Trust framework - Gaia-X Trust Framework - DRAFT version 18803bf1

Table of Contents

1. Gaia-X Trust Framework

For Gaia-X to ensure a higher and unprecedented level of trust in digital platforms, we need to make trust an easy to understand and adopted principle. For this reason, Gaia-X developed a Trust Framework â€“ formerly known as Gaia-X Compliance and Labelling Framework that safeguards data protection, transparency, security, portability, and flexibility for the ecosystem as well as sovereignty and European Control.

The Trust Framework is the set of rules that define the minimum baseline to be part of the Gaia-X Ecosystem. Those rules ensure a common governance and the basic levels of interoperability across individual ecosystems while letting the users in full control of their choices.[1]

In other words, the Gaia-X Ecosystem is the virtual set of participants and service offerings following the requirements from the Gaia-X Trust Framework.

The Trust Framework uses verifiable credentials and linked data representation to build a FAIR[2] knowledge graph of verifiable claims from which additional trust and composability indexes can be automatically computed.

The set of computable rules known as compliance process is automated and versioned. It means that this document will also be versioned.

1.1 Trust Framework scope

Those rules apply to all Gaia-X Self-Descriptions and there is a Self-Description for all the entities defined as part of the Gaia-X Conceptual model described in the Gaia-X Architecture document:

This list mainly comprises:

- Participant including Consumer, Federator, Provider
- Service Offering
- Resource

1.1.1 Gaia-X Labels

The Labelling Framework itself is further detailed and translated into concrete criteria and measures in the Gaia-X Labelling Criteria document 22.04.

Framework	Notes
Trust Framework	Compulsory set of rules to comply with in other to be part of the Gaia-X Ecosystem. Individual ecosystems can extend those rules.

Framework	Notes
Labelling Framework	Optional set of criteria for Service Offerings.

1.2 Gaia-X Self-Description

Gaia-X Self-Descriptions are:

- machine readable texts
- cryptographically signed, preventing tampering with its content
- following the Linked Data principles[3] to describe attributes

The format is following the W3C Verifiable Credentials Data Model.

1.3 Gaia-X Trust Framework

There are 4 types of rules:

- serialization format and syntax.
- cryptographic signature validation and validation of the keypair associated identity.
- attribute value consistency.
- attribute veracity verification.

2. Trust anchors

For compliance, Trust anchors are Gaia-X endorsed entities responsible to manage certificate to sign claims.

To be compliant with the Gaia-X Trust Framework, all keypairs used to sign claims must have at least one of the Trust Anchors in their certificate chain.

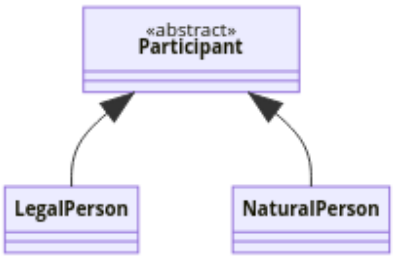
At any point in time, the list of valid Trust Anchors is stored in the Gaia-X Registry.

2.1 List of defined trust anchors

Name	Defined as
State	The Trust Service Providers (TSP) must be a state validated identity issuer. - For `participant`, if the `legalAddress.country` is in EEA, the TSP must be eiDAS compliant. - Until the end of 2022-Q1, to ease the onboarding and adoption this framework, DV SSL can also be used. - Gaia-X Association is also a valid TSP for Gaia-X Association members.
eiDAS	Issuers of Qualified Certificate for Electronic Signature as defined in eIDAS Regulation (EU) No 910/2014 (homepage: https://esignature.ec.europa.eu/efda/tl-browser/#/screen/home) (machine: https://ec.europa.eu/tools/lotl/eu-lotl.xml)
DV SSL	Domain Validated (DV) Secure Sockets Layer (SSL) certificate issuers are considered to be temporarily valid Trust Service Providers. (homepage: https://wiki.mozilla.org/CA/Included_Certificates) (machine: https://ccadb-public.secure.force.com/mozilla/IncludedCACertificateReportPEMCSV)
Gaia-X	*To be defined after 2022Q1.*
EDPB CoC	List of Monitoring Bodies accreditated to the Code of Conduct approved by the EDBP (list of EDBP's CoC: https://edpb.europa.eu/our-work-tools/documents/our-documents_fr?f%5B0%5D=all_publication_type%3A61&f%5B1%5D=all_topics%3A125)

3. Participant

A Participant is a Legal Person or Natural Person, which is identified, onboarded and has a Gaia-X Self-Description. Instances of Participant neither being a legal nor a natural person are prohibited.



