Token and Identity Chaining Between OAuth Protected Resources in a Multiple ICAM Ecosystem Using OAuth Token Exchange

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

This document has been approved for public release 21-1422

Annapolis Junction, MD

Beth Abramowitz
Kelley Burgin
Neil McNab
Michael Peck
Mark Russell
Roger Westman

May 2021

This page intentionally left blank.

Table of Contents

List of Figures

1 Introduction

This document extends the Enterprise Mission Tailored OAuth 2.0 Profile [OAuth-Profile] to enable token and identity chaining in a multiple Identity, Credential, and Access Management (ICAM) ecosystem by profiling OAuth 2.0 Token Exchange [RFC8693], an Internet Engineering Task Force (IETF) Request for Comments (RFC) that defines a protocol that enables exchanging an access token with an authorization server (AS) for another access token. Readers of this document are expected to have a thorough understanding of the Enterprise Mission Tailored OAuth 2.0 (or newer) Profile.

All components described in the following are assumed to be profile-compliant with the Enterprise Mission Tailored OAuth 2.0 (or newer) Profile. The requirements in this document assume a multiple ICAM ecosystem, where the protected resources (and possibly the users) are in different ICAM ecosystems, meaning that they trust different authorization servers. An ICAM ecosystem refers to a system that performs authentication and authorization services within a given security domain (such as a corporation or government organization). A separate profile [Token-Chaining] provides requirements for a single ICAM ecosystem involving protected resources that trust the same authorization server.

The Enterprise Mission Tailored OAuth 2.0 Profile describes use of OAuth 2.0 by an OAuth client to obtain an OAuth access token to access an OAuth protected resource, such as a backend database, on a user's behalf. As described in [OAuth-Profile], the OAuth client may be a web application running on a remote web server, or it may be a native application running on the user's own endpoint system. The type of OAuth client and the method (if any) by which the user authenticates to the OAuth client is out of scope for this profile.

This profile describes how to handle the situation where, in a multiple ICAM ecosystem, a protected resource (PR1) may need to call a second protected resource (PR2) such as a second backend database in order to satisfy a query received from a client. PR1 cannot simply replay Token1 at PR2 since PR2 trusts a different authorization server and the Enterprise Mission Tailored OAuth 2.0 Profile requires that the tokens be sender and/or audience constrained, so PR1 must request a new access token, Token2, from an authorization server that is valid for PR1 to use at PR2 (in this usage, PR1 is acting as an OAuth client). If PR2 needs to access a third protected resource (PR3), then PR2 must request a new access token, Token3, and so on. This process of exchanging Token1 (which grants access to PR1) to obtain a new access token, Token2 (which grants access to PR2) is called ***token chaining***. This profile additionally enables ***identity chaining*** by ensuring that the identities of the user, client, and protected resources are propagated in the exchanged tokens, so that each protected resource can, as necessary, use the set of identities to make appropriate access decisions.

This profile describes only the case where an OAuth protected resource receives an OAuth access token and is exchanging it for a new OAuth access token. Another use case may exist where an OAuth client (or protected resource acting as an OAuth client) needs to obtain an access token to act on behalf of a user but does not have an access token to exchange and cannot perform the OAuth authorization code flow as described in [OAuth-Profile] to obtain an access token. Also, use cases may exist where other types of tokens, such as Security Access Markup

Language (SAML) tokens, need to be exchanged. This profile does not describe those use cases. Requirements to meet those use cases would need to be specified separately.

1.1 Requirements Notation and Convention

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

1.2 Terminology

This specification uses the terms "Access Token", "Authorization Server", "Client", "Protected Resource", "Resource Server", and "Token Endpoint" defined by OAuth 2.0 [RFC6749], the term "Assertion" defined by [RFC7521], the terms "Token Endpoint" and "Token Introspection" defined by [RFC7662], the term "Token Exchange" defined by [RFC8693], and the terms defined by OpenID Connect Core 1.0 [OIDC-Core].

1.3 Conformance

This specification defines requirements for the following components:

- OAuth 2.0 or 2.1 protected resources
- OAuth 2.0 or 2.1 authorization servers

The requirements include details of interactions between these components:

- Protected resource (acting as a client) to authorization server
- Protected resource (acting as a client) to another protected resource
- Authorization server (acting as a client) to another authorization server

When a profile-compliant component is interacting with other profile-compliant components in any valid combination, all components MUST implement the requirements as stated in this specification. All interaction with non-profile components is outside the scope of this specification.

A profile-compliant OAuth 2.0/2.1 protected resource PR1 acting in the role of a client to exchange an access token to receive a second access token for use at another protected resource MUST support and utilize certain features as described in the PR1 Profile in at least one of the options in Section 2 of this specification (Sections 2.1.1, 2.2.1, 2.3.1, and 2.4.1).

A profile-compliant OAuth 2.0/2.1 protected resource PR2 receiving exchanged access tokens from another entity MUST support and utilize certain features as described in the PR2 Profile in at least one of the options in Section 2 of this specification (Sections 2.1.2, 2.2.2, 2.3.2, and 2.4.2). Furthermore, for interoperability, the option(s) selected for the PR2 Profile SHOULD be the same as the option(s) selected for the PR1 Profile.

A profile-compliant OAuth 2.0/2.1 authorization server in PR1's organization MUST support and utilize certain features as described in the AS1 Profile in at least one of the options in Section 3 of this specification (Sections 3.1.1, 3.2.1, 3.3.1, and 3.4.1). Furthermore, for interoperability, the option(s) selected for the AS1 Profile SHOULD be the same as the option(s) selected for the PR1 and PR2 Profiles.

A profile-compliant OAuth 2.0/2.1 authorization server in PR2's organization MUST support and utilize certain features as described in the AS2 Profile in at least one of the options in Section 3 of this specification (Sections 3.1.2, 3.2.2, 3.3.2, and 3.4.2). Furthermore, for interoperability, the option(s) selected for the AS2 Profile SHOULD be the same as the option(s) selected for the PR1, PR2, and AS1 Profiles.

1.4 Multiple ICAM Ecosystem

The following terms will be used throughout the rest of the document.

PR1	The protected resource receiving the client request and then acting as an OAuth client in OAuth Token Exchange to obtain a new access token to a second protected resource
PR2	The second protected resource being accessed by PR1
AS1	The authorization server in PR1's organization
AS2	The authorization server in PR2's organization

Token and identity chaining can take place between two protected resources in the same ICAM ecosystem or between protected resources in different ICAM ecosystems. The focus of this document is on the second case, a multiple ICAM ecosystem.

The Enterprise Mission Tailored OAuth 2.0 Profile limits each protected resource to trust only one authorization server. In a multiple ICAM ecosystem, each protected resource (PR1) will contact an authorization server to obtain an access token that can be used at another protected resource (PR2), and the protected resources (PR1 and PR2) trust different authorization servers.

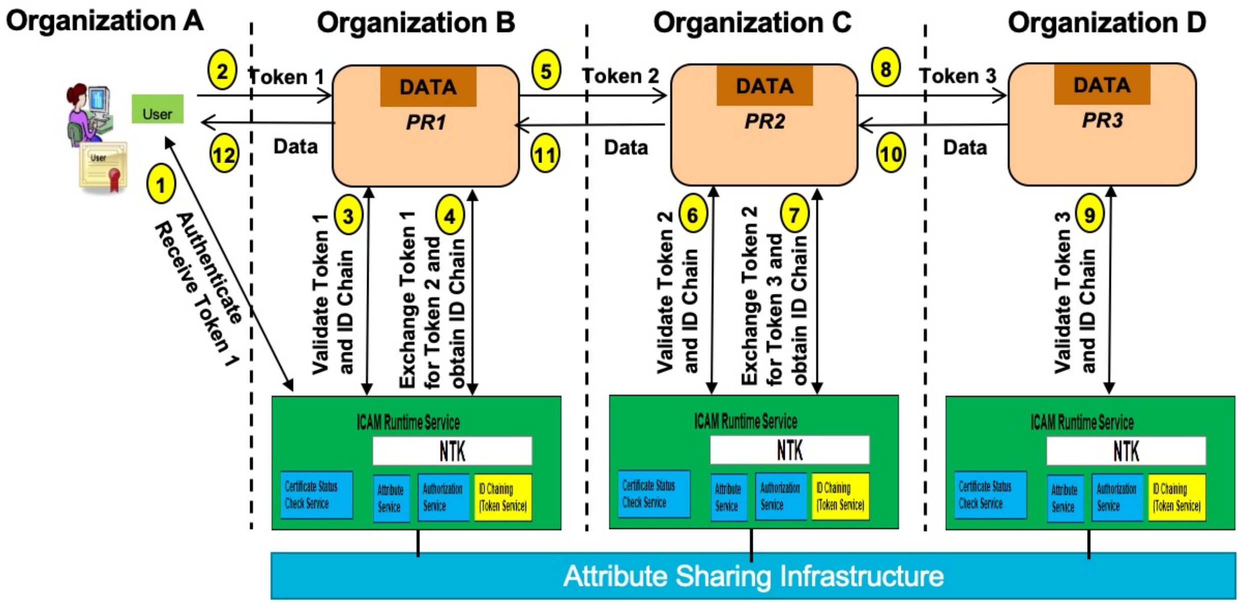
Organization A
Organization B
Organization C
Organization D
User
User
1
Authenticate
Receive Token 1
2
Token 1
12
Data
DATA
PR1
5
Token 2
11
Data
DATA
PR2
8
Token 3
10
Data
DATA
PR3
3
Validate Token 1 and ID Chain
4
Exchange Token 1 for Token 2 and obtain ID Chain
6
Validate Token 2 and ID Chain
7
Exchange Token 2 for Token 3 and obtain ID Chain
9
Validate Token 3 and ID Chain
ICAM Runtime Service
NTK
Certificate Status Check Service
Attribute Service
Authorization Service
ID Chaining (Token Service)
ICAM Runtime Service
NTK
Certificate Status Check Service
Attribute Service
Authorization Service
ID Chaining (Token Service)
ICAM Runtime Service
NTK
Certificate Status Check Service
Attribute Service
Authorization Service
ID Chaining (Token Service)
Attribute Sharing Infrastructure

Figure 1: Multiple ICAM Ecosystems OV-1

Token and identity chaining in a multiple ICAM ecosystem case is described in the following. The client follows the OAuth protocol flow as usual to obtain an access token, Token1, to access PR1. The client presents Token1 to PR1, which in turn needs to access PR2 to satisfy the client query. PR1 (acting as an OAuth client) uses OAuth Token Exchange [RFC8693] to exchange Token1 for a second token, Token2, that PR1 can use to access PR2. PR1 then presents Token2 to PR2 to obtain the data needed to satisfy the client request. Figure 1 provides a high-level view of a notional multiple ICAM ecosystem.

This process may continue if PR2 needs to access a third protected resource, PR3, that trusts its own, different authorization server, to satisfy the client request. This process may continue further if PR3 needs to access a fourth protected resource, PR4, and so on. In each case, the protected resources (PR2 and PR3, PR3 and PR4) involved satisfy the roles of PR1 and PR2 in the protocol described above.

Figure 2 illustrates a notional use case of a complex multiple ICAM ecosystem involving five organizations. Note that the user can belong to the same ICAM ecosystem as PR1 or can be part of a different ICAM ecosystem. In either case, as long as the authorization server AS1 has the ability to authenticate the user, the steps outlined above, shown in Figure 1, and described throughout this profile can be used to achieve token chaining.

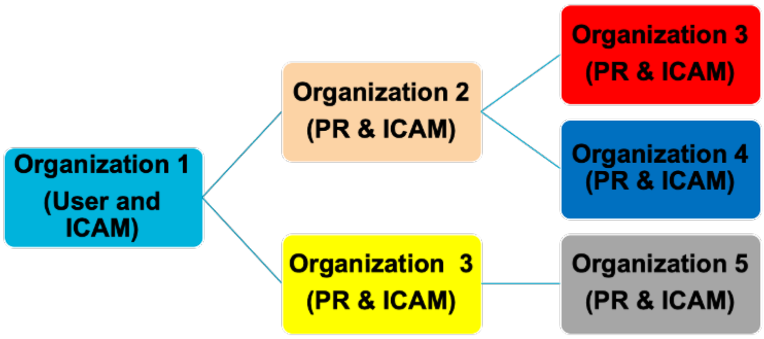

Figure 2: Multiple ICAM Ecosystem Complex Use Case

The process above is not specific about whether PR1 will perform token exchange at the authorization server AS1 in its organization or AS2 in PR2's organization. Due to the immaturity and lack of implementation experience of this requirement, this profile provides three different options for implementing token and identity chaining in multiple ICAM ecosystems. Interoperability requires that all entities interacting with one another must implement at least one option in common. A future version of this profile may down-select to only one of the options. The profile authors welcome feedback on implementation experience.

