

SD-JWT & SD-JWT VC 101

# SD-JWT & SD-JWT VC

- Formats for
  - enabling **selective disclosure and key binding for JWS/JWT** (SD-JWT)
  - **credentials** based on that format (SD-JWT VC)
- Attributes are structured as JSON
- Specified in the OAuth Working Group at the IETF
  - SD-JWT: Working group last call
  - SD-JWT VC: Working group draft

# SD-JWT

## **Selective Disclosure for JWTs**

using a simple, salted-hash based format  
— for verifiable credentials and more.

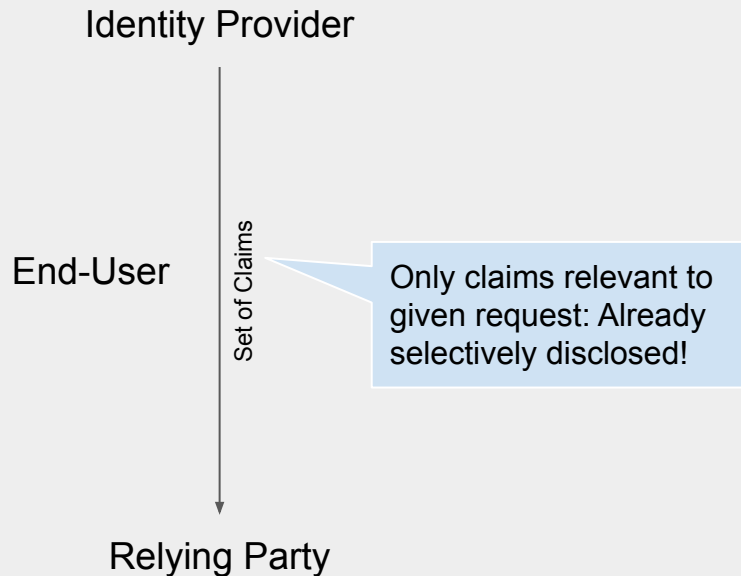


IETF Draft: <https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/>

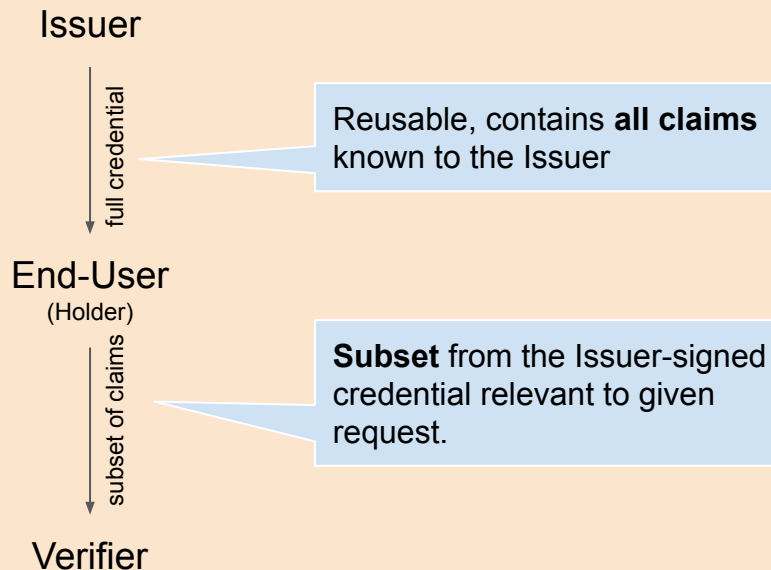
Daniel Fett  
Kristina Yasuda  
Brian Campbell

# Credential Issuance & Presentation Decoupled

## Identity Federation



## Wallet Model



# Selective Disclosure

**Issuer** issued a whole set of claims:

```
{
  "iss": "https://server.example.com",
  "sub": "some-user-identifier",
  "aud": "s6BhdRkqt3",
  "given_name": "John",
  "family_name": "Doe",
  "email": "johndoe@example.com",
  "phone_number": "+1-202-555-0101",
  "address": {
    "street_address": "123 Main St",
    "locality": "Anytown",
    "region": "Anystate",
    "country": "US"
  },
  "birthdate": "1940-01-01"
}
```

✓ signed  
by Issuer



But **Verifier** only needs a subset in a given request:

```
{
  "iss": "https://server.example.com",
  "sub": "some-user-identifier",
  "aud": "s6BhdRkqt3",
  "given_name": "John",
  "family_name": "Doe",
  "email": "johndoe@example.com",
  "phone_number": ████████████████████
  "address": {
    ████████████████████████████████████
    ████████████████████████████████████
    ████████████████████████████████████
    ████████████████████████████████████
  },
  "birthdate": ████████████████████
}
```

✓ signed  
by Issuer

# SD-JWT in 5 Simple Steps

## Step 1: Prepare User Data

```
{
  "iss": "https://example.com",
  "type": "IdentityCredential",
  "cnf": {"jwk": {"kty": "RSA", "n": "0vx....Kgw", "e": "AQAB" } },

  "given_name": "Max",
  "family_name": "Mustermann",
  "email": "mustermann@example.com",
  "address": {
    "street_address": "Musterstr. 23",
    "locality": "Berlin",
    "country": "DE"
  }
}
```

# SD-JWT in 5 Simple Steps

## Step 2: Create Disclosures

```
{
  "iss": "https://example.com",
  "type": "IdentityCredential",
  "cnf": {"jwk": {"kty": "RSA", "n": "0vx....Kgw", "e": "AQAB" } },

  "given_name": "Max", ..... ["G00r26n0-iw50ZcAoOilFw", "given_name", "Max"]
  "family_name": "Mustermann", ..... ["cSlbR135i0NjhsouMxrjjg", "family_name", "Mustermann"]
  "email": "mustermann@example.com", ..... ["oHDt43Vwuhpo8mzaprgCcw", "email", "mustermann@example.com"]
  "address": {
    "street_address": "Musterstr. 23", ..... ["rGc0KtY6WmflywTTKEWIEQ", "street_address", "Musterstr. 23"]
    "locality": "Berlin", ..... ["pGQMqx-2tH2XwC_eQCFn4g", "locality", "Berlin"]
    "country": "DE" ..... ["TI15M8G5UIxPiWNZ-VLYBA", "country", "DE"]
  }
}
```

↑                           ↑                   ↑  
salt                    claim name    claim value

