Queensland Government Enterprise Architecture

Federated identity blueprint

Pattern library

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 – Pattern library* was developed and updated by Queensland Government Chief Information Office.

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

Copyright

*Federated identity blueprint – Pattern library*

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 Introduction 4](#_Toc484094374)

[1.1 Federated authentication - application resources (single IdP) 5](#_Toc484094375)

[1.2 Federated authentication - application resources (multiple IdPs) 6](#_Toc484094376)

[1.3 Federated Authentication/Authorisation - API resources (single IdP) 9](#_Toc484094377)

[1.4 Federated Authentication/Authorisation - API resources (multiple IdPs) 14](#_Toc484094378)

[1.5 Identity provider integration methods (e.g. Partner IdP’s) 16](#_Toc484094379)

[1.6 Identity Provider integration with authoritative sources of truth 17](#_Toc484094380)

[1.7 Linking Customer and Client Identities 19](#_Toc484094381)

[1.8 Federated authentication – identity provider to identity provider 21](#_Toc484094382)

# Introduction

This section contains are a number of common architectural patterns to achieve:

* federated authentication to application resources
* federated authentication API resources (using single or multiple IdP’s)
* identity provider integration methods (e.g. partner IdP's)
* identity provider integration with authoritative sources
* linking of client and customer identities
* federated authentication between two Identity Providers.

These patterns have been visually depicted below:



## Federated authentication - application resources (single IdP)

The following patterns apply for Federated Authentication to application resources across domains when using a single Identity Provider.

|  |  |  |
| --- | --- | --- |
| Pattern 1 | Pattern 2 | Pattern 3 |
|  |  |  |
| *Resources is claims-aware* | *Resource is not claims-aware, bridge standards-based token to the local domain security model* | *Resource may or may not claims-aware, additional functionality is required* |
| Termination of the connection occurs directly at the web application resource | * Termination of the connection occurs via a Relying Parties SAML or ODIC federation provider which acts as a bridge * Protocol translation or bridging is required when either the IdP or RP does not support the desired authentication protocol. Translation in some cases can be solely achieved by the RP, or require co-operation with the IdP as well or an intermediately STS. See Decision Framework for treatment approaches for non-claims aware resources | An RP SAML or ODIC federation provider may be used to provide additional functionality such as:   1. consolidating federation for multiple apps in the same resource domain 2. offloading the connection (token validation, key management) 3. providing trust elevation (setup authentication or risk-based controls) 4. augmenting an identity assertion with additional local attributes 5. provisioning of user accounts 6. additional authorisation logic, including acting as an ABAC PEP 7. acting as a Federation Service |

For more information, refer to the Decision Framework - Determine connection termination endpoint.

## Federated authentication - application resources (multiple IdPs)

The following patterns apply to Federated Authentication when relying upon multiple identities/Identity Providers. Most customer facing applications also require authentication of QLD Government staff. For example, authenticated agency staff uploading documents to a shared portal which customers also access to download or digitally sign the documents.

Consuming multiple identity sources can be problematic as applications typically only support one issuer (even those which are claims-aware). A Federation Service to broker is often required to abstract the multiple issuers (IdP’s or multiple CP’s).

A Federation Service is a logical role that can be fulfilled by the RP, or an IdP providing or a standalone intermediately (e.g. VANGuard, the Australian Access Federation (AAF)). For more information, see the Decision Framework - Determine the Federation Service.

The table below outlines for each combination of user constituencies, the possible Federation Service options:

### Agency Staff + Customers

|  |  |  |
| --- | --- | --- |
| Pattern 1 (recommended) | Pattern 2 | Pattern 3 (discouraged) |
|  |  |  |
| *Agency uses QGov Customer IdP, which brokers Agency Staff*  *IdPs via QGSFAS* | *Agency brokers Staff IdP + QGov Customer IdP* | *Agency Brokers QGov Customer IdP + the QGSFAS* |
| * Leverages integration established between QGov and QGSFAS to support this common use case. * QGov uses QGSFAS to avoid the need to federate directly with each Agency Staff IdP. | * May supports complex integration requirements for staff access e.g. attributes, authorisation information, provisioning requirements unable to be accommodated using a brokered connection to QGSAF via QGov. * The FS role may be provided by the Agency Staff IdP | * RP connections to QGSFAS are only permitted for shared Cross-Agency applications. Unnecessary connections add complexity and impact licensing. * The integration between QGov and QFSFAS should provide all required functionality. |

### Agency Staff + Partners

|  |  |  |
| --- | --- | --- |
| Pattern 1 | Pattern 2 | Pattern 3 |
|  |  |  |
| *Rely upon Partner(s) as an Identity Provider (and Credential Provider). Agency Federates directly.* | *Rely upon Partner(s) as an Identity Provider (and Credential Provider). QGov used to federate.* | *Rely upon Partner(s) as a Credential Provider, utilising QGov to verify Identity.* |
| * Supports multiple partners * Recommend where partners are fully trusted for identity verification | * May be more desirable if a common partner interacts with multiple QLD Govt agencies to mitigate duplicating federations per agency. * Industry Hubs may be best placed to perform this role. * There is currently no identified business driver or validated business case to provide a consolidated federation point. | * The QGov service can provide individual identity verification against DVS if required and still allow re-use of a partner’s credentials. * Recommended when identity verification is required (if the partner’s identity lifecycle management processes are not trusted |

Note: The scope of QGSFAS is for internal public sector staff federation only.