1.5 Recommended Solutions Using Token Exchange

Support for multiple ICAM ecosystems (the ability to interact with protected resources that trust different authorization servers) is optional. If implemented, each of the protected resources acting in the role of PR1, the protected resource acting in the role of PR2, the authorization server in PR1's organization, and the authorization server in PR2's organization MUST comply with the requirements in either Option 1, Option 2, or Option 3 found in Sections 2 and 3 in this profile.

1.5.1 Solution Option 1 – PR1 Performs Token Exchange at AS1

In Option 1, PR1 performs token exchange with the authorization server AS1 in its organization to receive an access token that it can use to access PR2. When PR1 presents the access token it received during token exchange to PR2, PR2 uses introspection to ask its authorization server AS2 to validate the token received from PR1, since the token was issued by AS1 in PR1's organization. High-level views of token chaining in a multiple ICAM ecosystem using Option 1 are shown in Figure 3 and Figure 4.

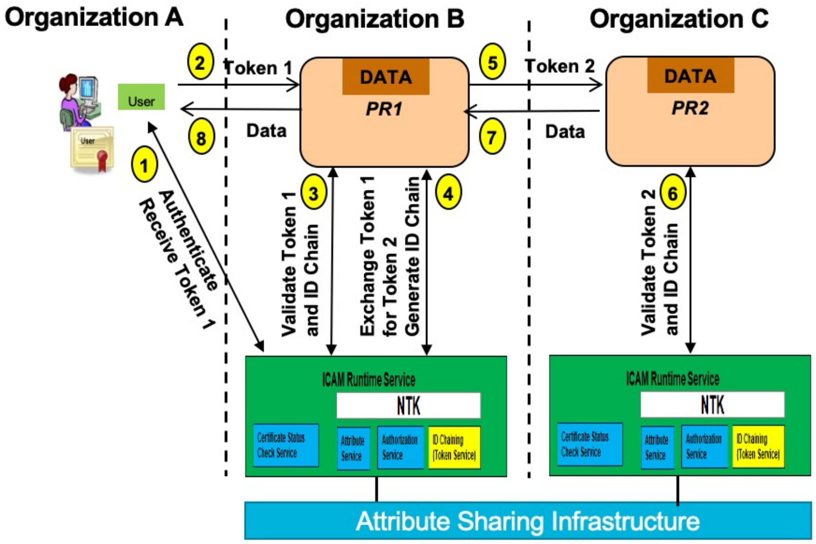

Figure 3: High-level view of token chaining in a multiple ICAM ecosystem using Option 1

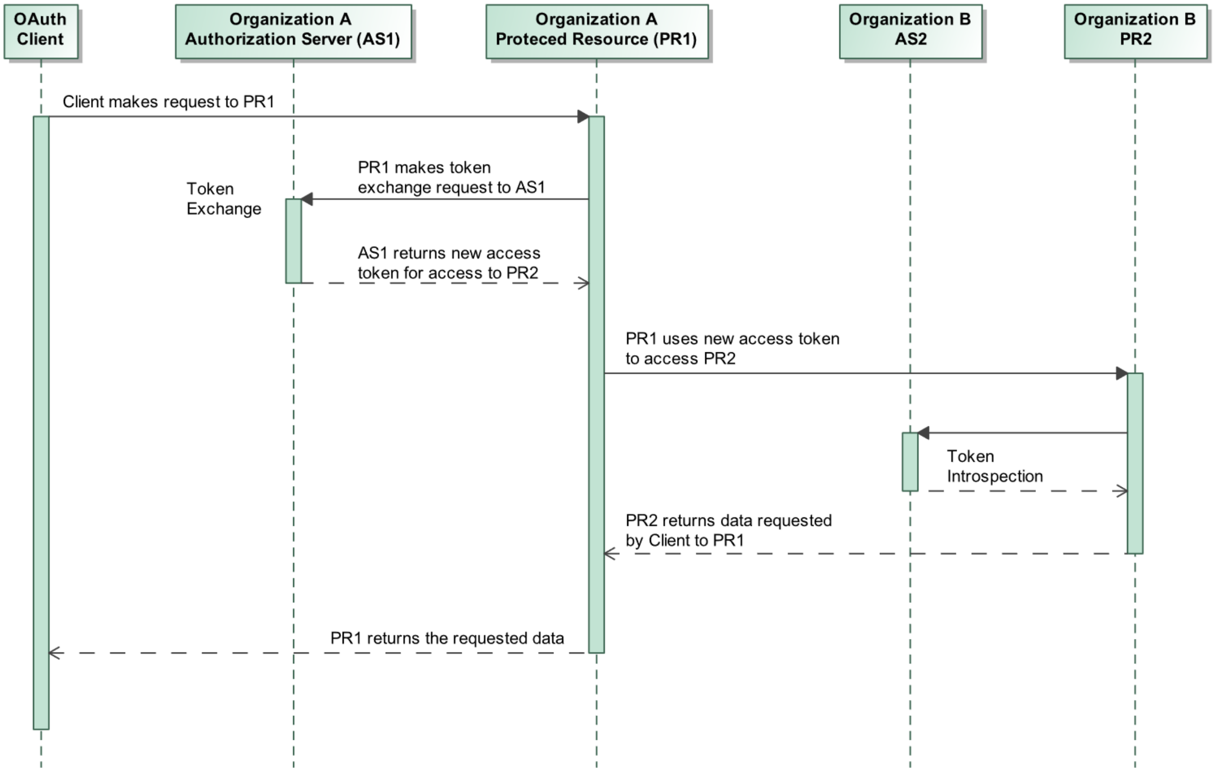
OAuth Client
Organization A Authorization Server (AS1)
Organization A Proteced Resource (PR1)
Organization B AS2
Organization B PR2
Client makes request to PR1
PR1 makes token exchange request to AS1
Token Exchange
AS1 returns new access token for access to PR2
PR1 uses new access token to access PR2
Token Introspection
PR2 returns data requested by Client to PR1
PR1 returns the requested data

Figure 4: Option 1 for Token and Identity Chaining in a Multiple ICAM Ecosystem

A benefit of using Option 1 is

- The protected resources in one organization do not need to register with the authorization servers of every other organization they may need to access.

Some challenges of using Option 1 are

- PR2 needs to trust tokens issued by another organization's authorization server.
- PR2's authorization server AS2 needs to be able to verify access tokens issued by PR1's authorization server AS1.
- "Off-label" use of the token introspection protocol.

The use of token introspection between PR2 and its own authorization server likely mitigates the first two challenges above. Furthermore, the use of token introspection in the protocol appears to meet the usage as described in [RFC7662], even if such use may not have been anticipated by the RFC's authors.

1.5.2 Solution Option 2 – PR1 Performs Token Exchange at AS2

In Option 2, PR1 performs token exchange with the authorization server AS2 in PR2's organization to receive an access token it can use to access PR2. High-level views of token chaining in a multiple ICAM ecosystem using Option 2 are shown in Figure 5 and Figure 6.

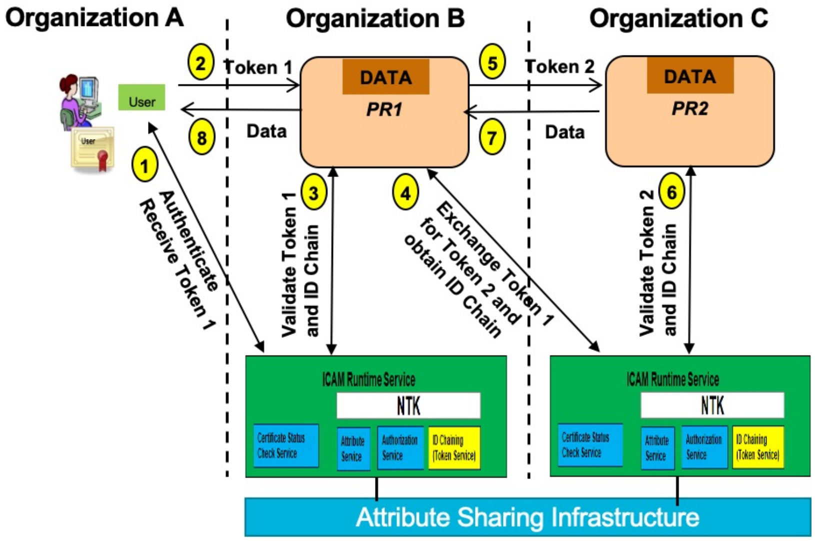

Figure 5: High-level view of token chaining in a multiple ICAM ecosystem using Option 2

A benefit of using Option 2 is

- PR2 only needs to be able to validate tokens issued by its own authorization server (AS2) rather than tokens issued by other authorization servers.

Some challenges of using Option 2 are

- The protected resources in one organization must register with the authorization server of every other organization it may need to access.
- AS2 must be able to trust, interpret, and verify access tokens issued by AS1 (and all other relevant organizations) in order to complete token exchange.

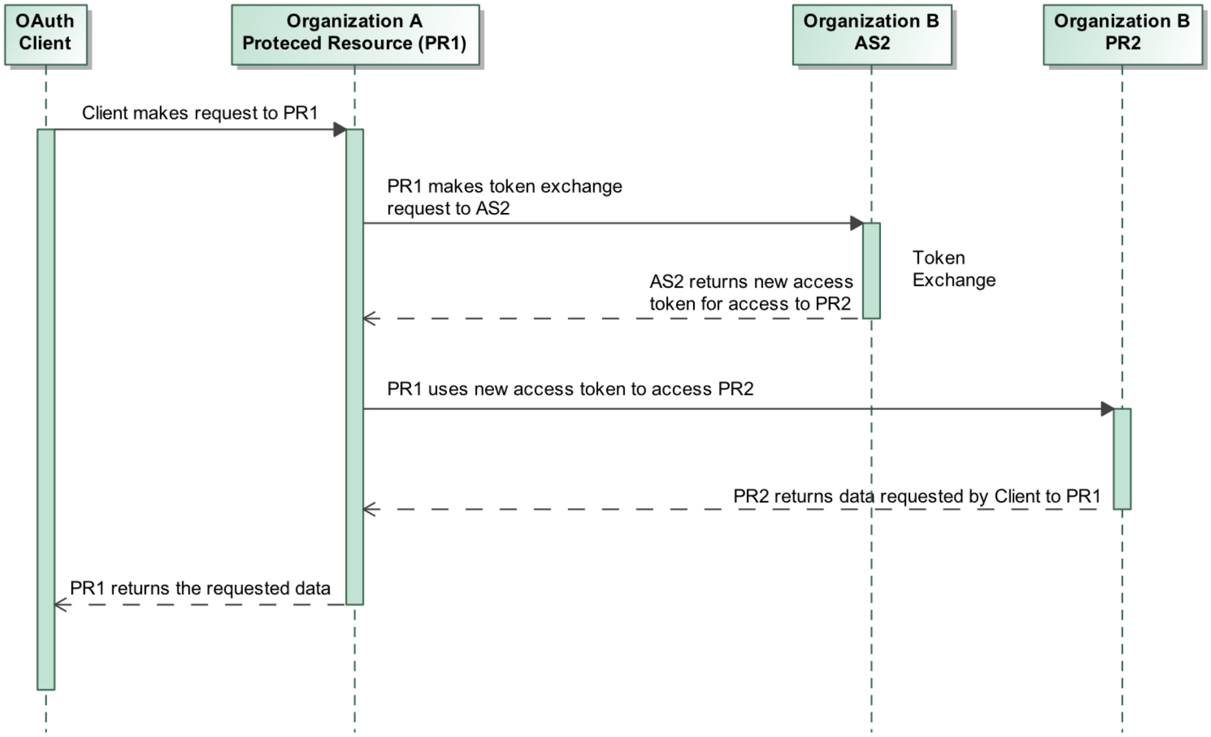
OAuth Client
Organization A Proteced Resource (PR1)
Organization B AS2
Organization B PR2
Client makes request to PR1
PR1 makes token exchange request to AS2
Token Exchange
AS2 returns new access token for access to PR2
PR1 uses new access token to access PR2
PR2 returns data requested by Client to PR1
PR1 returns the requested data

Figure 6: Option 2 for Token and Identity Chaining in a Multiple ICAM Ecosystem

1.5.3 Solution Option 3 – An Assertion Grant Is Used to Obtain a New Access Token[1]

In Option 3, a JSON Web Token (JWT) assertion is used as an intermediate step for PR1 to obtain an AS2-issued access token that it can use to access PR2. Option 3 is divided into Options 3a and 3b, described below.