The Architecture Document defines three roles a Participant can have within the Gaia-X Ecosystem (Provider, Consumer, and Federator). These are not yet part of Trust Framework and are to be defined in future releases.

3.1 Legal person

For legal person the attributes are

Attribute	Cardinality	Trust Anchor	Comment
`registrationNumber`	1	State	Country's registration number, which identifies one specific entity.
`headquarterAddress` . `countryCode`	1	State	Physical location of head quarter 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.
`parentOrganisation[]`	0..*	State	A list of direct `participant` that this entity is a subOrganization of, if any.
`subOrganisation[]`	0..*	State	

Attribute	Cardinality	Trust Anchor	Comment
			A list of direct `participant` with an legal mandate on this entity, e.g., as a subsidiary.
`termsAndConditions`	1	State	SHA512 of the Generic Terms and Conditions for Gaia-X Ecosystem as defined below

3.1.1 registrationNumber

The list of valid entity `registrationNumber` type are described below:

Attribute	Comment
`local`	the state issued company number
`EUID`	the European Unique Identifier (EUID) for business located in the European Economic Area, Iceland, Liechtenstein or Norway and registered in the Business Registers Interconnection System (BRIS). This number can be found via the EU Business registers portal
`EORI`	the Economic Operators Registration and Identification number (EORI).
`vatID`	the VAT identification number.
`leiCode`	Unique LEI number as defined by https://www.gleif.org.

Consistency rules

- if `EORI` is provided, the number will be verified against the European Commission API.
- if `leiCode` is provided, the number will be verified against the Global Legal Entity Identifier (GLEIF) API
- if `local` is provided, the number will be verified with `headquarterAddress.countryCode` against the OpenCorporate API.
- if `vatID` is provided and `headquarterAddress.countryCode` belongs to the European member states or North Ireland, the number will be checked against the VAT Information Exchange System (VIES) API
- if several numbers are provided, the information provided by each number must be consistent.

3.1.2 Gaia-X Ecosystem Terms and Conditions

The PARTICIPANT signing the Self-Description agrees as follows:
- to update its descriptions about any changes, be it technical, organisational, or legal - especially but not lim
- wrongful statements will reflect a breach of contract and may cumulate to unfair competitive behaviour.
- in cases of systematic and deliberate misrepresentations, Gaia-X Association is, without prejudice to claims

Alongside, the PARTICIPANT signing the Self-Description is aware and accepts that:
- the SERVICE OFFERING will be delisted where Gaia-X Association becomes aware of any inaccurate statem

