

BEIS Digital Business Identity

# D2 & D4 – Operational Runbook & Data Schemas

**Service ID**:

**Supplier**: KPMG LLP - Enabling Data Sharing to improve Public Services

**Department**: BEIS DDaT

**Project**: BEIS Digital Identity for Business Programme

*This proposal is made by KPMG LLP, a UK Limited Liability Partnership and a member firm of the KPMG network of independent firms affiliated with KPMG International Cooperative (“KPMG International”), a Swiss entity. The proposals set out in this document do not constitute an offer capable of acceptance. They are in all respects subject to satisfactory completion of KPMG's procedures to evaluate prospective clients and engagements, including independence and conflict checking procedures, and the negotiation, agreement, and signing of a specific engagement letter or contract. KPMG International provides no client services. No KPMG member firm has any authority to obligate or bind KPMG International or any other member firm vis-à-vis third parties, nor does KPMG International have any such authority to obligate or bind any member firm*

Version 1.0 – 25th March 2020

Contents

[1. Purpose 5](#_Toc36038998)

[2. Core Components 6](#_Toc36038999)

[3. Solution Setup (Product agnostic) 8](#_Toc36039000)

[Required Skills 8](#_Toc36039001)

[What do you need? 8](#_Toc36039002)

[Configuration 8](#_Toc36039003)

[LDAP Data Source 8](#_Toc36039004)

[Federation Hub 9](#_Toc36039005)

[Assurance Engine 10](#_Toc36039006)

[Registration Application 11](#_Toc36039007)

[BEIS Grants Application 12](#_Toc36039008)

[4. AWS Environment 13](#_Toc36039009)

[5. PingDirectory Configuration & Data Schema 14](#_Toc36039010)

[Configurations Details 14](#_Toc36039011)

[Schema Customisations 14](#_Toc36039012)

[Custom Attributes 14](#_Toc36039013)

[Custom Object Class 17](#_Toc36039014)

[Run/Stop Commands 18](#_Toc36039015)

[Run Directory Server 18](#_Toc36039016)

[Stop Directory Server 18](#_Toc36039017)

[6. SAML Identity Provider (IdP) 19](#_Toc36039018)

[Installation 19](#_Toc36039019)

[Generate IDP Signing Certificate 19](#_Toc36039020)

[Run Command 19](#_Toc36039021)

[Metadata URL 19](#_Toc36039022)

[7. PingFederate Configuration 20](#_Toc36039023)

[Data Stores 20](#_Toc36039024)

[Password Credential Validators 20](#_Toc36039025)

[Identity Provider Adapter 21](#_Toc36039026)

[IDP Connections 22](#_Toc36039027)

[Authentication Policy 23](#_Toc36039028)

[OIDC Clients 23](#_Toc36039029)

[Run/Stop Commands 23](#_Toc36039030)

[8. Build & Deployment Instructions 24](#_Toc36039031)

[OIDC Applications 24](#_Toc36039032)

[Assurance Engine 24](#_Toc36039033)

[Encrypt Text using jasypt 24](#_Toc36039034)

# Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Person | Role | Date |
| 0.1 | Deepansh Mathur | Technical Architect | 02/Mar/2020 |
| 0.2 | Steve Ford | Engagement Manager | 04/Mar/2020 |
| 0.3 | Mike Nelsey | Commercial/Business | 06/Mar/2020 |
| 1.0 | Steve Ford | Engagement Manager | 25/Mar/2020 |

Please see the engagement document repository for detailed revision and version increment information.

# Purpose

The purpose of this document is to provide a low level understanding of the technical design used for development of the Digital Business Identity (DBI) Minimum Viable Product (MVP). The core components defined in the architecture design for MVP with details of configuration and data schemas used for the development are detailed in this document. This document also provides a list of commands/scripts used to run each of the components and configuration to connect them together. The configurations of commercial tools utilised has been provided for reference; while the configuration used is one way of achieving the requirements for the development of MVP, there may alternatives.

# Core Components

The MVP developed for BEIS Digital Business Identity programme comprises of 5 core components. These components are currently hosted on KPMG AWS environment. Each of these components play a unique role in user journey and are described below –

1. Federation Hub (FH)

A Federation Hub (FH) has been developed using PingFederate 9.3 as the foundation service. The purpose of the FH is to provide federated authentication to DBI account users and enable GOV.UK Verify users to register for a DBI account.

