Gaia-X 101

Ewann Gavard – Tech lead – Gaia-X AISBL CTO Team

Summary

TECHNOLOGIES AND STANDARD USED IN GAIA-X

GAIA-X SPECIFICATIONS & DOCUMENTS

CURRENT STATE OF THE IMPLEMENTATION

WRAP EVERYTHING TO GET GAIA-X CREDENTIALS

Let's start outside of Gaia-X

Verifiable Credentials

JSON-LD

JsonWebSignature

DID/DID Web

SHACL

Verifiable Credentials

Represents any form of credential, permits, license

Used in Gaia-X to represent everything, companies, people, services

VCs are cryptographically signed by the issuer, allowing to check data tampering and issuer's legitimacy

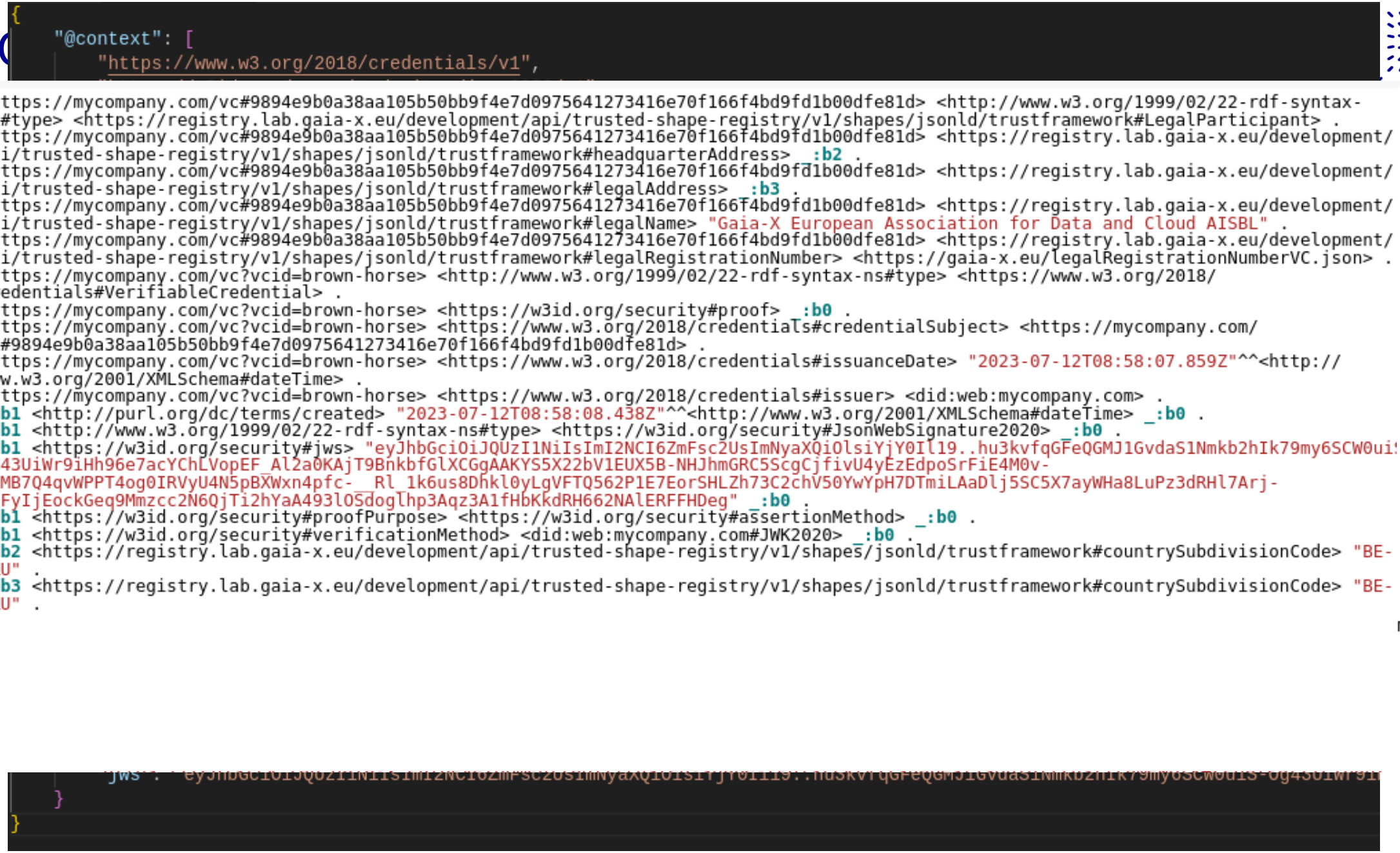
VCs are written using JSON-LD, allowing to intricate and bind credentials and claims

