National Identity Exchange Federation

REST Services Profile

Version 1.0

July 31, 2018

Table of Contents

1. Target Audience and Purpose

This document specifies technical interoperability requirements for connection to operational endpoints that leverage the National Identity Exchange Federation (NIEF) and that adhere to the Representational State Transfer (REST) paradigm[1]. The target audience includes technical representatives of organizations that intend to participate in NIEF as Identity Provider Organizations (IDPOs), Service Provider Organizations (SPOs), Service Consumer Organizations (SCOs), Attribute Provider Organizations (APOs), or some combination of these roles.[2] It also includes vendors, contractors, and consultants who, as part of their project or product implementation, have a requirement to establish technical interoperability with NIEF endpoints.

This document focuses only on issues of technical interoperability. It does not cover governance, policy, or other nontechnical interoperability requirements. For more information about those topics, see [NIEF Bylaws] and [NIEF OPP]. In addition, this document focuses only on REST services. It does not address SOAP Web Services; see [NIEF S2S] for SOAP Web Services interaction profiles.

2. NIEF Identity Trust Framework and Terminology

This document is one component of the NIEF Identity Trust Framework. See [NIEF OPP] for more information about the full NIEF Identity Trust Framework.

This document contains language that uses technical terms related to federations, identity management, Web services, and other related technologies. To minimize confusion for readers, it is important that each technical term have a precise definition. Accordingly, all technical terms in this document are to be interpreted as described in [NIEF Terms], [OIDC Core], and [OAuth Core].

3. References

Table 1 and Table 2 contain a list of documents that pertain to the specifications and requirements described in this document (including components from the NIEF Identity Assurance Framework and industry standards).

Document References for NIEF Identity Assurance Framework Components	
Document ID	**Document Name and URL if Applicable**
NIEF Bylaws	NIEF Center Bylaws
NIEF OPP	NIEF Center Operational Policies and Procedures
NIEF S2S	NIEF Web Services System-to-System Profile
NIEF Terms	NIEF Terminology Reference
NIEF Trust	NIEF Cryptographic Trust Model

Table 1: Document References for NIEF Identity Assurance Framework Components

[1] See http://en.wikipedia.org/wiki/Representational_state_transfer for more information about REST.
2 See [NIEF Terms] for terminology related to various organizational and technical roles in NIEF.

Document References for Industry and Government Standards	
Document ID	**Document Name and URL**
FIPS 140-2	Federal Information Processing Standard (FIPS) Publication 140-2, *Security Requirements for Cryptographic Modules* December 3, 2002 http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
RFC 2119	Key Words for Use in RFCs to Indicate Requirement Levels Internet Engineering Task Force (IETF) Request for Comments (RFC) 2119 https://tools.ietf.org/html/rfc2119
OAuth Core	The OAuth 2.0 Authorization Framework IETF RFC 6749 http://tools.ietf.org/html/rfc6749
OAuth Bearer	The OAuth 2.0 Authorization Framework: Bearer Token Usage IETF RFC 6750 http://tools.ietf.org/html/rfc6750
JSON	JavaScript Object Notation (JSON) Data Interchange Format IETF RFC 7159 http://tools.ietf.org/html/rfc7159
JWT	JSON Web Token (JWT) IETF RFC 7519 https://tools.ietf.org/html/rfc7519
OAuth Assertions	Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants IETF RFC 7521 https://tools.ietf.org/html/rfc7521
OAuth SAML2	SAML 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants IETF RFC 7522 https://tools.ietf.org/html/rfc7522
OAuth JWT	JWT Profile for OAuth 2.0 Client Authentication and Authorization Grants IETF RFC 7523 https://tools.ietf.org/html/rfc7523
OAuth DCR	OAuth 2.0 Dynamic Client Registration Protocol IETF RFC 7591 https://tools.ietf.org/html/rfc7591
OIDC Core	OpenID Connect Core 1.0 http://openid.net/specs/openid-connect-core-1_0.html
OIDC Disc	OpenID Connect Discovery 1.0 http://openid.net/specs/openid-connect-discovery-1_0.html
OIDC DCR	OpenID Connect Dynamic Client Registration 1.0 http://openid.net/specs/openid-connect-registration-1_0.html
SAML2	Security Assertion Markup Language, Version 2.0 http://wiki.oasis-open.org/security
SAML2 Profiles	Profiles for the OASIS Security Assertion Markup Language (SAML) Version 2.0. OASIS Standard, March 15, 2005 http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
SAML2 Delegation	SAML 2.0 Condition for Delegation Restriction, Version 1.0 OASIS Committee Specification 01, November 15, 2009 http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation-cs-01.html
SOAP	W3C SOAP Note, May 8, 2000 http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
RFC4648	The Base16, Base32, and Base64 Data Encodings IETF RFC 4648 https://tools.ietf.org/html/rfc4648

Table 2: Document References for Industry Standards

4. Notation for Normative Content

This document contains both normative and non-normative content. Sections containing normative content are marked appropriately. In those sections, the key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" are to be interpreted as described in [RFC 2119].

5. Overview

This document specifies service interaction profiles for secure Representational State Transfer (REST) service use cases. REST is a set of service-oriented architecture (SOA) guidelines, principles, and constraints for designing services that are efficient, simple, and scalable; and REST describes the architecture of the World Wide Web. Web services and application programming interfaces (APIs) to which REST constraints are applied are often described as "RESTful". For a more thorough overview of REST, see the Wikipedia article on REST at http://en.wikipedia.org/wiki/Representational_state_transfer.

REST is often positioned in contrast to the Simple Object Access Protocol (SOAP), and while both can be used to specify Web services, there are many differences between the two. SOAP is a standardized specification, while REST is an architectural paradigm that has no standardized specification. SOAP is transport agnostic while REST assumes the use of HTTP. The WS-* suite of specifications extend SOAP with security features, while specifications such as OAuth and OpenID Connect were designed to provide Web security in a RESTful manner. This document provides RESTful alternatives to the SOAP-based Web Services interaction profiles defined in [NIEF S2S].

This Profile is concerned with secure federated identity and secure service interactions. API design and data payload-specific protections are out of scope.

Note that since the term "Web service" is often tied to the use of SOAP, this document uses the term "REST service" instead.

Note that all references to OAuth refer to OAuth 2.0 and all references to SAML refer to SAML 2.0.

6. NIEF REST Service Interaction Profiles

This section defines the service interaction profiles (SIPs) for REST services.

6.1 NIEF OpenID Connect Single Sign-On SIP

The NIEF OpenID Connect (OIDC) Single Sign-On (SSO) SIP profiles and constrains OpenID Connect 1.0 (see [OIDC Core]) for implementing RESTful SSO.

6.1.1 Motivating Use Case (Non-Normative)

This SIP derives its motivation from the need for an SSO alternative that lends itself to ease of implementation and integration with other NIEF REST Profiles. This SIP profiles the OIDC 1.0 SSO protocols defined in [OIDC Core]. OIDC has three different methods of accomplishing SSO, called Authorization Code Flow, Implicit Flow, and Hybrid Flow.

In the Authorization Code Flow, the OIDC Relying Party (RP) first redirects the user agent (UA) to obtain an authorization code, which is an OAuth authorization grant, from the Identity Provider's (IDP's) Authorization Endpoint. The RP then uses the authorization code to retrieve an ID token directly from the IDP's Token Endpoint. The RP can also retrieve an OAuth access token from the IDP's Token Endpoint; this access token can be used to retrieve supplement user attributes from the IDP's UserInfo Endpoint. In the Implicit Flow, the RP receives an ID token, and optionally an access token, from the IDP's Authorization Endpoint. The Hybrid Flow allows for optionality in how the Client receives the ID token and the access token.

In general, OIDC SSO consists of the following steps.

1. A user connects his/her UA to an OIDC RP in order to access a resource at that RP. The user is currently unauthenticated at the RP. The RP discovers the user's OIDC IDP. OIDC IDP discovery is out of scope for this SIP.[3] The RP sends an OIDC authentication request to the IDP by redirecting the UA with the request to the IDP's Authorization Endpoint.

2. The Authorization Endpoint processes the request. In this step, it authenticates the End-User and obtains consent from the user for releasing a user assertion about the End-User to the RP. The details of this are out of scope of this SIP.

3. Depending on the details in the metadata the Authorization Endpoint has about the RP and in the authentication request, upon successful authentication and consent, the Authorization Endpoint returns an authentication response that contains some combination of an authorization code, OIDC ID token, and access token. An OIDC ID token contains the verified user identifier, metadata about the authentication event, and optionally, attributes about the End-User. An access token can be used by the RP to retrieve supplemental user attributes from the IDP's UserInfo Endpoint.

4. The RP processes the response. In this step, if the RP received an OIDC ID token, then it validates that token, optionally retrieves supplemental attributes, and provides the End-User with access to the requested resource in accordance with the RP's access control policy.

[3] See Section **Error! Reference source not found.** for guidance on OIDC IDP discovery.

5. [Optional] If the RP received an authorization code, then it uses that code in a token request to the IDP's Token Endpoint to obtain an OIDC ID token and/or access token, as necessary.

6. [Optional] The Token Endpoint processes and validates the token request. In this step, the Token Endpoint authenticates the RP.

7. [Optional] The Token Endpoint returns a token response that contains an OIDC ID token, and if requested, an access token.

8. [Optional] The Client processes the token response, optionally retrieves supplemental attributes, and provides the End-User with access to the requested resource in accordance with the RP's access control policy.

Figure 1 depicts this SIP.

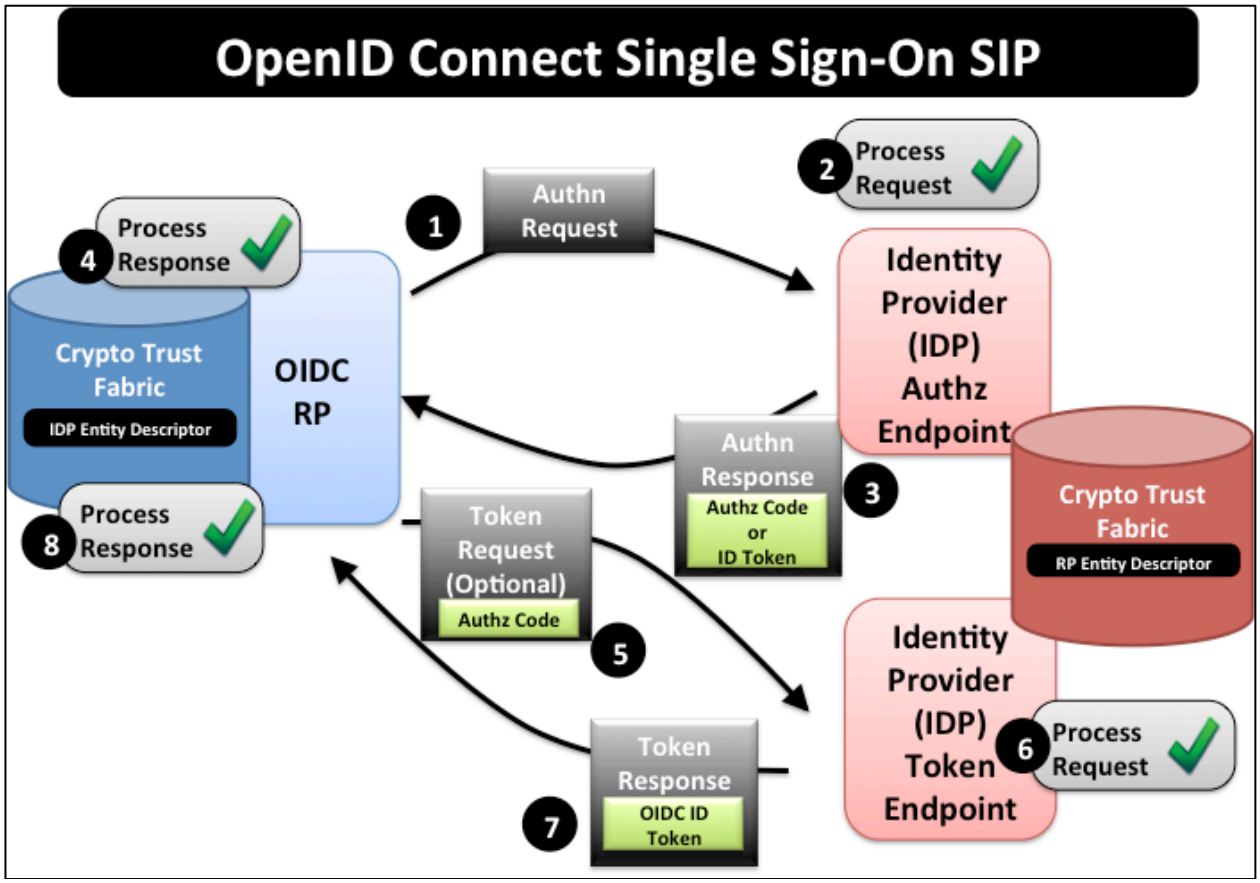

Figure 1: Diagram of the OpenID Connect Single Sign-On SIP

6.1.2 OpenID Connect Relying Party Requirements

1. The RP MUST conform to [OIDC Core] as a Relying Party.

2. When authenticating to the Token Endpoint of the IDP, the RP MUST authenticate in accordance with Section 7.1.

6.1.3 OpenID Connect Identity Provider Requirements

1. The OIDC IDP (IDP) MUST conform to [OIDC Core] as an OpenID Provider.

2. The Token Endpoint of the IDP MUST authenticate the RP in accordance with Section 7.1.

6.2 NIEF REST Consumer-Provider SIP

The NIEF REST Consumer-Provider SIP enables a REST Service Consumer (RSC) to connect to a REST Service Provider (RSP) to access a hosted resource, without acting directly on behalf of a user.