1. Data Store (DS)

The FH uses PingDirectory to provide credential validation for DBI account users accessing applications utilising federated authentication. The PingDirectory is a standard LDAP directory with customisations detailed in [section 4.2](#_4.2_Schema_Customisations) to include custom object classes and attributes required for the processing of MVP.

1. Assurance Engine (AE)

Assurance Engine is a RESTful web service engine to support registration process, facilitating searching and validation of business attributes provided during the business to identity linking. In addition, the AE is used to validate information provided by the business owner during the addition of a business. The AE connects with various government agencies as well as private data holders via REST APIs to complete the validations and verifications.

1. SAML Identity Provider (IdP)

Due to technical glitches experienced during the integration of GOV.UK Verify with the federation hub, it was decided to use a simple SAML Identity Provider to impersonate Verify and provide exactly same LOA 1 attributes which Verify would have provided. Please note that this Identity Provider is developed for demonstration/PoC purposes only and should NOT be used in any production environment.

1. OpenID Connect 1.0 (OIDC) Relying Parties

The MVP also comes with 2 Java Spring Boot applications. The Java version used for the development of both the application is long support version 1.8.0\_202 which is also compatible with the version of Java used by the FH. The 2 applications are:

* BEIS Grants Application

This application is used to simulate BEIS Grants application management system and will be used when performing any transactional activities involving BEIS. It has been designed as an OIDC Relying Party and communicates with the FH for authentication using OIDC protocol. The FH allows use of DBI accounts to authenticate users to access this application. This application will also have links to the Registration Application to allow user to view/update DBI profile as well as update business information.

* Registration Application

The Registration Application is a similar OIDC Relying Party connected with the FH for authentication and attribute retrieval. This application allows users to create a DBI account by verifying their identity using IdP (simulating Verify) for authentication via FH. The user will be allowed to link businesses to their identities using this application once a DBI account is created. This application connects with the AE for validations etc.

# Solution Setup (Product agnostic)

The reference architecture used to build the MVP is not dependent on the commercial tools used but is designed to utilise open standards and technology. So any commercial product supporting these open standards can be used to setup a similar MVP in future. This section provides information on how to run a prototype similar to the MVP using any commercial tool.

## Required Skills

The following minimum skills are required for developers working on the solution setup –

* Advanced Java (Minimum version 8) including Spring Boot
* Basic LDAP commands
* Understanding of federation protocols (for e.g. SAML 2.0, OAuth 2.0, OIDC 1.0)
* Implementation experience of the tools used

## What do you need?

In order to setup a prototype for Digital Business Identity, you will need

* Linux or Windows machines (could be VMs) with access to the internet
* Oracle Java JDK 1.8.202 (minimum)
* Apache Maven 3.6 or later
* A data source to store identities and linked businesses (for e.g. LDAP Directory or a relational database)
* A tool to provide federated authentication with federation hub functionality
* A simple SAML 2.0 Identity Provider (to simulate Verify)
* The Assurance Engine (delivered with the MVP)
* OIDC sample applications (delivered with the MVP)

## Configuration

The MVP deliverables are created to make it configurable so it could be plugged-in with any commercial product with minimum changes. This section provides details of configuration changes required for each component.

### LDAP Data Source

A data source will be required to store Digital Business Identities and their linked business information. The two applications created to simulate registration process and act as a BEIS Grants application have code to connect to an LDAP data source to submit the registration information. It is possible to use a different type of data source for e.g. RDBMS or similar but that will require appropriate code to be updated in the applications to connect and submit registration data.

LDAP Directory data source needs to be prepared before connecting it with the application. The LDAP schema extension required to setup the solution needs to be exactly same as done for PingDirectory in the MVP. The configuration changes required are mentioned [here](#_Schema_Customisations). In order to reuse the simulating applications it is expected that the format of extended JSON attributes are not changed. However, any extension to the JSON format is accepted but as a minimum the values should have JSON keys given in the examples [here](#_Custom_Attributes).

