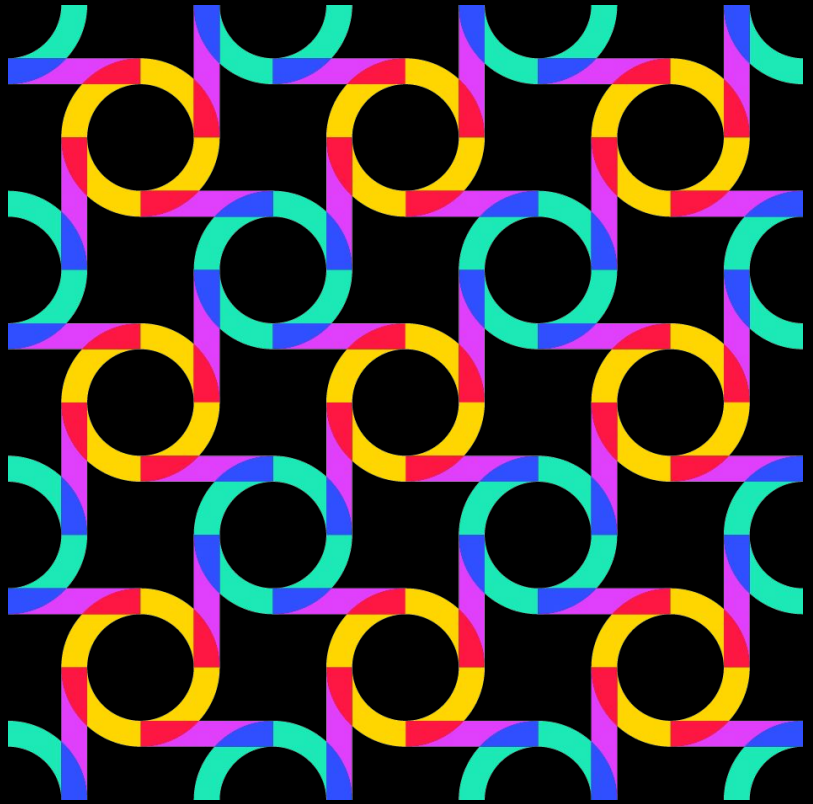
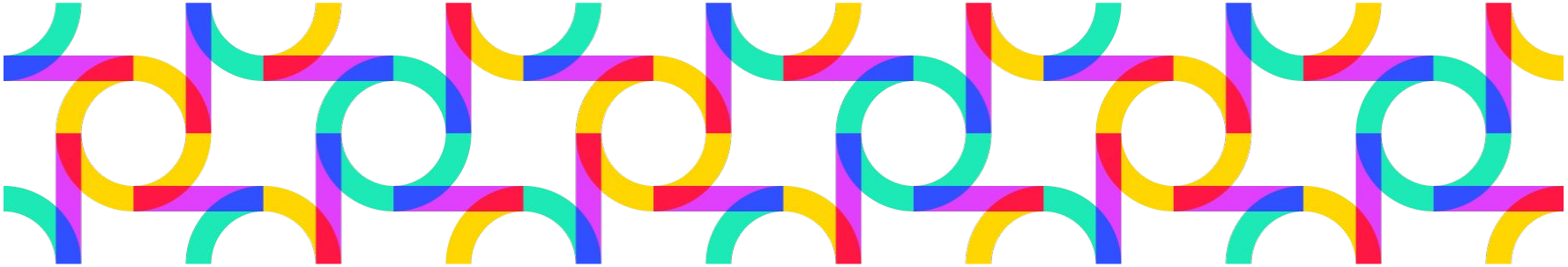
iSHARE - Deepdive & i4Trust Data Sovereignty & Trust aspects

Speakers:

Rajiv Rajani - CTO iSHARE Foundation

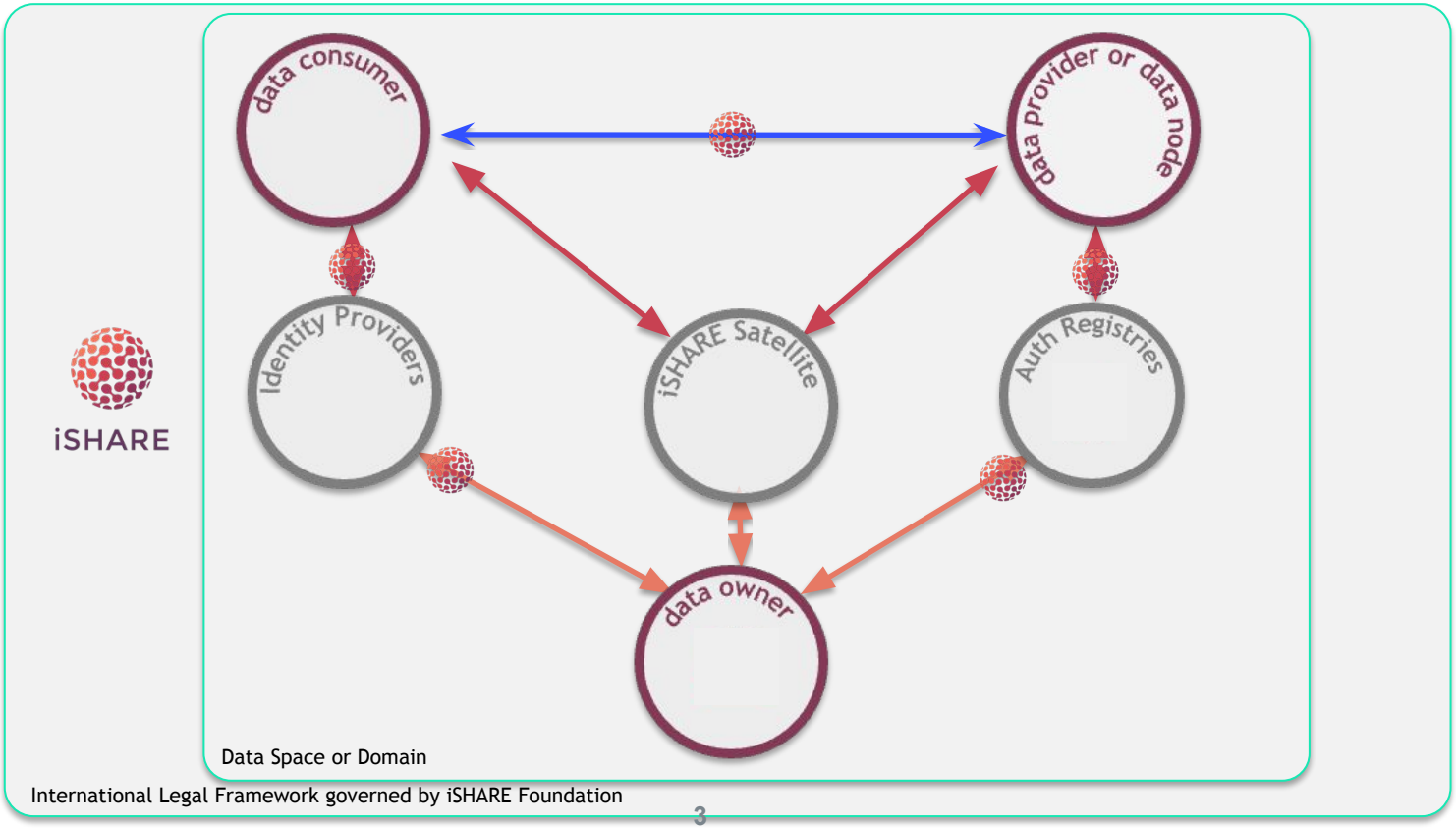




Technical Deepdive - iSHARE

i4Trust

Recap: iSHARE role model



i4Trust

Service consumer

- Service consumer is usually data consumer in general contexts.
- It needs data or updates data in order for it to play its part in the business process
- Data owner gives rights to service consumer to get data directly at source
- Data consumer can be a machine or human in which case it uses identity provider to identify him/herself
- Abides by the data license, attached to the authorisation it receives from data owner, on the data it receives
- Human service consumers uses an Identity Provider (registered in the network) to identify itself to service providers

Service Provider

- Service provider is usually the data provider (IT partner) who has data owner's data
- Examples are cloud SaaS solutions, data lakes, data hubs, hosted software platforms, etc.
- They are responsible for making sure that data is shared with only right organisations' machines or representatives
- Service provider and service consumer share data directly with each other
- iSHARE specifies API specification for Identification, Authentication and Authorisation for exposing data services
- The specifications are modified version of oAuth 2.0 which support verification of credentials inline with verifiable credentials principles
- Reference open source implementations:
  - https://github.com/iSHAREScheme/ServiceProvider
  - https://github.com/i4Trust/tutorials/tree/main/PacketDelivery-ReferenceExample/Data-Service-Provider

Entitled Party

- Entitled party is a party who is data owner at a given time and context of that data
- Entitled party has the first right on the data set in question and they can determine what can be done with that data
- They use authorization registries to specify which consumer has access to which data set from the data set they are entitled to
- They also specify the data license for the data set and consumer which determines the data usage rights for consumers
- Entitled party chooses it own authorization registry provider to maintain authorisations over its data for other organizations

What is iSHARE Satellite?

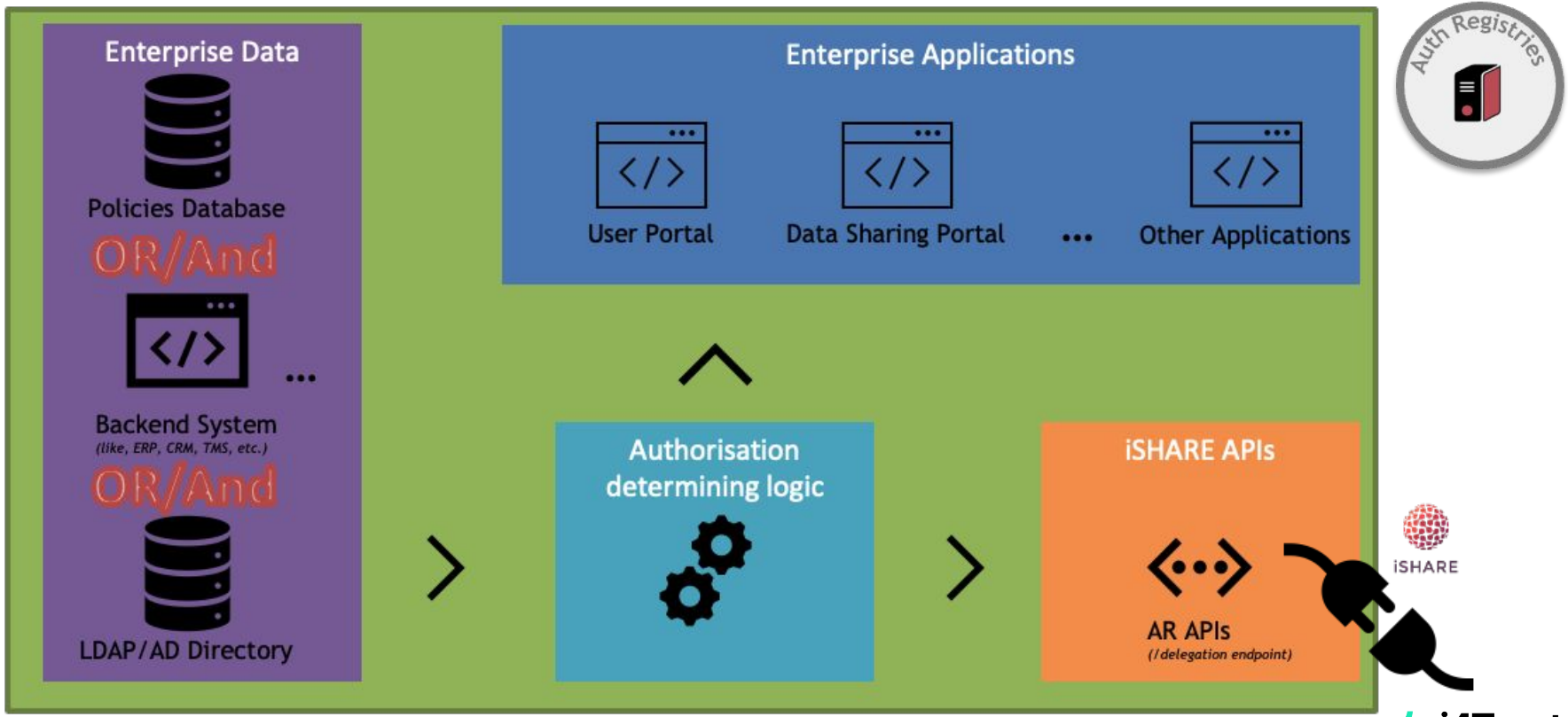
- iSHARE satellite is a federated certified role which needs to be played by a organisation within the dataspace
- It essentially does:
  - Onboard and maintain list of participants and its statues
  - Enables multilateral trust when participants signup
  - Provide APIs for participants to get detailed information along with status of other participants with whom they want to interact or are interacting
  - Plays the facilitator role for common agreements like, but not limited to business agreements, operational agreements, etc.
  - Is federated part of the governing body of interoperable data spaces
- Each dataspace is expected to have at least one trust provider
- When the dataspace uses iSHARE framework, the trust provider is known as iSHARE Satellite

Identity Provider role is designed for users to reuse their existing identity provider at various service/data providers

Identity Provider

- The Identity Provider role deals with the human identities with varying level of assurances, as defined in eIDAS framework, to support various use cases
- It is based on OpenID Connect as standard, however, with slight adjustment to make it better suitable for B2B, B2G, G2G and of course B2C and G2C
- Depending on the criticality of the data, appropriate level of assurance for an identity can be requested
- iSHARE specifications are designed such that service/data provider does not necessarily need to pre-register an identity provider as it can verify if it is an iSHARE certified provider from iSHARE satellite

Authorisations Registry (provider) role



Trustworthy Identification, Authentication and Authorisations

Interaction

Machine-to-Machine (M2M)
Communication between machines, without interference by a human

Human-to-Machine (H2M)
Communication between a human and (a) machine(s). Requires a user interface

Facilitate

Flexible authorizations
- Coarse-grained: broad authorization
- Fine-grained: specific authorization
- Flexibility on where to store authorizations

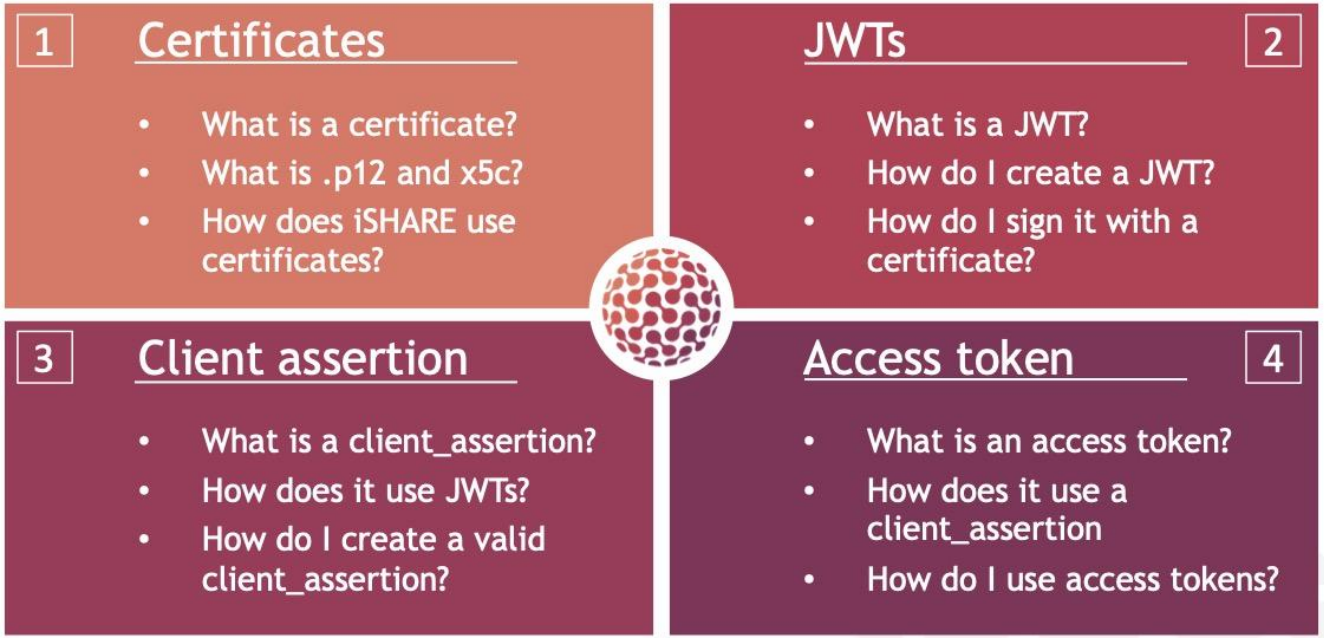
Portable identities
Identities can be spread out and recognised, i.e. portable, across multiple, independent systems

Enable

Delegations
Functions as evidence that a party is directly or indirectly operating on behalf of a known party

Customer in control
parties are allowed to modify or withdraw access rights to their data or services, whenever they wish

Basics of every IAM transaction



A certificate is a digital file used for integrity and authenticity

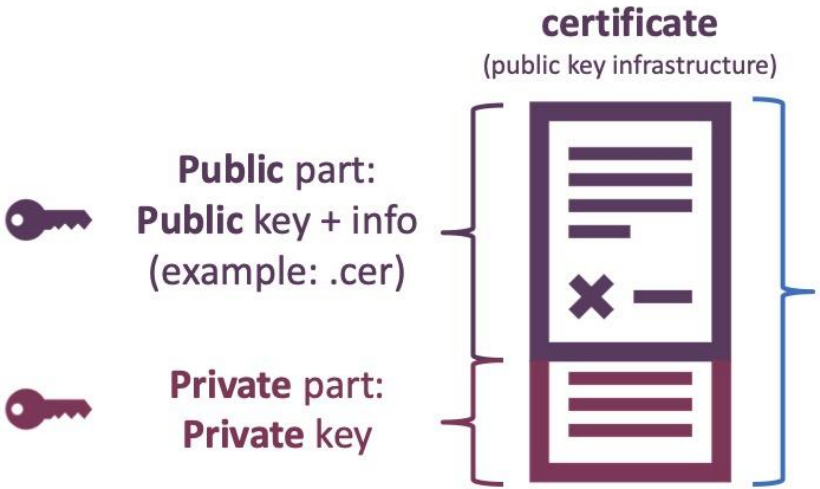
certificate
(public key infrastructure)

- Digital file that proves ownership of public key
- Certificate provides information on the owner
- Certificate is issued by a Certificate Authority
- Keys used to sign or encrypt other files

More info:
Cheat Sheet
or Dev. Portal

i4Trust

A certificate consists of a public and a private part



Full certificate:
Public + Private
(example: .p12)

More info:
Cheat Sheet
or Dev. Portal

i4Trust

iSHARE relies on PKI-certificates to verify organisational identities

certificate
(public key infrastructure)

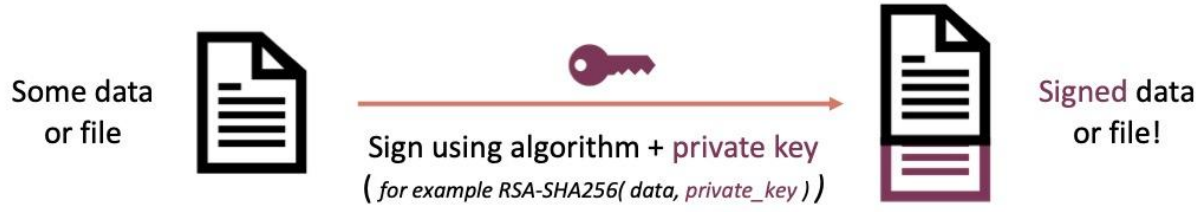
In *general*
- digital file used to sign or encrypt other files
- contains both a private key and public key
- part of **public-key infrastructure** (e.g. PKIoverheid or eIDAS)

In *iSHARE*
- **x5c** - value
  - String of letters representing public part of certificate
  - Used to *verify* signing and identity
- **.p12** file containing both private and public part
  - Used to sign (*integrity* and *authenticity*)
  - Password protected

More info:
Cheat Sheet
or Dev. Portal

i4Trust

A private keys signs data
A public keys verifies this signature



More info:
Cheat Sheet
or Dev. Portal

i4Trust

X.509 certificate encoding formats and extensions

- Digital certificates comes in various forms →
- You can usually convert them from one form to another
- Openssl is the most common tool used to manage and convert digital certificates
- Digital certificates cheat sheet can be found here
- For managing some format conversions a ready script is available here

- Base64 (ASCII)
  - PEM
    - .pem
    - .crt
    - .ca-bundle
  - PKCS#7
    - .p7b
    - .p7s
- Binary
  - DER
    - .der
    - .cer
  - PKCS#12
    - .pfx
    - .p12

JSON Web Tokens (JWTs) are base64- encoded JSON objects

A basic JWT consists of three base64-encoded strings joined together by a dot '.'

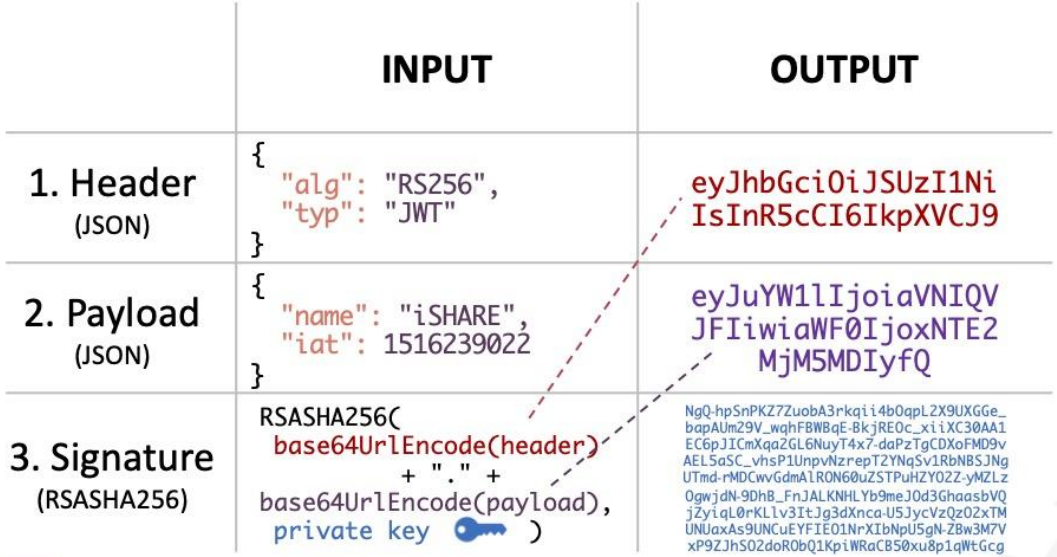
	INPUT	OUTPUT
1. Header (JSON)	{ "alg": "RS256", "typ": "JWT" }	
2. Payload (JSON)	{ "name": "iSHARE", "iat": 1516239022 }	
3. Signature (RSASHA256)		

More info: https://jwt.io or Dev. Portal

i4Trust

JSON Web Tokens (JWTs) are base64- encoded JSON objects

A basic JWT consists of three base64-encoded strings joined together by a dot '.'

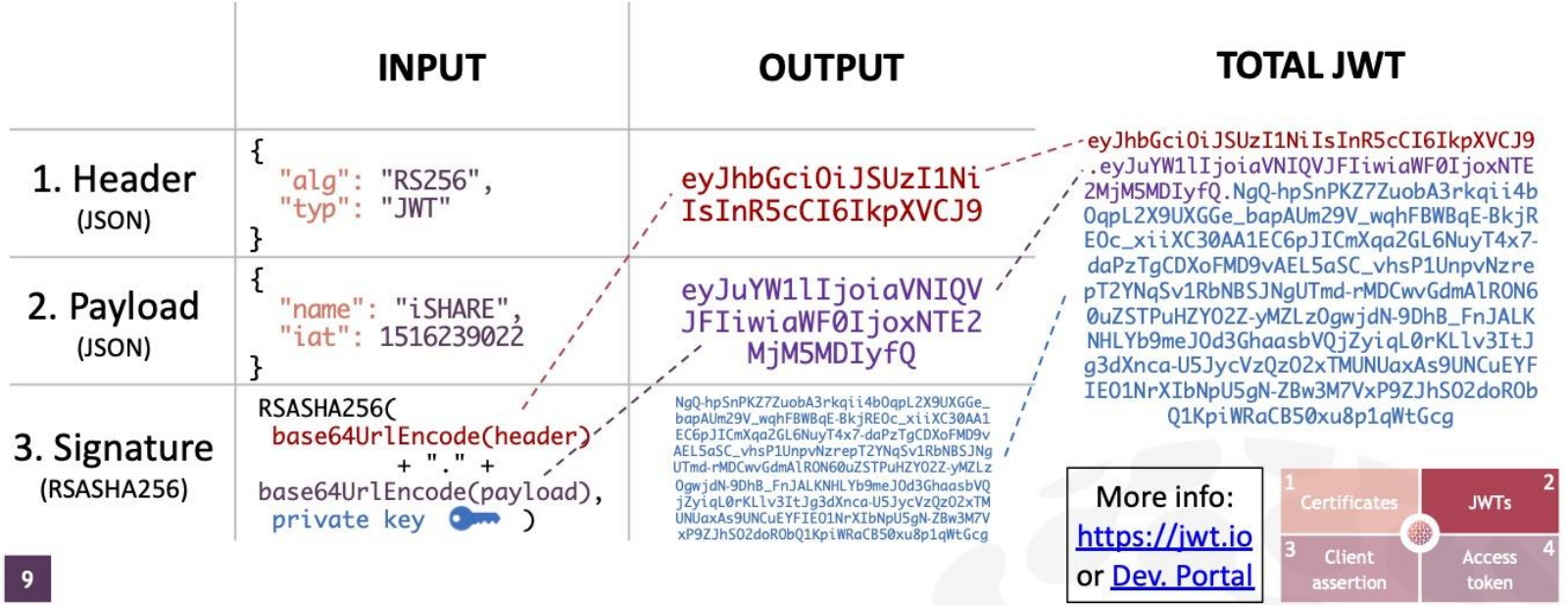


More info:
https://jwt.io
or Dev. Portal

i4Trust

JSON Web Tokens (JWTs) are base64- encoded JSON objects

A basic JWT consists of three base64-encoded strings joined together by a dot '.'



i4Trust

In iSHARE, client assertions are JWTs with iSHARE specific Header and Payload

A client assertion is a JWT used to prove your identity to another party.
To do so, iSHARE prescribes a specific format for the Header and Payload of the JWT.

HEADER + "." + PAYLOAD

```
{
  "alg": "RS256",
  "typ": "JWT",
  "x5c": ["MIIEgTCCAmmg……
          (long string)
         ……reK18+JkAjAJU="]
}
```

x5c-value of your certificate
(in this case of party ABC Trucking)

```
{
  "iss": "EU.EORI.NL000000001",
  "sub": "EU.EORI.NL000000001",
  "jti": "df7ffc54cf148d15b0",
  "iat": 1556210430,
  "nbf": 1556210430,
  "exp": 1556210460,
  "aud": "EU.EORI.NL000000003"
}
```

- Your EORI nr. *(here: ABC Trucking's EORI)*
- Should equal "iss"
- Identifier of the JWT
- Time of issuing in seconds (*UNIX time*)
- Should equal "iat"
- Expiry time ("iat" *+ 30 seconds*)
- Other party's EORI nr. *(here: Warehouse 13's EORI)*

In iSHARE, client assertions are JWTs with iSHARE specific Header and Payload

A client assertion is a JWT used to prove your identity to another party.
To do so, iSHARE prescribes a specific format for the Header and Payload of the JWT.