JSO

Co

Lir

Re


{
"@context": [
"https://www.w3.org/2018/credentials/v1",
<https://mycompany.com/vc#9894e9b0a38aa105b50bb9f4e7d0975641273416e70f166f4bd9fd1b00dfe81d> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <https://registry.lab.gaia-x.eu/development/api/trusted-shape-registry/v1/shapes/jsonld/trustframework#LegalParticipant> .
<https://mycompany.com/vc#9894e9b0a38aa105b50bb9f4e7d0975641273416e70f166f4bd9fd1b00dfe81d> <https://registry.lab.gaia-x.eu/development/api/trusted-shape-registry/v1/shapes/jsonld/trustframework#headquarterAddress> _:b2 .
<https://mycompany.com/vc#9894e9b0a38aa105b50bb9f4e7d0975641273416e70f166f4bd9fd1b00dfe81d> <https://registry.lab.gaia-x.eu/development/api/trusted-shape-registry/v1/shapes/jsonld/trustframework#legalAddress> _:b3 .
<https://mycompany.com/vc#9894e9b0a38aa105b50bb9f4e7d0975641273416e70f166f4bd9fd1b00dfe81d> <https://registry.lab.gaia-x.eu/development/api/trusted-shape-registry/v1/shapes/jsonld/trustframework#legalName> "Gaia-X European Association for Data and Cloud AISBL" .
<https://mycompany.com/vc#9894e9b0a38aa105b50bb9f4e7d0975641273416e70f166f4bd9fd1b00dfe81d> <https://registry.lab.gaia-x.eu/development/api/trusted-shape-registry/v1/shapes/jsonld/trustframework#legalRegistrationNumber> <https://gaia-x.eu/legalRegistrationNumberVC.json> .
<https://mycompany.com/vc?vcid=brown-horse> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <https://www.w3.org/2018/credentials#VerifiableCredential> .
<https://mycompany.com/vc?vcid=brown-horse> <https://w3id.org/security#proof> _:b0 .
<https://mycompany.com/vc?vcid=brown-horse> <https://www.w3.org/2018/credentials#credentialSubject> <https://mycompany.com/vc#9894e9b0a38aa105b50bb9f4e7d0975641273416e70f166f4bd9fd1b00dfe81d> .
<https://mycompany.com/vc?vcid=brown-horse> <https://www.w3.org/2018/credentials#issuanceDate> "2023-07-12T08:58:07.859Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<https://mycompany.com/vc?vcid=brown-horse> <https://www.w3.org/2018/credentials#issuer> <did:web:mycompany.com> .
_:b1 <http://purl.org/dc/terms/created> "2023-07-12T08:58:08.438Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> _:b0 .
_:b1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <https://w3id.org/security#JsonWebSignature2020> _:b0 .
_:b1 <https://w3id.org/security#jws> "eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..hu3kvfqGFeQGMJ1GvdaS1Nmkb2hIk79my6SCW0ui
Og43UiWr9iHh96e7acYChLVopEF_Al2a0KAjT9BnkbfGlXCGgAAKYS5X22bV1EUX5B-NHJhmGRC5ScgCjfivU4yEzEdpoSrFiE4M0v-
NbMB7Q4qvWPPT4og0IRVyU4N5pBXWxn4pfc-__Rl_1k6us8Dhkl0yLgVFTQ562P1E7EorSHLZh73C2chV50YwYpH7DTmiLAaDlj5SC5X7ayWHa8LuPz3dRHl7Arj-
sdFyIjEockGeq9Mmzcc2N6QjTi2hYaA493lOSdoglhp3Aqz3A1fHbKkdRH662NAlERFFHDeg" _:b0 .
_:b1 <https://w3id.org/security#proofPurpose> <https://w3id.org/security#assertionMethod> _:b0 .
_:b1 <https://w3id.org/security#verificationMethod> <did:web:mycompany.com#JWK2020> _:b0 .
_:b2 <https://registry.lab.gaia-x.eu/development/api/trusted-shape-registry/v1/shapes/jsonld/trustframework#countrySubdivisionCode> "BE-BRU" .
_:b3 <https://registry.lab.gaia-x.eu/development/api/trusted-shape-registry/v1/shapes/jsonld/trustframework#countrySubdivisionCode> "BE-BRU" .
}

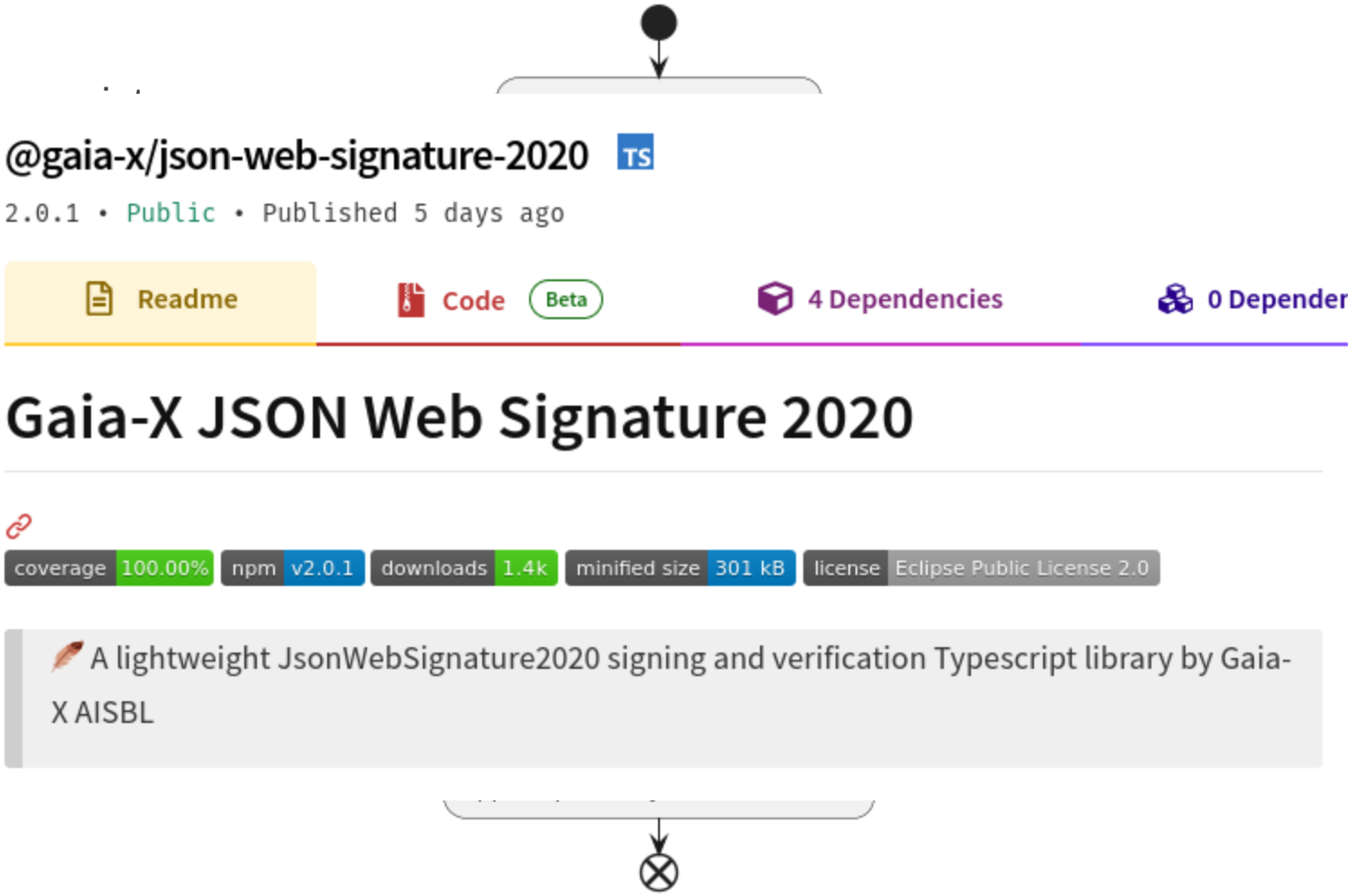
JWS: JsonWebSignature

Allow to ensure dat

References issuer's trustworthiness

Gaia-X uses "comp

Two marshalling co



DID: Decentralized Id

Self-declared and s

Contains cryptograp

One specification u

Examples:

- did:web:compliar x.eu/v1/did.json
- did:web:bakeup.

