**Connect.Gov – RP Integration – OpenID Connect - DRAFT**

Document Control Change Record

| Date | Revision | Section(s) | Editor | Change Reference |
| --- | --- | --- | --- | --- |
| 19/01/2015 | 1.0 | - | DH | Initial Version |
| 05/08/2015 | 1.1 | [Section 2.2](#_Session_Security_and) | MP | Corrected session security requirements (dropped SSL v3 from list). |
| [Section 3.1](#_RP_On-boarding)  [Section 3.4](#_Authentication_Endpoints) | Clarified specifications and corrected URLs. |
| [Appendix B.3](#C3_AuthenticationClaimRequest)  [Appendix B.4](#C4_AuthenticationClaimResponse) | Changed section from Authentication Claim/Response to Token Request/Response. |
| 26/08/2015 | 1.2 | [Section 2.1](#_OIDC_Protocol) | MP | Added external link to OAuth 2.0 Threat Model and Security Considerations. |
| [Section 3.2](#_CSP_Selection_Method) | Clarified use of authMode parameter and included possible values. |
| [Section 3.4](#_Authentication_Endpoints) | Updated terms to match those used in OpenID Connect spec. |
| [Appendix A.2](#Scopes) | New\* Added Connect.Gov scopes (attribute groupings). |
| [Appendix B](#LoA_ACR_VALUES) | New\* Added LoA/ACR Values. |
| 16/09/2015 | 1.4 | [Section 3.1](#_RP_On-boarding) | MP/JR/MV/DH | * Changed authentication method to private\_key\_jwt. * Added server (service) discovery location. * Clarified RP information requirements. |
| [Section 3.4](#_Authentication_Endpoints) | * Added descriptions for auth request/response parameters state, prompt, nonce, code, and authMode. * Added param requirements for token request: client\_assertion\_type and client\_assertion. * Clarified retrieve claims procedure. |
| [Appendix A.1](#A_FedAttributes) | * Added column for applicable “scope”. * Added email attribute. |
| [Appendix C.3](#C3_AuthenticationClaimRequest) | Modified example to reflect private\_key\_jwt authentication method. |
| [Appendix C.5](#C5_UserInfoRequest) | Clarified example for User Info Request. |
| [Appendix D](#F_ErrorCases) | Corrected examples of error responses. |

Table of Contents

[1. Introduction 5](#_Toc430180654)

[1.1 About Connect.Gov 5](#_Toc430180655)

[1.2 Audience 5](#_Toc430180656)

[1.3 Notation 5](#_Toc430180657)

[2. System Requirements 6](#_Toc430180658)

[2.1 OIDC Protocol 6](#_Toc430180659)

[2.2 Session Security and Sign-On 6](#_Toc430180660)

[2.3 Signature and Encryption Security 6](#_Toc430180661)

[2.3.1 Signatures 6](#_Toc430180662)

[2.3.2 Encryption (Optional) 7](#_Toc430180663)

[3. RP Integration Overview 8](#_Toc430180664)

[3.1 RP On-boarding 8](#_Toc430180665)

[3.2 CSP Selection Method 10](#_Toc430180666)

[3.2.1 Embedded Selector 10](#_Toc430180667)

[3.2.2 Full page selector 11](#_Toc430180668)

[3.3 User Consent 11](#_Toc430180669)

[3.3.1 Implicit Consent Page 11](#_Toc430180670)

[3.4 Authentication Flow and Endpoints 13](#_Toc430180671)

[3.5 Testing in Staging Environments 16](#_Toc430180672)

[Appendix A Federation Attributes 17](#_Toc430180673)

[A.1 Available Attributes in Connect.Gov 17](#_Toc430180674)

[A.2 Connect.Gov Scopes 21](#_Toc430180675)

[A.2.1 Group Attributes 21](#_Toc430180676)

[Appendix B Level of Assurance (ACR Values) 23](#_Toc430180677)

[Appendix C OIDC Protocols 24](#_Toc430180678)

[C.1 Authentication Request 24](#_Toc430180679)

[C.2 Authentication Response 25](#_Toc430180680)

[C.3 Token Request 26](#_Toc430180681)

[C.4 Token Response 26](#_Toc430180682)

[C.4.1 ID Token 27](#_Toc430180683)

[C.5 User Info Request 29](#_Toc430180684)

[C.6 User Info Response 29](#_Toc430180685)

[Appendix D Error Cases 30](#_Toc430180686)

[D.1 Invalid Authentication Request 30](#_Toc430180687)

[D.2 Invalid Authentication Claim Request 30](#_Toc430180688)

[Appendix E Clarification of MBUN and PAI 31](#_Toc430180689)

[E.1 MBUN Summary 31](#_Toc430180690)

[E.2 PAI Summary 31](#_Toc430180691)

[E.3 PAI Architecture 32](#_Toc430180692)

[Appendix F Acronyms 34](#_Toc430180693)

[Appendix G Glossary of Terms 35](#_Toc430180694)

[Appendix H References 36](#_Toc430180695)

# Introduction

## About Connect.Gov

Connect.Gov is a cloud-based broker for connecting large scale, consumer facing services with trusted identity and attribute providers. Connect.Gov delivers the following benefits:

* For consumers, Connect.Gov delivers a sign-on experience that is familiar, fast and simple.
* For online services, Connect.Gov removes the barriers of security, complexity and password fatigue to ensure online service success.
* For credential issuers, Connect.Gov provides new opportunities to leverage the success of their credentials platform and expand credential usage.

Connect.Gov is built upon strong privacy and security principles and enables a user-centric attribute consent model.

## Audience

This guide is intended for online service providers integrating with Connect.Gov as Relying Parties using the OpenID Connect protocol.

It is assumed the reader is generally familiar with the OpenID Connect (OIDC) v1.0 protocol.

## Notation

The following terms are used to describe entities in this document:

|  |  |
| --- | --- |
| Credential Service Provider (CSP) | Refers to an authentication provider in NIST’s Electronic Authentication Guidelines (special publication 800-63-2). This entity is also referred to as “Identity Provider”. |
| Relying Party (RP) | Refers to services (as used in NIST’s Electronic Authentication Guidelines). This entity is also referred to as “Service Provider”. |

# System Requirements

To perform integration with Connect.Gov, the following system and security requirements must be met:

## OIDC Protocol

An RP must support communication using the **Authorization Code** flow of the OpenID Connect v1.0 protocol. The RP must be able to consume and send OIDC messages to communicate with Connect.Gov.