### Federation Hub

The solution requires a mechanism to allow users accessing either of the two applications to authenticate themselves via different identity providers depending on what functionality they intend to use. This functionality has been achieved in the MVP using PingFederate Server 9.3 as it supports Federation Hub use cases. Any tool that can provide similar Federation Hub capability can be used to replace PingFederate Server. The tool used may have different ways to achieve these functionalities by making configuration changes but it should be able to fulfil following functional requirements:

* Support for open standards for e.g. SAML 2.0 and OIDC 1.0.
* Two separate identity providers within the Federation Hub.
  + Identity Provider connection (simulating Verify).
  + LDAP Authentication – The federation hub needs to be configured to perform credential validation against the data source used to store registered accounts.
* The Federation Hub should be able to act as an OAuth Authorisation Server supporting JSON Web Token based response.
* Both the applications should be added to Federation Hub as OIDC clients. These clients will authenticate using a client secret and should be allowed Authorisation Code grant types.
* The hub should be able to differentiate authentication requests coming from these clients in different user journeys and should be able to provide single sign-on between both of the applications. For e.g. an authentication request can come from Registration Application from a new user trying to setup a DBI account, then it should be taken to Verify (simulating) and not through LDAP authentication.
* After successful authentication, following attributes should be sent back in different OIDC responses –

Table 3.1 – OIDC Responses

| Ref. | User Attribute | OIDC Response |
| --- | --- | --- |
| AA1 | firstName | Access Token |
| AA2 | lastName | Access Token |
| AA3 | address | Access Token |
| AA4 | gender | Access Token |
| AA5 | username | Access Token |
| AA6 | displayName | Access Token |
| AA7 | birthDate | Access Token |
| IA1 | given\_name | ID Token |
| IA2 | family\_name | ID Token |
| IA3 | name | ID Token |
| IA4 | middle\_name | ID Token |
| IA5 | address | ID Token |
| IA6 | birthdate | ID Token |
| IA7 | gender | ID Token |
| IA8 | identityprovider | ID Token |
| IA9 | postal\_code | ID Token |
| UA1 | business | UserInfo |
| UA2 | delegates | UserInfo |

As long as the selected tool to support federation hub is able to provide context based switching of authentication source and returning listed attributes within different OIDC response the setup should be able to run similar to the MVP.

### Assurance Engine

The Assurance Engine is a Spring Boot application with two modules –

* Assurance Engine Processor
* BEIS Assurance Engine

Assurance Engine Processor has all the business logic and functionality to connect to external agencies for data verification. These external agencies expose RESTful web services which are invoked and consumed by the processor. Different agencies have different ways to identify or authorise access to these APIs; for example Companies House requires registration of the application invoking it and in response to that registration, an API key is issued which must be sent with all subsequent API requests. More details on the Companies House APIs can be found [here](https://developer.companieshouse.gov.uk/api/docs/index/gettingStarted/apikey_authorisation.html). Market Location utilises a similar registration process and requires a bearer token to be present in the authorisation header along with the client key for all requests.

These keys and tokens are stored in a properties file within Assurance Engine Processor.

Table 3.2 Assurance Engine configurable properties:

| Ref. | Property Name | Value/Format |
| --- | --- | --- |
| File Name - src/main/resources/serverconfig.properties | | |
| AE1 | companieshouse.rs.ws.api.key | API Key for Companies House APIs |
| AE2 | marketlocation.rs.ws.token | Bearer token for Market Location APIs |
| AE3 | marketlocation.rs.ws.client.id | Client ID for Market Location APIs |

You must update these properties with your own API keys and tokens before the Assurance Engine will be operational. Like all the component applications, the Assurance Engine also supports encrypted properties values. The process to encrypt a text value and the format to be used for encrypted values is provided [here](#_Encrypt_Text_using).

### Registration Application

The Registration Application has been developed to simulate the user experience in creating and managing a Digital Business Identity profile. This application has 3 interfaces:

* RAI 1 – Registration Application 🡪 Federation Hub
* RAI 2 – Registration Application 🡪 Assurance Engine
* RAI 3 – Registration Application 🡪 Identity Data Source (LDAP)

This Spring Boot application uses jasypt 1.8 library to read encrypted properties from two separate properties file within the application. These properties file may or may not contain encrypted values. The table below gives details about each property the file location where it can be updated.