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/jws-2020/v1"
```

@gaia-x/did-web-generator

1.0.1 • Public • Published a month ago

Readme | Code Beta | 2 Dependencies | 0 Dependent

Gaia-X AISBL DID Generator Library

This library allows you to generate a ready to use DID.

It uses your certificate to generate it, and thus relies on several x509/crypto libraries to work.

Usage

```
import {createDidDocument} from '@gaia-x/did-web-generator'
//...
function getDid(){
    return createDidDocument("https://mycompanydomain.com", "x509Certifica
}
```

```
    "type": "LinkedDomains",
    "serviceEndpoint": "https://bakeup.io/service.json"
  }]
}
```

SHACL

Know as shapes in o

Validates RDF struct

Similar to XSD for XM

Not all constraints ca
implemented in code

```
gx:LegalParticipantShape
    a sh:NodeShape ;
    sh:targetClass gx:LegalParticipant ;
    sh:property
        [
            sh:path gx:legalRegistrationNumber ;
            sh:node gx:legalRegistrationNumberShape ;
            sh:minCount 1 ;
        ],
        [
            sh:path gx:parentOrganization ;
            sh:node gx:LegalParticipantShape ;
        ],
        [
            sh:path gx:subOrganization ;
            sh:node gx:LegalParticipantShape ;
        ],
        [
            sh:path gx:headquarterAddress ;
            sh:minCount 1 ;
            sh:node gx:PostalAddressShape ;
        ],
        [
            sh:path gx:legalAddress ;
            sh:minCount 1 ;
            sh:node gx:PostalAddressShape ;
        ] .