6.2.1 Motivating Use Case (Non-Normative)

This SIP consists of the following steps:

1. The RSC sends a resource request to the RSP over a TLS channel in which the RSC authenticates the RSP.

2. The RSP processes the resource request. In this step, the RSP authenticates the RSC and makes an access control decision for the request.

3. The RSP sends a resource response to the RSC.

4. The RSC processes the resource response.

Figure 2 depicts this SIP.

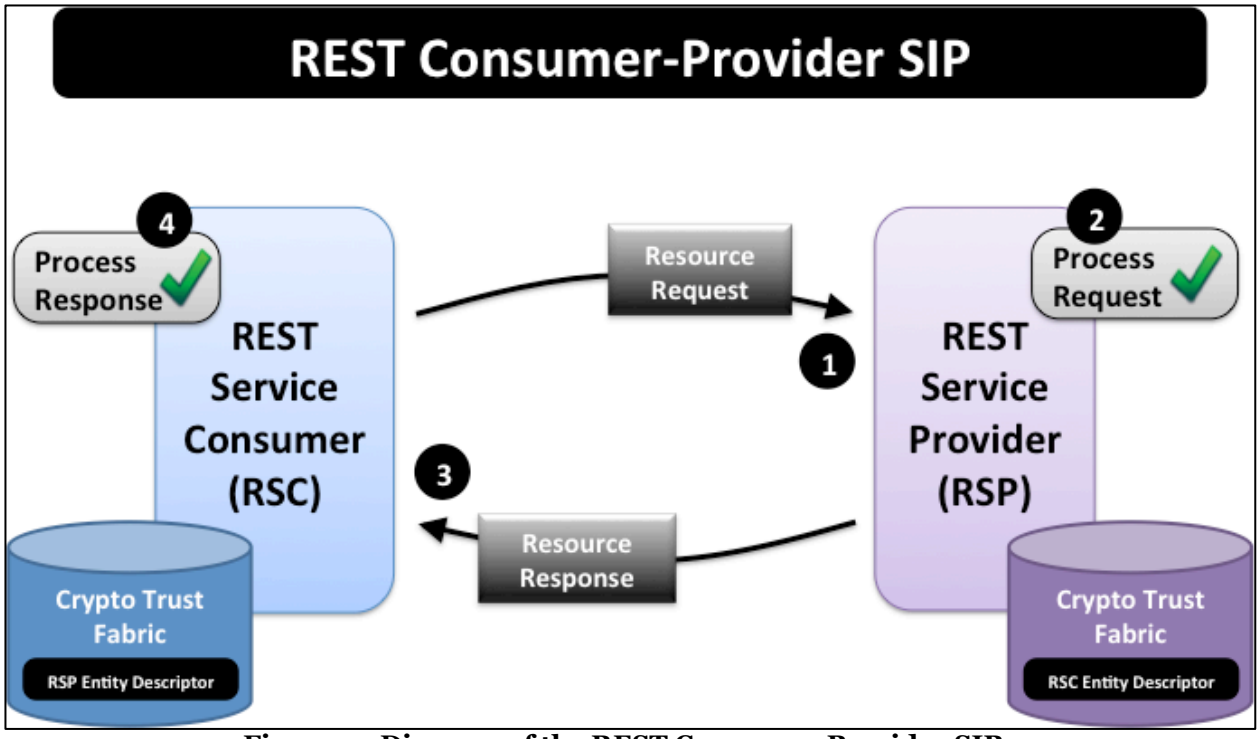

Figure 2: Diagram of the REST Consumer-Provider SIP

6.2.2 REST Service Consumer Requirements

1. The RSC MUST communicate with the RSP via HTTP over TLS.
2. The RSC MUST authenticate the RSP via its TLS server certificate.
3. The RSC MUST verify trust in the authenticated RSP.
4. The RSC MUST authenticate to the RSP via TLS client certificate authentication.

6.2.3 REST Service Provider Requirements

1. The RSP MUST expose its resources via HTTP over TLS.
2. The RSP MUST authenticate the RSC via TLS client certificate authentication.
3. The RSP MUST verify trust in the authenticated RSC.

6.3 NIEF REST Single Sign-On Consumer-Provider SIP

The NIEF REST Single Sign-On Consumer-Provider SIP enables a REST Service Consumer (RSC), while acting on behalf of an end-user, to connect to a REST Service Provider (RSP) to access a hosted resource, where the end-user is authenticated via a single sign-on mechanism.

6.3.1 Motivating Use Case (Non-Normative)

This SIP is useful in scenarios where the RSP can act as a single sign-on client to authenticate the end-user, the RSP needs to authenticate the RSC, and the RSC can act on behalf of the end-user by functioning as the end-user's HTTP user-agent.

In this SIP, the RSC must either deliver a user assertion to the RSP from a trusted IDP or deliver session tokens that have been previously provided by the RSP. The RSP uses the user assertion or session tokens to establish a user-authenticated session. When the RSC delivers a new user assertion to the RSP, the RSP then may supply new session tokens to the RSC. In addition, the RSP must authenticate the RSC over this session. Then, the RSC submits HTTP resource requests on behalf of the end-user over the session, and the RSP may make authorization decisions based on attributes about the user, RSC, and resource request.

This SIP relies on single sign-on (SSO) for the RSP to obtain the user assertion. Specifically, this SIP only allows the use of the NIEF Web Browser User-to-System Profile (SAML SSO), which is defined in [NIEF U2S], or the NIEF OpenID Connect (OIDC) SSO SIP, which is defined in Section 6.1 of this document. These SSO profiles allow SP/RP-initiated or IDP-initiated SSO transactions.

Figure 3 depicts the REST Single Sign-On Consumer-Provider SIP.

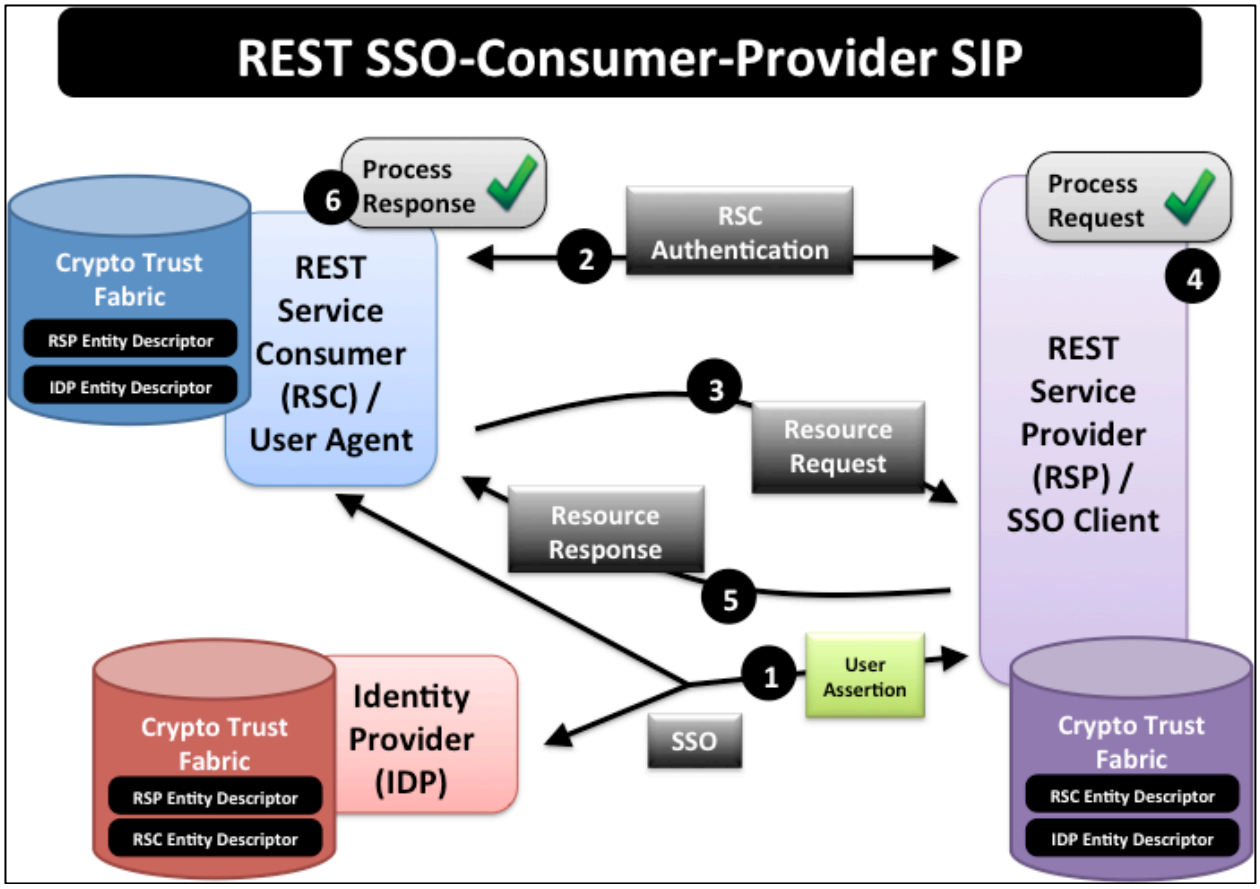

Figure 3: Diagram of the REST Single Sign-On Consumer Provider SIP

6.3.2 REST Service Consumer Requirements

1. The RSC MUST be able to facilitate at least one of the following SSO profiles as an HTTP user-agent.

    a. NIEF Web Browser User-to-System Profile

    b. NIEF OpenID Connect SSO SIP

2. The RSC MUST communicate with the RSP via HTTP over TLS.

3. The RSC MUST authenticate the RSP via the RSP's TLS server certificate.

4. The RSC MUST verify trust in the authenticated RSP.

5. The RSC MUST authenticate to the RSP via TLS client certificate authentication.

6.3.3 REST Service Provider Requirements

1. The RSP MUST meet at least one of the following SSO requirements.

    a. Conform to the NIEF OpenID Connect SSO SIP as a Relying Party (RP).

    b. Conform to the NIEF Web Browser User-to-System Profile as a Service Provider (SP).

2. The RSP MUST expose its resources via HTTP over TLS.

3. The RSP MUST authenticate the RSC via TLS client certificate authentication.

4. The RSP MUST verify trust in the authenticated RSC.

5. Upon initiation of a TLS channel with an RSC, the RSP MUST authenticate the end-user using one of the following methods.

    a. Conduct SSO via either the NIEF OpenID Connect SSO SIP or the NIEF Web Browser User-to-System Profile. After completion of the SSO process, the RSP MAY set a session cookie with the RSC.

    b. Receive a session cookie from the RSC that was previously issued by the RSP.

