ICANN 76 CANCÚN

DID to DNS with DNSSEC ☺

CIRA

March 15, 2023

Presented By
Jacques Latour – CIRA – in person
Jesse Carter – CIRA - remote

cira

ENHANCING THE DIGITAL TRUST INFRASTRUCTURE USING DNS & DNSSEC

When an entity is presented with a verifiable claim, there are three things they will want to ensure:

✓ 1 **That a claim hasn't been altered/falsified at any point in time**

(Cryptographic verifiability, Verifiable Data Registries - VDRs)

? 2 **That a claim has accurate representation**

(Authentication, DID Discovery/mapping within DNS)

? 3 **That a claim has authority**

(Authorization, Trust Registries/trust lists)

cira

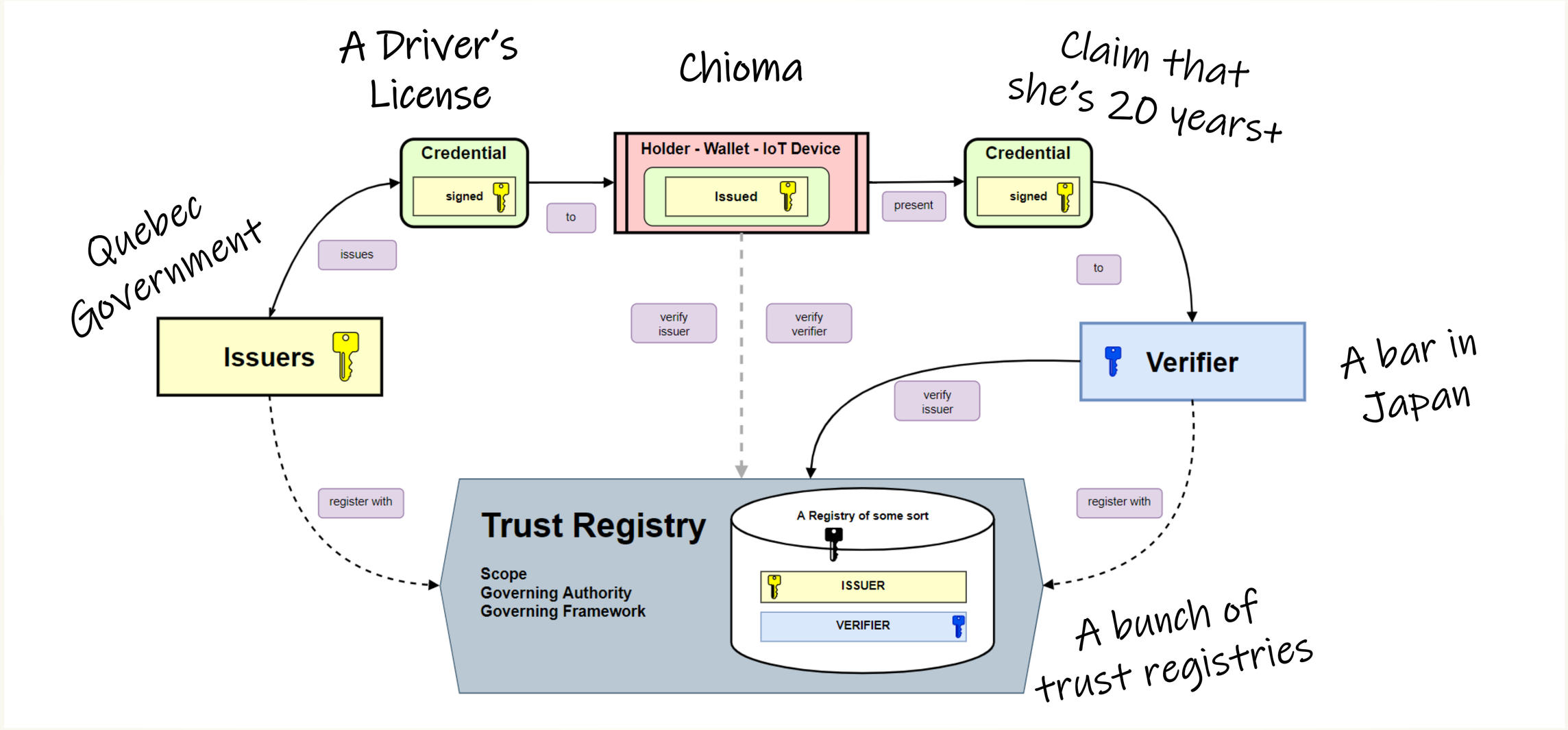
Here's the story

Chioma from Gatineau, Quebec is travelling in Japan. She goes to a bar where she is asked to prove she is 20 years or older (the legal drinking age in Japan). **Chioma** produces a proof of age claim with her driver's licence digital credential, which was issued to her by Quebec's driver's licence issuer.