Table 3.3 Registration Application Configurable Properties:

| Ref. | Property Name | Value/Format |
| --- | --- | --- |
| File Name - src/main/resources/credentials.properties | | |
| RP1 | client.id | OIDC Client ID configured for Registration Application on the Federation Hub. For e.g.  client.id=Registration Application |
| RP2 | client.secret | OIDC Client Secret configured for Registration Application on the Federation Hub. For e.g.  client.secret=2Federate |
| RP3 | ldap.server.url | LDAP URL to connect to the data source. For e.g.  ldap.server.url=ldaps://ldaphost.com:1636 |
| RP4 | ldap.base.dn | LDAP Base DN to connect to the data source to create a new DBI profile. For e.g.  ldap.base.dn=ou=dbiprofiles,dc=mvpfedhub,dc=com |
| RP5 | ldap.principal | User DN with read, write and search privileges on the base dn provided. For e.g.  ldap.principal=cn=Directory Manager |
| RP6 | ldap.secret | Password for the user added in ldap.principal property. For e.g.  ldap.secret=2Federate |
| RP7 | ae.client.id | Client ID provided after registration of the application on Assurance Engine. For e.g.  ae.client.id=djhsfhkjsfdhkhskdf |
| RP8 | ae.authorisation.token | Authorisation Token provided after registration of the application on Assurance Engine. For e.g.  ae.authorisation.token=shfkjhdfhsdhfljshdljfhlsdhfl |
| File Name - src/main/resources/resources.properties | | |
| RP9 | assurance.engine.rs.base.url | Update the hostname and port where you have the Assurance Engine application running. For e.g.  assurance.engine.rs.base.url=https://<aehost>:<aeport>/BEISAssuranceEngine/api/v1 |

### BEIS Grants Application

The BEIS Grants application is used to provide simulation of a grants application management system and to prove that the registered Digital Business Identity accounts can use their credentials to login into this application and see their linked businesses using it. This application has a single interface:

* GAI 1 – Grants Application 🡪 Federation Hub

Similar to the registration application, jasypt 1.8 libraries have been used in the grants application also. So this application supports encrypted as well as clear text values in properties file.

Table 3.4 Grants Application Configurable Properties:

| Ref. | Property Name | Value/Format |
| --- | --- | --- |
| File Name - src/main/resources/credentials.properties | | |
| GP1 | client.id | OIDC Client ID configured for BEIS Grants Application on the Federation Hub. For e.g.  client.id=Grants Application |
| GP2 | client.secret | OIDC Client Secret configured for BEIS Grants Application on the Federation Hub. For e.g.  client.secret=2Federate |

# AWS Environment

The core components are currently hosted on KPMG AWS environment. This section provides details of the hostname, ports and folder structures used by each component.

Table 4.1 AWS Hosts & components deployed:

| Ref. | Hostname | Components | Installation Directory |
| --- | --- | --- | --- |
| AWS1 | ec2-35-178-212-239.eu-west-2.compute.amazonaws.com | PingFederate Server 9.3  Ping Identity Directory Server 7.3.0.3  Java JDK 1.8.0\_202 | /usr/local/pingfederate-2  /u01/PingDirectory  /u01/jdk1.8.0\_202 |
| AWS2 | ec2-18-130-130-112.eu-west-2.compute.amazonaws.com | Assurance Engine 1.0  SAML Identity Provider  BEIS Grants Application 1.0  Registration Application 1.0  Apache Tomcat Server 8.5.50  Apache Maven 3.6.3  Java JDK 1.8.0\_202 | /u01/applications/AssuranceEngineMaster  /u01/applications/saml-idp-master  /u01/applications/BEISGrantsApplication  /u01/applications/RegistrationApplication  /u01/apache-tomcat-8.5.50  /u01/apache-maven-3.6.3  /u01/jdk1.8.0\_202 |

# PingDirectory Configuration & Data Schema

The Federation Hub uses PingDirectory as its local identity and business data store. All the users who registers for a DBI account are stored as an identity account in this directory. Once the DBI account is created in the directory server all the businesses user registers as a director of the company will also be added to same user account.

## Configurations Details

Table 5.1 – PingDirectory Server Configuration details:

| Attribute Name | Attribute Value |
| --- | --- |
| Instance Name | DIPMVPFedHubDir |
| Host Name | ec2-35-178-212-239.eu-west-2.compute.amazonaws.com |
| Root User DN | cn=Directory Manager |
| LDAP Listener Port | 1389 |
| Secure Access | Enable StartTLS  Enable LDAPS Port 1636  Self-Signed Certificate  Generate Default Truststore |
| HTTPS Listener Port | 8443 |
| Console URL | <https://ec2-35-178-212-239.eu-west-2.compute.amazonaws.com:8443/console> |
| Configuration URL | <https://ec2-35-178-212-239.eu-west-2.compute.amazonaws.com:8443/config> |
| Consent URL | <https://ec2-35-178-212-239.eu-west-2.compute.amazonaws.com:8443/consent> |
| Directory REST API | <https://ec2-35-178-212-239.eu-west-2.compute.amazonaws.com:8443/directory/v1> |
| Documentation | <https://ec2-35-178-212-239.eu-west-2.compute.amazonaws.com:8443/docs> |
| Swagger UI | <https://ec2-35-178-212-239.eu-west-2.compute.amazonaws.com:8443/api-docs> |
| Directory Data | Base DN: dc=mvpfedhub,dc=com |
| Identity Data OU | ou=federated-identities, dc=mvpfedhub, dc=com |

## Schema Customisations

The basic installation of PingDirectory has been extended to include new attributes required to store business information by the Registration Application. Following customisations were done –

### Custom Attributes

Three new attributes are created as follows:

* identityprovider – This attribute was created to store the information about the selected identity provider by the user during the registration process.

LDIF:

*objectClass: top*

*objectClass: ldapSubentry*

*objectClass: subschema*

*cn: schema*

*attributeTypes: ( identityprovider-OID NAME 'identityprovider' DESC 'Identity Provider selected by the user during registration' SYNTAX*

*1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE USAGE userApplications*

*X-SCHEMA-FILE '99-user.ldif' )*

Possible Values: postoffice, experian, Barclays

Format - Text

* delegates – This multivalued attribute was created to store information about the delegates for each business as delegated by the business owner in their Digital Business Identity profile.

LDIF:

*objectClass: top*

*objectClass: ldapSubentry*

*objectClass: subschema*

*cn: schema*

*attributeTypes: ( business-OID NAME 'business' DESC 'Businesses a user is authorised to act on behalf of.' SYNTAX 1.3.6.1.4.1.30221.2.3.4{0} USAGE userApplications X-SCHEMA-FILE '99-user.ldif' )*

Format: JSON

E.g. {“chBusinessId”:”Company Registration Number”,”uid”:”DBI Login ID of delegate”}

* Business – This multivalued attribute will store information about each business a director registers as an owner. This attribute will also be stored in delegate’s identity profile with a difference of role value.

LDIF:

*objectClass: top*

*objectClass: ldapSubentry*

*objectClass: subschema*

*cn: schema*

*attributeTypes: ( business-OID NAME 'business' DESC 'Businesses a user is*

*authorised to act on behalf of.' SYNTAX 1.3.6.1.4.1.30221.2.3.4{0} USAGE*

*userApplications X-SCHEMA-FILE '99-user.ldif' )*

Format: JSON

E.g.

*{*

*"chBusinessId":{*

*"value":"09731021",  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"CompaniesHouse"*

*},  
   "businessName":{*

*"value":"RUGS4DECOR LIMITED",  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"CompaniesHouse"*

*},  
   "businessAddress":{*

*"locality":"Bradford",  
      "premise":"Unit C St Stephen Mills",  
      "addressLine1":"Ripley Street",  
      "postcode":"BD5 7JW",  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"CompaniesHouse"*

*},  
   "businessPostCode":{*

*"value":"BD5 7JW",  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"CompaniesHouse"*

*},  
   "isOwner":true,  
   "isDelegate":false,  
   "businessCRN":{*

*"value":"09731021",  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"CompaniesHouse"*

*},  
   "businessUTR":{*

*"value":"UTR-09731021444444",  
      "verifiedDate":"2020-02-28 09:15:00",  
      "verificationSource":"HMRC"*

*},  
   "businessDUNN":{*

*"value":"DUNN-09731021",  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"D&B"*

*},  
   "businessVAT":{*

*"value":"VAT-09731021",  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"HMRC"*

*},  
   "businessPAYE":{*

*"value":"PAYE-09731021",  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"HMRC"*

*},  
   "businessSIC":{*

*"sicCodes":[*

*"47530"*

*],  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"CompaniesHouse"*

*},  
   "businessBankDetails":{*

*"sortcode":{*

*"value":"sort-number-09731021",  
         "verifiedDate":"2020-02-28 09:26:06",  
         "verificationSource":"LLOYDS"*

*},  
      "accountNumber":{*

*"value":"09731021",  
         "verifiedDate":"2020-02-28 09:26:06",  
         "verificationSource":"LLOYDS"*