6.4 NIEF REST Delegated-Consumer-Provider SIP

The NIEF REST Delegated-Consumer-Provider SIP enables a REST Service Consumer (RSC), while acting on behalf of a user, to connect to a REST Service Provider (RSP) to access a hosted resource, where the user is identified by a delegated user assertion.

6.4.1 Motivating Use Case (Non-Normative)

This SIP introduces a user authentication event and user assertion into the REST service transaction. In this SIP, the RSC is also a Web portal, and it performs a REST service transaction with an RSP on behalf of the user. It consists of the following steps:

1. The RSC, acting on behalf of a user, obtains a delegated user assertion from the user's IDP. The delegated user assertion must be an OIDC ID token or SAML assertion. How this happens is out of scope for this SIP. The REST Assertion Delegate Service (ADS) SIP, defined in Section 6.8 can be used by the RSC to obtain a delegated user assertion.

2. The RSC sends a resource request to the RSP over a TLS channel in which the RSC authenticates the RSP. The resource request contains the delegated user assertion.

3. The RSP processes the resource request. In this step, the RSP authenticates the RSC and makes an access control decision on the request. This decision may

be based on attributes about the RSC and about the user identified by the delegated user assertion.

4. The RSP returns a resource response to the RSC.

5. The RSC processes the resource response.

Figure 4 depicts the REST Delegated-Consumer-Provider SIP.

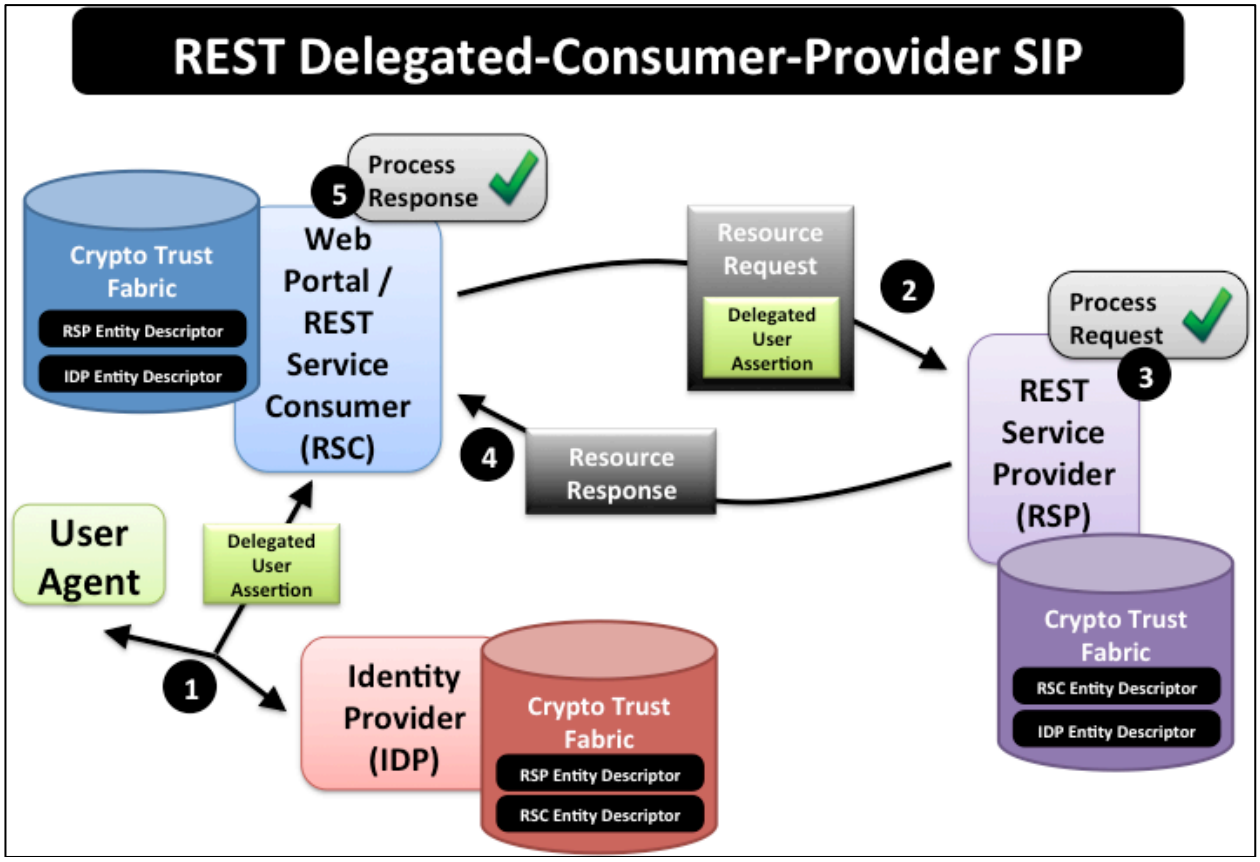

Figure 4: Diagram of the REST Delegated-Consumer-Provider SIP

6.4.2 REST Service Consumer Requirements

1. The RSC MUST conform to [OAuth Core] as an OAuth Client.

2. The RSC MUST acquire a delegated user assertion from a trusted IDP of the End-User that denotes to the RSP that the RSC is acting on behalf of the End-User. This delegated user assertion MUST be either an OpenID Connect (OIDC) ID token in accordance with Section 2 of [OIDC Core] that is signed and serialized using the JWS Compact Serialization in accordance with [JWS], or a SAML assertion in accordance with Section 7.2. The RSC is NOT required to verify that the acquired delegated user

assertion conforms to these requirements; however, the RSC MUST ensure that the IDP that issues the assertion is aware that it needs to issue a conforming assertion.[4]

3. The RSC MUST connect to the target resource via TLS.

4. When establishing a TLS channel, the RSC MUST authenticate the RSP via the RSP's TLS server certificate.

5. The RSC MUST verify trust in the RSP.

6. The RSC MUST use the acquired user assertion as a bearer token in the resource request it sends to the RSP, in accordance with [OAuth Bearer].

6.4.3 REST Service Provider Requirements

1. The RSP MUST conform to [OAuth Core] as an OAuth Resource Server.

2. The RSP MUST expose its resources via TLS.

3. The RSP MUST perform the validation steps in the following sub-items on the OAuth access token that it receives from the RSC.

    a. Verify that the access token is a valid OAuth bearer access token as defined in Section 7 of [OAuth Core] and in [OAuth Bearer].

    b. Verify that the access token is an OIDC ID token as defined in Section 2 of [OIDC Core] that has been signed and serialized with the JWS Compact Serialization in accordance with [JWS], or an SAML assertion that conforms to Section 7.2.

    c. Validate the digital signature.

    d. Verify that the signature certificate is associated with a trusted IDP.

    e. Verify that the RSP is identified in the audience of the token.

    f. Verify that the timestamp on the access token is not too far in the past according to RSP policy.

6.5 NIEF REST Consumer-Authorizer SIP

[4] The assertions issued by the IDP may be encrypted using an encryption key of the RSP; in this case, the RSC will not be able inspect the contents of the assertion. The RSC can use a protocol such as the NIEF Assertion Delegate Service (ADS) SIP to meet this requirement.

The NIEF REST Consumer-Authorizer SIP enables a REST Service Consumer (RSC) to obtain an authorization token from a REST Authorization Service (AS) to use to submit a pre-authorized resource request to a REST Service Provider (RSP), without acting directly on behalf of an End-User.

6.5.1 Motivating Use Case (Non-Normative)

This SIP addresses a scenario in which the RSP does not make its own authorization decisions, and must be accessed with an authorization token that has been issued by an Authorization Service (AS). In this SIP, the RSC is not acting on behalf of an End-User. It consists of the following steps:

1. The RSC sends an authorization token request to the AS over a TLS channel in which the RSC authenticates the AS. The authorization token request identifies the target RSP and may contain information about the requested access in the OAuth "scope" parameter.

2. The AS processes the authorization token request. In this step, the AS authenticates the RSC. Also, the AS uses information in the authorization token request to make an authorization decision about what resources the RSC can access at the RSP. This decision may be based on attributes about the RSC.

3. Upon successful authorization, the AS returns an authorization token response that contains an authorization token that the RSC can use to send authorized resource requests to the RSP.

4. The RSC processes the authorization token response and extracts the authorization token.

5. The RSC sends a resource request, which includes the authorization token, to the RSP. In this step, the RSC initiates establishment of a TLS channel with the RSP in which the RSC authenticates the RSP.

6. The RSP processes the resource request. In this step, the RSP validates the authorization token. Also, the RSP uses information in the authorization token to make an access control decision on the resource request.

7. The RSP returns a resource response to the RSC.

8. The RSC processes the resource response.

Figure 5 depicts this SIP.

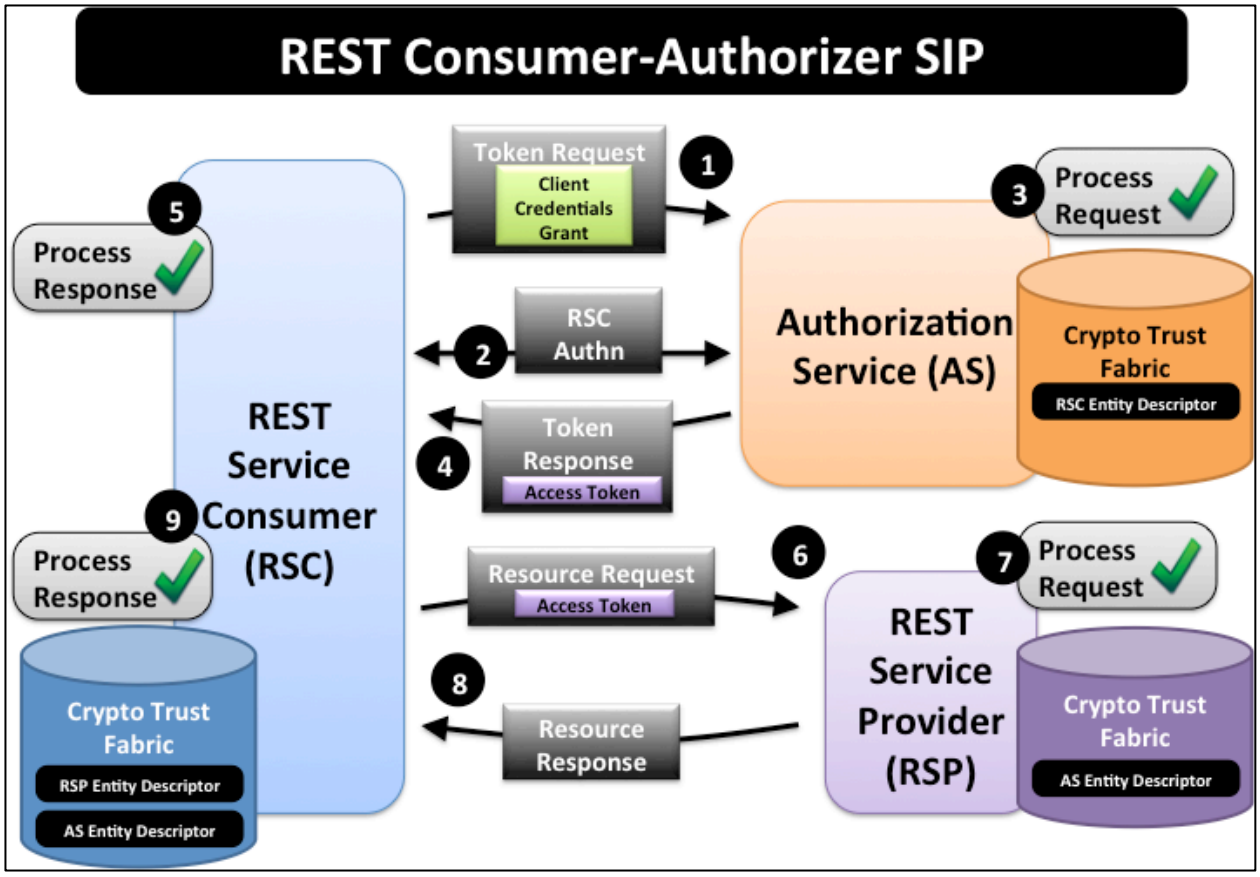

Figure 5: Diagram of the REST Consumer-Authorizer SIP

6.5.2 REST Service Consumer Requirements

1. The RSC MUST conform to the requirements in Section 7.3.1.

2. The RSC MUST use the OAuth Client Credentials Authorization Grant in accordance with Section 4.4 of [OAuth Core].

6.5.3 Authorization Service Requirements

1. The AS MUST conform to the requirements in Section 7.3.2.

2. The AS MUST verify that the token request it receives uses the Client Credentials Authorization Grant in accordance with Section 4.4 of [OAuth Core].

