Australian Government
Digital Transformation Agency

Technical Requirements

Trusted Digital Identity Framework
March 2019, version 1.0

dta

Digital Transformation Agency

Licence

Use of the Coat of Arms

Conventions

TDIF documents refenced by this document are denoted in italics. For example, *TDIF: Overview and Glossary* is a reference to the TDIF document titled Overview and Glossary.

The key words “**MUST**”, “**MUST NOT**”, “**SHOULD**”, “**SHOULD NOT**”, and “**MAY**” in this document are to be interpreted as described in the current version of the *TDIF: Overview and Glossary.*

Contact us

Digital Transformation Agency is committed to providing web accessible content wherever possible. If you are having difficulties with accessing this document, or have questions or comments regarding this document please email the Director, Digital Identity Policy at identity@dta.gov.au.

Document Management

The TDIF Accreditation Authority has reviewed and endorsed this document for release.

Change log

Version	Date	Author	Description of the changes
0.1	Mar 2018	TM	Initial version.
0.2	Apr 2018	TM	Edits following internal review.
0.3	Apr 2018	TM	Updates to align with updated identity proofing requirements.
0.4	Jun 2018	TM	Updates to align with updated authentication credential requirements.
0.5	Sep 2018	TM	Moved architecture overview to its own document.
0.6	Sep 2018	TM	Revised requirements for Attribute Providers.
0.7	Feb 2019	TM	Revision from consultation feedback for TDIF release 3.
1.0	Mar 2019		Endorsed for release by the TDIF Accreditation Authority.

Contents

1 Introduction

Agencies and organisations that apply to be accredited under the TDIF undergo a series of rigorous evaluations across all aspects of their identity service operations. The TDIF: Accreditation Process requires Applicants to demonstrate their identity service is usable, privacy enhancing and is secure and resilient to cyber threats. The intent of these evaluations is to determine whether the Applicant's identity service meets the TDIF Guiding Principles[1] and whether it is suitable to join the identity federation.

This document serves as the key reference for the technical requirements and technical integration standards for Accredited Providers in the identity federation.

The intended audience for this document includes:

- Applicants and Accredited Providers.
- Relying Parties.
- TDIF Accreditation Authority.

1.1 Purpose

The purpose of this document is to provide:

- Technical requirements that are common to all accredited entities in the federated identity eco-system.
- Technical requirements for each of the entities in the digital identity eco-system, either within this document or by providing a reference to relevant documents in the TDIF:
  - Identity Service Providers.
  - Credential Service Providers.
  - Attribute Providers.
  - Identity Exchange.

[1] See *TDIF: Overview and Glossary* for further information on the TDIF guiding principles.

1.2 Relationship to other TDIF documents

This document should be read in conjunction with the following TDIF documents:

- *TDIF: Overview and Glossary*, which provides a high-level overview of the TDIF including its scope and objectives, the relationship between its various documents and the definition of key terms.
- *TDIF: Architecture Overview,* which provides an architecture overview that describes the functions of the participants and how they interact with each other.
- The functional requirements for an Identity Service Provider are defined in the *TDIF: Identity Proofing Requirements.*
- The functional requirements for a Credential Service Provider are defined in the *TDIF: Authentication Credential Requirements.*
- *TDIF: Attribute Profile* specifies the attributes that are provided by an Identity Service Provider to a Relying Party via an Identity Exchange.
- *TDIF: OpenID Connect 1.0 Profile* specifies how the OpenID Connect 1.0 standards can be used to support authentication interactions.
- *TDIF: SAML 2.0 Profile* specifies how the SAML 2.0 standards can be used to support authentication interactions.

2 Technical Integration Standards Overview

2.1 Identity Federation Standards

The Trusted Digital Identity Framework enables the implementation of a national identity federation. This identity federation can support multiple Identity Exchanges that connect Relying Parties to Identity Service Providers using established and standardised federation protocols in an interoperable fashion. The currently supported federation protocols are OpenID Connect 1.0 (OIDC) and SAML 2.0 (SAML).

The following technical integration standards define how accredited entities interact to support the interactions in the federated digital identity eco-system.

- *TDIF: Attribute Profile* specifies the attributes that are be provided by an Identity Service Provider to a Relying Party via an Identity Exchange.

- *TDIF: OpenID Connect 1.0 Profile* specifies how the OpenID Connect 1.0 standards can be used to support authentication interactions.

- *TDIF: SAML 2.0 Profile* specifies how the SAML 2.0 standards can be used to support authentication interactions.

The following table notes the correspondence between the terminology used in the TDIF and the terms used to describe entities in these federation protocols.

TDIF Term	OIDC Term	SAML Term
Relying Party (RP)	Relying Party (RP)	Service Provider (SP)
Identity Service Provider (IdP)	OpenID Provider (OP)	Identity Service Provider (IdP)
Identity Service Provider (IdP)		

Figure 1: *Identity Federation Topologies.*

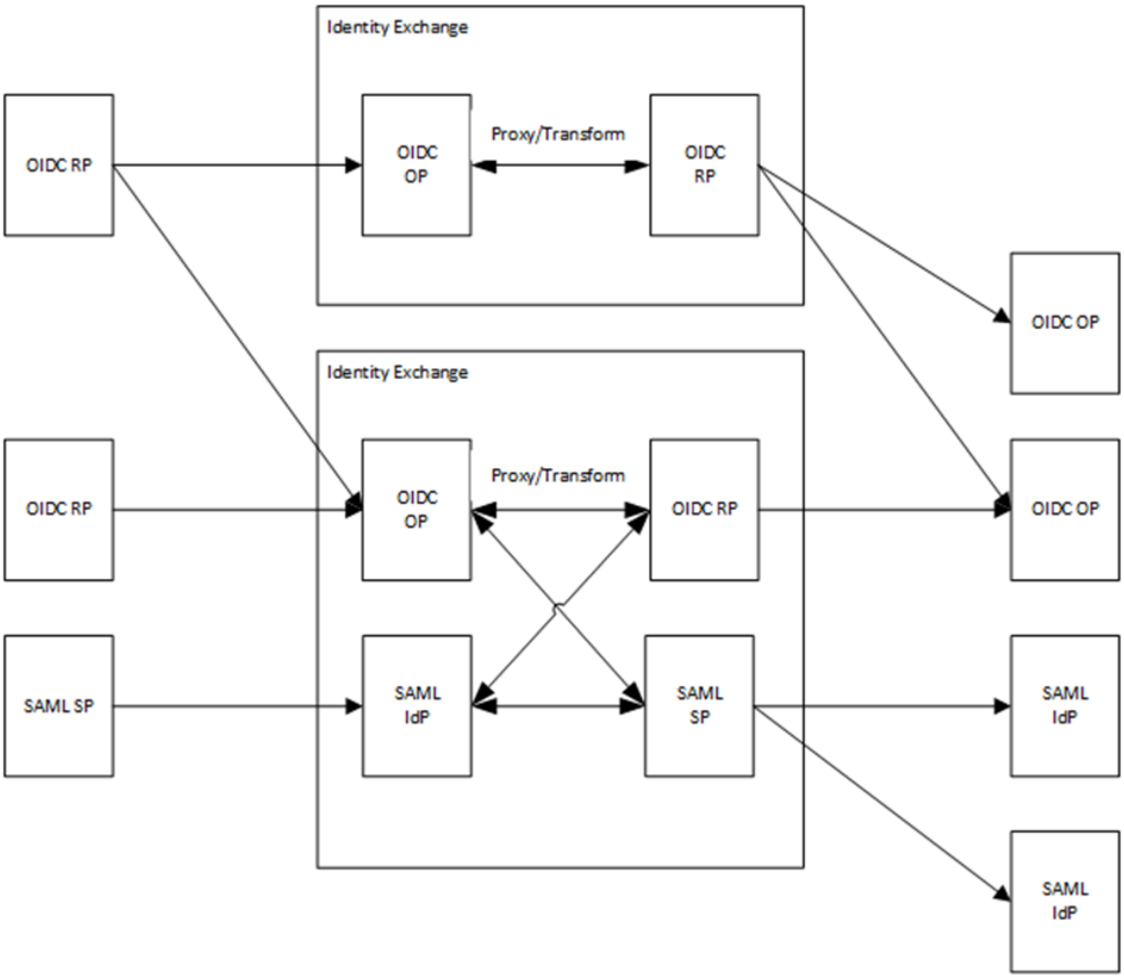


Figure 1 illustrates the possible identity federation topologies that exist in a mature identity eco-system. Digital services implemented that are relying on the identity federation can establish connections to any number of available Identity Exchanges that support their required federation protocol. These Identity Exchanges in turn can connect to any number of Identity Service Providers using their supported federation protocols.

