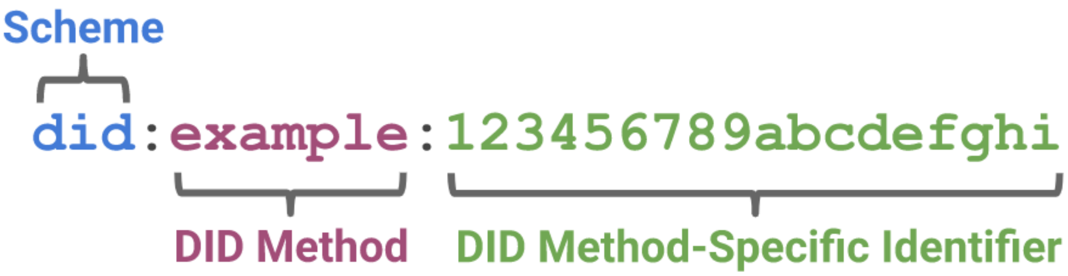
DID

URI字符串：

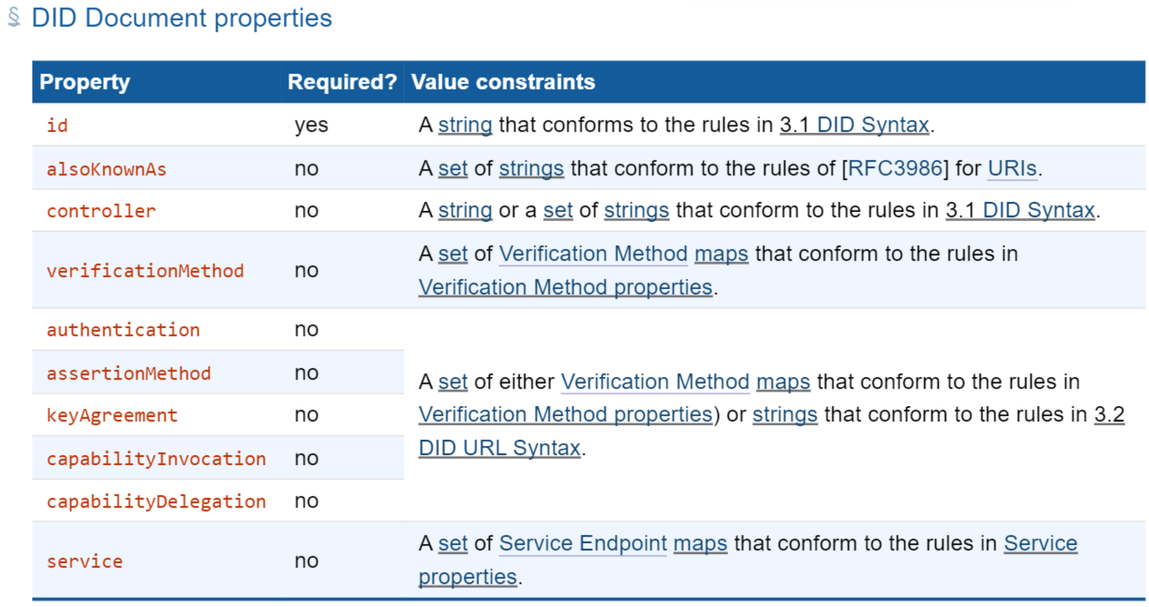

Scheme
did:example:123456789abcdefghi
DID Method
DID Method-Specific Identifier

例子：did:btcr:xys1-fd4g-f124-f5z4a

DID document

JSON:

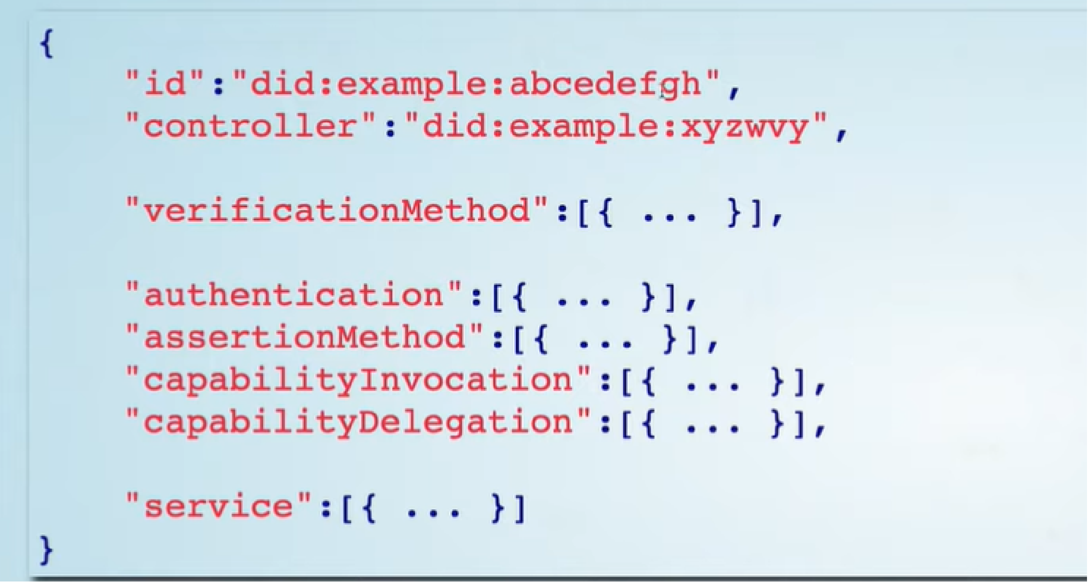
包含：did、pk、controllers、services



§ DID Document properties

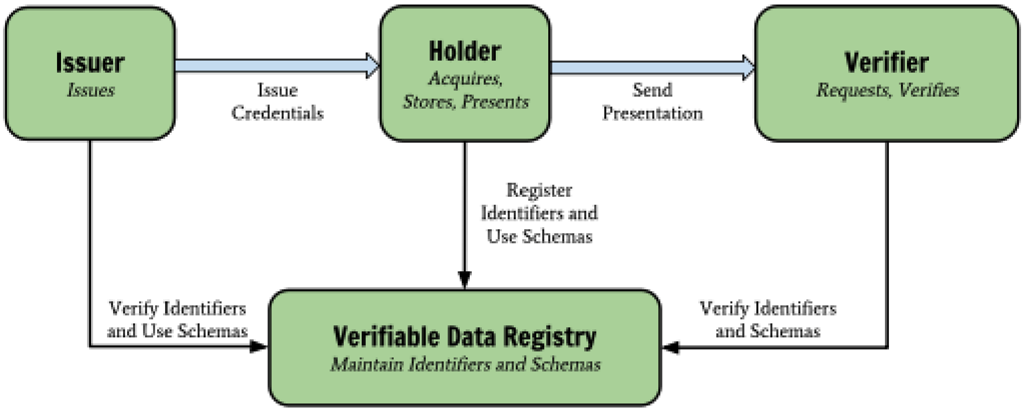
Property	Required?	Value constraints
id	yes	A string that conforms to the rules in 3.1 DID Syntax.
alsoKnownAs	no	A set of strings that conform to the rules of [RFC3986] for URIs.
controller	no	A string or a set of strings that conform to the rules in 3.1 DID Syntax.
verificationMethod	no	A set of Verification Method maps that conform to the rules in Verification Method properties.
authentication	no	A set of either Verification Method maps that conform to the rules in Verification Method properties) or strings that conform to the rules in 3.2 DID URL Syntax.
assertionMethod	no	
keyAgreement	no	
capabilityInvocation	no	
capabilityDelegation	no	
service	no	A set of Service Endpoint maps that conform to the rules in Service properties.

```
{
"@context": "https://w3id.org/did/v1",
"id": "did:example:123456789abcdefghi",
"authentication": [{
// 本DID文档对应的DID标识
"id": "did:example:123456789abcdefghi#keys-1",
"type": "RsaVerificationKey2018",
"controller": "did:example:123456789abcdefghi",
//本DID对应的公钥信息
"publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
}],
"service": [{
// 获取本DID对应的VC的服务接口
"id":"did:example:123456789abcdefghi#vcs",
"type": "VerifiableCredentialService",
"serviceEndpoint": "https://example.com/vc/"
}]
}
```



```
{
      "id":"did:example:abcedefgh",
      "controller":"did:example:xyzwvy",

      "verificationMethod":[{ ... }],

      "authentication":[{ ... }],
      "assertionMethod":[{ ... }],
      "capabilityInvocation":[{ ... }],
      "capabilityDelegation":[{ ... }],

      "service":[{ ... }]
}
```

- DID标识作为Key，DID文档可以作为Value存储（byte stream）到区块链中，利用区块链不可篡改、共享数据访问的特点，实现接下来在验证身份时能快速访问获取可信数据。
- 也可以不存储在，在需要的时候即时下载


Issuer
Issues
Issue Credentials
Holder
Acquires, Stores, Presents
Send Presentation
Verifier
Requests, Verifies
Register Identifiers and Use Schemas
Verify Identifiers and Use Schemas
Verify Identifiers and Schemas
Verifiable Data Registry
Maintain Identifiers and Schemas

VC (Verifiable Credential)

结构：

Metadata: 包含发行人、发行日期、声明的类型等信息
Credential Subject：一个或者多个关于主体的说明
Proofs：通常是颁发者的数字签名，保证了本VC能够被验证，防止VC内容被篡改以及验证VC的颁发者

示例：

```
{
  // VC内容所遵循的JSON-LD标准
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  // 本VC的唯一标识，也就是证书ID
  "id": "http://example.edu/credentials/1872",
  // VC内容的格式
  "type": ["VerifiableCredential", "AlumniCredential"],
  // 本VC的发行人
  "issuer": "https://example.edu/issuers/565049",
  // 本VC的发行时间
  "issuanceDate": "2010-01-01T19:73:24Z",

//Metadata

  // VC声明的具体内容
  "credentialSubject": {
    // 被声明的人的DID
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    // 声明的断言内容
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": [{
        "value": "Example University",
        "lang": "en"
      }, {
        "value": "Exemple d'Université",
        "lang": "fr"
      }]
    }
  },


//CredentialSubject

  // 对本VC的证明
  "proof": {
    // 签名算法
    "type": "RsaSignature2018",
    // 签名创建时间
    "created": "2017-06-18T21:19:10Z",
    // 本证明的目的
    "proofPurpose": "assertionMethod",
    // 验证本签名的公钥的ID
    "verificationMethod": "https://example.edu/issuers/keys/1",
    // 数字签名的内容
    "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5X
      sITJX1CxPCT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUc
      X16dUEMGlv50aqzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtj
      PAYuNzVBAh4vGHSrQyHUdBBPM"
  }
}
```

```
//Proof
```

