Workload Identity in Multi System Environments A. Schwenkschuster, Ed.
Internet-Draft   SPIRL
Intended status: Informational   3 July 2025
Expires: 4 January 2026

WIMSE Credential Exchange
draft-schwenkschuster-wimse-credential-exchange-02

Abstract

WIMSE defines Workload Identity and its representation through credentials. Typically, a credential is provisioned to the workload, allowing it to represent itself. The credential format is usually chosen by the platform. Common formats are JSON Web Tokens or X.509 certificates. However, workloads often encounter situations where a different identity or credential is required.

This document describes various situations where a workload requires another credential. It also outlines different ways this can be acchieved and compares them.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://ietf-wg-wimse.github.io/draft-ietf-wimse-s2s-protocol/draft-ietf-wimse-s2s-protocol.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-schwenkschuster-wimse-credential-exchange/.

Discussion of this document takes place on the Workload Identity in Multi System Environments Working Group mailing list (mailto:wimse@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/wimse/. Subscribe at https://www.ietf.org/mailman/listinfo/wimse/.

Source for this draft and an issue tracker can be found at https://github.com/arndt-s.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 4 January 2026.

Copyright Notice

Table of Contents

1. Introduction

Workloads operating across various platforms typically receive identity credentials in platform-specific formats such as JWT tokens, X509 certificates, Kubernetes Service Accounts Tokens, SPIFFE SVIDs, or cloud provider metadata documents. These credentials, including their format, issuer, subject, and other attributes are determined by the platform infrastructure rather than the workload itself.

When accessing external resources or other workloads, workloads must satisfy different authentication requirements specific to each resource they interact with. Resource access might require OAuth 2.0 Access Tokens, SPIFFE credentials, mutual TLS client certificates, or other authentication mechanisms that the workload cannot control or modify.

This credential mismatch creates a fundamental challenge: workloads must bridge the gap between their platform-issued identity and the various authentication requirements of the resources they need to access. Solutions typically involve credential exchange mechanisms or leveraging platform-specific functionality.

This specification:

* Defines abstract mechanisms for credential delivery and exchange in Section 2

* Examines the rationale behind credential exchange requirements in Section 3

* Documents concrete implementation patterns based on these mechanisms in Section 4

1.1. Static secrets

Credential exchange is a broad term and can be interpreted in many ways. This document focuses on the exchange of credentials that are issued to workloads on demand, such as JWTs, X.509 certificates, or other workload identity formats. It does not cover the exchange of static secrets, such as API keys or client secrets, which are typically provisioned out-of-band and do not involve a dynamic exchange process. This document rather sees static secrets as a special kind of resources that require strong authentication and authorization to access, but not as a credential exchange.

Credentials returned as by this specification are:

* short-lived
* issued on demand, but potentially cached
* are individual based on the credential used to authenticate

In contrast, static secrets are:

* medium to long-lived
* provisioned out-of-band, often manually
* do not differ based on authentication, different callers get access to the same credential

2. Mechanisms

Workloads have multiple options to acquire credentials in the way they are required. The following terms divide them into four primary mechanisms and outlines their approaches:

Initial manual provisioning
: Credentials are provisioned manually, often by an administrator. This is typically done out-of-band, such as through a configuration file, environment variable, or secret management system. This mechanism is not recommended for workload identity credentials. Within the document, this mechanism is not further discussed.

Initial platform provisioning

Credentials are issued during workload creation by the platform. The workload is "born" with them and they are available at startup. The format, scope, lifetime and other attributes are configured out of band and cannot be influenced at runtime. It is common that platforms allow only a single credential to be provisioned and only allow changes to scope.

On-demand platform issuance
: Workloads are able to obtain credentials on-demand from the platform. Parameters allow the workload to specify exactly the required format, scope, identity, lifetime, and other attributes the workload requires. No authentication is necessary to request on-demand credentials. The platform is able to strongly identify the workload so this request is typically unauthenticated from the workload's point of view. Implementations may use the workload's initial provisioned credential in the background to authenticate the request, but this is not visible to the workload.

On-demand credential exchange
: Workloads use an existing credential (provisioned manually or by the platform) to authenticate and authorize a request for a different credential. Based on parameters, the workload can specify the exact attributes of the credential it requires. The significant difference towards the on-demand platform issuance is that this is not necessarily within the platform and often an *authenticated* action.

