Workload Identity in Multi System Environments
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

Y. Rosomakho
Zscaler
J. Salowey
CyberArk

2 March 2026

Workload Identifier
draft-ietf-wimse-identifier-02

Abstract

This document defines a canonical identifier for workloads, referred to as the Workload Identifier. A Workload Identifier is a URI that uniquely identifies a workload within the context of a specific trust domain. This identifier can be embedded in digital credentials, including X.509 certificates and security tokens, to support authentication, authorization, and policy enforcement across diverse systems. The Workload Identifier format ensures interoperability, facilitates secure identity federation, and enables consistent identity semantics.

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-identifier/draft-ietf-wimse-identifier.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-wimse-identifier/.

Discussion of this document takes place on the Workload Identity in Multi System Environments 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/ietf-wg-wimse/draft-ietf-wimse-identifier.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

In modern distributed systems, workloads such as services, applications, or containerised tasks require cryptographically verifiable identities to support secure communication, access control, and auditability. As systems scale across trust domains, administrative boundaries, and heterogeneous platforms, the need for a consistent and interoperable identifier format becomes critical.

This document defines the Workload Identifier, a URI-based [URI] identifier intended to uniquely represent a workload within the context of an issuing authority. The identifier is designed to be stable, globally unique within a given trust domain, and suitable for use in digital credentials such as X.509 certificates , JSON Web Tokens (JWTs, [JWT]), and other security artifacts.

The Workload Identifier format is simple yet expressive. It enables organisations to define trust boundaries, delegate identity management, and identify workload instances and logical workloads in a uniform way across service meshes, cloud environments, and on-premises infrastructure. This specification defines the Workload Identifier used by the Workload Identity in Multi-System Environments (WIMSE) architecture [ARCH]. The format is defined in a general manner so that it can also be used by other systems that require stable, URI-based workload identities.

The primary goals of this specification are:

* To define the syntax and semantics of a Workload Identifier.

* To establish requirements for issuers and consumers of such identifiers.

* To promote interoperability across different identity systems and domains.

This document does not prescribe how identifiers are issued or verified. Instead, it focuses on the identifier' s format, uniqueness guarantees, and its relationship to trust domains.

2. 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.

3. Terminology

The following terms are used throughout this document:

Workload: An independently addressable and executable software entity. This may include microservices, containers, virtual machines, serverless functions, or similar components that initiate or receive network communications.

Workload Identifier: A URI-based identifier that uniquely represents a workload within a specific trust domain. A Workload Identifier MAY refer to a logical workload consisting of multiple instances, or to a specific workload instance, depending on the policies of the trust domain that issued the identifier. The identifier is intended to be included in security credentials and interpreted within the scope of an issuing authority.

Trust Domain: A security boundary defined and controlled by a single administrative authority. A trust domain establishes its own rules for identity issuance, validation, and policy enforcement.

Issuer: An entity responsible for assigning and validating Workload Identifiers.

Consumer: An entity that evaluates, verifies or uses a Workload Identifier, typically as part of authentication or authorisation decisions. This includes relying parties, verifiers, and policy enforcement points.

4. Workload Identifier Specification

A Workload Identifier is a URI [URI] that uniquely identifies a workload. It encodes both the trust domain and a workload-specific path, enabling unambiguous identification of workloads across administrative and organisational boundaries.

The identifier is designed to be stable and suitable for inclusion in digital credentials such as X.509 certificates and security tokens. This section defines the format, structure, and associated requirements for Workload Identifiers.

4.1. URI Requirements

A Workload Identifier MUST be an absolute URI, as defined in Section 4.3 of [URI]. In addition the URI MUST include a non-empty authority component that identifies the trust domain within which the identifier is scoped.

The scheme and scheme-specific syntax are not defined by this specification. The URI format allows different schemes (e.g., spiffe as defined in [SPIFFE-ID], wimse defined in Section 4.4) depending on deployment requirements. Example identifiers:

```
spiffe://incubation.example.org/ns/experimental/analytics/ingest
wimse://trust.corp.example.com/workload/af3e86cb-7013-4e33-b717-11c4edd25679
```

A Workload Identifier URI MUST NOT contain a query component, a fragment component, user information, or a port component.

Implementations that generate, parse, or otherwise process Workload Identifiers MUST support identifiers with a total length of at least 2048 bytes. Workload Identifiers SHOULD NOT exceed 2048 bytes in length.

Individual Workload Identifier schemes MAY define additional syntax or processing requirements, provided they do not conflict with the requirements defined in this document.

4.2. Scheme Specific Portion

The format and semantics of the scheme-specific part of the URI are determined by the issuer within the trust domain. The issuer defines the granularity at which identities are assigned.

A Workload Identifier MAY represent a specific workload instance, or a logical workload consisting of multiple instances that share the same identity within the trust domain.

