Workgroup: Digital Credentials Protocols
Published: 17 January 2025
Authors: K. Yasuda *SPRIND*
T. Lodderstedt *SPRIND*

# OpenID4VC High Assurance Interoperability Profile - draft 01

## Abstract

This document defines a profile of OpenID for Verifiable Credentials in combination with the credential formats IETF SD-JWT VC [I-D.ietf-oauth-sd-jwt-vc] and ISO mdoc [ISO.18013-5]. The aim is to select features and to define a set of requirements for the existing specifications to enable interoperability among Issuers, Wallets and Verifiers of Credentials where a high level of security and privacy is required. The profiled specifications include OpenID for Verifiable Credential Issuance [OIDF.OID4VCI], OpenID for Verifiable Presentations [OIDF.OID4VP], Self-Issued OpenID Provider v2 [OIDF.SIOPv2], IETF SD-JWT VC [I-D.ietf-oauth-sd-jwt-vc], and ISO mdoc [ISO.18013-5].

## Table of Contents

# 1. Introduction

This document defines a set of requirements for the existing specifications to enable interoperability among Issuers, Wallets and Verifiers of Credentials where a high level of security and privacy is required. This document is an interoperability profile that can be used by implementations in various contexts, be it a certain industry or a certain regulatory environment.

This document is not a specification, but a profile. It refers to the specifications required for implementations to interoperate among each other and for the optionalities mentioned in the referenced specifications, defines the set of features to be mandatory to implement.

The profile uses OpenID for Verifiable Credential Issuance [OIDF.OID4VCI] and OpenID for Verifiable Presentations [OIDF.OID4VP] as the base protocols for issuance and presentation of Credentials, respectively. The credential formats used are IETF SD-JWT VC as specified in [I-D.ietf-oauth-sd-jwt-vc] and ISO mdoc [ISO.18013-5]. Additionally, considerations are given on how the issuance of Credentials in both IETF SD-JWT VC [I-D.ietf-oauth-sd-jwt-vc] and ISO mdoc [ISO.18013-5] formats can be performed in the same transaction.

A full list of the open standards used in this profile can be found in Overview of the Open Standards Requirements (reference).

### 1.1. Audience Target audience/Usage

The audience of the document is implementers that require a high level of security and privacy for their solutions. A non-exhaustive list of the interested parties includes eIDAS 2.0, California Department of Motor Vehicles, Open Wallet Foundation (OWF), IDunion, GAIN, and the Trusted Web project of the Japanese government, but is expected to grow to include other jurisdictions and private sector companies.

## 2. Terminology

This specification uses the terms "Holder", "Issuer", "Verifier", "Wallet", and "Verifiable Credential" as defined in @!OIDF.OID4VCI] and [OIDF.OID4VP].

## 3. Scope

The following aspects are in scope of this interoperability profile:

- Profile of OpenID4VCI to issue IETF SD-JWT VCs, including
  - Wallet Attestation
- Profile of OpenID4VP to present IETF SD-JWT VCs
- Profile of OpenID4VP over the W3C Digital Credentials API [w3c.digital_credentials_api] to present
  - IETF SD-JWT VCs
  - ISO mdocs
- Protocol for User Authentication by the Wallet at a Verifier (SIOP v2)
- Profile of IETF SD-JWT VC that includes the following aspects
  - Status management of the Credentials, including revocation
  - Cryptographic Key Binding
  - Issuer key resolution
  - Issuer identification (as prerequisite for trust management)
- Crypto Suites

Note that when OpenID4VP is used, the Wallet and the Verifier can either be remote or in-person.