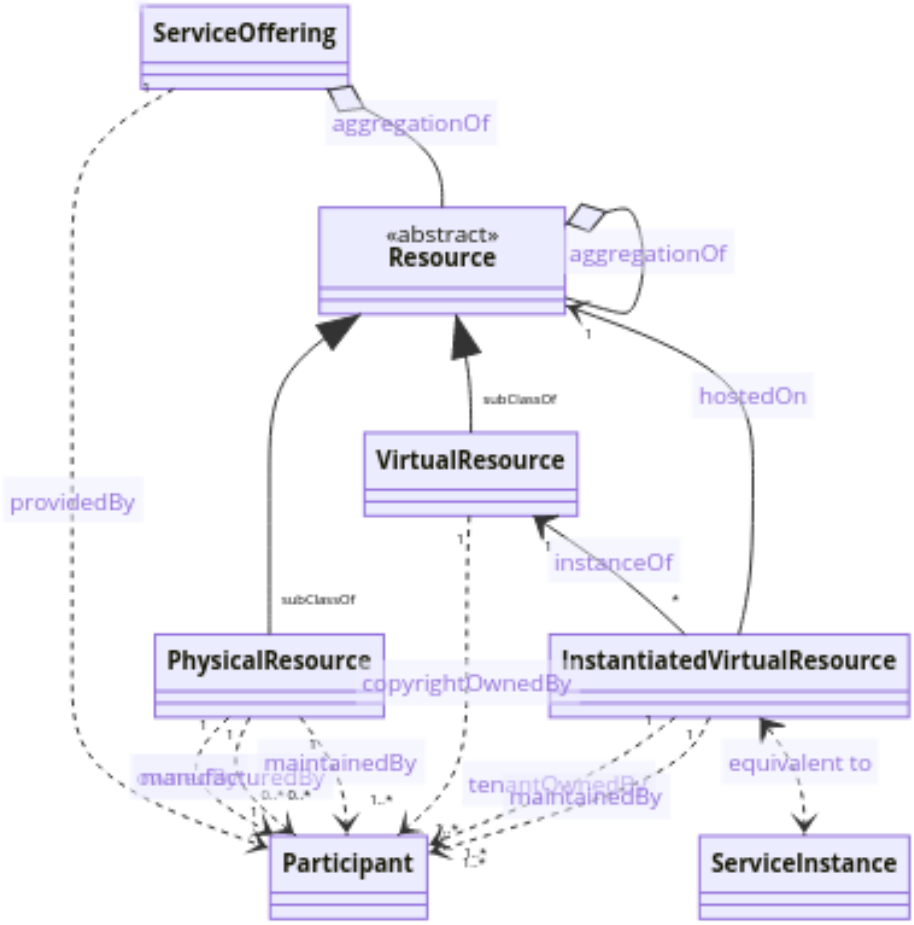
3.2 Natural person

To be defined in a future release.

4. Services & Resources

Here is the main model for service composition, also included in the Gaia-X Architecture document.

A `Service Offering` can be associated with other `Service Offering`s.



4.1 Service offering

This is the generic format for all service offerings

Attribute	Card.	Trust Anchor	Comment
`providedBy`	1	State	a resolvable link to the `participant` self-description providing the service

Attribute	Card.	Trust Anchor	Comment
`aggregationOf[]`	0..*	State	a resolvable link to the `resources` self-description related to the service and that can exist independently of it.
`dependsOn[]`	0..*	State	a resolvable link to the `service offering` self-description related to the service and that can exist independently of it.
`termsAndConditions[]`	1..*	State	a resolvable link to the Terms and Conditions appling to that service.
`policies[]`	0..*	State	a list of `policy` expressed using a DSL (e.g., Rego or ODRL)
`dataProtectionRegime[]`	0..*	State	a list of data protection regime from the list available below
`dataExport[]`	1..*	State	list of methods to export data out of the service

termsAndConditions structure

Attribute	Card.	Trust Anchor	Comment
`URL`	1	State	a resolvable link to document
`hash`	1	State	sha256 hash of the above document.

dataExport structure

The purpose is to enable the participant ordering the service to assess the feasability to export personal and non-personal data out of the service.

Attribute	Card.	Trust Anchor	Comment
`requestType`	1	State	the mean to request data retreival: `API` , `email` , `webform` , `unregisteredLetter` , `registeredLetter` , `supportCenter`
`accessType`	1	State	type of data support: `digital` , `physical`
`formatType`	1	State	type of Media Types (formerly known as MIME types) as defined by the IANA.

Data Protection Regime

To enable interoperability and automate policy negociation, the Gaia-X association strongly advocates to use the list of data protection regimes listed in the Gaia-X Registry

Non exclusive list of Data Protection regimes:

- `GDPR2016` : General Data Protection Regulation / EEA
- `LGPD2019` : General Personal Data Protection Law (*Lei Geral de Protecao de Dados Pessoais*) / BRA
- `PDPA2012` : Personal Data Protection Act 2012 / SGP
- `CCPA2018` : California Consumer Privacy Act / US-CA
- `VCDPA2021` : Virginia Consumer Data Protection Act / US-VA

Consistency rules

- the keys used to sign a SERVICE OFFERING description and the `providedBy` PARTICIPANT description should be from the same keychain.

4.2 Resource

A resource that may be aggregated in a Service Offering or exist independently of it.

Attribute	Card.	Trust Anchor	Comment
`aggregationOf[]`	0..*	State	`resources` related to the resource and that can exist independently of it.

