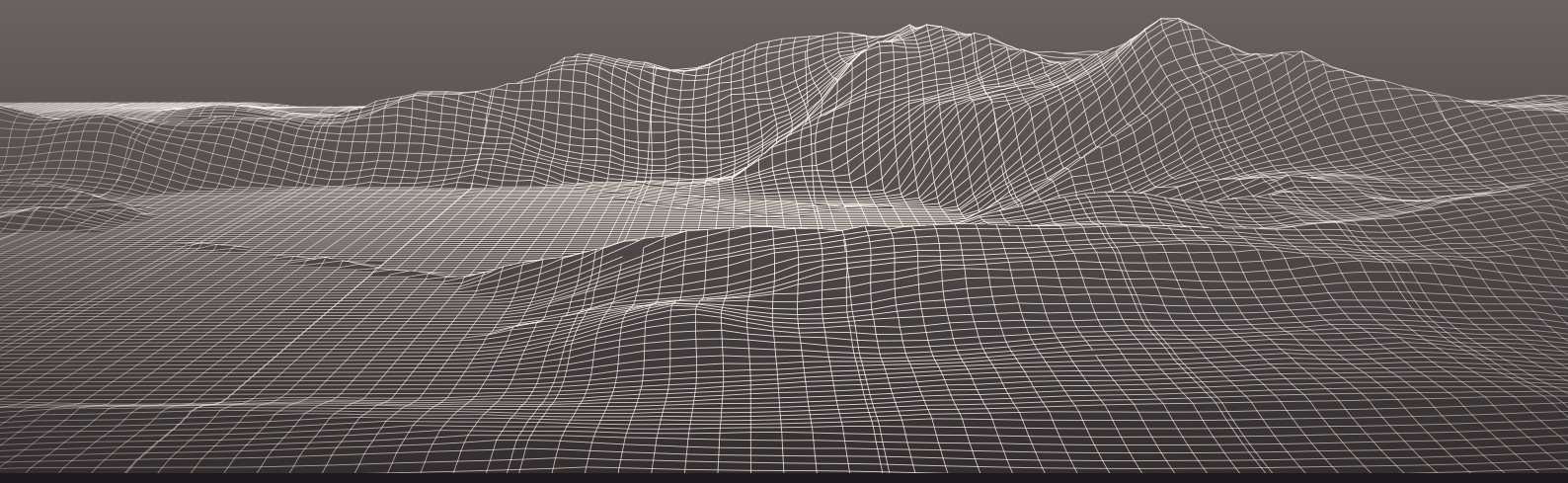
SPATIAL WEB
FOUNDATION

SWF STD-5:2025 (0.1.0)

The Spatial Web

DID Method for the Spatial Web (did:swid) Implementation Specification

September 26, 2025



Contents

List of figures

Spatial Web Foundation
5877 Obama Blvd
Los Angeles, CA 90016
USA

E-mail: info@spatialwebfoundation.org

Abstract

`did:swid` is a DID method (W3C WD-did-1.1-20250918) for the Spatial Web (Spatial Web Foundation), with the following design goals and properties:

- All Spatial Web Entities (including Credentials) have SWIDs.
- The Spatial Web SWID Registry ensures uniqueness of the SWIDs.
- Most SWIDs will be created by a DID Issuer at behest of the Entity and not recorded in a centralized registry.
- The Spatial Web will include Entities with DIDs created without interaction with any particular authority.

1. Scope

This specification defines the `did:swid` method for decentralized identifier (DID) (W3C did-core). DIDs using this method can be used as Spatial Web Identifiers (SWIDs). See SWF 3:2025 (0.1.0) for more details about SWIDs.

The Spatial Web will use SWIDs as identifiers.

The `did:swid` method leverages a registry known as Spatial Web SWID Registry, which is managed by a Designated Authority.

2. Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

W3C did-core, World Wide Web Consortium. *Decentralized Identifiers (DIDs) v1.0*. https://www.w3.org/TR/did-core/.

3. Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1. self-certifying identifier PREFERRED
SCID ADMITTED

object identifier derived from initial data such that an attacker could not create a new object with the same identifier

3.2. Data Integrity PREFERRED

specification of mechanisms for ensuring the authenticity and integrity of structured digital documents using cryptography, such as digital signatures and other digital mathematical proofs

SOURCE: W3C CG Data Integrity

3.3. decentralized identifier PREFERRED
DID ADMITTED

type of identifier that enable verifiable, decentralized digital identities

Note 1 to entry: A DID refers to any subject (e.g., a person, organization, thing, data model, abstract entity, etc.) as determined by the controller of the DID.