*}*

*},  
   "businessPSC":{*

*"personWithSignificantControl":[*

*{*

*"name":"Mrs Pavli Joshi",  
            "forename":null,  
            "surname":null,  
            "pscID":null,  
            "title":null*

*}*

*],  
      "verifiedDate":"2020-02-28 09:26:06",  
      "verificationSource":"CompaniesHouse"*

*},  
   "noOfEmployees":{*

*"value":"200",  
      "verifiedDate":"2020-02-26 15:13:50",  
      "verificationSource":"HMRC"*

*},  
   "businessTurnOver":{*

*"value":"100000",  
      "verifiedDate":"2020-02-26 15:13:50",  
      "verificationSource":"HMRC"*

*},  
   "hasCharges":false,  
   "registeredOfficeIsInDispute":false,  
   "businessStatus":"ACTIVE",  
   "hasInsolvencyNotices":false,  
   "riskScore":100*

*}*

### Custom Object Class

A new structural object class called “beisPerson” has been added to PingDirectory schema to allow addition of the above attributes to user profile. The Registration Application will create a new user with this object class and will add the above attributes in addition to various other optional and required attributes as applicable according to the object class hierarchy in the schema.

The “beisPerson” object class inetOrgPerson class as the parent to make sure we get all the required identity attributes in addition to the custom attributes mentioned above.

LDIF:

*objectClass: top*

*objectClass: ldapSubentry*

*objectClass: subschema*

*cn: schema*

*objectClasses: ( beisPerson-OID NAME 'beisPerson' DESC 'BEIS identity class' SUP*

*inetOrgPerson STRUCTURAL MAY ( business $ dateOfBirth $ delegates $ gender $*

*identityprovider $ uniqueIdentifier ) X-SCHEMA-FILE '99-user.ldif' )*

## Run/Stop Commands

PingDirectory is installed on AWS1 server in /u01/PingDirectory folder. Following are the commands that can be used to start and stop the directory server.

### Run Directory Server

To start the directory server, open a SSH terminal to AWS1 and run the following commands.

$ cd /u01/PingDirectory/bin

$ ./start-server

### Stop Directory Server

To stop the directory server, open a SSH terminal to AWS1 and run the following commands.

$ cd /u01/PingDirectory/bin

$ ./stop-server

# SAML Identity Provider (IdP)

The SAML Identity Provider app provides a simple SAML Identity Provider to test SAML 2.0 Service Providers with SAML 2.0 Web Browser SSO profile. The IdP uses nodejs to build and run the application as a standalone server.

## Installation

Run the following command from inside the installation directory –

$ cd /u01/applications/saml-idp-master

$ npm install

## Generate IDP Signing Certificate

Run the following command from the AWS host running the IdP application –

$ cd /u01/applications/saml-idp-master

$ openssl req -x509 -new -newkey rsa:2048 -nodes -subj '/C=US/ST=California/L=San Francisco/O=JankyCo/CN=Test Identity Provider' -keyout idp-private-key.pem -out idp-public-cert.pem -days 7300

## Run Command

Run the following command to start the IdP Server

$ cd /u01/applications/saml-idp-master/bin

$ nohup ./run.js –acsUrl <https://ec2-35-178-212-239.eu-west-2.compute.amazonaws.com:9031/sp/ACS.saml2> --audience DIPFedHub --port 8080 –host ec2-18-130-130-112.eu-west-2.compute.amazonaws.com > console.log &

## Metadata URL

The IdP server metadata is available on the following URL once it is started –

http://ec2-18-130-130-112.eu-west-2.compute.amazonaws.com:8080/metadata

# PingFederate Configuration

The Federation Hub has been configured based on a PingFederate Server 9.3 to provide multiple ways for authentication and attribute retrieval using the underlying PingDirectory server. The following artefacts have been configured in the PingFederate server for MVP development.

## Data Stores

The PingDirectory server instance has been installed on the same AWS1 host as PingFederate server. The directory instance has been added as a Data Store in PF.