gx:legalRegistrationNumberShape
    a sh:NodeShape ;
    sh:targetClass gx:legalRegistrationNumber ;
    sh:message "At least one of taxID, vatID, EUID, EORI or leiCode must be defined." ;
    sh:property
        [
            sh:path gx:taxID ;
            sh:datatype xsd:string ;
            sh:minLength 3 ;
        ] ;
    sh:property
        [
            sh:path gx:EUID ;
            sh:datatype xsd:string ;
            sh:minLength 3 ;
        ] ;
    sh:property
```

Gaia-X

Identity &

Policy Rul

Architectu

Data Exch

For Tagus

All availab

`ions`			Ecosystem as defined below

Example of T&C signed by the issuer

5.2 Legal person

For legal person the attributes are

Attribute	Cardinality	Trust Anchor	Comment
`registrationNumber`	1	registrationNumberIssuer	Country's registration number, which identifies one specific entity.
`headquartersAddress.countryCode`	1	State	Physical location of the headquarters in ISO 3166-2 alpha2, alpha-3 or numeric format.
`legalAddress.countryCode`	1	State	Physical location of legal registration in ISO 3166-2 alpha2, alpha-3 or numeric format.
`parentOrganization[]`	0..*	State	A list of direct `participant` that this entity is a subOrganization of, if any.
`subOrganization[]`	0..*	State	A list of direct `participant` with a legal mandate on this entity, e.g., as a subsidiary.

Gaia-X specifications in a slide

Everything is described using VerifiableCredentials in JSON-LD

Each issuer has to provide signed terms and conditions (TL;DR be nice)

Participant has to provide a Legal Registration Number issued by an accredited notary

On production, participant must use an EV-SSL or eIDAS certificate to sign their credentials

Few providers are accredited Gaia-X compliance issuers, more to come.

Having your credentials validated by the engine will result in a Gaia-X compliance VerifiableCredential

State of the implementation

1st production-ready release: Tagus (v1)

- Trust framework 22.10 fully implemented (Participants, ServiceOfferings, Resources)
- PRLD 22.11 partially implementated: Service Offering Labels level 1

A bit of tooling provided:

- Wizard
- DID Library
- Signature Library
- DID Validator Library

Running endpoints:

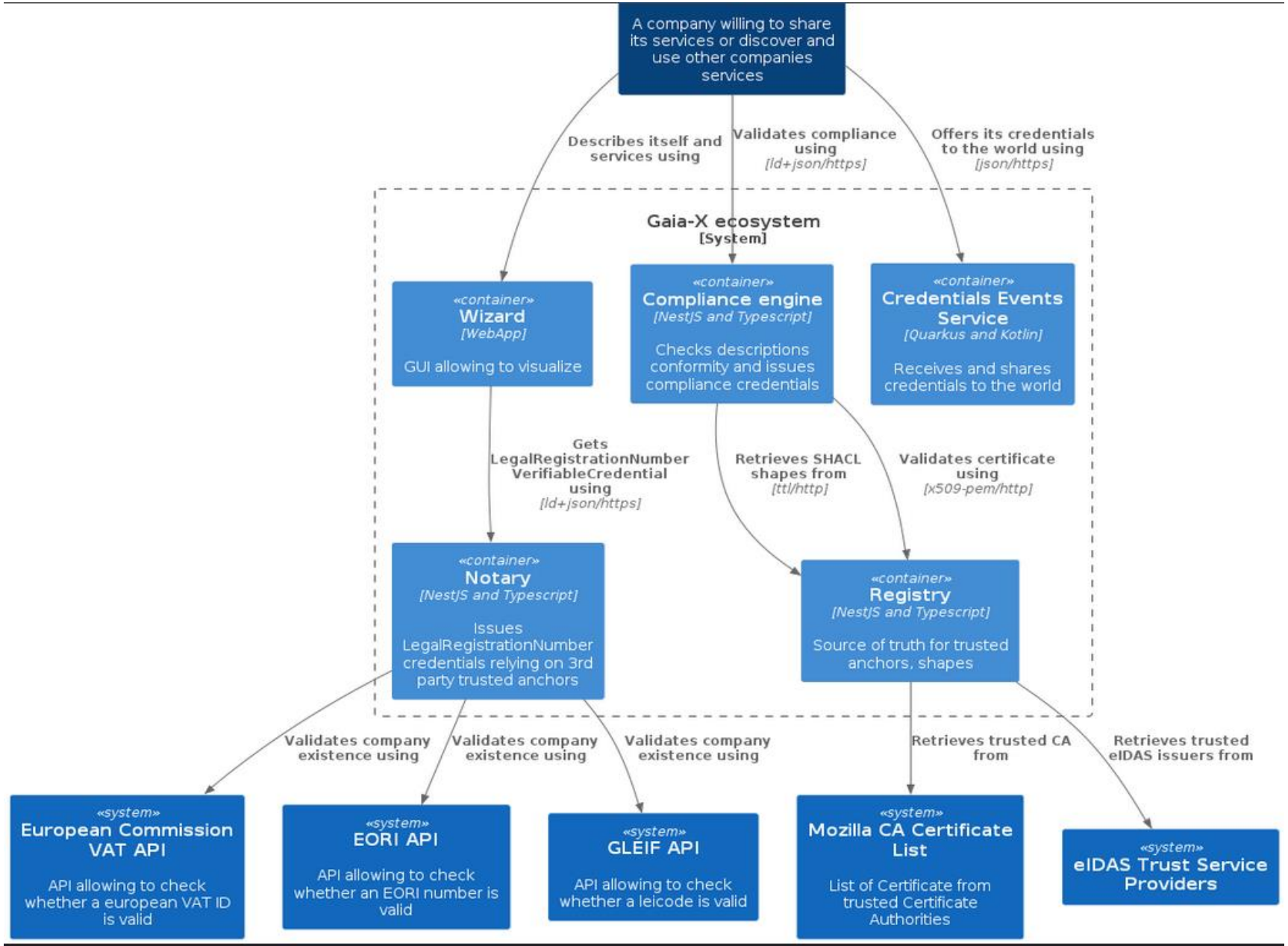
- https://docs.gaia-x.eu/framework/?tab=clearing-house

State of the implementation

Some mistakes exist:

- JSON-LD namespace complicated and referring to development in the URL
- Shapes are not perfectly aligned with specs (LegalParticipant != LegalPerson)
- Types need to be in credentialSubject to be valid (!55)

Software architecture



Wizard demo

Time to code

And after ?

- Let the world know & consume other people info
- Join a dataspace if you want
- Build your own catalog or use an existing one
- Participate in the specifications to move forward

gaia-x

Thank you for your attention

IPFS for the Gaia-x registry

A first step in distributing provably unmodified compliance artefacts

Why IPFS

What is IPFS?

- IPFS (InterPlanetary File System) is a peer-to-peer protocol designed for storing and sharing data in a distributed file system.
- Based on the Kademlia Distributed Hash Table
- Unlike traditional web protocols that locate data based on servers (location-based addressing), IPFS locates data based on its content (content-based addressing). This is using sha-256 content ids (CIDs).

Advantages of IPFS:

- **Decentralization:** *Eliminates* reliance on centralized servers, reducing points of failure and increasing resilience against censorship and outages.
