Network Working Group
Internet-Draft
Intended status: Informational
Expires: 11 June 2026

L. Johansson
SIROS Foundation

8 December 2025

An AuthZEN profile for trust registries
draft-johansson-authzen-trust-01

Abstract

Trust registries come in many forms; ETSI trust status lists, OpenID Federation, ledgers. This document describes a simple protocol in the form of a profile of AuthZen that provides a local interface to one or more trust registries. The protocol is mant to be used as a local abstraction layer for any application that needs to evaluate trust.

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://leifj.github.io/draft-johansson-authzen-trust. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-johansson-authzen-trust/.

Source for this draft and an issue tracker can be found at https://github.com/leifj/draft-johansson-authzen-trust.

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/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 11 June 2026.

Table of Contents

1. Introduction

Technical trust in systems using asymmetric cryptography often involves binding a name to a public key. One such example is, given an X.509 certificate (as a representative of a public key and name), determining its validity relative to a set of trust anchors by means of PKIX path construction and path validation. In this example the trust registry is the set of trust anchors together with the rules for path validation and construction set down in [RFC5280].

The proliferation of distributed identity systems have led to the development of a multitude of trust registries each with their own APIs for querying the registry and rules for evaluating trust.

Application developers are often faced with the choice of choosing one of these trust registries which leads to interoperability problems. It is often common for an service to register with multiple trust registries in order to reach all intended audiences.

This document describes an API for trust evaluation that is intended to fill a role similar to the stub resolver in the DNS architecture. The API is defined as a profile of [AUTHZEN]. AuthZen is a proposed standard for communicating between an authorization policy enforcement point (PEP) and a policy decision point (PDP).

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.

This specification uses the terms "PDP" and "PEP" defined by [NIST.SP.800-162] and [XACML]. A trust registry refers to any service that provides a binding (or mapping) between public keys and names. This is referred to as "name to key" or name-to-key binding.

3. Endpoints

Implementations of this specification MUST provide the /evaluation endpoint and SHOULD also provide the /evaluations and discovery endpoints. The /search endpoint MAY be provided but providing this endpoint may provide significant challenges for this profile and clients MUST NOT assume that it is present.

4. Authorization Request

This profile implements the following semantic: The client (PEP) requests that the server (PDP) authorizes the binding between the name specified by the Subject with the public key specified by the Resource. Optionally the Action is used to constrain the authorization to a specific role that the entity that the public key is bound to must have for the authorization to be approved. The PDP may also attempt to _resolve_ the name into metadata that provides additional information about the name-to-key binding.

4.1. Subject

Subject is used to represent the name part of the name-to-key binding.

The subject datafield MUST be present in requests and MUST contain the following elements:

* id MUST be the name bound to the public key to be validated or resolved

* type MUST be the constant string "key"

4.2. Resource

The resource datafield MUST be present in requests and MUST contain the following elements:

* id MUST be the name bound to the public key to be validated or resolved. Furthermore, the value of the resource.id element MUST be the same string as in the subject.id element.

* type MAY be present and if present MUST be one of "jwk" or "x5c".

* key If present, MUST be the public key in a format that depends on the type. If type is absent then key MUST NOT be present.

If typeis present then,

* If type is "jwk" then key MUST contain a JWK ([RFC7517]) format key.

* If type is "x5c" then key MUST contain an array of base64 encoded X.509 certificates formatted according to section 4.7 of [RFC7517].

Some trust registries support unambiguous name-to-key discovery. For such trust registries key and type MAY be elided from the Resource as described above.

When type and key is present however, a PDP implementing this specification MUST validate that the key is bound to subject.id even if subject.id is a name bound to a trust registry that supports unambiguous name-to-key discovery.

The PDP MAY include additional _metadata_ associated with subject.id in the result. The method by which this is done is an implementation detail but for instance when the subject.id is a "DID" then the resolution MAY be done by the lookup process of a supported DID method. It is RECOMMENDED that PDPs that support such trust registries return the appropriate metadata in the response.

Other specifications may define additional key formats in the future.

4.3. Action

The action datafield MAY be present in requests and SHOULD if present be used to represent the role associated with the name-to-key binding. This is used to distinguish different uses of the same name-to-key binding. For instance the action can be used to request authorization that a X.509 certificate is allowed to act as a TLS server endpoint or as a digital credential issuer.

