Queensland Government Enterprise Architecture

Federated identity blueprint

Decision framework

Final

May 2017

V1.0.0

PUBLIC

Document details

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **Security classification** | 1. PUBLIC | | | | | |
| 1. **Date of review of security classification** | 1. May 2017 | | | | | |
| 1. **Authority** | 1. Queensland Government Chief Information Officer | | | | | |
| 1. **Author** | 1. Queensland Government Chief Information Office | | | | | |
| 1. **Documentation status** |  | 1. Working draft |  | 1. Consultation release | 1. 🗹 | 1. Final version |

Contact for enquiries and proposed changes

All enquiries regarding this document should be directed in the first instance to:

Queensland Government Chief Information Office  
[qgcio@qgcio.qld.gov.au](mailto:qgcio@qgcio.qld.gov.au)

Acknowledgements

This version of the *Federated identity blueprint – Decision framework* was developed and updated Queensland Government Chief Information Office.

Feedback was also received from a number of agencies, which was greatly appreciated.

Copyright

*Federated identity blueprint – Decision framework*

Copyright © The State of Queensland (Queensland Government Chief Information Office) 2017

Licence

[](http://creativecommons.org/licenses/by/4.0/)

This work is licensed under a Creative Commons Attribution 4.0 International licence. To view the terms of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. For permissions beyond the scope of this licence, contact [qgcio@qgcio.qld.gov.au](mailto:qgcio@qgcio.qld.gov.au).

To attribute this material, cite the Queensland Government Chief Information Office.

The licence does not apply to any branding or images.

Information security

This document has been security classified using the Queensland Government Information Security Classification Framework (QGISCF) as PUBLIC and will be managed according to the requirements of the QGISCF.

Contents

[1 Decision framework 4](#_Toc484442636)

[1.1 Federated authentication 4](#_Toc484442637)

[1.2 Federated delegated authorisation 40](#_Toc484442638)

[1.3 Federated attribute exchange 42](#_Toc484442639)

[Appendix A OpenID connect flows and OAuth grant types 44](#_Toc484442640)

[Appendix B Attribute metadata standards 45](#_Toc484442641)

# Decision framework

This decision framework outlines a process for establishing identity federation across one or more domains to support:

* federated authentication or;
* federated delegated authorisation or;
* federated identity attribute exchange

## Federated authentication

The decision framework for federated authentication outlines a process for a relying party to select an identity provider and establish a federation trust. It is written from the perspective of a relying party:

* selecting an identity provider
* selecting a credential provider (if separate from the identity provider)
* determining the federation topology when there are multiple identity or credential providers
* establishing the required federation trust (connection) to the identity or credential providers which includes protocol, assertion and provisioning and assertion security considerations.

### Protected resource assessment

#### Determine identity and credentials assurance requirements

Undertake a business risk based assessment using the [QGAF](https://www.qgcio.qld.gov.au/products/qgea-documents/549-information-security/2415-queensland-government-authentication-framework) as a guideline to determine the desired:

* Identity Registration Assurance Level (IRAL) requirements
* Identity Authentication Assurance Level (IAAL) requirements

Note: The QGAF process will also require the data to be classified using the [Queensland Government Information Security Classification Framework (QGISCF)](https://www.qgcio.qld.gov.au/products/qgea-documents/549-information-security/2417-queensland-government-information-security-classification-framework).

The assessment process will assist in clarifying with stakeholders’ requirements pertaining too:

* identity registration and lifecycle management process
* credential management and authentication process
* access management and authorisation process.

|  |  |
| --- | --- |
| Result | |
| Identity registration strength: | <<insert assurance level or key business requirements e.g. client anonymity, service history and personalisation, real world identity link non-repudiation etc >> |
| Credential strength: | <<List each resource or key business requirements>> |

If the RP service does not require authentication, do not continue with this decision framework.

#### Determine resource type

Determine the type of resource to be protected. This may include a group of logically related resources for a particular user constituency.

|  |  |  |
| --- | --- | --- |
| **Regular n-tier web application** | A traditional web application hosted on a server, written in a server-side language e.g. Java, ASP.NET, Node.js, PHP, Python or Ruby on Rails | |
| **Backend API + client application** | **Client application type** | |
| **Native mobile application** | An application that runs natively on mobility devices. E.g. applications built using specific device SDK’s for iOS, Android, Windows Phone platforms |
| **Cross-platform applications** | An application that runs natively on mobility devices. E.g. applications built using a cross-platform framework such as Xamarin or React and complied to run natively on one or more platforms |
| **Hybrid mobile app** | A HTML5 and JavaScript based application hosted in a native wrapper which provides device functionality e.g. applications built using Cordova, Phonegap, Ionic or React |
| **Singe page application (SPA)** | A JavaScript front-end browser application e.g. application built using Angular, Backbone, Ember, jQuery |
| **Progressive web application (PWA)** | A mobile optimised web application with access to select native device features without app store installation |
| **Native desktop applications** | An application that runs natively on Microsoft Windows (e.g. Universal Windows Platform (UWP) apps for Windows 10), Mac OSX or Linux |
| **Hybrid desktop applications** | A HTML5 and Javascript application hosted in a native wrapper which provides OS functionality e.g. applications built using Electron |
| **Backend API type** | |
| **REST API** | Modern web service API’s and Web Hooks which use JSON to format data |
| **SOAP web service** | Traditional web service API’s which use XML to format data |

|  |  |
| --- | --- |
| Result | |
| Resource type: | <<list each resource and its type >> |

#### Determine resource ownership model

|  |  |
| --- | --- |
| First party developed/supplied | Bespoke applications or services developed and maintained by a Queensland Government Agency. This could include outsourced software development. |
| Third party developed/supplied | Commercial off-the-shelf (COTS) vendor products developed and maintained by external organisations. This could also include applications developed by an organisation for Queensland Government service used or sold directly to Queensland Government Customers (e.g. citizen developers or paid mobile applications for citizens) |

|  |  |
| --- | --- |
| Result | |
| Resource ownership model: | <<list each resource and its ownership model >> |

#### Determine resource support for claims-based authentication

Determine if the resource natively supports claims-based authentication using a standard federation protocol.

|  |  |
| --- | --- |
| **Supports claims-based authentication** | Claims-aware applications are applications which are designed to accept a claim about an authenticated user (issued by a trusted issuer) rather than authenticate the user themselves.  For example, support for federated identity standards such as SAML, ODIC, OAuth. Kerberos and HTTP Cookies also represent a claim, although these token formats are difficult to transport across domains and the internet in general. Kerberos was designed for use within a LAN environment and HTTP Cookies cannot be shared across web resource domains. |
| **Does not support claims-based authentication** | Applications that register and authenticate users directly typically against a local repository or shared enterprise directory. For example, username/password, local account registration, LDAP etc. |

If the resource does not support claims-based authentication for federation, an access security bridge (ASB) will be required to convert a standard based assertion to local security domain or application specific proprietary token format to set a valid session or other security context for the user represented by the identity attributes.

The ASB may be co-located within the resource domain for one or more resources or shared across resources and resource domains.

Available options for an ASB depending upon resource type are as follows:

|  |  |
| --- | --- |
| Resource type | Example ASB’s |
| Web application | * Using a reverse application proxy to terminate the connection and optionally generate a local security domain token to impersonate the user. * This approach is recommended because it avoids the need to install and maintain agents on each web server * This approach is recommended for presenting internal domain-joined web applications using Kerberos authentication externally - typically for employee remote access * Installing an SSO agent on to the web server which enables support for standard federation tokens or propriety tokens (working in combination with the approach above) * The IdP may be able to provide password vaulting functionality for secure storage and reply of passwords. This option should be carefully considered, however does mitigate the need for an ASB and RP application changes. |
| Web API’s or service | * Using an API gateway to accept a standard token format and translate into the local resource domain’s format * API management and some reverse application proxies may provide similar functionality to an API gateway |

|  |  |
| --- | --- |
| Result | |
| Federation standards supported: | <<list the federation standards supported>> |

### User Constituencies Assessment

Determine the types of users which need to be authenticated. Also consider the:

* type of users
* dimensions – scale/number of users
* identities held (and available identifiers)
* credentials held
* devices used.

|  |  |
| --- | --- |
| Result | |
| User constituencies: | <<list the primary user constituencies and types of users>> |

### Determine the identity provider (IdP)

An identity provider for each user constituency group needs to be determined. Repeat this process for each user constituency.

An IdP provider is generally not required if the:

* IRAL is 0 (anonymous access)
* IRAL is 1 (pseudonymous access) – sometimes the IdP may still be used for a shared pseudonymous identifier

#### Classify the user constituency relationship

Determine the type of relationship the RP has with the user constituency:

|  |  |
| --- | --- |
| Relationship type[[1]](#endnote-1) | Description and example |
| Individual relationship, known user | An agency has an established relationship with a known individual based on a need to conduct business with or provide services directly to the individual. An agency may choose to establish an account or role for this individual prior. |
| Individual relationship, unknown user | An agency has an established business or service relationship with a group of individuals that meet specified criteria, but does not know which individuals will need access to the resource at the time that the business or service is established. |
| Business-entity relationship with a known user base | An agency has an established business relationship with an entity and knows which users within that organisation will need access to the agency’s resources. Because the target user base is known, the agency has the option to provisioning the user account in advance. |
| Business-entity relationship with an indeterminate user base | An agency has an established business relationship with an entity, but the agency does not know which individuals will need access to the agency’s resources. The user population may frequently change and it is difficult to predict who will need access at a given point in time. For example, an educational institution relationship for student research. |

|  |  |
| --- | --- |
| Result | |
| User constituency relationship type: | <<select the relationship type>> |

#### Determine which type of identity provider

This determines which of the four types of identity provider organisations can provide the required identity proofing services.

Leveraging an existing IdP wherever suitable should be considered as this provides the ability to leverage results of identity proofing across multiple relying parties (particularly important for higher assurance identity proofing activities which have a higher impost).

Agencies should not immediately assume they will be both the IdP and RP. If there is an existing IdP that services the user constituency, the agency could consider using then as an IdP, provided they meet requirements for:

* user experience
* user privacy
* identity assurance
* security.

