other Email Address Object or Other Email Address Object Collection	ISP, IXP	Section 2.3.3 of this Schedule
Other Phone Number	`agdis_other_phone`	complex type	Other Phone Number Object or Other Phone Number Object Collection	ISP, IXP	Section 2.3.4 of this Schedule
Place of Birth	`agdis_birthplace`	string	-	ISP, IXP	Section 2.3.5 of this Schedule

Attribute	OIDC claim	JSON type	AGDIS type	Requestable from	Reference
Personal Title	`agdis_personal_title`	string	-	ISP, IXP	Section 2.3.6 of this Schedule

4.2 Data types

4.2.1 JavaScript Object Notation types

The primitive data types assigned to attributes conveyed via claims and scopes conform to the JSON data interchange format as outlined in RFC 8529. The primitive JSON data types are the building blocks used in the data type definitions for the simple and complex types used in the AGDIS.

4.2.2 Simple data types

Simple data types are JSON primitives with additional constraints applied that MUST be applied to the held value for an attribute. The AGDIS type and its associated JSON type with constraints are presented in Table 48 below.

4.2.2.1 NameString

A JSON string that can represent name attributes that are not mandatory.

The value has a:

(a) minimum length of zero characters; and
(b) maximum length of 100 characters.

4.2.2.2 RequiredNameString

A JSON string that can represent name attributes that are mandatory.

The value has a:

(a) minimum length of 1 character; and
(b) maximum length of 100 characters.

4.2.2.3 Boolean

A shorthand representation for a value that can assume one of the JSON literals `true` and `false`. All other values are illegal.

4.2.2.4 Date

RFC 3339 compliant date string that MUST conform to one of the following:

- YYYY
- YYYY-MM
- YYYY-MM-DD

This value is commonly used to represent birth dates.

4.2.2.5 DateTime

RFC 3339compliant date and time string capturing a significant date and time in Coordinated Universal Time (UTC). Values MUST assume one of the following forms:

- YYYY-MM-DDTHH:MM:SSZ
- YYYY-MM-DDTHH:MMZ

4.2.2.6 Unix Timestamp

A JSON number representing a date and time as the number of seconds since 1970-01-01T00:00:00Z up to when the time was observed.

Note: Follows the same format as the OIDC `updated_at` claim outlined in section 5.1 of the OpenID Connect Core 1.0 profile.

4.2.2.7 Email

A non-empty JSON string with a maximum length of 254 characters (defined by RFC 2821) with a format that conforms to the syntax defined in RFC 5322.

4.2.2.8 PhoneNumber

A non-empty JSON string with a maximum length of 15 characters with format the confirms to the phone numbering for prescribed in ITU-T E.164.

4.2.2.9 UUID

A JSON string that MUST be at most 36 characters in length and is generated in accordance with RFC 4122.

Selection of an appropriate version of the UUID version to be used is at the discretion of the participant. The UUID version choice MUST be suitable to avoid collisions.

Table 48 Simple data type definitions

AGDIS type	JSON type	Constraints
Name string	string	Minimum length 0 characters. Maximum length 100 characters.
Required Name String	string	A non-empty string. Maximum length 100 characters.
Boolean	JSON literal	Either `true` or `false`.
Date	string	RFC 3339 compliant date string that MUST conform to one of the following: • YYYY • YYYY-MM • YYYY-MM-DD
Date Time	string	RFC 3339 compliant date and time string of the form YYYY-MM-DDTHH:MM:SSZ.
Unix Timestamp	number	A JSON number that represents the time as the number of seconds from 1970-01-01T00:00:00Z as measured in coordinated universal time until the present time. Note: Follows the same format as the OIDC `updated_at` claim outlined in section 5.1 of the OpenID Connect Core 1.0 profile.
Email	string	A non-empty string with a maximum length of 254 characters (defined by RFC 2821) with a format that conforms to the syntax defined in RFC 5322.
Phone Number	string	A string with a maximum length of 15 characters with format the confirms to the phone numbering for prescribed in ITU-T E.164.
UUID	string	A string that MUST be 36 characters in length and is generated in accordance with RFC 4122.

4.2.3 Complex data types

4.2.3.1 Other Names Object

An Other Names Object is a JSON Object that represents an instance of a Verified Other Name.

The data types outlined in Table 46 above for the encapsulated claims MUST be applied.

At least one value of the claims MUST be present for the Other Name Object to be valid.

Table 49 Other Names Object fields

Attribute	Claim/Field name	Required
Family Name	`family_name`	N
Given Name	`given_name`	N
Middle Name	`middle_name`	N
Full Name	`full_name`	N

4.2.3.2 Address Object

The Address Object is a JSON Object that is intended to be used for the self-asserted address scope and claims. The specification for this data type maps directly to the definition outlined in Table 32 above.

Table 50 Address Object fields definitions

Attribute	Field name	Data type	Required
Address Type	address_type	URN from values outlined in Table 33 above	Y
Formatted	formatted	string	N
Street address	street_address	string	Y
Locality	locality	string	Y
Region	region	string	Y
Post Code	post_code	string	Y
Country	country	string	N
Validated	validated	Boolean	Y

4.2.3.3 Other Email Address Object

The Other Email Address Object is a JSON Object that is intended to be used for the self-asserted other email address scope and claim.

The data types for the fields of the Other Email Address JSON Object, outlined in section 2.3.3 of this Schedule, are presented in Table 51 below.

Table 51 Other Email Address type field definitions

Attribute	Field name	Data type	Required
Email	email	Email	Y
Email Validated	email_validated	Boolean	Y
Contact Type	contact_type	URN from the values prescribed in Table 35 above	Y

4.2.3.4 Other Phone Number Object

The Other Phone Number Type is a JSON Object that is intended to be sued for the self-asserted other phone numbers scope and claim.

The data types for the fields Other Phone Number JSON Object, outlined in section 2.3.4 of this Schedule, are presented in Table 52 below.

Table 52 Other Phone Number type field definitions

Attribute	Field name	Data type	Required
Other Phone	`other_phone`	Phone	Y
Other Phone validated	`other_phone_validated`	Boolean	Y
Other Phone Telephony Type	`telephony_type`	URN from the values prescribed in Table 34 above	Y
Contact Type	`contact_type`	URN from the values prescribed in Table 35 above	Y

4.3 Mutual attributes

4.3.1 Core

Each of the Core Attributes are assigned a simple data type (see Table 46 above). Additional requirements for the Core Attribute data representation are outlined in Table 53 below. Normative examples are available in section 4.8 of this Schedule.