6.5.4 REST Service Provider Requirements

1. The RSP MUST conform to the requirements in Section 7.3.3.

6.6 NIEF REST Single Sign-On Consumer-Authorizer SIP

The NIEF REST SSO Consumer-Authorizer SIP enables a REST Service Consumer (RSC), while acting on behalf of an End-User, to obtain an authorization token from a REST Authorization Service (AS) to use to submit a pre-authorized resource request to a REST Service Provider (RSP), where the AS authenticates the user via a single sign-on mechanism.

6.6.1 Motivating Use Case (Non-Normative)

This SIP addresses the scenario where an RSC acts on behalf of the end-user, the RSP relies on a REST AS to make authorization decisions, and the AS can act as a single sign-on client to authenticate the end-user. In this SIP, the RSC may function as the user-agent or rely on an external user-agent. RSC authentication by the AS is optional.

This SIP combines OAuth with SSO. The RSC acts as an OAuth Client and can use the OAuth Implicit Flow or OAuth Authorization Code Flow to obtain an OAuth access token from the AS. The AS acts as an OAuth Authorization Server, and the RSP acts as an OAuth-Protected Resource Server.

This SIP consists of the following steps.

1. The RSC sends an authorization request to the AS via redirecting the User-Agent.

2. The AS authenticates the End-User by facilitating SSO via the user-agent. This SIP supports the NIEF OpenID Connect (OIDC) SSO SIP and the NIEF Web Browser User-to-System Profile (SAML SSO).

3. After end-user authentication, the AS makes an authorization decision on whether to allow the RSC to act on behalf of the end-user to make the requested access. Also, the AS may obtain consent from the end-user for the requested access.

4. Upon successful authorization and consent, the AS redirects the user-agent back to the RSC with either an access token or an authorization code. If the RSC receives an authorization code, then it sends a token request to the Token Endpoint of the AS to exchange the authorization code for an access token.

5. The RSC sends resource requests to the RSP, passing the access token with the request.

6. The RSP validates the request and the access token, provides access according to the access token.

7. The RSP returns an appropriate resource response.

8. The RSC processes the resource response.

Figure 6 depicts this SIP.

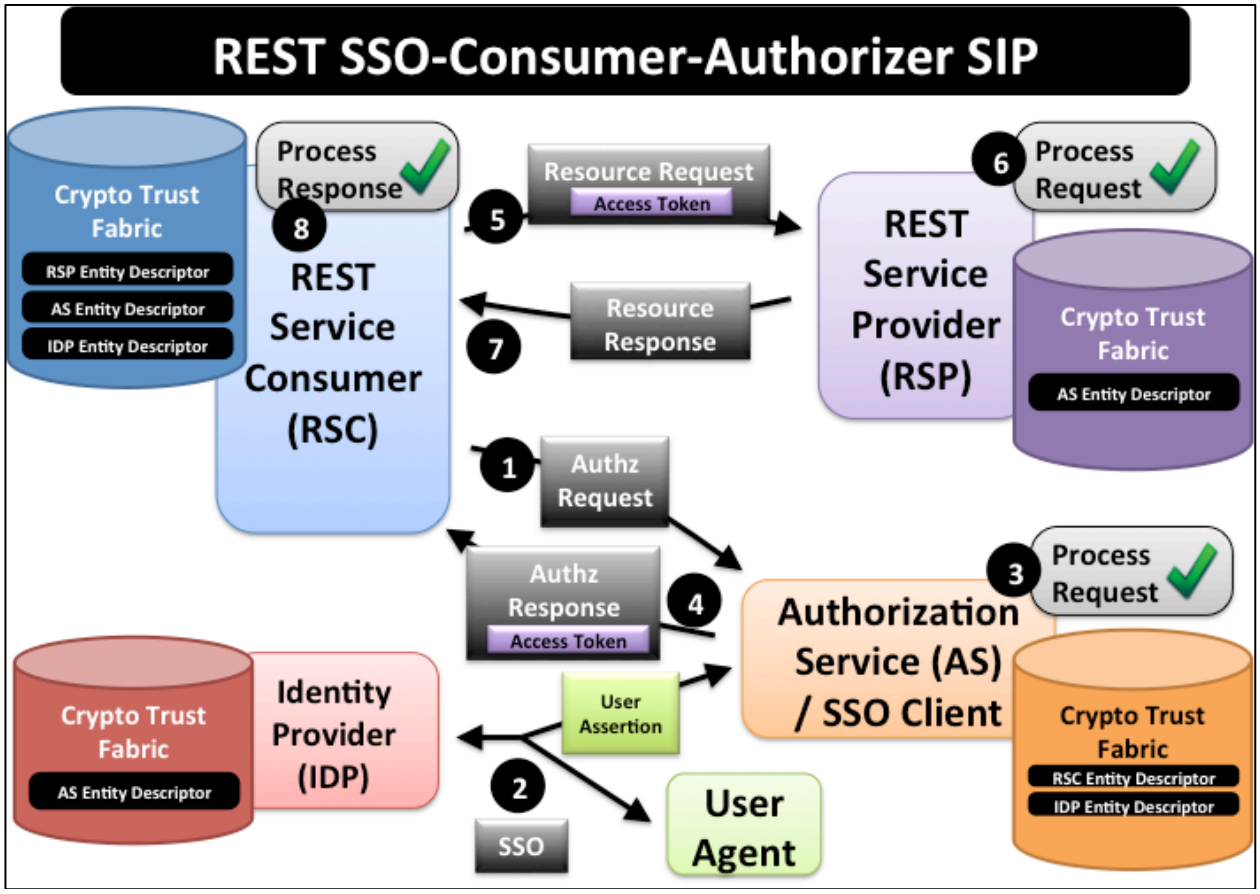

Figure 6: Diagram of the REST SSO Consumer-Authorizer SIP

6.6.2 REST Service Consumer Requirements

1. The RSC MUST conform to the requirements in Section 7.3.1.

2. The RSC MUST use the OAuth Implicit Flow as defined in Section 4.2 of [OAuth Core], or the OAuth Authentication Code Flow as defined in Section 4.1 of [OAuth Core].

6.6.3 Authorization Service Requirements

1. The AS MUST conform to the requirements in Section 7.3.2.

2. The AS MUST meet at least one of the following SSO requirements.

    a. Conform to the NIEF OpenID Connect SSO SIP as a Relying Party (RP).

    b. Conform to the NIEF Web Browser User-to-System Profile as a Service Provider (SP).

3. The AS MUST support either the OAuth Implicit Flow as defined in Section 4.2 of [OAuth Core], or the OAuth Authorization Code Flow as defined in Section 4.1 of [OAuth Core], or both.

4. The Authorization Endpoint of the AS MUST use one of the following methods to authenticate the End-User.

    a. Conduct SSO via either the NIEF OpenID Connect SSO SIP or the NIEF Web Browser User-to-System Profile. After completion of the SSO process, the AS MAY set a session cookie with the User-Agent.

    b. Receive a session cookie from the User-Agent that was previously issued by the AS.

6.6.4 REST Service Provider Requirements

1. The RSP MUST conform to the requirements in Section 7.3.3.

6.7 NIEF REST Delegated-Consumer-Authorizer SIP

The NIEF REST Delegated-Consumer-Authorizer SIP enables a REST Service Consumer (RSC), which is acting on behalf of an End-User, to obtain an authorization token from a REST Authorization Service (AS) to use to submit a pre-authorized resource request to a REST Service Provider (RSP), where the user is identified by a delegated user assertion.

6.7.1 Motivating Use Case (Non-Normative)

This SIP addresses a scenario in which the RSP does not make its own authorization decisions, and must be accessed with an authorization token that has been issued by an Authorization Service (AS). In this SIP, the RSC is acting on behalf of an End-User. It consists of the following steps:

1. The RSC, acting on behalf of an End-User, obtains a delegated user assertion from the user's IDP that can be provided to the AS. The assertion may be a delegated SAML assertion or an OIDC ID token. How the RSC obtains the assertion is out of scope for this SIP. However, it may use the REST Assertion Delegate Service (ADS) SIP, which is defined in Section 6.8, to accomplish this.

2. The RSC sends an OAuth token request to the Token Endpoint of the AS over a TLS channel in which the RSC authenticates the AS. The token request identifies the target RSP and may contain information about the requested access in the OAuth "scope" parameter. Also, the RSC includes the delegated user assertion obtained in step 1 in the token request.

3. The token endpoint processes the token request. In this step, the token endpoint authenticates the RSC. Also, the token endpoint uses information in

the token request to make an authorization decision about what resources the RSC can access at the RSP. This decision may be based on attributes about the RSC as well as attributes in the delegated assertion.

4. Upon successful authorization, the AS returns a token response to the RSC. The token response contains an OAuth access token that the RSC can use to send authorized resource requests to the RSP.

5. The RSC processes the token response and extracts the access token.

6. The RSC sends a resource request, which includes the access token, to the RSP. In this step, the RSC initiates establishment of a TLS channel with the RSP in which the RSC authenticates the RSP.

7. The RSP processes the resource request. In this step, the RSP validates and may honor the access token and process the request accordingly.

8. The RSP returns a resource response to the RSC.

9. The RSC processes the resource response.

Figure 7 depicts this SIP.