### Agency Staff + other Agencies Staff

|  |  |  |
| --- | --- | --- |
| Pattern 1 | Pattern 2 | Pattern 3  (recommended) |
|  |  |  |
| *Agency brokers local Staff IdP + other Agency Staff IdP* | *Agency brokers local Staff IdP, and use QSFAS to access other Staff IdPs* | *Agency uses QGSFAS to access local Staff IdP and other agency Staff IdPs* |
| * The FS role may be provided by the Agency Staff IdP * Suitable point-to-point model, but does not scale with large numbers of agencies. * Accommodates complex integration e.g. attributes, authorisation information, provisioning requirements | * Leverages QGSFAS to access other agencies Staff IdPs * Future proofs architecture for additional agency access | * Removes the need for an agency broker. * Future proofs architecture for MoG changes due to QGSFAS abstraction * Future proofs architecture for additional agency access |

### Agency Staff + Partner + Customer

|  |  |
| --- | --- |
| Pattern 1 | Pattern 2  (recommended) |
|  |  |
| *Agency federates directly*  *with each IdP* | *Agency federates with QGov which brokers Partner and Staff IdPs* |
| * Agency federation infrastructure required * Provides the most flexibility | * No agency federation infrastructure required * leverages WofG consolidated federation service pathways for common use cases |

## Federated Authentication/Authorisation - API resources (single IdP)

This section outlines patterns for federated authentication and authorisation to Relying Party API resources based upon the use of RESTful APIs which use OAuth as a resource delegation protocol to allow access to be authorised by the resource owner (e.g. the end user).

Using the OAuth protocol in a federated manner means the Identity Provider performs the role of ‘Authorisation Server’ (AS) to authenticate the user and prompts them to authorise access, whilst the relying party performs the role of the ‘Resource Server’ (RS) which protects the resource and grants access based upon the token provided by its trusted Authorisation Server. The OpenID Connect protocol (based upon OAuth) can be used interchangeably where Identity attributes are required by the API resource in addition to the list of permissions granted. ODIC utilises the JSON Web Token (JWT) specification as the token format to accommodate a rich attribute format.

It is important to note that in a federated model, the RP (and the resource owner) must explicitly trust the IdP to perform the Authorisation Server role which authenticates the user and obtains consent (performs authorisation). Whilst this architecture is similar to federated authentication models, in a typical OAuth implementation there is a reasonably tight binding between the Authorisation Server and Resource server which would typically both be operated by the same entity. However, the risk of an IdP inappropriately generating and signing a token to grant inappropriate access an RP resource remains the same in the federated model as the RP must trust the IdP for authentication and authorisation (even if the RP operates a local Authorisation Server and Resource Server as per models 4 and 5 below). Emerging standards such as UMA are better designed to operate in a federated model as they also allow a resource owner to select an entity they trust to manage authorisation.

For more information regarding API authentication refer to the Decision Framework specifically:

* Determine the Authorisation Server and Resource Server roles
* Determine Presentation of Tokens (for REST API’s)
* Determine Token Validation Model
* Determine the Session Management Strategy (token revocation).

### Single application client accessing a single API

There are five common architectural patterns to authenticate user (via the client application) to access a single API. These span from a fully collapsed model in which the IdP provides both the required Authorisation Server (AS) and Resource Server (RS) functionality to a fully distributed model.

For simplicity, the flow diagrams do not depict the first step where the user authenticates via the client application to the Authorisation Server. Only the corresponding flows showing the access token minted from the AS are depicted.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pattern 1 | Pattern 2 | Pattern 3 | Pattern 4 | Pattern 5 |
|  |  |  |  |  |
| *Use the IdP as an AS & RS* | *Use the IdP as an AS & the Resource itself as the RS* | *Use the IdP as an AS, RP operates own RS* | *RP operates own AS (as an IdP) & RP provides the STS* | *RP operates own AS (as an IdP) & provides the STS* |
| * Direct trust, fully hosted model * Native claims-aware API * The RP is able to offload token validation | * Native claims-aware API * API performs token validation | * Similar to pattern 2, with additional RP infrastructure to offload token validation * The RP may be used to support multiple API’s | Either:   * The IdP sends a bearer request for a token to the RP (used when the IdP doesn’t support the RP OAuth protocol) or; * The RP accepts the IdP token and issues a local RP token | The IdP AS is trusted to issue a token on be-half of the RP AS. |

Notes:

* An ESB or API Gateway deployed by the Relying Party can often provide the OAuth Resource Server functionality in pattern 3, and the STS functionality in Pattern 4. An ESB could also provide the RP OAuth Authorisation Server functionality in patterns 4 & 5.
* The decision framework addresses alternatives for legacy SOAP based web services and protocols translation.