1.5.3.1 Option 3a - PR1 Obtains a JWT Assertion from AS1 and an Access Token from AS2

In Option 3a, PR1 performs token exchange with the authorization server AS1 in its organization to receive a JWT assertion [RFC7523] that it sends to AS2 as part of an OAuth assertion grant request. AS2 then returns an access token to PR1 that it can use to access PR2. High-level views of token chaining in a multiple ICAM ecosystem using Option 3a are shown in Figure 7 and Figure 8.

[1] Thank you to Brian Campbell (Ping Identity) for suggesting use of JWT assertions.

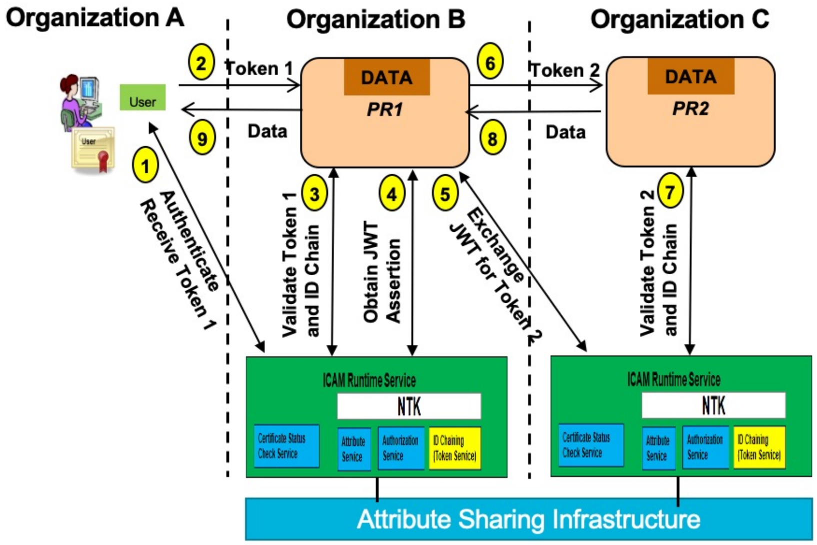

Figure 7: High-level view of token chaining in a multiple ICAM ecosystem using Option 3a

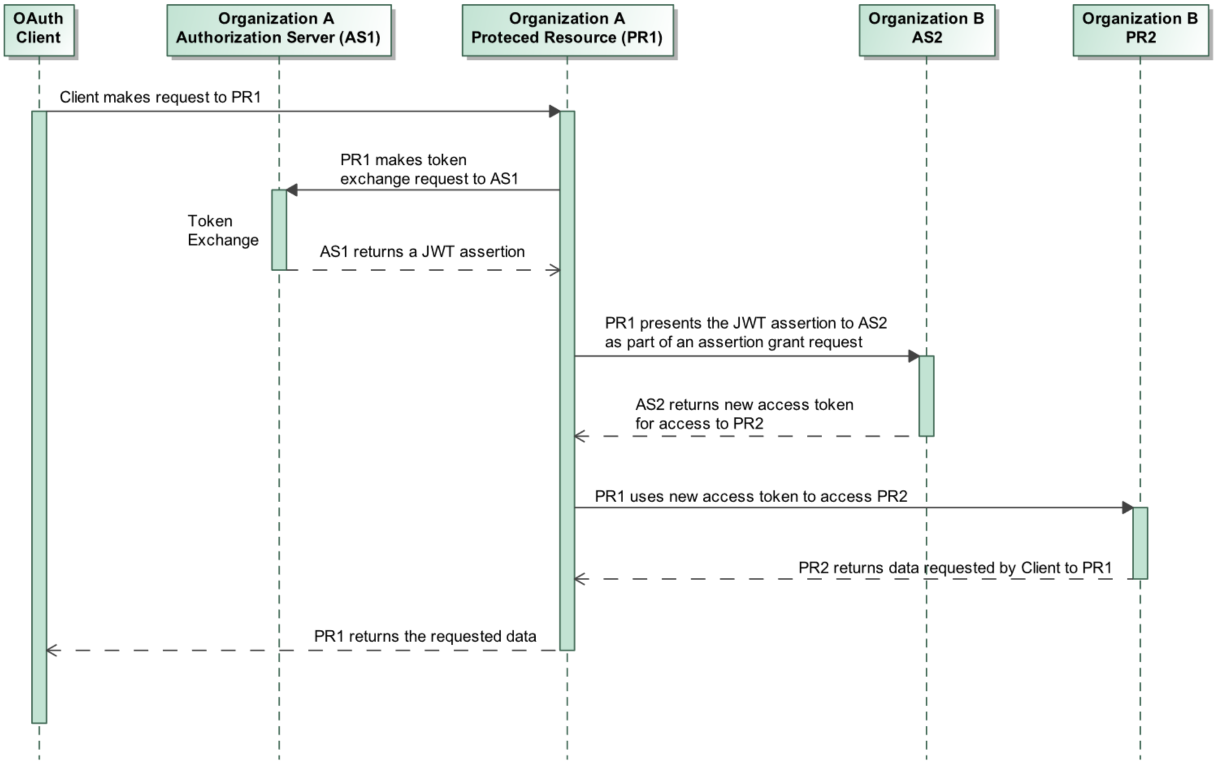
OAuth
Client
Organization A
Authorization Server (AS1)
Organization A
Proteced Resource (PR1)
Organization B
AS2
Organization B
PR2
Client makes request to PR1
PR1 makes token
exchange request to AS1
Token
Exchange
AS1 returns a JWT assertion
PR1 presents the JWT assertion to AS2
as part of an assertion grant request
AS2 returns new access token
for access to PR2
PR1 uses new access token to access PR2
PR2 returns data requested by Client to PR1
PR1 returns the requested data

Figure 8: Option 3a for Token and Identity Chaining in a Multiple ICAM Ecosystem

Some benefits of using Option 3a are

- The protected resources in one organization do not need to be registered at the authorization server of every other organization they may need to access.
- PR2 only receives access tokens issued by its authorization server AS2.
- JWT assertions are intended for use "across security domains" per RFC 7521.

Some challenges of using Option 3a are

- AS2 must trust assertions issued by AS1 in order to respond successfully to the assertion grant request.
- Authorization servers in different organizations must agree on the content of JWT assertions.
- Latency may be introduced due to multiple interactions.

1.5.3.2 Option 3b - PR1 Obtains an Access Token from AS1

In Option 3b, PR1 performs token exchange with the authorization server AS1 in its organization to receive a new access token it can use to access PR2. However, AS1 does not generate the access token it returns to PR1. Instead, AS1 generates a JWT assertion and (acting as an OAuth client) issues an assertion grant request to AS2 using the assertion AS1 generated to receive a new access token generated by AS2 that PR1 can use to access PR2. High-level views of token chaining in a multiple ICAM ecosystem using Option 3b are shown in Figure 9 and Figure 10.

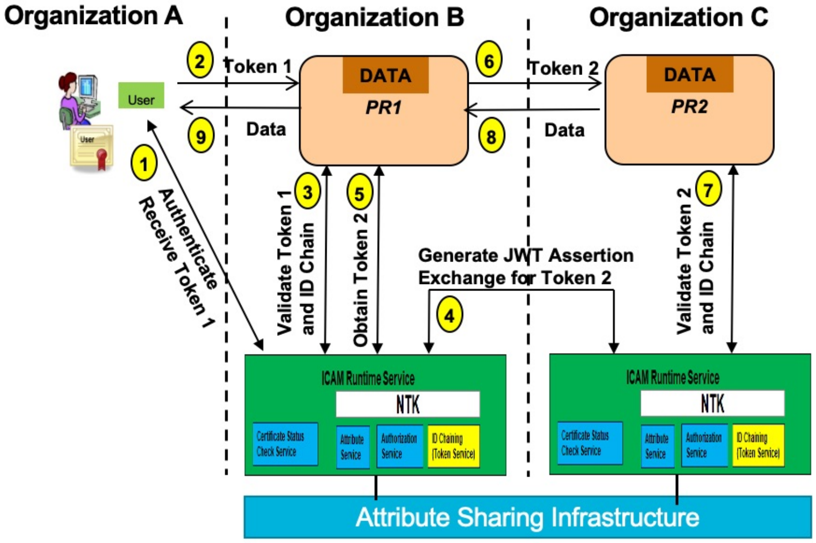

Figure 9: High-level view of token chaining in a multiple ICAM ecosystem using Option 3b

Some benefits of using Option 3b are

- The protected resources in one organization do not need to be registered at the authorization server of every other organization they may need to access.
- PR2 only receives access tokens issued by its authorization server AS2.
- PR1 only needs to contact its own authorization AS1 rather than both AS1 and AS2.
- JWT assertions are intended for use “across security domains” per RFC 7521.

Some challenges of using Option 3b are

- AS2 must trust assertions issued by AS1 in order to respond successfully to the assertion grant request.
- Authorization servers in different organizations must agree on the content of JWT assertions.
- Latency may be introduced due to multiple interactions.

- Since AS1 (rather than PR1) is connecting to AS2 to request the access token, it is unclear whether it's practical for AS2 to bind the issued access token to PR1's certificate. Furthermore, AS2 does not receive any proof that PR1 is involved in the transaction other than AS1 saying so, which may make it more difficult to audit transactions.

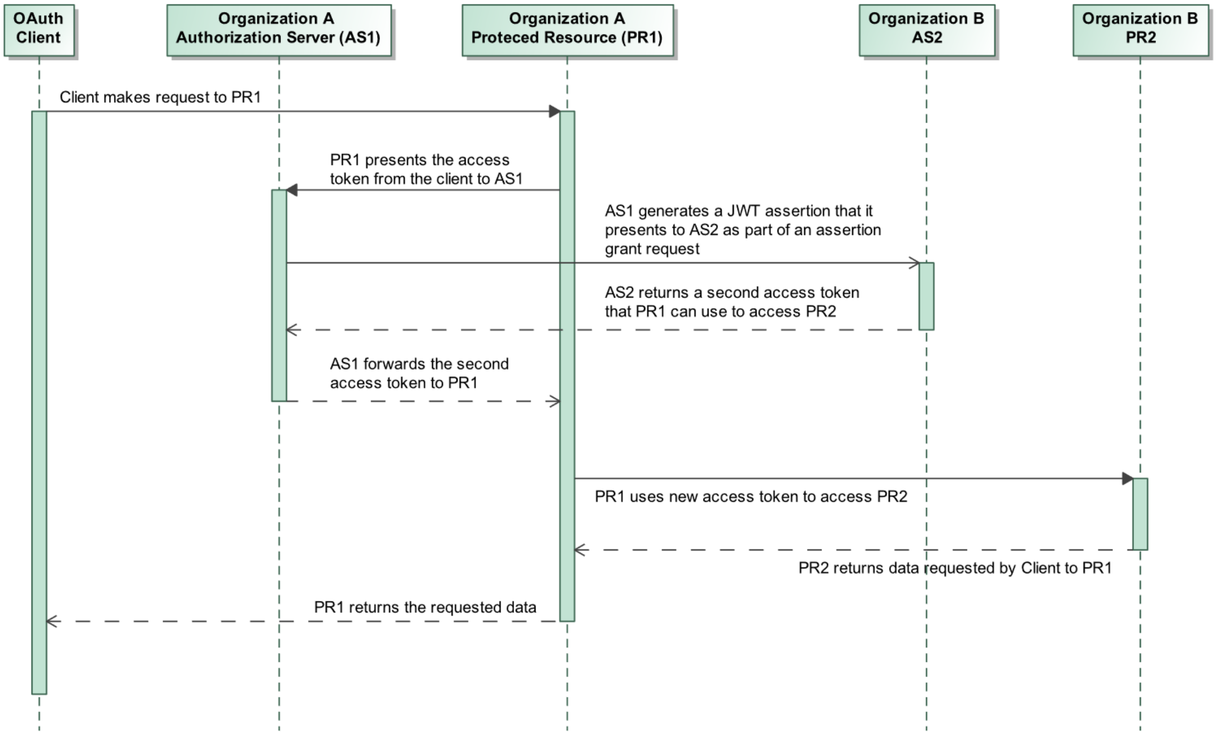
OAuth Client
Organization A Authorization Server (AS1)
Organization A Proteced Resource (PR1)
Organization B AS2
Organization B PR2
Client makes request to PR1
PR1 presents the access token from the client to AS1
AS1 generates a JWT assertion that it presents to AS2 as part of an assertion grant request
AS2 returns a second access token that PR1 can use to access PR2
AS1 forwards the second access token to PR1
PR1 uses new access token to access PR2
PR2 returns data requested by Client to PR1
PR1 returns the requested data

Figure 10: Option 3b for Token and Identity Chaining in a Multiple ICAM Ecosystem

2 Protected Resource Profiles

The protected resources acting in the roles of PR1 and PR2 MUST comply with the requirements described in Section 4 (Protected Resource Profile) of the Enterprise Mission Tailored OAuth 2.0 Profile.

