Electronic Consent Based SSN Verification (eCBSV) Service

Technical Information Guide

Version 3.6

Date: May 28, 2026

1 TABLE OF CONTENTS

2 INTRODUCTION 4

2.1 Overview and Background 4

2.2 Recommended Technical Expertise 5

3 REGISTRATION – TECHNICAL SPECIFICATIONS 6

3.1 Process Overview 6

3.2 Registration Flow 7

3.3 End-User Authorization Code Flow 8

3.4 Machine-to-Machine Flow 9

3.5 Full Systems Flow Diagram 10

3.6 Open ID Connect Provider 11

3.7 eCBSV OpenID Connect Requirements Summary 12

3.8 OIDC Discovery 13

3.9 Dynamic Client Registration Endpoint 14

3.10 JSON Web Key Set (JWKS) Endpoint 15

3.11 Authorization Endpoint 16

3.12 Token Endpoint 17

3.13 UserInfo Endpoint 17

4 REGISTRATION – TEST SERVICE 18

4.1 Entity OIDC URL Validation Tool 18

4.2 OIDC Validation Tool Screen Shots 18

4.3 OIDC Issuer URL Web Page Error Codes and Exception Handling 21

4.4 Successful Test and Next Steps 23

5 ENROLLMENT – CUSTOMER CONNECTION 24

5.1 Enrollment: Customer Connection Overview 24

5.2 Customer Connection: End-User Authorization Code Integration 24

5.3 Accessing the Customer Connection 28

6 VERIFICATION SERVICE – Authorization and Encryption 29

6.1 Machine-to-Machine Integration 29

6.2 Production Endpoint 29

6.3 Obtaining Access Token (M2M Flow) - Production 29

6.4 Sample Requests to Production Endpoint 31

6.5 Encryption Requirements - Production 33

7 VERIFICATION SERVICE – Requests and Responses 36

7.1 Data Content for Request 36
7.2 Data Content for Response 39
7.3 eCBSV Error Codes and Exception Handling 40
7.4 Sample Requests and Reponses 42
8 VERIFICATION SERVICE – External Testing Environment 48
8.1 Overview 48
8.2 Register for ETE 48
8.3 Accessing eCBSV Service – External Testing Environment (ETE) 49
8.4 ETE Test Data and Response Codes 49
8.5 Obtaining Access Token (M2M Flow) - ETE 49
8.6 Sample Request to ETE Endpoint 50
8.7 Encryption Requirements - ETE 52
9 HEALTH PING 55
9.1 Operation 55
9.2 Parameters 55
9.3 Responses 56
10 CONTACT US 58
10.1 When to contact eCBSV Technical Support 58
10.2 eCBSV Technical Support Contact Information 58
10.3 What is needed when contacting eCBSV Technical Support 58
11 CHANGE HISTORY 59
APPENDIX 61
Appendix A: Financial Institution Registration 61
Appendix B: Supported Certificate Authorities 62
Appendix C: eCBSV Screen Package 63
Appendix D: Acronyms. 64
Appendix E: eCBSV ETE Test Data and Scenarios. 65

2 INTRODUCTION

2.1 *Overview and Background*

The Social Security Administration's (SSA) Electronic Consent Based SSN Verification (eCBSV) service provides Permitted Entities with the capability to perform real-time Social Security Number (SSN) verifications. The eCBSV service is a Representational State Transfer (REST) service to verify whether the name, date of birth, and SSN obtained from a consenting Numberholder matches the data as it appears in SSA's records. Additionally, if SSA's records show that an individual is deceased, a death indicator will be provided to the customer as part of the verification. If the submitted data does not verify, explanatory information will be provided to identify the data element(s) that do not match the information in SSA's records.

In the Expanded Rollout, SSA will send email invitations to directly enroll in eCBSV to companies who applied during the initial application period in July 2019. In the future, SSA may announce open direct enrollment periods on its eCBSV website. The diagram displayed below provides a high-level view of the steps required to use the eCBSV service:

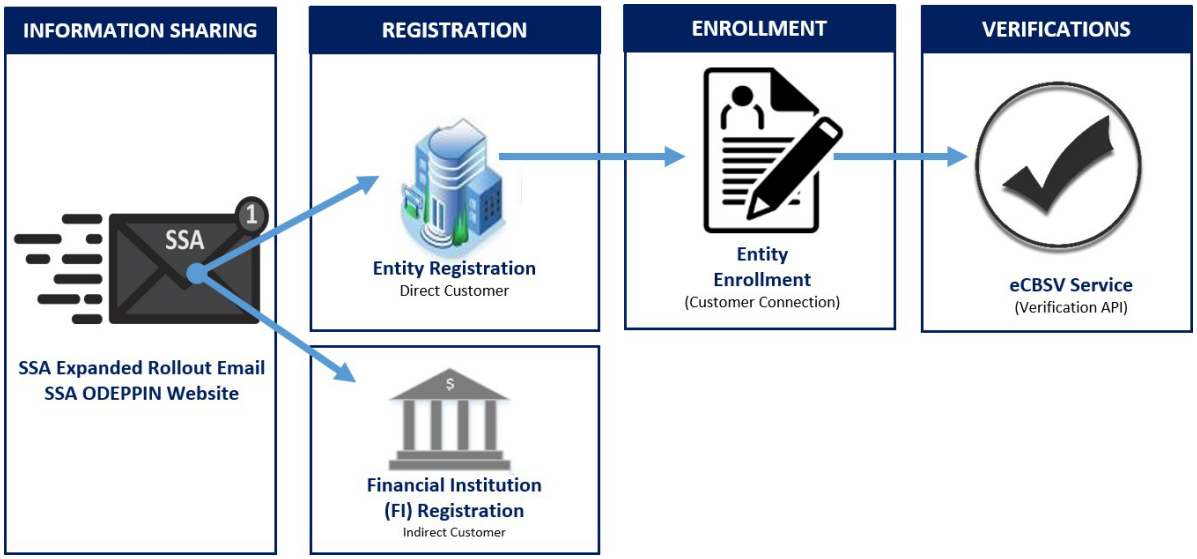


Figure - 2.1 Overview of required step to use eCBSV

More information about eCBSV program and business process may be found at the following link: https://www.ssa.gov/dataexchange/eCBSV/

NOTE: Financial Institutions have the option to **indirectly** participate in the eCBSV program through a Service Provider. Please see Appendix A for more information about the Indirect Registration for Financial Institutions.

2.2 *Recommended Technical Expertise*

Social Security recommends that each entity wishing to **directly** enroll in the eCBSV program have familiarity with the following concepts:

- Extended Validation SSL certificates
- OpenID Connect specification (OIDC), including Discovery, Dynamic Client Registration, and Authorization Code Flow
- JSON Web Tokens (JWTs)
- OAuth 2, including JWT client assertion
- Understanding of REST API requests and responses (JSON) and headers
- JSON Web Encryption (JWE)

3 REGISTRATION – TECHNICAL SPECIFICATIONS

3.1 *Process Overview*

The diagram displayed below provides the high-level steps required by an Entity to register to use the eCBSV service:

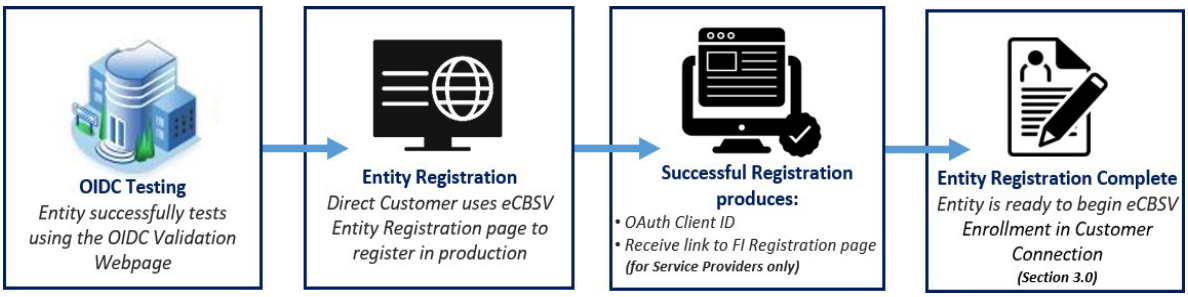


Figure 3.1 - Process Overview for Entity Registration

3.2 *Registration Flow*

(Refer to Figure 1 below)

During registration of an Entity, the SSA system will:

- Verify that Extended Validation (EV) SSL certificates are in place at relevant domains
- Create an SSA client in the Entity's OIDC Identity Provider (IdP) through dynamic client registration
- Create a mapping from the Entity's email domain to the Entity's OIDC IdP login page to facilitate the end-user authentication code flow
- Create the Entity's OAuth Client ID in the SSA authentication layer and email it to the Entity

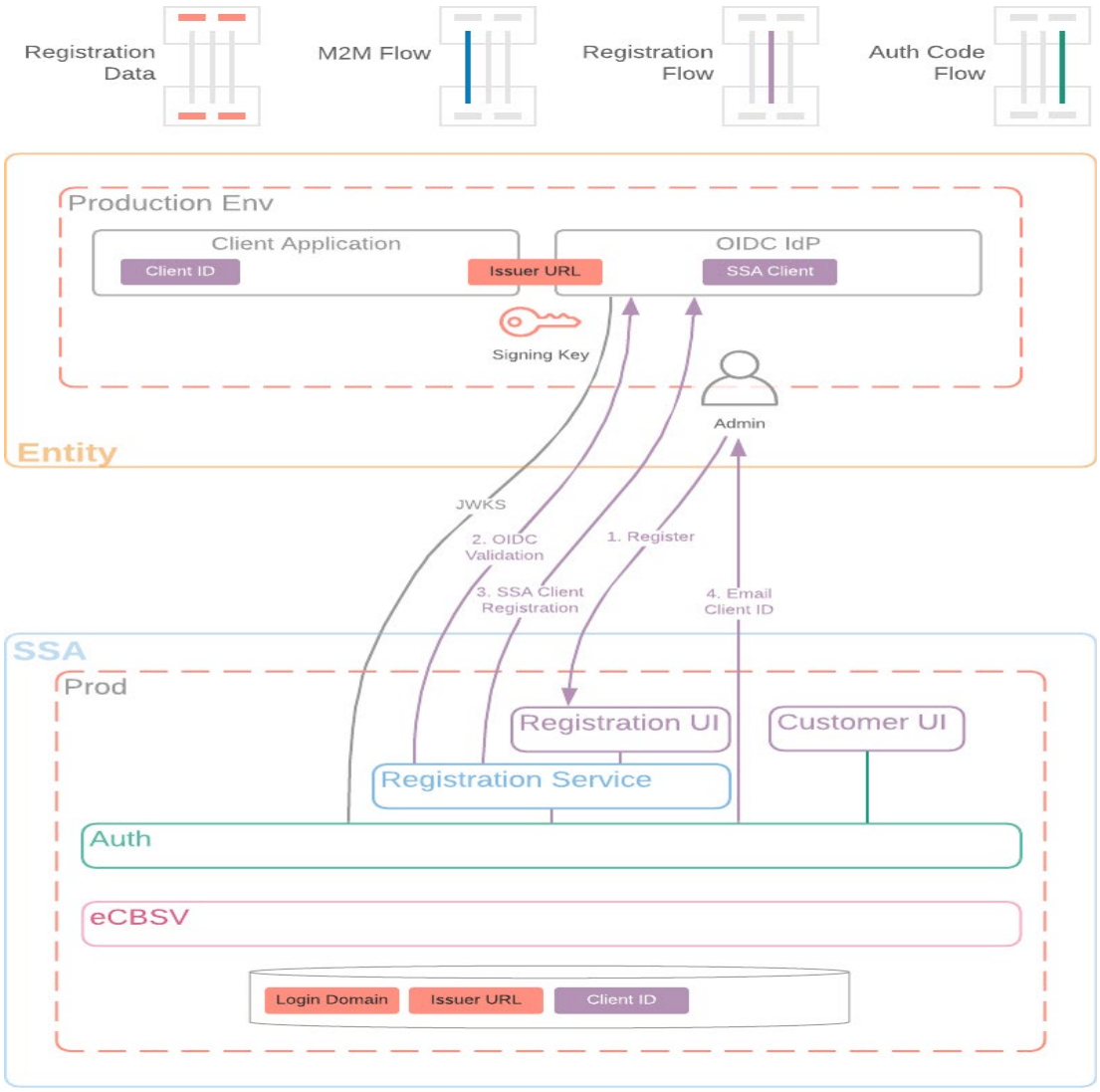


Figure 1.2 - Registration Flow

3.3 *End-User Authorization Code Flow*

(Refer to Figure 2 below)

In the end-user authorization code flow, the user is prompted to enter a corporate email address at SSA's user interface. The user is redirected to the Entity's OIDC IdP, where they can present their credentials to obtain an authorization code. SSA's authentication layer can use the authorization code to obtain a token from the Entity's OIDC IdP to verify and allow access to the eCBSV Customer Connection portal. For more information, please see section 5.2.