# SD-JWT in 5 Simple Steps

## Step 3: Hash Disclosures & Replace Original Claims

```
{
  "iss": "https://example.com",
  "type": "IdentityCredential",
  "cnf": {"jwk": {"kty": "RSA", "n": "0vx....Kgw", "e": "AQAB" } },

  "_sd": [ "EW1o0egqa5mGcbytT5S-kAubcEjYEUwRkXlu2vC5l20",
           "FEx-ITht41I8_cn0SS-hvoLneX_RGlJo_8o2xRNhfdk",
           "igg7H5fn2eBEMIEkE5Ckbn23QuwDJlTYoKRip08dYIc" ],
  "address": {
    "_sd": [ "gqB5kmAwryr88aHjaAe0-USX6J0MaojukKsheo3800c",
            "w8InvxsPXdKoowuVpyBMgl1b9_R2b6Xpa30Y0IjgQro",
            "v0n1Ytcjr872fP3Wa750z17c-6_MOVdIUNtwLKKxZw0" ]
  }
}
```

- ← ["G00r26n0-iW50ZcAoOilFw", "given\_name", "Max"]
- ← ["cSlbR135i0NjhsouMxrjjg", "family\_name", "Mustermann"]
- ← ["oHDt43Vvuhpo8mzaprgCcw", "email", "mustermann@example.com"]
- ← ["rGc0KtY6WmflywTTKEWIEQ", "street\_address", "Musterstr. 23"]
- ← ["pGQMqx-2tH2XwC\_eQCFn4g", "locality", "Berlin"]
- ← ["TI15M8G5UIxPiWNZ-VLYBA", "country", "DE"]



# SD-JWT in 5 Simple Steps

## Step 4: Sign SD-JWT & Encode for Transport

```
{
  "iss": "https://example.com",
  eyJhbGciOiAiUmlrNTYiLCJkaXIjOiJmNBRU1VcUowY21MekQxa3pHemhlaUJhZzBZ
  UkF6VmRsZnhOMjgwTmdIYUEifQ.eyJpc3MiOiAiAiaHR0cHM6Ly9leGFtcGxlLmNvbS9pc
  3N1ZXIiLCJkaXIjOiJmNBRU1VcUowY21MekQxa3pHemhlaUJhZzBZU0EiLCJ1Ijoi
  WJHY1FTdS4uLi4tY3NGQ3VyLWtFZ1U4YXdhcEp6S25xREtndyIsICJlIjogIkFRQUIi
  fX0sICJ0eXB1IjogIk1kZW50aXR5OQ3JlZGVudG1hbCIjcmVkaW50aWwFsu3ViamVjd
  CI6IHsiX3NkIjogWyJFVzFvMGVncWE1bUdjYn10VDVTLWtBdWJjRwpZRVV3UmtYbHUyd
  kM1bDIWIiwgIkZFeC1JVEh0NDFUJ0F9jbjBTUy1odm9MbmVYX1JHbEppvXzhvMnhSTmhmZ
  GsiLCiAUXhKVi0yVjFIOG1jbHRSNnZWQzRtM3JlVTVhTkG5d2RKEjJVZG1Sb0kxRSIsI
  CJhdFVUwMVRzd1JBbDRHUTdQZUV0WGFNdZJmNHVJVG1Kclg00DV3TTh2NjdFIiwgImZUT
  XczdmtRUX3TDFYtnVZSzhIN3pCS0NiDv91awY2MFNsRzFweVhJVVEiLCiaWdnN0g1Z
  m4yZUJFTU1FaOU1Q2tibTIZUXV3REpsVf1vS1JpcDA4ZF1JYyIsICJ0cFV0bDcwaHBVX
  3hucnZaaTBHaEdvUlIexam10MxpZZ3Z2NU1ZMEF4N0tjI10sICJhZGRyZXNzIjogeyJf
  c2Q0i0iBbImdxQjVrbUF3eXJ5ODhhSGphQWVPLVVTWdZKT01hb2p1a0tzaGvVmhPMGMiL
  CAidk9ubF10Y2pyODcyZ1AzV2E3NU96bDdjLTZfTU9WZElVTnR3TEtLefp3MCI3J30
  EludnhzUFhkS29vd3VwchICTwdsMWI5X1IyYjZyCGEzT1lPSWpnUXJvI119fSwgImIhd
  CI6IDE1MTYyMzkwMjIsICJleHAiOiAAXNTE2MjQ3MDIyLCiaic2RfZGlnZXN0X2Rlcm12Y
  XRpb25fYXN1IjogInNoYS0yNTYifQ.1UHEPtLLUXOT51jH3gg-3C-ZidWzsB9Un-VxmM
  VdQtTbLLhwDTB6HJtt15p43yCXtZdpiZxtDI6fr07Tp0Dy_Umg3Q5_FxFj4WHnsVuVzu
  ASU8cFlGPi6xgH9D3w1G2hqepBS8DyQ5bA_p5kN_tKJVoP1xWhcQujRJ8kkEKQsRia4F
  hrBld18f41wgu_ipPqh1Ix4BVI7GJClZNx94nWPT7JUFkI6Y6JkahLp3S6gB0MxtmLAe
  Y0qkuz8Ve0ZNF1_CDog55kVtKaArorfoL6D6TEji__-w6YyU0PnIRJXJ0wrYfoyhN18LK
  AP38QYmpdR7z_rsvHpQHzFAPTmevnhDg
}
```

- ← ["G00r26n0-iw50ZcAoOilFw", "given\_name", "Max"]
- ← ["cSlbR135i0NjhsouMxrjgg", "family\_name", "Mustermann"]
- ← ["oHdt43Vwuhpo8mzaprgCcw", "email", "mustermann@example.com"]
- ← ["rGc0KtY6WmflywTTKEWIEQ", "street\_address", "Musterstr. 23"]
- ← ["pGQMqX-2tH2XwC\_eQCfN4g", "locality", "Berlin"]
- ← ["TI15M8G5UIxPiWNZ-VLYBA", "country", "DE"]