Table 53 Core claim requirements

Claim	AGDIS data type	Requirements
`name`	RequiredNameString	Value MUST be a concatenation of given_name, middle_name and family_name. The joining character SHOULD be a space.
`family_name`	RequiredNameString	MUST NOT be an empty string.
`given_name`	NameString	MAY be excluded from attribute responses when value is an empty string.
`middle_name`	NameString	MAY be excluded from attribute responses when value is an empty string.
`birthdate`	Date	SHOULD be validated to ensure dates that satisfying the RFC 3339 format and are reasonable.
`updated_at`	Unix Timestamp	MUST be derived from the most recently updated Core Attribute.
`tdif_core_updated_at`	Unix Timestamp	IXPs MUST NOT broker this claim to PRPs. ISPs MUST supply this claim.

4.3.2 Validated Contact Details

As outlined in section 2.1.2 of this Schedule the Validated Contact attributes cover the individual's validated email and phone. The data types applied to the claims in the attribute response MUST conform to the values outlined in Table 46 above.

All claims MUST be returned when fulfilling the `email` and `phone` scopes.

An ISP MUST not fulfill this request if the related email or phone number have not been validated.

Table 54 Requirements for Validated Email claims

Claim	AGDIS data type	Required for scope	Constraints
`email`	Email	Y	
`email_verified`	Boolean	Y	If the email has not been successfully validated the scope for these claims should not be fulfilled.
`tdif_email_updated_at`	Unix Timestamp	Y	

Table 55 Requirements for Validated Phone Number claims

Claim	AGDIS data type	Required for scope	Constraints
`phone`	Phone Number	Y	
`phone_verified`	Boolean	Y	If the phone number has not been validated successfully the scope for these claims should not be fulfilled.
`tdif_phone_number_updated_at`	Unix Timestamp	Y	

4.3.3 Verified Other Names

The Verified Other Names scope and claim uses the Other Names object for each of the verified other names returned in an attribute request.

An ISP fulfilling this request MAY return a single Other Name object or a JSON array containing the single value when only one Verified Other Name set is available.

An ISP fulfilling this request MUST return a JSON array of Other Name object when two or more Other Verified Names are available.

Table 56 Requirements for Verified Other Names claims

Claim	AGDIS data type	Required for scope	Constraints
`tdif_other_names`	Other Names Object or an Array of Other Names Objects	Y	Returning a single OtherNameObject is only permitted when a single other name has been verified.
`tdif_other_names_updated_at`	Unix Timestamp	Y	-

4.3.4 Verified Documents

A verified document MUST be represented as a JSON Object with a structure conformant to the schema outlined in Table 10 above with the literal field names and data types outlined in Table 58 below.

The responses to an attribute request for a single verified document MAY be serialised as a single Verified Document Object or as wrapped in a JSON Array.

An attribute response for multiple verified documents MUST be serialised as a JSON Array of Verified Document Objects.

An ISP MUST apply the rules outlined in Table 10 above when preparing an attribute response if the attribute request for verified documents comes via the `scope` parameter.

Using the `claims` request parameter an attribute request for one or more specific documents can be made, using the document type code URNs outlined in Table 11 above. An example request is illustrated below in Figure 1.

```
{
    …,
    “userinfo”: {
        …,
        “tdif_docs”: {
            “values”: [
                “urn:id.gov.au:tdif:doc:type_code:MD”,
                “urn:id.gov.au:tdif:doc:type_code:DL”,
                “urn:id.gov.au:tdif:doc:type_code:PP”
            ]
        },
        …
    },
    …
}
```

Figure 1 Verified Documents sample claim request.

Table 57 Verified Document Object schema

Document attribute	Payload name	JSON type	AGDIS data type	Requirements
Document Type Code	`type_code`	string	URN	MUST be on of values defined in Table 11 above.
Document Verification Method	`verification_method`	string	-	MUST be one of values defined in Table 12 above. MUST be 1 character in length.
Document Verification Date	`verification_date`	string	Date Time	-
Document Identifiers	`identifiers`	Object	Document Identifiers Object	See section 4.3.4.2 of this Schedule.

Document attribute	Payload name	JSON type	AGDIS data type	Requirements
Document Names	`names`	Object	Document Names Object	See section 4.3.4.1 of this Schedule.
Document Date of Birth	`birthdate`	string	Date	Valid DVS dates are between 1753-01-01 to 3000-12-31
Document Attributes	`attributes`	Object	Document Attribute Object	

4.3.4.1 Document Identifiers Object

The Document Identifiers field of a Verified Document Object is a JSON Object with keys and values determined by the parent objects document type.

The document type, held in the `type_code` claim, is mapped to an enumeration of permitted values in Table 58 below. The mapping does not imply the required the presence of a document identifier, for the specified verified document type, for the payload to be valid. For a given document type, the availability of document identifiers is dependent upon an identity document's issuing authority and when it was issued.

For a Document Identifiers Object to be valid it MUST:

(a) only have keys that map to the parent object's Document Type field; and

(b) be compliant with value data types prescribed in Table 58 below for a given identifier.

Table 58 Verified Document Names with data types

Document name attribute literal	Value data type	Value requirements	Used by document types
`Family Name`	string	Maximum length 100 characters. May be empty if `Give Names` has a value for documents issued by NSW, VIC, or WA Births Deaths & Marriages.	BC, MC, NC, CC, RD, IM, VI, DL, PP

Document name attribute literal	Value data type	Value requirements	Used by document types
`Given Names`	string	Maximum length 100 characters. May be empty if `Family Name` has a value for a document issued by NSW, VIC, or WA Births Deaths & Marriages.	BC, MC, NC, CC, RD, IM, VI, DL, PP
`Family Name 2`	string	Maximum length 50 characters. May be empty if `Given Names 2` has a value for a document issued by NSW, VIC, or WA Births Deaths & Marriages.	MC
`Given Names 2`	string	Maximum length 60 characters. May be empty if `Family Name 2` has a value for a document issued by NSW, VIC, or WA Births Deaths & Marriages.	MC
`Middle Name`	string	Maximum length 20 characters.	DL
`Name`	string	Maximum length 32 characters	CO

