Warwickshire
County Council

OIX OPEN IDENTITY EXCHANGE

OIX IDAP Alpha Project - Technical Findings

Warwickshire County Council - using a Federated UK Government ID in trusted Local Authority transactions.

By Graham Dunnings and Ian Litton

Table of Contents

Introduction

This document accompanies the Alpha project White Paper. The White Paper gives an overview of the project and its outcomes, while this guide explains in more detail the technologies that were implemented by the Relying Party (RP)/Service Provider (SP)[1] to achieve the business objective.

We will describe how Warwickshire County Council (WCC - the SP):

- created two test applications typical of a local authority (reporting a pothole, and requesting a disabled parking bay)
- used Google accounts to allow customers to authenticate to the pothole application with an identity at LoA1 (Level of Assurance 1)[2]
- used the Gov.UK Identity Assurance Hub (the Hub) to authenticate customers to the disabled parking bay application with an identity at LoA2 (Level of Assurance 2).

These Technical Findings are intended to give SPs a good overview of what was required to connect to the Hub, the standards involved, and some issues to be aware of. Please note that the Hub is still in development and therefore some of the details in this report are subject to change.

Audience

The target audience is technical architects, technical managers and technical decision makers. There are also components that software development teams will find helpful for setting up and consuming identity tokens in their own applications.

Scope

This Technical Guide describes the technical components required by the SPs to consume authentication tokens from both the Hub and a social media IdP. It does not cover the requirements of Identity Providers (IdPs) or Federated Identity software providers. Low level detail on how to connect to the Hub and its SAML profiles will be provided by GDS in the form of the Identity Assurance Hub Service Profile - On-boarding Guide.

Architecture

This section gives a high level overview of the architectural components and their purpose. These are described in more detail in the sections that follow.

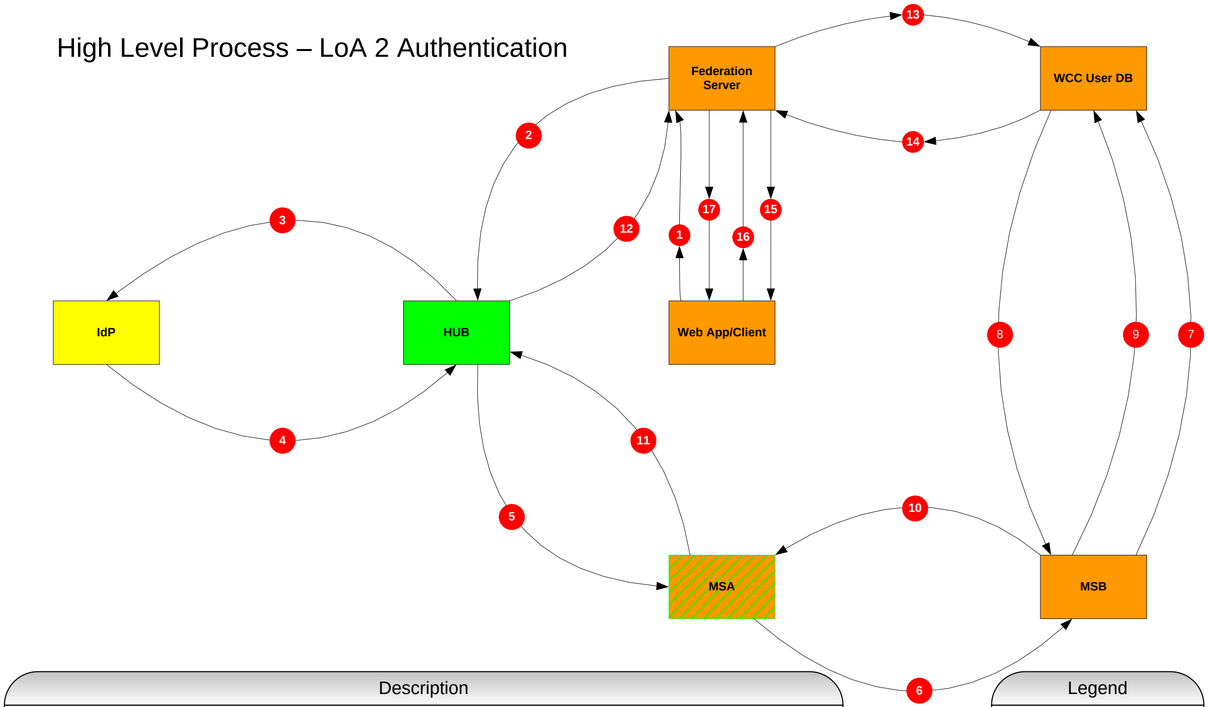
[1] A Service Provider and Relying Party are effectively the same entity. For clarity, this document will refer to that entity as the ‘Service Provider’