# SD-JWT in 5 Simple Steps

## Step 5: Base64url-encode Disclosures for Transport

```

{
  "iss": "https://example.com",
  eyJhbGciOiAiUmluYXNkaWkiLCIAia2lkIjogImNBRUlVcUowY21MekQxa3pHemhlaUJhZzBZ
UkF6VmRsZnhOMjgwTmdIYUEifQ.eyJpc3MiOiAiYHR0cHM6Ly91eGFtcGxlLmNvbS9pc3N1Z
XlIiLCAiY25mIjogeyJqd2siOiB7Imt0eSI6ICJSU0EiLCIAibiI6IClwdng3YWdvZ
WJHY1FTdS4uLi4tY3NGQ3VyLWtFZlU4YXdhcEp6S25xREtndyIsICJlIjogIkFRQUIif
XOsiLCJ0eXB1IjogIk1kZW50aXR5OjQ3JlZGVudG1hbCI6ICJlcjcmVkZW50aWwU3ViamVjd
Cl6IHsiX3NkIjogWyJFVzFvMGVncWE1bUdjYn10VDVTLWtBdWJjRwpZRVV3UmtYbHUyd
kM1bDIiWiIiwgIkZFeC1JVEhONDFjOF9jbJtUy1odm9MbmVYX1JHbEppvXzhvMnhSThmZ
GsiLCiAiuXhkiVi0yVjFIOG1jbHR5NnZwQzRtM3JlVTVhTkg5d2RKEjJVZG1Sb0kxRSIsI
CJhdFVuMVRzd1JBbDRHUtdQZUV0WGFndzJmNHVJVG1Kclg0ODV3TTh2NjdFIiwgImZUT
XczdmtRUX3TDFYtNVSZshIN3pCS0NidV91aWY2MFNsRzFweVhJVVeILCAiawDnN0g1Z
m4yZUJFTU1Fa0U1Q2tibTIZUXV3REpsVFlvS1JpcDA4ZFlJYyIsICJ0cFV0bDcwaHBVX
3hucnZaaTBHaEdvUlIexam10MXpZZ3Z2NUlZMEF4N0tjI10sICJhZGRyZXNzIjogeyJfc
2QiOiBbImdxQjVrbUf3eXJ5ODhhSGphQWVPLVVTWDZKT01hb2p1a0tzaGvvMzhPMGMiL
CAidk9ubF1OY2pyODcyZ1AzV2E3NU96bDdjLTZfTU9WZElVTnR3TETleFp3MCIsICJ30
EludnhzUFHks29vd3VwchlCTWdsMwiX1IyYjZyCGeZT1lPSWpnUXVjI119fSwgIm1hd
Cl6IDE1MTYyMzkWmjIsICJleHAiOiAxNTE2MjQ3MDIyLCIAic2RfZGlzNzX0X2Rlcm12Y
XRpb25fYXNjaWkiLCIAiX3NkIjogImNBRUlVcUowY21MekQxa3pHemhlaUJhZzBZUkF6
VMRsZnhOMjgwTmdIYUEifQ.1UHEPtLLUXOT51jH3gg-3C-ZidWzsB9Un-VxmM
VdQtTbLlhwDTB6HJtt15p43yCXtZdpiXztDI6fr07Tp0Dy_Umg3Q5_FxJf4WHnsVuVzU
ASU8cF1Gpi6xgH9D3w1G2hqepBS8DyQ5bA_p5kN_tKJVoP1xWhcQujRj8kkEKQsRia4F
hrBld18f41wgu_ipPqh1Ix4BVI7GJC1ZNX94nWPT7JUfKI6Y6JkahLf3S6gB0MxtmLAe
Y0qkuz8Ve0ZNfl_CDog55kVtKaRorfoL6D6TEji_-w6YyU0PnIRJXJ0wrYfoyhN18LK
AP38QYMPdr7z_rsvHpQHfAPtmevnhDG
}

```

```

- ["G00r26n0-iw50ZcAoOilFw", "given_name", "Max"]
  ["cSlbR135i0NhsouMyriig", "family_name", "Mustermann"]
~WyJHTzByMjZuTy1pVzUwWmNBb09pbEZ3IiwgImdpdmVhZlV0bG95bWUiLCIAiTWf4I10
~WyJjU2xiUjEzZlV0eXNkIiwgImZhbWlseV9uYW1lIiwgIk11c3Rlcm1
hbm4iXQ
~WyJvSER0NDNwd3VocG84bXphcHJnQ2N3IiwgImVtYWlsIiwgIm11c3Rlcm1hbm5AZXh
hbXBzZS5jb20iXQ
~WyJyR2MwS3RZnlldtZmx5d1RUS0VSUVRiIiwgImN0cmVldF9hZGRyZXNzIiwgIk11c3R
lcnN0ci4gMjMiXQ
~WyJwcm9ncFV4IiwgImNkbnN0IiwgIm9uYXNkaWkiLCIAiX3NkIjogImNBRUlVcUowY21
MekQxa3pHemhlaUJhZzBZUkF6VmRsZnhOMjgwTmdIYUEifQ.1UHEPtLLUXOT51jH3gg-3C-ZidWzsB9Un-VxmM
VdQtTbLlhwDTB6HJtt15p43yCXtZdpiXztDI6fr07Tp0Dy_Umg3Q5_FxJf4WHnsVuVzU
ASU8cF1Gpi6xgH9D3w1G2hqepBS8DyQ5bA_p5kN_tKJVoP1xWhcQujRj8kkEKQsRia4F
hrBld18f41wgu_ipPqh1Ix4BVI7GJC1ZNX94nWPT7JUfKI6Y6JkahLf3S6gB0MxtmLAe
Y0qkuz8Ve0ZNfl_CDog55kVtKaRorfoL6D6TEji_-w6YyU0PnIRJXJ0wrYfoyhN18LK
AP38QYMPdr7z_rsvHpQHfAPtmevnhDG
~WyJ1STE1TThHNVVJeFBpV05aLVZMmWUJBIiwgImNvdw50cnkiLCIAiREUiXQ

```

# Design Principles

## SD-JWT

Complexity	Selective disclosure, as simple as possible
Algorithms	Standard cryptography: JWS Signature + Hash function
Format	JWT & JSON
Security	Security-by-design Easy to understand & verify Hardware binding possible Cryptographic agility
Availability	Widely-available JWT libraries can be leveraged Already five independent implementations
Use Cases	Universal (beyond identity use cases)

Issuer

Issuance

**JWT**  
plain-text claims  
+ hashed Disclosures

```
{  
  "iss": "https://example.com",  
  "typ": "JWT",  
  "alg": "ES256",  
  "sub": "1234567890",  
  "aud": "https://example.com",  
  "exp": 1609459200,  
  "iat": 1609455600,  
  "nbf": 1609455600,  
  "claims": {  
    "disclosures": [ "disclosure1", "disclosure2" ]  
  }  
}
```

