App Subscription Function

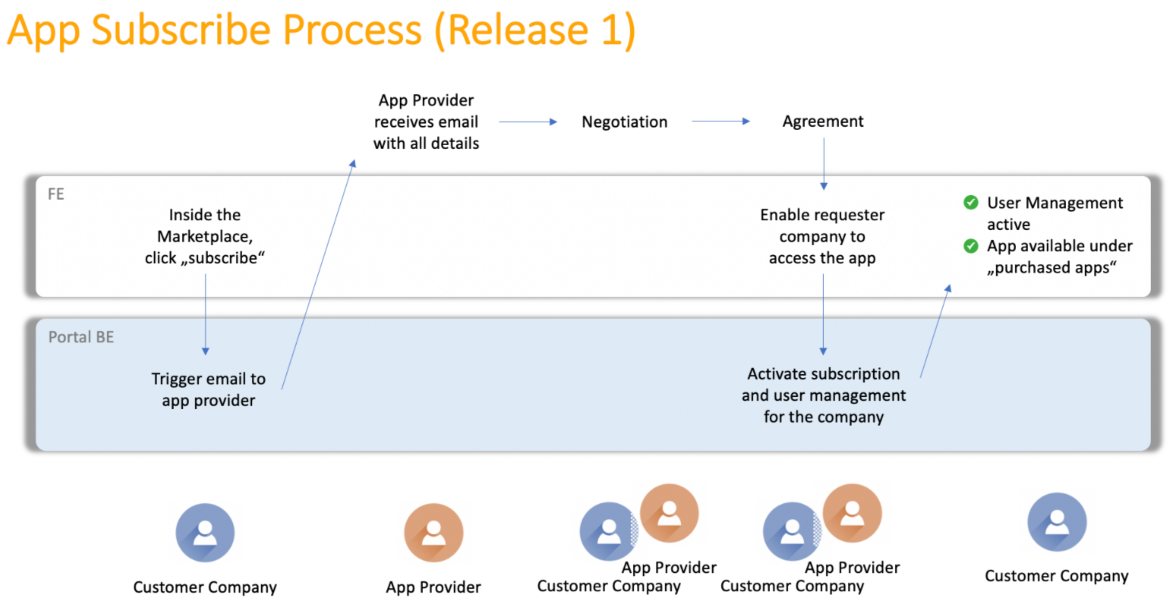
CUSTOMER APP PROVIDER

Functionality

The app subscription process is essential for the usage of an business application, no matter if an app is right away usable or any pre-requisites are given, the app subscription process will manage the relationship between interested app user and app provider.

In the drawing below, the initial process is drafted. Important: this is a draft only and relevant for the initial implementation. As soon as this process is enabled an enhancement is planned which should enable to whole app provider / customer communication on the CX platform.

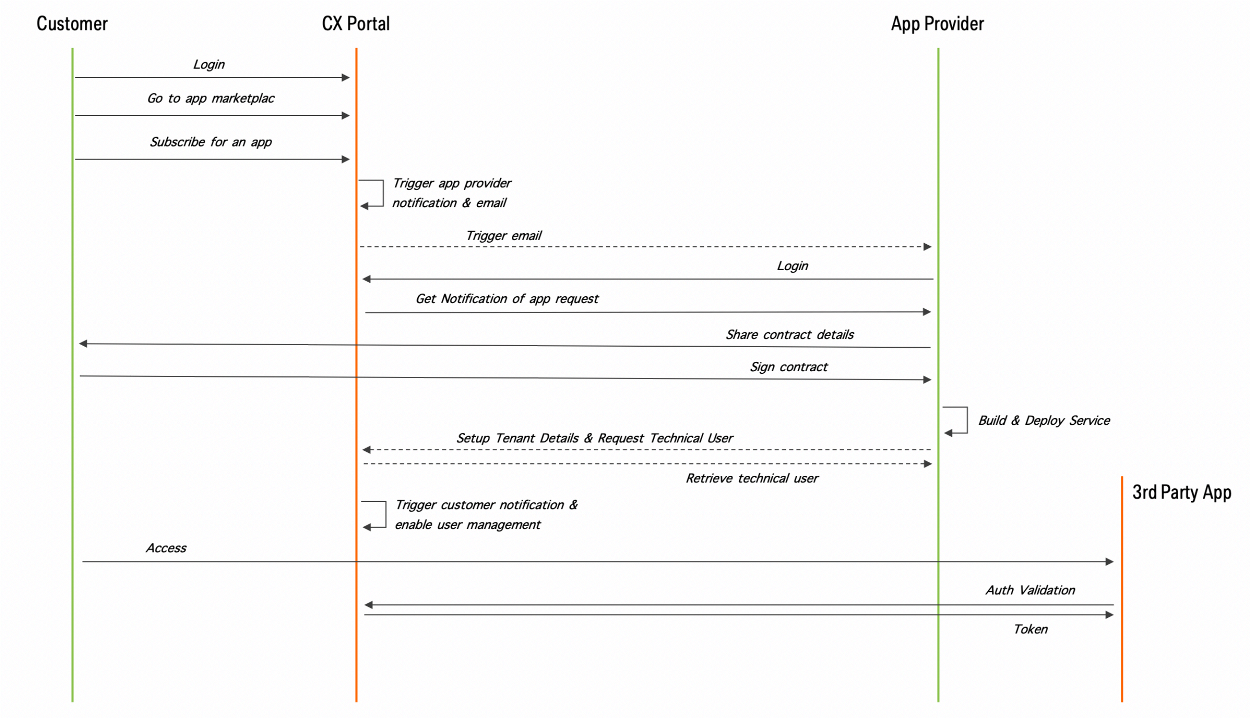
App Subscription Journey



App Subscribe Process (Release 1)