Assumptions made are the following:

- The issuers and verifiers cannot pre-discover Wallet's capability
- The issuer is talking to the Wallet supporting the features defined in this profile (via Wallet invocation mechanism)
- There are mechanisms in place for the verifiers and issuers to discover each other's capability

### 3.1. Out of Scope

The following items are out of scope for the current version of this document, but might be added in future versions:

- Trust Management, i.e. authorization of an issuer to issue certain types of credentials, authorization of the Wallet to be issued certain types of credentials, authorization of the Verifier to receive certain types of

credentials.
- Protocol for presentation of Verifiable Credentials for offline use-cases, e.g. over BLE.
- Profile of OpenID4VCI to issue ISO mdoc [ISO.18013-5] is defined in ISO 23220-3.
- Profile of OpenID4VP without using W3C Digital Credentials API to present ISO mdocs is defined in [ISO.18013-7]. For more details, also see Annex B.3 in [OIDF.OID4VP].

### 3.2. Scenarios/Business Requirements

- Combined Issuance of SD-JWT VC and mdoc
- Both issuer-initiated and wallet-initiated issuance
- eIDAS PID and (Q)EAA as defined in eIDAS ARF 1.0

### 3.3. Standards Requirements

This specification enables interoperable implementations of the following flows:

- Issuance of IETF SD-JWT VC using OpenID4VCI
- Presentation of IETF SD-JWT VC using OpenID4VP
- Presentation of IETF SD-JWT VC using OpenID4VP over W3C Digital Credentials API
- Presentation of ISO mdocs using OpenID4VP over W3C Digital Credentials API

Implementations of this specification do not have to implement all of the flows listed above, but they MUST be compliant to all of the requirements for a particular flow they chose to implement.

## 4. OpenID for Verifiable Credential Issuance

Both the Wallet and the Credential Issuer:

- MUST support both pre-authorized code flow and authorization code flow.
- MUST support protocol extensions for the SD-JWT VC credential format profile as defined in Section 8.2.
- MUST support sender-constrained tokens using the mechanism defined in [RFC9449].
- MUST support [RFC7636] with S256 as the code challenge method.

Both Wallet initiated and Issuer initiated issuance is supported.

### 4.1. Credential Offer

- The Grant Types `authorization_code` and `urn:ietf:params:oauth:grant-type:pre-authorized_code` MUST be supported as defined in Section 4.1.1 in [OIDF.OID4VCI]
- For Grant Type `authorization_code`, the Issuer MUST include a scope value in order to allow the Wallet to identify the desired credential type. The Wallet MUST use that value in the `scope` Authorization parameter. For Grant Type `urn:ietf:params:oauth:grant-type:pre-authorized_code`, the pre-authorized code is used by the issuer to identify the credential type(s).
- As a way to invoke the Wallet, at least a custom URL scheme `haip://` MUST be supported. Implementations MAY support other ways to invoke the Wallets as agreed by trust frameworks/ecosystems/jurisdictions, not limited to using other custom URL schemes.

Note: The Authorization Code flow does not require a Credential Offer from the Issuer to the Wallet. However, it is included in the feature set of the Credential Offer because it might be easier to implement with existing libraries and on top of existing implementations than the pre-authorized code Grant Type.

Both sending Credential Offer same-device and cross-device is supported.

## 4.2. Authorization Endpoint

- MUST use Pushed Authorization Requests (PAR) [RFC9126] to send the Authorization Request.
- Wallets MUST authenticate itself at the PAR endpoint using the same rules as defined in Section 4.3 for client authentication at the token endpoint.
- MUST use the `scope` parameter to communicate credential type(s) to be issued. The scope value MUST map to a specific Credential type. The scope value may be pre-agreed, obtained from the Credential Offer, or the Credential Issuer Metadata.
- The `client_id` value in the PAR request MUST be a string that the Wallet has used as the `sub` value in the client attestation JWT.

## 4.3. Token Endpoint

- The Wallets MUST perform client authentication as defined in [I-D.ietf-oauth-attestation-based-client-auth].
- Refresh tokens are RECOMMENDED to be supported for credential refresh. For details, see Section 13.5 in [OIDF.OID4VCI].
- The Wallet Attestation JWT scheme is defined in Section 4.3.1.