HEADER + "." + **PAYLOAD** → Sign using private key with RSASHA256 → **CLIENT ASSERTION**

```
{
  "alg": "RS256",
  "typ": "JWT",
  "x5c": ["MIIEgTCCAmmg……
          (long string)
         ……reK18+JkAjAJU="]
}
```

```
{
  "iss": "EU.EORI.NL000000001",
  "sub": "EU.EORI.NL000000001",
  "jti": "df7ffc54cf148d15b0",
  "iat": 1556210430,
  "nbf": 1556210430,
  "exp": 1556210460,
  "aud": "EU.EORI.NL000000003"
}
```

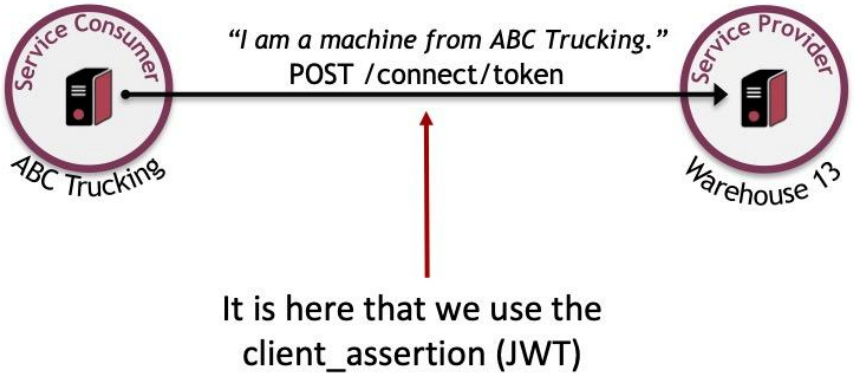
```
eyJhbGciOiJSUzI1NiIsInR5c...
      (very long string)
……D2vek8KTfhmzUw2i9UG
D5SsE0yOluDzUGU-Q
```

signed by your private key
(in this case of party ABC Trucking)

i4Trust

An iSHARE party uses the client_assertion to get an access token from another iSHARE Party

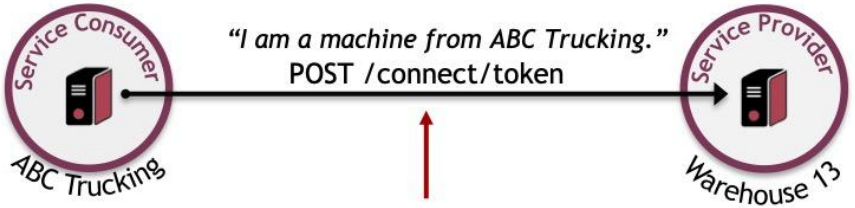
Every iSHARE Party that exposes services has a /token endpoint, where you can get an access token for the services it exposes.



i4Trust

An iSHARE party uses the client_assertion to get an access token from another iSHARE Party

Every iSHARE Party that exposes services has a /token endpoint, where you can get an access token for the services it exposes.



Values in the body this request

"grant_type":	"client_credentials",	MUST be "client_credentials"
"scope":	"iSHARE",	MUST be "iSHARE"
"client_id":	"EU.EORI.NL000000001",	Your EORI nr. *(here: ABC Trucking's EORI)*
"client_assertion_type":	"urn:ietf:params:oauth:client-assertion-type:jwt-bearer",	MUST be this string
"client_assertion":	"eyJhbGciOiJSUzI1NiIsInR5c…… (very long string) ……D2vek8KTf5SsE0yOluDzUGU-Q"	The JWT we created earlier

i4Trust

An iSHARE party uses the client_assertion to get an access token from another iSHARE Party

Every iSHARE Party that exposes services has a /token endpoint, where you can get an access token for the services it exposes.


```
Values in this response (if 200 OK)
{
    "access_token": "eyJhbGciOi…(long string)…ahvEQ5w",
    "expires_in": 3600,
    "token_type": "Bearer"
}
```

The access token, for your further use

The expiry time, in seconds

The type of token (*should be "bearer"*)

Trustworthy Identification, Authentication and Authorisations

Interaction

Machine-to-Machine (M2M)
Communication between machines, without interference by a human

Human-to-Machine (H2M)
Communication between a human and (a) machine(s). Requires a user interface

Facilitate

Flexible authorizations
- Coarse-grained: broad authorization
- Fine-grained: specific authorization
- Flexibility on where to store authorizations

Portable identities
Identities can be spread out and recognised, i.e. portable, across multiple, independent systems

Enable

Delegations
Functions as evidence that a party is directly or indirectly operating on behalf of a known party

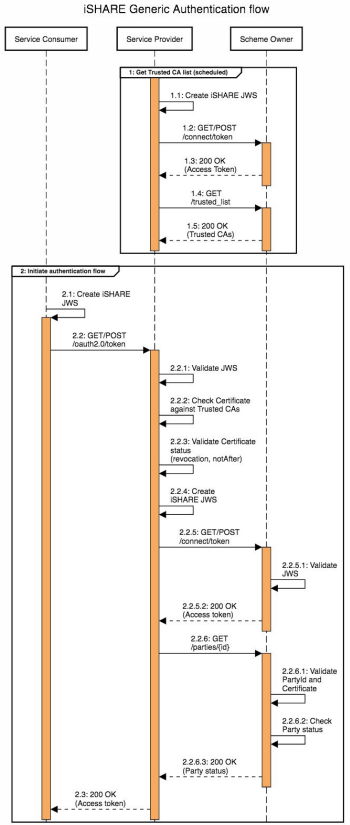
Customer in control
parties are allowed to modify or withdraw access rights to their data or services, whenever they wish

Authentication sequence diagram (M2M flow)

i4Trust

Authentication sequence diagram (M2M)

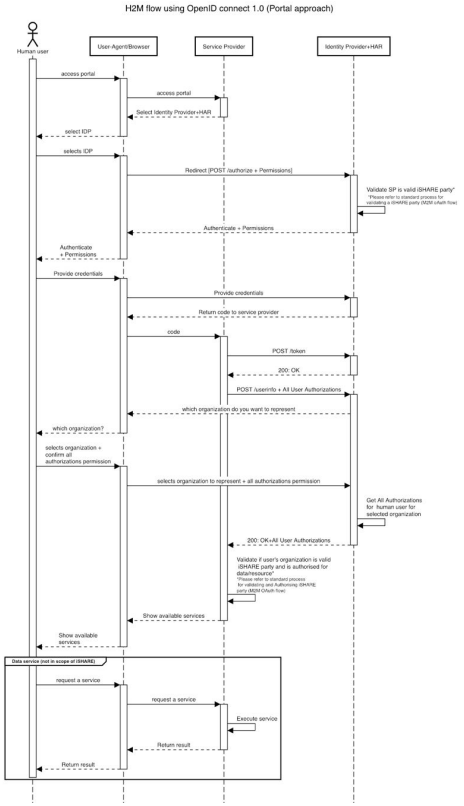
- iSHARE M2M authentication flow applies when servers from 2 organisations want to connect with each other
- Data is always shared peer to peer between these 2 organisations
- The process begins when a Service Consumer sends a client assertion to Service Provider to authenticate
- **Client Assertion** is a **JWT** containing **claims** about the **identity** of Service Consumer which is also **signed** by Service Consumer with its own eIDAS or equivalent supported certificate
- Following successful authentication, actual data request is processed, provided right authorisations are in place
- Service Provider relies on Authorisation Registries to determine the authorisations for Service Consumer
- The flow is explained in detail here



i4Trust

Authentication sequence diagram (H2M)

- iSHARE H2M authentication flow applies when Human Service Consumer wants to get data from Service Provider
- Service Provider usually has a Portal or an Application from where Human Service Consumer can access that data
- The process begins when a Human Service Consumer tries to **access data via Portal/Application of Service Provider**
- Service Provider relies on **Identity Provider** to **authenticate** Human Service Consumer
- Human Service Consumer can choose **any* Identity Provider** where its identities are registered to authenticate himself to the Service Provider
- Following successful authentication, actual data request is processed, provided right authorisations are in place
- Service Provider **relies on Authorisation Registries** to determine the **authorisations** for Human Service Consumer
- The flow is explained in detail [here]



i4Trust

Example model for policy storage in reference AR

```
{
    "delegationEvidence": {
        "notBefore": "1509633681",
        "notOnOrAfter": "1509633781",
        "policyIssuer": "EU.EORI.NL123456789",
        "target": {
            "accessSubject": "EU.EORI.NL987654321"
        },
        "policySets": [
            {
                "maxDelegationDepth": "5",
                "target": {
                    "environment": {
                        "licenses": ["ISHARE.0001"]
                    }
                },
                "policies": [
                    {
                        "target": {
                            "resource": {
                                "type": "GS1.CONTAINER",
                                "identifiers": ["*"],
                                "attributes": ["ATTRIBUTE.ETA", "ATTRIBUTE.WEIGHT"]
                            },
                            "actions": ["ISHARE.READ", "ISHARE.WRITE"],
                            "environment": {
                                "serviceProviders": ["EU.EORI.NL678912345"]
                            }
                        },
                        "rules": [
                            {
                                "effect": "Permit"
                            },
                            {
                                "effect": "Deny",
                                "target": {
                                    "resource": {
                                        "type": "GS1.CONTAINER",
                                        "identifiers": ["GS1.SSCC18.725391630826493716"],
                                        "attributes": ["ATTRIBUTE.WEIGHT"]
                                    },
                                    "actions": ["ISHARE.READ"]
                                }
                            }
                        ]
                    }
                ]
            }
        ]
    }
}
```

Issuer: who provides the authorization?
Subject: to whom is the authorization applicable?

License: what 'usage conditions' has the data owner connected to the data?

Type, ID, Attribute: description of the data set to which the authorization is applied. Please note: description is NOT a part of the iSHARE agreements

Read, Write: examples of allowed actions on data

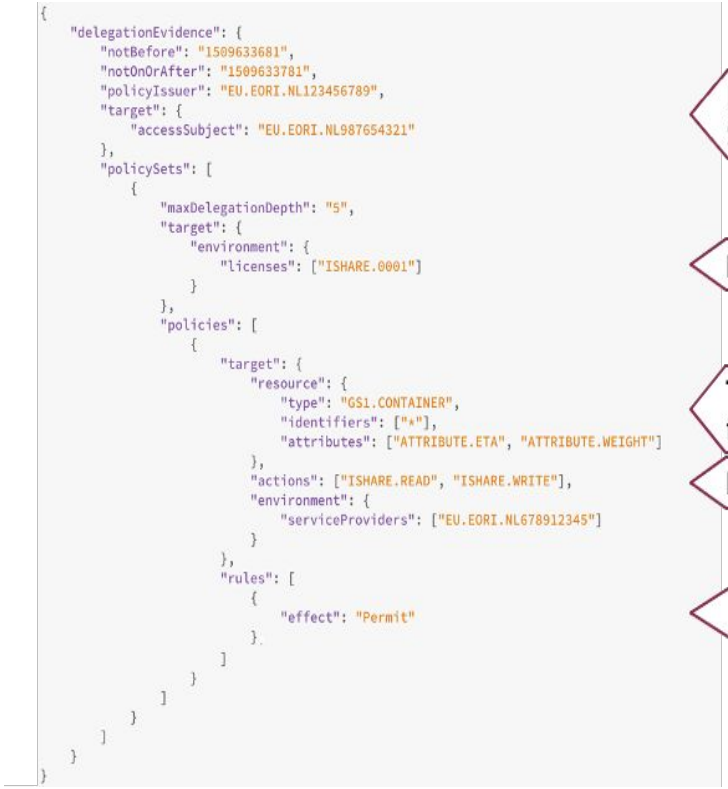
Permit: everything that complies with the rules above is allowed

Deny: everything hereafter is an exception to the above, and therefore not allowed

Important: This policy storage model is only for reference and is not mandatory to define policies as per AR iSHARE specifications

i4Trust

Response model as specified in iSHARE specifications



```
{
    "delegationEvidence": {
        "notBefore": "1509633681",
        "notOnOrAfter": "1509633781",
        "policyIssuer": "EU.EORI.NL123456789",
        "target": {
            "accessSubject": "EU.EORI.NL987654321"
        },
        "policySets": [
            {
                "maxDelegationDepth": "5",
                "target": {
                    "environment": {
                        "licenses": ["ISHARE.0001"]
                    }
                },
                "policies": [
                    {
                        "target": {
                            "resource": {
                                "type": "GS1.CONTAINER",
                                "identifiers": ["*"],
                                "attributes": ["ATTRIBUTE.ETA", "ATTRIBUTE.WEIGHT"]
                            },
                            "actions": ["ISHARE.READ", "ISHARE.WRITE"],
                            "environment": {
                                "serviceProviders": ["EU.EORI.NL678912345"]
                            }
                        },
                        "rules": [
                            {
                                "effect": "Permit"
                            }
                        ]
                    }
                ]
            }
        ]
    }
}
```

Issuer: who provides the authorization?
Subject: to whom is the authorization applicable?

License: what 'usage conditions' has the data owner connected to the data?

Type, ID, Attribute: description of the data set to which the authorization is applied. Please note: description is NOT a part of the iSHARE agreements

Read, Write: examples of allowed actions on data

Permit: everything that complies with the rules above is allowed

i4Trust

Response model depicting coarse-grained authorisation

```
{
    "delegationEvidence": {
        "notBefore": "1509633681",
        "notOnOrAfter": "1509633781",
        "policyIssuer": "EU.EORI.NL123456789",
        "target": {
            "accessSubject": "EU.EORI.NL987654321"
        },
        "policySets": [
            {
                "maxDelegationDepth": "*",
                "target": {
                    "environment": {
                        "licenses": ["*"]
                    }
                },
                "policies": [
                    {
                        "target": {
                            "resource": {
                                "type": "GS1.CONTAINER",
                                "identifiers": ["*"],
                                "attributes": ["*"]
                            },
                            "actions": ["*"]
                        },
                        "rules": [
                            {
                                "effect": "Permit"
                            }
                        ]
                    }
                ]
            }
        ]
    }
}
```

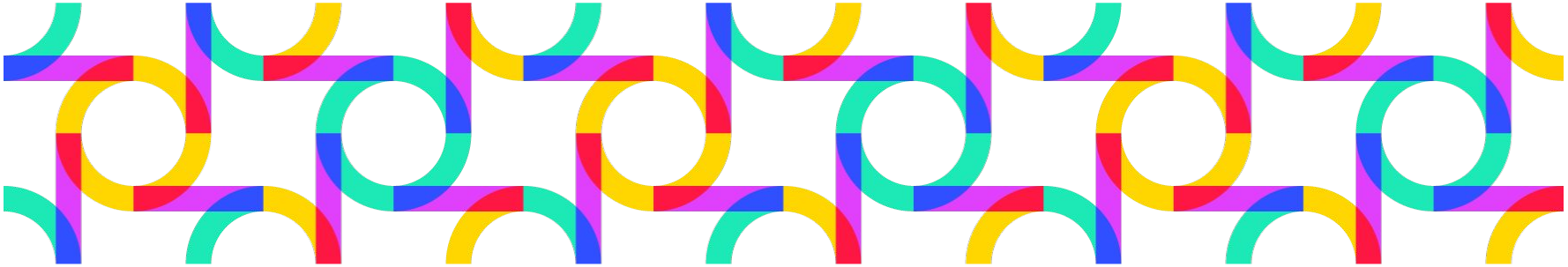
Issuer: who provides the authorization?
Subject: to whom is the authorization applicable?

License: no limitations to ‘usage conditions’ for data

Authorization applies to all **Types, IDs, Attributes**

All sorts of **actions** are allowed with the data

i4Trust



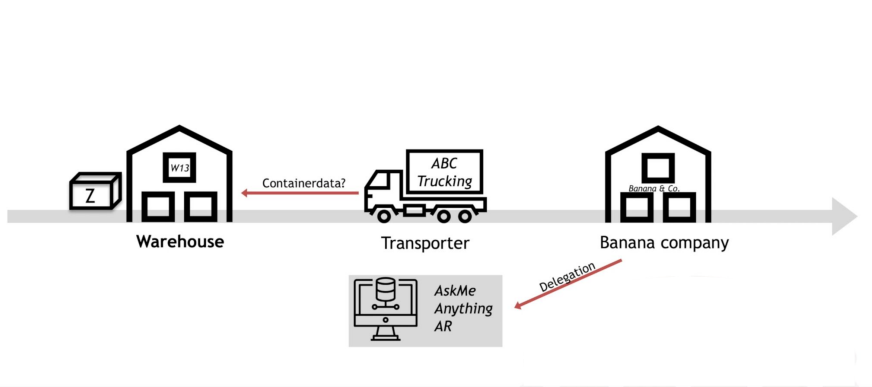
Fictional Use Case

i4Trust

Fictional use case

Case:

- ABC Trucking wants to access container data at the terminal on behalf of Banana & Co.
  - Banana & Co needs to register at AskMeAnything Authorization Registry that they delegate access rights to ABC Trucking
  - When ABC Trucking requests the data at the Service Provider, Warehouse 13, it checks with AskMeAnything for delegation evidence
  - If AskMeAnything returns valid delegation evidence, Warehouse 13 can exchange data with ABC Trucking



All calls are iSHARE calls, except if noted otherwise. These exceptions are marked with a *RED* arrow.

i4Trust

Participants of the fictional use case

- **Banana & Co**
  Freight owner, Entitled Party
- **Warehouse 13**
  Terminal, Service Provider
- **ABC Trucking**
  Transporter, Service Consumer
- **AskMeAnything**
  Authorisation Registry

Technical preparatory steps

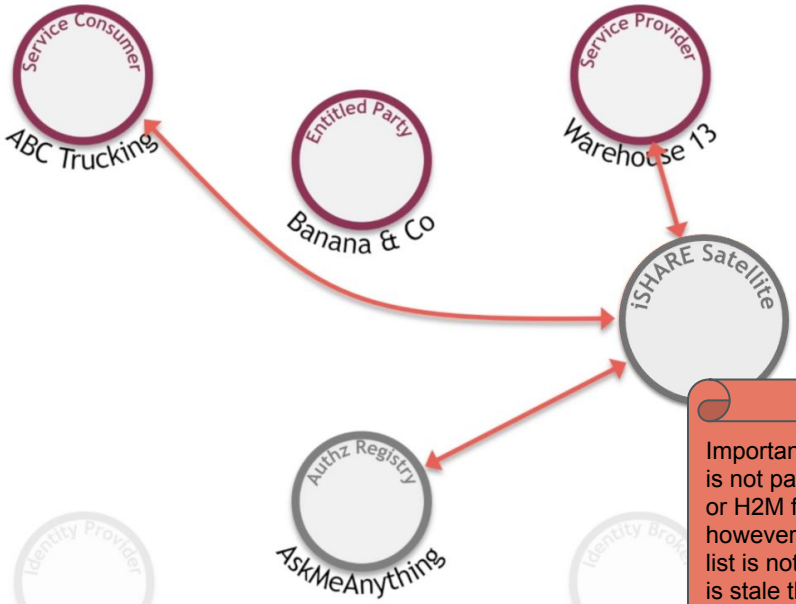
- For all participants of iSHARE following are the preparatory steps
  - They should on regular basis query and cache the trusted list locally within their own environment. As per iSHARE specifications, participants should only trust identity/data tokens from other participants only when they are signed using certificates issued by one of the Certificate Authorities (CA) from this list
  - Trusted list is a list of Certificate Authorities (CA) whose issued digital certificates are trusted in the iSHARE network. This list only contains CAs which follow high level of assurance and are inline with principles of trust framework. More importantly, the certificates they issue can be used to digitally sign data and these signatures are legally recognised
- Participant may play different roles depending on use case, so for example participant A is service consumer in use case I while it is service provider in use case II
- Depending on the role participant is playing in a particular use case relevant flow and APIs must be in place. TIP! Having sequence diagrams depicting use case helps in better understanding of role and steps one has to take

Beforehand: Getting the trusted list

- Banana & Co
  Freight owner, Entitled Party
- **Warehouse 13**
  Terminal, Service Provider
- **ABC Trucking**
  Transporter, Service Consumer
- **AskMeAnything**
  Authorisation Registry

Actions for all validating parties

1. Request access token at iSHARE Satellite
2. Receive access token
3. Request trusted_list at iSHARE Satellite
4. Receive trusted_list_jwt



Important: This call is not part of M2M or H2M flow, however, when the list is not cached or is stale then this call should be made before the flows

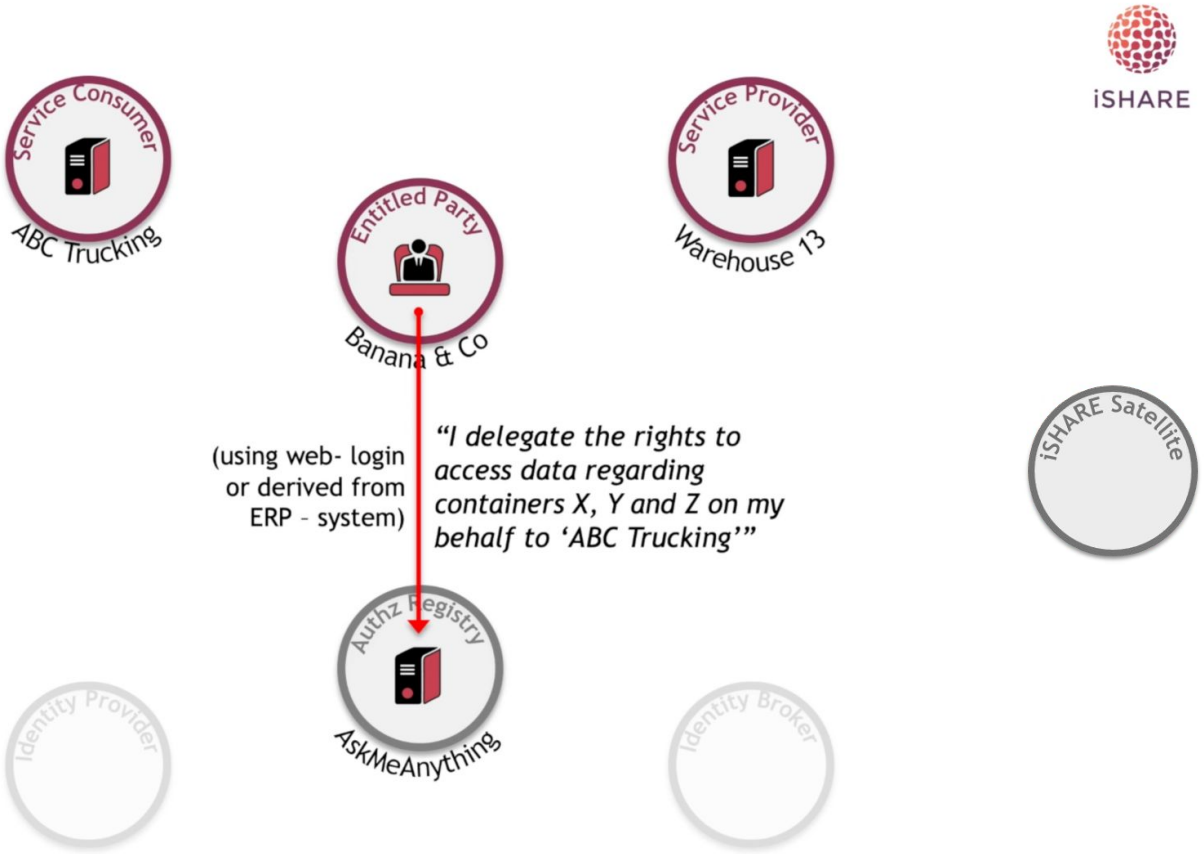
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Delegate rights

1. Update AR
2. Update SC



i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Delegate rights

1. Update AR
2. Update SC

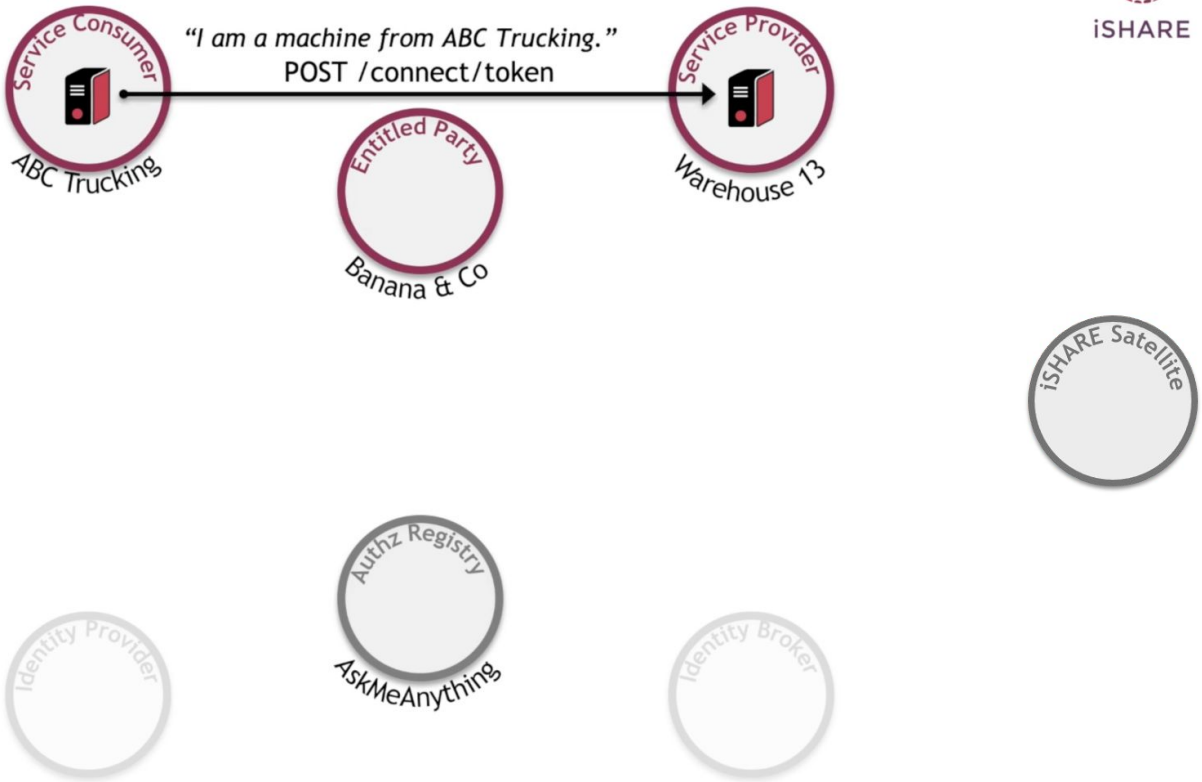
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Identification & authentication

1. Request token
2. Verify SC certificate
3. Get SO access token
4. Get Party info
5. Return access token



iSHARE

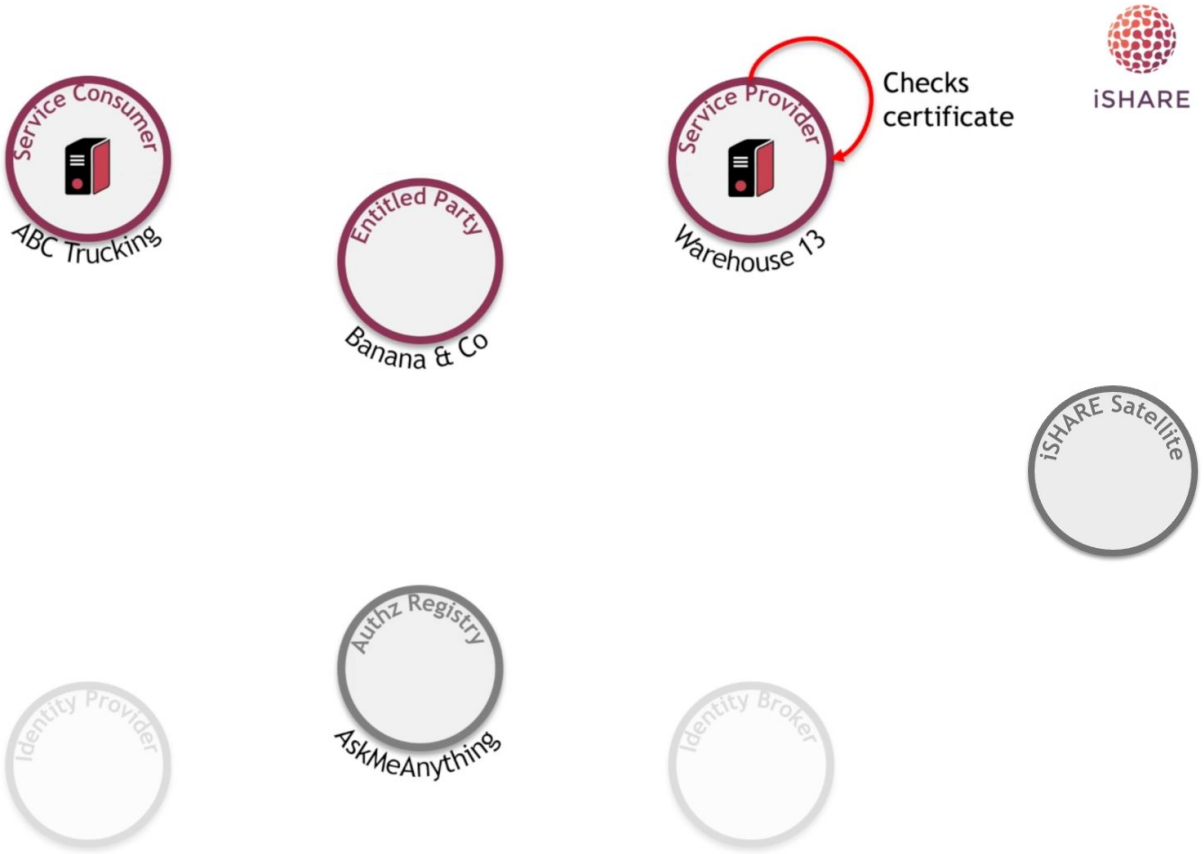
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Identification & authentication

