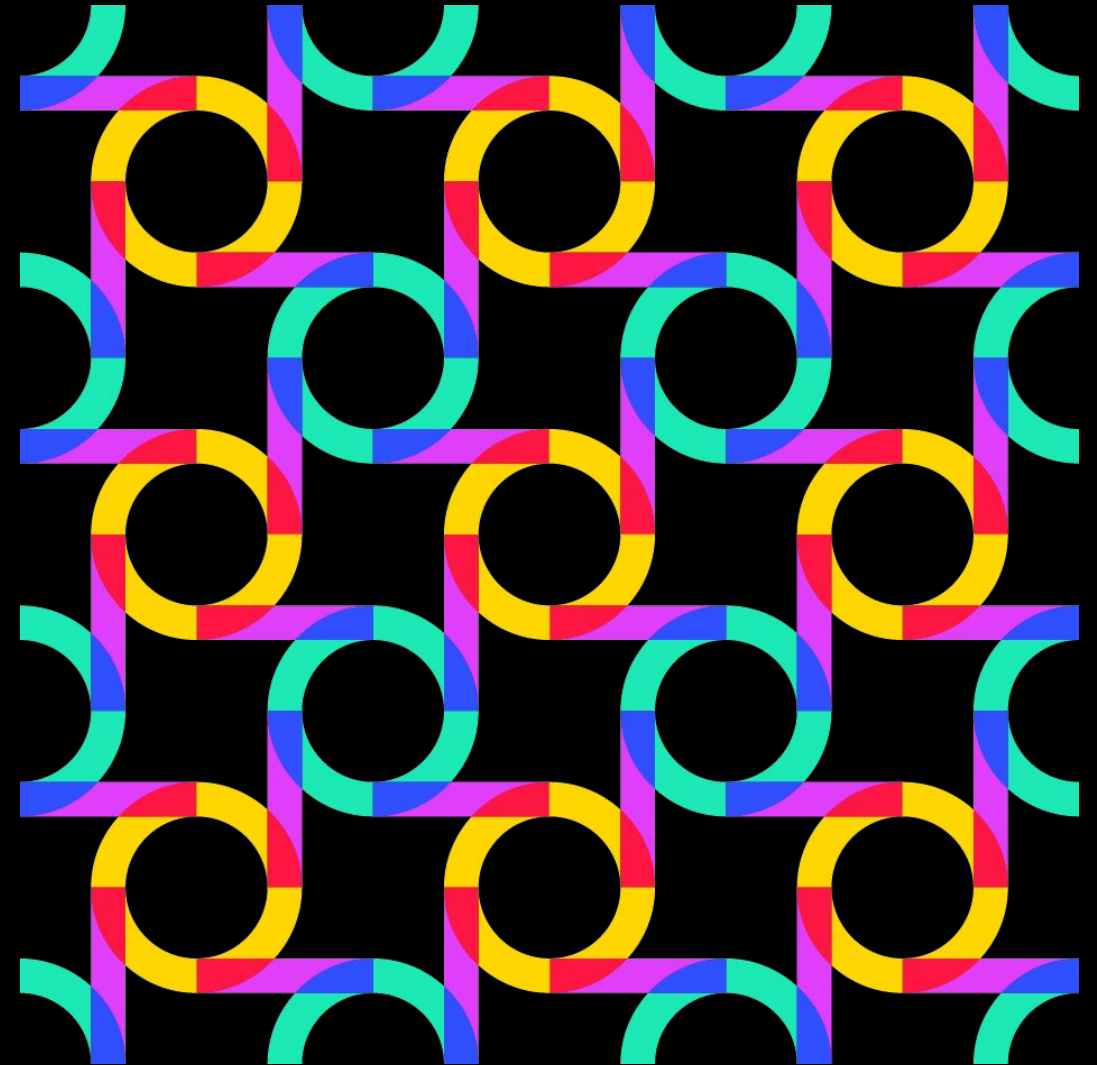


# Bringing the pieces together / Detailed walkthrough of the reference example

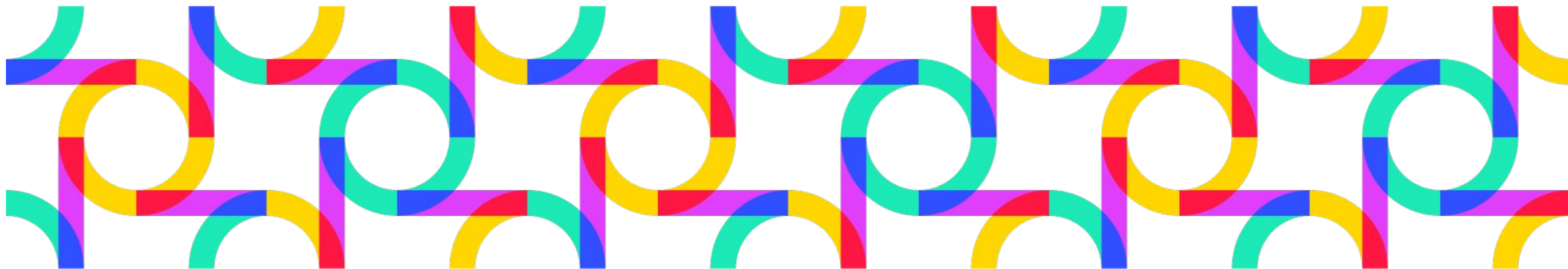
Dr. Dennis Wendland - Technical Lead & Architect  
FIWARE Foundation - [dennis.wendland@fiware.org](mailto:dennis.wendland@fiware.org)

Stefan Wiedemann - Technical Lead & Architect  
FIWARE Foundation - [stefan.wiedemann@fiware.org](mailto:stefan.wiedemann@fiware.org)



# Outline

1. Example overview
2. Prerequisites
3. Demo (OIDC/iSHARE)
4. What happened in the background (OIDC/iSHARE)
5. Demo (OIDC4VP/SIOP2)
6. What happened in the background (OIDC4VP/SIOP2)
7. Setup of components on K8s



## Example overview

# Example Overview

## Showcase:

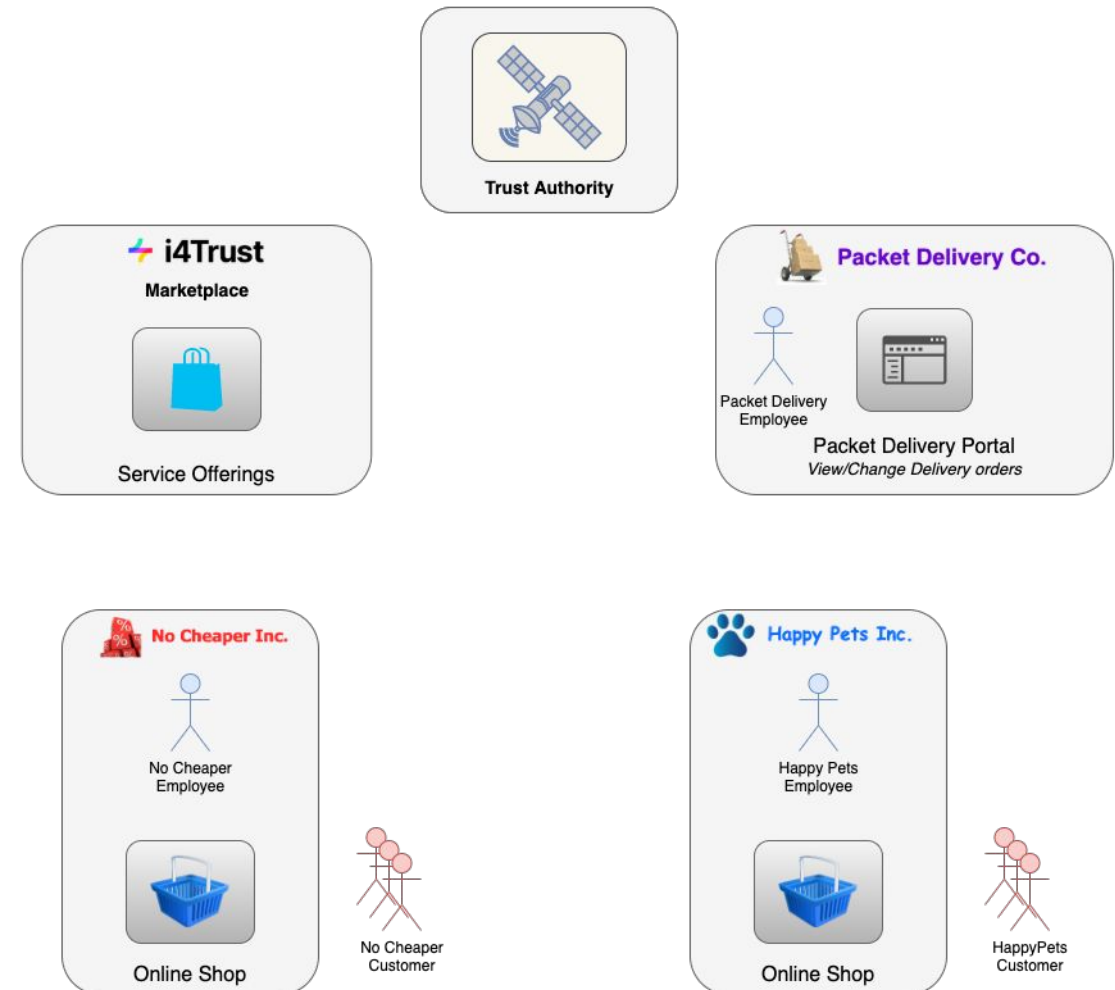
- Delegation of data service access rights within trusted data space
- Delegated from an organizational level to an user level
- Service provider does not need knowledge about users which finally access the service

## Parties:

- **Packet Delivery Co.** offers delivery order service on **i4Trust Marketplace**
- Different **retailers** (shops) acquire access to service offering
- Retailers delegate service access to their **customers**
- **Trust Authority** ensures trust among different organisations

## Steps:

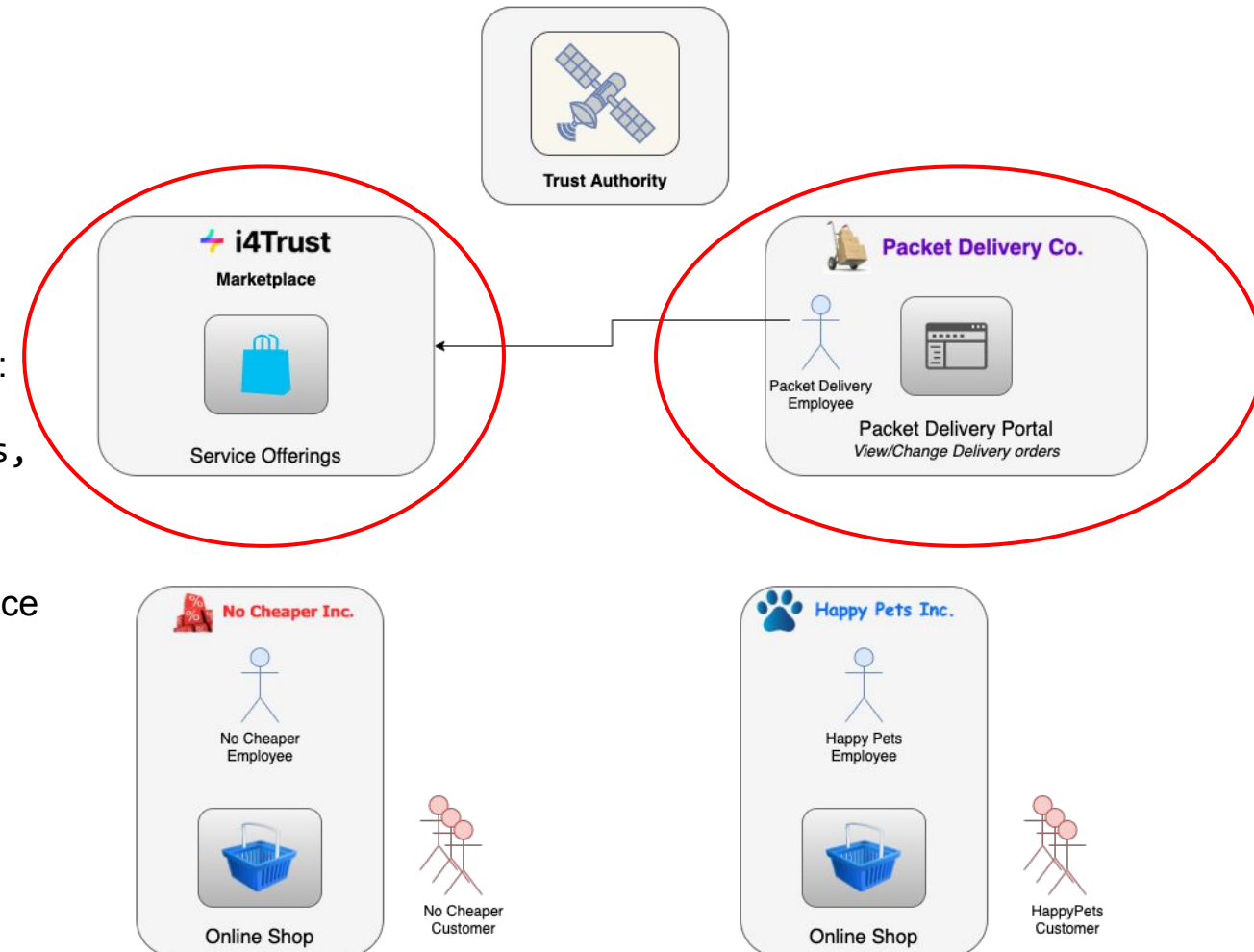
1. Creating an offering (by Packet Delivery Co.)
2. Acquiring an offering (by retailers)
3. Accessing packet delivery order service (by retailer customers)



# Example Overview

## Packet Delivery Company + Marketplace

- Packet Delivery Co. provides digital service about its packet delivery orders
- Delivery orders map to Digital Twin entities characterized by
  - issuer, destinee
  - originAddress, deliveryAddress
  - pda/pta (planned date/time of arrival)
  - eda/eta (estimated date/time of arrival)
- Different service levels offered for users accessing the service:
  - **Standard:** **Read** access to delivery orders (**HTTP GET**)
  - **Gold:** Allowed to **change** destinee, deliveryAddress, pta, pda of delivery orders (**HTTP PATCH**)
- Packet Delivery Co. publishes its offering on i4Trust Marketplace for acquisition by organisations (e.g. retailers/shops)
  - **Basic:** Organisations can only offer **PacketDelivery:Standard** service level to its customers
  - **Premium:** Organisations can assign **PacketDelivery:Standard** and **PacketDelivery:Gold** service level to its customers



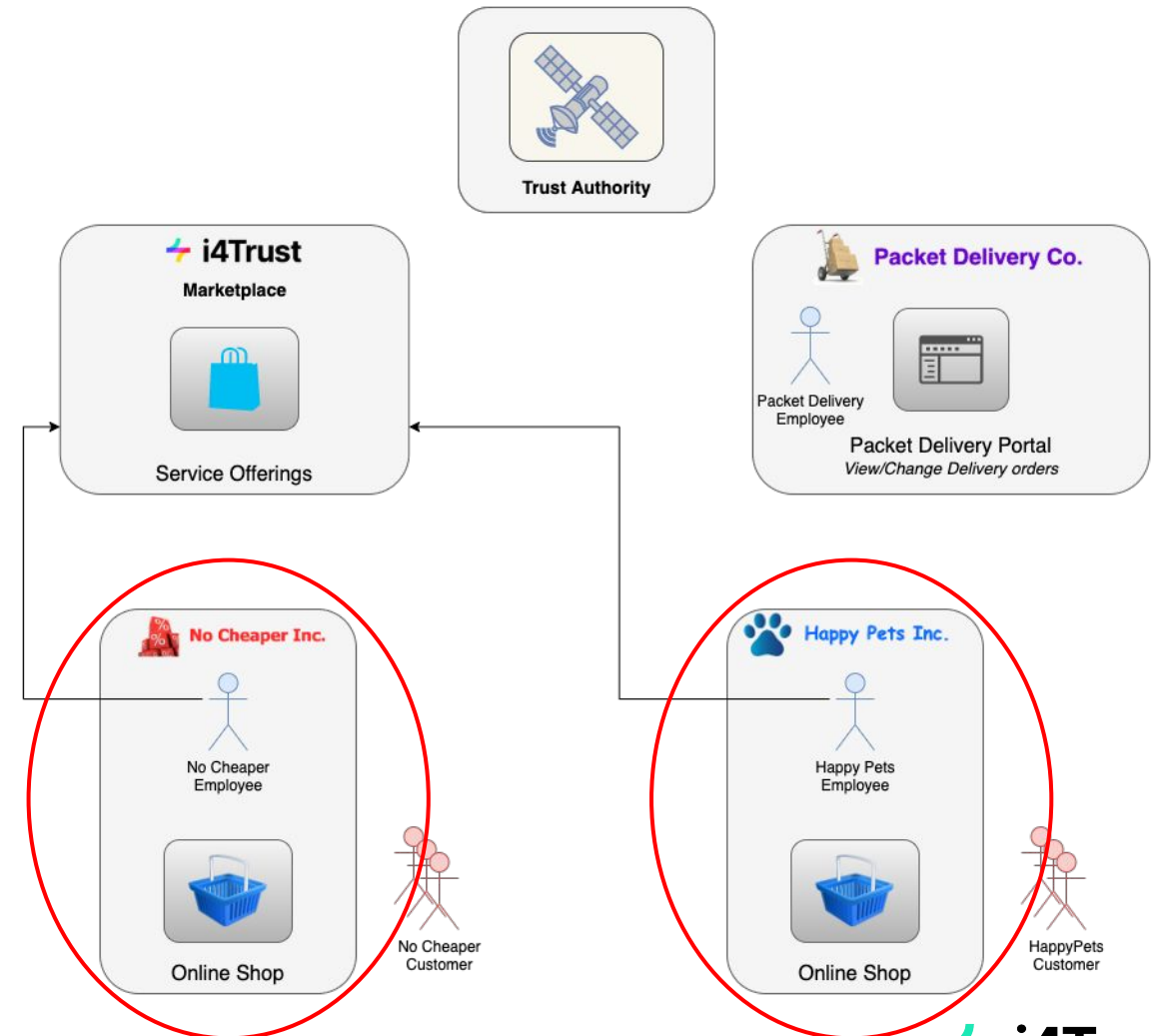
# Example Overview Shops

## Happy Pets Inc.:

- Shop of premium products of pets
- Acquires Premium packet delivery offering on Marketplace
  - Can offer Standard and Gold service to its customers

## No Cheaper Inc.:

- Shop of bargain clothes at big discounts
- Acquires Basic packet delivery offering on Marketplace
  - Can offer only Standard service to its customers

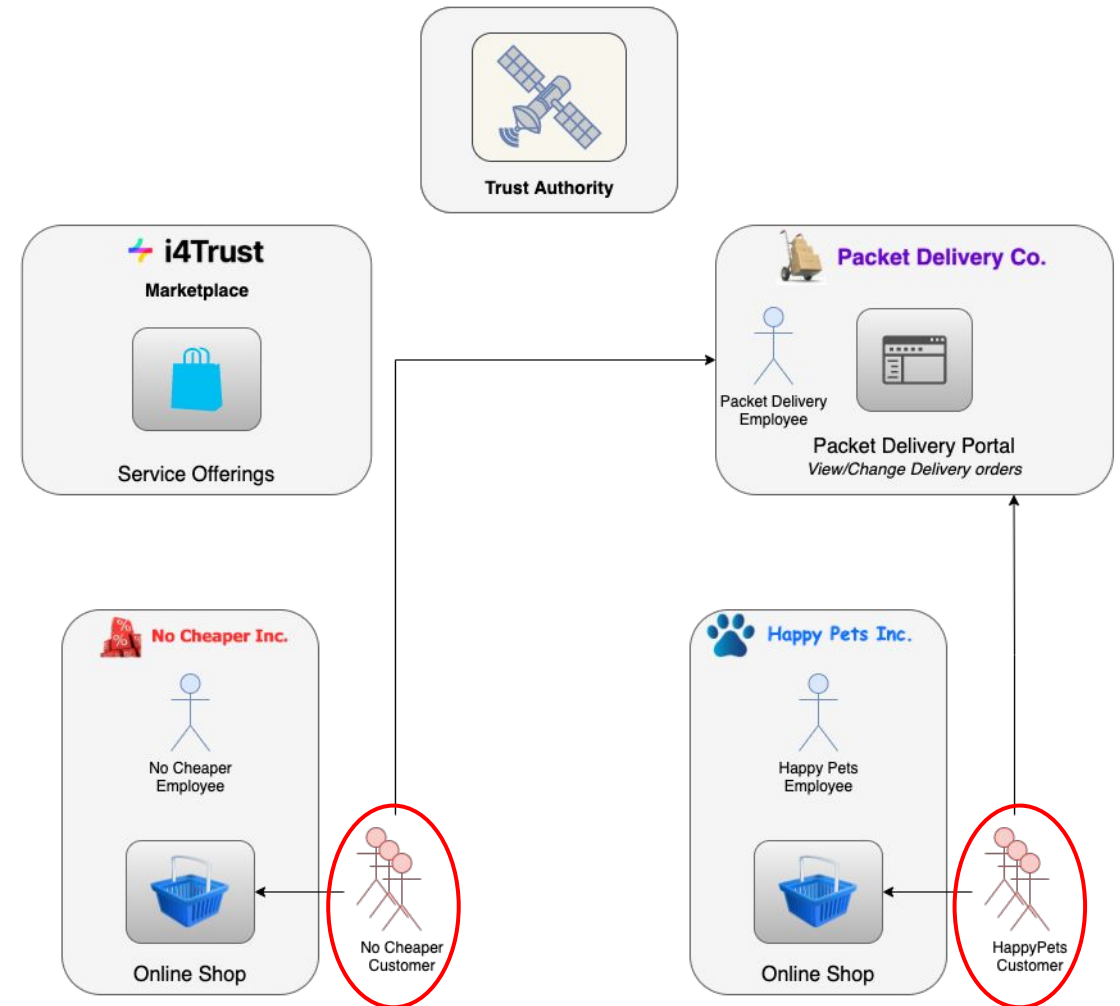


# Example Overview

## Customers

### Customers:

- Both shops have (human) customers
- After checkout at the shop, delivery orders at Packet Delivery Co. are generated
- Customers might want to view and/or change their delivery orders at the Packet Delivery Co. portal
  - Change of delivery orders only allowed for customers of Happy Pets Inc.