Note: It is RECOMMENDED to use ephemeral client attestation JWTs for client authentication in order to prevent linkability across Credential Issuers.

Note: Issuers should be mindful of how long the usage of the refresh token is allowed to refresh a credential, as opposed to starting the issuance flow from the beginning. For example, if the User is trying to refresh a credential more than a year after its original issuance, the usage of the refresh tokens is NOT RECOMMENDED.

### 4.3.1. Wallet Attestation Schema

Wallets MUST use attestations following the definition given in [I-D.ietf-oauth-attestation-based-client-auth].

In addition to this definition, the Wallet Attestation MAY contain the following claims in the `cnf` element:

- `key_type`: OPTIONAL. JSON String that asserts the security mechanism the Wallet uses to manage the private key associated with the public key given in the `cnf` claim. This mechanism is based on the capabilities of the execution environment of the Wallet, this might be a secure element (in case of a Wallet residing on a smartphone) or a Cloud-HSM (in case of a cloud Wallet). This specification defines the following values for `key_type`:
  - `software`: It MUST be used when the Wallet uses software-based key management.
  - `hardware`: It MUST be used when the Wallet uses hardware-based key management.
  - `tee`: It SHOULD be used when the Wallet uses the Trusted Execution Environment for key management.
  - `secure_enclave`: It SHOULD be used when the Wallet uses the Secure Enclave for key management.
  - `strong_box`: It SHOULD be used when the Wallet uses the Strongbox for key management.
  - `secure_element`: It SHOULD be used when the Wallet uses a Secure Element for key management.
  - `hsm`: It SHOULD be used when the Wallet uses Hardware Security Module (HSM).
- `user_authentication`: OPTIONAL. JSON String that asserts the security mechanism the Wallet uses to authenticate the user to authorize access to the private key associated with the public key given in the `cnf` claim. This specification defines the following values for `user_authentication`:

- `system_biometry`: It MUST be used when the key usage is authorized by the mobile operating system using a biometric factor.
- `system_pin`: It MUST be used when the key usage is authorized by the mobile operating system using personal identification number (PIN).
- `internal_biometry`: It MUST be used when the key usage is authorized by the Wallet using a biometric factor.
- `internal_pin`: It MUST be used when the key usage is authorized by the Wallet using PIN.
- `secure_element_pin` It MUST be used when the key usage is authorized by the secure element managing the key itself using PIN.

The Wallet Attestation MAY also contain the following claim:

- `aal`: OPTIONAL. JSON String asserting the authentication level of the Wallet and the key as asserted in the `cnf` claim.

To obtain the Issuer's Public key for verification, Wallet attestions MUST support web-based key resolution as defined in Section 5 of [I-D.ietf-oauth-sd-jwt-vc]. The JOSE header `kid` MUST be used to identify the respective key.

This is an example of a Wallet Instance Attestation:

```
{
  "typ": "wallet-attestation+jwt",
  "alg": "ES256",
  "kid": "1"
}
.
{
  "iss": "<identifier of the issuer of this Wallet attestation>",
  "sub": "<`client_id` of the OAuth client>",
  "iat": 1516247022,
  "exp": 1541493724,
  "aal" : "https://trust-list.eu/aal/high",
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "TCAER19Zvu3OHF4j4W4vfSVoHIP1ILilDls7vCeGemc",
      "y": "ZxjiWWbZMQGHVWKVQ4hbSIirsVfuecCE6t4jT9F2HZQ"
    },
    "key_type": "strong_box",
    "user_authentication": "system_pin",
  }
}
```

## 4.4. Credential Endpoint

- The JWT proof type MUST be supported.

## 4.5. Server Metadata

- The Credential Issuer MUST publish a mapping of every Credential Type it supports to a scope value.

