

iSHARE - Deepdive & i4Trust Data Sovereignty & Trust aspects

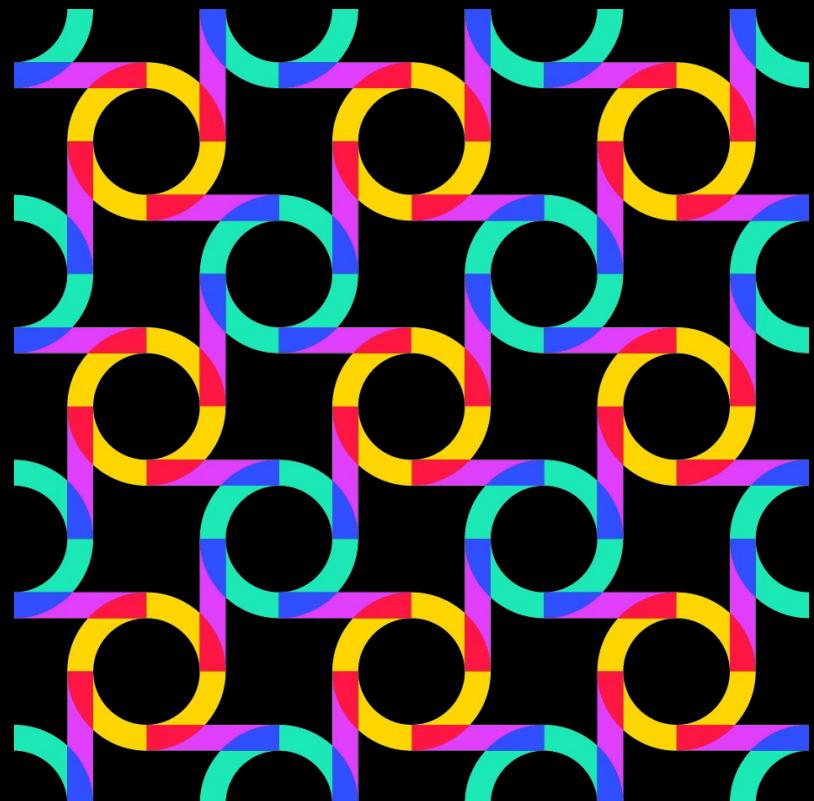
Speakers:

Rajiv Rajani - CTO iSHARE Foundation



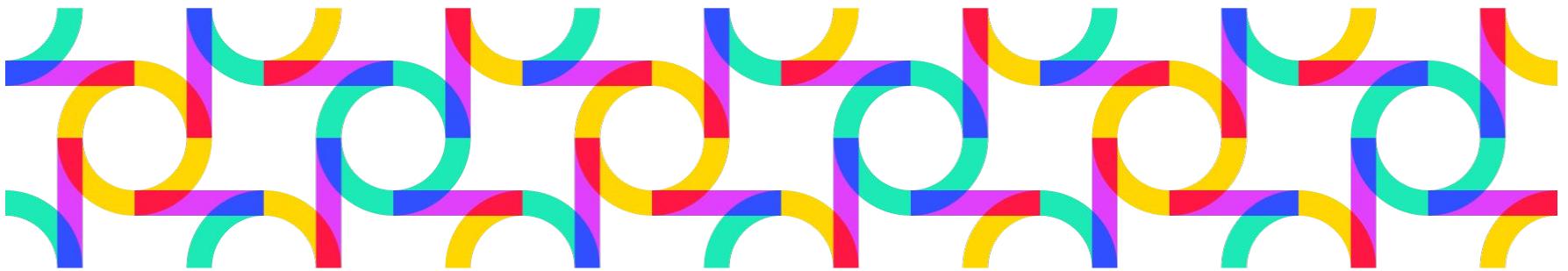
[i4Trust Website](#)

[i4Trust Community](#)



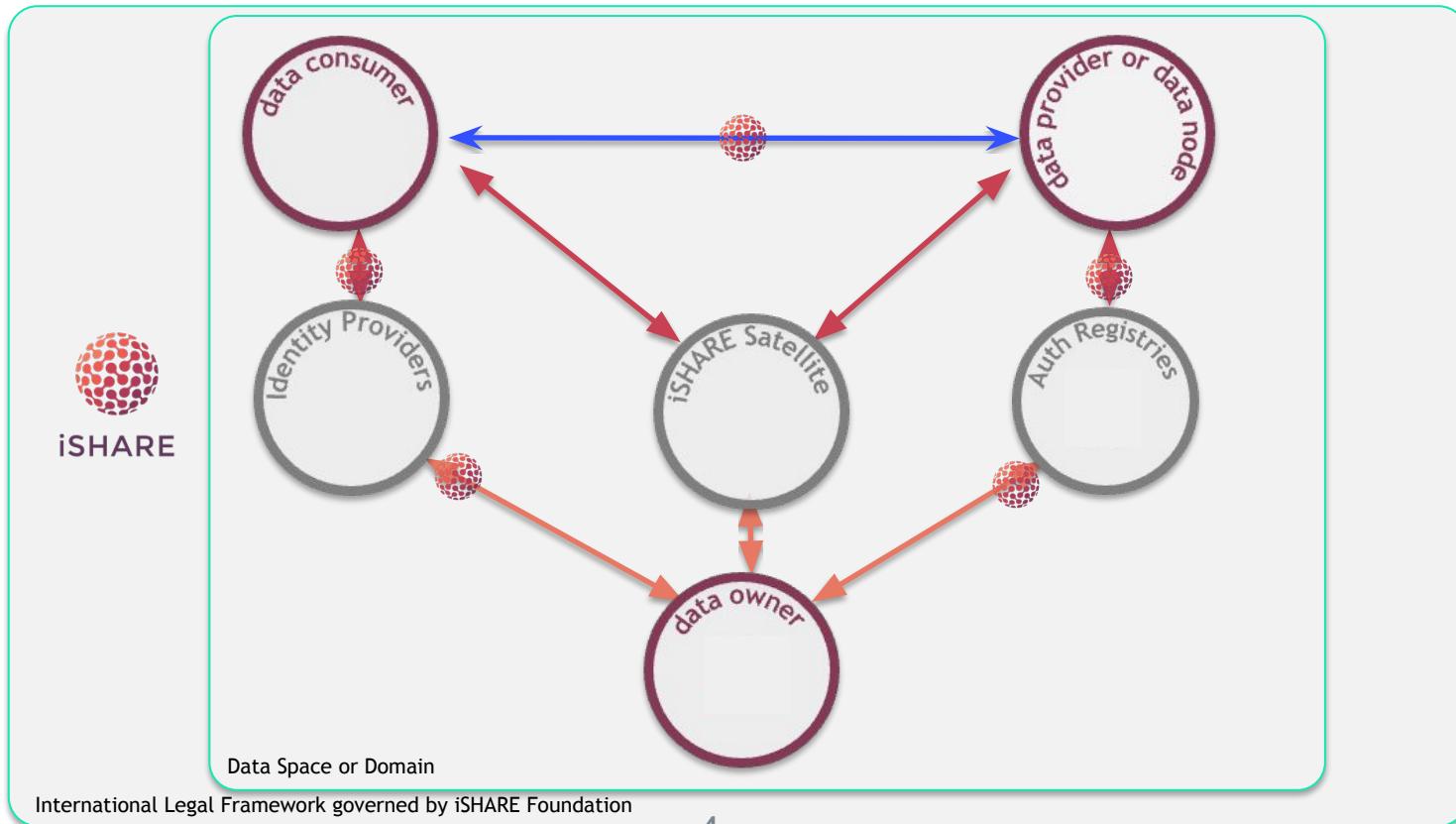
i4Trust Identity and Access Management (IAM)

- About:
 - This part of the session will introduce you to i4Trust IAM components in detail.
- This session will:
 - Explain iSHARE Identity and Access Management concepts in detail
 - Explain the FIWARE components which comply with iSHARE specifications in detail
 - How to deploy them
 - How to use them
- Goals:
 - After this session you will be able to implement a service using i4Trust IAM components
- Target Audience:
 - LEBDs



Technical Deepdive - iSHARE

Recap: iSHARE role model



What is iSHARE Satellite?

- iSHARE satellite is a federated certified role which needs to be played by an organisation within the dataspace
- It essentially does:
 - Onboard and maintain list of participants and its statuses
 - 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

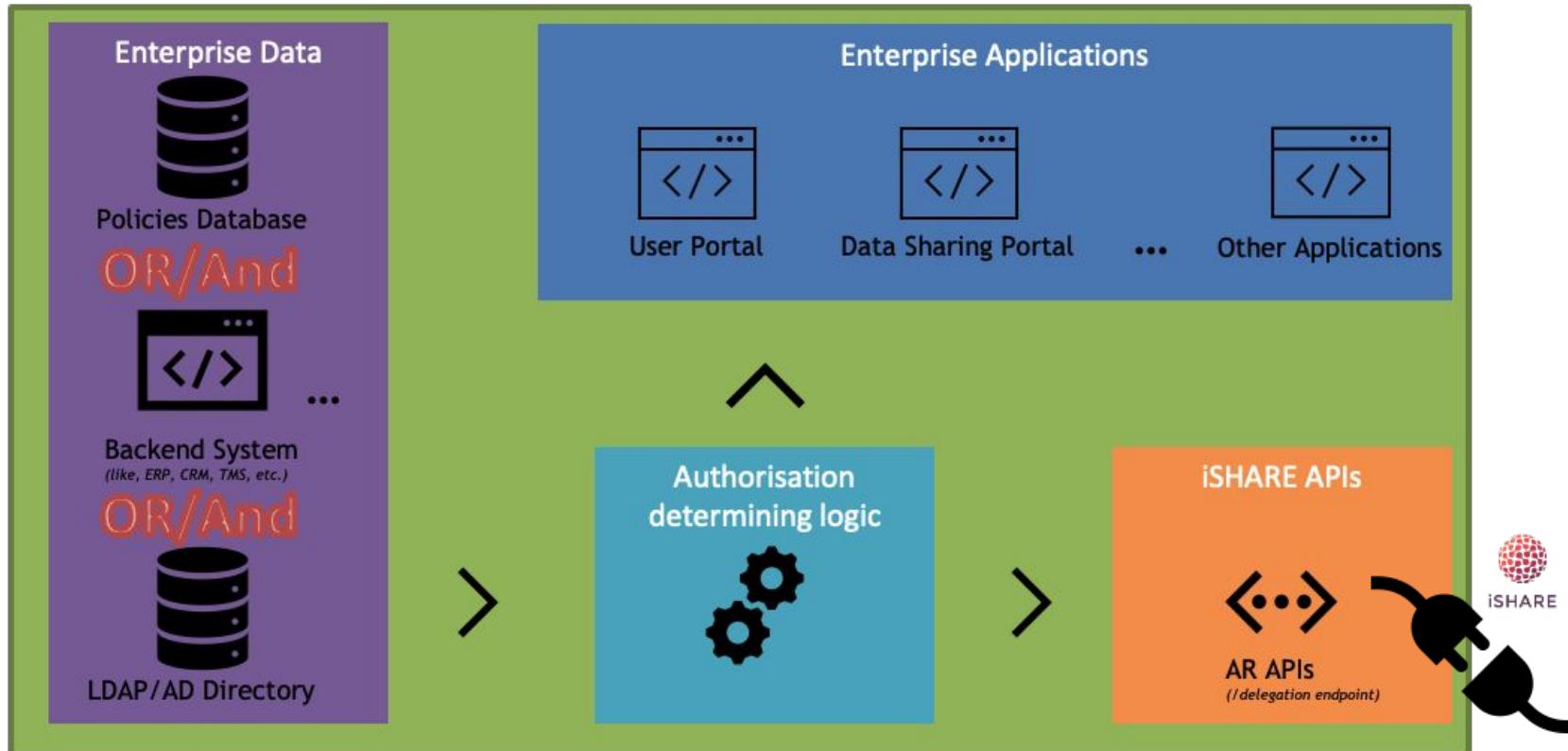
Identity Provider role is designed for users to reuse their existing identity provider at various service/data providers

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



Example response of a fine-grained authorisation

```
3 {
4   "delegationEvidence": {
5     "notBefore": "1509633681",
6     "notOnOrAfter": "1509633781",
7     "policyIssuer": "EU.EORI.NL123456789",
8     "target": {
9       "accessSubject": "EU.EORI.NL987654321"
10    },
11   "policySets": [
12     {
13       "maxDelegationDepth": "5",
14       "target": {
15         "environment": {
16           "licenses": ["ISHARE.0001"]
17         }
18       },
19       "policies": [
20         {
21           "target": {
22             "resource": {
23               "type": "GS1.CONTAINER",
24               "identifiers": ["*"],
25               "attributes": ["ATTRIBUTE.ETA", "ATTRIBUTE.WEIGHT"]
26             },
27             "actions": ["ISHARE.READ", "ISHARE.WRITE"],
28             "environment": {
29               "serviceProviders": ["EU.EORI.NL678912345"]
30             }
31           },
32           "rules": [
33             {
34               "effect": "Permit"
35             },
36             {
37               "effect": "Deny",
38               "target": {
39                 "resource": {
40                   "type": "GS1.CONTAINER",
41                   "identifiers": ["GS1.SSCC18.725391630826493716"],
42                   "attributes": ["ATTRIBUTE.WEIGHT"]
43                 },
44                 "actions": ["ISHARE.READ"]
45               }
46             }
47           ]
48         }
49       ]
50     }
51   ]
52 }
53 }
```

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

Example of a coarse-grained authorisation

```
3 % {  
4   "delegationEvidence": {  
5     "notBefore": "1509633681",  
6     "notOnOrAfter": "1509633781",  
7     "policyIssuer": "EU.EORI.NL123456789",  
8     "target": {  
9       "accessSubject": "EU.EORI.NL987654321"  
10    },  
11    "policySets": [  
12      {  
13        "maxDelegationDepth": "*",  
14        "target": {  
15          "environment": {  
16            "licenses": ["*"]  
17          }  
18        },  
19        "policies": [  
20          {  
21            "target": {  
22              "resource": {  
23                "type": "GS1.CONTAINER",  
24                "identifiers": ["*"],  
25                "attributes": ["*"]  
26              },  
27              "actions": ["*"]  
28            },  
29            "rules": [  
30              {  
31                "effect": "Permit"  
32              }  
33            ]  
34          }  
35        ]  
36      }  
37    ]  
38  }  
39 }
```

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

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

1

Certificates

- What is a certificate?
- What is .p12 and x5c?
- How does iSHARE use certificates?

JWTs

2

- What is a JWT?
- How do I create a JWT?
- How do I sign it with a certificate?

3

Client assertion

- What is a client_assertion?
- How does it use JWTs?
- How do I create a valid client_assertion?

Access token

4

- What is an access token?
- How does it use a client_assertion
- How do I use access tokens?



A certificate is a digital file used for integrity and authenticity

certificate
(public key infrastructure)

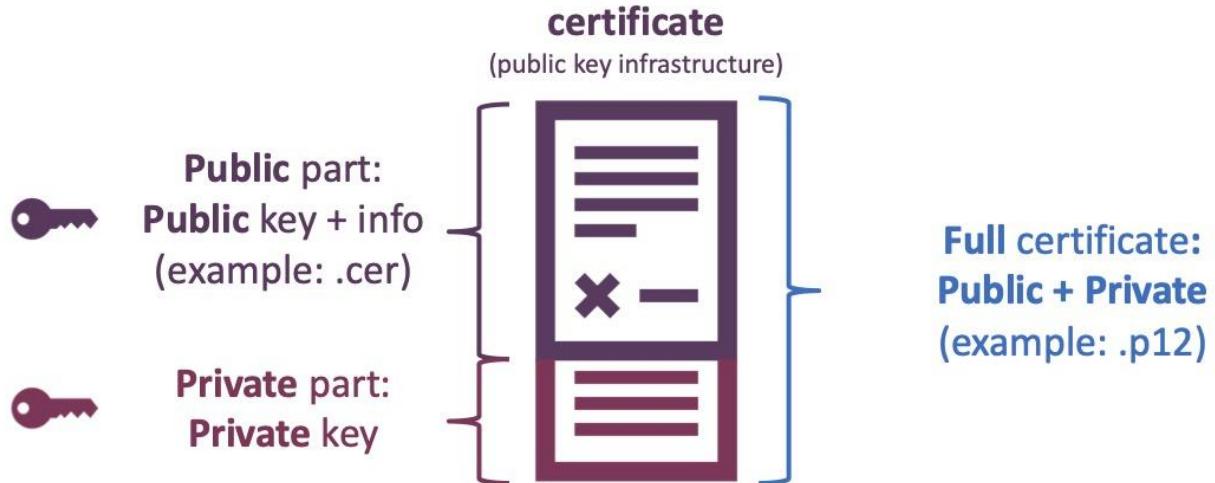


- Digital file that proves ownership of public key
- Certificate provides information on the owner
- Certificate is issued by a Certificate Authority
- Keys used to sign or encrypt other files

More info:
[Cheat Sheet](#)
or [Dev. Portal](#)



A certificate consists of a public and a private part



More info:
[Cheat Sheet](#)
or [Dev. Portal](#)

1 Certificates	2 JWTs
3 Client assertion	4 Access token

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. PKoverheid or eIDAS)