## Prerequisites



# Prerequisites

## Architectural overview

### Marketplace

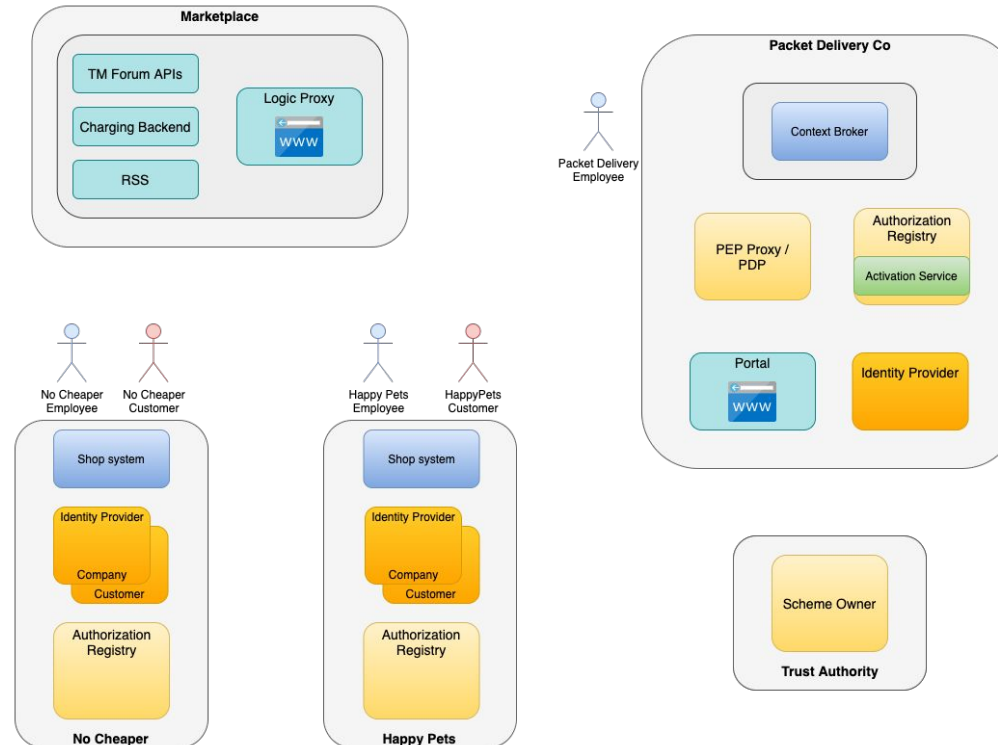
- Business API Ecosystem

### Happy Pets Inc.

- Identity Provider(s)
- Authorisation Registry (AR)
- Shop system application

### No Cheaper Inc.

- Identity Provider(s)
- Authorisation Registry (AR)
- Shop system application



### Packet Delivery Co.

- Context Broker
- Identity Provider
- PEP Proxy/PDP
- Authorisation Registry (AR) + Activation Service (AS)
- Portal application

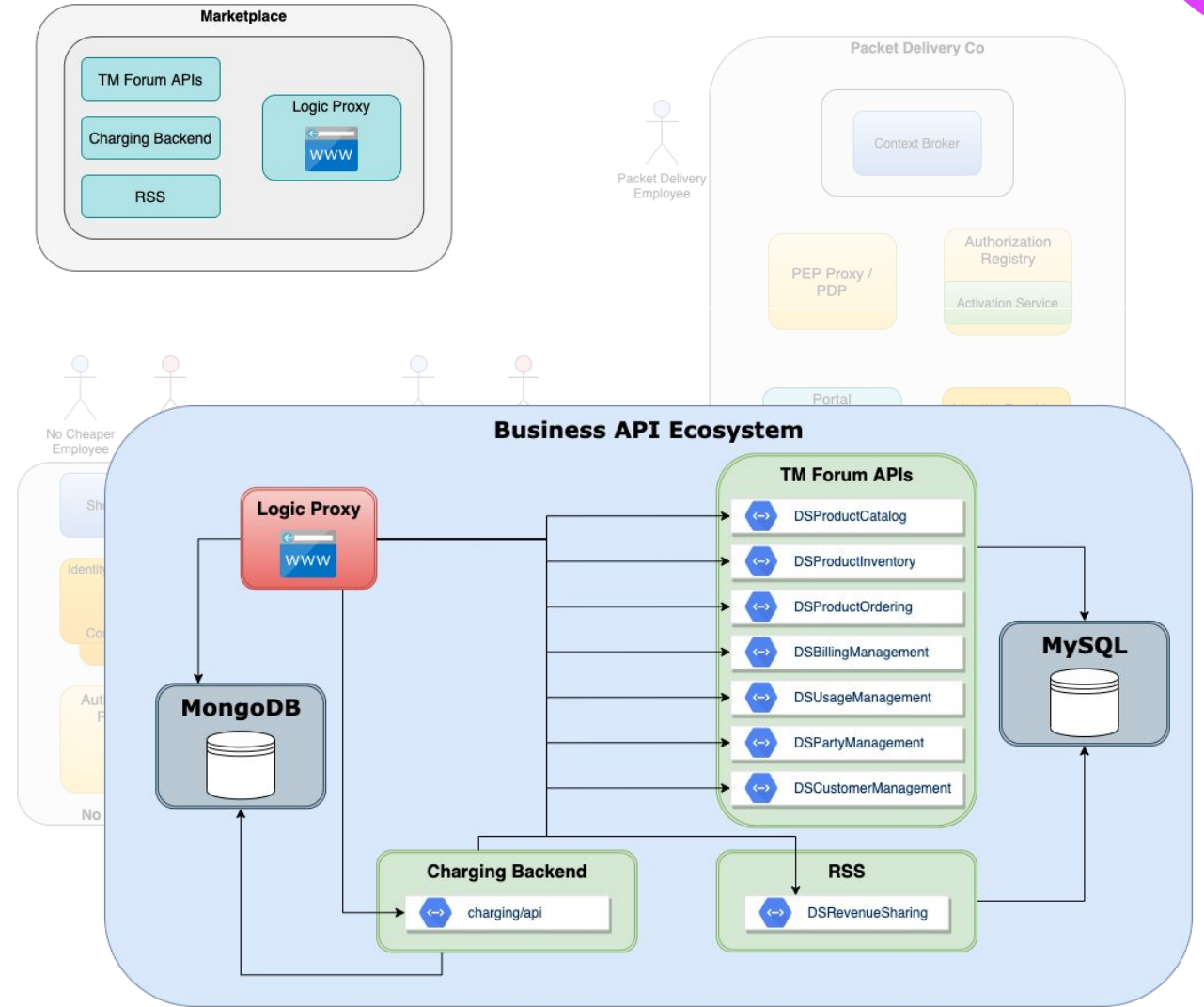
### Trust Authority

- iSHARE Satellite

# Prerequisites Marketplace

- Databases used by different marketplace components
  - MongoDB
  - MySQL
- TM Forum APIs
  - Reference implementation
- Charging Backend
  - Rating, charging and billing
- RSS
  - Revenue settlement and sharing
- Logic Proxy
  - UI portal, authentication and API orchestrator

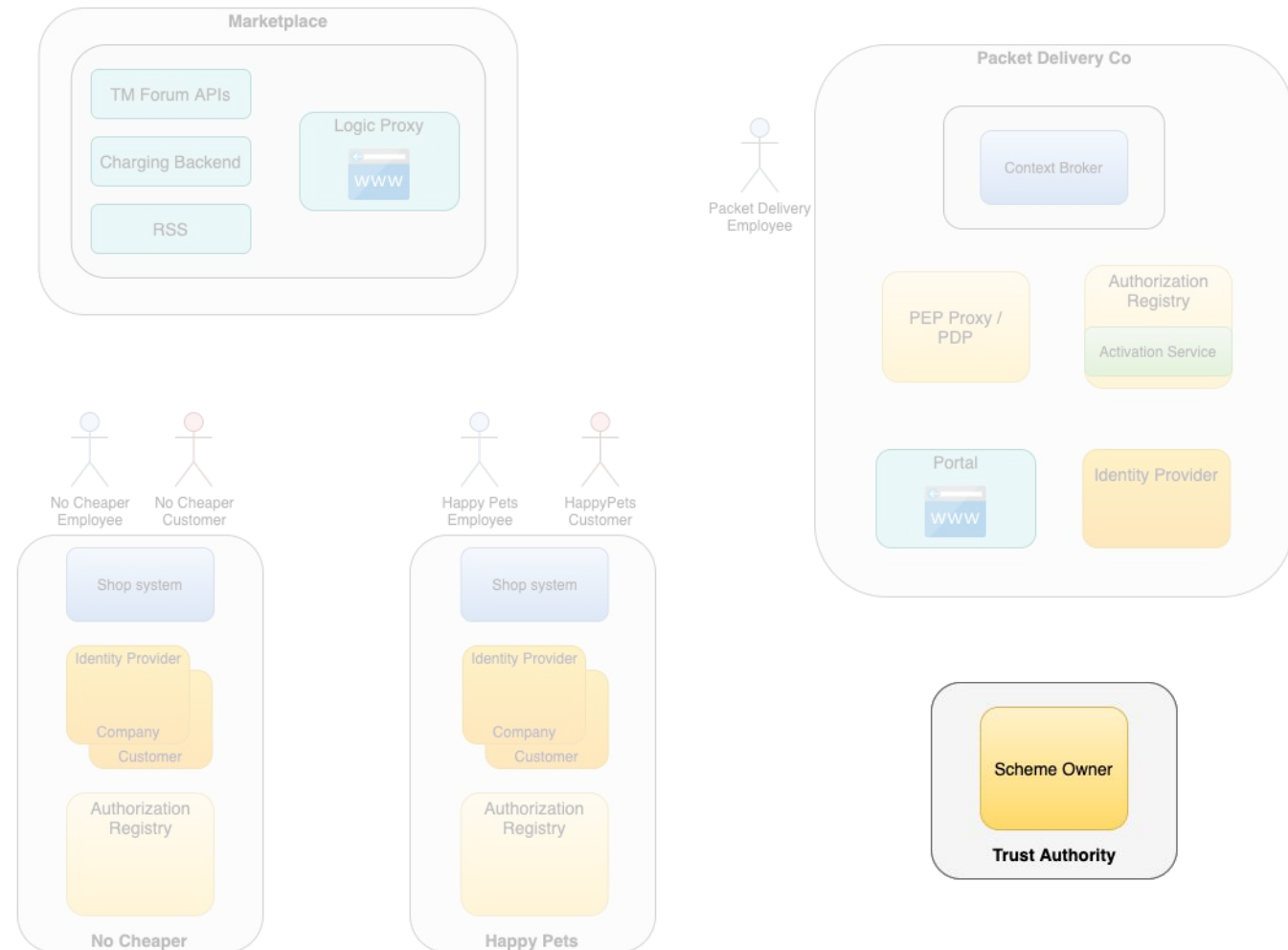
No further prerequisites needed



# Prerequisites

## Trust Authority

- Installed per data space
  - Using iSHARE Satellite
    - <https://scheme.isharetest.net/swagger/index.html>
- Each participating organisation, procures certificates/private keys and EORI IDs for registration
  - Marketplace: EU.EORI.NLMARKETPLA
  - Packet Delivery: EU.EORI.PACKETDEL
  - Happy Pets: EU.EORI.NLHAPPYPETS
  - No Cheaper: EU.EORI.NLNOCHEAPER
  - iSHARE Satellite: EU.EORI.NL000000000
- This information must be known to trust authority
  - Allows all participants to verify status of each organisation



# Prerequisites

## Shop: Happy Pets Inc.

### 2 Identity providers to be deployed with pre-registered users

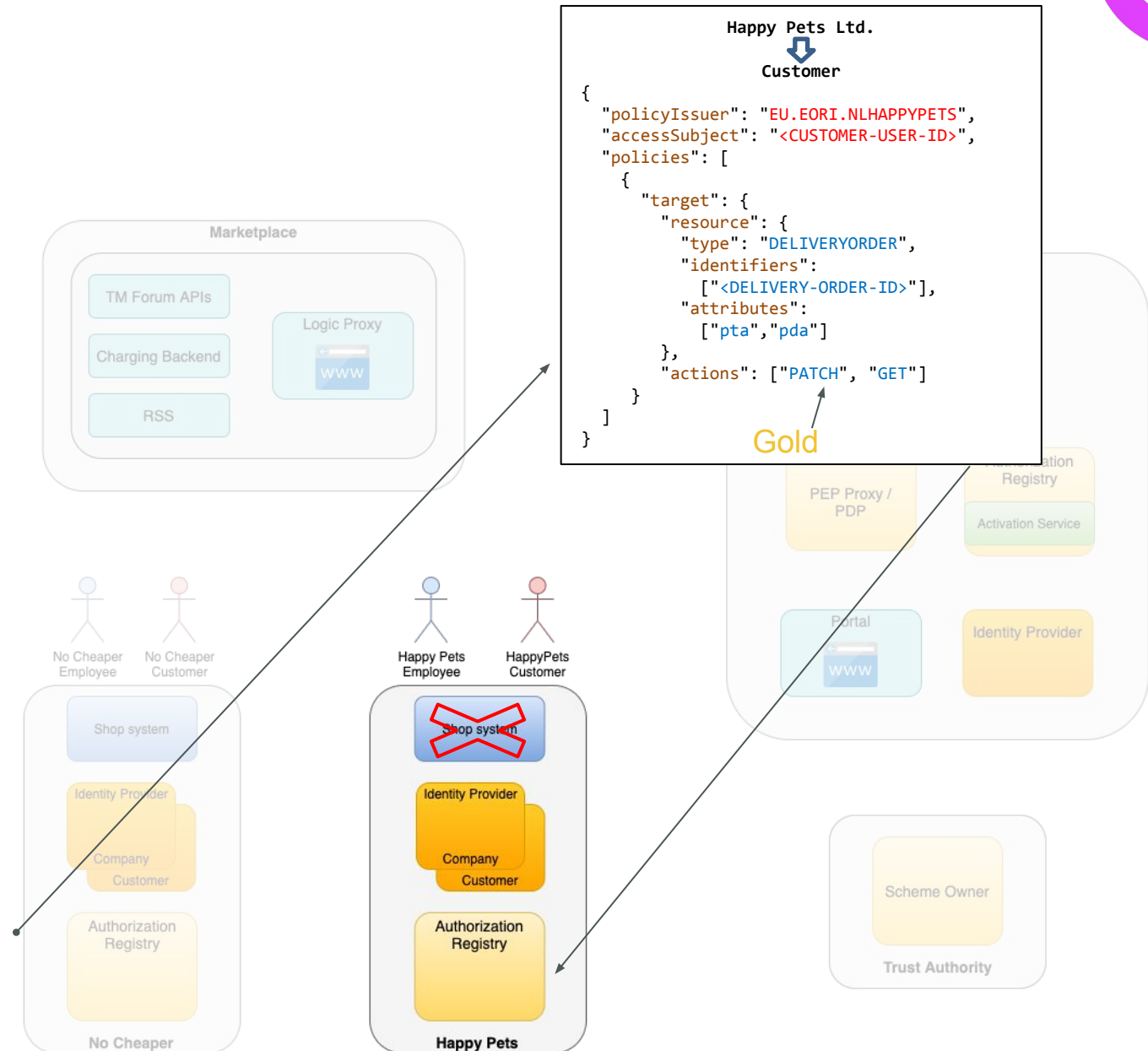
- Company (login @ marketplace)
  - Employee user
- Customer (login @ shop and packet delivery portal)
  - Standard customer user
  - Gold customer user

### Authorisation Registry instance

- Holds customer access policies
- Using iSHARE AR test instance:  
<https://ar.isharetest.net/swagger/index.html>

### Shop system

- Skipped for simplicity
- In the following assume:
  - Customer users already registered
  - Pre-existing delivery orders for each customer
  - Standard/Gold policies already assigned to customers
- Initial delivery orders to be created manually (next slides)
- Customer policies (Standard/Gold) to be created manually



