Secure Asset Transfer Protocol
Internet-Draft
Intended status: Informational
Expires: 19 September 2026

D. Avrilionis
Compellio S.A.
T. Hardjono
MIT

18 March 2026

Artefacts Registry
draft-avrilionis-satp-artefacts-registry-00

Abstract

This memo describes the Artefacts Registry for Asset Exchange API. The Registry is a component that exposes an API allowing gateways to fetch information related to the SAT protocol. Examples information stored in the Artefacts Registry are network identifiers, entities identifiers, asset profiles, or asset instances. Registries are are acting as persistent storage locations for records. Once registered, records can be updated in an append-only manner.

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://compellio.github.io/draft-avrilionis-satp-artefacts-registry/draft-avrilionis-satp-artefacts-registry.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-avrilionis-satp-artefacts-registry/.

Discussion of this document takes place on the Secure Asset Transfer Protocol Working Group mailing list (mailto:sat@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/sat/. Subscribe at https://www.ietf.org/mailman/listinfo/sat/.

Source for this draft and an issue tracker can be found at https://github.com/compellio/draft-avrilionis-satp-artefacts-registry.

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

This memo proposes an API intended to be implemented by Artefact Registries in the context of SATP. A Registry is a component that exposes an API allowing gateways to fetch artefacts related to the SAT protocol ("API3"). Examples of SATP artefacts are network identifiers, entities identifiers, asset profiles, or asset instances.

Registries play an important role in maintaining record of artefacts that are important during the Setup Stage of SATP (a.k.a "Stage 0"). Registries are are acting as persistent storage locations for such artefacts. Once registered, artefacts cannot be removed. New versions are appended in the Registry without removing previous versions (append-only principle). Readers are directed first to [REGARCH] for a description of the architecture underlying the Registries.

All API calls are assumed to run over TLS1.2 or higher, and the endpoints of the registry are associated with a certificate indicating the legal owner (or operator) of the Registry. HTTPS must be used instead of plain HTTP.

2. Conventions used in this document

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying significance described in RFC 2119.

3. Terminology

Please refer to [TERM] for the terminology used in this document.

The artefacts recorded via registries are called Tokenized Asset Records or TARs. Please refer to [REGARCH] for more information about TARs.

4. The Registry API

4.1. Overview

The Registry API pertains to the interaction between gateways. In [ARCH] such interface was identied as "API3" - see [ARCH] for more details.

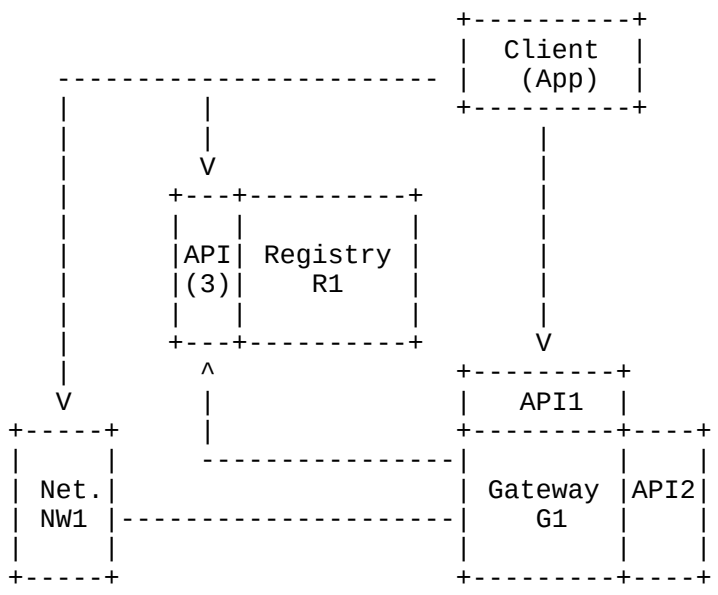

Figure 1

4.2. The role of the Registry in SATP

The three stages of the SATP protocol are described in [CORE]. Prior to the initiation of SATP the peer gateways may access artefacts related to networks, assets or the gateway themselves. Registries are used to maintain record of such artefacts. Registries are of particular importance in the interactions between the peer gateways during the setup stage (Stage-0) [STAGE0]

Records stored in a registry are persisted in the form of Tokenized Artefacts Record (TAR). In summary the main concepts of a TAR are as follows:

* The Artefact: This is a piece of information containing digitized data or pointing to assets. It can range from configuration data, execution log, network identifier, as well as any form of tangible or intangible asset, such as real estate, art, company shares, or even intellectual property.

* The Token: A digital token is created on a network to represent a specific piece of information or an asset.

* The Record: The token acts as the immutable record of ownership. It contains vital data about the artefact (metadata), ownership history, and rules for transfer, all secured by a network.

