Queensland Government Enterprise Architecture

Federated identity blueprint

Process models

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 – Process models* 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 – Process models*

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 Process and function models 4](#_Toc484431685)

[1.1 Federated authentication 4](#_Toc484431686)

[1.2 Federated delegated authorisation 6](#_Toc484431687)

[1.3 Federated attribute exchange 12](#_Toc484431688)

# Process and function models

## Federated authentication

The flows for federated authentication are broken into two parts:

* relying party to identity provider (IdP consolidation) – where multiple relying party (RP) or service providers (SP) use a common identity provider/shared user base
* identity provider to identity provider (IdP federation) – where identity providers are federated to allow users registered (known) by one identity provider to access relying party services within the other identity provider’s community.

### IdP consolidation

Federated authentication takes place across multiple systems and organisations on the basis of mutual trust – the user authenticates himself to this primary domain so that this primary domain authenticates the user to other domains in the federation. This way, the user only registers once (or may be already registered) with an identity provider and authenticates only once per session – providing so called single sign-on (SSO).

To achieve this, two requirements must be met – an IdP capable of issuing identity assertions, an RP capable of accepting assertions (support for claims based authentication) and an agreement of trust between all parties including users as to the rules/procedures. Namely, the:

* RP trusts the IdP to authenticate users appropriately and release attributes as per the agreement
* IdP trusts the SP to not abuse the release attributes and to issue them only for the stated purchases as per the agreement
* user trusts the IdP.

The diagram on page 5 outlines a typical scenario of IdP consolidation, where multiple relying parties ‘outsource’ authentication to a common IdP which also provides a means to leverage the IdP existing user base. The diagram below depicts the primary federation roles of an identity provider, the relying party, user, intermediary, attribute provider, the relationships and typical domain boundaries.



The identity provider comprises of multiple roles, which in themselves may be federated (provided by other entities).

As depicted below, the IdP or RP may also use one or more attribute providers (or attribute brokers) as external authoritative sources for identity attribute information or verification of identity attributes.

Integration between an IdP and RP may to support:

* authentication via an identity assertion
* attribute exchange to support resolution of additional attributes for authorisation
* provisioning of user accounts at the RP.

For more information, see the decision framework [federated authentication – relying party to identity provider](#_Federated_Authentication_–).

In order for the relying party to accurately and reliably grant the subject access to protected resources, the relying party must be able to associate the identity and/or credential information of the subject provided by an identity provider to entitlement information managed by the relying party.

For more information regarding the attribute provider (ap) role see [federated attribute exchange models](#_Federated_Attribute_Exchange).

### IdP federation

Identity federation is a business model in which a group of two or more trusted (business) parties (legally) bind themselves with a business and technical contract to provide services to users[[1]](#endnote-1).

Within the identity ecosystem system, there will be multiple identity providers covering the same or similar user constituencies. For example, different agencies within Queensland Government servicing the same individual customer or the same individual obtaining services from providers at a Federal, State and local levels.

The federation may be mutual (bi-directional) and each party may consider the other as either a credential provider or identity provider.

Identity providers wishing to federate (partner) need to determine if the:

* federation is mutual (bi-directional) - where each IdP accepts the other’s credentials and/or identities
* identity provider is relying on the other entity as a:

credential provider

identity provider

both a credential provider and identity provider

attribute provider e.g. address information, licenses/certification status.

The decision framework for federated authentication (IdP to IdP) outlines a number of use cases and requirements for consideration (to be negotiated) between two identity providers.

The left of the diagram below depicts federation between two identity providers as both identity and credential providers. The diagram on the right depicts an identity provider acting as a broker other providers of credential, identity or attributes to compose a digital identity. This is referred to as a federations service role – see [if a federation services role required](#_Determine_the_Federation) for more information.



## Federated delegated authorisation

There are three main models for establishing delegated authorisation:

* single resource delegation for person-to-self sharing (user controlled consent) – where an individual owner of a resource delegates access 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 multiple resources uses a central authority to manage delegations for one or more parties (individuals, businesses or agents)
* central delegated authorisation (enterprise controlled) – where resources in multiple domains delegate access control decisions to an authority that evaluates access against a single policy.

Different industry protocols have evolved to support each use case:

* OAuth standard provides a simple method for a user to interactively delegate access to third party applications
* UMA standard is centred around an individual managing predefined delegations for multiple resources and extends the concept of delegation to include authentication of the requesting party
* 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.

User controlled consent can be defined as:

* context – the right moment to make the decision to share
* control – the ability to share just the right amount
* choice – the true ability to say no and change one’s mind
* respect – regards for one’s wishes and preferences.

#### Single resource delegation for person-to-self sharing – user controlled consent (OAuth Protocol)

The diagram below depicts the four primary roles of resource owner, resource server, authorisation server and application client, their relationships and typical domain boundaries. The authorisation server will authenticate the resource owner, or may federate with an IdP for authentication.