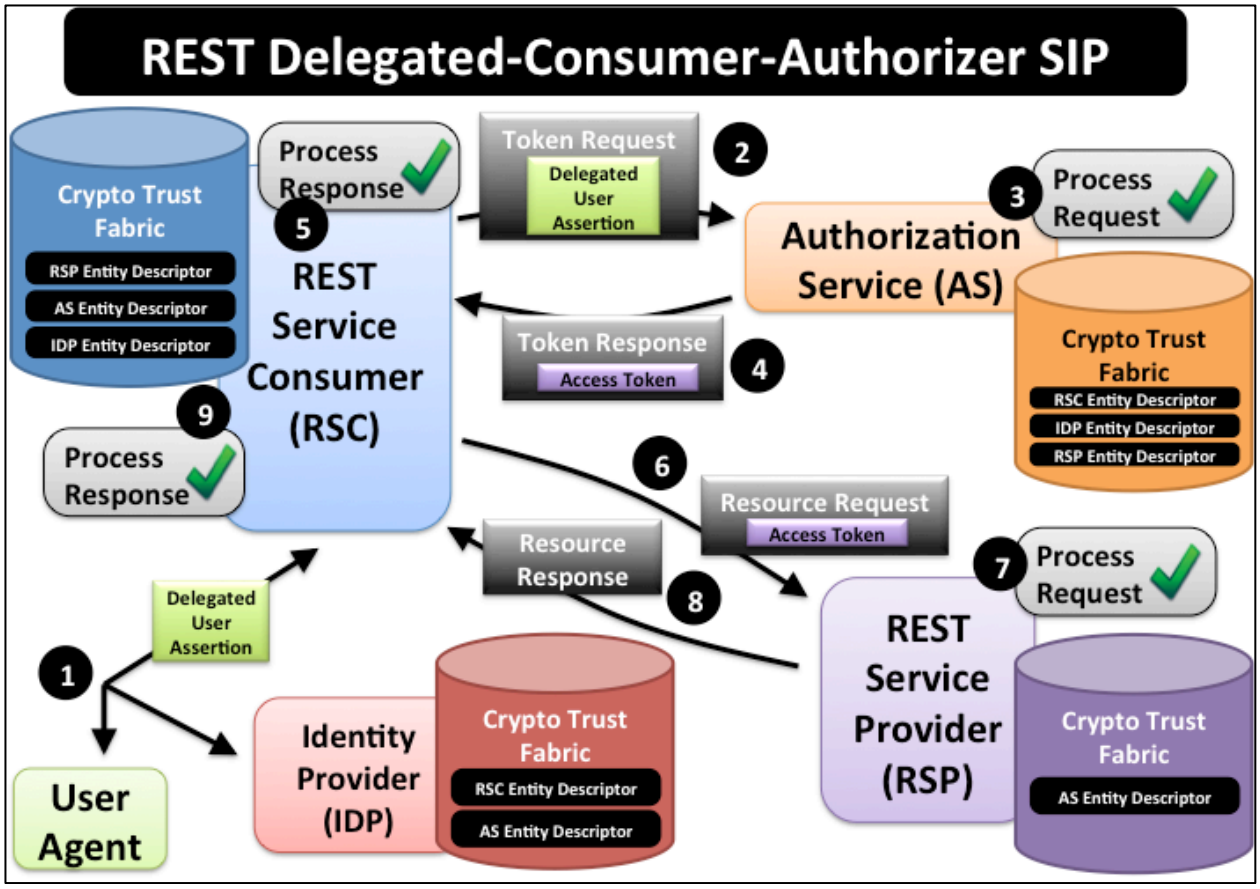

Figure 7: Diagram of the REST Delegated-Consumer-Authorizer SIP

6.7.2 REST Service Consumer Requirements

1. The RSC MUST conform to the requirements in Section 7.3.1.

2. The RSC MUST acquire a delegated user assertion, from a trusted IDP, that denotes to the RSP that the RSC is acting on behalf of the End-User. This assertion MUST be either an OpenID Connect (OIDC) ID token in accordance with Section 2 of [OIDC Core] that is signed and serialized using the JWS Compact Serialization in accordance with [JWS], or a SAML assertion that conforms to Section 7.2. The RSC is NOT required to verify that the acquired user assertion conforms to these requirements; however, the RSC MUST ensure that the IDP that issues the assertion is aware that it needs to issue a conforming assertion.[5]

3. The RSC MUST present the delegated user assertion as the OAuth authorization grant to the token endpoint of the AS in accordance with either [OAuth JWT] or [OAuth SAML2].

[5] The assertions issued by the IDP may be encrypted using an encryption key of the RSP; in this case, the RSC will not be able inspect the contents of the assertion. The RSC can use a protocol such as the NIEF Assertion Delegate Service (ADS) SIP to meet this requirement.

6.7.3 Authorization Service Requirements

1. The AS MUST conform to the requirements in Section 7.3.2.

2. The Token Endpoint of the AS MUST accept from the RSC an authorization grant that either conforms to [OAuth JWT] or conforms to [OAuth SAML].

3. The Token Endpoint of the AS MUST perform the validation steps in the following sub-items, in addition to the validation steps in [OAuth JWT] or [OAuth SAML], to validate the assertion in the authorization grant.

    a. The Token Endpoint MUST verify that the signing certificate of the assertion is associated with a trusted IDP.

    b. The Token Endpoint MUST verify that the RSC is an authorized party of the assertion.

    c. The Token Endpoint MUST be a member of the audience specified by the assertion.

6.7.4 REST Service Provider Requirements

1. The RSP MUST conform to the requirements in Section 7.3.3.

6.8 NIEF REST Assertion Delegate Service SIP

The NIEF REST Assertion Delegate Service SIP enables a REST Service Consumer (RSC) to obtain a delegated user assertion from an Assertion Delegate Service (ADS) for use at a REST Service Provider. The RSC may redirect the User Agent to the ADS so the End-User can provide in-band consent, or if the RSC has been previously issued a user assertion for the End-User, then it may exchange that assertion for the delegated assertion via directly communicating with the ADS.

6.8.1 Motivating Use Case (Non-Normative)

Some SIPs, such as the REST Delegated-Consumer-Provider SIP and the REST Delegated-Consumer-Authorizer SIP, require an RSC to obtain a delegated user assertion for use at an RSP. This SIP provides a method for an RSC to obtain such an assertion from a REST Assertion Delegate Service (ADS). The assertion may be a SAML assertion or an OIDC ID token. The ADS obtains in-band or out-of-band consent from the End-User, based on the delegation request sent by the RSC. To accomplish in-band consent, the RSC needs to be able to redirect the End-User's User Agent (UA) to the ADS. To accomplish out-of-band consent, the RSC needs to have previously been issued a user assertion for the End-User.

This SIP is based on OIDC, with the RSC acting as an OIDC Relying Party (RP) and the ADS acting as an OpenID Provider. The ADS provides a delegated user assertion to the RSC in a "delegated_assertion" parameter defined in this SIP.

Figure 8 depicts this SIP with the use of in-band consent, and Figure 9 depicts this SIP with the use of out-of-band consent.

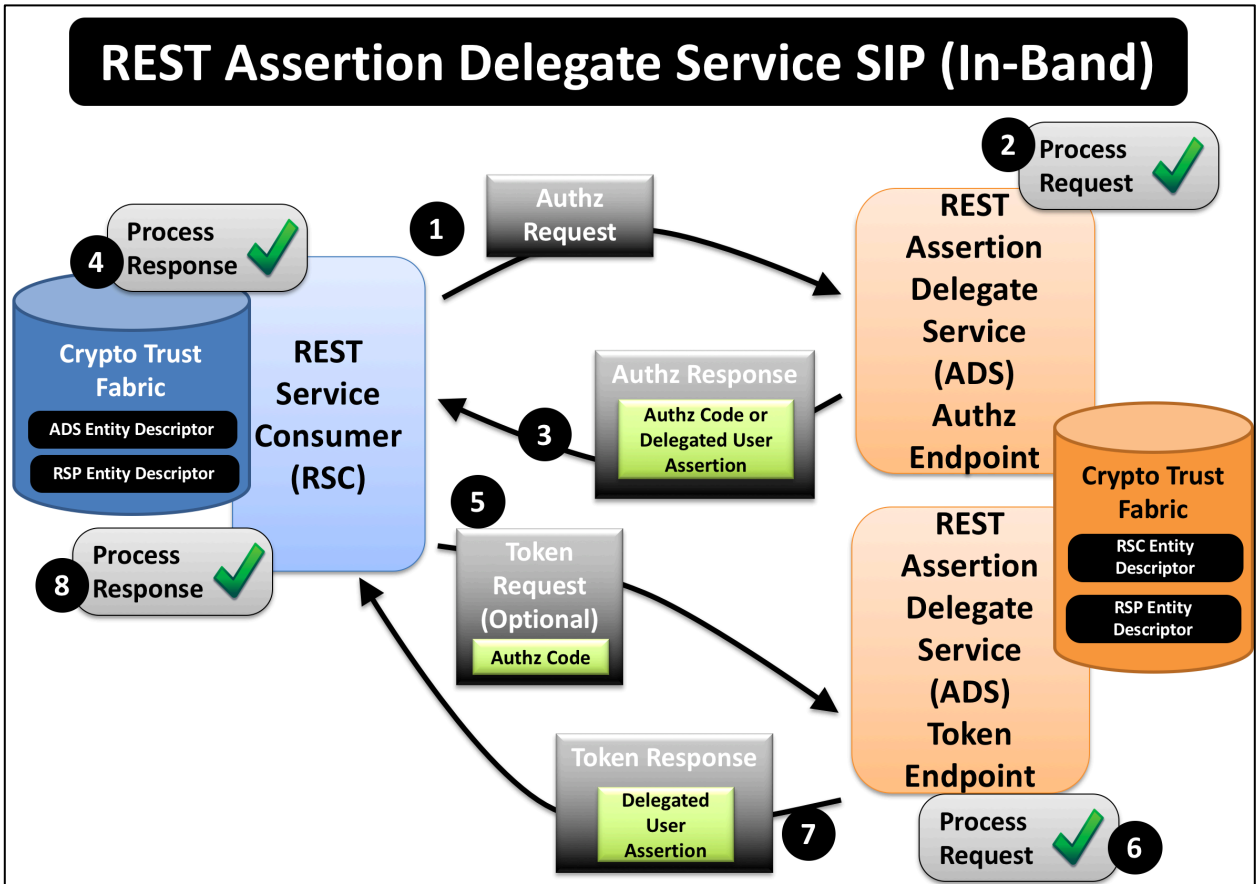

Figure 8: Diagram of the REST Assertion Delegate Service SIP with In-Band Consent

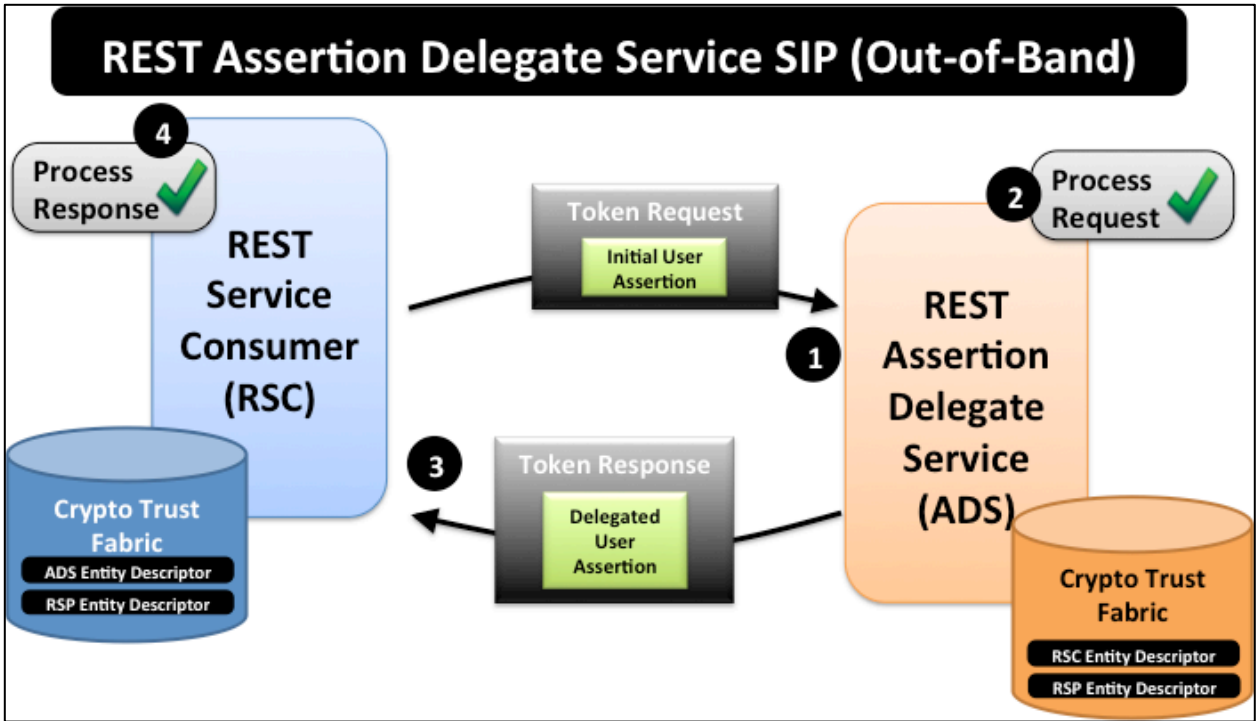

Figure 9: Diagram of the REST Assertion Delegate Service SIP with Out-of-Band Consent

6.8.2 REST Service Consumer Requirements

1. The RSC MUST conform to [OIDC Core] as a Relying Party.

2. The RSC MUST discover the appropriate ADS to use for the current user.[6]

3. Requests submitted by the RSC to the Authorization Endpoint of the ADS MUST be ADS In-Band (ADS-IB) authorization requests that conform to the requirements in Section 7.4.3.

4. If the RSC obtains an OAuth authorization code from the ADS in response to submitting an ADS-IB authorization request to the Authorization Endpoint in which the value of the "response_type" parameter is "code", then the RSC MAY submit an access token request to the Token Endpoint of the ADS. This access token request MUST conform to the requirements in Section 3.1.3.1 of [OIDC Core].

5. When interacting with the Token Endpoint, the RSC MUST authenticate to the Token Endpoint in accordance with Section 7.1.