5. Registry API Message Format, identifiers and Descriptors

5.1. Overview

This section describes the Registry API message-types, the format of the messages, the format for resource descriptors and other related parameters.

5.2. API Digital Signatures and Key Types

All API calls must be signed, using JSON Web Signatures mechanism (RFC7515).

Registries SHOULD support the algorithms defined in the JSON Web Algorithms (JWA) specification [RFC7518] and key types defined in the JSON Web Key (JWK) specification [RFC7517].

All registries implementing the API must implement at minimal the ECDSA signature algorithm with the P-256 curve and the SHA-256 hash function.

Additional signature algorithms and keying parameters may be implemented by the registries. However, these are outside the scope of this specification.

5.3. Registry Message Format and Payloads

All registry API messages are in JSON format [JSON].

5.3.1. Protocol version

This refers to the registry API Version, encoded as "major.minor" (separated by a period symbol).

The endpoints of the registry should clearly indicate the version of the API. The current version is "1.0" defined in this specification. Implementations not understanding a future option value should return an appropriate error response and cease the negotiation.

5.3.2. Client Credential Types Supported by Registries

Registries must support JSON Web Tokens (JWT) [RFC 7519] with OAUth2.0 [RFC6749] as the minimal credential type for authenticating incoming API calls.

A registry may support additional credential mechanisms, which may be advertised by the registry through different mechanisms (e.g. config file at a well-known endpoint). However, these mechanisms are out of scope for the current specification.

5.3.3. Registry Supported TLS Schemes

Registries must suport TLS1.2 or higher [RFC8448].

The TLS scheme is used by client applications to call the Registry endpoints. Registries must a minimal support the AES-128 in GCM mode with SHA-256 (TLS_AES_128_GCM_SHA256).

5.3.4. Client Offers Other Supported TLS Schemes

If a client application wishes to use TLS schemes other then the basic scheme (AES-128 in GCM mode with SHA-256), then the client may choose to send a JSON block containing information regarding the client’s supported TLS schemes.

5.3.5. Registry Identifier

This is the unique identifier of the registry service. The registry identifier MUST be uniquely bound to its endpoint (e.g. via X.509 certificates).

This registry identifier is distinct from the registry operator business identifier (e.g., legal entity identifier (LEI) number).

A registry operator may operate multiple registries. Each of the registries MUST be identified by a unique registry identifier.

The mechanisms to establish the registry identifier or the operator identifier is outside the scope of this specification.

5.3.6. Signature Algorithms Supported

This is a JSON list of digital signature algorithms supported by a registry. Each entry in the list should be either an Algorithm Name value registered in the IANA "JSON Web Signature and Encryption Algorithms" registry established by [JWA] or be a value that contains a Collision-Resistant Name.

All implementations must support a common default of "ES256", which is the ECDSA signature algorithm with the P-256 curve and the SHA-256 hash function.

5.3.7. JSON Canonicalisation

Registries must suport JSON Canonicalization [RFC8785].

6. Overview of API endpoints

Registries MUST support the use of the HTTP GET and POST methods defined in RFC 2616 [RFC2616] for the endpoint.

Clients MAY use the HTTP GET or POST methods to send messages to the registry.

If using the HTTP GET method, the request parameters may be serialized using URI Query String Serialization.

(NOTE: Flows occur over TLS. Nonces are not shown).

6.1. TAR Creation

6.1.1. Call

This endpoint stores the input data in the regitry permanent storage and returns a token identifier related to the TAR. The call should return as soon as the input data are persisted in the registry. In case of asynchronous creation of the token identifier associated to the TAR a temporary receipt should be returned. Such receipt should be substituted with the permanent token identifier.

This endpoint uses an HTTP POST method

Here is an example representation in JSON format:


```
{
  "@context": "urn:tar:eip155.444444444500:81d0782847297956410ec1a674e60a78fff14b69",
  "performance": {
    "venueID": "urn:tar:eip155.137:2C3a4C8a34404Ee0145f588536E94D47421dC891",
    "location": "Poetry Bar",
    "url": "https://poetry.bar/20241116/triplicity",
    "description": {
      "en": "Triplicity band live performance"
    },
    "startTime": "2024-11-16T21:30:00Z+02:00",
    "doorTime": "2024-11-16T21:00:00Z+02:00"
  },
  "owner": {
    "ownerID": "urn:tar:eip155.137:d9D6916A0A65Fe2aC212243E8f2252143D0c7dE4"
  },
  "price": {
    "currency": "EUR",
    "value": "10"
  },
  "seat": {
    "seatNumber": "A1",
    "seatLocation": "VIP"
  },
  "validity": {
    "used": false
  }
}
```


