EUDI Verifiable Credentials

Format Alignment Proposal

Daniel Fett, Alen Horvat, Oliver Terbu

2024-02-15

Agenda

- Problem statement
- Overview and Examples
- Selective disclosure and key binding
- Roadmap

Problem Statement

Existing Formats

Several established credential formats exist for JSON-based credentials, in particular

- W3C Verifiable Credentials Data Model v1.1/v2
- SD-JWT VC based on JWT/JWS

Note: ISO mdoc is out of scope.

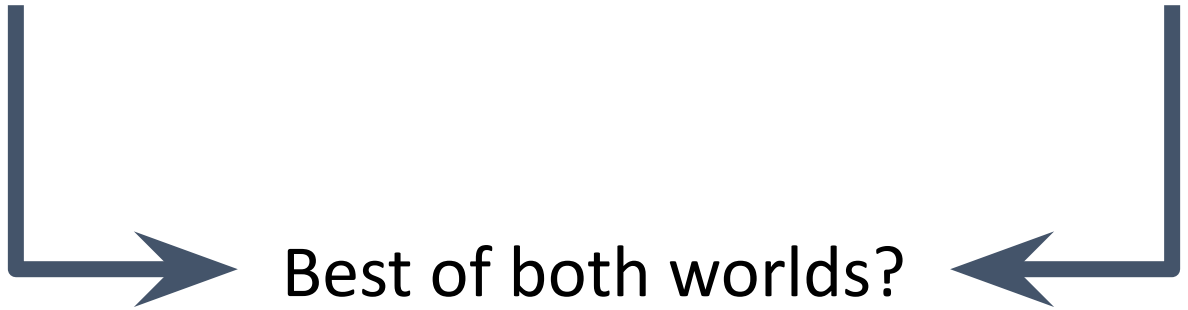
Both are not ideal

W3C VCDM drawbacks:

- Lacks selective disclosure
- JSON/JSON-LD processing ambiguity
- Complexity for simple credentials

SD-JWT VC drawbacks:

- No schemas or vocabularies
- Not immediately AdES compatible



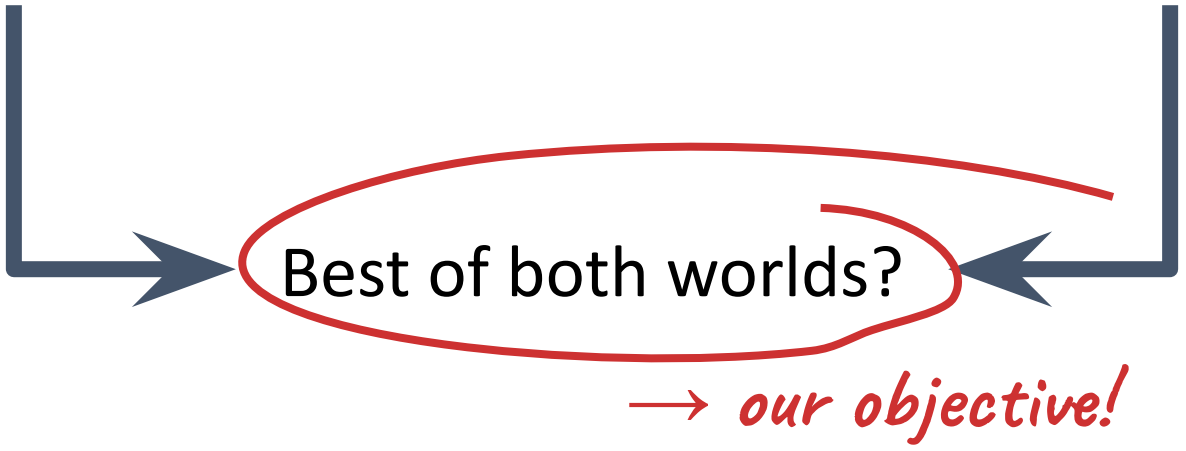
Both are not ideal

W3C VCDM drawbacks:

- Lacks selective disclosure
- JSON/JSON-LD processing ambiguity
- Complexity for simple credentials

SD-JWT VC drawbacks:

- No schemas or vocabularies
- Not immediately AdES compatible



Objective

Define a format for creating and securing JSON-based PIDs and (Q)EAAs based on Verifiable Credentials taking into consideration the existing data models, formats, and securing mechanisms.