The TDIF requires the presence of the Identity Exchange as a trusted broker between Relying Parties and Identity Service Providers. The Identity Exchange acts as an IdP Proxy from the perspective of a Relying Party. The Identity Exchange proxies the original request from a Relying Party to the user’s selected Identity Service Provider. Hence there are two hops in the interaction between a Relying Party and an Identity Service Provider via the Identity Exchange. Each of these hops is an instantiation of the federation protocols, with the Identity Exchange being responsible for maintaining the correspondence between the hops. In each of the hops the Identity Exchange acts as a different entity in terms of the federation protocols:

- Relying Party to Identity Exchange: The Identity Exchange acts as an Identity Service Provider.
- Identity Exchange to Identity Service Provider: The Identity Exchange acts as a Relying Party.

With more advanced Identity Exchanges, this process includes a translation in the federation protocol used. For example, a Relying Party connecting to an Identity Exchange using the SAML 2.0 federation has their requests serviced by the Identity Exchange performing a protocol translation to provide authentication from an Identity Service Provider that uses the OpenID Connect 1.0 federation protocol.

As well as providing the ability to proxy requests from a Relying Party to an Identity Service Provider chosen by the user, The Identity Exchange, as a trusted intermediary, enforces the attribute sharing policies required by the TDIF, such as any requirement for user consent.

2.2 Technical Integration Interactions

2.2.1 Base Authentication Interactions

Figure 2: *User Authentication Sequence Diagrams (steps 1 to 5).*

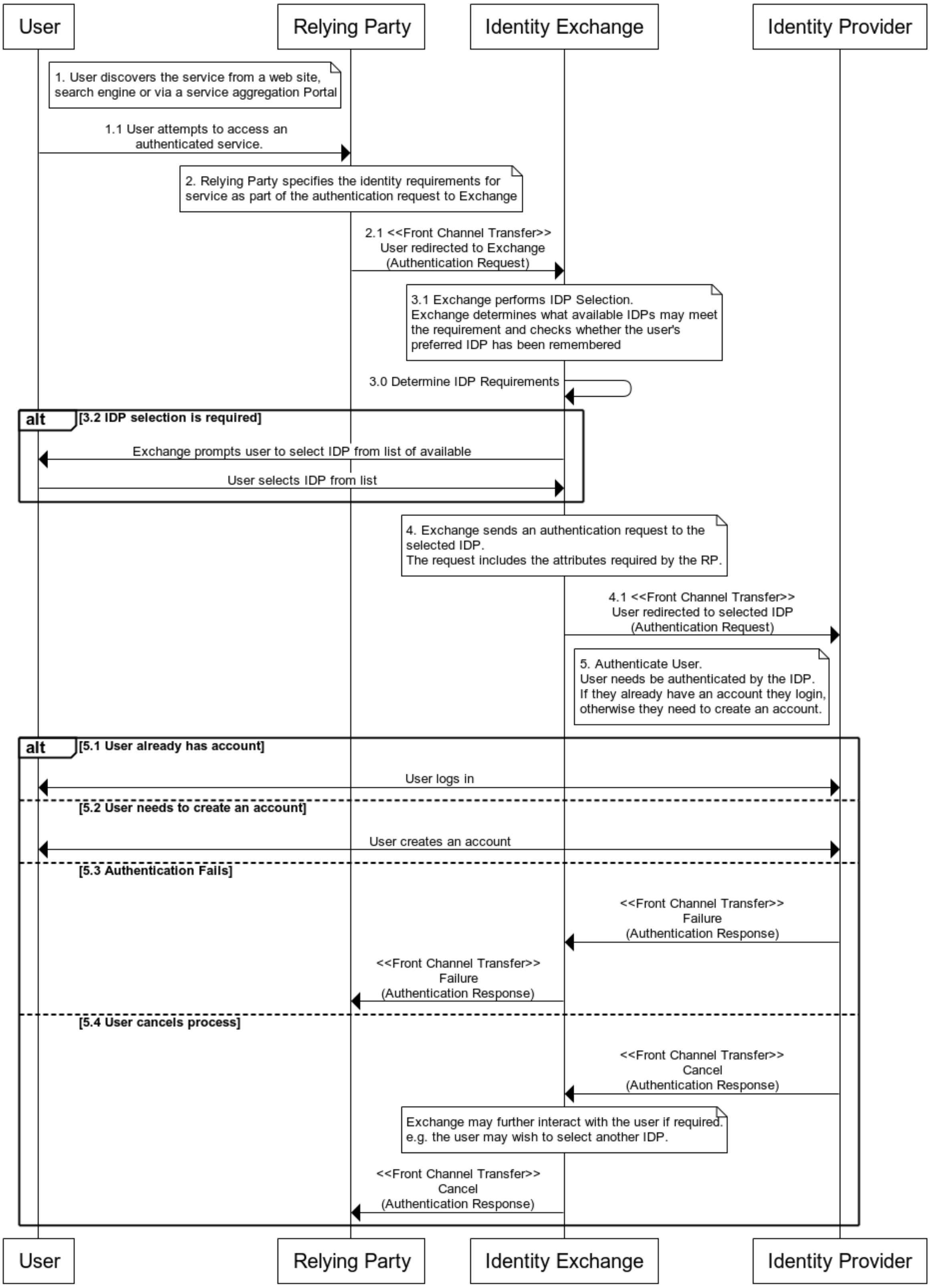

User
Relying Party
Identity Exchange
Identity Provider
1. User discovers the service from a web site, search engine or via a service aggregation Portal
1.1 User attempts to access an authenticated service.
2. Relying Party specifies the identity requirements for service as part of the authentication request to Exchange
2.1 <<Front Channel Transfer>> User redirected to Exchange (Authentication Request)
3.1 Exchange performs IDP Selection. Exchange determines what available IDPs may meet the requirement and checks whether the user's preferred IDP has been remembered
3.0 Determine IDP Requirements
alt
[3.2 IDP selection is required]
Exchange prompts user to select IDP from list of available
User selects IDP from list
4. Exchange sends an authentication request to the selected IDP. The request includes the attributes required by the RP.
4.1 <<Front Channel Transfer>> User redirected to selected IDP (Authentication Request)
5. Authenticate User. User needs be authenticated by the IDP. If they already have an account they login, otherwise they need to create an account.
alt
[5.1 User already has account]
User logs in
[5.2 User needs to create an account]
User creates an account
[5.3 Authentication Fails]
<<Front Channel Transfer>> Failure (Authentication Response)
<<Front Channel Transfer>> Failure (Authentication Response)
[5.4 User cancels process]
<<Front Channel Transfer>> Cancel (Authentication Response)
Exchange may further interact with the user if required. e.g. the user may wish to select another IDP.
<<Front Channel Transfer>> Cancel (Authentication Response)
User
Relying Party
Identity Exchange
Identity Provider