✓ signed  
by Issuer

**Disclosures**  
salt + claim name + claim value

```
WyJrSEhwOTEtZEFadDhtOUU0Smw0WGJRIiwgImdpdmVUX25hbWU1LCA1Sm9ob1Jld  
WyJQak1xcEdXbDRlQjRRcm9EaHFRdzb3IiwgImZhbn1seV9uYW11IiwgIkRvZS5Jd  
WyJ4bW1QNEpadE5XSUgTGTGtFRHQtb3IiwgImN0cmV1dF9hZGRyZXNzIiwgIkRvZS5Jd  
WyJLdGZzeHhUbTJtdzBZTFVjS1pvOHRIiwgImxvY2FsaXR5IiwgIkFueXRvd241XQ
```

End-User  
(Holder)

Presentation

Verifier

Issuer

Issuance

End-User  
(Holder)

Presentation

Verifier

**JWT**  
plain-text claims  
+ hashed Disclosures

```
{
  "iat": "https://example.com",
  "type": "id_token",
  "iss": "https://example.com",
  "exp": 1612134567,
  "sub": "1234567890",
  "aud": "https://example.com",
  "nonce": "1234567890",
  "disclosures": [
    {
      "salt": "1234567890",
      "claim_name": "scope",
      "claim_value": "openid profile email"
    }
  ]
}
```

✓ signed  
by Issuer

**Disclosures**  
salt + claim name + claim value

```
WyJrSEhwOTEtdEFadDhtOUU0Smw0WGJRIiwgImdpdmVx25hbWU1LCAiSm9obiJld
WyJQak1xcEdXbDRlQjRRcm9EaHFRdzB3IiwgImZhbn1seV9uYW11IiwgIkRvZSJD
WyJ4bm1QNEpadE5XSUgtTgtFRHQtbY1BIiwgImN0cmV1dF9hZGRyZXNzIiwgIkRvZXR0cmV1dCAXI1O
WyJLdGZzeHhUbTJtdzBZTFVjS1pVOHRBIiwgImxvY2FsaXR5IiwgIkFueXRvd24lXQ
```

**JWT**  
plain-text claims  
+ hashed Disclosures

```
{
  "iat": "https://example.com",
  "type": "id_token",
  "iss": "https://example.com",
  "exp": 1612134567,
  "sub": "1234567890",
  "aud": "https://example.com",
  "nonce": "1234567890",
  "disclosures": [
    {
      "salt": "1234567890",
      "claim_name": "scope",
      "claim_value": "openid profile email"
    }
  ]
}
```

✓ signed  
by Issuer

**Selected Disclosures**  
salt + claim name + claim value

```
WyJrSEhwOTEtdEFadDhtOUU0Smw0WGJRIiwgImdpdmVx25hbWU1LCAiSm9obiJld
WyJQak1xcEdXbDRlQjRRcm9EaHFRdzB3IiwgImZhbn1seV9uYW11IiwgIkRvZSJD
```



```
-----
```

Issuer

Issuance

End-User  
(Holder)

Presentation

Verifier

**JWT**  
plain-text claims  
+ hashed Disclosures

```
{
  "iat": "https://example.com",
  "type": "id_token",
  "iss": "https://example.com",
  "exp": 1516239024,
  "sub": "1234567890",
  "aud": "https://example.com",
  "scope": "openid",
  "nonce": "1234567890",
  "sd_hash": "1234567890",
  "sd": [
    {
      "salt": "1234567890",
      "claim_name": "sd_claim",
      "claim_value": "sd_claim_value"
    }
  ]
}
```

✓ signed  
by Issuer

**Disclosures**  
salt + claim name + claim value

```
WyJrSEhWOTEtZEFadDhtOUU0Smw0WGJRIiwImdpdmVx25hbWU1LCA1Sm90b1Jld
WyJQak1xcEdXbDRlQjRRcm9EaHFRdzB3IiwgImZhbWlseV9uYw11IiwgIkRvZSJD
WyJ4bmlQNEpadeE5XSUgtTgtFRHQtbY1BIiwgImN0cmV1dF9hZGRyZXNzIiwgIkRvZ
XN0cmV1dCAx11O
WyJLdGZzeHhUbTJtdzBZTFVjS1pVOHRBIiwgImxvY2FsaXR5IiwgIkFueXRvd241XQ
```

**JWT**  
plain-text claims  
+ hashed Disclosures

```
{
  "iat": "https://example.com",
  "type": "id_token",
  "iss": "https://example.com",
  "exp": 1516239024,
  "sub": "1234567890",
  "aud": "https://example.com",
  "scope": "openid",
  "nonce": "1234567890",
  "sd_hash": "1234567890",
  "sd": [
    {
      "salt": "1234567890",
      "claim_name": "sd_claim",
      "claim_value": "sd_claim_value"
    }
  ]
}
```