* Please visit the OpenID website for an overview of OpenID Connect v1.0 protocol: <http://openid.net/specs/openid-connect-basic-1_0.html>.
* Please see the OAuth 2.0 Threat Model and Security Considerations document for additional security considerations: <http://tools.ietf.org/html/rfc6819>.

## Session Security and Sign-On

The RP initiates the sign-on process by passing an authentication request to Connect.Gov. The request is passed in the redirect to Connect.Gov. RPs have the option of redirecting the entire page or part of the page via inline frame (iframe).

For web flow options and user interface considerations, see [Section 3.2](#_Implement_UI_for).

During the authentication process, Connect.Gov and the CSP will display pages to the user that are accessible from the internet:

* TLS v1.1 (and higher) MUST be used to protect all protocol endpoints.
* The use ofTLS 1.2 is recommended.
* It is recommended that the TLS implementation conform to NIST SP 800-52 (also, refer to the Best Current Practices on the use of TLS here: <https://tools.ietf.org/html/bcp195>).

## Signature and Encryption Security

Signing and encryption keys must be shared between entities to ensure they can process OIDC messages and verify the sender signature.

For more details about exchanging keys, see [Section 3.1](#_RP_On-boarding).

### Signatures

OIDC ID Tokens and (optionally) User Info messages from Connect.Gov are digitally signed and must be validated by the receiver using established signature validation keys. Signed messages ensure message integrity, authentication of message origin, and non-repudiation of origin.

The following signing algorithms are supported by Connect.Gov:

* RS256
* RS384
* RS512
* HS256
* HS384
* HS512

### Encryption (Optional)

Encryption of OIDC messages is optional.

OIDC Token and User Info responses from Connect.Gov can be encrypted using approved encryption keys with Connect.Gov.

The following encryption key algorithm is supported by Connect.Gov:

* RSA1\_5

The following encryption methods are supported by Connect.Gov:

* A192CBC-HS384
* A256CBC-HS512
* A128CBC-HS256

# RP Integration Overview

Connect.Gov uses the OIDC v1.0 protocol to define message syntax and processing rules for communication between system entities. To integrate with Connect.Gov, Relying Parties (RPs) must be able to pass and process OIDC messages using the Authorization Code Flow.

Each communicating entity must have specific knowledge about the other such as identifiers, endpoints, keys (or locations of key sets), and so on. These details are shared by the Connect.Gov administrator. Furthermore, user interface components and user flows must be established with Connect.Gov to ensure an integrated user experience across the system.

## RP On-boarding

During the on-boarding phase, the Connect.Gov administrator should provide the RP with the following information:

**Note:** The host portion in the following URL examples has the placeholder {federation-host}. This value is provided by your Connect.Gov administrator.

* **Client ID**

This identifies the RP in the request sent to Connect.Gov.

* **Issuer URI**

This is the root URI of the Connect.Gov OIDC server. This value is returned in the ID Token and is used in server discovery.

Root URI Convention: https://{federation-host}/federation

The server configuration information, including all endpoint URIs, is available as a JSON document from a URL based on the Issuer URI above.

https://{federation-host}/federation/.well-known/openid-configuration

* **Authorization Endpoint**

This endpoint is set up to receive an OIDC authentication request.

Authorization Endpoint:

https://{federation-host}/federation/oidc/authorization

* **Token Endpoint**

This endpoint is set up to exchange the Authorization Code for an Access Token and ID Token. The client authenticates using the **private\_key\_jwt** authentication method.

Token Endpoint: https://{federation-host}/federation/oidc/token

* **User Info Endpoint**

This endpoint is set up to accept an Access Token and return the corresponding user attributes (if applicable).

UserInfo Endpoint: https://{federation-host}/federation/oidc/userinfo

* **JSON Web Key Set (JWKS) Endpoint**

This endpoint contains the necessary keys for the RP to verify signed inbound messages such as signed ID tokens.

JWKS URI: https://{federation-host}/federation/oidc/jwks

An example JSON Web Key Set for an RSA public key follows:

{

"keys": [

{

"e": "AQAB",

"n": "rMbjxLvpzLh6s3Pdb623H7TRYms2wR8TRU3Le64V4\_Qhbboq6JbWNR\_UAnvVd-PPKo2bH\_a2eahamfLocMpAmTwzfbENOXsfosrx3xJyLgqwWHq3ygH-nGOKvuozu3ryX8Y-UfCrb4YusS55lMg5VMk3TalkRRMoXp9oOoOr6-y4V\_Kju4PVpeWybmTlcGEbn8Qo3QmYI4Nq5tyjKSiNLr4Hmv0Sygz6Pu0FCw5tkzl55UZGZm2JASEaowGZ6K5fOaBwbhMe9NQ-B5rWJIuQgbGCeHMC989fS5XcvLnM2XJryTZ7CacIihgEoOjsSdAlaMIHSlMu-jcs646TfDcNWw",

"kty": "RSA",

"kid": "connect-gov-rsa-1"

}

]

}

The RP must provide the Connect.Gov administrator with the following information:

* **RP Display Name**

The display name appears on user screens where the requesting entity (that is, the RP) is identified to the end user.

* **JSON Web Key Set Endpoint**

This endpoint is published by the RP and contains the necessary keys for the RP to sign messages to Connect.Gov and for Connect.Gov to encrypt outbound messages to the RP. Since the RP must use the **private\_key\_jwt** authentication method, this is required.

* **Redirection URI(s)**

These are a list of valid endpoints where Connect.Gov will redirect the authorization response. These URIs must be the exact full string used during the authentication request. The authentication request must specify which valid URI to redirect the response to, which will be compared using the exact string matching against the registered redirect URIs.

* **Default CSP Selection Method (Optional)**

If RPs want to specify a default CSP selection method, they can configure a default selection with Connect.Gov. This value will be used when no authMode parameter is passed with the authentication request.

## CSP Selection Method

In addition to the standard parameters defined in OIDC, Connect.Gov supports three modes for the authentication selector through use of an optional extension parameter:

* Pre-selected CSP\*
* Full Page Selector, and
* Embedded Selector

\*For more information about creating a custom CSP selector using the pre-selected option, refer to the Custom CSP Selector Guide.

The CSP selection method is specified by the authMode extension query parameter in the authentication request. If the authMode parameter is not passed, the default selection method – configured for the RP with Connect.Gov during on-boarding – will be used.

If no default is configured, the default is the Full Page selector.

The possible values for authMode are:

* widget – indicating the embedded selector.
* classic – indicating the full page selector.