Figure 3: *User Authentication Sequence Diagrams (steps 6 to 11).*

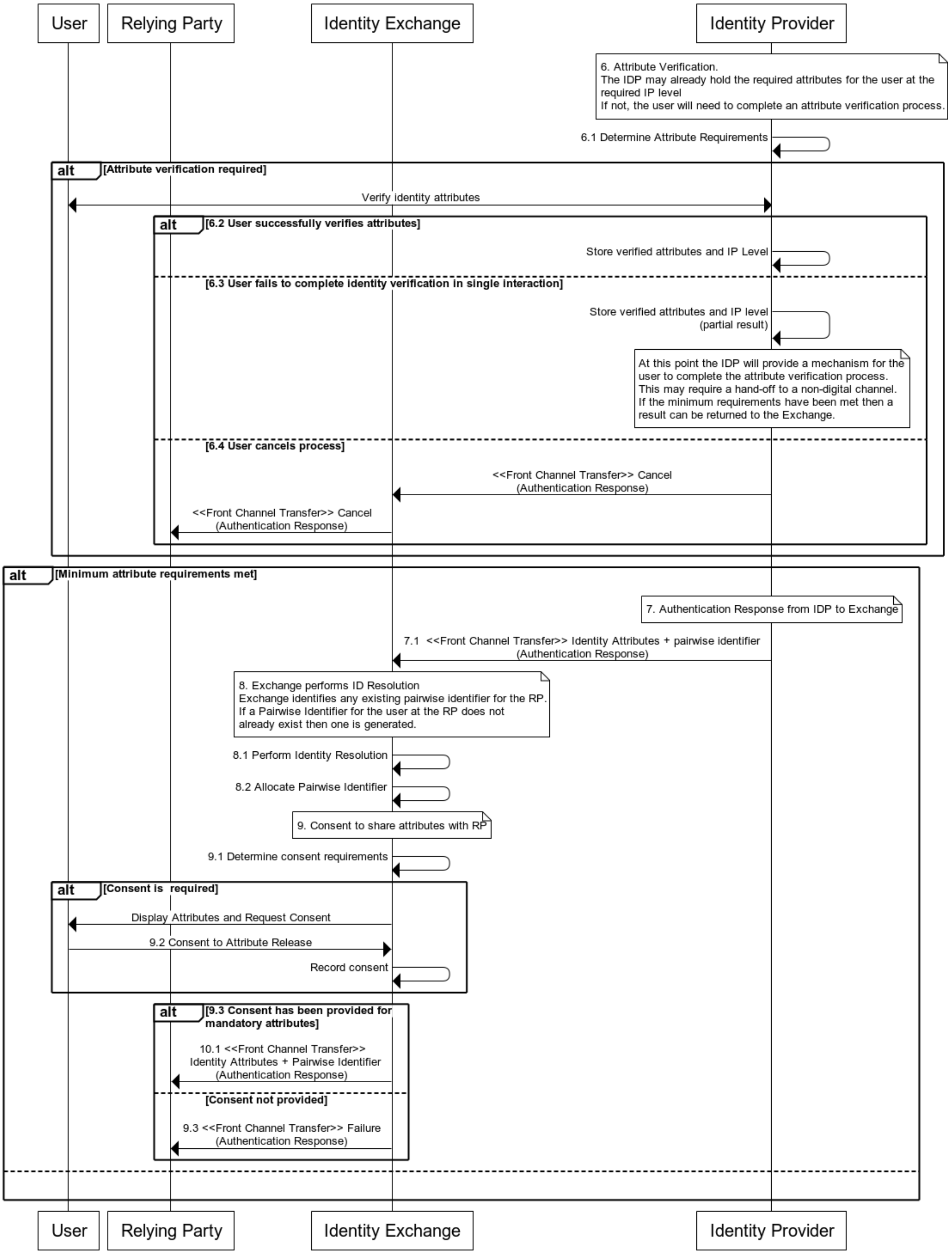

User
Relying Party
Identity Exchange
Identity Provider
6. Attribute Verification.
The IDP may already hold the required attributes for the user at the required IP level
If not, the user will need to complete an attribute verification process.
6.1 Determine Attribute Requirements
alt [Attribute verification required]
Verify identity attributes
alt [6.2 User successfully verifies attributes]
Store verified attributes and IP Level
[6.3 User fails to complete identity verification in single interaction]
Store verified attributes and IP level (partial result)
At this point the IDP will provide a mechanism for the user to complete the attribute verification process.
This may require a hand-off to a non-digital channel.
If the minimum requirements have been met then a result can be returned to the Exchange.
[6.4 User cancels process]
<<Front Channel Transfer>> Cancel (Authentication Response)
<<Front Channel Transfer>> Cancel (Authentication Response)
alt [Minimum attribute requirements met]
7. Authentication Response from IDP to Exchange
7.1 <<Front Channel Transfer>> Identity Attributes + pairwise identifier (Authentication Response)
8. Exchange performs ID Resolution
Exchange identifies any existing pairwise identifier for the RP.
If a Pairwise Identifier for the user at the RP does not already exist then one is generated.
8.1 Perform Identity Resolution
8.2 Allocate Pairwise Identifier
9. Consent to share attributes with RP
9.1 Determine consent requirements
alt [Consent is required]
Display Attributes and Request Consent
9.2 Consent to Attribute Release
Record consent
alt [9.3 Consent has been provided for mandatory attributes]
10.1 <<Front Channel Transfer>> Identity Attributes + Pairwise Identifier (Authentication Response)
[Consent not provided]
9.3 <<Front Channel Transfer>> Failure (Authentication Response)
User
Relying Party
Identity Exchange
Identity Provider

Figure 2 and Figure 3 are sequence diagrams that show the sequence of interactions for the authentication of a user. Each step in the diagram is described in detail below. These interactions are intended to illustrate the application of the technical interactions to an end-to-end user experience. Where is the user is transferred between entities via the user agent, e.g. web browser, the interaction is annotated with the `<<Front Channel Transfer>>` label. Each step in the diagram is described in detail below.

1. User discovers the digital service.

    1.1. User attempts to access an authenticated digital service.

    - The user discovers the digital service at a Relying Party. This can be from content on unauthenticated web site, a search engine, or from within a service aggregation portal.
    - The user accessing the service triggers the authentication process and any required verification of identity attributes occurs as part of this authentication process.
    - A user could initiate the attribute verification process independently of accessing a service by going directly to an Identity Service Provider.

2. Authentication Request from Relying Party to Exchange.

    2.1. User redirected to Exchange by the Relying Party using an authentication request.

    - The Relying Party specifies the identity requirements for the digital service as part of the authentication request. The request includes the required TDIF Assurance Levels and required identity attributes.
    - The Relying Party specifies the minimum assurance level that is required. The minimum assurance level is specified as mandatory or optional. If the specified minimum IP level is mandatory it must be reached for a successful authentication response to be returned to the Relying Party.

- The identity attributes are specified as optional or mandatory. If a mandatory attribute cannot be returned (not available or consent not provided) then the authentication response will be a failure.[2]

3. Identity Service Provider Selection.

3.1. The Identity Exchange determines the Identity Service Providers that will meet the requirements of the Authentication Request from the Relying Party. The Identity Exchange will determine what Identity Service Providers are available to meet the request. It will also check when a preferred Identity Service Provider for the user has been remembered.

3.2. If more than one Identity Service Provider is available then the user will be prompted to select an Identity Service Provider from a list. This selection may be remembered to streamline further interactions.

3.3. User Cancels Process. An Authentication Response indicating the cancellation of the process is sent back to the Relying Party.

4. Authentication Request from Identity Exchange to Identity Service Provider.

4.1. Exchange redirects the user to the selected Identity Service Provider using an authentication request. The request includes the attributes and assurance levels that were originally requested by the Relying Party.

5. Authenticate User. The user will either login to an existing account at the Identity Service Provider or create a new one.

5.1. User already has an account at the Identity Service Provider.

- The user logs into the Identity Service Provider using their existing credentials. If the existing credentials do not meet the required credential level the user will need to enrol additional credentials.

5.2. User does not have an account at the Identity Service Provider.

[2] The *TDIF: Attribute Profile* does not currently specify any attributes that may be requested as Mandatory by a Relying Party. In general, an authentication response should always be returned to the Relying Party as per the point above. This is consistent with the requests for claims in the OIDC standard, see https://openid.net/specs/openid-connect-core-1_0.html section 5.5.1.

- The user creates an account and is issued with credentials at the required credential level.

5.3. Authentication Fails.

- If the user fails to authenticate at the required credential level then an Authentication Response indicating the authentication failure is sent back to the Identity Exchange. The Identity Exchange then sends the same Authentication Response back to the Relying Party.

5.4. User Cancels Process.

- An Authentication Response indicating the cancellation of the process is sent back to the Identity Exchange. The Identity Exchange may interact with the user to determine if an alternate pathway is required to complete the process, e.g. to select a different Identity Service Provider. The Identity Exchange then sends the same Authentication Response back to the Relying Party if there is no identified alternate pathway.

6. Verify Attributes. The Identity Service Provider interacts with the user to verify the attributes at the required IP level, unless the Identity Service Provider already hold the attributes at the required IP level for the user.

6.1. Identity Service Provider determines attribute requirements.

- The Identity Service Provider checks the attributes already held for the user and determine if any further attribute verification is required. If attribute verification is required then steps 6.2 to 6.4 are possible paths.

6.2. User successfully verifies attributes.

- The user is able to successfully verify attributes at the required level.

6.3. The user is unable to complete the attribute verification process to the desired IP level in a single digital interaction.

- The Identity Service Provider will store the partial result and provide a process for the user to complete the attribute verification. This may require a hand-off to a non-digital channel. If the Relying Party originally specified a minimum IP level that has been met then a response can be returned to the Relying Party, otherwise this sequence of interactions end here.

6.4. User Cancels Process.

- An Authentication Response indicating the cancellation of the process is sent back to the Exchange. The Identity Exchange then sends the same Authentication Response back to the Relying Party if there is no identified alternate pathway.

7. Authentication Response is sent back to the Identity Exchange.

7.1. The Authentication Response from the Identity Service Provider includes:

- Achieved acr level.
- A pairwise identifier for the user at the Identity Service Provider.
- Identity attributes.

8. Exchange performs Identity Resolution.

- Identity Exchange identifies any existing pairwise identifier user at the Relying Party. If a pairwise Identifier for the user at the Relying Party does not already exist then one is generated.

8.1. Perform Identity Resolution.

- If a pairwise identifier is already mapped to the pairwise identifier from the Identity Service Provider then the Identity Exchange will use the pairwise identifier that is already allocated for the user.

8.2. Allocate Pairwise Identifier

- If required, a pairwise identifier is generated for the user. A pairwise identifier is an anonymous, unique identifier for the user at the Relying Party.