✓ signed  
by Issuer

**Selected Disclosures**  
salt + claim name + claim value

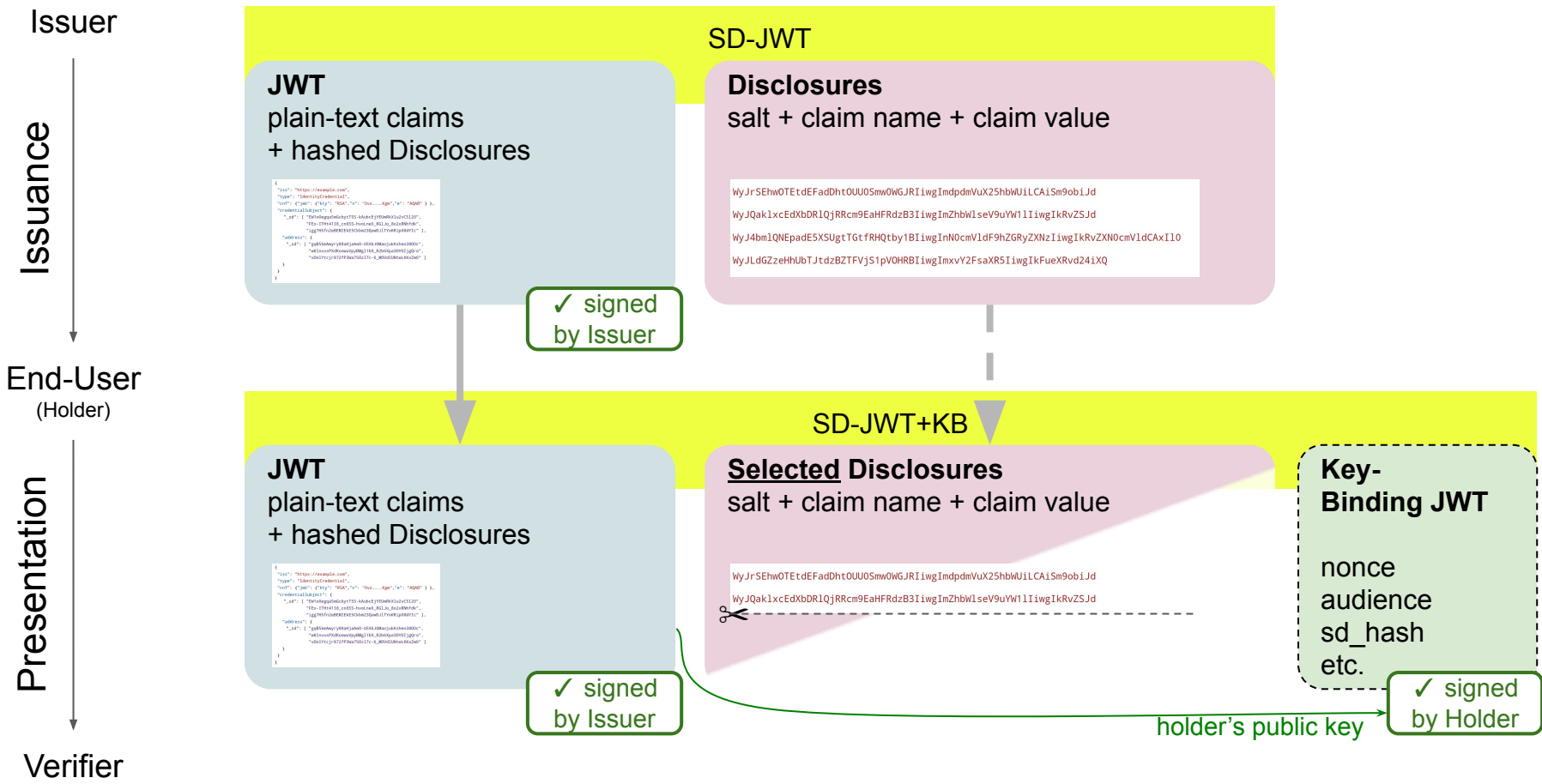
```
WyJrSEhWOTEtZEFadDhtOUU0Smw0WGJRIiwImdpdmVx25hbWU1LCA1Sm90b1Jld
WyJQak1xcEdXbDRlQjRRcm9EaHFRdzB3IiwgImZhbWlseV9uYw11IiwgIkRvZSJD
-----
sd_claim_value
```

**Key-Binding JWT**

nonce  
audience  
sd\_hash  
etc.

✓ signed  
by Holder

holder's public key



# Example Presentation

Issuer-signed SD-JWT~Disclosures~KB-JWT

```
eyJhbGciOiAiRVMyNTYifQ.eyJfc2QiOiBBIkNyUWU3UzVrcUJBSHQtbk1ZWGdjNmJkd
DJTSDVhVfKxc1VfTS1QZ2tqUEkiLCAiSnzZag0c3ZsaUgwUjNqUUVNzVadTZkdDY5d
TVxZWabzdGN0VQWwXTRISiSICJQb3JGYnBLdVZ1Nnh5bUphZ3ZrRnNgWEFiUm9jMkpHb
EFVQTJQTRvN2NJIiwgI1RHZjRvTGJnd2Q1S1FhSH1LV1FaVT1LVZEdFMHc1cnREc3Jae
mZVYw9tTG8iLCAiWFFfM2tQS3QxwH1YN0tBTmtxV1I2eVoyVmE1TnJQsXZQWwJ5TXZSS
0JNTSisICJYekZyendzY002R242Q0pEYzZ2Vks4QmtNbmZHOHZPU0tmcFBjWmRBZmRFI
iwgImdiT3NJNEVkcTJ4Mkt3LXc1d1BFemFrb2I5aFyXy1JEMEFUTjNvUw5Sk0iLCAia
nN1OXlWdWx3UVFsaEZsTV8zSmx6TWfTRnnpnGhRRZBEc6ZheVF3TFVLNCLdLCAiaXNZi
jogImhdHbZoi8vaXNzdWYyLmV4Yw1wbGUuY29tIiwgImldhdCI6IDE2ODMwMDAwMDAsI
CJleHAiOiAxODgzMDAwMDAwLCAic3ViIjogInVzZXJfNDIiLCAibmF0aW9uYwYxpdiG1lc
yI6IjE7Ii4uLiI6ICJwRm5kamtaX1ZDem15VGE2VWpsWm8zZGgta284YU1LUWM5RGxHe
mhhV1lvIn0sIHsiLi4uIjogIjZKa1B1ZHU5M2xjYndIZ2VaOGtoQXYxVTFPU2xlc
lAwVmtCSnJXWjAiFv0sICJfc2RfYXNjogInNoY0s0NTYiLCAiY25mIjogeyJqd2siO
iB7Imt0eSI6ICJFQyIsICJjcnYiOiAiUC0yNTYiLCAieCI6ICJUU0FFUjE5WnZ1M09IR
jRqNfC0dmZTVm9ISVAXsUxpbERsczd2Q2VHZW1jIiwgInkiOiAiwnhqaVdXYlplNUUdIV
ldlV1E0aGJTSWlyc1ZmdWVjQ0U2dDRqVDlGMkhaUSJ9fX0.0eQr inudSFTXNysZ2NuQ
rwwJv-P9gQ-Ce3wEYZkxngaA4GKfP fApdNzBa40dH1urt8tXhW2wQL-I00v8teuw-Wy
JlbHVWNU9m2dTtklJOEVZbnN4QV9BIiwgImZhbWlseV9uYw1IiwgIkRvZsJd-WyJBS
ngtMDk1V1BycFR0tjRRTU9xUk9BIiwgImFKZHLj3m1LCB7InN0cmVlF9hZGRyZXNZi
jogIjEyMyBNYwluIFN0IiwgImxvY2FsaXR5IjogIkFueXRvd24iLCAicmVnaW9uIjogI
kFueXNOYXRlIiwgImNvdW50cnkiOiAiVVMiFv0-WyIyR0xDNDJzS1F2ZUNmR2ZyeU5ST
j13IiwgImdpdmVuX25hbWUiLCAiSm9obiJd-WyJsa2x4RjVqTV1sR1RQVW92TU5JdkNB
IiwgI1VTI10-eyJhbGciOiAiRVMyNTYiLCAidHlwIjogImt0eSI6ICJlc291LCAiY25mIjog
6Ic1XmJmONTY3ODkwIiwgImF1ZCI6ICJodHRwc2ovl3Zlcm1malWYyLmV4Yw1wbGUUb3J
nIiwgImldhdCI6IDE2OTgwNzc30TAsICJfc2RfaGFzaCI6ICZlNHQ4dkNDX2Nfd1ZMbk9
hZEJ0d2g0ZEZ2QkVyU2w5ektPcXdtNm1oVf9VIn0.ZlotfwqF9NUTRAShrd8jGSJEBE
3Z3EKm-AD5udfzggxfK-1QM4TCKbHK81eV088YTKL-Ufm7W5yQpx5wpNpZw
```