### Embedded Selector

RPs can pass control to Connect.Gov via iframe to achieve a more integrated user experience during sign-on. RPs can customize the iframe style and dimensions to their own specifications.

We recommend that iframe dimensions, at minimum, have a **width of 490px** and a **height of 190px**.

The following shows a page on the RP’s web page with an embedded Sign-in Partners selector.

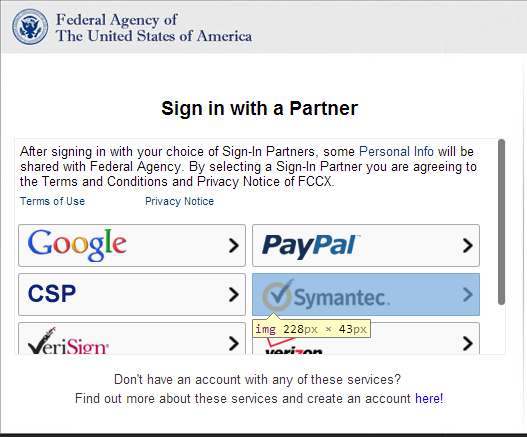


Figure 3: Example of Embedded Selector

Following the selection of a CSP, the iframe flow will end and the selected CSP will acquire the full page once the appropriate authentication request is made.

**Note:** The necessary OIDC (and cookie domain) redirects are handled within the iframe, per specification. An additional query parameter must be passed in the redirect (with the authentication request) to indicate use of embedded selector, OR a request can be made to Connect.Gov to use the embedded selector *by default* if the RP cannot support query parameters.

### Full page selector

The full page selector follows the standard practice of redirecting the browser page to Connect.Gov for authentication. An OIDC authentication request is included in the redirect. Once the request is received and verified, Connect.Gov presents the CSP selector using the full browser page.

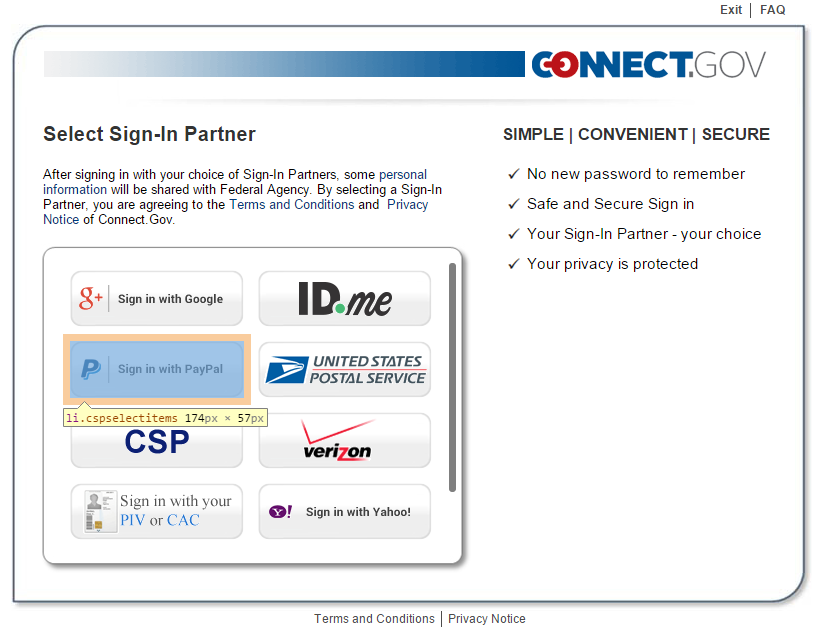


Figure 4: Example of Full Page Selector

## User Consent

Authentications where user attributes are requested must include pages to facilitate user consent for release of personal information. Connect.Gov supports the following method for facilitating user consent.

### Implicit Consent Page

Implicit consent pages inform the user which attributes are being requested by the RP. The RP name is shown on the page, as defined during RP On-Boarding.

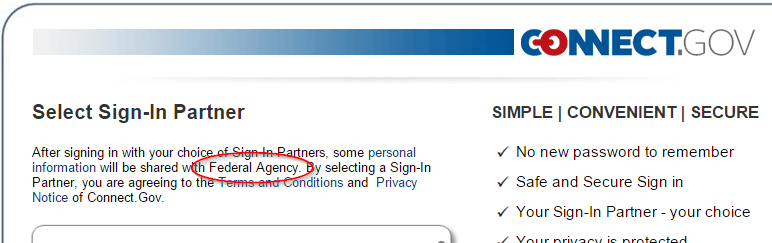


Figure 5: RP Organization Name on Consent Page

One the selector page, the user can click the “personal information” link to see a detailed list of the attributes requested (by scope) by the RP.



Figure 6: Personal Information Link

Connect.Gov indicates the required attributes for this request and notes the requesting RP.

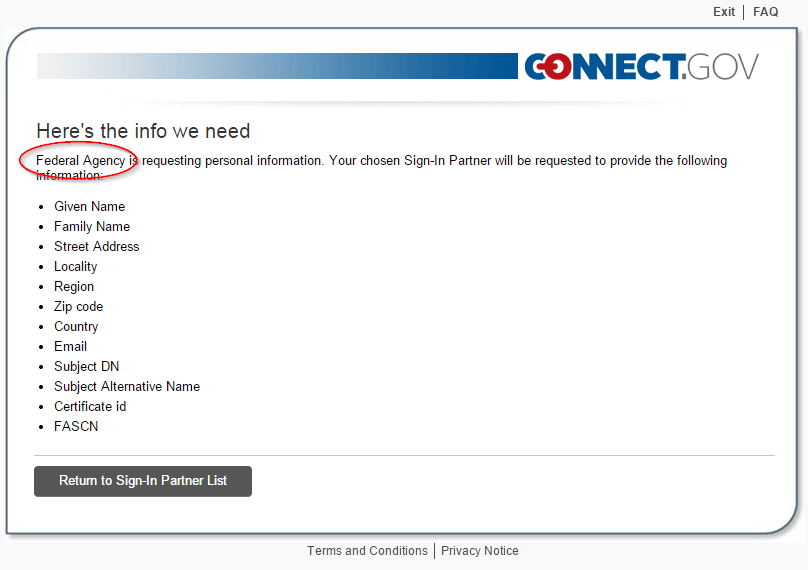


Figure 7: Implicit Consent Page

## Authentication Flow and Endpoints

Relying Parties (RPs) must implement several handlers for passing and processing OIDC messages to ensure trusted interoperability in the Connect.Gov environment. In addition, RPs must make direct requests to Connect.Gov to retrieve claims (attributes).