9. Consent to share attributes.

9.1. Determine consent requirements.

  - Identity Exchange determines the user consent requirements for the attributes requested by the Relying Party. It will include checking for any ongoing consent for sharing the attributes with the Relying Party.

9.2. Consent to Attribute Release.

  - If user consent is required, the Identity Exchange will interact with the user to gather consent to release the attributes to the Relying Party. The Identity Exchange will record the provided consent and the user's preference for remembering this consent.

9.3. Consent not provided.

  - If user consent is not provided then these attributes are not returned in the authentication response to the Relying Party.
  - If user consent is not provided for any mandatory attribute then a failure Authentication Response is returned to the Relying Party.[2]

10. Authentication Response to Relying Party.

10.1. Authentication Response is sent back to the Relying Party.

  - The Response includes:
    - Achieved acr level.
    - Pairwise identifier for user at the Relying Party.
    - Identity attributes for which consent has been provided.

11. User accesses digital service.

11.1. Relying Party uses the identity attributes to enable the user to access the digital service.

  - The first time the user accesses a Relying Party, the Relying Party may need to determine if there is an existing customer record by using the identity attributes as part of an Identity Matching process. Where a Relying Party performs Identity Matching, the Relying Party is responsible for ensuring that the matching process is sufficient to manage risks of authorised access to a person's record and is accountable for any privacy breach that may occur as a result of improper matching. Once a customer record has been located or

created at the Relying Party the Pairwise identifier is stored by the Relying Party, subsequent interaction by the user with the digital service will simply use the pairwise identifier to locate the customer record.
    - Note: some transactions are one-off and not require the above process.

2.2.2 Additional Authentication Interactions

2.2.2.1 Known Subject Authentication

Known subject authentication refers to authentication scenarios where the identity (service record) of the user at a Relying Party is already known. These authentication scenarios include:

- Relying Party clients where the identity of the user can be remembered, such as a mobile client.
- Scenarios where a previous authentication event has established the identity of the user at the Relying Party. These scenarios include:
    - Re-authentication. The user's authentication session has expired at the Relying Party and the user needs to be re-authenticated.
    - Step-up Authentication. The user has previously authenticated at the Relying Party, but the Relying Party has determined that a higher level of assurance is required, so triggers an additional authentication interaction for the user.

The TDIF technical integration standards support known subject authentication scenarios.

2.2.2.2 Force Authentication

In a force authentication scenario, the Relying Party has determined that it wants the user to immediately authenticate at the Identity Service Provider regardless of a pre-existing authenticated session at the Relying Party. This could occur when the user accesses a higher risk transaction, and hence be part of a known subject authentication scenario.

The TDIF technical integration standards support force authentication scenarios.

2.2.2.3 Single Sign On (SSO).

Single Sign On (SSO) refers to the ability for a user to be authenticated to a Relying Party and then being able to be authenticated again to another Relying Party without being required to provide their credentials again. This requires a single authenticated session to be maintained by a session broker. In a federation that includes a trust broker the role of the session broker is performed by the Identity Exchange. SSO can enhance the convenience and user experience for users but it also increases security risks from session hijacking and implementation of SSO **MUST** address these risks in agreement with the Relying Parties that it supports.

An Identity Exchange **MAY** support Single Sign On across Relying Parties. Where the an Identity Exchange supports Single Sign On, it **MUST** also support Known Subject Authentication (Section 2.2.2.1) and Force Authentication (Section 2.2.2.2).

A Relying Party **MUST** **NOT** assume that an authenticated session exists at the Identity Exchange, i.e. that the user will not be required to authenticate to the Identity Service Provider.

An Identity Exchange **MUST** expire the authenticated session in accordance with any minimum requirements specified for credentials in *TDIF: Authentication Credential Requirements*. An Identity Exchange **MAY** further restrict the expiration period for an authentication session to manage security risks.

An Identity Exchange **MAY** securely cache attributes from an Identity Service Provider for the duration of an authenticated session. This cached information **MUST** not be accessible to the operator of the Identity Exchange.

2.2.2.4 Single Logout

Single Logout (SLO) refers to be the ability for a user to initiate a logout process for all Relying Parties that relied on a single logon session for the user at an Identity Service Provider.

Standard federation protocols generally provide two mechanisms for enabling single logout that can be characterised follows:

- Front-channel Single Logout. The front-channel single logout model uses front-channel communication via the user agent (web browser).

- Back-channel Single Logout. The back-channel single logout model relies on direct server to server communication between the session participants.

In a federation where the RP and IdP directly interact, the IdP performs the roles of the session broker in the descriptions below. In a federation that involves a trust broker, such as an Identity Exchange, the trust broker performs the role of the session broker.

There are two generic Single Logout use-cases:

RP-initiated Single Logout. In this use-case following steps occur:

- The user initiates the SLO at a RP.
- The RP then sends the logout request to the session broker.
- The session broker determines every other participant that has been signed in during the current logon session at the session broker.
- The session broker sends that logout request to every other session participant (RP). Each RP terminates its logon session.
- The session broker terminates its own logon session, and sends a logout response to the initiating RP.
- The initiating RP terminates their logon session.

IdP-initiated Single Logout.

- The user initiates the SLO at the session broker.
- The session broker determines every other participant that has been signed in during the current logon session at the session broker.
- The session broker sends that logout request to every other session participant (RP). Each RP terminates its logon session.
- The session broker terminates its logon session.

In a federation that involves a trust broker, the implementation of single logout is more complex as the trust broker needs to propagate any logout requests between the parties:

- In RP-initiated Single Logout use-case, the Identity Exchange performs that role of session broker. In addition, a logout request is sent to the IdP that the user authenticated with.
- In the IdP-initiated Single Logout use-case, the Identity Exchange accepts a logout request from the IdP that the user authenticated with. The Identity Exchange acts as session broker in the SLO interaction and send a logout request to all RPs that have been authenticated as part of the same logon session at the IdP. In addition, a logout response is sent to the initiating IdP.

Single Logout is frequently not implemented by identity federations and tends to be treated an optional feature for the following reasons:

- The user experience outcome that the SLO process is intended to support is frequently poorly elaborated.
- Single Logout mechanisms have technical limitations, and are brittle.
- Single Logout is not an effective security mechanism. Closing a browser is the only effective mechanism for ensuring logout.
- Support for SLO results in significant overheads for development and overheads to operate at scale for the session broker, for questionable value.

The TDIF technical integration standards do not mandate the implementation of Single Logout by an Identity Exchange. This is an implementation decision for an Identity Exchange that considers:

- The user experience across all Relying Parties that it integrates with.
- The technical complexity of implementing it, especially if an Identity Exchange is required to support multiple federation protocols.
- Presence of other logical session brokers, such as common access point for digital services.

The TDIF technical integration standards have adopted the following approach to Single Logout:

- An Identity Service Provider **SHOULD** implement a SLO mechanism defined by the federation protocol that it supports. This requirement ensures that an Identity Service Provider can interoperate with any Identity Exchange that elects to implement a SLO scheme based on standard federation protocols.

- An Identity Exchange **MAY** implement a SLO mechanism, based on the needs of the users and Relying Parties that it will be supporting.

3 Common Technical Requirements

3.1 Overview

This section describes technical requirements that are common to all entities in the digital identity eco-system.

3.1.1 Key Privacy Requirements

Table 1 highlights requirements that are stated in the *TDIF: Privacy Requirements* and have a technical impact that is not specifically covered elsewhere in the TDIF technical standards.

Table 1: Key Privacy Requirements

Requirement Summary	Requirement Statement	Technical Impact	Reference
Separation of Identity Exchange	Identity Exchanges **MUST** operate separately from other identity federation participants and **MUST** establish and maintain its own privacy management arrangements.	Operation of Identity Exchange by an organisation must be administratively and logically separated from other systems operated by the organisation.	Section 2.1 General requirements

3.2 Technical Integration Standards

3.2.1 Relying Parties

Relying Parties operating as part of the federation **MUST** implement the Relying Party to Identity Exchange Profile specified in:

- *TDIF: OpenID Connect 1.0 Profile* as a OIDC Relying Party (RP).
- *TDIF SAML 2.0 Profile* as a SAML Service Provider (SP).

OIDC is the preferred protocol for new implementations.

3.2.2 Identity Service Providers

Identity Service Providers operating as part of the identity federation **MUST** implement the Identity Exchange to Identity Service Provider Profile specified in:

- *TDIF: OpenID Connect 1.0 Profile* as an OIDC Provider (OP).
- *TDIF SAML 2.0 Profile* as an Identity Service Provider (IdP).

OIDC is the preferred protocol for Identity Service Providers.

3.2.3 Identity Exchanges

Identity Exchanges operating as part of the identity federation **MUST** implement the Relying Party to Identity Exchange Profile specified in:

- *TDIF: OpenID Connect 1.0 Profile* as an OIDC Provider (OP).