[2] The definition of Level of Assurance (LoA) is as per GPG 45, Chapter 5.

For the purposes of the Alpha we worked to the following assumptions:

- Use Open Source software where appropriate
- Adhere to Open Standards in order to consume authentication tokens
- Write applications in Ruby on Rails (RoR)
- Host WCC's solution on Linux (RHEL5) VM's
- Restrict network connectivity as we hosted test applications internally

Although we use used Open Source software where appropriate, we did implement proprietary components where, at that time of the Alpha, the skills did not exist in WCC, the open standards were complex, or an alternative would have taken too long to develop. For this reason it was decided that the Federation Server as shown in Figure 1 would be sourced from a commercial vendor. We used Ping Federate[3] from Ping Identity to perform this task.



High Level Process – LoA 2 Authentication

Description

1	Redirect client to Federation Server
2	Redirect client to Gov.uk Hub
3	Client Selects IdP and logs on
4	IdP sends Assertion to Gov.uk Hub
5	Gov.uk Hub, sends MDS to MSA
6	MSA Sends request to Match to MSB
7	MSB Queries User repository (LDAP)
8	Results returned
9	Update of Persistent ID if Successful
10	Results passed back
11	Result passed back (Match:No-Match)
12	Assertion sent to Federation Server
13 14	Attribute Lookup & return results
15 16	Client initiated, Exchange token for Attribs.
17	Attribs. returned to Application

Legend

IdP	IdP Infrastructure
HUB	Gov.uk Infrastructure
MSA	Quasi-SP Infratstructure
SP	SP Infrastructure
0	Process Step Number

Figure 1. This is a high level process; it does not cater for browser redirects between domains

[3] https://www.pingidentity.com/products/pingfederate/

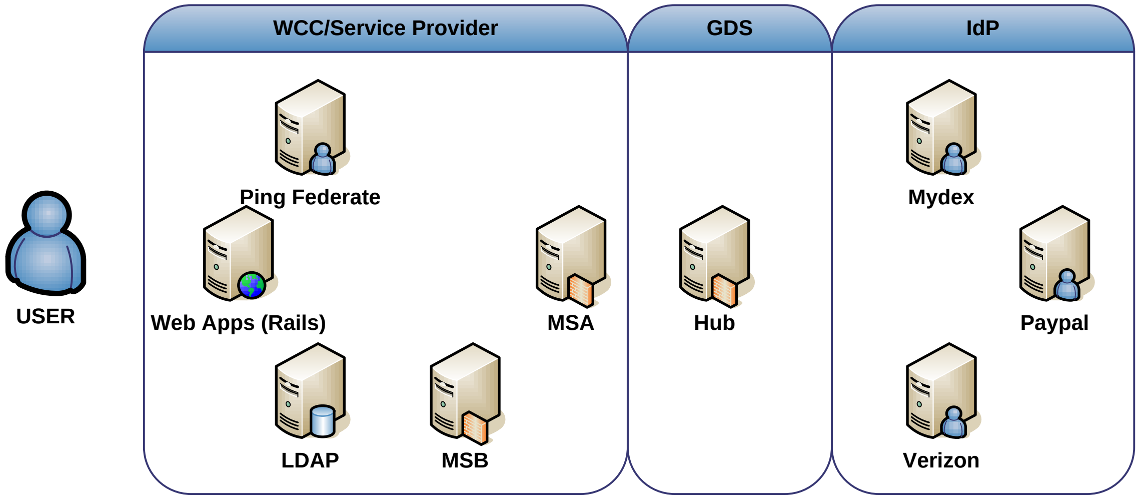


Figure 2. High level architecture showing boundaries and separation of functions

A key principle of the Hub is to ensure anonymity between SPs and IdPs. We do not, therefore, describe any transactions with IdPs in this document as all SP transactions are with the Hub. This also simplifies the authentication process for SPs as there is only one Hub connection to make regardless of the number of IdPs providing identity assurance services.