The following sequence diagram illustrates the authentication flow for RPs, with an emphasis on integration points necessary for the RP to integrate with Connect.Gov:

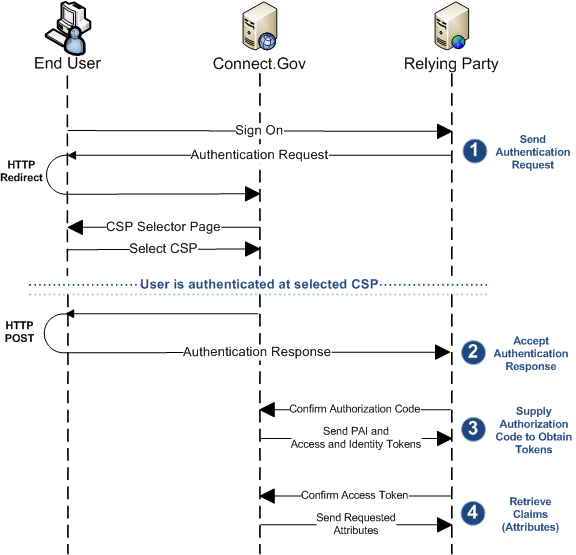


Figure 8: RP Sequence Diagram

Additional explanations for each step are provided below:

1. **Send Authentication Request**

When the end user indicates they want to sign on, the RP sends an authentication request to Connect.Gov. For Connect.Gov, the following values should be specified:

|  |  |
| --- | --- |
| responseType | [required] must be set to code. |
| acr\_values | [required] lists the acceptable LoAs (see [Appendix B](#LoA_ACR_VALUES)). |
| scope | [required] indicates the attributes being requested (see [Appendix A.2](#Scopes)). |
| redirect\_uri | [required] indicates a valid endpoint where the RP will receive the authentication response. |
| state | [required] unguessable random string generated by the RP, used to protect against CSRF attacks. Must contain a sufficient amount of entropy to avoid guessing. Returned to the RP in the authentication response. |
| prompt | [required] This value must be set to select\_account. Connect.Gov will prompt the user to select their preferred CSP. |
| nonce | [optional] unguessable random string generated by the RP, used to protect against CSRF attacks. Must contain a sufficient amount of entropy to avoid guessing. Returned to the RP in the ID Token. |
| authMode | [optional] specifies the CSP selection method (see [Section 3.2](#_Implement_UI_for)). Values include:   * classic to specify the full page selector * widget to specify the embedded selector |

The endpoint for this request is:

https://{federation host}/federation/oidc/authorization

For an example, see [Appendix C.1: Authentication Request](#C1_AuthenticationRequest).

1. **Accept Authentication Response**

The authentication response is sent via HTTP redirect to the redirect URI specified in the request. For Connect.Gov, this URI contains the parameters:

* state *required*, the value of the state parameter passed in by the RP in the authentication request. This value must match exactly.
* code *required,* the authorization code, a random string issued by the IdP to be used in the request to the token endpoint.

For an example, see [Appendix C.2: Authentication Response](#C2_AuthenticationResponse).

1. **Supply Authorization Code to Obtain Tokens**

To retrieve the access and ID tokens, the RP sends a token request to Connect.Gov’s token endpoint using HTTP POST with parameters in the form encoded body. For Connect.Gov, the following values are specified:

|  |  |
| --- | --- |
| grant\_type | [required] must be set to [authorization\_code]. |
| code | [required] the value of the code parameter returned in the authorization response. |
| client\_assertion\_type | [required] must be set to urn:ietf:params:oauth:client-assertion-type:jwt-bearer |
| client\_assertion | [required] the value of the signed client authentication JWT generated as described below. The RP must generate a new assertion JWT for each call to the token endpoint. |

To authenticate, the RP generates a JSON Web Token with the following *required* claims:

* iss the client ID of the RP
* sub the client ID of the RP
* aud the URL of the token endpoint.
* jti an unguessable random string generated by the RP.
* exp an integer timestamp (in Unix Epoch format) of the expiration of this assertion. This should be a very small time in the future, such as five minutes from issuance.

The client signs this JWT with its own private key and sends it to the server as described above in the client\_assertion field. Since the client has registered its public key, the server will be able to verify the signature of the JWT and therefore authenticate the client.

The endpoint for this request is:

https://{federation host}/federation/oidc/token

For an example, see [Appendix C.3: Token Request](#C3_AuthenticationClaimRequest).

The token response includes an access token (which can be used to make a UserInfo request) and ID token (a signed and optionally encrypted JSON Web Token). Of note, ID token values in Connect.Gov have the following meanings:

* iss indicates the issuer URL of the federation host.
* sub indicates the [PAI](#G_MBUNandPAI) for the user
* acr indicates the LoA the user was authenticated at.

1. **Retrieve Claims (Attributes) – Optional**

To retrieve claims (attributes), the RP sends a user info request to Connect.Gov as an HTTP GET or an HTTP POST with the access token passed in the HTTP Authorization header as an OAuth bearer token.

The endpoint for this request:

https://{federation-host}/federation/oidc/userinfo

The response from this request is a JSON document containing the attributes for the user. The attributes may alternatively be returned in a signed or encrypted JWT, depending on configuration.

For an example, see [Appendix C.5: User Info Request](#C5_UserInfoRequest).

## Testing in Staging Environments

Connect.Gov will provide a test package and the RP is expected to satisfy a number of testing requirements to deploy to a production environment. There are three phases to testing:

* **Integration Testing**

This phase involves testing each component of the overall interface, including:

* + Public key locations
  + Sending Authentication Requests to Connect.Gov
  + Receiving Authentication Responses
  + Requesting/Handling Authentication Claims with appropriate LoA and attributes
  + Requesting/Handling User Info
  + Error Handling and Response
* **System Integration (End-to-End) Testing**

This phase involves testing the entire process from end to end, including:

* + Round trip testing of successful use case as illustrated in [Section 3.4](#_Authentication_Flow_and).
  + Testing of all error messages and error scenarios.

See [Appendix D: Error Cases](#F_ErrorCases) for a non-exhaustive list of error scenarios.

* **Customer Acceptance Testing**

Final acceptance testing is done in this phase, and involves authenticating with CSPs during the execution of test cases.

As part of the integration project, your Connect.Gov representative will provide a Connect.Gov Test Package which contains details on how to access the test environments, as well as Test Cases and instruction on how to use the Connect.Gov Test Console.