Identity Exchanges operating as part of the identity federation **MUST** implement the Identity Exchange to Identity Service Provider Profile specified in:

- *TDIF: OpenID Connect 1.0 Profile* as a Relying Party (RP).

Identity Exchanges operating as part of the identity federation **MAY** implement the Relying Party to Identity Exchange Profile specified in:

- *TDIF SAML 2.0 Profile* as a SAML Identity Service Provider.

Identity Exchanges operating as part of the identity federation **MAY** implement the Identity Exchange to Identity Service Provider Profile specified in:

- *TDIF SAML 2.0 Profile* as a SAML Service Provider (SP).

The need for an Identity Exchange to implement SAML will be driven by the needs of the Relying Parties that are connecting to it, and by the presence of any Identity Service Provider that requires the use of SAML 2.0. Accreditation of an Identity Exchange is not predicated on the up-front need to support SAML 2.0.

Identity Exchanges operating as part of the federation **MUST** broker authentication requests from a Relying Party to an Identity Service Provider. The response from an Identity Service Provider to an Identity Service Provider **MUST** use a pairwise identifier to identify the subject of the authentication. In turn, the response from the

Identity Exchange to Relying Party **MUST** use different pairwise identifier to identify the subject of the authentication to the Relying Party. This use of pairwise identifiers is a key privacy mechanism. The definition of a Relying Party for the purposes of the allocation of pairwise identifiers is determined by the Relying Party in accordance with the *TDIF: Privacy Requirements* and any specific privacy and administrative arrangements that operate in its jurisdiction, including any including legislative requirements concerning how personal information should be collected, accessed and stored correctly. Additional technical guidance on the allocation of pairwise identifiers is provided in Section 4.1.2.

3.3 Recommended Cryptographic Algorithms

Cryptographic algorithms are used to secure the authentication interactions between entities in the digital identity eco-system. The use of cryptographic algorithms is specified in the *TDIF: Protective Security Requirements*. This section defines the recommended minimum key lengths for the cryptographic algorithms that are used in the interactions in the technical integration standards. These recommendations are consistent with the current edition of the Australian Government Information Security Manual and the *TDIF: Protective Security Requirements*.

3.3.1 Hashing

The recommended minimum algorithm for hashing operations is SHA-2. The minimum hash size is 256 bits.

3.3.2 Digital Signatures

The recommended minimum key size and algorithms for the creation of digital signatures are:

- RSA: 2048 bits.
- ECC: 160 bits.

Both the DSA and ECDSA algorithms **MAY** be used for digital signatures although ECDSA is recommended for longevity of a solution.

3.3.3 Session Key Agreement

The minimum recommended key sizes for the agreement of a session key are:

- DH: 2048 bits.
- ECDH 160 bits.

3.3.4 Symmetric Encryption

The recommended algorithm for symmetric encryption is AES. The minimum key size is 128 bits although 256 bits is recommended. When using AES, electronic code book mode (ecb) **SHOULD NOT** be used.

3.3.5 Key Transport Algorithms

Asymmetric encryption for the transport of keys using RSA or ECC **SHOULD** use:

- RSA: RSA OAEP, RSA OAEP + AES with the minimum recommended key length of 2048 bits.
- ECC: ECIES.

3.3.6 JSON Web Algorithms

When using JSON Web Tokens (JWT) the algorithms specified in the JSON Web Algorithms (JWA) **[RFC 7518]** where there is an alignment with the algorithms specified above these algorithms and key lengths **SHOULD** be used as a minimum.

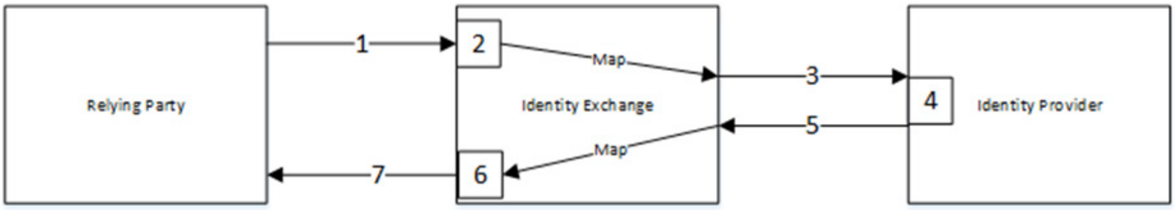
4 Identity Exchange Requirements

The Identity Exchange is the core component of the identity federation and acts as a trust broker between the Relying Parties and the Identity Service Providers.

4.1 Functional Requirements

Figure 4 illustrates the interactions that an Identity Exchange **MUST** implement in order perform the function of a trust broker. These are the logical interactions that are implemented using the federation protocols, the implementation of the logical interactions typically involve multiple protocol steps.

Figure 4 - *Trust Broker Interactions.*



The Trust Broker interactions are:

1. The Relying Party.
    a. Generates an authentication request and sends request to the Identity Exchange.
    b. Logs the authentication request.
2. The Identity Exchange:
    a. Generates a unique audit Id for the request and logs the authentication request from the Relying Party including the audit Id.
    b. Performs any processing required to determine the Identity Service Provider to send the request to.
3. The Identity Exchange:

   a. Maps the elements from the request from the Relying Party to the required elements to request them from an Identity Service Provider and generates the authentication request to send to the Identity Service Provider.
   b. Logs the generated authentication request using the audit Id generated for the request from the Relying Party in step 2.
   c. Sends the generated authentication request to the Identity Service Provider.
4. The Identity Service Provider:
   a. Logs the request from the Identity Exchange.
   b. Performs any additional processing to service the request.
   c. Generates a pairwise identifier for the user for the Identity Exchange if one does not already exist.
   d. Generates the authentication response to the Identity Exchange.
   e. Logs the authentication response to the Identity Exchange.
5. The Identity Service Provider sends the authentication response to the Identity Exchange.
6. The Identity Exchange:
   a. Logs the response from the Identity Service Provider, including the audit Id generated for the request in step 2.
   b. Performs any processing required to apply the attribute sharing policies.
   c. Generates a pairwise identifier unique to the user for the Relying Party if one does not already exist.
   d. Maps the elements from the response from the Identity Service Provider to the required elements in the authentication response to the Relying Party and generates the authentication response to send to the Relying Party.
   e. Logs the response to the Relying Party, including the audit Id generated for the request in step 2.
7. The Identity Exchange sends the authentication response to the Relying Party.

A Relying Party **MUST** generate the authentication request in step 1 in accordance with one of the following profiles:

- *TDIF: OpenID Connect 1.0 Profile* – using the Relying Party to Identity Exchange Profile.
- *TDIF: SAML 2.0 Profile* – using the Relying Party to Identity Exchange Profile.

The Identity Exchange **MUST** generate a unique audit Id for the request from a Relying Party as described in step 2. It **MUST** log the all related interactions between Relying Parties and Identity Service Providers using this unique audit id. The Identity Exchange **MUST** return this audit Id to the Relying Party using the RP_audit_id attribute described in the *TDIF: Attribute Profile*. The Identity **MUST** **NOT** send this audit id to the Identity Service Provider.

The Identity Exchange **MUST** implement the mapping described in steps 3 and 6 using the mappings specified in Section 4.1.1.

The Identity Exchange and Identity Service Provider **MUST** implement pairwise identifiers as specified in Section 4.1.2.

4.1.1 Trust Broker Protocol Mappings

The following mappings apply when an Identity Exchange brokers a request from a Relying Party to an Identity Service Provider:

4.1.1.1 OIDC Authentication Request from Relying Party

- Where the authentication request from the Relying Party is OIDC, and is brokered to an Identity Service Provider using OIDC, then the processing rules specified in Section 0 **MUST** be applied.
- Where the authentication request from the Relying Party is OIDC, and is being brokered to an Identity Service Provider using SAML 2.0, then the processing rules specified in Section 4.1.5 **MUST** be applied.

4.1.1.2 SAML Authentication Request from Relying Party

- Where the authentication request from the Relying Party is SAML, and is brokered to an Identity Service Provider using OIDC, then the processing rules specified in Section 4.1.7 **MUST** be applied.

- Where the authentication request from the Relying Party is SAML, and is being brokered to an Identity Service Provider using SAML 2.0, then the processing rules specified in Section 4.1.6 **MUST** be applied.

