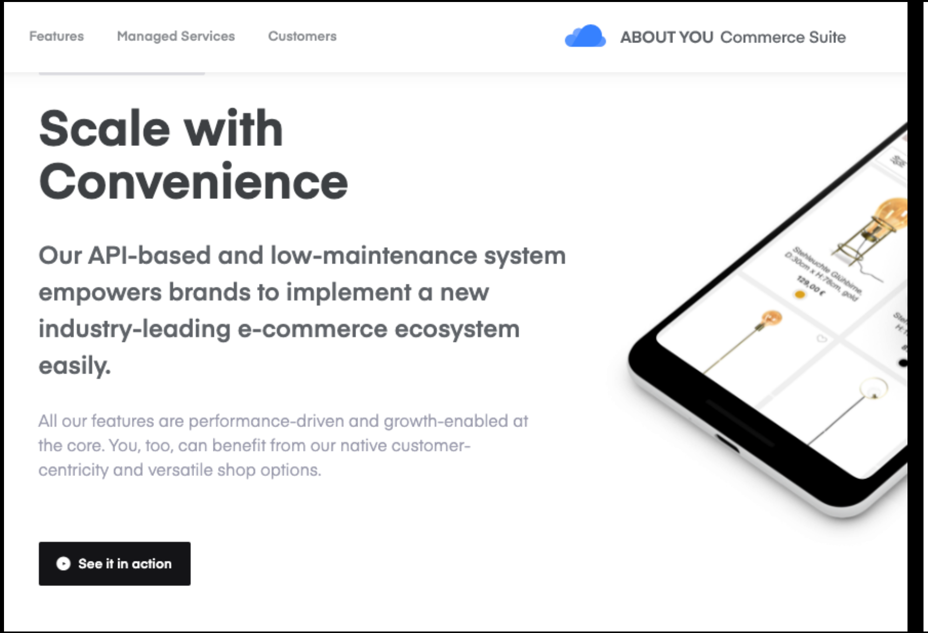
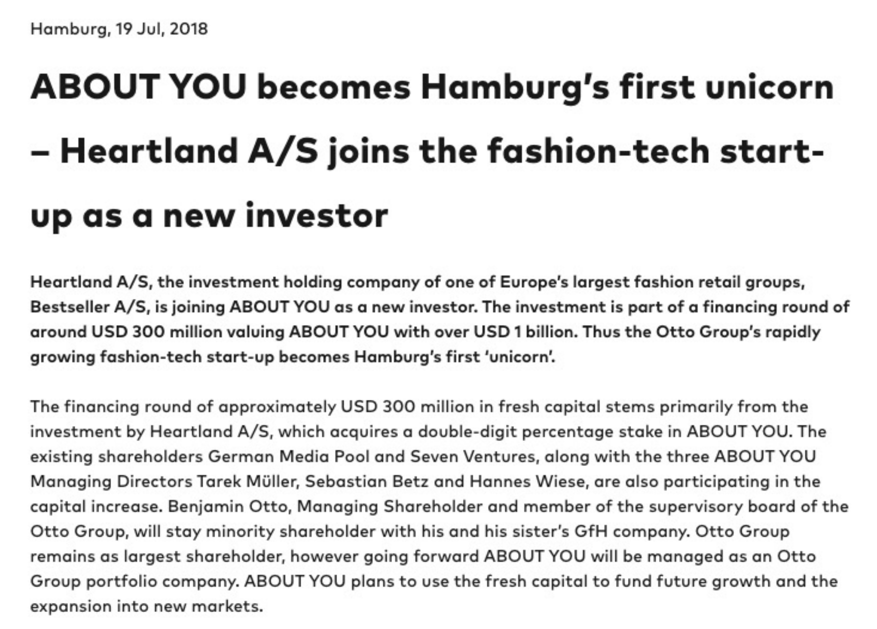
APIS ARE THE KEYSTONE OF SUCCESSFUL E-COMMERCE PLATFORMS



Scale with Convenience

Our API-based and low-maintenance system empowers brands to implement a new industry-leading e-commerce ecosystem easily.