**Note:** The RP is responsible for all testing.

1. Federation Attributes
   1. Available Attributes in Connect.Gov

The following table outlines the attributes that are accessible to RPs in Connect.Gov.

| Attribute Name | Friendly Name | Description | Scope |
| --- | --- | --- | --- |
| sub | Subject | Unique identifier for this user (PAI). | openid |
| given\_name | First Name | Given name(s) or first name(s) of End User.  Format: String  Limit: 100  Example: John | profile |
| family\_name | Family Name | Surname(s) or last name(s) of End User.  Format: String  Limit: 100  Example: Smith | profile |
| middle\_name | Middle Name | Middle name of End User.  Format: String  Limit: 100  Example: Carl | profile |
| middle\_initial | Middle Initial | Initial(s) of middle name of End User.  Format: String  Limit: 10  Example: C | profile |
| birthdate | Date of Birth | Date of birth of End User.  Format: String (YYYY-MM-DD)  Limit: 100  Example: 1985-04-18 | profile |
| email | Email address | Email address of the end user  Format: String (user@domain)  Limit: 100  Example: Alice@example.org | Email |
| ssn9 | Social Security Number | 9 digit social security number of End User.  Format: String (AAA-GG-SSSS)  Limit: 100  Example: 100-09-555 | ssn |
| country | Country | Country name component about End User.  Format: String  Limit: 50  Example: US | address |
| locality | Locality | City or locality component about End User.  Format: String  Limit: 100  Example: San Diego | address |
| region | Region | State, province, prefecture or region component about End User.  Format: String  Limit: 100  Example: California | address |
| postal\_code | Zip code | Zip code or postal code component about End User.  Format: String  Limit: 10  Example: 92123-4020 OR 92123 | address |
| street\_address | Street Address | Current street address of End User.  Format: String  Limit: 100  Example: 123 Myhouse Rd. | address |
| subject\_dn | Certificate Owner Name | The Subject Distinguished Name from an X.509 Certificate.  **Note:** Only applicable for FPKI Credentials (derived from PIV card).  Format: String  Mandatory: No  Limit: 200  Example: John Smith C | piv |
| subject\_alternative\_name | Certificate Owner Alias | Alias for an X.509 Certificate Owner.  **Note:** Only applicable for FPKI Credentials (derived from PIV card).  Format: String  Mandatory: No  Limit: 200  Example: John Smith | piv |
| uuid | Certificate Unique Identifier | The Universally Unique Identifier from a PIV-I Authentication Certification.  **Note:** Only applicable for FPKI Credentials (derived from PIV card).  Format: String  Mandatory: No  Limit: 100  Example: c3279704-8aaf-11e3-ae4d-d231feb1dc81 | piv |
| fasc\_n | Federal Certificate Unique Identifiers | The Federal Agency Smart Credential Number from a PIV Authentication Certificate.  **Note:** Only applicable for FPKI Credentials (derived from PIV card).  Format: String  Mandatory: No  Limit: 50  Example: e7c0ffec-8aaf-11e3-ae4d-d231feb1dc81 | piv |

* 1. Connect.Gov Scopes

User claims (attributes) can be requested in the authentication request using the scope parameter. The scope name represents a collection of claims. In Connect.Gov, the scope name maps to a configured attribute expression.

The following table lists the scopes for Connect.Gov and their associated attribute expressions.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Group 1**  **(Per. Info)** | **Group 3 or Group 9** | | **Group 3**  **(Email)** | **Group 4**  **(USDA)** | **Group 5**  **(VA)** | **Group 4 or**  **Group 9** | **Group 5**  **or**  **Group 9** | **Group 8** | **Group 9**  **PIV** |
| **Supported LOA:** | **LOA 1+** | | **LOA 1+** | **LOA 1+** | **LOA 2+** | **LOA 2+** | **LOA 2+** | | **LOA 1+** | **LOA 1+** |
| **Scope\*** | scope\_1 | scope\_2 | | scope\_3 | scope\_4 | scope\_5 | scope\_6 | scope\_7 | scope\_8 | scope\_9 |
| **Attribute Expression\*\*** | **A** or **F** | **C** or **F** | | **C** | **D** | **E** | **D** or **F** | **E** or **F** | **G** | **F** |

\* Scope – The possible values for scope parameter sent in the Authentication Request.

\*\* Attribute Expression – Expression representing requested attributes or combination of groups.

* + 1. Group Attributes

The following tables outline the required (**R**) and optional (**O**) attributes per grouping.

|  | **Group 1**  **(Per. Info)** | **Group 3 or Group 9** | **Group 3**  **(Email)** | **Group 4**  **(USDA)** | **Group 5**  **(VA)** | **Group 4 or**  **Group 9** | **Group 5**  **or**  **Group 9** | **Group 8** | **Group 9**  **PIV** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **First Name** | O |  |  | R | R |  |  | O |  |
| **Last Name** | O |  |  | R | R |  |  | O |  |
| **Middle Name** |  |  |  | O | O |  |  |  |  |
| **Middle Initial** |  |  |  | O |  |  |  |  |  |
| **Email** | O |  | R | O | O |  |  | R |  |
| **Street** | O |  |  | R | R |  |  |  |  |
| **City** | O |  |  | R | R |  |  |  |  |
| **State** | O |  |  | R | R |  |  |  |  |
| **Zip**  **(XXXXX-XXX OR XXXXX)** | O |  |  | R | R |  |  |  |  |
| **Country** | O |  |  | O | O |  |  |  |  |
| **DOB**  **(YYYY-MM-DD)** |  |  |  |  | R |  |  |  |  |
| **SSN9**  **(AAA-GG-SSSS)** |  |  |  |  | R |  |  |  |  |
| **SSN4** |  |  |  |  |  |  |  |  |  |
| **Subject DN** |  |  |  |  |  |  |  |  | R |
| **Subject Alternate Name** |  |  |  |  |  |  |  |  | R |
| **UUID** |  |  |  |  |  |  |  |  | O |
| **FASC-N** |  |  |  |  |  |  |  |  | O |

1. Level of Assurance (ACR Values)

Authentication requests identify the level of assurance (LoA) required using the   
acr\_values (Authentication Context Class Reference) parameter. Multiple values may be provided in the request. The ID token response will specify the actual acr value used to perform authentication.