SOURCE: W3C did-core

3.4. DID controller PREFERRED

entity that controls (create, updates, deletes) a given *DID* (3.3)

SOURCE: W3C did-core

3.5. DID document PREFERRED

document returned when a *DID* (3.3) is resolved

SOURCE: W3C did-core

3.6. DID method PREFERRED

mechanism by which a particular type of *DID* (3.3) and its associated *DID document* (3.5) are created, resolved, updated, and deactivated

Note 1 to entry: DID methods are defined using separate DID method specifications.

3.7. JSON Canonicalization Scheme PREFERRED

method for canonicalizing a JSON structure such that is suitable for verifiable hashing or signing according to IETF RFC 8785

SOURCE: IETF RFC 8785

3.8. multibase PREFERRED

specification for encoding binary data as a string using a prefix that indicates the encoding

3.9. multikey PREFERRED

verification method that encodes key types into a single binary stream that is then encoded as a *multibase* (3.8) value

3.10. multihash PREFERRED

mechanism for differentiating instances of hashes

Note 1 to entry: Software creating a hash prefixes (according to the specification) data to the hash indicating the algorithm used and the length of the hash, so that software receiving the hash knows how to verify it. Although multihash supports many hash algorithms, for interoperability, term `DID controllers` not resolved via ID `DID-controllers` MUST only use the hash algorithms defined in the specification as permitted.

SOURCE: Internet-Draft draft-multiformats-multibase-08

3.11. W3C VCDM PREFERRED

verifiable credential according to W3C REC-vc-data-model-20220303

SOURCE: W3C REC-vc-data-model-20220303

4. Identifier

4.1. Target system

The target system of the `did:swid` DID method is a registry known as Spatial Web SWID Registry, which is managed by a Designated Authority.

4.2. Method name

The namestring that identifies this DID method is: `swid`. A DID that uses this method MUST begin with the following prefix: `did:swid`. Per the DID specification, this string MUST be in lowercase. The remainder of the DID, after the prefix, is the 4.3, specified below.

4.3. Method-specific identifier

The `did:swid` method-specific identifier contains a self-certifying identifier (SCID) for the DID.

As specified in the following Augmented Backus-Naur Form (ABNF) notation IETF RFC 2234 the self-certifying identifier MUST be present in the DID string. See examples below.

EXAMPLE 1 — Example method-specific identifier

```
swid-did = "did:swid:" "z" scid
scid = 46(base58-btc-alphabet)
```

The `scid` part of the `did:swid` identifier is generated by creating a cryptographic digest (hash) of the first entry of a data structure known as Cryptographic Event Log Cryptographic Event Log (also see Clause 6). The hash is encoded using multihash (Internet-Draft draft-multiformats-multibase-08).

The characters in the `base58-btc-alphabet` are as defined in the W3C Controller Documents specification (W3C REC-cid-1.0-20250515).

EXAMPLE 2 — Example `did:swid`

```
did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv
```

5. DID Method Operations

5.1. Create (Register)

5.1.1. Creating a did:swid

To create a did:swid, the Entity first generates a private/public key pair that controls the DID. The public key is included as a verification method in the initial version of the DID document.

This initial version of the DID document MUST be a conformant SWID Document, i.e. it MUST have an HSTP Service Endpoint.

See SWF 3:2025 (0.1.0) for more details about SWIDs and SWID Documents.

EXAMPLE 1 — Example initial DID document

```
{
   "@context": [
      "https://www.w3.org/ns/did/v1.1",
      "https://spatialwebfoundation.org/contexts/did/1.0"
   ],
   "verificationMethod": [{
      "id": "#keys-1",
      "type": "Multikey",
      "publicKeyMultibase": "z6MkmM42vxfqZQsv4ehtTjFFxQ4sQKS2w6WR7em
ozFAn5cxu"
   }],
   "authentication": [
     "#keys-1"
   ],
   "service": [{
      "id": "#hstp",
      "type": "HSTPEndpoint",
      "serviceEndpoint": "https://hstp.example.com/hstpendpoint"
   }]
}
```

Using the initial version of the DID document, the first entry of a Cryptographic Event Log (Cryptographic Event Log) is constructed (also see Clause 6). A cryptographic digest (hash) of this entry is created and encoded using multihash (Internet-Draft draft-multiformats-multibase-08), and is used as the scid part of the did:swid identifier (see Clause 4).