The Federation Server provides an abstraction layer that allows us to simplify our applications by removing the majority of the identity assurance processing from the application and transferring it to the Federation Server. The Federation Server brokers all the conversations necessary between the application and the Hub or social media IdP, and between the application and our internal LDAP server. The Federation Server manages the SAML messages and the identity tokens.

The Matching Service Adapter (MSA) is provided by GDS but sits on the SP's infrastructure. It provides a simplified way for SPs to handle the Matching Data Set (MDS). The MDS originates from the IdP and is used to match user attributes to identities in the SP's local directory. The SP is responsible for creating the Matching Service Backend to interrogate the local directory (LDAP in our case) and returning the result to the MSA. The Hub expects a match; if no match is found the Hub will not return an assertion in the identity token to the Federation Server and the transaction will fail. For the purposes of the Alpha project we only tested "happy" user journeys where matches were found. We would need to build local user registration, or Just in Time (JiT) user provisioning for LoA2 into any live implementation.

Once an identity token containing an assertion is returned to the Federation Server, the Federation Server interrogates the LDAP directory and returns requested attributes and roles to the application that originated the login request. Although the attributes may well be in the MDS already passed from the Hub to the MSA, these cannot be used to populate form fields. Attribute exchange is logically separate from identity assurance and has not yet been implemented on the Hub. The constraints on the use of the MDS are related to the privacy principles agreed with the Identity Assurance Programme's Privacy and Consumer Advisory Group.[4]

Any role-based application permissions need to be handled by the originating application based on information returned from the LDAP directory i.e. Organisational Unit or security group membership.

Standards Used

OpenID

OpenID is a standard that allows any compliant Service Provider to accept authentication tokens from any compliant Identity Provider (IdP). For example, a user may create an OpenID compliant account with Google that can then be used on any web site that accepts OpenID accounts.

It is important to note that OpenID does not in itself give any degree of assurance that the identity presented to the Service Provider is valid. This is entirely dependent on how the IdPs registration process is managed. One benefit of the Hub is that it assures identities to LoA2, whereas social media identities are only assured to LoA1.

OpenID can be extended in order to transfer additional attributes (e.g. gender and address) from the IdP to the Service Provider. This is OpenID Attribute Exchange. Although in this Alpha we used OpenID Attribute Exchange with the LoA1 Google identities, this standard has not yet been implemented in the Hub.

SAML 2.0 (Security Assertion Markup Language)

SAML is a data format for securely exchanging authentication and authorisation data, typically between IdPs and SPs. SAML ensures the integrity and confidentiality of the message between the intended endpoints. The data exchanged allows the Service Provider to decide whether or not to give the user access to the services requested.

[4] https://www.gov.uk/government/consultations/draft-identity-assurance-principles/privacy-and-consumer-advisory-group-draft-identity-assurance-principles

It is important to note that SAML does not in itself give any degree of assurance that the identity of the person presented to the Service Provider is valid. This is entirely dependent on the trust framework the IdP's are working within.

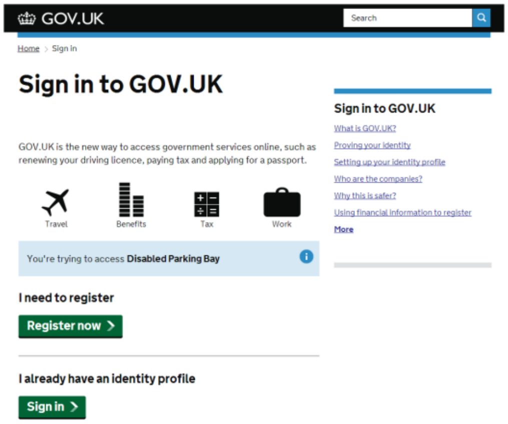
In the context of the Hub part of the 'onboarding' process includes the Service Provider and the Hub exchanging X.509 server certificates and details of endpoints to successfully communicate. This is called 'Metadata'. The contents of the SAML 'envelope' are encrypted and cryptographically signed. In the case of the IDA architecture, two certificates are used in the solution to sign assertions. It is essential that your solution can handle multiple public keys to verify signatures from the Hub. "Assertion Consumer Service" and "Signing" are the key descriptors in the SAML Specification.

