

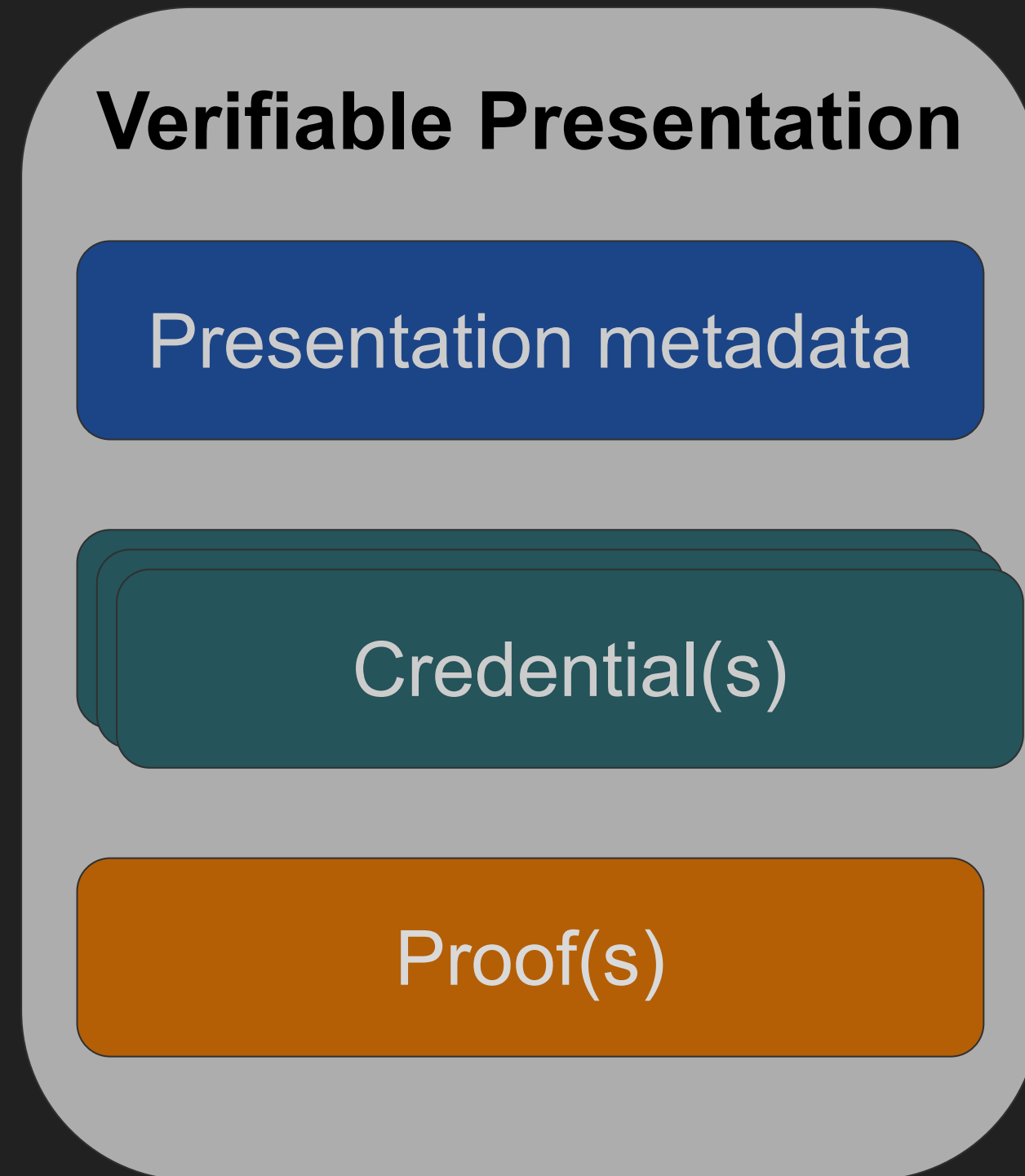
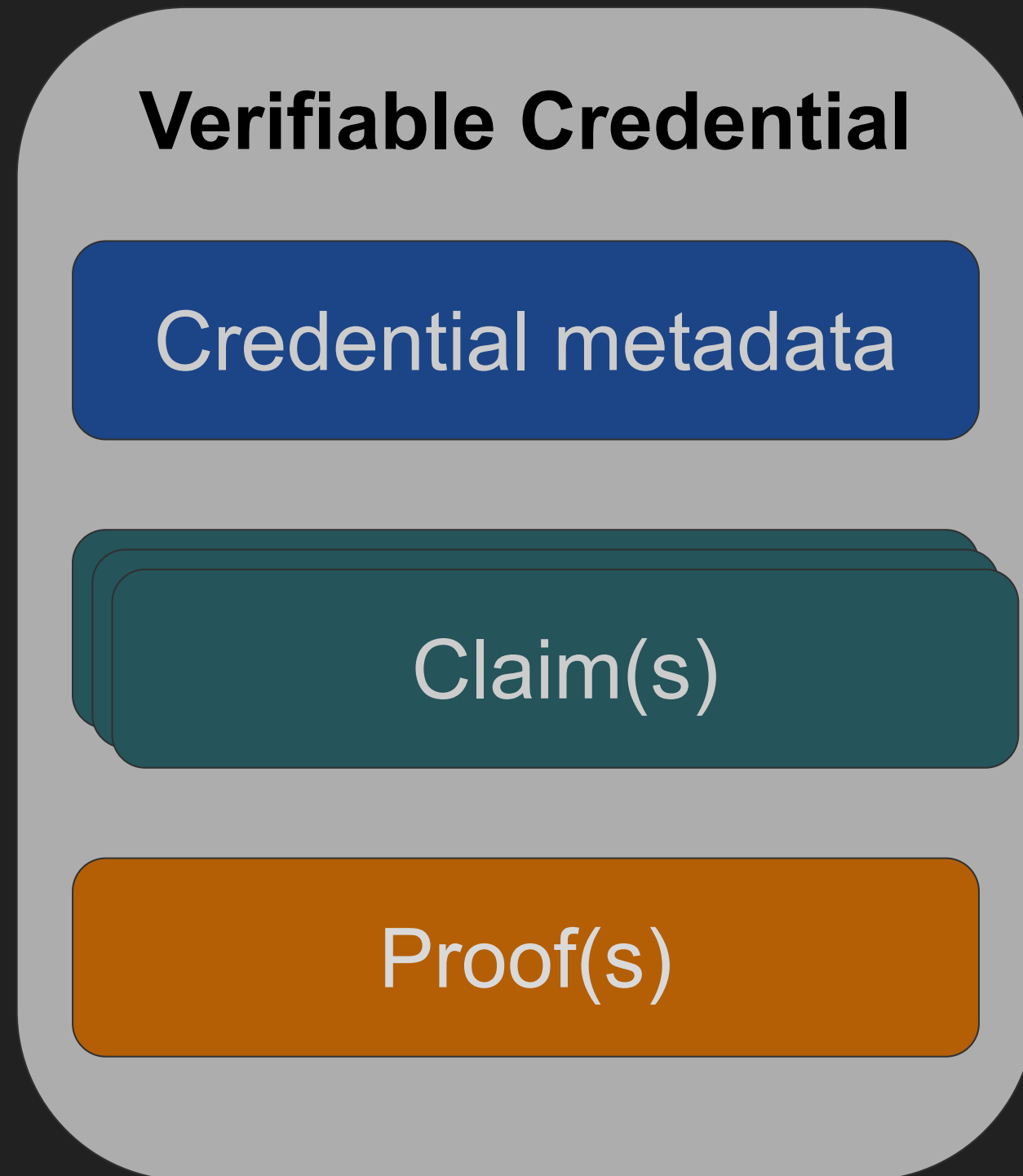
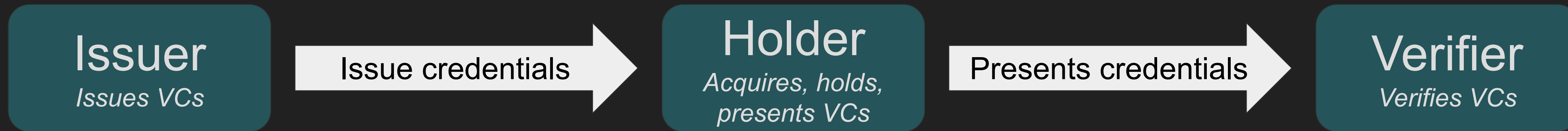
W3C Verifiable Credentials