4.1.2 Pairwise Identifiers and Identity Resolution

To operate as a Trust Broker in the identity federation an Identity Exchange is required to implement an Identity Resolution process whereby the Identity Exchange maps the identities managed by Identity Service Providers to the records held by the Relying Parties that consume these identities. Central to the function of identity resolution is the use of pairwise identifiers to map the identity of a user at an Identity Service Provider to a record at a Relying Party. Separate pairwise identifiers are used in the following interactions for a user as follows:

- Between the Identity Exchange and an Identity Service Provider. An unique pairwise identifier is generated by an Identity Service Provider for each user that they authenticate as part of a Trust Broker interaction.
- Between the Identity Exchange and a Relying Party. A unique pairwise identifier for the user is allocated by the Identity Exchange for each Relying Party that the takes part in a Trust Broker Interaction.

An Identity Exchange **MUST** implement an identity mapping process that maps the pairwise identifier presented by an Identity Service Provider in response to an authentication request to the pairwise identifier for the user at the Relying Party that initiated the authentication interaction. A Relying Party will always be presented with the same pairwise identifier as the subject identifier whenever the user uses the same Identity Service Provider as part of a Trust Broker interaction. There is no correlation of identity across Identity Service Providers.

Recommendations for the creation Pairwise Identifiers are contained in Section 8.1 of **[OpenID.Core]** for OIDC and as described in Section 3.4 of **[SAML-SubjectID-v1.0]** for SAML. The requirements for an Identity Exchange interacting with Relying Parties are more flexible to cater for the transition of Relying Parties from a legacy federation to an Identity Exchange.

Identity Service Providers **MUST NOT** generate Identifiers greater than 255 ASCII characters.

The Identify Exchange **MUST** be able to receive pairwise identifiers of up to 255 ASCII characters.

An Identity Exchange **SHOULD** generate Identifiers in accordance with the OIDC specification **[OpenID.Core]** and use these to interact with Relying Parties regardless of federation protocol.

An Identity Exchange **MAY** advertise a maximum length of the pairwise Identifiers it generated based on the mechanism it uses.

4.1.2.1 OIDC Relying Party Sector Identifiers

The OIDC specification closely couples the concept of a Relying Party to a client, or an software application instance. A TDIF Relying Party may need to register multiple OIDC clients for the different digital services that it provides but still require the same underlying pairwise identifier for an authenticated user to be passed to all of its registered OIDC clients.

The OIDC standard provides a mechanism to enable multiple clients to receive the same pairwise identifier. This mechanism is termed a Sector Identifier and is defined in OIDC Core specification https://openid.net/specs/openid-connect-core-1_0.html#PairwiseAlg and is further expanded on the specification for Dynamic Client Registration https://openid.net/specs/openid-connect-registration-1_0.html#SectorIdentifierValidation.

The Exchange **MUST** support the configuration of a Sector Identifier for a TDIF Relying Party in Identifiers in accordance with the OIDC specification **[OpenID.Core].**

The process for the registration of OIDC clients by an Identity Exchange **MUST** ensure that only valid and authorised clients for the TDIF Relying Party can use the same configured sector_identifier_uri.

4.1.3 Assurance Levels

TDIF assurance levels are represented in the technical integration standards by values of an Authentication Context Class Reference (acr). Acr is a concept supported by both the OpenID Connect 1.0 and SAML 2.0 standards.

TDIF assurance levels are represented using the values that defined in Section 2.2.8 of the *TDIF: Attribute Profile.*

Required acr values are represented in an OIDC authentication request using either the `acr_values` parameter or the `acr` claim. Required acr values are represented in a SAML request using the `<saml:AuthnContextClassRef>` element. Multiple acr values **MAY** be included in OIDC and SAML authentication requests. OIDC does not provide a mechanism for specifying an acr value as a miniumum required acr. The OIDC profile **[TDIF.OIDC]** describes a mechanism whereby a Relying Party can specify a single acr value as a minimum required value in an OIDC authentication request to an Identity Exchange and have it reliably proxied to a SAML or OIDC Identity Service Provider.

4.1.4 OIDC to OIDC Brokering

When the Identity Exchange is accepting requests from an OIDC Relying Party and translating those requests to an OIDC Identity Service Provider, the Identity Exchange **MUST** interact with the Identity Service Provider as per the OpenID Connect 1.0 Profile **[TDIF.OIDC]**, with the following processing rules.

4.1.4.1 Mapping Claims to Scopes

Scopes and Claims that are received from the Relying Party **MUST** be included in the request to the Identity Service Provider in accordance with the following processing rules

- Scopes and Claims that are defined within the *TDIF: Attribute Profile* **MUST** be included.
- If the `sub` (subject) claim is specified then it **MUST** be processed as per 4.1.4.2.

Any other Scopes and Claims **MUST** be ignored. Where Scopes or Claims are ignored, the Identity Exchange **MUST** **NOT** raise an error.

4.1.4.2 Handling of sub claim

An Identity Exchange **MAY** support the `sub` (subject) claim.

As the subject identifier is pairwise identifier for the user at the Relying Party, the Identity Exchange **MUST** resolve a pairwise identifier included in the `sub` (subject) claim in request from a Relying Party to an existing pairwise identifier for the user at the required Identity Service Provider. If no pairwise identifier for the user at the Identity Service Provider can be resolved then the Identity Exchange **SHOULD** return an error.

4.1.4.3 Mapping Assurance Levels

Where the `acr_values` or `acr` claim received from the Relying Party is a single value the Identity Exchange **MUST** pass the set of acr values that meet or exceed the value of the requested acr value to the Identity Service Provider in the generated authentication request. Where the `acr` claim is marked as essential within the request from the Relying Party it **MUST** be marked as essential when sent to the Identity Service Provider.

The Identity Exchange **MUST** evaluate the acr returned from the Identity Service Provider and if the acr meets or exceeds the originally requested value, return the originally requested value.

An example is shown below:

`acr_values` received from the Relying Party:

```
acr_values=urn:id.gov.au:tdif:acr:ip3:cl2
```

`acr_values` mapped by Identity Exchange in request to the Identity Service Provider.

```
acr_values=urn:id.gov.au:tdif:acr:ip3:cl2
urn:id.gov.au:tdif:acr:ip3:cl3 urn:id.gov.au:tdif:acr:ip4:cl3
```

Value of acr claim returned from the Identity Service Provider to the Identity Exchange as part of the ID Token:

```
“acr”:”urn:id.gov.au:tdif:acr:ip3:cl3”
```

Value of `acr` claim returned to the Relying Party as part of the ID Token

```
“acr”:”urn:id.gov.au:tdif:acr:ip3:cl2”
```

4.1.4.4 Other OIDC Request Parameters

The following sections specify processing rules for OIDC parameters that a Relying Party **MAY** include in an OIDC request to an Identity Exchange.

4.1.4.4.1 prompt Parameter

Table 4.1: *Processing rules for prompt parameter.*

Value received in OIDC request from Relying Party	Value sent in OIDC request to Identity Service Provider
`none`	`none`
`consent`	Ignored. The Identity Exchange **MUST** implement consent for the release of attributes in accordance with the Attribute Sharing Policy defined within the *TDIF: Attribute Profile*
`login`	`login`
`select_account`	`select_account`

4.1.4.4.2 id_token_hint Parameter

A Relying Party **MAY** include an ID Token previously issued by the Identity Exchange in the request to identify a specific user that requires authentication.

This specification does not require support for this mechanism by an Identity Exchange, but where it is supported the following processing rules **MUST** apply:

Where the Identity Exchange receives an `id_token_hint` within an authentication request from a Relying Party the Identity Exchange is required to validate the token and extract the subject. The Identity Exchange **MUST** resolve this to a subject identifier at the Identity Service Provider as per 4.1.4.2. The Identity Exchange **SHOULD** include the resolved subject identifier in the authentication request to the Identity Service Provider using the `sub` (subject) Claim.

4.1.5 OIDC to SAML Brokering

When the Identity Exchange is accepting requests from an OIDC Relying Party and translating those requests to a SAML Identity Service Provider, the Identity Exchange **MUST** interact with the Identity Service Provider as per the SAML 2.0 profile **[TDIF.SAML]** with the following processing rules.

4.1.5.1 Mapping Claims to Scopes

Scopes and Claims that are received from the Relying Party **MUST** be included in the request to the SAML 2.0 Identity Service Provider in accordance with the following processing rules:

- Claims that are defined within the within the *TDIF: Attribute Profile* **MUST** be included using the OIDC to SAML mapping described therein.
- Scopes that are defined in the Authentication Request **SHOULD** be expanded into the underlying claims and mapped as per the *TDIF: Attribute Profile.*
- If the `sub` (subject) claim is specified then it **MUST** be processed as per 4.1.4.2.
- Any other Scopes and Claims **MUST** be ignored. Where Scopes or Claims are ignored, the Identity Exchange **MUST NOT** raise an error.

4.1.5.2 Mapping Assurance Levels

Where the `acr_values` or `acr claim` received from the Relying Party is a single value the Identity Exchange **MUST** pass the set of `<saml:AuthnContextClassRef>` values that meet or exceed the value to the requested acr to the Identity Service Provider in the generated authentication request. Where the `acr` claim is marked as essential within the request from the RP the `<samlp:RequestedAuthnContext> comparison` attribute **MUST** be set to `minimum` when sent to the Identity Service Provider.