The bar in Japan has never heard of Quebec, but its verifying technology can see that the claim is made from a Canadian based governmental organization. This triggers:

1. the bar's verifying technology to reference Japan's apex trust registry,
2. which then using DNS can reference Canada's apex trust registry,
3. which can then "route the call" to Quebec trust registry,
4. which can in turn confirm that Quebec's driver licence issuer is both a **legitimate** entity and has the **authority** to issue a proof of age claim (via a driver's licence).

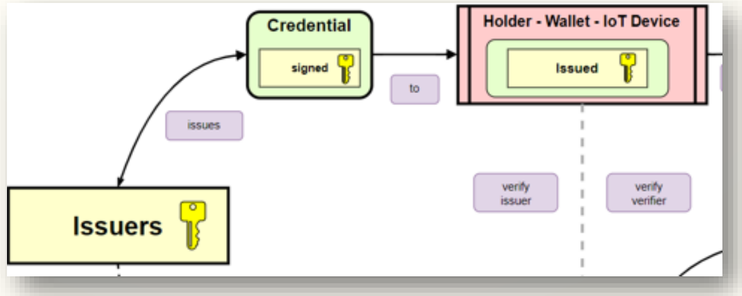
The Digital Credential Ecosystem



WHERE'S THE BEGINNING?

Let's start with verifiable credentials

- The individual, in this case **Chioma**, owns a digital wallet capable of holding and presenting a verifiable credential
- A verifiable credential in this example is based on the W3C Verifiable Credential Data Model v1.1 (https://www.w3.org/TR/vc-data-model/)
- The issuer, tr-demo.ciralabs.ca, issues the credential to **Chioma's** digital wallet by proxy of the DID they control.
    - did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM.



cira

THE DIGITAL CREDENTIAL

- The credential contains metadata about the subject, **Chioma**, and the claims it is representing (could be a degree, age, driver's license class, etc.)
- The credential points to an issuer, did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM
- The credential contains a digital proof, a cryptographic signature of the credential content ensuring its integrity
- The verificationMethod points to a public key which can be used to verify that signature

```
{
    // set the context
    "@context": [
        "https://www.w3.org/2018/credentials/v1"
    ],
    // specify the identifier of the credential
    "id": "https://tr-demo.ciralabs.ca/driverslicenses/01",
    // the credential types, which declare what data to expect in the credential
    "type": ["VerifiableCredential", "DriversLicenseCredential"],
    // the entity that issued the credential
    "issuer": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
    // when the credential was issued
    "issuanceDate": "2023-03-07T19:23:24Z",
    // claims about the subject of the credential
    "credentialSubject": {
        "id": "did:example:chiomasdid1234",
        "licenseClass": 7,
        "name": "Chioma Doe",
        "dateOfBirth": "1999/01/01",
        "height": "165 cm",
        "address": "11 Credential Road, Prince Albert, SK S0N 7V1",
    },
    // digital proof that makes the credential tamper-evident
    "proof": {
        // the cryptographic signature suite that was used to generate the signature
        "type": "Ed25519Signature2018",
        // the date the signature was created
        "created": "2023-03-07T19:23:24Z",
        // purpose of this proof
        "proofPurpose": "assertionMethod"
        // the identifier of the public key that can verify the signature
        "verificationMethod": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1",
        // the digital signature value
        "proofValue": "78f9de868f8f468f9becab8fa20ac3160159ee656a7ebe704cb08f8f
        ba3d0da311bf2dd9caaab47411f6519e047616ae2a983b1c9922fee00185d3aba373780d"
    }
}
```