- **Efficiency and Performance:** Fetches files from the nearest node rather than a central server, *potentially* speeding up web content delivery.
- **Permanence:** Content addressing ensures that data **cannot be tampered**.
- **Interoperability:** Facilitates an open web where applications can communicate more freely, enhancing data sharing and collaboration across different platforms.
- **Cost Reduction:** Can reduce hosting and bandwidth costs by distributing the load across multiple nodes.

IPFS in the GXDCH context

Potential cons of using IPFS for storing registry artefacts in practice:

- In order to really eliminate the points of failure and increase resilience, we will need actors (GXDCH operators) to participate by pining and seeding our artefacts.
- IPFS can be slow if we have a low number of peers hosting and sharing our content.
- We’re introducing a new external dependancy in our registry service deployment, in the form of Kubo, although since both are communicating through the standard RPC API, any compliant IPFS implementation (for example, an externally subscribed managed IPFS service) would work.
- On small text based files like ours (trust framework shapes, revocation lists, trusted gxdch list, etc.) the cost of exposing an IPFS node IP/dns through Load Balancers will probably be greater than centralized hosting bandwidth costs.

What will and won't change

Impacted artefacts:

- Trust framework shapes and schemas;
- Issuers revocation list;
- Trusted GXDCHs list.

Current release (TAGUS v1):

Those files were previously either bundled statically on each registry instance, or served from our gitlab. They were then served through the endpoints provided by the registry services.

Current implementation (on our development branch):

Those files are retrieved (both on startup and at regular intervals at runtime) from IPFS by registry instance. They are then still served through the endpoints provided by the registry services.

Next, LOIRE release:

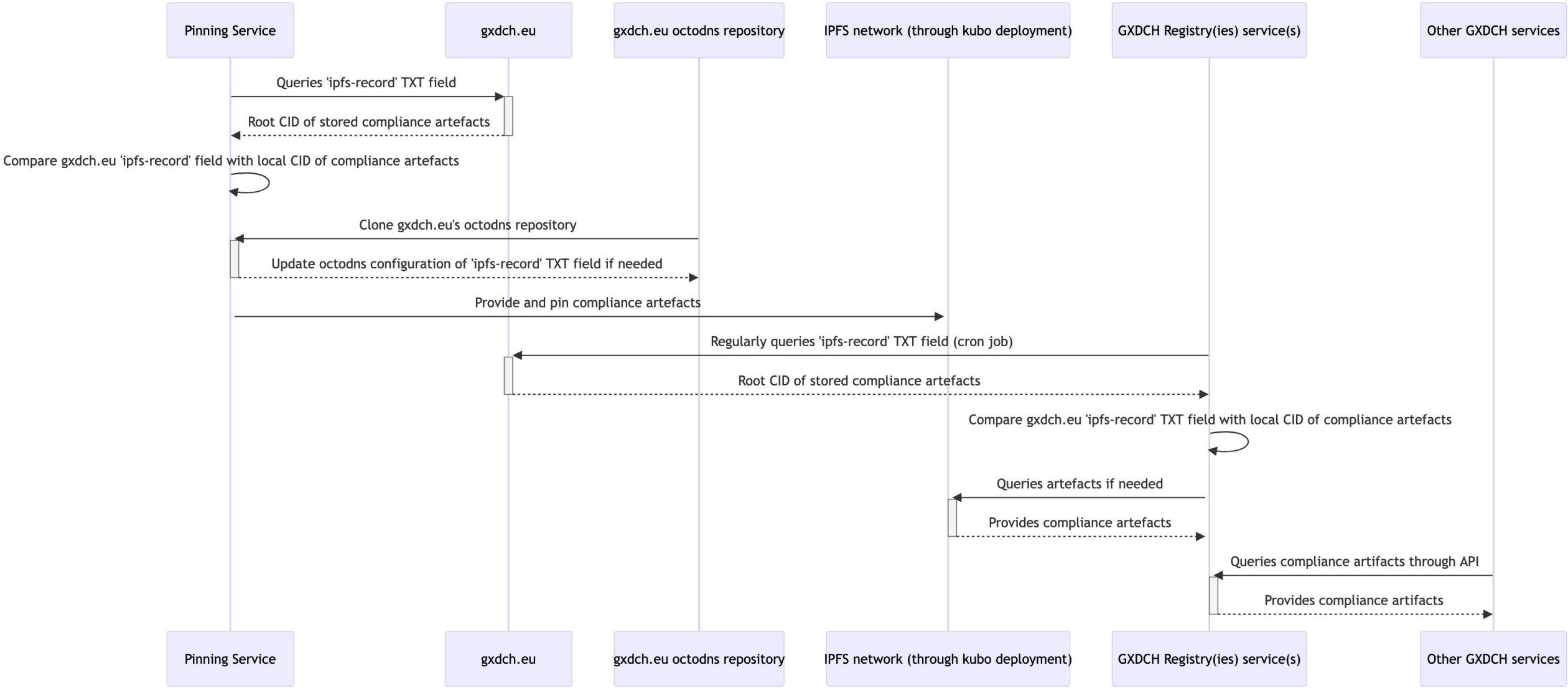
Current implementation and Gaia-x Trust Anchors lists (described in a standardized format, such as ETSI 119 612).

A « new » Gaia-x IPFS service

IPFS pinning service:

- Is a new service managing our compliance artefacts through IPFS pinning and seeding (https://gitlab.com/gaia-x/lab/compliance/gx-ipfs-pinning);
- Is the new place where we'll be updating & unit testing the trust framework shapes;
- Advertise the root CID of our artefacts to registry services instances by leveraging our octodns configuration (https://gitlab.com/gaia-x/lab/octodns);
- Will not need to be hosted by GXDCHs

A deeper look



Looking forward

« Weakly » distributed, not decentralized:

- Although we're leveraging technologies that are distributed like dns, and also decentralized like IPFS we're still updating those in a dictatorial and permissioned fashion. Some of those artefacts could probably benefit from more decentralization (ie: trust anchors & revocation lists)
- IPFS as a protocol does not incentivize sharing by non involved parties. This means unless we use paid services we'll only be as distributed as the number of seeding GXDCHs. It's still ok regarding the size and bandwidth profile of our content, but could be better. This should not be an issue as those requirements would automatically scale with GXDCHs deployments.