生成：

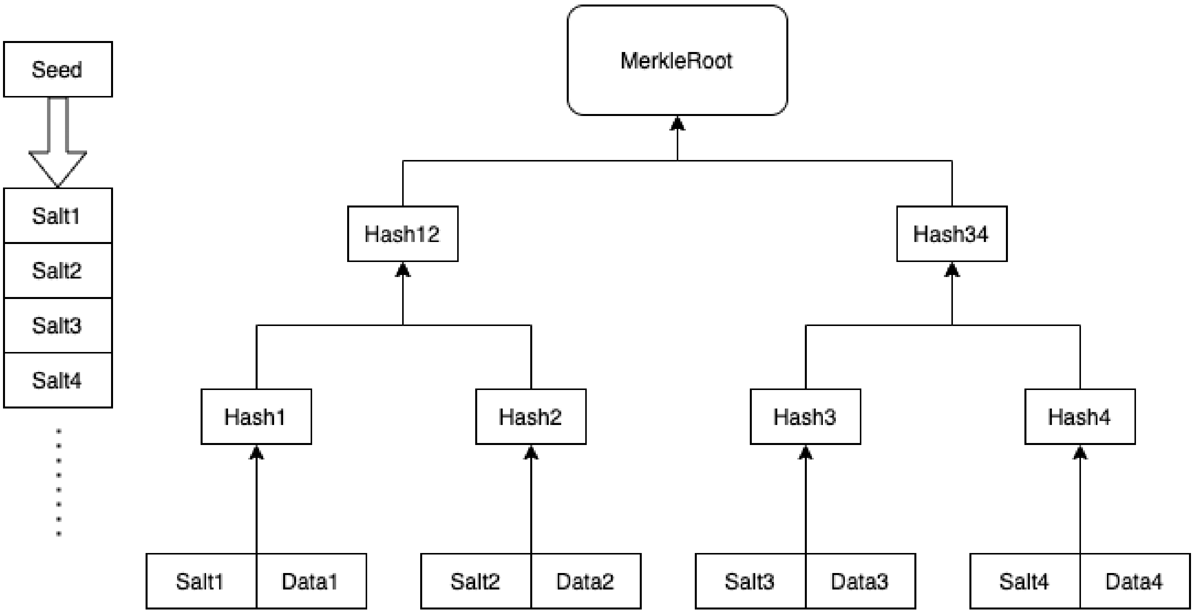
issuer根据本人的did以及其相关信息生成

随机种子seed，默克尔根，发证机关对默克尔根的签名

Seed

CA的属性作为data来构建Merkle树；

防止隐私泄露，采用随机种子seed生成N个序列作为salt，添加在data字段前



VP (Verifiable Presentation)

结构：

Metadata：主要包含了版本，本JSON对象的类型等信息
Verifiable Credential：对外展示的VC的内容
Proofs：主要是持有者对本VP的签名信息

示例：

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "type": "VerifiablePresentation",

//Metadata
```

```
  // 本VP包含的VC的内容
  "verifiableCredential": [{
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "id": "http://example.edu/credentials/1872",
    "type": ["VerifiableCredential", "AlumniCredential"],
    "issuer": "https://example.edu/issuers/565049",
    "issuanceDate": "2010-01-01T19:73:24Z",
    "credentialSubject": {
      "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
      "alumniOf": {
        "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
        "birthdate":"2000-01-01",
        //以下是验证披露字段有效性的数据
        //数据在默克尔树中的索引
        "dataIndex":2,
        //本数据加盐的值

"salt":"6b264354ed367ced527a86d38f75f9c3888bd3939f548cc48d93af435890b84a",
        //默克尔验证路径

"merklesibling":"34b64151443c3124620bf4ff69a05e97d580f0878b374b8343c6a5c3d822343
5 9d2b5b35ccb5bf18747c1f5dc05771c68ce613e6eb0c5f5ef77cec8ba3e9da67
bb82c63d4e21525125bf66a6724fbb4dcbded26aae2baa2633235dc12730016e",
        //默克尔根哈希

"merkleRoot":"ea59a369466be42d1a4783f09ae0721a5a157d6dba9c4b053d407b5a4b9af145",
    //公安机关对默克尔根的签名

"rootSignature":"3066022051757c2de7032a0c887c3fcef02ca3812fede7ca748254771b9513d
8e266",
    //用的公安机关哪个Key进行的签名
        "signer":"did:公安部门ID#keys-1"
      }
    },



//Verifiable Credential

    "proof": {
      "type": "RsaSignature2018",
      "created": "2017-06-18T21:19:10Z",
      "proofPurpose": "assertionMethod",
      "verificationMethod": "https://example.edu/issuers/keys/1",
      "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5X
        sITJX1CxPCT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUc
        X16dUEMGlv50aqzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtj
        PAYuNzVBAh4vGHSrQyHUdBBPM"
    }
  }],
  // Holder对本VP的签名信息
  "proof": {
    "type": "RsaSignature2018",
    "created": "2018-09-14T21:19:10Z",
    "proofPurpose": "authentication",
    "verificationMethod": "did:example:ebfeb1f712ebc6f1c276e12ec21#keys-1",
    // challenge和domain是为了防止重放攻击而设计的
    "challenge": "1f44d55f-f161-4938-a659-f8026467f126",
    "domain": "4jt78h47fh47",
    "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..kTCYt5
      XsITJX1CxPCT8yAV-TVIw5WEuts01mq-pQy7UJiN5mgREEMGlv50aqzpqh4Qq_PbChOMqs
      LfRoPsnsgxD-WUcX16dUOqV0G_zS245-kronKb78cPktb3rk-BuQy72IFLN25DYuNzVBAh
      4vGHSrQyHUGlcTwLtjPAnKb78"
  }
}


//Proof
```

生成：

基于VC

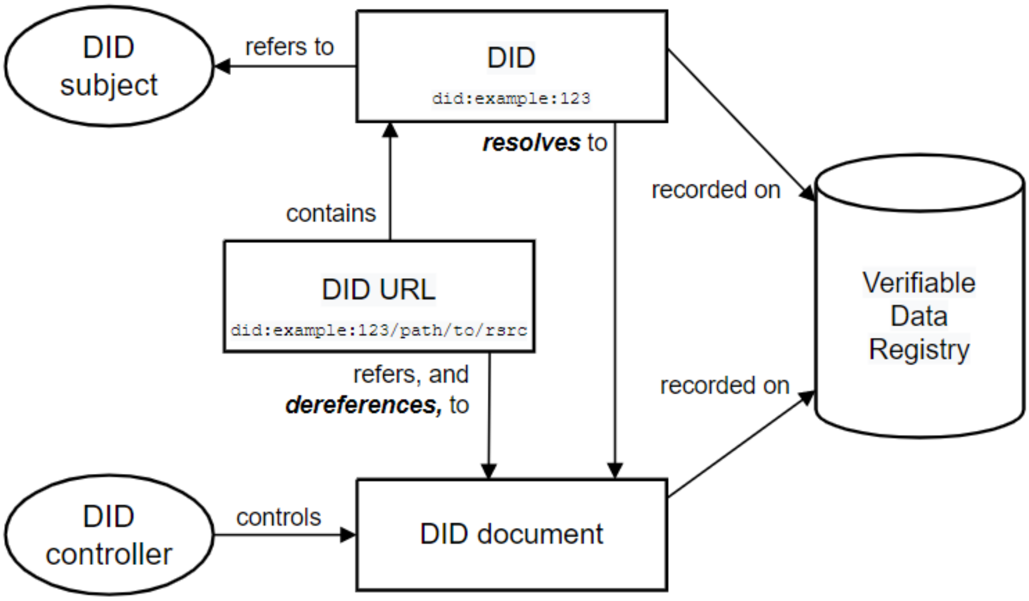
Verifier零知识证明请求 -> QR code（数字身份App）

验证：