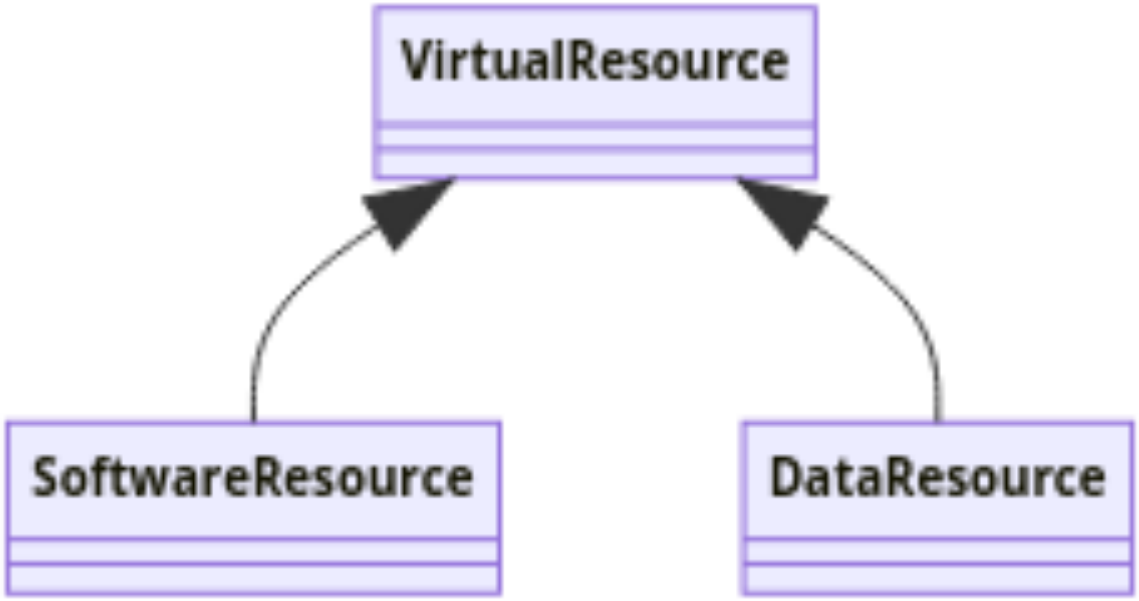
4.2.1 Physical Resource

A Physical Resouce inherits from a Resource.
A Physical resource is, but not limited to, a datacenter, a baremetal service, a warehouse, a plant. Those are entities that have a weight and position in physical space.

Attribute	Card.	Trust Anchor	Comment
`maintainedBy[]`	1..*	State	a list of `participant` maintaining the resource in operational condition and thus having physical access to it.
`ownedBy[]`	0..*	State	a list of `participant` owning the resource.
`manufacturedBy[]`	0..*	State	a list of `participant` manufacturing the resource.
`locationAddress[].country`	1..*	State	a list of physical location in ISO 3166-1 alpha2, alpha-3 or numeric format.
`location[].gps`	0..*	State	a list of physical GPS in ISO 6709:2008/Cor 1:2009 format.

4.2.2 Virtual Resource

A Virtual Resource inherits from a Resource.
A Virtual resource is a resource describing recorded information such as, and not limited to, a dataset, a software, a configuration file, an AI model. Special sub-classes of Virtual Resource are `SoftwareResource` and `DataResource`.



Attribute	Card.	Trust Anchor	Comment
`copyrightOwnedBy[]`	1..*	State	A list of copyright owners either as a free form string or `participant` URIs from which Self-Descriptions can be retrieved. A copyright owner is a person or organization that has the right to exploit the resource. Copyright owner does not necessarily refer to the author of the resource, who is a natural person and may differ from copyright owner.
`license[]`	1..*	State	A list of SPDX license identifiers or URL to license document