Table 59 Verified Document Identifiers with data types

Document identifier string	Value data type	Value requirements	Used by document types
`Registration Number`	string	Maximum length 10 characters. Only available for ACT, NT, NSW, TAS, VIC, and WA.	BC, MC, NC
`Registration Date`	Date	-	BC, MC, NC
`Registration Year`	Number	Integer value for the year.	BC, MC, NC
`Certificate Number`	string	Max length 11 characters. Only composed of numeric characters. Only available for NSW, TAS, ACT, NT, SA, and VIC.	BC, MC, NC
`Stock Number`	string	Minimum length 1 character. Maximum length 10 characters.	CC, RD
`ImmiCard Number`	string	Minimum length 1 character. Maximum length 9 characters.	IM
`Passport Number`	string	Minimum length 1 character. Maximum length 14 characters.	VI
`License Number`	string	Minimum length 1 character. Maximum length 10 characters.	DL
`Card Number`	string	Maximum length 10 characters.	DL, MD
`Individual Ref Number`	number	Value from 1 to 9.	MD

Document identifier string	Value data type	Value requirements	Used by document types
`Travel Document Number`	string	Minimum length 8 character. Maximum length 9 characters.	PP
`CRN`	string	Length MUST be 10 characters.	CO

4.3.4.2 Document Names Object

The Document Names Object field of a Verified Document Object is a JSON Object with keys and values determined by the parent objects document type.

The document type, held in the `type_code` claim, is mapped to an enumeration of permitted values in Table 58 above. The mapping does not imply the presence of a document name attribute is required for the specified verified document type. For a given document type, the availability of document name attributes is dependent upon the underlying identity document's issuing authority and when it was issued.

For a Document Names Object to be valid it MUST:

(a) only have keys that map to the parent object's Document Type field; and
(b) be compliant with value data types prescribed in Table 58 above for a given Name Attribute.

4.3.4.3 Document Attributes Object

The Document Attributes field of a Verified Document Object is a JSON Object with keys and values determined by the parent objects document type.

The document type, held in the `type_code` claim, is mapped to an enumeration of permitted values in Table 60 below. The mapping does not imply the required presence of a document attribute, for the specified verified document type, for the payload to be valid. For a given document type, the availability of document attributes is dependent upon an identity document's issuing authority and when it was issued.

For a Document Identifiers Object to be valid it MUST:

- only comprise keys that map to the parent object's Document Type field; and
- have values that are compliant with data types prescribed in Table 60 below for a given identifier.

Table 60 Verified Document Attributes with data types

Document attribute string	Value data type	Value requirements	Used by document types
`Registration State`	string	An up to 3 letter abbreviation of the registration state name. See Table 13 above for accepted values.	BC, MC, NC
`Date of Event`	Date	DVS valid dates are between 1753-01-01 to 3000-12-31.	MC
`Acquisition Date`	Date	DVS valid dates are between 1753-01-01 to 3000-12-31.	CC, RD
`Country of Issue`	string	A non-empty string. DVS does not specify any additional constraints on the verified value.	VI
`State of Issue`	string	An up to 3 letter abbreviation of the registration state name. See Table 13 above for legal values.	DL
`Card Type`	string	Single character matching the enumeration in Table 14 above.	MD
`Card Expiry`	string	Medicare Card expiry values are determined by the Card Type. See Table 15 above for patterns.	MD
`Full Name 1`	string	Maximum length 27 characters. Line 1 of the Medicare Card full name.	MD

Document attribute string	Value data type	Value requirements	Used by document types
`Full Name 2`	string	Maximum length 25 characters. Line 2 of the Medicare Card full name.	MD
`Full Name 3`	string	Maximum length 23 characters. Line 3 of the Medicare Card full name.	MD
`Full Name 4`	string	Maximum length 21 characters. Line 4 of the Medicare Card full name.	MD
`Gender`	string	Single character enumeration of the gender value the travel document. See Table 16 above for accepted values.	PP
`CardType`	string	3-character enumeration of the concession card type. Set Table 17 above for accepted values.	CO
`CardExpiry`	Date	DVS valid dates are between 1753-01-01 to 3000-12-31	CO

4.4 Identity System Metadata

4.4.1 Common

OpenID Connect Core 1.0 data representation for the common attribute sets defined in section 2.2.1 of this Schedule.

Table 61 Common attributes definition

Claim	AGDIS Data Type	Constraints
`sub`	string	Maximum length 255 characters.
`auth_time`	Unix Timestamp	-
`acr`	URN or Array of URNs	See section 2.2.1.3 of this Schedule.
`amr`	URN	Must be one of the values outlined in Table 30 above.
`updated_at`	Unix Timestamp	-

4.4.1.1 Subject ID

Commonly referred to as the subject identifier or subject ID the `sub` claim's data definition is outlined in OpenID Connect attribute found in Table 47 above.

ISPs and IXPs assigned subject IDs MUST be conform to the requirements outlined in section 1.2 of Schedule 1 (AGDIS Onboarding Specifications).

4.4.1.2 Authentication Time

An ISP MUST implement support for authentication time, the `auth_time` claim, as outlined in Table 47 above.

4.4.1.3 Assurance Level

When making, brokering, or fulfilling request for an assurance level, in compliance with the role specific requirements outlined in Schedule 2 (AGIDS OpenID Connect Profile), all entities participating the AGDIS (including PRPs) MUST submit ACR values conforming to the URNs enumerated in Table 29 above.

4.4.1.4 Authentication Method

When fulfilling the Authentication Method claim an IXP MUST use URN values conformant with the values outlined in Table 30 above to communicate the ISP chosen and authenticated at by the individual.

The addition of other values Internet Assigned Numbers Authority registered in line with the specification for the Authentication Method Reference response are permitted.

4.4.1.5 Last Updated

An ISP MUST implement support for this claim as outlined in Table 44 above using a Unix Timestamp.

4.4.2 Audit

The claim outlined in this section is for the RP Audit ID only.

Audit and transaction identifiers for the ISP to IXP transactions are left to the discretion of IXPs and accordingly not defined here.