In iSHARE

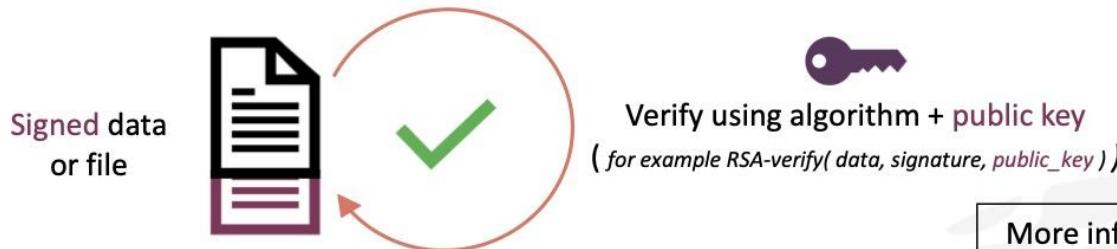
- **x5c** - value
 - String of letters representing public part of certificate
 - Used to *verify* signing and identity
- **.p12** file containing both private and public part
 - Used to sign (*integrity* and *authenticity*)
 - Password protected

More info:
[Cheat Sheet](#)
or [Dev. Portal](#)

1 Certificates	2 JWTs
3 Client assertion	4 Access token

A private key signs data

A public key verifies this signature



More info:
[Cheat Sheet](#)
or [Dev. Portal](#)

1 Certificates	2 JWTs
3 Client assertion	4 Access token

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)		

7

More info:
<https://jwt.io>
or [Dev. Portal](#)



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" }	eyJhbGciOiJSUzI1Ni IsInR5cCI6IkpXVCJ9
2. Payload (JSON)	{ "name": "iSHARE", "iat": 1516239022 }	eyJuYW1lIjoiaVNIQV JFIiwiaWF0IjoxNTE2 MjM5MDIyfQ
3. Signature (RSASHA256)	RSASHA256(base64UrlEncode(header) + ". " + base64UrlEncode(payload), private key 	NgQ-hpSnPKZ7ZuobA3rkqii4b0qpL2X9UXGe_ bapAUm29v_wqhFBWBqf-BkjRE0c_xiIXC30AA1 EC6pJJCMxQap2GL6NuyT4x7daPzTgCDXoFMD9v AEL5aSC_vhsP1UpnvNzrepT2YNeSv1RbNBSJNg UTmd-rMDcvwGdmALRON60uZSTPuHZY02Z-yMzLz OgwjdN-9DhB_FnJALKNHLyb9meJ0d3GhaasbVQ jZyiql0rKLlv3ItJg3dxnca-USjycVzq020xTM UNUjaxAs9UNCuEYFIE01nrXibNpU5gN-ZBw3M7V xP9ZJhS02doR0bQ1KpiWRacB50xu8p1qWtGcg

8

More info:
<https://jwt.io>
or [Dev. Portal](#)



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	TOTAL JWT
1. Header (JSON)	{ "alg": "RS256", "typ": "JWT" }	eyJhbGciOiJSUzI1Ni IsInR5cCI6IkpXVCJ9	eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9. eyJJuYW1lIjoiaVNIQVJFIiwiaWF0IjoxNTE2MjM5MDIyfQ.
2. Payload (JSON)	{ "name": "iSHARE", "iat": 1516239022 }	eyJuYW1lIjoiaVNIQVJFIiwiaWF0IjoxNTE2MjM5MDIyfQ	eyJuYW1lIjoiaVNIQVJFIiwiaWF0IjoxNTE2MjM5MDIyfQ.
3. Signature (RSASHA256)	RSASHA256(base64UrlEncode(header) + ". " + base64UrlEncode(payload), private key 	NgQ-hpSnPKZ7ZuobA3rkqii4b0qpL2X9UXGGe_bapAUm29V_wqhfBWBqf-BkjRE0c_xiiXC30AA1EC6pJICmXqa2GL6NuyT4x7-daPzTgCDXoFMD9vAEI5aSC_vhsP1UnpvNzrepT2YNaSv1RbNBSJNgUTmd-rMDCwvGdmAlRON60uZSTPuHZY02Z-yMZLzOgwjdN-9DhB_FnJALKNHLYb9meJ0d3GhaasbVQjZyiql0rKLlv3ItJg3dXnca-U5JycVzQz02xTMUNUaxAs9UNCuEYFIE01NrXIBNpU5gN-ZBw3M7VxP9ZJhS02doR0bQ1KpiWRaCB50xu8p1qWtGcg	NgQ-hpSnPKZ7ZuobA3rkqii4b0qpL2X9UXGGe_bapAUm29V_wqhfBWBqf-BkjRE0c_xiiXC30AA1EC6pJICmXqa2GL6NuyT4x7-daPzTgCDXoFMD9vAEI5aSC_vhsP1UnpvNzrepT2YNaSv1RbNBSJNgUTmd-rMDCwvGdmAlRON60uZSTPuHZY02Z-yMZLzOgwjdN-9DhB_FnJALKNHLYb9meJ0d3GhaasbVQjZyiql0rKLlv3ItJg3dXnca-U5JycVzQz02xTMUNUaxAs9UNCuEYFIE01NrXIBNpU5gN-ZBw3M7VxP9ZJhS02doR0bQ1KpiWRaCB50xu8p1qWtGcg

9

More info:
<https://jwt.io>
or [Dev. Portal](#)



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 + ":" + **PAYOUT**

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

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

```
{  
  "iss": "EU.EORI.NL0000000001",  
  "sub": "EU.EORI.NL0000000001",  
  "jti": "df7ffc54cf148d15b0",  
  "iat": 1556210430,  
  "nbf": 1556210430,  
  "exp": 1556210460,  
  "aud": "EU.EORI.NL0000000003"  
}
```

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

+

"

PAYOUT

Sign using private key
with RSASHA256

CLIENT ASSERTION

```
{  
  "alg": "RS256",  
  "typ": "JWT",  
  "x5c": ["MIIEgTCCAmg.....  
           (long string)  
           ....reK18+JkAjAJU-"]  
}
```

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

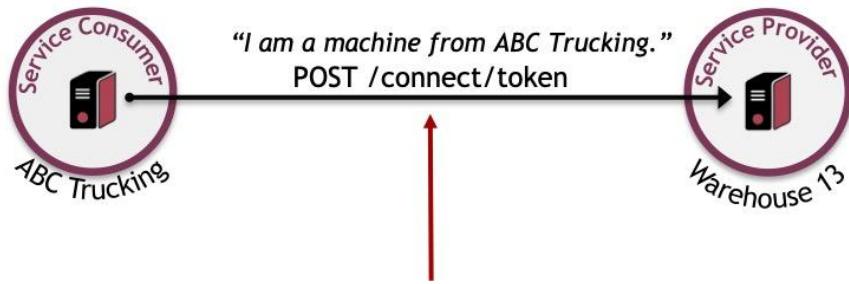
eyJhbGciOiJSUzI1NlslnR5c...
(very long string)
.....D2vek8KTfhmzUw2i9UG
D5SsE0yOluDzUGU-Q

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



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.

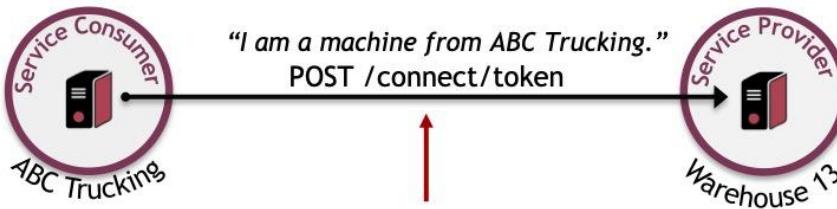


It is here that we use the
`client_assertion` (JWT)

1 Certificates	2 JWTs
3 Client assertion	4 Access token

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.



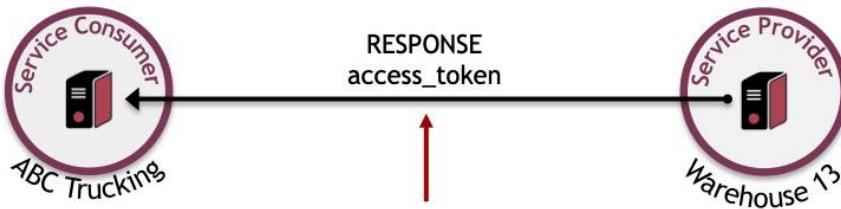
Values in the body this request

"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)D2vek8KTf5SsE0y0luDzUGU-Q"	The JWT we created earlier

1 Certificates	2 JWTs
3 Client assertion	4 Access token

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)

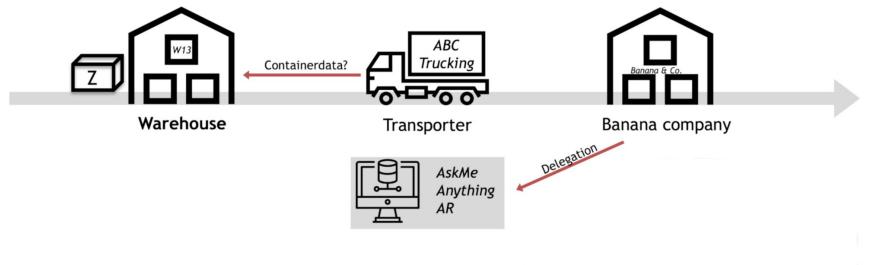
```
{  
    "access_token": "eyJhbGciOi...long string...ahvEQ5w", ← The access token, for your further use  
    "expires_in": 3600, ← The expiry time, in seconds  
    "token_type": "Bearer" ← The type of token (should be "bearer")  
}
```

1 Certificates	2 JWTs
3 Client assertion	4 Access token

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.