The proposal covers

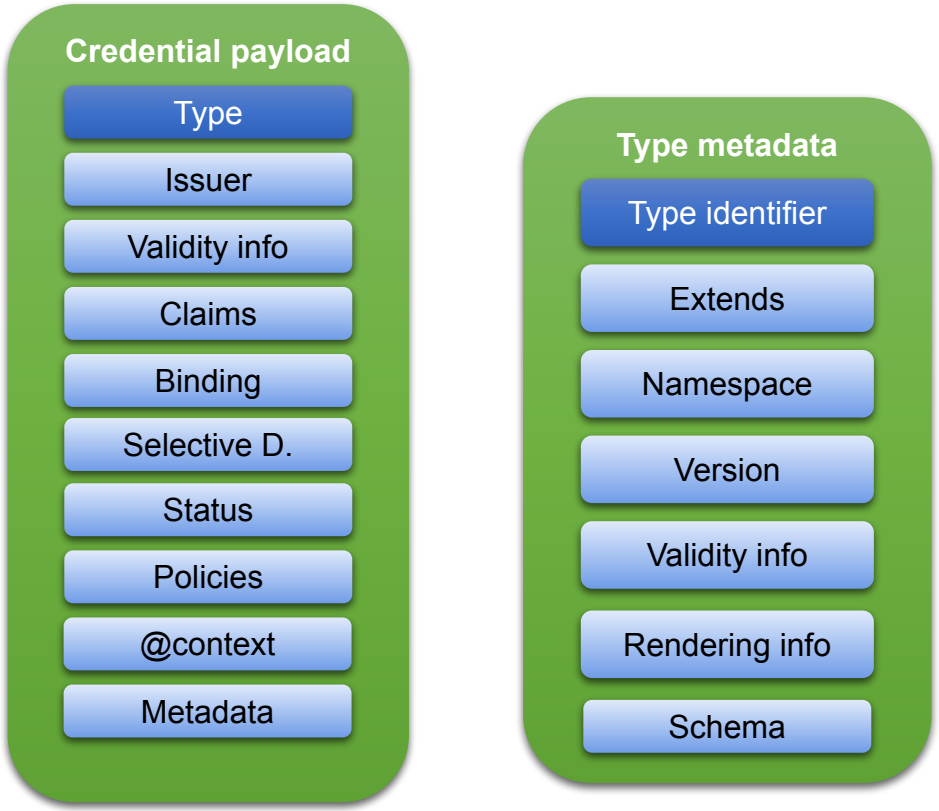
- Data model
- Data format
- Securing mechanisms
- Signature format

SD-JWT VC DM

Feature	SD-JWT VC	VCDM	SD-JWT VC DM
(Q)EAAs with nested data structures and arrays	🟩	🟩	🟩
Simple credentials	🟩	❌	🟩
Schemas and Vocabularies	❌	🟩	🟩
Selective Disclosure	🟩	❌	🟩
Signing Algorithms (ETSI/SOG-IS)	🟩	🟩	🟩
Key Binding Approaches (cryptographic, non-cryptographic)	🟩	🟩	🟩
Short, Medium, and Long-Lived Credentials	🟩	🟩	🟩
Different Identifiers (x509-based, cnf, DIDs)	🟩	🟩	🟩
Online and Offline Exchange of Credentials	🟩	🟩	🟩
Revocation/Suspension	🟩	🟩	🟩
Policies	🟩	🟩	🟩

Data model and format - best of both worlds!