Table 62 Audit attribute definitions

Claim	AGDIS data type	Constraints
`tdif_audit_id`	UUID	-

4.4.2.1 RP Audit Identifier

An IXP MUST implement RP Audit ID, the `tdif_audit_id` claim, as outlined in section 2.2.2 of this Schedule.

4.5 Self-Asserted Attributes

Assumed Self-Asserted Attributes are not mandatory under this profile, however if an ISP supports these attributes, they MUST implement the definitions outlined in Table 63 below to ensure interoperability.

Table 63 Self-Asserted Attributes types

Claim	AGDIS Data Type	Constraints
`preferred_username`	NameString	Section 2.1.1, section 2.3.1 of this Schedule. Non-empty string with a maximum length 100 characters.
`agdis_address`	Address Object or Array of Address Objects	Section 2.3.2, section 4.5.2 of this Schedule.
`agdis_other_email`	Other Email Object or Array of Other Email Objects	Section 2.3.3, section 4.5.3 of this Schedule.
`agdis_other_phone`	Other Phone Object or Array of Other Phone Objects	Section 2.3.4, section 4.5.4 of this Schedule.
`agdis_birthplace`	String	Section 2.3.5, section 4.5.5 of this Schedule. Non-empty string with a maximum length 255 characters.
`agdis_personal_title`	String	Section 2.3.6, section 4.5.6 of this Schedule. Non-empty string with a maximum length of 100 characters.

4.5.1 Preferred Name

The Preferred Name follows the same data definitions for name claims in the Core Attribute set.

If fulfilled it MUST be a non-empty string with a maximum length of 100 characters.

4.5.2 Addresses

The Address claim MAY represent one or more Address Objects.

A request for a single address MAY be returned as a single Address object or as a JSON Array containing a single entry.

A request for one or more addresses MUST be returned as a JSON array of Address Objects.

An attribute requests for multiple or a specific address type can be done via the claim requests parameter using the Address Type URNs in Table 33 above.

A request MAY include the default address along with other address types.

The example request is illustrated in Figure 2 will return the default address and the residential address, if they are not the same value.

4.5.3 Other Email Addresses

The Other Email Addresses scope MUST only return the default email address nominated by the individual.

The Other Email Address claim MAY represent one or more Other Email Address Objects.

A request for a single other email address MAY be returned as a single Other Email Address object or as a JSON Array containing a single entry.

A request for one or more other email addresses MUST be returned as a JSON array of Other Email Address Objects.

An attribute requests for multiple or a specific other email address type can be done via the claim requests parameter using the Contact Type URNs defined in Table 36 above. An example request is illustrated in Figure 3.

A request MAY include the default contact type along with other contact type URNs.

```
{
    …,
    “userinfo”: {
        …,
        “agdis_address”: {
            “values”: [
                “urn:id.gov.au:agdis:address:default”,
                “urn:id.gov.au:agdis:address:residential”
            ]
        },
        …
    },
    …
}
```

Figure 2 Self-asserted address sample claim request

4.5.4 Other Phone Numbers

The Other Phone Numbers scope request MUST only return the default other phone number nominated by the individual.

The Other Phone Numbers claim MAY represent one or more Other Phone Number Objects.

A request for a single other phone number MAY be returned as a single Other Phone Number object or as a JSON Array containing the object as its single entry.

A request for one or more other phone numbers MUST be returned as a JSON array of Other Phone Number Objects.

An attribute requests for multiple or a specific other phone number types can be done via the claim requests parameter using the Telephony (Table 34) or Contact Type (Table 35) URNs. An example request is illustrated in Figure 3.

```
{
    …,
    “userinfo”: {
        …,
        “agdis_email_address”: {
            “values”: [
                “urn:id.gov.au:agdis:contact:personal”,
                “urn:id.gov.au:agdis:contact:work”
            ]
        },
        …
    },
    …
}
```

Figure 3 Self-asserted other email address sample claim request

```
{
    …,
    “userinfo”: {
        …,
        “agdis_email_address”: {
            “values”: [
                “urn:id.gov.au:agdis:contact:default”,
            ]
        },
        …
    },
    …
}
```

Figure 4 Self-asserted other email address scope equivalent claim request

```
{
    …,
    “userinfo”: {
        …,
        “agdis_other_phone”: {
            “values”: [
                “urn:id.gov.au:agdis:contact:personal”,
                “urn:id.gov.au:agdis:contact:work”
            ]
        },
        …
    },
    …
}
```

Figure 5 Self-asserted other phone number sample claim request using the contact type

```
{
    …,
    “userinfo”: {
        …,
        “agdis_other_phone”: {
            “value”: “urn:id.gov.au:agdis:tel:video”,
        },
        …
    },
    …
}
```

Figure 6 Self-asserted other phone number sample claim request using telephony type

```
{
    …,
    “userinfo”: {
        …,
        “agdis_other_phone”: {
            “value”: “urn:id.gov.au:agdis:tel:default”,
        },
        …
    },
    …
}
```

Figure 7 Self-asserted other phone number scope equivalent claim request

4.5.5 Place of Birth

Place of Birth MUST be a non-empty string with a maximum length of 255 characters.

4.5.6 Personal Titles

A personal title MUST be a non-empty string with a maximum length of 100 characters.

4.6 Computed Attributes Data Definitions

Presently no Computed Attributes are defined in this Schedule.

4.7 Attribute Service Providers

The OpenID Connect attribute for AGIS ASP is presented in this section.

4.7.1 Business Authorisations

The business authorisations attribute set mapping in Table 64 below maps the field names defined in section 3.1 of this Schedule to concrete payload names with type definitions.

As the ASP providing these attributes RAM SHOULD provide guidance to PRPs consuming these attributes to support use.

Table 64 Business Authorisation data representation

Field	OIDC payload field name	Data type	Constraints
Schemas	`schemas`	Arry of URNs	MUST contain a URN that maps to the schema version follow the format defined in Table 39 above.
Unique Entity Relationship ID	`id`	string	A unique value as a non-empty string with a format defined by RAM, and a maximum length of 256 characters.