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

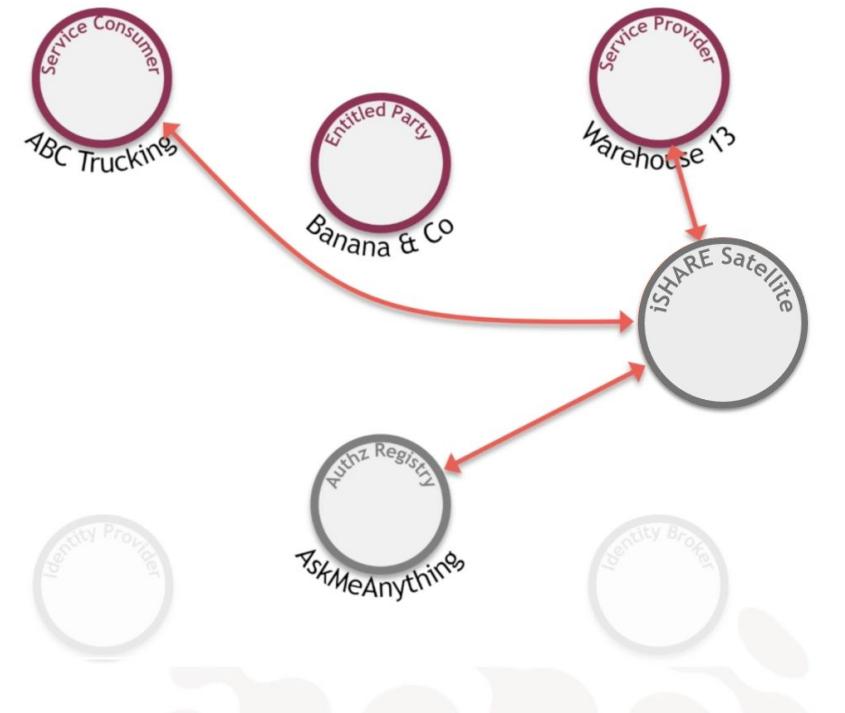


Beforehand: Getting the trusted list

- 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



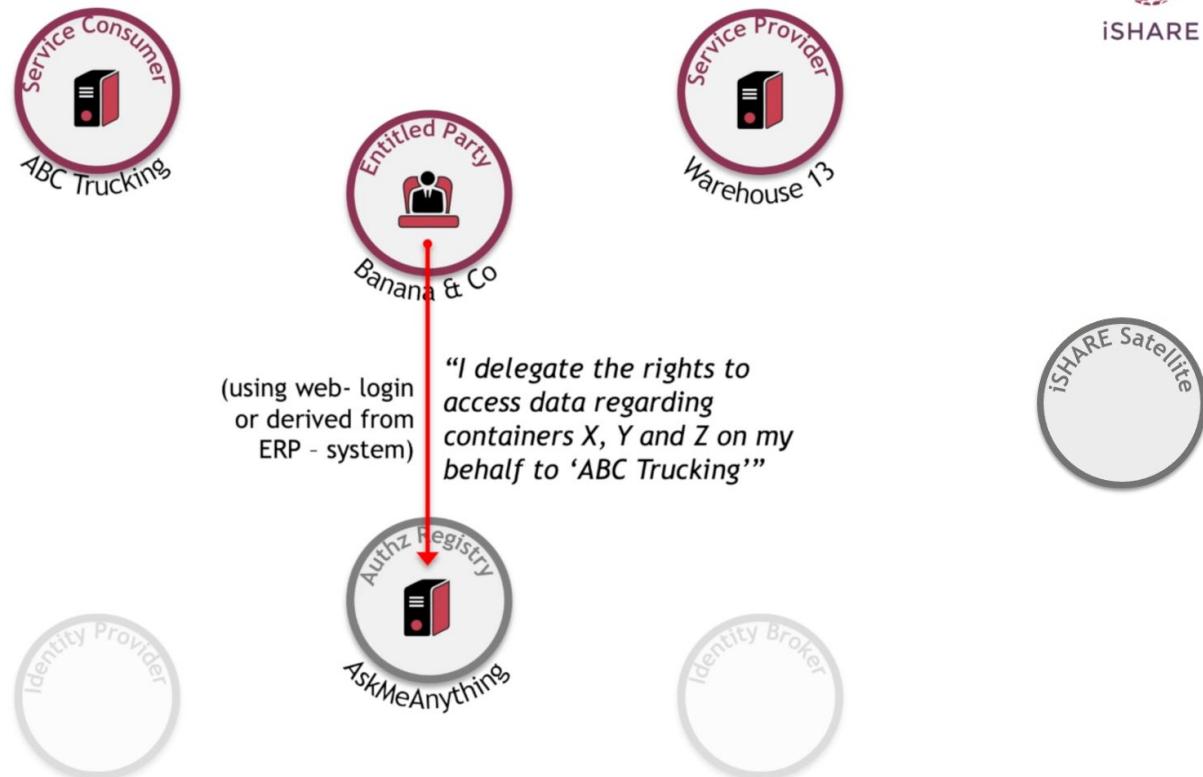
Case 1a

M2M

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

Delegate rights

1. Update AR
2. Update SC



RED: not necessarily an iSHARE call



iSHARE

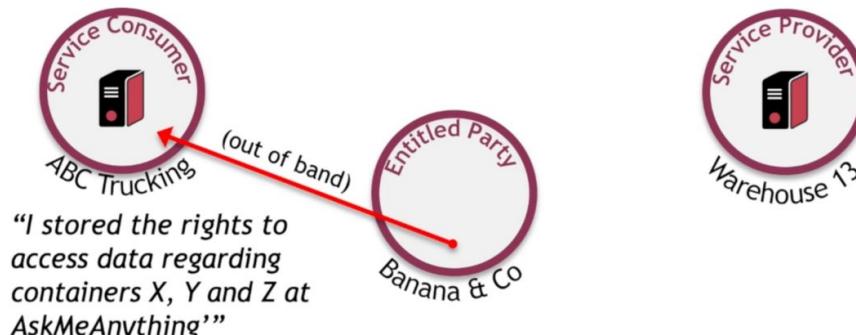
Case 1a

M2M

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

Delegate rights

1. Update AR
2. Update SC





iSHARE

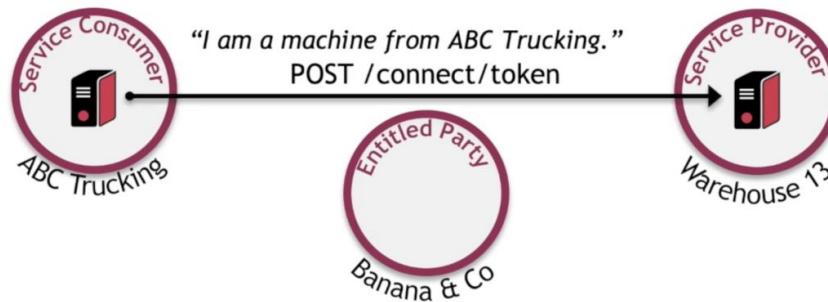
Case 1a

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



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



RED: not necessarily an iSHARE call



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

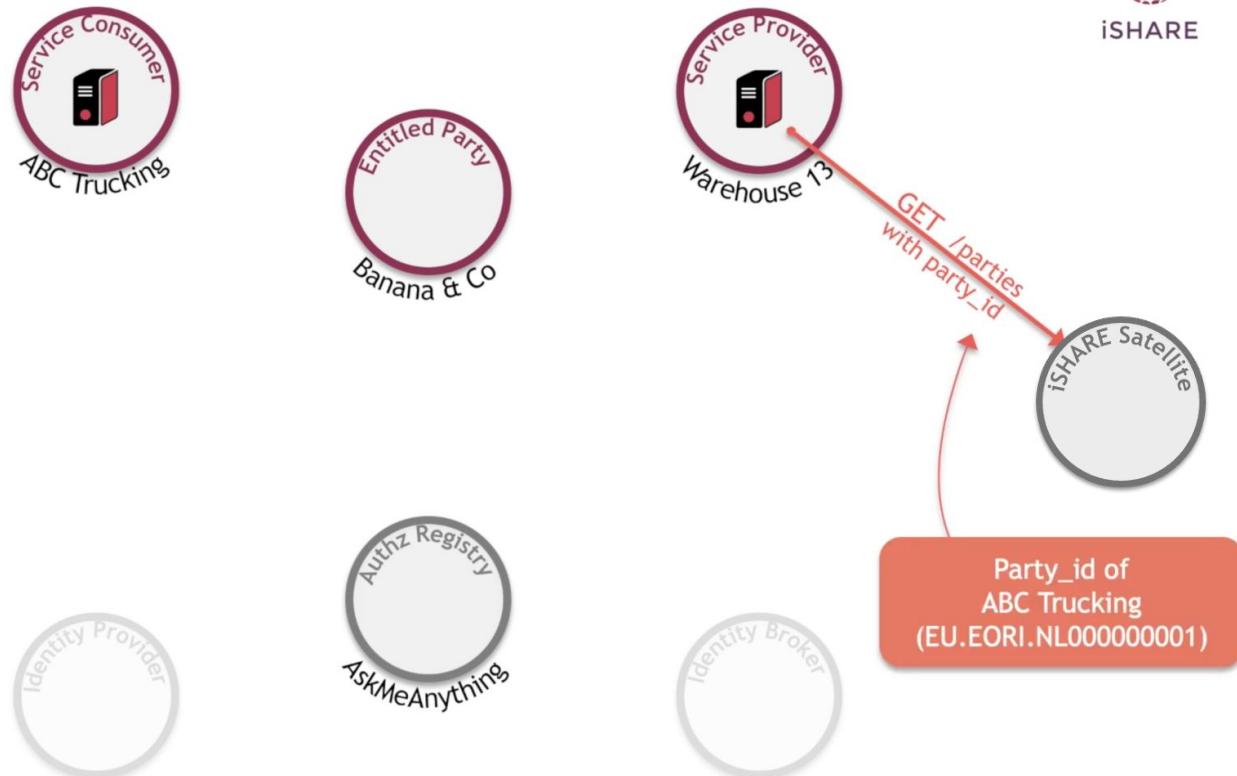
Case 1a

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

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



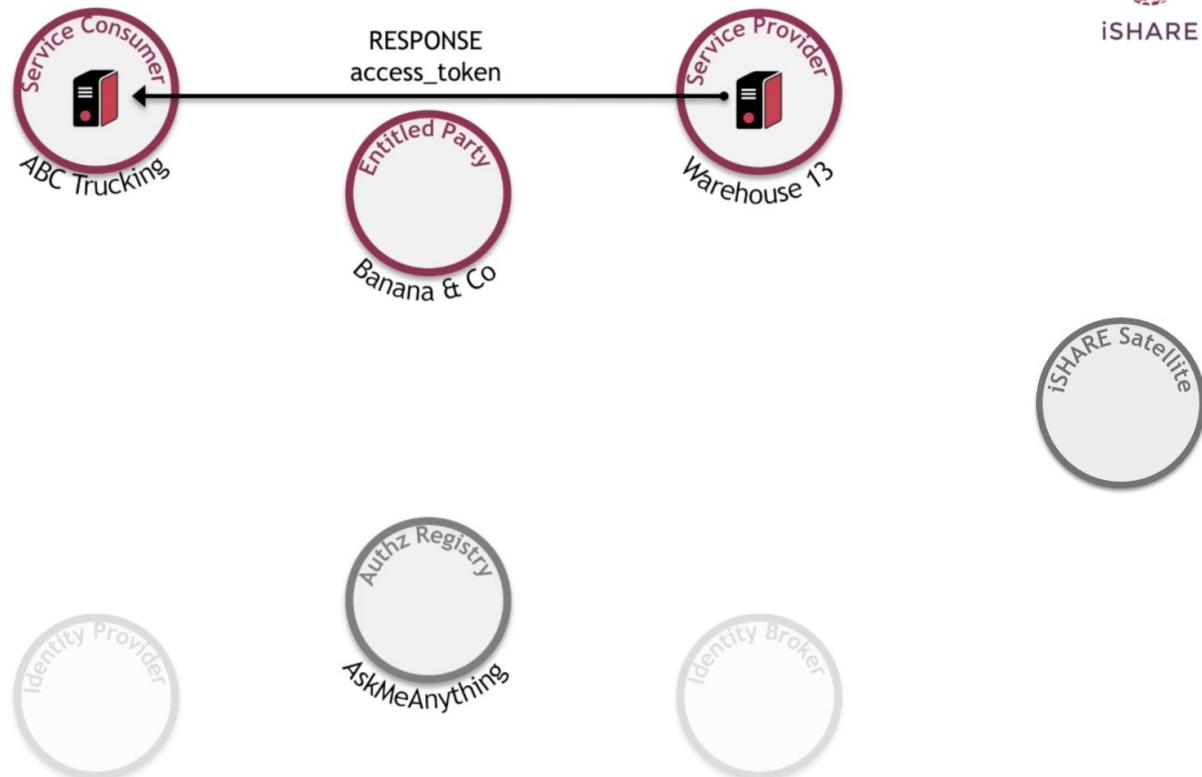
Case 1a

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



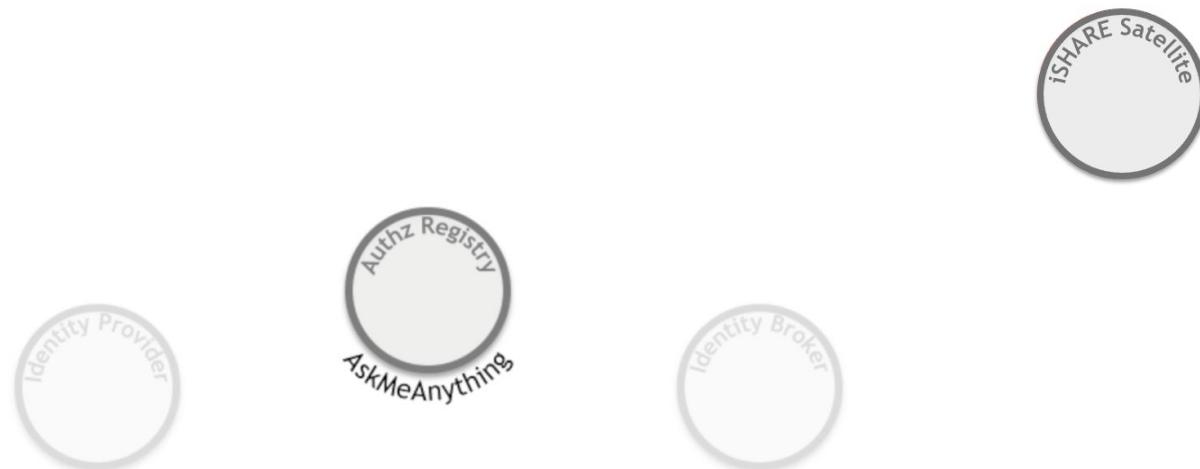
Case 1a

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



RED: not necessarily an iSHARE call



iSHARE

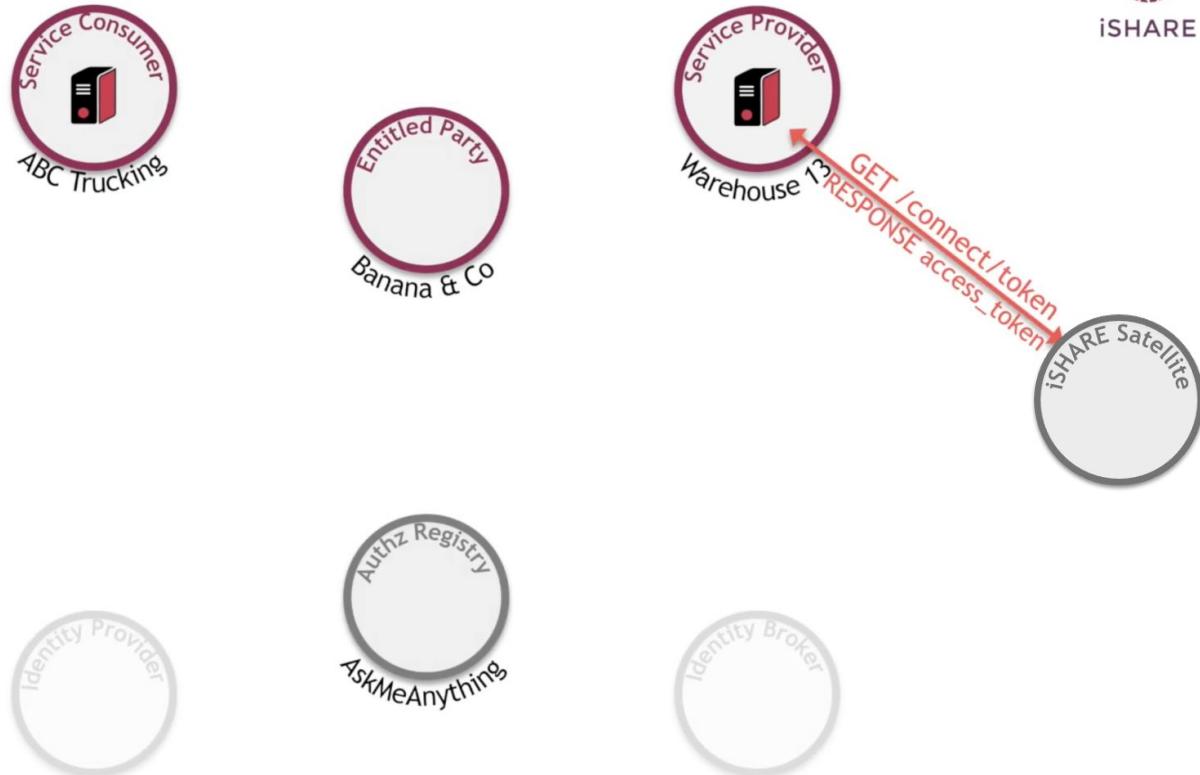
Case 1a

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





iSHARE

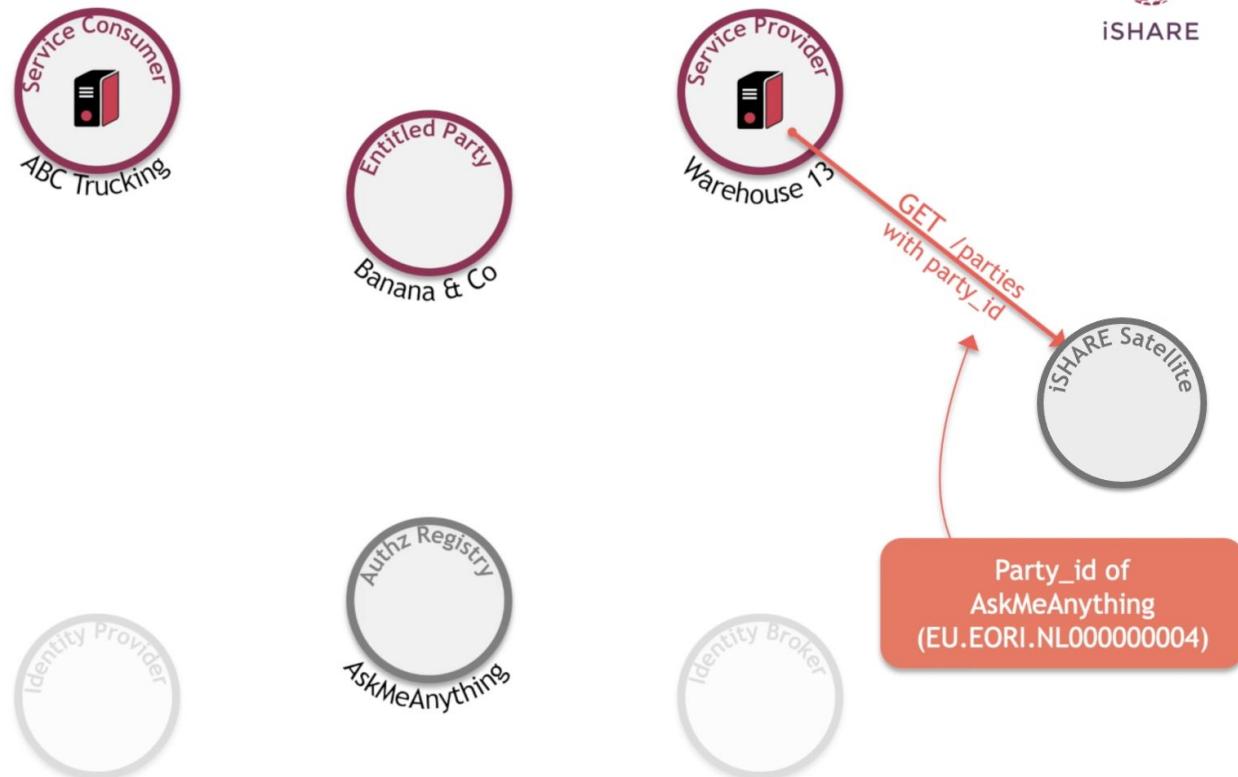