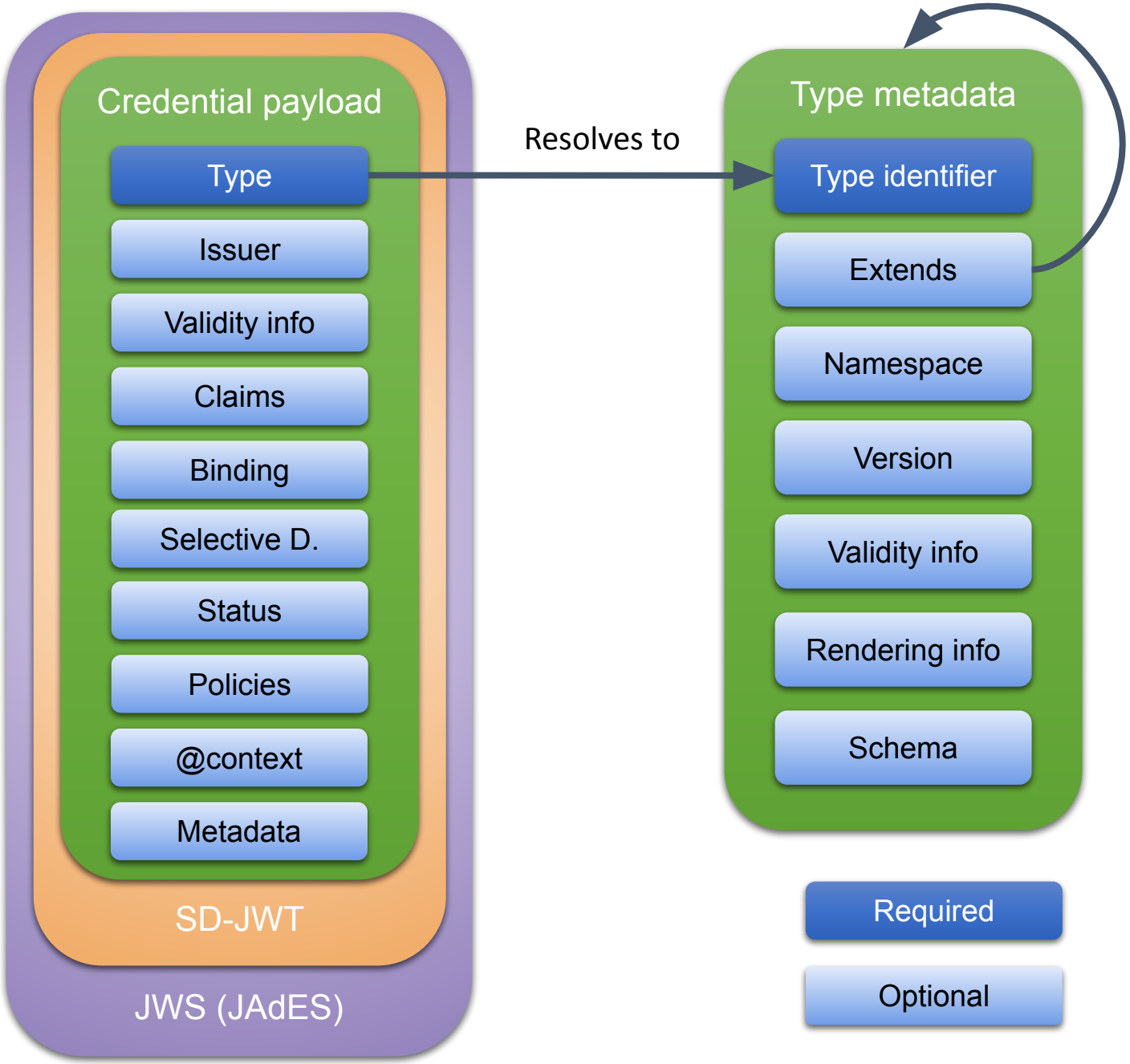
- SD-JWT VC(1) with Type Metadata
- Base format: JSON
- Supports open-world data modelling
- Compatible to W3C VCDM v2
- JSON-LD support



(1) With minor updates

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



Example: Simplified PID

The data model represents a simplified PID without selective disclosure

Exact claim names, definitions and the PID signature profile are out of scope.

(All examples shortened for presentation.)


```
{
 "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-KQ5aXNL91R8Eufg1aOf8Z5pZJnIvuCzNGfdnzo"
   }
 }
}
```

Example: Simplified PID

Same as before, with selective disclosure.

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

(All examples shortened for presentation.)

Example: PDA-1

Simplified Portable Document A1.


```
{
 "vct": "empl:pda1",

 "valid_from": "2022-11-10T19:19:47.287Z",
 "valid_until": "2022-11-10T19:19:47.287Z",

 "id": "635ba519cd19764e84ea67dd",
 "legal_entity_verifiable_id": {
   "legal_name": "Ministry of Wonderland"
 },
 "claims": {
   "personal_information": {
     "personal_identification_number": "1",
     "sex": "01",
     "surname": "Dalton",
     "forenames": "Joe Jack William Averell",
     "date_birth": "1985-08-15",
     "nationalities": [
       "BE"
     ],
     "state_of_residence_address": {
       "street_no": "sss, nnn ",
       "post_code": "ppp",
       "town": "ccc",
       "country_code": "BE"
     }
   }
 },

 "cnf": {
   "jwk": { … }
 }
}
```


(All examples shortened for presentation.)