The primary objective should be to enable API connectivity using Patterns 1-3 which are the simplest means provided:

* both the IdP and RP natively supports either OAuth or OIDC standards. Use of these standards is strongly advisable for all QLD Government IdPs and 1st Party developed API’s to maximize interoperability
* the RP does not operate as an IdP (authenticates other subjects for the same API resource)

Patterns 4 and 5 require either the IdP or RP to translate the token used in the IdP domain to the RP domain’s format. This is required when:

* The RP also operates as an IdP and authenticates subjects for the same API resource - when:

utilsing 3rd party cloud services e.g. Salesforce has a local user accounts, but also supports federated SSO for enterprise customers

agencies operating their own Customer Identity Provider

* The RP cannot directly consume the IdP’s token – this will occur when:

The token format does not match for example when:

* + - the IdP uses WS-Trust (SAML), and the RP uses OAuth or OIDC
    - the RP uses a propriety token format (typical for 3rd party cloud services)

The RP API resource(s) cannot or does not wish to directly consume or trust the IdP issued token. The RP then must construct a new token for the local resource domain in order to map permission models, or augment with additional claims information from local data sources or obtain additional user consent is required.

Pattern 5 where the IdP generates a token suitable to the RP will only be possible if the RP uses industry standards, or if the propriety format is well supported.

### Single application client accessing multiple APIs

This scenario deals with the user (via the client application) also needing access a second API. This is a common scenario for example:

* WofG portals aggregating information from multiple agency systems
* Single agency applications aggregating information from multiple backend systems, APIs or entities.
* Simple mobile applications which often require access to multiple API services for data access, push notifications, real-time messaging, paywalls, in-app chat and analytics services (which need to communicate user context information).

For security reasons, an IdP treats each RP connection/resource differently and issues a separate signed and scoped token so that a compromise to one resource or it’s token may not easily escalate access to other resources. The line colouring on the diagrams is used to depict these differently scoped and signed tokens.

|  |  |  |
| --- | --- | --- |
| Pattern 1 | Pattern 2  (recommended) | Pattern 3 |
|  |  |  |
| *Both RP API’s share a single IdP connection / token.* | *The client web application obtains a token to access an API* | *The second API resource is protected by its own AS (the RP is also an IdP)* |
| * A single API can accept (trust) a shared token also used by other API’s. * This model may be suitable where the resources have a close affinity, are within the same security domain (common with stateless micro service architectures), or are provided by the same RP organization. * Does not provide finite control over token revocation (session management) or permission and consent models. In the event of compromise multiple resources may be effected. | * The client application (on behalf of the user) can request a new token from the IdP to access the other RP API. To avoid the user needing to re-authenticate, the IdP should support leveraging an existing authentication (this may be a session established with the IdP from a federated authentication web SSO flow or involve exchanging a previous token from another API which the target API trusts). | Patterns 4 or 5 (single API access patterns) is required. The pattern selected depends upon which party provides the token translation. Pattern 4 is shown as an example. |

Notes:

* In regards to pattern 3, if the IdP does not support OAuth or provide an STS functionality, and needs access to a RP API whilst leveraging an existing Web SSO session, a SAML 2.0 Bearer Assertion Profile or JSON Web Token Bearer Profile (when using ODIC for Web SSO) if supported by the IdP and RP can be used as an authorisation grant to obtain OAuth token from the RP. This is a common mechanism used for token translation when enterprise Staff IdP’s only support SAML and need to obtain access a 3rd party cloud API’s using OAuth.
* Whilst these patterns utilize OAuth as the recommend protocol for token-based authentication for API’s (as per the benefits listed in the standards for API section), patterns 1 and 2 above can be achieved using simple cookie-based authentication if both the client web application and the API are served from the same web domain or subdomain. Both cookie-based and token-based authentication can be mixed as required where there are multiple API’s and security domains e.g. cookies for API within the agency domain and tokens for remote domains.
* Should one of the API’s subsequently need to call another API in another domain in context of the same user (referred to as API chaining) patterns 4 or 5 which utilise an STS may be used to bridge trust and perform token translation across the chained domains.

## Federated Authentication/Authorisation - API resources (multiple IdPs)

The following patterns support federated authentication and authorisation to API resources by multiple user constituencies identified by different Identity Providers.

### A common API accessed by different users associated with different IdPs

The API is provided by a common RP which needs to authenticate multiple user constituencies against their associated IdP. Patterns 1 and 2 assume there is no direct trust between the separate IdP’s as trust is bridged by the RP as the provider of the API, although one option shown in pattern 3 is for an IdP to act as a primary IdP and STS.

|  |  |  |
| --- | --- | --- |
| Pattern 1 | Pattern 2 | Pattern 3 |
|  |  |  |
| *The RP RS accepts tokens from multiple IdP’s/AS’s* | *The RP operates an STS that trusts the other IdP AS’s and translates/bridges each domain* | *A primary IdP federates with the other IdP(s) for authentication only* |
| * The RP RS infrastructure needs to support validating tokens from multiple issuers for a single resource. This can generally be easily achieved in cases where the API itself acts as the RS and validates the token in arbitrary code using a 3rd party library and conditional logic. * Some ESB’s rely upon a remote token validation by the AS using a specific URL per issuer similar to PKI revocation checking using OCSP. | * Either IdP 1 or IdP 2 could perform an STS role, however typically once the RP is required to consume (trust) more than one token, an STS will be required. | * The primary IdP must act as a federation service to terminate the 2nd IdP connection and be capable of leveraging the existing SSO session to mint an API token. * User consent from the chained IdP is unable to be obtained in this model, or would require a propriety implementation. |