- Other parts of our infrastructure are still fully centralized, notably the Credentials Event Service (https://gitlab.com/gaia-x/lab/credentials-events-service). We're looking into ways to improve this.

Solutions?:

- DAO, Governance & voting mechanisms: There are a lot of options allowed by smart contracts that could allow us to update artefacts in a decentralized fashion. Those range from simple multisigs contracts to advanced and customisable voting systems leveraging non transferable voting tokens.
- Much like NFTs, a storage solution combining txdata inscribed on a blockchain through calling a smart contract, pointing to IPFS content, could also be imagined in the future to decentralize the CES. This would need to be cheap and fast but there are solutions (L2s, application specific chains, etc.).

Our community is encouraged to provide feedback, ideas or concerns on those points.

Get more info or contribute

- Slack : https://join.slack.com/t/gaia-xworkspace/shared_invite/zt-2dr9bj9hx-IM7nwpv3DABR02UVhgQnzw
- Mailing lists: https://list.gaia-x.eu/postorius/lists/oss-community.list.gaia-x.eu/
- OSS Community call, every thursday, 9am CEST
- Gitlab releases, issues, merge requests: https://gitlab.com/gaia-x/lab

Thank you

Policy Reasoning using ODRL Profile for Attribute based access/usage control using Verifiable Credential claims

Yassir SELLAMI, Software Engineer, Gaia-X

Enforceable Policies using Verifiable Credentials

- Using Verifiable Credential Claims within a policy definition
- Use cases
- ODRL (Open Digital Rights Language)
- Policy definition example
- Reasoning Engine
- Using Ontologies
- ODRL Profile
- What’s next ?

Using Verifiable Credential Claims within a policy definition

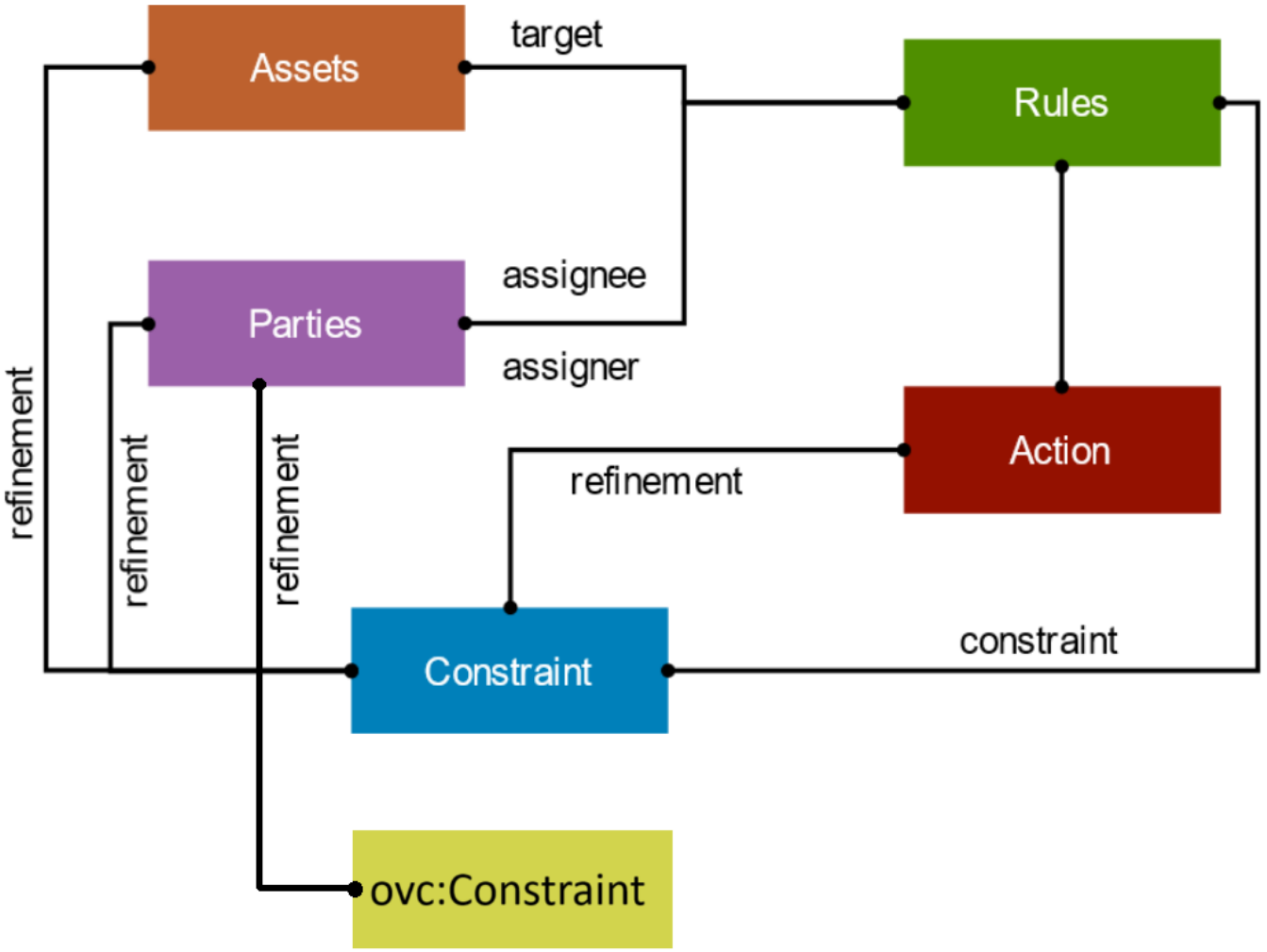
- The Open Digital Rights Language (ODRL) is a policy expression language that provides a flexible and interoperable information model for representing statements about the usage of content and services.
  - But…
- There is no easy way to verify and assess an access request in a trustworthy verifiable manner from that same policy, especially using a software component.
  - So why not leverage Verifiable Credentials inside an ODRL Policy !

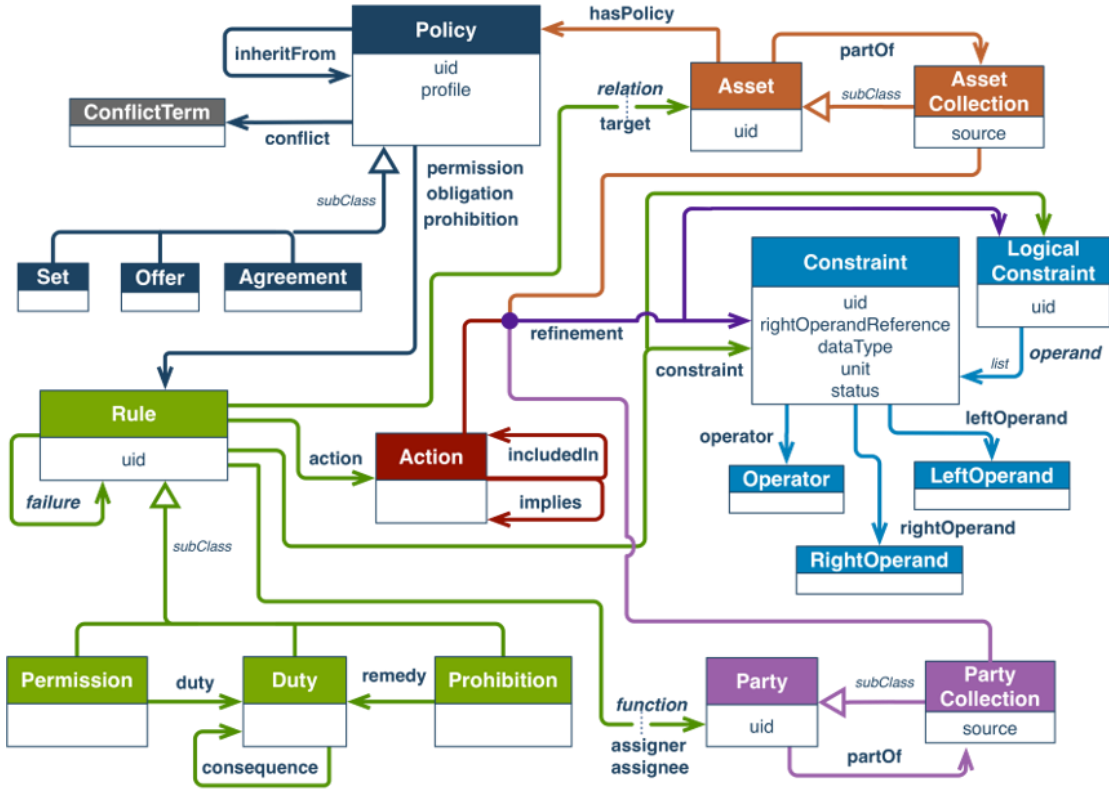
Use cases

gaia-x

- A catalog only accepting Gaia-x compliant participants
- A provider communicating only with valid Legal Participant VC holders
- A natural personal giving consent to a PDI
- A provider giving access to only certain countries/regions
- A company sharing data with only partner companies
- A company giving access to employees (Right delegation)
- Company legally appointed represetative (LEAR)
- The government giving access to only certified profesionnals
- A catalog displaying more features for certain user

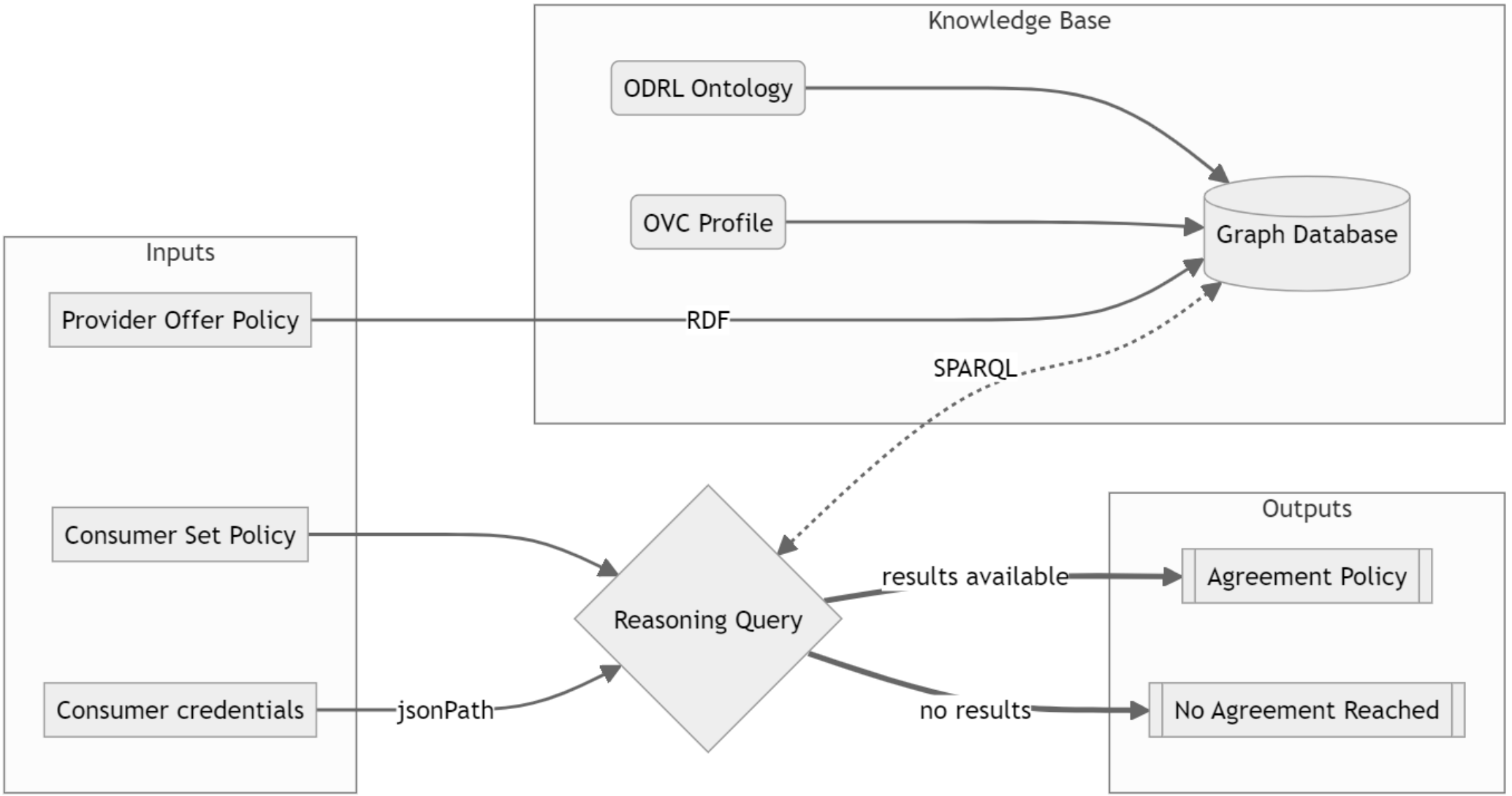
ODRL


Assets
target
Rules
Parties
assignee
assigner
Action
refinement
refinement
refinement
refinement
Constraint
constraint
ovc:Constraint

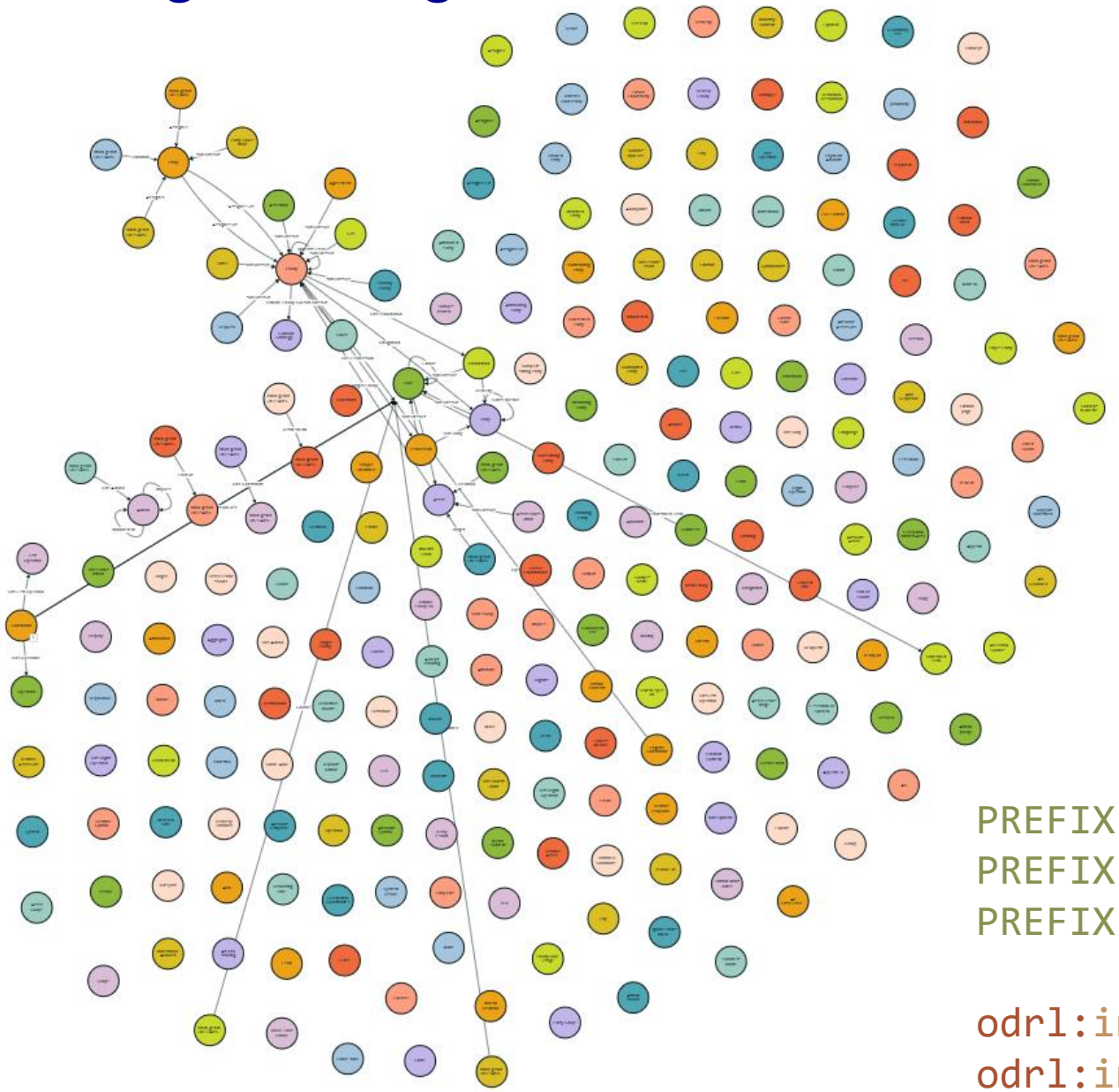

Policy
uid
profile
hasPolicy
inheritFrom
ConflictTerm
conflict
partOf
relation
target
Asset
uid
subClass
Asset Collection
source
subClass
permission
obligation
prohibition
Set
Offer
Agreement
Constraint
uid
rightOperandReference
dataType
unit
status
Logical Constraint
uid
refinement
constraint
list
operand
Rule
uid
failure
action
Action
includedIn
implies
operator
leftOperand
Operator
LeftOperand
rightOperand
RightOperand
subClass
Permission
duty
Duty
remedy
Prohibition
consequence
function
assigner
assignee
Party
uid
subClass
Party Collection
source
partOf