Example: PDA-1 Metadata

As resolved from `"vct": "empl:pda1"` type identifier

(All examples shortened for presentation.)


```
{
 "language": "en-gb",
 "namespace": "empl",

 "vct": "empl:pda1",
 "extends": "iana:sd-jwt-vc",
 "extends#integrity": "sha256-786b8dfc26a9b...1854dd2",

 "version": "1.0",
 "name": "Portable Document A1",
 "description": "Example metadata for PDA1",

 "schema": {
   "json_schema": {
     "uri": "https://empl.eu/credential-schema-1.0",
     "uri#integrity": "sha256-742289d058bc...5aef1620ac02",
   }
 },
 "display": [
   {
     "en-GB": {
       "name": "Portable Document A1",
       "rendering": {
         "simple": {
           "logo": {
             "uri": "https://empl.eu/pda1/logo.png",
             "uri#integrity": "sha256-e737d7...da26762acb",
             "alt_text": "a square logo of a university"
           },
           "background_color": "#12107c",
           "text_color": "#FFFFFF"
         }
       }
     }
   }
 ]
}
```

Example: ELM (1/2)

Data structure


```
{
 "@context": [
   "https://www.w3.org/2018/credentials/v1",
   "http://data.europa.eu/snb/model/context/edc-ap"
 ],
 "vct": "empl:europeanDigitalCredential",

 "id": "http://example.org/credential132",
 "authentic_source": { ... },
 "credentialProfiles": {
   "id": "http://data.europa.eu/snb/credential/bdc47cb449",
   ...
 },
 "displayParameter": { ... },

 "evidence": {
   "elm:evidence": {
     "id": "http://example.org/evidence123",
     "dcType": {
       "id": "http://data.europa.eu/snb/evidence-type/c_18016257",
       "type": "Concept",
       "inScheme": {
         "id": "http://data.europa.eu/snb/evidence-type/25831c2",
         "type": "ConceptScheme"
       }
     }
   }
 },
 "terms_of_use": {
   "elm:terms_of_use": {
     "id": "http://example.org/termsOfUse1",
     "type": "TermsOfUse"
   }
 },
 "status": {
   "elm:credential_status": {
     "id": "http://example.org/credentialStatus1",
     "type": "CredentialStatus"
   }
 },
```


(All examples shortened for presentation.)

continued on next slide …

Example: ELM (2/2)

Data structure

… continued from previous slide

```
"claims": {
  "id": "http://example.org/pid1",
  "type": "Person",
  "birthName": { "en": "Maxi" },
  "familyName": { "en": "Power" },
  "fullName": { "en": "Max Power" },
  "givenName": { "en": "Max" },
  "hasClaim": {
    "id": "http://example.org/cl1",
    "type": "LearningAchievement",
    "awardedBy": {
      "id": "http://example.org/awardingProcess1",
      "type": "AwardingProcess",
      "awardingBody": {
        "id": "http://example.org/org1",
        "type": "Organisation",
        "legalName": { "en": "some legal name of the organisation" },
        "location": {
          "id": "http://example.org/loc2",
          "type": "Location",
          "address": { … }
        }
      },
      "educationalSystemNote": {
        "id": "http://example.org/someEducationalSystem",
        "type": "Concept",
        "definition": {
          "en": "the definition of the the concept for the educational system"
        }
      }
    },
    "title": {
      "en": "some kind of learning achievement",
      "fr": "une sorte de réussite scolaire"
    }
  }
 }
}
```

(All examples shortened for presentation.)

