Service Marketplace

CX Core ART

Exported on 10/27/2022

Table of Contents

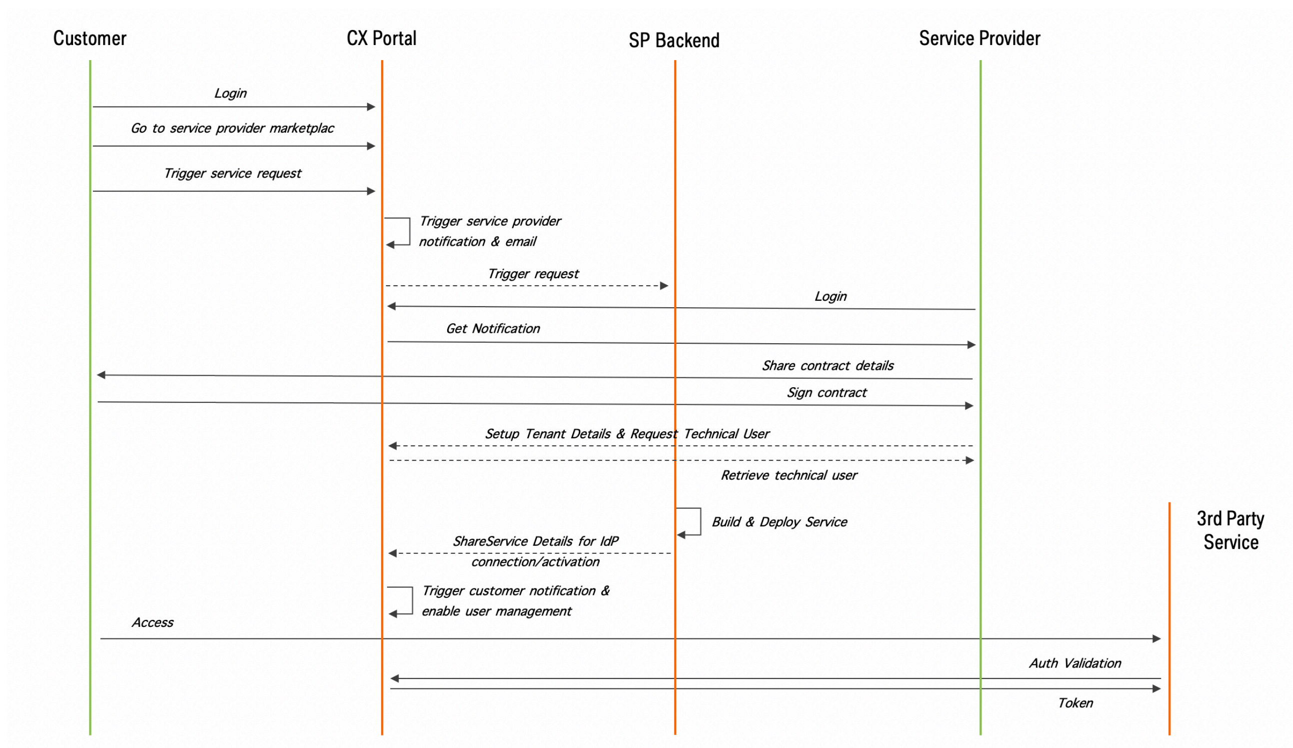
1 Service Marketplace Details Journey

1.1 Summary

Service Marketplace is used to join a service provider with a costumer.

1.2 Service Request Flow

In the flow diagram below, the first draft of the communication/flow between Costumer, Portal and Service Provider is shown.