6. The RSC MAY submit ADS Out-Of-Band (ADS-OOB) token requests that conform to the requirements in Section 7.4.4, to the Token Endpoint of the ADS. The RSC MUST use an assertion that it has previously been issued by an IDP associated with the ADS, as the assertion in the ADS-OOB token request.

[6] See Section **Error! Reference source not found.** for guidance on performing ADS discovery.

6.8.3 Assertion Delegate Service Requirements

1. The ADS MUST conform to [OIDC Core] as an OpenID Provider.

2. Upon receipt of an ADS authorization request at the Authorization Endpoint of the ADS (i.e., if the "scope" parameter of the request includes "openid" and an ADS scope value in accordance with Section 7.4.1), the Authorization Endpoint MUST perform the following steps to validate the request.

    a. Verify that the request conforms to Section 7.4.3.

    b. Verify that the value of the "client_id" parameter identifies a trusted RSC.

    c. Verify that the target resource specified in the "resource_uri" parameter of the request is a base URI of a trusted RSP.

    The failure of any validation step constitutes an error condition.

3. The Authorization Endpoint MUST process every validated ADS authorization request in accordance with the following requirements.

    a. The Authorization Endpoint MUST authenticate the End-User in accordance with Section 3.1.2.3 of [OIDC Core].

    b. The Authorization Endpoint MUST obtain End-User consent for releasing the requested delegated user assertion in accordance to Section 3.1.2.4 of [OIDC Core].

4. If the value of the "response_type" of a validated ADS authorization request is "code", and if the Authorization Endpoint successfully obtained End-User consent, then the Authorization Endpoint MUST return an OAuth authorization code in accordance with Section 3.1.2.5 of [OIDC Core]. The authorization code MUST signify, to the Token Endpoint of the ADS, authorization to release the requested delegated user assertion to the RSC.

5. If the value of the "response_type" of a validated ADS authorization request is "delegated_assertion", and if the Authorization Endpoint successfully obtained End-User consent, then the Authorization Endpoint MUST return a response message in accordance with the requirements in the following sub-items.

    a. The response message MUST be sent to the redirect URI that was specified in the "redirect_uri" parameter of the request.

    b. The response message MUST include the "delegated_assertion" parameter added to the fragment component of the redirect URI. The value of this

parameter MUST be an assertion that is either an OIDC ID token or a SAML assertion in accordance with the ADS scope value of the request.

c. The ADS MUST be the issuer of the assertion.

d. The RSC MUST be an authorized party of the assertion.

e. The value of the "resource_uri" parameter of the request MUST be included in the audience of the assertion.

f. The ADS MUST digitally sign the assertion.

g. The response message MUST include the "state" parameter added to the fragment component of the redirect URI if the request contained the "state" parameter. The value of the "state" parameter of the response, if it exists, MUST match the value of the "state" parameter of the request.[7]

6. Upon receipt of a token request, the Token Endpoint MUST authenticate the RSC in accordance with Section 7.1.

7. Upon receipt of a token request that uses an OAuth grant type (see [OAuth Core]) of "code" at the Token Endpoint, the Token Endpoint MUST validate the request in accordance with the requirements in the following sub-items.

    a. The Token Endpoint MUST verify that the request is an OAuth authorization code flow access token request as defined in Section 4.1.3 of [OAuth Core]. If this validation step fails, then the request does not apply to this SIP and the ADS behavior is undefined.

    b. The Token Endpoint MUST verify that the value of the "code" parameter of the request is an OAuth authorization code that was provided to the RSC in response to an ADS authorization request in accordance with Section 7.4.3. If this validation step fails, then the request does not apply to this SIP and the ADS behavior is undefined.

    c. The Token Endpoint MUST validate the request in accordance with Section 3.1.3.2 of [OIDC Core].

8. After a Token Endpoint successfully validates a token request that uses an OAuth grant type (see [OAuth Core]) of "code", the Token Endpoint MUST provide a response in accordance with the following requirements.

[7] The parameter SHOULD be used for preventing cross-site request forgery as described in Section 10.12 of [OAuth Core].

a. The response message MUST use the "application/json" media type and the content of the response message payload MUST be a JSON object in accordance with [JSON].

b. The payload MUST include a "delegated_assertion" member. The value of this member MUST be an assertion that is either an OIDC ID token or a SAML assertion in accordance with the ADS scope value of the original ADS-IB authorization request.

c. The ADS MUST be the issuer of the assertion.

d. The RSC MUST be an authorized party of the assertion.

e. The value of the "resource_uri" parameter of the request MUST be included in the audience of the assertion.

f. The ADS MUST digitally sign the assertion.

9. Upon receipt of an ADS-OOB token request at the Token Endpoint (i.e., if the request conforms to Item #1 of Section 7.4.4), the ADS MUST perform the steps in the following sub-items to validate the request.

    a. The ADS MUST verify that the request is valid in accordance with Section 7.4.4.

    b. The ADS MUST validate the signature of the assertion used in the authorization grant.

    c. The ADS MUST verify that the ADS is associated with the signing key used to sign the assertion.

    d. The ADS MUST verify that the issuer value of the assertion matches the issuer value of the ADS.

    e. The ADS MUST verify that the RSC that supplied the assertion is identified in the assertion's audience.

    f. The Token Endpoint MUST verify trust in the identified RSC.

    If the request is not valid, then the ADS MUST process an error condition in accordance with Section 3.1.3.4 of [OIDC Core].

10. After the ADS successfully validates an ADS-OOB token request, the ADS MUST obtain, or have previously obtained, out-of-band consent from the subject for releasing the requested assertion. If this step fails, then the ADS MUST process an error condition in accordance with Section 3.1.3.4 of [OIDC Core].

11. Upon obtaining successful consent, the ADS MAY issue the requested delegated user process the request in accordance with the following requirements. For these requirements, the subject is the entity identified by the subject of the assertion used in the authorization grant in the request.

    g. The response message MUST use the "application/json" media type and the content of the response message payload MUST be a JSON object in accordance with [JSON].

    h. The payload of the response message MUST include a "delegated_assertion" member. The value of this member MUST be an assertion that is either an OIDC ID token or a SAML assertion in accordance with the ADS scope value of the original ADS-IB authorization request.

    i. The ADS MUST be the issuer of the assertion in the response message.

    j. The RSC MUST be an authorized party of the assertion in the response message.

    k. The value of the "resource_uri" parameter of the request MUST be included in the audience of the assertion in the response message.

    l. The ADS MUST digitally sign the assertion in the response message.

6.9 NIEF REST Attribute Provider SIP

The NIEF REST Attribute Provider SIP enables a REST Attribute Consumer (AC) to obtain supplemental claims about an End-User from a REST Attribute Provider (AP). This SIP supports ACs redirecting the User Agent to the AP so that the End-User can grant in-band consent, as well as ACs communicating directly with the AP and the AP obtaining out-of-band End-User consent.

6.9.1 Motivating Use Case (Non-Normative)

OIDC defines a UserInfo Endpoint, which provides user claims to OIDC Relying Parties (RPs) where the End-User grants in-band consent during his/her current session with the RP. This SIP provides for in-band End-User consent via mandating the use of an OIDC UserInfo Endpoint. This is suitable for REST Service Consumers (RSCs), which when providing resources to an End-User, will be connected to the End-User's user agent (UA).

However, NIEF REST Service Providers (RSPs) that expose service interfaces, and not user interfaces, do not establish active sessions directly with End-Users via their UAs. To address this scenario, this SIP also provides a method for RSPs to obtain user claims from APs about End-Users who have granted consent to the release of their claims out-of-band from the AP transactions.

Requesters of user claims are called REST Attribute Consumers (ACs), and are OAuth Clients. REST APs are OpenID Providers that behave in accordance with [OIDC Core] and extensions as defined in the normative requirements for this SIP.

Figure 10 depicts this SIP with the use of in-band user consent, and Figure 11 depicts this SIP with the use of out-of-band user consent.