1. Request token
2. Verify SC certificate
3. Get SO access token
4. Get Party info
5. Return access token



i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Identification & authentication

1. Request token
2. Verify SC certificate
3. Get SO access token
4. Get Party info
5. Return access token

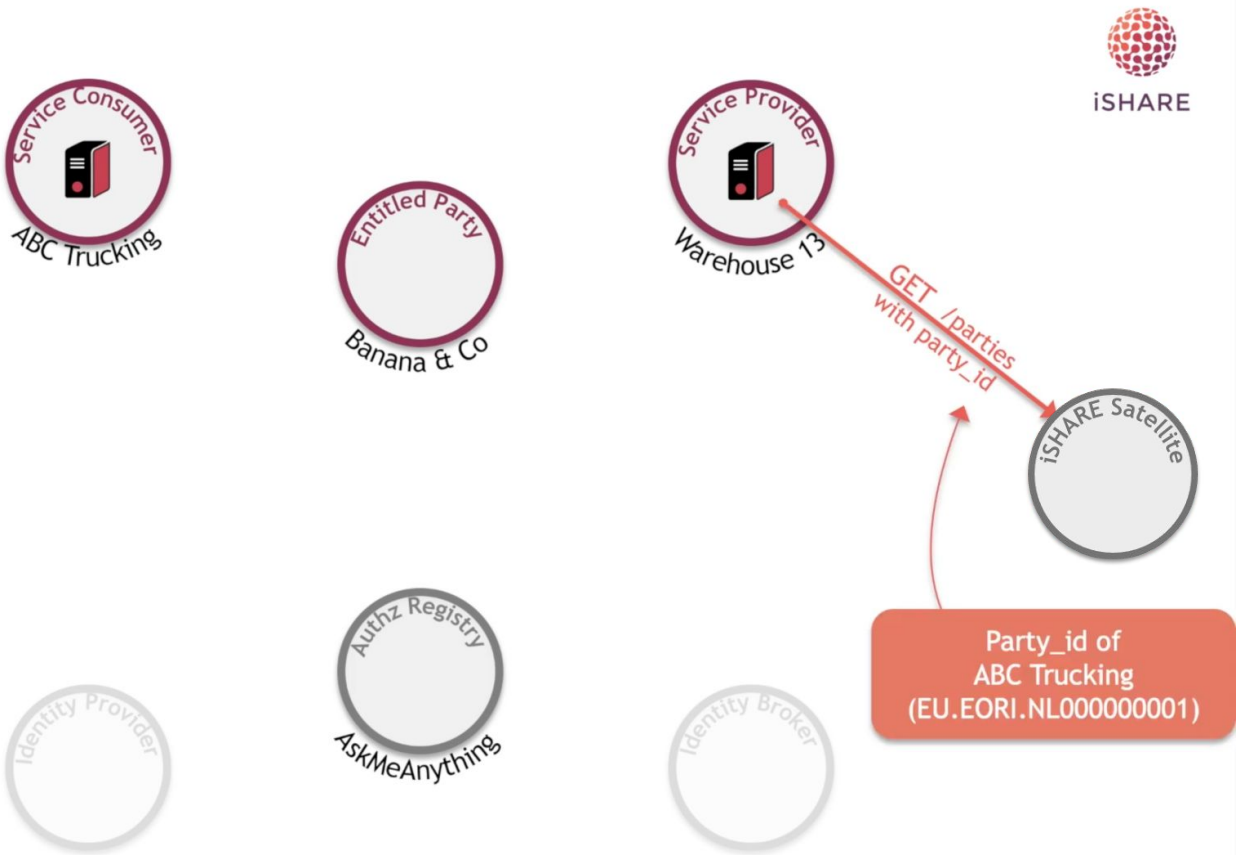
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Identification & authentication

1. Request token
2. Verify SC certificate
3. Get SO access token
4. **Get Party info**
5. Return access token



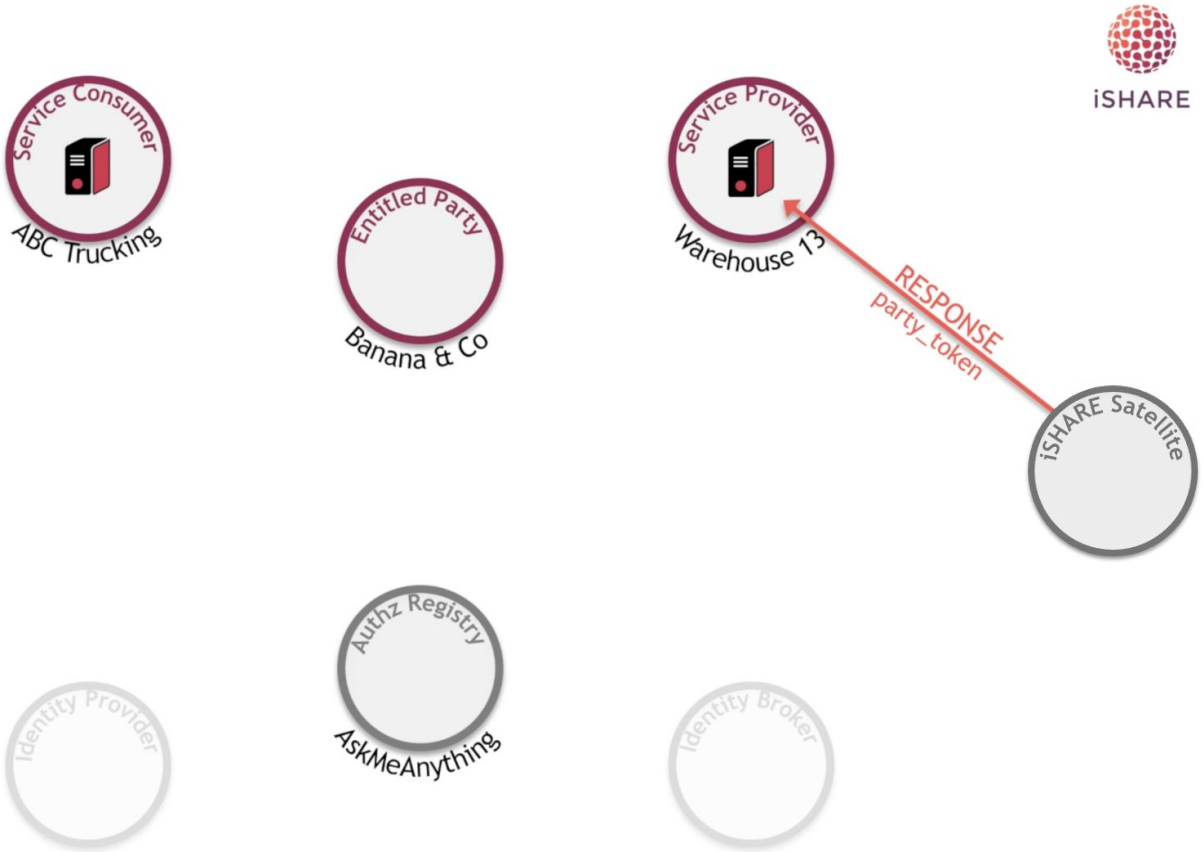
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Identification & authentication

1. Request token
2. Verify SC certificate
3. Get SO access token
4. **Get Party info**
5. Return access token



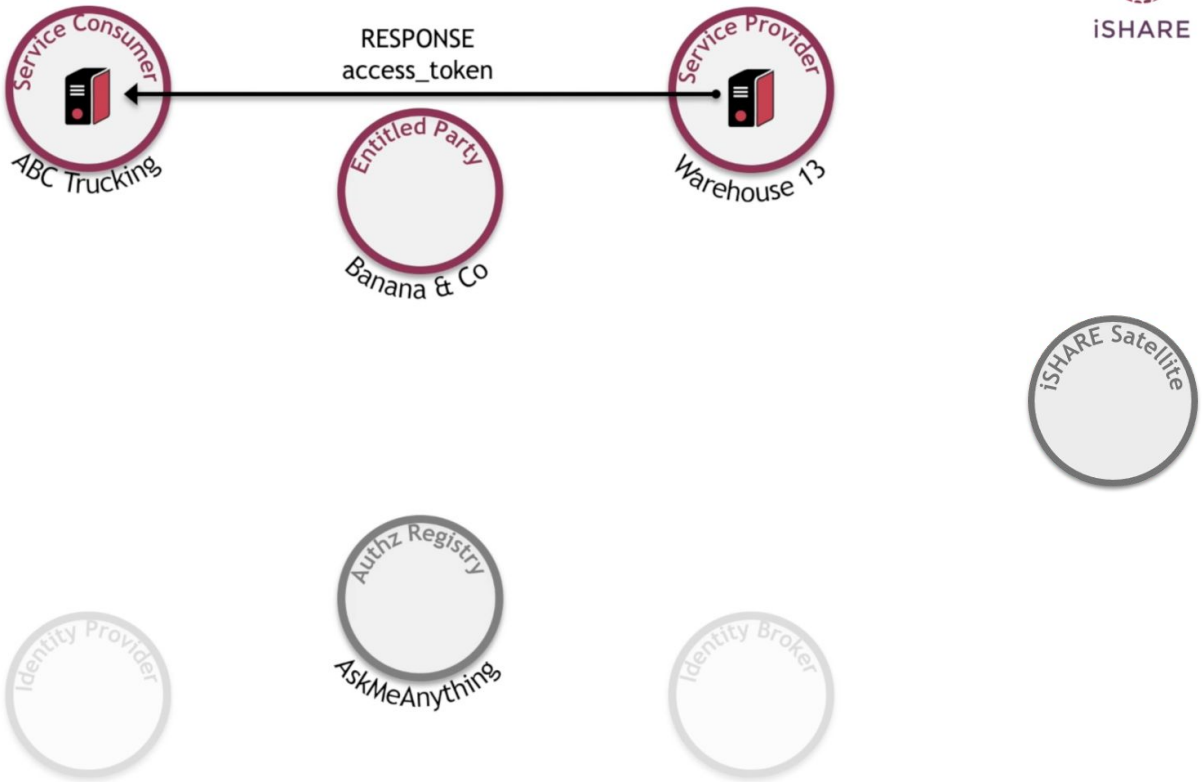
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Identification & authentication

1. Request token
2. Verify SC certificate
3. Get SO access token
4. Get Party info
5. Return access token



i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

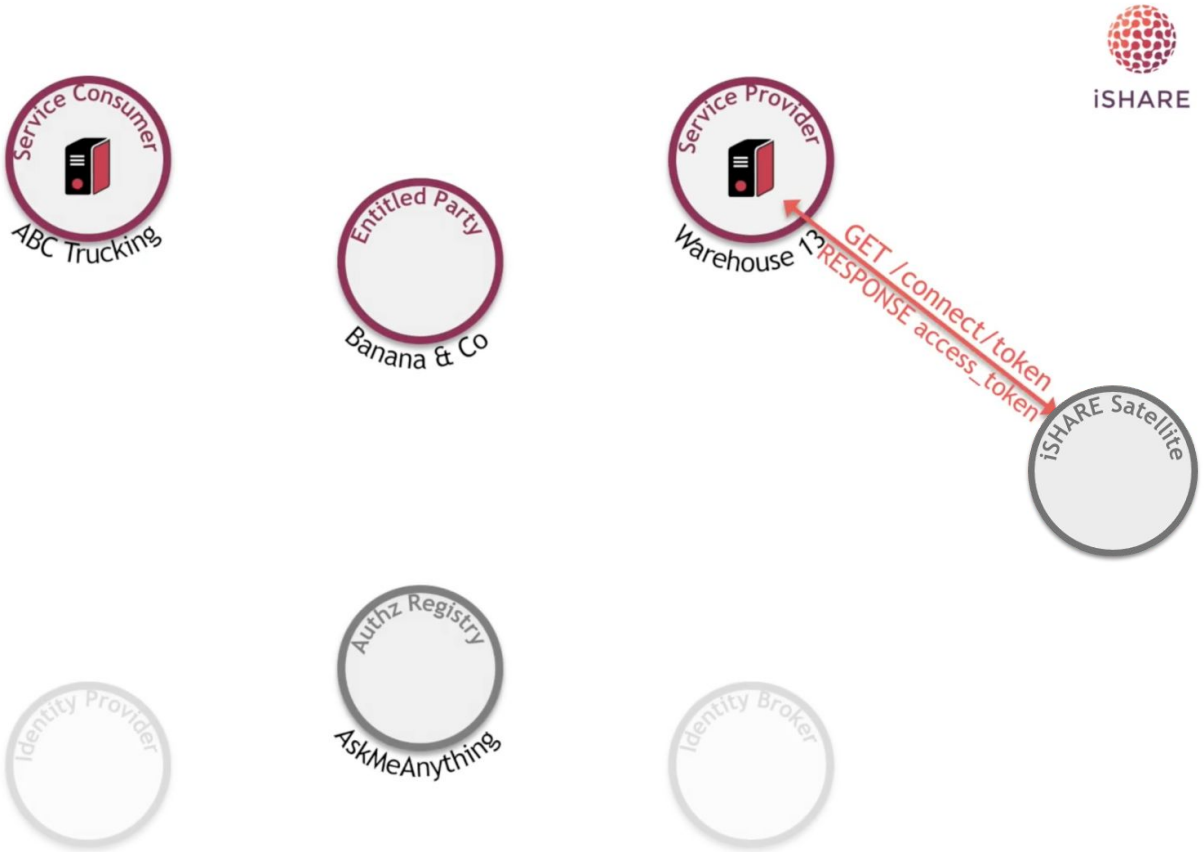
1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange



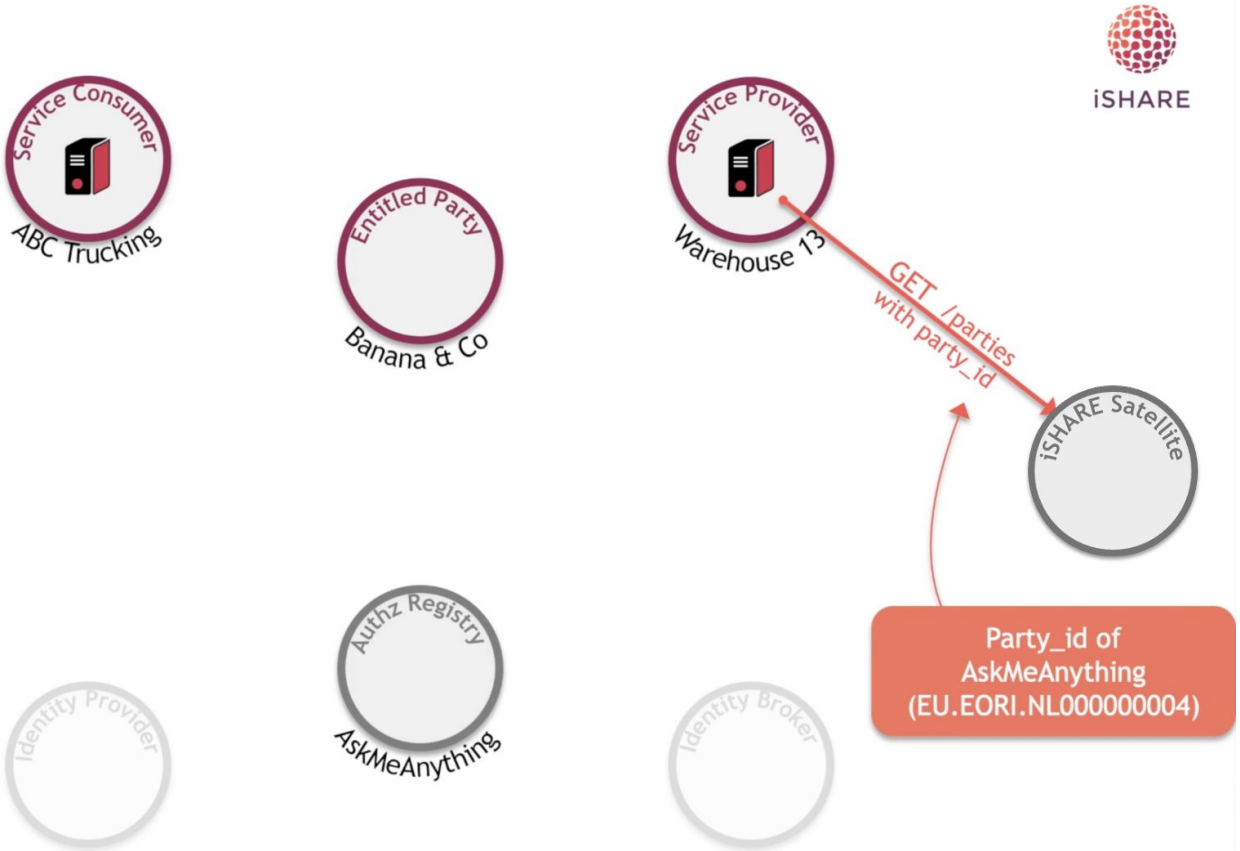
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange



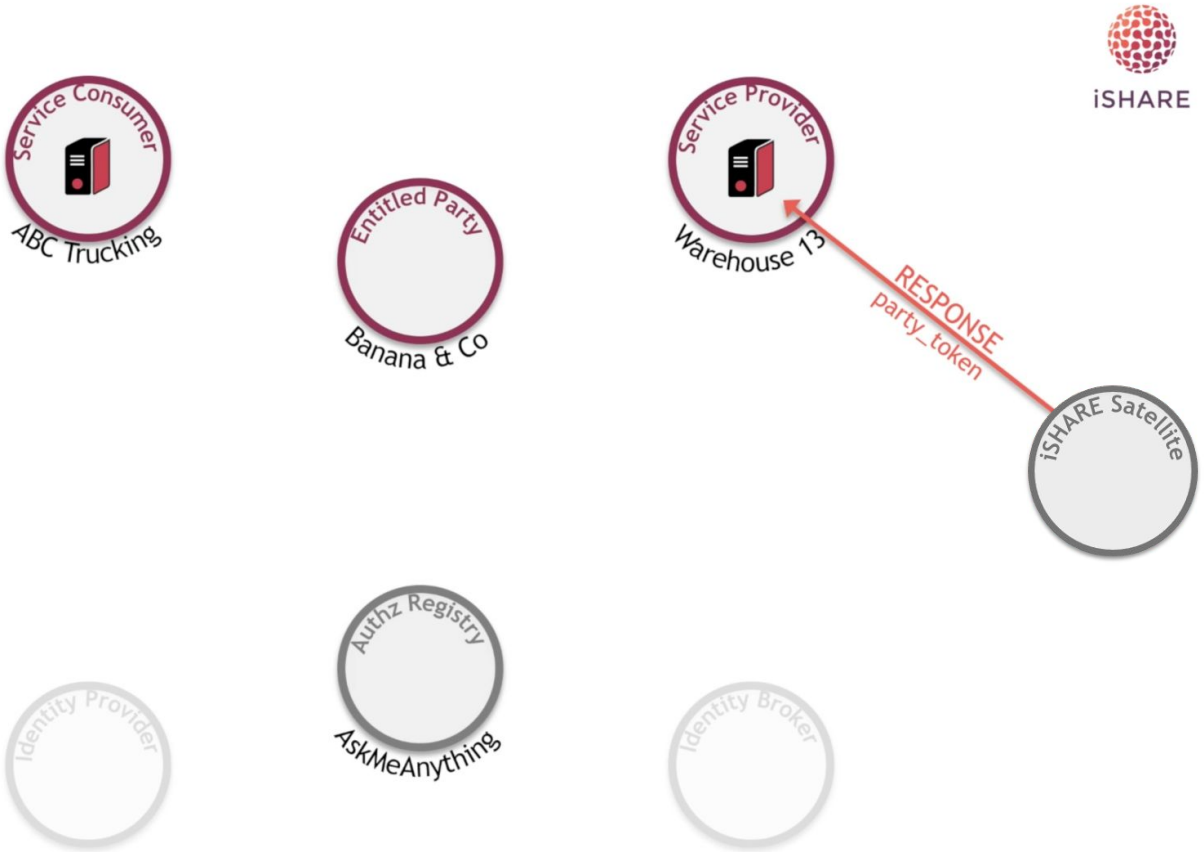
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange



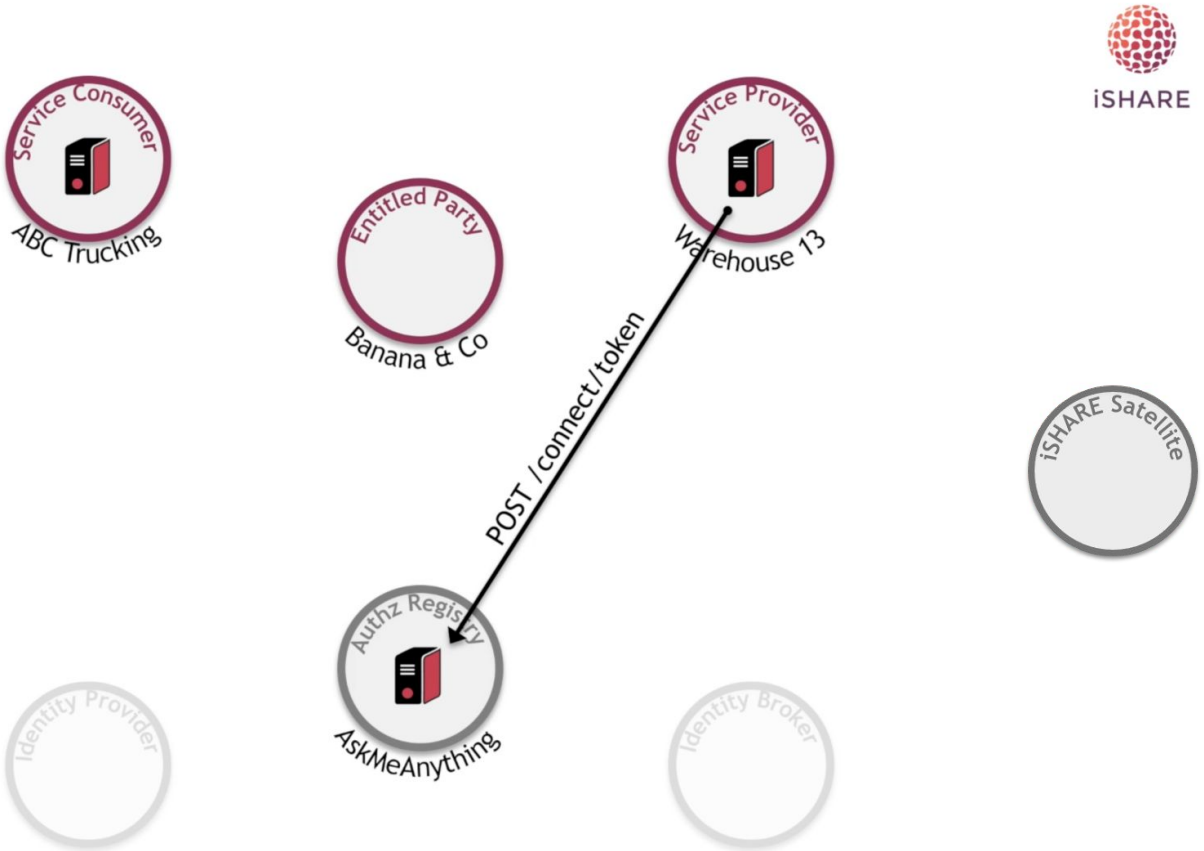
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange

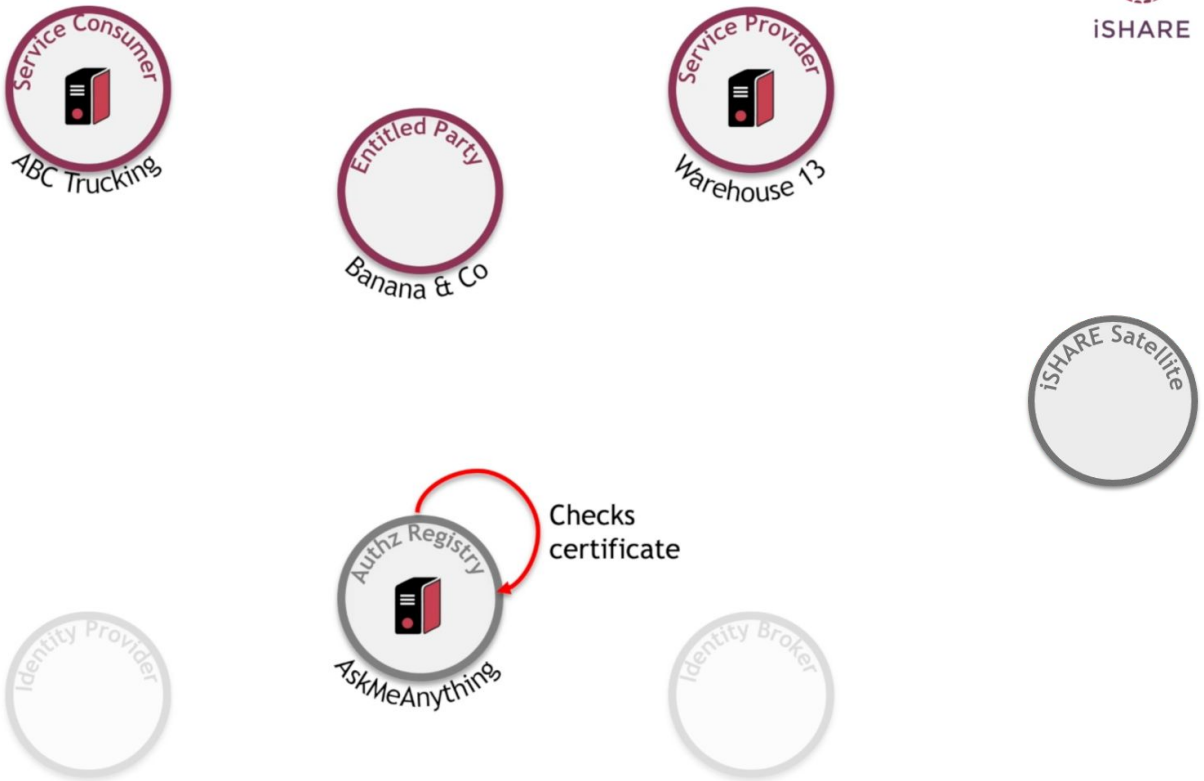


Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange



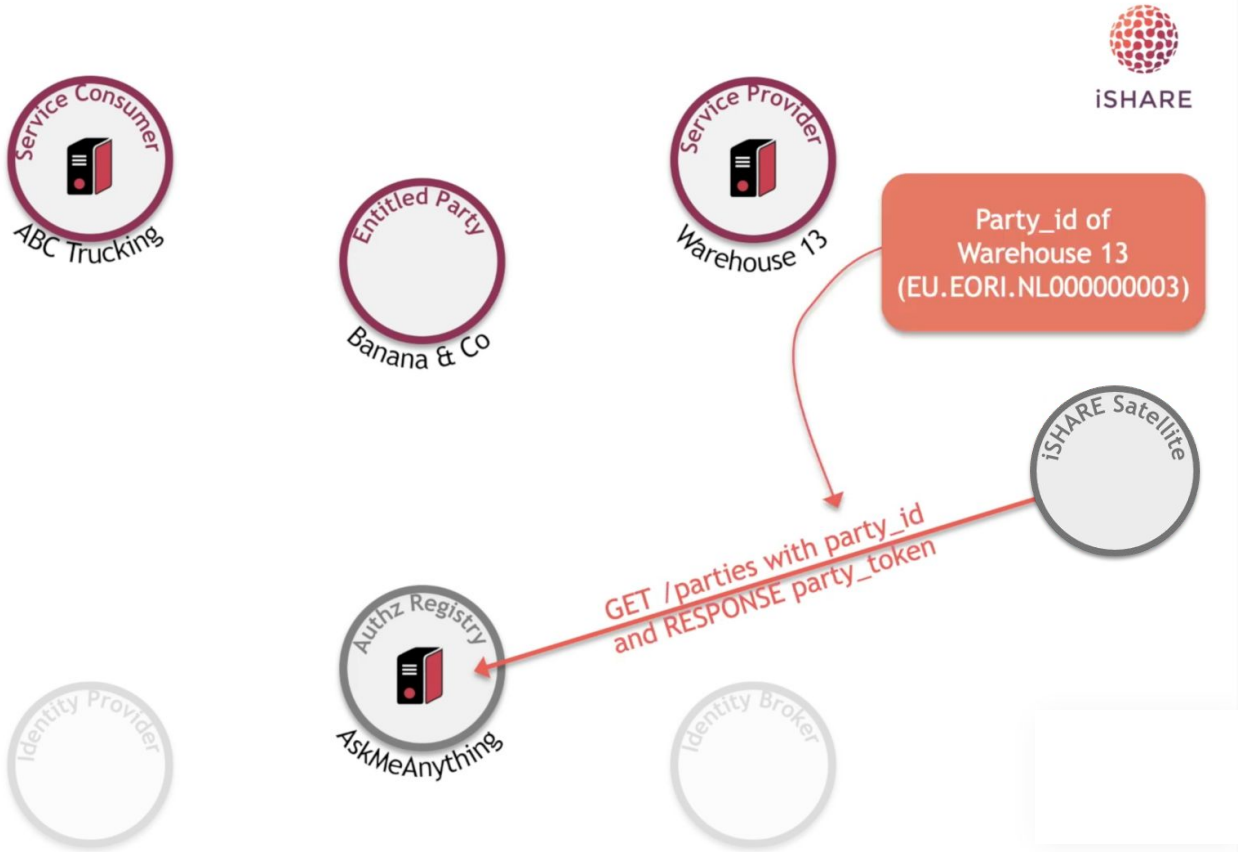
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange



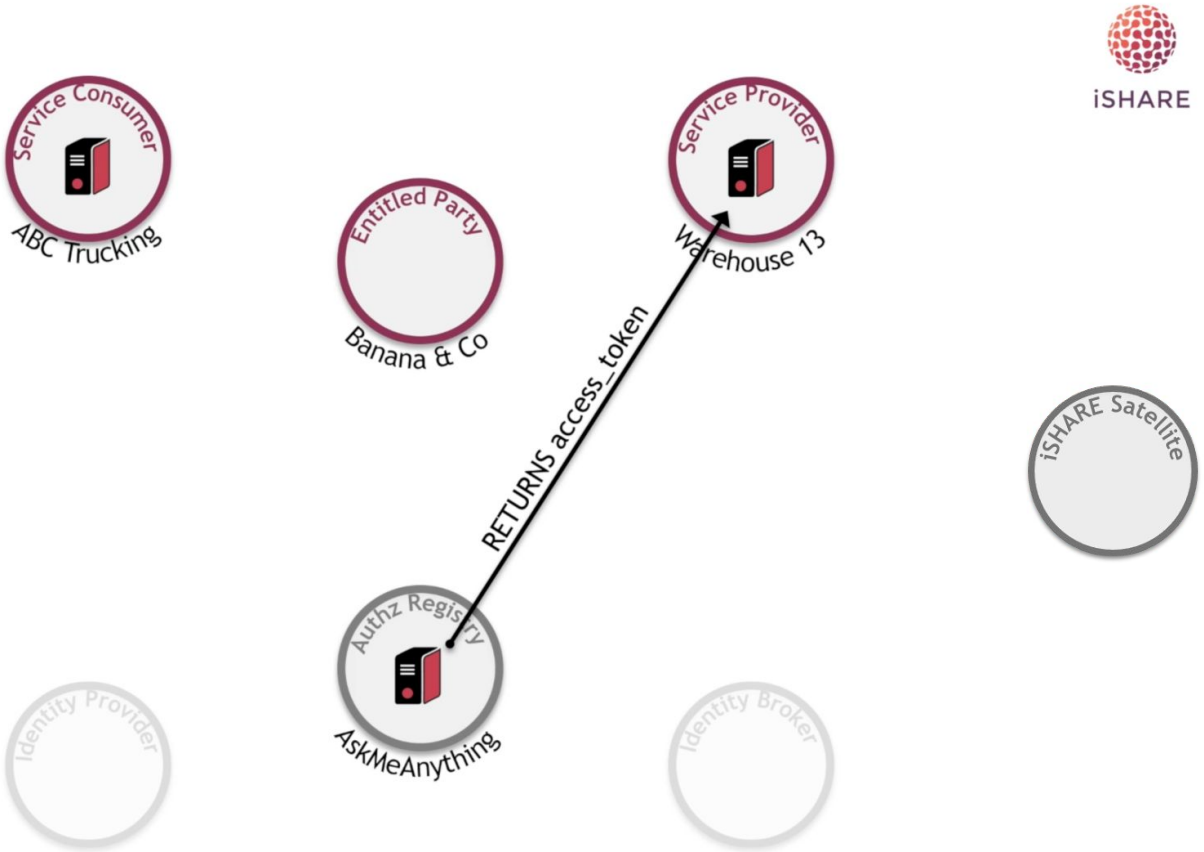
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. **Return access token**
5. Delegation evidence
6. Data exchange



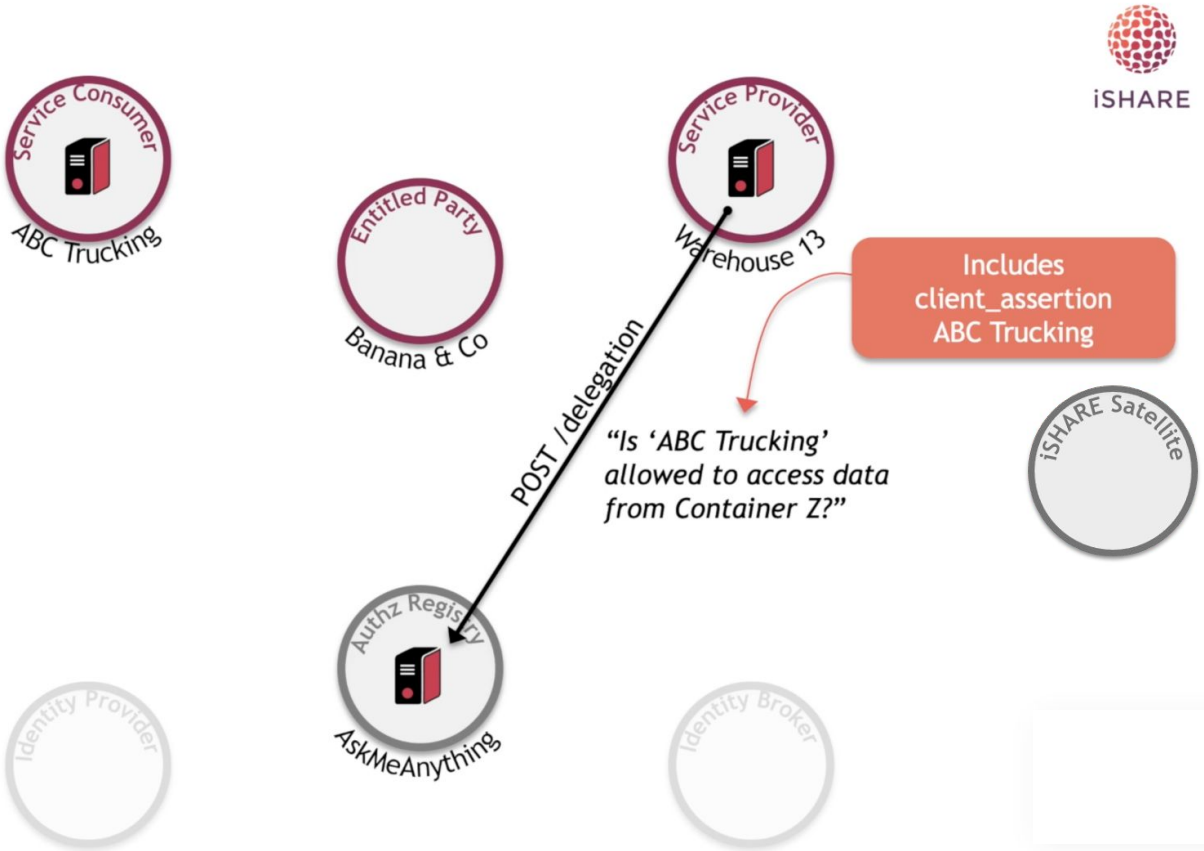
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. **Delegation evidence**
6. Data exchange



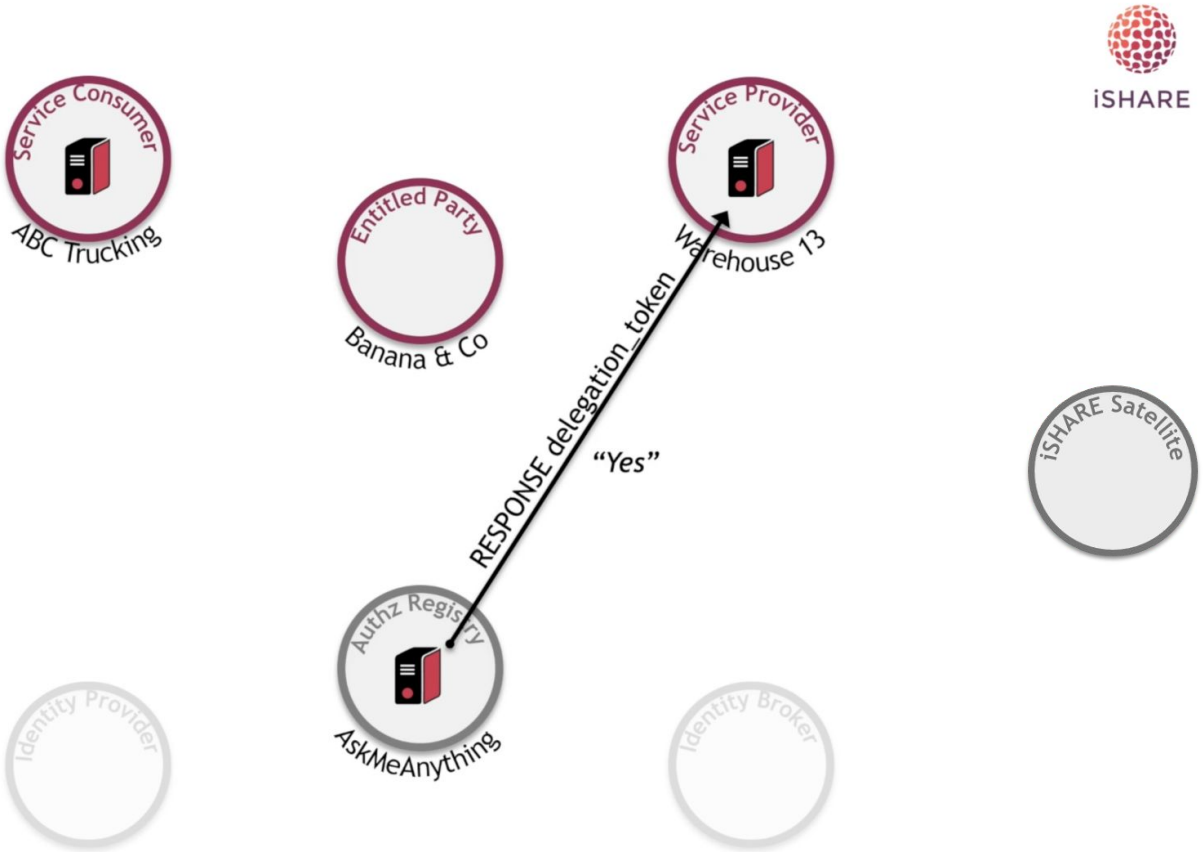
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. **Delegation evidence**
6. Data exchange



i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. **Delegation evidence**
6. Data exchange

Checks evidence and certificate

iSHARE

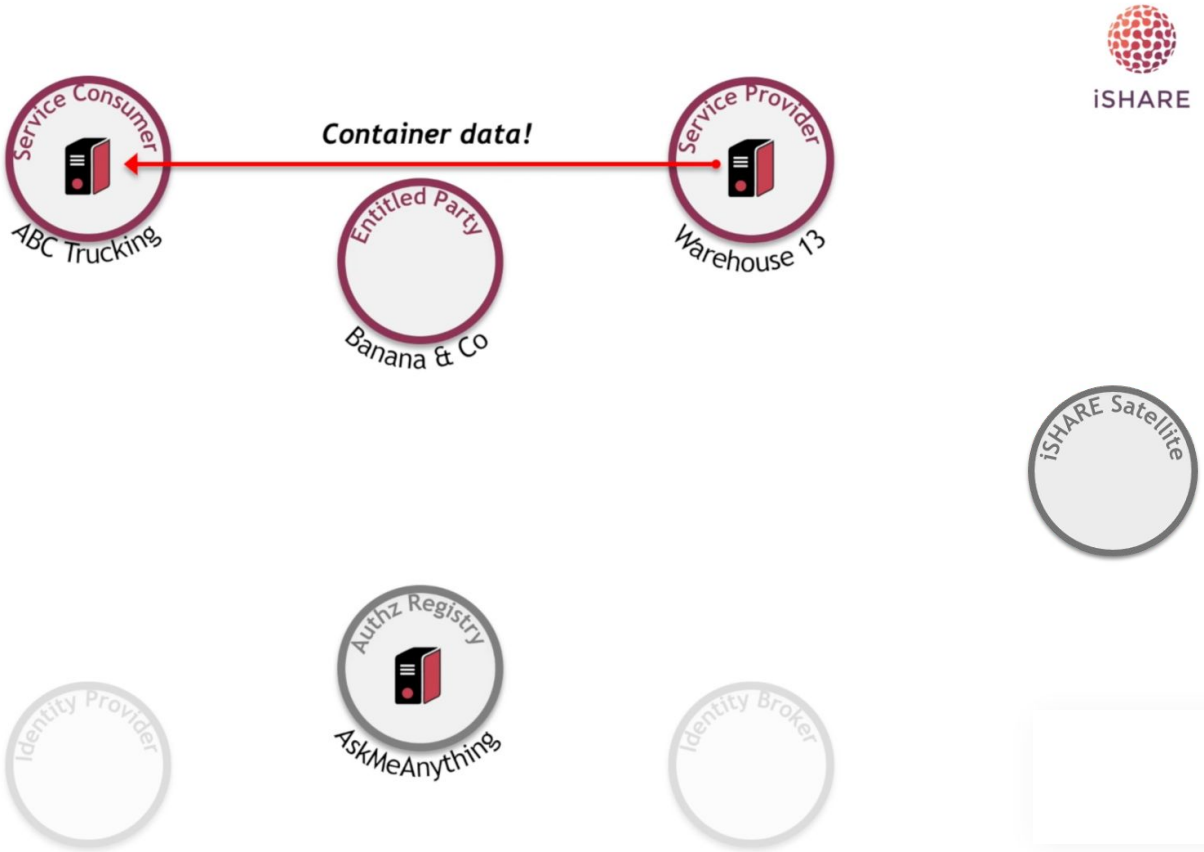
i4Trust

Case 1a
M2M

ABC Trucking wants to access container data at the terminal, on behalf of Banana & Co

Authentication & exchange

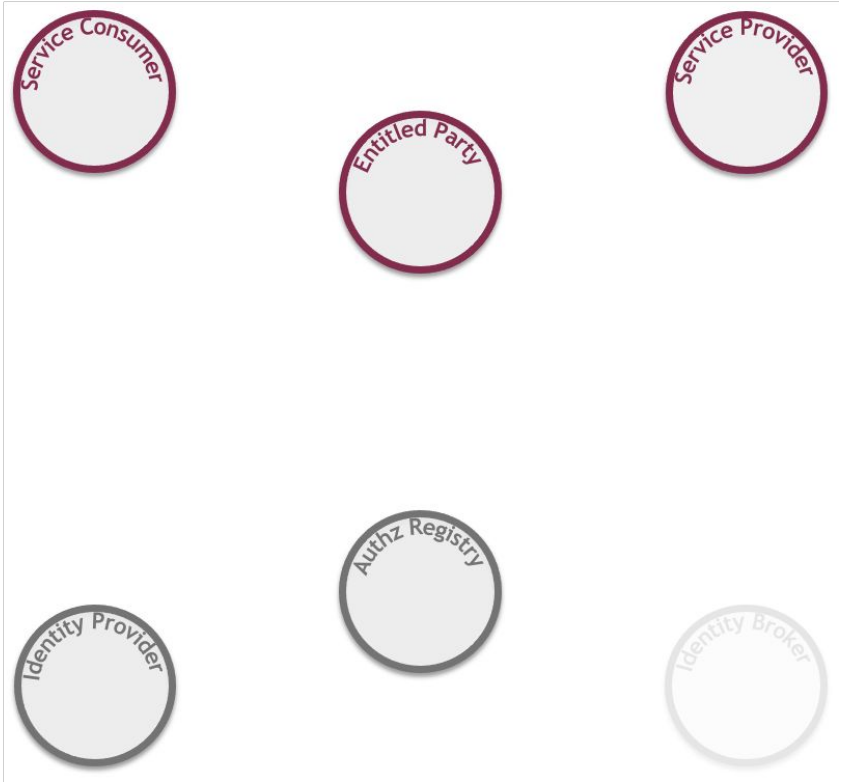
1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange



i4Trust

Participants of the fictional use case

- **Banana & Co**
  Frieght owner, Entitled Party
- **Warehouse 13**
  Terminal, Service Provider
- **Driver X**
  Trucker of ABC Trucking, Service Consumer
- **Ishare enabled ("Registry X")**
  Authorisation Registry
- **Secure Logistics**
  Identity Provider



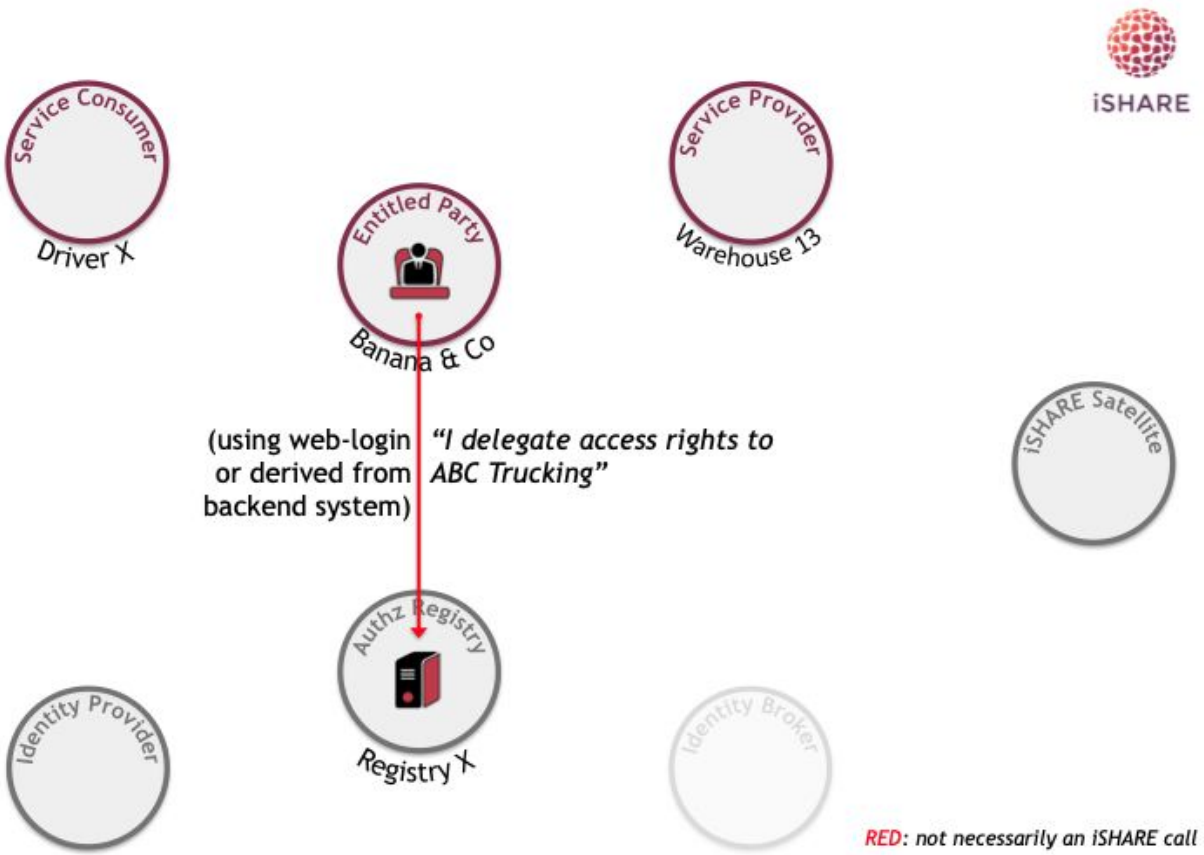
i4Trust

Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.

Case 2
H2M

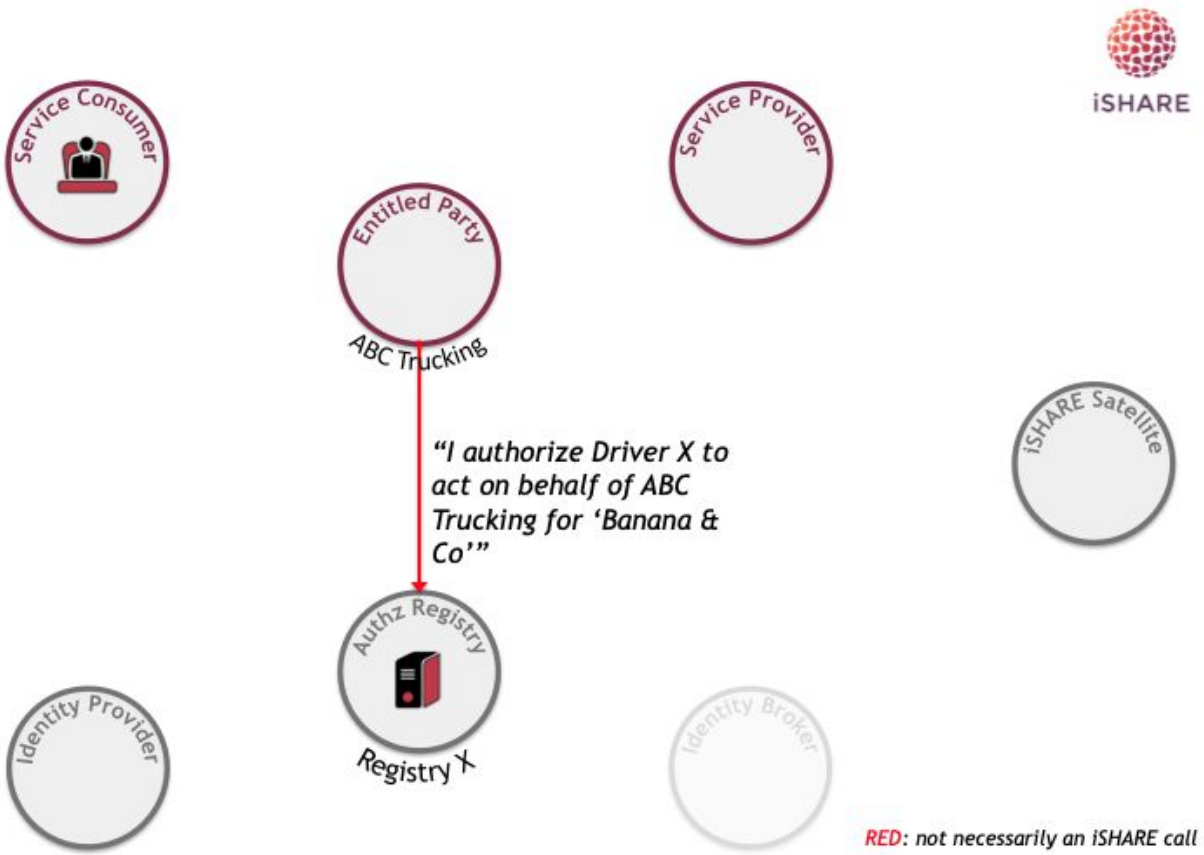
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

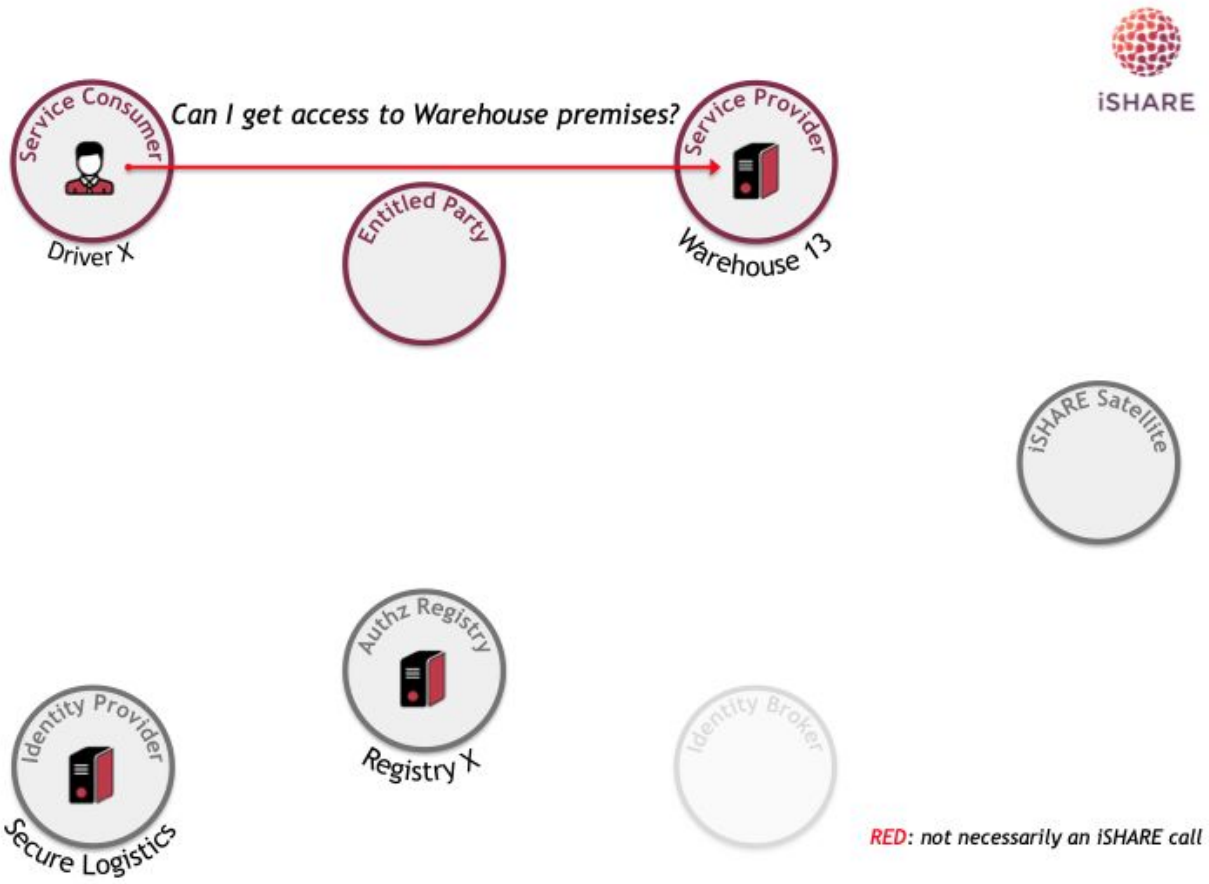
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

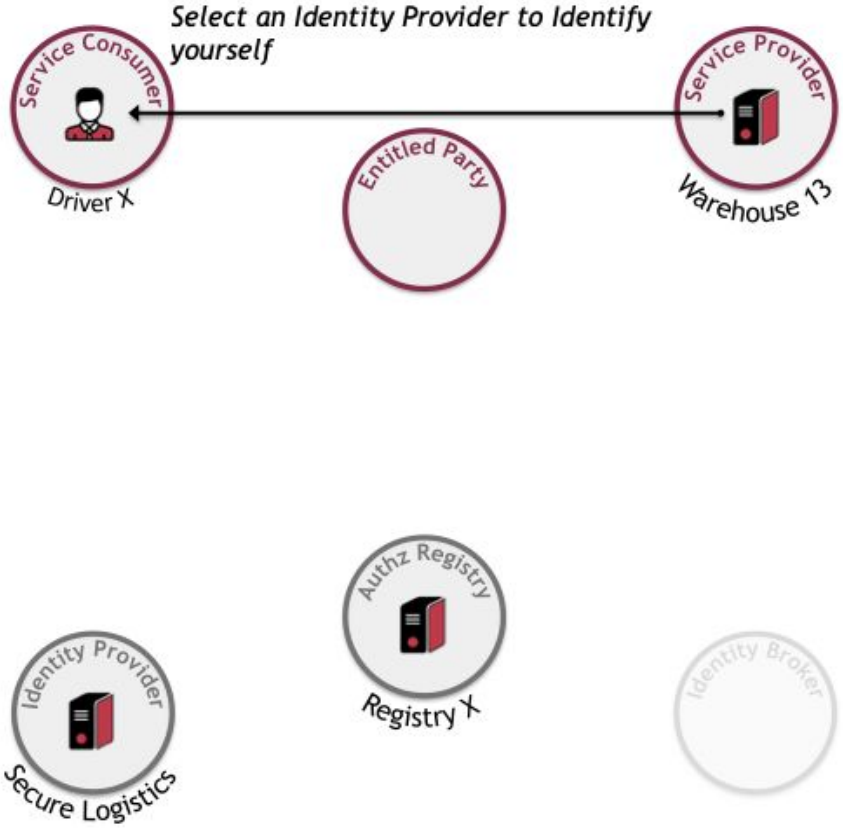
Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.