The table below lists the four generic types of IdP. The list is not exclusive, nor are the models listed in any particular order.

| IdP Type | Description |
| --- | --- |
| A social media organisation as the IdP | * Social identity providers make available a standard credential for free on standard terms. * it can make more sense to let the user bring their own identity (BYOI) when: * The organisation has no existing user store for the user constituency * A low level of assurance is sufficient * pseudonymous or anonymous Identifiers are required * the organisation doesn't know:   + who the end user is (such as with a consumer prospect),   + the end user very well (volunteer or even short-term, low-level employee). * The strengths of social provided identities are as follows: * Are a proven approach for low-assurance use cases, especially for new users and transient relationships? * May be combined with step-up authentication, adaptive access and other trust-elevation methods for more sensitive use cases. * Able to provide SSO across web properties - many consumer websites are composite websites (constructed from multiple web applications) or may be part of a set of related websites. * Social identities tend to be more ‘sticky’ on a mobile device. That is, the login often persists for a longer period of time, so a single sign-on can be leveraged more times. * Volunteer collaboration: convenience for volunteers and reduces overhead for the organisation. Because many of their activities may be open, the fact that social identities currently tend to be low-assurance (not support a high level of security) is not a problem. * Acceptance of social identities can allow the organisation to delay local account creation until the user is fully motivated to go through the account setup process. The practice also reduces the cost and effort involved in account creation by defaulting user information from the social identity into the registration fields. * The weaknesses of social provided identities are as follows: * Most existing, free social identities are only suitable for a limited number of low assurance scenarios. To date, there has not been a market for freely available and easy-to-use high-assurance credentials. |
| Organisation as the IdP (and the RP) | * The organisation itself may be in a good position to be the IdP where the Organisation: * Is best placed to know who the end users are and to validate them? * Has an existing identity data store - this is commonly the case for employees (excluding high-turnover or lower-skilled positions) and customers? * The identity proofing processes if conducted by the organisation itself: * can be as stringent as need be * can be based on information available from the organisation and/or other sources the organisation has access to * The organisation may use an IDaaS solution or on premise solution * An organisation may need to directly register users due to: * legislation or regulation * the need to maintain historic information records * the need to manage accountability for users acting in particular roles or * the RP application supports long running processes that cloud be disrupted should the user be removed or no longer available. * Where the organisation undertakes the identity proofing, the remote credential is often little more than a token (i.e. shared secret) and may have little to no assurances of identity * Given the organisation is the IdP and RP, in some cases it may (as per QGAF) choose to recognise ‘known customers’. This refers to clients that may already have been registered for the same or other RP provided service rather than requiring re-registration. In this case, the previous registration must meet the required IRAL, consideration given to the suitability of existing credentials for use with the new service. * The organisation may act as an IdP for partners to consume as a RP. |
| Government as the IdP | * Accessing government services online requires a higher degree of identity proofing. As there is not a broad market where an individual can acquire a strongly verified identity for little or no cost, governments globally have created programs to issue high-assurance and reusable identities digital identities to their citizens[[2]](#endnote-2). * In some cases, the government itself has contracted to be the identity provider, and in other cases, the government has partnered with commercial organisations such as banks, insurance companies, postal agencies to be the identity providers. In some countries, these identity credentials can be used to access both government and commercial RPs[[3]](#endnote-3). * Government IdPs for customers typically have additional requirements to control personal information, consent models and stronger processes for identity protection to reduce risk of identity theft. The credential management strategy also differs to focus on providing choice of credentials which can be upgraded and a standardised account. * The Queensland Government commissioned a similar program (QGCIDM) in 2012 under the One Stop-Shop strategy. * Most Government agencies currently manage the identity of their customers, and in some cases act as a provider of identity information to other agencies (RP’s) and digital access or credentials relied up on for their own online service delivery. |
| Partner as the IDP | Sometimes, the organisation doesn't control the primary relationship with a group of end users. This is often the case with partner, vendor and third party NGO relationships. In this situation, it can be more effective to have the partner manage the user identities because the partner knows, for example, when its employees come and go. In situations where the partner has sufficient expertise and resources, it can make sense for the partner to run the IDP for its end users.   * The strengths of partner provided identities are as follows: * responsibility for maintaining user identities and other organisational information such as roles, delegations, cost centre information is delegated to the partner * usability –partner’s staff can use their existing credentials and in most cases have a SSO experience. * liability – Should the employee leave the partner, they will no longer have access, and the RP may still add additional authorisation policies. |
| An industry federation hub acts as the IdP | Federation hubs are hosted offerings that broker multiple organisations as identity providers and/or relying parties. Some industries such as higher education, aerospace and defence, and automotive, pharmaceutical and health sciences areas have used industry hubs for a long time. These operate under a variety of business models. A new organisation in an industry with an existing hub can save a lot of time and effort by joining an existing hub[[4]](#endnote-4).  As the use cases for federated identity increase, as well as the costs for providing high-quality identities, more alliances (federations) will be formed to provide strong reusable identities. Digital identity credentials that have been proofed to a high level of assurance are inherently expensive, unless the IDP is just digitising an existing trust relationship[[5]](#endnote-5). |

As a guideline, based upon the user constituency relationship type above the following IdP types should be considered.

|  |  |
| --- | --- |
| Condition | Consider |
| Individual relationship, known user | * Organisation is the IdP * if a small number of users e.g. may choose to register the users e.g. known volunteers for a drug test * Social media organisation as the IdP * if a large number of users * Government as the IdP * if requiring higher identity assurance * Industry federation hub as the IdP * If the hub also provides individual user authentication services |
| Individual relationship, unknown user | * Social Media Organisation as the IdP * Government as the IdP * if requiring higher identity assurance |
| Business-entity relationship, known user base | * Organisation is the IdP * if a small number of users * Partner is the IDP * if a large number of users * Industry federation hub as the IdP |
| Business-entity relationship, indeterminate user base | * Partner as the IdP * Industry federation hub as the IdP |

|  |  |
| --- | --- |
| Result | |
| Identity provider type/s: | <<for each user constituency>> |

#### Determine which organisation will be the identity provider

Once the type of identity provider and requirements are known, consult Appendix A for a list of existing identity provider organisations.

|  |  |
| --- | --- |
| Result | |
| Identity provider/s: |  |

#### Determine identity provisioning model

Identity provisioning across federated domains involves the setup of accounts, profiles, and/or access privileges for users, either ahead of or during the login process at the RP[[6]](#endnote-6).

Using the identity information from trusted third-party identity providers, the RP must use the information to uniquely identify the external user in the local context, verify that an account does not already exist for that user, and, if necessary, establish a linked user account.

**Is provisioning required?**

Determine the how the identities managed (proofed) by the IdP are registered/provisioned and maintained in the RP organisation/systems, or not at all.

|  |  |
| --- | --- |
| Option | Description |
| No registration (temporary access session) | * No persistent user account/record/identity is required by the RP * Common for pseudonymous access scenarios * Common for interactions which require temporary access to an agency resource or information that is not publicly available * Examples: * Online surveys * Authenticated information portals for volunteers * Authenticated access to online research and library’s |
| Provisioning required | * It is common for federated identity implementations to require user accounts or persistent identity data at both the IDP and RP domains. * In some cases, the IDP is the authoritative source for identity and provisioning is merely a means to supply the identity information for the RP. In other cases, both sites are authoritative sources for different attributes. * Ongoing management of provisioned user attributes must be considered e.g. de-provisioning.   Gartner discourages the creation of shadow accounts, unless there is a specific business need to do so, because managing the user data in two places reduces much of the benefit of using a third-party identity in the first place. Often, the compelling reason for a relying party/service provider to create their own account is that it enables them to maintain a local copy of additional attributes that they control and that can be used for personalisation purposes etc.  Identity provisioning actions may also include group membership entitlements in addition to users – common for federation scenarios to enterprise applications for Partner or enterprise cloud adoption.  It is also important to consider Identity de-provisioning actions (often de-activation rather than deletion) which may be required to ensure access is revoked, for licensing management, data archiving or data retention management |

|  |  |
| --- | --- |
| Result | |
| Identity provisioning require? | <<yes/no>> |

#### Determine the provisioning method

The following options are available for identity provisioning and/or de-provisioning:

| Model | Description |
| --- | --- |
| Direct registration | The RP provisions a local identity manually based upon a service request from the IdP. Manual entry is the simplest form of user provisioning. |
| Bulk load | The IdP provides a flat file for bulk entry which includes new accounts and updates to existing accounts. The file maybe generated by the IdP in an automated fashion using IDM tools. |
| Delegated self-service portal | RP provides a self-service portal and workflow for delegated administrators with the IdP organisation to request/provision access to systems. This typically implemented using the RP’s identity management (IDM) system capability.  Delegated administrators are given privileged access to create, manage and disable user accounts and their associated privileges. In addition to granting access, delegated administrators are often tasked with auditing users' access and performing access reviews.  The delegation model may be based upon administration boundaries (e.g. business units, application, geographic locations)  An alternative model if the RP does not have a self-service capability is for the RP to utilise a cloud hosted IDaaS service to provide the capability. This service can integrate directly with the RP systems or in a standard manner using federated provisioning standards. |
| Federated provisioning  (Run-time model) | An RP may choose to only provision an account when the user first attempts to access the application. This can be achieved in an automated fashion by leveraging a federation established for authentication. This is commonly referred to as Just in time (JIT) provisioning.  This approach may result in delayed access to the resource if the user must provide additional attributes not included in the identity assertion, but can prevent creation of unnecessary accounts which may never be used.  In this case, deprovisioning actions are manual or need to occur out of band through a reconciliation process. |
| Federated provisioning  (administrative model) | This approach pre-provisions user accounts from the IdP to the RP systems in an automated fashion prior a user accessing a resource. The provisioning process is not necessarily dependent upon federated authentication.  Administrative pre-provisioning and synchronisation may not be possible for Customer IdP’s which require prior user consent and selected attribute release.  Pre-provisioning may be required in cases where the RP systems require a persistent identity record e.g. email systems where a persistent mailbox, contact or calendar must exist prior to a user’s first login.  Federated provisioning is implemented using an orchestration authority.  If the IdP is not able to support federated provisioning, particularly if the partner is not large enough or technically capable, the RP may provide a hosted delegated self-service portal using a cloud IDaaS solution which supports federated provisioning. In a sense, the organisation ‘federates with itself’ as it owns the issuer. |

As a guideline, based upon the identity provider type above the following identity provisioning models should be considered:

|  |  |
| --- | --- |
| Condition (IdP Type) | Consider |
| A social media organisation as the IdP | * Federated provisioning (Run-time model) only * often due to consent/attribute release model and privacy enhancing design employed by Social IdP’s. |
| Organisation as the IdP (and the RP) | * Direct registration * Federated provisioning internal to other RD’s / RP within the organisation |
| Government as the IdP | * Federated provisioning (Run-time model) only * often due to the consent/attribute release model and privacy enhancing design employed by customer IdP’s. |
| Partner as the IDP | * Direct registration * Bulk load * Delegated self-service portal * Federated provisioning (Run-time model) * Suitable for relationships with external partners not tightly integrated and may be more dynamic in nature that alleviates the need to synchronise large user populations * Federated Provisioning (Administrative Model) |
| An industry federation hub acts as the IdP | * Federated provisioning (administrative model) * Federated provisioning (Run-time model) * Delegated self-service portal * Some federations also provide self-hosted IdP services or user authentication services (e.g. VanGUARD, Australian Access Federation (AAF)) |

The preferred option is federated provisioning (administrative or run-time) where possible. This model improves efficiency, lower costs and improves accuracy through automation.

|  |  |
| --- | --- |
| Result | |
| Identity provisioning model (if required): | <<list>> |

### Determine the credential provider

Organisations must determine one or more credential providers (CP) that cover the relevant user population (or induce their user constituents to register).