```
{
  "@context": [
    "http://www.w3.org/ns/odrl.jsonld",
    {
      "gx": "https://registry.lab.gaia-x.eu/development/api/trusted-shape-registry/v1/jsonld/trustframework#"
    },
    {
      "ovc": "https://w3id.org/ovc/1/"
    }
  ],
  "@type": "Offer",
  "uid": "http://example.com/policy/123",
  "profile": "https://w3id.org/ovc/1/",
  "permission": [
    {
      "@type": "Permission",
      "target": "http://example.com/asset/456",
      "action": "http://www.w3.org/ns/odrl/2/play",
      "assigner": "http://example.com/provider",
      "assignee": {
        "ovc:constraint": [
          {
            "ovc:leftOperand": "$.credentialSubject.gx:legalAddress.gx:countrySubdivisionCode",
            "operator": "http://www.w3.org/ns/odrl/2/in",
            "rightOperand": [
              "FR-HDF",
              "BE-BRU"
            ],
            "ovc:credentialSubjectType": "gx:LegalParticipant"
          }
        ]
      }
    }
```

Policy reasoning engine

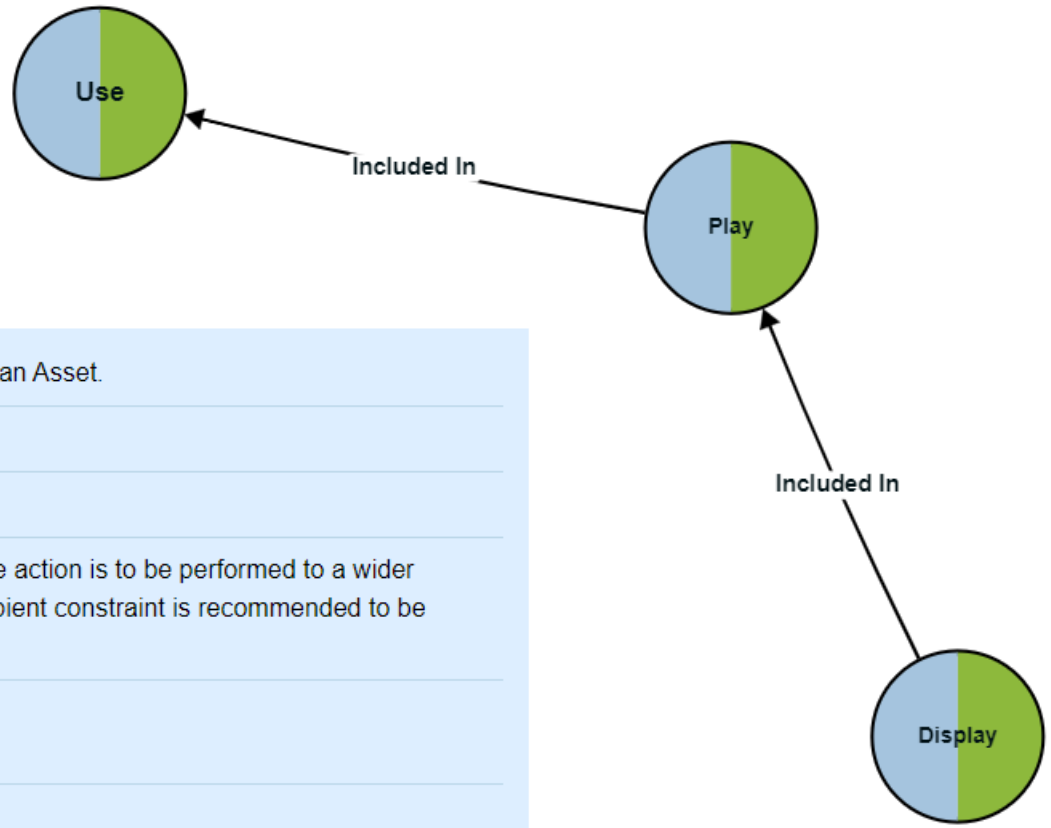


Using Ontologies