The Identity Exchange **MUST** evaluate the `<saml:AuthnContextClassRef>` returned from the Identity Service Provider and if the `<saml:AuthnContextClassRef>` meets or exceeds the originally requested acr value, return the originally requested value.

An example is show below:

`acr_values` received from the Relying Party:

```
acr_values=urn:id.gov.au:tdif:acr:ip3:cl2
```

`acr_values` mapped to SAML 2.0 by Identity Exchange in request to the Identity Service Provider:

```
<samlp:RequestedAuthnContext Comparison="minimum">
    <saml:AuthnContextClassRef>
      urn:id.gov.au:tdif:acr:ip3:cl2
    </saml:AuthnContextClassRef>
    <saml:AuthnContextClassRef>
      urn:id.gov.au:tdif:acr:ip3:cl3
    </saml:AuthnContextClassRef>
    <saml:AuthnContextClassRef>
      urn:id.gov.au:tdif:acr:ip4:cl3
    </saml:AuthnContextClassRef>
</samlp:RequestedAuthnContext>
```

acr values returned from the Identity Service Provider to the Identity Exchange as part of the SAML 2.0 Response:

```
<saml:AuthnContext>
    <saml:AuthnContextClassRef>
      urn:id.gov.au:tdif:acr:ip3:cl3
    </saml:AuthnContextClassRef>
</saml:AuthnContext>
```

Value of `acr` claim returned to the Relying Party as part of the ID Token

```
“acr”:”urn:id.gov.au:tdif:acr:ip2:cl3”
```

4.1.5.3 Other OIDC Request Parameters

The following sections provide information on the transformation and passing of specific attributes from the OIDC request from a Relying Party to a SAML 2.0 Identity Service Provider.

4.1.5.3.1 Prompt parameter

Table 2: *Processing rules for prompt parameter.*

Value received in OIDC request from Relying Party	Value sent in SAML 2.0 request to Identity Service Provider
`none`	`isPassive` attribute is set to true on the `<AuthnRequest>` message
`consent`	Ignored. The Identity Exchange **MUST** implement consent for the release of attributes in accordance with the Attribute Sharing Policy defined within *TDIF: Attribute Profile*
`login`	`ForceAuthn` attribute is set to true on the `<AuthnRequest>` message
`select_account`	Ignored.

4.1.5.3.2 id_token_hint Parameter

A Relying Party **MAY** include an ID Token previously issued by the Identity Exchange in the request to identify a specific user that requires authentication.

This specification does not require support for this mechanism by an Identity Exchange, but where it is supported the following processing rules **MUST** apply:

Where the Identity Exchange receives an `id_token_hint` within an authentication request from a Relying Party the Identity Exchange is required to validate the token and extract the subject. The Identity Exchange **MUST** resolve this to a subject identifier at the Identity Service Provider per 4.1.4.2.The Identity Exchange **SHOULD** include the resolved subject identifier in the authentication request to the Identity Service Provider by including it in a `<saml:Subject>` element in the SAML 2.0 `<AuthnRequest>` message.

4.1.5.3.3 max_age Parameter

A Relying Party **MAY** optional include a value for the max_age parameter in the OIDC request, as per Section 3.1.2.1 of the OpenID Connect Core specification [OpenID.Core]: ‘Specifies the allowable elapsed time in seconds since the last time the End-User was actively authenticated by the OP. If the elapsed time is greater than this value, the OP **MUST** attempt to actively re-authenticate the End-User.’

There is no equivalent functionality in SAML 2.0 protocol, so in order to support this functionality the Identity Exchange **MUST** implement the following processing:

- On receiving the authentication response, the Identity Service Provider **MUST** calculate the elapsed time since the user was authenticated using the value of `AuthInstant` attribute in the SAML 2.0 response from the Identity Service Provider.
- If the elapsed time is greater than the `max_age` value requested by the Relying Party then the Identity Exchange **MUST** generate a fresh authentication request with the `ForceAuthn` attribute is set to true on the `<AuthnRequest>` message.

4.1.5.3.4 Other Parameters

Table 3 *Processing rules for other OIDC parameters*

OIDC request parameter from Relying Party	Equivalent in SAML 2.0 request to Identity Service Provider
`display`	No SAML 2.0 equivalent. The Identity Service Provider is responsible for detecting the capabilities of the user agent and presenting the appropriate display.
`login_hint`	Ignored.

4.1.6 SAML to SAML Brokering

When the Identity Exchange is accepting requests from a SAML Relying Party and translating those requests to a SAML Identity Service Provider, the Identity Exchange **MUST** interact with the Identity Service Provider as per the SAML profile, specified in this document, with the following processing rules.

4.1.6.1 Mapping Attributes

Where the attributes required are predefined within the Relying Parties metadata, the set of required attributes **MUST** be included in the request to the Identity Service Provider with the following processing rules:

- Where the requested attributes contained within the Relying Party's metadata are the same as the Identity Exchanges requested attributes in its metadata exchanged with the Identity Service Provider; the Identity Exchange creates a standard authentication request.
- Where the requested attributes are not available in the requested attributes as part of the metadata shared with the Identity Service Provider by the Identity

Exchange; the Identity Exchange is required to create an authentication request to the Identity Exchange using extensions to request the attributes required by the Relying Party.

Where the attributes requested by a Relying Party are requested via extensions the Identity exchange **MUST** copy those attributes into the authentication request to the Identity Service Provider as extensions.

4.1.6.2 Subjects within Requests

The Relying Party **MAY** include a SAML Subject in the authentication request. As the subject identifier is pairwise identifier for the user at the Relying Party, the Identity Exchange **MUST** resolve this pairwise identifier in any request to an existing pairwise identifier for the user at the required Identity Service Provider. If no pairwise identifier for the user at the Identity Service Provider can be resolved then the Identity Exchange **SHOULD** return an error.

4.1.6.3 Mapping Assurance Levels

Where the Relying Party includes a `<RequestedAuthnContext>` in the authentication request, the Identity Exchange is required to send the set of `<AuthnContextClassRef>` to the Identity Service Provider that meet or exceed the originally requested `<RequestedAuthnContext>`.

The `Comparison` attribute for the <RequestedAuthnContext> **MUST** be set to `exact` or `minimum`.

4.1.6.4 Other SAML Request Parameters

4.1.6.4.1 ForceAuthn Attribute

When the ForceAuthn attribute is set to true within the Authentication Request from the Relying Party this **MUST** be passed through in the Authentication sent by the Identity Exchange to the Identity Service Provider.

4.1.6.4.2 isPassive Attribute

When the isPassive attribute is set to true within the Authentication Request from the Relying Party this **MUST** be passed through in the Authentication sent by the Identity Exchange to the Identity Service Provider.

4.1.7 SAML to OIDC Brokering

When the Identity Exchange is accepting requests from a SAML Relying Party and translating those requests to an OIDC Identity Service Provider, the Identity Exchange **MUST** interact with the Identity Service Provider as per of OIDC Profile [TDIF.OIDC] with the following processing rules.

4.1.7.1 Mapping Attributes to Claims or Scopes

The attributes requested within the Authentication Request either through extensions or via the Relying Party's metadata **MUST** be processed in accordance with the following rules:

- All attributes included in the Relying Party's Authentication Request **MUST** be included in the authentication request sent to the Identity Service Provider in either Scopes or Claims.
- Where the attributes can be mapped fully into an available scope the Identity Exchange **SHOULD** request those scopes from the Identity Service Provider.
- Where the attributes do not map fully into a Scope the Identity Exchange **MUST** requests those attributes as claims from the Identity Service Provider.

4.1.7.2 Mapping Assurance Levels

Where the Relying Party includes a `<RequestedAuthnContext>` in the authentication request, the Identity Exchange is required to send the set of `acr` values to the Identity Service Provider that meet or exceed the originally requested `<RequestedAuthnContext>`. The set of `acr` values **SHOULD** use the `acr` claim. The `acr` claim **SHOULD** be marked as essential.

The `Comparison` attribute for the `<RequestedAuthnContext>` **MUST** be set to `exact` or `minimum`.

4.1.7.3 Other SAML Request Parameters

4.1.7.3.1 ForceAuthn

Where the `ForceAuthn` attribute is included in the authentication request from the Relying Party, the Identity Exchange **MUST** set the `prompt` parameter to `login` in the OIDC authentication request to the Identity Service Provider.

4.1.7.3.2 isPassive

Where the `isPassive` attribute is included in the authentication request from the Relying Party, the Identity Exchange **MUST** set the `prompt` parameter to `none` in the OIDC authentication request to the Identity Service Provider.