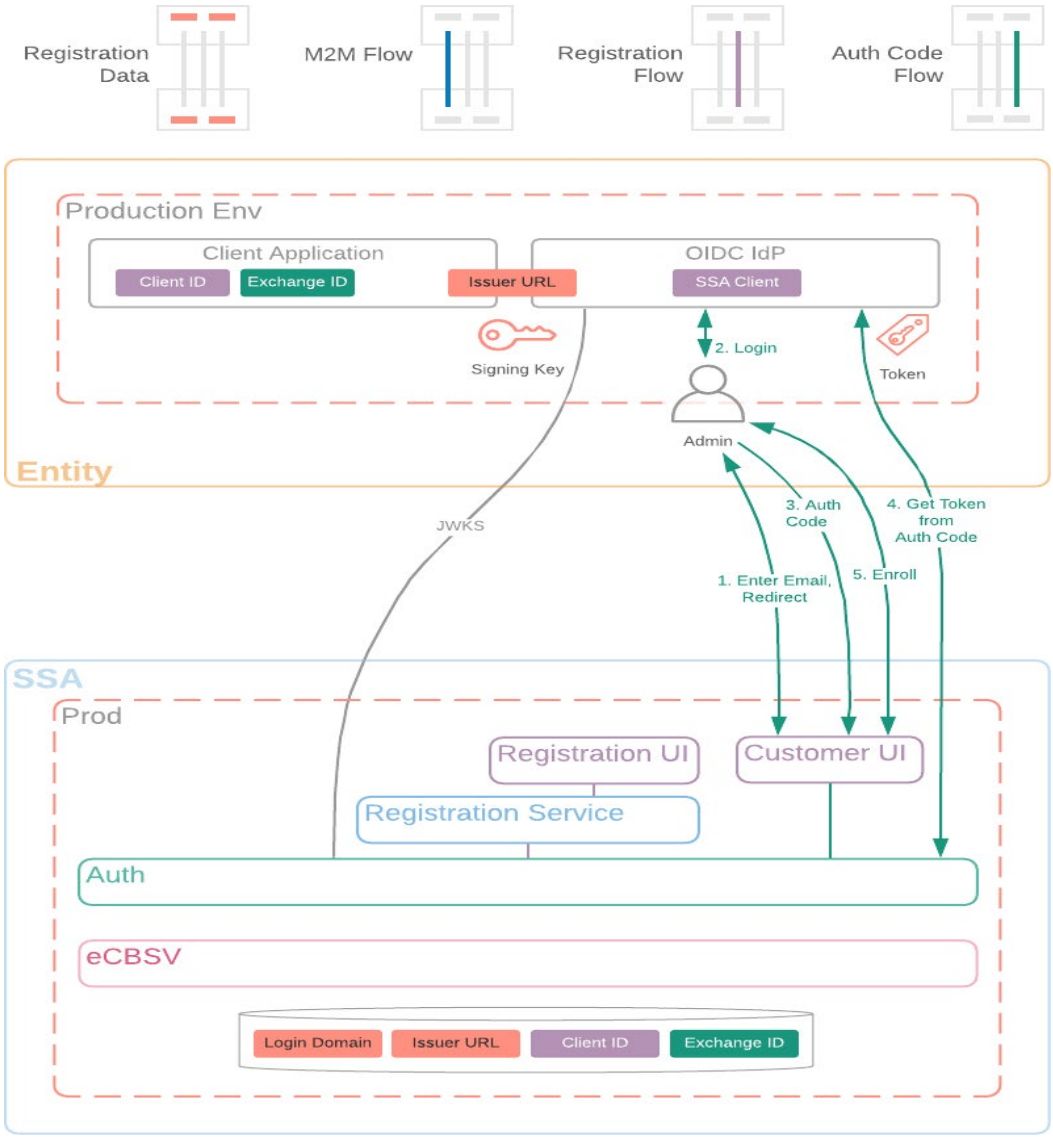


Figure 3.3 - Authentication Code Flow

3.4 *Machine-to-Machine Flow*

(Refer to Figure 3 below)

In machine-to-machine flows, the Entity's client application creates a client assertion JSON Web Token (JWT) using a designated issuer URL and signing key (that the OIDC IdP serves at its JWKs endpoint). That JWT is presented to SSA's authentication layer to obtain an access token, which can then be used in REST calls to eCBSV services along with the Exchange ID received after completing enrollment.

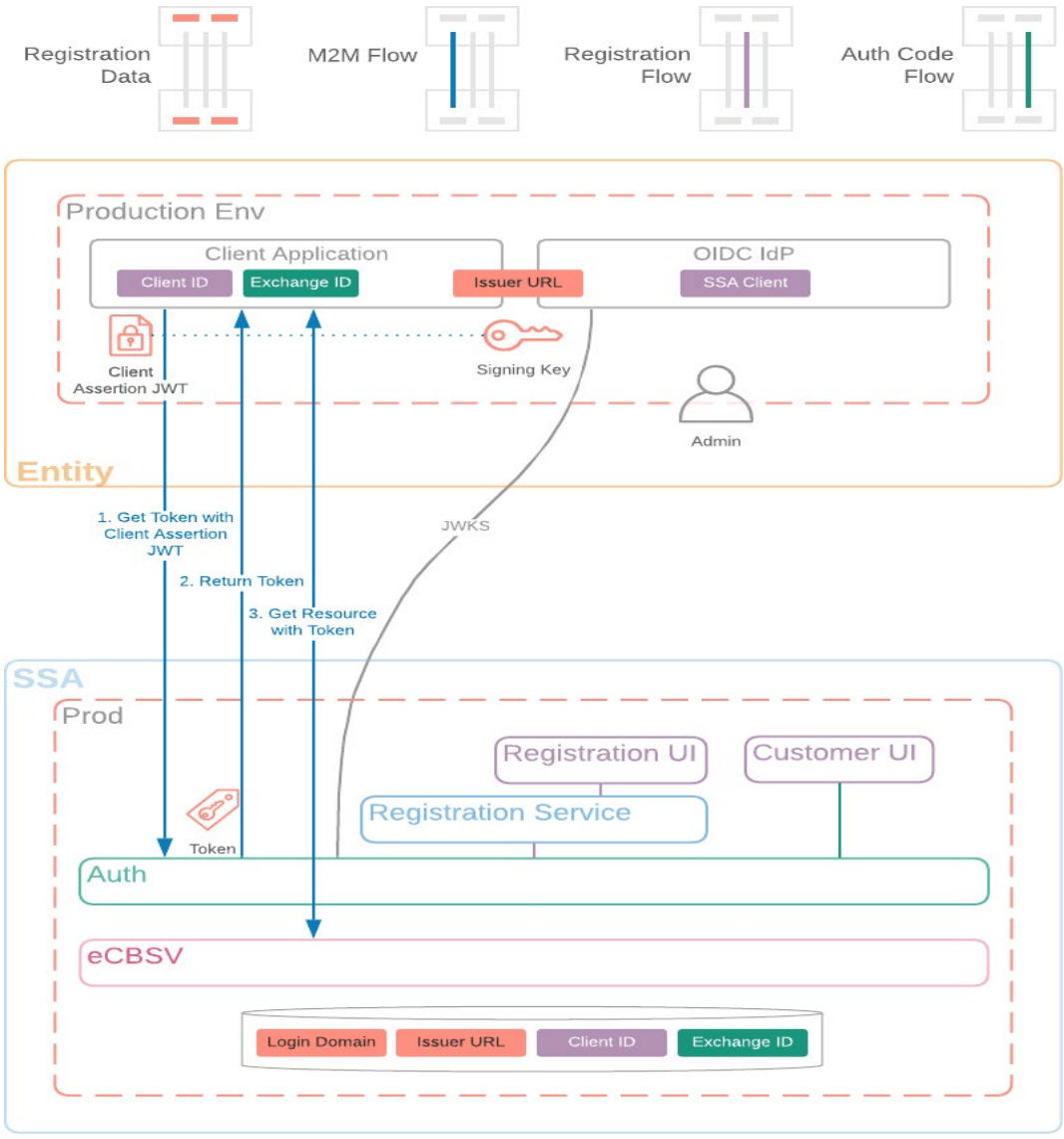


Figure 3. 4 - Machine to Machine Flow

3.5 *Full Systems Flow Diagram*

(Refer to Figure 4 below)

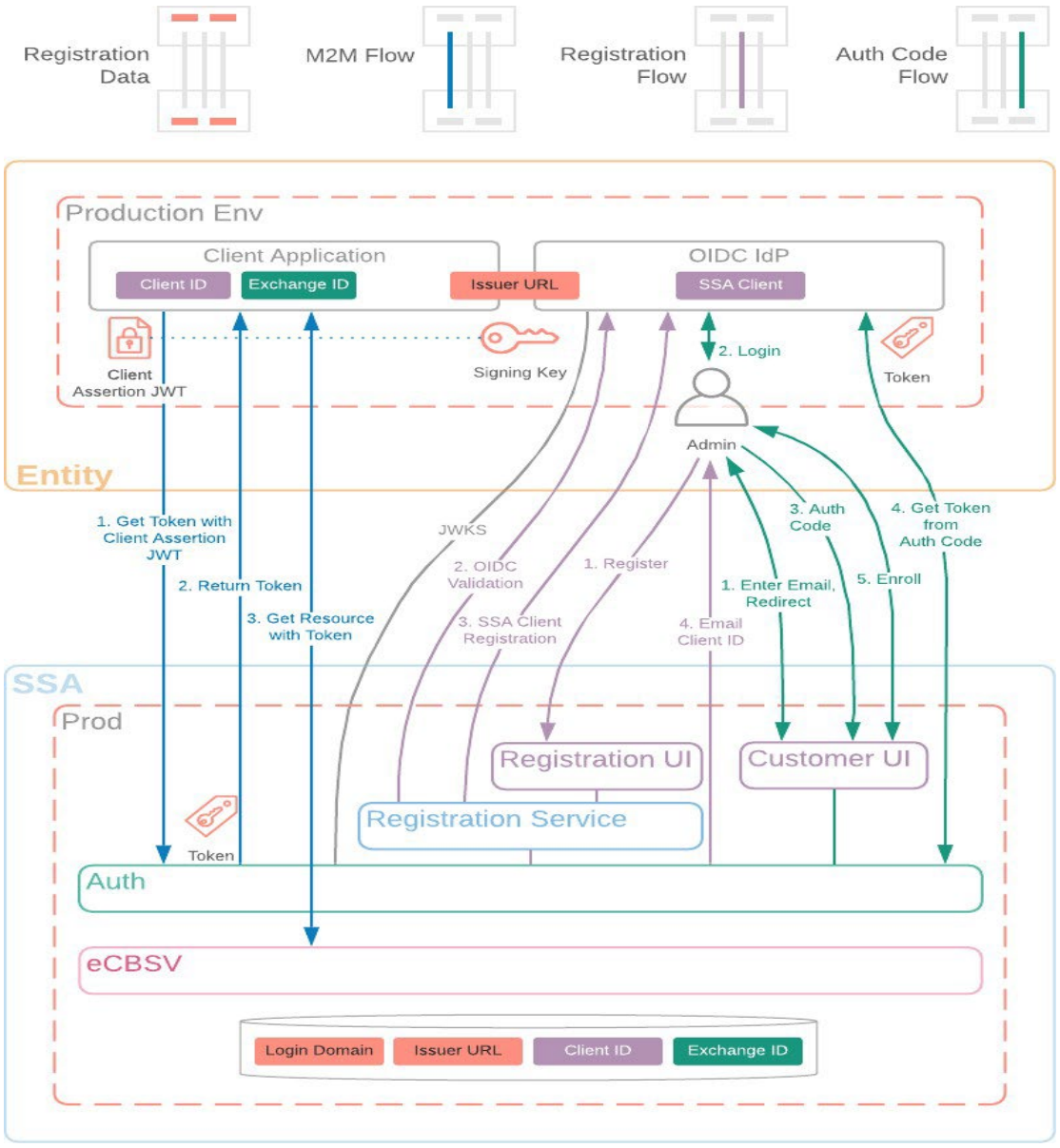


Figure 3.5 - Full System Flow

3.6 *Open ID Connect Provider*

In order to login with a corporate identity to access SSA's business services, entities are required to host an OpenID Connect Provider (OP) that supports the following capabilities:

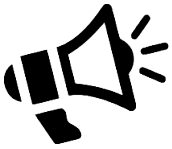
- Dynamic Client Registration
- Authentication using the Authorization Code Flow

REQUIRED: refer to the OIDC Technical Specifications in the table, eCBSV OpenID Connect Requirements Summary**, to ensure the requisite criteria are met to register with eCBSV. In some cases, the OIDC Connect Configuration specifications for eCBSV differ from OIDC/JWT specifications.**

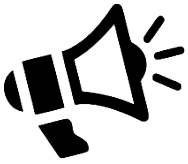
In order to support these features, the entity ***MUST*** host and implement the following endpoints:

- Well known OpenID Configuration Endpoint
- Dynamic Client Registration Endpoint
- JWKS Endpoint
- Authorization Endpoint
- Token Endpoint
- UserInfo Endpoint

The entity ***MUST*** use *Extended Validation (EV) SSL certificates* for endpoint authentication and utilize TLS 1.2[1] for any communication with these endpoints. The Extended Validation (EV) certificate ***MUST*** conform to the specification defined in Entity Extended Validation Certificate Requirements and be issued from a supported certificate authority defined in Appendix B of this document.



NOTE: To successfully register, the entity's company name and the company name on the EV certification **MUST** be an exact match.



It is ***strongly recommended*** that entities use one of the many OpenID Connect Provider products, SaaS Providers, or open source projects available that **ALREADY** meet the requirements defined here, rather than attempting to develop their own solution.