EXAMPLE 2 — Example SWID

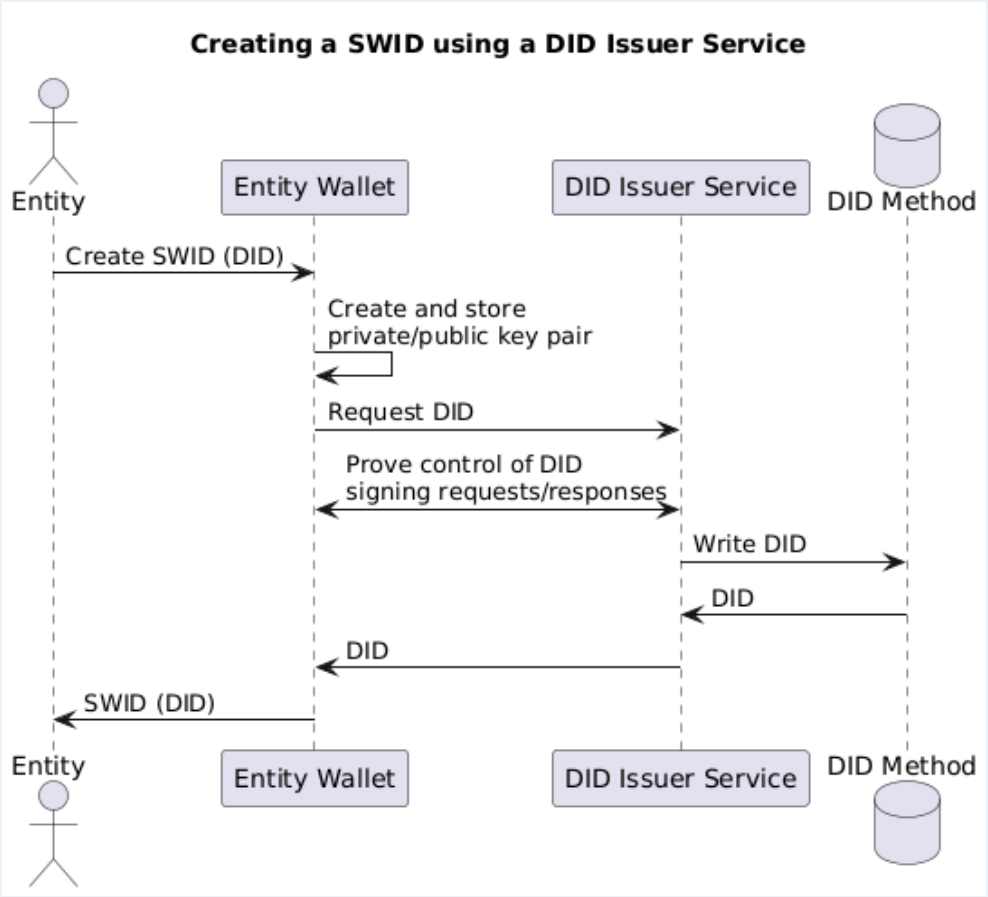
did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv

5.1.2. Using a DID Issuer Service

A did:swid or other type of SWID may be created by a "DID Issuer Service" at behest of the Entity.

The following diagram describes the process of creating a SWID using a "DID Issuer Service":

FIGURE 1: Creating a SWID using a DID Issuer Service



When creating a `did:swid` or other type of SWID, the "DID Issuer Service" MUST enforce that the DID document associated with the SWID is a conformant SWID Document, i.e. it MUST have an HSTP Service Endpoint. Otherwise, the "DID Issuer Service" MUST return an error response.

See SWF 3:2025 (0.1.0) for more details about SWIDs and SWID Documents.

To create a SWID using a "DID Issuer Service", the `create` function of the DIF specification (DIF did-registration) is used.