It is important to note the resource server is paired with an authorisation server that controls access within the same domain/organisation.



The table below outlines some examples of single resource delegation and the respective actors of each role:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resource owner** | **Resource domain** | | **Third party consuming domain** | |
| **Organisation** | **Resource** | **Organisation** | **Application client** |
| **Use case**: John downloads the ‘SmartCal’ mobile app wishes to grant the app access to his Google Calendar | | | | |
| John Phillips | Google | John’s personal calendar | Calendars R Us | SmartCal mobile app |
| **Use case**: A citizen via an online website chat can delegate access to a customer service agent to perform a specific function for the next 30 minutes on their behalf against a backend agency franchise system | | | | |
| John Phillips | Queensland Government agency | Heavy Vehicle Registration | RegoChecker | App |

#### Process flow – obtain authorisation process

The following describes the process for a resource owner to provide delegated access to a resource they own when using a third party application:

* the resource owner using the third party application client attempts to access a resource they own hosted at the resource server
* the resource owner is prompted to authenticate to the resource server’s authorisation server (responsible for protecting the resource)
* once authenticated, the resource owner must consent to the permissions requested from the third party application client
* if approved, the authorisation server provides the application client with an access token
* the third party application client presents the access token when making requests to the protected resource.

#### Central delegation across resources for party-to-party (person-to-person and person-to-organisation) sharing – user controlled consent (UMA Protocol)

The diagram on page 9 depicts the five primary roles of resource owner, resource server, authorisation server, application client and requesting party, their relationships and domain boundaries. The roles are intended to work in an interoperable fashion where each is operated by an independent party (for example, different organisations).

The notable differences to the single resource delegation model are:

* authorisation is managed centrally for multiple resource domains
* the central authorisation service may be provided by another entity chosen by the individual
* delegations can be managed/established at any time and ahead of time
* the requesting party (an individual or business) is known and can be authenticated (if required)

The authorisation server will authenticate the resource owner, or may federate with an IdP for authentication. In this case, the OAuth-based OpenID connect protocol can be used as a means of collecting identity claims from a requesting party in order to attempt to satisfy the authorising user's access policy[[2]](#endnote-2).



The table below outlines some examples of central resource delegation and the respective actors of each role:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Resource owner** | **Resource domain/s** | | **Central authorisation manager service** | **Third party consuming domain** | | |
| **Organisation** | **Resource** | **Organisation** | **Application client** | **Requesting party** |
| **Use case**: A patient provides consent/authorisation for clinicians or healthcare providers to access their health record | | | | | | |
| Patient | Queensland Health | Health record (e.g. discharge summary) | MyGov | Healthcare provider | Medical Record system | Healthcare provider or individual clinician |
| Patient | Healthcare provider | Health record (e.g. discharge summary) | MyGov | Queensland Health | Integrated electronic medical records (IEMR) | Queensland Health or individual clinician |
| **Use case**: A job seeker shares their Record of Achievement with a potential employer | | | | | | |
| Student | Education  Queensland | Record of achievement | Queensland ‘My Account’ service | Potential employer | OnlineRecruit SaaS | HR Manager |

#### Process flows

The UMA process is broken into two parts – the initial setup of a user selecting a central authorisation manager and linking resources they own to be able to establish pre-defined access rules, and the subsequent process flows for requesting parties to obtain authorisation.

#### Protect a resource

* The resource owner of resources at resource server, introduces it to the authorisation server so that later it can protect access to the resource.
* The resource owner configures the authorisation server with access policies associated with the resource.

#### Obtain authorisation

* An application client attempts to access to a protected resource on behalf of its requesting party
* The requesting party may need to be authenticated (undergo a process of trust elevation, for example, supply identity claims)
* If approved, the authorisation server provides the application client with an access token
* The third party application client presents the access token when making requests to the protected resource.

The New Zealand office of the Government Chief Information Officer ran a Proof of Concept (POC) project to test the concept of a common delegation’s capability for government using the UMA protocol.

The main scenarios tested included:

* immigration: someone overseas wants help to immigrate to New Zealand
* health: a patient asks their family member to help them do things on their patient portal
* education (careers): a job seeker shares their record of achievement with a potential employer
* education (early child): in the future parents can delegate after school pick up for their children.

The PoC found that UMA is a valuable tool when attempting to allow a member of the public to manage ‘delegations’ or access control, however there were many caveats[[3]](#endnote-3):

* UMA does not manage relationships in general. A well-structured approach, above and beyond UMA will be required to manage complex or abstract delegations and relationships.
* Agencies consuming a common delegation capability service are almost certain to need access control systems in addition to what is provided by a common delegation service.
* Parties wishing to create a suite of delegations across several government agencies in a single act would need a bespoke solution to do so. The standard UMA model would only facilitate a user setting up delegations independently at each agency.
* Prior to implementing a common delegation capability, the numerous policy questions must be resolved.