*please note: changes will still get applied, however the flow diagram is giving a first good impression of the service provider flow

1.3 Service Provider Functionalities

The following portal system functions are relevant for service providers

- [ ] Service Publishing Process - Priority: medium, Status: in design
- [ ] Service Marketplace - Priority: medium, Status: in design
- [ ] Self Description - Priority: low, Status: in validation
- [ ] Service Management - Priority: medium, Status: in design
- [ ] Service Provider Technical User Request

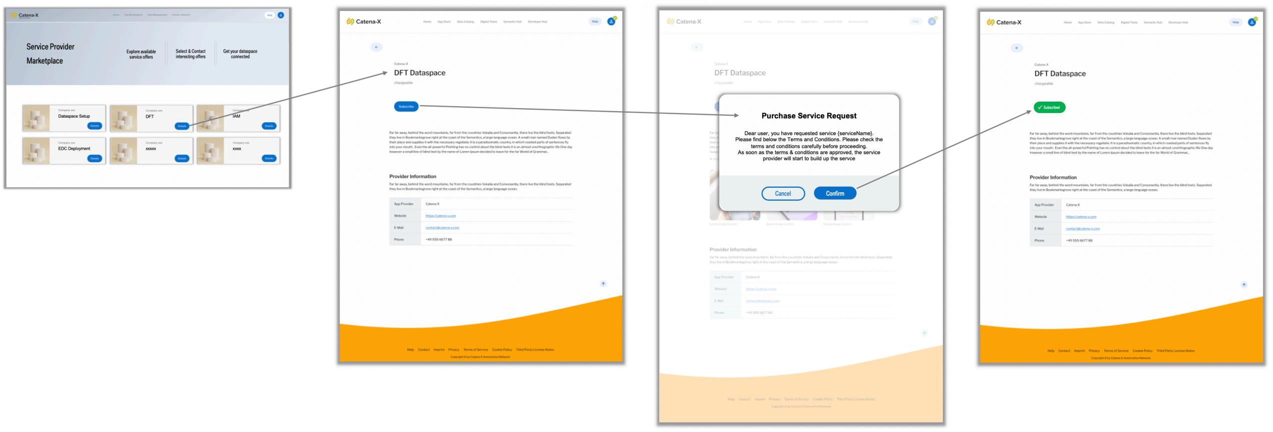
2 Service Marketplace Service Purchasing

2.1 Summary

Service Marketplace is used to join a service provider with a costumer.

2.2 Service Marketplace Flow

First DRAFT Version



Dear user, you have requested service {serviceName}. Please find below the Terms and Conditions. Please check the terms and conditions carefully before proceeding. As soon as the terms & conditions are approved, the service provider will start to build up the servce

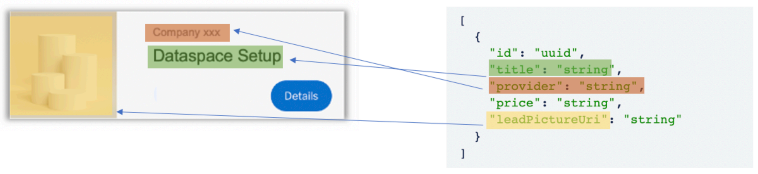
2.3 Implementation - Marketplace

2.3.1 #1 Get Service Offers

Get Service offers can get viewed / displayed by all users of the network. Its used to display the available services in status "ACTIVE" inside the marketplace.

Endpoint: GET /api/services/active

Data mapping logic:



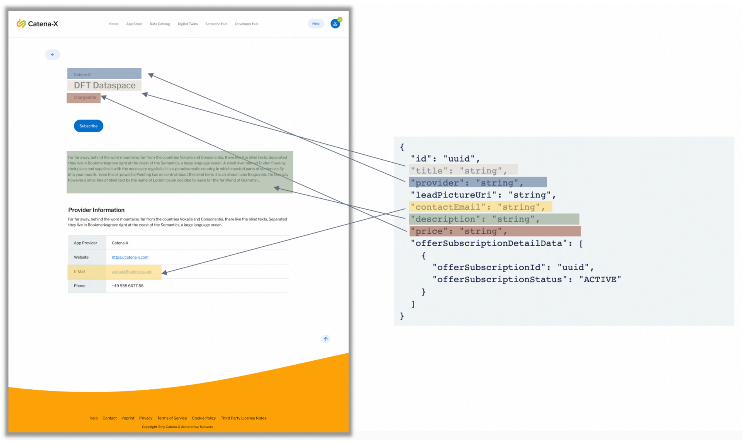
```
[
  {
    "id": "uuid",
    "title": "string",
    "provider": "string",
    "price": "string",
    "leadPictureUri": "string"
  }
]
```

2.3.2 #2 Get Service Offer Details

When selecting an offered service, the service details are getting fetched.