Multiple instances MAY share the same Workload Identifier when they are intended to be treated as the same workload for the purpose of authentication, authorization, and auditing.

The scheme-specific part of the URI MAY be an opaque value that is only meaningful to the issuing authority, or it MAY encode structured information used within the trust domain.

Some examples of these concepts are given below:

* Opaque identifier

spiffe://prod.trust.domain/89a6ec51-f877-44c0-9501-b213597f2d1d

* Application role

spiffe://prod.trust.domain/ns/prod-01/sa/foo-service

* Specific instance of application role

spiffe://prod.trust.domain/ns/prod-01/sa/foo-service/iid-1f814646-87b5-4e26-bb55-1d13caccdd8d

* Specific code for an application role

spiffe://prod.trust.domain/foo-service/sha256/c4dbb1a06030e142cb0ed4be61421967618289a19c0c7760bdd745ac67779ca7

Other concepts may be defined within the trust domain depending on what is important in the system and what information is available when the identity is issued. A trust domain should define the scheme specific portion of the URI to meet their auditing and authorization needs.

4.3. Trust Domain Association

The authority component of the URI defines the trust domain which is responsible for issuing, validating, and managing Workload Identifiers within its scope. The trust domain SHOULD be a fully qualified domain name belonging to the organization defining the trust domain to help provide uniqueness for the trust domain identifier. While IP addresses are allowed as host names in the URI encoding rules, they MUST NOT be used to represent trust domains except in the case where they are needed for compatibility with legacy naming schemes.

Workload Identifiers are interpreted in the context of the trust domain that issued the credential. Identifiers with identical path components but different trust domains represent different workloads.

Issuers within a trust domain MUST ensure uniqueness of all Workload Identifiers they assign.

4.4. The "wimse" URI Scheme

A Workload Identifier using the wimse scheme has the generic form:

wimse://<trust-domain>/<path>

The URI MUST satisfy all requirements defined in Section 4.1.

The structure of the path component is deployment-specific and is not interpreted by this specification.

Examples:

```
wimse://trust.example.com/service/payment
wimse://trust.example.com/service/payment/instance/1234
wimse://prod.corp.example/workload/89a6ec51-f877-44c0-9501-b213597f2d1d
```

4.5. Stability and Uniqueness

Workload Identifiers are intended to be stable over time. An identifier assigned to a workload SHOULD NOT be reassigned to a different workload unless explicitly intended by the policies of the trust domain. Multiple workload instances MAY share the same Workload Identifier when they represent the same logical workload within the trust domain.

4.6. Workload Identifier Origin

A Workload Identifier Origin is a specification of a namespace under which a Workload Identifier is meaningful for a given use case. An origin consists of the URI scheme and trust domain components of a Workload Identifier, omitting the path component.

Workload Identifier Origins serve as hints about the set of identifiers an entity may present in a particular protocol instance or usage context without revealing specific identifier.

Examples of Workload Identifier Origins:

```
spiffe://prod.trust.domain
wimse://trust.corp.example.com
```

5. Identifier Interpretation and Mapping

Workload Identifiers are carried in credentials and tokens and are used for authentication, authorization, and auditing. However, the identifier itself does not define how a workload is reached over the network.

In many deployments, workloads are accessed using external handles such as DNS names, service names, load balancer addresses, or routing paths. These handles are deployment-specific and do not necessarily match the Workload Identifier presented in credentials.

To enable correct authentication decisions, implementations MUST support a deployment-defined mapping between the external handle used to access a workload and the Workload Identifier expected for that workload.

This mapping is outside the scope of this specification and MAY be provided by configuration, service discovery systems, orchestration platforms, or other local policy mechanisms.

Consumers MUST NOT assume that the Workload Identifier can be derived from network-layer information such as IP address, DNS name, or request path without such mapping.

Deployments using Workload Identifiers with the WIMSE credential formats defined in [WIMSE-CREDENTIALS] MUST ensure that a consistent mapping exists between workload access handles and the Workload Identifiers contained in credentials.

6. Usage in Credentials and Tokens

Workload Identifiers are designed to be embedded in cryptographic credentials and security tokens that are used to assert the identity of workloads during authentication, authorisation, and auditing. Representation of workload identifier in commonly used formats is defined in [WIMSE-CREDENTIALS].

7. Security Considerations

The Workload Identifier is intended to be used as a stable, verifiable identity for workloads. Its use in cryptographic credentials means it must be protected against spoofing, ambiguity, and misinterpretation. This section outlines security considerations for issuers, consumers, and system designers.

7.1. URI Parsing and Processing Considerations

Workload Identifiers are encoded as URIs and therefore rely on correct and secure URI parsing. Implementations MUST apply a standards-compliant URI parser.