### A user associated with one IdP accessing an API associated with another IdP

These patterns support a user identified by IdP 1 using a Relying Party application (or an API) associated with IdP 1 needing access to a Relying Party API resource associated with IdP 2. The user should not need to re-register with IdP 2 (IdP 2 trusts IdP 1). The inverse use case of an IdP 2 user accessing an IdP 1 resource can be supported through a trust arrangement.

|  |  |  |
| --- | --- | --- |
| Pattern 1 | Pattern 2 | Pattern 3 |
|  |  |  |
| *IdP 2 trusts a token from IdP 1 and issues a token compatible with the IdP 2 RP API* | *The RP API accepts a token from IdP 1 or a request for a token from IdP 1* | *IdP 1 is trusted by IdP 2 (or IdP 2’s RP) to generate a token on-behalf of IdP 2 to access the IdP 2 RP API* |
| * The token from IdP 1 may be the IdP 1 RP token or another scoped token for the purposes of delegation as shown on the diagram. * If IdP 2 translates the token, the infrastructure may be able to be leveraged for other RP 2 APIs | * This is the same as pattern 4 (single IdP, single API) as the 2nd IdP is out of play * Each RP would need to provide the infrastructure, which may not scale with multiple API’s from different RPs. * RP 2’s API can consume the existing IdP 2 token via the same STS infrastructure | * Simple point solution within the control of IdP 1 which doesn’t necessitate involvement from IdP 2 * RP 2’s API can easily consume multiple tokens (particularly if the same token presentation model is maintained) |

## Identity provider integration methods (e.g. Partner IdP’s)

The following patterns outline methods for a Relying Party to integrate (accept identities) from an Identity Provider such as a trusted partner. Options range from the simplest form of manual registration to fully automated federated models.

Whilst it is preferable to integrate using a fully federated model with Single Sign-on (as this improves efficiency, lower costs and improves accuracy through automation), in some cases depending upon the organisation it may not be technically possible, or based upon the nature of the relationship for the RP to carry out some aspects of the identity management lifecycle.

The models below apply to both credential management and/or identity management processes. Also see the Decision Framework for guidance as to the split of responsibilities for authorisation between an IdP and RP.

|  |  |  |  |
| --- | --- | --- | --- |
| Pattern 1 | Pattern 2 | Pattern 3 | Pattern 4 |
| *The Agency provisions a local identity manually based upon a service request from the Partner. Manual entry is the simplest form of user provisioning and account management.* | *The Partner provides a flat file for bulk entry which includes new accounts and updates to existing accounts. The file maybe generated by the Partner in an automated fashion using IDM tools.* | *The Agency provides a self-service portal and workflow for delegated administrators of the Partner to create, manage and disable user accounts and associated privileges. This will typically include responsibility for performing access reviews.*  *A cloud hosted IDaaS service could be utilised.* | *The Partner and Agency are both capable of supporting federated authentication. The federation may support either run-time or administrative identity provisioning\*.* |

\*Federated provisioning may include either the following models:

* **Run-time Model** - An Agency may choose to only provision an account when the user first attempts to access the application (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. Deprovisioning actions are manual or need to be reconciled out-of-band.
* **Administrative Model** – User accounts are pre-provisioned from the Partner to the Agency systems in an automated fashion prior a user accessing a resource. The provisioning process is not dependent upon the integration for federated authentication. Pre-provisioning may be required in cases where the agency 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 and/or without being logged into.

## Identity Provider integration with authoritative sources of truth

There are four example models for an identity provider, such as a Customer IdP to integrate with authoritative sources to:

* verify self-asserted attributes
* obtain verified attributes
* facilitate access control or entitlement decisions
* facilitate service personalization

Pattern 1 is the preferred digital-first means to obtain verified attributes whereby the claimant authenticates to the authoritative source using a previously issued credential and provides consent to release the attributes, along with an option to provide ongoing delegated access to maintain the information through a person-to-self sharing model.