## Key-Binding JWT Body:

```
{
  "nonce": "1234567890",
  "aud": "https://verifier.example.org",
  "iat": 1698077790,
  "sd_hash": "34t8vCC_c_vVLnOadBtwh4dFvBERs19zK0qwm6ihT_U"
}
```





# Reconstructing the Original Data

find in data



```
{
  "_sd": [
    "f0BUSQvo46yQ0-wRwXBcGqvnbKIueISEL961_Sjd4do"
  ],
  "iss": "https://issuer.example.com",
  "iat": 1683000000,
  "exp": 1883000000,
  "sub": "6c5c0a49-b589-431d-bae7-219122a9ec2c",
  "_sd_alg": "sha-256",
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "TCAER19Zvu30HF4j4W4vfSVoHIP1ILiDls7vCeGemc",
      "y": "ZxjiWbZMQGHVWkVQ4hbSIirsVfuecCE6t4jT9F2HZQ"
    }
  }
}
```

```
[ "2GLC42sKQveCfGfryNRN9w", "address", { "street_address":
  "Schulstr. 12", "locality": "Schulpforta", "region":
  "Sachsen-Anhalt", "country": "DE"} ]
```

f0BUSQvo46yQ0-wRwXBcGqvnbKIueISEL961\_Sjd4do

```
{
  "nonce": "1234567890",
  "aud": "https://verifier.example.org",
  "iat": 1721206049,
  "sd_hash": "GpLqgSVPUPqeetF7H4jxV3GkPn3BkT1pf3yY2I4G4ew"
}
```

# Reconstructing the Original Data

replace



```
{
  "_sd": [
    "f0BUSQvo46yQ0-wRwXBcGqvnbKIueISEL961_Sjd4do"
  ],
  "iss": "https://issuer.example.com",
  "iat": 1683000000,
  "exp": 1883000000,
  "sub": "6c5c0a49-b589-431d-bae7-219122a9ec2c",
  "_sd_alg": "sha-256",
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "TCAER19Zvu30HF4j4W4vfSVoHIP1ILiLDls7vCeGemc",
      "y": "ZxjiWbZMQGHVwKVQ4hbSIirsVfuecCE6t4jT9F2HZQ"
    }
  }
}
```

```
["2GLC42sKQveCfGfryNRN9w", "address", {"street_address":
  "Schulstr. 12", "locality": "Schulpforta", "region":
  "Sachsen-Anhalt", "country": "DE"}]
```

f0BUSQvo46yQ0-wRwXBcGqvnbKIueISEL961\_Sjd4do

```
{
  "nonce": "1234567890",
  "aud": "https://verifier.example.org",
  "iat": 1721206049,
  "sd_hash": "GpLqgSVPUPqeetF7H4jxV3GkPn3BkT1pf3yY2I4G4ew"
}
```

# Reconstructing the Original Data

```
{
  "address": {
    "street_address": "Schulstr. 12",
    "locality": "Schulpforta",
    "region": "Sachsen-Anhalt",
    "country": "DE"
  },
  "iss": "https://issuer.example.com",
  "iat": 1683000000,
  "exp": 1883000000,
  "sub": "6c5c0a49-b589-431d-bae7-219122a9ec2c",
  "_sd_alg": "sha-256",
  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "TCAER19Zvu30HF4j4W4vfSVoHIP1ILi1D1s7vCeGemc",
      "y": "ZxjiWbZMQGHVwKVQ4hbSIrsVfuecCE6t4jT9F2HZQ"
    }
  }
}
```

... repeat for all disclosures ...

Done!

# Any Element may be Selectively Disclosable

in sub-structures

```
{
  "iss": "https://issuer.example.com",
  "iat": 1683000000,
  "exp": 1883000000,
  "sub": "6c5c0a49-b589-431d-bae7-219122a9ec2c",
  "address": {
    "_sd": [
      "6vh9bq-zS4GKM_7GpggVbYzzu6oOGXrmNVGPHP75Ud0",
      "9gjVuXtdFR0CgRrtNcGUXmF65rdezi_6Er_j76kmYyM",
      "KURDPH4ZC19-3tiz-Df39V8eidy1oV3a3H1Da2N0g88"
    ],
    "country": "DE"
  },
  "_sd_alg": "sha-256"
}
```

array elements

```
{
  "iss": "https://issuer.example.com",
  "iat": 1683000000,
  "exp": 1883000000,
  "sub": "user_42",
  "nationalities": [
    {
      "...": "pFndjkZ_VCzmyTa6Uj1Zo3dh-ko8aIKQc9D1GzhaVYo"
    },
    {
      "...": "7Cf6JkPudry3lcbwHgeZ8khAv1U10S1erP0VkJrWZ0"
    }
  ],
  "_sd_alg": "sha-256",
  "cnf": {...}
}
```

# Recursive Selective Disclosure for Fine-Grained Release

```
{
  "_sd": [
    "HvrKX6fPV0v9K_yCVFbiLFHsMaxcD_114Em6VT8x1lg"
  ],
  "iss": "https://issuer.example.com",
  "iat": 1683000000,
  "exp": 1883000000,
  "sub": "6c5c0a49-b589-431d-bae7-219122a9ec2c",
  "_sd_alg": "sha-256",
  "cnf": {
    "jwk": {
      "kty": "EC",