# Prerequisites

## Shop: No Cheaper Inc.

### 2 Identity providers to be deployed with pre-registered users

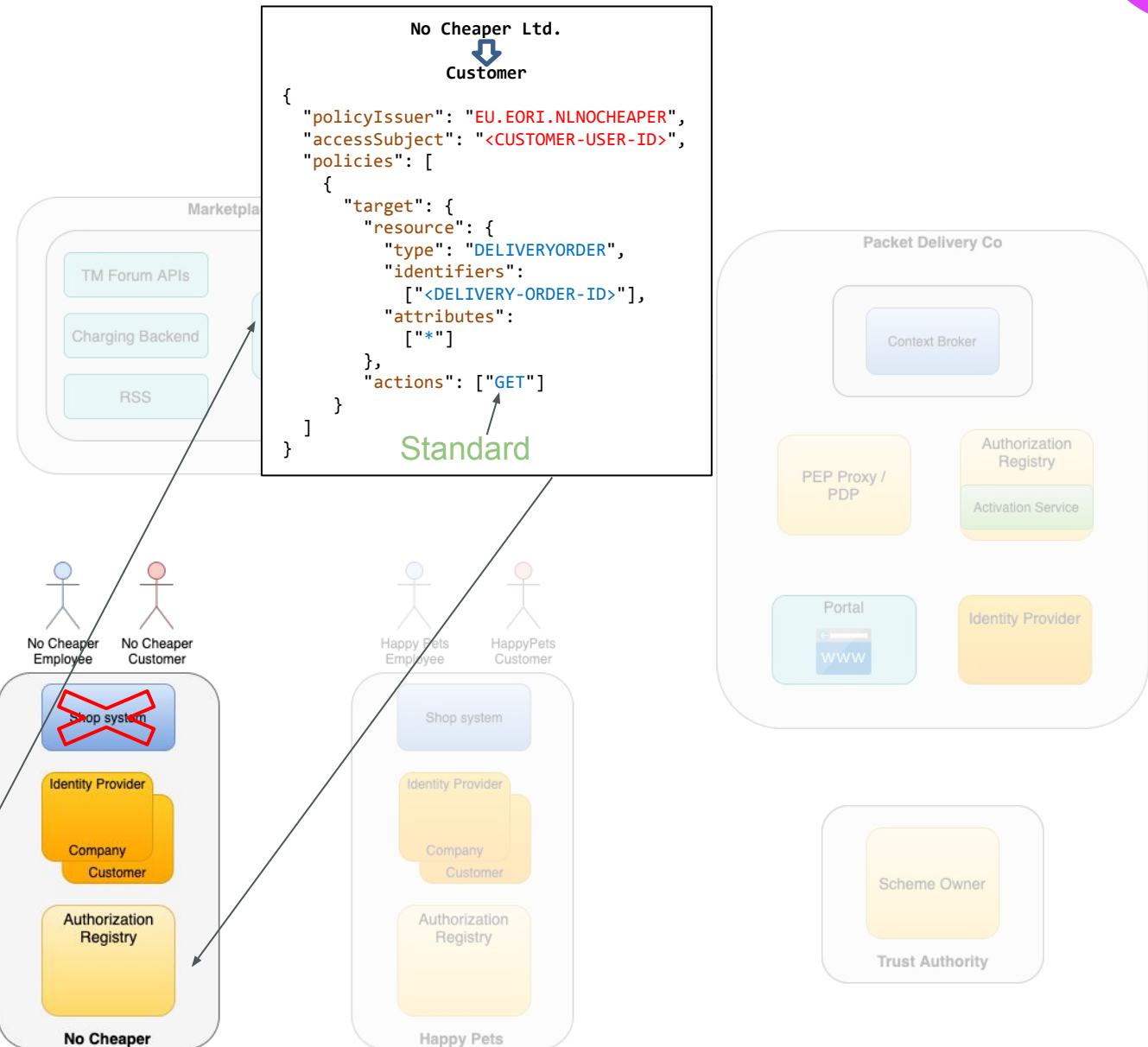
- Company (login @ marketplace)
  - Employee user
- Customer (login @ shop and packet delivery portal)
  - Standard customer user
  - Gold customer user could also be created to demonstrate denied access with PATCH request

### Authorisation Registry instance

- Holds customer access policies
- Using iSHARE AR test instance:  
<https://ar.isharetest.net/swagger/index.html>

### Shop system

- Skipped for simplicity
- In the following assume:
  - Customer users already registered
  - Pre-existing delivery orders for each customer
  - Standard policies already assigned to customers
- Initial delivery orders to be created manually (next slides)
- Customer policies (Standard) to be created manually



# Prerequisites

## Packet Delivery Co.

### Packet Delivery Portal application

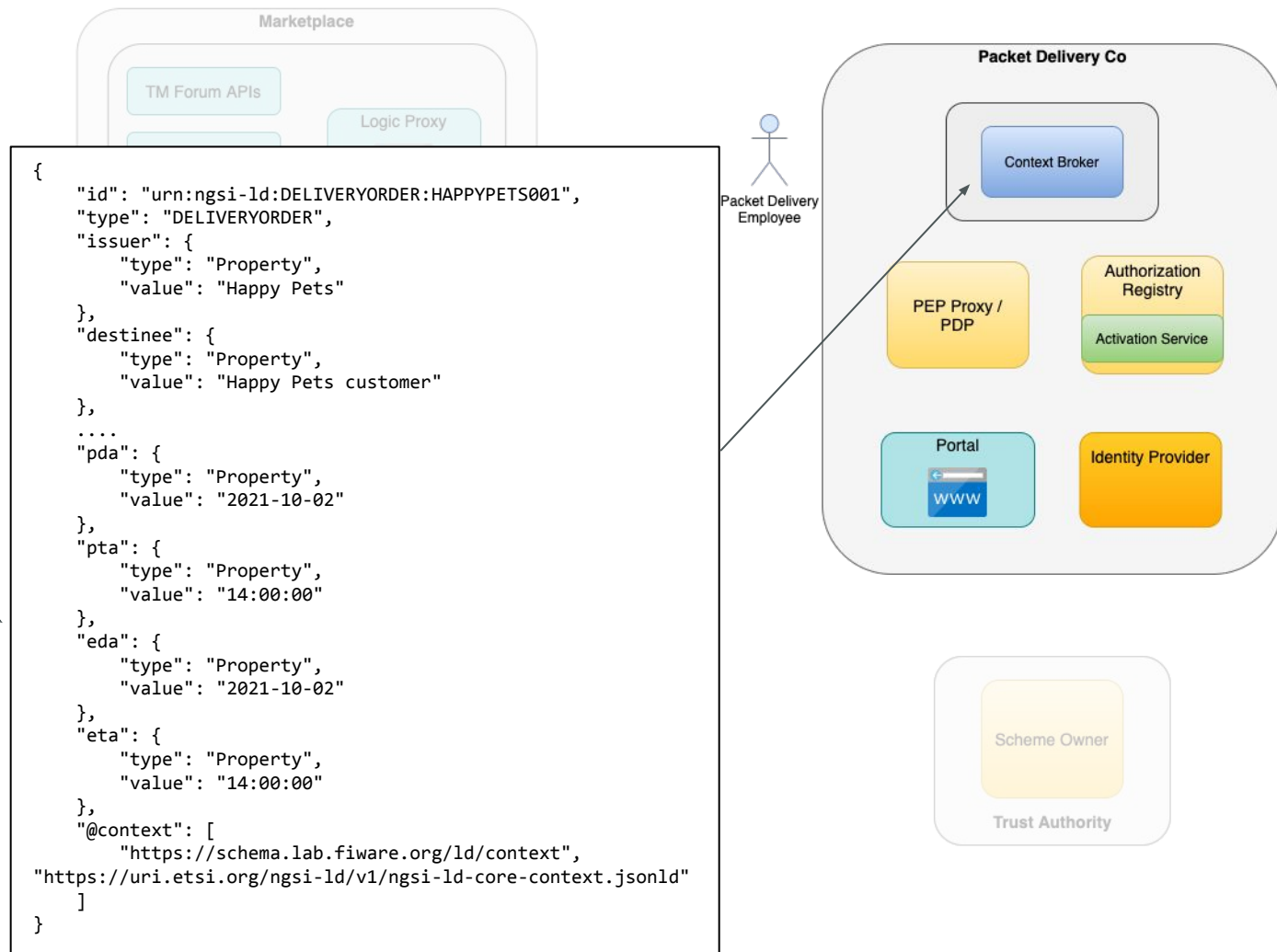
- Demo application available

### Identity Provider to be deployed with pre-registered user

- Employee user

### Context Broker as Service provider

- Provides NGSI-LD API for
  - Reading delivery order entities (HTTP GET)
  - Changing delivery order entities (HTTP PATCH)
- Initial delivery order entities to be created manually



# Prerequisites

## Packet Delivery Co.

### PEP Proxy / PDP

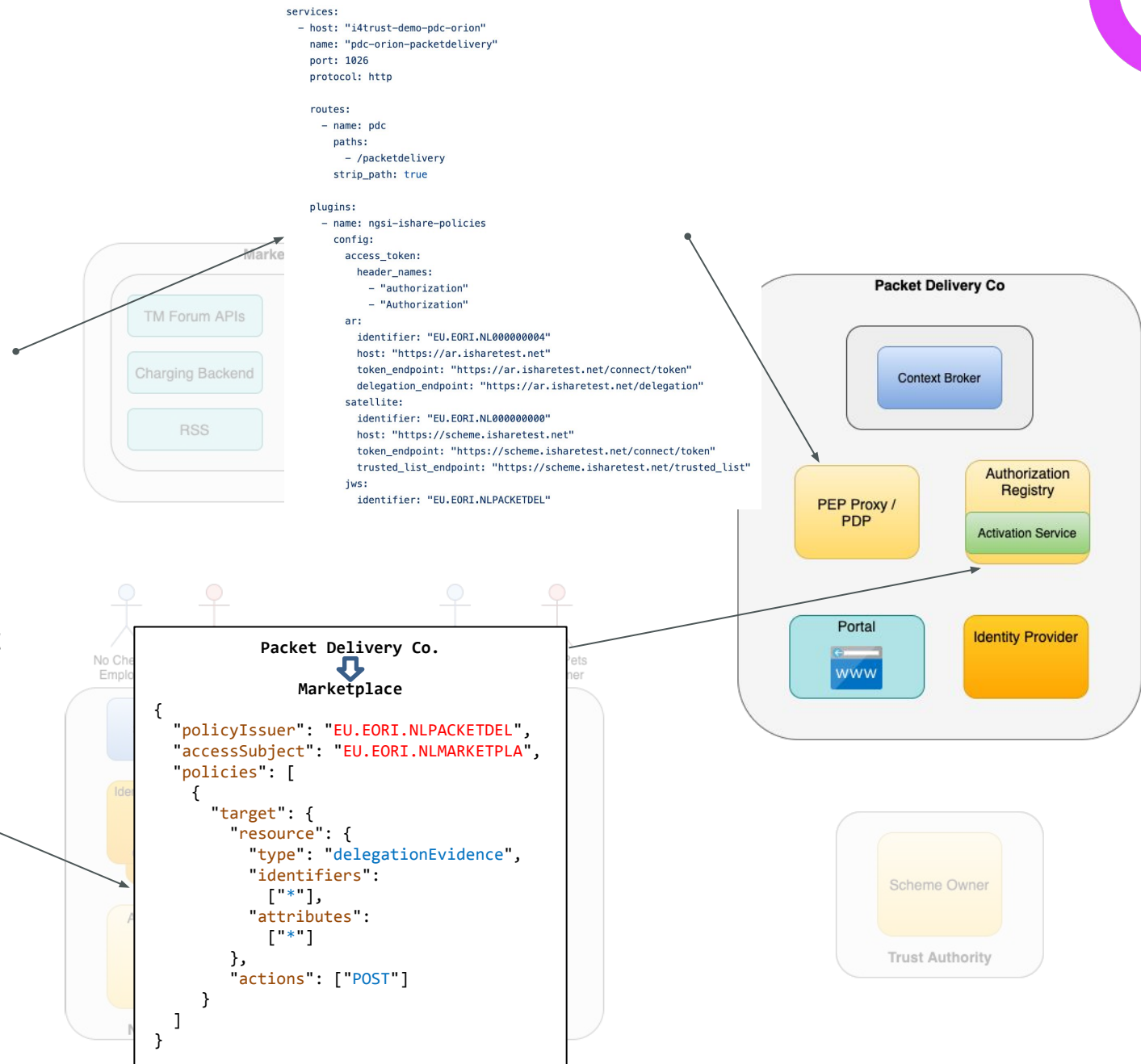
- Using Kong with NGSI/iSHARE plugin
- Kong service (Context Broker) configured with plugin ngsi-ishare-policies

### Authorisation Registry instance

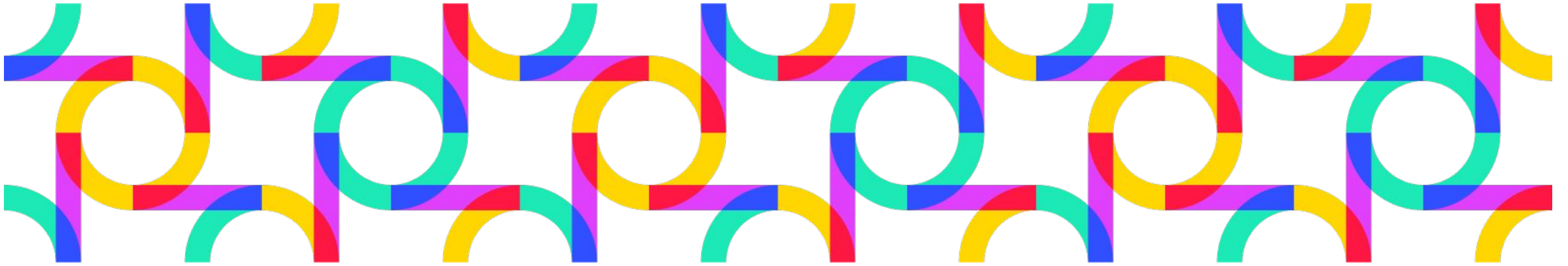
- Holds organisational access policies
- Using central iSHARE AR test instance

### Activation Service

- Demo application available
- Required for enabling Marketplace to create policies at AR of Packet Delivery
- Marketplace policy to be created initially



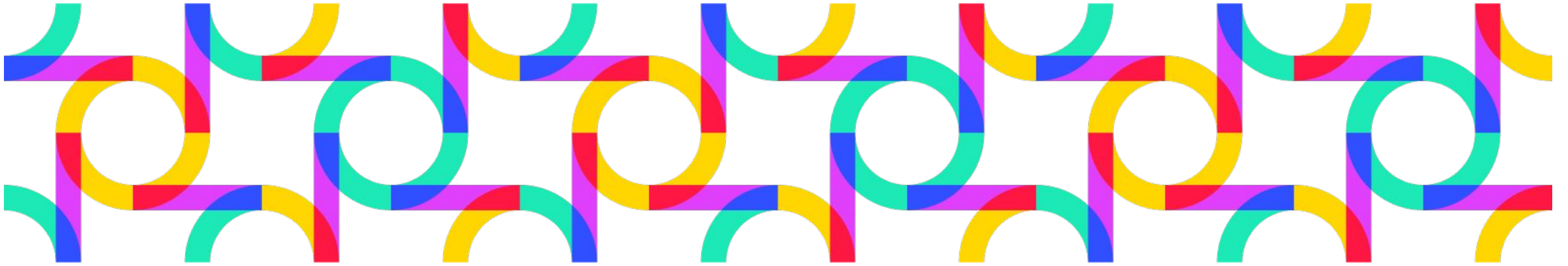




## Demo (OIDC/iSHARE flow)

- Create an offering (not shown)
- Acquire access to an offering
- Access the service



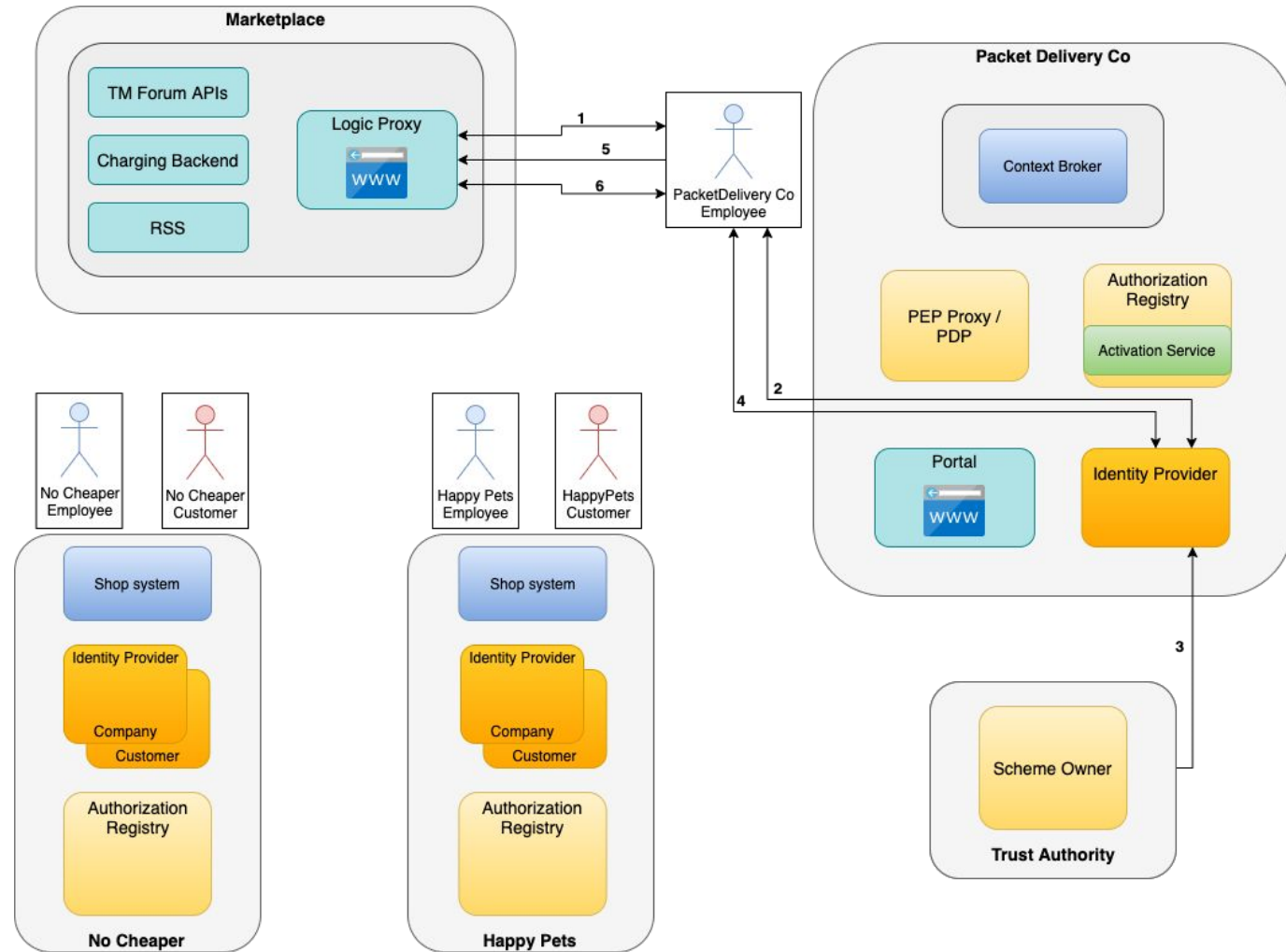


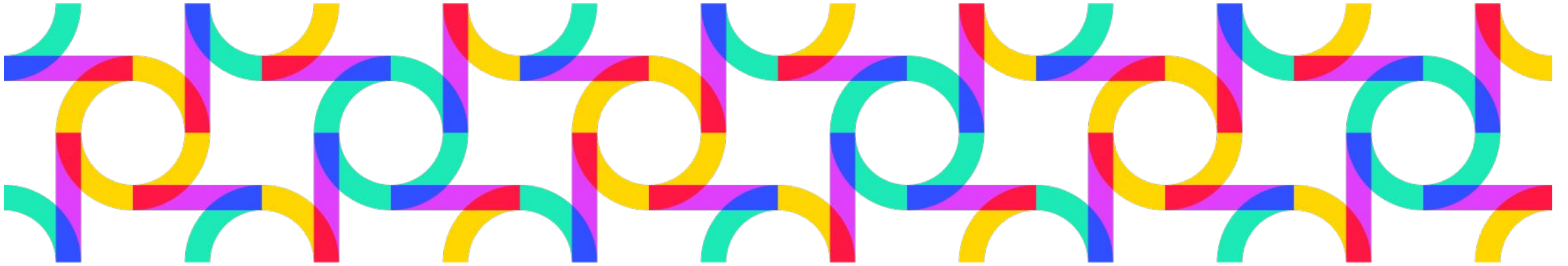
What happened in the background  
Creating an offering

# Creating an offering

## Employee of Packet Delivery Co. creates offering on Marketplace for its basic and premium services

- Employee requests login on Marketplace based on OIDC using its own IDP (1-2)
- IDP validates against Trust Authority whether Marketplace is a trusted party (3)
- Employee performs login based on OIDC standard (4-5)
- Employee has logged in and creates offerings for standard and premium services (6)  
Provides information to offering description about:
  - Service endpoint (+ API specification)
  - Data model used
  - Security requirements (e.g. policies supported)
  - Other terms and Conditions, pricing