EXAMPLE 1 — Example Request to create a SWID: HTTP POST to `https://<did-issuer>/create?method=swid`

```
{
  "options": {
    "clientSecretMode": true
  },
  "secret": { },
  "didDocument": {
     "@context": [
        "https://www.w3.org/ns/did/v1.1",
        "https://spatialwebfoundation.org/contexts/did/1.0"
```

```
      ],
      "verificationMethod": [{
         "id": "#keys-1",
         "type": "Multikey",
         "publicKeyMultibase": "z6MkmM42vxfqZQsv4ehtTjFFxQ4sQKS2w6WR7
emozFAn5cxu"
      }],
      "authentication": [
        "#keys-1"
      ],
      "service": [{
         "id": "#hstp",
         "type": "HSTPEndpoint",
         "serviceEndpoint": "https://hstp.example.com/hstpendpoint"
      }]
   }
}
```

EXAMPLE 2 — Example Response

```
{
  "didState": {
    "state": "finished",
    "did": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv
",
    "secret": { }
  },
  "didRegistrationMetadata": { },
  "didDocumentMetadata": { }
}
```

During the process of creating a SWID, the Entity MAY be required to prove control of its SWID's associated public key, using Signing Requests (DIF did-registration) and Signing Responses (DIF did-registration).

EXAMPLE 3 — Example Signing Request

```
{
 "jobId": "96202012-41d8-424a-85a9-3673bda6abc7",
 "didState": {
  "state": "action",
  "action": "signPayload",
  "signingRequest": {
   "signingRequestCreate": {
    "kid": "#keys-1",
    "alg": "EdDSA",
    "purpose": "authentication",
    "serializedPayload": "<-- base 64 encoded -->"
   }
  }
 },
 "didRegistrationMetadata": { ... },
 "didDocumentMetadata": { ... }
}
```

EXAMPLE 4 — Example Signing Response

```
{
    "jobId": "96202012-41d8-424a-85a9-3673bda6abc7",
    "options": {
```

```
            "clientSecretMode": true
        },
        "secret": {
            "signingResponse": {
                "signingRequestCreate": {
                    "signature": "<-- base64 encoded -->"
                }
            }
        },
        "didDocument": {}
}
```

5.2. Read (Resolve)

To resolve a `did:swid` or other type of SWID, the `resolve` function W3C did-resolution is used.

EXAMPLE 1 — Example Request to resolve a SWID: HTTP
GET to `https://<swid-resolver>/identifiers/did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv`

EXAMPLE 2 — Example Response

```
{
  "didDocument": {
    "@context": [
      "https://www.w3.org/ns/did/v1.1",
      "https://spatialwebfoundation.org/contexts/did/1.0"
    ],
    "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv
",
    "verificationMethod": [{
      "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3j
v#keys-1",
      "type": "Multikey",
      "controller": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY
2Hcjpe3jv",
      "publicKeyMultibase": "z6MkmM42vxfqZQsv4ehtTjFFxQ4sQKS2w6WR7em
ozFAn5cxu"
    }],
    "authentication": [
      "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv#keys
-1"
    ],
    "service": [{
      "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3j
v#hstp",
      "type": "HSTPEndpoint",
      "serviceEndpoint": "https://hstp.example.com/hstpendpoint"
    }]
  },
  "didResolutionMetadata": { },
  "didDocumentMetadata": { }
}
```

5.3. Update (Rotate)

To update a `did:swid` or other type of SWID, the `update` function DIF did-registration is used.

Updating a SWID can include the following:

- Rotating the public key that controls the SWID.
- Changing the HSTP service endpoint of the SWID.

EXAMPLE 1 — Example Request to update a SWID: HTTP POST to `https://<swid-registry>/update`

```
{
  "did": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv",
  "options": {
    "clientSecretMode": true
  },
  "secret": { },
   "didDocument": {
      "@context": [
         "https://www.w3.org/ns/did/v1.1",
         "https://spatialwebfoundation.org/contexts/did/1.0"
      ],
      "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3j
v",
      "verificationMethod": [{
         "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjp
e3jv#keys-1",
         "type": "Multikey",
         "controller": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKr
McY2Hcjpe3jv",
         "publicKeyMultibase": "z6MkmM42vxfqZQsv4ehtTjFFxQ4sQKS2w6WR
7emozFAn5cxu"
      }],
      "authentication": [
         "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv#k
eys-1"
      ],
      "service": [{
         "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjp
e3jv#hstp",
         "type": "HSTPEndpoint",
         "serviceEndpoint": "https://hstp.example.com/new_
hstpendpoint"
      }]
   }
}
```

EXAMPLE 2 — Example Response

```
{
  "didState": {
    "state": "finished",
    "did": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv
",
    "secret": { }
  },
```

```
  "didRegistrationMetadata": { },
  "didDocumentMetadata": { }
}
```

During the process of updating a SWID, the Entity MAY be required to prove control of its SWID's associated public key, using Signing Requests and Signing Responses.