```

```
WyJRZ19PNjR6cUF4ZTQxMmExMDhpcm9BIiwgImFkZHZJlc3MiLCB7I19zZCI6
IFsiNnZoOWJxLXpTNEdLTV83R3BnZ1ZiWxp6dTzVt0dYcm10VkdQsFA3NVVk
MCIsICl5Z2pWdVh0ZEZST0NnUnJ0TmNHVhtRjY1cmR1em1fNkVyX2o3NmTt
WX1NIiwgIktVUKRQaDRaQzE5LTVnOaXotRGYzOVY4ZWlkeTFvVjNhM0gxRGEy
TjBnODgiLCAiV045cjlkQ0JK0EhUQ3NTMmpLQVN4VGpFeVc1bTV4NjVfWl8y
cm8yamZYTSJdfV0
```

decode

```
[
  "Qg_064zqAxe412a108iRoA",
  "address",
]
{
  "_sd": [
    "6vh9bq-zS4GKM_7GpggVbYzZu6oOGXrmNVGPH75Ud0",
    "9gjVuXtdFROCGRrtNcGUXmF65rdezi_6Er_j76kmYyM",
    "KURDPH4ZC19-3tiz-Df39V8eidy1oV3a3H1Da2N0g88",
    "WN9r9dCBJ8HTCsS2jKASxTjEyW5m5x65_Z_2ro2jfXM"
  ]
}
]
```

```
WyJ1SThaV205UW5LUHBOUGVOZW5IZGhRIiwgImNvdW50cnkiLCAiREUiX
```

decode

```
[
  "eI8Zwm9QnKPPnPeNenHdhQ",
  "country",
  "DE"
]
```

# Security Considerations (I)

**Signature verification: Verifiers could verify the signature inadequately/partially and accept tampered credentials**

Mitigating measures:

- Simple processing model, specified in detail in the standard
- Established algorithms enable the use of existing implementations

**Manipulation of disclosures: If the hashes of the disclosures are not checked by the verifier, manipulated plaintext values could be accepted.**

Mitigating measures:

- Design: Generally no assignment to the document possible without hash calculation
- Processing model specified in detail

# Security Considerations (II)

**Missing check of key binding: Verifiers could accept credentials without key binding**

Mitigating measures:

- Different formats with/without key binding
- Differentiation in terminology
- Detailed discussion in the standard



# Privacy Considerations

**Unlinkability (“unlinkability”): Several presentations of the same credential can be traced back to the same person (due to the same hash values).**

Mitigating measures:

- Single use: Credentials are always issued in groups - same data, different salt values. Each individual credential is then only used once.

SD-JWT VC

# SD-JWT VC

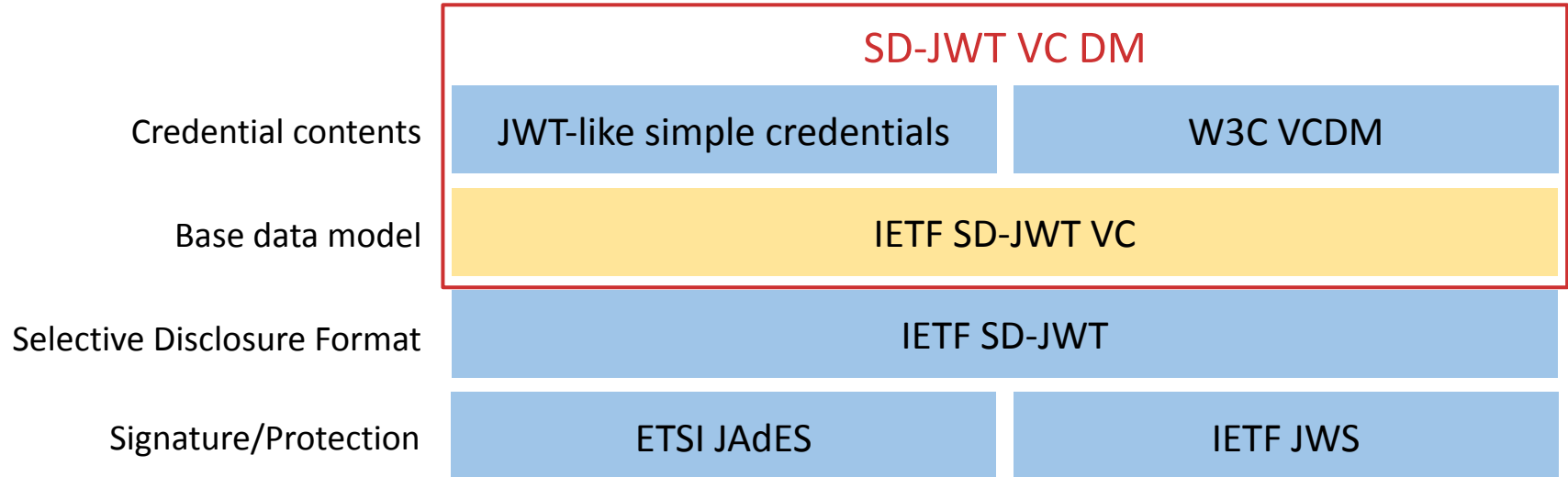
## **Credentials based on SD-JWT VC** using an extensible data model