6.1.2. Return result

```
{
  "id": "urn:tar:eip155.444444444500:6850be85c8c264ef1562ebae547fd7086c281774",
  "receipt": "23f2769a-2cce-48f7-9612-4c3dd7a918b8",
  "data": {
    "@context": "urn:tar:eip155.444444444500:81d0782847297956410ec1a674e60a78fff14b69"
,
    "owner": {
      "ownerID": "urn:tar:eip155.137:d9D6916A0A65Fe2aC212243E8f2252143D0c7dE4"
    },
    "performance": {
      "description": {
        "en": "Triplicity band live performance"
      },
      "doorTime": "2024-11-16T21:00:00Z+02:00",
      "location": "Poetry Bar",
      "startTime": "2024-11-16T21:30:00Z+02:00",
      "url": "https://poetry.bar/20241116/triplicity",
      "venueID": "urn:tar:eip155.137:2C3a4C8a34404Ee0145f588536E94D47421dC891"
    },
    "price": {
      "currency": "EUR",
      "value": "10"
    },
    "seat": {
      "seatLocation": "VIP",
      "seatNumber": "A1"
    },
    "validity": {
      "used": false
    }
  },
  "checksum": "0xBA66E005328F45E1AE3CCE97F3404E4D4365D13443214CB968A49BB5948F98F3",
  "version": 1
}
```


6.1.3. Error Message

TBD

6.2. TAR Read

This endpoint retrieves a TAR previously created or updated in the Registry.

6.2.1. Call

The parameters of this message consist of the following: * tarid REQUIRED: urn:tar:eip155.444444444500:81d0782847297956410ec1a674e60a78fff14b69

6.2.2. Return result


```
   {
     "id": "urn:tar:eip155.444444444500:6850be85c8c264ef1562ebae547fd7086c281774",
     "receipt": "23f2769a-2cce-48f7-9612-4c3dd7a918b8",
     "data": {
       "@context": "urn:tar:eip155.444444444500:81d0782847297956410ec1a674e60a78fff14b69"
,
       "owner": {
         "ownerID": "urn:tar:eip155.137:d9D6916A0A65Fe2aC212243E8f2252143D0c7dE4"
       },
       "performance": {
         "description": {
           "en": "Triplicity band live performance"
         },
         "doorTime": "2024-11-16T21:00:00Z+02:00",
         "location": "Poetry Bar",
         "startTime": "2024-11-16T21:30:00Z+02:00",
         "url": "https://poetry.bar/20241116/triplicity",
         "venueID": "urn:tar:eip155.137:2C3a4C8a34404Ee0145f588536E94D47421dC891"
       },
       "price": {
         "currency": "EUR",
         "value": "10"
       },
       "seat": {
         "seatLocation": "VIP",
         "seatNumber": "A1"
       },
       "validity": {
         "used": false
       }
     },
     "checksum": "0xBA66E005328F45E1AE3CCE97F3404E4D4365D13443214CB968A49BB5948F98F3",
     "version": 1,
     "_sdHashes": []
   }
```


6.2.3. Error Message

TBD

6.3. TAR Update

Updates the TAR by registering a new version for the specific TARID

6.3.1. Call

The parameters of this message consist of the following: * tarid REQUIRED: urn:tar:eip155.444444444500:81d0782847297956410ec1a674e60a78fff14b69

* tarBody REQUIRED:


```
{
  "@context": "urn:tar:eip155.444444444500:81d0782847297956410ec1a674e60a78fff14b69",
  "owner": {
    "ownerID": "urn:tar:eip155.137:d9D6916A0A65Fe2aC212243E8f2252143D0c7dE3"
  },
  "performance": {
    "description": {
      "en": "Triplicity band live performance"
    },
    "doorTime": "2024-11-16T21:00:00Z+02:00",
    "location": "Poetry Bar",
    "startTime": "2024-11-16T21:30:00Z+02:00",
    "url": "https://poetry.bar/20241116/triplicity",
    "venueID": "urn:tar:eip155.137:2C3a4C8a34404Ee0145f588536E94D47421dC891"
  },
  "price": {
    "currency": "EUR",
    "value": "10"
  },
  "seat": {
    "seatLocation": "VIP",
    "seatNumber": "A2"
  },
  "validity": {
    "used": false
  }
}
```