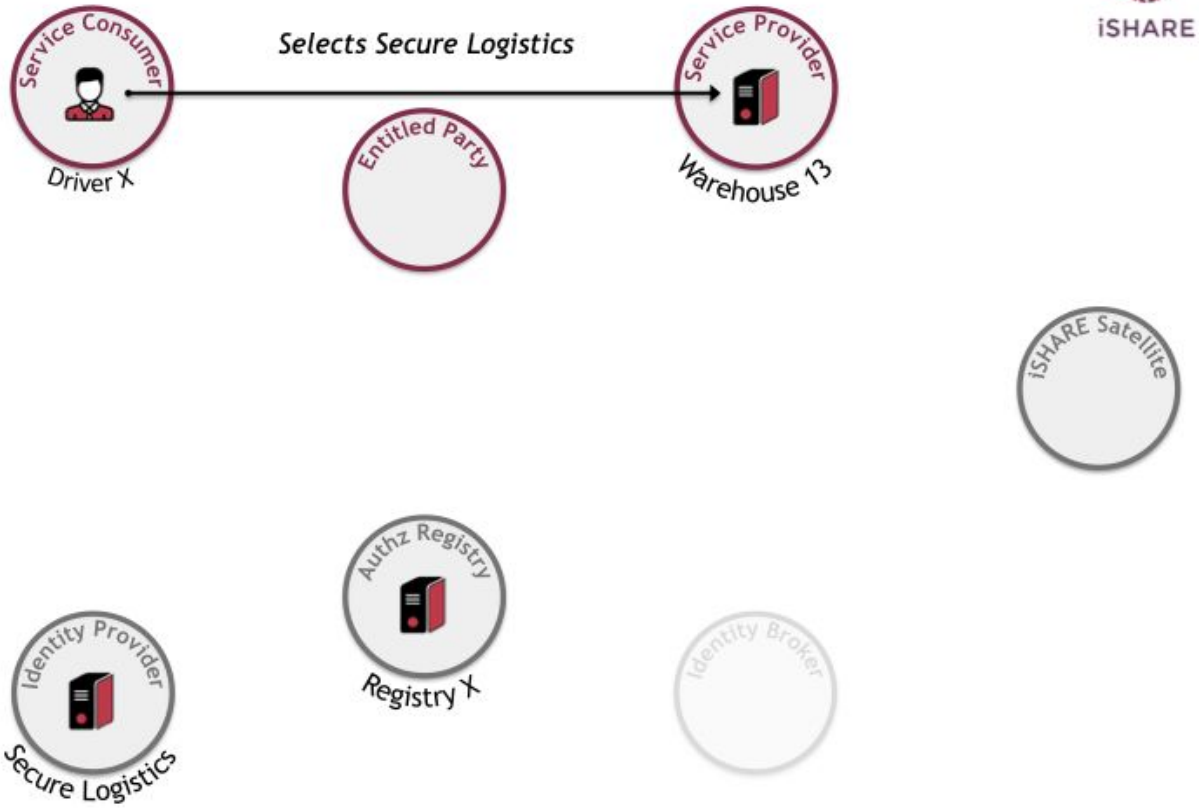


iSHARE

i4Trust

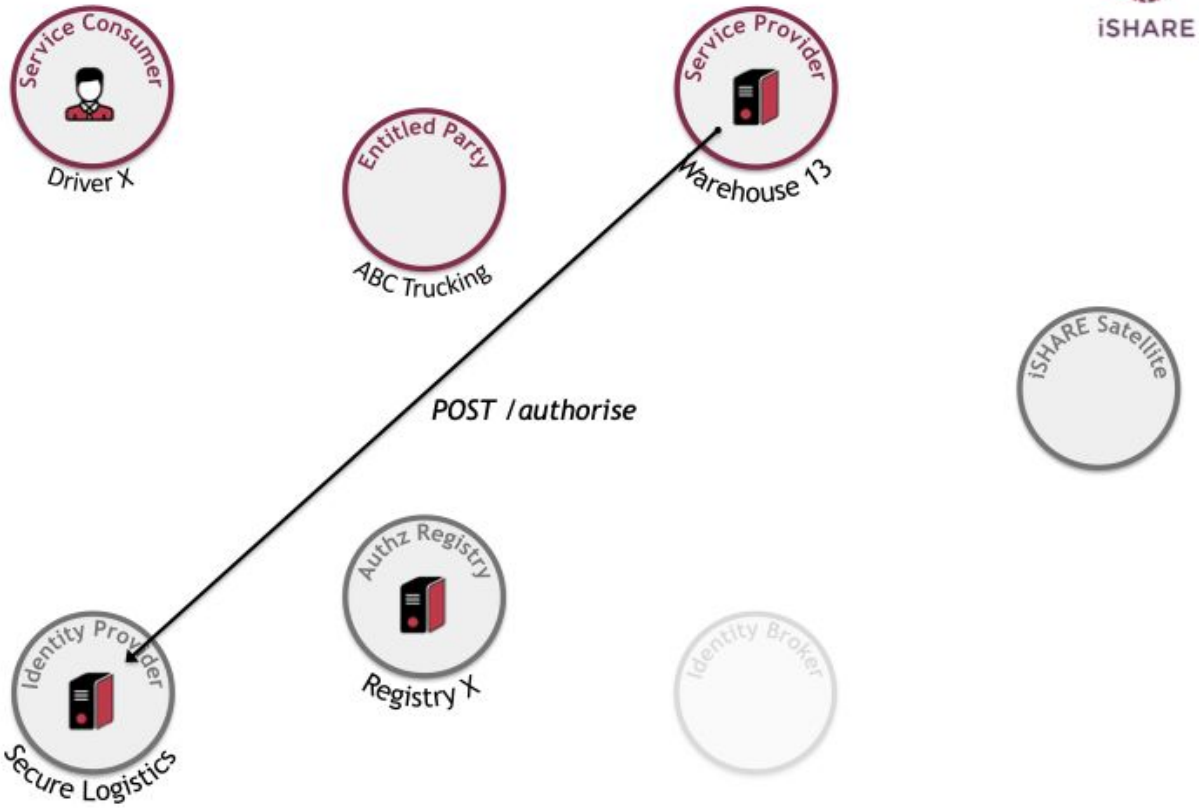
Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



iSHARE

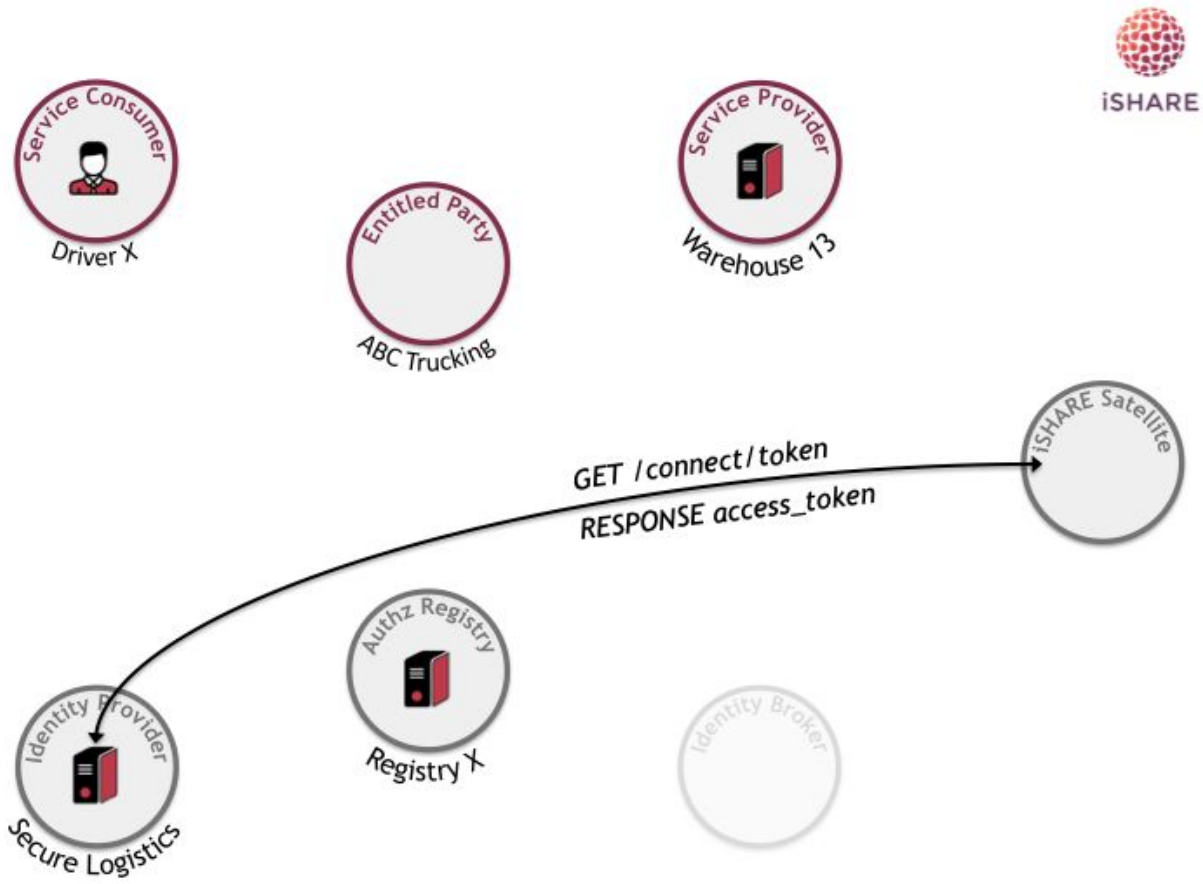
i4Trust

Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.

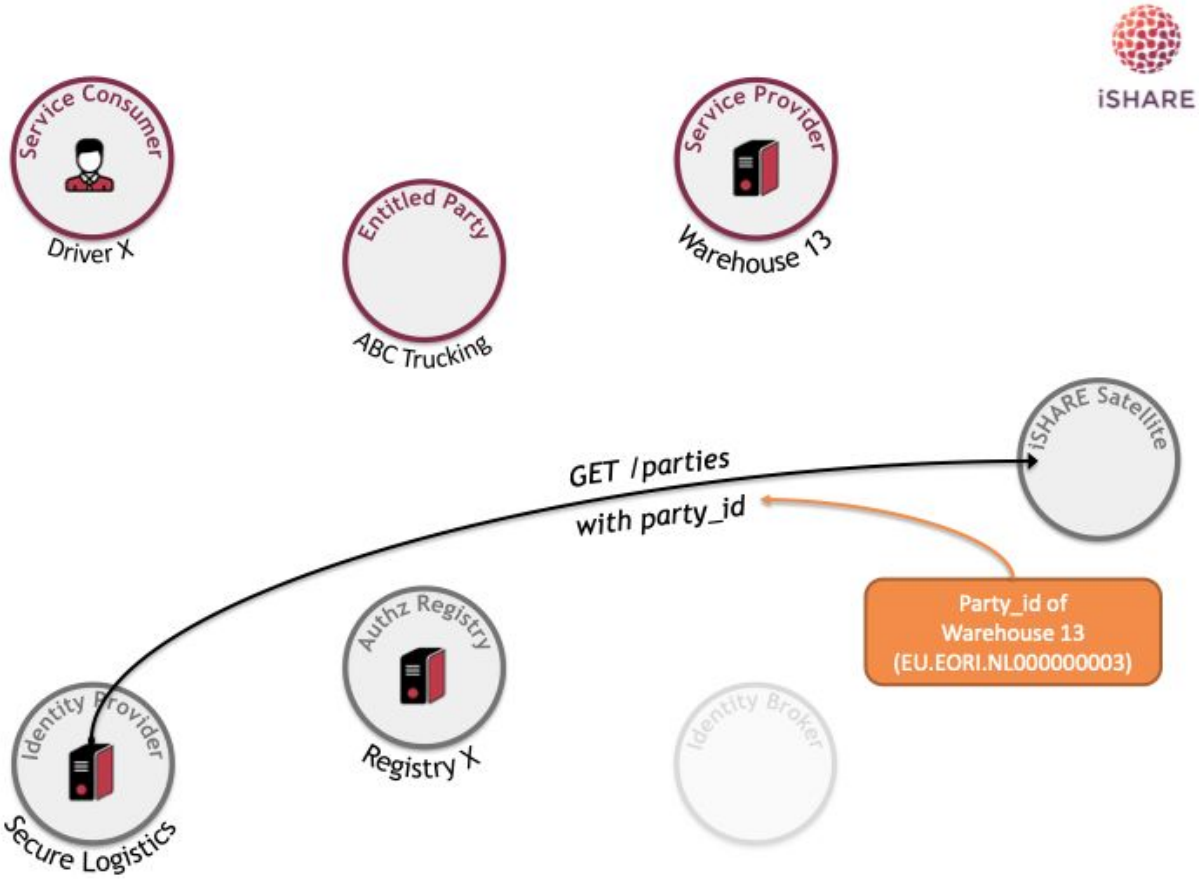
Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



Case 2
H2M

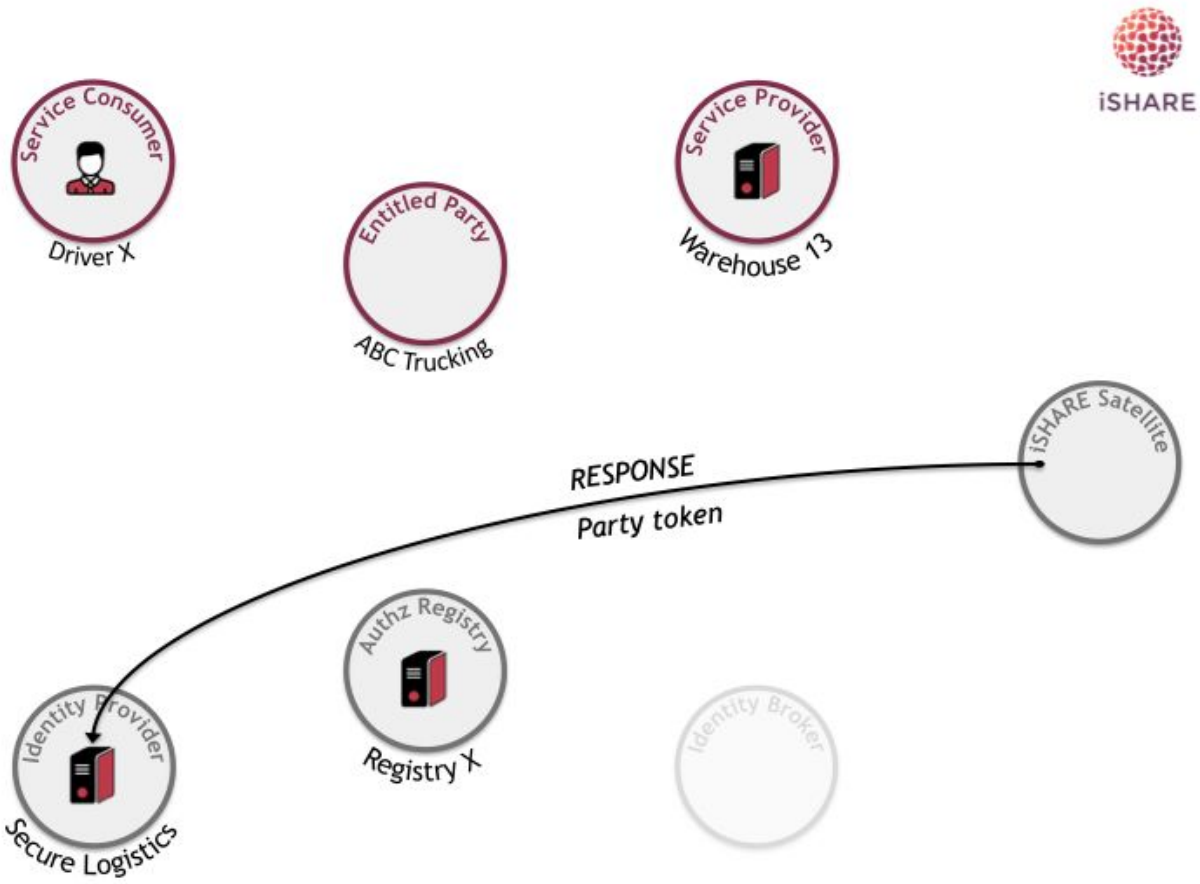
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

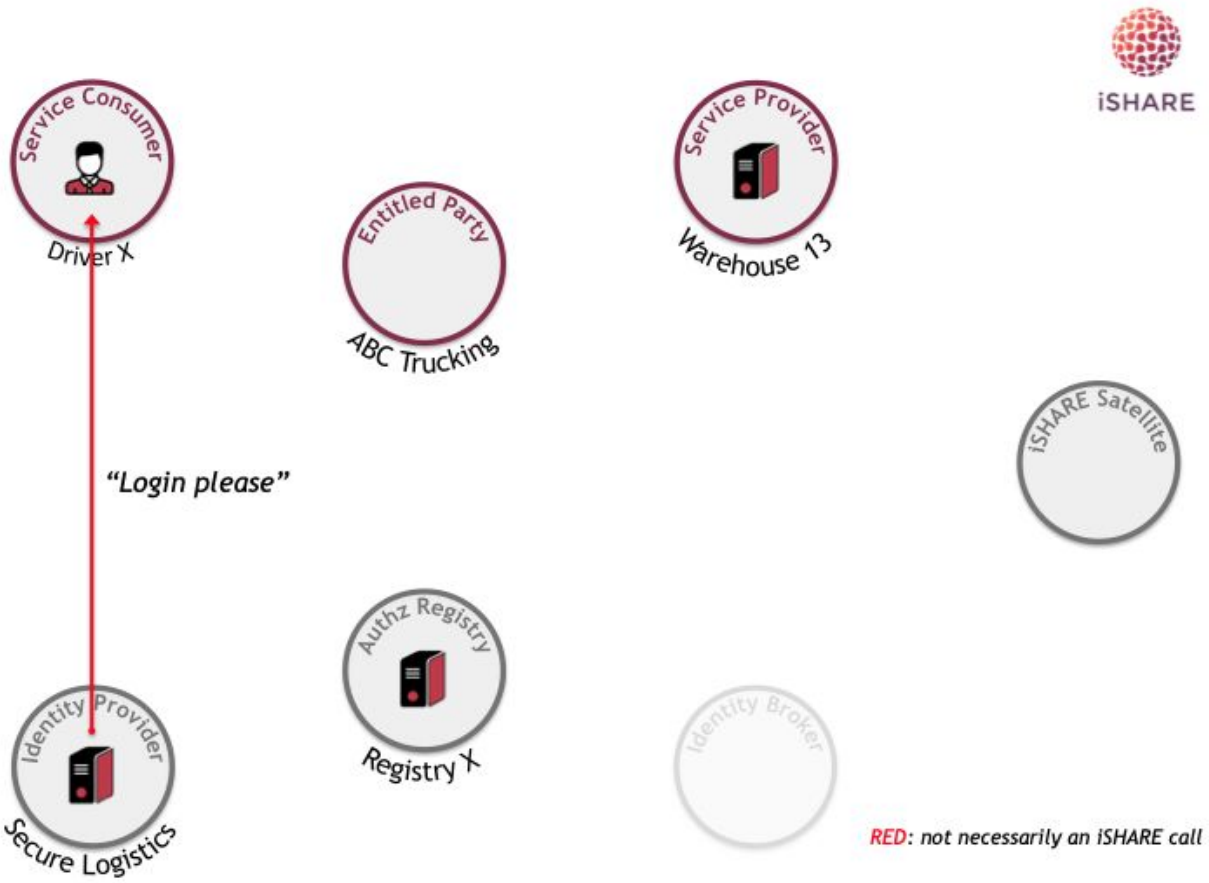
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

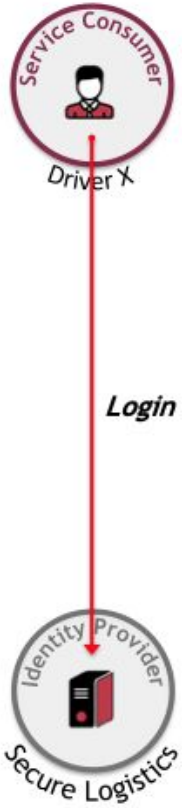
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.

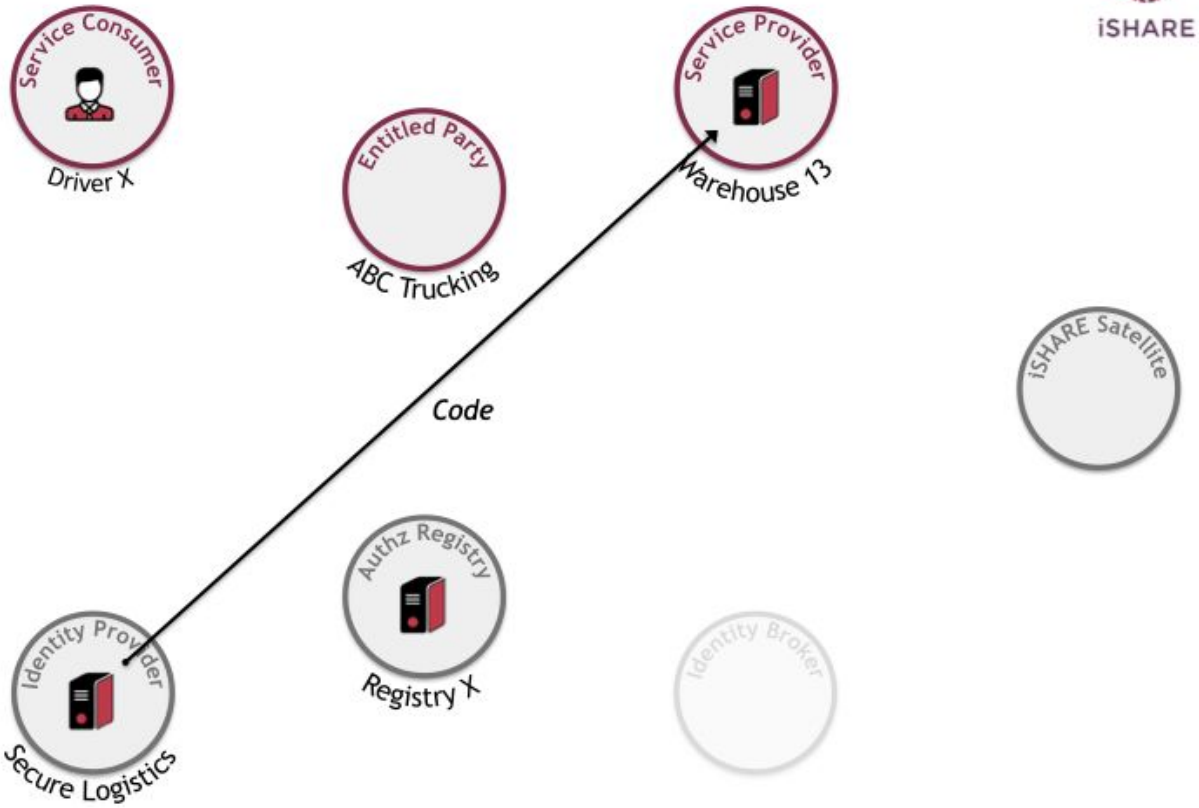


RED: not necessarily an iSHARE call

i4Trust

Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

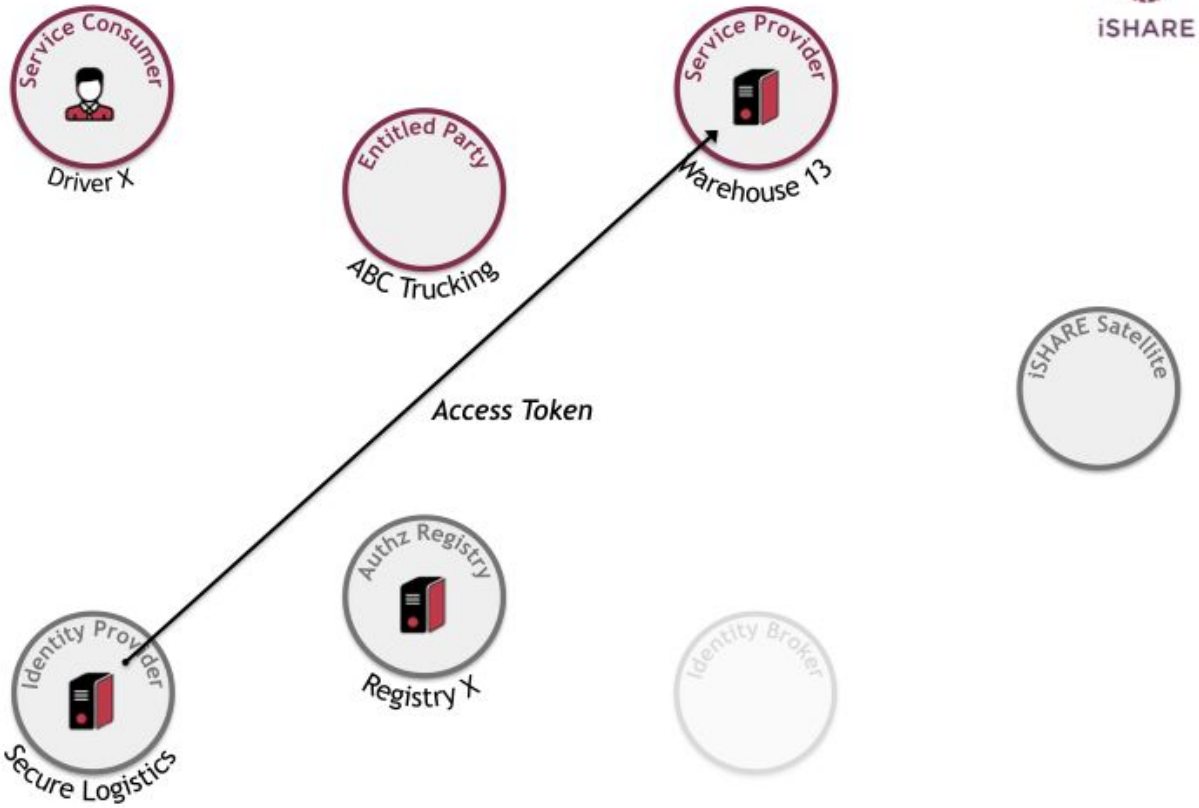
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.

iSHARE

i4Trust

Case 2
H2M

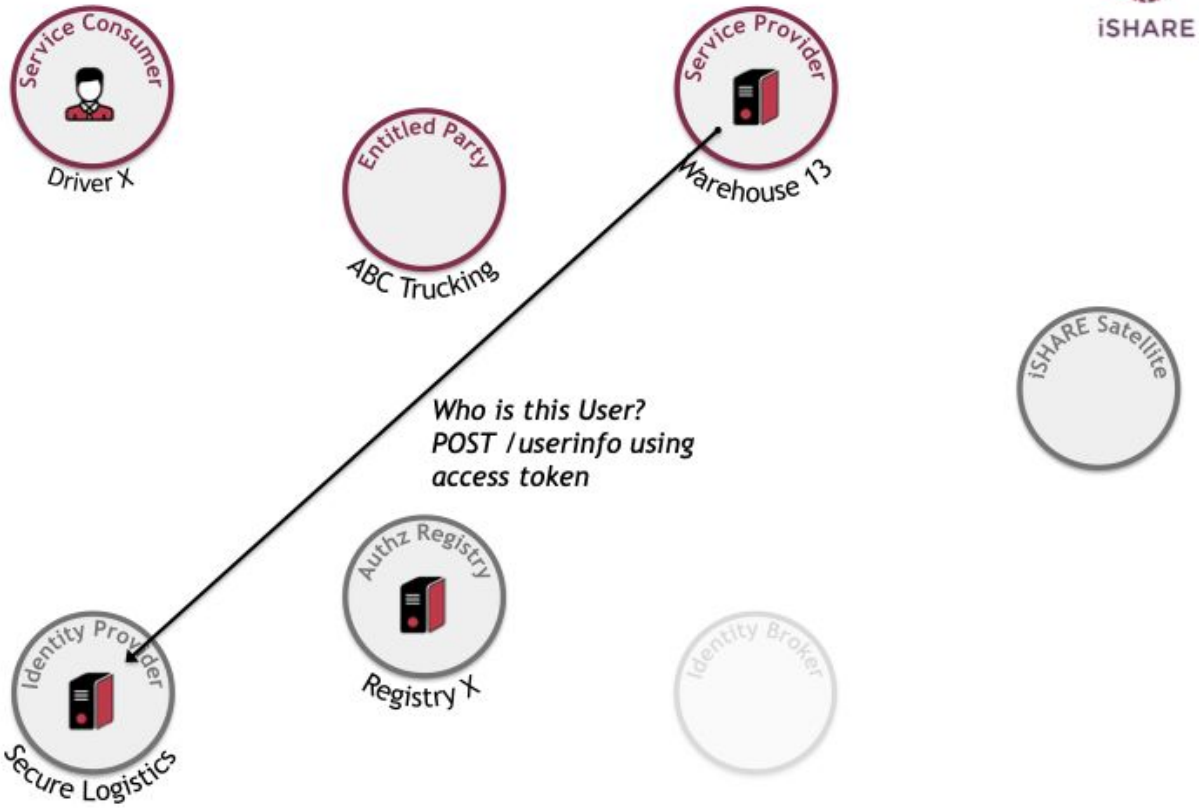
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.