2.1 Option 1 (PR1 Performs Token Exchange at AS1)

PR1 performs token exchange with the authorization server AS1 in its organization to receive an access token that it can use to access PR2. When PR1 presents the access token it received during token exchange to PR2, PR2 uses introspection to ask its authorization server AS2 to validate the token received from PR1.

2.1.1 Protected Resource 1 (PR1) Profile

This section imposes requirements on and describes the actions taken by PR1 to obtain a new access token from an authorization server valid for use by PR1 at PR2. When interacting with the authorization server and with PR2, PR1 is acting in the role of an OAuth client. If PR2 then needs to exchange the access token to access PR3, then PR2 would adopt the role of PR1 as described in this profile, and PR3 would adopt the role of PR2.

2.1.1.1 Connection to AS1 in PR1's Organization

When performing token exchange, PR1 MUST authenticate to the token endpoint of AS1 using mutually authenticated Transport Layer Security (TLS), in compliance with Section 2.1 of RFC8705, using a Public Key Infrastructure (PKI) certificate and corresponding private key.

PR1, when complying with this profile, MUST set the fields of its token exchange requests as follows (note that this table is equivalent to the one in section 2.2.1.1 defining requirements for Option 2).

grant_type	REQUIRED	Value set to "urn:ietf:params:oauth:grant-type:token-exchange" as required by Section 2.1 of [RFC8693].
client_id	REQUIRED	Value set to PR1's client_id at the authorization server as required by Section 2 of [RFC8705].
resource	OPTIONAL - at least one of "resource" or "audience" MUST be set	Set as described in [RFC8693]. The requirement that at least one of "resource" or "audience" must be set is per this profile.
audience	OPTIONAL - at least one of "resource" or "audience" MUST be set	Set as described in [RFC8693]. The requirement that at least one of "resource" or "audience" must be set is per this profile.

scope	OPTIONAL	Set as described in [RFC8693].
requested_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access_token" as described in Section 3 of [RFC8693]. The requirement that requested_token_type must be set is per this profile.
subject_token	REQUIRED	Value set to the access token sent to PR1 from its client. The requirement to include subject_token is per [RFC8693] Section 2.1. The requirement that it be set to the access token is per this profile.
subject_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access_token". The requirement to include subject_token_type is per [RFC8693] Section 2.1. The requirement that it identify an access token is per this profile.
actor_token	NOT ALLOWED	PR1 is the actor and identifies itself to the authorization server through TLS client certificate authentication per [RFC8705]; therefore, this field is not permitted. This field is optional per [RFC8693] Section 2.1 and is prohibited per this profile.
actor_token_type	NOT ALLOWED	PR1 is the actor and identifies itself to the authorization server through TLS client certificate authentication per [RFC8705]; therefore, this field is not permitted. This field is prohibited per [RFC8693] Section 2.1 when actor_token is not present.

2.1.1.2 Connection to PR2

For connections between PR1 and PR2, where PR1 is acting in an OAuth Client role, PR1 MUST comply with the requirements described in Section 2.3 (Client Connection to the Protected Resource) of the Enterprise Mission Tailored OAuth 2.0 Profile.

2.1.2 Protected Resource 2 (PR2) Profile

As described by Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile, PR2 (the recipient of an access token presented by PR1) may directly make authorization decisions based on the scopes or other claims that are optionally found in the access token. Alternatively, PR2 can make use of applicable enterprise authorization services to determine the allowed access.

This access determination can be based on the user’s identity (as asserted by the "sub" claim), PR1’s identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange (as asserted by the "act" claim).

If the protected resource acting in the role of PR2 needs to make a request to an additional protected resource, this initiates a new token chaining transaction. Such protected resources that both receive and request chained tokens must comply with the PR1 profile in the context of requesting new tokens for further resource access, and with the PR2 profile in the context of receiving tokens from the prior protected resource in the chain. No additional requirements are imposed on protected resources that perform both roles.

However, risks exist that must be accepted if PR2 chooses to use identities asserted by nested "act" claims within the access token. [RFC8693] states, "[f]or the purpose of applying access control policy, the consumer of a token MUST only consider the token's top-level claims and the party identified as the current actor by the ‘act’ claim. Prior actors identified by any nested ‘act’ claims are informational only and are not to be considered in access control decisions."

2.1.2.1 Connection to the Authorization Server AS2 in PR2’s Organization

In Option 1, PR2 uses token introspection [RFC7662] to validate the access token received from PR1. PR2 connects using mutual TLS to the introspection endpoint at the authorization server AS2 in its organization to validate the token, since it was issued by a different authorization server (AS1 in PR1’s organization). When performing introspection, PR2 MUST comply with the requirements in [RFC7662]. The details of the arrangement between the authorization servers in the two organizations that allows for validating access tokens is out of scope for this document.

2.2 Option 2 (PR1 Performs Token Exchange at AS2)

PR1 performs token exchange with the authorization server AS2 in PR2’s organization to receive an access token it can use to access PR2.

2.2.1 Protected Resource 1 (PR1) Profile

This section imposes requirements on and describes the actions taken by PR1 to obtain a new access token from an authorization server valid for use by PR1 at PR2. When interacting with the authorization server and with PR2, PR1 is acting in the role of an OAuth client. If PR2 then needs to exchange the access token to access PR3, then PR2 would adopt the role of PR1 as described in this profile, and PR3 would adopt the role of PR2.

The requirements for PR1 are the same as in the previous section specified for Option 1, except that PR1 performs token exchange at AS2 instead of AS1. The requirements for PR2 are the same as for Option 1.

2.2.1.1 Connection to the Authorization Server Performing Token Exchange

When performing token exchange, PR1 MUST authenticate to the token endpoint of AS2 using mutually authenticated TLS, in compliance with Section 2.1 of [RFC8705], using a PKI certificate and corresponding private key.

PR1, when complying with this profile, MUST set the fields of its token exchange requests as follows (note that the below table is equivalent to the table found in section 2.1.1.1 defining the Option 1 requirements).

grant_type	REQUIRED	Value set to "urn:ietf:params:oauth:grant-type:token-exchange" as required by Section 2.1 of [RFC8693].
client_id	REQUIRED	Value set to PR1's client_id at the authorization server as required by Section 2 of [RFC8705].
resource	OPTIONAL - at least one of "resource" or "audience" MUST be set	Set as described in [RFC8693]. The requirement that at least one of "resource" or "audience" must be set is per this profile.
audience	OPTIONAL - at least one of "resource" or "audience" MUST be set	Set as described in [RFC8693]. The requirement that at least one of "resource" or "audience" must be set is per this profile.
scope	OPTIONAL	Set as described in [RFC8693].
requested_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access_token" as described in Section 3 of [RFC8693]. The requirement that requested_token_type must be set is per this profile.
subject_token	REQUIRED	Value set to the access token sent to PR1 from its client. The requirement to include subject_token is per [RFC8693] Section 2.1. The requirement that it be set to the access token is per this profile.
subject_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access_token". The requirement to include subject_token_type is per [RFC8693] Section 2.1. The requirement that it identify an access token is per this profile.
actor_token	NOT ALLOWED	PR1 is the actor and identifies itself to the authorization server through TLS client certificate authentication per [RFC8705]; therefore, this field is not permitted. This field is optional per [RFC8693] Section 2.1 and is prohibited per this profile.

actor_token_type	NOT ALLOWED	PR1 is the actor and identifies itself to the authorization server through TLS client certificate authentication per [RFC8705]; therefore this field is not permitted. This field is prohibited per [RFC8693] Section 2.1 when actor_token is not present.

2.2.1.2 Connection to PR2

For connections between PR1 and PR2, where PR1 is acting in an OAuth Client role, PR1 MUST comply with the requirements described in Section 2.3 (Client Connection to the Protected Resource) of the Enterprise Mission Tailored OAuth 2.0 Profile.

2.2.2 Protected Resource 2 (PR2) Profile

As described by Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile, PR2 (the recipient of an access token presented by PR1) may directly make authorization decisions based on the scopes or other claims that are optionally found in the access token. Alternatively, PR2 can make use of applicable enterprise authorization services to determine the allowed access.

This access determination can be based on the user's identity (as asserted by the "sub" claim), PR1's identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange (as asserted by the "act" claim).

If the protected resource acting in the role of PR2 needs to make a request to an additional protected resource, this initiates a new token chaining transaction. Such protected resources that both receive and request chained tokens must comply with the PR1 profile in the context of requesting new tokens for further resource access, and with the PR2 profile in the context of receiving tokens from the prior protected resource in the chain. No additional requirements are imposed on protected resources that perform both roles.

However, risks exist that must be accepted if PR2 chooses to use identities asserted by nested "act" claims within the access token. [RFC8693] states, "[f]or the purpose of applying access control policy, the consumer of a token MUST only consider the token's top-level claims and the party identified as the current actor by the 'act' claim. Prior actors identified by any nested 'act' claims are informational only and are not to be considered in access control decisions."

2.3 Option 3a (PR1 Obtains a JWT Assertion from AS1 and an Access Token from AS2)

PR1 performs token exchange with the authorization server AS1 in its organization to receive a JWT assertion [RFC7523] that it sends to AS2 as part of an OAuth assertion grant request. AS2 then returns an access token to PR1 that it can use to access PR2.

2.3.1 Protected Resource 1 (PR1) Profile

This section imposes requirements on and describes the actions taken by PR1 to obtain a new access token from an authorization server valid for use by PR1 at PR2. When interacting with the authorization servers AS1 and AS2 and with PR2, PR1 is acting in the role of an OAuth client. If PR2 then needs to exchange the access token to access PR3, then PR2 would adopt the role of PR1 as described in this profile, and PR3 would adopt the role of PR2.

2.3.1.1 Connection to the Authorization Server AS1 in PR1's Organization

When performing token exchange, PR1 MUST authenticate to the token endpoint of AS1 using mutually authenticated TLS, in compliance with Section 2.1 of [RFC8705], using a PKI certificate and corresponding private key.