OAuth 2.0

OAuth 2.0 is an open standard for authorisation that complements OpenID. It allows a user to authorise service provider 1 to access defined account information from service provider 2 based on an access token, and without the need to share the user's username and password with provider 1. A pre-existing "contract" must be in place between service provider 1 and service provider 2 to allow service provider 1 to access APIs from service provider 2. It is mostly used to allow RESTful services to consume tokens to access services and data.

Government Hub

At the time the project commenced, the Gov.uk Identity Hub was at version 1.0. During the project, the specification was updated and the solution was delivered to version 1.1[5]. The term 'Hub' is apt. The architecture is such that all other parties, IdP and SPs alike, communicate with the Hub alone. An SP cannot communicate directly with an IdP and vice versa. The reason behind this is one of privacy. Each SP or IdP will perform an on-boarding process to ensure that the endpoints using SAML are all compliant and working. Once onboarding has been completed, the SP initiated SSO (Single Sign On) adapter within the Federation Server will converse with the Hub using SAML to secure all messaging between the two endpoints.



Sign in to GOV.UK

GOV.UK is the new way to access government services online, such as renewing your driving licence, paying tax and applying for a passport.

You're trying to access **Disabled Parking Bay**

I need to register

I already have an identity profile

[5] Identity Assurance Hub Service SAML 2.0 Profile v1.1

The Hub will only talk to endpoints which have gone through an on-boarding process. On-boarding will be covered later in the document.

Federation Server

The Federation server provides core functionality to allow an SP to easily participate in the IDA Scheme. For the purposes of the Alpha we used Ping Federate as our Federation Server. It facilitates the creation of "connectors" or "adapters" to link to the Hub and other IdPs. This section talks about the functionality a Federation Server should provide, without going into the details of the specific implementation of Ping Federate. SPs may choose to use different Federation Software; this section will help them determine the features to look for.

The Federation Server consumes information through one connector and transposes it into another format whilst retaining its session and integrity. This means that different attribute naming conventions from different IdPs can be handled by the Federation Server rather than being embedded at application level. This becomes significant as the number of IdPs proliferates.

The Federation Server allows us to securely converse with the Hub using SAML, consume tokens returned from the Hub and then look up additional attributes from a local identity store. The results are then packaged up in a JSON format that can be consumed by the originating application.

The Federation Server needs to be able to handle multiple certificates (see SAML 2.0 above). In the SAML spec for the Hub, the "Assertion Consumer Service" and "Signing" are the key descriptors.

Although JiT user account provisioning is not yet supported by the Hub, it would be wise to ensure any Federation Server can handle this feature in future.

The Federation Server will need to be accessible from the internet as the user's browser is redirected to the Federation Server to kick the process off with the Hub.

Identity Provider (IdP)

Although the Hub is treated as an IdP by the Federation Server (the Hub acts as the broker and hides the IdP from the SP), Service Providers may wish to use social media IdPs, and may be involved with Trust Frameworks other than the Hub in future. Different IdPs may work to different protocols. The Federation Server should be able to handle multiple IdPs by defining the relevant connection details and protocols they use.

For the purposes of the Alpha we used Google Mail accounts for LoA1 access and UK Gov.UK accounts for LoA2 access.

Google OpenID connector

In order to allow users to authenticate to applications using their Google account it is necessary to use a Google Connector. Google operate two authentication connectors, Google OpenID and Google+ which is oAuth 2 compliant. For the purposes of the Alpha we chose the former.