[1] *TLS 1.3 will be required by January 1, 2024 as per NIST SP 800-52 Rev. 2. SSA may offer TLS 1.3 as an option prior to that date.*

3.7 *eCBSV OpenID Connect Requirements Summary*

OIDC/JWT Specification Item	OIDC/JWT Specification Requirement	EAZE/eCBSV-Specification Requirement
ENDPOINTS		
EV SSL Certificates	Not required (TODO Reference)	EV SSL Certificates
TLS	*TLS is required. Version is not dictated.*	TLS 1.2 (This is a NIST Requirement)
OIDC CONFIG		
token_endpoint	**REQUIRED** unless only the Implicit Flow is used	**REQUIRED** because Authorization Code Flow is used
userinfo_endpoint	RECOMMENDED	**REQUIRED**
registration_endpoint	RECOMMENDED	**REQUIRED** Needed to create Auth Code SSA client at Entity IdP
grant_types_supported	Dynamic OpenID Providers MUST support the authorization_code	**REQUIRED**
token_endpoint_auth_methods_supported	OPTIONAL	**REQUIRED** Must contain client_secret_post
scopes_supported	RECOMMENDED	**REQUIRED** Must contain openid, email, roles
DYNAMIC CLIENT REGISTRATION		
client_secret	OPTIONAL	**REQUIRED** for Auth Code Flow
client_secret_expires_at	REQUIRED if client_secret is issued	MUST be 0

JWKS ENDPOINT			
	R256 sig keys	No specification	Keys must expire after 367 days and rotate
AUTHORIZATION ENDPOINT			
	state	RECOMMENDED	**REQUIRED**
	nonce	OPTIONAL	**REQUIRED**
CLIENT ASSERTION JWT			
	Kid (key)	Essentially optional. The kid must be available but not necessarily via JWKS endpoint or the JWKS endpoint as an IdP.	**REQUIRED** Signed by a kid (key) available at the JWKS Endpoint
	iss (issuer)	REQUIRED, though not necessarily an IdP Issuer URL	*iss (issuer) must match the OIDC IdP Issuer URL*

3.8 *OIDC Discovery*

The well-known OpenID Configuration endpoint **MUST** conform to Section 4.1 of the OpenID Connect Discovery specification.

The endpoint ***MUST*** respond to **GET** requests at the path **/.well-known/openid-configuration** and ***MUST*** serve a valid OpenID Configuration document as described in Section 4.2.

The entity ***MUST*** support all the **REQUIRED** values described in Section 3 as well as the following **ADDITIONAL** values:

- **token_endpoint**
  This value **MUST** be the URL of the Token endpoint (required to support the authorization code flow).

- **userinfo_endpoint**
  This value **MUST** be the URL of the UserInfo endpoint.

- **registration_endpoint**
  This value **MUST** be the URL of the Dynamics Client Registration endpoint.

- **grant_types_supported**
  At a minimum, the entity **MUST** support the ***authorization_code*** grant type.

- scopes_supported
  At a minimum, the entity **MUST** support the openid, email, and roles scopes.
  - Entities **MUST** ensure controls are in place to properly set attributes that allow Authorized Users access to the eCBSV service. See the user agreement for more information.

- userinfo_signing_alg_values_supported
  At a minimum, the entity MUST support the RS256 signing algorithm.

- token_endpoint_auth_methods_supported
  At a minimum, the entity MUST support the client_secret_post method.

- claim_types_supported
  The entity MUST support normal claims. If omitted, SSA assumes only normal claims. The SSA RP will ignore distributed claims.

3.9 *Dynamic Client Registration Endpoint*

The entity MUST host a dynamic client registration endpoint in accordance with **Section 3 of the OpenID Connect Dynamic Client Registration**. The endpoint **MUST** use TLS 1.2[2] and **MUST** be secured using an EV SSL certificate. The URL for this endpoint **MUST** match the value of the ***registration_endpoint*** in the OpenID Connect configuration document.

The entity may **OPTIONALLY** provide an Initial Access Token or other Authorization header during entity registration which restricts dynamic client registration requests to only SSA's services. The entity may also restrict dynamic client registration calls to only be permitted from SSA source IP addresses in the CIDR block: **137.200.0.0/16**.

The entity **MUST** support all the **REQUIRED** values described in **Section 3.2** as well as the following **ADDITIONAL** values:

- **client_secret**
  This value along with the ***client_id***, **MUST** be unique for the SSA RP. Furthermore, this value MUST be confidential and issued in accordance with **Section 5.1.4.2.2 of RFC 6819**.

[2] *TLS 1.3 will be required by January 1, 2024 as per NIST SP 800-52 Rev. 2 here. SSA may offer TLS 1.3 as an option prior to that date.*

- **client_secret_expires_at**
  This value is required to be 0, indicating that the client secret does not expire.

If a **client_secret** is compromised, the entity SHALL immediately take the following action:

- Notify SSA of the compromised ***client_secre***t value.
- Expire the ***client_secret*** to force re-authentication.

3.10 *JSON Web Key Set (JWKS) Endpoint*

The entity MUST host a JSON Web Key Set (JWKS) endpoint that conforms to the specification defined in Section 5 of RFC 7517. The endpoint **MUST** use TLS 1.2 and **MUST** be secured using an EV SSL certificate. The URL for this endpoint **MUST** match the value of ***jwks_uri*** in the OpenID Connect configuration document.

The array of **KEYS** retrieved from the endpoint **MUST** contain at least **ONE** JSON Web Key (JWK) value that utilizes the RSASSA-PKCS1-v1_5 scheme as defined in Section 3.3 of RFC 7518.

The entity **MUST** specify the following attribute values for the key(s):

- **use**
  This value **MUST** be ***sig*** for the key.
- **alg**
  This value **MUST** be ***RS256*** for the key.

The entity OP must use the associated private key to sign any JSON Web Tokens (JWTs) (such as the id_token or user's claims) when communicating with SSA. SSA will utilize the public keys to verify the signature of the JWT. The entity must not disclose the private keys used for signing to any third-parties.

Finally, the keys **MUST EXPIRE** after a maximum of 367 days and **MUST** be **ROTATED** accordingly.

It is recommended that both the expiring and new keys be available during rotation to avoid interruption of service.

In the case that a private key is compromised, the entity **SHALL** immediately take the following action:

- Notify SSA immediately of the public key that corresponds to the compromised private key.
- Delist the compromised key at the JWKS endpoint.
- Generate a new public-private key pair (in accordance with the specifications described above) and list the new public key at the JWKS endpoint.

3.11 *Authorization Endpoint*

The entity **MUST** host an Authorization endpoint that conforms to the specification defined in Section 3.1.2 of OpenID Connect Core. The endpoint **MUST** use TLS 1.2[3] and **MUST** be secured using an EV certificate. The URL for this endpoint **MUST** match the value of ***authorization endpoint*** in the OpenID Connect configuration document. The endpoint **MUST** support authentication using **Authorization Code Flow** as defined in Section 3.1.1 of OpenID Connect Core.

The entity **MUST** support Authentication requests with all the **REQUIRED** values described in Section 3.1.2.1 of OpenID Connect Core as well as the following **ADDITIONAL** values:

- **state**
  This value is used to mitigate Cross-Site Request Forgery (CSRF) attacks and **MUST** be passed as-is when the entity OP invokes the callback specified in the ***redirect_uri*** Authentication request parameter.
- **nonce**
  The value is passed through unmodified from the Authentication request to the ID Token.

  The following value is **RECOMMENDED** in order to improve user experience:

  - **login_hint**
    This is a hint about the login identifier that the end-user might use to log in (typically a corporate email address).

If the end-user is authenticated successfully, the entity OP **MUST** respond to an Authentication request with a valid success response in accordance with Section 3.1.2.5 of OpenID Connect Core.

If the Authentication request object coming from the SSA Relying Party (RP) is invalid or there is an error during authentication, the entity OP **MUST** respond with a valid error response as defined in Section 3.1.2.6 of OpenID Connect Core.

[3] *TLS 1.3 will be required by January 1, 2024 as per NIST SP 800-52 Rev. 2 here. SSA may offer TLS 1.3 as an option prior to that date.*

3.12 *Token Endpoint*

The entity **MUST** host a Token endpoint that conforms to the specification defined in **Section 3.1.3 of OpenID Connect Core**. The endpoint **MUST** use TLS 1.2 and **MUST** be secured using an EV certificate. The URL for this endpoint **MUST** match the value of ***token_endpoint*** in the OpenID Connect configuration document. The entity OP MUST validate all Token requests from the SSA RP in accordance with **Section 3.1.3.2 of OpenID Connect Core**. Furthermore, the endpoint **MUST** authenticate Token requests from the SSA RP using the ***client_secret_post*** method defined in **Section 9 of OpenID Connect Core**.

If the Token request was successfully validated, the entity OP **MUST** respond with a valid success response in accordance with **Section 3.1.3.3 of OpenID Connect Core**.

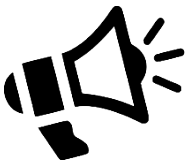
If the Token request object coming from the SSA RP is invalid or there is an error during validation, the entity OP **MUST** respond with a valid error response as defined in **Section 3.1.3.4 of OpenID Connect Core**.

3.13 *UserInfo Endpoint*

The entity **MUST** host a UserInfo endpoint that conforms to the specification defined in **Section 5.3 of OpenID Connect Core**. The endpoint **MUST** use TLS 1.2[4] and **MUST** be secured using an EV certificate. The URL for this endpoint **MUST** match the value of ***userinfo_endpoint*** in the OpenID Connect configuration document. The entity OP **MUST** authorize all UserInfo requests from the SSA RP via an OAuth 2.0 Bearer Token as specified in **RFC 6750**.

If the UserInfo request was successfully authorized the entity OP **MUST** respond with a valid UserInfo response in accordance with **Section 5.3.2 of OpenID Connect Core**. At a minimum, the entity OP **MUST** use JSON format and sign all UserInfo response objects. As such, the content-type header for the HTTP response **MUST** be ***application/jwt***. As defined in the specification the signed response **MUST** include ***iss*** (issuer) and ***aud*** (audience) claims.

If the UserInfo request object coming from the SSA RP is invalid or there is an error during authorization, the entity OP **MUST** respond with a valid error response as defined in **Section 5.3.3 of OpenID Connect Core**.



NOTE: Entities **CANNOT** move forward until technical development from this section has been completed

[4] TLS 1.3 will be required by January 1, 2024 as per NIST SP 800-52 Rev. 2. SSA may offer TLS 1.3 as an option prior to that date.

4 REGISTRATION – TEST SERVICE

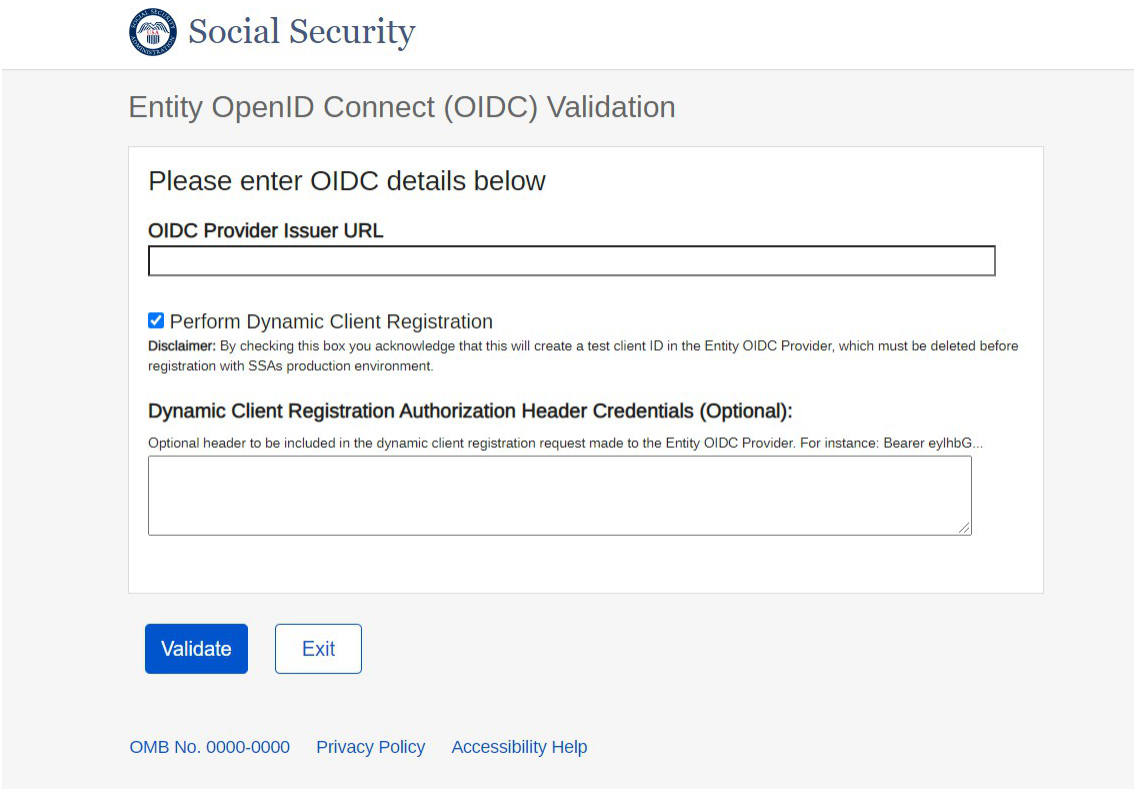
4.1 *Entity OIDC URL Validation Tool*

Once an entity has met all prerequisites and completed their development as specified in Section 3 of this document, the entity is highly encouraged to test prior to attempting to register in Production. The entity's OIDC URL and Dynamic Client Registration Authorization Header Credential should be tested to ensure there are no issues ***prior to attempting registration in production in order to prevent business downtime*.**

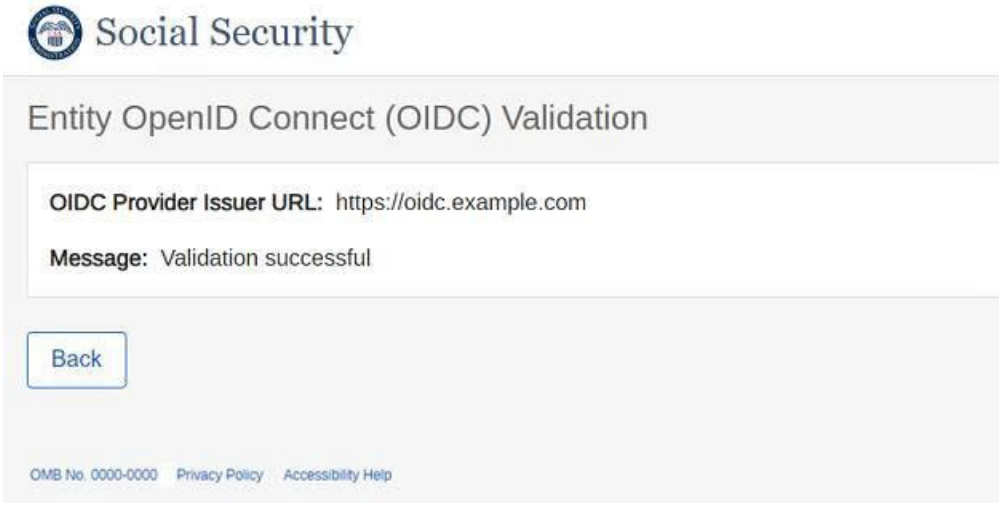
A link to the OIDC URL Validation Tool is found here.