cira

VERIFYING THE VERIFIABLE CREDENTIAL

- To verify the credential, the verifier must first resolve the public key in the proof section.
  - In this case, the verificationMethod points to a key held by the same DID which issued the credential: did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM
- Resolving that DID will yield a DID Document, a set of data describing the DID subject.

```
Please enter a DID you wish to verify via DNS/TLSA:
did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM
```

```
Resolved DID document:{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2018/v1",
    "https://w3id.org/security/suites/x25519-2019/v1"
  ],
  "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
  "verificationMethod": [
    {
      "type": "Ed25519VerificationKey2018",
      "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1",
      "controller": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
      "publicKeyBase58": "HdXo5kegxgPze3tAw6QYU7vvJg4gbxztqSidt8LsB6eS"
    },
    {
      "type": "X25519KeyAgreementKey2019",
      "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-agreement-1",
      "controller": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
      "publicKeyBase58": "3shqJTQvoTa7JwhRGGEjwyKhcC2xaVTiyRt2pznp7XsR"
    }
  ],
  "authentication": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1"
  ],
  "assertionMethod": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1"
  ],
  "keyAgreement": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-agreement-1"
  ]
}
```


cira

TRY: https://dev.uniresolver.io/ DID-URL: did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM

VERIFYING THE VERIFIABLE CREDENTIAL

- The resolved DID document contains the public key we need to verify the credential: did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1

- By converting the public key represented in base 58 to hex we get: f716b82bd54ecddb908d804cd031512eb5cb3a5454d0c1024e30b04a88251cff

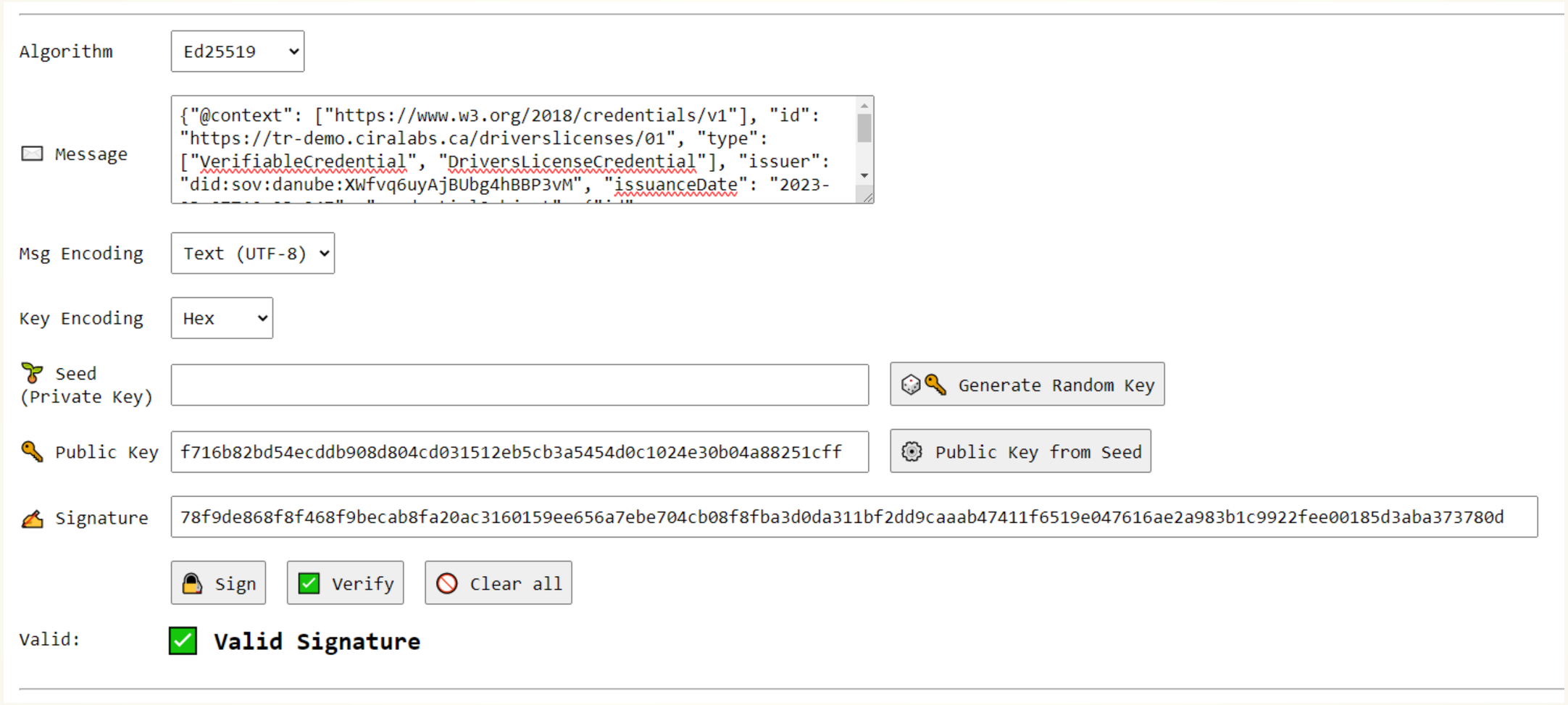
- The signature of the content in the verifiable credential can then be referenced against the public key to determine if that key pair really did sign the credential…

```
"verificationMethod": [
  {
    "type": "Ed25519VerificationKey2018",
    "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1",
    "controller": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
    "publicKeyBase58": "HdXo5kegxgPze3tAw6QYU7vvJg4gbxztqSidt8LsB6eS"
  },
```

```
// digital proof that makes the credential tamper-evident
"proof": {
    // the cryptographic signature suite that was used to generate the signature
    "type": "Ed25519Signature2018",
    // the date the signature was created
    "created": "2023-03-07T19:23:24Z",
    // purpose of this proof
    "proofPurpose": "assertionMethod",
    // the identifier of the public key that can verify the signature
    "verificationMethod": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1",
    // the digital signature value
    "proofValue": "78f9de868f8f468f9becab8fa20ac3160159ee656a7ebe704cb08f8f
    ba3d0da311bf2dd9caaab47411f6519e047616ae2a983b1c9922fee00185d3aba373780d"
}
```