Selective Disclosure and Key Binding

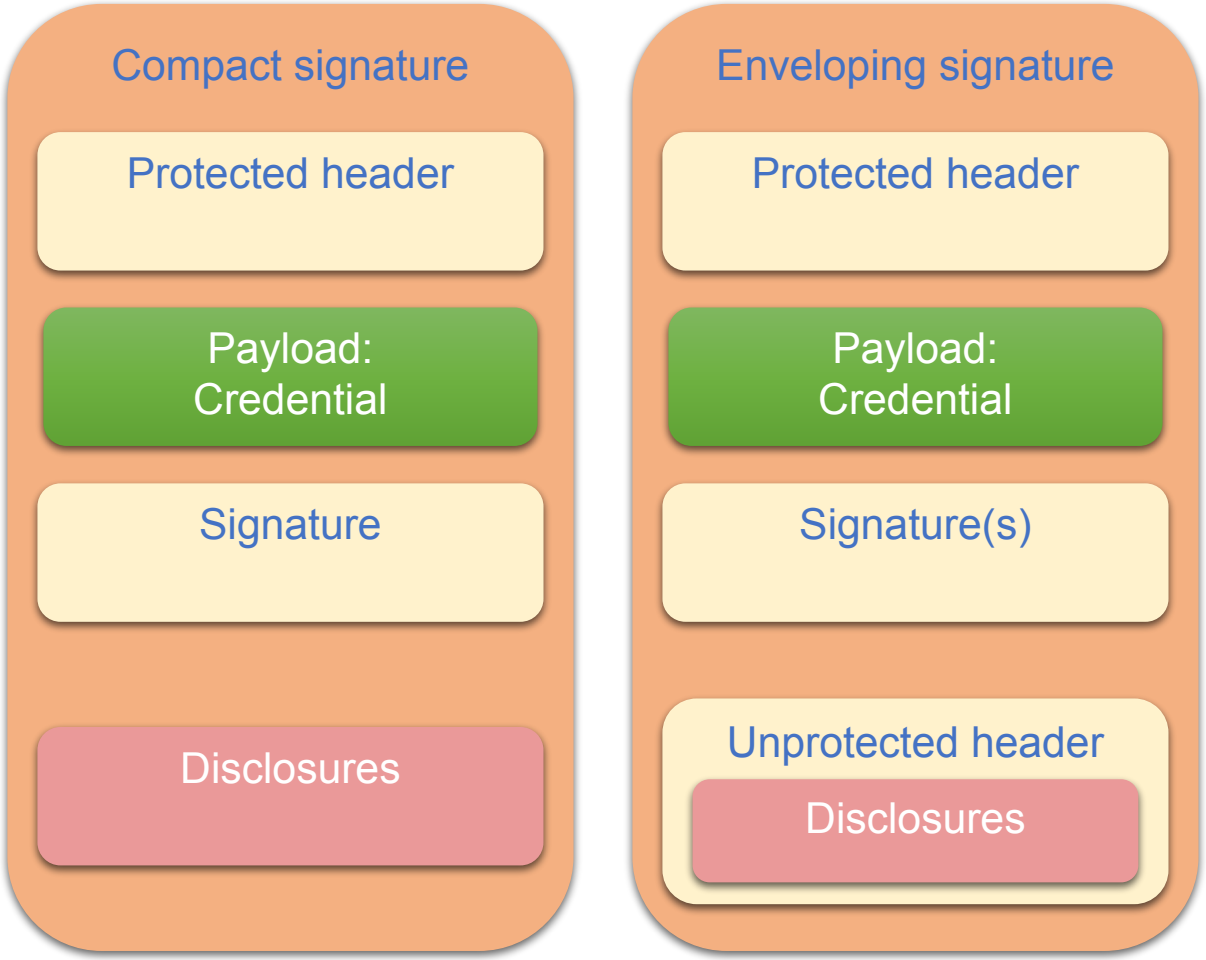
Credential issuance and presentation options

- The format and data model support a variety of credential issuance and presentation options as summarised in the table
- The proposed signature format **SD-JWT** supports
  - Selective disclosure
  - Key binding
  - Signature profile definitions
- SD-JWT signature format extends the well-established JWS signature format