4.2 *OIDC Validation Tool Screen Shots*

1. Enter OIDC Provider Issuer URL

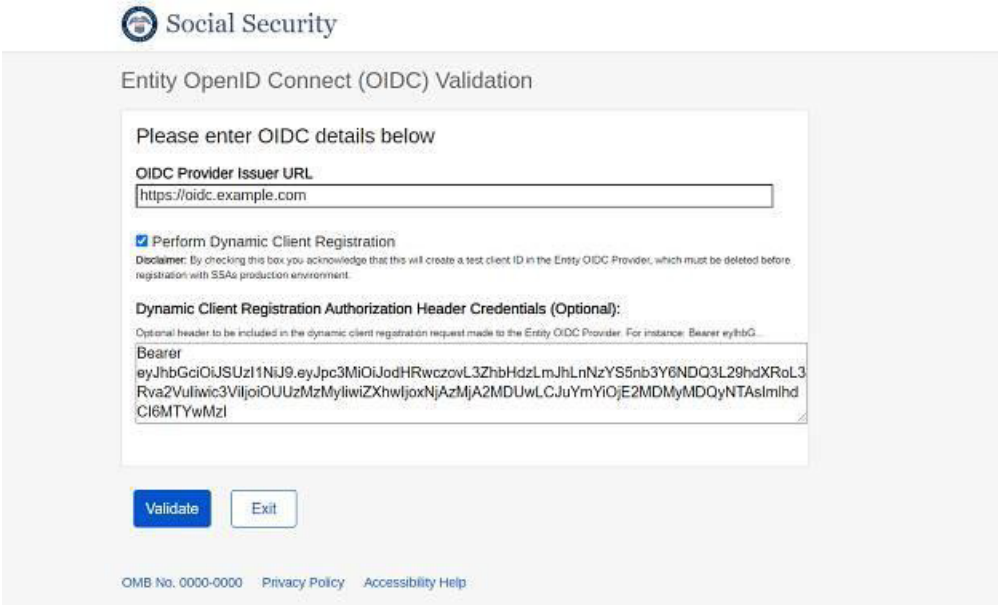


2. *Validation Successful*

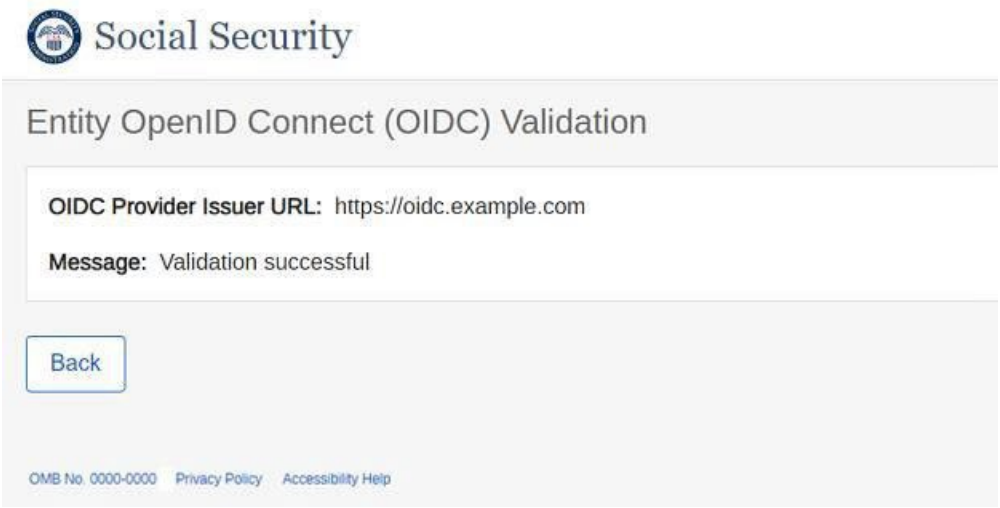


3. *Invalid Issuer URL Validation – Failure Message*

4. Enter Dynamic Client Registration Authorization Header Credentials *(Optional)*



5. *Validation Successful*



6. *Validation Failure*

4.3 *OIDC Issuer URL Web Page Error Codes and Exception Handling*

The test validation process will provide an Error Code with a corresponding http code indicating that there is a problem with the OIDC Issuer URL, the EV SSL certificate, and/or the Dynamic Client Registration Authorization Header Credential.

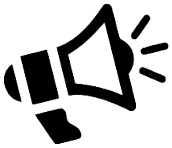
Error Code	Error Code Description	http Code
400.1.0	The issuer URL must be a valid URL	**400**
400.1.1	Failed GET request for the OIDC configuration	**400**
400.1.2	The OIDC configuration is missing the following claim [field]	**400**
400.1.3	The OIDC configuration claim [a field] must contain a value	**400**
400.1.4	The JWKS at [URL] cannot be retrieved	**400**
400.1.5	The JWKS must contain at least one key	**400**
400.1.6	The JWKS should have a key with alg:RS256 and use:sig	**400**
400.1.7	The following URL failed the SSL Validation Service [URL]	**400**
400.1.10	Failed POST request for the Dynamic Client Registration Endpoint	**400**

400.1.11	The Dynamic client registration response does not meet out requirements. [A field] is null	**400**
400.1.12	The Dynamic client registration response does not meet out requirements.	**400**
400.1.14	The issuer URL must be provided	**400**
400.1.15	The domain name must be provided	**400**
400.2.0	URL must not be empty	**400**
400.2.1	URL must be a valid HTTPS URL	**400**
400.2.2	A connection could not be established to the given URL	**400**
400.2.4	The certificate at the given URL could not be parsed	**400**
400.2.5	The certificates for the given URL could not be retrieved	**400**
400.2.6	The certificate at the given URL is not an Extended Validation Certificate from an SSA trusted Certificate Authority	**400**
400.2.7	The certificate at the given URL is untrusted	**400**
400.2.8	A connection could not be made to a valid hostname using the provided URL	**400**
401	Authentication Failure. Occurs when no valid bearer token is provided with the request	**401**
403	Authorization Failure. Occurs when a valid bearer token is provided with the request, but the token does not have the role required to access the resource	**403**
404	Not Found	**404**
500	Internal Server Error	**500**

4.4 *Successful Test and Next Steps*

When a successful validation message is displayed at the bottom of the screen the entity MUST complete the following steps to continue with the registration process:

1. Entity MUST delete the OAuth Client ID generated during validation testing
2. Access the eCBSV Entity Registration webpage to register in production.
    - The link to the eCBSV Entity Registration webpage can be found in the Expanded Roll-Out invitation email provided by the ^eCBSV mailbox.
    - Screen shots of the Entity Registration webpage can be found in the eCBSV Screen Package (Appendix C)



NOTE: To successfully register in Production, the entity's company name and the company name on the EV certification **MUST** be an exact match.

Ensure that the OIDC Issuer URL entered during registration is the final URL that will be used in the production environment. Do not use a temporary or staging URL as SSA does not have a way to make changes to this URL once the entity goes live.

5 ENROLLMENT – CUSTOMER CONNECTION

5.1 *Enrollment: Customer Connection Overview*

Once successfully registered in production, the entity is ready to complete the eCBSV Enrollment Process in the eCBSV Customer Connection.

The eCBSV Customer Connection is an automated workflow tool that will guide the entity through the enrollment process. During the enrollment process, the entity is required to provide their Permitted Entity Certification, sign the User Agreement, and purchase the Tier Subscription.

The eCBSV Customer Connection screens can be viewed in the eCBSV Screen Package (Appendix C).

More information about the eCBSV Enrollment process can be found on SSA's eCBSV Webpage: https://www.ssa.gov/dataexchange/eCBSV/

5.2 *Customer Connection: End-User Authorization Code Integration*

Access the Customer Connection follows the end-user authorization code flow, as displayed on the system diagram in **Section 3.3, Figure 1-2**.

To access the Customer Connection, entities need to configure the Entity's OIDC IdP redirect URI and id token JWT per the information below:

Entity's OIDC IdP redirect URI Requirements

When attempting to access the eCBSV Customer Connect portal (URL provided in the next section), the user is prompted to enter a corporate email address. Upon submit, the user is redirected to the Entity's OIDC IdP. The redirected URL will need OIDC parameters below:

- **redirect_uri –** Entity's OIDC IdP will check if the URL in the redirect_uri matches the one registered with the OIDC Client (client_id below). After successful authentication at IdP, the request will be redirected to this redirect_uri.
- **Authorization code** & **state** should be included as part of the URI.
- **client_id –** SSA's OIDC client id – dynamically registered
- **response_type -** code
- **state –** alpha numeric characters; should be included in the response to redirect URL along with the authorization code
- **login_hint –** User's email id; pre-populates the IdP's login form.
- **nonce – alpha numeric characters**

Sample redirect URI provided here:
https://oidc.entity1.com/authorize?*nonce*=jNBeTYDaLRQ8&*redirect_uri*=https%3A%2F%2Fapiauth.ssa.gov%2Fisam%2Fsps%2Foidc%2Frp%2FEAZE%2Fredirect%2Fentity1.com&*response_type*=code&*state*=****&*scope*=openid+email+roles&*login_hint*=test%40entity1.com&*client_id*=**********

id_token JWT Requirements

- The JWT header **MUST** have the following attributes:
  - **alg** This value MUST be RS256.
  - **typ** This value MUST be JWT.

- The JWT body **MUST** have the following claims:
  - **iss** The issuer of this JWT. This value MUST be the same issuer URL as specified in the Entity's OIDC Configuration.
  - **sub** This is the Subject Identifier. Its value MUST be the authenticated user at ID (usually user's email address).
  - **aud** This is the Audience Identifier. Its value MUST be SSA's OAuth Token Endpoint: https://apiauth.ssa.gov/mga/sps/oauth/oauth20/token.
  - **exp** This value MUST be the expiration time on or after which the JWT is not accepted for processing and should be short-lived, on the order of a few minutes.
  - **iat** This value MUST be the time at which the JWT was issued.
  - **given_name** This value MUST be the user's first name.
  - **family_name** This value MUST be the user's last name.
  - **email** This value MUST be the user's email address.

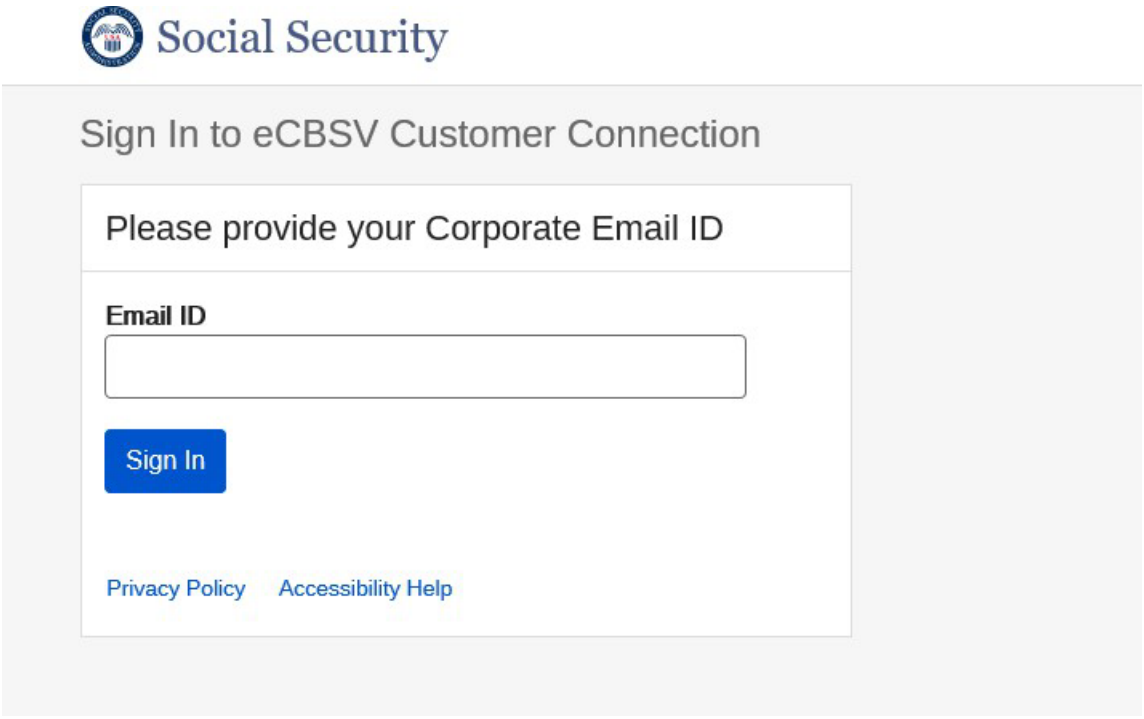
SAMPLE DECODED JWT – id_token


```
{
 "alg": "RS256",
 "typ": "JWT"
}

{
 "aud": "https://apiauth.ssa.gov/mga/sps/oauth/oauth20/token",
 "iss": "https://oidc.entity1.com/production",
 "sub": "test@entity1.com",
 "exp": 1652799651,
 "iat": 1652799351,
 "given_name": "John",
 "family_name": "Doe",
 "email": "test@entity1.com"
}
```