Brent Zundel (Yubico)

AC Meeting 2026 - DevMeetup



The basic terminology



Verifiable presentations are essential!

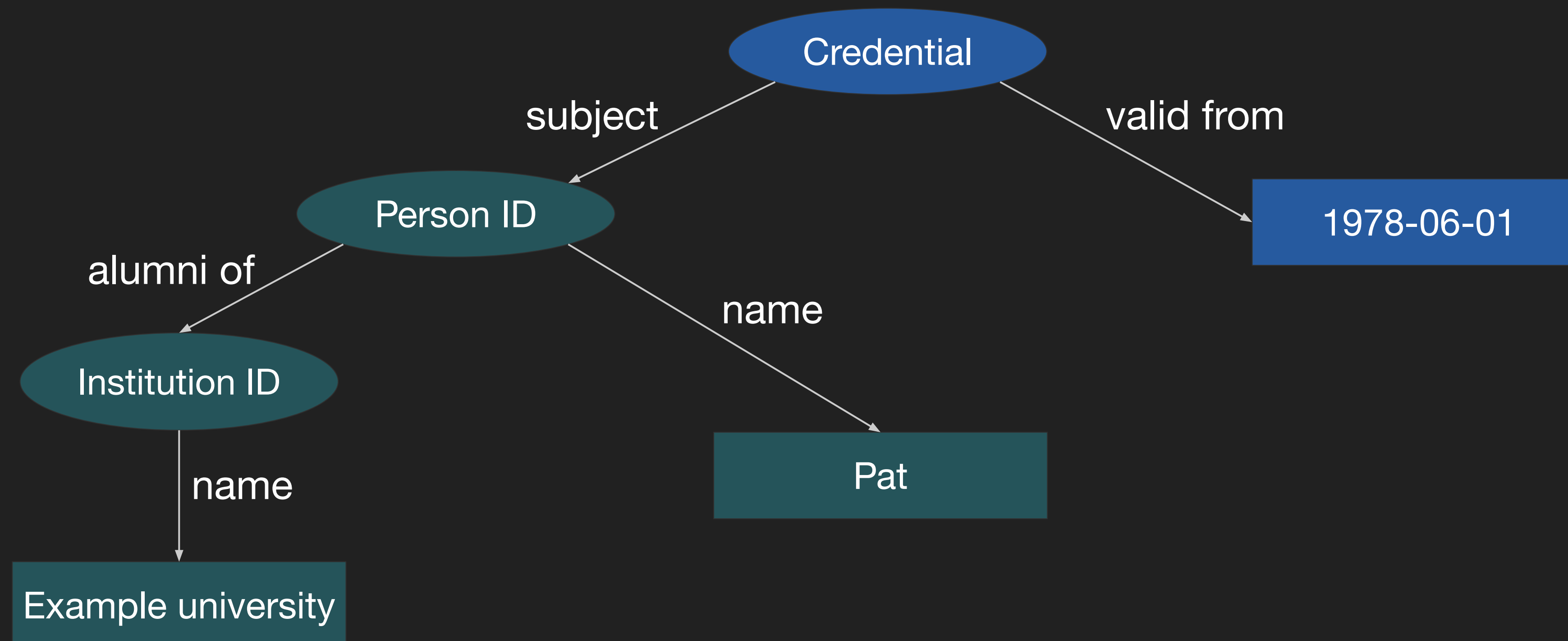
- The holder may present:
 - a modified credential (e.g., for selective disclosure)
 - a combination of several credentials on the same subject (e.g. diploma & work permit)
 - a series of credentials for different subjects (e.g., travel documents for a family)

- Proofs on the credentials and on the presentation are different:
 - the credential proofs are provided by, or are derived from, the issuer
 - the presentation proofs are provided by the holder
 - both are required for a presentation to ensure data integrity



The VC model: collection of claims

- Is a simple model of individual claims (a.k.a. statements):
 - subject → predicate → object
- By combining such statements, we get a graph
- In VC, the graph is serialized in JSON



Advantages of a graph based model

- It is conceptually easy to *combine* graphs: just add new connections binding them together
 - e.g., combining standard terms with application specific ones
- Thinking in terms of atomic claims is needed for selective disclosure schemes, and this model facilitates this

In VC, the graph is serialized in JSON

- More exactly, a dialect of JSON called JSON-LD, specialized in representing of such graphs
 - the standard processing steps in the recommendations are expressed in terms of JSON
 - can be implemented with standard JSON tools
- There is work going on to define specialized CBOR conversions to ensure efficient storage and transmission