Case 1a

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





iSHARE

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





iSHARE

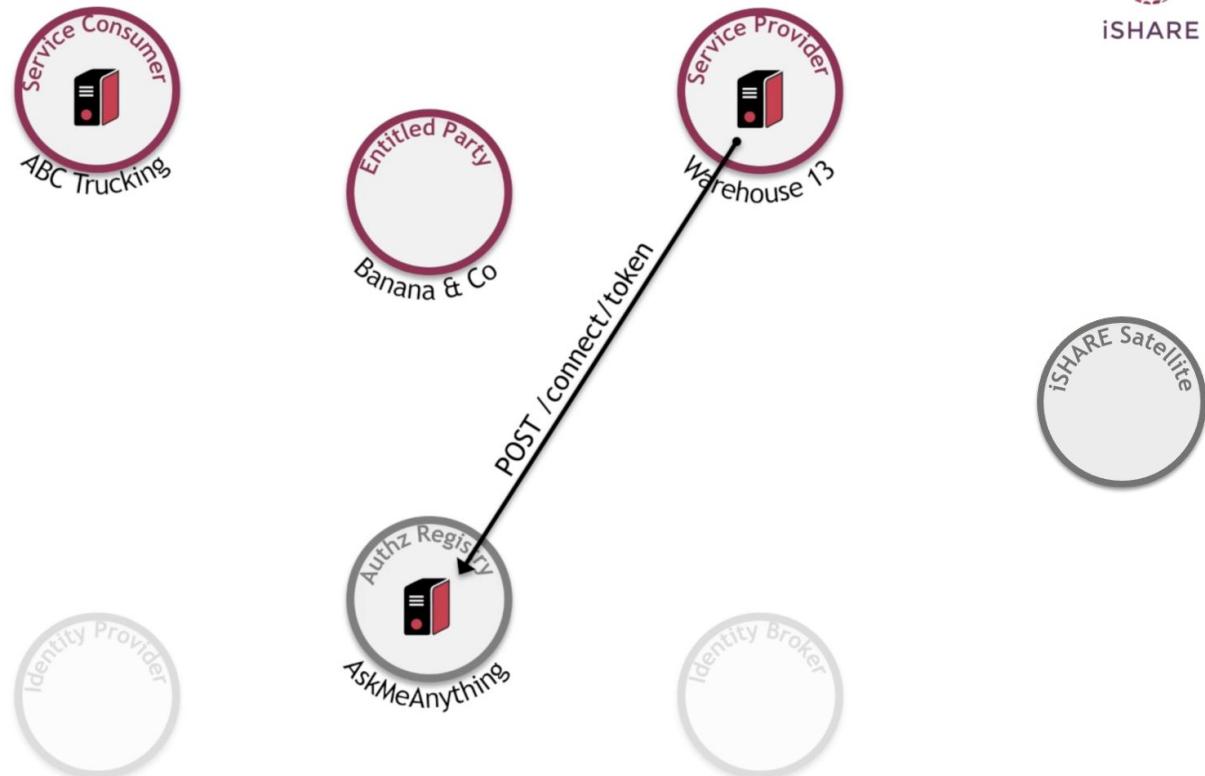
Case 1a

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



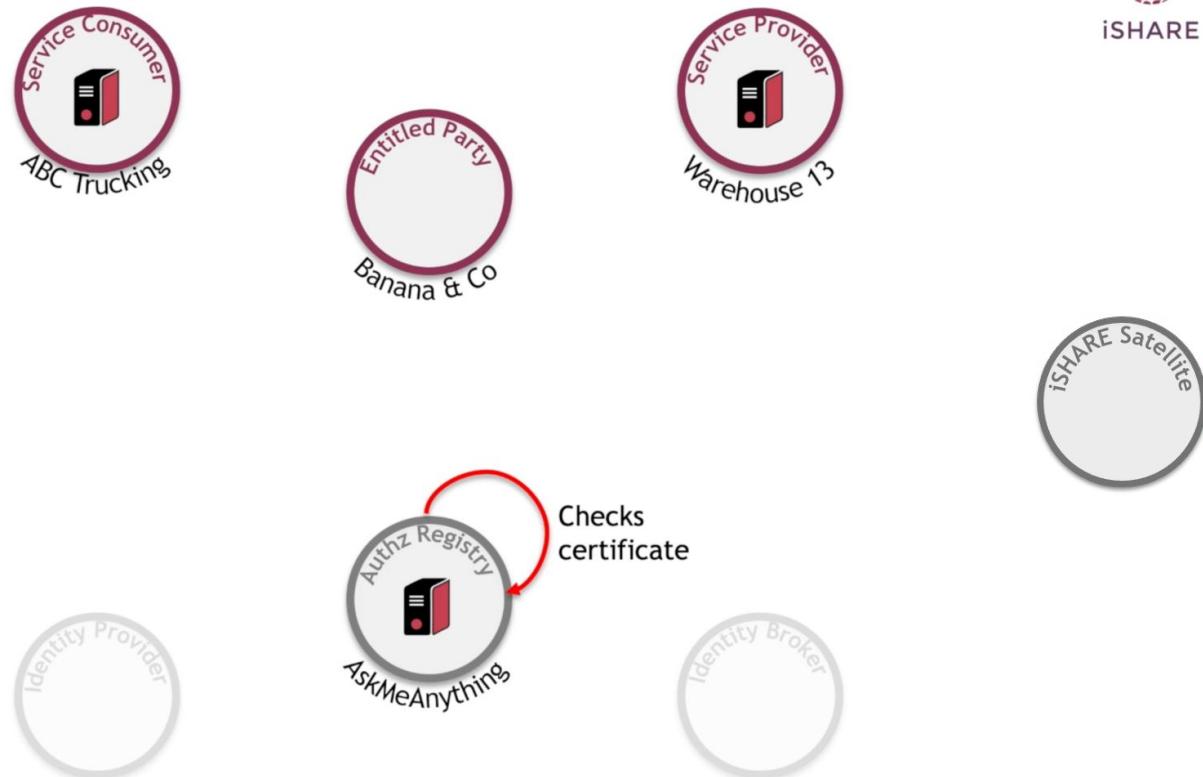
Case 1a

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



RED: not necessarily an iSHARE call

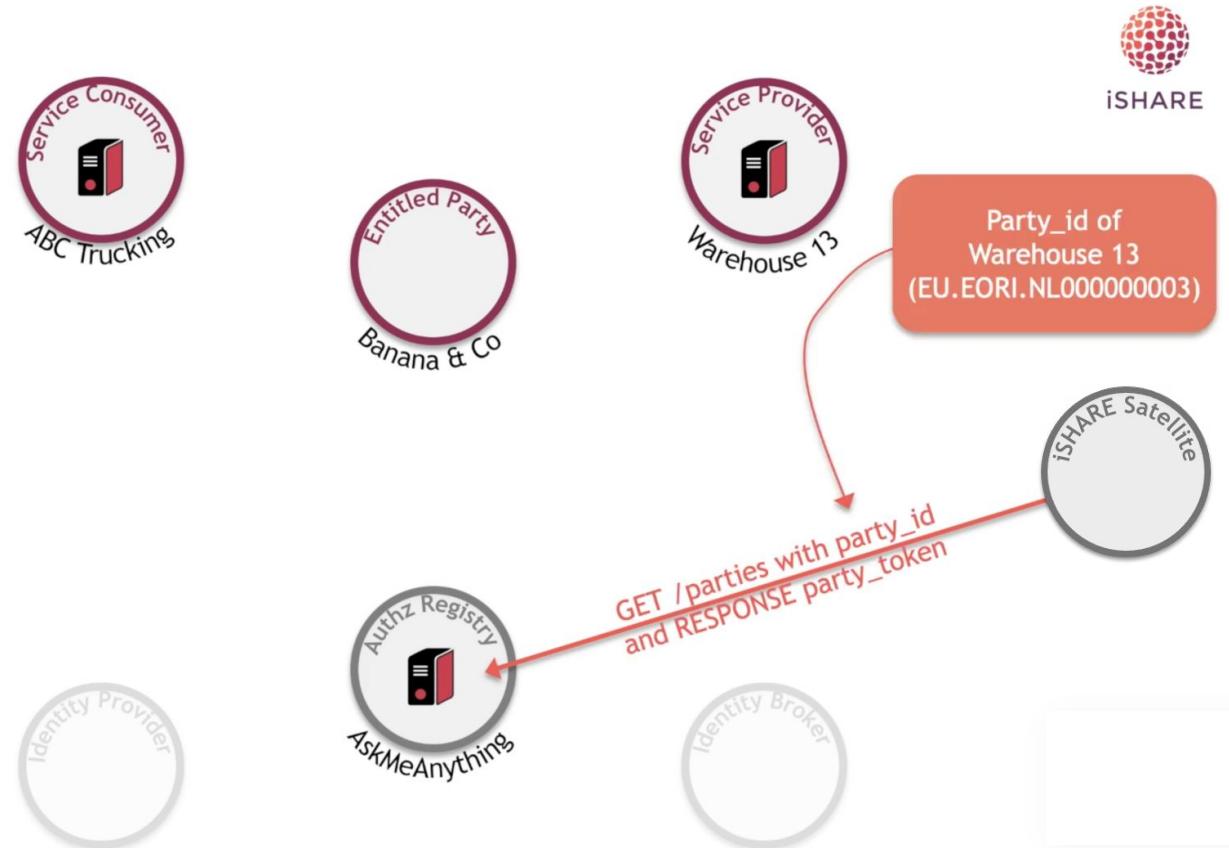
Case 1a

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





iSHARE

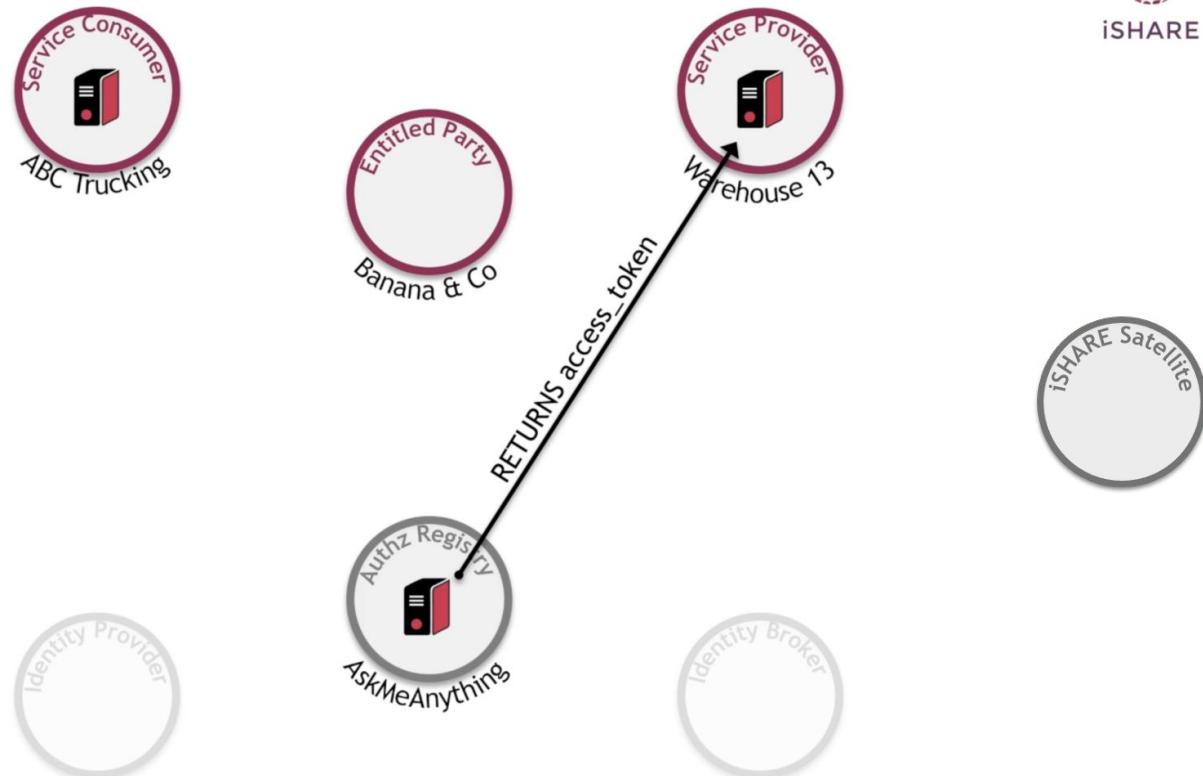
Case 1a

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





iSHARE

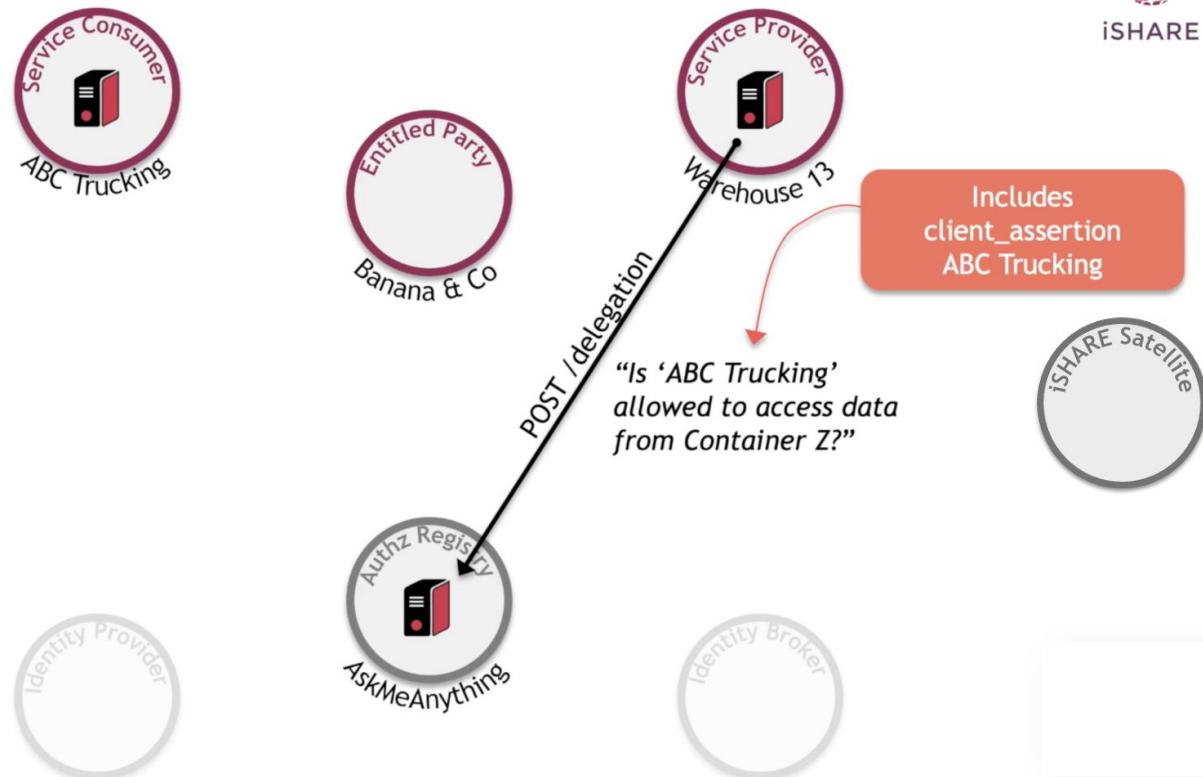
Case 1a

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





iSHARE

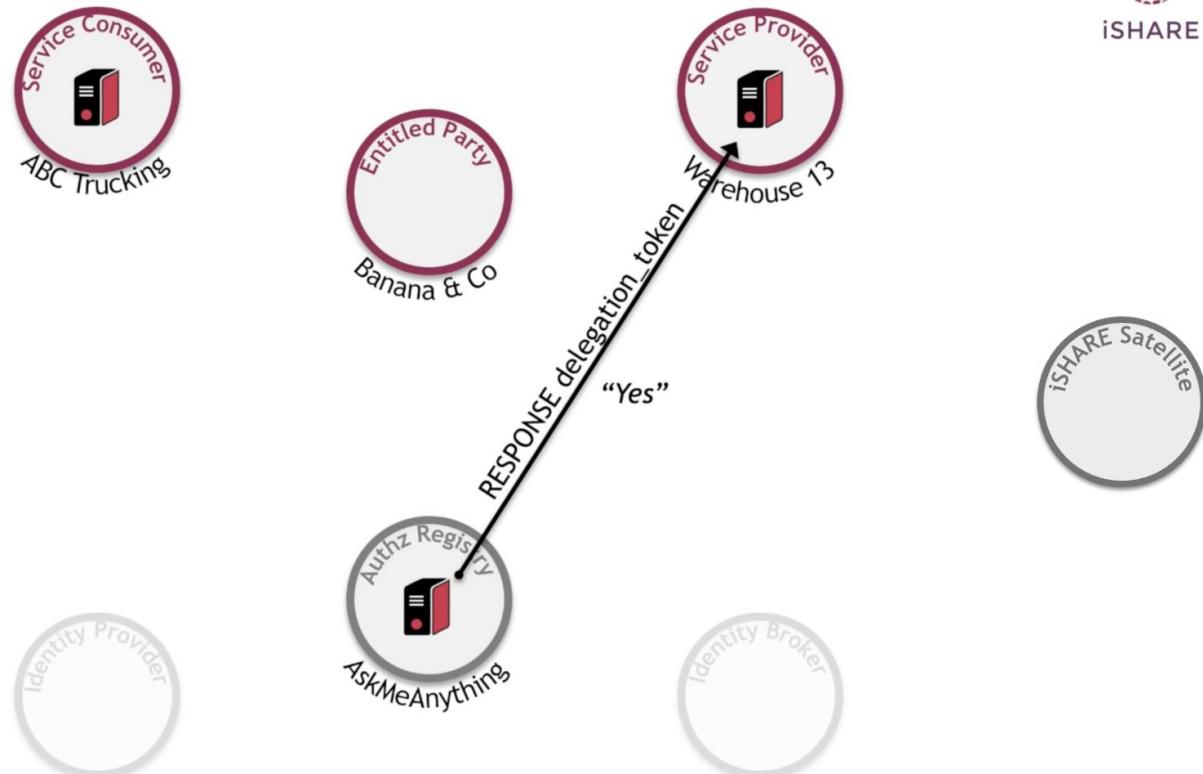
Case 1a

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





iSHARE

Case 1a

M2M

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

Authentication & exchange

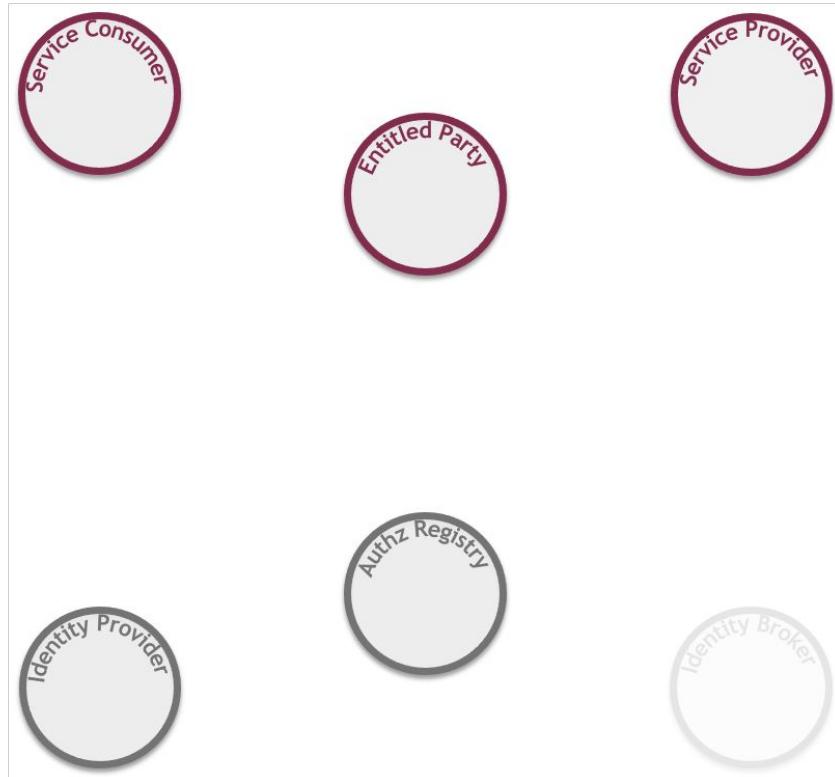
1. Data request
2. SP verifies AR
3. AR verifies SP
4. Return access token
5. Delegation evidence
6. Data exchange



RED: not necessarily an iSHARE call

Participants of the fictional use case

- **Banana & Co**
Freight 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



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 Trucking, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