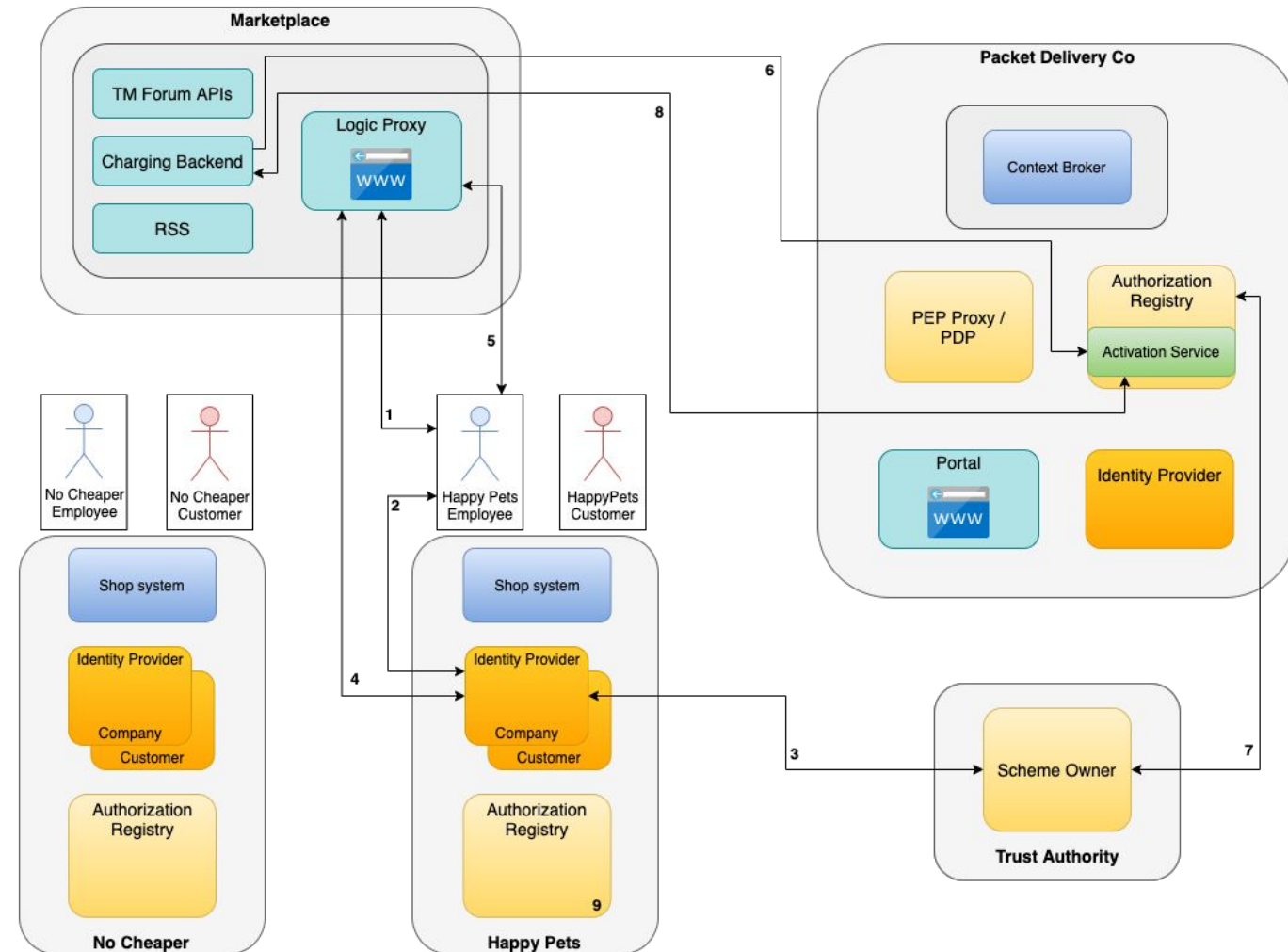


What happened in the background  
Acquiring access rights and activating the service

# Acquisition of offerings

## Employee of Happy Pets Ltd acquires access to premium service offering

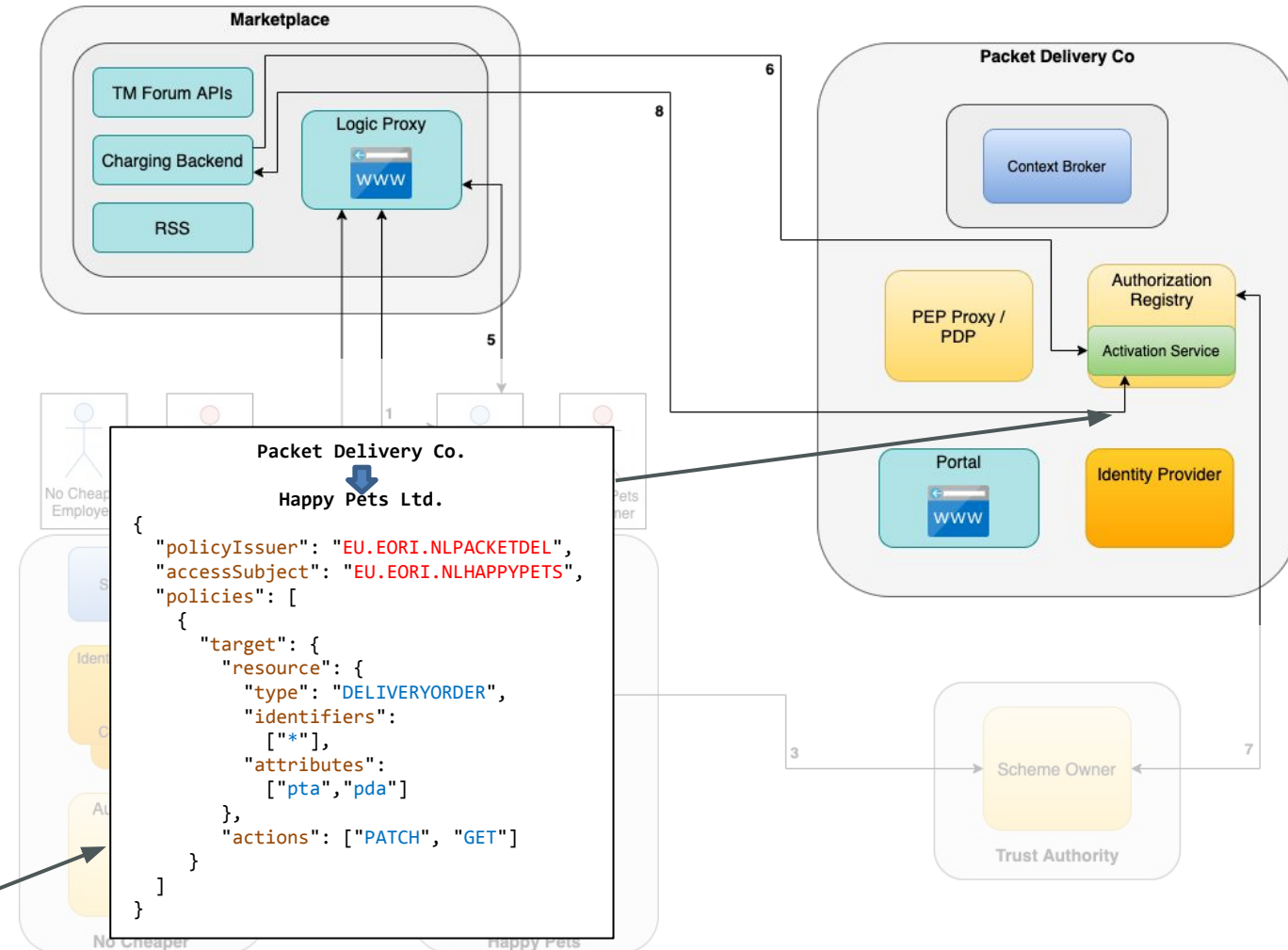
- Employee requests login on Marketplace based on OIDC using its own IDP (1-2)
- IDP validates Marketplace as trusted party, login is performed based on OIDC and Employee is presented available offerings (3-4)
- Employee selects the Packet Delivery **premium service** and performs the checkout process incl. providing payment details (5)
- Marketplace requests at Packet Delivery AR **to register HappyPets as organization able to assign Packet Delivery premium service policies to its customers** (6)
  - Authentication at AR based on OAuth2
- AR validates Marketplace as trusted party (7)
- Policy is created at AR, allowing Happy Pets to delegate premium service access to its customers (8)
- Policy allowing premium access needs to be assigned at Happy Pets AR to its customers (9)



# Acquisition of offerings

## Employee of Happy Pets Ltd acquires access to premium service offering

- Employee requests login on Marketplace using its own IDP (1-2)
- IDP validates Marketplace as trusted party, login is performed based on OIDC and Employee is presented available offerings (3-4)
- Employee selects the Packet Delivery premium service and performs the checkout process incl. providing payment details (5)
- Marketplace requests at Packet Delivery AR to register HappyPets as organization able to assign Packet Delivery premium service policies to its customers (6)
- AR validates Marketplace as trusted party (7)
- Policy is created at AR, allowing Happy Pets to delegate premium service access to its customers (8)
- Policy allowing premium access needs to be assigned at Happy Pets AR to its customers (9)

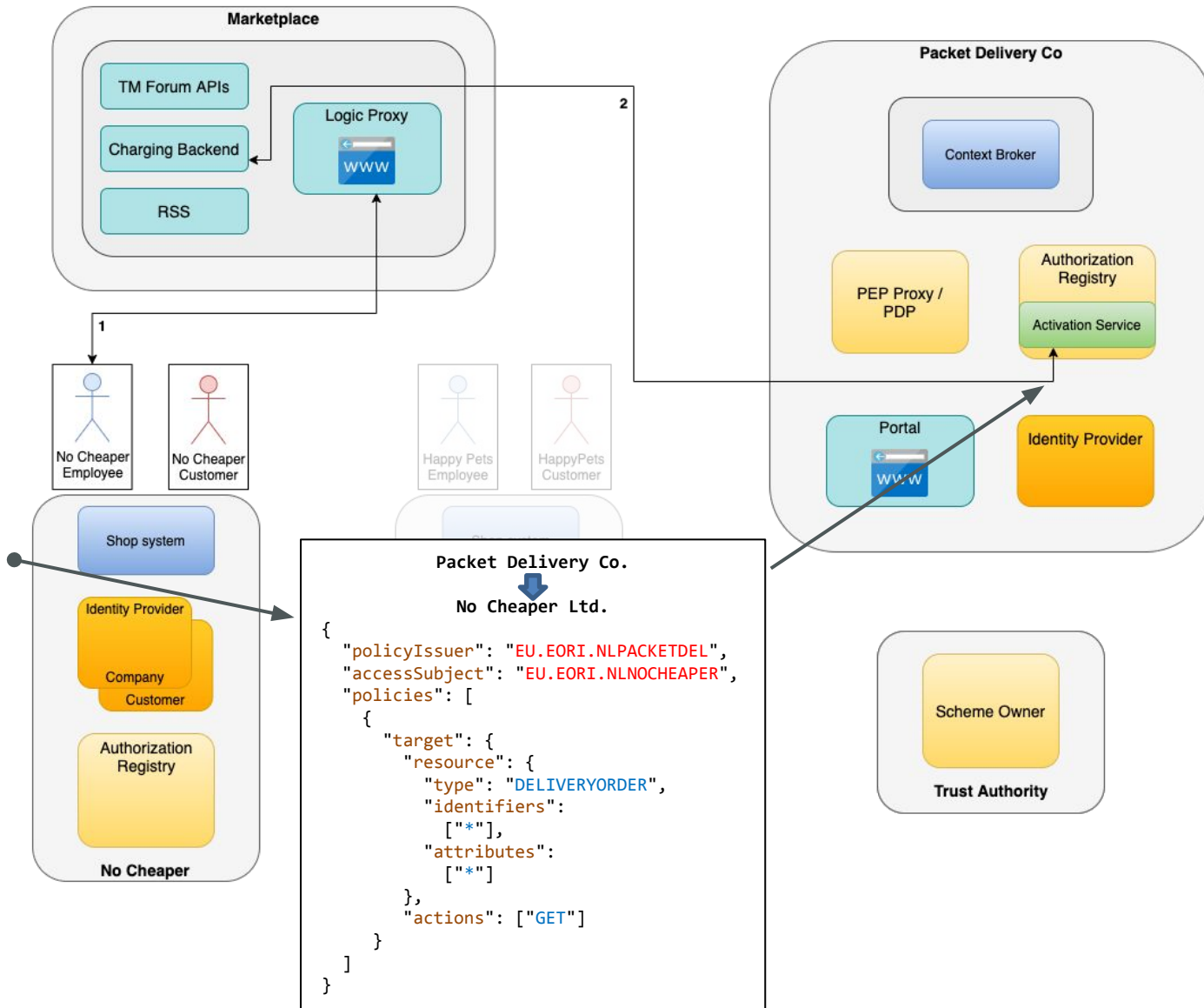


# Acquisition of offerings

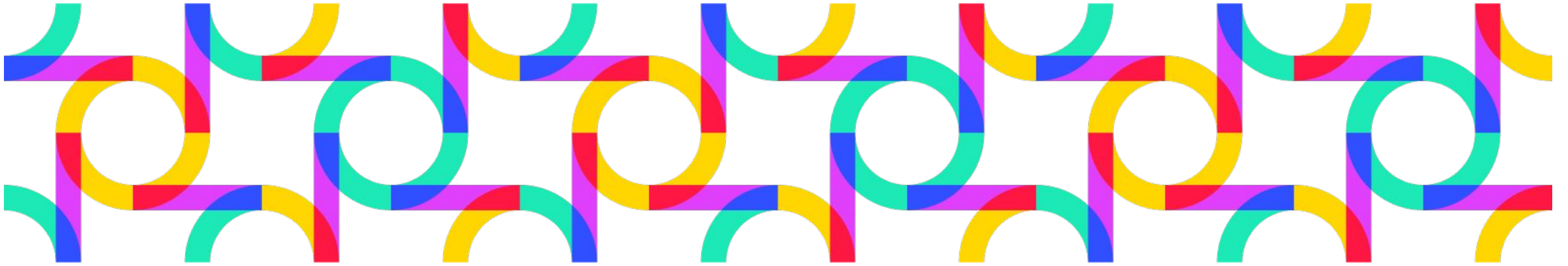
## Scenario: No Cheaper

### Employee of No Cheaper Ltd acquires access to basic service offering

- Employee signs in at marketplace and selects Packet Delivery **Basic Service** offering (1)
- Marketplace requests at Packet Delivery AR **to register No Cheaper as organization able to assign Packet Delivery basic service policies to its customers** (2)
  - AR validates Marketplace as trusted party
  - Policy is created at AR, allowing Happy Pets to delegate premium service access to its customers
- Policy allowing basic access needs to be assigned at No Cheaper AR to its customers







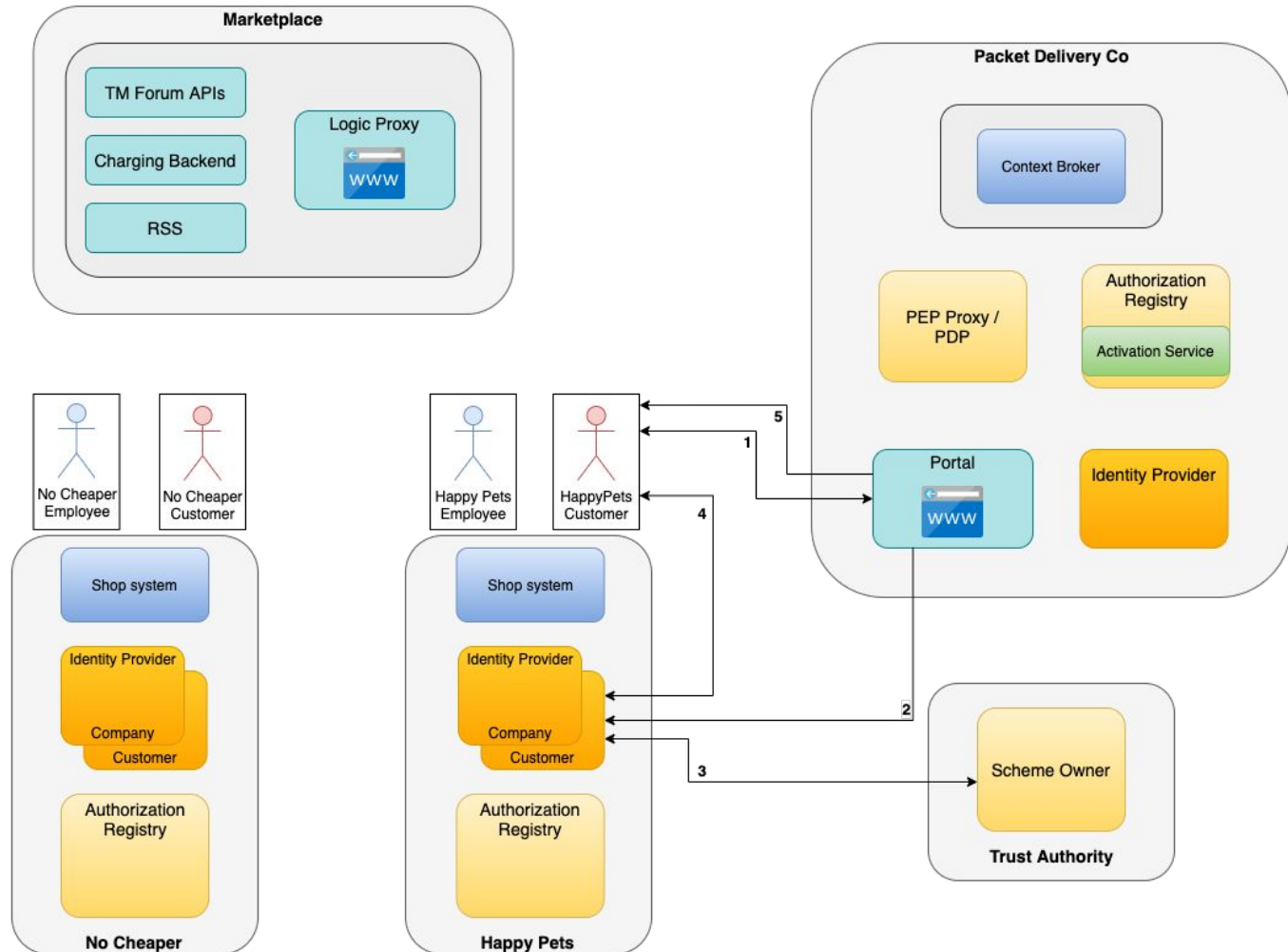
What happened in the background  
Accessing the service

# Usage of data service

## Login at portal

### Customer of Happy Pets Ltd. Performs login at Packet Delivery portal

- Customer requests login at portal using Happy Pets IDP (1-2)
- IDP validates against Trust Authority whether portal is a trusted party (3)
- Login is performed based on OIDC standard, the issued signed JWT contains info about Happy Pets AR where to retrieve user policies from (4)
  - JWT could also directly contain user policies
- Customer is presented details about its delivery order (5)