All our features are performance-driven and growth-enabled at the core. You, too, can benefit from our native customer-centricity and versatile shop options.

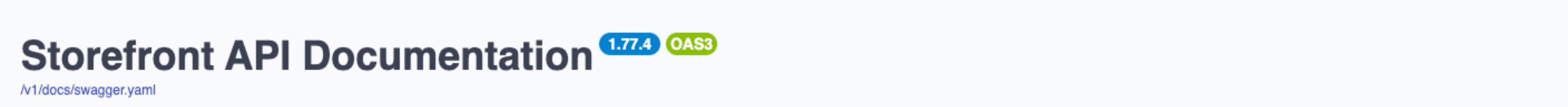


Hamburg, 19 Jul, 2018

ABOUT YOU becomes Hamburg's first unicorn – Heartland A/S joins the fashion-tech start-up as a new investor

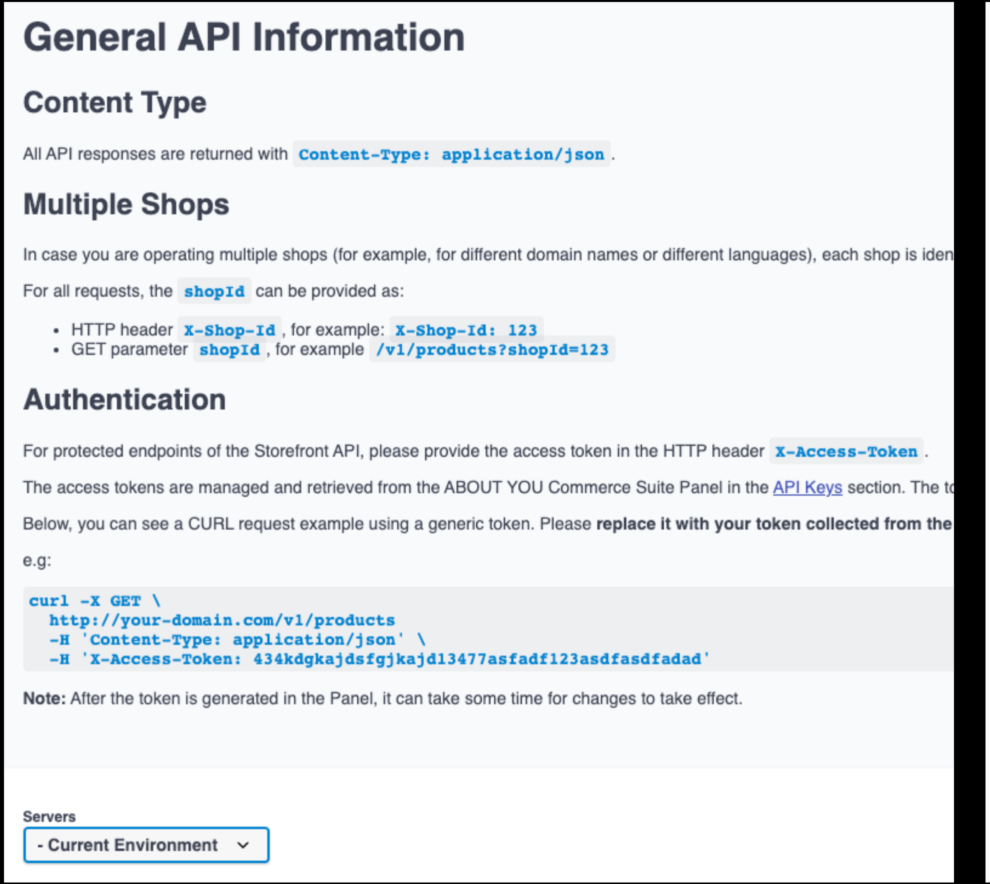
Heartland A/S, the investment holding company of one of Europe's largest fashion retail groups, Bestseller A/S, is joining ABOUT YOU as a new investor. The investment is part of a financing round of around USD 300 million valuing ABOUT YOU with over USD 1 billion. Thus the Otto Group's rapidly growing fashion-tech start-up becomes Hamburg's first 'unicorn'.