6.3.2. Return result

```
   {
     "id": "urn:tar:eip155.444444444500:6850be85c8c264ef1562ebae547fd7086c281774",
     "receipt": "23f2769a-2cce-48f7-9612-4c3dd7a918b8",
     "data": {
       "@context": "urn:tar:eip155.444444444500:81d0782847297956410ec1a674e60a78fff14b69"
,
       "owner": {
         "ownerID": "urn:tar:eip155.137:d9D6916A0A65Fe2aC212243E8f2252143D0c7dE4"
       },
       "performance": {
         "description": {
           "en": "Triplicity band live performance"
         },
         "doorTime": "2024-11-16T21:00:00Z+02:00",
         "location": "Poetry Bar",
         "startTime": "2024-11-16T21:30:00Z+02:00",
         "url": "https://poetry.bar/20241116/triplicity",
         "venueID": "urn:tar:eip155.137:2C3a4C8a34404Ee0145f588536E94D47421dC891"
       },
       "price": {
         "currency": "EUR",
         "value": "10"
       },
       "seat": {
         "seatLocation": "VIP",
         "seatNumber": "A1"
       },
       "validity": {
         "used": false
       }
     },
     "checksum": "0xBA66E005328F45E1AE3CCE97F3404E4D4365D13443214CB968A49BB5948F98F3",
     "version": 1,
     "_sdHashes": []
   }
```


6.3.3. Error Message

TBD

6.4. TAR Deletion

Archives the TAR

6.4.1. Call

The parameters of this message consist of the following: * tarid REQUIRED: urn:tar:eip155.444444444500:81d0782847297956410ec1a674e60a78fff14b69

6.4.2. Return result

A success code

6.4.3. Error Message

TBD

7. IANA Consideration

The tar namespace might follow a hierachichal structure under the general URN satp namespace, i.e. urn:satp:tar.

7.1. Registry API Error Codes

TBD

7.2. URN Registration

URN: Request to be assigned by IANA.

Common Name: urn:ietf:satp

Registrant Contact: IESG

Description: The secure asset transfer protocol (SATP) requires message types, endpoints and parameters to be defined within a unique namespace to prevent collision.

8. Error Types and Codes

This appendix defines the error codes that may be returned in SATP protocol messages.

8.1. Protocol Error Codes

TBD

9. Acknowledgements

The authors would like to thank the following people for their input and support:

Andre, Rafael, Rama.

10. References

10.1. Normative References

[BASE64] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/rfc/rfc4648>.

[DATETIME] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/rfc/rfc3339>.

[JSON] 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>.

[JWA] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/rfc/rfc7518>.

[JWS] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/rfc/rfc7515>.

[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>.

[REQ-LEVEL] 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>.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, DOI 10.17487/RFC2616, June 1999, <https://www.rfc-editor.org/rfc/rfc2616>.

[X.500] ITU-T, "The Directory: Overview of concepts, models and services", 2005.

10.2. Informative References

[ARCH] Hardjono, T., Hargreaves, M., Smith, N., and V. Ramakrishna, "Secure Asset Transfer (SAT) Interoperability Architecture", June 2024, <https://datatracker.ietf.org/doc/draft-ietf-satp-architecture/>.

[CORE] Hargreaves, M., Hardjono, T., Belchior, R., and V. Ramakrishna, "Secure Asset Transfer Protocol (SATP) Core", November 2025, <https://datatracker.ietf.org/doc/draft-ietf-satp-architecture/>.

[ECDSA] "Digital Signature Standard (FIPS 186-5)", February 2023, <https://doi.org/10.6028/NIST.FIPS.186-5>.

[MICA] European Commission, "EU Directive on Markets in Crypto-Assets Regulation (MiCA)", June 2023, <https://www.esma.europa.eu/esmas-activities/digital-finance-and-innovation/markets-crypto-assets-regulation-mica>.

[NIST] Yaga, D., Mell, P., Roby, N., and K. Scarfone, "NIST Blockchain Technology Overview (NISTR-8202)", October 2018, <https://doi.org/10.6028/NIST.IR.8202>.

[REGARCH] Avrilionis, D. and T. Hardjono, "Asset Schema Architecture for Asset Exchange", November 2025, <https://datatracker.ietf.org/doc/draft-ietf-satp-architecture/>.

[RFC5939] Andreasen, F., "Session Description Protocol (SDP) Capability Negotiation", RFC 5939, DOI 10.17487/RFC5939, September 2010, <https://www.rfc-editor.org/rfc/rfc5939>.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, DOI 10.17487/RFC8785, June 2020, <https://www.rfc-editor.org/rfc/rfc8785>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/rfc/rfc9334>.

[STAGE0] Avrilionis, D. and T. Hardjono, "SATP Setup Stage", November 2025, <https://datatracker.ietf.org/doc/draft-avrilionis-satp-setup-stage/>.

[TERM] Hardjono, T., "Secure Asset Transfer Protocol (SATP) Terminology", November 2025, <https://datatracker.ietf.org/doc/draft-hardjono-satp-terminology/>.

Authors’ Addresses

Denis Avrilionis
Compellio S.A.
Email: denis@compellio.com

Thomas Hardjono
MIT
Email: hardjono@mit.edu