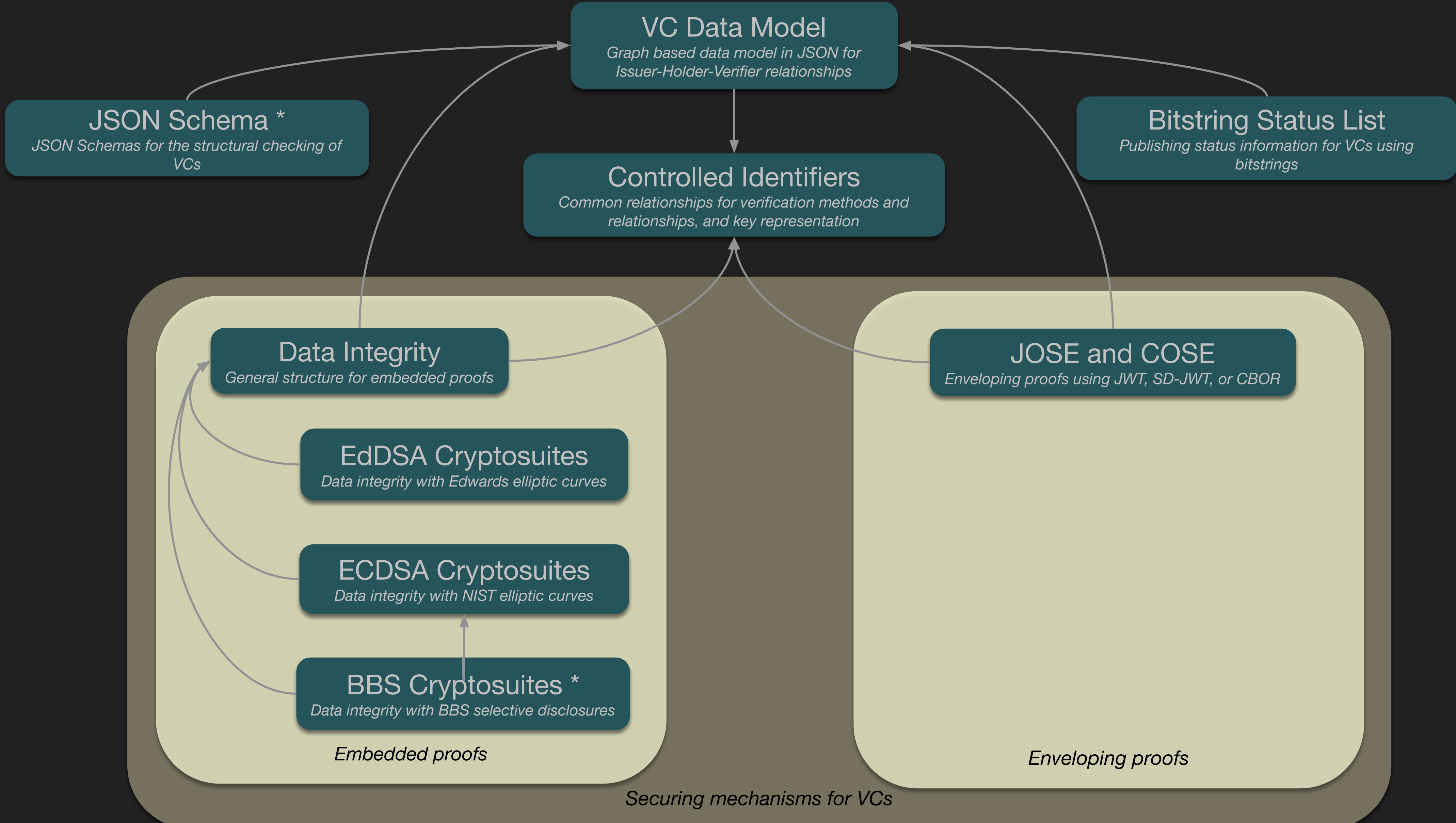
Advantages of a Linked Data graph based model

- Identifiers and attributes are expressed as URLs
 - can be HTTPS, UUID, DID, DOI...
 - makes it possible to reuse existing vocabularies / ontologies
 - developed at W3C (e.g. DPV), ETSI (e.g. SAREF), etc.
 - and to combine / integrate vocabularies developed independently
 - very important when combining graphs (e.g. university degree & work permit)
 - or when reusing VCs outside their original context (e.g., **cross-border**)

- Interoperable with other initiatives
 - Dataspaces (IDSA, Gaia-X)
 - Digital Product Passport (GS1, CIRPASS)
 - Industry 4.0 (RAMI 4.0, WoT, SAREF)



W3C VC Specifications



W3C VC Specifications

- Published as W3C Recommendations on 15 May 2025
 - Except for
 - Verifiable Credentials JSON Schema Specification
(lack of implementation feedback so far)
 - Data Integrity BBS Cryptosuites v1.0
(pending the official publication of BBS by IETF)
- Overview document:
<https://www.w3.org/TR/vc-overview/>



Future of the VC Working Group?

- Continuing to maintain published specifications
- Also developing the following:
 - VC Render Method v1.0
 - VC Confidence v1.0
 - VC API for Lifecycle Management v1.0
 - VC Barcodes v1.0
 - Verifiable Issuer and Verifiers v1.0



Future of the VC Working Group?

VCWG may also develop the following:

- Vocabularies for Digital Product Passports
- Vocabularies for Business Wallets
- VC over Wireless v1.0
- VC Refresh v1.0
- Data Integrity Quantum Safe Cryptosuites v1.0
- Data Integrity SM2 Cryptosuites v1.0



Appendix:
some more details
with an example

An example of a basic credential (without proof)

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.example.org/vocabs/alumni"
  ],
  "id": "https://uni.example/Credential12",
  "type": ["VerifiableCredential", "ExampleAlumniCredential"],
  "issuer": "did:example:2g55q91",
  "validFrom": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "https://www.example.org/persons/pat",
    "name": "Pat",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": "Example University"
    }
  }
},
...
```

An example of a basic credential (without proof)

Identification of

- the terminologies
- the credential itself
- the type of the credential

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.example.org/vocabs/alumni"
  ],
  "id": "https://uni.example/Credential12",
  "type": ["VerifiableCredential", "ExampleAlumniCredential"],
  "issuer": "did:example:2g55q91",
  "validFrom": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "https://www.example.org/persons/pat",
    "name": "Pat",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": "Example University"
    }
  }
},
...
```

An example of a basic credential (without proof)