The identifying URIs for Connect.Gov levels of assurance (acr\_values) are:

* http://idmanagement.gov/icam/2009/12/saml\_2.0\_profile/assurancelevel1
* http://idmanagement.gov/icam/2009/12/saml\_2.0\_profile/assurancelevel2
* http://idmanagement.gov/icam/2009/12/saml\_2.0\_profile/assurancelevel3
* http://idmanagement.gov/icam/2009/12/saml\_2.0\_profile/assurancelevel4

1. OIDC Protocols
   1. Authentication Request

Authentication requests outline a set of properties and conditions that must be met to authenticate an end user.

Details on Authentication Request Protocol can be found in “OpenID Connect Basic Client Implementer’s Guide” (OpenID Basic Client Implementation Guide, 2014), Section 2.1.1.

The following is an example of an authentication request:

1. https://Connect.Gov.fed1.connect.gov/federation/oidc/authorization?
2. **response\_type**=code
3. &**client\_id**=827937609728-m2mvqffo9bsefh4di90saus4n0diar2h
4. &**scope**=d+openid
5. &**redirect\_uri**=https%3A%2F%2Fint-rp.fed1.connect.gov%2Frp1%2F

web%2Foidc%2FloginResponse

1. &**state**=2ca3359dfbfd0
2. &**prompt**=select\_account
3. &**acr\_values**=http%3A%2F%2Fidmanagement.gov%2Fns%2Fassurance%2Floa%2F1

+http%3A%2F%2Fidmanagement.gov%2Fns%2Fassurance%2Floa%2F2

+http%3A%2F%2Fidmanagement.gov%2Fns%2Fassurance%2Floa%2F3

+http%3A%2F%2Fidmanagement.gov%2Fns%2Fassurance%2Floa%2F4+

| No. | Request Properties | Description |
| --- | --- | --- |
| 1. | location | The authorization endpoint as set up by the Connect.Gov administrator. An RP passes the authentication request to this location. |
| 2. | response\_type | This value must be set to code to indicate use of the OIDC Authorization flow. |
| 3. | client\_id | This value is a unique RP identifier assigned to the RP by the Connect.Gov administrator. |
| 4. | scope | This value must include openid to indicate the use of the OpenID protocol.  An RP can also specify an attribute group index to request user attributes. The attribute group index is assigned by the Connect.Gov administrator. |
| 5. | redirect\_uri | This value is the redirection URI where Connect.Gov will send the authentication response. This value must match the URI provided to the Connect.Gov administrator during RP onboarding. |
| 6. | state | This value is a unique RP supplied value used to maintain state between the request and response. |
| 7. | prompt | This value must be set to select\_account. Connect.Gov will prompt the user to select their preferred CSP. |
| 8. | acr\_values | This value indicates the acceptable LoA(s) the user can be authenticated at.  In the case where the RP specifies **multiple LoAs** in the request, the CSP must honor specified order and meet the first LoA that they can support. |
| - | authMode | (Optional) This value specifies the CSP selection method (see [Section 3.2](#_Implement_UI_for)). |
| - | nonce | (Optional) This value is used to associate the session with the ID Token. The nonce will be included in the ID Token. |

* 1. Authentication Response

Authentication responses, if successful, are composed of assertions that satisfy the authentication request or, if unsuccessful, include appropriate status codes to indicate the request failed.

Details on Authentication Response Protocol can be found in “OpenID Connect Basic Client Implementer’s Guide” (OpenID Basic Client Implementation Guide, 2014), Section 2.1.5.

The following example is a response to the authentication request example from the previous section.

1. https://int-rp.fed1.connect.gov/rp1/web/oidc/loginResponse?
2. **state**=2ca3359dfbfd0
3. &**code**=gOIFJ1hV6Rb1sxUdFhZGACWwR1sMhYbJJcQbVJN0wHA

| No. | Response Properties | Description |
| --- | --- | --- |
| 1. | location | The redirect URI as specified by the RP in the request and during RP onboarding. |
| 2. | state | This value is echoed back from the value provided in the authentication request. |
| 3. | code | This value is the Authorization Code that the RP can use to retrieve the authentication claim. |

* 1. Token Request

Details on Token Request can be found in “OpenID Connect Basic Client Implementer’s Guide” (OpenID Basic Client Implementation Guide, 2014), Section 2.1.6. The following example shows authentication using the method **private\_key\_jwt**.

1. **grant\_type**=authorization\_code&
2. **code**=gOIFJ1hV6Rb1sxUdFhZGACWwR1sMhYbJJcQbVJN0wHA&
3. **redirect\_uri**= https%3A%2F%2Fint-rp.fed1.connect.gov%2Frp1%2Fweb%2Foidc%2FloginResponse&
4. **client\_id**=827937609728-m2mvqffo9bsefh4di90saus4n0diar2h&
5. **client\_assertion\_type**=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer&
6. **client\_assertion**= eyJhbGciOiJSUzI1NiIsImtpZCI6IjIyIn0.

eyJpc3Mi[...omitted for brevity...].

cC4hiUPo[...omitted for brevity...]

| No. | Request Properties | Description |
| --- | --- | --- |
| 1. | grant\_type | This value is set to authorization\_code. |
| 2. | code | This value is the Authorization Code. Use this value to retrieve the authentication claim from the Token Endpoint. |
| 3. | redirect\_uri | This value is the redirection URI where Connect.Gov will send the authentication claim (identity token and access token). This value must match the URI provided to the Connect.Gov administrator during RP onboarding. |
| 4. | client\_id | This value is a unique RP identifier assigned to the RP by the Connect.Gov administrator. |
| 5. | client\_assertion\_type | This value is set to urn:ietf:params:oauth:client-assertion-type:jwt-bearer |
| 6. | client\_assertion | A JWT generated by the RP and signed by the RP’s private key. |

* 1. Token Response

Details on the Token Response can be found in “OpenID Connect Basic Client Implementer’s Guide” (OpenID Basic Client Implementation Guide, 2014), Section 2.2.

