

SD-JWT-based Verifiable Credentials (SD-JWT VC)

[draft-terbu-oauth-sd-jwt-vc-00](#)

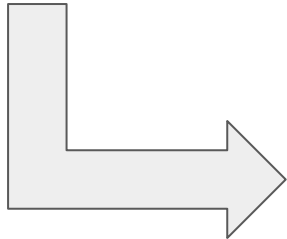
- Oliver Terbu (SpruceID)
- Daniel Fett (Authlete)

Recap SD-JWT

- Adopted by the OAuth WG
- Enables selective disclosure (SD) of individual claims included in a JWT
- No assumption about the included JWT claims
- Not a Verifiable Credential format

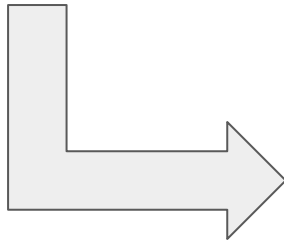
Recap SD-JWT

```
{  
  "sub": "user_42",  
  "given_name": "John",  
  "family_name": "Doe"  
}
```



Issue SD-JWT
and
Disclosures
with given
input claim set

```
{  
  "_sd": [  
    "CrQe7...PgkJP",  
    "JzYjH...4svli"  
  ],  
  "iss": "https://example.com",  
  "iat": 1683000000,  
  "exp": 1883000000,  
  "sub": "user_42",  
  "_sd_alg": "sha-256",  
}
```



Selectively disclose claims with
given SD-JWT and Disclosures

1. Issuer gives Holder:
SD-JWT + all
Disclosures

**2. Holder gives
Verifier:** SD-JWT +
selected Disclosures

3. Verifier checks:
signature of SD-JWT +
Disclosures are
included in SD-JWT

Claim given_name:

* SHA-256 Hash: jsu9yVuIwQQlhFLM_3JlzMaSFzgLhQG0DpfayQwLUK4

* Disclosure:

WyIyR0xDNDJzS1F2ZUNmR2ZyeU5STjI3IiwgImdpdmVuX25hbWUiLCAiSm9o
biJd

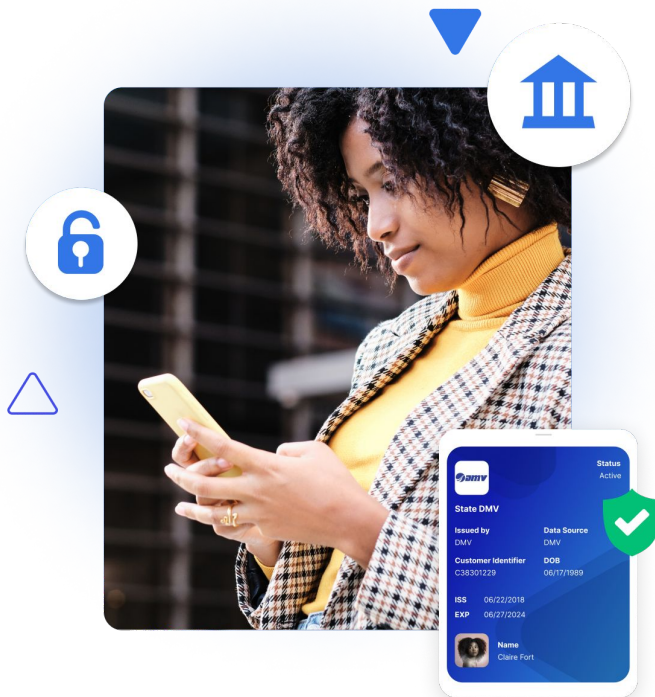
* Contents: ["2GLC42sKQveCfGfryNRN9w", "given_name", "John"]