cira

SURPRISE, IT DID!



https://cyphr.me/ed25519_applet/ed.html

WITH THE CREDENTIAL VERIFIED, HOW DO WE VERIFY THE ISSUER?

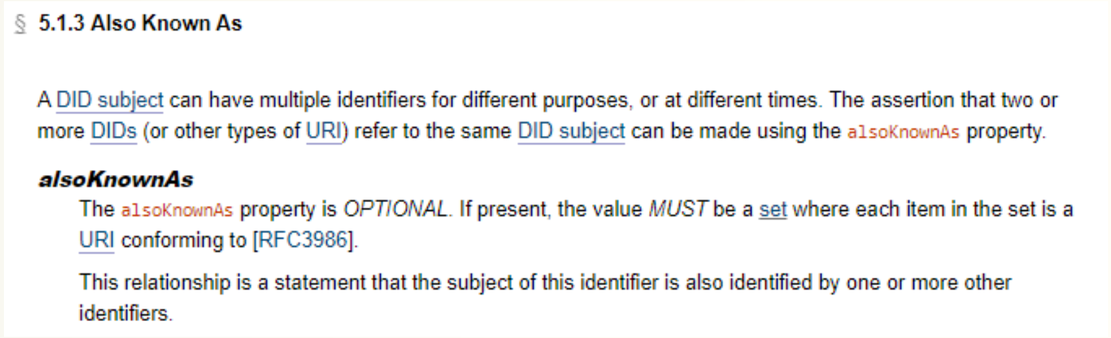
- Through the cryptographic verification process we just outlined; we have confirmed that the key pair in possession of did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM did in fact sign and issue the digital credential to **Chioma**.
- But the question remains, how do we verify the issuer, did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM?
    - Is did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM associated with an organization or entity we trust?
    - Does did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM have the authority to issue that credential to **Chioma**?
    - Where and how do I verify all the above?

WWW.CIRA.CA

The answer is in the DNS…

HOW DO WE LINK AN ISSUER'S DID TO A DOMAIN?

Unique DID to DNS name mapping



§ 5.1.3 Also Known As

A DID subject can have multiple identifiers for different purposes, or at different times. The assertion that two or more DIDs (or other types of URI) refer to the same DID subject can be made using the `alsoKnownAs` property.