When selecting a CP there are a number of considerations outlined below.

It is preferable to re-use existing credentials where suitable (i.e. federate) as this reduces user friction, enhances user experience, especially for new user registration and recurrent login. This greatly reduces the need for password resets, especially for rarely used applications.

In cases where the IdP is a partner or the IdP is the organisation, the CP is also often same organisation as the IdP e.g. to allow the partner’s staff to use their same credentials

In some cases, multiple CP’s may be required to:

* support multi-factor authentication requirements
* provide user choice of credentials e.g. social login options

When selecting a CP, it is important to consider the:

* user constituency and likely credentials they may hold
* types of devices used and support for biometrics which may serve as additional factors
* level of trust placed on each credential e.g. strength and resistance to tampering

The following frameworks may be used as a reference the:

* One Stop-Shop Credential Provider Selection Framework
* US Government FICAM framework

### Determine the federation topology

It is important to plan the federation topology used to connect to IdPs or CPs – particularly where there are multiple entities. The RP needs to determine and known which services will rely on which authorities it will accept assertions and tokens from.

When determining the federation connection and relationships there are two key dimensions to consider:

1. The nature of the federation relationships, being either at a:

business level relationship – legal agreements between the two entities regarding assurance, roles, responsibilities, audit, privacy and liability.

technical level relationship – agreements between the two entity regarding standard, protocol, data formats and security.

1. The topology model, in which there are three typical models.

|  |  |
| --- | --- |
| Topology | Description |
| Point-to-Point | Where the organisation establishes a federation with one or more organisations directly. This arrangement will encompass both a trust agreement and technical integration. The federation may be unidirectional or bi-lateral. |
| Hub-and-Spoke | Where a single entity acts as a central point of communication and exchange between IdPs and RPs. The central entity facilitates and normalises the technical integration requirements such that IdPs and RPs need only integrate once with the central hub. Optionally, the central entity may enforce a common trust framework and accredit IdPs against the framework. |
| Networked | A peer-to-peer model where all entities are interconnected and can communicate and exchange data directly with all others. This model is similar to a scaled point-to-point model, however whilst the technical connection is point-to-point, at a business layer a common trust framework may be in operation. |

Across the Queensland Government and broader identity ecosystem all three models will exist.

#### Determine the federation topology

As a guideline, the following models should be considered when:

|  |  |
| --- | --- |
| Topology | Suitable when |
| Point-to-point | * A single organisation relies directly upon a single IdP * There are a small number of entity (IdP’s/CPs) (or partners with a close relationship between a few partners needing to exchange identity information) * Domains already have or can or can establish direct trust relationships * Arrangements for trust/identity assurance are bespoke in nature and not or can’t easily be standardised * No pre-existing identity hub is available |
| Hub-and-Spoke | * Needing to federate with a more than one IdP/CP * Needing to federate with a large number of entities * Avoiding the need to integrate * The federation comprises a community of domains centred around a single central organisation (which may be a large customer or a provider of services to the community) * If using an industry hub |
| Networked | * Needing to federate with a large number of entities, those entities also need to interoperate * No pre-existing identity hub is available – connections can be established organically as required * There are multiple IdP’s and/or multiple trust frameworks in operation |

Although point-to-point federations often deliver significant business benefits on their own, setting up multiple point-to-point or bilateral relationships can be an inefficient approach due to the need to establish business or legal agreements and technical interconnections for each relationship[[7]](#endnote-7). When establishing new point-to-point federations, agencies should look for opportunities to reuse existing federations or to create the point-to-point federation in a reusable manner[[8]](#endnote-8).

#### Determine the federation service (if required)

A federation service is typically required when an RP uses multiple IdP’s or multiple CP’s.

A federation service is a logical role that can be fulfilled by either an:

* IdP brokering multiple CP’s e.g. QGCIM brokering social media and business credentials
* intermediately federation service providing brokerage of common IdP’s or CP’s e.g. the Queensland Government Staff Federation Service or an industry federation such as VANGuard or the AAF
* RP brokering multiple IdP’s e.g. an agency RP federating with multiple Partner IdP’s or a web site supporting social logins and local account registration.

The last model is the most common where a single RP application must authenticate multiple user constituencies identified by different IdP’s and there is no existing established common federation service. Refer to Appendix A for a listing of well-known federation services available within and outside the Queensland Government categorised according to this blueprint.

The table below outlines for common user constituencies, the federation service options:

|  |  |  |
| --- | --- | --- |
| User constituencies | Federation service options | Considerations |
| **Organisation IdP + Government IdP** | | |
| Home agency staff + customer(s) | * The Agency RP ASA acts as a FS and federates with and the Agency IdP | Point solution |
| * The agency RP ASA federates with a government IdP (QGCIDM) which is already federated with all agency staff IdP’s via QGFAS | Public facing applications also requiring customer authentication should be federated with QGCIDM which can also authenticate Queensland Government Staff via a federation with QGFAS.  Queensland Government Staff shown as login option when configured for Application (not required if using an IdP initiated-SSO URL) |
| * The Agency RP ASA federates with QGFAS as a broker for staff and customer (QGCIM) identifies. | This model is not supported under the blueprint. |
| **Organisation IdP + Social IdP** | | |
| Home agency staff + customer(s) | 1. The agency RP resource acts as a FS federates with required social IdP(s) and the agency IdP (which may simply be local account registration) | Multiple social IdP may need to be supported. Application may provide build-in support e.g. social media walls and not support an external FS. |
| 1. The agency RP ASA acts as a FS and federates with required Social IdP(s) and the agency IdP |  |
| 1. The agency IdP acts as a FS and federates with required Social IdP(s) | Consolidation within an agency |
|  | 1. The agency RP resource, agency RP ASA, agency IdP federation with QGCIDM | Use of QGCIDM may provide SSO unless IdP-initiated SSO is support |
| **Organisation IdP + Partner IdP** | | |
| Home agency staff + partner(s) | 1. The agency RP ASA acts as a FS and federates with the agency staff IdP and partner IdP(s) |  |
| 1. The agency staff IdP acts as a FS and federates with partner IdP(s) |  |
| 1. The agency uses QGCIDM as a FS which federates with the agency Staff IdP (via QGFAS) and Partner IdP(s) | QGCIDM could provide EOI services for Partners if required |
| **Organisation IdP + Government IdP + Partner IdP** | | |
| Home agency staff + citizens + partner(s) | 1. The agency RP ASA acts as a FS and federates with the agency staff IdP, QGCIDM IdP and Partner IdP(s) |  |
| 1. The agency staff IdP acts as a FS and federates with QGCIDM and partner IdP(s) |  |
| 1. The agency uses QGCIDM as a FS which federates with the agency staff IdP (via QGFAS), QGCIDM (itself) and Partner IdP(s) | QGCIDM can provide EOI services for Partner identities if required |
| **Organisation IdP + Organisation IdP** | | |
| Home agency + other agency(s) | 1. The agency RP ASA acts as a FS and federates with the agency staff IdP and other agency IdP | Also consider other existing industry hubs |
| 1. The agency uses QGFAS as a FS which is already federated with all agencies | It is preferred to use the QGFAS service for any cross-agency federation, particularly when multiple agencies are involved. |

When using a central hub topology or providing a central service, there are a few additional considerations:

* Assertion proxying – A central hub may in some implementations simply acts as a router and proxy the assertion from an IdP to a RP. In other cases, the hub may generate a new derived assertion which is signed by the hub. This is common particularly where the hub provider accredits IdP under a common trust framework and it is the intermediately that vouches for the identity and must be trusted by the RP. In these cases, it may not be possible to determine the actual IdP, unless the full trust chain is preserved and included in the assertion by the hub.
* Pseudonymous identifiers or pairwise identifier algorithms – A central hub in some implementations may generate a different identifier for a subject per spoke (RP) connection to prevent a subject’s account at one IdP from being linked across one or more RPs through use of a shared identifier without the user’s consent. This is typically a privacy enhancing feature, although consideration needs to be given to the benefits of sharing a common identifier within a particular group of domains or RPs. For example, using the same identifier for the user within Queensland Government or cluster of agencies, but not across tiers of government. For more information on the use of Pairwise Identifier Algorithms, refer to the recruitment considered mandatory for identity providers in the OpenID Connect standard.

#### Determine the IdP discovery process (if required)

IdP discovery during authentication is a unique-to-federation problem caused by the need for user selection of an authentication source. Within a single domain, multiple authentication sources are often combined by ‘stacking’, in which credentials are played against multiple back-ends in some combination[[9]](#endnote-9). This doesn't work across domains because it would expose user credentials to authentication systems controlled by unrelated organisations[[10]](#endnote-10). As a result, the authentication source has to be selected before credentials are supplied, either explicitly through user choice, or by deriving something from a user identifier[[11]](#endnote-11).

There are a number of well-known patterns and options for IdP discovery (sometimes referred to as Home Realm Discovery)

* programmatically determine the user’s IdP based upon the authentication request:

contextual information e.g. the user’s IP range

an indication of the IdP/realm to use via an SP-imitated SSO parameter.