|  |  |  |  |
| --- | --- | --- | --- |
| Pattern 1  Linked Credential | Pattern 2  Attribute verification | Pattern 3  Attribute Query | Pattern 4  Policy Decision |
| *The claimant authenticates to the authoritative source (a federated identity provider) using a previously issued credential and releases the requested attributes.*  *The ‘credential’ or process used to authenticate the subject may be either (or a combination of):*  *A traditional online account (e.g. user-name/password login)*  *A one-time password or invitation code sent by the client IDM to a previously verified mobile number or registered mobile app (push-to-swipe) or postal address or human messenger.*  *A knowledge-based authentication process [See Note 3]* | *A bundle of self-asserted attributes provided by the claimant is verified against the authoritative source (typically via data matching). [See Note 1]*  *For example, using DVS to match identity details listed on paper-based identity documents against the source issuer database.* | *Either:*   1. *Retrieving attributes for the subject using a shared identifier previously established between the IdP and authoritative source (linked through a previous enrolment process/authentication e.g. a private QGCIDM QID identifier or verification of a public identifier e.g. TMR CRN). [See Note 2]* 2. *Retrieving additional attributes of a subject by providing the subjects identity attributes which have been verified previously by a trusted entity (e.g. the IdP) as a means to authenticate the subject to their record at the authoritative source. [See Note 1]* | *Determining if the claimant has a specific entitlement for authorisation by providing a policy question to the authoritative source e.g. can subject X drive Y type of vehicle in QLD?*  *The subject will typically need to be identified/authenticated first via patterns 2 or 3.* |
| **Options to maintain currency/revocation:**   * Periodic re-query through the claimant re-undergoing EOI * Can be achieved through a resource delegation model for person-to-self sharing where an access token provides ongoing API access | **Options to maintain currency/revocation:**   * Periodic re-query through the claimant re-undergoing EOI * Periodic re-query using shared identifier (a transactional identifier might need to be established by the authoritative source in some cases) * Push events e.g. bulk load from authoritative source | **Options to maintain currency/revocation:**   * Periodic re-query through the claimant re-undergoing EOI * Periodic re-query using shared identifier * Push events e.g. bulk load from authoritative source | **Options to maintain currency/revocation:**   * Periodic re-query through the claimant re-undergoing EOI * Periodic re-query |

Notes:

1. This requires the Attribute Provider to have an identity resolution process to resolve a remote identity to a local identity record. See the patterns below regarding linking customer to client identities.   
     
   Is important for stand-alone attributes providers to support attribute query and validation flows which encompass identity resolution (as opposed to key-based lookup mechanisms) to be able to service a broad customer base without the need to standardize common identifiers across multiple federated domains, which can have cost and privacy impacts.
2. Whilst establishing shared identifiers can assist with unique resolution, they can be non-trivial to establish and may have privacy implications. However, once a link has been established, shared subject identifiers (private or public) are important to avoid re-resolution for subsequent interactions.
3. Agencies should refer to AS4860-2007: Knowledge-based identity authentication – Recognizing Known Customers.

## Linking Customer and Client Identities

The patterns below indicate how a customer identity record could be linked to an agency client identity record. This process may be initiated by either the Customer IdP or Client IDM to link either an existing identity or establish a new identity through relying upon either an existing customer or client identity. The examples below cover online and offline scenarios, including a hand-off between channels.

The patterns below outline examples and possible options to establish the initial link between the two identities. Once the link is established a mapping between either the customer identifier to the client identifier or client identifier to the customer identifier needs to be stored to avoid re-linking for subsequent interactions.

This mapping can be maintained or held by either the Customer IdP or Client IDM system. The system performing and/or maintaining the mapping must consider a process to handle or an ability to support a one to many mapping scheme given a customer may by choice have multiple identities that may be mapped to a single unique client record.

|  |  |  |  |
| --- | --- | --- | --- |
| Pattern 1  Customer IdP initiated (existing Client Identity) | Pattern 2  Customer IdP initiated (new Client Identity) | Pattern 3  Client IDM initiated  (existing Customer Identity) | Pattern 4  Client IDM initiated  (new Customer Identity) |
| *The individual has a customer identity and an existing client identity.* | *The individual has a customer identity, but does not have an established agency client identity* | *The individual has a client identity with an agency, and an existing customer identity* | *The individual has a client identity with the agency, but no established customer identity.* |
| Examples:   * An individual registered with QGov accesses an agency online service (e.g. a customer portal) to interact with their existing client record. * An individual wishes to update their address with multiple agencies, some agency records have not been linked previously (note: may only be able to rely on data matching only – interactive knowledge-based authentication may not be possible) | Examples:   * An individual applies for a new agency service and uses their existing QGov customer identity to identify themselves. | Examples:   * An individual undertakes an additional verification with the agency e.g. in-person verification/interview or meets a new eligibility service requirement (e.g. proof of license, proof of training) that can be added to their digital customer identity profile for reuse with other agencies. * An individual engages an agency for services and requests that they be able to interact with the agency online noting they already have a QGov Identity. * An individual needs to authenticate to the agency in-person or over the phone (their QGov account could be used to facilitate authentication via MFA SMS) | Examples:   * An individual registers an identity in-person at an agency counter for services and is offered/wishes to create a QGov digital identity to be able to interact with the agency online or other agencies through re-use of the existing verification. * An agency launches a new digital service which uses QGov to authenticate customers. The agency runs a marketing campaign and invite their client to register online for access. |
| **Linking process options:**   * Identity resolution - data matching against verified identity attributes, and other interactive knowledge-based authentication processes [See Note 1] * Linked credential options from pattern 1 above (if available and meets assurance requirements). May be relevant for migration between customer IdPs. * Using a shared public identifier that has been verified by the Customer IdP e.g. TMR CRN | **Linking process options:**   * Nothing specific provided the QGov identity meets the identity proofing requirements of the agency service. | **Linking process options:**   * A one-time password or invitation code sent by the client IDM a previously verified mobile number or registered mobile app (push-to-swipe) or postal address or human messenger. | **Linking process options:**   * A one-time password or invitation code sent by the client IDM a previously verified mobile number or registered mobile app (push-to-swipe) or postal address or human messenger. |
| **Identifier mapping:**  Best maintained by Client IDM undertakes the resolution process as have access to the existing client identity for knowledge-based authentication. No easy means to communicate the client identifier back to the customer IdP as web SSO authentication is one-way and privacy issues with storing all client identifiers online in centralised storage. | **Identifier mapping:**  Best maintained by Client IDM which will established a new identity and client identifier (following existing schemes as the customer IdP identifier is private). See Pattern 1. | **Identifier mapping:**  Generally, the entity which performs the authentication is best placed, however the responsibilities for authentication event may be split in some cases or solely managed by the Customer IdP if the client IDM is not capable (legacy) | **Identifier mapping:**  As per pattern 3. |