Endpoint: Get /api/services/{serviceID}

Data mapping logic:



```
{
  "id": "uuid",
  "title": "string",
  "provider": "string",
  "leadPictureUri": "string",
  "contactEmail": "string",
  "description": "string",
  "price": "string",
  "offerSubscriptionDetailData": [
    {
      "offerSubscriptionId": "uuid",
      "offerSubscriptionStatus": "ACTIVE"
    }
  ]
}
```

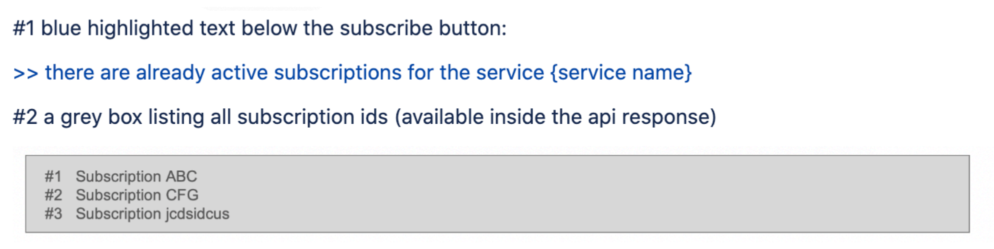
2.3.3 #3 Get Service Order Details

When the user is accessing a service detail page and in case there are already active or pending subscriptions, the subscription status will show up below the "Subscribe" button.

Details regarding active subscriptions will be included in the get service details endpoint.

Endpoint: Get /api/services/{serviceID}

Details on the UI implementation



#1 blue highlighted text below the subscribe button:

>> there are already active subscriptions for the service {service name}

#2 a grey box listing all subscription ids (available inside the api response)

```
#1  Subscription ABC
#2  Subscription CFG
#3  Subscription jcdsidcus
```

2.4 Implementation - Service Subscription

Julia Jeroch to be added

2.4.1 #1 Post Subscription Request - ONGOING

.....

Endpoint: POST /api/services/{serviceId}/subscribe

important, this endpoint also includes the post to the third party. We need to test this jointly asap.

Details to the implemented logic can get found here:

Service Autosetup Interface

2.4.2 #2 Get Service Agreements

When the user is requesting to subscribe a service, the service agreements show up which the user needs to "confirm".

Via api, the agreements will get fetched from the backend to display them to the user.

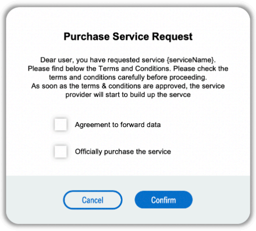
Endpoint: GET /api/services/serviceAgreementData

Data mapping

api example response

```
"agreements": [
    {
      "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "name": "Agreement to forward data"
    },
    {
      "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "name": "Officially purchase the service"
    }
   ]
```

With this example body, the overlay would show up as displayed below:



Purchase Service Request

Dear user, you have requested service {serviceName}.
Please find below the Terms and Conditions. Please check the terms and conditions carefully before proceeding.
As soon as the terms & conditions are approved, the service provider will start to build up the servce

☐ Agreement to forward data

☐ Officially purchase the service

2.4.3 #3 Post Subscription Consent - ONGOING

.....

Endpoint:

2.4.4 #5 Activate Subscription & Tenant - Not UI supported

For all tenant based services / apps, an endpoint is created where app/service provider can active the customer subscription by creating the relevant auth client and enables the customer to assign app roles to their users and login via SSO into the service.

Additionally the interface will send a technical user with credentials to the respective service provider. With those credentials, the service provider can enable the AAS registration, DAPS and create the Self Description.

Endpoint: POST /api/services/autoSetup

CPLP-1212 - Service Provider Marketplace Service CLOSED

CPLP-1209 - Service Provider Marketplace UI CLOSED