# 5. OpenID for Verifiable Presentations profile for IETF SD-JWT VC

Requirements for both the Wallet and the Verifier:

- As a way to invoke the Wallet, at least a custom URL scheme `haip://` MUST be supported. Implementations MAY support other ways to invoke the wallets as agreed by trust frameworks/ecosystems/jurisdictions, not limited to using other custom URL schemes.
- Response type MUST be `vp_token`.
- Response mode MUST be `direct_post.jwt`. The Verifier MUST return `redirect_uri` in response to the HTTP POST request from the Wallet, where the Wallet redirects the User to, as defined in Section 7.2 of [OIDF.OID4VP]. Implementation considerations for the response mode `direct_post.jwt` are given in Section 12.4 of [OIDF.OID4VP].
- Authorization Request MUST be sent using the `request_uri` parameter as defined in JWT-Secured Authorization Request (JAR) [RFC9101].
- The Client Identifier Scheme as introduced in Section 5.10 of [OIDF.OID4VP] MUST be either `x509_san_dns` or `verifier_attestation`. The Wallet MUST support both. The Verifier MUST support at least one.
- To obtain the issuer's public key for verification, verifiers MUST support Web-based key resolution, as defined in Section 5 of [I-D.ietf-oauth-sd-jwt-vc]. The JOSE header `kid` MUST be used to identify the respective key.
- Presentation Definition JSON object MUST be sent using a `presentation_definition` parameter.
- The following features from the DIF Presentation Exchange v2.0.0 MUST be supported. A JSON schema for the supported features is in Appendix B:
  - In the `presentation_definition` object, `id`, `input_descriptors` and `submission_requirements` properties MUST be supported.
  - In the `input-descriptors` object, `id`, `name`, `purpose`, `group`, `format`, and `constraints` properties MUST be supported. In the `constraints` object, `limit_disclosure`, and `fields` properties MUST be supported. In the `fields` object, `path` and `filter` properties MUST be supported. A `path` MUST contain exactly one entry with a static path to a certain claim. A `filter` MUST only contain `type` elements of value `string` and `const` elements.
  - In the `submission_requirements` object, `name`, `rule (pickonly)`, `count`, `from` properties MUST be supported.

# 6. OpenID for Verifiable Presentations over W3C Digital Credentials API

The following requirements apply for both, the Wallet and the Verifier, unless specified otherwise:

- MUST support Annex A in [OIDF.OID4VP] that defines how to use OpenID4VP over the W3C Digital Credentials API.
  - The Wallet MUST support both signed and unsigned requests as defined in Annex A.3.1 and A.3.2 of [OIDF.OID4VP]. The Verifier MAY support signed requests, unsigned requests, or both.
- Wallet Invocation is done via the W3C Digital Credentials API or an equivalent platform API. Any other mechanism, including Custom URL schemes, MUST NOT be used.
- Response Mode MUST be `dc_api.jwt`. The response MUST be encrypted.

- Response encryption MUST be performed as specified in Section 7.3 of [OIDF.OID4VP]. The JWE `alg` (algorithm) header parameter (see Section 4.1.1 of [RFC7516]) value `ECDH-ES` (as defined in Section 4.6 of [RFC7518]), with key agreement utilizing keys on the `P-256` curve (see Section 6.2.1.1 of [RFC7518]) MUST be supported. The JWE `enc` (encryption algorithm) header parameter (see Section 4.1.2 of [RFC7516]) value `A128GCM` (as defined in Section 5.3 of [RFC7518]) MUST be supported.
- The DCQL query and response as defined in Section 6 of [OIDF.OID4VP] MUST be used. Presentation Exchange as defined in Sections 5.4 and 5.5 of [OIDF.OID4VP] MUST NOT be used. Below is the list of features in the DCQL query and response that MUST be supported:
  - tbd (https://github.com/openid/oid4vc-haip/issues/142)

### 6.1. ISO mdoc specific requirements for OpenID for Verifiable Presentations over W3C Digital Credentials API

Requirements for both the Wallet and the Verifier:

- The Credential Format Identifier MUST be `mso_mdoc`.
- ISO mdoc Credential Format specific DCQL parameters as defined in Annex B.3.1 of [OIDF.OID4VP] MUST be used.
- Verifier MAY request more than one Credential in the same request.
- When multiple ISO mdocs are being returned, each ISO mdoc MUST be returned in a separate `DeviceResponse` (as defined in 8.3.2.1.2.2 of [ISO.18013-5]), each matching to a respective DCQL query. Therefore, the resulting `vp_token` contains multiple `DeviceResponse` instances.
- The `SessionTranscript` and `Handover` CBOR structures MUST be generated in accordance with Annex B.3.4.1 of [OIDF.OID4VP].

### 6.2. IETF SD-JWT VC specific requirements for OpenID for Verifiable Presentations over W3C Digital Credentials API

Requirements for both the Wallet and the Verifier:

- The Credential Format identifier MUST be `dc+sd-jwt`.
- IETF SD-JWT VC Credential Format specific DCQL parameters as defined in Section 6.4.1 of [OIDF.OID4VP] MUST be used.

## 7. Self-Issued OP v2

To authenticate the user, subject identifier in a Self-Issued ID Token MUST be used as defined in [OIDF.SIOPv2].

- As a way to invoke the Wallet, at least a custom URL scheme `haip://` MUST be supported. Implementations MAY support other ways to invoke the Wallets as agreed by trust frameworks/ecosystems/jurisdictions, not limited to using other custom URL schemes.
- `subject_syntax_types_supported` value MUST be `urn:ietf:params:oauth:jwk-thumbprint`

## 8. SD-JWT VCs

This profile defines the following additional requirements for IETF SD-JWT VCs as defined in [I-D.ietf-oauth-sd-jwt-vc].

- Compact serialization MUST be supported as defined in [I-D.ietf-oauth-selective-disclosure-jwt]. JSON serialization MAY be supported.

- The following JWT Claims MUST be supported Content (differentiate issuance & presentation)

| Claim | SD-JWT as issued by the Issuer | Normative Definition |
|---|---|---|
| iss | MUST | [RFC7519], Section 4.1.1 |
| iat | MUST | [RFC7519], Section 4.1.6 |
| exp | SHOULD (at the discretion of the issuer) | [RFC7519], Section 4.1.4 |
| cnf | MUST | [RFC7800] |
| vct | MUST | [I-D.ietf-oauth-sd-jwt-vc] |
| status | SHOULD (at the discretion of the issuer) | [I-D.ietf-oauth-status-list] |

*Table 1*

- The Issuer MUST NOT make any of the JWT Claims in the table above to be selectively disclosable, so that they are always present in the SD-JWT-VC presented by the Holder.
- It is at the discretion of the Issuer whether to use `exp` claim and/or a `status` claim to express the validity period of an SD-JWT-VC. The Wallet and the verifier MUST support both mechanisms.
- The `iss` claim MUST be an HTTPS URL. The `iss` value is used to obtain Issuer's signing key as defined in Section 8.1.
- The `vct` JWT claim as defined in [I-D.ietf-oauth-sd-jwt-vc].
- The `cnf` claim [RFC7800] MUST conform to the definition given in [I-D.ietf-oauth-sd-jwt-vc]. Implementations conforming to this profile MUST include the JSON Web Key [RFC7517] in the `jwk` sub claim.

Note: Currently this profile only supports presentation of credentials with cryptographic Holder Binding: the holder's signature is required to proof the credential is presented by the holder it was issued to. This profile might support claim-based and biometrics-based holder binding once OpenID for Verifiable Credentials adds support for other forms of Holder Binding. See https://bitbucket.org/openid/connect/issues/1537/presenting-vc-without-a-vp-using-openid4vp

Note: Re-using the same Credential across Verifiers, or re-using the same JWK value across multiple Credentials gives colluding Verifiers a mechanism to correlate the User. There are currently two known ways to address this with SD-JWT VCs. First is to issue multiple instances of the same credentials with different JWK values, so that if each instance of the credential is used at only one Verifier, it can be reused multiple times. Another is to use each credential only once (ephemeral credentials). It is RECOMMENDED to adopt one of these mechanisms.