Note 1: Implementing an identity resolution process has a number of considerations which includes privacy, data minimization, data matching algorithms and addressing multiple matches when the minimal data set is not enough to uniquely resolve a person.   
  
Matching is not absolute and typically needs to be associated with a confidence level, for example bands segmenting a 'clear' winner (e.g. 98% match), no close second, multiple plausible (e.g. 98-90%) and no plausible candidate (e.g. > 50%). Supplemental attributes that can be used to prevent collisions in cases where core biographic attributes are not enough.

For more information, see the:

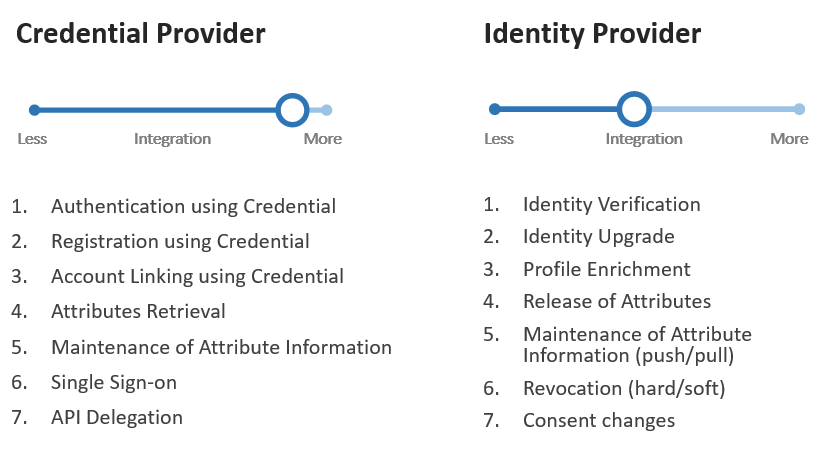
* The Federal Government Better Practice Guidelines for Data Matching 2009 developed under the National Identity Security Strategy (NISS).
* Decision Framework - determine the Identity Resolution process which discusses the effectiveness of matching one attribute over another based upon the American National Standards Institute (ANSI) / North American Security Products Organization (NASPO) Identity Proofing and Verification (IDPV) Standard Development Project’s minimum requirements for ‘reconciling an asserted identity to a single individual’.

## Federated authentication – identity provider to identity provider

This pattern outlines a model for establishing a federation trust between two identity providers which each have an established user base and community of relying parties.

The two Identity Providers who have chosen form a business partnership must determine if they will rely upon the other entity as a credential and/or identity provider and level of integration required. The more integrated, generally the better the end user experience.

The federation may be mutual (bi-directional) whereby each IdP accepts the other’s credentials and/or identities.



The trust framework model between two IdP will typically require the ‘dynamic trust model (attribute driven)’ option. For a detailed explanation, refer to section 10 - Trust Models and Frameworks.

When two identity providers decide to form a partnership and federate, if there is benefit in seeding user databases, either or both IdP’s may agree to run a marketing campaign to ‘invite’ their users to register with the other partner IdP to streamline subsequent access attempts. Should an IdP not have a digital credential, a one-time credential such as an invitation code may be used to establish the link.

The table below outlines a number of optional use cases and requirements for consideration (negotiation). They are written from the perspective of IdP 1 relying upon IdP 2 as either a Credential Provider or Identity Provider.