When PR1 performs token exchange at AS1 to exchange the access token AS1 issued to the client for a JWT assertion, the fields of its token exchange requests are the same as for Options 1 and 2 with the exception that PR1 is requesting a JWT assertion instead of an access token. PR1, when complying with this profile, MUST set the fields of its token exchange requests as follows.

grant_type	REQUIRED	Value set to "urn:ietf:params:oauth:grant-type:token-exchange" as required by Section 2.1 of [RFC8693].
client_id	REQUIRED	Value set to PR1's client_id at the authorization server as required by Section 2 of [RFC8705].
resource	OPTIONAL - at least one of "resource" or "audience" MUST be set	Set as described in [RFC8693]. The requirement that at least one of "resource" or "audience" must be set is per this profile.
audience	OPTIONAL - at least one of "resource" or "audience" MUST be set	Set as described in [RFC8693]. The requirement that at least one of "resource" or "audience" must be set is per this profile.
scope	OPTIONAL	Set as described in [RFC8693].
requested_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:jwt" as described in Section 3 of [RFC8693]. The requirement that requested_token_type must be set is per this profile.
subject_token	REQUIRED	Value set to the access token sent to PR1 from its client. The requirement to include subject_token is per [RFC8693] Section 2.1. The requirement that it be set to the access token is per this profile.
subject_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access_token". The requirement to

		include subject_token_type is per [RFC8693] Section 2.1. The requirement that it identify an access token is per this profile.
actor_token	NOT ALLOWED	PR1 is the actor and identifies itself to the authorization server through TLS client certificate authentication per [RFC8705]; therefore, this field is not permitted. This field is optional per [RFC8693] Section 2.1 and is prohibited per this profile.
actor_token_type	NOT ALLOWED	PR1 is the actor and identifies itself to the authorization server through TLS client certificate authentication per [RFC8705]; therefore, this field is not permitted. This field is prohibited per [RFC8693] Section 2.1 when actor_token is not present.

2.3.1.2 Connection to the Authorization Server AS2 in PR2's Organization

After receiving the assertion grant from token exchange with AS1, PR1 presents the assertion to AS2 as part of an assertion grant token request.

PR1, when complying with this profile, MUST set the fields of its assertion grant token requests to AS2 as follows.

grant_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:jwt-bearer".
assertion	REQUIRED	Value set to the JWT assertion returned to PR1 from the token exchange with AS1.
scope	OPTIONAL	If present, the scope claim SHOULD be a subset of the values in the JWT assertion.

2.3.1.3 Connection to PR2

For connections between PR1 and PR2, where PR1 is acting in an OAuth Client role, PR1 MUST comply with the requirements described in Section 2.3 (Client Connection to the Protected Resource) of the Enterprise Mission Tailored OAuth 2.0 Profile.

2.3.2 Protected Resource 2 (PR2) Profile

As described by Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile, PR2 (the recipient of an access token presented by PR1) may directly make authorization decisions based on the scopes or other claims that are optionally found in the access token. Alternatively, PR2 can make use of applicable enterprise authorization services to determine the allowed access.

This access determination can be based on the user's identity (as asserted by the "sub" claim), PR1's identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange (as asserted by the "act" claim).

If the protected resource acting in the role of PR2 needs to make a request to an additional protected resource, this initiates a new token chaining transaction. Such protected resources that both receive and request chained tokens must comply with the PR1 profile in the context of receiving tokens from the prior protected resource in the chain, and with the PR2 profile in the context of requesting new tokens for further resource access. No additional requirements are imposed on protected resources that perform both roles.

However, risks exist that must be accepted if PR2 chooses to use identities asserted by nested "act" claims within the access token. [RFC8693] states, "[f]or the purpose of applying access control policy, the consumer of a token MUST only consider the token's top-level claims and the party identified as the current actor by the ‘act’ claim. Prior actors identified by any nested ‘act’ claims are informational only and are not to be considered in access control decisions."

2.4 Option 3b (PR1 Obtains an Access Token from AS1)

PR1 performs token exchange with the authorization server AS1 in its organization to receive a new access token it can use to access PR2. However, AS1 does not generate the access token it returns to PR1. Instead, AS1 generates a JWT assertion and (acting as an OAuth client) issues an assertion grant request to AS2 using the assertion AS1 generated to receive a new access token generated by AS2 that PR1 can use to access PR2

2.4.1 Protected Resource 1 (PR1) Profile

This section imposes requirements on and describes the actions taken by PR1 to obtain a new access token from an authorization server valid for use by PR1 at PR2. When interacting with the authorization server AS1 and with PR2, PR1 is acting in the role of an OAuth client. If PR2 then needs to exchange the access token to access PR3, then PR2 would adopt the role of PR1 as described in this profile, and PR3 would adopt the role of PR2.

2.4.1.1 Connection to the Authorization Server AS1 in PR1’s Organization

When performing token exchange, PR1 MUST authenticate to the token endpoint of AS1 using mutually authenticated TLS, in compliance with Section 2.1 of [RFC8705], using a PKI certificate and corresponding private key.

When PR1 performs token exchange at AS1 to exchange the access token AS1 issued to the client for a new access token PR1 can use to access PR2, the fields of its token exchange requests are the same as for Options 1 and 2.

PR1, when complying with this profile, MUST set the fields of its token exchange requests as follows.

grant_type	REQUIRED	Value set to "urn:ietf:params:oauth:grant-type:token-exchange" as required by Section 2.1 of [RFC8693].
client_id	REQUIRED	Value set to PR1's client_id at the authorization server as required by Section 2 of [RFC8705].

resource	OPTIONAL - at least one of "resource" or "audience" MUST be set	Set as described in [RFC8693]. The requirement that at least one of "resource" or "audience" must be set is per this profile.
audience	OPTIONAL - at least one of "resource" or "audience" MUST be set	Set as described in [RFC8693]. The requirement that at least one of "resource" or "audience" must be set is per this profile.
scope	OPTIONAL	Set as described in [RFC8693].
requested_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access_token" as described in Section 3 of [RFC8693]. The requirement that requested_token_type must be set is per this profile.
subject_token	REQUIRED	Value set to the access token sent to PR1 from its client. The requirement to include subject_token is per [RFC8693] Section 2.1. The requirement that it be set to the access token is per this profile.
subject_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access_token". The requirement to include subject_token_type is per [RFC8693] Section 2.1. The requirement that it identify an access token is per this profile.
actor_token	NOT ALLOWED	PR1 is the actor and identifies itself to the authorization server through TLS client certificate authentication per [RFC8705]; therefore, this field is not permitted. This field is optional per [RFC8693] Section 2.1 and is prohibited per this profile.
actor_token_type	NOT ALLOWED	PR1 is the actor and identifies itself to the authorization server through TLS client certificate authentication per [RFC8705]; therefore, this field is not permitted. This field is prohibited per [RFC8693] Section 2.1 when actor_token is not present.

2.4.1.2 Connection to PR2

For connections between PR1 and PR2, where PR1 is acting in an OAuth Client role, PR1 MUST comply with the requirements described in Section 2.3 (Client Connection to the Protected Resource) of the Enterprise Mission Tailored OAuth 2.0 Profile.

2.4.2 Protected Resource 2 (PR2) Profile

As described by Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile, PR2 (the recipient of an access token presented by PR1) may directly make authorization decisions based on the scopes or other claims that are optionally found in the access token. Alternatively, PR2 can make use of applicable enterprise authorization services to determine the allowed access.

This access determination can be based on the user's identity (as asserted by the "sub" claim), PR1's identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange (as asserted by the "act" claim). If the protected resource acting in the role of PR2 needs to make a request to an additional protected resource, this initiates a new token chaining transaction. Such protected resources that both receive and request chained tokens must comply with the PR1 profile in the context of receiving tokens from the prior protected resource in the chain, and with the PR2 profile in the context of requesting new tokens for further resource access. No additional requirements are imposed on protected resources that perform both roles.

However, risks exist that must be accepted if PR2 chooses to use identities asserted by nested "act" claims within the access token. [RFC8693] states, "[f]or the purpose of applying access control policy, the consumer of a token MUST only consider the token's top-level claims and the party identified as the current actor by the "act" claim. Prior actors identified by any nested "act" claims are informational only and are not to be considered in access control decisions."

3 Authorization Server (AS) Profiles

In all options, the authorization servers AS1 and AS2 MUST comply with the requirements described in Section 3 (Authorization Server Profile) of the Enterprise Mission Tailored OAuth 2.0 Profile.

3.1 Option 1 (PR1 Performs Token Exchange at AS1)

PR1 performs token exchange with the authorization server AS1 in its organization to receive an access token that it can use to access PR2. When PR1 presents the access token it received during token exchange to PR2, PR2 uses introspection to ask its authorization server AS2 to validate the token received from PR1 since the token was issued by AS1 in PR1's organization.

3.1.1 Authorization Server 1 (AS1) Profile

This section imposes requirements on and describes the actions taken by AS1 when performing token exchange with PR1 so that PR1 can obtain a new access token from an authorization server valid for use by PR1 at PR2.

3.1.1.1 Connection from PR1 to Perform Token Exchange

AS1 MUST allow token exchange only if it has authenticated PR1 using mutually authenticated TLS in compliance with Section 2.1 of [RFC8705]. PR1 MUST be registered as an OAuth client at the AS, with the subject distinguished name of PR1's PKI certificate associated with that client's registration for authentication purposes.

AS1 MUST ensure before allowing token exchange that the subject_token field in the token exchange request contains a valid, unexpired OAuth access token (compliant with the format specified in Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile). This access token to be exchanged MUST contain an "aud" claim, and the claim MUST specifically identify PR1 as a valid recipient of the token.

AS1 MUST provide the ability to set and enforce an authorization policy that determines under what conditions token exchange is permitted and how claims will be populated in the issued token. The authorization policy MUST specify which protected resources are allowed to perform token exchange. If tokens issued as a result of token exchange are to contain "scope", "resource", "aud" or similar claims, the authorization policy MUST specify the allowed values for these claims. For example, in most cases it would be desired that a new access token's "scope" claim must contain a subset of the values in the access token to be exchanged, not new values, as PR1 should not be able to obtain new authorizations that were not originally granted by the user to the client. **It is critical that each authorization server's administrators appropriately configure the token exchange authorization policy to meet the organization's security objectives; otherwise, serious privilege escalation threats may be introduced.**

Note that Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile states that issued access tokens "are not required to contain scopes or other claims conveying detailed authorization information." If they do not, the protected resource (PR2) consuming the newly issued token can make use of applicable enterprise authorization services to determine the allowed access. This access can be based on the user's identity (as asserted by the "sub" claim), PR1's identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange chain (as asserted by the "act" claim described below).

If the token exchange request passes the AS's checks, the AS will generate a new access token compliant with Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile.
Since PR1 was identified using mutually authenticated TLS, the AS MUST populate a "cnf" claim in the new access token as specified by Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile.

Each organization should perform a risk analysis to determine an appropriate policy for populating the "exp" (expiration) claim of new access token. Authorization servers SHOULD make the token expiration behavior configurable. In some cases, the appropriate position would be to ensure that the "exp" claim's value is less than or equal to the "exp" claim of the access token to be exchanged, to prevent the token exchange process from being abused to create new access tokens with longer validity than the original access token. However, there may be cases where an operation takes a lengthy amount of time and potentially involves a chain of many protected resources, where it may be necessary to extend the lifetime of exchanged tokens beyond the original token's expiration.

AS1 MUST populate an "act" claim in the new access token as specified by Section 4.1 of [RFC8693]. The "act" claim MUST contain a "sub" claim identifying PR1 and an "iss" claim identifying the AS. If an "act" claim is present in the access token to be exchanged, the AS MUST copy it into the new access token as a nested claim within the new access token's outer "act" claim. If an "act" claim is not present in the access token to be exchanged, the AS MUST add a nested "act" claim containing a "sub" claim with the identity of the client that presented the access token to be exchanged to PR1 (found in the access token's "client_id" claim) and an "iss" claim identifying the AS. Informative examples of "act" contents within issued access tokens are in Section 3.2 below.