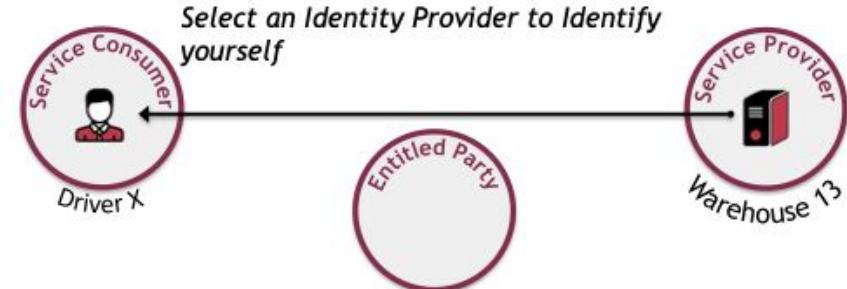
Case 2 H2M

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



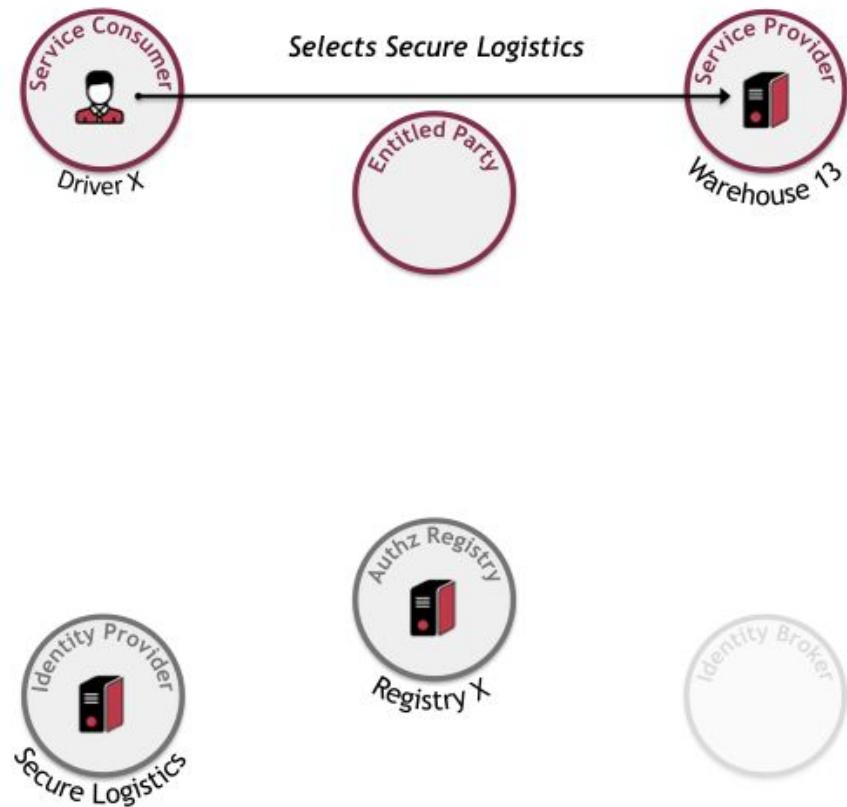
Case 2 H2M

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



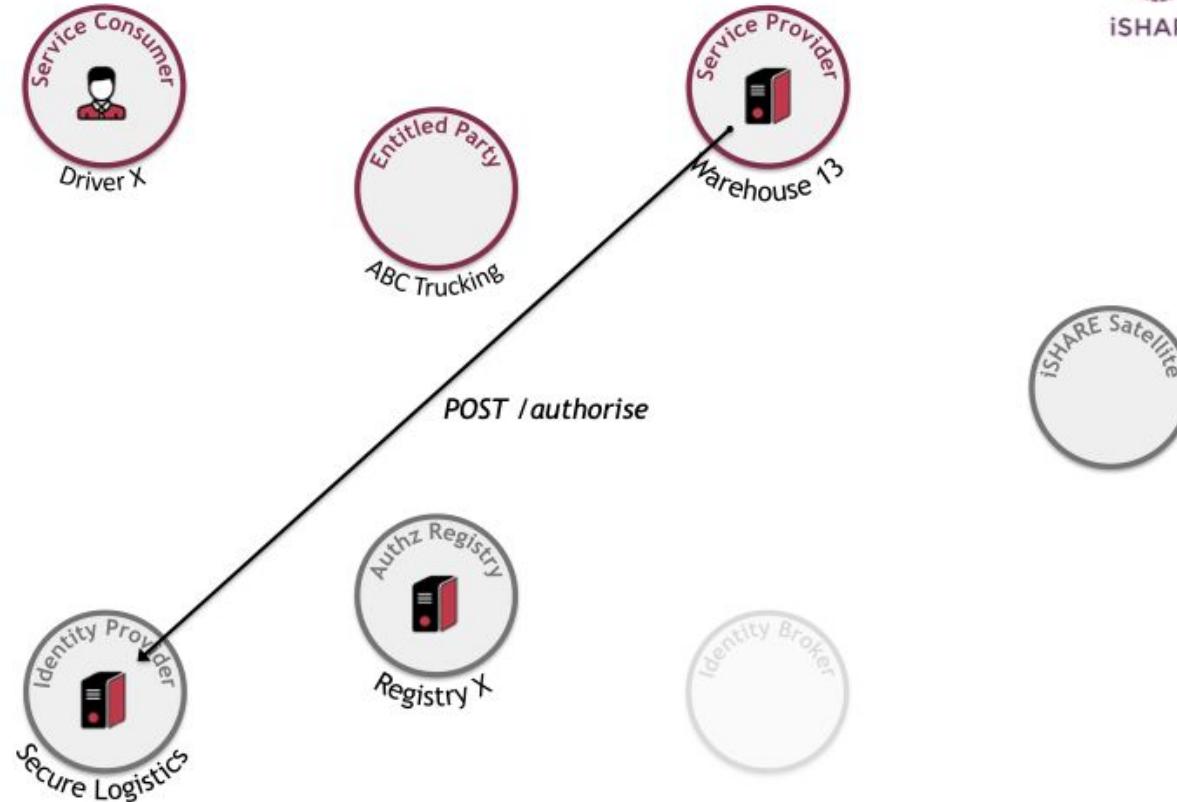
Case 2 H2M

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.



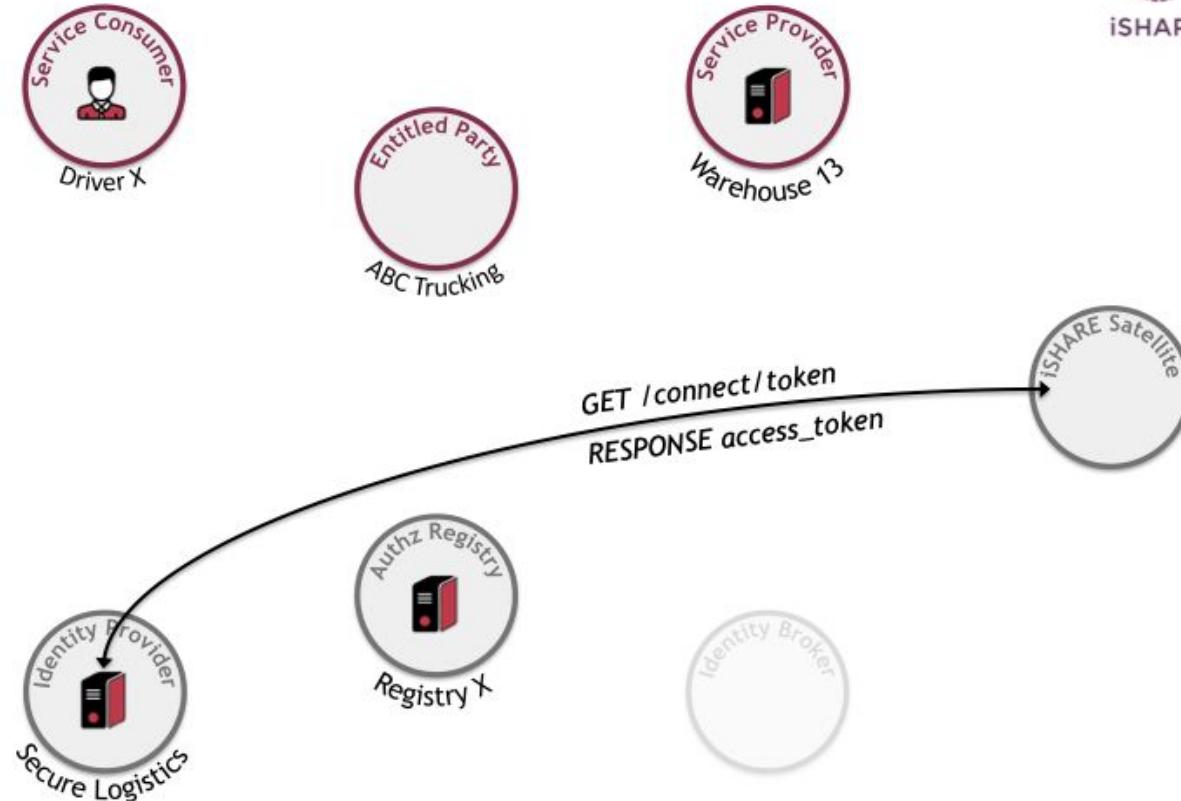
Checks certificate



RED: not necessarily an iSHARE call

Case 2 H2M

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



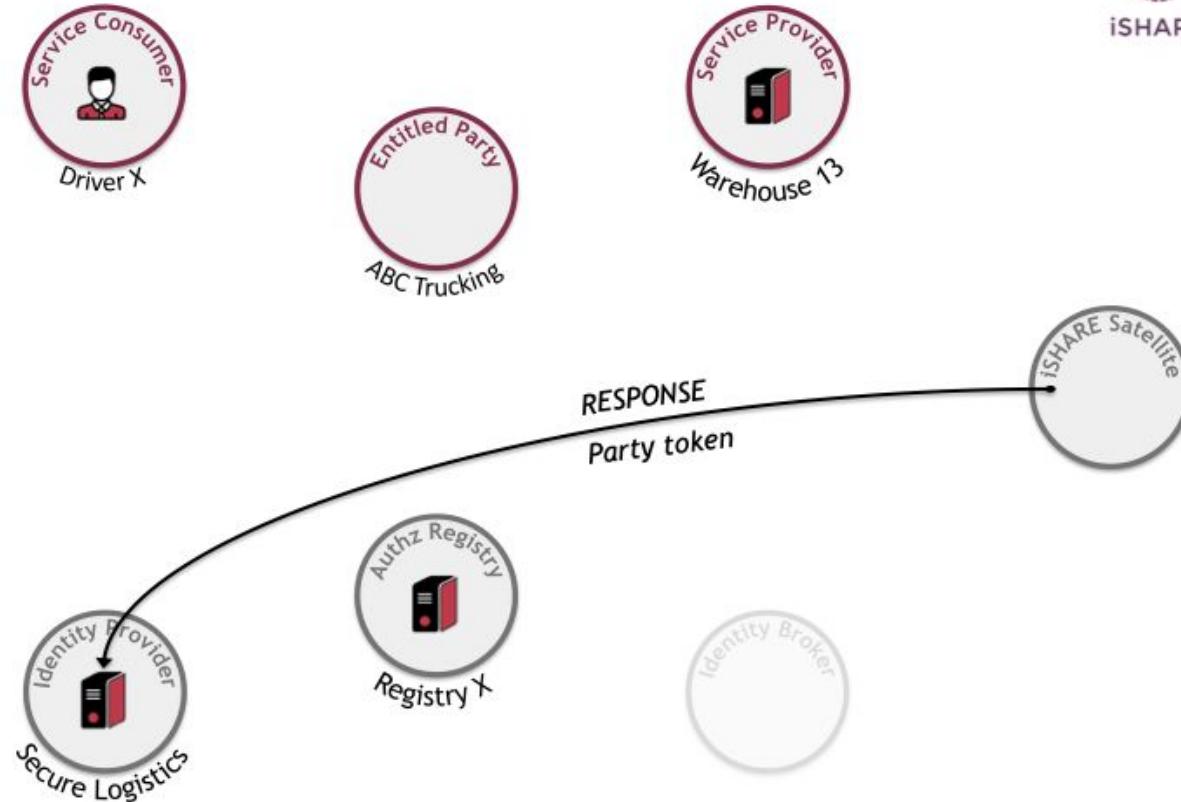
Case 2 H2M

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.



Case 2 H2M

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

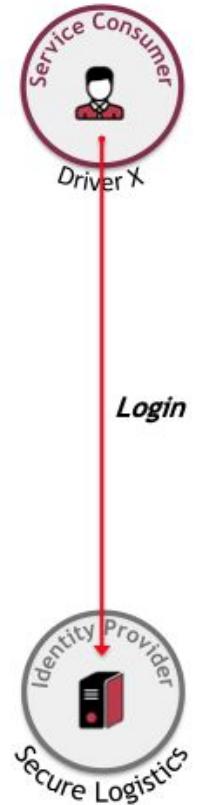


RED: not necessarily an iSHARE call



Case 2 H2M

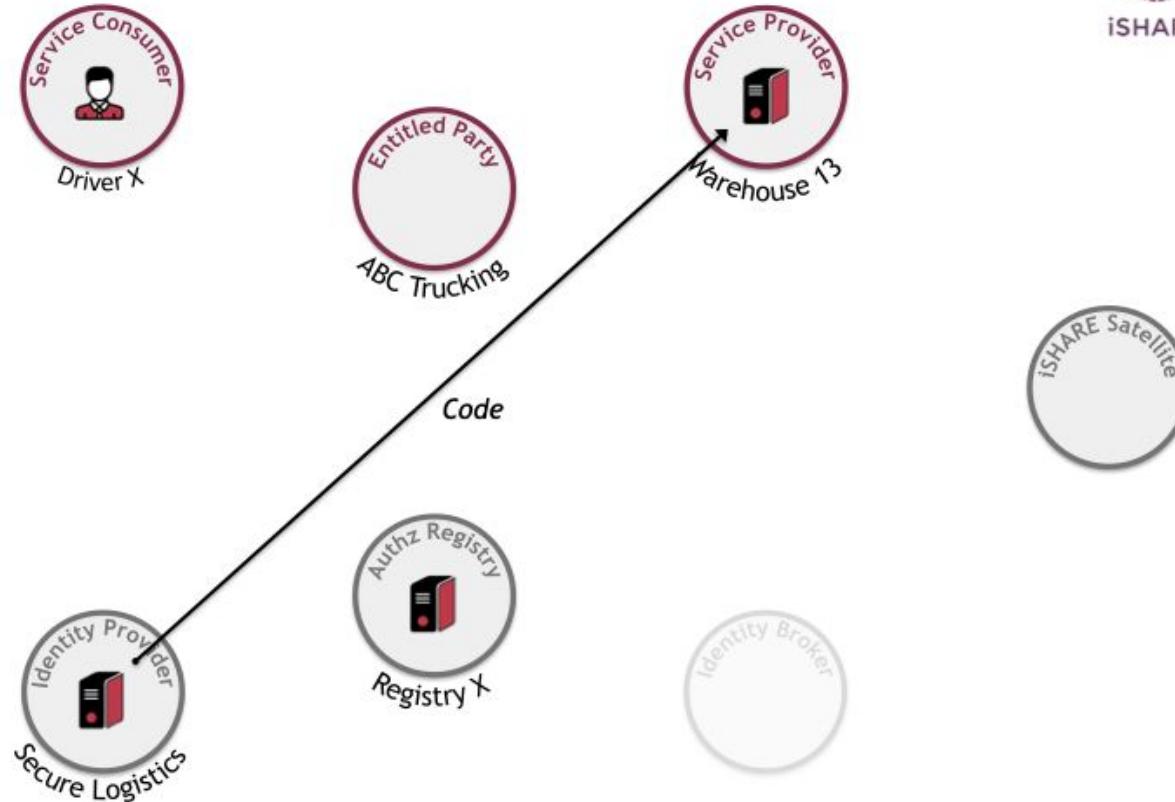
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