### IdP 1 relying upon IdP 2 as a credential provider

|  | Use case/requirements |
| --- | --- |
| 1 | **Choice of federated credential**  IdP 1 should offer users the choice of using a IdP 2 credential (in additional other credential choices already offered by IdP 1) |
| 1.1 | **User registration using federated credential**  A new user can register at IdP 1 using an IdP 2 credential (and on a subsequent occasion authenticate at IdP 1 using the credential) |
| 1.2 | **User authentication using federated credential**  A user registered at IdP 1 can authenticate using an IdP 2 credential. |
| 1.3 | **Account linking using the federated credential**  An existing user can:   * Link an IdP 2 credential to their IdP 1 account (and on a subsequent occasion authenticate an IdP 1 using the credential). * Unlink an IdP 2 credential at any time   IdP 1 must support the reconciliation of multiple identities into a single account. |
| 1.4 | **Release of attribute information**  As appropriate and consented by the user, IdP 2 should release to IdP 1 along with the credential where appropriate any associated:   * identifiers * credential attributes * identity attributes (see 2.2.1) which may be released time of authentication * attribute metadata   This may involve:   * A copy of the actual attribute data and metadata to populate the IdP 1 user profile * Only the metadata of the attributes and an access\_token as a means to obtain the actual data when required (referred to as distributed claims in industry standards). |
| 1.2 | **Maintenance of attribute information**  As appropriate, IdP 1 should have the ability to maintain the currency of any attribute data (or metadata) sourced from the IdP 2 (whilst the user’s original authorisation/consent remains valid e.g. only once/3 months/6 months/12 months, indefinitely). Maintenance of attribute information can assist with information or authorisation attribute changes and revocation. |
| 1.3 | **Identifiers**  IdP 1 and IdP 2 must agree on a shared identifier for the subject (either from either IdP1 or IdP2 or a new unique identifier) and establishing a mapping if required. If IdP 2 is only a credential provider, this may be a pseudonymous identifier or an arbitrary generated identifier for the credential. |
| 2 | **Resource authentication and authorisation across RP services connected to each IdP**  The following table outlines a model to support authentication and authorisation of web and API resources across RP services connected to each IdP. |
| 2.1 | **Web single sign-on**  IdP 1 should support IdP-initiated SSO to an IdP 1 RP application to allows IdP 2 to create a smart link which automatically logs an IdP 2 user into the RP application connected to IdP 1 using the agreed subject identifier.  The smart link avoids the IdP 2 user manually selecting their IdP/credential at IdP 1 (known has home realm discovery) which is required with SP-initiated SSO if the IdP 2 user accessed the IdP 1 RP application directly. |
| 2.2 | **API authentication and authorisation**  An IdP 1 RP application should have the ability to access an IdP 2 RP API (service functionality) in the case of:   * a new authentication or; * via leveraging an existing web SSO authentication   Based upon the requested and authorised permissions (scopes) for the API e.g. read\_infringements, read\_overdue\_payments, make\_payment, IdP 2 should provide a signed access token for the requested IdP 2 RP API either:   * as part of a web SSO assertion or; * provide a means for IdP 1 (or the IdP 1 RP application) to exchange the IdP 2 authentication/session token for an access\_token to access the IdP 2 RP API   IdP 2 should provide a refresh\_token (if requested by IdP 1 and approved by the IdP 2 user) to enable the IdP 1 RP Application to subsequently access the API on behalf of the user without needing the user to re-authenticate.  Note: All API access is to be delegated with consent by the individual resource owner (user). Non-interactive system-to-system authentication/integration which provides ‘carte blanche’ access to the other IdP to impersonate any user should be avoided. |
| 2.3 | **Revocation of access**  See 2.4. The permissions (scopes) will reflect functionality e.g. read\_infringements, read\_overdue\_payments, make\_payment. |
| 2.4 | **Changes to pre-authorised access**  See 2.5. The permissions (scopes) will reflect functionality e.g. read\_infringements, read\_overdue\_payments, make\_payment. |
| 2.5 | **Increased scope of access**  See 2.6. The permissions (scopes) will reflect functionality e.g. read\_infringements, read\_overdue\_payments, make\_payment. |