* interactive determination by the end user through:

selecting the IdP e.g. by name/logo

entering an identifier e.g. an email address which indicates the IdP e.g. the email domain name in Office 365 determines the organisation’s IdP.

A combination of these techniques may be used sequentially e.g. attempt to programmatically determine the IdP, otherwise prompt the user.

Once the user’s IdP has been discovered (particularly for interactive mechanisms), it is common to save this ‘preference’ using a temporary user storage mechanism such as a HTTP cookie to minimise the need re-ask the user for a period of time.

### Determine the trust framework

Addressing trust is critical to achieving the vision of an interoperable government.

Essentially an identity federation is a business model in which a group of two or more trusted (business) parties bind themselves with a business and technical contract to provide services to users.

Trust is ‘the willingness of a party to take action based on its relationship with another party’. Trust is developed through a number of building blocks, including business and legal agreements, shared policies, encryption and key management, assurance, and assertions.

The following definitions of identity and credential assurance are also relevant:

* Identity assurance is a measure of certainty that an individual, organisation or device is who or what it claims to be. Identity risk is the risk that an individual, organisation or device is not who or what it claims to be.
* Credential assurance is the assurance that an individual, organisation or device has maintained control over (binding) what has been entrusted to him or her (e.g., key, token, document, identifier, etc.) and that the credential has not been compromised (e.g. tampered with, modified). A credential risk is the risk that an individual, organisation or device has lost control over the credential that has been issued to him or her.

For a relying party to be satisfied that they are in fact dealing with the intended client, an understanding as the process used by the identity provider is required to determine the accuracy of the claimed identity for the requisite level of trust.

Identity proofing is evolving to become an ongoing risk-based data collection and analysis process that executes throughout the identity life cycle, rather than just the ‘trust’ part of a registration process. Gartner also states a hybrid approach to proofing is required as new identity-proofing models are based on user-controlled identities and trusted third-party identity providers. This requires a dynamic ‘step-up’-style proofing to bridge the gap between the level of trust in the third-party identity and business requirements.

There are two common approaches to inter-organisational trust:

|  |  |
| --- | --- |
| Model | Description |
| Static trust model (Levels of Assurance) | A trust framework and model that defines a small number of standardised and agreed levels of credential and identity assurance between federated parties. |
| Dynamic trust model (attribute driven) | A model that allows trust and confidence to be communicated between federated organisations at a granular level i.e. each attribute comprising an identity. This increases the ability to re-use and recognise previous verifications undertaken by other organisations (for example attributes which provide proof of name, proof of age, proof of address, proof of certification). |

For a detailed explanation, refer to section 10 - Trust models and frameworks. Use the decision table below to determine the most appropriate trust model:

|  |  |
| --- | --- |
| Model | Favour when: |
| Static trust model (levels of assurance) | Federated authentication (RP to IdP consolidation)   * There is an existing trust framework that is applicable and fit-for-purpose (or that can be used as a template) * The levels of assurance supported meet the business requirements of relying parties for identity and credential management with suitable variance that accommodates * The identity provider has a well-established user-base which RP’s desire to leverage (and are willing to accept trade-offs in requirements and standardise) * There are small number of RP’s with similar requirements * Federation of internal domains within an administrative boundary with the same security and privacy policies, risk management frameworks, compliance programs and accountabilities are clear.   Federated authentication (IdP to IdP Federation)   * Both IdP’s follow the same Trust Framework * There is close alignment between the IdP’s Trust Framework and an agreed equivalence between levels of interest OR the impact of re-verifying or is low or unlikely due to banded requirements |
| Dynamic trust model (attribute driven) | For federated authentication (RP to IdP consolidation or IdP to IdP federation)   * Any of the above are untrue. * There are multiple pre-established trust frameworks that are fit-purpose within respective domains. * Standardisation of business process requires large scale organisation change and funding AND the effect a process change may be an extended time based up on the frequency of identity re-verification.   Note: A dynamic model is more common for federation between established identity providers with pre-established trust frameworks.  Overtime, a level of standardised may be possible to promote progressive harmonisation of different trust frameworks and common requirements across organisations to further increase reuse of verifications as appropriate. |

An example of each model is provided below. The examples focus on third party assurance of identity. Relying upon a third parties for credential assurance can be achieved in a similar manner through sharing granular attribute metadata between federated parties. Credential assurance generally does not often present the same challenges as identity assurance because:

* a limited number of credential types exists and have been commoditised by the market
* credentials are relatively easier to describe in terms their characteristics e.g. the ability to resist tampering compared to various implementations of identity registration business process (which typically incorporate a level of human discretion).
* industry accepted frameworks exists to band credentials
* the relying party can easily elevate the trust as required by combing multiple credentials e.g. multi-factor authentication.

#### Static trust framework model (levels of assurance)

The diagram below depicts an example of a typical static model of trust for federated authentication based upon a central consolidated IdP.



In this this model:

* Relying parties ‘outsource’ and entrust identity registration (proofing) and credentialing processes to a common identity provider.
* The identity provider defines common ‘levels of assurance’ which represent specific business process designed to achieve the specified ‘strength’ of registration or authentication.
* A relying party undertakes a business risk assessment (involving data classification) to determine the required ‘strength’ of registration and authentication, and maps this to a comparable ‘assurance level’ offered by the IdP.
* It is common that there are only a small number of assurance levels implemented due to cost and standardisation of business requirements across relying parties.
* In this example, the identity provider establishes (registers) a new digital identity through verifying the subjects self-asserted attributes against the source issuer (in this case the paper document credential). The identity provider is this example supports multiple sources for a single attribute e.g. name and requires and algorithm to align/match attributes
* Once an identity is verified to the agreed standard, the identity is stored at this level by the IdP.
* The IdP also implements business processes to ‘upgrade’ the strength of registration or authentication as necessary based upon the strength previously achieved by the subject, and the strength required by the RP for a subsequent transaction.
* Should the IdP need to federate (accept an identity) from another IdP, both IdP’s must agree too and adhere to the same trust framework and level of assurances, including following the same business process for the levels federated.

#### Dynamic trust framework model (attribute driven)

Within an identity ecosystem encompassing multiple agencies, tiers of government and the private sector, different frameworks for trust and identity assurance will operate for various business purposes. It must be acknowledged that there is currently a wide variance in the policies, practices and standards followed to established identities within the Queensland Government, individual departments, business units and across jurisdictions. Subtle nuances in each organisations culture, policies, practices and standards can further hinder interoperability. Equally, there is no universally accepted or overarching framework for ‘describing the concerns that go into a determination of inter-organisational and transactional identity trust’.

Modern dynamic approaches enable trust and confidence to be communicated between federated organisations at a granular level i.e. each attribute comprising an identity. This provides a number of benefits:

* Increases the ability to recognise (and re-use) previous identity verifications and attestations undertaken by other organisations to verify the attributes of an identity (for example attributes which provide proof of name, proof of age, proof of address, proof of certification).
* Allows the identity assurance of the verification proofs used to be further communicated to a RP as the accountable entity through accompanying attribute metadata. This enables the RP to:

choose to trust all or some of the proof statements asserted by an IdP

maintain an auditable chain of custody when for attribute information that is obtained through one or more sources

evaluate the trustworthiness of an asserted attribute and implement additional enforce verifications where required

for sensitive use cases, know how an attribute was verified and how recently it was validated in order to determine if the attribute meets the RP’s internal requirements

rely upon third parties for portions of the identity life cycle and managing certain portions themselves.

* Allows an identity broker to assemble an ‘identity’ from multiple verification proofs as required to meet RP assurance requirements

As identity sources become more distributed, metadata about identity credentials and attributes becomes more important, and depending on the source of the identity, the relying party organisation may want to perform additional validation steps[[12]](#endnote-12).

A dynamic model favours transparency over the quality of the information, rather than getting all information to the same level of quality which is aim of traditional static models designed to operate on the principle of the ‘golden record’ where all identity information held must meet the same standard.

The inclusion of attribute metadata is used to convey information about a subject’s attribute(s) to allow for a relying party to obtain a greater understanding of how the attribute and its value were obtained, determined, and vetted. Specifically:

* At ‘runtime’:

the strength of the identity verification and registration process through individual attributes and the confidence of each (described through meta-data)

the strength of authentication, including contextual information such as device identity, location, health that may be useful to the RP in enforcing security policies

* At ‘design time’:

the strength of the credential, token issuance and management processes

binding between identity proofing and tokens (if done separately)

In a static trust model with pre-agreed processes a level of assurance (typically conveyed as a scalar number) is commonly the ‘low watermark’ of these components based upon design-time considerations only.

The diagram below depicts an example of a dynamic trust model for federated authentication using a central IdP based upon attribute meta-data. In this instance the central IdP also acts as a broker to consume (federate) granular identity attributes verifications from other identity providers.



In this this model:

* Relying parties ‘outsource’ and entrust identity registration (proofing) and credentialing processes to a common identity provider.
* A relying party undertakes a business risk assessment (involving data classification) to determine the required ‘strength’ of registration and authentication, and determines an access policy that reflects the identity attributes to be verified and the metadata such as the trusted source for the attribute and or number of sources to be verified against. Attribute requirements are typically described in business terms and reflects service eligibility requirements relating to proof of name, proof of age, proof of address.
* To meet the identity requirements defined by the RP, the IdP looks for any compatible existing attribute verifications undertaken by the user matching the policy. If suitable verification is held, the IdP guides the user through a process to obtain verification according to the RP policy. Once an attribute has been verified (proofed) the attribute data (or just the metadata) of the attestation is stored by the IdP.
* A user can assemble (and reveal) ‘proofs’ (or verifications) as required to meet a Relying Parties assurance requirements.

Departments wishing to implement a dynamic attribute based trust model should agree (negotiate) the metadata to be exchanged in addition to identity attributes and data formats as part of standard federation agreements. Refer to appendix B for example attribute metadata as suggested by the US Government National Institute of Standards (NIST).

#### Trust frameworks

Trust relationships are relatively easy to establish when the federating domains all exist within a single enterprise (or legal entity). Security and privacy policies may provide the necessary framework, if not, formal memoranda of understanding (MOU) can be established in lieu of contracts[[13]](#endnote-13).