4.2.3 Instantiated Virtual Resource

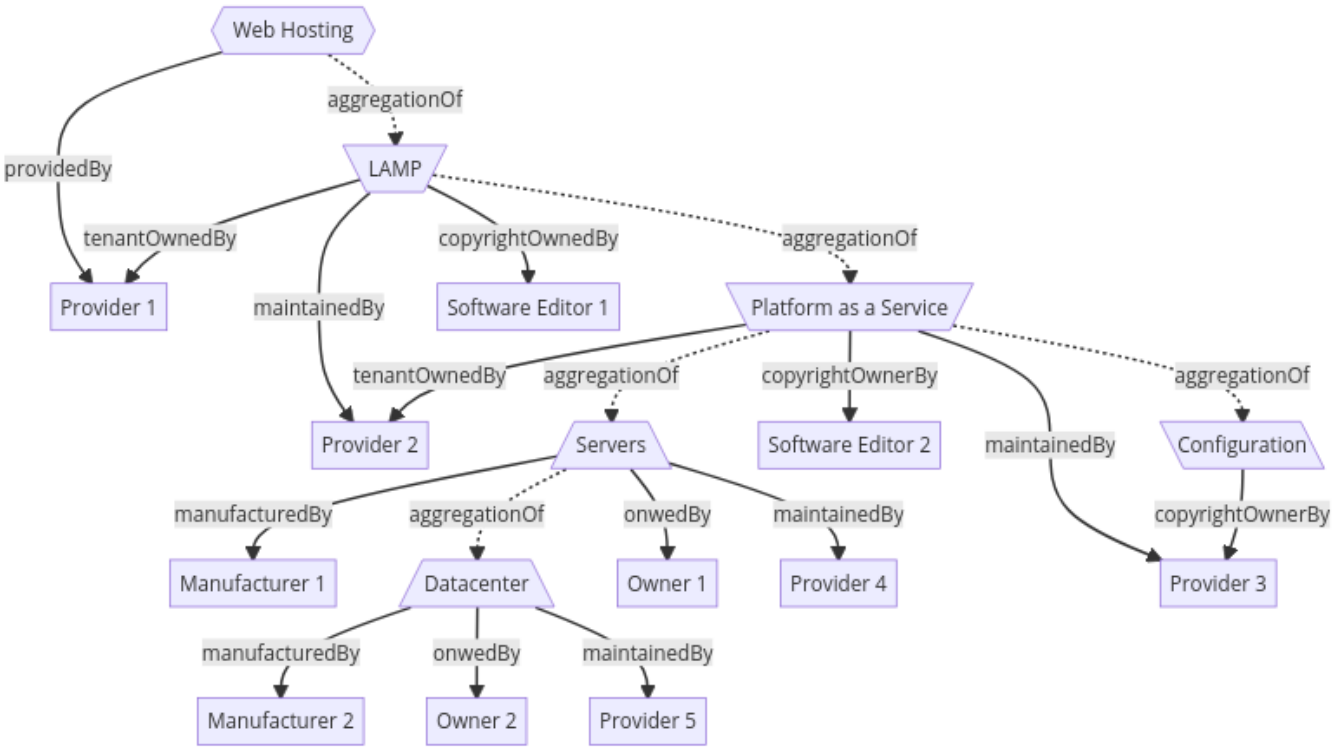
An Instantiated Virtual Resource is an instance from a Virtual Resource.
An Instantiated Virtual resource is a running resource exposing endpoints such as, and not limited to, a running process, an online API, a network connection, a virtual machine, a container, an operating system.

Attribute	Card.	Trust Anchor	Comment
`maintainedBy[]`	1..*	State	a list of `participant` maintaining the resource in operational condition.
`hostedOn`	1	State	a `resource` where the process is running, being executed on.
`instanceOf`	1	State	a `virtual resource` (normally a `software` resource) this process is an instance of.
`tenantOwnedBy[]`	1..*	State	a list of `participant` with contractual relation with the resource.
`serviceAccessPoint[]`	1..*	State	a list of Service Acccess Point which can be an endpoint as a mean to access and interact with the resource