### IdP 1 relying upon IdP 2 as an identity provider

|  |  |
| --- | --- |
|  |  |
| 2.1 | **EOI processes should accept previous IdP 2 verifications**  As appropriate, IdP 1’s EOI process should accept (trust to a given level) previous identity verifications performed by IdP 2 (for a given attribute or collection of attributes). This applies to EOI processes used to support:   * new registrations (online or in-person) or; * trust elevation for existing registrations   This could be facilitated by either the user registered with IdP 2:   1. Authenticating to IdP 2 using their IdP 2 credential and releasing requested identity attributes (and an access token to support maintain currency of attribute information if required and consented).  Note: Where IdP 2 user’s do not have an associated digital credential and there is no common shared identifier enrolled, to avoid the user at IdP1 needing to re-prove their identity to IdP 2 through self-asserted attributes or knowledge based questions, IdP 2 may provide the user with a one-time token/credential through an existing registered channel to facilitate registration with IdP 1 and establish a shared identifier or link. 2. Retrieving attributes from IdP 2 using a common shared identifier:    * An established identifier (e.g. the QGCIDM QID) from IdP 1 which may have been previously linked to IdP 2’s record through an enrolment process online/offline or previous authentications    * An established identifier (e.g. a TMR CRN) from the IdP 2 which IdP 1 uses to identify the subject in IdP2 3. Specifying self-asserted attributes to be verified against IdP 2 as a source of truth (IdP1 may choose to use other sources in addition). Optionally, to support maintenance of the attribute information, upon a successful match an access token (or other generated subject identifier) may be returned to allow subsequent attribute queries.  Note: This process has a number of considerations including privacy, data minimisation and multiple matches when the minimal data set is not enough to uniquely resolve a person. |
| 2.2 | **Augmentation of user profile (read/pull from IdP 2)**  An IdP 1 user where appropriate should be able to augment their profile with additional identity information from IdP 2. This information may be used by IdP 1 for various purposes e.g. pre-fill forms, represent a previous verification of identify information or authorisation information e.g. role/entitlement.  This could occur though either the user registered with IdP 2:   1. Authenticating to IdP 2 using their IdP 2 credential and releasing their identity attributes (and an access token to support maintenance of attribute information) 2. Specifying self-asserted attributes to be verified against IdP 2 as a source of truth (IdP1 may choose to use other sources in addition). To support maintenance of attribute information, upon a successful match an access token (or generated identifier) may be returned to allow subsequent retrievals. |
| 2.2.1 | **Release of attribute information**  Same as 1.4 |
| 2.2.2 | **Maintenance of attribute information**  Same as 1.5 |
| 2.3 | **Augmentation of user profile (write/push from IdP 2)**  IdP 2 may wish to advise (notify) IdP 1 that an event of interest has taken place. IdP 1 may choose to take some action such as revoke or modify a previous verification or add new verification. E.g. A user at IdP 2 may have updated their details (change of name, email, address etc), re-verified some attributes (e.g. identity expiry) or undertaken additional verifications to be shared with IdP 1.  The notification (or web-hook) may contain:   * A trigger for IdP 1 to check for updates to existing attributes for a given subject (as per the standard 2.2.2 Maintenance of Attribute Information). This may include a new access\_token (or means to obtain a new access\_token) with additional permissions to fetch additional attributes. * Within the payload Create, Read, Update, Delete (CRUD) statements and information for the given subject to be processed by IdP 1.   The push from Id2 to IdP 1 may be authenticated via mutual system-to-system authentication or IdP 1 may delegate IdP 2 authorisation to the API. |
| 2.4 | **Revocation of access to identity profile information** |
| 2.4.1 | **Soft disconnection at IdP 1**  The IdP 2 user when logged into their IdP 1 profile may choose to ‘remove’ the profile information (attributes) retrieved from IdP 2.  When the user chooses to remove the last or all attributes sourced from IdP 2, IdP 1 removes the stored access\_token and refresh\_token. Optionally IdP 1 may notify IdP 2 to backlist the access\_token and refresh\_token to ensure no further use by any party. |
| 2.4.2 | **Forced revocation from IdP 2**  An IdP 2 user should be able to revoke IdP 1’s (or an IdP 1 RP application’s) access to an IdP 2 RP API service (in the event IdP 1 or the IdP 1 RP Application is not trusted)  An IdP 2 user should be able to revoke IdP 1’s access to their identity profile information.  There are 2 options:   1. Full revocation (basic) - the user at IdP 2 chooses to ‘revoke IdP 1 access to their profile’.  IdP 2 revokes (blacklists) the access\_token and refresh\_token IdP 1 uses to access the user profile. This prevents IdP 1 from querying IdP 2 to maintain the currency of existing attribute information (2.2.2). However, if IdP 1 has previously stored the attributes this will remain on file at IdP 1. 2. Full revocation (advanced) - the user at IdP 2 chooses to ‘revoke IdP 1 access to my profile and advise IdP 1 to remove my data from their systems’.   As above, with a notification to IdP 1 to remove the data as per the push mechanism outlined in 2.2  Note: This does not unlink the IdP 2 credential from the IdP 1 identity which is covered as per 1.1.3 and would typically be initiated by the IdP 2 user at IdP 1 (e.g. as part of an IdP 1 account closure). |
| 2.5 | **Changes to pre-authorised user consent to identity profile information** |
| 2.5.1 | **Soft change at IdP 1**  The IdP 2 user when logged into their IdP 1 profile may choose to ‘remove’ some or all profile information (attributes) retrieved from IdP 2. |
| 2.5.2 | **Forced change from IdP 2**  The user at IdP 2 may choose to modify the scope of IdP 1’s access to their IdP 2 user profile attributes based upon what was previously granted to IdP 1 e.g. the user at IdP 2 ‘unchecks’ IdP 1’s access to their address or photograph attributes, but leaves name and DOB ‘checked’.  In this case, IdP 2 revokes (blacklists) the access\_token and refresh\_token IdP 1 uses to access the user profile. IdP 2 generates a new access token with the amended scopes and notifies IdP 1 as per the push mechanism outlined in 2.2. This may include removal of any stored data for the revoked attributes. |
| 2.6 | **Increased scope of access to identity profile information**  This can occur through either (or both):   * re-authentication and linking by the IdP 2 user at IdP 1 (see 2.2 Augmentation of user profile) * a change pushed by the IdP 2 user from IdP 2 (see 2.5.2 Changes to pre-authorised user consent) |
| 2.7 | **Shared attributes and metadata**  IdP 1 and IdP 2 must agree which identity attributes are to be exchanged.  Where appropriate any attributes supplied by IdP 2 should include associated metadata wherever possible. IdP 1 may choose to accept, override or assign metadata based for its own purposes based upon its own determination. |
| 2.8 | **Identifiers**  IdP 1 and IdP 2 must agree on a shared identifier for the subject (either from either IdP1 or IdP2 or a new unique identifier) and establishing a mapping. |

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 |