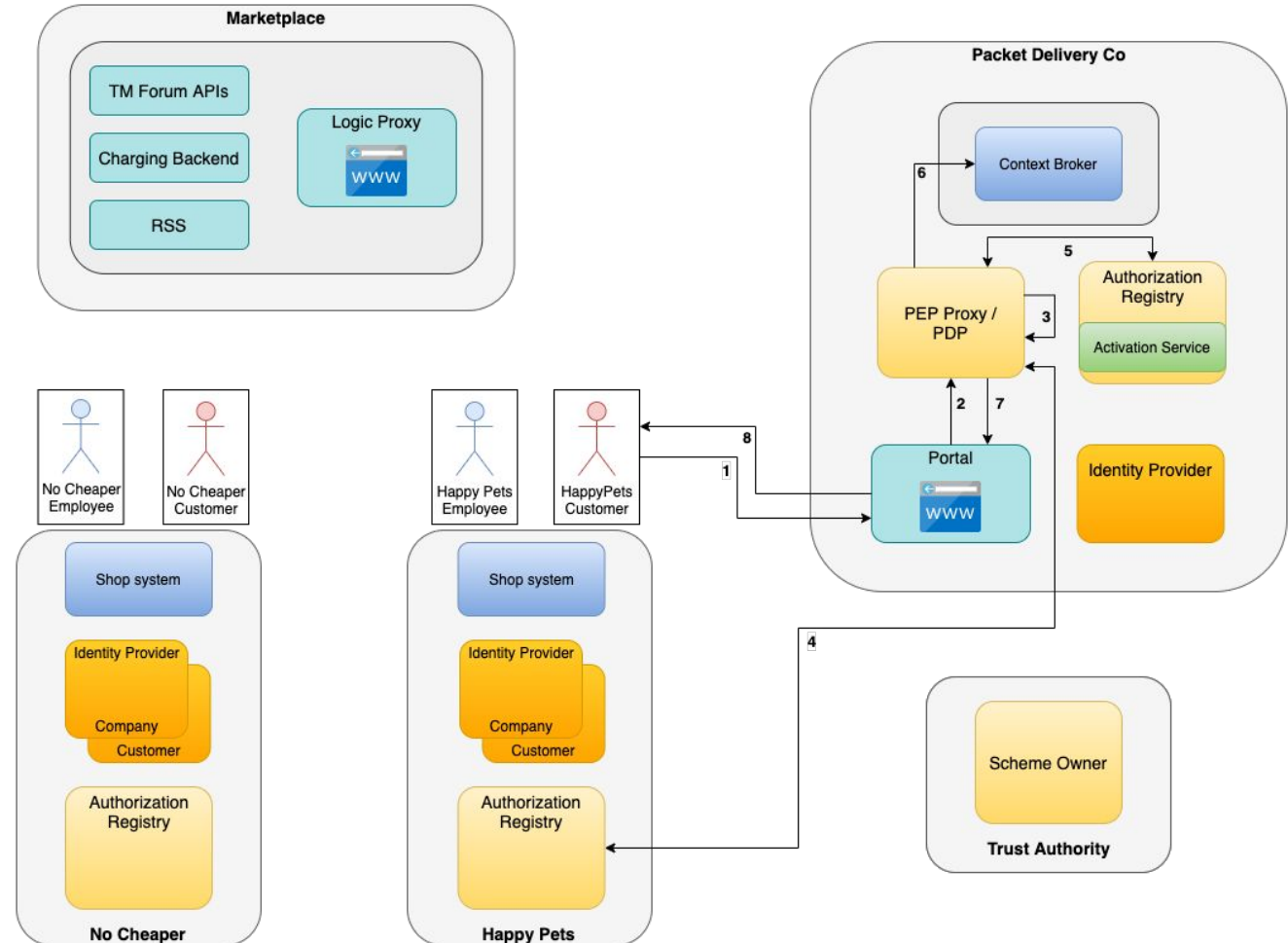


# Usage of data service

## Change Delivery Order

### Customer of Happy Pets Ltd. changes PDA/PTA of its delivery order

- Customer triggers sending of request to change PDA/PTA which includes the signed JWT (1-2)
- Proxy validates and verifies JWT (3)
- Proxy requests user policies at Happy Pets AR, and validates if customer was assigned the necessary premium service policy (4)
  - Alternatively JWT could already contain user policies and this step could be skipped
- Proxy checks at Packet Delivery AR if Happy Pets was able to assign premium service policy to its customers (5)
- Request is forwarded to Context Broker, attributes are changed and confirmation is displayed at portal (6-8)

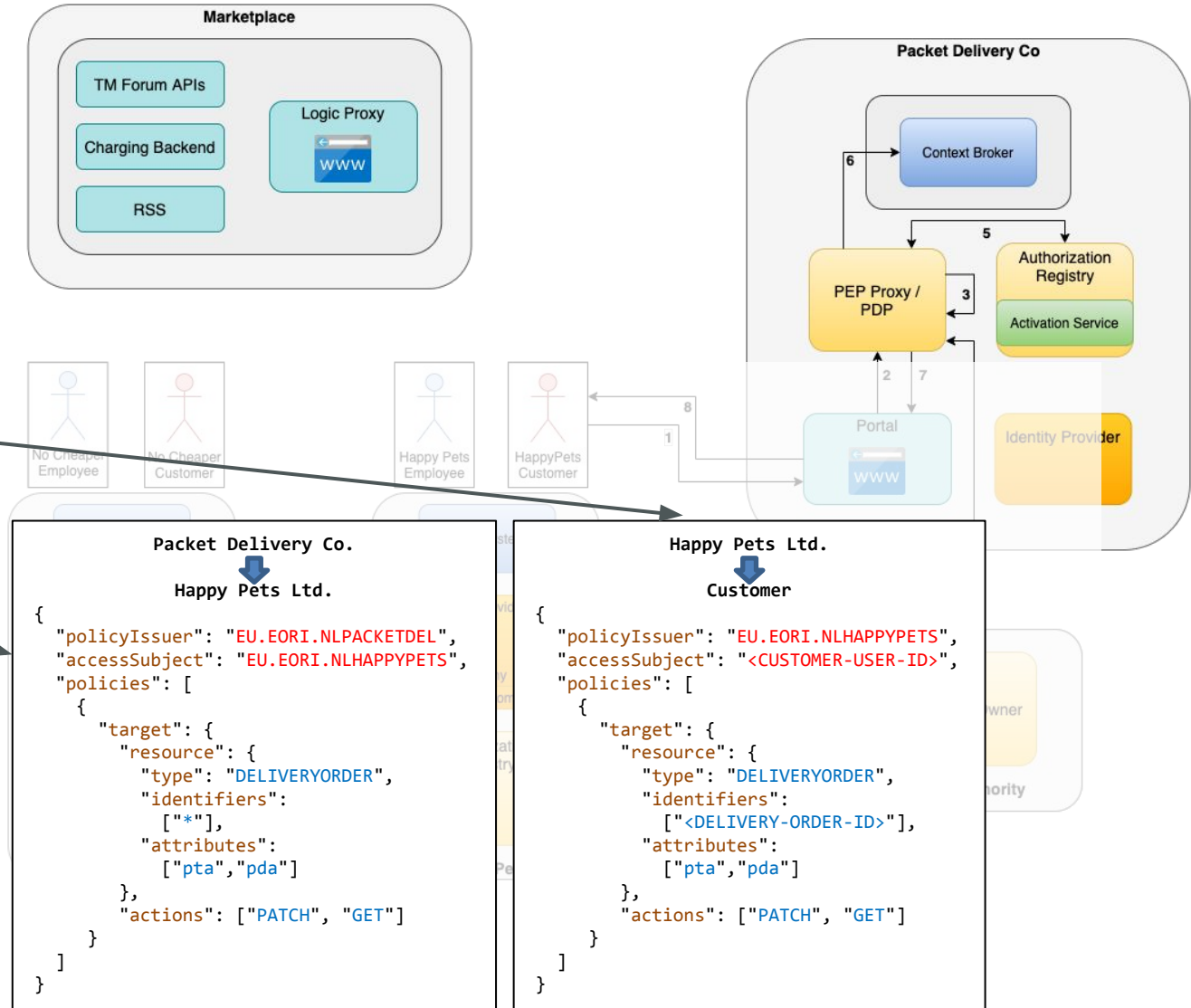


# Usage of data service

## Change Delivery Order

### Customer of Happy Pets Ltd. changes PDA/PTA of its delivery order

- Customer triggers sending of request to change PDA/PTA which includes the signed JWT (1-2)
- Proxy validates JWT (3)
- Proxy requests user policies at Happy Pets AR, and validates if customer was assigned the necessary premium service policy (4)
  - Alternatively JWT could already contain user policies and this step could be skipped
- Proxy checks at Packet Delivery AR if Happy Pets was able to assign premium service policy to its customers (5)
- Request is forwarded to Context Broker, attributes are changed and confirmation is displayed at portal (6-8)

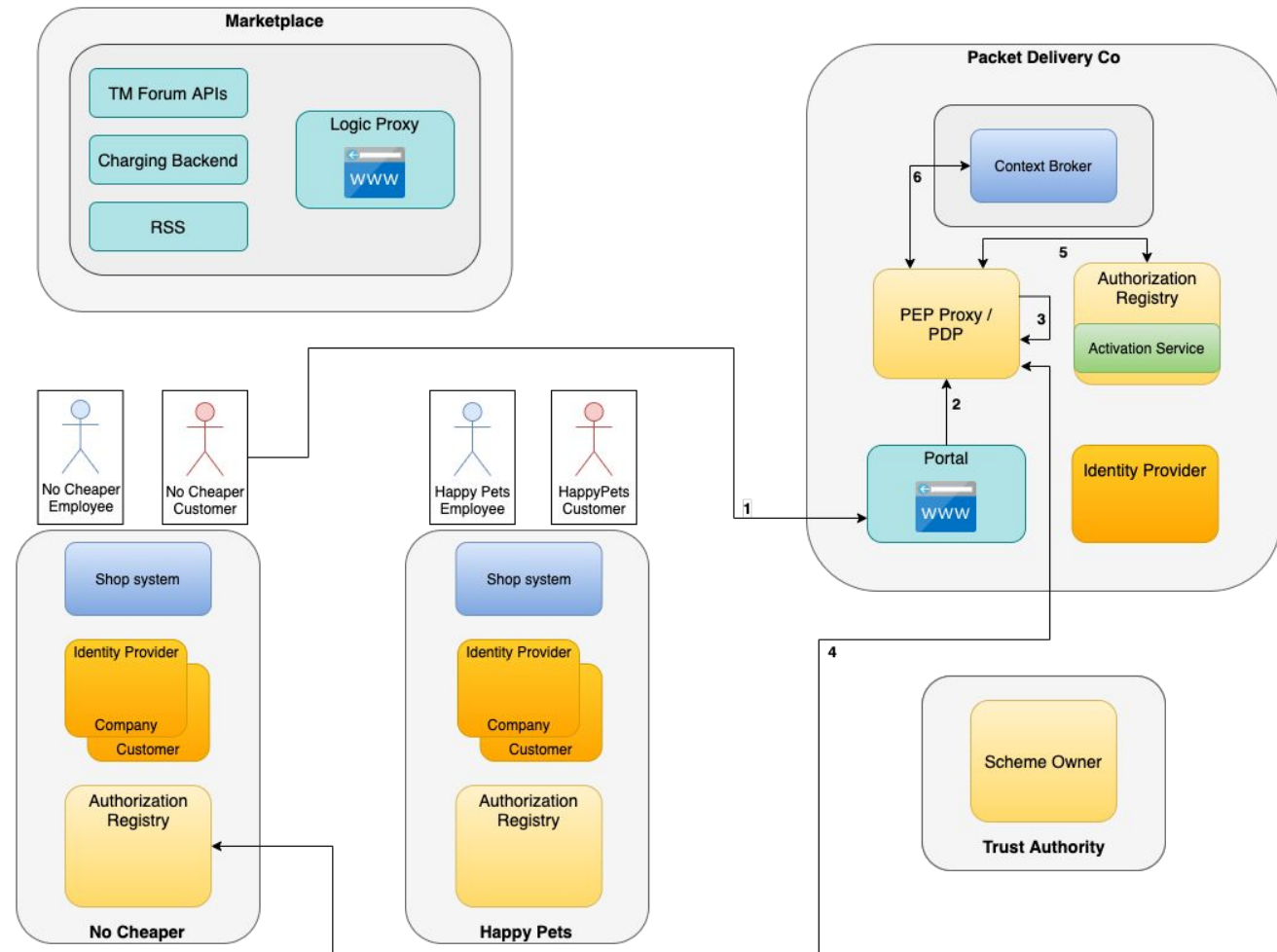
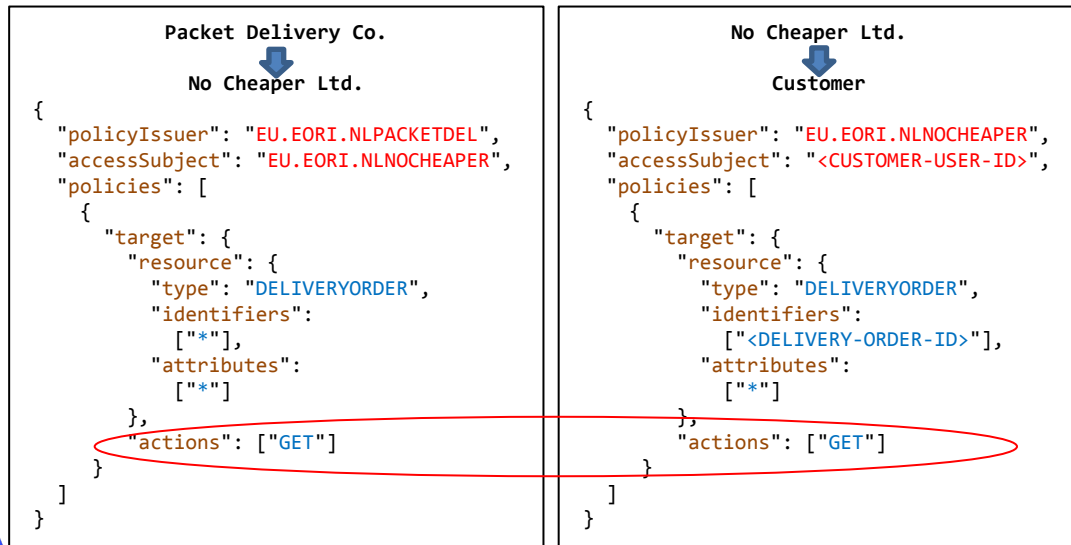


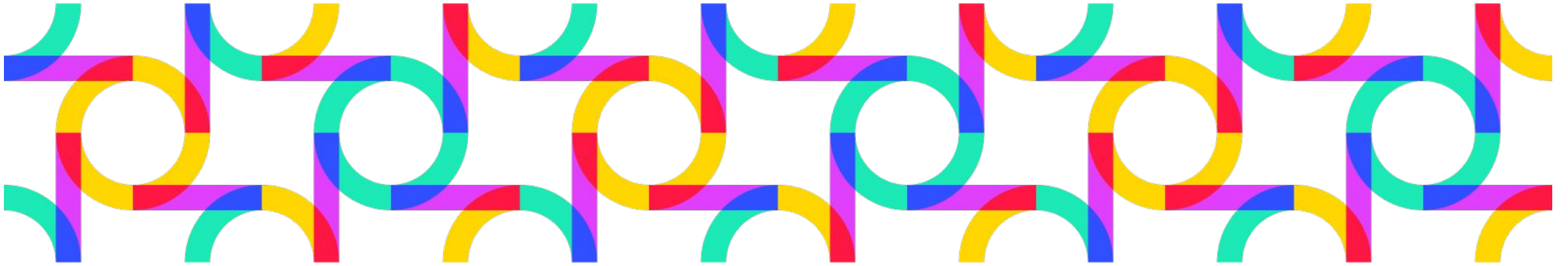
# Usage of Data Service

## Scenario: No Cheaper

### Additional scenario for No Cheaper organization

- No Cheaper acquired basic service offering
  - Only read access allowed at Packet Delivery service
  - No Cheaper is only allowed to assign standard service policies to its customers
- A request for changing PTA/PDA by No Cheaper customer will get rejected at PEP Proxy / PDP
- Request would also get rejected, even if No Cheaper would assign premium service policy to its customers





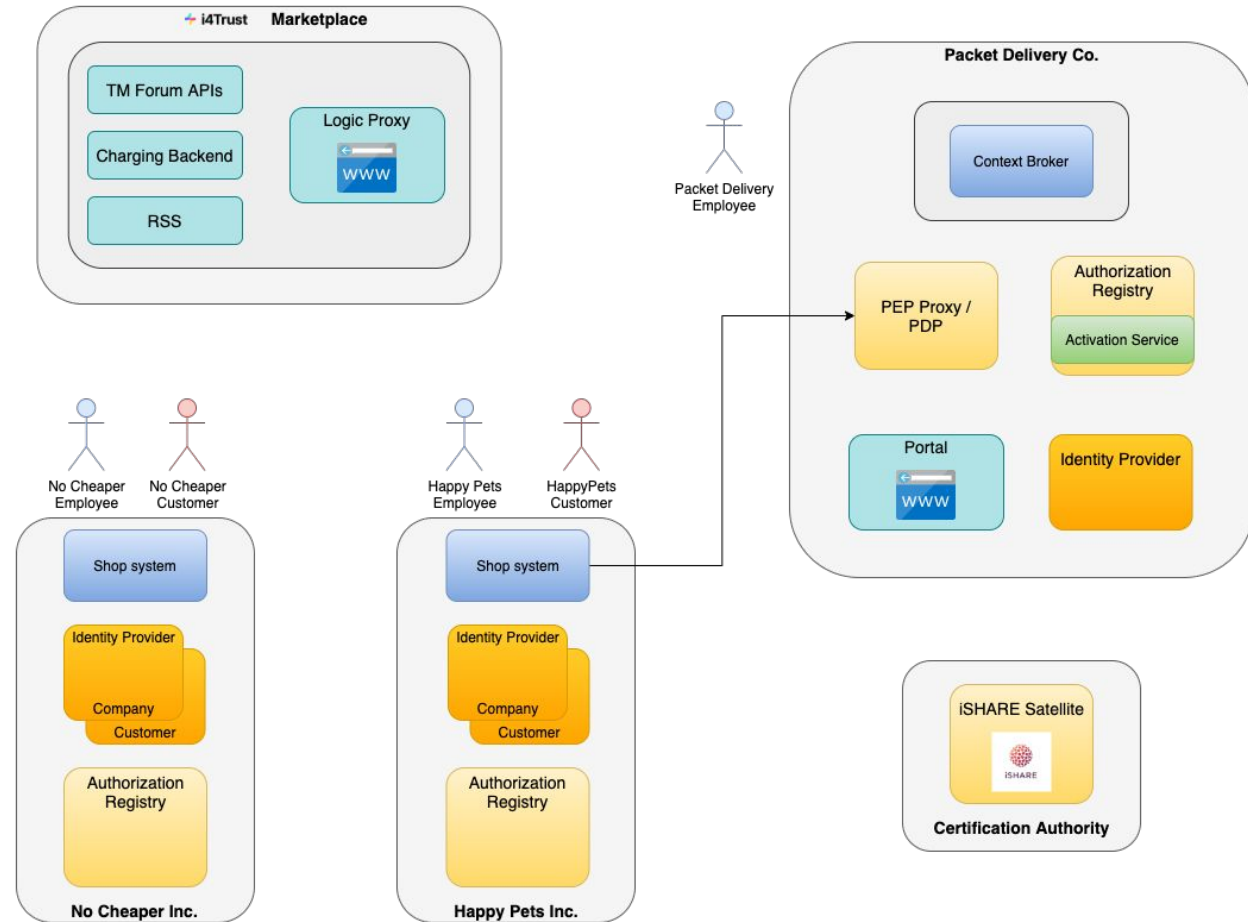
Another scenario (not shown)  
M2M Interaction

# M2M Interaction

- Previous use case showcased Human-to-Machine (H2M) interaction
  - End-user was accessing the service
- Framework also supports Machine-to-Machine (M2M) interactions

## Possible use case: Creation of delivery order

- When customer ordered a product, a delivery order needs to be created when product is ready for pickup
- Could be done directly by shop system at Packet Delivery Service
  - Requires additional offering
  - Created policies would be issued to retailer organisation and allow POST requests for entities of type DELIVERYORDER
- No human involved