FE
Inside the Marketplace, click „subscribe“
Portal BE
Trigger email to app provider
App Provider receives email with all details
Negotiation
Agreement
Enable requester company to access the app
Activate subscription and user management for the company
✓ User Management active
✓ App available under „purchased apps“
Customer Company
App Provider
App Provider
Customer Company
App Provider
Customer Company
Customer Company

App Subscription Flow



Important flow chart details: the app subscription process does not handle the contract workflow inside the portal, instead the contract agreement is planned to take place outside the marketplace/portal

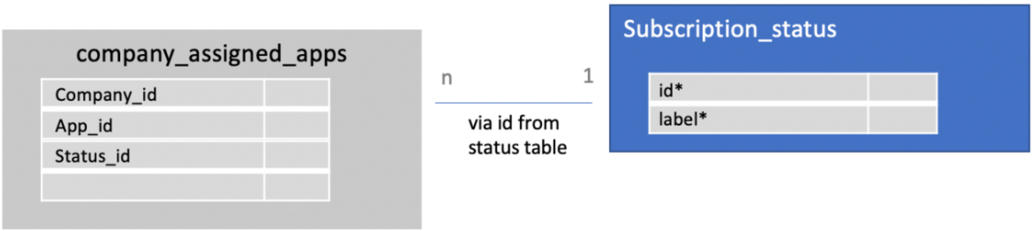
After the successful subscription (request by customer and activation by provider) the user management will get enabled for the respective customer company.

With that, the customer can manage user roles for their own users (pre-requisite: those users need to have an active cx account)

Implementation

DB implementation

To support the app subscription status, the portal db model was enhanced by a new table "Subscription_status" and the status attribute which got added inside the company_assigned_apps table



#1 POST Subscribe Request

The post subscribe request is getting triggered by the service user company. Means the company which is interested to use the app/service is calling out the interest to the service provider.

API: /api/apps/{appId}/subscribe

[Request path parameter]

-
  - appId

[Response body]

-
  - Success
  - Error
    - Service unavailable
    - Subscription already existing
    - AppId not existing

        - missing permission

WHAT - API business logic / implementation logic details

With the POST api, the backend service will

-
    - create a record inside the company_assigned_apps table
    - status_id for the record will get set to "PENDING"

*permission: "subscribe_apps"

#2 PUT Subscribe Activation

The put subscription activation is.........

subscribe request is getting triggered by the service user company. Means the company which is interested to use the app/service is calling out the interest to the service provider.

API: /api/apps/{appId}/subscription/company/{companyId}/activate

Request path parameter

-
    - appId
    - companyId

Response

-
    - Success
    - Error
        - Missing Permission
        - Missing Input Values

        - Status incorrect
        - AppID not existing

WHAT - API business logic / implementation logic details

With the PUT api, the backend service will

-
    - First Validation
        - Check if the AppID is a valid AppID
            - if yes, proceed
            - if no, stop - error "AppID doesn't exist"
        - Check if user who is calling the API belongs to the same company as the APP for which the update should get made
            - If yes, proceed
            - if no, stop - error "Missing Permission"
        - Check if the subscriptionCompanyId and appId is having an existing record in the company_assigned_apps table in status "PENDING"
            - if yes, proceed
            - if no, stope - error "No pending subscription existing"
    - Second functionality (if all validations have been successfully, proceed with the functional flow)
        - update the status of the 'company x appId' record insight the table company_assigned_apps to "ACTIVE"

Permission: "activate_subscription"

#3 Get View Subscriptions Customer

Get Subscription Customer Endpoint is used to receive all the subscriptions which the company has subscribed for.

Subscriptions with status PENDING, ACTIVE and INACTIVE will get distracted.

API: /api/apps/subscribed/subscription-status

Request Body

n/a

Respond Body

```
[
  {
    "appId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "appSubscriptionStatus": "PENDING"
  }
]
```

#4 Get View Subscription Provider

Get Subscription Provider Endpoint is used to receive all subscriptions for any of my apps.

Subscriptions with status PENDING, ACTIVE and INACTIVE will get distracted.

API: /api/apps/provided/subscription-status

Request Body

n/a

Respond Body

```
[
  {
    "appId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "companySubscriptionStatuses": [
      {
        "companyId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "appSubscriptionStatus": "PENDING"
      }
    ]
  }
```

]

#4 Put Inactivate App Subscriptions

Get Subscription Provider Endpoint is used to receive all subscriptions for any of my apps.

Subscriptions with status PENDING, ACTIVE and INACTIVE will get distracted.

API: /api/apps/{appId}/unsubscribe

Request Body

n/a

Respond Body

success or error

Important Links