Field	OIDC payload field name	Data type	Constraints
Entity ID	`subjectId`	string	A JSON string that MUST adhere the lengths constraints of the entity type.
Entity Type	`subjectIdType`	string	A value from the enumeration defied in Table 41 above.
Entity Name	`subjectName`	string	A non-empty string with a maximum length of 200 character.
Contact Details	`email`	Email	Only a single email address is provided.
Relationship Type	`relationshipType`	string	A non-empty string.
Relationship Start Time	`startTimestamp`	DateTime	-
Relationship End Time	`endTimestamp`	DateTime	-
Attributes	`attributes`	JSON Object	A JSON object of objects with the fields `name` and `value`. The literals use for the names field, and data types for values are defined by RAM.
Roles	`roles`	Array of strings	Values defined by RAM.
Entitlements	`permissions`	Array of strings	Values defined by RAM
Attributes Last updated	`lastModified`	DateTime	-

4.7.2 myGov

Detailed definition of the data representation for the myGov Link ID is not defined in this Schedule.

An IXP or a PRP seeking further information on the format of this attribute SHOULD consult the myGov member service documentation.

4.8 Normative OIDC Profile Attribute Examples

4.8.1 Core

Each Core Attribute example is presented on its own. In practice these values would be presented in the ID Token or UserInfo response.

Table 65 Core Attributes normative examples for claims

Attribute	OIDC Claim	Normative Example
Family Name Full Name	`family_name` `name`	Valid: • `“family_name”: “Moore”` • `“name”: “John David Citizen”` Invalid • `“family_name”: “”` • `“family_name”: <string longer than 100 characters>` • `“family_name”: null`

Given Name
Middle Name
Preferred Name
	`given_name`
	`middle_name`
	`preferred_username`
		Valid:
		• `“given_name”: “Trentino”`
		• `“given_name”: “”`
		Invalid:
		• `“given_name”: <string longer than 100 characters>`
		• `“given_name”: null`

Attribute	OIDC Claim	Normative Example
Date of birth	`birthdate`	Valid: - `“birthdate”: “1970-04-01”` - `“birthdate”: “1970-04”` - `“birthdate”: “1970”` Invalid: - `“birthdate”: 1970` - `“birthdate”: “70-04-01”` - `“birthdate”: “1984-30-04”`
Updated at claims	`updated_at` `tdif_core_updated_at`	Valid: - `“tdif_core_updated_at”: 1674539150` - `“tdif_other_names_updated_at”: 1674539150` Invalid: - `“tdif_core_updated_at”: “1674539150”` - `“tdif_core_updated_at”: “2022-03-19 00:03:33.33”`

4.8.2 Validated Contact Details

Note grouped examples claims are where the data types are the same. The examples provided for the group claims may be used interchangeably.

Table 66 Validated contact details normative examples for claims

Attribute	OIDC Claim	Normative Example
Validated Email	`email`	Valid: • `“email”: “jane.citizen@example.com”` Invalid • `“email”: null` • `“email”: “malformed.email.address”`
Validated Phone	`phone_number`	Valid: • `“given_name”: “Trentino”` • `“given_name”: “”` Invalid: • `“given_name”: <string longer than 100 characters>` • `“given_name”: null`

Attribute	OIDC Claim	Normative Example
Email verified Phone number verified	`email_verified` `phone_number_verified`	Valid: • `“email_verified”: true` • `“phone_number_verified”: true` Invalid: • `“email_verified”: false` • `“email_verified”: “true”` • `“email_verified”: null` • `“phone_number_verified”: false` • `“phone_number_verified”: “true”` • `“phone_number_verified”: null`
Email updated at Phone updated at	`tdif_email_updated_at` `tdif_phone_number_updated_at`	Valid: • `“tdif_email_updated_at”: 1674539151` • `“tdif_phone_number_updated_at”: 1674539150` Invalid (not the claims should be interpreted interchangeably): • `“tdif_email_updated_at”: “1674539151”` • `“tdif_phone_number_updated_at”: “1674539150”` • `“tdif_email_updated_at”: “2024-04-01T00:00.00Z”` • `“tdif_email_updated_at”: “”` • `“tdif_phone_number_updated_at”: null` • `“tdif_email_updated_at”: “2024-04-01T00:00.00Z”` • `“tdif_email_updated_at”: “2024-04-01T00:00.00Z”`

Table 67 Validated contact details normative examples for scopes

Attribute	OIDC Scope	Normative Example

Validated Email	`email`	Each example presented here is a subset of the top-level fields in the JWT response.
		Valid
		All fields present and with legal values.
		`{`
		`    …`
		`    “email”: “jane.citizen@example.com”,`
		`    “email_verified”: true,`
		`    “tdif_email_updated_at”: 956386037`
		`    …`
		`}`
		Invalid
		Email verified MUST NOT be false:
		`{`
		`    …`
		`    “email”: “jane.citizen@example.com”,`
		`    “email_verified”: false,`
		`    “tdif_email_updated_at”: 956386037`
		`    …`
		`}`
		The updated at field is not a Unix Timestamp.

Attribute	OIDC Scope	Normative Example

		{
		    …
		    “email”: “jane.citizen@example.com”,
		    “email_verified”: true,
		    “tdif_email_updated_at”: “2024-01-01T00:00Z”
		    …
		}
		Missing fields:
		{
		    …
		    “email”: “jane.citizen@example.com”,
		    “email_verified”: true,
		    …
		}

Validated Phone	phone	**Valid**
		All fields present and have legal values.
		{
		    …
		    “phone_number”: “jane.citizen@example.com”,
		    “phone_number_verified”: True,
		    “tdif_email_updated_at”: 956386037

Attribute	OIDC Scope	Normative Example

		…
		 }
		 Invalid
		 Phone number verified MUST NOT be false:
		 {
		 …
		 “phone_number”: “jane.citizen@example.com”,
		 “phone_number_verified”: false,
		 “tdif_email_updated_at”: 956386037
		 …
		 }
		 The updated at field is not a Unix Timestamp.
		 {
		 …
		 “phone_number”: “jane.citizen@example.com”,
		 “phone_number_verified”: true,
		 “tdif_phone_numner_updated_at”: “2024-01-01T00:00Z”
		 …
		 }
		 Missing fields (e.g., the last updated field):

Attribute	OIDC Scope	Normative Example

		{
		    …
		    “phone_number”: “jane.citizen@example.com”,
		    “phone_number_verified”: true,
		    …
		}