```
PREFIX odrl: <http://www.w3.org/ns/odrl/2/>
PREFIX rdfs: <http://www.w3.org/2000/01/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

odrl:includedIn a owl:TransitiveProperty .
odrl:includedIn a rdfs:subClassOf .
```



4.4.32 Play

Definition:	To create a sequential and transient rendition of an Asset.
Label:	Play
Identifier:	http://www.w3.org/ns/odrl/2/play
Note:	For example, to play a video or audio track. If the action is to be performed to a wider audience than just the Assignees, then the Recipient constraint is recommended to be used.
Included By:	display
Included In:	use
Class:	Action

ODRL Profile

- Compatible with the base ODRL Information Model
- Would rely on ODRL, VC and JSONPath base specification
- Give clear definition and syntax for custom constraint values
- Custom value formats for ovc:`leftOperand` and ovc:credentialSubjectType
  - ovc:leftOperand to contain a JSONPath for the intended attribute to evaluate
  - ovc:credentialSubjectType refer to the credential type

What’s next

- Provide a Typescript library
- Provide a Java library
- Component within Clearing House
- Integration with EDC

Useful links

- The ODRL Profile
  - https://gitlab.com/gaia-x/gaia-x-community/open-source-community
  - https://gitlab.com/gaia-x/
- Gaia-x Slack
- OSS Community Call & newsletter
- Our Jira Backlog