RED: not necessarily an iSHARE call

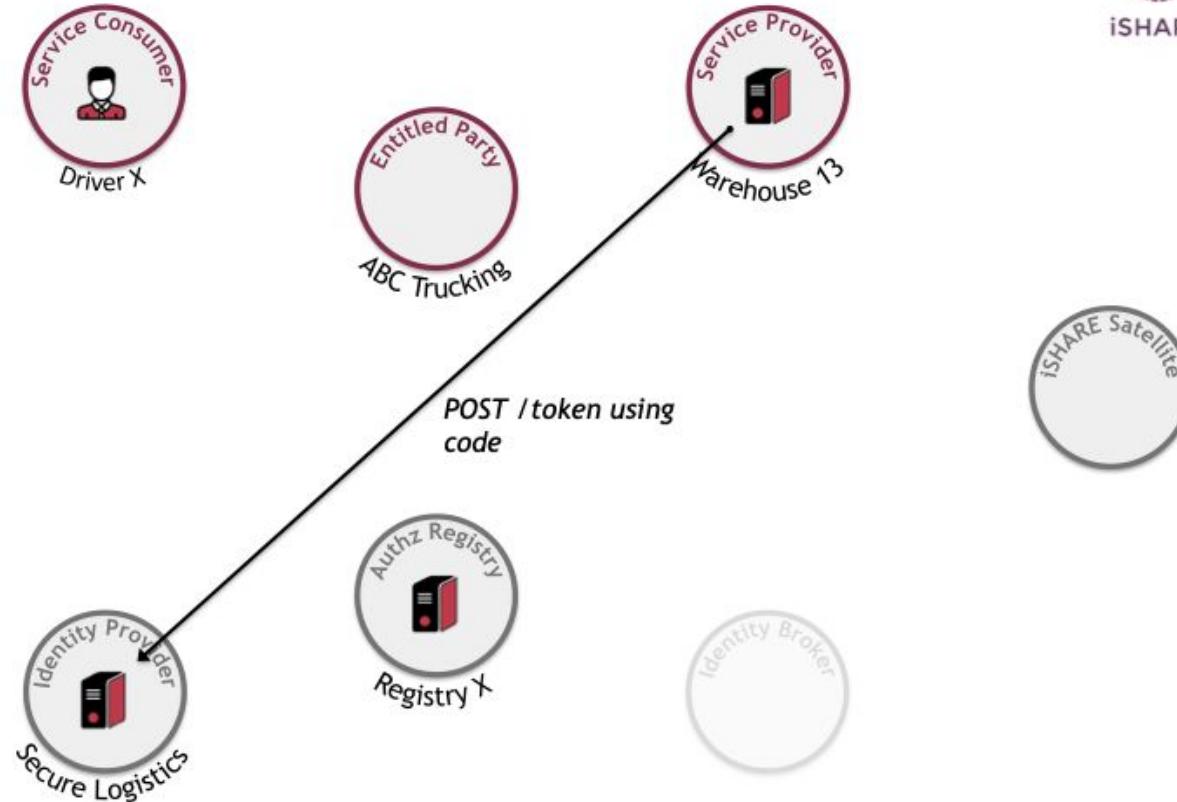
Case 2 H2M

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



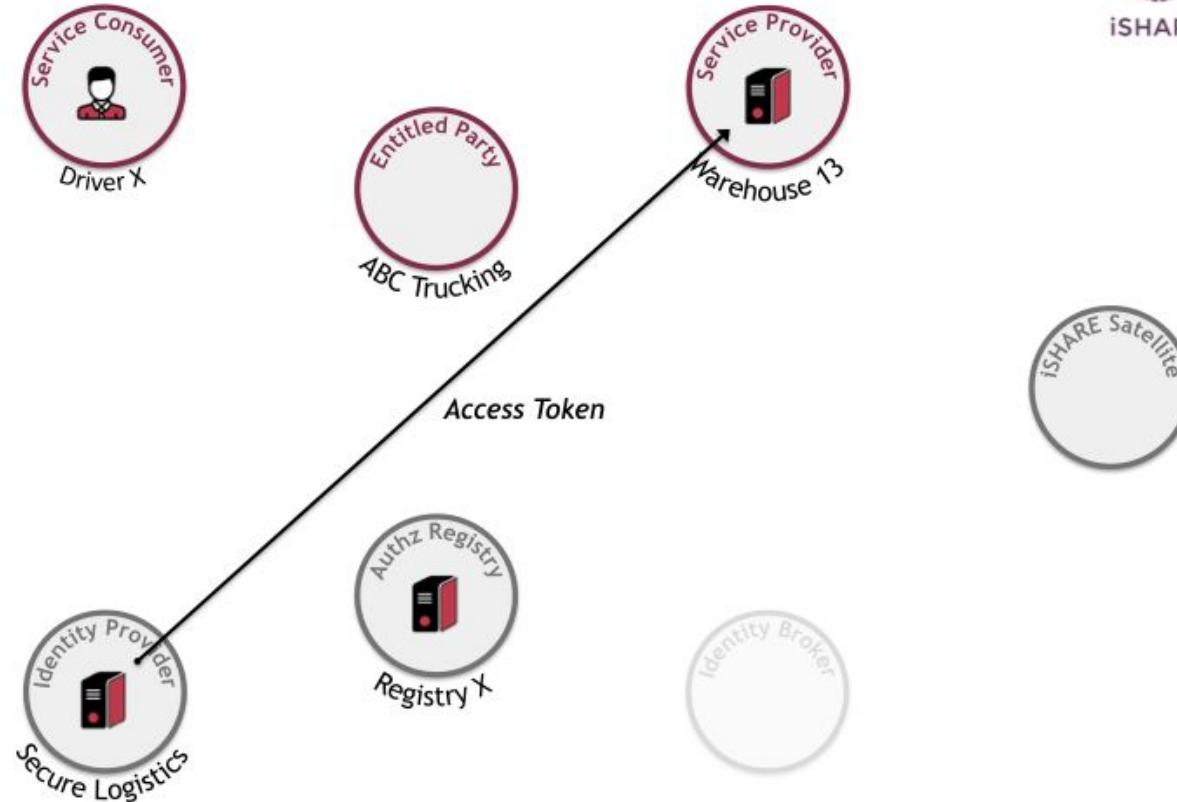
Case 2 H2M

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



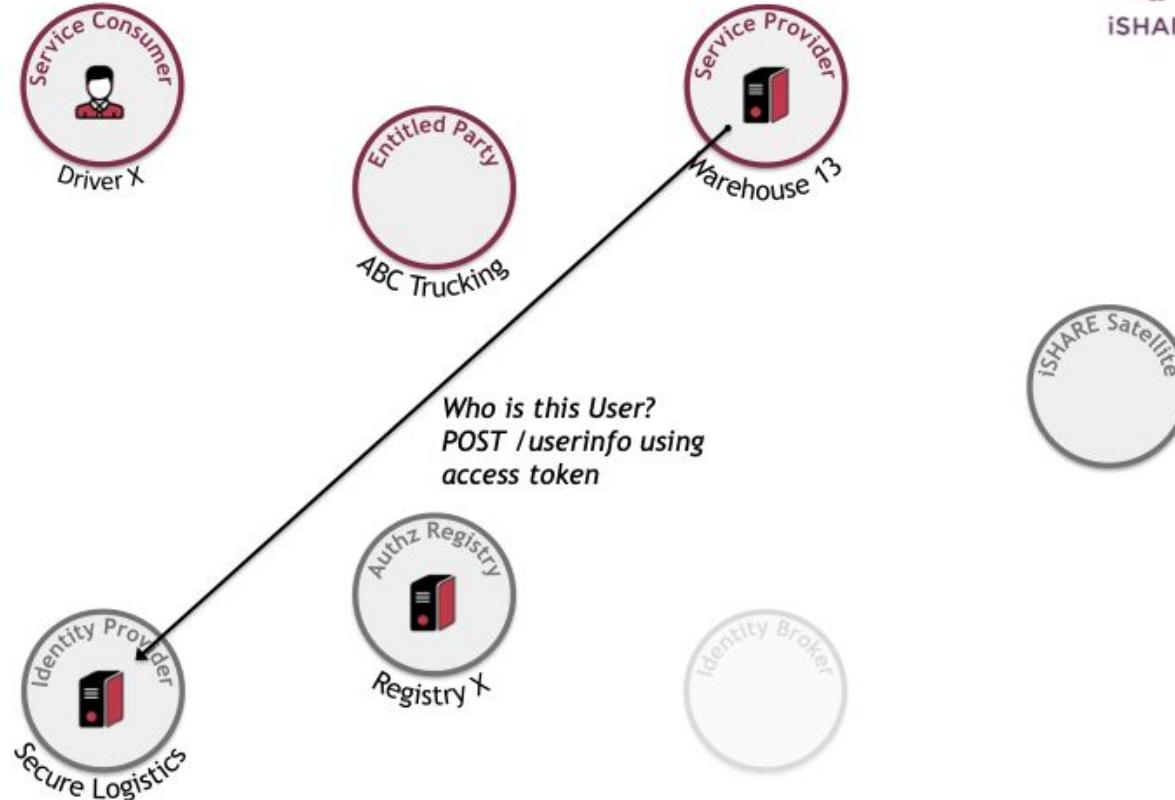
Case 2 H2M

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



Case 2 H2M

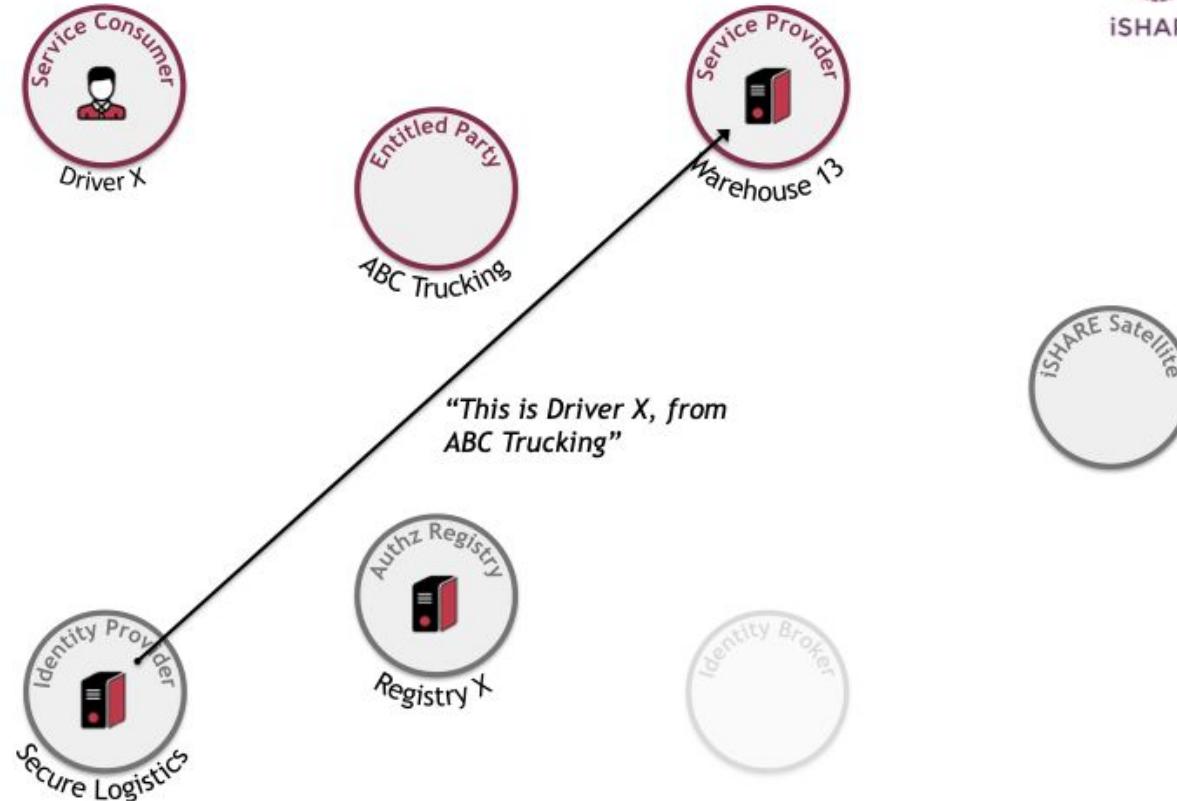
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.



Case 2

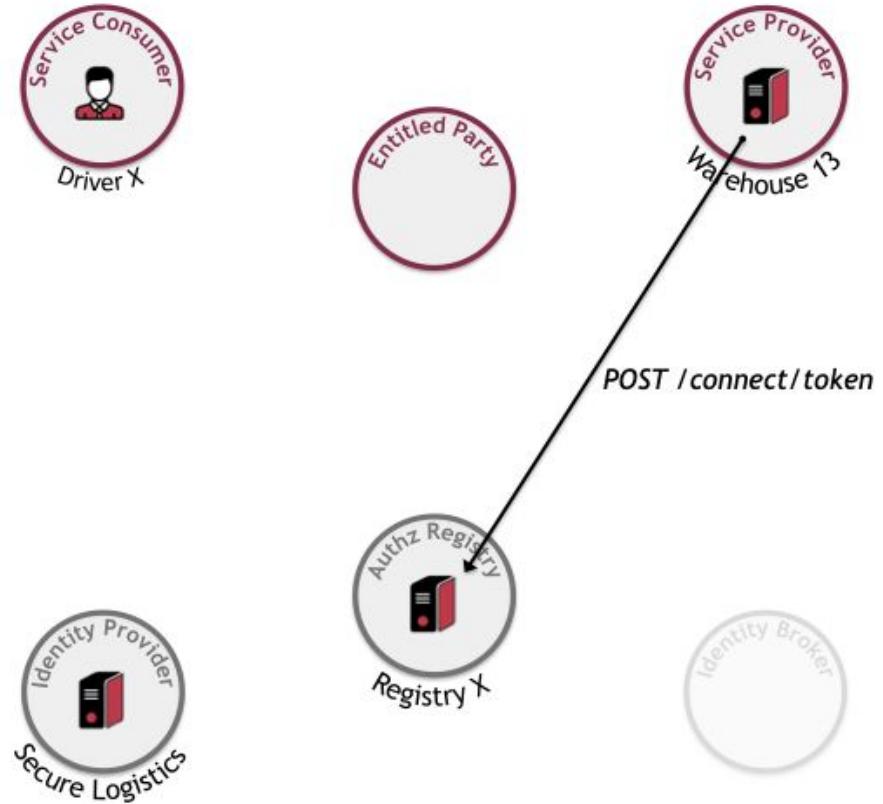
H2M

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.