{

1. **"expires\_in"** : 360,
2. **"token\_type"** : "Bearer",
3. **"id\_token"** : "eyJhbGciOiJSU0ExXzUiLCJjdHkiOiJKV1QiLCJlbmMiOiJBMTkyQ...",
4. **"access\_token"** : "gyfmzoE3T7P5Dck0oLe-9fvLYAQ"

}

| No. | Response Properties | Description |
| --- | --- | --- |
| 1. | expires\_in | The time in UTC seconds (epoch seconds) format that indicates when the access token expires. |
| 2. | token\_type | This value is set to Bearer. |
| 3. | id\_token | This value is a signed (and optionally encrypted) JSON Web Token (JWT) containing details about the authentication claim. |
| 4. | access\_token | This value is used to retrieve any attributes from the Connect.Gov User Info Endpoint, if requested. |

* + 1. ID Token

The ID Token is a signed (and optionally encrypted) JWT that contains details about the authentication event. Its payload is a JSON object with the following structure:

{

1. **"exp"** : 1406829176,
2. **"sub"** : "1e6032d7-7e14-486e-b494-4ed50cf8caeb",
3. **"iss"** : "https://int.fed1.connect.gov/federation",
4. **"aud"** : ["827937609728-m2mvqffo9bsefh4di90saus4n0diar2h"],
5. **"iat"** : 1406828576,
6. **"acr"** : http://idmanagement.gov/icam/2009/12/saml\_2.0\_profile/assurancelevel1,
7. **"nonce" :** "hrTs64K0"
8. **"jti" :** "de305d54-75b4-431b-adb2-eb6b9e546014"

}

| No. | Response Properties | Description |
| --- | --- | --- |
| 1. | exp | The time in UTC seconds (epoch seconds) format that indicates when the authentication claim is set to expire. After the expiration time, the claim must not be accepted for processing. |
| 2. | sub | The PAI value of the end user issued by Connect.Gov. |
| 3. | iss | The identifier of the response issuer. The value is a case-sensitive URL using the HTTPS scheme. |
| 4. | aud | The client identifier the authentication claim is intended for. |
| 5. | iat | The time in UTC seconds (epoch seconds) format for when the JWT was issued. |
| 6. | acr | The LoA at which the user was authenticated at. |
| 7. | nonce | The nonce value that was provided in the authentication request. Included if provided in authentication request. |
| 8. | jti | A unique identifier for the token, which can be used to prevent reuse of the token. |

* 1. User Info Request

Details on UserInfo Request can be found in “OpenID Connect Basic Client Implementer’s Guide” (OpenID Basic Client Implementation Guide, 2014), Section 2.3.

1. GET **/federation/oidc/userinfo** HTTP/1.1
2. Host: **int.fed1.connect.gov**
3. Authorization: Bearer **gyfmzoE3T7P5Dck0oLe-9fvLYAQ**

| No. | Request Properties | Description |
| --- | --- | --- |
| 1. | location | The User Info Endpoint as set up by the Connect.Gov administrator. An RP passes the access token to this location. |
| 2. | host | The federation endpoint as set up by the Connect.Gov administrator. |
| 3. | access\_token | The access token as returned by Connect.Gov when retrieving the authentication claim. |

* 1. User Info Response

The UserInfo response is a JSON object with attributes about the user. This may also be returned as a signed or encrypted JWT with the user attributes in the JWT payload, depending on configuration.

{

1. **"sub"** : "9c9bd6bb-56f8-44e5-bdea-92fe96f7e3e8",
2. **"iss"** : "https:\/\/int.fed1.connect.gov/federation"
3. **"given\_name":** **"John",**

**"family\_name": "Doe"**

/\* Requested *attributes* consented by end user \*/

}

| No. | Response Properties | Description |
| --- | --- | --- |
| 1. | sub | The PAI value of the end user issued by Connect.Gov |
| 2. | iss | The identifier of the response issuer. The value is a case-sensitive URL using the HTTPS scheme. |
| 3. | *attributes* | The set of attributes as requested by the RP. See [Appendix A: Federation Attributes](#A_FedAttributes) for attribute names and descriptions. |

1. Error Cases
   1. Invalid Authentication Request

The following scenarios result in an error page displayed to the user. There is no response returned to the RP in these events:

* Invalid client identifier, client\_id, specified in the authentication request.
* Invalid redirect URI, redirect\_uri, specified in the authentication request.

The following scenarios result in an error message returned to the RP by passing parameters to the RP’s redirect URI with a machine readable error code and a human readable error description (which may vary depending on circumstances). In these cases, RPs are expected to handle the error gracefully, as Connect.Gov does not display an error page to the user:

* Invalid response type, response\_type, specified in the authentication request.

?error=unsupported\_response\_type&error\_description=Unsupported%20response%20type%3A%20token

* Invalid scope, scope, specified in the authentication request.

?error=invalid scope&error\_description=Unsupported%20scope%3A%20batman

* 1. Invalid Authentication Claim Request

The following scenarios result in an error message returned to the RP as JSON objects with a machine-readable error code and a human readable error\_description (which may vary depending on circumstances). In these cases, RPs are expected to handle the error gracefully, as Connect.Gov does not display an error page to the user:

* Invalid client identifier, client\_id, or invalid signed JWT specified in the authentication claim (token) request.

{ "error": "invalid\_client",

"error\_description" : "Unkown client identifier" }

* Invalid request URI, redirect\_uri, specified in the authentication claim (token) request.

{ "error": "invalid\_request",

"error\_description": "Invalid redirect URI" }

* Invalid authorization code, code, specified in the authentication claim (token) request.

{ "error": "unsupported\_grant\_type",

"error\_description": "Unsupported grant type: password" }

1. Clarification of MBUN and PAI

Each entity in the system (outside the CSP) must be able to uniquely identify users without compromising user privacy or data integrity. Connect.Gov uses the concepts of MBUN and PAI to anonymously pass persistent user identifiers across the system to enable entities to identify unique users without compromising privacy. There are multiple PAI types in the Connect.Gov model. The term MBUN is used to identify the PAI passed by the CSP to Connect.Gov.

The following is a technical explanation of MBUNs and PAIs in the Connect.Gov system.

* 1. MBUN Summary
* Meaningless But Unique Number (MBUN) which is unique per user.
* Term used to describe the PAI value provided by CSPs as the Subject of the Authentication Claim.
* Format of MBUN determined by CSP and at most 200 characters long (alphanumeric string).
  1. PAI Summary
* Persistent Anonymous Identifier (PAI) which is unique per user.
* Term used to describe the persistent user identifiers generated by Connect.Gov for RPs in the authentication claim (ID Token).