5.4. Deactivate (Revoke)

To deactivate a `did:swid` or other type of SWID, the `deactivate` function of the DIF DIF did-registration specification is used.

EXAMPLE 1 — Example Request to deactivate a SWID: HTTP POST to `https://<swid-registry>/deactivate`

```
{
  "did": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv",
  "options": {
    "clientSecretMode": true
  },
  "secret": { }
}
```

EXAMPLE 2 — Example Response

```
{
  "didState": {
    "state": "finished",
    "did": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3jv
",
    "secret": { }
  },
  "didRegistrationMetadata": { },
  "didDocumentMetadata": { }
}
```

During the process of deactivating a SWID, the Entity MAY be required to prove control of its SWID's associated public key, using Signing Requests (DIF did-registration) and Signing Responses (DIF did-registration).

6. The SWID Registry

The SWID Registry contains a list of SWIDs. For each SWID, the SWID Registry contains a cryptographic event log following the Cryptographic Event Log specification (Cryptographic Event Log). Every entry in the event log is cryptographically signed by the entity that controls a SWID.

EXAMPLE — Example SWID Registry entry: The SWID Registry could contain the following cryptographic event log for a SWID that has been created, and then updated once:


```
{
  "log": [{
    "event": {
      "operation": {
        "type": "create",
        "data": {
          "@context": [
            "https://www.w3.org/ns/did/v1.1",
            "https://spatialwebfoundation.org/contexts/did/1.0"
          ],
          "id": "did:swid:",
          "verificationMethod": [{
            "id": "#keys-1",
            "type": "Multikey",
            "controller": "did:swid:",
            "publicKeyMultibase": "z6MkmM42vxfqZQsv4ehtTjFFxQ4sQKS2w
6WR7emozFAn5cxu"
          }],
          "authentication": [
            "#keys-1"
          ],
          "assertionMethod": [
            "#keys-1"
          ],
          "service": [{
            "id": "#hstp",
            "type": "HSTPEndpoint",
            "serviceEndpoint": "https://hstp.example.com/
hstpendpoint"
          }],
          "proof": {
            "type": "DataIntegrityProof",
            "cryptosuite": "ecdsa-jcs-2019",
            "created": "2024-11-29T13:56:28Z",
            "verificationMethod": "#keys-1",
            "proofPurpose": "assertionMethod",
            "proofValue": "z5obCSsrQxuFJdq6PrUMCtqY93gBHqGDBtQLPFxpZ
xzwVWgHYrXxoV"
          }
        }
      }
    }
  }, {
    "event": {
      "previousEvent": "uEkoYyQ6YVtUmER8pN24wLZcLK9EBguM5WZlbAgfXBDu
QiA",
      "operation": {
        "type": "update",
        "data": {
          "@context": [
            "https://www.w3.org/ns/did/v1.1",
            "https://spatialwebfoundation.org/contexts/did/1.0"
          ],
```

```
          "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcj
pe3jv",
          "verificationMethod": [{
            "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2H
cjpe3jv#keys-1",
            "type": "Multikey",
            "controller": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemH
AKrMcY2Hcjpe3jv",
            "publicKeyMultibase": "z6MkmM42vxfqZQsv4ehtTjFFxQ4sQKS2w
6WR7emozFAn5cxu"
          }],
          "authentication": [
            "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3j
v#keys-1"
          ],
          "assertionMethod": [
            "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2Hcjpe3j
v#keys-1"
          ],
          "service": [{
            "id": "did:swid:zQmQoeG7u6XBtdXoek5p3aPoTjaSRemHAKrMcY2H
cjpe3jv#hstp",
            "type": "HSTPEndpoint",
            "serviceEndpoint": "https://hstp.example.com/new_
hstpendpoint"
          }],
          "proof": {
            "type": "DataIntegrityProof",
            "cryptosuite": "ecdsa-jcs-2019",
            "created": "2024-11-29T13:56:28Z",
            "verificationMethod": "did:swid:zQmQoeG7u6XBtdXoek5p3aPo
TjaSRemHAKrMcY2Hcjpe3jv#keys-1",
            "proofPurpose": "assertionMethod",
            "proofValue": "z5obCSsrQxuFJdq6PrUMCtqY93gBHqGDBtQLPFxpZ
xzwVWgHYrXxoV"
          }
        }
      }
    }
  }]
}
```


7. Security and Privacy Considerations

EDITORIAL NOTE

TODO

Annex A
(informative)

Version log

The following lists the substantive changes in each version of the specification.