Credential metadata

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.example.org/vocabs/alumni"
  ],
  "id": "https://uni.example/Credential12",
  "type": ["VerifiableCredential", "ExampleAlumniCredential"],
  "issuer": "did:example:2g55q91",
  "validFrom": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "https://www.example.org/persons/pat",
    "name": "Pat",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": "Example University"
    }
  }
},
...
```



An example of a basic credential (without proof)

Credential claims

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.example.org/vocabs/alumni"
  ],
  "id": "https://uni.example/Credential12",
  "type": ["VerifiableCredential", "ExampleAlumniCredential"],
  "issuer": "did:example:2g55q91",
  "validFrom": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "https://www.example.org/persons/pat",
    "name": "Pat",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": "Example University"
    }
  }
},
...
```



Securing mechanisms



Enveloping proofs: JOSE/COSE

- The JSON-LD representation is fed into a JWT pipeline
 - reusing the JOSE toolkit, registered signature mechanisms, etc.
- Using COSE (i.e., CBOR) means a very small footprint for credentials
- Using IETF's SD-JWT provides selective disclosure
- Automatically includes updates to JOSE/COSE such as PQ algorithms



Enveloping proofs: JOSE/COSE

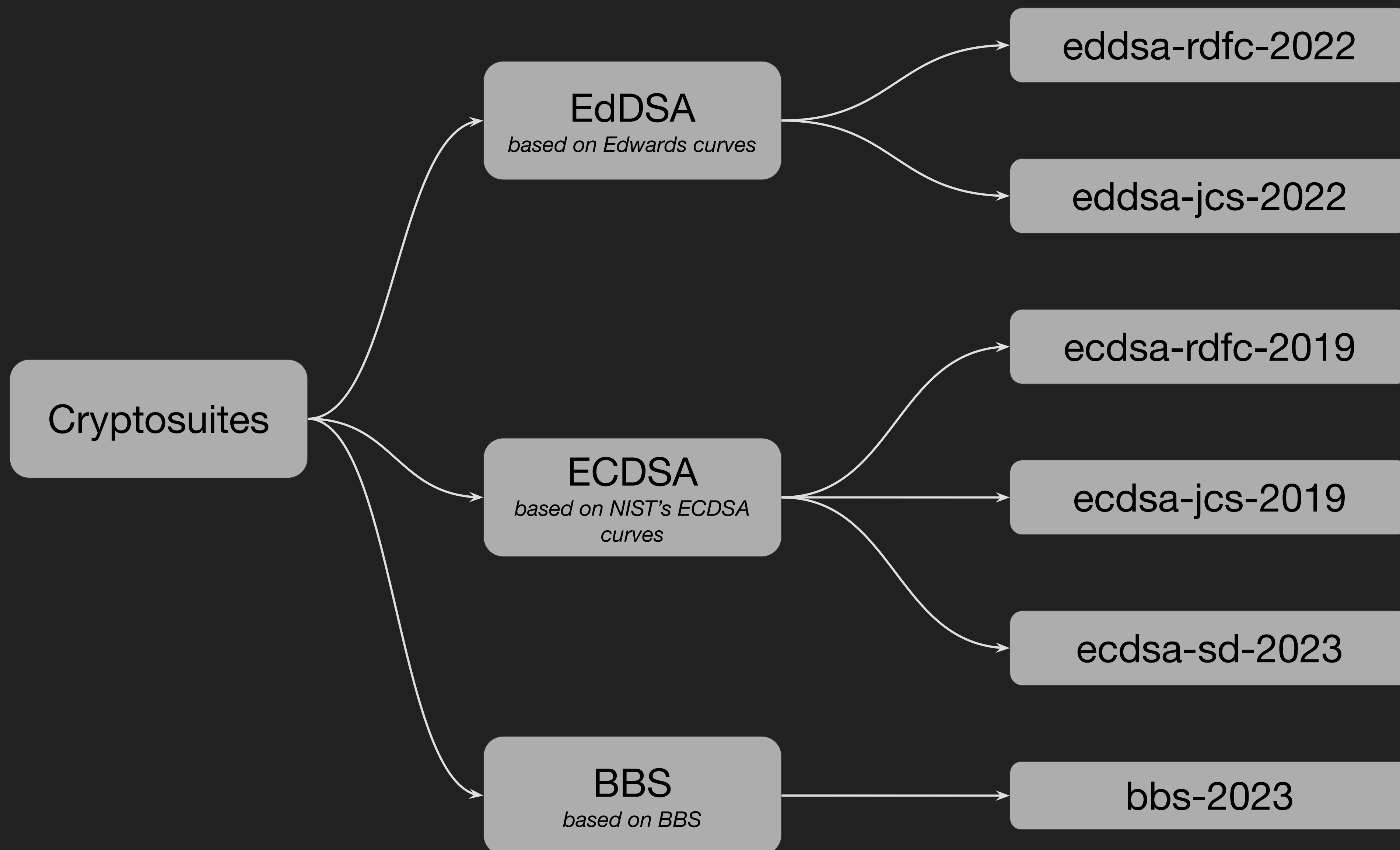
- Using JWT/CWT means reusing off-the shelf tools
 - means easier deployment
- Relies on the IANA registry for cryptographic schemes that can be used
- Enveloping Verifiable *Presentations* allow holders to use JWT/CWT tools to sign
 - remember that a VP contains separate issuer and holder proofs...