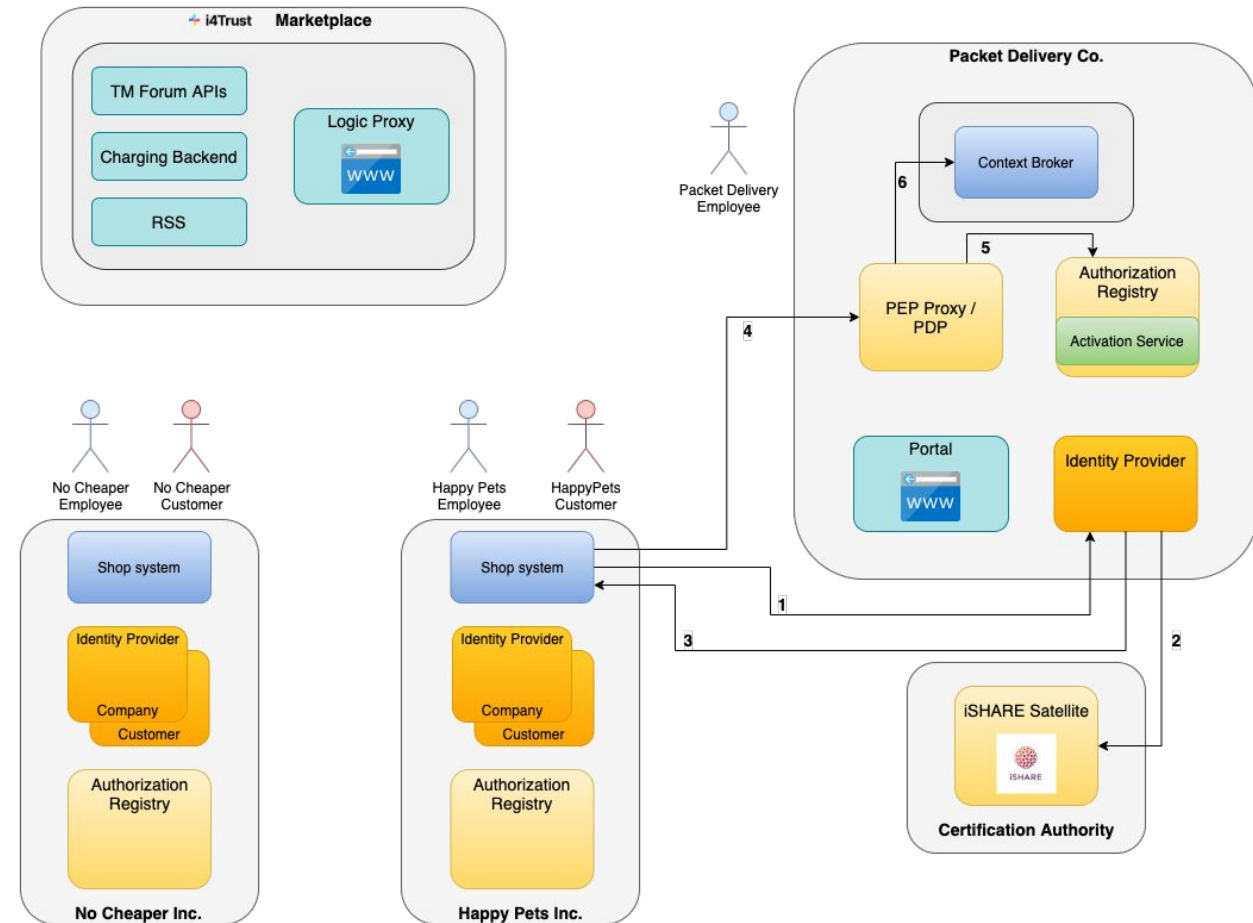
# M2M Interaction

**Shop system automatically creates a delivery order after a customer placed an order and product is ready to be picked up**

- Shop system creates signed JWT for identification and requests an access token at token endpoint\* of Packet Delivery (1)
- Packet Delivery validates HappyPets as trusted participant (2)
- A signed JWT is returned as access token (3)
- Shop system sends POST request (with access token) for creating a delivery order to proxy of Packet Delivery (4)
- Proxy validates and verifies access token, then checks at Packet Delivery AR if Happy Pets was issued policy for creating delivery orders (5)
- Request is forwarded to Context Broker and delivery order gets created (6)

\*: Service Provider needs to be provide a token endpoint

⇒ Using IDP here, since this functionality is implemented in Keyrock





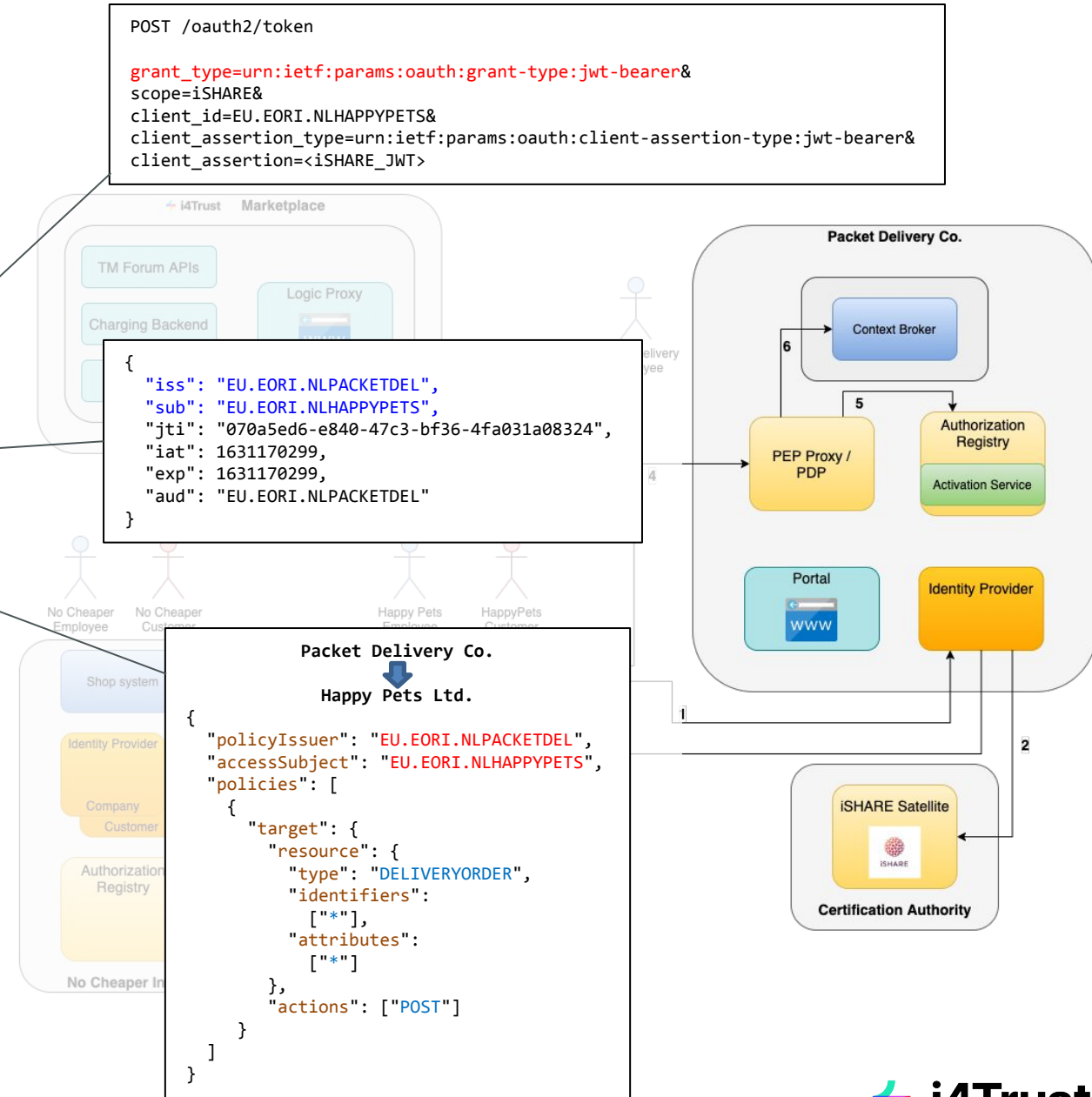
# M2M Interaction

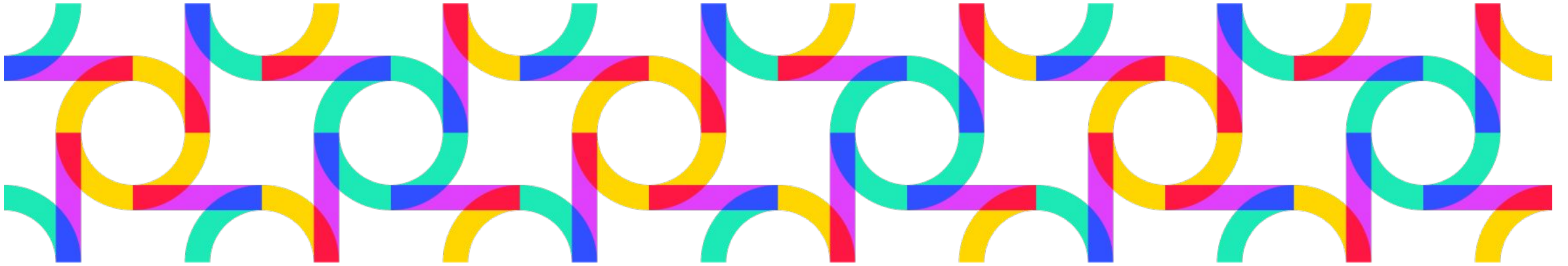
**Shop system automatically creates a delivery order after a customer placed an order and product is ready to be picked up**

- Shop system creates signed JWT for identification and requests an access token at token endpoint\* of Packet Delivery (1)
- Packet Delivery validates HappyPets as trusted participant (2)
- A signed JWT is returned as access token (3)
- Shop system sends POST request (with access token) for creating a delivery order to proxy of Packet Delivery (4)
- Proxy validates and verifies access token, then checks at Packet Delivery AR if Happy Pets was issued policy for creating delivery orders (5)
- Request is forwarded to Context Broker and delivery order gets created (6)

\*: Service Provider needs to be provide a token endpoint

⇒ Using IDP here, since this functionality is implemented in Keyrock

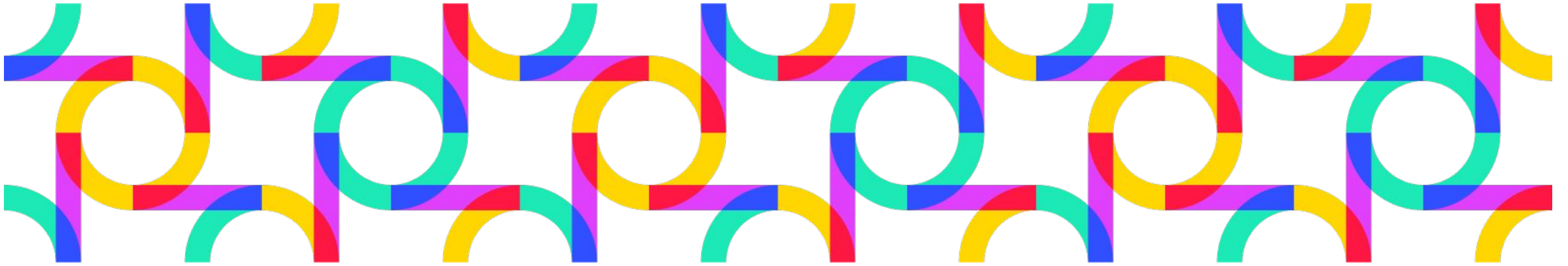




## Demo (OIDC4VP/SIOP2)

- Create an offering
- Acquire access to an offering
- Access the service





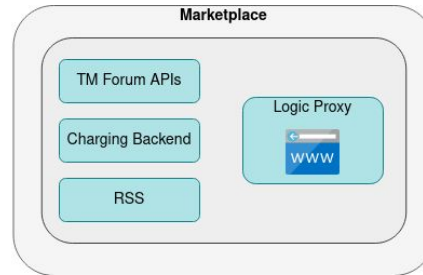
## Usage of Verifiable Credentials - OIDC4VP/SIOP2

# Prerequisites

## Architectural overview

### Marketplace

- Business API Ecosystem



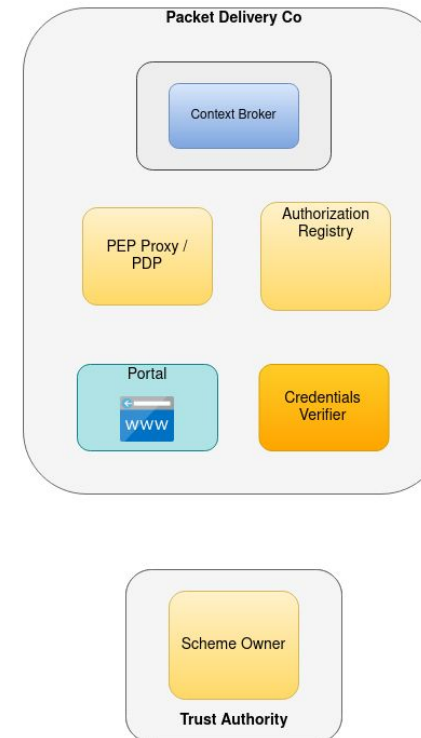
### Happy Pets Inc.

- Credentials Issuer
- Shop system application



### No Cheaper Inc.

- Credentials Issuer
- Shop system application



### Packet Delivery Co.

- Context Broker
- Credentials Verifier
- PEP Proxy/PDP
- Authorisation Registry (AR) + Activation Service (AS)
- Portal application

### Trust Authority

- iSHARE Satellite

# Additional configuration

## Marketplace

- Configure issuer DID for the participants

## PacketDelivery Co

- Deploy Credentials Verifier
- Setup DSBA-PDP
- Create the roles to be offered

## Happy Pets Inc.

- Deploy Credentials Issuer

## No Cheaper Inc.

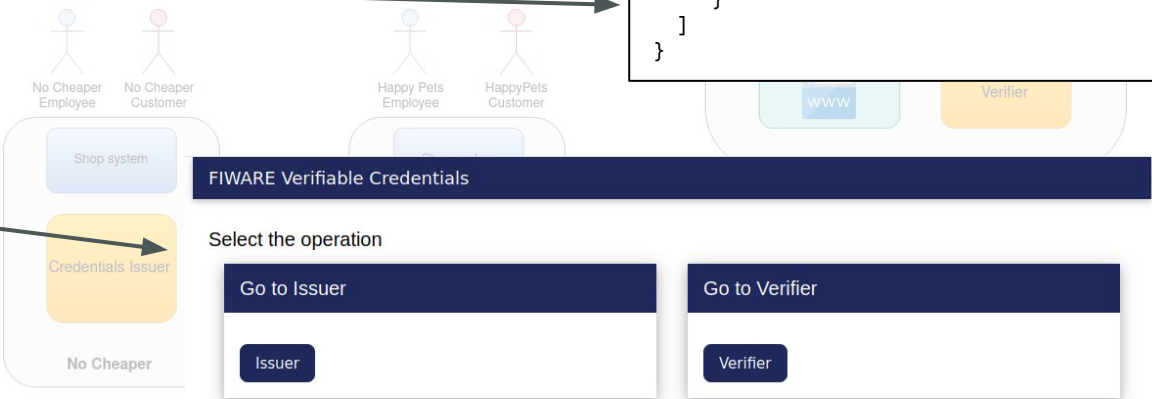
- Deploy Credentials Issuer

The screenshot shows a configuration form for a participant. It has tabs for 'List' and 'Detail'. Under the 'General' tab, there are fields for 'IDP EORI' (EU.EORI.NLHAPPYPETS), 'Issuer DID' (did:key:z6MkuKFXcv3qa86jx2KzdyvxYkS5Fr8knpzbejzeHCVnvFoK), 'Name' (Happy Pets Inc.), 'Server' (https://happypets-keyrock.i4trust-demo.fiware.dev), and 'Description (optional)' (Company IDP of happy Pets). An 'Update' button is at the bottom right.

```
Packet Delivery Co.

PREMIUM_CUSTOMER
('accessSubject')

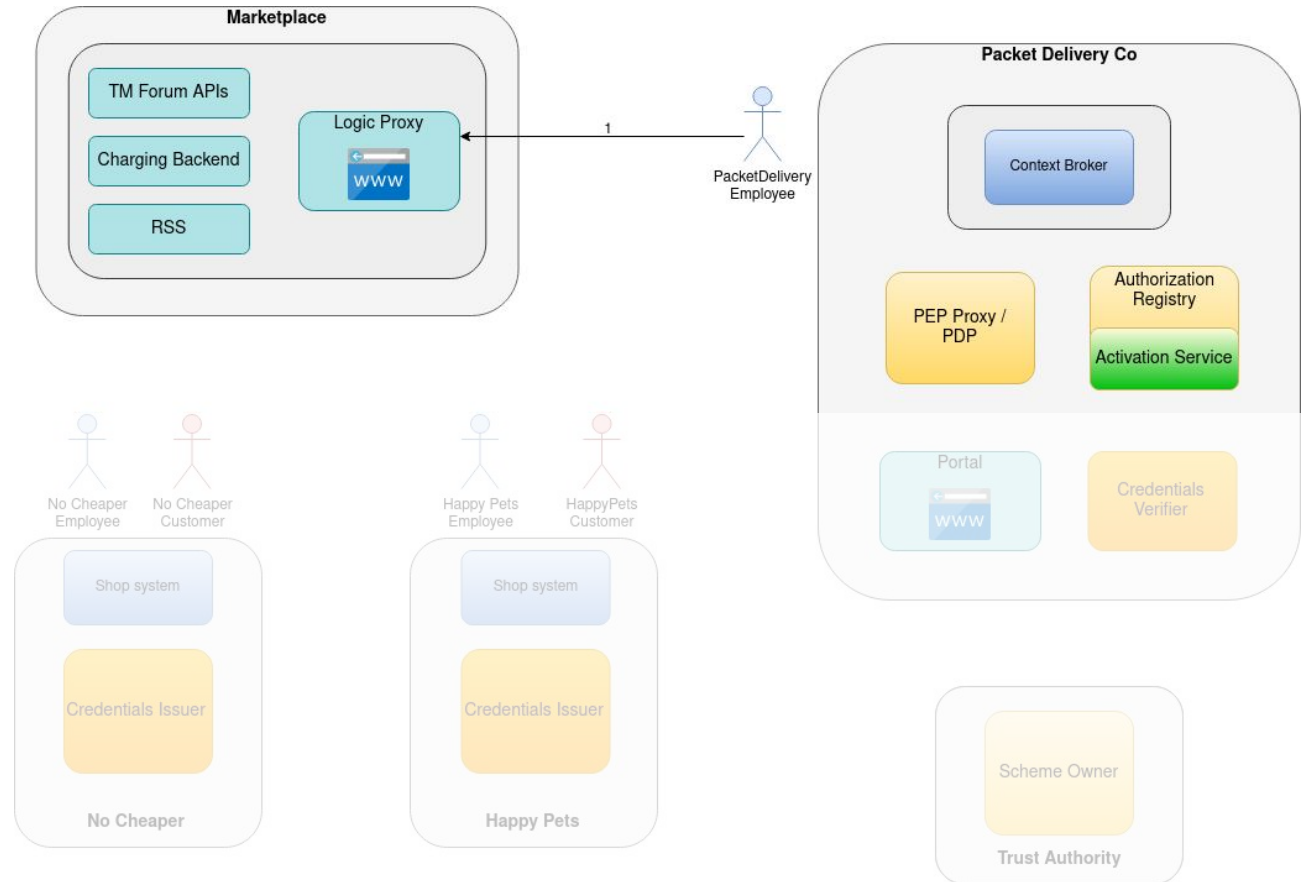
{
  "policyIssuer": "EU.EORI.NLPACKETDEL",
  "accessSubject": "PREMIUM_CUSTOMER",
  "policies": [
    {
      "target": {
        "resource": {
          "type": "DELIVERYORDER",
          "identifiers":
            ["*"],
          "attributes":
            ["*"]
        },
        "actions": ["GET", "PATCH"]
      }
    }
  ]
}
```