When trust relationships extend across organisations, legal agreements should be established to outline roles, responsibilities and liabilities of both parties, identity data usage, audit or assessment criteria for compliance[[14]](#endnote-14). Legal agreements are expensive to negotiate, and negotiation is made doubly difficult when liabilities such as penalties or indemnifications are required[[15]](#endnote-15).

Examples of static assurance frameworks include the:

* Queensland Government Authentication Framework (QGAF)
* Australian Federation Government National e-Authentication Frameworks (NeAF)
* Australian National Identity Proofing Guidelines expand upon identity proofing/identity registration requirements of NeAF
* US Government NIST SP 800-63
* Kantara Identity Assurance Framework
* ISO 29115.

The IDESGs also publishes a number of ‘guiding principles’ for ‘trust framework providers’ to incorporate into their trust frameworks.

Choose a trust framework model between the RP and IdP.

|  |  |
| --- | --- |
| Result | |
| Trust framework model: | <<static (level of assurance) or dynamic (attribute metadata)>> |

If selected a static framework, a particular trust framework must be selected.

### Determine the authorisation model

Authentication is not a substitute for authorisation. Federated authentication protocols such as SAML and OIDC leave the authorisation requirements to the relying party.

There are two general models for how authorisation requirements are addressed for federated authentication.

* Local RP authorisation (with distributed IdP authentication) – authorisation enforcement is left to the relying party to implement a local authorisation model (typically role-based). The application must use the identifier from the federated identity assertion to link to the locally-maintained entitlements, or provision an account for new users.
* Identity provider authorisation (with distributed IdP authentication) – a level of authorisation (coarse-grained) is delegated to the authenticating organisation (the Identity Provider). Federated identity assertions contain ‘authorisation’ attributes which indicate to an RP entitlement such as a group or role entitlement known and managed by the IdP. This is in addition to other fine-grained authorisation policies implemented locally by the RP.

The first model is more conventional which separates authentication from authorisation. This allows fine-grained authorisation polices which control visibility of particular records or functionality that is typically to be managed by the RP.

The second model which delegates some coarse-grained authorisation logic has become more common (e.g. for enterprise cloud SSO) when the authorisation policy is simple (typically all or nothing) and applies to large numbers of users.

Where a fine-grained level of authorisation for a resource needs to be delegated to another party (other than the RP), this is best achieved using a delegated authorisation model which is based upon an industry standard designed to supports external authorisation and access control decisions using an expressive authorisation policy language that considers multiple conditions including the subject, resource and other contextual attributes.

#### Establishing the connection

This section is broken into two sections, depending upon the type of resource:

1. web application resources
2. API Resources for client applications

Access to API resources may also be required in addition to access to a web application resource. In this case, the existing Web SSO authentication session can be leveraged to obtain an OAuth token for API access.

#### Web application resources

**Determine authentication protocol**

Selection of the most appropriate identity federation standard is predominately based upon the Resource Type and in some cases resource ownership model or other relevant considerations.

The two relevant industry standards for federated authentication of a subject are:

* SAML - The web SSO passive flow is the most widely used flow for federated authentication. Other flows such as back-end attribute exchange (BAE) may be used to support authentication. Refer standards for a complete list of flows as per the SAML standard.
* OpenID Connect and OAuth - As the OpenID connect protocol is based upon the OAuth standard, with two of the common OAuth grant types (authorisation code and implicit) available in OIDC, including a newer hybrid flow which combines aspects of the authorisation code flow and the implicit flow. The other two OAuth flows (resource owner and client credentials) may also be used when using OpenID connect depending upon vendor support.

Note: The OAuth is a protocol designed for access delegation, although can be used for authentication purposes (delegating access to an API to retrieve user profile). This is a common approach for social provider logins, although the newer ODIC protocol provides similar functionality.

Whilst the federated organisations are encouraged to determine the most suitable authentication protocol and flow, the RP is often limited to what the IdP or the RP/protected resource can support.

As a general guideline:

* Use OpenID connect for cross-domain Web SSO authentication for first or third Party developed applications.
* Use OAuth for API authentication or OpenID Connect (as an OAuth based protocol) where identity information/claims are required for first Party or third Party developed applications. WS-Trust should be avoided unless the web service is SOAP based (although OAuth can still be supported).
* Use SAML only for Cross-domain Web SSO authentication for traditional sever-side applications where required by third Party developed applications due to vendor support (e.g. Cloud SaaS). Any first party developed applications should use OpenID Connect.
* The Shibboleth protocol/standard may be used if the IdP or RP is active in the higher education and/or research communities.

For a definition of each flow or grant type refer to the standards section.

This decision framework assumes federated credentials which identify an individual subject will be used. For system to system authentication for API, the following authentication schemes may be suitable:

* basic authentication – username/password
* key authentication - generate API keys for distribution to third-party developers
* HMAC authentication – shared secret.

**Determine authentication protocol flow**

Authentication protocols support a number of sub-flows (or grant types) for various purposes. It is important to select the most appropriate corresponding flow.

Flow selection generally depends upon the:

* type of resource
* resource ownership model
* type of tokens required (in particular refresh tokens for Session Management – see Session Management Strategy)
* need for an interactive browser-based session to support:

a variety of authentication mechanisms e.g. password, one-time password, adaptive etc.

layering additional interactive security measure (e.g. MFA)

layering interactive or consent models (more common for G2C scenarios – rate for enterprise).

* security considerations and options.

As a guideline:

* The authorisation code flow is intended for clients that can securely maintain a client Secret between themselves and the authorisation server, the implicit flow is intended for clients that cannot[[16]](#endnote-16).
* The implicit flow is best suited for:

native desktop applications

native mobile applications

Cross-platform applications

JavaScript clients in the browser (e.g. hybrid mobile apps, singe page applications (SPA) or progressive web applications (PWA))

* The authorisation code flow is best suited for:

traditional sever-side applications

stronger security (client authentication), particularly where PKCE is not used on Implicit flows.

* However, the authorisation code flow is sometimes also used by native applications and other clients in order to be able to obtain a refresh token, even when they cannot ensure the secrecy of the client secret value.
* The hybrid flow combines aspects of the authorisation code flow and the implicit flow to enable clients to obtain an ID token and optionally an access token with only one round trip to the authorisation server, possibly minimising latency, while still enabling clients to later get tokens from the token endpoint – especially a refresh token.
* The resource owner flow allows the user (resource owner) to authenticate directly at the RP by providing their credentials (username/password) rather than redirecting the user to the IdP. For example, where the IdP may act as a credential provider.
* The client credentials flow is for unattended services and non-interactive clients (e.g. CLI’s, Daemos or backend services) that require access to an API.

For more information regarding the tokens (access, ID or refresh) provided for each flow and protocol, refer to Appendix X - OpenID connect flows and OAuth grant types.

For web application resources, OpenID connect is preferred over SAML. OpenID connect for authentication provides a cross-domain, interoperable identity token. First developed party applications developed by or for government should support OIDC from the outset. If the application is provided by a third party, the choice of authentication standard may be constrained. third party cloud service typically only support SAML.

**Determine connection termination endpoint**

The termination of the connection (token validation) can either occur directly at the web application resource (if claims-aware) or a SAML or ODIC federation provider may be used even if the resource is claims-aware to provide additional functionality such as the following:

* offload the connection (token acceptance, validation, key and certificate management)
* provide trust elevation e.g. setup authentication or risk-based controls
* augment an identity assertion with additional claims e.g. against a local data source
* act as an ASB for protocol translation should the resource not be claims-aware (proprietary or standards-based)
* provisioning of user accounts
* additional authorisation logic, including acting as an ABAC PEP
* acting as a federation service to broker multiple IdP’s or CP’s.

**Determine if protocol translation is required**

Protocol translation or bridging is required when either the IdP or RP does not support the desired authentication protocol determined above.

Translation in some cases can be solely achieved by the RP, or require co-operation with the IdP as well or an intermediately STS.

The table below lists common protocol and token format combinations and options to translate between.

|  |  |  |  |
| --- | --- | --- | --- |
| IdP | RP | | Protocol and Token format translation options |
| Protocol (Token Format) | Protocol (Token Format) | Resource Type |
| SAML | OIDC (JWT) | Regular Web Application | The RP ASA will need to act as a Federation Service to broker a SAML RP federation trust with the IdP and correspondingly act as a OIDC IdP |

In some cases, the actual protocol version used by IdP and RP (e.g. SAML 1.1 to SAML 2.0) may not match. There is generally no standard means for major version differences to translate protocol versions unless the RP is supporting a higher version and the standard supports backwards compatibility. Industry standards in general, typically promote migration paths allowing the two versions to operate in tandem. This means the IdP or RP must support the same protocol version.

#### Determine the assertion format

RP must identify the claims information required to communicate a successful authentication event and represent the identity to the RP. Specifically, the:

* authentication claims – the subject identifier, authentication method/means
* identity claims - identity and preference attributes

Required e.g. to support service personalisation, provisioning and identity enrolment requirements or authorisation requirements.

Determine how these attributes will be sourced (authoritative sources)? What level of verification? (using verified or user-asserted attributes).

* authorisation (entitlement) claims

The entitlement attributes required to support authorisation/access control decisions.

Refer section IM identity assertions for more information on each type of claim.

Identity and authorisation claims may be sourced from the IdP or other sources the RP deems authoritative and trustworthy. An attribute issuer that is not authoritative for the attribute it issues should not be trusted. As an example, an attribute indicating proof of training and training status may be held by another organisation and is not gathered as part of the identity proofing process.

|  |  |
| --- | --- |
| Result | |
| Authentication claims: | << list attributes required and authoritative source(s)>> |
| Identity claims: | << list attributes required and authoritative source(s)>> |
| Authorisation claims: | << list attributes required and authoritative source(s)>> |

#### Determine identity attribute sourcing methods

If Identity or entitlement attributes are to be sourced:

from the IdP, determine whether this is to occur via the front-channel or backend-channel below

from an attribute provider or other authoritative source refer to the federated identity attribute exchange models.