In order for the Federation Server to consume services from Google and broker the conversations with the applications, the applications have to be registered with Google under an organisational account. An application "secret" is stored in the Federation Server and shared with Google to secure individual transactions. The Federation Server also stores an endpoint URL for the applications. Once this configuration is in place the individual applications are able to request identity tokens and user attributes via the Federation Server whenever a user attempts to authenticate using Google.

Ping Federate has a proprietary connector for Facebook, and uses the OAuth 2.0 connector for Twitter. We did not implement either of these for this Alpha.

More information on the configuration can be found at:
https://developers.google.com/accounts/docs/OpenID
https://developers.google.com/accounts/docs/OAuth2Login

Hub connector

In order to connect the Federation Server to the hub, GDS will provide an on-boarding process for the exchange of metadata, including the SAML requirements of the endpoints for both the Hub and the Federation Server. Separate on-boarding documentation from GDS goes into this in much more detail.

Service Provider (SP)

Just as the Hub acts as a broker to the Government assured IdPs, so the Federation server acts as a single endpoint to the Hub. The Hub has one place to send assertions and the Federation Server handles internal communication with the local directory and individual applications. In our case we developed a Ruby Gem to communicate with the Federation Server and to consume authentication tokens from the Federation Server. In this way we were able to abstract common code out of individual applications. (Currently the Hub only assures identities at LoA2. However, the specification caters for several levels of assurance and the assertion will contain which level the principal has been assured to.[6])

Local Directory and Attribute mapping

We used an LDAP directory for the pilot. Initially we wanted to implement the OpenID Connect schema on OpenLDAP but the OpenID Connect schema was found to be overly complex for the purposes of the Alpha and we hit issues when we tried to manipulate the schema. Instead we installed OpenDJ and standard inetOrgPerson, person and organizationalPerson objects in one schema as at this point we were collecting minimal attributes. Additional attributes can be added if required.

The Federation Server needs to interrogate the Local Directory from where it can retrieve user attributes once an authentication token is sent from the Hub.

An adapter needs to be set up in the Federation Server to specify the connection details, User DN and credentials relevant to the local directory. As different IdPs may use different attribute naming conventions, the Federation Server also needs to be able to map the IdP's attribute names to those found in the SP's LDAP schema or database.

Just in time (JiT) provisioning of identities in our local directory was out of scope for this Alpha. JiT provisioning would be required to provision details of new users in the local directory. For the purposes of this Alpha we worked with pre-registered identities. Ping Federate, although it can handle JiT provisioning with databases, cannot be configured for JiT provisioning on LDAP. This is due to the complexity of LDIF (the LDAP Data Interchange Format - for data and schema import) and class constraints within the hierarchy of LDAP schemas; as such it is almost impossible to cater for all eventualities. It is essential though, when considering architectural components, to ensure that they are fully compatible end to end. Because WCC wish to use JiT user provisioning to register new users, LDAP would not be a suitable identity repository for us to use with the current architecture.

Web Application Architecture

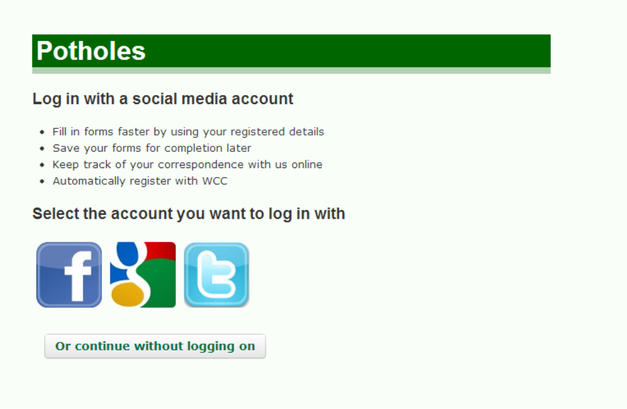
For the purposes of the Alpha project we engineered two test applications in Ruby on Rails, and a Ruby Gem to handle communication between the applications and the Federation Server. In the test cases we simply needed a user to be authenticated at the appropriate level (LoA1 or

[6] Identity Assurance Hub Service Profile – Authentication Contexts v1.1
6

LoA2). In some applications we would need to apply role-based access and permissions at the application level. This was out of scope for the Alpha project.