Note: If there is a requirement to communicate information about the verification status and identity assurance data of the claims about the subject, the syntax defined by [OIDF.ekyc-ida] SHOULD be used. It is up to each jurisdiction and ecosystem, whether to require it to the implementers of this profile.

Note: If there is a requirement to provide the Subject's identifier assigned and maintained by the Issuer, the `sub` claim MAY be used. There is no requirement for a binding to exist between the `sub` and `cnf` claims. See the Implementation Considerations section in [I-D.ietf-oauth-sd-jwt-vc].

Note: In some credential types, it is not desirable to include an expiration date (eg: diploma attestation). Therefore, this profile leaves its inclusion to the Issuer, or the body defining the respective credential type.

### 8.1. Issuer identification and key resolution to validate an issued Credential

This profile supports two ways to represent and resolve the key required to validate the issuer signature of an SD-JWT VC, the web PKI-based key resolution and the x.509 certificates.

- Web-based key resolution: The key used to validate the Issuer's signature on the SD-JWT VC MUST be obtained from the SD-JWT VC issuer's metadata as defined in Section 5 of [I-D.ietf-oauth-sd-jwt-vc]. The JOSE header `kid` MUST be used to identify the respective key.
- x.509 certificates: the SD-JWT VC contains the issuer's certificate along with a trust chain in the `x5c` JOSE header. In this case, the `iss` value MUST be an URL with a FQDN matching a `dNSName` Subject Alternative Name (SAN) [RFC5280] entry in the leaf certificate.

Note: The issuer MAY decide to support both options. In which case, it is at the discretion of the Wallet and the Verifier which key to use for the issuer signature validation.

#### 8.1.1. Cryptographic Holder Binding between VC and VP

- For Cryptographic Holder Binding, a KB-JWT, as defined in [I-D.ietf-oauth-sd-jwt-vc], MUST always be present when presenting an SD-JWT VC.

### 8.2. OpenID4VC Credential Format Profile

A Credential Format Profile for Credentials complying with IETF SD-JWT VCs [I-D.ietf-oauth-sd-jwt-vc] is defined in Annex A.3 of [OIDF.OID4VCI] and Annex A.4 of [OIDF.OID4VP].

## 9. Crypto Suites

Issuers, holders and verifiers MUST support P-256 (secp256r1) as a key type with ES256 JWT algorithm for signing and signature validation whenever this profiles requires to do so:

- SD-JWT-VC
- Wallet Instance Attestation
- DPoP
- HB JWT
- Authorization request during presentation

SHA256 MUST be supported by all the entities as the hash algorithm to generate and validate the digests in the SD-JWT VC.

Note: When using this profile with other cryptosuites, it is recommended to be explicit about which entity is required to support which curve for signing and/or signature validation

## 10. Implementations Considerations

### 10.1. Validity Period of the Signature and the Claim Values

`iat` and `exp` JWT claims express both the validity period of both the signature and the claims about the subject, unless there is a separate claim used to express the validity of the claims.

# 11. Normative References


**[I-D.ietf-oauth-attestation-based-client-auth]** Looker, T., Bastian, P., and C. Bormann, "OAuth 2.0 Attestation-Based Client Authentication", Work in Progress, Internet-Draft, draft-ietf-oauth-attestation-based-client-auth-04, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-attestation-based-client-auth-04>.

**[I-D.ietf-oauth-sd-jwt-vc]** Terbu, O., Fett, D., and B. Campbell, "SD-JWT-based Verifiable Credentials (SD-JWT VC)", Work in Progress, Internet-Draft, draft-ietf-oauth-sd-jwt-vc-08, 3 December 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-sd-jwt-vc-08>.