The table below outlines various options:

| Option | Front-channel assertion | Back-channel query  (single subject) | Back-channel (multiple subject) |
| --- | --- | --- | --- |
| Strengths | The identity assertion from the IdP produced at the authentication event also contains identity and entitlement attributes.  The subject is directly involved in the process, typically as part of the authentication event.  Provides the ability for the user to consent to release attributes  Example:  SAML Web Browser SSO Profile | The RP obtains attribute information post the authentication event via a direct connection to the IdP in real-time.  Examples:  SAML backend attributes exchange (BAE) profile using the subject’s Identifier.  OIDC user info endpoint using the opaque token. | A RP fetches or synchronises attributes of interest for a number of subjects ahead of an authentication event.  This option essentially represents the federation identity provisioning model.  Examples:  SPML or SCIM |
| Weakness | Dependent upon security of user agent and transport. | Additional attributes can be fetched if required for RP enrolment/identity resolution.  Consent or saved attribute release preferences remain possible but may be complex. | No option for user consent or selected attribute release  RP must store of identity data for all relevant subjects. |

#### Determine the identity resolution process

If identity is defined as a set of attributes that uniquely describe an individual, identity resolution is the confirmation that an identity has been resolved to a unique individual within a particular context (the set of identity records it holds).

Enrolment is defined here as the process by which a link is established between the IdP identifier of a person and the identifier used within an RP to uniquely identify the record of that person. Therefore, the next time the same IdP identifier is presented, the RP can find the associated record based upon the previous mapping to grant access to that account.

Attribute matching can occur against verified identity attributes from the IdP - typically core biographic information, for example full name and date of birth (partial/full). There may be data-matching problems where departments have old or slightly different information; such as previous addresses or maiden names; or addresses and names in different formats[[17]](#endnote-17).

It is common that the identity attributes provided by the IdP alone may not be sufficient to achieve a match in the local RP context (or a higher degree of confidence is required). Additional shared private information known by the subject and issued by the RP can be used to establishing a link between the IdP identifier such as:

* a local RP identifier - a locally unique shared piece of information between the subject and the RP – typically an identifier that is already established/well known e.g. driver license number
* personally Identifiable Information held by the RP – e.g. address information (postal code, city, state, country)
* other information held by the RP which is unique to the service, business process or entitlement e.g. car registration number, make/model.

This information may typically have been established previously as part of an offline relationship. Information used for matching does not constitute proof of identity as such information is already known to the RP and is used only as a claim of identity by the subject to facilitate a match to an existing record. Before identity resolution, the subject must still undertake an identity proofing process to verify that it is indeed that person making this claim, and not an identity thief who has simply obtained the shared information.

Matches need to be associated with a confidence level. For example, the following bands:

* a 'clear' winner e.g. 98%, no close 2nd
* multiple plausible e.g. 98-90%
* no plausible candidate e.g. > 50%

Matching may occur pragmatically (in an automated fashion) or required manual intervention for lower confidence cases.

The American National Standards Institute ([ANSI) / [North American Security Products Organisation (NASPO)](http://www.naspo.info/) Identity Proofing and Verification (IDPV) Standard Development Project](https://blog.aniljohn.com/2013/05/an-emerging-standard-for-identity-proofing-and-verification.html)) has developed minimum requirements for ‘reconciling an asserted identity to a single individual’. The standard defines five attribute sets across the dimensions of Name, Location, Time and Identifier (social security number) that provide an equivalent level of effectiveness when it comes to identity resolution to a single identity. This was determined based upon extensive data modelling and analysis on LexisNexis data sets that covered the U.S. population. As an example, attribute Bundle #2 (full name and full date of birth) provided a ~ 96% resolution or Attribute Bundle #1 (full name + partial address + partial date of birth) which provided ~97% resolution[[18]](#endnote-18).

The standard provides a list of supplemental attributes that can be used to prevent collisions in cases where the core attributes are not enough.

The standard recommends that the following factors be used to choose one particular attribute over another:

* EFFECTIVENESS: How EFFECTIVE is the attribute at distinguishing an identity?
* SENSITIVITY: Is the person SENSITIVE about the data attribute? Is the individual concerned that providing the requested information may make them vulnerable to harm?
* ACCESSIBILITY: Can the enroller verify the attribute? Does the enroller have ACCESS to resources that have details about the attribute?
* PERMANENCE: Is this attribute stable over time?
* UNAVOIDABILITY: Is the attribute something that is required to be collected as part of a business processes or for other reasons?

The Federal Government Better Practice Guidelines for Data Matching 2009 developed under the National Identity Security Strategy is also a good reference point.

#### Determine provisioning protocol and flow (if using federated provisioning)

If federated provisioning has been selected, the provisioning method and/or federation provision protocol needs to be determined.

|  |  |
| --- | --- |
| Identity Provisioning Model | Options |
| Federated provisioning  (run-time model) | The RP is responsible for provisioning the identity locally into its systems (typically using an orchestration authority).  The identity attributes are sourced either from:   * the identity assertion presented via the federated authentication protocol selected (front-channel) * back-channel fetching of attributes from the IdP or other authoritative sources * aggregated locally from additional RP attributes   The federated provision standard (SCIM) can also be used for run-time (just-in-time provisioning) as SCIM bindings are available for SAML to carry identity attributes. |
| Federated provisioning  (administrative model) | The RP and IdP must agree and establish a connection for federated identity provisioning (separate to that required for federated authentication) in order to synchronised identities between domains.  There are two established industry standards for federated provisioning standards, however to date proprietary implementations are more common. Refer to the Standard for guidance.   |  |  |  | | --- | --- | --- | |  | **SCIM (preferred)** | **SPML** | | **Published** | 2015 | 2007 | | **Schema** | Defines common schema for users and groups and is extensible | No standard object schema | | **Data Forma** | JSON | XML, DSML | | **Protocol** | REST over HTTP | SOAP over HTTP | | **Authentication** | OAuth2 or other | Proprietary | | **Meta-data** | Meta-data for schema and functionality discovery | Not supported | |

#### Determine the session management strategy

The session management strategy depends on the type of resource and federation protocol. It is important to select the session management approach that aligns with requirements for revoking or expiring user access.

|  |  |
| --- | --- |
| Resource Type | Description |
| Regular n-tier web application | Although details vary based on the exact federation protocol in use, for web applications, an assertion should be used only to represent a single log-in event at the RP. After the RP consumes the assertion, session management at the RP comes into play and the assertion is no longer used directly. For browser session, often the RP will exchange the token for local credentials e.g. a session cookie.  There are a number of considerations surrounding logout (localised IdP only or federated single logout which can occur via a front or back channel). |

Determine SSO for mobile client applications (if resource type is mobile client)

If SSO is required across one or more mobile native applications using the same IdP on a single device, the following two options regarding use of either the system web browser or embedded web-view for federation exists. Neither selection is ideal as both options described below have security and usability compromises.

Note, as of the last quarter of 2015 mobile SSO is now possible as mobile Operating Systems now provide the necessary services for an SSO framework for native applications. Both iOS and Android have added ‘intermediate browser’ functionality that will bridge the gap between the system web browser and an embedded browser component[[19]](#endnote-19).

|  |  |  |
| --- | --- | --- |
|  | Embedded web browser control | System web browser |
| Security | Mitigates MITM attacks on access codes and tokens don’t pass through the mobile OS | The use of the system browser introduces MITM attacks as authorisation codes and tokens pass through the OS on their way from the system browser to the application.  Note that the use of the emerging proof key for code exchange (see the PKCE proof of possession section) can mitigate these attacks. |
| Security trust marks | The embedded browser has limited visual cues to help the user understand the security posture of the web session | Standard visual security cues of web browsers are present |
| User credentials | Less secure because user credentials are passing through the application. This may suitable for first party developed applications where the RP and IdP are trusted, however is not suitable for untrusted third party developed applications using a government IdP | More secure as user credentials are provided directly to the IdP, without needing to trust the intermediate application. |
| Usability | Better usability as the user never leaves the application | Dependent upon two separate interfaces |
| SSO support | Yes. This was previously not possible prior to iOS and Android OS updates.  Note: At this point in time embedded web views cannot access X.509 certificate stores (may be relevant for corporate/enterprise developed mobile apps) | Yes |

#### Determine the assertion security model

Both SAML and ODIC support signed and encrypted tokens. In the case of OIDC, these are JavaScript Object Notation (JSON) Web Tokens (JWTs) that have been signed using the JavaScript Object Signing and Encryption (JOSE) framework.

Refer to the US Government NIST 800-63 e-Authentication Framework v4 Part C Federation and Assertions for guidance of the following:

* assertion possession

holder-of-key assertions

bearer assertion.

* assertion presentation

indirect presentation

direct presentation

assertion proxying.

* assertion protection

signing (symmetric or asymmetric key algorithms)

encryption.

#### Certificate and key management for token transport security, token signing or token encryption

Public-key infrastructure (PKI) is an important consideration for federation as it extends a fabric of trust across boundaries.

There are generally three primary interactions for Federated Authentication that need to be secured:

* transport layer security - certificates used to establish a secure sockets layer/transport layer security (SSL/TLS) session for message confidentiality and integrity between federation endpoints
* token signing - digital signatures used to ensure the identity of both parties and to validate that a message was received from a particular partner
* token encryption - an optional mechanism to achieve increased privacy.

The last two apply when using an RSA signing method for assertion protection for SAML or ODIC standards.

The chosen certificate authority (CA) could reside within the IdP’s Management Domain or be a well-known public/commercial CA. There is no specific guidance regarding which CA to use for a given trust relationship. Private CA’s may provide for stronger assurance over enrolment which may be more important than the convenience of public CA’s.

#### API resources

The following guidance applies to establishing a connection from an Identity Provider to a relying party API.

API authentication (authorisation) can occur in a direct model where the API is the only resource being accessed, but is mostly typically combined with a federated authentication flow for a web application resource which authenticates identities where they are stored, and then needs access to remote API’s on behalf of the user. The IdP leverages the existing web SSO session to avoid the need to re-authenticate access to the remote API’s. This may also involve federation where the API is in another domain.

#### Determine authentication protocol flow

For resource-driven applications based, OAuth should be used as a resource delegation protocol to allow access to the API resource to be authorised by the resource owner e.g. the end user. Using this protocol, the identity provider performs the role of ‘authorisation server’ to authenticate the user and authorise access, whilst the relying parties performs the role of the ‘Resource Server’ protecting the resource and granting access based upon the access token provided by its trusted authorisation server.

The JWT token format may be used instead where Identity claims are required by the API resource as opposed to the standard access token.