Table 7.1 – PingFederate Data Store configuration:

| Attribute Name | Attribute Value |
| --- | --- |
| Data Store Name | PingDir-SSL |
| Hostname(s) | localhost:1636 |
| LDAP Type | PingDirectory |
| LDAPS | True |
| Username | Cn=Directory Manager |
| Mask Values in Log | False |
| Test Connection on Borrow | False |
| Test Connection on Return | False |
| Create New Connections in Necessary | True |
| Verify LDAPS Hostname | True |
| Minimum Connections | 10 |
| Maximum Connections | 100 |
| Maximum Wait (ms) | -1 |
| Time Between Eviction (ms) | 60000 |
| Read Timeout (ms) | 3000 |
| Connection Timeout (ms) | 3000 |
| DNS TTL (ms) | 60000 |
| LDAP DNS SRV Record prefix | \_ldap.\_tcp |
| LDAPS DNS SRV Record prefix | \_ldaps.\_tcp |

## Password Credential Validators

A password credential validator was created to utilise the PingDirectory data store connection to perform credential validations for a DBI login user flow. The configuration details are given below –

Table 7.2 – PingFederate PingDirectory PCV configuration:

| Attribute Name | Attribute Value |
| --- | --- |
| Instance Name | PCV: PingDir |
| Instance ID | PCVPingDir |
| Type | LDAP Username Password Credential Validator |
| Class Name | org.sourceid.saml20.domain.LDAPUsernamePasswordCredentialValidator |
| Parent Instance Name | None |
| LDAP Datastore | PingDir-SSL |
| Search Base | Dc=mvpfedhub,dc=com |
| Scope of Search | Subtree |
| Case-Sensitive Matching | False |
| Display Name Attribute | displayName |
| Mail Attribute | Mail |
| SMS Attribute | *<Blank>* |
| PingID Username Attribute | *<Blank>* |
| Mail Search Filer | mail=${mail} |
| Username Attribute | *<Blank>* |
| Enable PingDirectory Detailed Password Policy Requirement Messaging | True |
| Extended Contract | DN  givenName  mail  username |

## Identity Provider Adapter

The PingFederate server, acting as a Federation Hub, acts as both a Service Provider for an external Identity Provider as well as an Identity Provider for Service Providers or OIDC Clients. In order to enable PingFederate as an Identity Provider we need to create an Adapter to configure PCV:PingDir for authentication.

Table 7.3 PingFederate PingDirAuthAdapter configuration:

| Attribute Name | Attribute Value |
| --- | --- |
| Instance Name | PingDirAuthAdapter |
| Instance ID | PingDirAuthAdapter |
| Type | HTML Form IdP Adapter |
| Class Name | com.pingidentity.adapters.htmlform.idp.HtmlFormIdpAuthnAdapter |
| Parent Instance Name | None |
| Credential Validators | PCV: PingDir |
| Challenge Retries | 3 |
| Session State | Globally |
| Core Contract | policy.action  username 🡪 Pseudonym |
| Extended Contract | address  birthDate  firstName  lastName  gender |

As mentioned in the Adapter configuration above, we are extending the contract from this adapter to include additional attributes. The fulfilment of these extended attributes is done using the PingDir-SSL data source with following configuration for LDAP Directory Search –

Table 7.4 PingFederate Extended Contract fulfilment configuration

| Attribute Name | Attribute Value |
| --- | --- |
| Base DN | ou=federated-identities,dc=mvpfedhub,dc=com |
| Search Scope | SUBTREE\_SCOPE |
| Filter | uid=${username} |

## IDP Connections

In order to get the Service Provider role of Federation Hub we need to provide information about the SAML IdP Provider connection details to the PingFederate Server. Following are the configuration details provided to create the IdP Connection –

Table 7.5 PingFederate SAML IdP Connection configuration

| Attribute Name | Attribute Value |
| --- | --- |
| Connection Type | Browser SSO Profile Protocol SAML 2.0 |
| Metadata URL | <http://ec2-18-130-130-112.eu-west-2.compute.amazonaws.com:8080/metadata> |
| Partner’s Entity ID (Connection ID) | urn:example:idp |
| Base URL | <http://ec2-18-130-130-112.eu-west-2.compute.amazonaws.com:8080> |
| SAML Profiles | SP-Initiated SSO – True  SP-Initiated SLO – True |
| Identity Mapping | Enable No Account Mapping |
| Attribute Contract | SAML\_SUBJECT  address  birthDate  firstName  lastName  gender |
| SSO Service URL | POST - /saml/sso |
| SLO Service URL | POST - /saml/slo |
| Signature Verification | Selected Certificate – Import the certificate generated on IdP server and select it. |

## Authentication Policy

An authentication policy is a tree of authentication sources, selector instances, or a combination of them, on which the decision to route a request through a series of approved authentication sources with an optional authentication policy contract or a local identity profile at the end or to deny the request are based.