Below are the details of OIDC Authorization Code Flow:

1. User tries to access SSA's Customer Connect Login https://apiauth.ssa.gov/entityLogin.html page

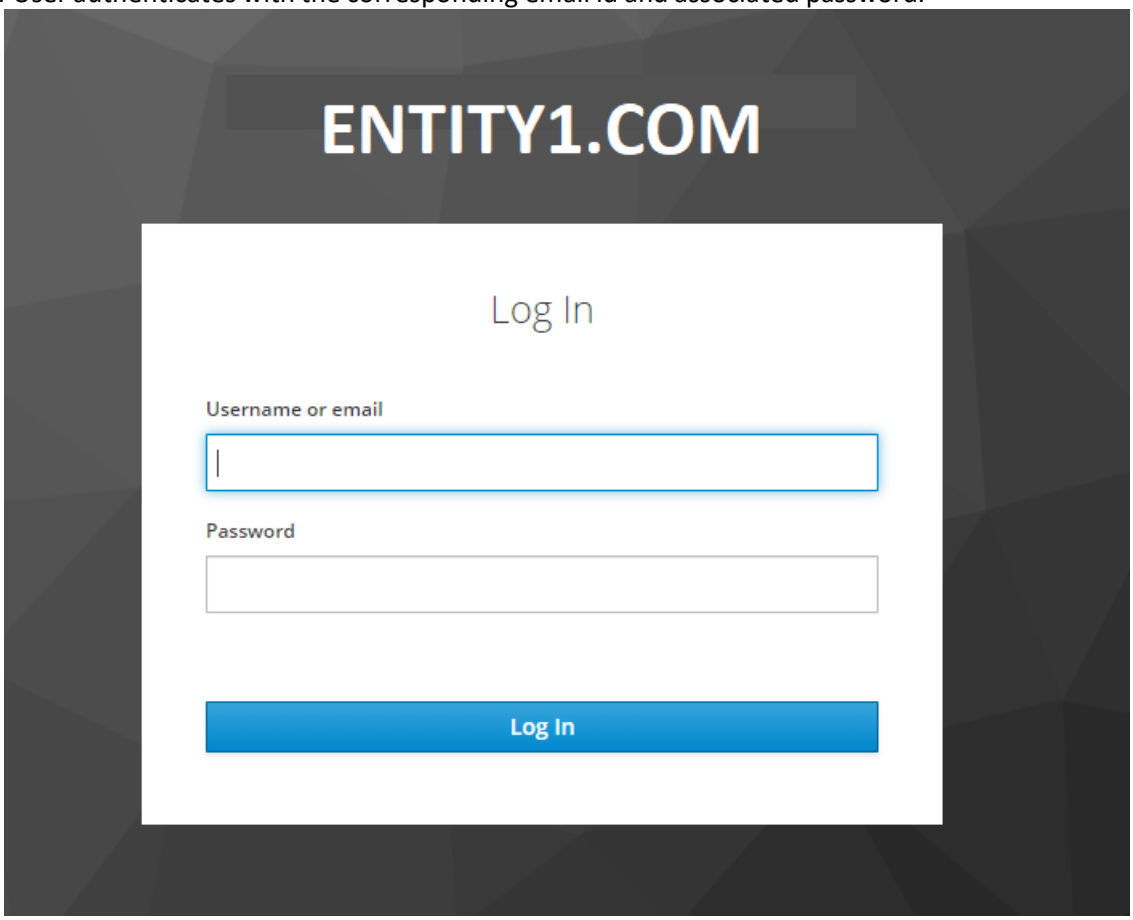


2. User enters corporate email id and submits the form

   Ex: test@entity1.com

3. Browser now redirects the request to entity1's OIDC IDP (The Request URL will have OIDC related info – redirect_uri, client_id,response_type etc) – SAMPLE:
https://oidc.entity1.com/authorize?nonce=jNBeTYDaLRQ8&redirect_uri=https%3A%2F%2Fapiauth.ssa.gov%2Fisam%2Fsps%2Foidc%2Frp%2FEAZE%2Fredirect%2Fentity1.com&response_type=code&state=****&scope=openid+email+roles&login_hint=test%40entity1.com&client_id=**********

4. User authenticates with the corresponding email id and associated password.



5. Entity1's OIDC IDP should validate the credentials. If valid redirect the browser's request to the "redirect_uri" as provided in the initial request above. This redirected request should include the authorization code (SSA's Redirect URL – SAMPLE : https://apiauth.ssa.gov/isam/sps/oidc/rp/EAZE/redirect/entity1.com?code=AUTHORIZATION_CODE_GENERATED_AT_IDP&state=AS_INCLUDED_IN_THE_INITIAL_REQUEST)

6. SSA's OIDC Relying Party will now connect to entity1's OIDC Token endpoint – requesting id_token & access_token. This request includes the code – i.e., AUTHORIZATION_CODE as provided by entity1 in the previous step.

7. Entity1's OIDC Server should now respond with the id_token. This **id_token** is JWT formatted, should be RS256 signed using the Certificate as indicated in the OIDC JWKS endpoint.
The id_token should have the authenticated user's **email, given_name, family_name** claims along with the standard **iss, sub, aud, iat, exp** claims

5.3 *Accessing the Customer Connection*

Once the steps in 5.2 are completed, the link to the eCBSV Customer Connection is as follows:

eCBSV Customer Connection
https://apiauth.ssa.gov/entityLogin.html

6 VERIFICATION SERVICE – AUTHORIZATION AND ENCRYPTION

6.1 *Machine-to-Machine Integration*

In machine-to-machine flows, the Entity's client application creates a client assertion JSON Web Token (JWT) using a designated issuer URL and signing key (that the OIDC IdP serves at its JWKs endpoint). That JWT is presented to SSA's authentication layer to obtain an access token, which can then be used in REST calls to eCBSV services along with the Exchange ID received after completing enrollment. Please reference the System Diagram found in Appendix C of this document.

6.2 *Production Endpoint*

SSA's eCBSV service will be available at the following endpoints:

SERVICE ENDPOINT	HEALTH PING ENDPOINT
https://ecbsvws.ssa.gov/eden/verify	https://ecbsvws.ssa.gov/eden/ping

To consume the eCBSV service, the Entity's API client has to pass an access token in the HTTP Authorization header as a bearer token.

Please reference Section 10 of this document for more information about the Health Ping.

6.3 *Obtaining Access Token (M2M Flow) - Production*

The API client will obtain the access token from SSA's OAuth Authorization Server.

SSA's OAuth Authorization server will be available at the following endpoint:

SERVICE ENDPOINT
https://apiauth.ssa.gov/mga/sps/oauth/oauth20/token

SSA's OAuth Authorization Server follows the JSON Web Token (**JWT) Profile for OAuth 2.0 Client Authentication & Authorization Grants** as described in **RFC7523** for issuing an access token and uses the JWT (JSON Web Token) format.

The entity's API client must authenticate with the OAuth Authorization Server using HTTP POST to the token endpoint. The following request parameters MUST be included:

- **grant_type - MUST** contain the value "client_credentials"
- **client_assertion_type - MUST** contain the value "urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
- **client_assertion - MUST** contain a single JWT - ***Requirements for the content of this JWT are described below.***

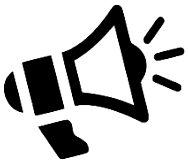
If the request contains a client_id parameter, this client_id value **MUST** *match the "sub" value claim in client_assertion.*

client_assertion JWT requirements

- The JWT MUST be signed with the Entity's Private key. Details about the Public Key MUST be available at the Entity's OpenID Connect (OIDC) JSON Web Key Set (JWKS) endpoint.
- The JWT header MUST have the following attributes:
  - **alg** This value MUST be RS256
  - **kid** This value MUST be the Key ID of the Private Key used to sign the JWT

  The Entity's OIDC JWKS Endpoint MUST have a reference to this "kid" and include the corresponding Public Key information.

- The JWT body MUST have the following claims:
  - **iss** The issuer of this JWT. This value MUST be the same issuer URL as specified in the Entity's OIDC Configuration.
  - **sub** This is the Subject Identifier. Its value MUST be the **client id** provided by SSA following successful registration with the eCBSV service.
  - **aud** This is the Audience Identifier value and its value MUST be SSA's OAuth Token Endpoint: *https://apiauth.ssa.gov/mga/sps/oauth/oauth20/token*
  - **exp** This value MUST be the expiration time on or after which the JWT is not accepted for processing and **should be short-lived**, on the order of a few minutes
  - **iat** This value MUST be the time at which the JWT was issued



NOTE: *The access token is valid for 30 minutes. A new token MUST be requested at the end of its expiry for continued access.*

The signature of this token will be validated using the information in the Entity's JWKS endpoint (jwks_uri attribute in the Entity's OIDC configuration).

Entity's API client MUST use Extended Validation (EV) Secure Socket(s) Layer (SSL) certificates for the OIDC endpoints and MUST utilize TLS 1.2[5] to communicate with SSA's OAuth Authorization Server and API service endpoints.

If you encounter an error when accessing this endpoint, please refer to the OAuth HTTP Error Codes table, located in Section 4.3 of document.

6.4 *Sample Requests to Production Endpoint*

SAMPLE ACCESS TOKEN REQUEST

```
POST /mga/sps/oauth/oauth20/token HTTP/1.1
Host: apiauth.ssa.gov
Content-Type: application/x-www-form-urlencoded
Accept: application/json
'grant_type=client_credentials&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer&
client_assertion=eyJraWQiOiIyV0hQU...'
```

DECODED JWT IN CLIENT_ASSERTION FROM THE SAMPLE ABOVE

```
{
  "kid":"2WHP5YmLrVhNlWmxWe01xeNY5amIul-qHKnS955IIfY",
  "alg":"RS256"
}
{
  "iss":"https://test.entity.com:7443/auth/realms/gcp",
  "sub":"780e78d2-007a-49af-b916-5cf36978705a",
  "aud":"https://apiauth.ssa.gov/mga/sps/oauth/oauth20/token",
  "exp":1570725265,
  "nbf":1570120405,
  "iat":1570120465
}
```

[5] *TLS 1.3 will be required by January 1, 2024 as per NIST SP 800-52 Rev. 2 here. SSA may offer TLS 1.3 as an option prior to that date.*

SSA's OAuth Authorization server will verify the client assertion and issue the access token in JWT format.

ACCESS TOKEN RESPONSE


```
{
  "access_token":"eyJraWQiOiJaRVNkb3Y2dWJSQVEJrIiwiYWxnIjoiUlMyNTYifQ...",
  "token_type":"bearer",
  "expires_in":1800
}
```


The entity's API Client can now include this JWT Access Token in HTTP Header and call SSA's eCBSV Service (Ex: be https://ecbsvws.ssa.gov/eden/verify)

ACCESS ECBSV SERVICE

```
GET /eCBSV HTTP/1.1
Host: ecbsvws.ssa.gov
Content-Type: application/json
Accept: application/json
Authorization: Bearer eyJraWQiOiJaRVNkb3Y2dWJSQVVQcndxSFlILXNzTEJrIiwiYWxnIjoiUlMyNTYifQ..."
exchangeID: XXXXXXXX
externalTransactionID: XXXXXXXXXX
```

exchangeID value is provided by SSA following successful registration with the eCBSV service.

exchangeID HTTP header MUST be included.

The value for **externalTransactionID** HTTP header is the entity's transaction ID. This is optional. This ID helps in correlating requests and troubleshooting.

6.5 *Encryption Requirements - Production*

The JSON data request payload **must** be encrypted using JSON Web Encryption (JWE).

SSA's public JSON Web Key (JWK) with details about the public key meant for encrypting the request payload will be available in the JSON Web Key Set (JWKS) endpoint:

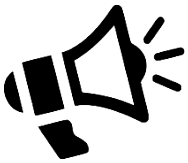
JWKS ENDPOINT
https://apiauth.ssa.gov/mga/sps/jwks

If an error is encountered when accessing this endpoint:

- Refer to OAuth HTTP Error Codes table located in the Appendix
- Follow-up with eCBSV Technical Support, as needed

This will be indicated by the attribute/value "use":"enc" in the JWK.

Error Code	Error Code Description	http Code
400	Decryption failure	**400**



NOTE: *The public key to encrypt JSON payload should not be fetched for every request as the encryption key is valid for a year.*

The client should obtain the encryption public key, cache it in memory and poll every 24 hours. If it has changed, invalidate the cache and update with new key.

When SSA encryption key gets updated, client requests using the old encryption key will fail to decrypt at SSA. Clients should invalidate the cache and update with new key.

EXAMPLE:

SSA'S JWKS ENDPOINT WITH SIGNING & ENCRYPTION JWK SAMPLE


```
{
  "keys":[
    {
      "kty":"RSA",
      "kid":"gCMwMdea-fQKPYjvnG0RftNb8JLCDpY1HUGDm0BuEH8",
      "use":"sig",
      "n":"vx3yHbw3fsowtlrz9Q82tvB2mPwCjWgUu3DhKHhv1quLmg5...kxAcB1UQ",
      "e":"AQAB"
    },
```

```
    {
      "kty":"RSA",
      "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0",
      "use":"enc",
      "n":"mPeQt-WxaX9STiil4EZhgt2FFw9MbhlQLI4tHfeCPYnXX...ltSnSWh",
      "e":"AQAB"
    }
  ]
}
```


Entities should include the **JWK** key id (**“use”:”enc”**) in the JWE header.