If the action is present then it MUST contain at least the name parameter which MUST contain a string that represents the role. The interpretation of the role depends on the deployment.

4.4. Context

The context datafield MAY be present in requests but MUST NOT contain information that is critical for the correct processing of the request.

5. Authorization Response

The Authorization Response MAY return metadata associated with subject.id in the response using the trust_metadata field. When the request type is absent then the trust_metadata field SHOULD be present.

6. Examples (non-normative)

The following example is a query to check if a provided certificate chain is bound to the name "did:foo:bla" and is allowed act as a EUDI wallet provider.

```
{
  "type": "authzen",
  "request": {
      "subject": {
        "type": "key",
        "id": "did:foo:bla"
    },
    "resource": {
      "type": "x5c",
      "id": "did:foo:bla",
      "key": ["... x5c data ..."]
    },
    "action": {
      "name": "http://ec.europa.eu/NS/wallet-provider",
    }
  }
}
```

The following example is a query to check if a provided certificate chain is bound to "www.example.com" and is allowed to act as a TLS server.

```
{
  "type": "authzen",
  "request": {
      "subject": {
        "type": "key",
        "id": "www.example.com"
    },
    "resource": {
      "type": "x5c",
      "id": "www.example.com",
      "key": ["... x5c data ..."],
    },
    "action": {
      "name": "TODO:oid:tls-server",
    }
  }
}
```

The following is an example response with no additional context:

```
{
  "decision": true
}
```

The following is an example response with trust_metadata that contains an (abbreviated) DID document.

```
   {
     "decision": true,
     "context": {
       trust_metadata: {
         {
           "@context": [
             "https://www.w3.org/ns/did/v1",
             "https://w3id.org/security/suites/ed25519-2020/v1"
           ],
           "id": "did:example:123",
           ....
       }
     }
   }
```

The following is an hypothetical response with error messages:

```
{
  "decision": false,
  "context": {
    "reason": {
      "403": "Unknown service - contact helpdesk@registry.example.com for support using t
he following identifier: #ID4711"
    }
  }
}
```

7. AuthZen Trust as a DID resolver

As should be obvious from the specification above, a DID resolver as specified in section 7 of [DID] share many properties with this specification. Notable differences is that error handling is slightly different and content negotiation is handled by the PDP which means that DID resolution options (section 7.1.1 of [DID]) isn’t needed in this case.

8. Security Considerations

The protocol described in this specification is meant to be used by applications that share a common security domain and it may be perfectly reasonable for deployments of this specification to be deployed without authentication on "localhost" or in situations where security requirements for the protocol is provided elsewhere in the stack. In general implementations of this specification MAY implement [RFC6749] authentication for the purpose of authenticating the client (PDP) to the server (PEP) and SHOULD provide a way for the PDP to be authenticated to the client.

In addition to the above the security considerations for authentication for AuthZen applies in equal measure to this profile.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[AUTHZEN] Gazitt, O., Brossard, D., and A. Tulshibagwale, "OpenID AuthZEN Authorization API", July 2024, <https://openid.github.io/authzen/>.

[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>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/rfc/rfc6749>.

[RFC7517] Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015, <https://www.rfc-editor.org/rfc/rfc7517>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

10.2. Informative References

[DID] "Decentralized Identifiers (DIDs) v1.0", 2022, <https://www.w3.org/TR/did-1.0/>.

[NIST.SP.800-162] Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., and NIST, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations", NIST Special Publications (General) 800-162, DOI 10.6028/NIST.SP.800-162, January

2014,
<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-162.pdf>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC9525] Saint-Andre, P. and R. Salz, "Service Identity in TLS", RFC 9525, DOI 10.17487/RFC9525, November 2023, <https://www.rfc-editor.org/rfc/rfc9525>.

[XACML] Godik, S., Ed. and T. M. (Ed.), Ed., "eXtensible Access Control Markup Language (XACML) Version 1.1", n.d., <https://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf>.

Acknowledgments

TODO acknowledge.

Author’s Address

Leif Johansson
SIROS Foundation
Email: leifj@siros.org