4.8.3 Verified Other Names

Table 68 Verified other names normative examples for claims

Attribute	OIDC Claim	Normative Example

Verified Other Names	`tdif_other_names`	**Valid**
		Single Other Verified Name available (no array):
		`“tdif_other_names”: {`
		`    “family_name”: “Moore”,`
		`    “given_name”: “Trentino”`
		`}`
		Valid
		Single Other Verified Name available in an array:
		`“tdif_other_names”: [`
		`   {“family_name”: “Moore”, “given_name”: “Trentino” }`
		`]`

Attribute	OIDC Claim	Normative Example

		Valid:
		Multiple Other Verified Name available:
		`“tdif_other_names”: {`
		`    {“family_name”: “Moore”, “given_name”: “Trentino”},`
		`    {“family_name”: Moore”, “given_name”: “Trentino Vino”}`
		`}`
		Multiple Other Verified Name available:
		`“tdif_other_names”: [{`
		`    “family_name”: “Vass”,`
		`    “middle_name”: “Steven”,`
		`    “given_name”: “Ahgan”`
		`}, {`
		`    “family_name”: “Vass”,`
		`    “given_name”: “Ahgan”`
		`}]`

Attribute	OIDC Claim	Normative Example

		Invalid
		Family MUST NOT be an empty value:
		`“tdif_other_names”: [{`
		`    “family_name”: “”, “given_name”: “Ahgan”`
		`}]`
		Invalid
		`null` values are not permitted:
		`“tdif_other_names”: [{`
		`    “family_name”: “Vass”, “given_name”: null`
		`}]`

4.8.4 Verified Documents

4.8.4.1 Birth Certificate

Table 69 Birth certificate verified document claim normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:BC",
			"verification_method": "S",
			"verification_date": "2022-12-06T23:21:19.0231031Z",
			"names": {
			"family_name": "Maximus",
			"given_name": "Michael"
			},

verification_method		Y	
verification_date		Y	
names[1]	family_name	Y	
	given_name	Y	
birthdate		Y	
identifiers[2]	Registration Number	N	
	Certificate Number	N	

[1] On birth certificates issued by the NSW, VIC, WA BDM, given name may be left blank if family name has a value, and family name may be left blank if given name has a value.

[2] The identifiers used for birth certificates will vary significantly depending on the state and territory which issued the birth certificate.

Field	Sub-field(s)	Required	Example

attributes	Registration Date	N	"birthdate": "1989-01-03",
			"identifiers": [
			{
			"type": "Registration Number",
			"value": "1234567890"
			}
			],
			"attributes": [
			{
			"type": "Registration State",
			"value": "ACT"
			}
			]
			}

	Registration State	Y	
	Registration Year	N	

4.8.4.2 Centrelink Concession Card

Table 70 Centrelink concession card verified document claim normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:CO",
			"verification_method": "S",
			"verification_date": "2022-12-06T23:21:19.0231031Z",
			"names": {
			"name": "Jane Citizen"
			},
			"birthdate": "1976-02-15",
			"identifiers": [
			{
			"type": "CRN",
			"value": "1234567890"

verification_method		Y	
verification_date		Y	
names	name	Y	
birthdate		N	
identifiers	CRN	Y	
attributes	CardExpiry	Y	
	CardType	Y	

Field	Sub-field(s)	Required	Example

			`        }`
			`    ],`
			`    "attributes": [`
			`        {`
			`            "type": "CardExpiry ",`
			`            "value": "2024-03"`
			`        }, {`
			`            “type”: “CardType”,`
			`            “value”: ”PCH”,`
			`        }`
			`    ]`
			`}`

4.8.4.3 Change of Name Certificate

Please see footnotes 1 and 2 to clarify name handling.

Table 71 Change of Name Certificate verified document claim normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:NC",
			"verification_method": "S",
			"verification_date": "2022-12-07T00:52:48.2Z",
			"names": {
			"family_name": "Southsoil",
			"given_name": "Jakub"
			},
			"birthdate": "1992-12-07",

verification_method		Y	
verification_date		Y	
names[3]	family_name	Y	
	given_name	Y	
birthdate		N	
identifiers	Registration Number	N	
	Certificate Number	N	
attributes	Registration	N	

[3] For Change of Name Certificates issued by the NSW, VIC or WA BDM, given name may be left blank if family name has a value, and family name may be left blank if given name has a value.

Field	Sub-field(s)	Required	Example

	Date		"identifiers": [
			{
			"type": "Registration Number",
			"value": "1563160"
			}
			],
			attributes": [
			{
			"type": "Registration State",
			"value": "ACT"
			}
			]
			}

	Registration State	Y	
	Registration Year	N	

4.8.4.4 Citizenship Certificate

Table 72 Citizenship Certificate verified document claim normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:CC",
			"verification_method": "S",
			"verification_date": "2022-12-06T23:22:01.6116772Z",
			"names": {
			"family_name": "Doe",
			"given_name": "John"
			},
			"birthdate": "1986-04-18",
			"identifiers": [

verification_method		Y	
verification_date		Y	
names	family_name	Y	
	given_name	N	
birthdate		Y	
identifiers	Stock Number	Y	
attributes[4]			

[4] Please note that the Acquisition Date field, while supported by the DVS, is not currently available on the AGDIS.

Field	Sub-field(s)	Required	Example

			`        {`
			`            "type": "Stock Number",`
			`            "value": "ACC1000112"`
			`        }`
			`    ],`
			`    "attributes": [],`
			`}`

4.8.4.5 Registration by Descent Certificate

Table 73 Registration by Descent Certificate verified document normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:RD",
			"verification_method": "S",
			"verification_date": "2022-12-06T23:22:01.6116772Z",
			"names": {
			"family_name": "Citizen",
			"given_name": "Birdy"
			},
			"birthdate": "1986-04-18",
			"identifiers": [

verification_method		Y	
verification_date		Y	
names	family_name	Y	
	given_name	N	
birthdate		Y	
identifiers	Stock Number	Y	
attributes[5]			

[5] Please note that the Acquisition Date field, while supported by the DVS, is not currently available on the AGDIS.