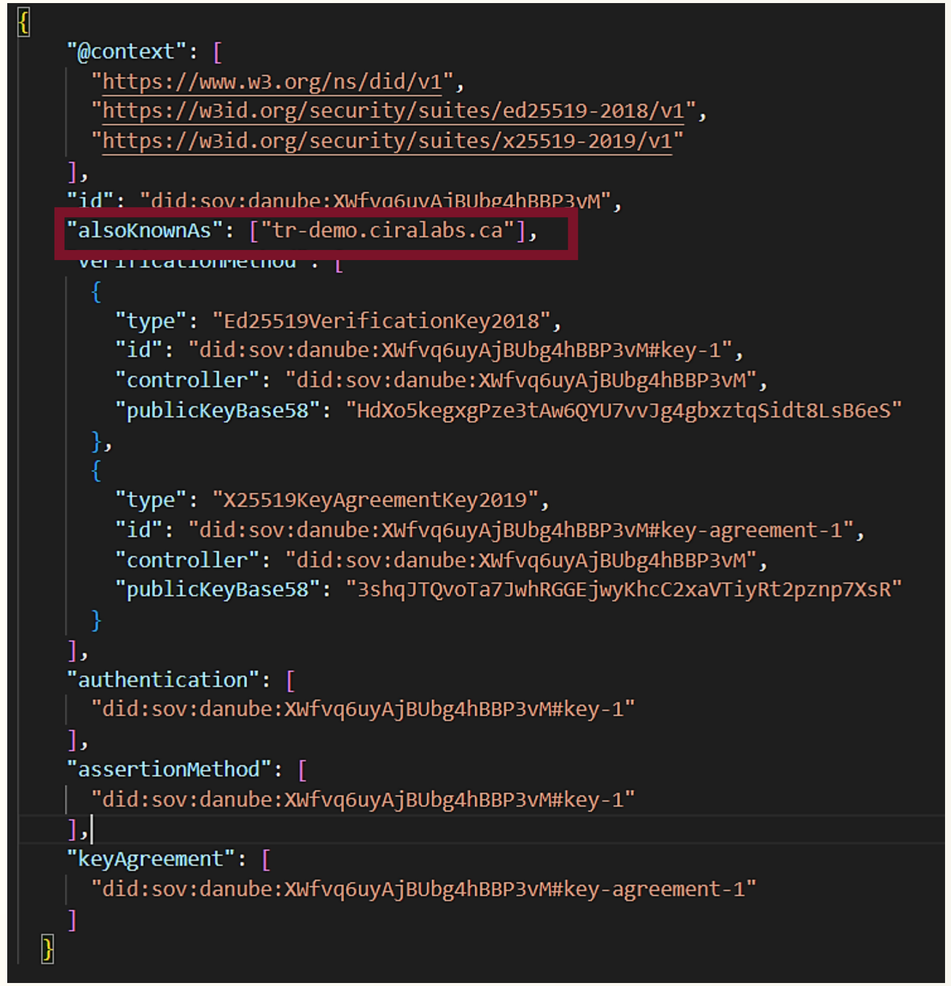
alsoKnownAs

The `alsoKnownAs` property is *OPTIONAL*. If present, the value *MUST* be a set where each item in the set is a URI conforming to [RFC3986].

This relationship is a statement that the subject of this identifier is also identified by one or more other identifiers.

https://www.w3.org/TR/did-core/#also-known-as

- As specified in the W3C DID Core spec v1.0, the "alsoKnownAs" field can contain a DID or URI, making the assertion that the URI in the field refers to the same subject.
- The "alsoKnownAs" assertion can then be used to indicate under which domain the relevant DNS records (URI and TLSA) for this DID can be found.



```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2018/v1",
    "https://w3id.org/security/suites/x25519-2019/v1"
  ],
  "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
  "alsoKnownAs": ["tr-demo.ciralabs.ca"],
  "verificationMethod": [
    {
      "type": "Ed25519VerificationKey2018",
      "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1",
      "controller": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
      "publicKeyBase58": "HdXo5kegxgPze3tAw6QYU7vvJg4gbxztqSidt8LsB6eS"
    },
    {
      "type": "X25519KeyAgreementKey2019",
      "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-agreement-1",
      "controller": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
      "publicKeyBase58": "3shqJTQvoTa7JwhRGGEjwyKhcC2xaVTiyRt2pznp7XsR"
    }
  ],
  "authentication": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1"
  ],
  "assertionMethod": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1"
  ],
  "keyAgreement": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-agreement-1"
  ]
}
```


cira

HOW DO WE AN ISSUER'S DID TO A DOMAIN?