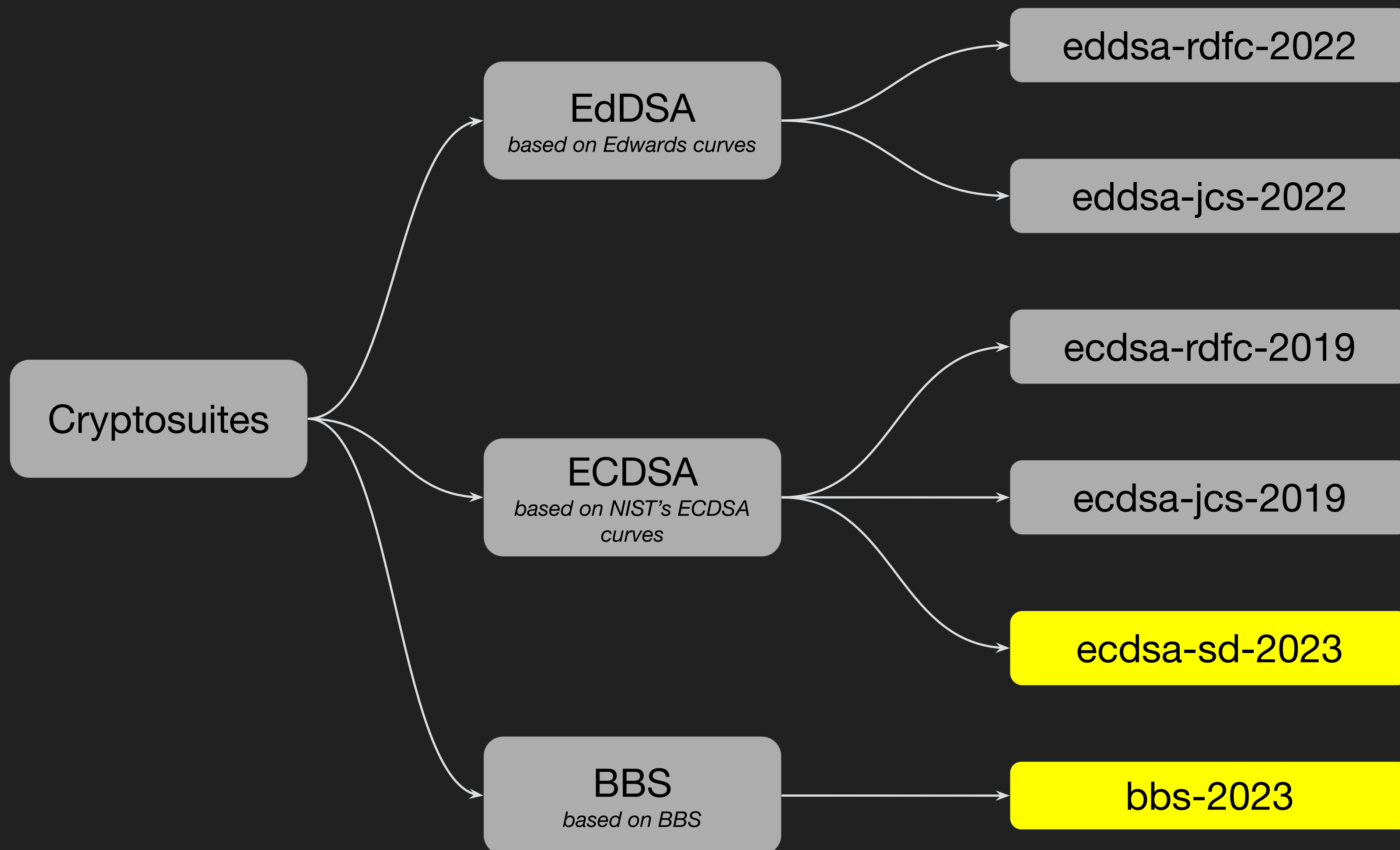
Embedded proofs: the proof is “in” the credential

- The Data Integrity specification provides a general framework to represent proofs within the JSON-LD structure
- The various “cryptosuite” specifications map cryptographic schemes to this framework
- Communities may add their cryptosuite to use other schemes (there is no central registration mechanism)
 - for example, the Chinese community may want to use SM2 instead of the NIST curves