AS1, when complying with this profile, MUST set the fields of successful token exchange responses as follows:

access_token	REQUIRED	Value set to the JWT assertion issued in response to the token exchange request. Note the requirements above on the contents of the assertion. Requirement to include this field is per [RFC8693]; requirement to set it to a JWT assertion is per this profile.
issued_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:jwt". Requirement to include this field is per [RFC8693]; requirement to set it to the particular value is per this profile.

token_type	REQUIRED	Value set to “Bearer”. Even though the issued access token must be sender constrained per [RFC8705], the RFC does not define a distinct OAuth Access Token Type in the Internet Assigned Numbers Authority (IANA) registry. Requirement to include this field is per [RFC8693]; requirement to set it to “Bearer” is per this profile.
expires_in	RECOMMENDED	As specified by [RFC8693].
scope	OPTIONAL or REQUIRED depending upon request	As specified by [RFC8693], this field is OPTIONAL if the scope is identical to the scope in the request; otherwise, this field is REQUIRED. It is acceptable for this field to be either omitted or set to an empty value if it was not present in the request or was set to an empty value in the request and is not present or is empty in the issued access token.
refresh_token	NOT ALLOWED	Token exchange pursuant to this profile cannot be used to obtain refresh tokens. If the issued access token expires and a new access token is needed, another token exchange can be performed. Expiration times in access tokens issued from a token exchange can be lengthened when necessary to minimize the need to obtain new access tokens. Future guidance may be provided on obtaining refresh tokens if warranted. This field is OPTIONAL in [RFC8693] and per this profile is NOT ALLOWED.

3.1.2 Authorization Server 2 (AS2) Profile

This section imposes requirements on and describes the actions taken by AS2 to enable PR2 to interact with AS2 to validate tokens.

3.1.2.1 Connection from PR2 to Perform Introspection

Option 1 requires PR2 to perform introspection with the authorization server AS2 in its organization to validate the token presented by PR1. When implementing Option 1, the authorization server AS2 in PR2’s organization MUST comply with the requirements described in in Section 3.4.1 (Connections with Protected Resources: Introspection) of the Enterprise

Mission Tailored OAuth 2.0 profile. The details of the arrangement between the authorization servers in the two organizations that allows for validating access tokens is out of scope for this document.

3.2 Option 2 (PR1 Performs Token Exchange at AS2)

PR1 performs token exchange with the authorization server AS2 in PR2's organization to receive an access token it can use to access PR2.

3.2.1 Authorization Server 1 (AS1) Profile

AS1 does not participate in Option 2 so there are no additional requirements on AS1.

3.2.2 Authorization Server 2 (AS2) Profile

This section imposes requirements on and describes the actions taken by AS2 when performing token exchange with PR1 so that PR1 can obtain a new access token from an authorization server valid for use by PR1 at PR2.

3.2.2.1 Connection from PR1 to Perform Token Exchange

AS2 MUST allow token exchange only if it has authenticated PR1 using mutually authenticated TLS in compliance with Section 2.1 of [RFC8705].

AS2 MUST ensure before allowing token exchange that the subject_token field in the token exchange request contains a valid, unexpired OAuth access token (compliant with the format specified in Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile). This access token to be exchanged MUST contain an "aud" claim, and the claim MUST specifically identify PR1 as a valid recipient of the token.

AS2 MUST provide the ability to set and enforce an authorization policy that determines under what conditions token exchange is permitted and how claims will be populated in the issued token. The authorization policy MUST specify which protected resources are allowed to perform token exchange. If tokens issued as a result of token exchange are to contain "scope", "resource", "aud" or similar claims, the authorization policy MUST specify the allowed values for these claims. For example, in most cases it would be desired that a new access token's "scope" claim must contain a subset of the values in the access token to be exchanged, not new values, as PR1 should not be able to obtain new authorizations that were not originally granted by the user to the client. **It is critical that each authorization server's administrators appropriately configure the token exchange authorization policy to meet the organization's security objectives; otherwise, serious privilege escalation threats may be introduced.**

Note that Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile states that issued access tokens "are not required to contain scopes or other claims conveying detailed authorization information." If they do not, the protected resource (PR2) consuming the newly issued token can make use of applicable enterprise authorization services to determine the allowed access. This access can be based on the user's identity (as asserted by the "sub" claim), PR1's identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange chain (as asserted by the “act” claim described below).

If the token exchange request passes the AS’s checks, the AS will generate a new access token compliant with Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile.
Since PR1 was identified using mutually authenticated TLS, the AS MUST populate a “cnf” claim in the new access token as specified by Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile.

Each organization should perform a risk analysis to determine an appropriate policy for populating the “exp” (expiration) claim of new access token. Authorization servers SHOULD make the token expiration behavior configurable. In some cases, the appropriate position would be to ensure that the “exp” claim’s value is less than or equal to the “exp” claim of the access token to be exchanged, to prevent the token exchange process from being abused to create new access tokens with longer validity than the original access token. However, there may be cases where an operation takes a lengthy amount of time and potentially involves a chain of many protected resources, where it may be necessary to extend the lifetime of exchanged tokens beyond the original token’s expiration.

AS2 MUST populate an “act” claim in the new access token as specified by Section 4.1 of [RFC8693]. The “act” claim MUST contain a “sub” claim identifying PR1 and an “iss” claim identifying the AS. If an “act” claim is present in the access token to be exchanged, the AS MUST copy it into the new access token as a nested claim within the new access token’s outer “act” claim. If an “act” claim is not present in the access token to be exchanged, the AS MUST add a nested “act” claim containing a “sub” claim with the identity of the client that presented the access token to be exchanged to PR1 (found in the access token’s “client_id” claim) and an “iss” claim identifying the AS. Informative examples of “act” contents within issued access tokens are in Section 3.2 below.

AS2, when complying with this profile, MUST set the fields of successful token exchange responses as follows:

access_token	REQUIRED	Value set to the JWT assertion issued in response to the token exchange request. Note the requirements above on the contents of the assertion. Requirement to include this field is per [RFC8693]; requirement to set it to a JWT assertion is per this profile.
issued_token_type	REQUIRED	Value set to “urn:ietf:params:oauth:token-type:jwt”. Requirement to include this field is per [RFC8693]; requirement to set it to the particular value is per this profile.
token_type	REQUIRED	Value set to “Bearer”. Even though the issued access token must be sender constrained per [RFC8705], the RFC does

		not define a distinct OAuth Access Token Type in the IANA registry. Requirement to include this field is per [RFC8693]; requirement to set it to “Bearer” is per this profile.
expires_in	RECOMMENDED	As specified by [RFC8693].
scope	OPTIONAL or REQUIRED depending upon request	As specified by [RFC8693], this field is OPTIONAL if the scope is identical to the scope in the request; otherwise, this field is REQUIRED. It is acceptable for this field to be either omitted or set to an empty value if it was not present in the request or was set to an empty value in the request and is not present or is empty in the issued access token.
refresh_token	NOT ALLOWED	Token exchange pursuant to this profile cannot be used to obtain refresh tokens. If the issued access token expires and a new access token is needed, another token exchange can be performed. Expiration times in access tokens issued from a token exchange can be lengthened when necessary to minimize the need to obtain new access tokens. Future guidance may be provided on obtaining refresh tokens if warranted. This field is OPTIONAL in [RFC8693] and per this profile is NOT ALLOWED.

3.3 Option 3a (PR1 Obtains a JWT Assertion from AS1 and an Access Token from AS2)

PR1 performs token exchange with the authorization server AS1 in its organization to receive a JWT assertion [RFC7523]. PR1 then sends the JWT assertion to AS2 as part of an OAuth assertion grant token request. AS2 then returns an access token to PR1 that it can use to access PR2.

3.3.1 Authorization Server 1 (AS1) Profile

This section imposes requirements on and describes the actions taken by AS1 when performing token exchange with PR1 so that PR1 can obtain a new access token from an authorization server valid for use by PR1 at PR2.

3.3.1.1 Connection from PR1 to Perform Token Exchange

AS1 MUST allow token exchange only if it has authenticated PR1 using mutually authenticated TLS in compliance with Section 2.1 of [RFC8705]. PR1 MUST be registered as an OAuth client at the AS, with the subject distinguished name of PR1's PKI certificate associated with that client's registration for authentication purposes.

AS1 MUST ensure before allowing token exchange that the subject_token field in the token exchange request contains a valid, unexpired OAuth access token (compliant with the format specified in Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile). This access token MUST contain an "aud" claim, and the claim MUST specifically identify PR1 as a valid recipient of the token.

AS1 MUST provide the ability to set and enforce an authorization policy that determines under what conditions token exchange is permitted and how claims will be populated in the issued JWT assertion. The authorization policy MUST specify which protected resources are allowed to perform token exchange. If assertions issued as a result of token exchange are to contain "scope", "resource", "aud", or similar claims, the authorization policy MUST specify the allowed values for these claims. For example, in most cases it would be desired that the "scope" claim of an issued assertion must contain a subset of the values in the access token to be exchanged, not new values, as PR1 should not be able to obtain new authorizations that were not originally granted by the user to the client. **It is critical that each authorization server's administrators appropriately configure the token exchange authorization policy to meet the organization's security objectives; otherwise, serious privilege escalation threats may be introduced.**

If the token exchange request passes the AS's checks, the AS will generate a new JWT assertion compliant with RFC 7523.

Since PR1 was identified using mutually authenticated TLS, the AS MUST populate a "cnf" claim in the new JWT assertion as specified by Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile.

Each organization should perform a risk analysis to determine an appropriate policy for populating the "exp" (expiration) claim of issued assertions. Authorization servers SHOULD make the assertion expiration behavior configurable. In some cases, the appropriate position would be to ensure that the "exp" claim's value is less than or equal to the "exp" claim of the access token to be exchanged, to prevent the token exchange process from being abused to create new assertions with longer validity than the access token to be exchanged. However, there may be cases where an operation takes a lengthy amount of time and potentially involves a chain of many protected resources, where it may be necessary to extend the lifetime of issued assertions beyond the original token's expiration.

AS1, when complying with this profile, MUST set the fields of successful token exchange responses as follows:

access_token	REQUIRED	Value set to the JWT assertion issued in response to the token exchange request. Note the requirements above on the contents of the assertion. Requirement to include this field is per [RFC8693]; requirement to set it to a JWT assertion is per this profile.
issued_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:jwt". Requirement to include this field is per [RFC8693]; requirement to set it to the particular value is per this profile.
token_type	REQUIRED	Value set to "N_A" as specified by [RFC8693].
expires_in	RECOMMENDED	As specified by [RFC8693].
scope	OPTIONAL	As specified by [RFC8693]. It is acceptable for this field to be either omitted or set to an empty value if it was not present in the request or was set to an empty value in the request and is not present or is empty in the issued access token.
refresh_token	NOT ALLOWED	Token exchange pursuant to this profile cannot be used to obtain refresh tokens. If the issued access token expires and a new access token is needed, another token exchange can be performed. Expiration times in access tokens issued from a token exchange can be lengthened when necessary to minimize the need to obtain new access tokens. Future guidance may be provided on obtaining refresh tokens if warranted. This field is OPTIONAL in [RFC8693] and per this profile is NOT ALLOWED.

3.3.2 Authorization Server 2 (AS2) Profile

This section imposes requirements on and describes the actions taken by AS2 when receiving an assertion grant request from PR1 so that PR1 can obtain a new access token from an authorization server valid for use by PR1 at PR2.

3.3.2.1 Connection from PR1 in Response to an Assertion Grant Token Request

AS2 MUST return a successful response to the assertion grant token request by PR1 only if AS2 has authenticated PR1 using mutually authenticated TLS in compliance with Section 2.1 of [RFC8705]. PR1 does not need to be registered as an OAuth client at AS2. However, AS2 MUST ensure that the “cnf” field of the JWT assertion presented by PR1 corresponds with the client certificate presented by PR1 when establishing the mutually authenticated TLS session.

AS2 MUST ensure before responding successfully to the assertion grant token request from PR1 that the assertion field in the request contains a valid, unexpired JWT assertion.

AS2 MUST provide the ability to set and enforce an authorization policy that determines under what conditions an assertion grant request is permitted and how claims will be populated in the issued token. The authorization policy MUST specify which protected resources are allowed to present an assertion grant request. If tokens issued as a result of the assertion grant token request are to contain "scope", "resource", "aud", or similar claims, the authorization policy MUST specify the allowed values for these claims. For example, in most cases it would be desired that a new access token’s "scope" claim must contain a subset of the values in the assertion presented during the assertion grant request, not new values, as PR1 should not be able to obtain new authorizations that were not originally granted by the user to the client. **It is critical that each authorization server's administrators appropriately configure the token exchange authorization policy to meet the organization’s security objectives; otherwise, serious privilege escalation threats may be introduced.**

Note that Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile states that issued access tokens "are not required to contain scopes or other claims conveying detailed authorization information." If they do not, the protected resource (PR2) consuming the newly issued token can make use of applicable enterprise authorization services to determine the allowed access. This access can be based on the user's identity (as asserted by the "sub" claim), PR1’s identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange chain (as asserted by the "act" claim described below).