Verifiable Credential

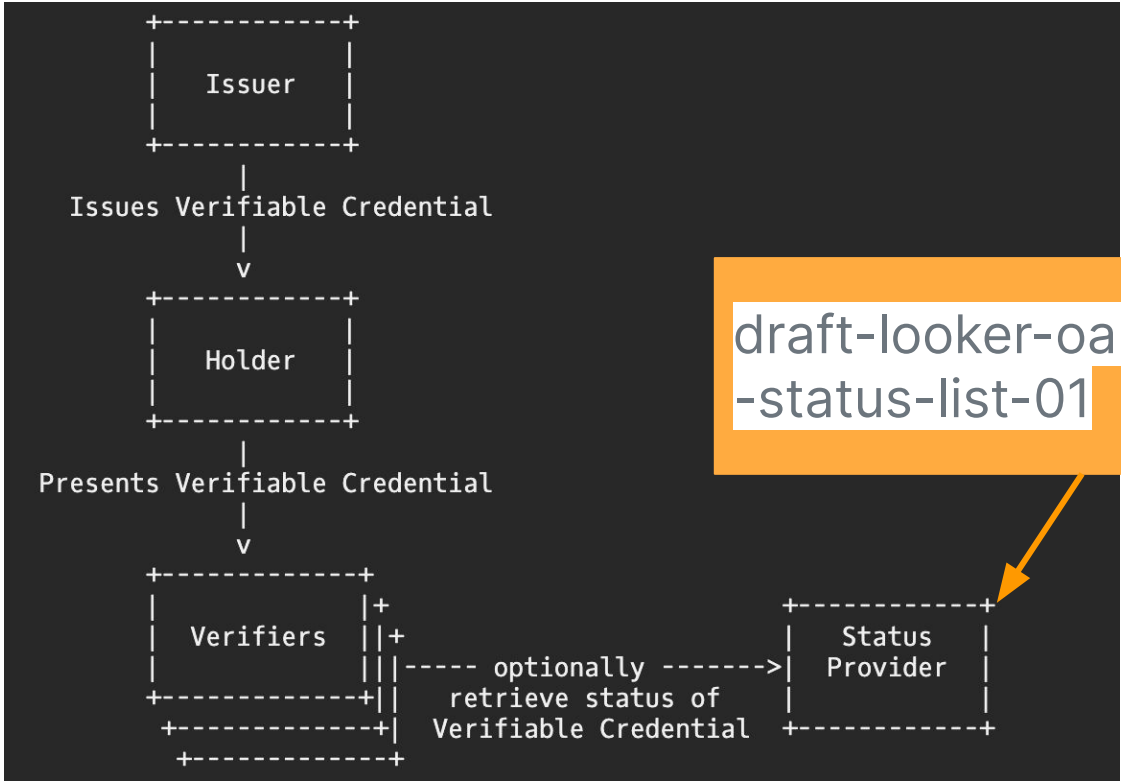
- An Issuer-signed assertion with claims about a Subject.

SD-JWT Verifiable Credentials (SD-JWT VC)

- Defines a profile of SD-JWT for Verifiable Credentials, similar to an ID Token is a specific profile of JWT
- Defines data formats and media types for Verifiable Credentials based on SD-JWTs with JSON payloads
- Defines validation and processing rules for verifiers and holders of Verifiable Credentials based on SD-JWTs
- Includes support for plain JWTs (non-SD)



Issuer-Holder-Verifier Model



Use Cases

- Issuing and presenting digital identity documents
- Supply chain integrity and transparency (SCITT) and content authenticity
- Client attestations
- Covers Issuer-Verifier-Holder Model cases with high security requirements ...
 - Enables cloning/forgery/MITM protection
 - Can use hardware cryptography
 - (Cloud) HSMs
 - Hardware-backed keystores on mobile devices
 - Can be combined with FIPS-compliant or PQC algorithms

Why not other Technologies?

- “Simple” is a feature
 - Uses a JSON-based data model and JWT Claim Sets
 - Builds on well-understood technologies
 - Other technologies are
 - more complex
 - rely on less proven building blocks
 - less composable
 - Have licensing issues
 - Do not support hardware-backed cryptography
- SD-JWT and JWTs?
 - Good basis, but do not define specific requirements for Verifiable Credentials

SD-JWT VC

- Work started in April 2023
- Referenced by OIDF OID4VC High Assurance Profile
- Interested: eIDAS 2.0 ARF
- Current Github activity
 - 28 open issues
 - 12 contributors

Why OAuth WG?

- SD-JWT VC is based on specifications from OAuth WG
 - Data model uses JWT Claims and is JSON only
 - Uses existing (SD-)JWT security
 - Not based on W3C Verifiable Credential Data Model (VCDM)
- Community
 - Existing contributors are members of OAuth WG
 - OAuth members are the right experts for SD-JWT VC with focus on ...
 - Security & privacy
 - Online and remote flows

Next Steps

- Reviews from OAuth WG
- Work on open issues
- Call for adoption?

Deep Dive

Data Format

- Based on SD-JWT and contains
 - Issuer-signed JWT
 - Disclosures
 - Optional key binding JWT for presentations
- Has new media type: `application/vc+sd-jwt`
 - IANA consideration → media type registry
 - Serialized SD-JWT with specific JWT Claim Sets
 - Indicates JWT claim sets with plain JSON data
 - Registered IANA JWT claims; existing and new ones
 - Public and private claims

Data Format

- Required issuer-signed JWT Headers:
 - "typ": "vc+sd-jwt"

Data Format

- JWT Claim Sets in issuer-signed JWT
 - `iss` (required)
VC issuer
 - `iat` (required)
Issuance date of the VC proof
 - `nbf` (optional)
Valid from date of the VC proof
 - `exp` (optional)
Valid until date of the VC proof
 - `cnf` (optional)
Cryptographic key binding method for cloning protection

Data Format

- New JWT Claim Sets in issuer-signed JWT
(IANA considerations)
 - type (required)
VC type (IANA considerations)
 - status (optional)
Information how to read VC status

As defined in

<https://datatracker.ietf.org/doc/draft-looker-oauth-jwt-cwt-status-list/>

Data Format

- `type`
 - IANA consideration → JWT claim registry
 - Defines required and optional claims in Issuer-signed JWT and SD-JWT Disclosures
 - SD and non-SD JWT claims

Data Format

- JWT Claim Set in key binding JWT
 - Now, as defined in latest SD-JWT spec
 - *Note, prior version of SD-JWT did not specify claims*