5. Examples

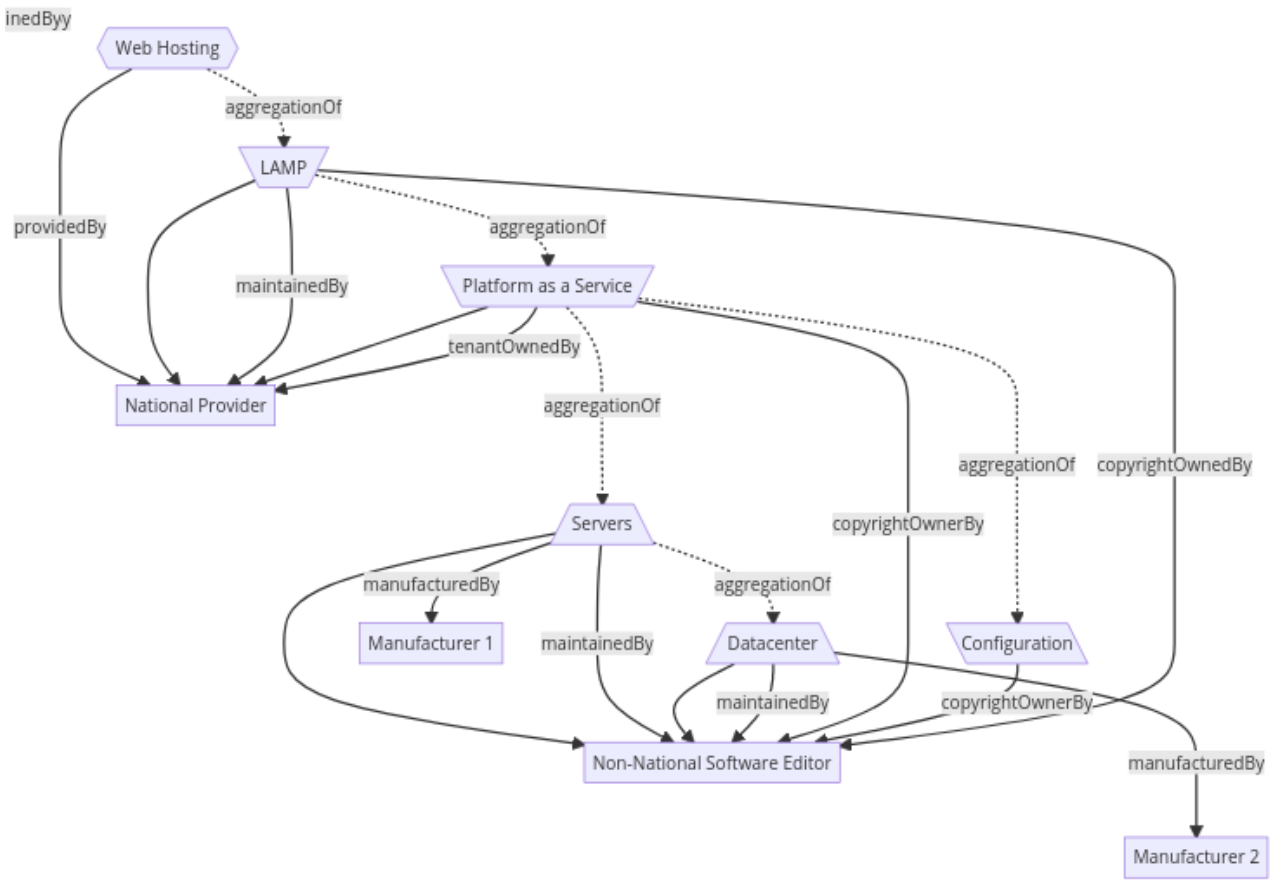
5.1 Generic LAMP offering

LAMP is an acronym for Linux, Apache, MySQL, PHP. It is a software stack consisting of the operating system, an HTTP server, a database management system and an interpreted programming language, and is used to set up a web server.



5.1.1 LAMP offering using one software vendor

Example of a LAMP offering with one software vendor.
This diagram can be used to illustrate how several "Trusted Cloud" offers are built.

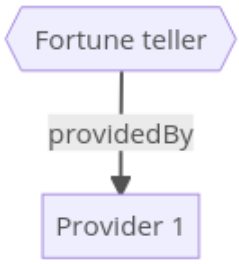


5.2 Simple Fortune teller

Example of a simple API endpoint returning a fortune from the BSD packet fortune.

For the same service offering, 3 examples of service offering are detailled with 3 different transparency level: Trust_Index(Service Offering 1 v1.0) < Trust_Index(Service Offering 1 v2.0) < Trust_Index(Service Offering 1 v3.0)

5.2.1 Fortune teller v1.0



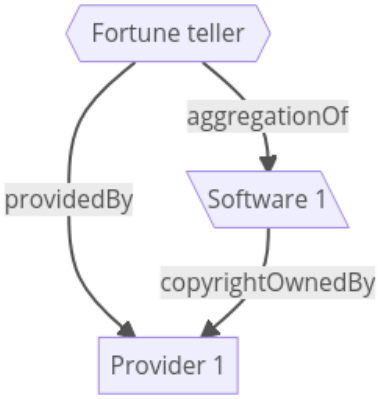
Service Offering

```
name: Fortune teller
description: API to randomly return a fortune
providedBy: url(provider1)
termsAndConditions:
  - https://some.url.for.terms.and.condition.example.com
```