The Federation Hub has been configured to allow users to authenticate using one of the two mechanisms based on the user journey. During the registration process a new user will always be authenticated by SAML Identity Provider (simulating Verify) and if user has already created a basic DBI profile then they can authenticate using DBI credentials only. After successful authentication, user information from respective sources is used to fulfil an attribute contract which is further used to insert information in access token and id token sent to the OIDC Clients.

## OIDC Clients

The simulating applications are integrated with the Federation Hub as OIDC Clients. The client authentication is performed using the ‘Client Secret’ which has been shared and embedded in the client code. The access tokens granted by the PingFederate is of JSON Web Tokens type for these clients. The ID Token signing algorithm used is HMAC Using SHA-256. Both the applications have been granted access to Session Revocation API to be able to terminate user’s session from service provider itself.

## Run/Stop Commands

Following are the commands to run/stop PingFederate Server. Open an SSH terminal to AWS1 and run the below commands –

**Start Server** –

$ cd /usr/local/pingfederate-2/sbin/

$ ./pingfederate-run.sh

**Stop Server** –

$ cd /usr/local/pingfederate-2/sbin/

$ ./pingfederate-shutdown.sh

**Check Server Status** –

$ cd /usr/local/pingfederate-2/sbin/

$ ./pingfederate-status.sh

# Build & Deployment Instructions

## OIDC Applications

As part of MVP development, 2 distinct applications have been created. Both the applications are developed as Java Spring Boot Application and are deployed on AWS2. To run these application a script has been provided in the application installation folder. Run the following commands –

**Start BEIS Grants Application**

$ cd /u01/applications/BEISGrantsApplication

$ ./startApplication.sh

**Start Registration Application**

$ cd /u01/applications/RegistrationApplication

$ ./startApplication.sh

## Assurance Engine

The Assurance Engine has two core modules – AssuranceEngineProcessor and BEISAssuranceEngine. The AssuranceEngineProcessor has all the business processing code and serves as the backend. BEISAssuranceEngine is the front-end part of AE.

The Assurance Engine is a REST web service engine which runs as a Spring Boot application on AWS2 on port 9091.

**Build**

To build Assurance Engine code open an SSH terminal to AWS2 and run the commands below –

$ cd /u01/applications/AssuranceEngineMaster

$ mvn clean install

The above command will generate a BEISAssuranceEngine-0.0.1-SNAPSHOT.war file in the local maven repository.

**Run**

To start the AssuranceEngine Server, run the following commands –

$ cd /u01/applications/AssuranceEngineMaster/BEISAssuranceEngine

$ mvn spring-boot:run

## Encrypt Text using jasypt

Jasypt supplies a bunch of Command Line Interface (CLI) tools, useful for performing encryption, decryption and digest operations from the command line. In order to use this tools, you should download the distribution zip file (named jasypt-$VERSION-dist.zip) and unzip it. This tool also requires icu4j-3.4.4.jar to be added in the classpath before running any command. Once the utility is setup run the following command to encrypt a text value.

$ cd <jasypt-dist>/src/main/bin

$ ./encrypt.sh input=<Text to be encrypted> password=<Encryption Password> algorithm=PBEWithHMACSHA512AndAES\_256 ivGeneratorClassName=org.jasypt.iv.RandomIvGenerator

The output of the command will give you the encrypted value to be added in properties file.

The encrypted values should be added in properties file enclosed within “ENC()”. For e.g.

client.id=ENC(Encrypted Value here)

*This speculative document is made by KPMG LLP, a UK Limited Liability Partnership and a member firm of the KPMG network of independent firms affiliated with KPMG International Cooperative (“KPMG International”), a Swiss entity. The proposals set out in this document do not constitute an offer capable of acceptance. They are in all respects subject to satisfactory completion of KPMG's procedures to evaluate prospective clients and engagements, including independence and conflict checking procedures, and the negotiation, agreement, and signing of a specific engagement letter or contract. KPMG International provides no client services. No KPMG member firm has any authority to obligate or bind KPMG International or any other member firm vis-à-vis third parties, nor does KPMG International have any such authority to obligate or bind any member firm.*