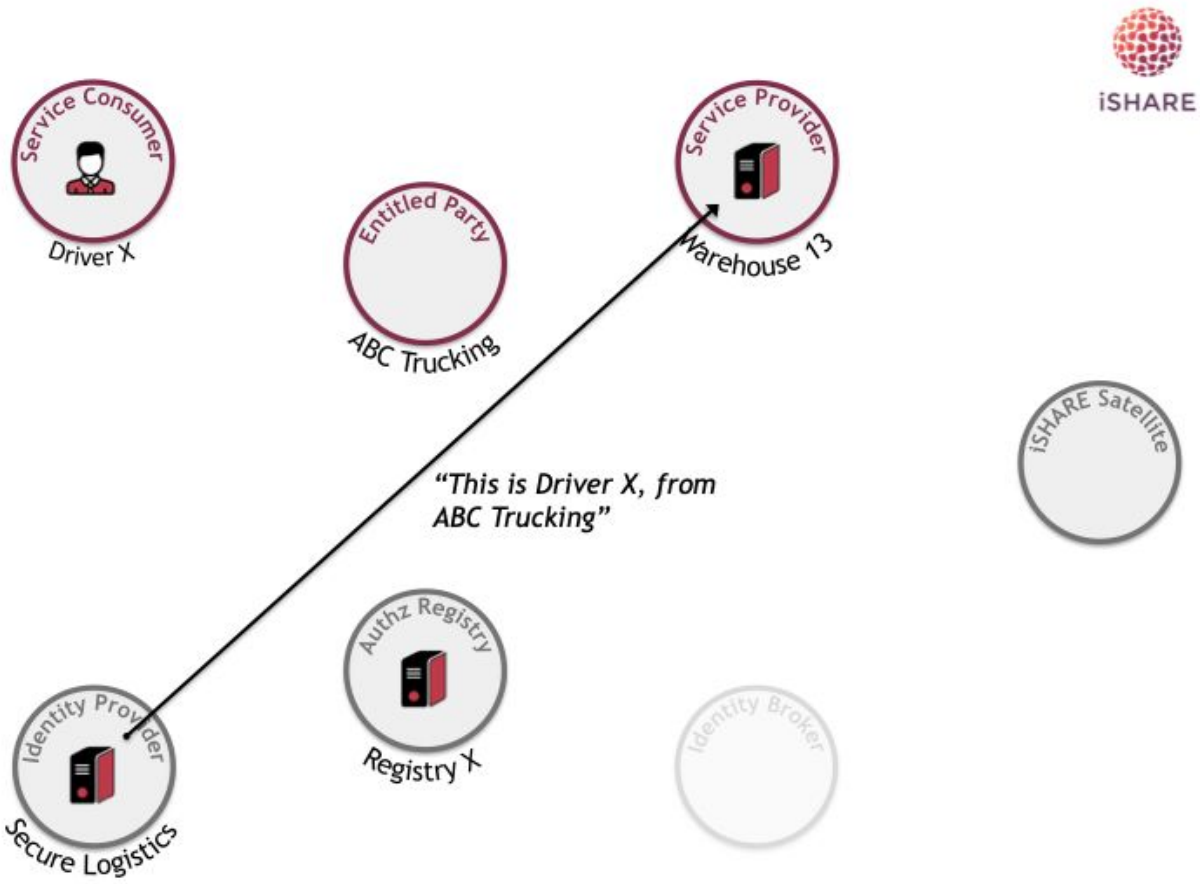


iSHARE

i4Trust

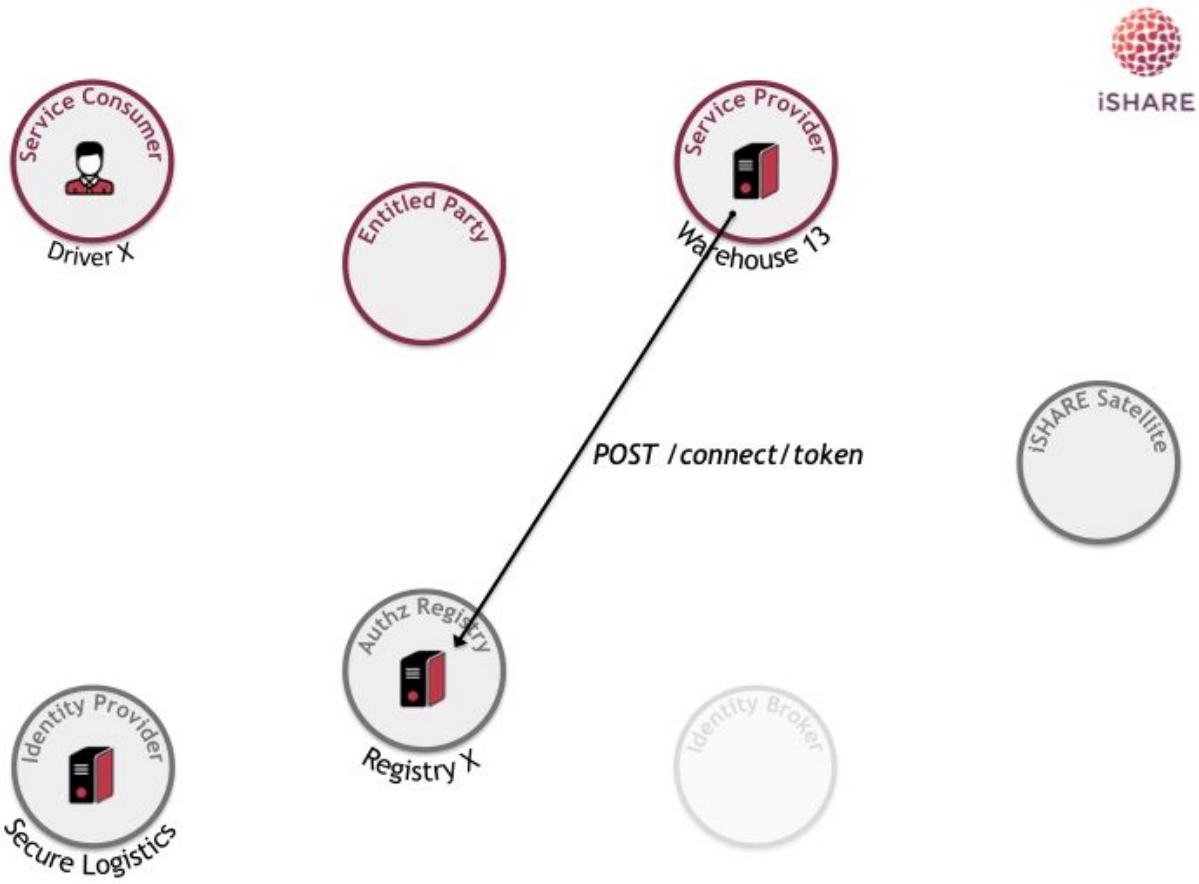
Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



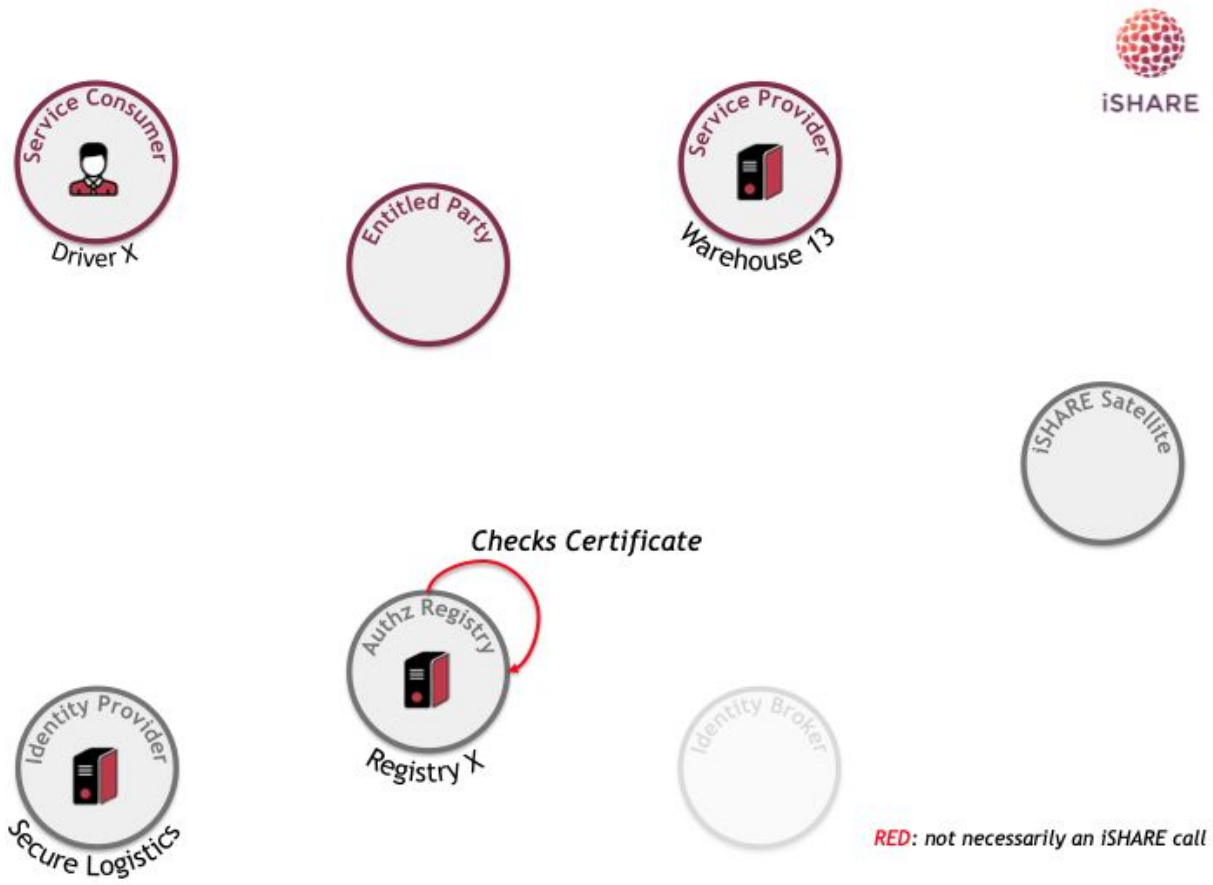
Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



Case 2
H2M

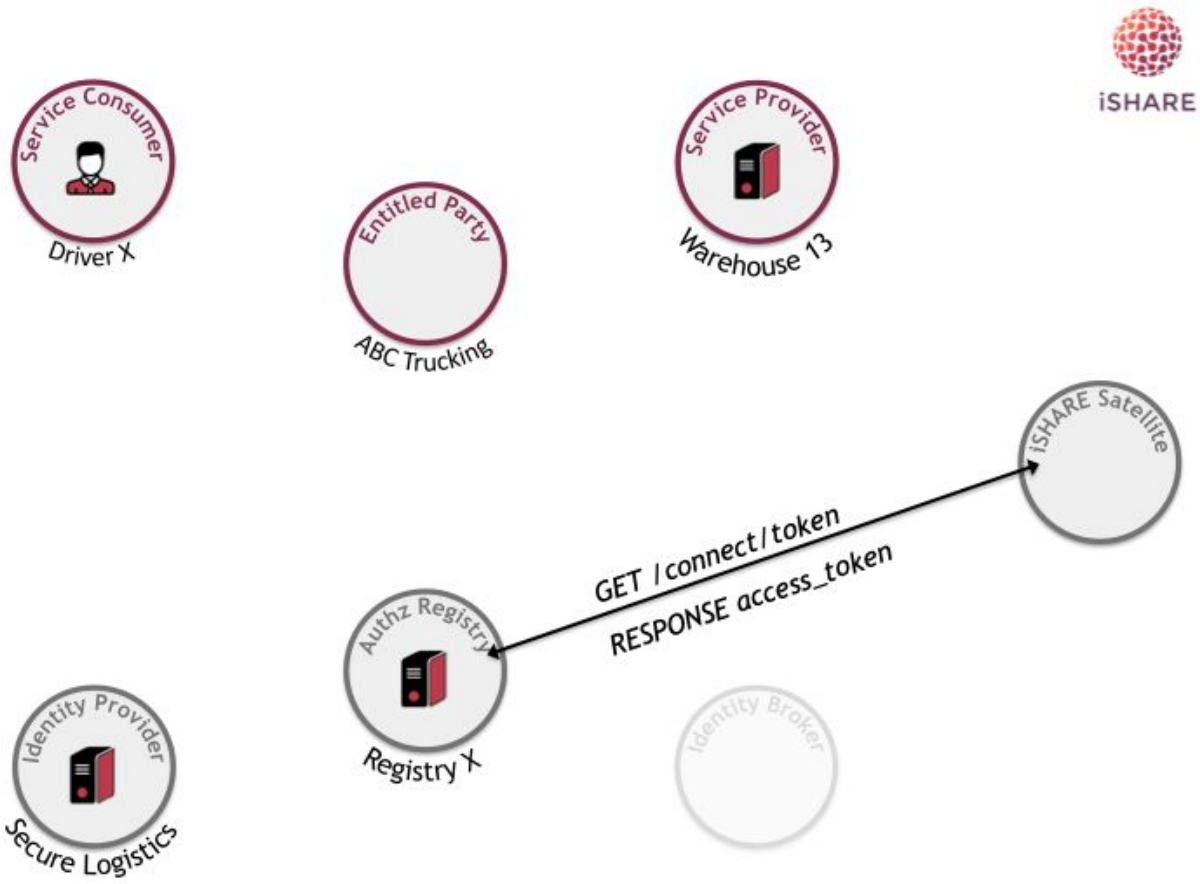
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

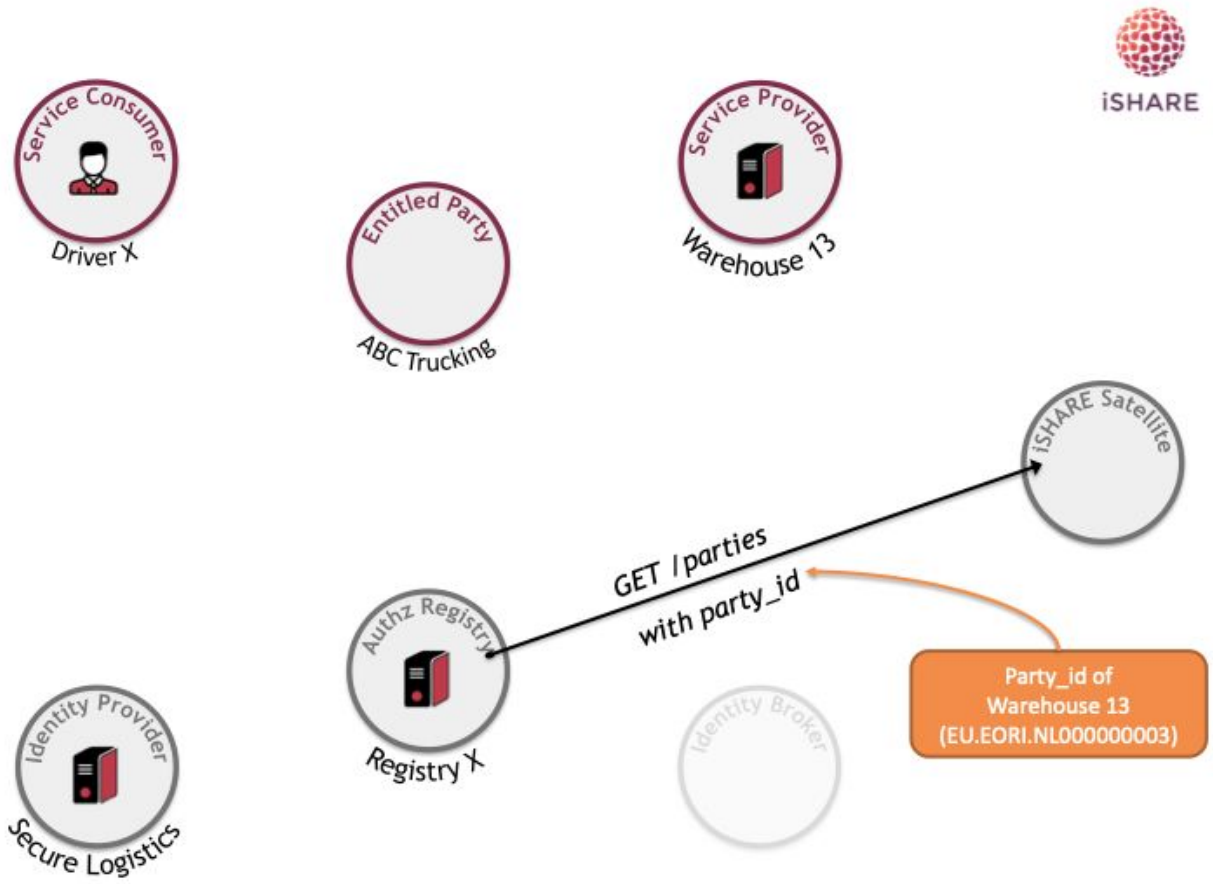
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

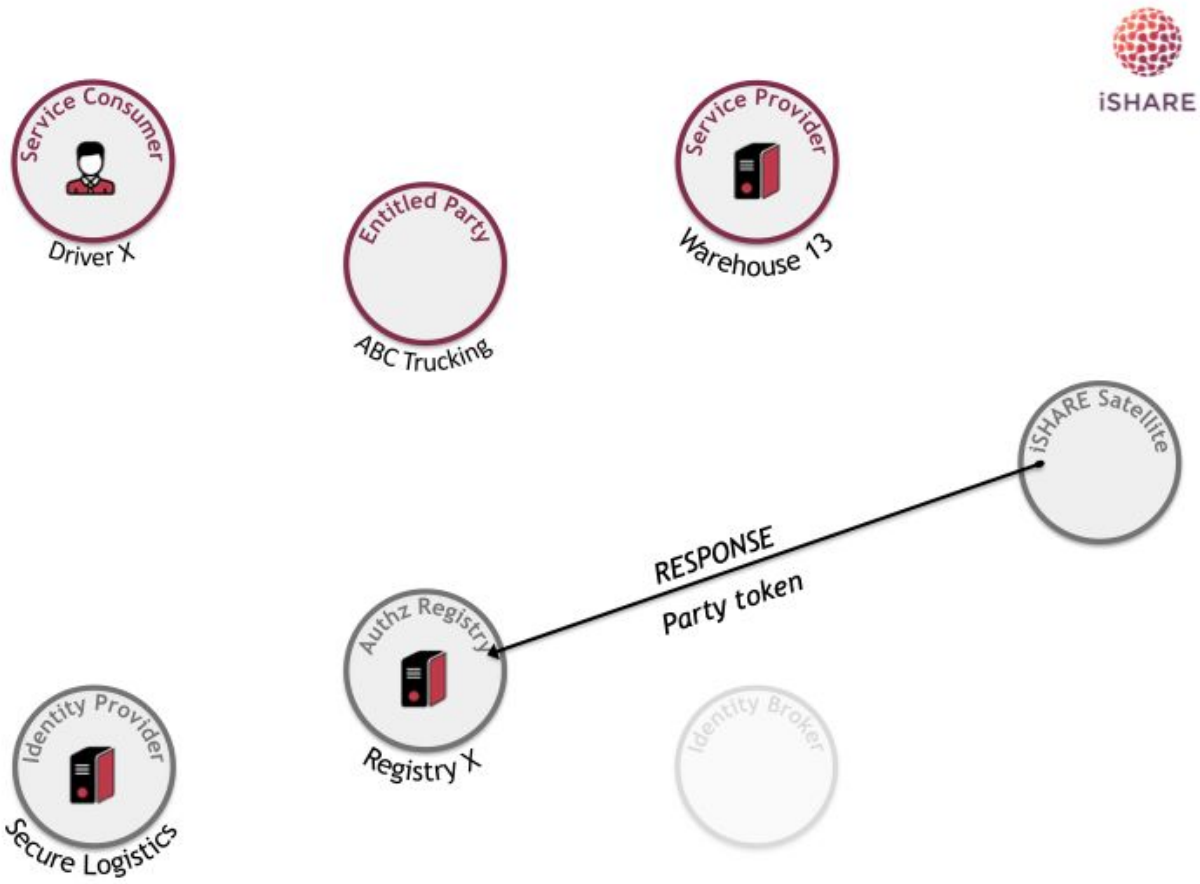
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

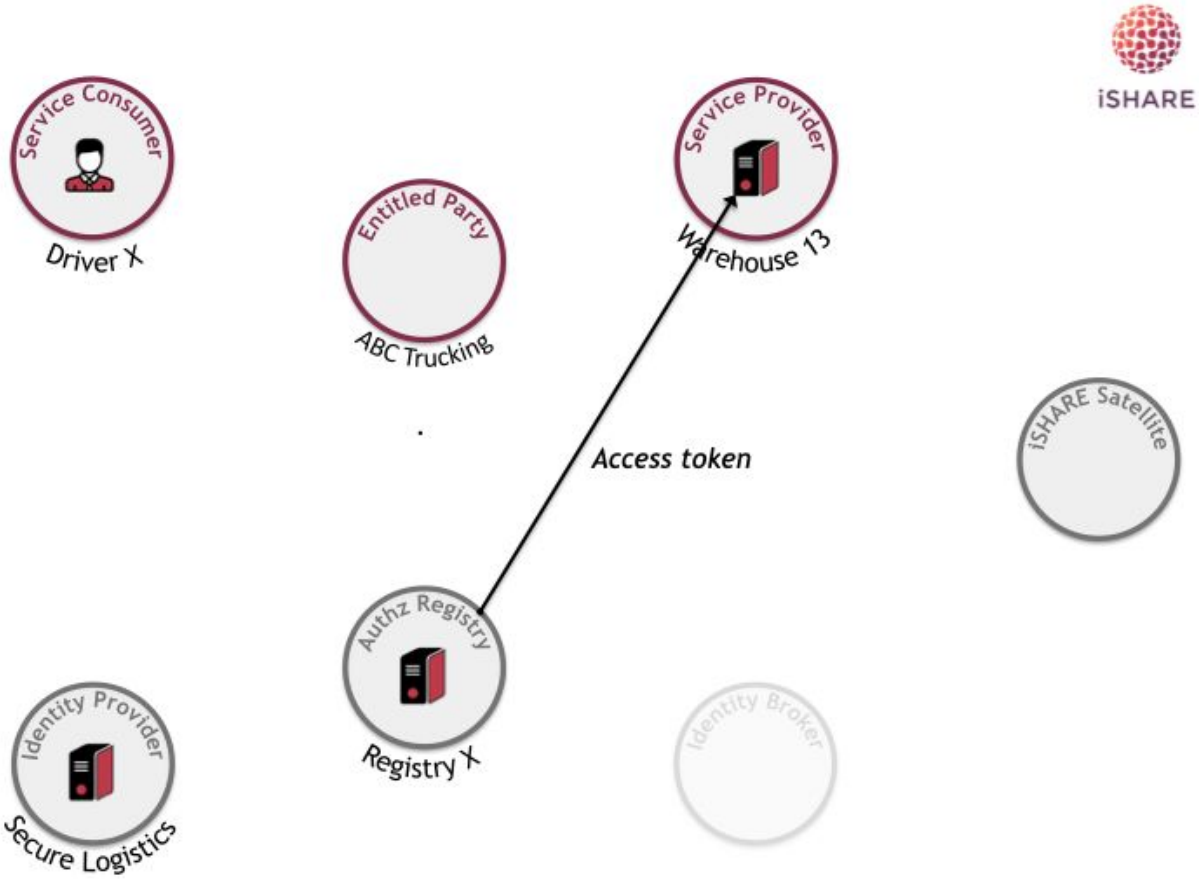
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