Web Application to consume LoA1

We created a Ruby on Rails test application to simulate the ability to report a pothole to WCC. One of the aims of the Alpha was to test users' attitudes to using social media identities in the context of local government transactions. We therefore gave the user the choice to report a pothole without logging on, or to authenticate using a Google account. (The user was also offered the choice on the user interface to authenticate using Facebook and Twitter, but these options were not implemented for user experience testing).



Potholes

Log in with a social media account

- Fill in forms faster by using your registered details
- Save your forms for completion later
- Keep track of your correspondence with us online
- Automatically register with WCC

Select the account you want to log in with

The application itself interfaces with the Federation Server and not directly with Google. This allows us to abstract most of the identity assurance functions from individual applications and implement them centrally in the Identity Broker. This keeps the individual applications as simple as possible and ensures there is a single place to manage the connections to Google and other IdPs.

In terms of identity assurance the Ruby on Rails applications and Gem are responsible for:

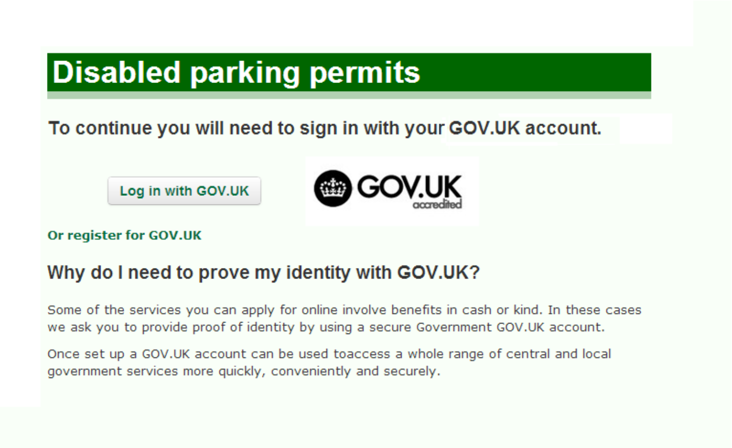
- identifying if the user is logged on with an identity that confers the required level of assurance (LoA1 in the case of reporting a pothole)
- offering the user the choice to log on with Google if they are not currently authenticated
- passing the user to the Federation Server to broker the conversation with Google if the user decides to log on
- consuming a token passed from Google to the Federation Server and then to the application if the user chooses to log on
- consuming additional attributes passed from the Google account to the Federation server
- displaying the additional attributes on the pothole form

Issues to address include:

- distinguishing between users who are authenticated to LoA1 from those authenticated to LoA2. It is not sufficient to simply identify users as logged in or not.
- ensuring that session variables are correctly handled as users move between applications on a web site

Web Application to consume LoA2

We created a Ruby on Rails test application to simulate the ability to request a disabled parking bay. This was the means to test the need for a LoA2 identity and links to the Hub. In contrast to the Pothole application, the user was required to log in using a Gov.UK account to complete transaction.



Disabled parking permits

To continue you will need to sign in with your GOV.UK account.

Or register for GOV.UK

Why do I need to prove my identity with GOV.UK?

Some of the services you can apply for online involve benefits in cash or kind. In these cases we ask you to provide proof of identity by using a secure Government GOV.UK account.

Once set up a GOV.UK account can be used toaccess a whole range of central and local government services more quickly, conveniently and securely.

As with the LoA1 application, we abstracted as much of the identity assurance functionality as possible from the application and into the Federation Server

In terms of identity assurance the application is responsible for:

- identifying if the user is logged on with an identity that confers the required level of assurance (LoA2 in the case of requesting a disabled parking bay)
- passing the user to Federation Server to broker the conversation with the Hub if the user is not logged on
- consuming a token passed from the Hub to the Federation Server when the user logs on
- granting access to the application if the logon is successful

Issues to address include:

- distinguishing between users who are authenticated to LoA1 from those authenticated to LoA2. It is not sufficient to simply identify users as logged in or not.
- ensuring that session variables are correctly handled as users move between applications on a web site

Matching Service Adapter (MSA)

The MSA is a java application that needs to be installed on a Service Provider server that has bi-directional connectivity back to the Hub. In our case we installed it on the same server as Ping Federate. The MSA uses completely different ports, but could use the existing external DNS entry to minimise the number of hosts. It is a self-contained service that requires minimal configuration. The configuration is covered in the GDS on-boarding documents, and involves the exchange of entity IDs, and setting up private/public keys. Dependant on the platform you run the MSA on, it is best to run it as a service or daemon in its own right.