Entities can use the following key management algorithm (alg) and content encryption algorithm (enc) combinations to encrypt the request payload as shown in the sample below. **"alg":"RSA-OAEP-256","enc":"A256GCM" is preferred.**

Supported algorithm:


```
{
  "alg":"RSA-OAEP",
  "enc":"A256CBC-HS512",
  "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
}
{
  "alg":"RSA-OAEP",
  "enc":"A256GCM",
  "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
}
{
  "alg":"RSA-OAEP-256",
  "enc":"A256CBC-HS512",
  "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
}
{
  "alg":"RSA-OAEP-256",
  "enc":"A256GCM",
  "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
}
```

Sample request

BEARER JWT
POST /eden/verify HTTP/1.1 Host: ecbsvws.ssa.gov Accept: application/json Authorization: Bearer eyJraWQiOiJaRVNkb3Y2dWJSQVVQcndxSFlILXNzTEJrIiwiYWxnIjoiUlMyNTYifQ... exchangeID: XXXXXXXX externalTransactionID: XXXXXXXXXX eyJhbGciOiJSU0EtT0FFUC0yNTYiLCJlbmMiOiJBMjU2R0NNIiwia2lkIjoiZXd6bi1iSkh3Z1A5Q0VhX3p6V2cyN 185X3E2cnV4bXo0RzJnSVNYRU14MCJ9.Cb_kYnv3hm.sAALXJ1k tkwqMkMiIyHR4L61o9J668g..JlUYp1IXT4B59xg.S8eKQhVpvdKC9qn9q9igKQ

COMPONENT	SAMPLE JWE SNIPPET	NOTES

JWE Header	Decoded Value
	
	{
	"alg":"RSA-OAEP-256",
	"enc":"A256GCM",
	"kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
	}
		Base64 encoded
		
		This header can also contain other attributes like Key ID (“kid”) , JSON Web Key , JWK Set URL (jku) , (X.509 Certificate) x5c etc – and these are Registered Header Parameter Names

7 VERIFICATION SERVICE – REQUESTS AND RESPONSES

7.1 *Data Content for Request*

Description: This operation will verify an individual using the input SSN, Name, and Date of Birth against SSA's Master Files.

Produces: application/json

The following data must be transmitted in the Request to perform the SSN verification. Any records with erroneous information will not be processed and will be returned to the customer.

Request Parameters

Field Name	Description	Max Field Length	Field Type / Format	REQUIRED /OPTIONAL
REQUEST HEADER				
exchangeID	Provided by SSA after successful enrollment.	20	Alpha/ Numeric (A/N)	REQ
Authorization	A unique access token which is valid for 30 minutes. e.g. Bearer XXXXXXXXXXXXXXXXXXX **Refer to section 3.2 Obtaining Access token**	n/a	A/N	REQ
externalTransactionID	An optional identifier field, as generated by the user.	36	A/N	OPT (highly recommend-ed for trouble-shooting)
Content-Type	Identifying the content of the payload as JSON. Only accept "application/json".	n/a	A/N	REQ

accept	Identifying the content of the payload as JSON. Only accept "application/json".	n/a	A/N	REQ
REQUEST BODY (PER TRANSACTION)				
EIN	Employer Identification Number of the entity that obtained the number holder consent for matching in the verification service.	9	Numeric	REQ
REQUEST BODY (PER RECORD)				
externalSeqNumber	Sequence number of the record request sent by the external customer	10	Numeric	OPT; highly recommended for trouble-shooting
ssn	Numberholder's SSN No special characters or spaces allowed.	9	Numeric	REQ
dateOfBirth	Numberholder's Date of Birth Fomat = **MMDDYYYY** **MM** is *month*; enter two digit value (01 – 12) **DD** is *day*; enter two digits (01 – 31) **YYYY** is *year*; enter 4 digit value Use numeric characters only. Letters, hyphens, slashes, spaces or any other characters are not allowed.	8	Numeric	REQ
lastName	Number Holder's last name If the Last Name is longer than 20, enter the first 20 characters.	20	Alpha	REQ

	Must contain at least 1 character. Alphabetic characters only. Numbers, hyphens, slashes, or any other characters are not allowed and should be replaced with a space. Example: O'BRIEN should be entered as O BRIEN. Spaces are allowed. No suffixes (Jr., Sr., etc.)			
firstName	Number Holder's first name If the First Name is longer than 15, enter the first 15 characters. Must contain at least 1 character. Alphabetic characters only. Numbers, hyphens, slashes or any other characters are not allowed and should be replaced with a space. Example: O'BRIEN should be entered as O BRIEN. Spaces are allowed.	15	Alpha	REQ
middleName	Input middle name If supplied, and middle name is longer than 15, enter the first 15 characters. Alphabetic characters only. Numbers, hyphens, slashes or any other characters are not allowed and should be replaced with a space. Example: O'BRIEN should be entered as O BRIEN. Spaces are allowed.	15	Alpha	OPT
signatureType	Indicates Numberholder consent signature type. Valid values are:	1	Alpha	REQ

	"E" or "e"– customer provided electronic signature "W" or "w"– customer provided wet/ink signature			

7.2 *Data Content for Response*

The following data will be returned in the Response from the eCBSV Service to the client.

Records returned in a bulk request will be in the same order as the records submitted.

Response

Field Name	Max Field Length	Field Type / Format	Comments
RESPONSE HEADER			
externalTransactionID	36	A/N	From Request Header, if supplied
globalTransactionID	24	A/N	Generated by SSA
exchangeID	20	A/N	From JWT token
RESPONSE BODY			
externalSeqNumber	10	Numeric	From request body
verificationCode	1	Alpha	"**Y**" – verified (SSN data matches SSA's records) "**N**" – not verified (SSN data did not match SSA's records)
deathIndicator	1	Alpha	"**Y**" – deceased "**N**" – not deceased Blank – not checked *DI is only populated when verificationCode = Y)*
errorCode	4	A/N	Error Code at the transaction level

errorCodeDesc	100	A/N	Error code description at the transaction level
recordErrorCode	4	A/N	Error Code at the record level
recordErrorCodeDesc	100	A/N	Error code description at the record level
***reason**			
ssnMatch	1	Alpha	"T" - SSN matches SSA's records "F" - SSN does not match SSA's records
nameMatch	2	Alpha	"T" - Name matches SSA's records "F"- Name does not match SSA's records "NA" – not checked when ssnMatch = "F"
dobMatch	2	Alpha	"T" – DOB matches SSA's records "F" - DOB does not match SSA's records "NA" – not checked when ssnMatch = "F"
other	1	Alpha	"T" - In rare circumstances, SSA is unable to disclose the reason why an SSN did not verify. In these cases, SSA will provide a reason of "other". After verifying the input data is correct, the individual should be directed to visit an SSA field office to determine why their information was not able to be verified.

** Reason will either contain ssnMatch/nameMatch/dobMatch or other. If response contains 'other' parameter, it will not contain the ssnMatch/nameMatch/dobMatch.*

7.3 *eCBSV Error Codes and Exception Handling*

The eCBSV service will provide an Error Code with a corresponding http Code indicating that a run time exception occurred during the call, as explained below. Your account balance will not be decremented in the event that any of the following errors has occurred.

Error Code	Error Code Description	http Code
429	Too many requests. Exceeding requests per second limit*	**429**
4000	Exchange ID is required	**403**
4001	Exchange ID is invalid	**403**
4002	Your account is not in good standing	**403**
4003	Forbidden	**403**
8000	EIN is required	**400**
8001	EIN is invalid	**422**
8002	The Permitted Entity Certification is invalid	**422**
8003	Insufficient balance	**422**
8004	Bulk transaction: number of submitted records exceeded maximum**	**400**
8100	Input Date of Birth is invalid	**400**
8101	Signature type must be W or E	**400**
8103	Input SSN is invalid	**400**
8104	Input first name is invalid	**400**
8105	Input last name is invalid	**400**
8106	Input middle name is invalid	**400**
8107	Input external sequence number is invalid	**400**
8201	An error occurred – your account was not decremented***	**500**
8202	An error occurred – your account was not decremented***	**500**
8203	An error occurred – your account was not decremented***	**500**
8204	An error occurred – your account was not decremented***	**500**
n/a	External Sequence Number is invalid	**400**

** Every entity will have a throttling limit set on how many requests per second the entity is able to send.*

***Number of records allowed in a bulk transaction is 10*

****Please resubmit your transaction. If problem persists, contact eCBSV Help Desk (See Section 11 of this document)*

Your account balance may or may not be decremented in the event that the following error has occurred.

Error Code	Error Code Description	http Code

8300	A problem has occurred. Please contact eCBSV User Support	**500**

http Code	Description
200	OK
400	Bad request
401	Unauthorized
403	Forbidden
404	Not Found
405	Invalid method
422	Unprocessable entity
429	Too many requests
500	Internal server error
503	Service unavailable

7.4 *Sample Requests and Reponses*

Please refer to the Request and Response requirements in Section 7.1 and 7.2 of this document for proper tag names and formatting of the data.

SUBMITTING A SINGLE VERIFICATION TRANSACTION

```
{
  "ein":"912355220",
  "cvsRequestList":[
    {
```

```
        "externalSeqNumber":"1234567891",
        "ssn":"903526700",
        "dateOfBirth":"12041977",
        "firstName":"MICKEY",
        "lastName":"MOUSE",
         "middleName":"M",
        "additionalParams":{
          "signatureType":"E"
        }
      }
   ]
}
```

PRODUCES A SINGLE RESPONSE

```
{
  "errorCode":null,
  "errorCodeDesc":null,
  "cvsResponseList":[
    {
      "verificationCode":"Y",
      "verificationData":{
        "reason":{"dobMatch":"T","ssnMatch":"T","nameMatch":"T"},
         "deathIndicator":"N"
       },
       "recordErrorCode":null,
       "recordErrorCodeDesc":null,
       "cvsRequest":{
         "externalSeqNumber":"1234567891"
       }
    }
  ]
}
```

SUBMITTING A BULK VERIFICATION REQUEST

```
{
  "ein":"912355220",
  "cvsRequestList":[
    {
      "externalSeqNumber":"1234567891",
      "ssn":"903526700",
      "dateOfBirth":"12041977",
      "firstName":"MICKEY",
      "lastName":"MOUSE",
       "middleName":"M",
      "additionalParams":{
        "signatureType":"E"
      }
    },
```

```
    {
      "externalSeqNumber":"1234567892",
      "ssn":"912765604",
      "dateOfBirth":"03081976",
      "firstName":"DONALD",
      "lastName":"DUCK",
      "middleName":"",
      "additionalParams":{
        "signatureType":"E"
      }
    }
  ]
}
```

PRODUCES A BULK RESPONSE

```
{
  "errorCode": null,
  "errorCodeDesc": null,
  "cvsResponseList": [
     {
      "verificationCode": "Y",
      "verificationData": {
       "reason":{"dobMatch": "T","ssnMatch": "T","nameMatch":"T"},
       "deathIndicator": "N"
      },
      "recordErrorCode": null,
      "recordErrorCodeDesc": null,
      "cvsRequest": {
        "externalSeqNumber": "1234567891"
      }
    },
    {
      "verificationCode": "Y",
      "verificationData": {
       "reason":{"dobMatch": "T","ssnMatch": "T","nameMatch": "T"},
       "deathIndicator": "N"
      },
      "recordErrorCode": null,
      "recordErrorCodeDesc": null,
      "cvsRequest": {
        "externalSeqNumber": "1234567892"
     }
   }
  ]
  }
```

SAMPLE BULK REQUEST WITH MIX OF ERROR and VALID SSN RECORD


```
{
  "ein":"912355220",
  "cvsRequestList":[
    {
      "externalSeqNumber":"1234567890",
      "ssn":"903526700",
      "dateOfBirth":"12041977",
      "firstName":"",
      "lastName":"MOUSE",
      "middleName":"M",
      "additionalParams":{
        "signatureType":"E"
      }
    },
    {
      "externalSeqNumber":"1234567891",
      "ssn":"912765604",
      "dateOfBirth":"03081976",
      "firstName":"DONALD",
      "lastName":"DUCK",
      "middleName":"",
      "additionalParams":{
        "signatureType":"E"
      }
    }
  ]
}
```


SAMPLE BULK RESPONSE WITH MIX OF ERROR and VALID SSN RECORD


```
{
  "errorCode": null,
  "errorCodeDesc": null,
  "cvsResponseList": [
    {
      "verificationCode": null,
      "verificationData": null,
      "recordErrorCode": "8104",
      "recordErrorCodeDesc": "Input first name is invalid",
      "cvsRequest": {
        "externalSeqNumber": "1234567890"
      }
    },
    {
```

```
      "verificationCode": "Y",
      "verificationData": {
        "reason":{"dobMatch": "T","ssnMatch": "T","nameMatch": "T"},
         "deathIndicator": "N"
      },
      "recordErrorCode": null,
      "recordErrorCodeDesc": null,
      "cvsRequest": {
        "externalSeqNumber": "1234567891"
      }
     }
  ]
}
```


SUBMITTING A SINGLE VERIFICATION TRANSACTION THAT WILL NOT VERIFY #1


```
{
  "ein":"912355220",
  "cvsRequestList":[
    {
      "externalSeqNumber":"1234567891",
      "ssn":"945013760",
      "dateOfBirth":"11102009",
      "firstName":"DORA",
      "lastName":"EXPLORER",
      "middleName":"",
      "additionalParams":{
        "signatureType":"E"
      }
    }
  ]
}
```


PRODUCES A SINGLE RESPONSE NOT VERIFIED WITH REASON “OTHER” = T


```
{
  "errorCode": null,
  "errorCodeDesc": null,
  "cvsResponseList": [
    {
      "verificationCode": "N",
      "verificationData": {
      "reason":{"other": "T"},
      "deathIndicator": ""
      },
      "recordErrorCode": null,
      "recordErrorCodeDesc": null,
      "cvsRequest": {
```

```
      "externalSeqNumber": "1234567891"
    }
   }
  ]
 }
```


SUBMITTING A SINGLE VERIFICATION TRANSACTION THAT WILL NOT VERIFY #2


```
{
  "ein":"912355220",
  "cvsRequestList":[
    {
      "externalSeqNumber":"1",
      "ssn":"903526700",
      "dateOfBirth":"12041977",
      "firstName":"DAFFY",
      "lastName":"DUCK",
      "middleName":"M",
      "additionalParams":{
        "signatureType":"E"
      }
    }
  ]
}
```


PRODUCES A SINGLE RESPONSE NOT VERIFIED WITH REASON "NAMEMATCH" = F


```
{
  "errorCode":null,
  "errorCodeDesc":null,
  "cvsResponseList":[
    {
      "verificationCode":"N",
      "verificationData":{
        "reason":{"dobMatch":"T","ssnMatch":"T","nameMatch":"F"},
      "deathIndicator":""
    },
    "recordErrorCode":null,
    "recordErrorCodeDesc":null,
    "cvsRequest":{
      "externalSeqNumber":"1"
    }
   }
  ]
}
```

8 VERIFICATION SERVICE – EXTERNAL TESTING ENVIRONMENT

8.1 *Overview*

SSA will provide an External Testing Environment (ETE) for the eCBSV Service so that clients in development can connect to this test environment and perform Interface testing of their software with the eCBSV Service.