- Version 0.1
  - Initial version

Bibliography

[1] Spatial Web Foundation, Spatial Web Foundation. *Spatial Web Foundation*. https://spatialwebfoundation.org/.

[2] DIF did-registration, Decentralized Identity Foundation, Markus Cihan SABADELLO, Ahamed SAGLAM and AZEEM. *DID Registration*. 2025. https://identity.foundation/did-registration.

[3] W3C did-resolution, World Wide Web Consortium. *Decentralized Identifier Resolution (DID Resolution) v0.3*. https://www.w3.org/TR/did-resolution/.

[4] W3C CG Data Integrity, *Data Integrity 1.0, W3C Final Community Group Report*. 2022. https://www.w3.org/community/reports/credentials/CG-FINAL-data-integrity-20220722/.

[5] W3C WD-did-1.1-20250918, SPORNY, Manu and Dmitri ZAGIDULIN (eds.). *Decentralized Identifiers (DIDs) v1.1*. 2025. World Wide Web Consortium. https://www.w3.org/TR/2025/WD-did-1.1-20250918/.

[6] Internet-Draft draft-multiformats-multibase-08, JUAN BENET and MANU SPORNY. *The Multibase Data Format*. 2023. https://datatracker.ietf.org/doc/html/draft-multiformats-multibase-08.

[7] IETF RFC 2234, P. OVERELL. *Augmented BNF for Syntax Specifications: ABNF*. 1997. RFC Publisher. https://www.rfc-editor.org/info/rfc2234.

[8] IETF RFC 8785, A. RUNDGREN, B. JORDAN and S. ERDTMAN. *JSON Canonicalization Scheme (JCS)*. 2020. RFC Publisher. https://www.rfc-editor.org/info/rfc8785.

[9] SWF 3:2025 (0.1.0), Spatial Web Foundation and Markus SABADELLO. *SWIDs and SWID Documents*. 2025. https://spatial-web-foundation.github.io/swid-document-spec/.

[10] ISO 8601 (all parts), International Organization for Standardization. *Date and time – Representations for information interchange*. First edition. 2019. Geneva. https://www.iso.org/standard/70907.html.

[11] W3C REC-vc-data-model-20220303, SPORNY, Manu, Grant NOBLE, Dave LONGLEY, Daniel BURNETT, Brent ZUNDEL and Kyle DEN HARTOG (eds.). *Verifiable Credentials Data Model v1.1*. 2022. World Wide Web Consortium. https://www.w3.org/TR/2022/REC-vc-data-model-20220303/.

[12] W3C REC-cid-1.0-20250515, JONES, Michael and Manu SPORNY (eds.). *Controlled Identifiers v1.0*. 2025. World Wide Web Consortium. https://www.w3.org/TR/2025/REC-cid-1.0-20250515/.

[13] Cryptographic Event Log, Digital Bazaar. *Cryptographic Event Log*. https://digitalbazaar.github.io/cel-spec/.

[14] W3C REC-vc-di-eddsa-20250515, SPORNY, Manu, Ted THIBODEAU JR and Greg BERNSTEIN (eds.). *Data Integrity EdDSA Cryptosuites v1.0*. 2025. World Wide Web Consortium. https://www.w3.org/TR/2025/REC-vc-di-eddsa-20250515/.

Document contributors

Markus Sabadello (lead editor), Kurt Cagle, Scott Carroll, Bastiaan den Braber, Stephane Fellah, Jacqueline Hynes, George Percivall, Christine Perey, Capm Petersen, Reese Plews, Gabriel René, Lior Saar, Hari Thiruvengada, Ronald Tse

Spatial Web Foundation leadership

Gabriel René	Executive Director / Founder
Dan Mapes	Managing Director / Founder
Bastiaan den Braber	Director of Operations
George Percivall	Distinguished Engineering Fellow
Dan Richardson	Director of Market Analysis
Dr. Sarah Grace Manski	Senior Ethics Advisor

Comments about the Spatial Web and this document can be sent to the Spatial Web Foundation at info@spatialwebfoundation.org