The outlined mechanisms are from the point of view of the workload. Specific mechanisms are often implemented by a combination of other mechanisms. For example, a platform on-demand provisioning mechanism may use an initial provisioned credential in combination with a credential exchange to issue a new credential. However, from the workload's perspective it is a platform on-demand provisioning mechanism, as it is not aware of the underlying implementation details.

3. Rationale

Workloads often require credentials that differ from the ones they were initially provisioned with. This can be due to various reasons, such as changes in format, scope, identity, trust domain, or lifetime. The need for credential exchange arises when the workload needs to adapt to these changing requirements. The following sections outline the most common reasons for credential exchange, gives examples and provides recommendations for the mechanisms to use.

3.1. Change in format

Workloads may require a different format representing the same identity in the same trust domain. Some concrete examples are:

* The initial credential was an X.509 certificate but resources require application-level authentication such as JWT or Workload Identity Tokens as defined in (TODO).

* The initial credential was a JWT bound to a key to be presented along with proof of possession, but the resource does not support it and requires a bearer credential.

"Credential format" is difficult to define abstractly. Some formats are opaque to the workload and should remain that way. For instance, how an OAuth 2.0 Bearer token is constructed, and whether it carries claims or not, is not a concern of the workload. That a bearer token is required, however, is known to the workload. So a change in format between a bearer token and an X.509 certificate is certainly a change in format the workload can require. A different encoding of a bearer token, on the other hand, is not and this specification does not address those cases.

Mechanism	Recommendation	Reason
Initial platform provisioning	Caution	Risks preemptively issuing credentials that aren't used. See security considerations (Section 5.3) for details.
On-demand platform issuance	Prefer	Credentials are issued on a need basis, allowing the workload to specify the format it requires & keep lifetime short.
On-demand credential exchange	Caution	Requires an additional credential to authenticate the exchange. See security considerations (Section 5.2) for details.

Table 1

3.2. Change in scope

A credential in the same format may represent the same identity, but is scoped differently. Examples are:

* A JWT credential with an audience set to interact with the workload platform, but access to other workloads are required. The workload is in need of JWTs with different, dedicated audiences.

* An X.509 credential is constrained to a certain key usage, but the workload requires difference usage bits set. For instance, the existing certificate allows for digitalSignature but keyEncipherment or dataEncipherment is required.

Generally, scope should already be present and configured appropriately with the workload platform only issuing narrowly scoped credentials to the workload. However, the platform may only support the provisioning of a single credential and doesn't allow custom scoping.

Mechanism	Recommendation	Reason
Initial platform provisioning	Caution	Risk of over-provisioning and over-scoping. Particularly when different scopes are required but initial provisioning only allows a single credential. See security considerations (Section 5.3) for details.
On-demand platform issuance	Prefer	Credentials are issued on a need basis and can be scoped to the exact requirements of the workload.
On-demand credential exchange	Caution	Requires an additional credential to authenticate the exchange. See security considerations (Section 5.2) for details.

Table 2

3.3. Change in identity

A workload may be known under multiple identities. For example:

* A workload identity representing an exact physical instance of the workload may be eligible for a workload identity representing a logical unit that groups many physical instances together. Another example is a workload running in a specific region being eligible for a broader, geographically scoped identity.

* A workload that can act on behalf of other workloads. These workloads often are part of infrastructure such as API gateways, proxies, or service meshes in container environments.

Mechanism	Recommendation	Reason
Initial platform provisioning	Neutral. Avoid for on-behalf-of situations.	The authors believe that on-behalf-of should be an explicit operation and not by default to avoid ambiguity and keep trust boundaries clear.
On-demand platform issuance	Prefer	Credentials are issued on a need basis and can be scoped to the exact requirements of the workload.
On-demand credential exchange	Prefer for on-behalf-of situations.	Requires an additional credential to authenticate the exchange. See security considerations (Section 5.2) for details.

Table 3

3.4. Change in trust domain

A provisioned workload identity is often part of a trust domain that is coupled to infrastructure or deployment. Workloads often interact with other workloads or access outside resources located in different trust domains. This may require the client workload to retrieve an identity of the other trust domain. Examples here include:

* Federation (a workload identity federates to a identity in a different trust domain). In existing workload identity environment OAuth2 with Token Exchange (TODO) and Assertion framework (TODO) are popular.

* A workload requires a credential of "higher trust" to interact with other workloads. This "higher trust" is facilitated by another trust domain. For instance, a workload may require a WebPKI certificate to offer a service to clients with "default" trust stores.