Unique DID to DNS name mapping

§ 5.4 Services

Services are used in DID documents to express ways of communicating with the DID subject or associated entities. A service can be any type of service the DID subject wants to advertise, including decentralized identity management services for further discovery, authentication, authorization, or interaction.

https://www.w3.org/TR/did-core/#also-known-as

- As specified in the W3C DID Core spec v1.0, the "services" field can be leveraged to indicate a way to communicate with the DID subject or an associated entity.
- Like the "alsoKnownAs" assertion, the "services" field can also be used to indicate under which domain the relevant DNS records (URI and TLSA) for this DID can be found.


```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2018/v1",
    "https://w3id.org/security/suites/x25519-2019/v1"
  ],
  "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
  "service": [{
      "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#linked-domain",
      "type": "LinkedDomains",
      "serviceEndpoint": "https://tr-demo.ciralabs.ca"
  }],
  "verificationMethod": [
    {
      "type": "Ed25519VerificationKey2018",
      "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1",
      "controller": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
      "publicKeyBase58": "HdXo5kegxgPze3tAw6QYU7vvJg4gbxztqSidt8LsB6eS"
    },
    {
      "type": "X25519KeyAgreementKey2019",
      "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-agreement-1",
      "controller": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
      "publicKeyBase58": "3shqJTQvoTa7JwhRGGEjwyKhcC2xaVTiyRt2pznp7XsR"
    }
  ],
  "authentication": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1"
  ],
  "assertionMethod": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1"
  ],
  "keyAgreement": [
    "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-agreement-1"
  ]
}
```

cira

THE ROLE OF THE ISSUER

How do we link the domain of the issuer and their DID in DNS?

- A URI record allows us to associate a hostname (like tr-demo.ciralabs.ca) to a URI, like did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM.
- Through this association, a domain can show which DIDs are under their jurisdiction, in a publicly accessible and resolvable manner.
- Ex: URI Record
  - _**did**.tr-demo.ciralabs.ca 1 0 “did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM”
- But this association with URI records only represents a claim, and a claim needs support. We once again turn to cryptography and the help of TLSA records as support for this claim.

```
$ dig _did.tr-demo.ciralabs.ca URI
```

THE ROLE OF THE ISSUER

We need TLSA records to provide the cryptographic glue holding the association between a domain and a DID together.

- **A TLSA record allows us to host cryptographic information in the DNS.**
  - These TLSA records will hold the public half of the key pair used by the DID/Issuer to issue a digital credential.
  - Ex: TLSA Records
    - _did.tr-demo.ciralabs.ca 3 1 0 F716B82BD54… public key un-hashed)
    - _did.tr-demo.ciralabs.ca 3 1 1 8A9E530067D…(sha256 public key hash)
  - By hosting the public keys in the DNS, the Verifier can not only check the digital credential against the verification method in the DID document, but also against the public keys via DNS as well.
  - In a similar manor, a cryptographic challenged can be done against both the domain and the DID to ensure they both have access to the associated private key.

```
$ dig _did.tr-demo.ciralabs.ca TLSA
```


cira

```
Attempting to resolve TLSA records for: did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM._did.tr-demo.ciralabs.ca...
DNS resolution successful.
TLSA records: ['3 1 1 8a9e530067d29d399ac190006344822889385e91d70ba249c98dcd669d849460', '3 1 0 f716b82bd54ecddb908d804cd031512eb5cb3a5454d0c1024e30b04a8825
1cff']

Please select which key you would like to match against the TLSA records: ['did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1', 'did:sov:danube:XWfvq6uyAjBUbg4hBB
P3vM#key-agreement-1']
did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1

verificationMethod selected: {
  "type": "Ed25519VerificationKey2018",
  "id": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM#key-1",
  "controller": "did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM",
  "publicKeyBase58": "HdXo5kegxgPze3tAw6QYU7vvJg4gbxztqSidt8LsB6eS"
}
verificationMethod public key as hex: f716b82bd54ecddb908d804cd031512eb5cb3a5454d0c1024e30b04a88251cff

Attempting to match TLSA record: 3 1 1 8a9e530067d29d399ac190006344822889385e91d70ba249c98dcd669d849460...
Matching type: 1, matching SHA-256 hash of public key...
Key SHA256 hash: 8a9e530067d29d399ac190006344822889385e91d70ba249c98dcd669d849460
Successfull key match. TLSA verification successful.

Attempting to match TLSA record: 3 1 0 f716b82bd54ecddb908d804cd031512eb5cb3a5454d0c1024e30b04a88251cff...
Matching type: 0, matching unhashed public key...
Successfull key match. TLSA verification successful.
```