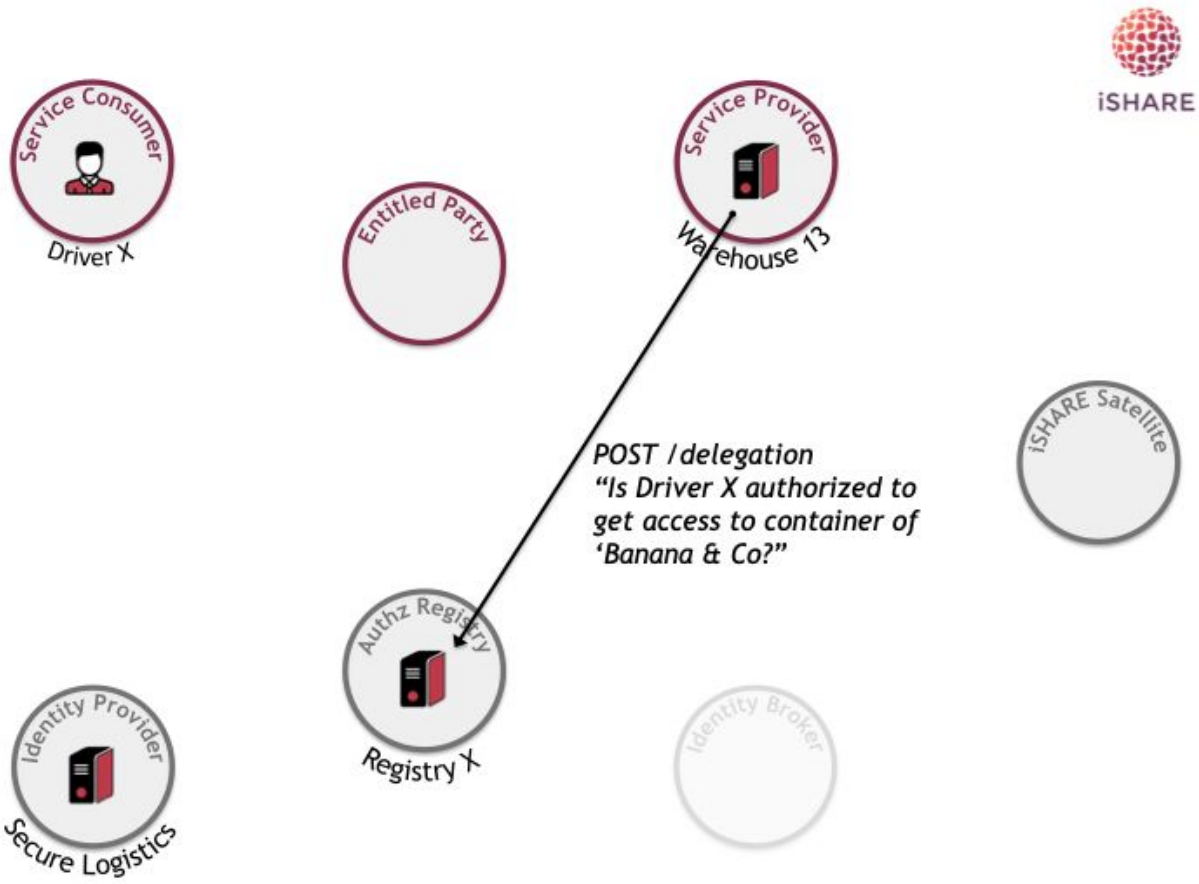
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

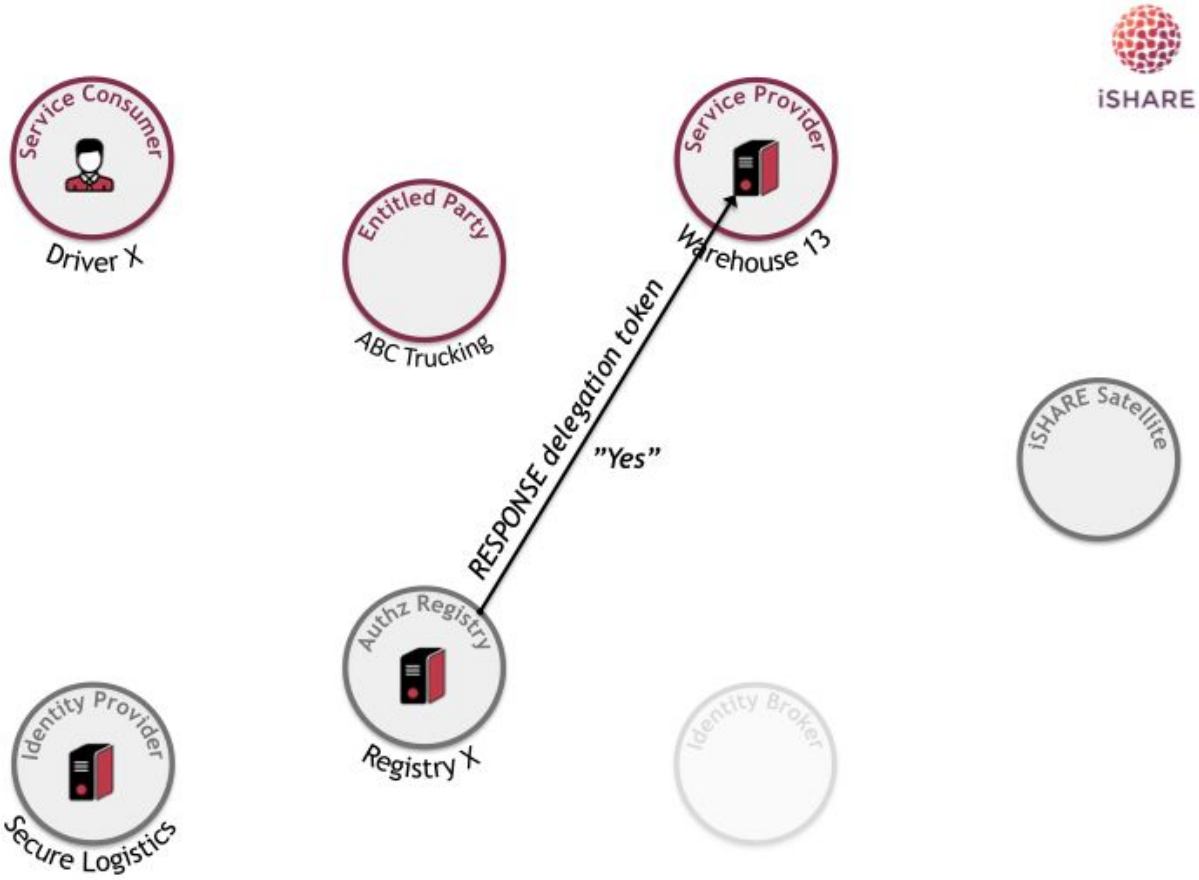
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

Case 2
H2M

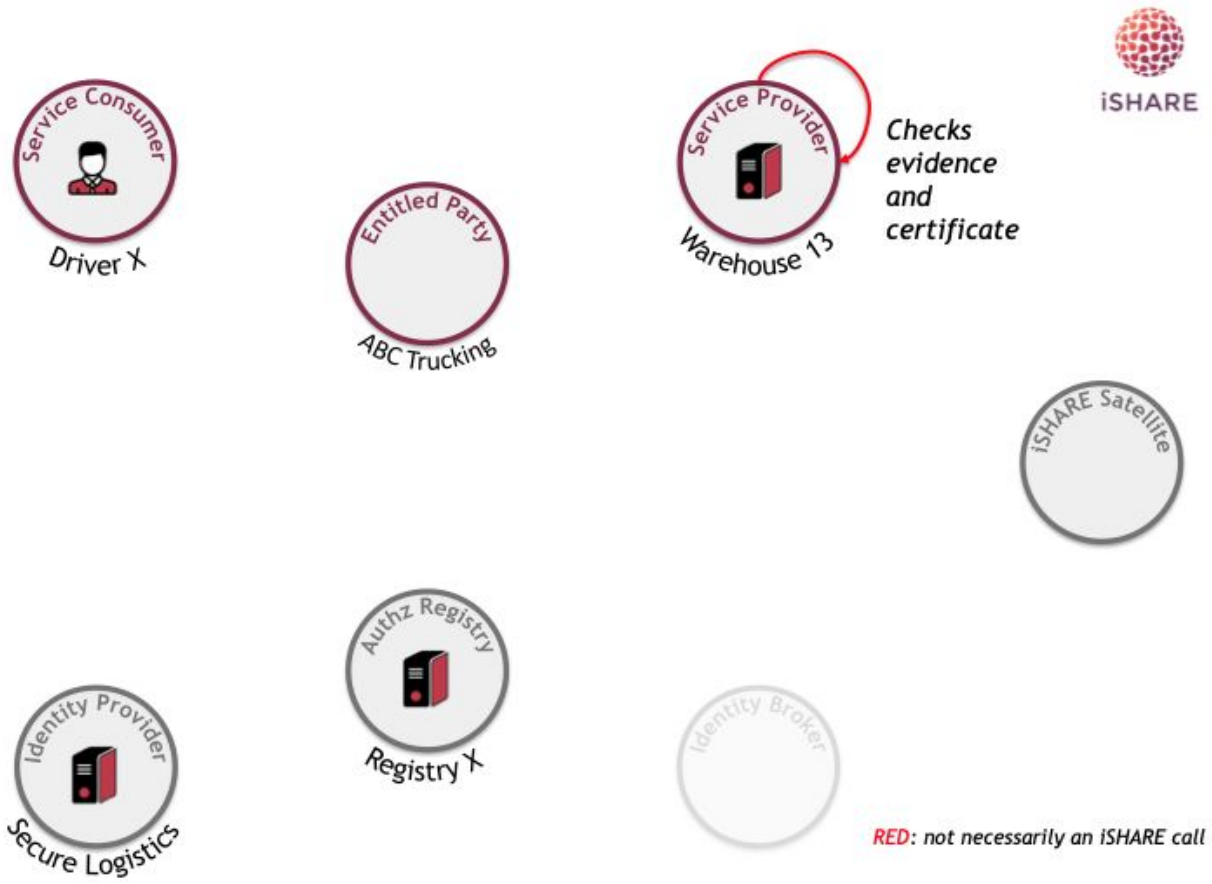
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



i4Trust

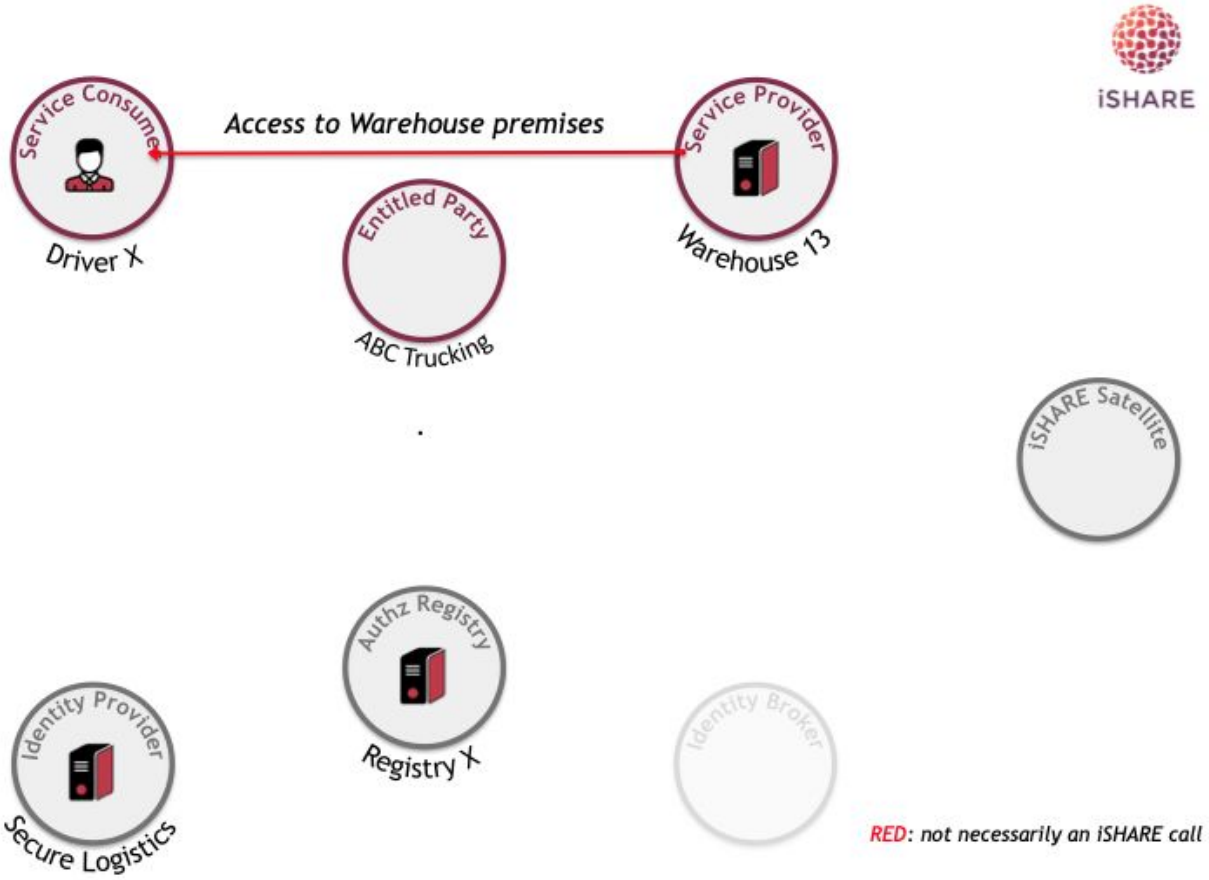
Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



Case 2
H2M

Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



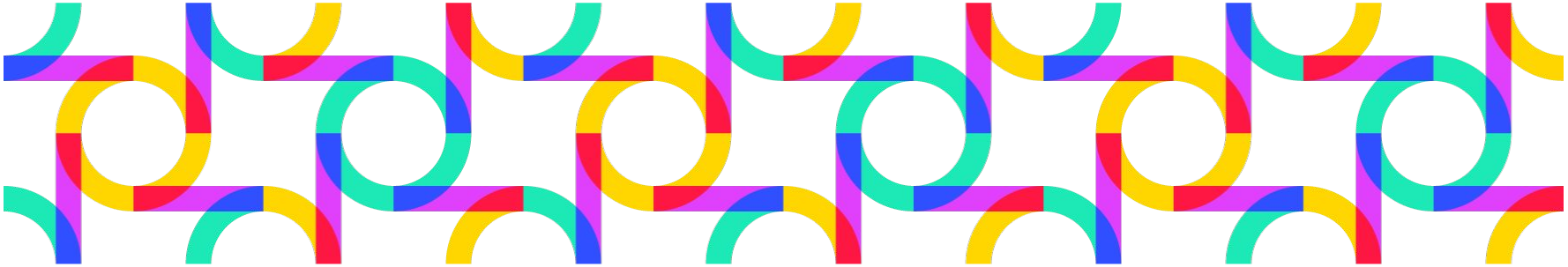
Becoming iSHARE participant involves following steps

1. Identify your role within the iSHARE framework. Note you may play multiple roles from iSHARE framework, in which case you should focus on one role at a time.
    a. Refer to iSHARE Role model here:
2. For the selected role in step 1, refer to the steps given below for fulfilling the requirements for that role. Develop, test and get your implementation certified for compliance with iSHARE specifications
    a. Refer to specifications for each role here:
    b. Explore and play with the postman collection to get better understanding of the roles and specifications
    c. Develop and implement it in your development and test environments.
    d. For testing, you can join iSHARE's test network, where you can test along with other participants or with dummy participants. To test in iSHARE test environment, you can get test eIDAS certificates from https://ca7.isharetest.net:8442/ejbca/ra/
    e. For Autorisation Register testing iSHARE Foundation provides a testing environment or existing providers can be contacted alternatively
    f. For testing with the iSHARE satellite to become the satellite in your data space, reach out to the iSHARE team (info@ishare.eu) to receive the full docker setup of the Satellite component for testing.
3. Once you have done the PoC/Pilot and you are ready to move to production, procure eIDAS* Qualified Seal certificate and doing the joining formalities
    a. Procure eIDAS* Qualified Seal certificate
    b. Sign the iSHARE agreements + dataspace specific agreements if any
    c. Provide chamber of commerce extract, public key of the eIDAS certificate, any other documents as requested by the satellite based on the role
    d. Provide the Conformance Test Tool report

i4Trust

Further reading and useful links

- iSHARE framework covering all aspects - https://scheme.ishareworks.org
- iSHARE developer portal - https://dev.ishareworks.org
  - Postman collections for fictional use case - https://dev.ishareworks.org/demo-and-testing/postman.html#postman-collections
  - Dummy participants in various roles for testing purposes - https://dev.ishareworks.org/demo-and-testing/test-participants.html
- JSON Web Tokens (JWT) reference and libraries - https://jwt.io
- OAuth specifications and other materials - https://oauth.net/2/
- OpenID Connect specifications and other materials - https://openid.net/connect/
- eIDAS regulation
- PKI wikipedia - https://en.wikipedia.org/wiki/Public_key_infrastructure



i4Trust Role Model

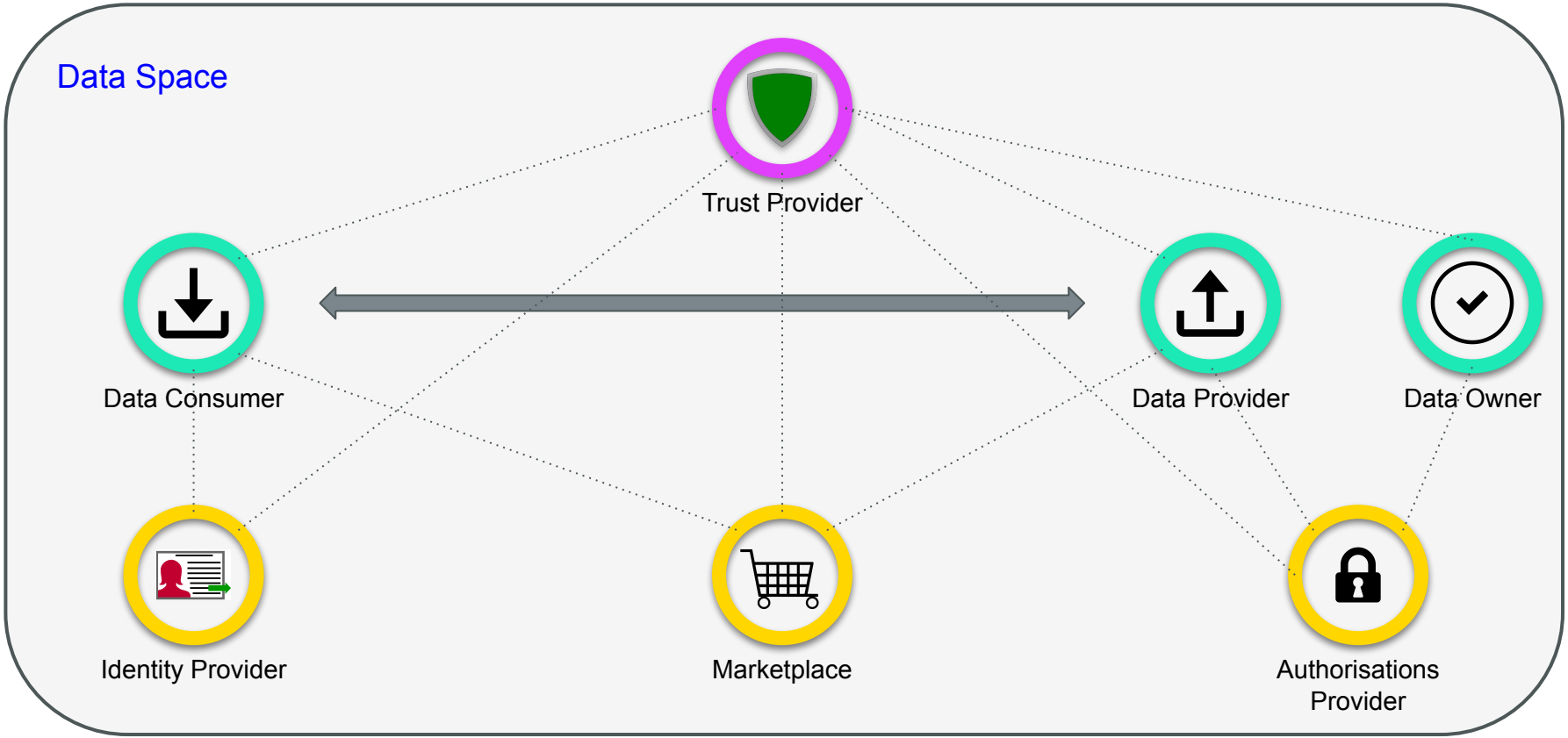
i4Trust

Recap: A data space is an decentralized data ecosystem built by the parties with common purpose

- Further, it can be defined as a decentralized data ecosystem built around commonly agreed building blocks enabling an effective and trusted sharing of data among participants.
- Data consumers, data providers and identity and authorisation service providers form an ecosystem with a governance structure in place to create a basic data space
- Additionally, service providers like Marketplaces, Brokers, Billing and Clearing etc. can be part of data space to support variety of use cases
- In i4Trust iSHARE and FIWARE bring the necessary components along with basic governance structure to create the **i4Trust data space**

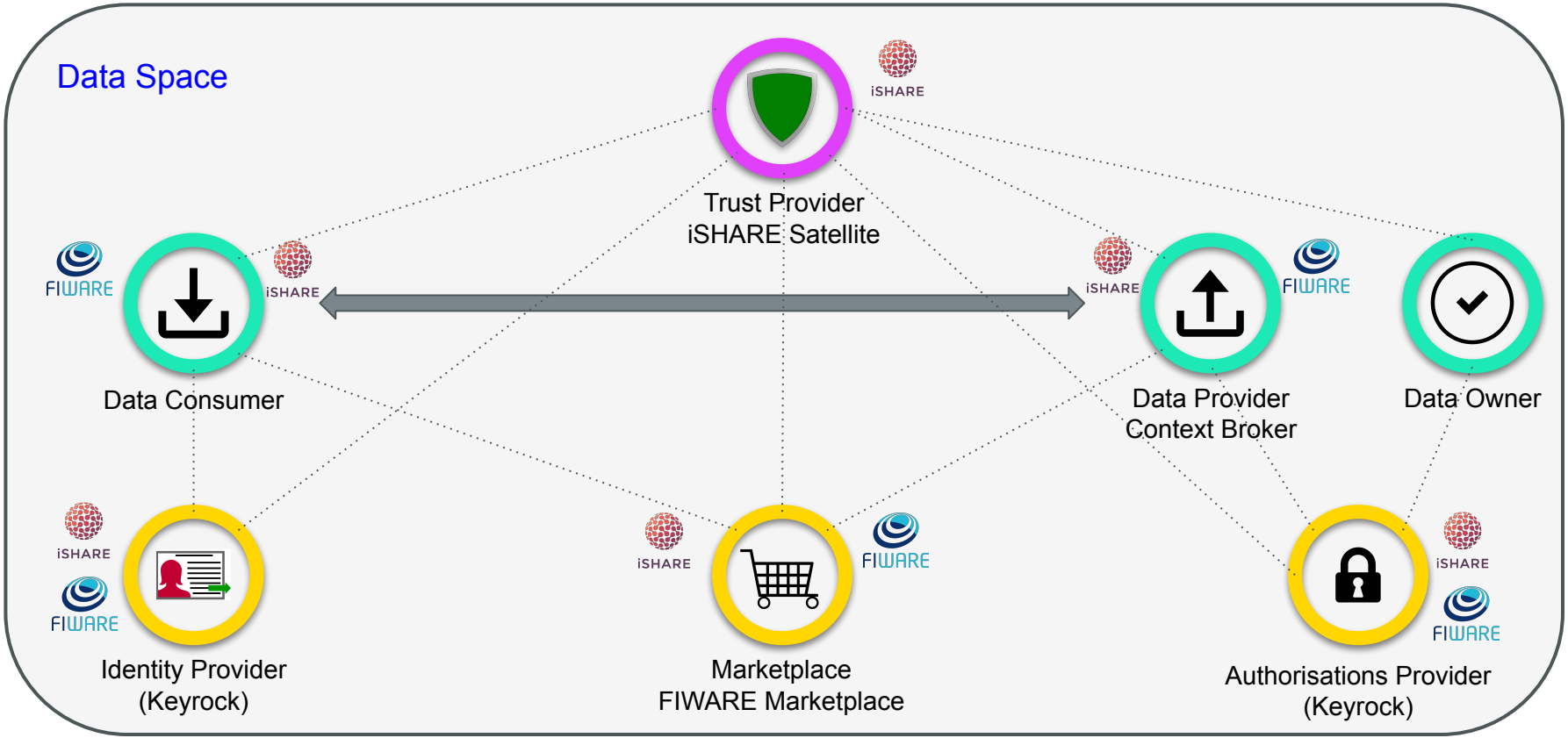
i4Trust

Roles in a typical i4Trust based Data Space



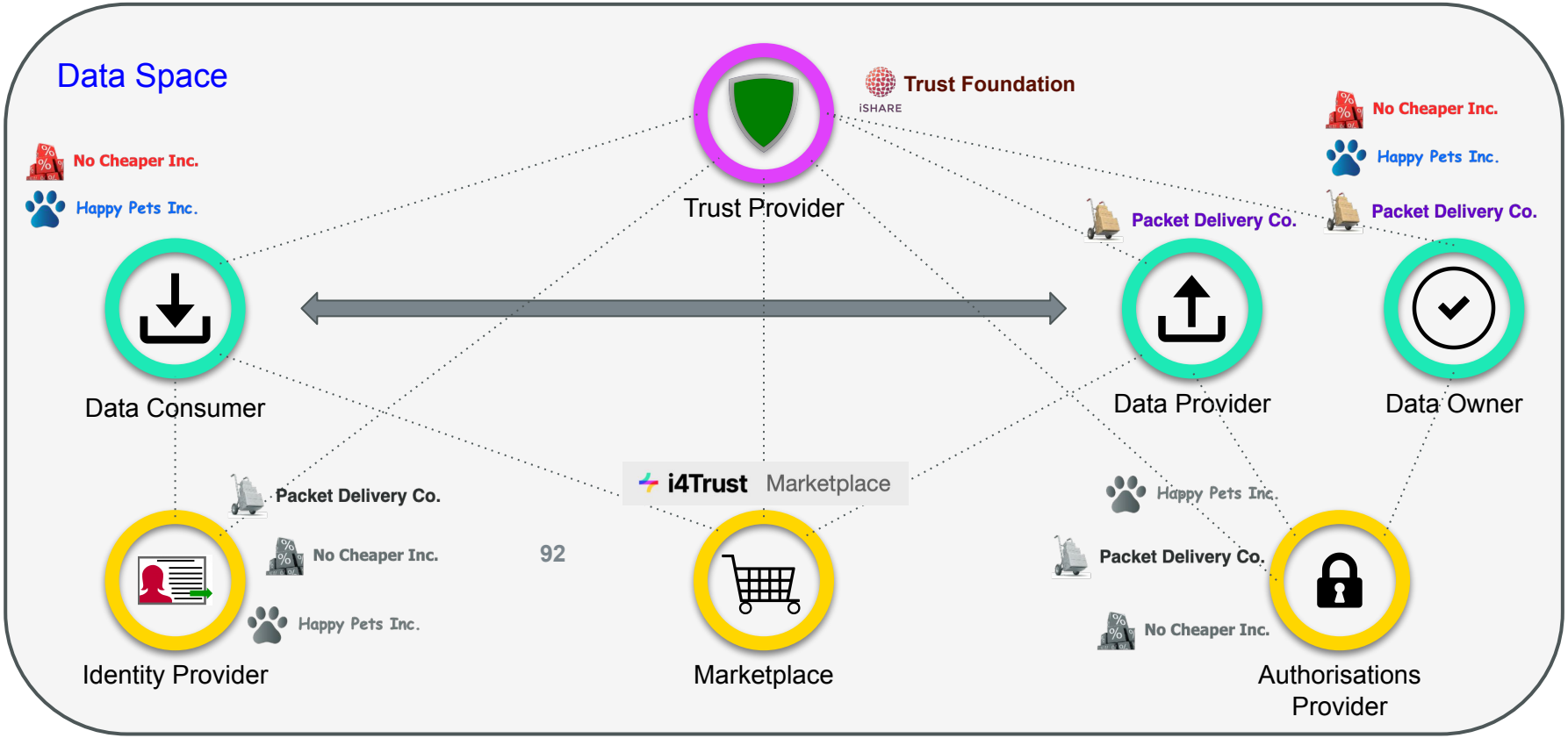
i4Trust

Open source components are available for these roles



i4Trust

Example participants mapped to the role model



i4Trust

Trust Provider essentially facilitates trust in a data space and acting as a guardian

Trust Provider

- Every data space needs a trust provider who acts as register of participants of the data space and keeps their status up to date
- They follow predefined and agreed upon procedures to maintain the list of participants
- Since they play important role in a data space it is paramount that trust provider is a neutral party or a party providing this service within the confines of strict agreements on neutrality
- Every data space should have one trust provider at minimum, however, same trust provider organisation can provide service in multiple data spaces
- In a large data space there can be more than one trust provider
- In i4Trust this role will be fulfilled by iSHARE satellite role

Identity Provider role is designed for users to reuse their existing identity provider at various service/data providers