Field	Sub-field(s)	Required	Example

			{
			"type": "Stock Number",
			"value": "ACC1000342"
			}
			],
			"attributes": [
			{
			“type”: “Acquisition Date”,
			“value”: “2013-04-20”
			}
			],
			}

4.8.4.6 Australian Driver License

Table 74 Australian Driver License verified document normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:DL",
			"verification_method": "S",
			"verification_date": "2022-12-06T23:22:01.6116772Z",
			"names": {
			"family_name": "Bisan",
			"given_name": "Wizard"
			},
			"birthdate": "1986-04-18",

verification_method		Y	
verification_date		Y	
names	family_name	Y	
	given_name	Y	
	middle_name	N	
birthdate		Y	
identifiers	Licence Number	Y	
	Card Number[6]	N	
attributes	State of Issue	Y	

[6] Card Number is required for a NSW, NT, SA, WA, Tas and ACT drivers licence verified after the 1st of September 2022. If it is recorded by an IDP, it is expected to be passed as part of a driver licence tdif_doc.

Field	Sub-field(s)	Required	Example

			"identifiers": [
			{
			"type": "Licence Number",
			"value": "65493621"
			}
			],
			"attributes": [
			{
			“type”: “State of Issue”,
			“value”: “QLD”
			}
			]
			}

4.8.4.7 ImmiCard

Table 75 ImmiCard verified document normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:IM",
			"verification_method": "S",
			"verification_date": "2022-12-06T23:22:01.6116772Z",
			"names": {
			"family_name": "Moore",
			"given_name": "Trentino Bici"

verification_method		Y	
verification_date		Y	
names	family_name	Y	
	given_name	Y	
birthdate		Y	
identifiers	ImmiCard Number	Y	
attributes	*empty*	Y	

Field	Sub-field(s)	Required	Example

			`    },`
			`    "birthdate": "1986-04-18",`
			`    "identifiers": [`
			`        {`
			`            "type": "ImmiCard Number",`
			`            "value": "PRE123456"`
			`        }`
			`    ],`
			`    "attributes": []`
			`}`

4.8.4.8 Marriage Certificate

Table 76 Marriage Certificate verified document normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:MC",
			"verification_method": "S",
			"verification_date": "2022-12-06T23:21:19.0231031Z",
			"names": {
			"family_name": "O’Keeffe",
			"given_name": "Mickey",
			"family_name2": "Louis",
			"given_name2": "Jesse"
			},
			"birthdate": "1989-01-03",
			"identifiers": [

verification_method		Y	
verification_date		Y	
names	family_name	Y	
	given_name	Y	
	family_name2	Y	
	given_name2	Y	
birthdate		Y	
identifiers	Registration Number	N	
	Certificate Number	N	
attributes	Registration Date	N	

Field	Sub-field(s)	Required	Example

	Registration State	Y	`{`
			`"type": "Registration Number",`
			`"value": "1234567890"`
			`}`
			`],`
			`"attributes": [`
			`{`
			`"type": "Registration State",`
			`"value": "ACT"`
			`}`
			`]`
			`}`

	Registration Year	N	

4.8.4.9 Medicare Card

Table 77 Medicare Card verified documents normative example

Field	Sub-field(s)	Required	Example
type_code		Y	{ "type_code": "urn:id.gov.au:tdif:doc:type_code:MD", "verification_method": "S", "verification_date": "2022-12-06T23:21:19.0231031Z", "birthdate": "1989-01-03", "identifiers": [{ "type": "Card Number", "value": "1234567890" }, { "type": "Individual Ref Number",
verification_method		Y	
verification_date		Y	
birthdate		Y	
identifiers	Card Number	Y	
	Individual Ref Number	Y	
attributes	Card Type	Y	
	Card Expiry	Y	
	Full Name 1	Y	
	Full Name 2	Y	
	Full Name 3	Y	
	Full Name 4	Y	

Field	Sub-field(s)	Required	Example

			`        "value": "1"`
			`    },`
			`],`
			`"attributes": [`
			`    {`
			`        "type": "Card Type",`
			`        "value": "G"`
			`    },`
			`    {`
			`        "type": "Card Expiry",`
			`        "value": "2022-12"`
			`    },`
			`    {`
			`        "type": "Full Name 1",`

Field	Sub-field(s)	Required	Example
			"value": "Lachlan Murramarang" }, { "type": "Full Name 2", "value": null }, { "type": "Full Name 3", "value": null }, { "type": "Full Name 4", "value": null },

Field	Sub-field(s)	Required	Example
			] }

4.8.4.10 Australian Travel Document

Table 78 Australian Travel Document verified document normative example

Field	Sub-field(s)	Required	Example

`type_code`		Y	`{`
			`    "type_code": "urn:id.gov.au:tdif:doc:type_code:PP",`
			`    "verification_method": "S",`
			`    "verification_date": "2022-12-06T23:22:01.6116772Z",`
			`    "names": {`
			`        "family_name": "Mason",`
			`        "given_name": "Master"`
			`    },`
			`    "birthdate": "1930-04-18",`
			`    "identifiers": [`
			`        {`

`verification_method`		Y	
`verification_date`		Y	
`names`	`family_name`	Y	
	`given_name`	N	
`birthdate`		Y	
`identifiers`	`Travel Document Number`	Y	
`attributes`	`Gender`	N	

Field	Sub-field(s)	Required	Example

			"type": "Travel Document Number",
			"value": "PA0002020"
			}
			],
			"attributes": [
			{
			“type”: “Gender”,
			“value”: “Female”
			}
			]
			}

4.8.4.11 Visa

Table 79 Visa verified documents normative example

Field	Sub-field(s)	Required	Example

type_code		Y	{
			"type_code": "urn:id.gov.au:tdif:doc:type_code:VI",
			"verification_method": "S",
			"verification_date": "2022-12-06T23:22:01.6116772Z",
			"names": {
			"family_name": "Vino",
			"given_name": "Trentino"

verification_method		Y	
verification_date		Y	
names	family_name	Y	
	given_name	Y	
birthdate		Y	
identifiers	Passport Number	Y	
attributes[7]			

[7] Please note that the country of issue field, while supported by the DVS, is not currently available on the AGDIS.