```
1. Holder提交的——验证数字签名（Proof: Holder's did->did document->pk->sig）
2. Issuer可信机构颁布——（Proof: creator's did-> ...）
3. 验证——MerkleRoot
4. 验证披露字段——salt、验证路径
```

应用

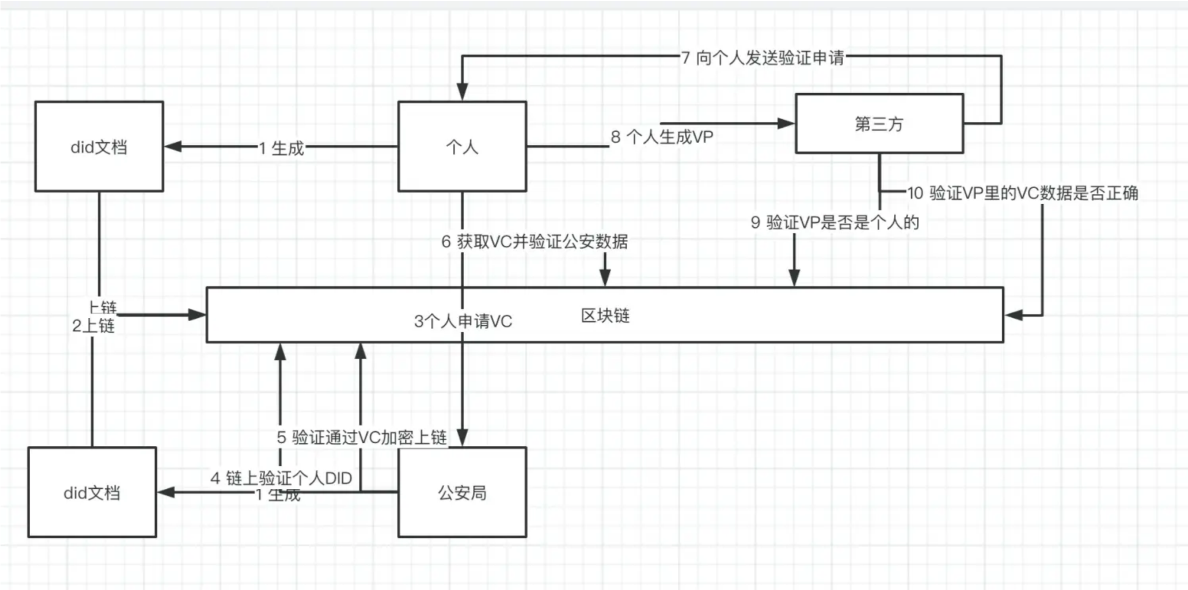
- 无密码安全登录（类似于微信扫码登录，但信息全局掌握在自己手里）
- 身份验证（学籍认证、电子签名...）
- IoT（自动更新产品信息、防伪...）



区块链的作用：

可信

存储



6.个人获取VC并验证

个人通过VC的ID 从链上下载VC

1. 通过我的私钥解密 `credentialSubject` 获得明文所有属性
2. 根据所有属性生成MerkleTree，并获取MerkleRoot
3. 验证VC的 `signatureValue` 是否是公安局的MerkleRoot 签名
4. 通过签名后，表示该VC就是公安局背书的，并且数据没有被修改
5. 将VC属性明文保存到本地

智能合约

1. 定义DID Document的存储结构和读写方式
2. DID存在更新需求

评估一下使用w3c/ens

ens的标准（更像域名注册的web3版本）

dns合约

域名注册流程

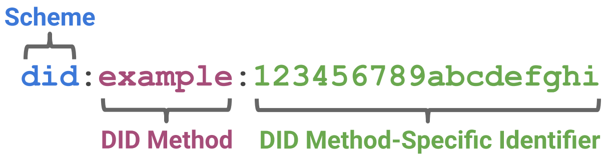
w3web3的标准

bid的范围更广

DID 重新梳理

DID & DID Document

- DID


Scheme
did:example:123456789abcdefghi
DID Method
DID Method-Specific Identifier

- DID Document

```
EXAMPLE 1: A simple DID document
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ]
  "id": "did:example:123456789abcdefghi",
  "authentication": [{

    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "Ed25519VerificationKey2020",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
  }]
}
```

authentication is a process

an entity can prove:

- it has a specific attribute
- it controls a specific secret

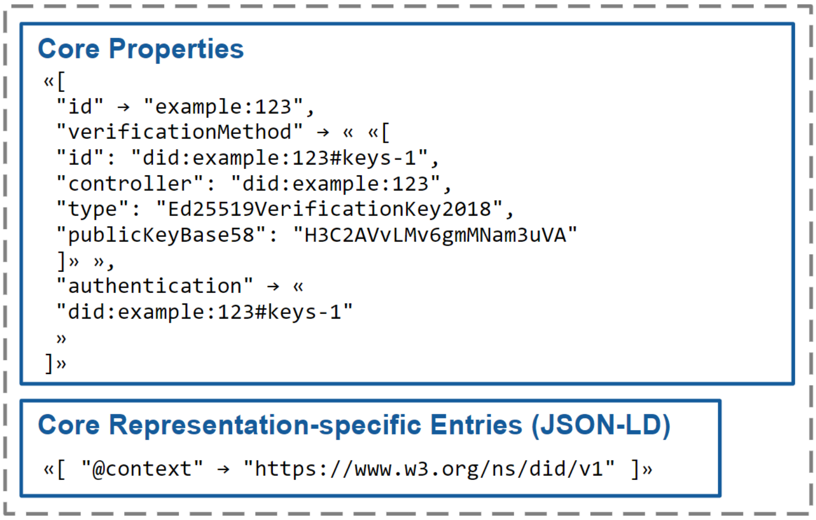
Data Model - Map of Entries

*Can be serialize to a representation

at least 2 entries, each entry consists of a **key/value** pair

> key type: string
>
> value type: list, map, datetime...(each type specify the serialize method)

- **core properties**
- **core representation-specific entries**

Data Model	Example
	