4.1.7.3.3 Subject

Where a `Subject` is included in the authentication request from the Relying Party the Identity Exchange is required to validate the token and extract the subject. The Identity Exchange **MUST** resolve this to a subject identifier at the Identity Service Provider as per 4.1.4.2. The Identity Exchange **SHOULD** include the resolved subject identifier in the authentication request to the Identity Service Provider using the `sub` (subject) Claim.

4.2 Attribute Sharing and User Consent

The Identity Exchange **MUST** implement the Attribute Sharing Policies defined in the *TDIF: Attribute Profile*.

Note that is includes the ability for the user to revoke any ongoing consent that has been provided using an Identity Service Provider.

4.3 IdP Selection

An Identity Exchange **MUST** allow a user to select an Identity Service Provider when accessing a Digital Service from a list of Identity Service Providers that are integrated with the Identity Exchange.

The list of Identity Service Providers presented to the user **MUST** be capable of meeting the assurance levels (Credential Level and Identity Proofing Level) that is requested by a Relying Party.

An Identity Exchange **SHOULD** provide a mechanism for a user's selection of Identity Service Provider to be remembered so that the user does not have to select an Identity Service Provider when accessing digital services at Relying Parties. The user **MUST** agree to the Identity Exchange remembering an Identity Service Provider selection, and there **MUST** be a mechanism available for the user to remove the remembered Identity Service Provider selection.

4.4 Audit History

An Identity Exchange **MUST** maintain a log of all interactions that a user undertakes with a Relying Party via an Identity Exchange.

An Identity Exchange **MUST NOT** persistently store any values of any personal attributes retrieved from an Identity Service Provider or Attribute Provider.

The retention period for audit logs will be determined in a future TDIF release. This will take into consideration any statutory record keeping requirements, requirements for an audit trails, and consider what aspects can potentially be user-controlled.

4.5 Consumer History and User Dashboard

The Consumer History refers to the access that a user is provided to the Audit History.

An Identity Exchange **MUST** provide a Consumer History than can be accessed by the user authenticating using an Identity Service Provider that is integrated with the Identity Exchange.

The Consumer History **MUST** enable the user to perform the following actions for the Identity Service Provider they authenticated with:

- View the history of all interactions (Audit History) performed via the Identity Exchange using the Identity Service Provider.

- View the consent that the user has provided to share attributes using the Identity Service Provider with a Relying Party. These attributes are provided by the Identity Service Provider or by an Attribute Provider.

The User Dashboard is collective term for the features that an Identity Exchange provides to a user that has been authenticated by an Identity Service Provider. This includes:

- Access to the Consumer History, as described above.
- Ability to revoke ongoing consent to shared attributes with a Relying Party as notes in Section 4.2.

5 Identity Service Provider Requirements

5.1 Functional Requirements

Identity Service Provider **MUST** implement the requirements specified in the *TDIF: Identity Proofing Requirements.*

Identity Service Provider **MUST** be integrated with a Credential Service Provider that provides the credentials that are used to authenticate a user.

5.2 Technical Integration Standards

An Identity Service Provider **MUST** implement the technical integration standards specified in Section 3.2.

The technical integration of a Credential Service Provider with an Identity Service Provider is not specified in the TDIF. The primary requirement is that this integration is secure and this will be assessed as part of the accreditation of an Identity Service Provider.

6 Credential Provider Requirements

6.1 Functional Requirements

Credential Service Provider **MUST** implement the requirements specified in the *TDIF: Authentication Credential Requirements.*

7 Attribute Provider Requirements

7.1 Overview

An Attribute Provider is an entity that provides additional attributes other than those that are supplied by an Identity Service Provider.

7.2 Functional Requirements

Attribute Provider **MUST** implement the requirements specified in the *TDIF: Attribute Provider Requirements.*

An Attribute Provider **MUST** be authoritative for the attributes that it provides to a TDIF identity federation as verified attributes.

An Attribute Provider **MUST** share attributes in accordance with the privacy principles of the TDIF identity federation.

Specific functional requirements for an Attribute Provider will be elaborated as an Attribute Provider is identified. These requirements **MUST** state any specific requirements that an Attribute Provider **MUST** meet in order to provide a Relying Party with attributes that have a consistent and defined level of assurance. If the binding of attributes to a digital identity can achieve more than one level of assurance then attributes that express these levels of assurance **MUST** be defined and made available to Relying Parties via an Identity Exchange in conjunction with the attributes themselves.

7.3 Technical Integration Standards

An Attribute Provider is a broad concept present in many digital eco-systems that provide Relying Parties with authorised access to attributes. The TDIF technical integration standards are not intended to support all possible patterns or scenarios for the sharing of any arbitrary set of attributes. The TDIF technical standards are concerned with attributes that have a close relationship with digital identity.

The technical integration standards detailed here assume the following:

- An Attribute Provider is integrated with an Identity Exchange as a Relying Party in order to make the attributes available to Relying Parties.
- The user authenticates to the Attribute Provider using their chosen Identity Service Provider.
- The Identity Exchange makes the attributes available to Relying Party in accordance with the technical requirements detailed in Section 4.
- Attributes provided by Attribute Providers are specified in *TDIF: Attribute Profile*.
- Attribute sets provided by different Attribute Providers are disjoint, i.e. there is a 1:1 correspondence between attribute sets and an Attribute Provider.

7.4 Technical Requirements

The technical requirements detailed in this section apply when an Attribute Provider is integrated with an Identity Exchange as a Relying Party.

7.4.1 General Requirements

Attributes **MUST** be shared with Relying Parties in accordance with the Attribute Sharing Policies defined in the *TDIF: Attribute Profile*.

The gathering of user consent may be able to be delegated to an Attribute Provider if these is explicitly stated in the *TDIF: Attribute Profile*. In order for this user consent to be specific as required in the *TDIF: Privacy Requirements*, the Attribute Provider must be permitted to have visibility of the Relying Party requesting the attributes.

7.4.2 Attribute Provider Requirements

An Attribute Provider **MUST** be integrated with an Identity Exchange as a Relying Party in accordance with technical standards listed in Section 3.2.1.

An Attribute Provider **MUST** use the pairwise identifiers (RP Links) generated by an Identity Exchange for it as a Relying Party to associate the attributes that it provides with the digital identity brokered by an Identity Exchange.

An Attribute Provider **MUST** provide an API that enables the attributes it provides available to be shared with Relying Parties. The call to this API **MUST** use the pairwise identifier issued by Identity Exchange to identify the required attributes.

The Attribute Provider **MUST** authorise the Identity Exchange to securely access the API it provides.

It is recommended that the API provided by an Attribute Provider be implemented as a REST API. Where a REST API is provided the Attribute Provider **SHOULD** authorise access in accordance with JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants **[RFC 7523]**.

7.4.3 Identity Exchange Requirements

When an Identity Exchange receives an authentication request from a Relying Party that includes attributes supplied by an Attribute Provider then it **MUST** call the API provided by the Attribute Provider to make these attributes available to the Relying Party in the authentication response. The Identity Exchanges makes the attributes available to Relying Parties by operating as a Trust Broker as described in Section 4.1.

Attributes provided to a Relying Party **MAY** be made available by authorising the Relying Party to directly retrieve the attributes from the Attribute Provider. This mechanism **MUST** only be used when specified as being permissible in the *TDIF: Attribute Profile* as it **MUST** be consistent with the privacy requirements of the identity federation. This mechanism requires the Identity Exchange to return a security token to the Relying Party. The Relying Party can use this security token to retrieve the attributes from the Attribute Provider. Any security token issued by an Identity Exchange to a Relying Party **MUST NOT** reveal the pairwise identifier of the Attribute Provider.

Where the retrieval of attributes directly from an Attribute Provider by a Relying Party is permitted it is recommended that this be implemented using distributed claims as detailed in section of the OpenID Connect 1.0 Core Specification https://openid.net/specs/openid-connect-core-1_0.html#AggregatedDistributedClaims.

7.4.4 Audit Logging

The section details additional technical requirements that **MUST** be met to enable an audit trail of attribute binding and retrieval.

An Attribute Provider **MUST** maintain a log of the following events:

- The binding of any attributes to a digital identity brokered by an Identity Exchange. This occurs as a result of authentication interaction with the Identity Exchange and the response from the Identity Exchange will include the RP Audit Id attribute. These logged events **MUST** include this value of the RP Audit Id attribute.
- The retrieval of attributes by an Identity Exchange or Relying Party. The call to the API provided by the Attribute Provider **MUST** include the value of RP Audit Id attribute that has generated by the Identity Exchange for the Relying Party that requested the attributes. These logged events **MUST** include this value of the RP Audit Id attribute.
- Any user consent managed by an Attribute Provider then enables the sharing of attributes with a Relying Party.