OAuth should be used for REST-based API’s. WS-Trust is generally paired with SOAP-based API’s. OAuth is preferred for modern REST-based API’s. New API’s should not be developed using SOAP which is considered legacy. Token translation to support existing and third party SOAP-based API’s is addressed below.

Note: Identity federation standards e.g. OAuth are only required for API access where the API resides in another security domain. If both the web app and the API are served from the same domain (according to the web browser standards for cross-original requests which allows sub-domains to be consider under the same higher order domain) simple cookie-based authentication already deployed for the web-application can be used instead of token-based authentication. Both cookie-based authentication and token-based authentication can be mixed as required where there are multiple API’s and security domains.

#### Determine the authorisation server and resource server roles

There are four common architectural models based upon the following considerations from a fully collapsed to fully distributed model.

| Model | Considerations |
| --- | --- |
| 1) Use the IdP as an AS and the resource itself as the RS | No RP infrastructure required.  Typical for modern API’s that natively support claims-based authentication. The API resource itself validates tokens and enforces the required scopes. |
| 2) Use the IdP as an AS and RS (direct trust) | No RP infrastructure required.  Directly trusts the IdP as both an AS for the API to authenticate users and RS to protect the API, validate tokens and enforce required scopes.  Typical for modern API’s that natively support claims-based authentication, where there is a high level of trust in the IdP (either due to the IdP being operated by the same organisation or a trusted organisation). |
| 3) Use the IdP as an AS, RP operates own RS | RP infrastructure required.  Directly trusts the IdP as an AS for the API to authenticate users, however the RP operates its own RS to validate tokens and enforce and required scopes.  Typical for modern API’s that natively support claims-based authentication or where the RP uses an API Gateway or ESB that provides the RS function to offload this requirement from the actual API resource. This may be a shared function/service across multiple resource domains. |
| 4) RP operates own AS (as an IdP) and operates own RS | RP infrastructure required.  Common where an RP which already operates an AS to manage authentication to its API resources (the RP is also an IdP).  The RP’s own AS needs to ‘federate’ with another IdP’s AS to accept/trust its previous authentications.  This is a typical scenario for third party cloud services e.g. SaaS services where the provider supplies a mobile application and allows its users to authenticate. The cloud provider may also allow its enterprise customers to federate with their own IdP in addition.  ‘Federation’ with another AS requires Token Translation via a security token service provided by either party (also referred to as API delegation). The following options exist:  the RP AS to be capable of accepting a token from the IdP AS and issuing a local token (this typically involves local or remote token validation – see below)  the IdP AS being capable of generating a bearer token request that the RP AS accepts and issues a local token  the IdP AS being capable of issuing a token a token on be-half of the RP AS (this may involve the IdP AS generating the actual token or the IdP AS requesting a signed token from the RP AS)  Some identity products can ‘short-circuit’ this process where they host both the IdP role and RP roles e.g. The IdP can generate a SAML assertion or OIDC JWT and also request an OAuth token at the same time. |

Within the Queensland Government, models 1 -3 are common for simple user cases where the agency doesn’t already operate as an IdP and authenticate users (as in the 4th model).

#### Determine Protocol Translation (required only for model four above)

|  |  |  |  |
| --- | --- | --- | --- |
| IdP | RP | | Protocol and Token format translation options |
| Protocol (Token Format) | Protocol (Token Format) | Resource Type |
| SAML | OAuth | REST API | * Both IdP and RP must support the RFC7522 SAML 2.0 bearer assertion Profile. The IdP must be capable of generating a SAML bearer assertion (typically requiring a WS-Trust STS) in which the RP ASA must be capable of accepting in order to provide a local OAuth access token in response for subsequent API requests * Should the IdP be capable of acting as an OAuth AS or STS for the RP resource (generally within the same management domain), it may be possible for the IdP to generate an OAuth token and include it in the SAML assertion |
| SAML | ODIC (JWT) | REST API | * As per the above. The RP ASA will return a local token in the format required by the ASA i.e. a JWT in this case. |
| SAML | WS-Trust (SAML) | SOAP Web Service | * The web service client (WSC) via the IdP or STS must be able to generate a WS-TRUST assertion |
| OIDC (JWT) | OAuth | REST API | * To accept a JWT, the RP ASA needs to support RFC7523 JSON web token bearer profile such that the RP ASA accepts a JWT bearer assertion and provides a local OAuth access token in response for subsequent requests |
| OIDC (OAuth) | WS-Trust | SOAP Web Service | * The IdP should embed the OAuth access in header. |
| OAuth | WS-Trust | SOAP Web Service | * OAuth access token into the standard 'sessionID' SOAP header to make subsequent calls to the SOAP API. * The IdP or an intermediate STS can covert an OAuth Bearer token to a WS-TRUST token with embedded SAML |

Note:

* The SAML and JWT bearer assertion are popular OAuth extension grants support leveraging an existing Web SSO Authentication to obtain an OAuth token for API access. Refer standards for IdP and RP support considerations
* WS-Trust is a standard which is specifies how to encrypt/decrypt/sign SOAP messages and how to bind security tokens to a SOAP message. SAML tokens are an example of a security token that could be bound to a SOAP message[[20]](#endnote-20). WS-Trust may include/support other less common web services security (WSS) token formats other than SAML such as Username, X.509, Kerberos, Open Token or other proprietary WAM binary tokens which are not shown.

**API chaining**

API ‘chaining’ (API to downstream API) is another scenario that may arise when a resource (System/API) in one domain needs to call another resource (system/SPI) in another domain in context of a particular user. API chaining requires either party to provide as STS (as per model four above). The downstream token may be more narrowly scoped.

Determine the session management strategy

The session management strategy depends on the type of resource and federation protocol. It is important to select the session management approach that aligns with requirements for revoking or expiring user access.

|  |  |
| --- | --- |
| Resource type | Description |
| Web API | API access tokens can be set to expire after a set period of time. Refresh tokens provide a means:   * to determine if a session should be extended (when a client requests a new access token) * make the session indefinitely renewable/persistent * revoke access if the situation changes   Access tokens should be given lifetimes long enough to prevent users from having to authentic multiple times during a single work session, however short enough to be able to revoke access should the need arise. Some implementations can blacklist/revoke individual access tokens for immediate revocation.  A benefit of tokens over passwords is that access to a particular device or service can be revoked, whilst others retain unaffected. |

#### Determine token presentation of tokens (for REST API’s)

The OpenID Connect and OAuth specifications do not define how bearer tokens should be presented in REST requests (some applications present tokens in the message body or request header). As a general rule the following standard should be followed RFC 6750, The OAuth 2.0 Authorisation Framework: Bearer Token Usage.

#### Determine token validation model

An important consideration for API authentication is how the access token is validated by the by the resource server or resource to ensure the token has been tampered, has expired or been revoked.

* Local validation – where the ASA/resource server validates the token signature using a pre-exchanged secret, issuer and expiry checks
* Remote validation – where the ASA/resource server calls the issuing AA’s validation URL to validate the token which can include additional checks (as outlined below).

The OpenID Connect and OAuth specifications do not define a standard for remote validation of an access or identity token by an authorisation server. Some vendor implementation provides propriety OAuth grant type for access token verification.

A single or combination of both approaches can be used. There can be a performance impact of with remote validation where the issuer resides in another security domain, although this can be mitigated through local caching/whitelisting.

|  |  |  |
| --- | --- | --- |
| Validation criteria | Local validation | Remote validation |
| Token hasn’t been tampered with | yes | yes |
| Token hasn’t expired | yes | yes |
| Token hasn’t been revoked | no | yes |
| Account hasn’t been disabled, and hasn’t been deleted | no | yes |
| Issuer is valid | yes | yes |
| Issuing application is still enabled | no | yes |
| Account is still in an account store for the issuing application | no | yes |
| Validation against authorised scopes | Yes | yes |

## Federated delegated authorisation

### Determine delegation model

As outlined in, there are three main models for establishing delegated authorisation:

* single resource delegation (user controlled consent) – where an individual owner of a resource delegates access (through consent) to a third party application to act on their behalf.
* central delegation across Resources for party-to-party information sharing (user controlled consent) – where an individual owner of one or more resources, uses a central authority to manage delegations for one or more requesting parties (being individuals, organisations or a mediated agent of the requesting party)
* central delegated authorisation (enterprise controlled) – where access control decisions for multiple resources are delegated to a central authority which executes an agreed policy e.g.

|  |  |
| --- | --- |
| Result | |
| Delegation model: | << option 1,2 or 3>> |

### User controlled delegation (consent based)

#### Determine protocol

Different industry protocols have evolved to support each use case:

* OAuth provides a simple method for a user to interactively delegate access to third party applications they use as required
* UMA is centred around an individual to managing predefined delegations for multiple resources and extends the concept of delegation to include authentication of the requesting party. E.g. I want to share this information selectively and proactively among my own apps, with family and friends, with organisations.

The table below outlines the main differences between OAuth and UMA.

|  |  |  |
| --- | --- | --- |
|  | OAuth (basic) | UMA (advanced) |
| Overview | Designed for an individual to delegate (grant) access to a resource they owner by a third party application client. | Designed for centralising the authorisation process for distributed resources. The authorising user may be granting access to a truly autonomous party |
| Obtaining consent | The resource owner must be present at time of the access request from the third party application client to grant or deny access. | The resource owner can manage access policy to their all their registered resources across multiple parties via central portal at any time. |
| Governed resources | Supports one resource access delegation per request | Supports managing access permissions for many resources provided by many different parties. |
| Access | The authorisation server provides access based on the resource owners ability to authenticate. | The authorisation manager grants access based on the user policy and the ‘claims’ (requester identification) conveyed by the requesting party |
| Architecture | The resource server is paired with its own authorisation server for access control. | A central authorisation server manages access control for multiple resources enrolled by the user. |

#### Determine resource owner authentication process

The process to authenticate the resource owner can be either handled by the Authorisation Server, or the Authorisation Server may federate with an external IdP via Federated Authentication (see decision framework)

#### Determine requesting party authentication process

The same process as above applies.

#### Determine entitlement (scope) design

When using these protocols for delegating third party access to a particular resource, the resource owner is asked to grant/approve specific entitlements. Suggested entitlements are requested by the third party, however the resource owner may only consent to a subset.

‘Scopes’ are a concept in the OAuth and UMA delegation protocols that are used to describe entitlements (permissions). Access to a resource requires specific scopes.