Provider 1

```
registrationNumber: FR5910.899103360
headquarterAddress:
  country: FR
legalAddress:
  country: FR
```

5.2.2 Fortune teller v2.0



Service Offering

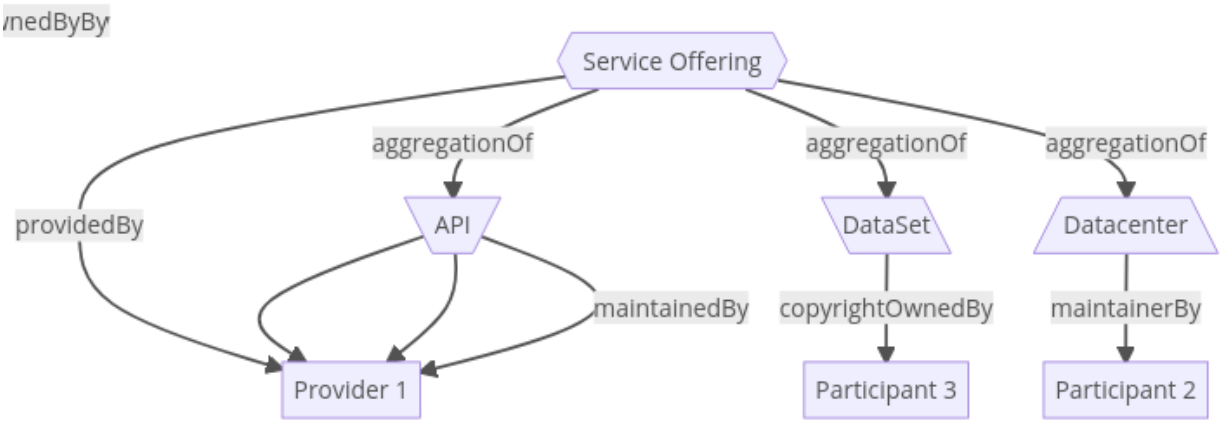
```
name: Fortune teller
description: API to randomly return a fortune
providedBy: url(provider1)
```

```
aggregationOf:
  - url(software1)
termsAndConditions:
  - https://some.url.for.terms.and.condition.example.com
```

Software 1

```
name: api software
copyrightOwnedBy:
  - url(provider1)
license:
  - EPL-2.0
```

5.2.3 Fortune teller v3.0



Service Offering

```
name: Fortune teller
description: API to randomly return a fortune
providedBy: url(provider1)
aggregationOf:
  - url(software1)
  - url(dataset1)
  - url(datacenter1)
termsAndConditions:
  - https://some.url.for.terms.and.condition.example.com
policies:
  - type: opa
    content: |-
      package fortune
      allow = true {
```

```
      input.method = "GET"
    }
```

API 1

```
name: api software
maintainedBy:
 - url(provider1)
tenantOwnedByBy:
 - url(provider1)
copyrightOwnedBy:
 - url(provider1)
license:
 - EPL-2.0
```

Dataset 1

```
name: fortune dataset
copyrightOwnedBy:
 - name: The Regents of the University of California
   registrationNumber: C0008116
   headquarterAddress:
     state: CA
     country: USA
   legalAddress:
     state: CA
     country: USA
license:
 - BSD-3
 - https://metadata.ftp-master.debian.org/changelogs//main/f/fortune-mod/fortune-mod_1.99.1-7.1_copy
```

Participant 2

```
name: Cloud Service Provider
registrationNumber: FR5910.424761419
headquarterAddress:
 country: FR
legalAddress:
 country: FR
```

Datacenter 1

```
name: datacenter
maintainedBy: url(participant2)
location:
 - country: FR
```

---

1. https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/651992/EPRS_BRI(2020)651992_EN.pdf ↩

2. FAIR = findable, accessible, interoperable, reusable; cf. https://www.go-fair.org/fair-principles/ ↩

3. https://www.w3.org/standards/semanticweb/data ↩