#### Central delegated authorisation across resources – enterprise controlled (XACML Protocol)

A centralised authorisation model allows an enterprise authority (or delegated policy authorities) to define access control policies that are applied consistently across federated multiple resources (applications, data and content) in different security domains. These resources delegate authorisation decisions and logic to a central authority and policy (although this may be distributed from a deployment perspective).

Access policies are written in business terms based upon attributes which allow for precise granular and real-time access control based upon a number of discrete inputs based on user information using external attributes (subject attributes, resource attributes, and other contextual information). For examples of each type of attribute, refer [Information models](#_Information_and_entity).

Attributes can reside in multiple sources where the level of confidence an RP has in an attribute may vary. The critical components of an ABAC solution are the governing organisational policies, attribute syntax and semantics and authoritative sources. The policies define the business objectives and the authoritative sources provide critical attribute verification, but syntactic and semantic agreement between the information exchange endpoints is the linchpin of attribute sharing[[4]](#endnote-4).

In December 2011, the US Government issued a directive recommending ABAC as the preferred access control model for enabling information sharing between diverse and disparate organisations[[5]](#endnote-5).

This same model for federated authorisation may be deployed in an enterprise context to better manage complex security policies for multiple subjects (employees, contractors, partners, and customers). With ABAC, when a new subject joins the organisation, existing rules and objects do not need to be modified to grant access as long as the subject is assigned the attributes necessary for the access required. This is one of the primary benefits employing ABAC when accommodating the external federated users.

The diagram on page 12 depicts the five common roles of an ABAC architecture:

* enforce – policy enforcement point (PEP)
* decide – policy decision point (PDP)
* support – policy information point (PIP) and policy retrieval point (PRP)
* manage – policy administration point (PAP)

The roles may be fully collapsed or distributed. The grey dashed boundaries represent an example deployment only.



## Federated attribute exchange

The diagram below depicts a typical model for sharing and/or validation of identity attribute information between parties from authoritative sources typically residing in another domain. Specifically:

* sharing identity attribute information to provide access to attributes held by one entity that are needed by another
* validation of identity attribute information against authoritative sources – typically for identity proofing or authorisation access control.



### Attribute providers

An attribute provider is responsible for the processes associated with establishing and maintaining identity attributes. Attribute maintenance includes validating, updating, and revoking attribute claims. An attribute provider asserts trusted, validated attribute claims in response to attribute requests[[6]](#endnote-6).

An IdP can be considered an attribute provider for the identities it manages.

### Attribute consumers

An IdP will commonly use an attribute provider to:

* validate self-asserted attribute claims from an authoritative source as part of identity proofing processes
* retrieve attributes for an authoritative source to support:

access control entitlement decisions (typically course-grained)

obtain other identifiers (e.g. a payroll number from the payroll system)

enrich the identity profile to fulfil RP attribute contract requirements

An RP will commonly use an attribute provider to retrieve or validate additional subject attributes to support:

* access control entitlement decisions (ABAC)
* service personalisation
* identity provisioning requirements.

### Attribute brokers

An attribute consumer may choose to leverage an attribute broker which can act a ‘hub of attributes’ by aggregating data from the various sources. By aggregating and correlating identity data, attributes can be found in one central location which improves consistency and provides a single source. An attribute broker is commonly required when one or more authoritative sources of attributes for a subject exist in multiple domains.

The attribute broker can reside:

* in the local trust domain
* outside the local trust domain (e.g. the IdP)
* in a mutual trust domain e.g. an intermediary.

An attribute broker is also capable of extending the schema for an identity with additional attributes for not included in any one authoritative source schema. The broker is designed to correlate identity attributes from the various authoritative data sources to provide a single authoritative source. The broker may also maintain different view for different consumers based upon authorisation to protect sensitive data (attributes).

The attribute broker must handle a number of complex scenarios regarding the following four situation. This material has been sourced from the US Government Federal Identity, Credential and Access Management (FICAM) Implementation Guidance Roadmap[[7]](#endnote-7).

**Defining identities**. The Authoritative Attribute manager uses built-in queries to aggregate identity attributes from different sources. In some cases, there may be multiple sources for digital identity data. When this occurs, the Attribute Manager should be capable of determining which source is the authoritative and be used to define the enterprise digital identity. For example, if there are two different sources the following scenarios could be present:

* Identity exists in both sources and consists of mutually exclusive attributes;
* Identity exists in both sources and consists of overlapping attributes; or
* Identity exists in one source but not the other.