If the token request passes the AS’s checks, the AS will generate a new access token compliant with Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile.
Since PR1 was identified using mutually authenticated TLS, the AS MUST populate a "cnf" claim in the new access token as specified by Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile.

Each organization should perform a risk analysis to determine an appropriate policy for populating the "exp" (expiration) claim of new access token. Authorization servers SHOULD make the token expiration behavior configurable. In some cases, the appropriate position would be to ensure that the "exp" claim’s value is less than or equal to the "exp" claim of the access token to be exchanged, to prevent the token exchange process from being abused to create new access tokens with longer validity than the original access token. However, there may be cases where an operation takes a lengthy amount of time and potentially involves a chain of many protected resources, where it may be necessary to extend the lifetime of exchanged tokens beyond the original token’s expiration.

AS2 MUST populate an "act" claim in the new access token as specified by Section 4.1 of [RFC8693]. The "act" claim MUST contain a "sub" claim identifying PR1 and an "iss" claim identifying the AS. If an "act" claim is present in the access token to be exchanged, the AS MUST copy it into the new access token as a nested claim within the new access token's outer "act" claim. If an "act" claim is not present in the access token to be exchanged, the AS MUST add a nested "act" claim containing a "sub" claim with the identity of the client that presented the access token to be exchanged to PR1 (found in the access token's "client_id" claim) and an "iss" claim identifying the AS. Informative examples of "act" contents within issued access tokens are in Section 3.5 below.

AS2, when complying with this profile, MUST set the fields of successful assertion grant token responses as follows:

access_token	REQUIRED	Value set to the access token issued in response to the token request. Note the requirements above on the contents of the access token.
token_type	REQUIRED	Value set to "Bearer". Even though the issued access token must be sender constrained per [RFC8705], the RFC does not define a distinct OAuth Access Token Type in the IANA registry.
expires_in	RECOMMENDED	As specified by [RFC6749].
scope	OPTIONAL or REQUIRED depending upon request	As specified by [RFC6749], this field is OPTIONAL if the scope is identical to the scope in the request; otherwise, this field is REQUIRED. It is acceptable for this field to be either omitted or set to an empty value if it was not present in the request or was set to an empty value in the request and is not present or is empty in the issued access token.

3.4 Option 3b (PR1 Obtains an Access Token from AS1)

PR1 performs token exchange with the authorization server AS1 in its organization to receive a new access token it can use to access PR2. However, AS1 does not generate the access token it returns to PR1. Instead, AS1 generates a JWT assertion and (acting as an OAuth client) issues an assertion grant request to AS2 using the assertion AS1 generated to receive a new access token generated by AS2 that PR1 can use to access PR2.

3.4.1 Authorization Server 1 (AS1) Profile

This section imposes requirements on and describes the actions taken by AS1 when performing token exchange with PR1 so that PR1 can obtain a new access token from an authorization server valid for use by PR1 at PR2. When interacting with the authorization server AS2, AS1 is acting in the role of an OAuth client to present an assertion grant request to AS2.

3.4.1.1 Connection from PR1 to Perform Token Exchange

AS1 MUST allow token exchange only if it has authenticated PR1 using mutually authenticated TLS in compliance with Section 2.1 of [RFC8705]. PR1 MUST be registered as an OAuth client at the AS, with the subject distinguished name of PR1's PKI certificate associated with that client's registration for authentication purposes.

AS1 MUST ensure before allowing token exchange that the subject_token field in the token exchange request contains a valid, unexpired OAuth access token (compliant with the format specified in Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile). This access token to be exchanged MUST contain an "aud" claim, and the claim MUST specifically identify PR1 as a valid recipient of the token.

AS1 MUST provide the ability to set and enforce an authorization policy that determines under what conditions token exchange is permitted and how claims will be populated in the JWT assertion it generates. The authorization policy MUST specify which protected resources are allowed to perform token exchange. If assertions generated as a result of the token exchange request from PR1 are to contain "scope", "resource", "aud", or similar claims, the authorization policy MUST specify the allowed values for these claims. For example, in most cases it would be desired that an assertion's "scope" claim must contain a subset of the values in the access token to be exchanged, not new values, as PR1 should not be able to obtain new authorizations that were not originally granted by the user to the client. **It is critical that each authorization server's administrators appropriately configure the token exchange authorization policy to meet the organization's security objectives; otherwise, serious privilege escalation threats may be introduced.**

Note that Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile states that issued access tokens "are not required to contain scopes or other claims conveying detailed authorization information." If they do not, the protected resource (PR2) consuming the newly issued token can make use of applicable enterprise authorization services to determine the allowed access. This access can be based on the user's identity (as asserted by the "sub" claim), PR1's identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange chain (as asserted by the "act" claim described below).

If the token exchange request passes the AS's checks, the AS will generate a new JWT assertion compliant with Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile. AS1 will then present the JWT assertion to AS2 to obtain an access token for PR1's use.

AS1, when complying with this profile, MUST set the fields of successful token exchange responses as follows:

access_token	REQUIRED	Value set to the access token issued by AS2 in response to the assertion grant request by AS1.

		Requirement to include this field is per [RFC8693]; requirement to set it to an access token is per this profile.
issued_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access-token". Requirement to include this field is per [RFC8693]; requirement to set it to the particular value is per this profile.
token_type	REQUIRED	Value set to "Bearer". Even though the issued access token must be sender constrained per [RFC8705], the RFC does not define a distinct OAuth Access Token Type in the IANA registry. Requirement to include this field is per [RFC8693]; requirement to set it to "Bearer" is per this profile.
expires_in	RECOMMENDED	As specified by [RFC8693].
scope	OPTIONAL or REQUIRED depending upon request	As specified by [RFC8693], this field is OPTIONAL if the scope is identical to the scope in the request; otherwise, this field is REQUIRED. It is acceptable for this field to be either omitted or set to an empty value if it was not present in the request or was set to an empty value in the request and is not present or is empty in the issued access token.
refresh_token	NOT ALLOWED	Token exchange pursuant to this profile cannot be used to obtain refresh tokens. If the issued access token expires and a new access token is needed, another token exchange can be performed. Expiration times in access tokens issued from a token exchange can be lengthened when necessary to minimize the need to obtain new access tokens. Future guidance may be provided on obtaining refresh tokens if warranted. This field is OPTIONAL in [RFC8693] and per this profile is NOT ALLOWED.

3.4.1.2 Connection to AS2 to Request Access Token Using Assertion Grant

In order to complete token exchange with PR1, AS1 must generate a JWT assertion that it can use to obtain a new access token from AS2 for use at PR2.

When AS1 presents the JWT assertion it generated to AS2 as part of an assertion grant token request, the corresponding fields in the request MUST be as follows.

grant_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:jwt".
assertion	REQUIRED	Value set to the JWT assertion generated by AS1.
scope	OPTIONAL or REQUIRED depending upon request	As specified by [RFC8693], this field is OPTIONAL if the scope is identical to the scope in the request; otherwise, this field is REQUIRED. It is acceptable for this field to be either omitted or set to an empty value if it was not present in the request or was set to an empty value in the request and is not present or is empty in the issued access token.

3.4.2 Authorization Server 2 (AS2) Profile

This section imposes requirements on and describes the actions taken by AS2 when responding to an assertion grant request from AS1 to obtain a new access token for use by PR1 at PR2. When interacting with the authorization server AS2, AS1 is acting in the role of an OAuth client to present an assertion grant request to AS2.

3.4.2.1 Connection from AS1 to Request Access Token Using Assertion Grant

AS2 MUST respond successfully to an assertion grant request from AS1 only if AS2 has authenticated AS1 using mutually authenticated TLS in compliance with Section 2.1 of [RFC8705].

AS2 MUST ensure before successfully responding to an assertion grant request that the assertion field in the request contains a valid, unexpired JWT assertion.

AS2 MUST provide the ability to set and enforce an authorization policy that determines under what conditions to respond successfully to the assertion grant request and how claims will be populated in the issued token. The authorization policy MUST specify which authorization servers are allowed to present an assertion grant request. If tokens issued in response to the assertion grant request are to contain "scope", "resource", "aud", or similar claims, the authorization policy MUST specify the allowed values for these claims. For example, in most cases it would be desired that a new access token’s "scope" claim must contain a subset of the values in the assertion presented by AS1, not new values, as AS1 should not be able to obtain new authorizations that were not originally granted by the user to the client. **It is critical that each authorization server’s administrators appropriately configure the token exchange**

authorization policy to meet the organization's security objectives; otherwise, serious privilege escalation threats may be introduced.

Note that Section 3.9 of the Enterprise Mission Tailored OAuth 2.0 Profile states that issued access tokens "are not required to contain scopes or other claims conveying detailed authorization information." If they do not, the protected resource (PR2) consuming the newly issued token can make use of applicable enterprise authorization services to determine the allowed access. This access can be based on the user's identity (as asserted by the "sub" claim), PR1's identity (as asserted by the "client_id" claim), and the identity of the original client and any other protected resources involved in the token exchange chain (as asserted by the "act" claim described below).

If the assertion grant request passes the AS's checks, the AS will generate a new access token compliant with Section 3.3 of the Enterprise Mission Tailored OAuth 2.0 Profile.
Since AS2 only identified AS1 using mutually authenticated TLS, not PR1, AS2 is unlikely be to be able to populate the "cnf" claim in the new access token unless some other means are used to obtain PR1's certificate. This will require further exploration.

Each organization should perform a risk analysis to determine an appropriate policy for populating the "exp" (expiration) claim of new access token. Authorization servers SHOULD make the token expiration behavior configurable. In some cases, the appropriate position would be to ensure that the "exp" claim's value is less than or equal to the "exp" claim of the JWT assertion presented with the assertion grant request to prevent the process from being abused to create new access tokens with longer validity than the assertion. However, there may be cases where an operation takes a lengthy amount of time and potentially involves a chain of many protected resources, where it may be necessary to extend the lifetime of issued tokens beyond the expiration of the assertion.

AS2 MUST populate an "act" claim in the new access token as specified by Section 4.1 of [RFC8693]. The "act" claim MUST contain a "sub" claim identifying PR1 and an "iss" claim identifying the AS. If an "act" claim is present in the JWT assertion presented during the assertion grant request, the AS MUST copy it into the new access token as a nested claim within the new access token's outer "act" claim. If an "act" claim is not present in the assertion, the AS MUST add a nested "act" claim containing a "sub" claim with the identity of the client that presented the assertion to PR1 (found in the assertion's "client_id" claim) and an "iss" claim identifying the AS. Informative examples of "act" contents within issued access tokens are in Section 3.5 below.

AS2, when complying with this profile, MUST set the fields of successful token exchange responses as follows:

access_token	REQUIRED	Value set to the access token issued in response to the token exchange request. Note the requirements above on the contents of the access token.

		Requirement to include this field is per [RFC8693]; requirement to set it to an access token is per this profile.
issued_token_type	REQUIRED	Value set to "urn:ietf:params:oauth:token-type:access_token". Requirement to include this field is per [RFC8693]; requirement to set it to the particular value is per this profile.
token_type	REQUIRED	Value set to "Bearer". Even though the issued access token must be sender constrained per [RFC8705], the RFC does not define a distinct OAuth Access Token Type in the IANA registry. Requirement to include this field is per [RFC8693]; requirement to set it to "Bearer" is per this profile.
expires_in	RECOMMENDED	As specified by [RFC8693].
scope	OPTIONAL or REQUIRED depending upon request	As specified by [RFC8693], this field is OPTIONAL if the scope is identical to the scope in the request; otherwise, this field is REQUIRED. It is acceptable for this field to be either omitted or set to an empty value if it was not present in the request or was set to an empty value in the request and is not present or is empty in the issued access token.
refresh_token	NOT ALLOWED	Token exchange pursuant to this profile cannot be used to obtain refresh tokens. If the issued access token expires and a new access token is needed, another token exchange can be performed. Expiration times in access tokens issued from a token exchange can be lengthened when necessary to minimize the need to obtain new access tokens. Future guidance may be provided on obtaining refresh tokens if warranted. This field is OPTIONAL in [RFC8693] and per this profile is NOT ALLOWED.