```
«[
 "id" → "example:123",
 "verificationMethod" → « «[
 "id": "did:example:123#keys-1",
 "controller": "did:example:123",
 "type": "Ed25519VerificationKey2018",
 "publicKeyBase58": "H3C2AVvLMv6gmMNam3uVA"
 ]» »,
 "authentication" → «
 "did:example:123#keys-1"
 »
]»
```

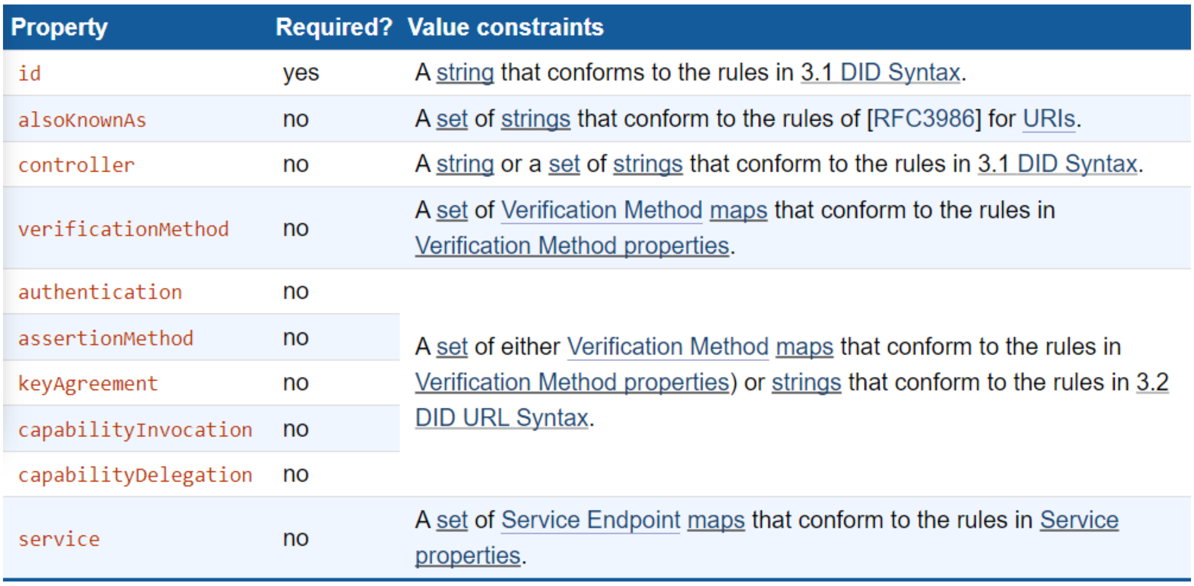
```
«[ "@context" → "https://www.w3.org/ns/did/v1" ]»
```

- extended **properties/representation** should use the W3C DID Specification Registries mechanism [DID-SPEC-REGISTRIES]

Core representation-specific entries

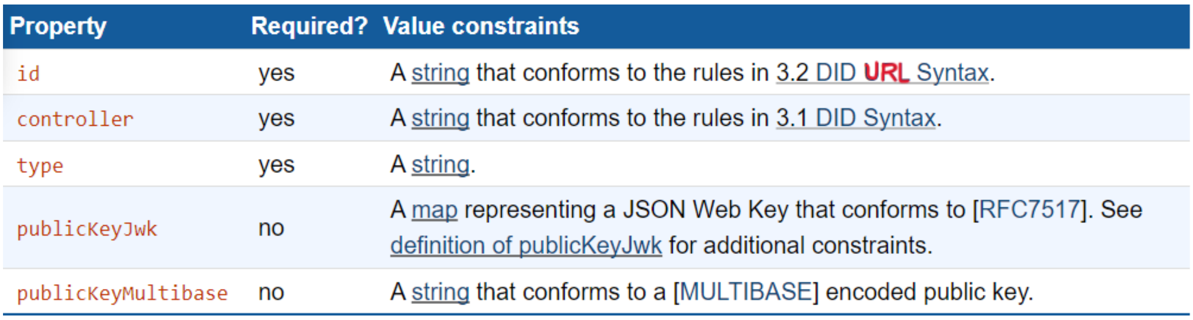
```
"@context": ...
```

Core Properties



Property	Required?	Value constraints
id	yes	A string that conforms to the rules in 3.1 DID Syntax.
alsoKnownAs	no	A set of strings that conform to the rules of [RFC3986] for URIs.
controller	no	A string or a set of strings that conform to the rules in 3.1 DID Syntax.
verificationMethod	no	A set of Verification Method maps that conform to the rules in Verification Method properties.
authentication	no	A set of either Verification Method maps that conform to the rules in Verification Method properties) or strings that conform to the rules in 3.2 DID URL Syntax.
assertionMethod	no	
keyAgreement	no	
capabilityInvocation	no	
capabilityDelegation	no	
service	no	A set of Service Endpoint maps that conform to the rules in Service properties.

verificationMethod Properties



Property	Required?	Value constraints
id	yes	A string that conforms to the rules in 3.2 DID URL Syntax.
controller	yes	A string that conforms to the rules in 3.1 DID Syntax.
type	yes	A string.
publicKeyJwk	no	A map representing a JSON Web Key that conforms to [RFC7517]. See definition of publicKeyJwk for additional constraints.
publicKeyMultibase	no	A string that conforms to a [MULTIBASE] encoded public key.

- 可以用来验证proof的一组参数
- 例如，它可以是加密公钥，用来验证数字签名（是否被私钥加密）
- 此外，controller可以授权verificationMethod给delegate以进行authentication
- 关于DID Document core property里的controller和verificationMethod里的controller的区别。如果把DID subject当作一个房子，则前者controller相当于房东（房东有钥匙），verificationMethod相当于备用钥匙，里面的controller（可以有多个）相当于钥匙的所有者。
- 后两个属性：publicKeyJwk & publicKeyMultibase被称为verification Material

verification relationship

Relationship between **subject** and **verification method**

E1. authentication

how subject is expected to be authenticated (e.g. for purpose logging into a website)

2 ways:

- referenced