**Correlation of identity attributes**. Given the number of potential sources for identity attributes, an agency should determine a mechanism for correlating those attributes into a single digital identity within the Attribute Manager. That is, each individual needs to be uniquely identified and attributes of the same identity from different sources should be correlated to the same enterprise digital identity. As discussed in Section 7.1.3, attributes can be correlated using a unique person identifier or a combination of attributes (i.e., a multi-attribute key). If a reliable correlation key does not exist, a mechanism must be developed for accurately correlating identity information, perhaps involving human review of potentially conflicting records.

**Normalisation**. Normalization creates a common and consistent taxonomy for attributes. It provides a mapping between different attribute types and values. For example, one bureau/component’s organisation attribute can be equivalent to another’s division attribute. Establishing a process for normalizing these attributes (e.g., data modelling) across an agency is critical to enabling effective attribute sharing.

**Reconciliation of identity attributes**. During the course of correlating attributes to an identity, there may be scenarios where different authoritative sources have a discrepancy regarding the same attribute of an identity, or where an attribute is missing completely. An agency may experience difficulty determining which attributes are correct if there are discrepancies with a single identity. In this case, a process must be put in place to correct the discrepancy and store the correct attribute in the Authoritative Attribute Manager. This will often require use of an offline process involving human review of the attributes. It must be determined how to fix the problem upstream. This can be achieved by either creating a data scanning utility to check for error conditions prior to populating the Authoritative Attribute Manager, or by building a process to notify the appropriate personnel to correct the source data manually.

### Single or multiple attribute providers

A single attribute provider may be used (most commonly an identity provider) as an authoritative source for pre-agreed attributes. Attributes for the same subject may also be sourced from multiple attribute providers consulted independently (e.g. credit bureaus)[[8]](#endnote-8). Multi-source attribute provider model is less common because of the added complexity, extended trust model required and the lack of a standards-based approach to multi-source attribute aggregation[[9]](#endnote-9).

There are two primary flows:

#### Attribute validation flow

Attribute validation is the validation of a self-asserted identity attributes. This could occur via a:

Presenting a self-asserted attribute bundle and returning a MATCH/NO-MATCH (optionally on a per attribute basis). For example, the Federal Government Document Verification Service (DVS).

Presenting an authorisation requirement with an asserted a set of attributes. e.g. ‘Is this person allowed to drive a heavy vehicle in Queensland?’

#### Attribute query flow

Attribute query is the retrieval of verified identity attributes based upon:

* Providing a common identifier (GUSID or LUSID) to be used as a look-up key for the subject. This requires both parties to select and standardise upon a USID (unique shared identifier) as the lookup key that is shared between the requester and the attribute provider or map between two different identifiers typically via an enrolment process. Establishing shared identifiers can be non-trivial to establish and may have privacy implications.
* Providing a bundle of attributes to lookup the subject – requires the attribute provider to have an identity resolution process. A common identifier may also assist, but may not be mandatory. Support for flows which encompass identity resolution as opposed to a simple key-based lookup is important for stand-alone attributes providers being able to service a broad customer base across multiple federated domains without the need to standardise common identifiers, which can have cost and privacy impacts.

An attribute query can be for an individual subject’s attributes or a bulk query for attributes belonging to multiple subjects i.e. a synchronisation.

Attribute query can be based upon either a:

* pull model attribute data queried in real-time by the IdP or RP
* push model where the attribute provider proactively pushes (synchronises) attributes with an IdP or RP (typically in bulk).

Attribute query maybe be used as part of a federated authentication process for ‘back-channel attribute retrieval’ through a direct connection to an attribute source, compared to a front-channel attribute delivery where the claimant is directly involved in the process (typically as part of the authentication event).

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.researchgate.net/publication/235178793\_Identity\_federations\_A\_new\_perspective\_for\_Bangladesh [↑](#endnote-ref-1)
2. https://en.wikipedia.org/wiki/User-Managed\_Access [↑](#endnote-ref-2)
3. <http://kantarainitiative.org/confluence/download/attachments/76907066/NZ%20ProjectClosureReport_PoC%20Delegations-Final%20Version%20SIG%20redacted%20Rev1.pdf?api=v2> [↑](#endnote-ref-3)
4. http://ieeexplore.ieee.org/document/5655096/?reload=true [↑](#endnote-ref-4)
5. http://csrc.nist.gov/projects/abac/ [↑](#endnote-ref-5)
6. https://www.whitehouse.gov/sites/default/files/rss\_viewer/NSTICstrategy\_041511.pdf [↑](#endnote-ref-6)
7. https://www.idmanagement.gov/IDM/servlet/fileField?entityId=ka0t0000000TNNBAA4&field=File\_\_Body\_\_s [↑](#endnote-ref-7)
8. http://csrc.nist.gov/publications/drafts/nistir-7817/Draft-NISTIR-7817.pdf [↑](#endnote-ref-8)
9. http://csrc.nist.gov/publications/drafts/nistir-7817/Draft-NISTIR-7817.pdf [↑](#endnote-ref-9)