The ETE should not be used for high volume performance testing

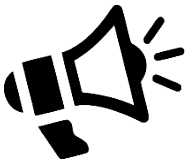
SSA recommends that the Requesting Party set up and configure an independent test environment to connect to SSA's ETE. The test environment must replicate the Production environment, including network connectivity, network security, and SSN Verifications to ensure proper handling of the responses returned to the client software. ETE is available every day of the week, except daily from 12:00 AM to 4:30 AM Eastern Time.

8.2 *Register for ETE*

Prior to using the ETE Verification Service, the entity is required to register for ETE in the eCBSV Customer Connection. In the eCBSV Customer Connection, entities will be required to enter the following information:

- OpenID Connect (OIDC) Issuer URL
- Optional: Dynamic Client Registration Authorization Header Credentials

Upon successful registration of your test environment, the entity will receive their ETE OAuth Client ID. Screen shots of the ETE Registration screens can be viewed in the eCBSV Screen Package (Appendix C).



NOTE: *At this time, registration may only be completed once. If you need to make updates, please email the eCBSVHelpDesk@ssa.gov.*

8.3 *Accessing eCBSV Service – External Testing Environment (ETE)*

SSA's eCBSV service will be available in ETE at the following endpoint:

ETE SERVICE ENDPOINT	ETE PING ENDPOINT
https://ecbsvwsete.ssa.gov/eden/verify	https://ecbsvwsete.ssa.gov/eden/ping

To consume the eCBSV service in ETE, the Entity's API client has to pass an access token in the HTTP Authorization header as a bearer token.

8.4 *ETE Test Data and Response Codes*

SSA is providing test data that generates given responses and messages in the test environment to replicate those generated by the eCBSV Service in the Production environment. Test data includes various Exchange IDs with associated EINs, and SSN data to simulate various use cases and error conditions. This data provides the flexibility to develop test scenarios to generate expected responses and specific error conditions related to the following:

- Access and connectivity
- SSN Verification response which generates a unique response code
- Response description and proper fault response in case of failures.

The test data provided is strictly for use in the External Testing Environment. ETE uses test SSNs that are impossible in the production environment. If you have any edits in your system to filter out impossible SSNs, you will need to disable those edits when testing in the ETE.

Test Data, Scenarios and Exchange IDs with the expected response codes and description of the messages generated by the eCBSV Service can be found in **Appendix E**.

8.5 *Obtaining Access Token (M2M Flow) - ETE*

The API client will obtain the access token from SSA's OAuth Authorization Server in ETE.

SSA's OAuth Authorization server in ETE will be available at the following endpoint:

SERVICE ENDPOINT
https://apiauthete.ssa.gov/mga/sps/oauth/oauth20/token

If you encounter an error when accessing this endpoint, refer to the **OAuth HTTP Error Codes** table located in the Section 4.3 of this document.

The JWT requirements in ETE are the same as those listed in the Production Section 6.5 above.

8.6 *Sample Request to ETE Endpoint*

SAMPLE ACCESS TOKEN REQUEST

POST /mga/sps/oauth/oauth20/token HTTP/1.1

Host: apiauthete.ssa.gov

Content-Type: application/x-www-form-urlencoded

Accept: application/json

'grant_type=client_credentials&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwtbearer&

client_assertion=eyJraWQiOiIyV0hQU...'

DECODED JWT IN CLIENT_ASSERTION FROM THE SAMPLE ABOVE


```
{
  "kid":"2WHP5YmLrVhNlWmxWe01xeNY5amIul-qHKnS955IIfY",
  "alg":"RS256"
}{
  "iss":"https://test.entity.com:7443/auth/realms/gcp",
  "sub":"780e78d2-007a-49af-b916-5cf36978705a",
  "aud":"https://apiauthete.ssa.gov/mga/sps/oauth/oauth20/token",
  "exp":1570725265,
  "nbf":1570120405,
  "iat":1570120465
}
```

SSA's OAuth Authorization server will verify the client assertion and issue the access token in JWT format.

ACCESS TOKEN RESPONSE


```
{
  "access_token":"eyJraWQiOiJaRVNkb3Y2dWJSQVEJrIiwiYWxnIjoiUlMyNTYifQ...",
  "token_type":"bearer",
  "expires_in":1800
}
```


The entity's API Client can now include this JWT Access Token in HTTP Header and call SSA's eCBSV Service in ETE (Ex: https://ecbsvwsete.ssa.gov/eden/verify)

ACCESS ECBSV SERVICE

GET /eCBSV HTTP/1.1

Host: ecbsvwsete.ssa.gov

Content-Type: application/json

Accept: application/json

Authorization: Bearer eyJraWQiOiJaRVNkb3Y2dWJSQVVQcndxSFlILXNzTEJrIiwiYWxnIjoiUlMyNTYifQ..."

exchangeID: XXXXXXXX

externalTransactionID: XXXXXXXXXX

exchangeID value is provided in the ETE Test Data (Section 8.3 above)

exchangeID HTTP header MUST be included.

The value for **externalTransactionID** HTTP header is the entity's transaction ID. This is optional. This ID is highly encouraged and significantly helps in correlating requests and troubleshooting.

8.7 *Encryption Requirements - ETE*

The JSON data request payload **must** be encrypted using JSON Web Encryption (JWE). SSA's public JSON Web Key (JWK) with details about the public key meant for encrypting the request payload will be available in the JSON Web Key Set (JWKS) endpoint.

JWKS ENDPOINT
https://apiauthete.ssa.gov/mga/sps/jwks

This will be indicated by the attribute/value "use":"enc" in the JWK.

EXAMPLE:

SSA'S JWKS ENDPOINT WITH SIGNING & ENCRYPTION JWK SAMPLE


```
{
  "keys":[
    {
      "kty":"RSA",
      "kid":"gCMwMdea-fQKPYjvnG0RftNb8JLCDpY1HUGDm0BuEH8",
      "use":"sig",
      "n":"vx3yHbw3fsowtlrz9Q82tvB2mPwCjWgUu3DhKHhv1quLmg5...kxAcB1UQ",
      "e":"AQAB"
    },
    {
      "kty":"RSA",
      "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2glSXEMx0",
      "use":"enc",
      "n":"mPeQt-WxaX9STiil4EZhgt2FFw9MbhlQLI4tHfeCPYnXX...ltSnSWh",
      "e":"AQAB"
    }
  ]
   }
```


Entities should include the **JWK** key id (**"use":"enc"**) in the JWE header.

Entities can use the following key management algorithm (alg) and content encryption algorithm (enc) combinations to encrypt the request payload as shown in the sample below. **"alg":"RSA-OAEP-256","enc":"A256GCM" is preferred.**

SUPPORTED ALGORITHM:


```
    {
      "alg":"RSA-OAEP",
      "enc":"A256CBC-HS512",
```

```
    "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
  }
  {
    "alg":"RSA-OAEP",
    "enc":"A256GCM",
    "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
  }
  {
    "alg":"RSA-OAEP-256",
    "enc":"A256CBC-HS512",
    "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
  }
  {
    "alg":"RSA-OAEP-256",
    "enc":"A256GCM",
    "kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
  }
```


SAMPLE REQUEST

BEARER JWT
POST /eden/verify HTTP/1.1 Host: ecbsvwsete.ssa.gov Accept: application/json Authorization: Bearer eyJraWQiOiJaRVNkb3Y2dWJSQVVQcndxSFlILXNzTEJrIiwiYWxnIjoiUlMyNTYifQ... exchangeID: XXXXXXXX externalTransactionID: XXXXXXXXXX eyJhbGciOiJSU0EtT0FFUC0yNTYiLCJlbmMiOiJBMjU2R0NNIiwia2lkIjoiZXd6bi1iSkh3Z1A5Q0VhX3p6V2cyN 185X3E2cnV4bXo0RzJnSVNYRU14MCJ9.Cb_kYnv3hm.sAALXJ1k tkwqMkMiIyHR4L61o9J668g..JlUYp1IXT4B59xg.S8eKQhVpvdKC9qn9q9igKQ

COMPONENT	SAMPLE JWE SNIPPET	NOTES

JWE Header	Decoded Value
	
	{
	"alg":"RSA-OAEP-256",
	"enc":"A256GCM",
	"kid":"ewzn-bJHwgP9CEa_zzWg27_9_q6ruxmz4G2gISXEMx0"
	}
		Base64 encoded ,
		
		This header can also contain other attributes like Key ID (“kid”) , JSON Web Key , JWK Set URL (jku) , (X.509 Certificate) x5c etc – and these are Registered Header Parameter Names

9 HEALTH PING

9.1 *Operation*

GET /ping

Description: The purpose of the PING operation is to indicate if the eCBSV system is UP or DOWN. Entities should call the PING operation when establishing initial connectivity and security. Entities may also use the PING operation during some planned SSA's maintenance activity to proactively check the overall eCBSV system status during that time.

NOTE: Entities should not be calling the PING operation for every Verify operation. A successful PING operation does not guarantee a successful Verify call. The eCBSV PING operation does not provide any details. Only the eCBSV Verify operation will provide a detailed error message of the failure.

PROD PING ENDPOINT	ETE PING ENDPOINT
https://ecbsvws.ssa.gov/eden/ping	https://ecbsvwsete.ssa.gov/eden/ping

9.2 *Parameters*

Type	Name	Description	REQ/OPT
Header	**Authorization**	Unique access token which is valid for 30 minutes. Ex. Bearer XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	REQ
Header	**Content-type**	Identifying the content of the payload as JSON. Only accept "application/json".	REQ
Header	**Accept**	Identifying the content of the payload as JSON. Only accept "application/json".	REQ
Header	**ExchangeID**	Provided by SSA after successful enrollment	REQ

9.3 *Responses*

When eCBSV system is UP and available, the PING response is as follows:

Field	Value	HTTP Code
status	UP	200

PING Response

```
{
  "status": "UP"
}
```

When eCBSV system is DOWN due to an unplanned outage, some error, and is not available, the PING response is as follows:

Field	Value	HTTP Code
errorCode **errorCodeDesc**	500 Internal Server Error	500

PING Response

```
{
  "errorCode": "500",
  "errorCodeDesc": " Internal Server Error"
}
```

When eCBSV system is DOWN due to a planned maintenance, the PING response during the maintenance time when eCBSV system is NOT available is as follows:

Field	Value	HTTP Code
errorCode **errorCodeDesc**	503 System Unavailable	503

PING Response


```
{
  "errorCode": "503",
  "errorCodeDesc": "System Unavailable"
}
```


When eCBSV system detects an unauthenticated or unauthorized request, the PING response is as follows:

Field	Value	HTTP Code
errorCode **errorCodeDesc**	401 Authentication Failure	401

PING Response


```
{
  "errorCode": "401",
  "errorCodeDesc": "Authentication Failure"
}
```

10 CONTACT US

10.1 *When to contact eCBSV Technical Support*

- After your in-house technical team has reviewed the issue(s) and they are unable to resolve.
- If you have received an error response in Production or the External Testing Environment (ETE) which, while you can work around it, causes you to take extra step(s) that you shouldn't have to.
- If the links provided in this document are not working.

10.2 *eCBSV Technical Support Contact Information*

- Email address: eCBSVHelpDesk@ssa.gov
- Hours of Operation:
  - 6:00 AM – 6:00 PM Eastern Time, Monday to Friday, excluding all Federal Holidays

10.3 *What is needed when contacting eCBSV Technical Support*

When you contact the eCBSV Help Desk, via email, please be prepare to provide the following information:

- ExchangeID
- Company's Name and EIN
- Domain Name
- OIDC URL
- External Transaction ID (optional)
- Date and time of issue
- A description of the problem (Example: "I can't successfully validate my OIDC Issuer URL")
  - If the problem can be reproduced:
    - List the steps you took to create it
    - Provide screen shot(s) of any error message(s) that is displayed
    - Include any additional supporting documentation
  - If the problem cannot be reproduced, i.e., occurs sporadically or inconsistently:
    - Describe the circumstances in which it occurred and the symptoms observed
    - Provide screen shot(s) of any error message(s) that is displayed
    - Include any additional supporting documentation.
- Please provide a point of contact:
  - Contact name,
  - Contact information (phone # and email),
  - An alternate contact person, if available.