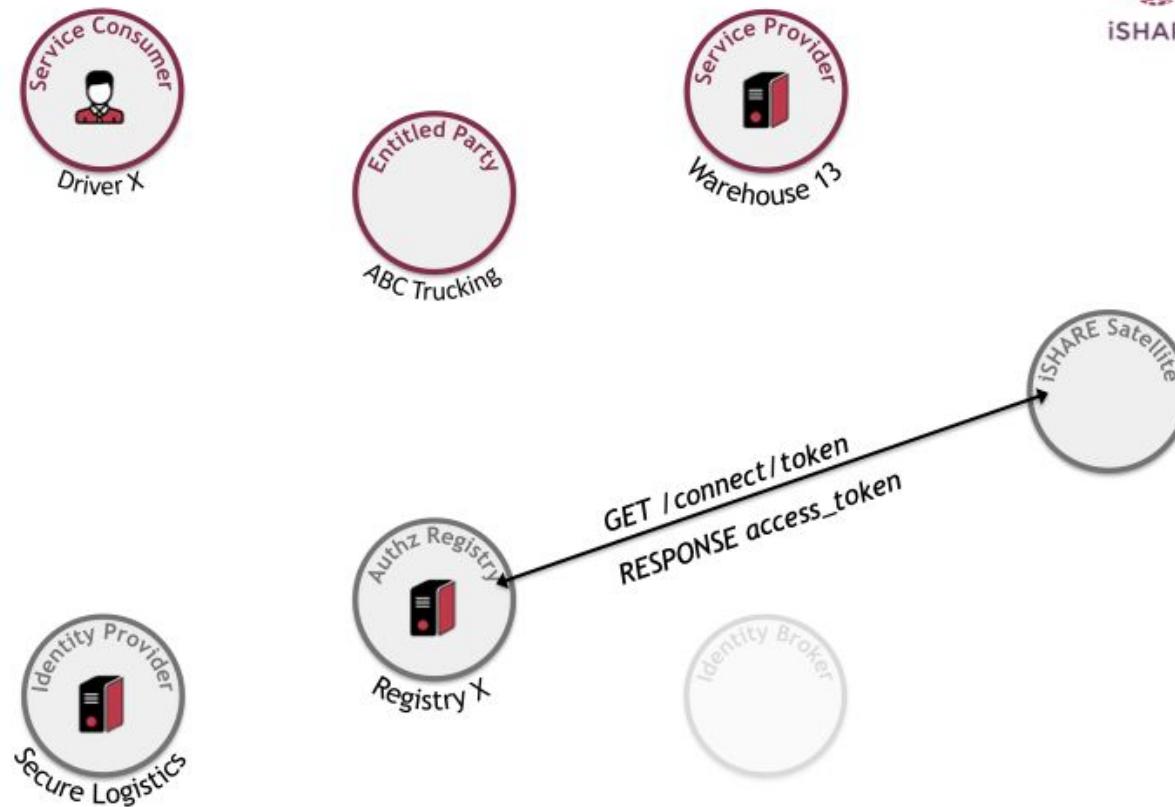
Case 2 H2M

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



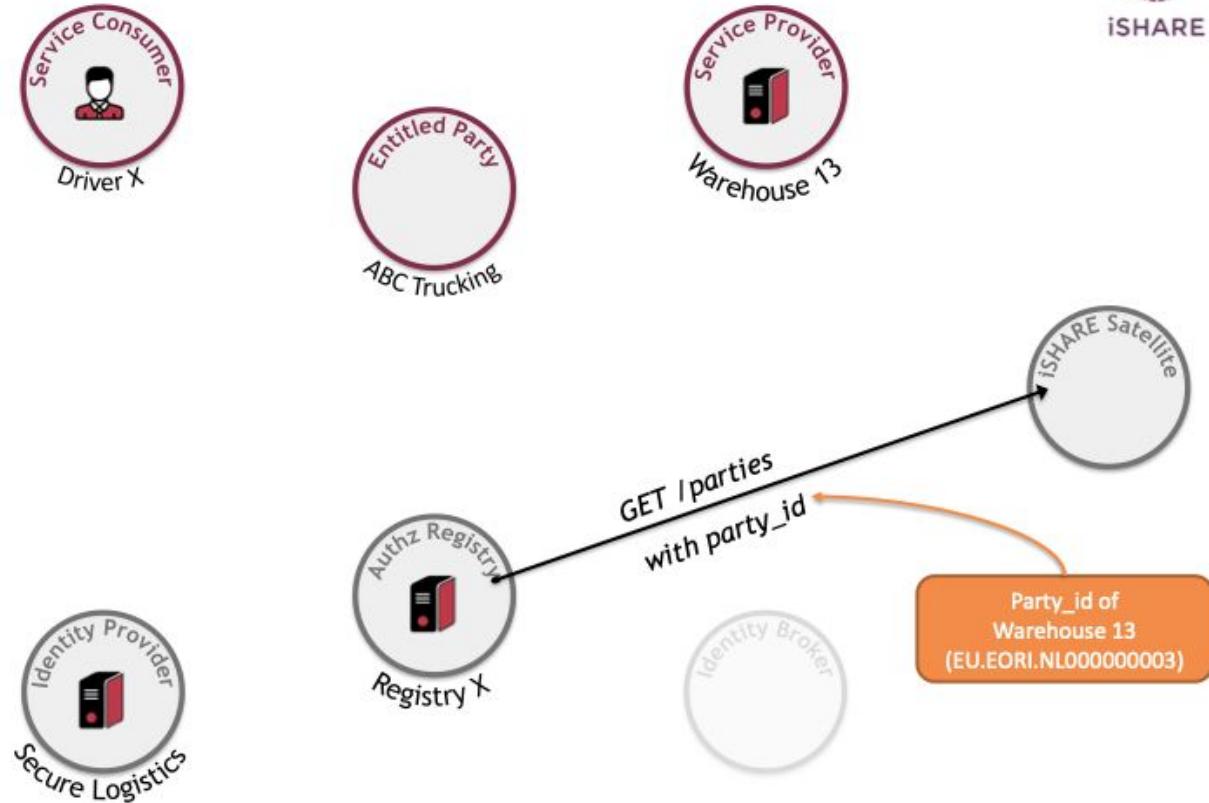
Case 2 H2M

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



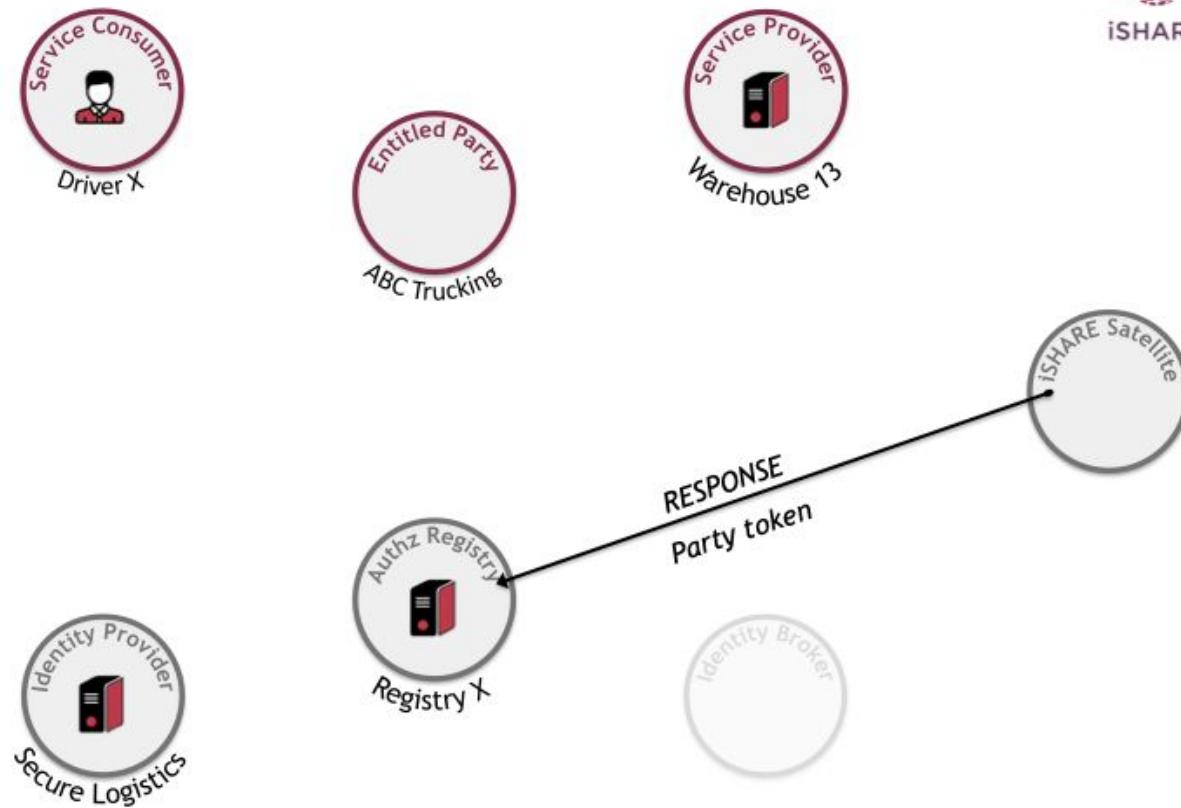
Case 2 H2M

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



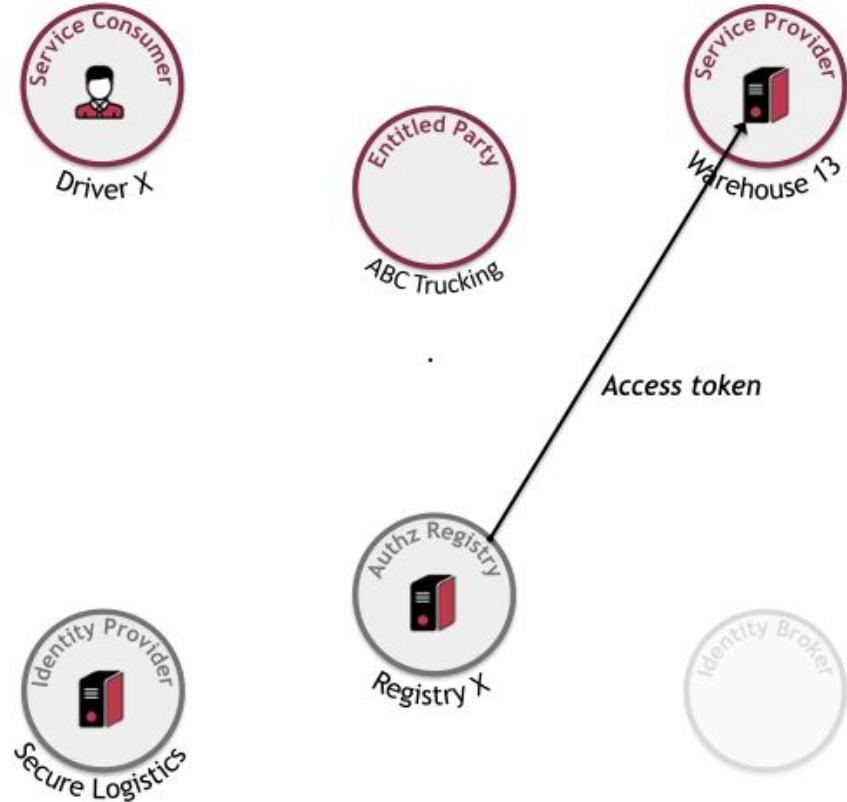
Case 2 H2M

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



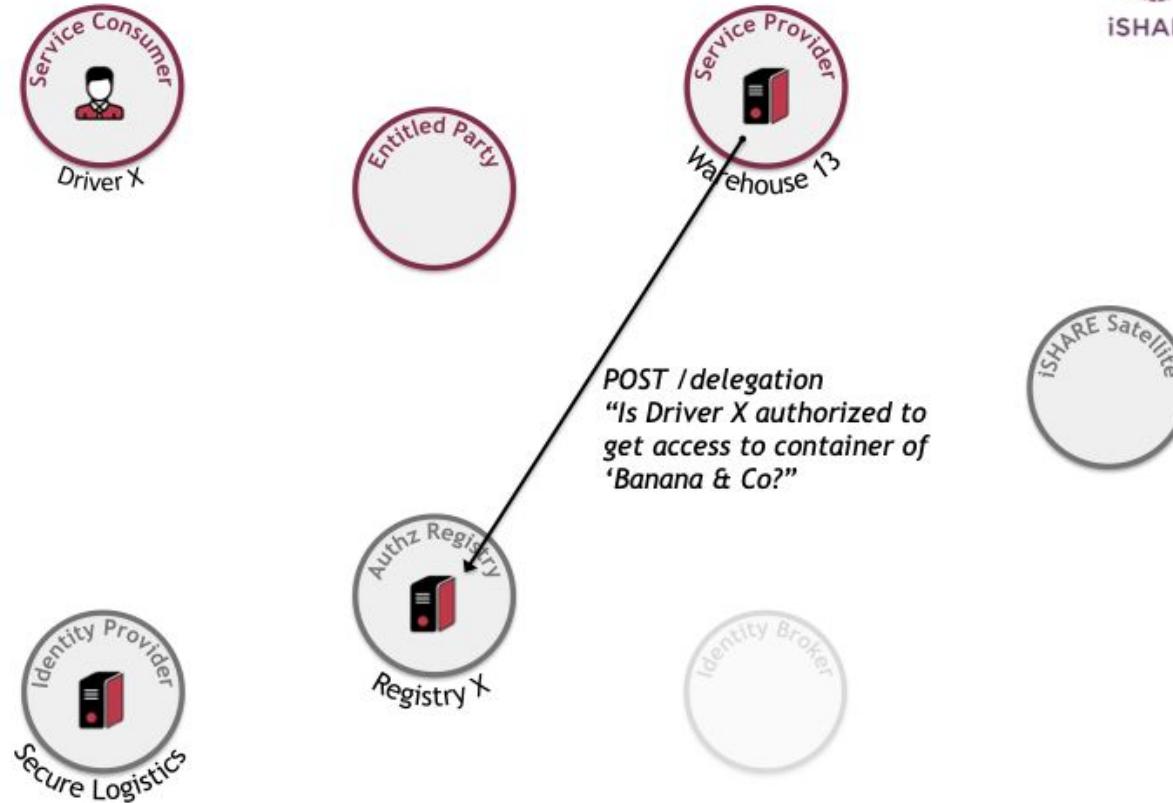
Case 2 H2M

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.



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.



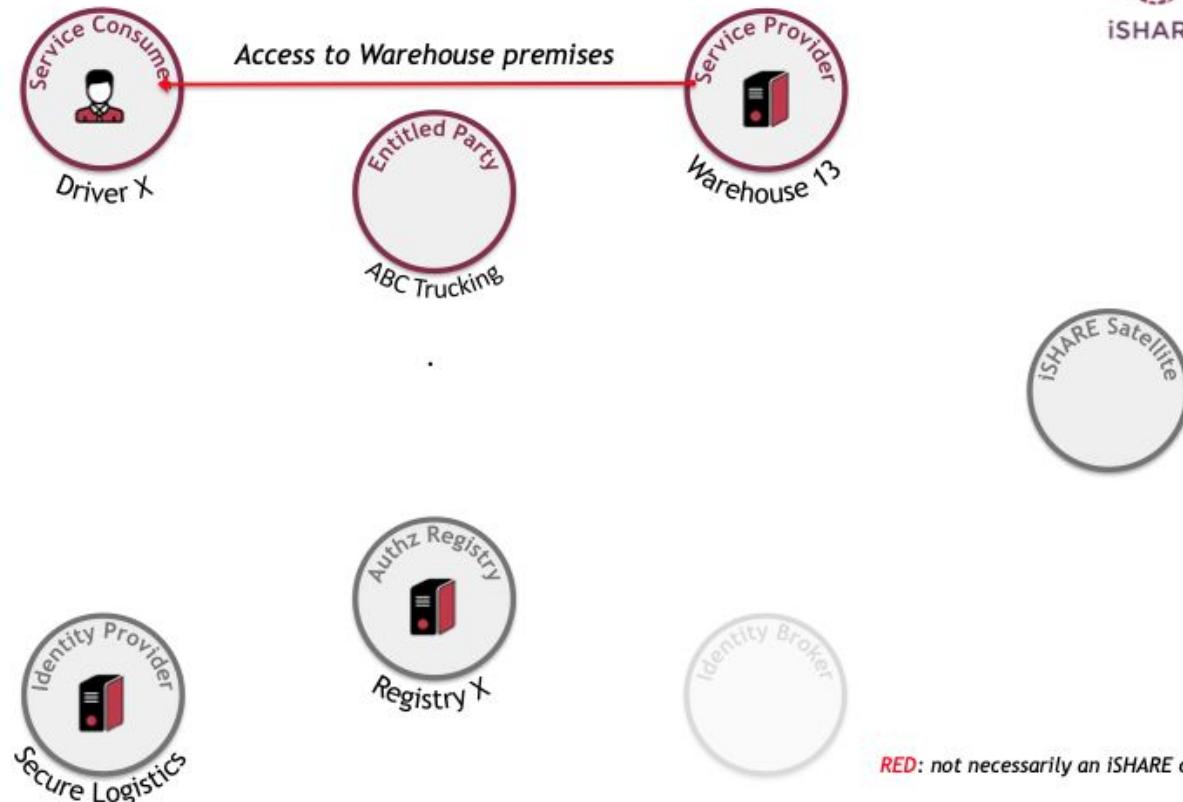
Checks evidence and certificate



RED: not necessarily an iSHARE call

Case 2 H2M

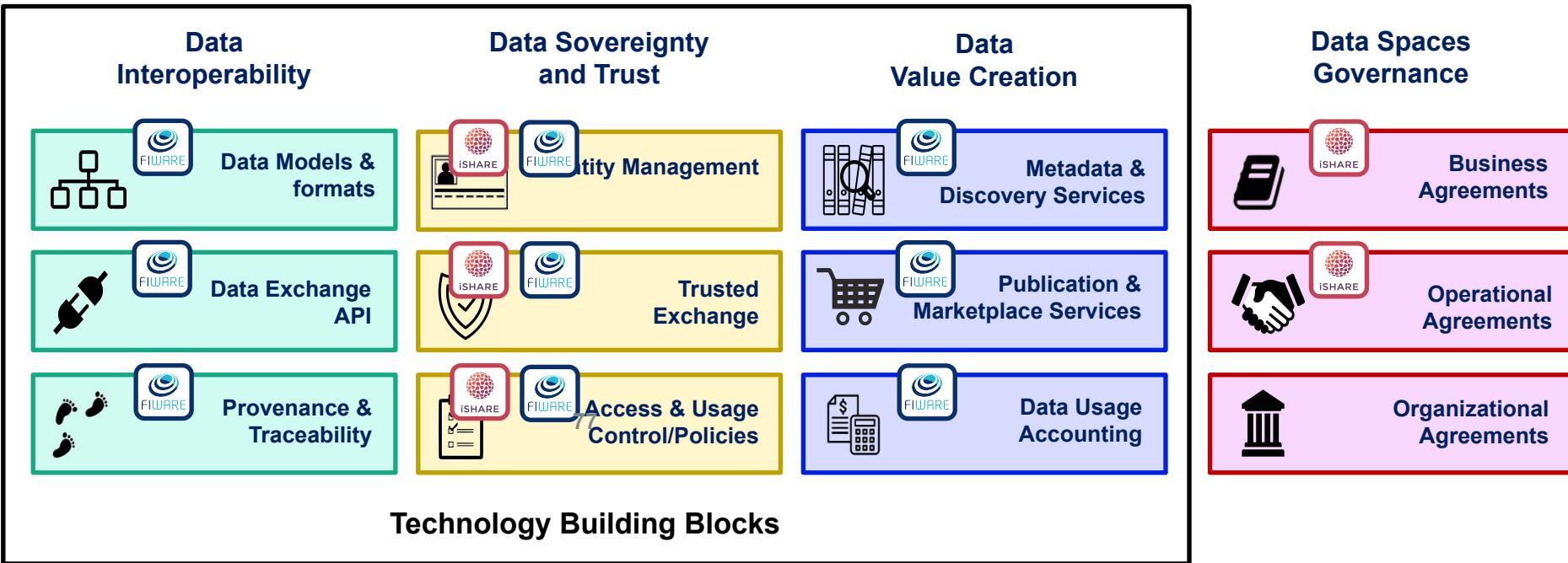
Driver X, trucker from ABC Trucker, wants to access Terminal premises on behalf of Banana & Co to pickup its container.

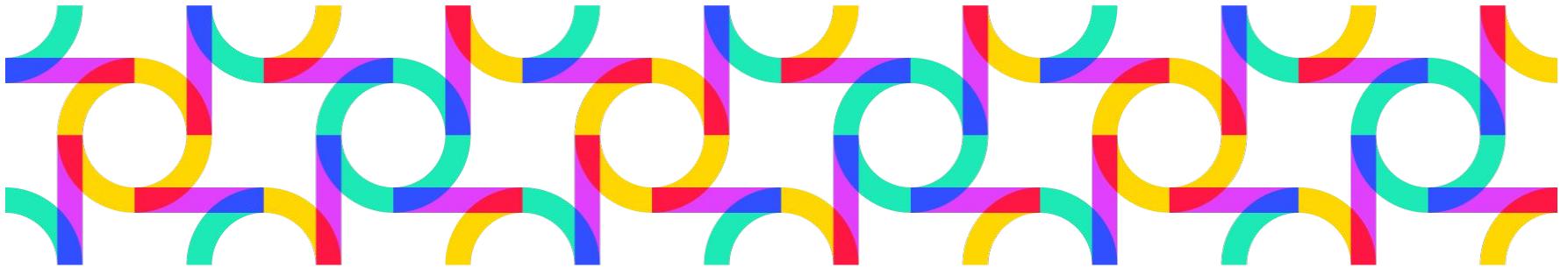


Further reading and useful links

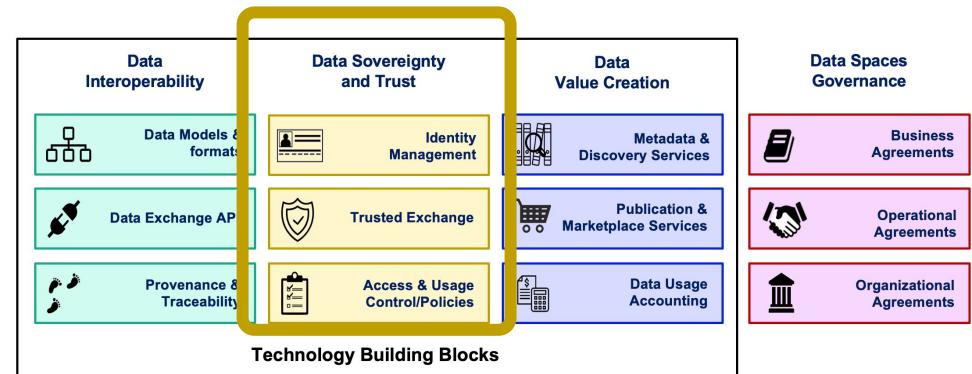
- iSHARE framework covering all aspects - <https://scheme.ishareworks.org>
- iSHARE developer portal - <https://dev.ishareworks.org>
 - Postman collections for fictional use case -
<https://dev.ishareworks.org/demo-and-testing/postman.html#postman-collections>
 - Dummy participants in various roles for 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: boilerplate for creating dataspaces