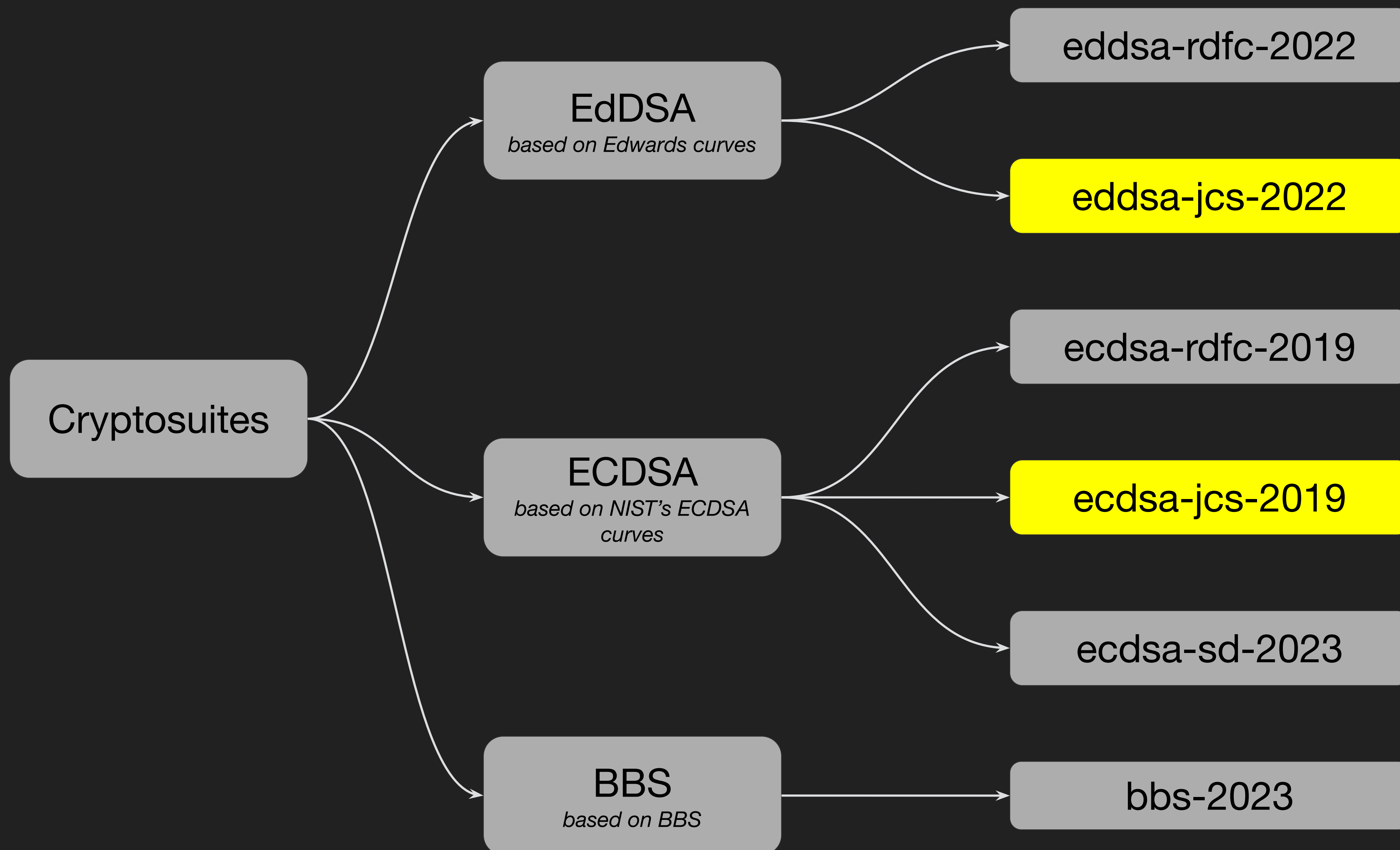
Cryptosuites defined by the VC Working Group