The financing round of approximately USD 300 million in fresh capital stems primarily from the investment by Heartland A/S, which acquires a double-digit percentage stake in ABOUT YOU. The existing shareholders German Media Pool and Seven Ventures, along with the three ABOUT YOU Managing Directors Tarek Müller, Sebastian Betz and Hannes Wiese, are also participating in the capital increase. Benjamin Otto, Managing Shareholder and member of the supervisory board of the Otto Group, will stay minority shareholder with his and his sister's GfH company. Otto Group remains as largest shareholder, however going forward ABOUT YOU will be managed as an Otto Group portfolio company. ABOUT YOU plans to use the fresh capital to fund future growth and the expansion into new markets.



Storefront API Documentation

/v1/docs/swagger.yaml



General API Information

Content Type

All API responses are returned with `Content-Type: application/json` .

Multiple Shops

In case you are operating multiple shops (for example, for different domain names or different languages), each shop is iden

For all requests, the `shopId` can be provided as:

- HTTP header `X-Shop-Id` , for example: `X-Shop-Id: 123`
- GET parameter `shopId` , for example `/v1/products?shopId=123`

Authentication

For protected endpoints of the Storefront API, please provide the access token in the HTTP header `X-Access-Token` .

The access tokens are managed and retrieved from the ABOUT YOU Commerce Suite Panel in the API Keys section. The to

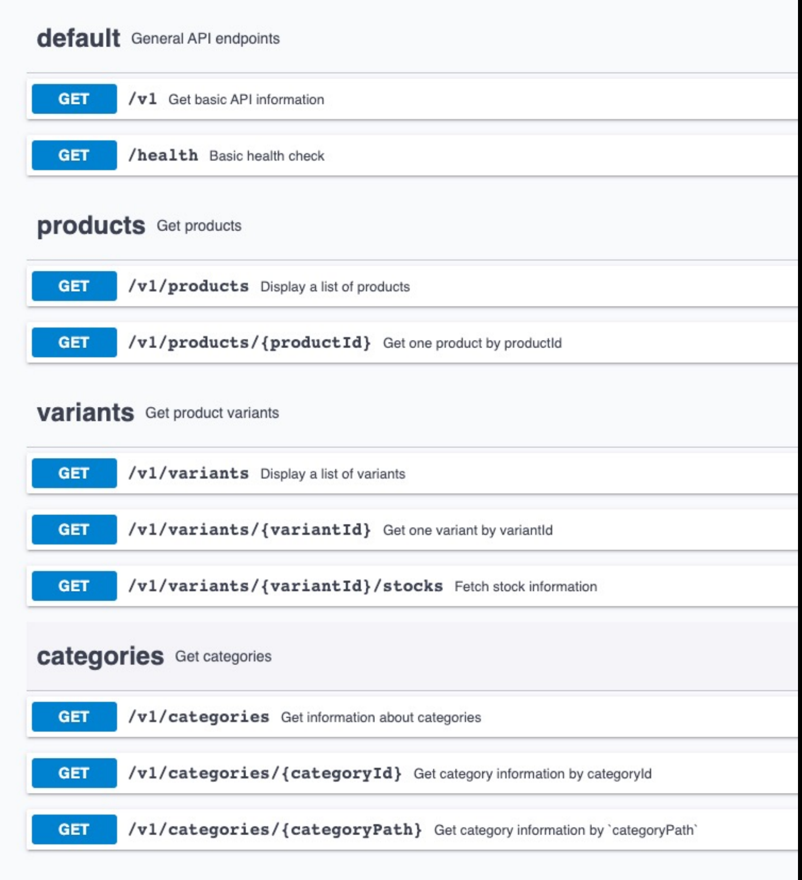
Below, you can see a CURL request example using a generic token. Please **replace it with your token collected from the**

e.g:

```
curl -X GET \
  http://your-domain.com/v1/products
  -H 'Content-Type: application/json' \
  -H 'X-Access-Token: 434kdgkajdsfgjkajd13477asfadf123asdfasdfadad'
```

Note: After the token is generated in the Panel, it can take some time for changes to take effect.