Field	Sub-field(s)	Required	Example

			`    },`
			`    "birthdate": "1962-08-10",`
			`    "identifiers": [`
			`        {`
			`            "type": "Passport Number",`
			`            "value": "NN123456"`
			`        }`
			`    ],`
			`    "attributes": []`
			`}`

4.8.5 Identity System Metadata

4.8.5.1 Digital ID Pairwise Identifier

Table 80 Digital ID pairwise examples

Attribute	OIDC Claim	Normative Example
Digital ID pairwise identifier	`sub`	Valid: • `“sub”: “234932875989812”` • `“sub”: <a string up to 255 characters in length>` Invalid: • `“sub”: <a string longer than 255 characters>` • `“sub”: null` • `“sub”: 123145124`

4.8.5.2 Authentication Time and Updated At

Table 81 Authentication time and Update at examples

Attribute	OIDC Claim	Normative Example
Authentication Time Last Updated	`auth_time` `updated_at`	Valid: • `“auth_time”: 1711929600` • `“updated_at”: 1585699200` Invalid: • `“auth_time”: “1711929600”` • `“updated_at”: “20240401T00:00Z”`

4.8.5.3 Authentication Method

Table 82 Authentication Method examples

Attribute	OIDC Claim	Normative Example
Authentication Method Reference	`amr`	Valid: • `“amr”: [“urn:id.gov.au:idp:mygovid”]` Invalid: • `“amr”: “urn:id.gov.au:idp:mygovid”`

4.8.5.4 Audit Identifiers

Table 83 Audit Identifier examples

Attribute	OIDC Claim	Informative Example
RP Audit ID	`tdif_audit_id`	Valid: • `“tdif_audit_id”: “702b45ce-d2d6-451d-84cd-9fbcf299533b”` Invalid: • `“tdif_audit_id”: 4093809` • `“tdif_audit_id”: “”` • `“tdif_audit_id”: null`

4.8.6 Self-Asserted Attributes

4.8.6.1 Addresses

Table 84 Addresses normative examples

Attribute	OIDC Claim	Normative Example

Addresses	agdis_address	Valid all fields:
		{
		    “address_type”: “urn:id.gov.au:agdis:address:postal”,
		    “street_address”: “1 Canberra Ave”,
		    “locality”: “Forrest”,
		    “region”: “ACT”,
		    “post_code”: “2603”,
		    “country”: “Australia”,
		    “formatted”: “1 Canberra Ave\nForrest ACT 2603\nAustralia”,
		    “validated”: false
		}
		Valid only required fields:
		{
		    “address_type”: “urn:id.gov.au:agdis:address:postal”,
		    “street_address”: “1 Canberra Ave”,
		    “locality”: “Forrest”,
		    “region”: “ACT”,
		    “post_code”: “2603”,
		    “validated”: false
		}

4.8.6.2 Other Email Addresses

Table 85 Other Email Addresses

Attribute	OIDC Claim	Normative Example

Other Email Addresses	agdis_other_email	Valid:
		{
		“email”: “jane.cizten@exmaplebusiness.com.au”,
		“email_validated”: false,
		“contact_type”: “urn:id.gov.au:agdis:contact:business”
		}
		{
		“email”: “jane.cizten@exmaple.com.au”,
		“email_validated”: true,
		“contact_type”: “urn:id.gov.au:agdis:contact:other”
		}
		Invalid:
		{
		“email”: “not-a-valid-email”,
		“email_validated”: true,
		“contact_type”: “urn:id.gov.au:agdis:contact:other”
		}

4.8.6.3 Other Phone Numbers

Table 86 Other Phone Numbers normative examples

Attribute	OIDC Claim	Normative Example

Other Phone Numbers	`agdis_other_phone_numbers`	Valid individual claims:
		{
		    “email”: “jane.cizten@exmaplebusiness.com.au”,
		    “email_validated”: false,
		    “contact_type”: “urn:id.gov.au:agdis:contact:business”
		}
		{
		    “email”: “jane.cizten@exmaple.com.au”,
		    “email_validated”: true,
		    “contact_type”: “urn:id.gov.au:agdis:contact:other”
		}
		Invalid:
		{
		    “email”: “not-a-valid-email”,
		    “email_validated”: true,
		    “contact_type”: “urn:id.gov.au:agdis:contact:other”
		}

Table 87 Birthplace Normative Examples

Attribute	OIDC Claim	Normative Example
Birthplace	`agdis_birth_place`	Valid: • `“agdis_birth_place”: “Ngunnawal Country”` • `“agdis_birth_place”: <a non-empty string up to 100 characters in length>` Invalid: • `“agdis_birth_place”: “”` • `“agdis_birth_place”: null` • `“agdis_birth_place”: <a string over100 characters in length>`

4.8.6.4 Personal Title Normative Examples

Table 88 Personal Titles Normative Examples

Attribute	OIDC Claim	Normative Example
Personal title	`agdis_personal_title`	Valid: • `“agdis_personal_title”: “Professor”` • `“agdis_personal_title”: <a non-empty string up to 100 characters in length>` Invalid: • `“agdis_personal_title”: “”` • `“agdis_personal_title”: null` • `“agdis_personal_title”: <a string over100 characters in length>`

4.8.7 Business Authorisations

Table 89 Business Authorisations

Attribute	OIDC Scope and Claim	Normative Example

Business authorisations	tdif_business_authorisations	{
		    "schemas": [
		        "urn:id.gov.au:tdif:authorisations:business:1.0"
		    ],
		    "id": "78dfcb8c-dbb7-4c36-9807-70125dca90ca",
		    "subjectId": "49090058647",
		    "subjectIdType": "ABN",
		    "subjectName": "ALTONWAY LTD",
		    "relationshipType":"ASSOCIATE",
		    "startTimestamp":"2019-08-09T14:00:00Z",
		    "endTimestamp":null,
		     "attributes":[
		        {"name":"pid","value":"668"},
		        {"name":"subId","value":"ABRP:49090058647_668"},
		        {"name":"previousPid","value":null},
		        {"name":"previousSubId","value":null}
		    ],
		    "permissions":[
		        "QLD_IDENTITY_LOGIN/FULL"
		    ],
		    "roles":[
		        "AUTHORISATION_ADMINISTRATOR",
		        "MACHINE_CREDENTIAL_ADMINISTRATOR",
		        "PRINCIPAL_AUTHORITY"
		    ],
		    "email": "IndustryRDTI1@test.gov.au",
		    "lastModified": "2021-01-04T00:20:14.2453821Z"
		}

Australian Government

Australia's
Digital ID System

digitalidsystem.gov.au