# Creating an offering

**Employee of Packet Delivery Co. creates offering on Marketplace for its basic and premium services**

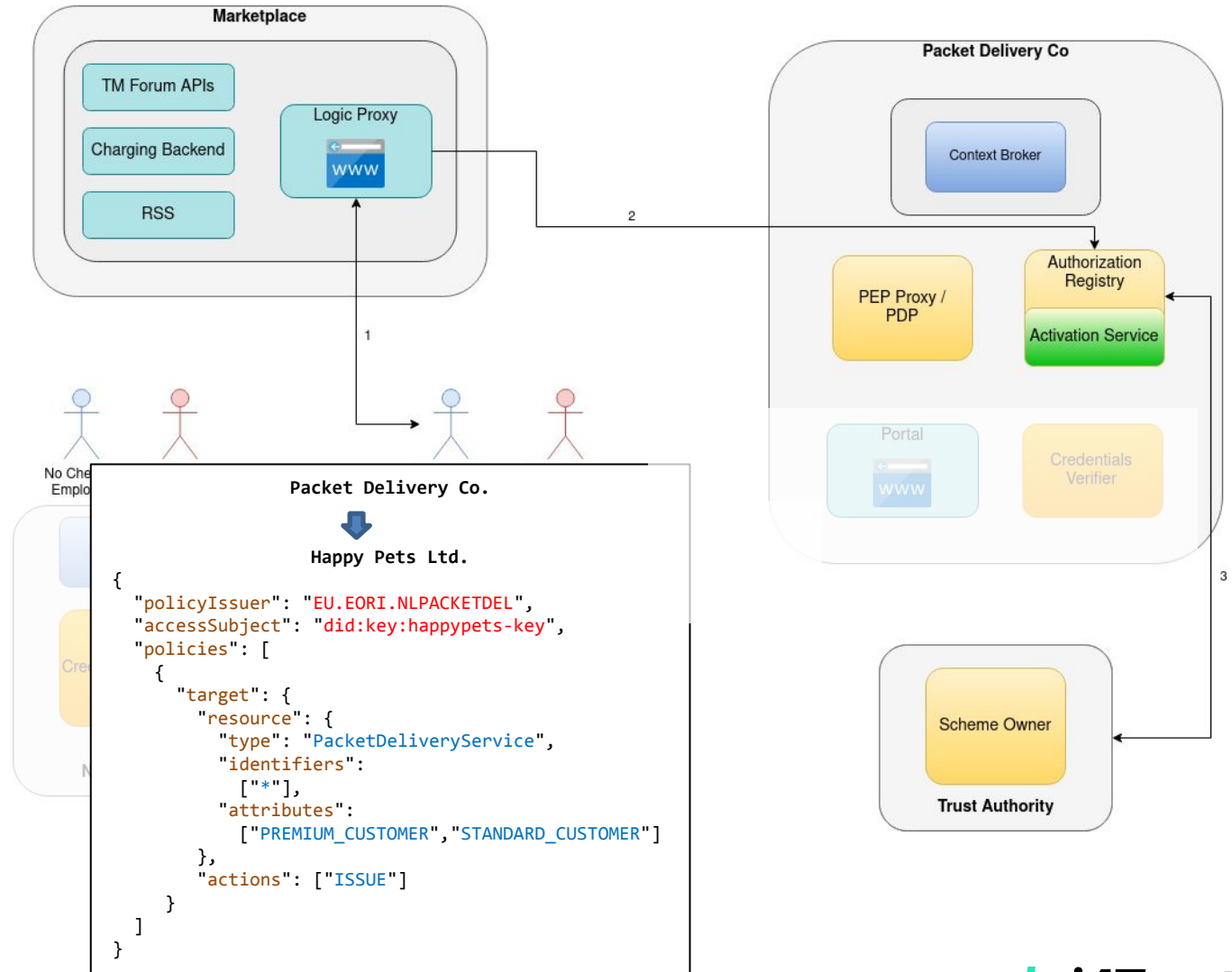
- Login steps still the same
- Employee has logged in and creates offerings for standard and premium services(1)  
Provides information to offering about:
  - Service endpoint (+ API specification)
  - Roles offered
  - Supported VC-Type(f.e. PacketDeliveryService)
  - Other terms and Conditions, pricing



# Acquisition of offerings

## Employee of Happy Pets Ltd acquires access to premium service offering

- Login steps still the same
- Employee selects the Packet Delivery **premium service** and performs the checkout process incl. providing payment details (1)
- Marketplace requests at Packet Delivery AR to register HappyPets as a Trusted Issuer, allowed to issue Verifiable Credentials to its customers (2)
  - Authentication at AR based on OAuth2
- AR validates Marketplace as trusted party (3)
- Policy is created at AR, allowing Happy Pets to issue credentials of type "PacketDeliveryService" with roles "PREMIUM\_CUSTOMER" and "STANDARD\_CUSTOMER" to its customers (4)

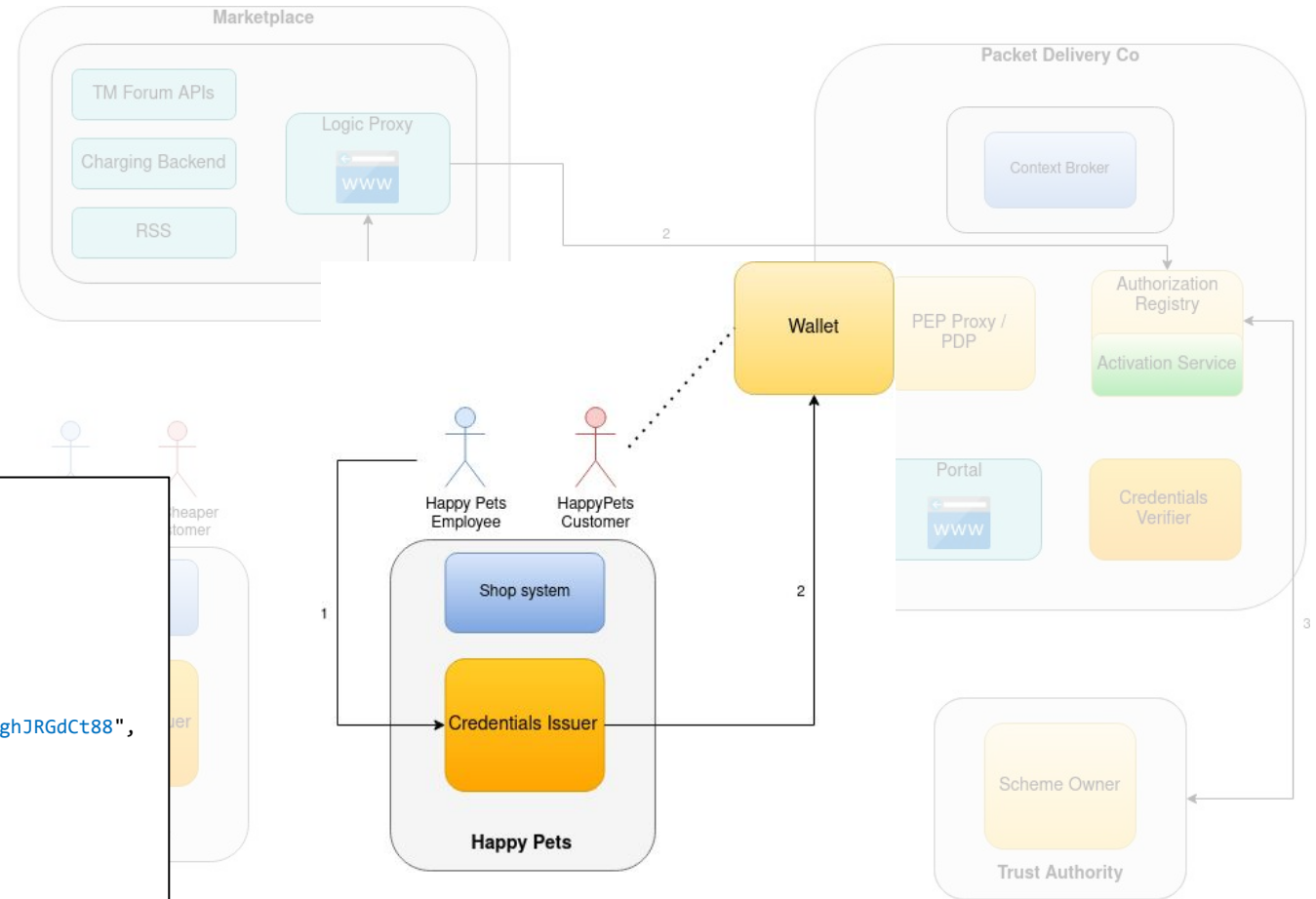


# Issuing of the Verifiable Credential

## Happy Pets Ltd. issues a Verifiable Credential to its customer

- Happy Pets Employee creates the Verifiable Credential(1)
- Credentials Issuer presents Verifiable credential to the Customers Wallet, VC is stored in the Wallet(2)

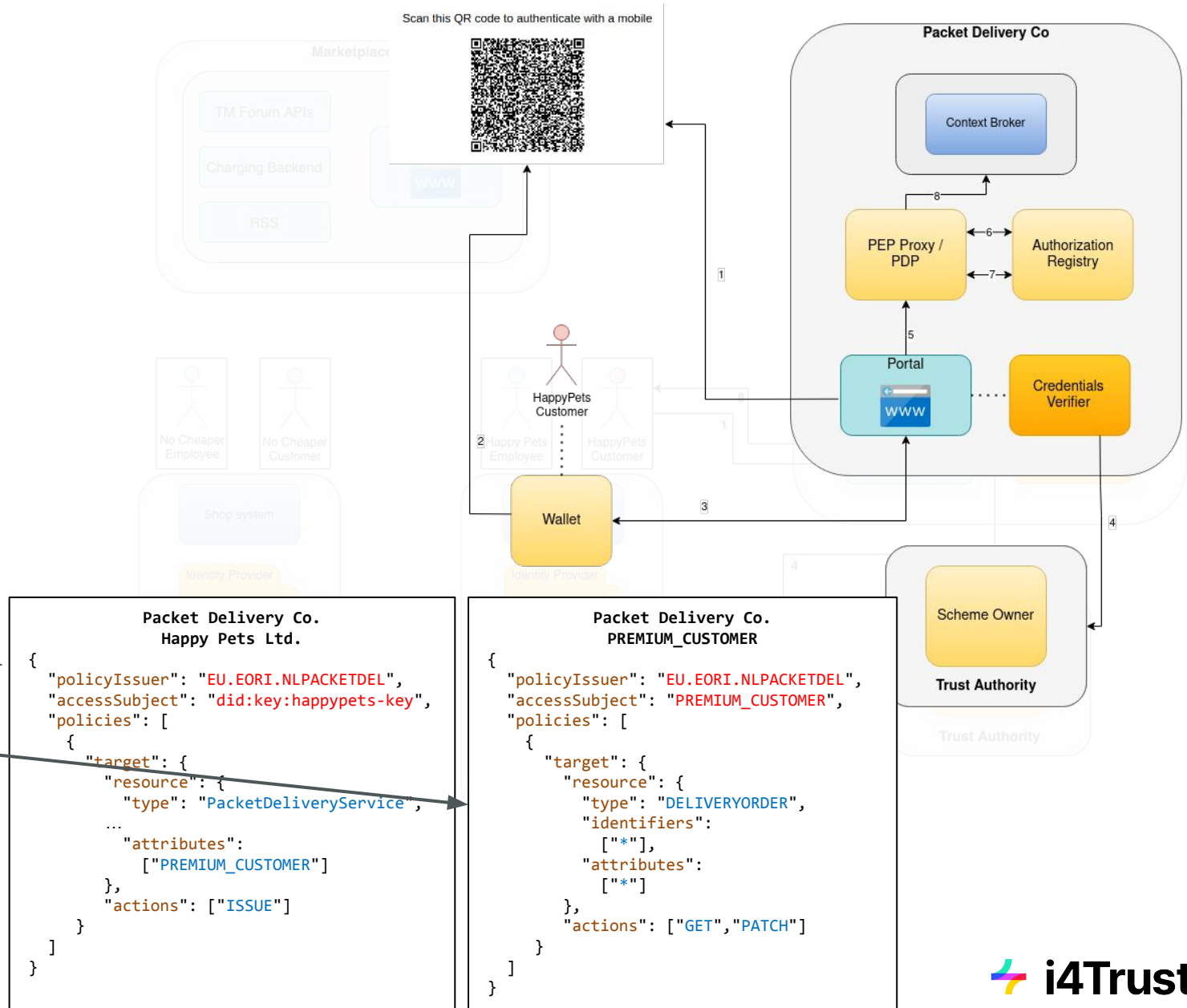
```
{
  "@context": [...],
  "credentialSchema": {
    "id": "https://.../schema.json",
    "type": "FullJsonSchemaValidator2021"
  },
  "credentialSubject": {
    "familyName": "Customer",
    "firstName": "Test",
    "id": "did:key:z6Mkfdio1n9SKoZUtKdr9GTCZsRPbwHN8f7rbJghJRGdCt88",
    "roles": [
      {
        "names": [
          "PREMIUM_CUSTOMER"
        ],
        "Target": "EU.EORI.NLPACKETDEL"
      }
    ]
  },
  ...,
  "issuer": "did:key:happypets-key",
  ...
}
```

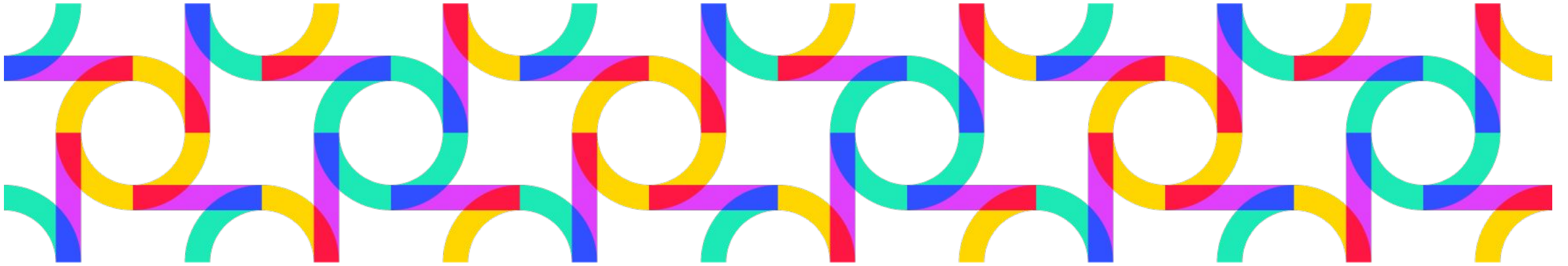


# Usage of data service

## Customer of Happy Pets Ltd. logs into Packet Delivery Co, checks delivery status

- Portal presents the QR with Login Information(1)
- Customer scans Login-QR with the Wallet(2)
- Wallet sends VC, Verifier will verify and create a JWT for the portal (3)
- Credentials Verifier checks issuer at the Trusted Issuers Registry(4)
- Portal uses the JWT to access the PEP-Proxy(5)
  - Verify that the issuer is allowed to issue the credential(6)
  - Verify that the role in the VC is allowed to make the call(7)
- Request forwarded to the broker(8)





## Setup of components of the i4Trust experimentation framework on Kubernetes



# Deployment Marketplace: Business API Ecosystem

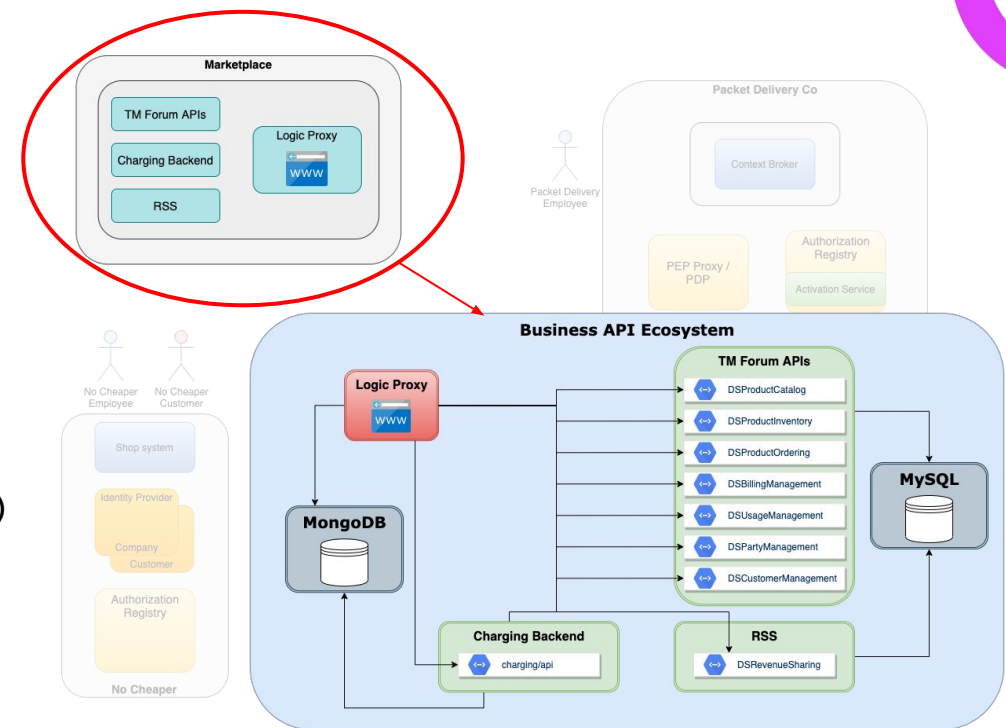
- Need to deploy databases, elasticsearch and BAE
- Helm charts available
  - MongoDB: bitnami/mongodb (DB: v3.6.21)
  - MySQL: t3n/mysql (DB: v5.7)
  - elasticsearch: elastic/elasticsearch (v7.5.1)
  - Business API Ecosystem: fiware/business-api-ecosystem (v8.0.0)
- Need to supply config parameters as values when deploying with helm

## Resources:

- <https://github.com/FIWARE-TMForum/Business-API-Ecosystem>
- <https://business-api-ecosystem.readthedocs.io/en/latest/>
- <https://github.com/i4Trust/tutorials/tree/main/i4Trust-Marketplace>

```
helm repo add elastic https://helm.elastic.co
helm repo add t3n https://storage.googleapis.com/t3n-helm-charts
helm repo add bitnami https://charts.bitnami.com/bitnami
helm repo add fiware https://fiware.github.io/helm-charts/
helm install -f ./values.yml elastic elastic/elasticsearch --version 7.5.1
helm install -f ./values.yml mysql t3n/mysql --version 0.1.0
helm install -f ./values.yml mongodb bitnami/mongodb --version 10.30.12
helm install -f ./values.yml business-api-ecosystem /path/to/chart --version 0.4.17
```

- Need to add service asset plugin to charging backend component
  - <https://github.com/i4Trust/bae-i4trust-service>
  - <https://business-api-ecosystem.readthedocs.io/en/latest/plugins-guide.html#installing-asset-plugins>



BAE Component	Docker Image
BAE APIs	<a href="#">fiware/biz-ecosystem-apis:v8.0.0</a>
BAE RSS	<a href="#">fiware/biz-ecosystem-rss:v8.0.0</a>
BAE Charging Backend	<a href="#">fiware/biz-ecosystem-charging-backend:v8.0.0</a>
BAE Logic Proxy	<a href="#">fiware/biz-ecosystem-logic-proxy:v8.1.0-dev</a>

# Deployment

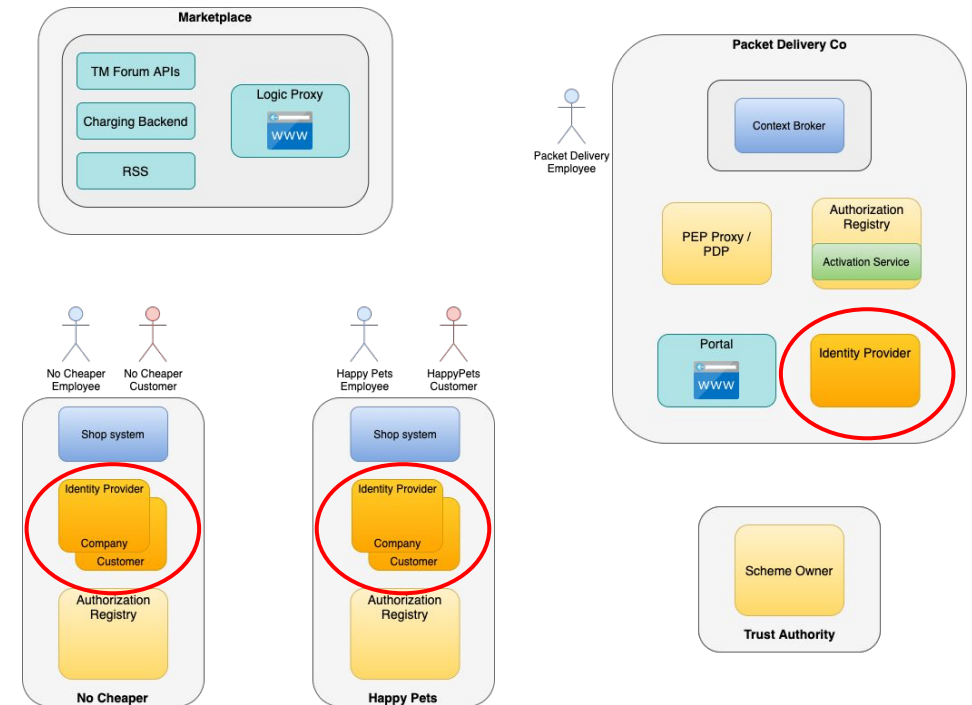
## Identity Providers: Keyrock

- Several IDP instances to be deployed for the different organisations/environments
- Helm chart available: fiware/keyrock (RC tag: [i4trust-rc8](#))
- Need to supply config parameters as values when deploying with helm
- Keyrock requires MySQL database, can use the one previously deployed
  - Need to specify separate database within MySQL