**[I-D.ietf-oauth-selective-disclosure-jwt]** Fett, D., Yasuda, K., and B. Campbell, "Selective Disclosure for JWTs (SD-JWT)", Work in Progress, Internet-Draft, draft-ietf-oauth-selective-disclosure-jwt-15, 16 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-selective-disclosure-jwt-15>.

**[I-D.ietf-oauth-status-list]** Looker, T., Bastian, P., and C. Bormann, "Token Status List", Work in Progress, Internet-Draft, draft-ietf-oauth-status-list-06, 3 December 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-status-list-06>.

**[ISO.18013-5]** ISO/IEC JTC 1/SC 17 Cards and security devices for personal identification, "ISO/IEC 18013-5:2021 Personal identification — ISO-compliant driving license — Part 5: Mobile driving license (mDL) application", 2021, <https://www.iso.org/standard/69084.html>.

**[OIDF.OID4VCI]** Lodderstedt, T., Yasuda, K., and T. Looker, "OpenID for Verifiable Credential Issuance", 20 June 2022, <https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html>.

**[OIDF.OID4VP]** Terbu, O., Lodderstedt, T., Yasuda, K., Lemmon, A., and T. Looker, "OpenID for Verifiable Presentations", 20 June 2022, <https://openid.net/specs/openid-4-verifiable-presentations-1_0.html>.

**[OIDF.SIOPv2]** Microsoft, Jones, M. B., and T. Lodderstedt, "Self-Issued OpenID Provider V2", 18 December 2021, <https://openid.net/specs/openid-connect-self-issued-v2-1_0.html>.

**[OIDF.ekyc-ida]** yes, Fett, D., Haine, M., Pulido, A., Lehmann, K., and K. Koiwai, "OpenID Connect for Identity Assurance 1.0", 19 August 2022, <https://openid.net/specs/openid-connect-4-identity-assurance-1_0-ID4.html>.

**[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/info/rfc5280>.

**[RFC7516]** Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/info/rfc7516>.

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

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

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

[RFC7636] Sakimura, N., Ed., Bradley, J., and N. Agarwal, "Proof Key for Code Exchange by OAuth Public Clients", RFC 7636, DOI 10.17487/RFC7636, September 2015, <https://www.rfc-editor.org/info/rfc7636>.

[RFC7800] Jones, M., Bradley, J., and H. Tschofenig, "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)", RFC 7800, DOI 10.17487/RFC7800, April 2016, <https://www.rfc-editor.org/info/rfc7800>.

[RFC9101] Sakimura, N., Bradley, J., and M. Jones, "The OAuth 2.0 Authorization Framework: JWT-Secured Authorization Request (JAR)", RFC 9101, DOI 10.17487/RFC9101, August 2021, <https://www.rfc-editor.org/info/rfc9101>.

[RFC9126] Lodderstedt, T., Campbell, B., Sakimura, N., Tonge, D., and F. Skokan, "OAuth 2.0 Pushed Authorization Requests", RFC 9126, DOI 10.17487/RFC9126, September 2021, <https://www.rfc-editor.org/info/rfc9126>.

[RFC9449] Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., and D. Waite, "OAuth 2.0 Demonstrating Proof of Possession (DPoP)", RFC 9449, DOI 10.17487/RFC9449, September 2023, <https://www.rfc-editor.org/info/rfc9449>.

## 12. Informative References

[ISO.18013-7] ISO/IEC JTC 1/SC 17 Cards and security devices for personal identification, "ISO/IEC DTS 18013-7 Personal identification — ISO-compliant driving license — Part 7: Mobile driving license (mDL) add-on functions", 2024, <https://www.iso.org/standard/82772.html>.

[w3c.digital_credentials_api] Caceres, M., Goto, S., and T. Cappalli, "Digital Credentials API", <https://wicg.github.io/digital-credentials/>.