```
"did:example:123456789abcdefghi#keys-1"
```

- embedded

```
    {
      "id": "did:example:123456789abcdefghi#keys-2",
      "type": "Ed25519VerificationKey2020",
      "controller": "did:example:123456789abcdefghi",
      "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
    }
```

E2. assertionMethod

Used to specify how the subject is expected to **express claims** (such as issuing a VC)

2 ways: `referenced` & `embedded`

- verifier收到VC之后，可以查到issuer的DID，然后resolve to DID Doc，找到assertionMethod，从而可以验证VC的有效性

E3. keyAgreement

Used to specify how entity产生**encryption material**，然后传送给**subject** (such as establishing a secure communication channel with the recipient)

- 通过DID resolve to DID Doc， 然后找到keyAgreement里的公钥，用此加密想要发送给subject的材料，实现消息的秘密传输

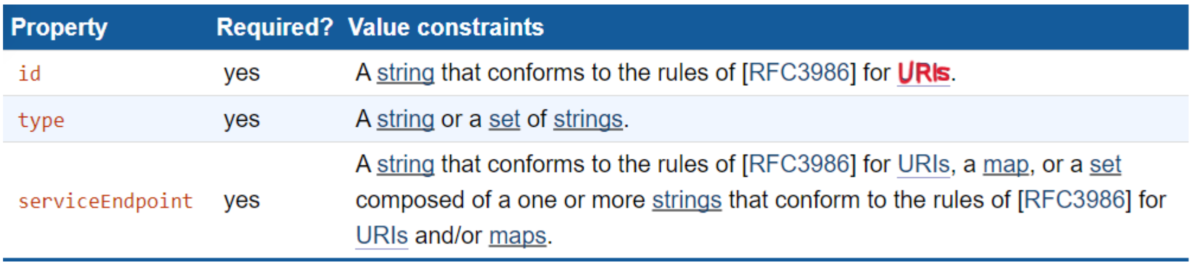
E4. capabilityInvocation

Used to specify a verification method being used by **subject** to **invoke a cryptographic capability** (such as the authorization to update the DID Document)

E5. capabilityDelegation

Used to specify a mechanism being used by **subject** to **delegate a cryptographic capability to another party (such as delegating the authority to access a specific HTTP API to a subordinate)

service Properties



Property	Required?	Value constraints
id	yes	A string that conforms to the rules of [RFC3986] for URIs.
type	yes	A string or a set of strings.
serviceEndpoint	yes	A string that conforms to the rules of [RFC3986] for URIs, a map, or a set composed of a one or more strings that conform to the rules of [RFC3986] for URIs and/or maps.

- 是可以通过service endpoint与subject交互的方式

DID resolver

input: DID

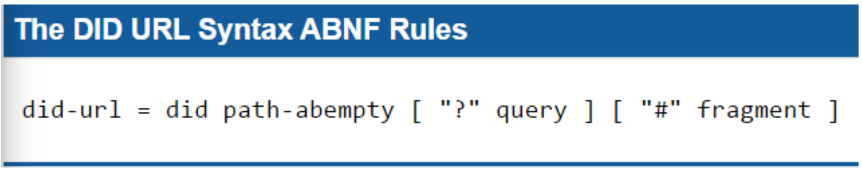
output: DID Document

DID URL & external resource

DID URL 扩展了基本 DID 的语法以包含其他标准 URI 组件，以便定位特定资源

（例如representations of DID subjects, verification methods, services, specific parts of a DID document...)

***语法和一般的URI类似**，包括路径path、查询query和片段fragment



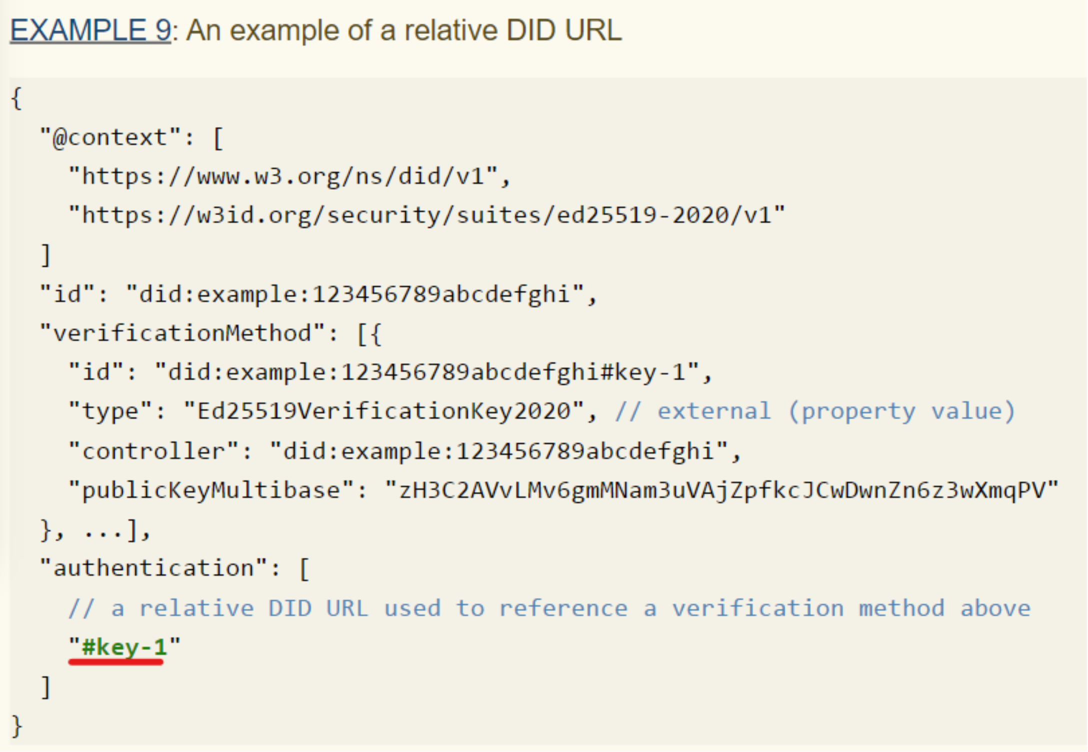
The DID URL Syntax ABNF Rules