### Resources:

- <https://github.com/qing/fiware-idm>
- <https://fiware-idm.readthedocs.io/en/latest/>
- <https://github.com/i4Trust/tutorials/tree/main/Data-Service-Provider#keyrock>

```
# To be performed for each Keyrock instance (change release name)
helm install -f ./values.yml keyrock /path/to/chart --version 0.4.9
```



Component	Docker Image
Keyrock	<a href="#">fiware/idm:i4trust-rc8</a>

# Deployment

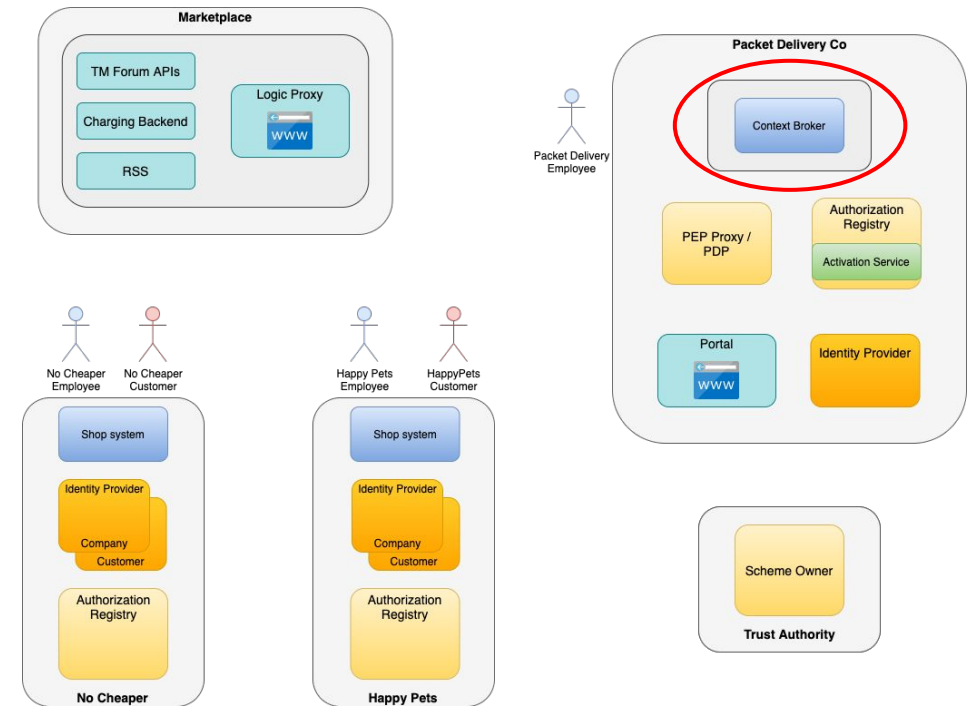
## Service provider: Context Broker

- Deploy instance of orion context broker
- Helm chart available: fiware/orion (1.0.1)
- Need to supply config parameters as values when deploying with helm
- Orion requires MongoDB database, can use the one previously deployed
  - Need to specify separate database within MongoDB

### Resources:

- <https://fiware-orion.readthedocs.io/en/master/>

```
helm repo add fiware https://fiware.github.io/helm-charts/  
helm install -f ./values.yml orion fiware/orion
```



Component	Docker Image
Orion Context Broker	<a href="#">fiware/orion-ld:1.0.1</a>

# Deployment

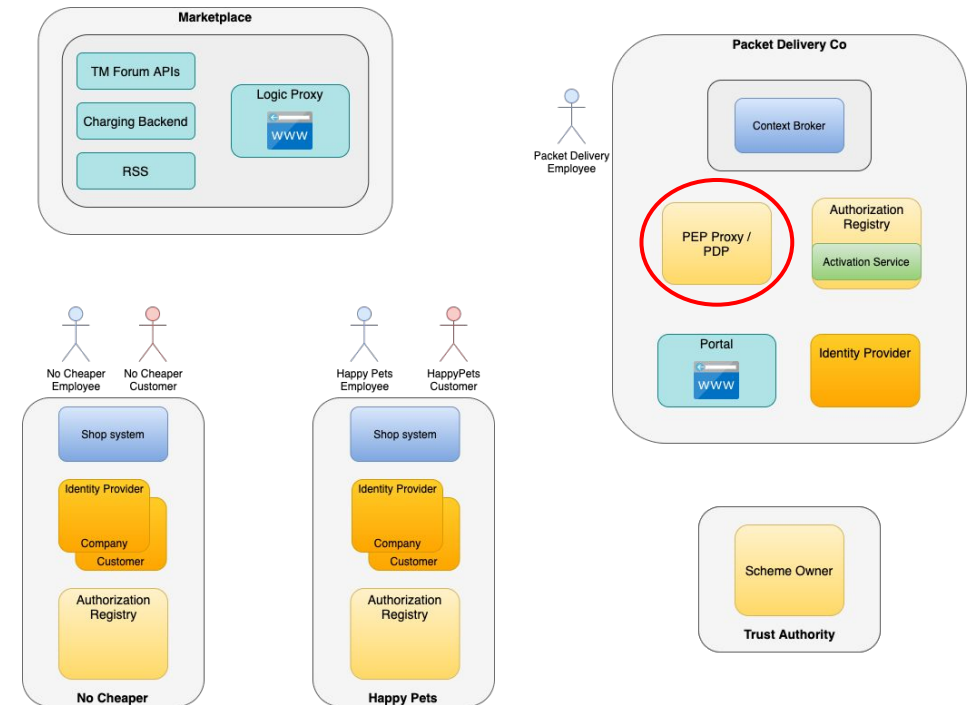
## PEP/PDP: Kong

- Deploy instance of Kong
- Official Helm chart available: kong/kong (Kong version: 2.8.1)
- Need to supply config parameters as values when deploying with helm
  - In addition need to supply kong.yml when using DB-less config
- Need to install `ngsi-ishare-policies` plugin
  - Also FIWARE Kong build available bundling all FIWARE plugins
- For VerifiableCredentials: needs to install the `pep-plugin`, also available in the FIWARE Kong build

### Resources:

- <https://konghq.com/>
- <https://github.com/FIWARE/kong-plugins-fiware/tree/main/kong-plugin-ngsi-ishare-policies>
- <https://github.com/FIWARE/kong-plugins-fiware/tree/main/kong-pep-plugin>
- <https://github.com/i4Trust/tutorials/tree/main/PacketDelivery-ReferenceExample/Data-Service-Provider#kong>

```
helm install -f ./values.yml kong kong/kong --set ingressController.installCRDs=false  
--set-file dblessConfig.config=./kong/kong.yml --version 2.8.0
```



Component	Docker Image
Kong + plugin	<a href="https://quay.io/fiware/kong:0.5.0">quay.io/fiware/kong:0.5.0</a>

# Deployment

## PEP/PDP: Kong

- Configure Kong service in kong.yml

### Example config (OIDC/iSHARE): Parameters

- Local orion endpoint
- External endpoint (path) for service
- Plugin config
  - Where to expect access token
  - Authorization Registry config
  - iSHARE Satellite config
  - Service provider identifier (EORI)
- Remove Authorization header before forwarding to orion

```
_format_version: "2.1"
_transform: true
services:
  - host: "i4trust-demo-pdc-orion"
    name: "pdc-orion-packetdelivery"
    port: 1026
    protocol: http

  routes:
    - name: pdc
      paths:
        - /packetdelivery
      strip_path: true

  plugins:
    - name: ngsi-ishare-policies
      config:
        access_token:
          header_names:
            - "authorization"
            - "Authorization"
        ar:
          identifier: "EU.EORI.NL000000004"
          host: "https://ar.isharetest.net"
          token_endpoint: "https://ar.isharetest.net/connect/token"
          delegation_endpoint: "https://ar.isharetest.net/delegation"
        satellite:
          identifier: "EU.EORI.NL000000000"
          host: "https://scheme.isharetest.net"
          token_endpoint: "https://scheme.isharetest.net/connect/token"
          trusted_list_endpoint: "https://scheme.isharetest.net/trusted_list"
        jws:
          identifier: "EU.EORI.NLPACKETDEL"

    - name: request-transformer
      config:
        remove:
          headers:
            - Authorization
            - authorization
```

# Deployment

## PEP/PDP: Kong

- Configure Kong service in kong.yml

### Example config (OIDC4VP/SIOP2): Parameters

- Local orion endpoint
- External endpoint (path) for service
- Plugin config
  - Type of PDP to be used
  - Endpoint of the pdp
- Remove Authorization header before forwarding to orion

```
- host: "i4trust-demo-pdc-orion"
  name: "orion-vc"
  port: 1026
  protocol: http

routes:
  - name: orion-vc
    paths:
      - /orion-vc
    strip_path: true

plugins:
  - name: pep-plugin
    config:
      pathprefix: "/orion-vc"
      authorizationendpointtype: ExtAuthz
      authorizationendpointaddress: http://i4trust-demo-pdc-pdp-dsba-pdp:8080/authz

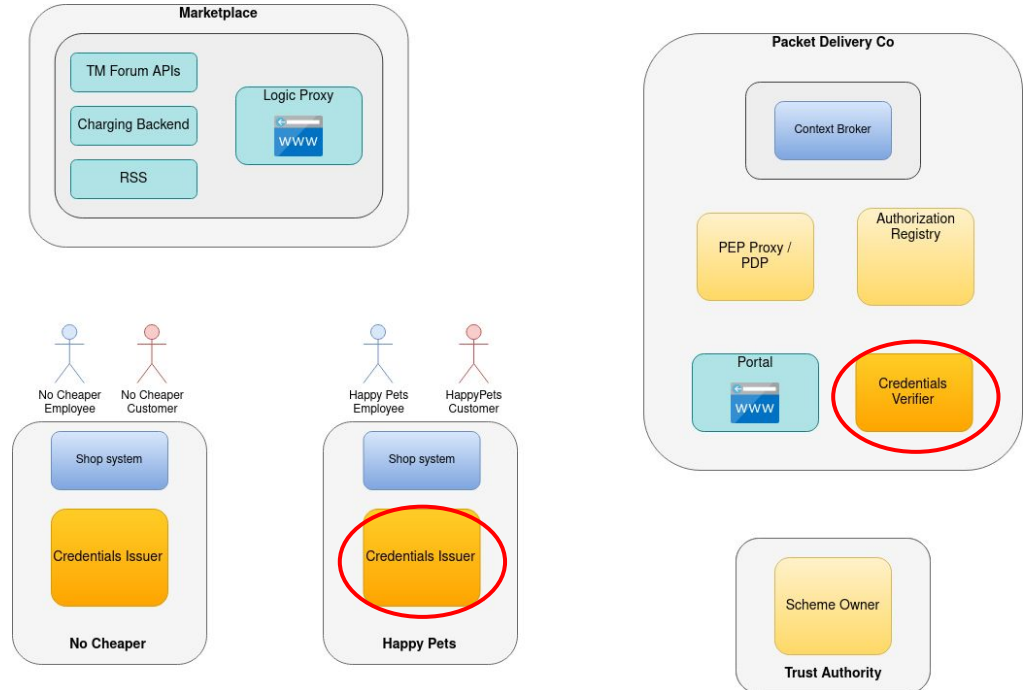
  - name: request-transformer
    config:
      remove:
        headers:
          - Authorization
          - authorization
```

# Deployment Verifier & Issuer

- Deploy VCBackend to provide issuer and verifier
- Deploy VCWaltid to serve the VCBackend
- Helm charts available:
  - VCBackend: `i4trust/vcbackend`
  - VCWaltid: `i4trust/vcwaltid`
- Need to supply config parameters as values when deploying with helm

## Resources:

- <https://github.com/FIWARE-Ops/fiware-gitops/tree/master/aws/i4trust/i4trust-demo/happypets-issuer-vcwaltid>
- <https://github.com/FIWARE-Ops/fiware-gitops/tree/master/aws/i4trust/i4trust-demo/nocheaper-issuer-vcbackend>



Component	Docker Image
VCBackend	<a href="#">i4trust/vcbackend:0.0.7</a>
VCWaltid	<a href="#">i4trust/vcwaltid:0.0.3</a>



# Deployment

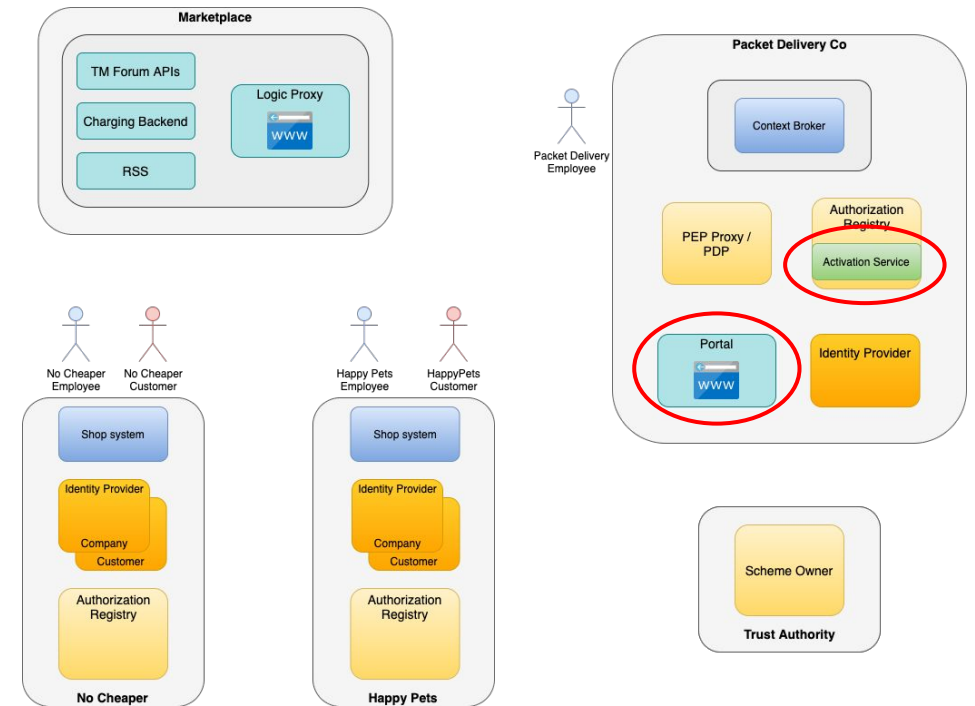
## Packet Delivery Co.: Portal + AS

- Deploy portal and activation service demo applications
- Helm charts available:
  - Portal: i4trust/pdc-portal
  - Activation Service: i4trust/activation-service
- Need to supply config parameters as values when deploying with helm

### Resources:

- <https://github.com/i4Trust/pdc-portal>
- <https://github.com/i4Trust/tutorials/tree/main/Data-Service-Provider#activation-service>
- <https://github.com/i4Trust/activation-service>
- <https://github.com/i4Trust/tutorials/tree/main/Data-Service-Provider#packet-delivery-portal-demo-application>

```
helm repo add i4trust https://i4trust.github.io/helm-charts
helm install -f ./values.yml pdc-portal i4trust/pdc-portal --version 2.1.1
helm install -f ./values.yml activation-service i4trust/activation-service --version 1.1.1
```



Component	Docker Image
PDC Portal	<a href="#">i4trust/pdc-portal:2.1.0</a>
Activation Service	<a href="#">i4trust/activation-service:1.2.0</a>

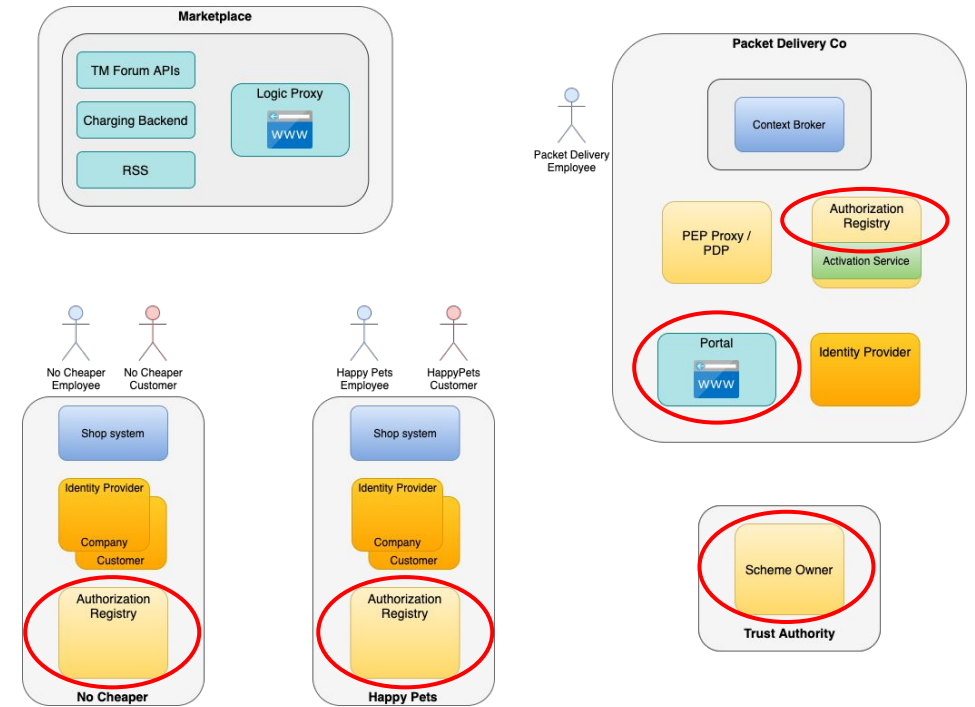
# Deployment iSHARE AR + SO

## Scheme Owner:

- Can use iSHARE Satellite test instance
  - <https://scheme.isharetest.net/>
  - Open Source version available soon
- API documentation:  
<https://dev.ishareworks.org/scheme-owner/parties.html>
- iSHARE can be contacted in order to get registered at the iSHARE Satellite test instance and to retrieve test certificates and EORIs

## Authorisation Registry:

- Can use iSHARE AR test instance
  - <https://ar.isharetest.net/>
- AR functionality has been integrated into Keyrock and can be used as well
  - [Keyrock Manual: Configuration - External participant](#)
- Another alternative is to set up own AR
  - Reference implementation:  
<https://github.com/iSHAREScheme/AuthorizationRegistry>
  - API documentation:  
<https://dev.ishareworks.org/delegation/endpoint.html>



# Deployment

**Deployment instructions and example Helm configurations can also be found at**

- FIWARE production on K8s repository
  - <https://github.com/FIWARE/production-on-k8s>
- i4Trust tutorials repository
  - <https://github.com/i4Trust/tutorials>
- Deployment of reference example via gitops
  - <https://github.com/FIWARE-Ops/fiware-gitops/tree/master/aws/i4trust/i4trust-demo>

**More information about Data Spaces can be found in the FIWARE Catalogue**

- <https://github.com/FIWARE/catalogue/tree/master/data-spaces>

# Thank you!

Dr. Dennis Wendland  
FIWARE Foundation  
[dennis.wendland@fiware.org](mailto:dennis.wendland@fiware.org)