PUTTING URI AND TLSA RECORDS TOGETHER

By leveraging URI and TLSA records, we create a bi-directional and cryptographic linkage of the DIDs in whatever distributed ledger they reside on to the publicly resolvable DNS and domain they are associated with.

- **Or in other words;**
  - The Verifier at the bar in Japan can confirm that **Chioma's** digital driver's license was issued by did:sov:danube:XWfvq6uyAjBUbg4hBBP3vM, which is under the control of tr-demo.ciralabs.ca, meaning the driver's license was ultimately issued by tr-demo.ciralabs.ca.
  - It is important to note that these technologies can work completely independently of one another or in tandem. The Verifier can leverage just the DNS records, just the DID and distributed ledger (blockchain), or a combination of both to whatever degree they deem necessary.

WWW.CIRA.CA

The Trust Registry Part…

TRUST REGISTRY IN DNS

The bar in Japan has confirmed the validity of **Chioma's** credential, but now they need to verify the authority of its issuer, tr-demo.ciralabs.ca.

By leveraging the same URI and TLSA records we used before to verify the credential, we can also verify the issuer's membership in a Trust Registry.

- **tr-demo.ciralabs.ca hosts a URI record, pointing to a trust registry it is claiming membership in.**
  - Ex: URI Record
    - _tr.tr-demo.ciralabs.ca 1 0 "trustregistry.ca"
  - This record indicates that tr-demo.ciralabs.ca is claiming it is registered under trustregistry.ca, the Apex Trust Registry for Canada.

```
$ dig _tr.tr-demo.ciralabs.ca URI
```

TRUST REGISTRY IN DNS

The bar in Japan can now reference that the public key used to issue **Chioma's** driver's licence is hosted by trustregistry.ca.

The verifier can check trustregistry.ca for proof of tr-demo.ciralabs.ca's membership. We use TLSA records to provide the cryptographic proof of that association by hosting the public key of demo.ciralabs.ca in trustregistry.ca's zone.

- **Ex: TLSA Record**
  - tr-demo.ciralabs.ca._tr.trustregistry.ca 3 1 0 (public key un-hashed)
  - tr-demo.ciralabs.ca._tr.trustregistry.ca 3 1 1 (sha256 hash of public key)
- **Please note that public key in this case is the same one we used earlier to verify the credential issued by tr-demo.ciralabs.ca to Chioma.**

```
$ dig tr-demo.ciralabs.ca._tr.trustregistry.ca  TLSA
```

NATIONAL TRUST REGISTRIES IN COLLABORATION

By having both Canada and Japan's Apex Trust Registries recognize one another, the chain of trust now spans across the globe and is publicly accessible and resolvable.

- **But how does the bar know trustregistry.ca is valid? trustregistry.ca would hold a reference to trustregistry.jp, the Apex Trust Registry for Japan, and vice versa. That way the bar can reference their own country's Apex Trust Registry to navigate the authority and issuance of credentials from another nation.**
- **This allows the root of trust to remain decentralized but still be easy to navigate and reference.**
- **Ex: URI Records**
  - canada._tr.trustregistry.jp 1 0 "trustregistry.ca"
  - japan._tr.trustregistry.ca 1 0 "trustregistry.jp"
- **Those trust relationships could be bolstered through the association of PKI (via TLSA records and public keys) or association with DIDs via URI records as required.**

cira

Thank You

https://www.cira.ca/labs

cira

Northern Block

Digital Identity LABORATORY