```
did-url = did path-abempty [ "?" query ] [ "#" fragment ]
```

- path　由did-method进一步确定　`did:example:123456/path`
- query
- fragment

relative DID URL

存在于DID Doc中，没有严格按照DID URL的语法



EXAMPLE 9: An example of a relative DID URL

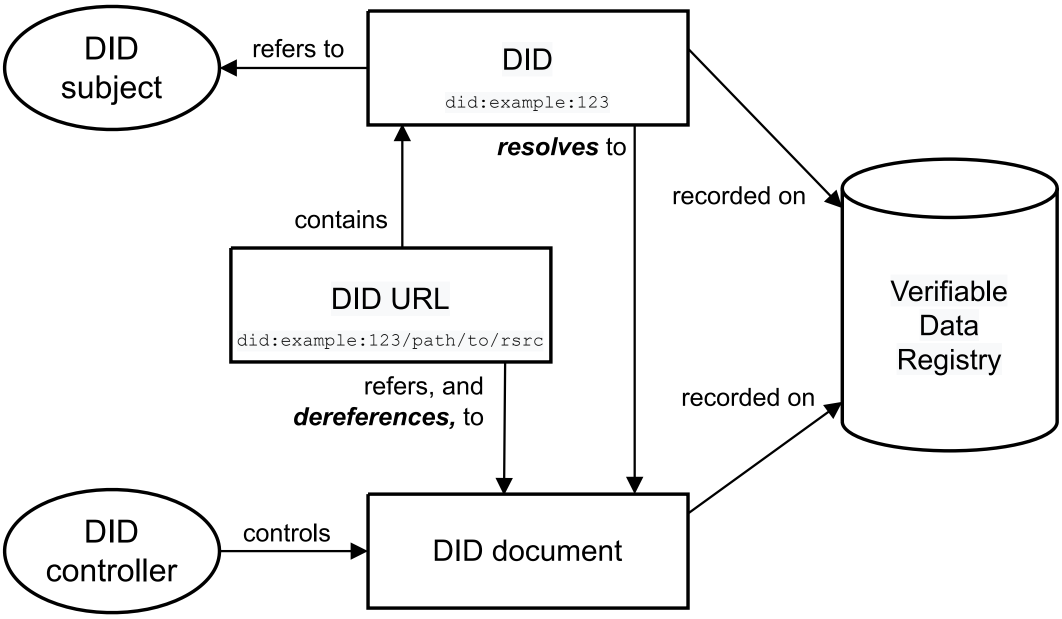
```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ]
  "id": "did:example:123456789abcdefghi",
  "verificationMethod": [{
    "id": "did:example:123456789abcdefghi#key-1",
    "type": "Ed25519VerificationKey2020", // external (property value)
    "controller": "did:example:123456789abcdefghi",
    "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
  }, ...],
  "authentication": [
    // a relative DID URL used to reference a verification method above
    "#key-1"
  ]
}
```

DID URL dereferencers

input: DID URL

output: external resource

Architecture



Representation

Representation is the concrete serialization of DID Doc

(DID Doc) data model -> **Produce** -> representation

representation -> **Consumption** -> data model

Verifiable Credentials Ecosystem

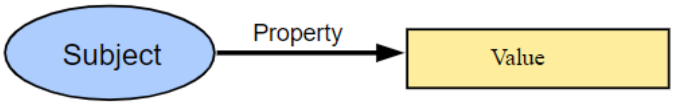
VC

Verifiable Credentials Data Model

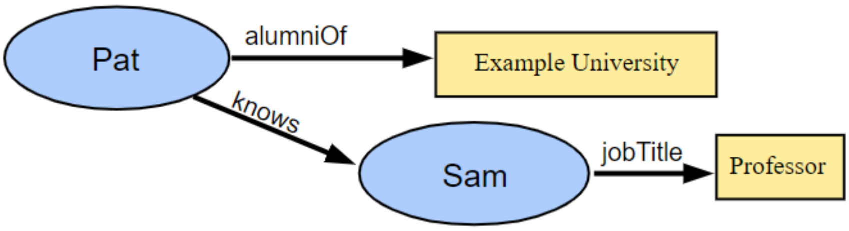
VC -> VP (Verifiable Presentation)

A claim is a statement about a subject

the data model of a claim is



各种关系的描述



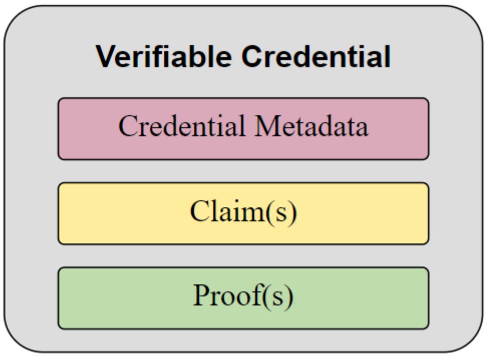
- stored in 可信的仓库（例如电子钱包）
- 校友 例子

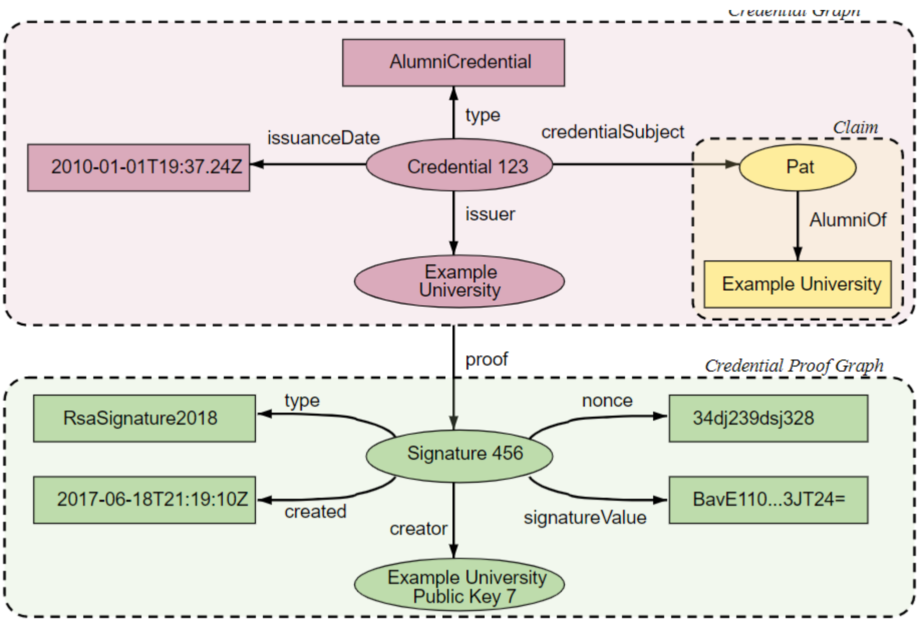
```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],