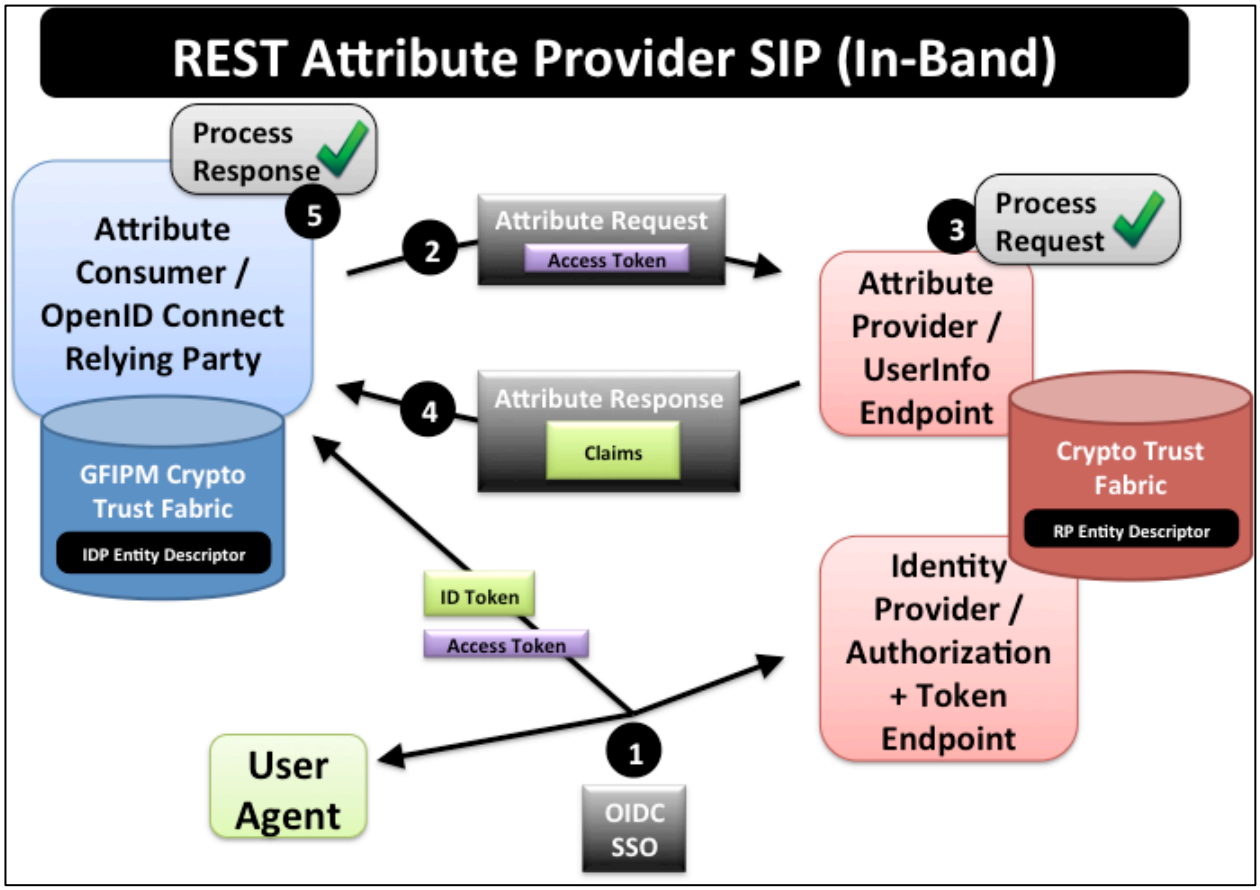

Figure 10: Diagram of the REST Attribute Provider SIP with In-Band Consent

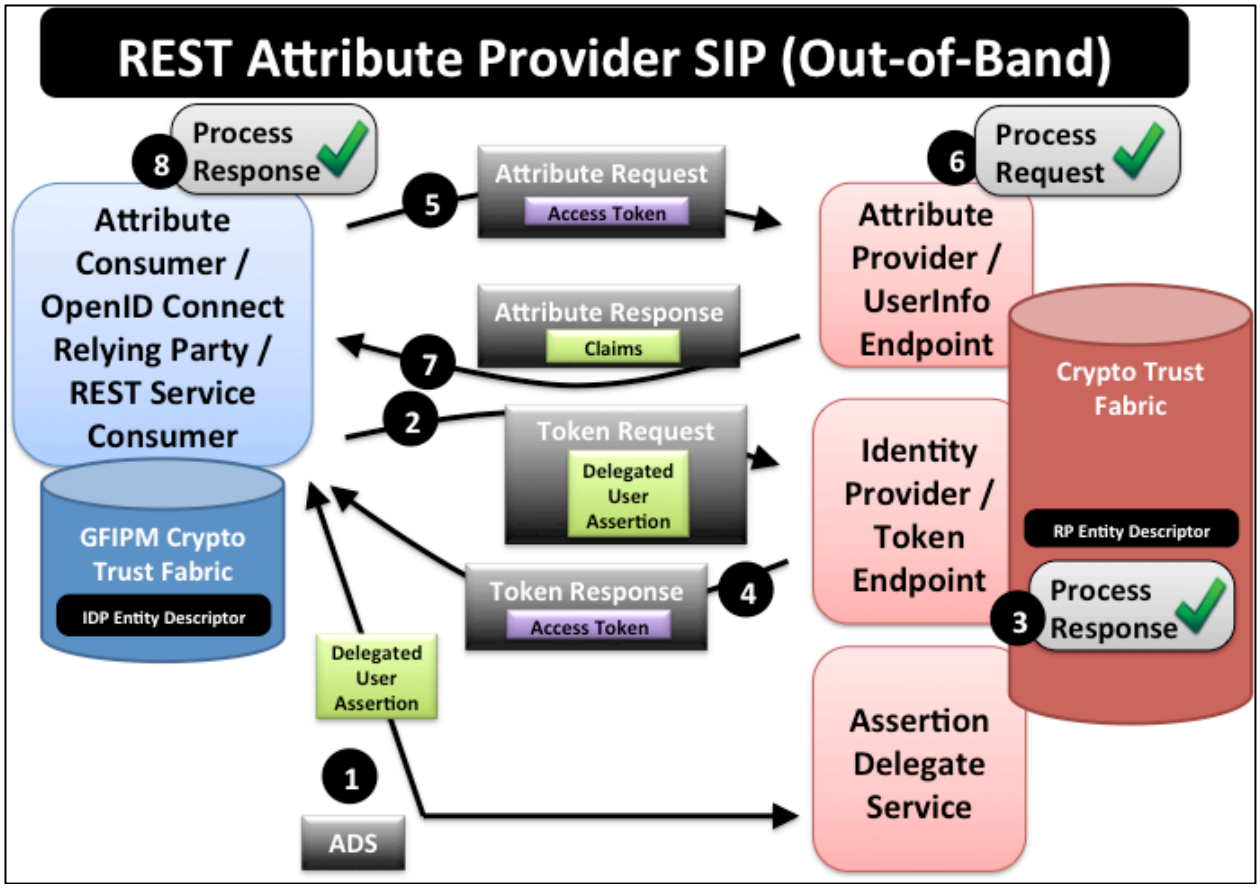

Figure 11: Diagram of the REST Attribute Provider SIP with Out-of-Band Consent

6.9.2 Attribute Consumer Requirements

1. The AC MUST conform to [OIDC Core] as a Relying Party.

2. The AC MUST only submit requests to trusted APs.

3. In order to obtain claims without having an active session with the End-User's User Agent, the AC MUST have previously obtained an OIDC ID token or SAML assertion about the subject from an IDP that's associated with the AP. This assertion MUST conform to the requirements in the following sub-items.

    a. The assertion MUST have been signed with a key associated with the AP.

    b. The issuer value of the assertion MUST match the issuer value of the AP.

    c. The AC MUST be identified in the audience of the assertion.

4. In order to obtain claims without having an active session with the End-User's User Agent, the AC MUST submit an AP Out-Of-Band (AP-OOB) access token request to

the Token Endpoint of the AP. This request MUST conform to the following requirements.

a. The request MUST conform to the requirements in Section 7.5.

b. The AC MUST supply the previously obtained assertion as the assertion used in the authorization grant.

5. When authenticating to the Token Endpoint of the AP, the AC MUST authenticate in accordance with Section 7.1.

6.9.3 Attribute Provider Requirements

1. The AP MUST conform to [OIDC Core] as an OpenID Provider.

2. The AP MUST deploy an Authorization Endpoint if the AP supports the authorization code, implicit flow, or hybrid flow in accordance with [OIDC Core].

3. The AP MUST deploy a Token Endpoint in accordance with [OIDC Core] if any of the following conditions hold.

a. The AP supports the authorization code flow in accordance with Section 3.1 of [OIDC Core].

b. The AP supports AP-OOB access token requests in accordance with Section 7.5.

4. The AP MUST deploy a UserInfo Endpoint in accordance with [OIDC Core].

5. When communicating with an AC, the Token Endpoint of the AP MUST authenticate the AC in accordance with Section 7.1.

6. Upon receipt of an AP-OOB access token request (i.e., if the request conforms to Section 7.5), then the Token Endpoint MUST validate the request in accordance with the requirements in the following sub-items.

a. The Token Endpoint MUST validate the signature of the assertion used in the authorization grant.

b. The Token Endpoint MUST verify that the AP is associated with the signing key used to sign the assertion.

c. The Token Endpoint MUST verify that the issuer value of the assertion matches the issuer value of the AP.

d. The Token Endpoint MUST verify that the REST AC that supplied the assertion is identified in the assertion's audience.

e. The Token Endpoint MUST verify trust in the identified REST AC.

7. After successfully validating an AP-OOB access token request, the Token Endpoint MAY return an access token, an OIDC ID token, or both, in accordance with the requirements in the following sub-items. For these requirements, the subject is the entity identified by the subject of the assertion used in the authorization grant in the request.

    a. The Token Endpoint MAY return an OAuth access token in accordance with [OAuth Core] and in accordance with [OIDC Core]. The access token MUST only allow the AC to access, from the UserInfo Endpoint, claims that the subject has authorized to release to the AC.

    b. The Token Endpoint MAY return an OIDC ID token in accordance with [OIDC Core]. The OIDC ID token MUST only include claims that the subject has authorized to release to the AC.

6.10 NIEF OpenID Connect Dynamic Client Registration SIP

The NIEF OpenID Connect Dynamic Client Registration SIP profiles the OIDC Dynamic Client Registration protocol (see [OIDC DCR]) to provide normative rules for the use of initial access tokens.

6.10.1Motivating Use Case (Non-Normative)

The OIDC Dynamic Client Registration (DCR) protocol defines a method for an OIDC Relying Party (RP) to dynamically register its configuration metadata with an OpenID Provider (Provider) at the Provider's Registration Endpoint. [OIDC DCR] provides a mechanism for Registration Endpoints to act as OAuth protected resources and, as such, require registration requests to be protected via the use of OAuth access tokens. These access tokens that are used in [OIDC DCR] are called "initial access tokens". The content and structure of initial access tokens are out of scope for [OIDC DCR]. The NIEF OpenID Connect Dynamic Client Registration SIP profiles [OIDC DCR] by specifying requirements for how OpenID Providers and OIDC RPs create and consume initial access tokens.

Note that this SIP applies generally to all OIDC Providers and Clients, not only those endpoints that conform to other NIEF REST SIPs.

6.10.2 OpenID Connect Relying Party Requirements

1. The OIDC RP (Client) MUST conform to [OIDC DCR] as an OIDC Relying Party.

2. When submitting a registration request to an OpenID Provider's registration endpoint, the Client MUST verify trust in the Provider's certificate that was used to establish the TLS connection.

3. When submitting a registration request to a Provider's Registration Endpoint, the Client MUST supply a self-issued initial access token that conforms to the requirements in the following sub-items.

    a. The token MUST conform to the requirements in Section 7.6.2.

    b. The value of the "aud" claim of the token MUST be the issuer identifier of the Provider.

4. The Client MUST supply the initial access token in accordance with [OAuth Bearer].

6.10.3 OpenID Provider Requirements

1. The OpenID Provider (Provider) MUST conform to [OIDC DCR] as an OpenID Provider.

2. Upon receipt of a registration request from an OIDC RP, the Provider MUST verify that the request contains an initial access token in accordance with [OAuth Bearer].

3. The Provider MUST validate the token by performing the verification steps in the following sub-items.

    a. The Provider MUST verify that the token is a JWT that conforms to the requirements in Section 7.6.2.

    b. The Provider MUST verify trust in the key used to sign the token.

    c. The Provider MUST verify that the key used to sign the token is associated with the entity identified by the "iss" claim in the token.

    d. The Provider MUST verify that the value "aud" claim of the token is its issuer identifier.

    If any of the above verification steps fail, then the Provider MUST reject the request in accordance with Section 3.3 of [OIDC DCR].

6.11 NIEF OAuth Dynamic Client Registration SIP

The NIEF OAuth Dynamic Client Registration SIP profiles the OAuth DCR protocol (see [OAuth DCR]) to provide normative rules for the use of initial access tokens.

6.11.1 Motivating Use Case (Non-Normative)

The OAuth DCR protocol defines a method for an OAuth Client (Client) to dynamically register its configuration metadata with an OAuth Authorization Server (Server) at the Server's Registration Endpoint, which may be an OAuth protected resource. [OAuth DCR] provides a mechanism for Registration Endpoints to act as OAuth protected resources and, as such, require registration requests to be protected via the use of OAuth access tokens. These access tokens that are used in [OAuth DCR] are called "initial access tokens". The content and structure of initial access tokens are out of scope for [OAuth DCR]. The NIEF OAuth Dynamic Client Registration SIP profiles [OAuth DCR] by specifying requirements for how OAuth Authorization Servers and Clients create and consume initial access tokens.

Note that this SIP applies generally to all OAuth Clients and Authorization Servers, not only those endpoints that conform to other NIEF REST SIPs.

6.11.2 OAuth Client Requirements

1. The OAuth Client (Client) MUST conform to [OAuth DCR] as an OAuth Client.

2. When submitting a registration request to an OAuth Authorization Server's (Server's) Registration Endpoint, the Client MUST verify trust in the Server's certificate that was used to establish the TLS connection.

3. When submitting a registration request to a Server's Registration Endpoint, the Client MUST supply a self-issued initial access token that conforms to the requirements in the following sub-items.

    a. The token MUST conform to the requirements in Section 7.6.2.

    b. The value of the "aud" claim of the token MUST be the issuer identifier of the Server.

4. The Client MUST supply the initial access token in accordance with [OAuth Bearer].

6.11.3 OAuth Authorization Server Requirements

1. The OAuth Authorization Server (Server) MUST conform to [OAuth DCR] as an OAuth Authorization Server.

2. Upon receipt of a registration request from an OAuth Client, the Server MUST verify that the request contains an initial access token in accordance with [OAuth Bearer].

3. The Server MUST validate the token by performing the verification steps in the following sub-items.

a. The Server MUST verify that the token is a JWT that conforms to the requirements in Section 7.6.2.

b. The Server MUST verify trust in the key used to sign the token.

c. The Server MUST verify that the key used to sign the token is associated with the entity identified by the "iss" claim in the token.

d. The Server MUST verify that the value "aud" claim of the token is its issuer identifier.

If any of the above verification steps fail, then the Server MUST reject the request in accordance with Section 4.2 of [OAuth DCR].

7. Supporting Profiles

This section contains sets of requirements that are used by the SIPs from Section 6.

7.1 Client Authentication Requirements for OAuth Token Endpoints

In several NIEF REST SIPs, an OAuth Token Endpoint is required to authenticate an RSC acting as an OAuth Client or OIDC RP. This section provides normative requirements for performing this authentication.

7.1.1 REST Service Consumer Requirements

1. The RSC MUST support at least one of the following HTTP client authentication mechanisms.

    a. A client authentication mechanism defined in Section 9 of [OIDC Core], except the "none" mechanism. These mechanisms include "client_secret_basic", "client_secret_post", "client_secret_jwt", and "private_key_jwt".

    b. SAML bearer token authentication as defined in [OAuth SAML2] and as follows.

        i. The Client MUST use a self-issued SAML assertion.

        ii. The SAML assertion MUST conform to the requirements in Section 7.2.

    c. TLS client certificate authentication.