Incorrect URI parsing can result in misinterpretation of identifier components, security policy bypass, or inconsistent trust domain evaluation across implementations.

Implementations MUST enforce the URI requirements defined in this document, including the absence of query, fragment, user information, and port components. Failure to validate these constraints may allow identifiers to carry unintended or ambiguous semantics.

Implementations MUST also take care to handle Workload Identifiers of the maximum supported length without causing excessive memory allocation, resource exhaustion, or denial-of-service conditions. Parsers SHOULD impose reasonable internal limits and reject identifiers that exceed implementation-defined constraints, consistent with the length requirements in this document.

7.2. Identifier Authenticity

Workload Identifiers MUST only be considered authenticated when presented in a credential or token that has been cryptographically verified. An identifier received outside such a context, such as a plaintext string in a request, MUST NOT be treated as authenticated.

Consumers MUST treat a Workload Identifier as authenticated only when it is obtained from a credential or token that has been successfully validated according to the rules of the credential format and the deployment policy.

Validation requirements for credentials carrying Workload Identifiers are defined in [WIMSE-CREDENTIALS] and in the protocols that use those credentials.

7.3. Trust Domain Validation

Consumers MUST validate that the trust domain in the Workload Identifier matches an expected or explicitly trusted domain. Failure to do so may allow identifiers from unauthorised domains to be accepted as legitimate.

Where appropriate, consumers should maintain an allowlist of trusted domains or trusted issuing authorities.

7.4. Identifier Reuse and Collision

Issuers SHOULD ensure that Workload Identifiers are not reused across different workloads unless such reuse is intentional and well-scoped. Reassignment of identifiers to unrelated entities can result in privilege escalation or confusion in audit trails.

Consumers SHOULD assume that identifiers are permanent within their domain of interpretation and treat unexpected reuse with suspicion.

7.5. Information Disclosure

Because Workload Identifiers may encode topological or semantic information, they may inadvertently reveal deployment details. Issuers and system designers should take care not to expose sensitive naming conventions in externally visible identifiers.

Where possible, identifier paths SHOULD be minimally descriptive and avoid exposing internal implementation details unless necessary for interoperation.

7.6. Wildcard and Prefix Matching

Consumers SHOULD NOT interpret Workload Identifiers using wildcard or prefix matching unless explicitly specified by policy. For example, treating all identifiers under prefix of spiffe://example.org/ns/db/ as equivalent may lead to incorrect authorisation.

8. IANA Considerations

8.1. URI Scheme Registration

IANA is requested to register the "wimse" scheme to the "URI Schemes" registry [IANA-URISCHEMES]:

Scheme name: wimse

Status: permanent

Applications/protocols that use this scheme name: any application and protocol interacting with workload identifiers.

Contact: IETF Chair chair@ietf.org

Change controller: IESG iesg@ietf.org

References:

Section 4.4 of this document.

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>.

[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>.

[URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[WIMSE-CREDENTIALS] Campbell, B., Salowey, J. A., Schwenkschuster, A., Sheffer, Y., and Y. Rosomakho, "WIMSE Workload Credentials", Work in Progress, Internet-Draft, draft-ietf-wimse-workload-creds-00, 3 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-wimse-workload-creds-00>.

9.2. Informative References

[ARCH] Salowey, J. A., Rosomakho, Y., and H. Tschofenig, "Workload Identity in a Multi System Environment (WIMSE) Architecture", Work in Progress, Internet-Draft, draft-ietf-wimse-arch-07, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-wimse-arch-07>.

[IANA-URISCHEMES] IANA, "Uniform Resource Identifier (URI) Schemes", <https://www.iana.org/assignments/uri-schemes>.

[JWT] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/rfc/rfc7519>.

[SPIFFE-ID] "The SPIFFE Identity and Verifiable Identity Document", January 2025, <https://github.com/spiffe/spiffe/blob/main/standards/SPIFFE-ID.md>.

Appendix A. Acknowledgments

Authors would like to thank Evan Gilman for his review of the initial text of this document and his guidance.

Appendix B. Changelog

RFC Editor's Note: Please remove this section prior to publication of a final version of this document.

B.1. since draft-ietf-wimse-identifier-00

* Defined Workload Identifier Scope
* Replaced specifics of usage in credentials and tokens with a reference to s2s-creds draft
* Added URI requirements

B.2. since draft-ietf-wimse-identifier-01

* Changed the term "Scope" to "Origin"
* Added wimse URI scheme definition
* Added more alignement and reduced overlap with draft-ietf-wimse-workload-creds
* Clarified separation of specific workload instances and logical workloads

Authors' Addresses

Yaroslav Rosomakho
Zscaler
Email: yaroslavros@gmail.com

Joe Salowey
CyberArk
Email: joe@salowey.net