{

"exp" : 1406829176,

**"sub" : "1e6032d7-7e14-486e-b494-4ed50cf8caeb",**

"iss" : "https://int.fed1.connect.gov/federation",

"aud" : ["827937609728-m2mvqffo9bsefh4di90saus4n0diar2h"],

"iat" : 1406828576,

"acr" : http://idmanagement.gov/icam/2009/12/saml\_2.0\_profile/assurancelevel1

}

* PAIs are NOT derived from MBUNs.
* MBUNs and PAIs are stored together in the Connect.Gov database:
  + Identity Broker Database stores: MBUN and iPAI (Connect.Gov internal PAI) pair
  + Federation Manager Database stores: iPAI and RPPAI (RP-specific PAI) pair

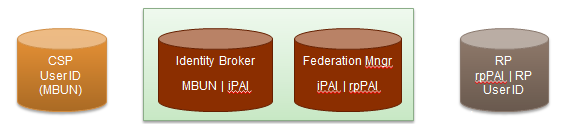


Figure 9: PAI Storage

* 1. PAI Architecture

The following diagram illustrates how the PAI value is generated in Connect.Gov:

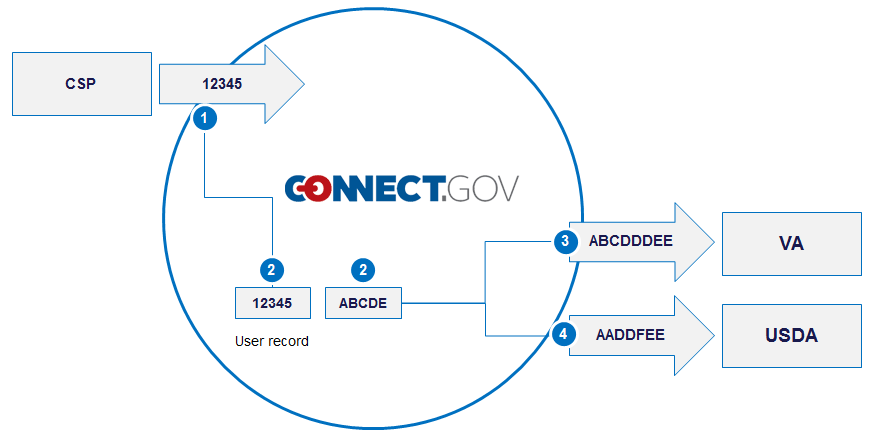


Figure 10: PAI Architecture

The above diagram shows the PAI values as they are generated at each stage of the authentication.

**Note:** The PAI values depicted in the diagram are simplified to illustrate their differences at each stage.

1. The CSP sends an MBUN to Connect.Gov in the authentication response of a successful user authentication.
2. Connect.Gov stores the MBUN and separately generates an internal PAI value, which is then sent to the federation manager.
3. The federation manager generates separate RP-specific PAI values for each RP in the authentication response. For example:
   * For the first RP, **VA**, the response can be:

{

"exp" : 1406829176,

**"sub" : "sdf223ce-14123aasdfsad-962dfdf-13ffsdsdfs22342",**

"iss" : "https:\/\/int.fed1.connect.gov/federation",

"aud" : ["client-id-for-VA"],

"iat" : 1406828576,

"acr" : "http:\/\/idmanagement.gov\/ns\/assurance\/loa\/1"

}

* + And for the second RP, **USDA**, the response can be:

**Note:** sub (the PAI) is different for both USDA and VA although the CSP authenticated only one user.

{

"exp" : 1406829176,

**"sub" : "124gsdf8ce-21312fsdf-fsd268-vd232-123fsdfs15as",**

"iss" : "https:\/\/int.fed1.connect.gov/federation",

"aud" : ["client-id-for-USDA"],

"iat" : 1406828576,

"acr" : "http:\/\/idmanagement.gov\/ns\/assurance\/loa\/1"

}

**Note the following:**

* The CSP PAI (MBUN) is not passed directly from the CSP to the RP.
* Connect.Gov generates and passes an internal PAI, a **new** anonymous identifier and not derived from the MBUN.
* The CSP PAI (MBUN) is saved in the Connect.Gov database.
* Each RP receives a consistent PAI to uniquely identify users; however, **no two RPs receive the same PAI**.

1. Acronyms

|  |  |
| --- | --- |
| Acronyms | Definition |
| ACI | Anonymous Credential Indicator |
| CSP | Credential Service Provider |
| FTP | File Transfer Protocol |
| UI | User Interface |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol Secure |
| IdP | Identity Provider (also known as CSP) |
| LoA | Level of Assurance |
| MBUN | Meaningless But Unique Number |
| NIST | National Institute of Standards and Technology |
| OIDC | OpenID Connect Protocol |
| PAI | Persistent Anonymous Identifier |
| PIV | Personal Identity Verification |
| PIV-I | Personal Identity Verification – Interoperability |
| RP | Relying Party |
| SFTP | Secure File Transfer Protocol |
| SLO | Single Logout |
| SP | Service Provider |
| SSO | Single Sign On |

1. Glossary of Terms

|  |  |
| --- | --- |
| Term | Definition |
| ACI | Anonymous Credential Identifier. Anonymously identifies the credential used to authenticate the user. In Exchange, the federation assigns a unique ACI to each [credential, PAI] pair. |
| Connect.Gov | An identity broker service platform. |
| Embedded selector | The embedded selector allows the RP to show the CSP selector as an embedded IFrame within the RP’s web page |
| Connect.Gov Federation | A specific deployment of Connect.Gov federation. Client specific federation identity ecosystem e.g.: Canadian Government, US Government. |
| Full page selector | The full page selector follows the standard practice of redirecting the browser page to the CSP (federation) for authentication. |
| CSP | Credential Service Provider. A party that issues and verifies credentials. |
| PAI | Persistent Anonymous Identifier. A unique identifier assigned by the federation to diversify a user to an RP. RP's use this as the subject |
| RP | Relying Party. The party making an authentication or attribute request to Exchange. This is not the end user but the organization that is providing the end user services. |

1. References

BCP 195. (2015, May). *Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS).* Retrieved from https://tool.ietf.org/html/bcp195

OpenID Basic Client Implementation Guide. (2014, August 27). *OpenID Connect Basic Client Implementer's Guide 1.0 - draft 33.* Retrieved from http://openid.net/specs/openid-connect-basic-1\_0.html

OpenID Connect. (2014, November 8). *OpenID Connect Core 1.0.* Retrieved from http://openid.net/specs/openid-connect-core-1\_0.html

RFC 1766. (1995, March). *Tags for the Identification of Languages.* Retrieved from http://www.ietf.org/rfc/rfc1766.txt

RFC 6749. (2012, November). *The OAuth 2.0 Authorization Framework.* Retrieved from https://tool.ietf.org/html/rfc6749

RFC 7519. (2015, May). *JSON Web Token (JWT).* Retrieved from https://tools.ietf.org/html/rfc7519