The application client must request the desired scopes from the authorisation server, which facilitates obtaining consent from the resource owner to authorise the access delegation. The protected resource compares the list of scopes (entitlements) granted in the access token against the scope required for the requested API or API function.

The specificity of the scopes may be either:

* fine-grained resource entitlements (e.g. READ\_PROFILE, INCREMENT\_FAVOURITES)
* coarse grained scopes representing a logical group of API functions, an entire API (e.g. BOOKS\_RESOURCE)

If the resource requires a basic set of permissions to function, optional permissions should be used to address advanced functionality at time of use to avoid initially deterring users from approving access. The same OAuth authentication flow can be re-run to grant access to additional scopes as needed.

#### Determine if an additional authorisation layer is required

The resource server owner must determine if the resource owner’s consent/authorisation alone is sufficient or if the resource server (the relying party) protecting the resource must also implement an additional access control layer.

For example, whilst the owner of a bank account (e.g. protected resource) can delegate someone else access to obtain or transfer funds, the bank (e.g. resource server) will still have final veto over whether the transaction goes through based upon business rules such as transfer limits. The New Zealand office of the GCIO ran a Proof of Concept (PoC) also confirmed this requirement in a Government context citing ‘an agency’s business rules may prevent user delegated access under certain circumstances or when legislation specifically allows or denies access to that resource’[[21]](#endnote-21).

Whilst a simple list of entitlement statements approved by the resource owner as consent may be well suited for consumer-based applications, this static authorisation model alone may be insufficient to address other authorisation requirements which must factor in multiple dimensions such as the Subject, Action, Resource and Environment Context. For example, only the resource owner will be able to access the resource during business hours Monday-Friday. The OAuth Authorisation server or resource server may utilise an ABAC model (see central delegated authorisation (enterprise controlled) model below) to create an XACML request that renders an access control decision based upon the subject, resource (the API action, user consented scope) and other contextual information e.g. time of day.

### Central delegated authorisation (enterprise controlled)

XACML provides a declarative fine-grained policy language for access control decisions implemented through separating layers of logic that enforce policy and make decisions about policy.

## Federated attribute exchange

Possible models for sharing identity information (attributes) varies depending up on the type of identity.

### Employee/staff Identity Information

The industry has developed common approaches and technology products to propagate/share staff identity attributes between corporate systems typically within a single enterprise given employee identity has a reasonably well-defined schema and is primarily stored within enterprise LDAP directories.

There common three patterns are:

* virtual directory (pull model)
* identity manager (push model) using either proprietary protocols or industry standards (SCIM)
* meta directory (hybrid).

For more information, refer US Government Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Guidance, 2011.

### Client and customer Identity Information

Due to the bespoke nature and diversity of this information there are not standard mechanisms for federated exchange. The patterns and mechanisms for the sharing of structured data (e.g. tabular records) across systems and entities is not necessarily unique to Identity data which needs to be shared as part of Identity and Access Management business processes.

As a general rule start by determining:

* Which attribute providers are considered authoritative for a given set of attributes
* If a single or multiple sources are required for the same attribute
* If an attribute broker is required, and which domain (local, intermediary) they reside
* Validation and/or query flows, including the use of shared identifier requirements

OpenID connect flows and OAuth grant types

Vendor implementations can often vary from the guidance in industry standard to address common business needs and normalise behaviour. For instance:

* refresh tokens may not be provided unless explicitly requested (e.g. via scopes)
* refresh tokens may be provided when requested on flows (e.g. implicit flow) not supported by the standard
* refresh tokens may be long-lived (never expire) or have a controllable expiry
* refresh token may be able to be administratively revoked (user or device)
* an access token may be able to be administratively revoked (user or device)
* the refresh token itself may be an access token or ID token
* ID token may only be provided when requested (using scope=openid)
* ID tokens may still be provided on flows (e.g. resource owner credentials) not supported by the OIDC standard

|  |  |  |  |
| --- | --- | --- | --- |
| Grant Type | Protocol | Client Authentication | Tokens Provided/Available |
| Authorisation code | OAuth  OIDC | Client ID Client secret | Access Token  ID Token  Refresh Token |
| Implicit | OAuth  OIDC | Client ID | Access Token  ID Token |
| Hybrid | OIDC | Client ID Client secret (optional) | Access Token  ID Token  Refresh Token |
| Resource owner credentials | OAuth | Client ID | Access Token  ID Token\*  Refresh Token\*  \* conditional – see above. |
| Client credentials | OAuth | Client ID  Client Secret | Access Token |

Note: No refresh token is available or SAML Bearer or JWT Bearer extension grant types.

Attribute metadata standards

The US Government National Institute of Standards (NIST) recently released (public draft as of July 2016) an Attribute Metadata schema (NISTIR-8112 related to special publication NIST 800-63-3 e-Authentication Framework) to ‘support cross-organisation confidence in attribute assertions’, as well as the semantics and syntax required to support interoperability. The inclusion of attribute metadata is used to convey information about a subject’s attribute(s) to allow for a relying party to obtain a greater understanding of how the attribute and its value were obtained, determined, and vetted.

The report states ‘this schema is intended to demonstrate the value of attributes and attribute value metadata in supporting U.S. Federal Government use cases and it is envisioned that the core set of metadata proposed here can serve as a library or menu from which both commercial and federal implementers can draw common semantics, syntaxes, and values to support their specific needs’.

This document introduces some of the technical, policy, and implementation considerations associated with the development of a schema by which federal agencies and other organisations could make risk- based decisions that are informed by confidence in attributes.

The optional schema outlines 15 metadata elements across nine categories that can be associated with an attribute. A summary and examples is listed below:

* Origin (origins name)
* Provider (providers name)
* Verifier (origin or provider)
* Verification Method (document verification, record verification, proof of possession, unverified)
* Pedigree (authoritative/sourced/self-asserted/derived)
* Verification Frequency
* Last Verification Date
* Last Refresh Data
* Expiration Date
* Acceptable Use (authorisation, secondary use, no further disclosure)
* Releaseability (none, public release)
* Classification (unclassified, In-confidence, protected)
* Individual Consented and Date
* Caching and Data Deletion Policy

The standard was developed off the back of a preliminary discussion paper released in December 2015 entitled Attribute Metadata and Confidence Scoring’, which sort to:

* Defining standardised attribute metadata that organisations can use to support business decisions.
* Establishing a scoring framework and its associated components to enable standardised attribute confidence scores.

Gartner also indicates many commercial RPs, especially in the financial services industry, calculate scores to determine whether to trust an authentication and/or authorisation. The RP’s weighing and scoring algorithm for attribute data is directly relevant to its risk mitigation strategy.

Document history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Key changes made |
| 0.0.1 | June 2016 | QGCIO | First draft |
| 0.1.0 | November 2016 | QGCIO | Draft for informal comment |
| 1.0.0 | May 2017 | QGCIO | No changes following informal or formal consultation. Administratively approved by Director, Strategy, Policy and Governance, QGCIO |

1. https://www.idmanagement.gov/IDM/servlet/fileField?entityId=ka0t0000000TNNBAA4&field=File\_\_Body\_\_s [↑](#endnote-ref-1)
2. <http://www.gartner.com/document/3029229?ref=solrAll&refval=174824899&qid=9386e94df270857d1adbf67c88f1be70> [↑](#endnote-ref-2)
3. <http://www.gartner.com/document/3029229?ref=solrAll&refval=174824899&qid=9386e94df270857d1adbf67c88f1be70> [↑](#endnote-ref-3)
4. http://www.gartner.com/document/3029229?ref=solrAll&refval=174824899&qid=9386e94df270857d1adbf67c88f1be70 [↑](#endnote-ref-4)
5. http://www.gartner.com/document/3029229?ref=solrAll&refval=174824899&qid=9386e94df270857d1adbf67c88f1be70 [↑](#endnote-ref-5)
6. https://spaces.internet2.edu/display/fedapp/Provisioning [↑](#endnote-ref-6)
7. http://www.gartner.com/document/3029229?ref=solrResearch&refval=174825228&qid=712c0e85f8dabe3a1e07a5d27ba78787 [↑](#endnote-ref-7)
8. http://www.gartner.com/document/3029229?ref=solrResearch&refval=174825228&qid=712c0e85f8dabe3a1e07a5d27ba78787 [↑](#endnote-ref-8)
9. https://spaces.internet2.edu/display/fedapp/IdP+Discovery [↑](#endnote-ref-9)
10. https://spaces.internet2.edu/display/fedapp/IdP+Discovery [↑](#endnote-ref-10)
11. https://spaces.internet2.edu/display/fedapp/IdP+Discovery [↑](#endnote-ref-11)
12. http://www.gartner.com/document/3029229?ref=solrAll&refval=174826508&qid=1ee43e867562ad73d7a9424cdeb72d35 [↑](#endnote-ref-12)
13. http://www.gartner.com/document/3029229?ref=solrAll&refval=174826280&qid=721156738a0b0e756260e30cf17c3495 [↑](#endnote-ref-13)
14. http://www.gartner.com/document/3029229?ref=solrAll&refval=174826280&qid=721156738a0b0e756260e30cf17c3495 [↑](#endnote-ref-14)
15. http://www.gartner.com/document/3029229?ref=solrAll&refval=174826280&qid=721156738a0b0e756260e30cf17c3495 [↑](#endnote-ref-15)
16. https://openid.net/specs/openid-connect-basic-1\_0.html [↑](#endnote-ref-16)
17. UK National Audit Office (NAO) Identity Assurance Programme briefing paper - https://www.nao.org.uk/report/identity-assurance-programme/ [↑](#endnote-ref-17)
18. https://blog.aniljohn.com/2013/06/how-to-choose-attributes-to-uniquely-identify-a-person.html [↑](#endnote-ref-18)
19. http://www.gartner.com/document/3103918?ref=solrAll&refval=174827371&qid=4228346b7cbc13ea537ce2e6c924a7af [↑](#endnote-ref-19)
20. <http://security.stackexchange.com/questions/655/what-is-the-exact-difference-between-samlp-and-ws-trust> [↑](#endnote-ref-20)
21. https://kantarainitiative.org/confluence/display/uma/Case+Study%3A+Users+Managing+Delegated+Access+to+Online+Government+Services [↑](#endnote-ref-21)