Identity Provider

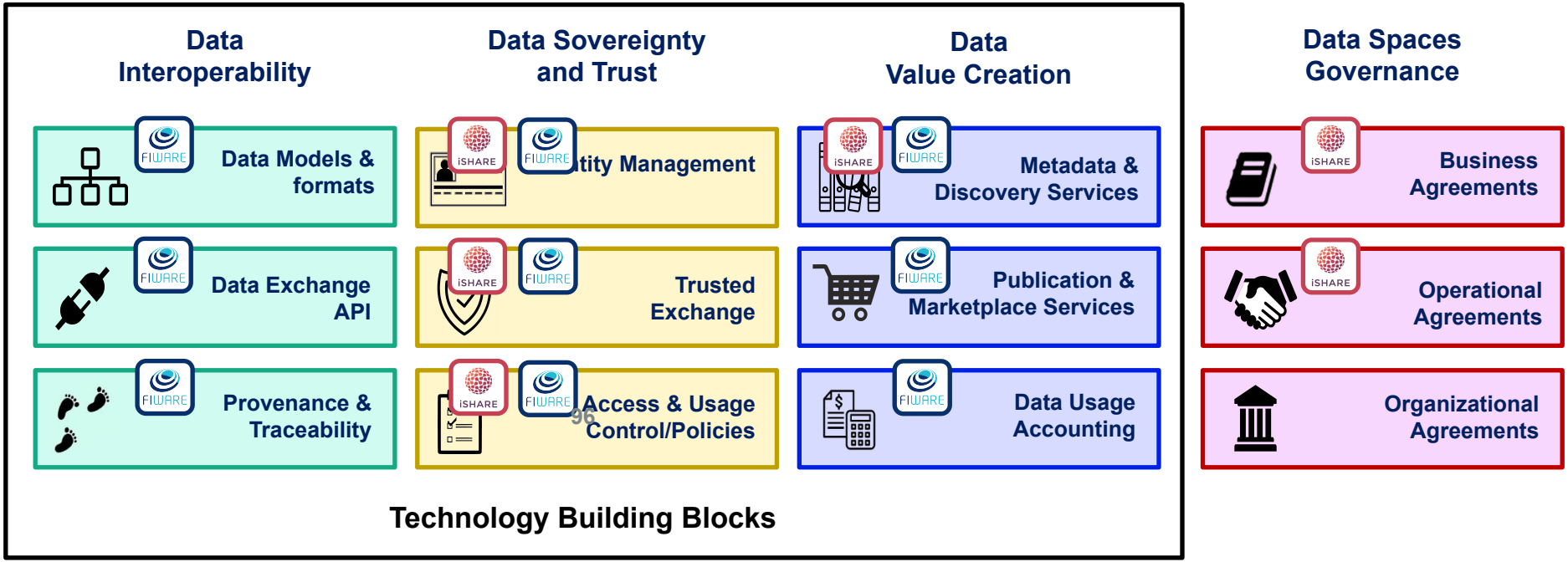
- The Identity Provider role deals with the human identities with varying level of assurances, as defined in eIDAS framework, to support various use cases
- Depending on the criticality of the data, appropriate level of assurance for an identity can be requested
- iSHARE specifications are designed such that service/data provider does not necessarily need to pre-register an identity provider as it can verify if it is an iSHARE certified provider from iSHARE satellite
- Keyrock an Identity Provider software from FIWARE already complies with iSHARE specifications and is available as open source

Authorisations Provider is a role which can be played by any organization as defined in iSHARE

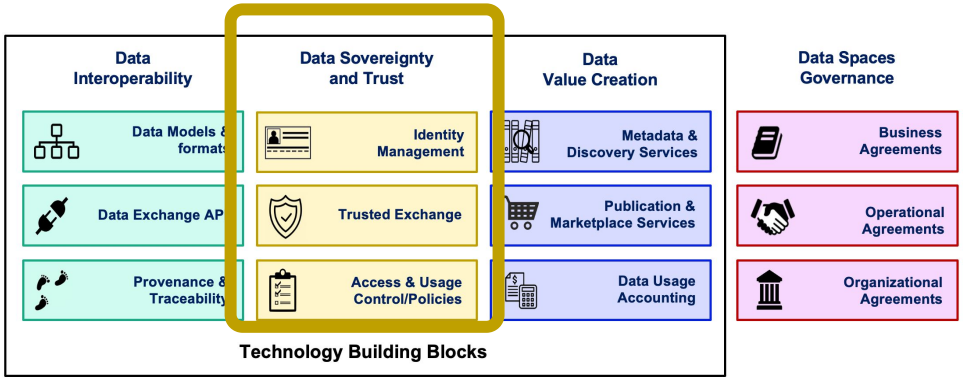
Authorisations Provider

- The role only defines the interface for asking about authorizations and how the response should look like.
- The authorizations can be determined by using a policy database and/or transactional data from backend systems and/or existing authorizations from LDAP/AD or a combination thereof.
- Keyrock powered by FIWARE will provide a policy database to define policies for authorizations which can be combined with other sources to determine authorization during runtime.

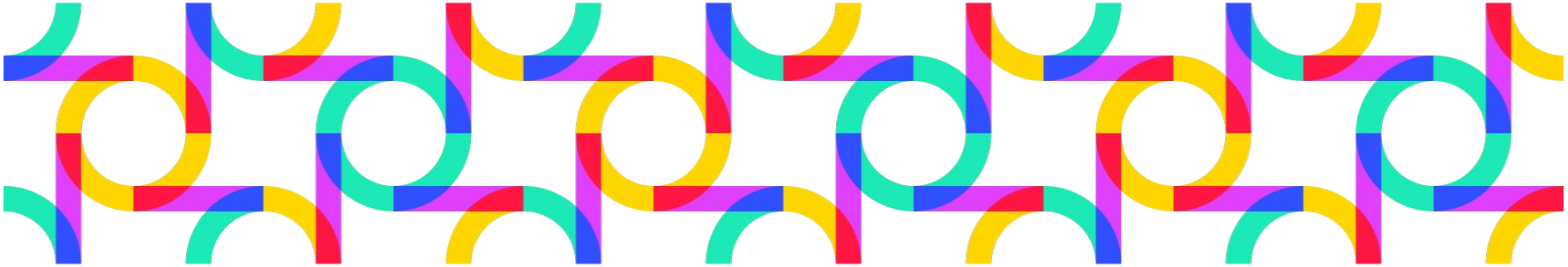
i4Trust: boilerplate for creating dataspaces



Data Sovereignty and Trust



i4Trust



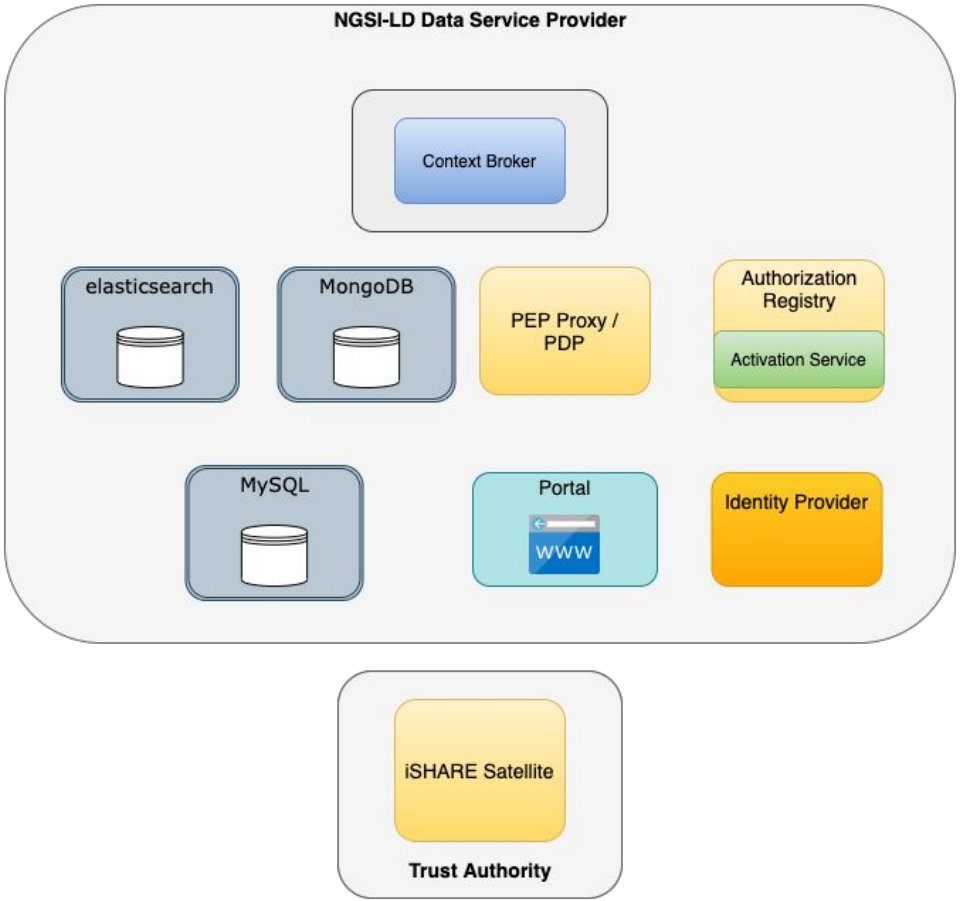
i4Trust IAM components

i4Trust

iSHARE Architecture using FIWARE

- Keyrock is the FIWARE component responsible for identity management.
  - It supports, among others, the OAuth2, Open ID Connect and eIDAS standards
  - It is being extended to support the iSHARE specifications
  - It covers technical implementation of an identity provider
  - In a near future, it will also provide the authorization registry features.
- API Umbrella is an open source API management platform for exposing and securing web service APIs.
  - It acts as a proxy that provides access control, rate limiting, analytics, … to API services. We will use it to protect our context broker instances.
  - It is being extended to comply with iSHARE specifications.
- Until officially released, release candidate docker images should be used. At the moment:
  - fiware/idm:i4trust-rc2 for Keyrock
  - And fiware/api-umbrella:i4trust-rc3 for API Umbrella

iSHARE Architecture using FIWARE



i4Trust

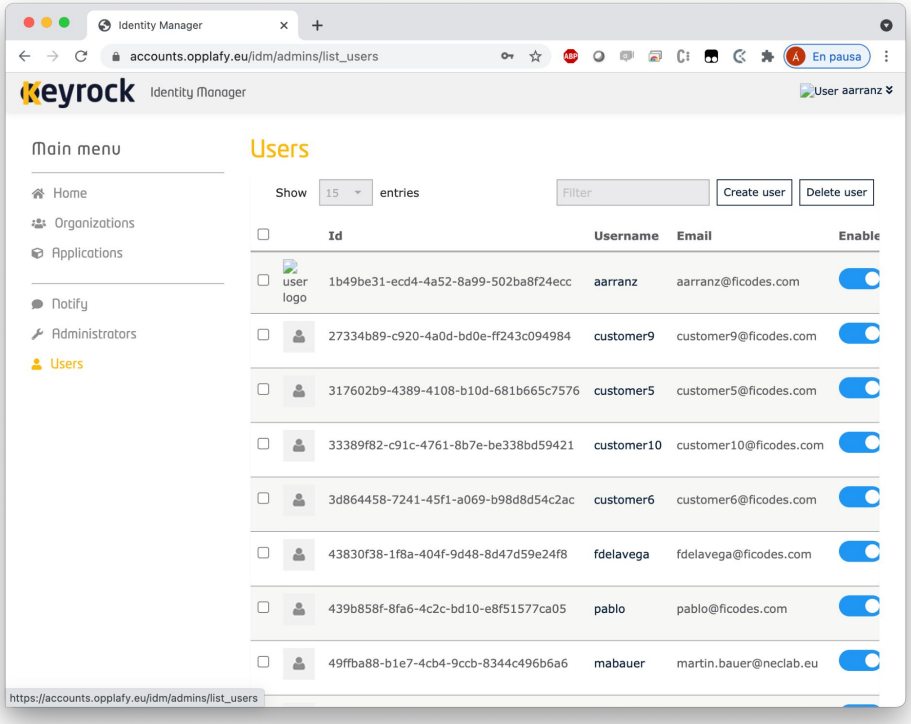
Requirements to deploy

- To be registered as a iSHARE party. This means to have a certificate signed by a trusted CA (e.g. a CA from eIDAS or from a test CA) with the associated private key. Also means to have access credentials to the iSHARE Satellite and Authorization Registry instance to be used.
- Please, contact i4trust.support@ishare.eu for requesting testing credentials and access to those services.

Setting up Keyrock: i4Trust related environment variables

- **IDM_PR_URL:**
  - URL of the iSHARE Satellite instance to use (e.g. https://scheme.isharetest.net)
  - This setting also is the one that enables i4Trust support on Keyrock
- **IDM_PR_CLIENT_ID:**
  - party identifier (EORI) associated with this Keyrock instance
- **IDM_PR_CLIENT_KEY_FILE:**
  - path to the private key file (PEM format)
- **IDM_PR_CLIENT_CRT_FILE:**
  - path to the certificate chain (PEM format)
- **IDM_AR_URL:**
  - URL of Authorization Registry instance to use (e.g. https://ar.isharetest.net)
- **IDM_AR_IDENTIFIER:**
  - Party identifier (EORI) associated with the Authorization Registry instance

Managing Keyrock accounts



i4Trust

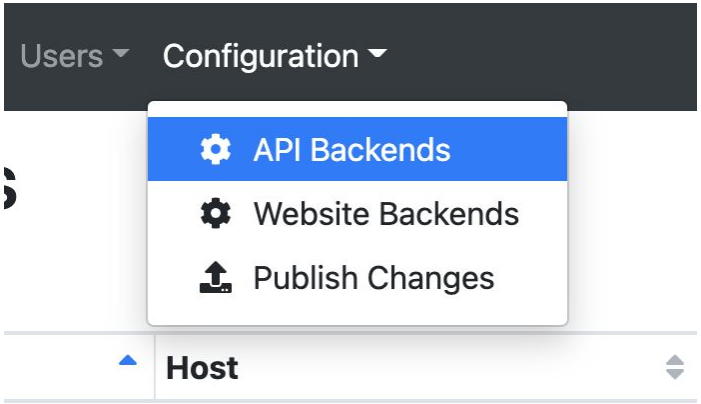
Setting up API umbrella: i4Trust related configuration

In this case, configuration is provided through the api-umbrella.yml file.

```
...

jws:
    identifier: EU.EORI.NLHAPPYPETS
    root_ca: |
        -----BEGIN CERTIFICATE-----
        ...
        -----END CERTIFICATE-----

    private_key: |
        -----BEGIN PRIVATE KEY-----
        MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQC8yeRuxDI2/vyJ
        ...
        -----END PRIVATE KEY-----

    x5c:
      - MIIE...Uw==
      - MIIF...vic=
      - MIIF...AQ==

authorisation_registry:
    host: https://ar.isharetest.net
    identifier: EU.EORI.NL000000004
    token_endpoint: https://ar.isharetest.net/token
    delegation_endpoint: https://ar.isharetest.net/delegation
```

API Management with API Umbrella

- It support two types of entries:
  - API Backends (this is the one we are interested in)
  - And Website Backends
- Changes are not automatically applied. Once you have created/updated/deleted the API Backend configuration, you have to publish them.

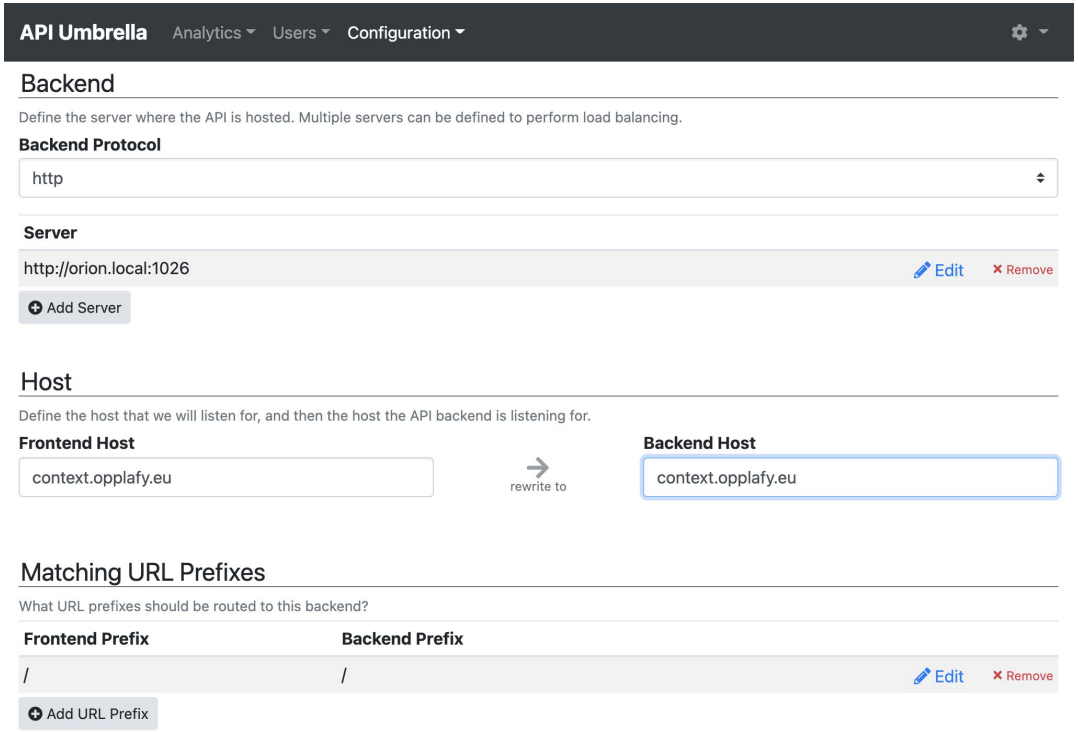


Basic API Backend configuration

You can configure a list of backend services. In this case there is only one backend server (http://orion.local:1026).

“Frontend Host” is the DNS that will be associated with this API backend (Umbrella can be associated with multiple DNS names). The “Backend Host” setting allows you to pass a different “Host” header to the backend server (usually not required).

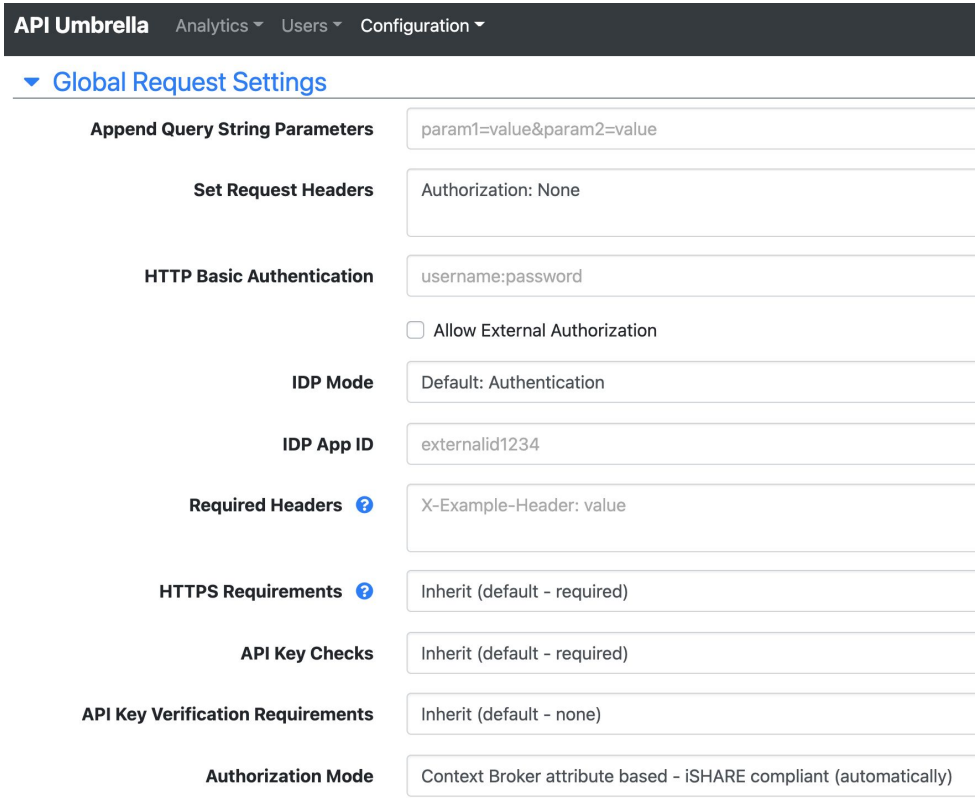
Also, it is possible to transform path prefixes.



i4Trust

Enabling i4Trust support

API Umbrella is able to automatically analyse the incoming NGSI-LD request and to determine the necessary policy required for each request. This is achieved by setting the “Authorization Mode” to “Context Broker attribute based - iSHARE compliant (automatically)” in the Global Request Settings section.



i4Trust

Context Broker policy example

This policy is telling that EU.EORI.NLPACKETDEL is allowing EU.EORI.NLNOCHEAPER to issue GET requests to read DELIVERYORDER entities, including any of their attributes.


```
{
  "delegationEvidence": {
        "notBefore": 1541058939,
        "notOnOrAfter": 2147483647,
        "policyIssuer": "EU.EORI.NLPACKETDEL",
        "target": {
            "accessSubject": "EU.EORI.NLNOCHEAPER"
        },
        "policySets": [
            {
                "maxDelegationDepth": 1,
                "target": {
                    "environment": {
                        "licenses": ["ISHARE.0001"]
                    }
                },
                "policies": [
                    {
                        "target": {
                            "resource": {
                                "type": "DELIVERYORDER",
                                "identifiers": ["*"],
                                "attributes": ["*"]
                            },
                            "actions": ["GET"]
                        },
                        "rules": [
                            {
                                "effect": "Permit"
                            }
                        ]
                    }
                ]
            }
        ]
    }
  }
}
```

Login (step 1)

First, generate an iSHARE JWT signed using the JSON Web Signature standard (JWS)

```
> Headers
{
    "alg": "RS256",
    "typ": "JWT",
    "x5c": [ // Complete certificate chain of the party
       "MIIEhjCC....Zy9w==",
       ...
    ]
}

> Payload
{
      "jti": "99ab5bca41bb45b78d242a46f0157b7d", // Unique JWT ID
      "iss": "EU.EORI.NLMARKETPLA",
      "sub": "EU.EORI.NLMARKETPLA",
      "aud": "EU.EORI.NLHAPPYPETS", // ID (EORI) of the IDP to be accessed
      "iat": "1540827435",
      "nbf": "1540827435",
      "exp": "1540827435",  // 30 seconds after iat
      "response_type": "code",
      "client_id": "EU.EORI.NLMARKETPLA",
      "scope": "openid iSHARE profile email",
      "redirect_uri": "https://www.marketplace.com/openid_connect1.0/return",
      "state": "af0ifjsldkj",
      "nonce": "c428224ca5a",
      "acr_values": "urn:http://eidas.europa.eu/LoA/NotNotified/high",
      "language": "en"
}
```

Login (step 2)

Make a POST request to the /authorize endpoint, so the target Identity Provider (in this case a Keyrock instance) can validate we are a trusted iSHARE party.

```
> Content-Type: application/x-www-form-urlencoded

POST https://idp-pdc.i4trust.fiware.io/authorize

response_type=code&
client_id=EU.EORI.NLMARKETPLA&
scope=iSHARE openid&
request=eyJ0eXA...YkNKOQ
```

The JWT generated in step 1 has to be provided in the request parameter.

Keyrock with validate the JWT and will contact the iSHARE Satellite to check that the client is a valid iSHARE participant. If everything goes well, a 204 status code will be returned along with a Location header.

Login (step 3)

Redirect user's browser to the URL provided on the Location headers. This way, the user can provide his credentials and consent through the login page of Keyrock.

If everything goes well, Keyrock will return an authorization code using the provided redirect URI.

```
< Location: https://marketplace.i4trust.fiware.io/openid_connect1.0/return?
    code=Dmn-TbSj7OcKl5ym1j5xZsgkabzVP8dMugC81nzmeW4&
    state=ZqVQm4zHaEDyBhzpm1ZRH7fsxy703lq2
```

Login (step 4)

A new request to the /token endpoint of the Identity Provider has to be made to retrieve the final access token.

The signed JWT created in step 1 has to be provided in the client_assertion parameter of the request.

The authorization code obtained in step 3 is provided within the code parameter.

```
> Content-Type: application/x-www-form-urlencoded

POST https://idp-pdc.i4trust.fiware.io/token

grant_type=authorization_code&
client_id=EU.EORI.NLMARKETPLA&
client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer&
client_assertion=eyJ0eXA...YkNKOQ&
redirect_uri=https://marketplace.i4trust.fiware.io/openid_connect1.0/return&
code=Dmn-TbSj7OcKl5ym1j5xZsgkabzVP8dMugC81nzmeW4
```

i4Trust

Login (step 5)

Finally, Keyrock will provide an access token that can be used to access services.

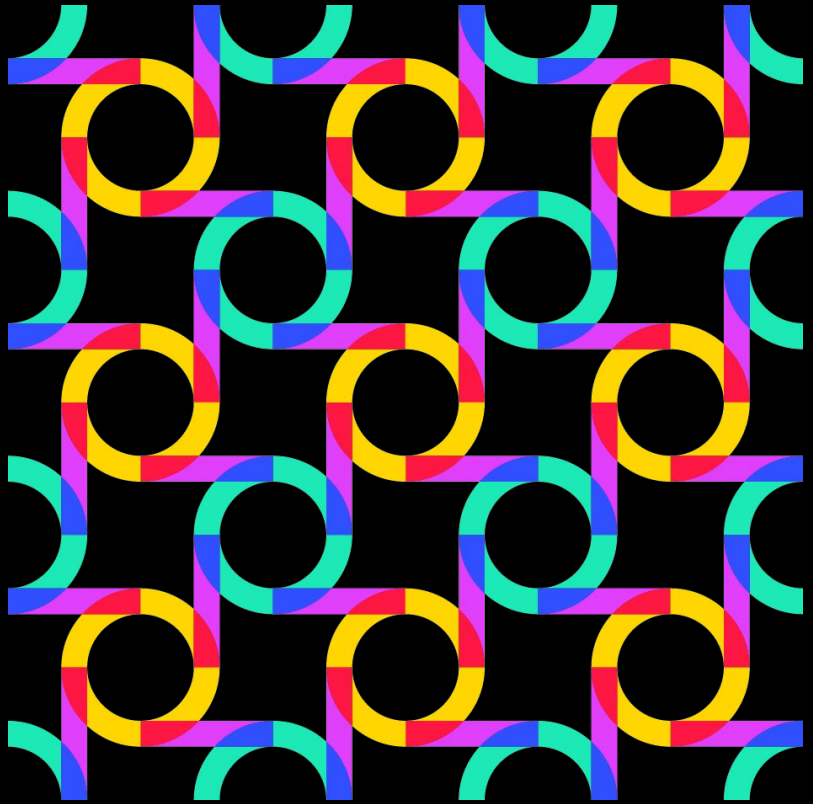
In addition, an iSHARE JWT compliant id_token is also returned which is associated with the authenticated session.

```
< Content-Type: application/json
< Cache-Control: no-store
< Pragma: no-cache

{
  "id_token": "eyJhb...V2jA",
  "access_token": "aW2ys...LIOw",
  "expires_in": 3600,
  "token_type": "Bearer"
}

Decoded id_token parameter
{
  "iss": "EU.EORI.NLPACKETDEL",
  "sub": "419404e1-07ce-4d80-9e8a-eca94vde0003de",
  "aud": "EU.EORI.NLMARKETPLA",
  "jti": "378a47c4-2822-4ca5-a49a-7e5a1cc7ea59",
  "iat": 1504683445,
  "exp": 1504683475,
  "auth_time": 1504683435,
  "nonce": "c428224ca5a",
  "acr": "urn:http://eidas.europa.eu/LoA/NotNotified/low",
  "azp": "EU.EORI.NLMARKETPLA",
}
```

Further reading and useful links

- Official Keyrock Documentation - https://fiware-idm.readthedocs.io/
- Tutorials and descriptions on how to setup the components within an i4Trust data space
  - https://github.com/i4Trust/tutorials/tree/main/Data-Service-Provider
- Example Keyrock Theme - https://github.com/i4Trust/keyrock-theme-pdc
- Helm Charts (kubernetes) to deploy i4Trust IAM components:
  - https://github.com/FIWARE/helm-charts/tree/i4trust
- Passport strategy for authentication with FIWARE Keyrock using iShare JWT:
  - https://github.com/Ficodes/passport-i4trust
- iSHARE framework covering all aspects - https://scheme.ishareworks.org
- iSHARE developer portal - https://dev.ishareworks.org

Thank you!



i4Trust

i4Trust has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement no 951975.