Verification and Processing

- Defines verification and processing rules for SD-JWT VCs
 - Requires SD-JWT verification and processing
 - And additional rules ...
 - Optional validation of the VC status based on the `status` claim
 - Optional mechanism to retrieve the public key of the issuer based on JWT Issuer Metadata

JWT Issuer Metadata

- JWT Issuer Metadata
 - Based on issuer identifier → `iss` claim
- JWT Issuer Metadata Request
 - Based on `.well-known` URL
 - `/.well-known/jwt-issuer`
- JWT Issuer Metadata Response Parameters
 - `issuer`
 - `jwks_uri`
 - `jwks`

JWT Issuer Metadata

```
{
  "issuer": "https://example.com",
  "jwks": {
    "keys": [
      {
        "kid": "doc-signer-05-25-2022",
        "e": "AQAB",
        "n": "nj3YJwsLU ... 0wMuzifQrMI9bQ",
        "kty": "RSA"
      }
    ]
  }
}

{
  "issuer": "https://example.com",
  "jwks_uri": "https://jwt-issuer.example.org/my_public_keys.jwks"
}
```

Verification and Processing

- Defines verification and processing rules for presentations of SD-JWT VCs
 - Requires SD-JWT verification and processing
 - And additional rules ...
 - Optional key binding JWT has to be verified by `cnf` claim

Example: Input JWT Claim Set for SD-JWT

```
{
  "iss": "https://pid-provider.memberstate.example.eu",
  "iat": 1541493724,
  "type": "PersonIdentificationData",
  "first_name": "Erika",
  "family_name": "Mustermann",
  "nationalities": [
    "DE"
  ],
  "birth_family_name": "Schmidt",
  "birthdate": "1973-01-01",
  "address": {
    "postal_code": "12345",
    "locality": "Irgendwo",
    "street_address": "Sonnenstrasse 23",
    "country_code": "DE"
  },
  "is_over_18": true,
  "is_over_21": true,
  "is_over_65": false
}
```


Example: Decoded Issuer-signed JWT Payload

```
{
  "_sd": [
    "09TbSuo12i2CqZbg31AFgbGy_UnMIXIHoMjsELpukqg",
    "0n9yzFSWvK_BUHiaMhm12ghrCtVahrGJ6_-kZP-ySq4",
    "4VoA3a1VmPxmdC8WIn3pOqQf3gfOVOvDYsN5E5R5Kd0",
    "5A88AmauAao-QANao95CYUkUPNTid_gAK8aYtZ9RZwc",
    "910byr3UVRqRzQoPzBsc20m-eMgpZAhLN6z8NoGF5mc",
    "Ch-DBcL3kb4VbHIwtknnZdNUHthEq9MZjoFdg6idiho",
    "I00fcFUoDXCucp5yy2ujqPssDVGaWNiUlinZ_awD0gc",
    "X9MaPaFWmQYpfHEdytRdaclnYoEru8EztBEUQuWOe44",
    "Y1urWJV_-HBGnSf9tFOwvH4cICRBCiKwEHfkXFSfjpo",
    "zNhKoraaq--x7BWWIVhbGXu1XXXLM8ivZXD3m2FZMgs",
    "xpsq6cxQHDSOnZWhrqBckTkOM_efElUnDFXOFmowLSE",
    "zU4521kGbEKh8ZuH_8Kx3CUvn1F4y1gZLqlDTgX_8Pk"
  ],
  "iss": "https://pid-provider.memberstate.example.eu",
  "iat": 1541493724,
  "exp": 1883000000,
  "type": "PersonIdentificationData",
  "_sd_alg": "sha-256",
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x":
        "TCAER19Zvu3OHF4j4W4vfSVoHIP1ILilDls7vCeGemc",
      "y":
        "ZxjiiWWbZMQGHVWKVQ4hbSIirsVfuecCE6t4jT9F2HZQ"
    }
  }
}
```

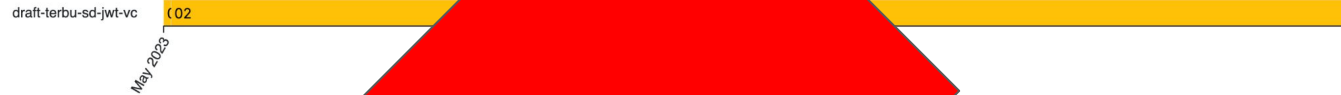
(Previous) IETF Data Tracker

SD-JWT-based Verification of Credentials (SD-JWT VC) Payloads (SD-JWT VC)
draft-terbu-sd-jwt-vc

Status [Email expansions](#) [History](#)

Versions:

[00](#) [01](#) [02](#)



Document	Type	Authors	Last updated

Current IETF Data Tracker

SD-JWT-based Verifiable Credentials (SD-JWT VC) draft-terbu-oauth-sd-jwt-vc-00

Status [Email expansions](#) [History](#)

Versions:

00



Document	Type	Active Internet-Draft (individual)
	Authors	Oliver Terbu ✉, Daniel Fett ✉
	Last updated	2023-07-10

Open Q&A

Thank you