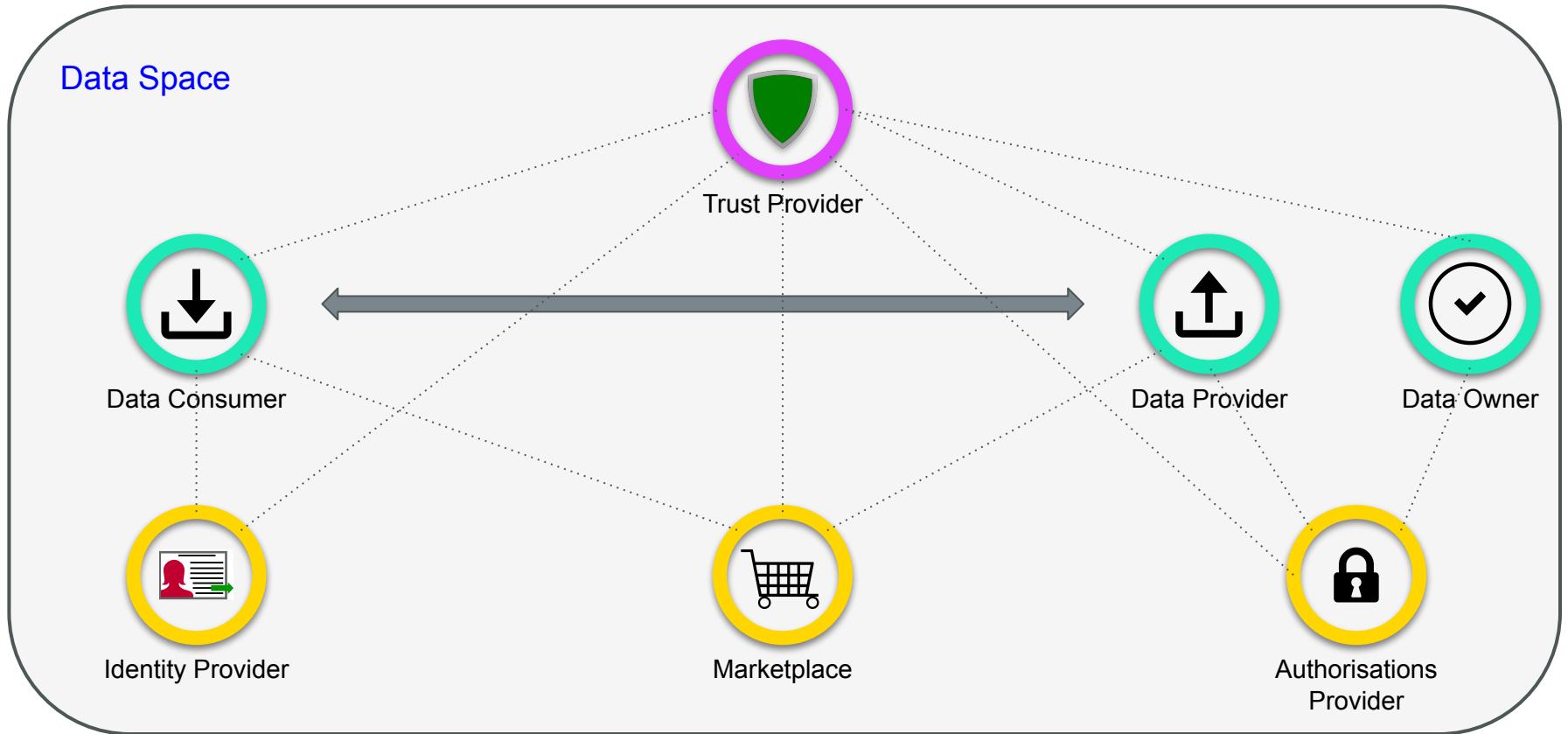




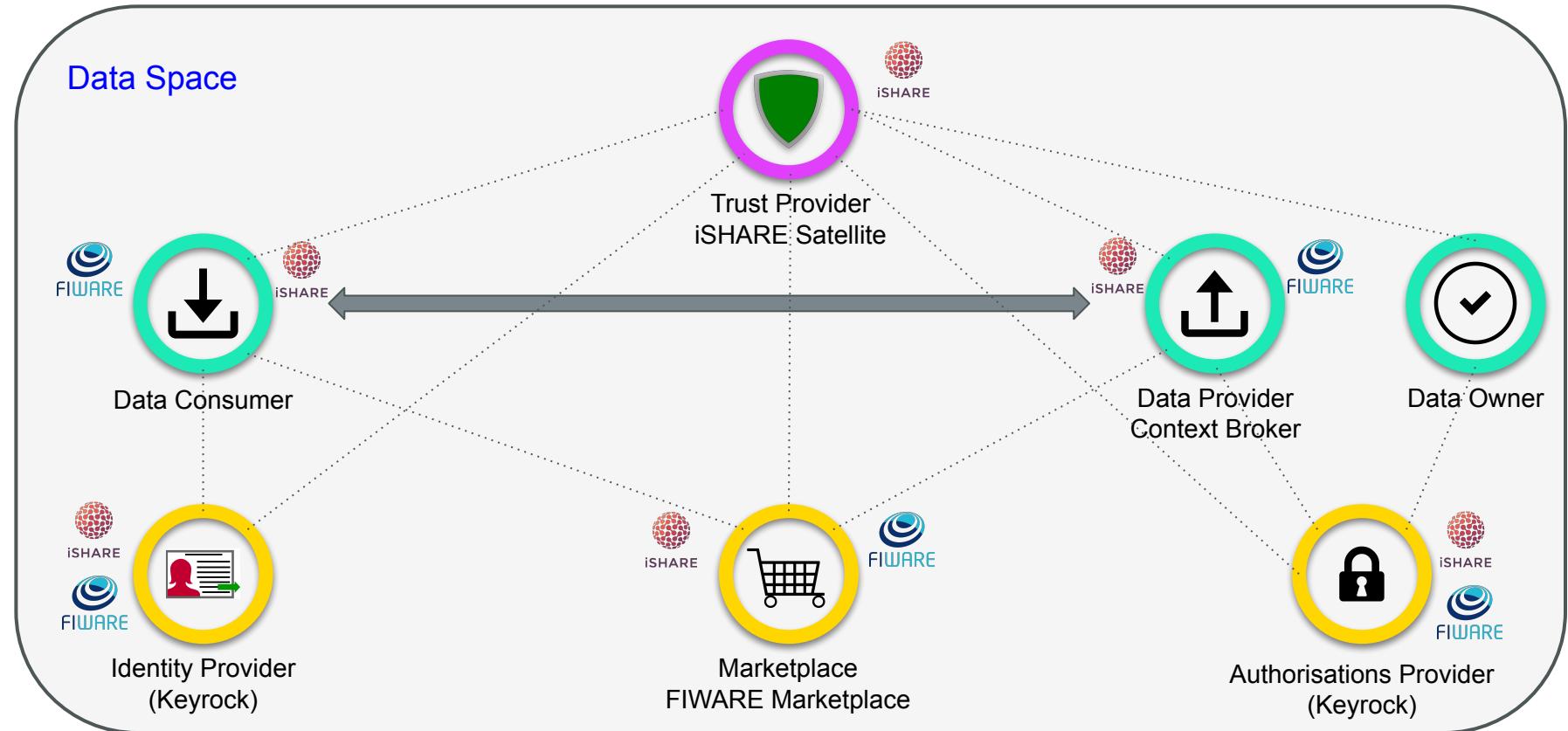
Data Sovereignty and Trust



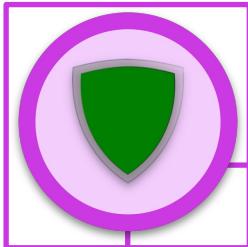
Recap: Roles in a Data Space



Open source components are available for these roles



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



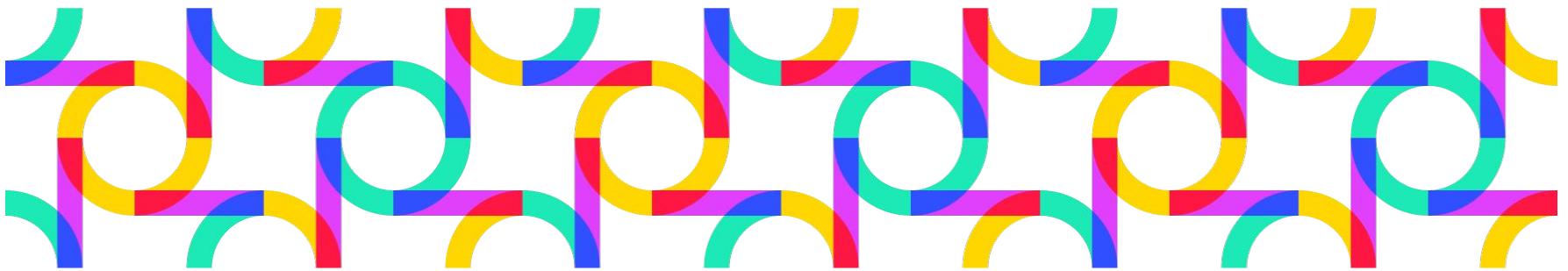
- The Identity Provider role deals with the human identities with varying level of assurances, as defined in eIDAS framework, to support various use cases
- Depending on the criticality of the data, appropriate level of assurance for an identity can be requested
- iSHARE specifications are designed such 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.

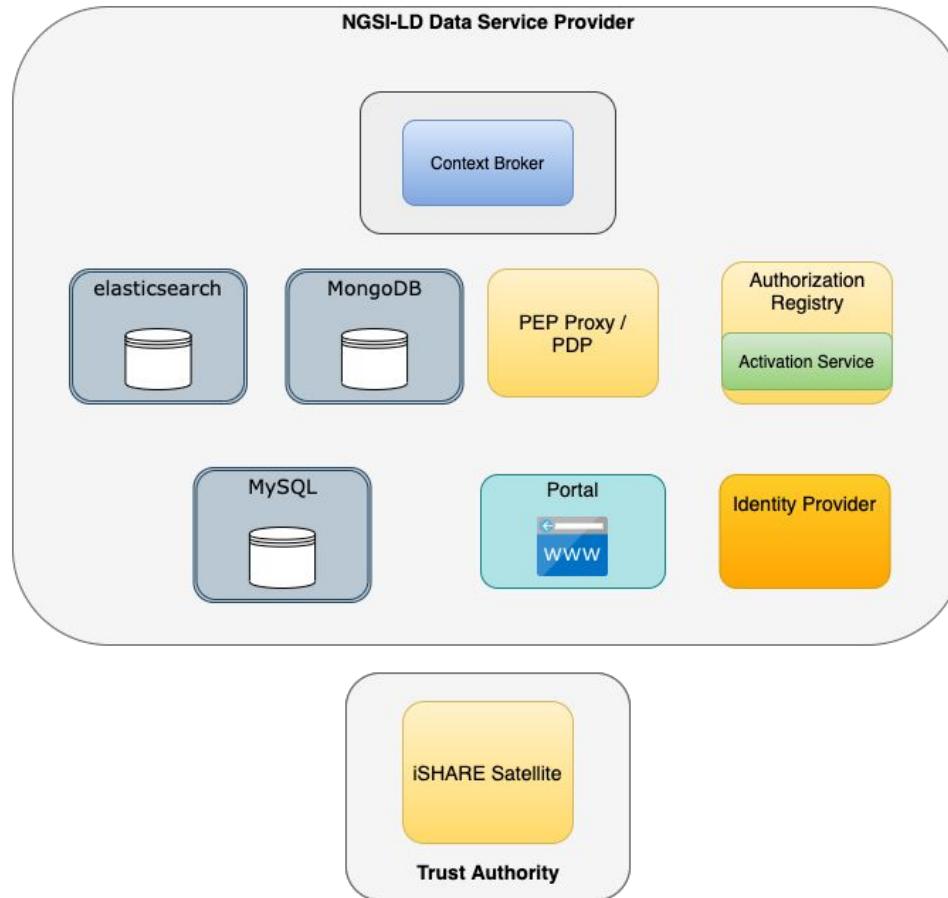


i4Trust IAM components

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



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

The screenshot shows the Keyrock Identity Manager web application running in a browser. The URL in the address bar is `https://accounts.opplafy.eu/idm/admins/list_users`. The browser status bar indicates "En pausa".

The interface includes a "Main menu" sidebar on the left with options: Home, Organizations, Applications, Notify, Administrators, and Users (which is selected). The main content area is titled "Users" and displays a list of users with the following columns: Id, Username, Email, and Enable.

	Id	Username	Email	Enable
<input type="checkbox"/>	1b49be31-ecd4-4a52-8a99-502ba8f24ecc	aarranz	aarranz@ficodes.com	<input checked="" type="button"/>
<input type="checkbox"/>	27334b89-c920-4a0d-bd0e-ff243c094984	customer9	customer9@ficodes.com	<input checked="" type="button"/>
<input type="checkbox"/>	317602b9-4389-4108-b10d-681b665c7576	customer5	customer5@ficodes.com	<input checked="" type="button"/>
<input type="checkbox"/>	33389f82-c91c-4761-8b7e-be338bd59421	customer10	customer10@ficodes.com	<input checked="" type="button"/>
<input type="checkbox"/>	3d864458-7241-45f1-a069-b98d8d54c2ac	customer6	customer6@ficodes.com	<input checked="" type="button"/>
<input type="checkbox"/>	43830f38-1f8a-404f-9d48-8d47d59e24f8	fdelavega	fdelavega@ficodes.com	<input checked="" type="button"/>
<input type="checkbox"/>	439b858f-8fa6-4c2c-bd10-e8f51577ca05	pablo	pablo@ficodes.com	<input checked="" type="button"/>
<input type="checkbox"/>	49ffa88-b1e7-4cb4-9ccb-8344c496b6a6	mabauer	martin.bauer@neclab.eu	<input checked="" type="button"/>

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-----

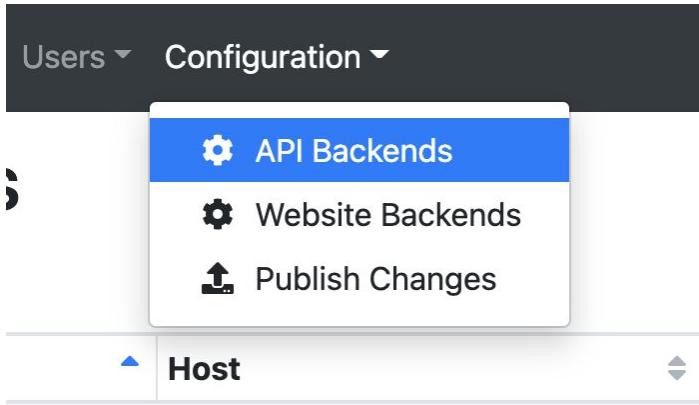
  private_key: |
    -----BEGIN PRIVATE KEY-----
    MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQC8yeRuxDI2/vyJ
    ...
    -----END PRIVATE KEY-----

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 supports 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.

The screenshot shows the API Umbrella configuration interface. At the top, there is a navigation bar with links for Analytics, Users, Configuration, and a gear icon. Below the navigation bar, the main content area is titled "Backend". It contains a section for "Backend Protocol" with a dropdown menu set to "http". Under "Server", the URL "http://orion.local:1026" is listed with edit and remove buttons. A "Add Server" button is also present. The next section is "Host", which defines the host for listening and the host the API backend is listening on. It includes "Frontend Host" (context.opplafy.eu) and "Backend Host" (context.opplafy.eu), with a "rewrite to" arrow between them. The final section is "Matching URL Prefixes", which asks what URL prefixes should be routed to this backend. It shows a table with one row where both "Frontend Prefix" and "Backend Prefix" are "/". An "Add URL Prefix" button is at the bottom of this section.

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.

The screenshot shows the 'Global Request Settings' section of the API Umbrella interface. It includes fields for 'Append Query String Parameters' (param1=value¶m2=value), 'Set Request Headers' (Authorization: None), 'HTTP Basic Authentication' (username:password), 'IDP Mode' (Default: Authentication), 'IDP App ID' (externalid1234), 'Required Headers' (X-Example-Header: value), 'HTTPS Requirements' (Inherit (default - required)), 'API Key Checks' (Inherit (default - required)), 'API Key Verification Requirements' (Inherit (default - none)), and 'Authorization Mode' (Context Broker attribute based - iSHARE compliant (automatically)).

Setting	Value
Append Query String Parameters	param1=value¶m2=value
Set Request Headers	Authorization: None
HTTP Basic Authentication	username:password
IDP Mode	Default: Authentication
IDP App ID	externalid1234
Required Headers	X-Example-Header: value
HTTPS Requirements	Inherit (default - required)
API Key Checks	Inherit (default - required)
API Key Verification Requirements	Inherit (default - none)
Authorization Mode	Context Broker attribute based - iSHARE compliant (automatically)

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 will 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.

```
< Location: https://marketplace.i4trust.fiware.io/openid_connect1.0/return?  
code=Dmn-TbSj70cK15ym1j5xZsgkabzVP8dMugC81nzmeW4&  
state=ZqVQm4zHaEDyBhzpm1ZRH7fsxy703lq2
```

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

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...YkNK0Q&  
redirect_uri=https://marketplace.i4trust.fiware.io/openid_connect1.0/return&  
code=Dmn-TbSj70cK15ym1j5xZsgkabzVP8dMugC81nzmeW4
```

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:https://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!