11 CHANGE HISTORY

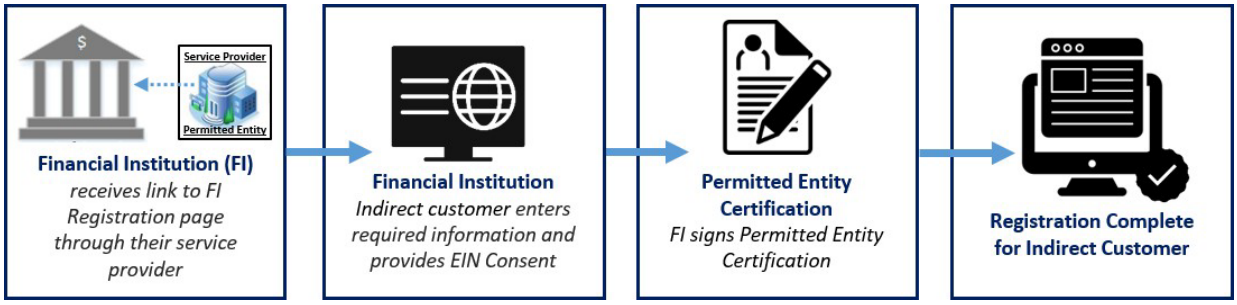
VERSION #	DATE	UPDATES
1.0	11/20/2020	Initial guide published
2.0	02/05/2021	Changes include: • Section 6.5 – Added helpful "Note" to encryption requirement section • Section 8 - Updates to ETE test instructions for "production-like testing" in ETE • Section 10 – Added description and response details to Health PING • Section 12.1 – Updates to OIDC Issuer URL Webpage Error Codes and Exception Handling • Appendix E – Added ETE Test Data
2.1	3/2/21	Changes include: • Section 3.8 – Added the role's scope value • Section 4.1 – Added the OIDC URL Validation Tool link.
2.2	3/5/21	Change include: • Section 4.2 – Added OIDC Validation Tool Screenshots
2.3	3/11/21	Change include: • Section 11.3 – Updated list of requirements when calling the helpdesk.
2.4	3/25/21 5/18/21	Changes include: • Updating document to meet 508 Accessibility requirements. • Updated OIDC Validation Tool Images
2.5	5/20/21	Changes include: • Removal of Availability and Performance section
3.0	**7/19/21**	**Expanded Roll Out – Go Live** Changes Include: • Section 3.6 – Added Registration note • Section 4.4 – Added Registration note • Section 5.2 and 5.3 – Added access clarification and link to eCBSV Customer Connection • Section 10.2 – Added phone line for Help Desk • Appendix C – Added eCBSV Screen Package
3.1	**02/17/22**	Changes include:

		• Section 7.4 – Updated samples with ETE test data from Appendix E • Section 10.2 – Updated Help Desk Hours • Appendix E, Table 4 - Updated scenario 3 for clarity; added scenarios 12-16
3.2	**7/20/22**	Changes include: • Section 5.2 - Expanded to provide more information on accessing the Customer Connection. • Section 3.8 - Removed roles value information • Section 6.3 - Changed jwtbearer to jwt-bearer (sample access token request table) • Section 7.3 - Updated footnote for 8201 and 8204.
3.3	**11/15/2023**	Changes include: • Section 4.4 – Added language to registration note regarding staging URL. • Section 10.2 – Removed contact us phone number and updated hours of operation.
3.4	**05/13/2025**	Changes include: • Section 7.2 – updated Response to include the Reason tags • Section 7.4 – updated the Sample Requests and Responses to include the reason • Appendix E – updated Table 1 Exchange ID and EIN combination for valid responses (ETEX00034/912355220). Added reason to the Options. Added Option #5.
3.5	**07/1/2025**	Changes include: • Section 8.1 – added ETE availability • Deleted Valid Certificates column from list of supported CAs table in appendix B • Deleted Link to Certificate Downloads column from list of supported CAs table in appendix B
3.6	**05/28/2026**	Changes include: • Section 7.3 – Updated 8107-Input external sequence number invalid error. • Appendix E - Updated error code for the external sequence number invalid error.

APPENDIX

Appendix A: Financial Institution Registration

The diagram displayed below provides a high-level overview of the steps required by a Financial Institution to register with eCBSV, which will allow them to work through a Service Provider to verify Numberholder information. For more information on Financial Institution Registration, go to https://www.ssa.gov/dataexchange/eCBSV/



NOTE - Screen shots of the Financial Institution Registration screen can be found in the eCBSV Screen Package (Appendix C)

Appendix B: Supported Certificate Authorities

The following is a list of supported CAs:

Name (Link)	Certification Practice Statement
Amazon Trust Services LLC	Statement
Digicert, Inc.	Statement
Entrust Datacard	Statement
GlobalSign	Statement
GoDaddy Inc	Statement
Network Solutions, LLC	Statement
SecureTrust	Statement
Sectigo	Statement
SSL.com	Statement

Appendix C: eCBSV Screen Package

Screens shots for eCBSV Entity Registration, eCBSV Financial Institution Registration, and the eCBSV Customer Connection can be viewed in the eCBSV Screen Package

Appendix D: Acronyms

The following is a list of acronyms used throughout this document

Acronyms	Definition
API	Application Programming Interface
eCBSV	Electronic Consent Based SSN Verification
EIN	Employer Identification Number
ETE	External Testing Environment
HTTP	Hypertext Transfer Protocol
IdP	Identity Provider
JSON	JavaScript Object Notation
JWE	JSON Web Encryption
JWT	JSON Web Token
JWK	JSON Web Key
JWKS	JSON Web Key Set
OAuth	Open Authorization
OIDC	OpenID Connect
REST	Representational State Transfer
SSL	Secure Socket(s) Layer
SSN	Social Security Number
URL	Uniform Resource Locator

Exchange IDs and EINs

Below is the list of Exchange IDs that can be used for eCBSV transactions in the External Testing Environment (ETE).

Note: Please use an Exchange ID in combination with the associated EIN (i.e.; ETEX00034/912355220) to avoid any unexpected results.

Table 1

Exchange ID	EIN	Error Code	Error Description	Notes
ETEX00034	912355220	N/A	N/A	Enough Balance
ETEX00011	912355211	4003	Forbidden	Client ID is not associated with Exchange ID
ETEX00012	912355201	4001	Exchange ID is invalid	Exchange ID not found in the table
ETEX00013	912355213	4002	Your account is not in good standing	Valid Exchange ID but Exchange is pending
ETEX00014	912355214	4002	Your account is not in good standing	Valid Exchange ID but Exchange is suspended
ETEX00015	912355215	4002	Your account is not in good standing	Valid Exchange ID but Exchange is terminated
ETEX00018	912355218	8002	The Permitted Entity Certification is invalid	
ETEX00019	912355219	8003	Insufficient Balance	

Test Data

Request Header

Authorization: JWT

Exchange ID: ETEX00034

External Transaction ID: Optional free form

Request Body

Option #1: match/no death

Submit any correctly formatted data in *Table 2*.

Verification code: Y

Death indicator: N

Reason

SSNMatch: T

NameMatch: T

DOBMatch: T

NOTE: The SSNs below are strictly for use in the External Testing Environment. They are impossible SSNs for the productions environment so if you have any edits in your system to filter out certain numbers you will need to disable those edits for the ETE.

Table 2

First Name	Middle Name	Last Name	SSN	DOB	Death Indicator
MICKEY	M	MOUSE	903526700	12/04/1977	N
DONALD		DUCK	912765604	03/08/1976	N
MINNIE	R	MOUSE	933887700	03/14/1990	N
ELMER		FUDD	941026505	09/04/1973	N
BUGS		BUNNY	942046305	12/31/1976	N
DAFFY		DUCK	944641208	11/07/1985	N
DAISY	A	DUCK	945109703	10/08/1989	N
FRED		FLINTSTONE	948887803	02/23/1983	N
BARNEY	D	RUBBLE	949545201	04/10/1978	N
WILMA		FLINTSTONE	971986104	04/28/1983	N
BETTY	MICHELLE	RUBBLE	987863809	09/17/1990	N
ROAD	Y	RUNNER	992622904	05/06/1984	N
INSPECTOR	S	GADGET	905728600	02/15/1972	N
WILE	E	COYOTE	905944409	06/02/1985	N
TWEETY	J	BIRD	929829103	01/06/2008	N
SYLVESTER	A	CAT	929927101	11/17/1996	N

SNOW	LEE	WHITE	933606203	01/24/2006	N
PORKY	ALEX	PIG	951926302	11/14/2004	N
GARFIELD		CAT	945477905	01/19/1992	N
ROBIN	A	HOOD	949817504	08/18/1960	N

Option #2: match/death

Submit any correctly formatted data that is in *Table 3*.

Verification code: Y

Death indicator: Y

Reason:

SSNMatch: T

NameMatch: T

DOBMatch: T

Table 3

First Name	Middle Name	Last Name	SSN	DOB	Death Indicator
OPTIMUS	K	PRIME	908727609	07/08/1911	Y
TASMANIAN	THE	DEVIL	908822208	10/18/1930	Y
MISS		PIGGY	923842200	07/08/1950	Y
JUDY	J	JETSON	925915904	07/10/1938	Y
RED	KELLY	RIDINGHOOD	965010501	09/18/1957	Y
WONDER		WOMAN	904942008	06/25/1930	Y
TINKER	ANNE	BELL	928784108	05/11/1924	Y
CAPTAIN		AMERICA	980924106	06/04/1969	Y
POWER	PUFF	GIRLS	919058708	01/31/1955	Y
PRINCESS		FIONA	920886407	01/08/1974	Y

Option #3: No Match with reason containing SSNMatch=F, NameMatch=NA, and DOBMatch=NA

Submit any correctly formatted data with an SSN ***not*** in *Table 2*, Table 3 or Table 5.

Verification code: N

Death Indicator: (blank)

Reason:

SSNMatch: F

NameMatch: NA

DOBMatch: NA

Option #4: errors

Table 4

	Error code	Error description	Scenario	Note
1	4000	Exchange ID is required	Submit a verification request with a missing or empty Exchange ID in the header.	This error is returned at the transaction level
2	8000	EIN is required	Submit a verification with a missing or empty EIN.	EIN must be present. This error is returned at the transaction level
3	8001	EIN is invalid	Submit a verification request with an EIN in the wrong format, not numeric, or does not match a valid test EIN.	The correct format of the EIN is 9 digits with no spaces and no dashes, and it must match a test EIN This error is returned at the transaction level.
4	8004	Bulk transaction: Number of submitted records exceeded maximum	Submit 1 verification request with more than 10 SSN records.	This error is returned at the transaction level.
5	8100	Input Date of Birth is invalid	Submit a verification request with a missing date of birth or a date of birth in the wrong format or not numeric.	The correct format of the date of birth is MMDDYYYY. This error is returned at the record level.
6	8101	Signature type must be W or E	Submit a verification request with a missing signature type or the wrong code.	The only correct signature type codes are W or w (for wet) and E or e (for electronic). This error is returned at the record level.
7	8103	Input SSN is invalid	Submit a verification request with a missing SSN or a SSN in the wrong format or not numeric.	The correct format of the SSN is 9 digits with no spaces and no dashes. This error is returned at the record level.
8	8104	Input First Name is invalid	Submit a verification request with a missing first name or name is greater than 15 characters.	This error is returned at the record level.
9	8105	Input Last Name is invalid	Submit a verification request with a missing last name or name that is greater than 20 characters.	This error is returned at the record level.

10	8106	Input Middle Name is invalid	Submit a verification request with a middle name that is longer than 15 characters.	Middle name can be empty or missing and, when entered should be between 1 character and 15 characters. This error is returned at the record level.
11	8107	Input External Sequence Number is invalid	Submit a verification with an external sequence number greater than 10 digits or not numeric.	The External Sequence Number can be empty or a number of up to 10 digits. This error is returned at the record level.
12	8002	The Permitted Entity Certification is invalid	Submit a verification request with an EIN that has an invalid Permitted Entity Certification	Refer to Table 1 in Appendix E for the appropriate test Exchange ID and EIN that matches this scenario. This error is returned at the transaction level.
13	8003	Insufficient balance	Submit a verification request with an Exchange ID that has an insufficient balance for transactions	Refer to Table 1 in Appendix E for the appropriate test Exchange ID and EIN that matches this scenario This error is returned at the transaction level.
14	4001	Exchange ID is invalid	Submit a verification request with an Exchange ID that is not valid	Refer to Table 1 in Appendix E for the appropriate test Exchange ID and EIN that matches this scenario This error is returned at the transaction level.
15	4002	Your account is not in good standing	Submit a verification request with an Exchange ID that has a pending, suspended, or terminated status	Refer to Table 1 in Appendix E for the appropriate test Exchange ID and EIN that matches this scenario This error is returned at the transaction level.
16	4003	Forbidden	Submit a verification request with a Client ID that is not associated with the Exchange ID	Refer to Table 1 in Appendix E for the appropriate test Exchange ID and EIN that matches this scenario

				This error is returned at the transaction level.

NOTE: If the Requesting Party receives a response code or failure not listed in this table, the Requesting Party must examine its own client software to diagnose the problem before reporting issue to SSA.

Option #5: No Match with reason containing other=T

Submit correctly formatted data in Table 5.

Verification code: N

Death Indicator: (blank)

Reason:

other: T

Table 5

First Name	Middle Name	Last Name	SSN	DOB	Death Indicator
DORA		EXPLORER	945013760	11/10/2009	