  "id": "http://example.edu/credentials/1872",
  "type": ["VerifiableCredential", "AlumniCredential"],
  "issuer": "https://example.edu/issuers/565049",
  "issuanceDate": "2010-01-01T19:23:24Z",

  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": [{
        "value": "Example University",
        "lang": "en"
        }, {
        "value": "Exemple d'Université",
        "lang": "fr"
        }]
      }
    },

  "proof": {
    "type": "RsaSignature2018",
    "created": "2017-06-18T21:19:10Z",
    "proofPurpose": "assertionMethod",
    "verificationMethod": "https://example.edu/issuers/565049#key-1",
    "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5X
      sITJX1CxPCT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUc
      X16dUEMGlv50aqzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtj
      PAYuNzVBAh4vGHSrQyHUdBBPM"
  }
}
```

- **id**： URI
- **credentialSubject**: claim - id: did
- **issuer**: URI
- **proof**: external(**J**SON **W**eb **T**oken) & embedded
  - 例如签名
    - 包含：signature 和 指向sig实体的reference

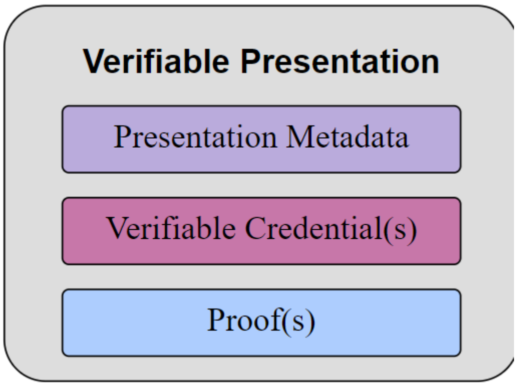
```
  "proof": {
    "type": "Ed25519Signature2020",
    "created": "2021-11-13T18:19:39Z",
    "verificationMethod": "https://example.edu/issuers/14#key-
1",
    "proofPurpose": "assertionMethod",
    "proofValue":
"z58DAdFfa9SkqZMVPxAQpic7ndSayn1PzZs6ZjWp1CktyGesjuTSwRdo
                   WhAfGFCF5bppETSTojQCrfFPP2oumHKtz"
  }
```

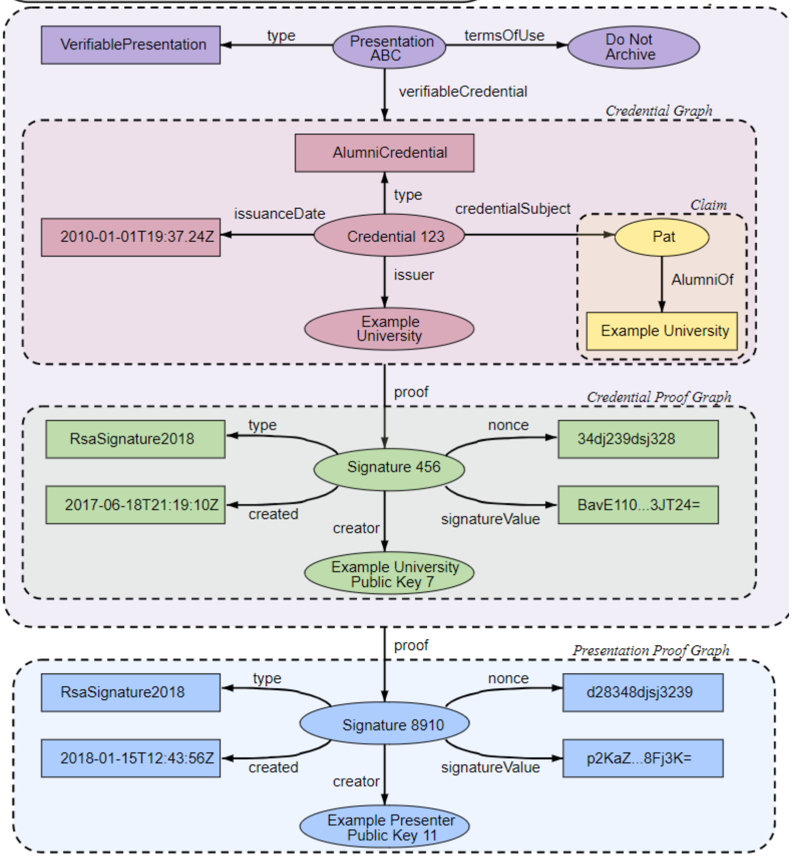
- 可以被holder transfer





VP





- CP可以引用自多个CV

其它

- credentialSchema
- refreshService