IETF Draft: <https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/>

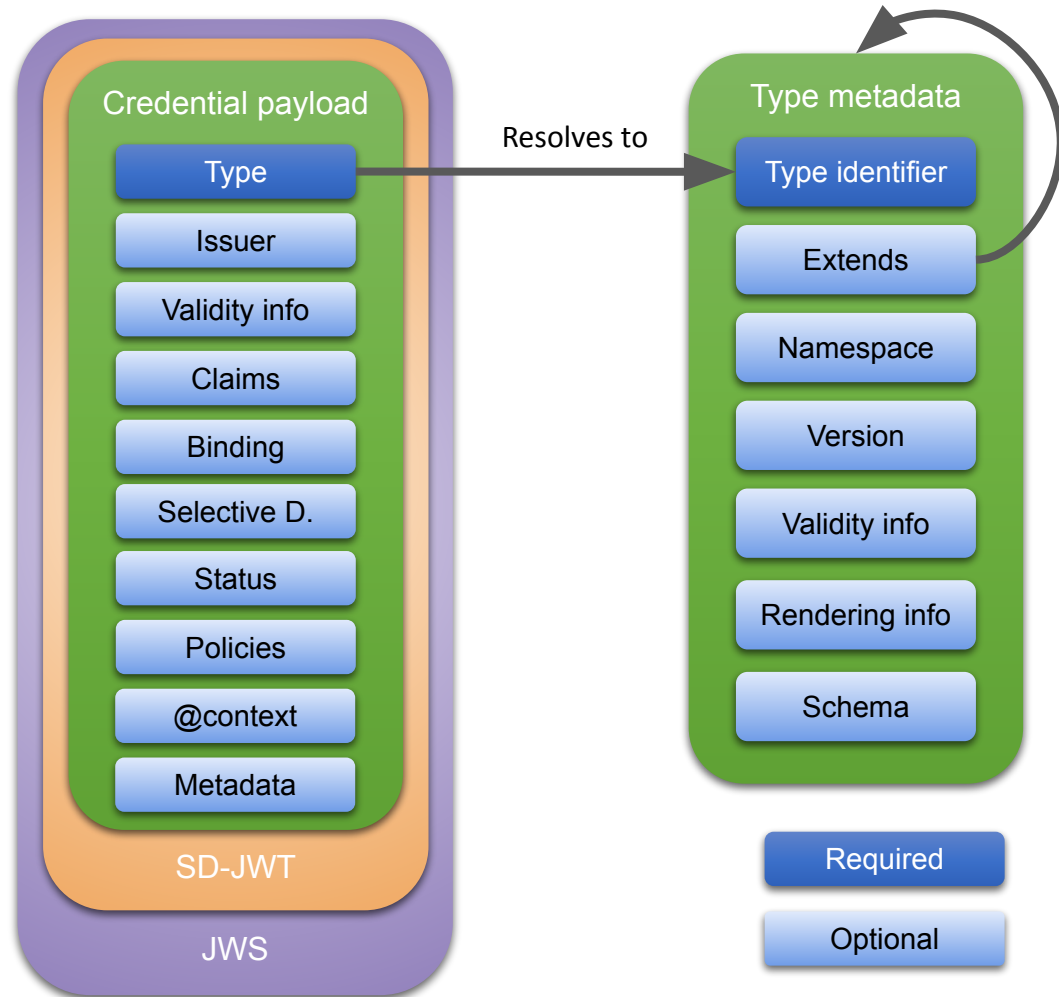
Daniel Fett  
Oliver Terbu  
Brian Campbell

# SD-JWT VC DM



# Overview

- The **core data model** consists of a set of required and optional claims
- The **type identifier** resolves to **type metadata** that contains additional information about the credential
- The data model allows to express simple and complex information sets



# Defined Claims

- **iss** — The Issuer of the Verifiable Credential. The value of iss MUST be a URI.
- **nbf** — The time before which the Verifiable Credential MUST NOT be accepted before validating.
- **exp** — The expiry time of the Verifiable Credential after which the Verifiable Credential is no longer valid.
- **cnf** — Contains the confirmation method identifying the proof of possession key. For proof of cryptographic Key Binding, the Key Binding JWT in the presentation of the SD-JWT MUST be signed by the key identified in this claim.
- **vct** — The type of the Verifiable Credential, e.g., [https://credentials.example.com/identity\\_credential](https://credentials.example.com/identity_credential).
- **status** — The information on how to read the status of the Verifiable Credential.
- **sub** — The identifier of the Subject of the Verifiable Credential. The Issuer MAY use it to provide the Subject identifier known by the Issuer. There is no requirement for a binding to exist between sub and cnf claims.
- **iat** — The time of issuance of the Verifiable Credential. See [RFC7519] for more information.

# Example: Simplified PID

The data model  
represents a simplified  
PID without selective  
disclosure

```
{
  "vct": "eudi:example:pid",

  "given_name": "Jack",
  "family_name": "Dougherty",
  "birthdate": "1980-05-23",

  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "52aDI_ur05n1f_p3jiYGUU82oKZr3m4LsAErM536crQ",
      "y": "ckhZ-KQ5aXNL91R8Eufg1a0f8Z5pZJnIvuCzNGfdnzo"
    }
  }
}
```

# Example: Simplified PID

Same as before, with selective disclosure.

After processing, data structure as shown on previous slide is restored.

```
{
  "vct": "eudi:example:pid",
  "_sd_alg": "sha-256" ,

  "_sd": [
    "09vKrJM0lyTWM0sjpu_pd0BVBQ2M1y3KhpH515nXkpY" ,
    "2rsjGbaC0ky8mT0pJrPioWTq0_daw1sX76poUlgCwbI" ,
    "Ek08dhW0dHEJbvUH1E_VCeuC9uRELOieLZhh7XbUTtA"
  ],

  "cnf": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "52aDI_ur05n1f_p3jiYGUU82oKZr3m4LsAErM536crQ",
      "y": "ckhZ-KQ5aXNL91R8Eufg1aOf8Z5pZJnIvuCzNGfdnzo"
    }
  }
}
```

(All examples shortened for presentation.)



# German EUDI Wallet PID Proposal (1/3)

```
{
  // Base data (SD-JWT VC DM)
  "vct": "https://example.bmi.bund.de/credential/pid/1.0",
  // metadata would define this as an extension of the base type, e.g.,
  // https://example.eudi.eu/credential/pid/1.0

  "vct#integrity": "sha256-jo8433ot48....utul8ura33",

  // Base dataset that always needs to be present

  "given_name": "Erika",
  "family_name": "Mustermann",
  "birthdate": "1963-08-12",

  // Additional data

  "source_document_type": "id_card",

  "address": {
    "street_address": "Heidestraße 17",
    "locality": "Köln",
    "postal_code": "51147",
    "country": "DE"
  },
}
```

# German EUDI Wallet PID Proposal (2/3)

```
"nationalities": [
  "DE"
],

"gender": "female",
"birth_family_name": "Gabler",

"place_of_birth": {
  "locality": "Berlin",
  "country": "DE"
},

"also_known_as": "Schwester Agnes",

// Derived claims
"age_equal_or_over": {
  "12": true,
  "14": true,
  "16": true,
  "18": true,
  "21": true,
  "65": false
},
```

# German EUDI Wallet PID Proposal (3/3)

```
// key binding (SD-JWT VC DM)
"cnf": {
  "jwk": {
    "kty": "EC",
    "crv": "P-256",
    "x": "52aDI_ur05n1f_p3jiYGUU82oKZr3m4LsAErM536crQ",
    "y": "ckhZ-KQ5aXNL91R8Eufg1a0f8Z5pZJnIvuCzNGfdnzo"
  }
},

"iat": 1712231700,
"exp": 1806839700,
"issuing_authority": "DE",
"issuing_country": "DE"
}
```