Key Binding	Selective Disclosure
❌	❌
✅	❌
❌	✅
✅	✅

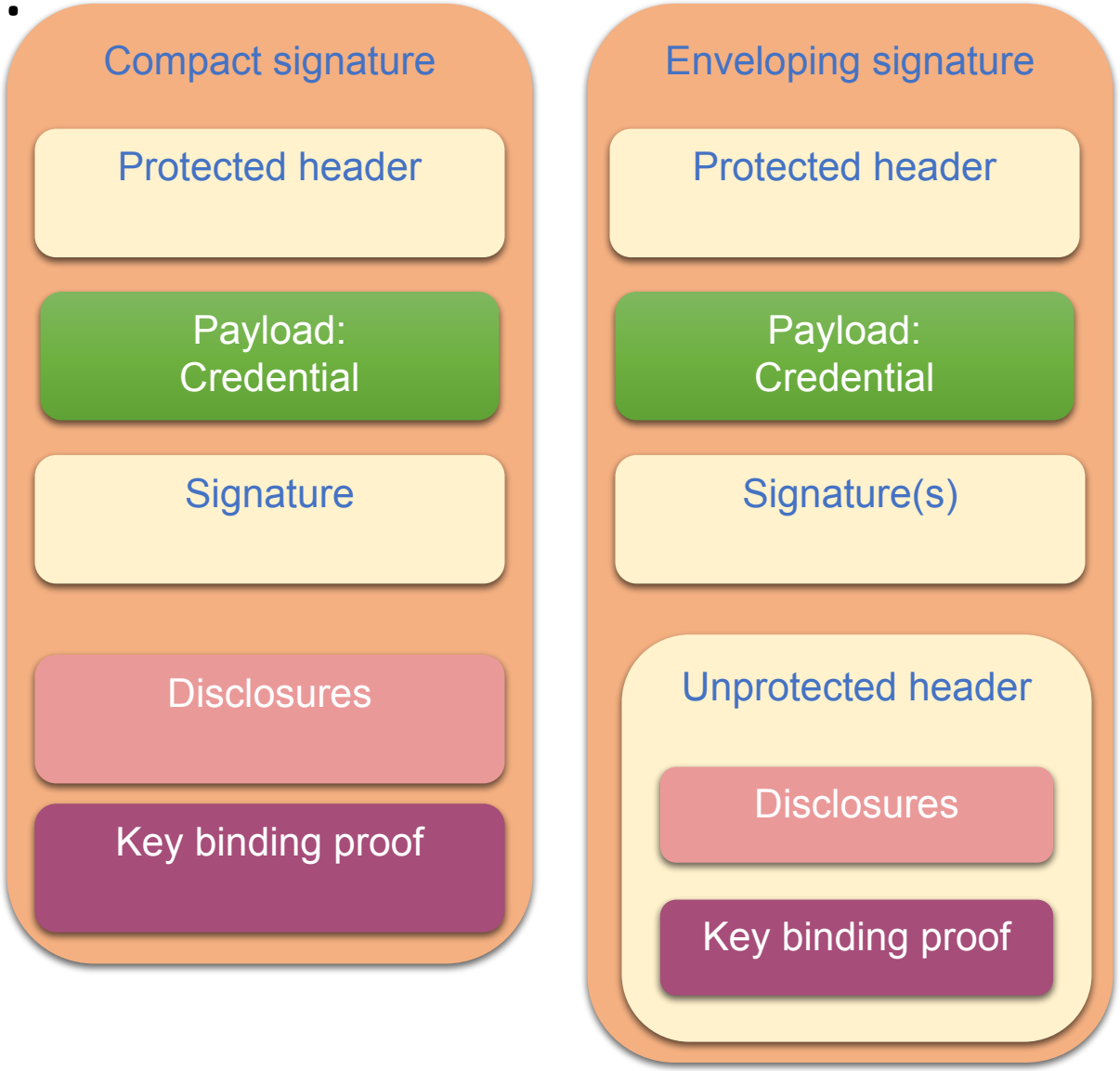
Keep simple things simple, make complex things possible.

- Compact SD-JWT signature format for simple credentials
- Enveloping (JSON serialised) signature format for rich signatures (self-contained credentials, multiple signatures, re-signing)
- Variety of key-binding key representations (raw, identifier, DID, …)
- Supports all ETSI/SOG-IS signing algorithms
- Supports JAdES*



Keep simple things simple, make complex things possible.

- Compact SD-JWT signature format for simple credentials
- Enveloping (JSON serialised) signature format for rich signatures (self-contained credentials, multiple signatures, re-signing)
- Variety of key-binding key representations (raw, identifier, DID, …)
- Supports all ETSI/SOG-IS signing algorithms
- Supports JAdES*



How to get there?

Roadmap

- ETSI JAdES
  - Update JAdES profiles
- IETF SD-JWT
  - Minor updates for JAdES alignment
- IETF SD-JWT VC
  - Integrate metadata schema specification, including how to handle namespaces

Thank you!