3.5 Informative Examples of "act" Contents Within Issued Access Tokens

3.5.1 Option 1 Example

1. If an "act" claim is present in the access token to be exchanged:

```
{
 ...
 "act":
 {
 "sub":"PR1",
 "iss":"AS1",
 "act":
 {
  "sub":"[client_id2]",
  "iss":"AS1",
  "act":
  {
  "sub":"[client_id1]",
  "iss":"AS1"
  }
 }
 }
}
```

2. If an "act" claim is not present in the access token to be exchanged:

```
{
 ...
 "act":
 {
 "sub":"PR1",
 "iss":"AS1",
 "act":
 {
  "sub":"[client_id from access token to be exchanged]",
  "iss":"AS1"
 }
 }
}
```

3.5.2 Option 2 Example

1. If an "act" claim is present in the access token to be exchanged:

```
{
 ...
 "act":
 {
```

```
 "sub":"PR1",
 "iss":"AS4",
 "act":
 {
  "sub":"[client_id2]",
  "iss":"AS3",
  "act":
  {
  "sub":"[client_id1]",
  "iss":"AS2"
  }
 }
 }
}
```

2. If an "act" claim is not present in the access token to be exchanged:

```
{
 ...
 "act":
 {
 "sub":"PR1",
 "iss":"AS3",
 "act":
 {
  "sub":"[client_id from access token to be exchanged]",
  "iss":"AS2"
 }
 }
}
```

4 Example Token and Identity Chaining Protocol Interactions

This section is non-normative and provides examples of protocol interactions involving token exchange. These steps occur after the client obtains an access token for use at PR1 (Steps 1–5 in Figure 4 and Figure 6).

4.1 Option 1 Example (PR1 Performs Token Exchange at AS1)

1. Data request from Client to PR1 (Step 1 in Figure 3):

```
GET /resource_PR1 HTTP/1.1
Host: rs1.example.com
Authorization: Bearer [client-to-PR1-access-token]
```

2. Token Exchange request from PR1 to AS1 (Step 2 in Figure 3):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to the AS1.)

```
POST /as1/token.oauth2 HTTP/1.1
Host: as1.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-
exchange
&client_id=[PR1's client_id]&resource=[PR2 resource]
&requested_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Aaccess_token&subject_token=[client-to-PR1-access-token]
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Aaccess_token
```

3. Successful token exchange response from AS1 to PR1 (Step 3 in Figure 3):

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store

{
"access_token":"[PR1-to-PR2-access-token]",
"issued_token_type":"urn:ietf:params:oauth:token-
type:access_token",
"token_type":"Bearer",
"expires_in":60
}
```

4. Data request from PR1 to PR2 (Step 4 in Figure 3):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to PR2.)

```
GET /resource_PR2 HTTP/1.1
Host: rs2.example.com
```

```
Authorization: Bearer [PR1-to-PR2-access-token]
```

5. Introspection request from PR2 to AS2 (Step 5 in Figure 3)

(Request must be sent over a mutually authenticated TLS connection, with PR2 using its PKI certificate to authenticate itself to PR2.)

```
POST /introspect HTTP/1.1
Host: as2.example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Bearer [PR1-to-PR2-access-token]
token=[PR1-to-PR2-access-token]
```

6. Successful introspection response from AS2 to PR2 (Step 6 in Figure 3):

```
HTTP/1.1 200 OK
Content-Type: application/json
{
"active": true,
"client_id": [PR1's client_id],
"scope": "read write dolphin",
"sub": "Z5O3upPC88QrAjx00dis",
"exp": 1419356238,
}
```

PR2 then returns the requested data to PR1, which in turn returns the data to the Client.

4.2 Option 2 Example (PR1 Performs Token Exchange at AS2)

1. Data request from Client to PR1 (Step 1 in Figure 4):

```
GET /resource_PR1 HTTP/1.1
Host: rs1.example.com
Authorization: Bearer [client-to-PR1-access-token]
```

2. Token Exchange request from PR1 to AS2 (Step 2 in Figure 4):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to the AS.)

```
POST /as2/token.oauth2 HTTP/1.1
Host: as2.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-
exchange&client_id=[PR1's client_id]&resource=[PR2 resource]
&requested_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Aaccess_token&subject_token=[client-to-PR1-access-token]
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Aaccess_token
```

3. Successful token exchange response from AS2 to PR1 (Step 3 in Figure 4):

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store

{
"access_token":"[PR1-to-PR2-access-token]",
"issued_token_type":"urn:ietf:params:oauth:token-
type:access_token",
"token_type":"Bearer",
"expires_in":60
}
```

4. Data request from PR1 to PR2 (Step 4 in Figure 4):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to PR2.)

```
GET /resource_PR2 HTTP/1.1
Host: rs2.example.com
Authorization: Bearer [PR1-to-PR2-access-token]
```

PR2 then returns the requested data to PR1, which in turn returns the data to the Client.

4.3 Option 3a Example (PR1 Obtains a JWT Assertion from AS1 and an Access Token from AS2)

1. Data request from Client to PR1 (Step 1 in Figure 5):

```
GET /resource_PR1 HTTP/1.1
Host: rs1.example.com
Authorization: Bearer [client-to-PR1-access-token]
```

2. Token Exchange request from PR1 to AS1 (Step 2 in Figure 5):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to the AS.)

```
POST /as1/token.oauth2 HTTP/1.1
Host: as1.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-
exchange&client_id=[PR1's client_id]&resource=as2.example.com
&requested_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Ajwt&subject_token=[client-to-PR1-access-token]
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Aaccess_token
```

3. Successful token exchange response from AS1 to PR1 (Step 3 in Figure 5):

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store
```

```
{
"access_token":"[PR1-to-AS2-assertion]",
"issued_token_type":"urn:ietf:params:oauth:token-type:jwt",
"token_type":"Bearer",
"expires_in":60
}
```

4. Assertion Grant token request from PR1 to AS2 (Step 4 in Figure 5):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to the AS.)

```
POST /as2/token.oauth2 HTTP/1.1
Host: as2.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-
bearer&assertion=[PR1-to-AS2-assertion]
```

5. Successful assertion grant token response from AS2 to PR1 (Step 5 in Figure 5):

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store

{
"access_token":"[PR1-to-PR2-access-token]",
"token_type":"Bearer",
"expires_in":60
}
```

6. Data request from PR1 to PR2 (Step 6 in Figure 5):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to PR2.)

```
GET /resource_PR2 HTTP/1.1
Host: rs2.example.com
Authorization: Bearer [PR1-to-PR2-access-token]
```

PR2 then returns the requested data to PR1, which in turn returns the data to the Client.

4.4 Option 3b Example (PR1 Obtains an Access Token from AS1)

1. Data request from Client to PR1 (Step 1 in Figure 6):

```
GET /resource_PR1 HTTP/1.1
Host: rs1.example.com
Authorization: Bearer [client-to-PR1-access-token]
```

2. Token Exchange request from PR1 to AS1 (Step 2 in Figure 6):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to the AS.)

```
POST /as1/token.oauth2 HTTP/1.1
Host: as1.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-
exchange&client_id=[PR1's client_id]&resource=as2.example.com
&requested_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Ajwt&subject_token=[client-to-PR1-access-token]
&subject_token_type=urn%3Aietf%3Aparams%3Aoauth%3Atoken-
type%3Aaccess_token
```

3. Assertion Grant token request from AS1 to AS2 (Step 3 in Figure 6):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to the AS.)

```
POST /as2/token.oauth2 HTTP/1.1
Host: as2.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-
bearer&assertion=[AS1-to-AS2-assertion (generated by AS1)]
```

4. Successful assertion grant token response from AS2 to PR1 (Step 4 in Figure 6):

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store

{
"access_token":"[PR1-to-PR2-access-token]",
"token_type":"Bearer",
"expires_in":60
}
```

5. Successful token exchange response from AS1 to PR1 (Step 5 in Figure 5):

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store

{
"access_token":"[PR1-to-PR2-access-token]",
"issued_token_type":"urn:ietf:params:oauth:token-
type:access_token",
"token_type":"Bearer",
"expires_in":60
}
```

6. Data request from PR1 to PR2 (Step 6 in Figure 6):

(Request must be sent over a mutually authenticated TLS connection, with PR1 using its PKI certificate to authenticate itself to PR2.)

```
GET /resource_PR2 HTTP/1.1
Host: rs2.example.com
Authorization: Bearer [PR1-to-PR2-access-token]
```

PR2 then returns the requested data to PR1, which in turn returns the data to the Client.

5 Security Rationale for Profile Requirements

This section is intended to provide rationale behind the requirements in this profile to help the reader understand the reason(s) certain decisions were made.

This profile requires that the token being exchanged must contain an "aud" field, and it must identify PR1 (the entity exchanging the token). This ensures that PR1 is the intended recipient of an access token in order to exchange it for another access token. This requirement is intended to prevent stolen access tokens from being exchanged for new access tokens by an unauthorized entity. [RFC8693] does not contain this explicit requirement.

This profile requires that access tokens obtained through token exchange must identify the entire chain of clients and protected resources that held previously exchanged access tokens. The newly issued access token must contain an "act" claim that identifies the protected resource that exchanged the token, the client that sent the token to the protected resource, and any other entities involved in exchanges of other access tokens in the chain. This enables the protected resource consuming the access token to, if desired, look up authorizations or privileges associated with each entity in the chain as part of deciding what access to allow. The access tokens can still include specific authorization information (e.g. in its scope claim, resource claim, or other environment-specific claim) that protected resources could use instead of or in addition to the chain information. [RFC8693] defines the "act" claim but does not explicitly require its use.

6 Normative References

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", IETF RFC 6749, October 2012, <http://www.rfc-editor.org/info/rfc6749>.

[RFC7521] B. Campbell, C. Mortimore, M. Jones, and Y. Goland, “Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants”, IETF RFC 7521, May 2015, <http://www.rfc-editor.org/info/rfc7521>.

[RFC7523] M. Jones, B. Campbell, and C. Mortimore, “JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants”, IETF RFC 7523, May 2015, <http://www.rfc-editor.org/info/rfc7523>.

[RFC7662] Richer, J., Ed., “OAuth 2.0 Token Introspection”, IETF RFC 7662, October 2015, <http://www.rfc-editor.org/info/rfc7662>.

[RFC8693] B. Campbell, Ed., "OAuth 2.0 Token Exchange", IETF RFC 8693, January 2020, <http://www.rfc-editor.org/info/rfc8693>.

[RFC8705] B. Campbell, J. Bradley, N. Sakimura, and T. Lodderstedt, "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens", IETF RFC 8705, February 2020, < http://www.rfc-editor.org/info/rfc8705>.

[OIDC-Core] OpenID Foundation. "OpenID Connect Core 1.0 incorporating errata set 1”, November 2014, <https://openid.net/specs/openid-connect-core-1_0.html>.

7 Informative References

[OAuth-Profile] B. Abramowitz, et al. " Enterprise Mission Tailored OAuth 2.0 Profile.", February 2020, <https://www.mitre.org/publications/technical-papers/enterprise-mission-tailored-oauth-20-and-openid-connect-profiles>

[Token-Chaining] B. Abramowitz, et al. " DRAFT Token Chaining in a Single ICAM Ecosystem using OAuth Token Exchange."

Appendix A Acronyms

AS1	The authorization server in PR1's organization
AS2	The authorization server in PR2's organization
IANA	Internet Assigned Numbers Authority
ICAM	Identity, Credential, and Access Management
IETF	Internet Engineering Task Force
JWT	JSON Web Token
OV-1	Operation View 1 (High-Level Operational Concept Graphic)
PKI	Public Key Infrastructure
PR1	The protected resource initiating the token exchange protocol
PR2	The protected resource containing the data requested by the client
RFC	Request For Comments
TLS	Transport Layer Security
Claims	
act	actor
aud	audience
cnf	confirmation
exp	expiration time
iss	issuer