Mechanism	Recommendation	Reason
Initial platform provisioning	Avoid	Initial provisioning is limited to the issuer of the workload platform. Making initial provisioned credentials multi-issuer creates ambiguity.
On-demand platform issuance	Neutral	See On-behalf-of exchange pattern (Section 4.2) for a combination of on-demand provisioning and exchange.
On-demand credential exchange	Prefer	A change in trust domain indicates a different issuer.

Table 4

3.5. Change in lifetime

Credentials often come with time restrictions, or usage may be restricted based on token lifetime. For instance:

* A resource denies the long-lived workload credential based on a maximum lifetime policy.

* An initial provisioned credentials has expired and renewal is unsupported.

* A credential with shorter lifetime would reduce replay risk.

Mechanism	Recommendation	Reason
Initial platform provisioning	Caution	Creates unnecessary long-lived credentials that are difficult to protect.
On-demand platform issuance	Prefer	Individual lifetimes that fit the exact need are possible.
On-demand credential exchange	Neutral	May be used to reduce lifetime, but should be avoided to increase or expand lifetime as a credential that expires later is effectively a higher trust. See security considerations (Section 5.1) for details.

Table 5

3.6. Missing provisioning support

A workload platform may not support the provisioning of credentials required by the workload. would likely fall under the reasons above, but it’s a very common reason and often falls into multiple categories. As an example:

* Workload platform provisions identity and credential in the form of a simple signed document that carries the attributes attested by the platform, but gives not access in any way.

Mechanism	Recommendation	Reason
Initial platform provisioning	-	This section applies when initial provisioning is not supported.
On-demand platform issuance	-	This section applies when on-demand provisioning is not supported.
On-demand credential exchange	Neutral	Credential exchange may be used when no other mechanism is supported or supports the desired outcome. However, credential exchange still requires authentication. See security considerations (Section 5.2) for details.

Table 6

3.7. Combinations

Reasons for requesting or exchanging credentials are often not binary. A change in trust domain is effectively a change in identity as well. A change in format can require a change in trust domain, because formats come with different trust structures and security promises. For example, a trust domain issuing JSON Web Tokens may not be able to issue WebPKI certificates.

4. Exchange patterns

4.1. Format-specific exchange

The existing trust and identity framework often consist of a protocol or framework to exchange credentials. Leveraging this makes use of existing adoption and specific guidelines.

The following bullets give an overview of the existing patterns and when to use them based on the needs given above:

* OAuth Token Exchange [RFC8693] is:
  - meant for a change in scope.
  - meant for a change in identity.

- to a certain extend meant for a change in format (limited).

- NOT meant for a change in trust domain.

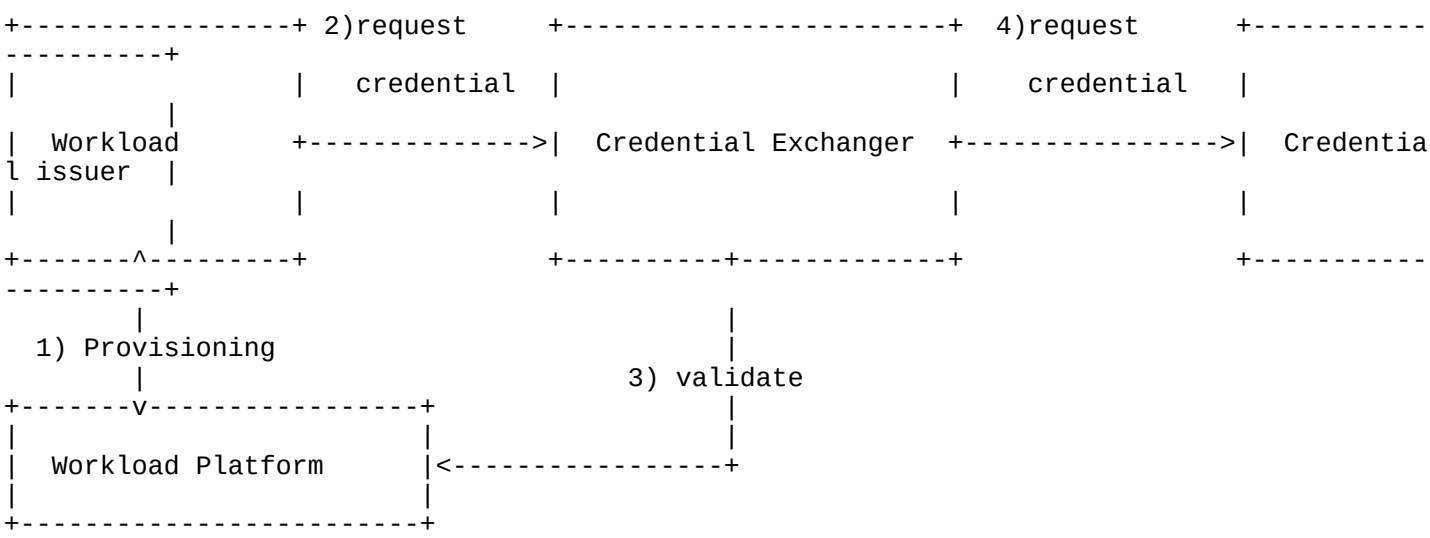
* OAuth Assertion Framework [RFC7521] is:

- meant for a change in trust domain. As a result of the change in trust domain, a change in identity, scope and, potentially, format is unavoidable but not the primary use case.

- NOT meant for exchanges within a trust domain.

4.2. On-behalf-of exchange

Workload environments can be highly dynamic and connected with a high variety of resources protected by different identity frameworks and formats. A format-agnostic component that exchanges credentials on behalf of the workload may be desired to remain control of credential issuance. For instance, it might enforce policy, collect audit trails, or aid management.



1. The Workload Platform issues credential to the workload. This can be either "initial", during workload startup or "on-demand", once the workload requires it. See Section 2 for more details.

2. The Workload requests a new credential from the Credential Exchanger by specifying at least the issuer, format, and identity. Potentially, it also specifies lifetime and scope. It authenticates itself with the credential it has received from the Workload Platform.

3. The Credential Exchanger validates the credential it receives. For simplicity, the diagram shows this as a interaction with the Workload Platform, but other means of validations are also possible.

4. The Credential Exchanger requests a credential from the Credential Issuer. Also, for simplicity this step shows the interaction with a third party. However, this may also be the Workload Platform itself. Authentication and other step details depend on the scenario, format, and trust framework.

The author believes that a specific protocol that fits all credential formats and trust frameworks is infeasible while maintaining the existing security promises. He rather believes that a profile for each scenario is the best way forward and welcomes everyone to profile this document for their concrete use cases. As a general guidance it is recommended to:

* narrowly scope the scenarios, instead of building a one-fits-all exchange for a specific format.

* decouple authentication and access control from the actual exchange as best as possible. For example, a credential of one profile should be allowed as a means of authentication to exchange to a credential of a different profile, whether or not the profiles are aware of each other.

* allow the workload to specify at least issuer, identity and format when requesting a credential. Lifetime and scope could be optionally specified, based on the need and support for it.

* keep multi-stepped issuance in mind. Some formats and trust frameworks may require the workload to perform challenges, like responding to a nonce or providing a signature.

The "Credential Exchanger" shown in the figure MAY be the Workload Platform itself that offers this capability. It MAY be offered during a "re-provisioning" without authentication.

5. Consideration

5.1. Credential exchange cannot increase trust

A credential exchange is an authenticated method to retrieve credential(s). Thus, the issued credential cannot be given a higher trust level than the credential that was used to authenticate the request. This is particularly relevant when a required credential, due to its format and framework, is of a higher trust than the one that was used to authenticate the request. This includes exchanging credentials without proof of key possession for credentials that do carry proof of possession.

These situations are not recommended. Workloads SHOULD be provisioned with the credential of the highest trust and only retrieve less-trusted credentials via credential exchange.

Alternatively, the authentication request should be enriched with additional identification that increases the level of authentication. For example, along with authentication, the workload would provide additional proof of platform attestation.

5.2. Credential exchange cannot replace initial or platform provisioning

Because credential exchange is authenticated it cannot be the first credential that is issued. Without an initial credential or a platform on-demand requested credential a workload cannot facilitate credential exchange, as there is no proof the workload is eligible for the requested credential.

5.3. Initial provisioning comes with over-provisioning risk

Provisioning credentials preemptively risks being exposed to overprovisioning credentials that are not required. For example, with initial provisioning, every workload is provisioned with a default credential, even though some don’t require it. This unnecessarily increases the risk of those credentials being exposed.

On-demand-based provisioning, on the other hand, only issues credential when requested and mitigates this. They are exactly in the scope, format, identity and lifetime that are required. This can significantly decrease the number of unnecessarily issued and provisioned credentials.