7.1.2 Token Endpoint Requirements

1. The Token Endpoint MUST authenticate the RSC using exactly one of the following HTTP client authentication mechanisms.

    a. A client authentication mechanism defined in Section 9 of [OIDC Core], except the “none” mechanism. These mechanisms include “client_secret_basic”, “client_secret_post”, “client_secret_jwt”, and “private_key_jwt”.

    b. SAML bearer token authentication as defined in [OAuth SAML2] and as follows.

        i. The Server MUST verify that the SAML assertion conforms to the requirements in Section 7.2.

    c. TLS client certificate authentication.

7.2 SAML Assertion Requirements

This section contains normative requirements for SAML assertions that are used in the NIEF REST SIPs. This includes requirements for delegated SAML assertions. A delegated SAML assertion is a SAML assertion that asserts that the entity to which the assertion was issued is acting on behalf of the subject of the assertion. The entity to which the assertion was issued is called an “authorized party” of the assertion.

1. The SAML assertion (assertion) MUST conform to the requirements in Section 2.3.3 of [SAML2 Core].

2. The assertion MUST contain a <Conditions> element with an <AudienceRestriction> element with an <Audience> element that identifies the intended audience.

3. The assertion MUST contain a <Subject> element.

4. The <Subject> element MUST contain at least one <SubjectConfirmation> element in accordance with the requirements in the following sub-items.

    a. The <SubjectConfirmation> element MUST have a Method attribute with a value of "urn:oasis:names:tc:SAML:2.0:cm:bearer".

    b. If the assertion does not have a suitable “NonOnOrAfter” attribute on the <Conditions> element, then the <SubjectConfirmation> element MUST contain a <SubjectConfirmationData> element.

    c. When present, the <SubjectConfirmationData> element MUST have a “Recipient” attribute.

5. The assertion MUST have an expiry that limits the time window during which it can be used. The expiry can be expressed either as the NotOnOrAfter attribute of the <Conditions> element or as the NotOnOrAfter attribute of a suitable <SubjectConfirmationData> element.

6. If the assertion denotes an authorized party acting on behalf of the assertion's subject, then the assertion MUST identify the authorized party as a delegate in accordance with [SAML2 Delegation].

7. The assertion MUST be digitally signed or have a message authentication code applied by the Issuer.

8. The assertion MUST be encoded using base64url where the padding bits are set to zero in accordance with [RFC4648], and the encoded assertion MUST NOT be line wrapped or contain pad characters (such as "=").[8]

7.3 Authorizer SIP Base Requirements

This section contains base requirements for REST Service Consumers (RSCs), Authorization Services (ASs), and REST Service Providers (RSPs) for the REST Consumer-Authorizer SIP (see Section 6.5) and the REST Delegated-Consumer-Authorizer SIP (see Section 6.6).

7.3.1 RSC Requirements

1. The RSC MUST conform to [OAuth Core] as an OAuth Client.

2. All redirection URIs used by the RSC MUST use TLS.

3. The RSC MUST submit OAuth authorization requests to only trusted ASes.

4. When communicating with the Token Endpoint of the AS, the RSC MUST authenticate to the Token Endpoint in accordance with Section 7.1.

5. When communicating with the Token Endpoint of the AS, the RSC MUST verify trust in the Token Endpoint.

7.3.2 AS Requirements

1. The AS MUST conform to [OAuth Core] as an OAuth Authorizer Server.

2. When communicating with an RSC, the Token Endpoint of the AS MUST authenticate the RSC in accordance with Section 7.1.

3. The Token Endpoint MUST verify trust in authenticated RSCs.

[8] These SAML assertion encoding rules come from [OAuth SAML2].

4. The AS MUST ensure that every OAuth access token (token) it issues conforms to the following requirements.

    a. The token MUST have mechanisms that allow RSPs to verify the authenticity and integrity of the token.

    b. The access token MUST have a mechanism that allows RSPs to determine when the access token has expired.

7.3.3 RSP Requirements

1. The RSP MUST conform to [OAuth Core] as an OAuth Resource Server.

2. The RSP MUST expose its resources via TLS.

3. The RSP MUST perform the validation steps in the following sub-items, in addition to the validation steps in Section 7 of [OAuth Core], to validate the access token presented by the RSC.

    a. The RSP MUST verify that it trusts the AS that issued the access token.

    b. The RSP MUST verify the authenticity and integrity of the access token.

    c. The RSP MUST verify that the access token has not expired.

7.4 REST Assertion Delegate Service Supporting Requirements

This section specifies supporting requirements for the REST Assertion Delegate Service SIP (see Section 6.8).

7.4.1 REST ADS Scope Requirements

A REST ADS scope value MUST be one of the following.

1. The value “ads-saml2-bearer” denotes a request for a delegated SAML assertion.

2. The value “ads-oidc-id-bearer” denotes a request for an OIDC ID token.

7.4.2 ADS Claims Object Requirements

1. An ADS Claims object MUST be a JSON object that conforms to Section 5.5 of [OIDC Core].

2. In addition, the object MAY contain a top-level member called “delegated_assertion”. This member is a JSON object that requests that the listed individual claims be

returned in the delegated assertion. The structure of this object MUST conform to the structure of the "userinfo" and "id_token" objects as defined in Section 5.5 of [OIDC Core].

7.4.3 ADS Authorization Request Requirements

1. An HTTP request is an ADS authorization request if it is an OAuth authorization request in accordance with [OAuth Core] that has a "scope" parameter whose value is a space delimited set of string scope values that includes an ADS scope value in accordance with Section 7.4, and includes the value "openid". The "scope" parameter MAY include other scope values.

2. An ADS authorization request is valid if it conforms to the requirements in the following sub-items.

    a. The value of the "response_type" parameter MUST be either "code" or "delegated_assertion".

    b. The request MUST have a "resource_uri" parameter. The value of this parameter MUST a base URI of the resource(s) to which the delegated user assertion is destined.

    c. The request MAY have a "display" parameter. If this parameter exists, it MUST conform to the "display" parameter requirements in Section 3.1.2.1 of [OIDC Core].

    d. The request MAY have a "prompt" parameter. If this parameter exists, it MUST conform to the "prompt" parameter requirements in Section 3.1.2.1 of [OIDC Core].

    e. The request MAY have a "max_age" parameter. If this parameter exists, it MUST conform to the "max_age" parameter requirements in Section 3.1.2.1 of [OIDC Core].

    f. The request MAY have a "ui_locale" parameter. If this parameter exists, it MUST conform to the "ui_locale" parameter requirements in Section 3.1.2.1 of [OIDC Core].

    g. The request MAY have an "id_token_hint" parameter. If this parameter exists, it MUST conform to the "id_token_hint" parameter requirements in Section 3.1.2.1 of [OIDC Core].

    h. The request MAY have a "saml_token_hint" parameter. If this parameter exists, it MUST conform to the "id_token_hint" parameter requirements in Section 3.1.2.1 of [OIDC Core], where a SAML assertion is used instead of an

OIDC ID token. SAML assertions MUST conform to the requirements in Section 7.2.

i. The request MAY have a "login_hint" parameter. If this parameter exists, it MUST conform to the "login_hint" parameter requirements in Section 3.1.2.1 of [OIDC Core].

j. The request MAY have an "acr_values" parameter. If this parameter exists, it MUST conform to the "acr_values" parameter requirements in Section 3.1.2.1 of [OIDC Core].

k. The request MAY have a "state" parameter. If this parameter exists, it MUST conform to the "state" parameter requirements in Section 3.1.2.1 of [OIDC Core].

l. The request MAY have "claims" parameter. If this parameter exists, its value MUST be a JSON object that conforms to Section 7.4.2.

7.4.4 ADS-OOB Token Request Requirements

1. An HTTP request that is sent to an OpenID Connect Token Endpoint is an Assertion Delegate Service Out-Of-Band (ADS-OOB) token request if it has a "scope" parameter whose value is a space delimited set of string scope values that includes an ADS scope value in accordance with Section 7.4, and includes the value "openid". The "scope" parameter MAY include other scope values.

2. An ADS-OOB access token request is valid if it conforms to the requirements in the following sub-items.

    a. The request MUST have a "resource_uri" parameter. The value of this parameter MUST a base URI of the resource(s) to which the delegated user assertion is destined.

    b. The request MAY have "claims" parameter. If this parameter exists, its value MUST be a JSON object that conforms to Section 7.4.2.

    c. The request MUST use an authorization grant that conforms to either [OAuth JWT] or [OAuth SAML2].

7.5 REST Attribute Provider Out-of-Band Access Token Requests

A REST Attribute Provider Out-Of-Band (AP-OOB) access token request is an OAuth access token request that uses either the [OAuth JWT] or [OAuth SAML2] authorization grant. The request uses the claims feature of OpenID Connect to express requests for particular claims.

7.5.1 REST AP-OOB Access Token Request Requirements

An HTTP request is an AP-OOB access token request if it conforms to the requirements in the following sub-items.

1. The request MUST be an OAuth access token request in accordance with [OAuth Core].

2. The request MUST use an authorization grant that conforms to either [OAuth JWT] or [OAuth SAML2].

3. The request MUST have a “scope” parameter whose value is a space delimited set of string values. The value of the “scope” parameter MUST include “openid”.

4. The request MAY have a “claims” parameter. The semantics and value of this parameter MUST conform to the requirements in Section 5.5 of [OIDC Core].

7.6 Self-Signed OAuth Access Token Profile

This profile specifies how self-signed JWTs can be used by OAuth Clients as access tokens when communicating with OAuth protected resources.

7.6.1 Motivating Use Case (Non-Normative)

Multiple NIEF SIPs require or allow OAuth Clients or OpenID Connect Clients (Clients) to communicate on behalf of themselves with OAuth-protected Resource Servers without the use of OAuth Authorization Servers. These SIPs include the NIEF OpenID Connect Dynamic Client Registration SIP (see Section 9) and the NIEF OAuth Dynamic Client Registration SIP (see Section 6.11). This profile specifies normative requirements on the use of self-signed JWTs that may be used as access tokens in these scenarios. These requirements are based on the requirements for using JWTs for OAuth client authentication as defined in [OAuth JWT].

Note that the use of the term “OAuth Client” herein includes OpenID Connect Clients and NIEF REST Service Consumers.

7.6.2 Self-Issued OAuth Access Token Requirements

1. The access token (token) MUST be a JWT that conforms to [JWT].

2. The token MUST contain an "iss" (issuer) claim. The semantics of this claim are as specified in Section 4.1.1 of [JWT]. The value of this claim MUST be a string that is the client identifier of the OAuth Client (Client) that is issuing the token.

3. The token MUST contain a "sub" (subject) claim. The semantics of this claim are as specified in Section 4.1.2 of [JWT]. The value of this claim MUST be a string that is the client identifier of the Client that is issuing the token.

4. The token MUST contain an "aud" (audience) claim. The semantics of this claim are as specified in Section 4.1.3 of [JWT]. The value of this claim MUST be a string that is the identifier of the OAuth protected resource to which the token is being sent

5. The token MUST contain an "exp" (expiration) claim. The semantics of this claim are as specified in, and use of this claim MUST conform to, Section 4.1.4 of [JWT].

6. The token MUST contain an "iat" (issued at) claim. The semantics of this claim are as specified in, and use of this claim MUST conform to, Section 4.1.6 of [JWT].

7. The JWT MUST be signed in accordance with [JWS].

8. The signed JWT MUST be encoded via the JWS Compact Serialization in accordance with [JWS].

7.7 Definition of Base URI

When used in this document, the term “base URI” is defined as follows. A URI (the first URI) is a base URI of a second URI if the following conditions hold.

1. The scheme part of both URIs are equivalent.

2. The authority part of both URIs are empty, or the authority part of both URIs are equivalent.

3. The path component of the second URI includes the path component of the first URI. This condition holds when the path component of both URIs are equivalent.

4. The query and fragment parts are not considered in this comparison.

7.7.1 Examples (Non-Normative)

1. “http://nief.org/trust/” is a base URI of “http://nief.org/trust/example/one/”.

2. “http://nief.org/trust” is NOT a base URI of “http://sub.nief.org/trust/example/one/”, since the authority component of both URIs are not equivalent.

7.8 TLS Requirements

TLS channels used in accordance with this Profile SHOULD use TLS version 1.1 or higher, and MUST be configured to provide anti-replay, confidentiality, and integrity protections.