## Appendix A. Combined Issuance of SD-JWT VC and mdocs

- If combined issuance is required, the Batch Credential Endpoint MUST be supported.

## Appendix B. JSON Schema for the supported Presentation Definition properties

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "title": "Presentation Definition for a High Assurance Profile",
    "type": "object",
    "properties": {
      "presentation_definition": {
        "$ref": "#/definitions/presentation_definition"
      }
    },
    "definitions": {
      "presentation_definition": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string"
          },
          "input_descriptors": {
            "type": "array",
```

```
            "items": {
              "$ref": "#/definitions/input_descriptor"
            }
          },
          "submission_requirements": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/submission_requirement"
            }
          }
        },
        "required": [
          "id",
          "input_descriptors"
        ],
        "additionalProperties": false
      },
      "input_descriptor": {
        "type": "object",
        "additionalProperties": false,
        "properties": {
          "id": {
            "type": "string"
          },
          "name": {
            "type": "string"
          },
          "purpose": {
            "type": "string"
          },
          "format": {
            "$ref": "http://identity.foundation/claim-format-
registry/schemas/presentation-definition-claim-format-designations.json"
          },
          "group": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "constraints": {
            "type": "object",
            "additionalProperties": false,
            "properties": {
              "limit_disclosure": {
                "type": "string",
                "enum": [
                  "required",
                  "preferred"
                ]
              },
              "fields": {
                "type": "array",
                "items": {
                  "path": {
                    "type": "array",
                    "items": {
                      "type": "string"
                    }
                  },
                  "filter": {
                    "$ref": "http://json-schema.org/draft-07/schema#"
                  }
```

```
                }
              }
            }
          }
        },
        "required": [
          "id",
          "constraints"
        ]
      },
      "submission_requirement": {
        "type": "object",
        "oneOf": [
          {
            "properties": {
              "name": {
                "type": "string"
              },
              "rule": {
                "type": "string",
                "enum": [
                  "pick"
                ]
              },
              "count": {
                "type": "integer",
                "minimum": 1
              },
              "from": {
                "type": "string"
              }
            },
            "required": [
              "rule",
              "from"
            ],
            "additionalProperties": false
          }
        ]
      }
    }
  }
```

## Appendix C. Acknowledgements

We would like to thank Paul Bastian, Christian Bormann, Mike Jones, Oliver Terbu, Daniel Fett, and Giuseppe De Marco for their valuable feedback and contributions to this specification.

## Appendix D. Notices

developing specifications, and (ii) implementing Implementers Drafts and Final Specifications based on such documents, provided that attribution be made to the OIDF as the source of the material, but that such attribution does not indicate an endorsement by the OIDF.

The technology described in this specification was made available from contributions from various sources, including members of the OpenID Foundation and others. Although the OpenID Foundation has taken steps to help ensure that the technology is available for distribution, it takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this specification or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any independent effort to identify any such rights. The OpenID Foundation and the contributors to this specification make no (and hereby expressly disclaim any) warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to this specification, and the entire risk as to implementing this specification is assumed by the implementer. The OpenID Intellectual Property Rights policy requires contributors to offer a patent promise not to assert certain patent claims against other contributors and against implementers. The OpenID Foundation invites any interested party to bring to its attention any copyrights, patents, patent applications, or other proprietary rights that MAY cover technology that MAY be required to practice this specification.

## Appendix E. Document History

[[ To be removed from the final specification ]]

-02

- Add specific requirements for response encryption
- Add SessionTranscript requirements

-01

- Rename specification to enable non-SD-JWT credential formats to be included
- Require encrypted responses
- Remove reference to `client_id_scheme` parameter that no longer exists in OpenID4VP
- Refresh tokens are now optional

-00

- initial revision

## Authors' Addresses

**Kristina Yasuda**
SPRIND
Email: kristina.yasuda@sprind.org

**Torsten Lodderstedt**
SPRIND
Email: torsten@lodderstedt.net