Matching Service Backend (MSB)

The Service Provider is responsible for writing an MSB to talk to the MSA. The MSB consumes the attributes in the Matching Service Dataset (MSD) passed to it by the MSA, uses these to interrogate the local directory, establishes if there is a match, and passes the match/no match result back to the MSA. If a match is established for the first time the Persistent Identifier (PID) contained in the matching dataset is added to the matching record in the Local Directory. The MSA then sends the result back to the Hub.

Although the functionality of the MSB is easily expressed, it is in the implementation of the MSB that a lot of the risk and complexity for the Service Provider rests. If the MSB reports false positives then an assured user account will be linked to the wrong local account. This could lead to incorrect service delivery, compromised case files and inappropriate data disclosures.

False negatives should not be an issue once a link is established between the MDS and the local directory. At this point a Persistent Identifier (PID) is exchanged that can be used as a primary key in future transactions, as long as the user authenticates with the same IdP.

Data matching is typically based on an algorithm and a confidence threshold based on a set of rules. The rules determine how to deal with common variations in names (Robert, Rob, Bob, Robby etc), which attributes confer greater or lesser levels of confidence in a match, when an “automatic” match is deemed suitable and so on. Different services may be prepared to bear a greater level of risk. The data matching process has to deal with time-dependent changes in data as the data in the local directory and that held by the IdP can get out of step over time.

The complexity of data matching may present a significant barrier to implementation by local authorities; this would seem to be an area where ‘develop once and reuse many times’ would make real sense.

During the Alpha project the MSD contained only two attributes - FirstName and Surname. Additional attributes will be added to the MSD before live implementation.

Test Data

In the absence of JiT user account provisioning, the Alpha project user experience testing concentrated on users pre-registered with the Service Provider and “happy” user journeys. A set of test data was created and shared with the IdPs to ensure their registration processes would work correctly and achieve suitable matches through the MSA and MSB. Different IdPs have different data and validation requirements in their registration processes, so the test data needs to be set up accordingly.

Contributors

WCC - Principal Investigator - Ian Litton
WCC - Identity Architect/OIX Project Lead Technical Contact - Graham Dunnings
WCC - Lead Developer - Rob Nicols
WCC - System Administration - Ranjit Khera
Ping Identity - Regional Solutions Architect - Chris Robins

Appendix 1

GPG 43 (Good Practice Guide 43):
https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/138004/GPG_43_RSDOPS_issue_1.1_Dec-2012.pdf

GPG 44:
https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/204447/GPG_44_-_authentication_credentials_in_support_of_HMG_online_services_issue_1.2_May_2013_1_.pdf

GPG 45:
https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/204448/GPG_45_Identity_proofing_and_verification_of_an_individual_2.0_May-2013.pdf

- oauth 2.0
- OpenID
- OpenID Connect

Firewall Rules from an SP perspective using our solution. Please note that all 3 of our VM servers were in a test environment, further segregation may be required for access to the Directory (LDAP protocol etc) via the use of reverse proxies etc as per your own Information Security standards.

Direction	Source	Destination	Service	Notes
Inbound	Any	Federation Server	9031	Federation Service
Inbound	Any	Web Application	80/443	Web Provisioning
Inbound	Hub	MSA	<from GDS>	
Outbound	MSA	Hub	<from GDS>	

High level technical implementation steps

- Configure servers and set up external IP & DNS issuance for Ping Federate and web application server

- Create all SSL certificates, self-signed for testing. In the future, all externally facing servers that will consume LoA2 identities will require EV certificates.
- Configure Ping Federate IdP Initiated SSO adapter
- Export the metadata from Ping Federate and share with GDS
- Configure LDAP
- Configure Ping Federate to consume attributes from LDAP
- Update applications to use Ping Federate as its point of authentication and consume the responses