5.4. Expanding credential lifetime

A change in lifetime of a credential can be critical if it can be used to effectively keep a credential alive. One example is an issued short-lived bearer credential that can be used to exchange for a new, longer-lived credentials. Thus, it is highly recommended to only use on-demand provisioning to re-request a new credential.

On the other hand, it is valid to leverage token exchange to request a shorter-lived credential whose lifetime is within the bound of the credential used for authenticating the request.

If an exchange to a longer-lived credential is required, it is recommended to prevent this from re-occurring and deploy policy to not allow a continuous exchange to longer-lived credentials.

5.5. Involvement of human, transactional or other contextual credentials

Although this document focuses heavily on workload identity, workloads often deal with other credentials carrying caller, transactional, or contextual information. This could include an access token of the caller used to authorize the request. or an OAuth Transaction Token that was part of the request coming from another workload carrying transactional data.

These credentials and their formats, lifetime, scope, etc. are not covered by this document. However, they may be used as parameters or authentication to request additional credentials that combine multiple identities into a single credential.

Some concrete examples are:

* An access token and a workload identity credentials are used to request an OAuth Transaction Token.

* An on-behalf-of scenario where a workload identity is used as actor, and a different, contextual credential unrepresentative of the workload is used as a subject in an OAuth Token Exchange.

On-demand platform provisioning or credential exchange MAY be used to issue any of those contextual credentials to the workload. Existing contextual credentials MAY be supplied as parameters. Initial-based provisioning is not suitable with existing contextual credentials as it does not support parameters. In situations where the workload’s identity does not play a role and only the contextual credentials are used as authentication, credential exchange is the preferred mechanism.

5.6. Credential formats supporting offline attenuation

Some credential formats allow the scope of the credential to be reduced offline, without interaction to an issuing party ("offline attenuation"). In these situations no exchange or on-demand provisioning is required and workloads can "act on their own." Examples of these formats are [Macaroons] or [Biscuit] tokens. The provisioning of a credential that supports offline attenuation is still required in the first place.

6. Conventions and Definitions

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

7. Security Considerations

TODO Security

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC7521] Campbell, B., Mortimore, C., Jones, M., and Y. Goland, "Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants", RFC 7521, DOI 10.17487/RFC7521, May 2015, <https://www.rfc-editor.org/rfc/rfc7521>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8693] Jones, M., Nadalin, A., Campbell, B., Ed., Bradley, J., and C. Mortimore, "OAuth 2.0 Token Exchange", RFC 8693, DOI 10.17487/RFC8693, January 2020, <https://www.rfc-editor.org/rfc/rfc8693>.

9.2. Informative References

[Biscuit] "Biscuit, a bearer token with offline attenuation and decentralized verification", n.d., <https://doc.biscuitsec.org/reference/specifications>.

[Macaroons] Birgisson, A., Politz, J. G., Erlingsson, U., Vrable, M., and M. Lentczner, "Cookies with Contextual Caveats for Decentralized Authorization in the Cloud", 2014, <https://theory.stanford.edu/~ataly/Papers/macaroons.pdf>.

Appendix A. Document History

// RFC Editor: please remove before publication.

A.1. draft-schwenkschuster-wimse-credential-exchange-02

* Rephrased introduction
* Added scope consideration for static secrets
* Moved to a 4-level mechanism classification, added manual, rephrased existing ones to make it more clear what is platform-based.

A.2. draft-schwenkschuster-wimse-credential-exchange-01

* Fix typo that wrongly said OAuth2 assertion flow is not meant for inter-trust domain exchanges (meant was "intra").
* Rephrased X509 change of scope example to be more clear.
* Sharpened ways of provisioning, renamed "provisioning" to "initial provisioning" and "re-provisioning" to "on-demand provisioning".
* Add "Change in lifetime" need.
* Add considerations for the involvement of contextual, transactional and human credentials
* Add consideration for credential formats supporting offline-attenuation.
* Describe "Credential Exchanger" pattern.
* Clean up for IETF 122.

A.3. draft-schwenkschuster-wimse-credential-exchange-00

* Initial individual draft & write up.

Acknowledgments

Big shoutout to the WIMSE token exchange design team (Dean Saxe, Yaroslav Rosomakho, Andrii Deinega, Dmitry Izumskiy, Ken McCracken and George Fletcher) that have done amazing groundlaying work in this area.

Contributors

Ken McCracken
Google

Marcel Levy
SPIRL

Andrew McCormick
Aembit

Author’s Address

Arndt Schwenkschuster (editor)
SPIRL
Email: arndts.ietf@gmail.com