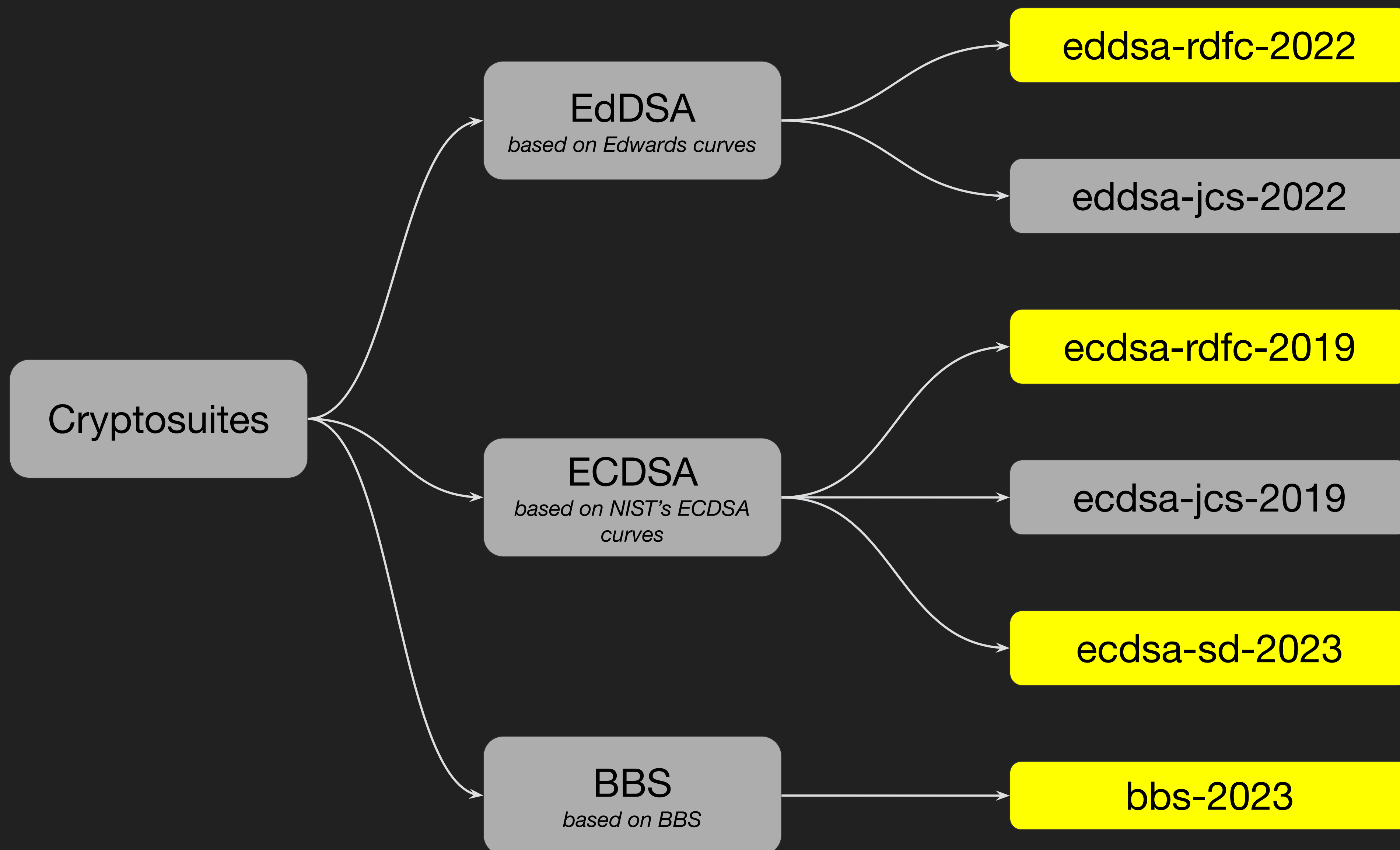
Cryptosuites with selective disclosure



Cryptosuites using JCS for canonicalization



Cryptosuites relying on the graph model



Example: Basic credential...

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.example.org/vocabs/alumni"
  ],
  "id": "https://uni.example/Credential12",
  "type": ["VerifiableCredential", "ExampleAlumniCredential"],
  "issuer": "did:example:2g55q91",
  "validFrom": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "https://www.example.org/persons/pat",
    "name": "Pat",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": "Example University"
    }
  }
},
...
```



...and signed using ECDSA

```
...  
"proof": {  
  "type": "DataIntegrityProof",  
  "cryptosuite": "ecdsa-rdfc-2019",  
  "created": "2010-01-01T00:00:00Z",  
  "expires": "2040-01-01T00:00:00Z",  
  "verificationMethod": "did:example:2...q91#ecdsa-public-key",  
  "proofPurpose": "assertionMethod",  
  "proofValue": "zQeVb...Wx"  
}  
}
```

...and signed using ECDSA

Proof metadata

```
...
"proof": {
  "type": "DataIntegrityProof",
  "cryptosuite": "ecdsa-rdfc-2019",
  "created": "2010-01-01T00:00:00Z",
  "expires": "2040-01-01T00:00:00Z",
  "verificationMethod": "did:example:2...q91#ecdsa-public-key",
  "proofPurpose": "assertionMethod",
  "proofValue": "zQeVb...Wx"
}
}
```

...and signed using ECDSA

Reference to the
ECDSA public key

```
...
"proof": {
  "type": "DataIntegrityProof",
  "cryptosuite": "ecdsa-rdfc-2019",
  "created": "2010-01-01T00:00:00Z",
  "expires": "2040-01-01T00:00:00Z",
  "verificationMethod": "did:example:2...q91#ecdsa-public-key",
  "proofPurpose": "assertionMethod",
  "proofValue": "zQeVb...Wx"
}
}
```

...and signed using ECDSA

```
...  
"proof": {  
  "type": "DataIntegrityProof",  
  "cryptosuite": "ecdsa-rdfc-2019",  
  "created": "2010-01-01T00:00:00Z",  
  "expires": "2040-01-01T00:00:00Z",  
  "verificationMethod": "did:example:2...q91#ecdsa-public-key",  
  "proofPurpose": "assertionMethod",  
  "proofValue": "zQeVb...Wx"  
}  
}
```

The ECDSA
signature itself