Servers



default General API endpoints

GET `/v1` Get basic API information

GET `/health` Basic health check

products Get products

GET `/v1/products` Display a list of products

GET `/v1/products/{productId}` Get one product by productId

variants Get product variants

GET `/v1/variants` Display a list of variants

GET `/v1/variants/{variantId}` Get one variant by variantId

GET `/v1/variants/{variantId}/stocks` Fetch stock information

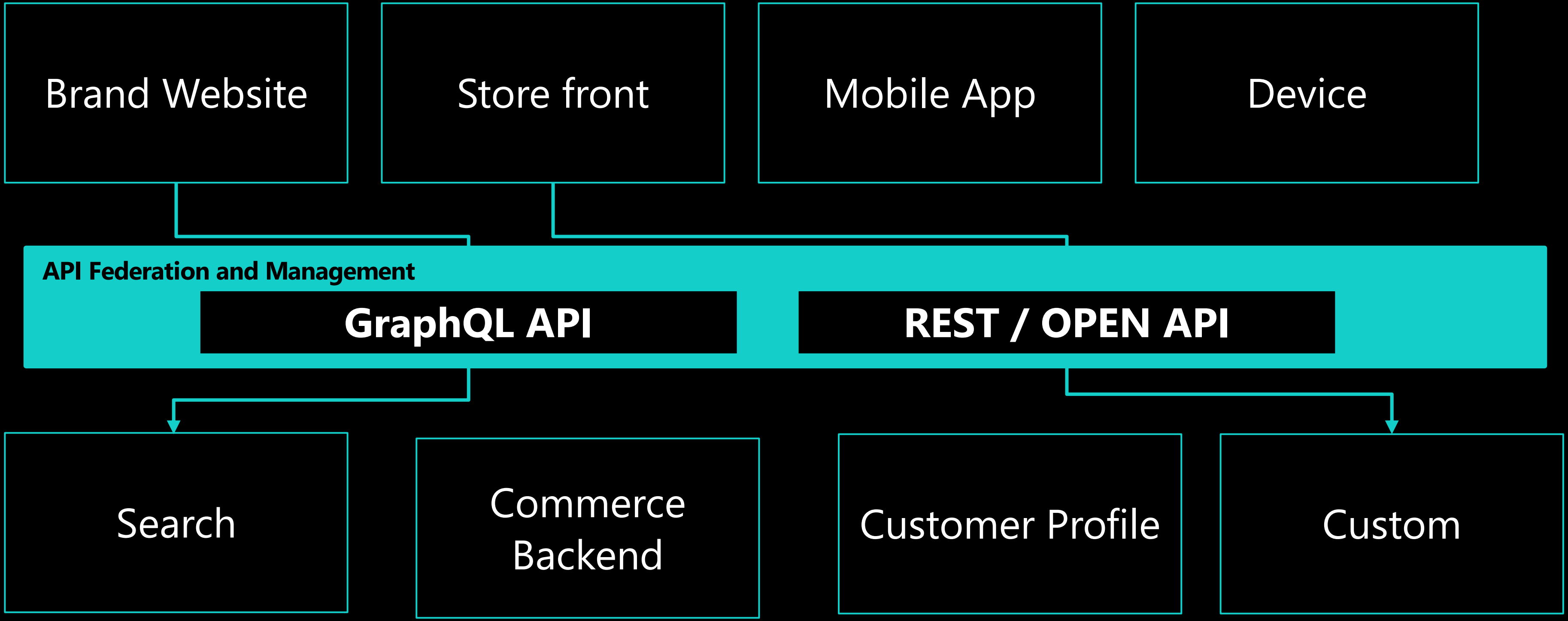
categories Get categories

GET `/v1/categories` Get information about categories

GET `/v1/categories/{categoryId}` Get category information by categoryId

GET `/v1/categories/{categoryPath}` Get category information by `categoryPath`

Modern API Integration Architecture for Omnichannel (“North-South”)



General API Guidelines

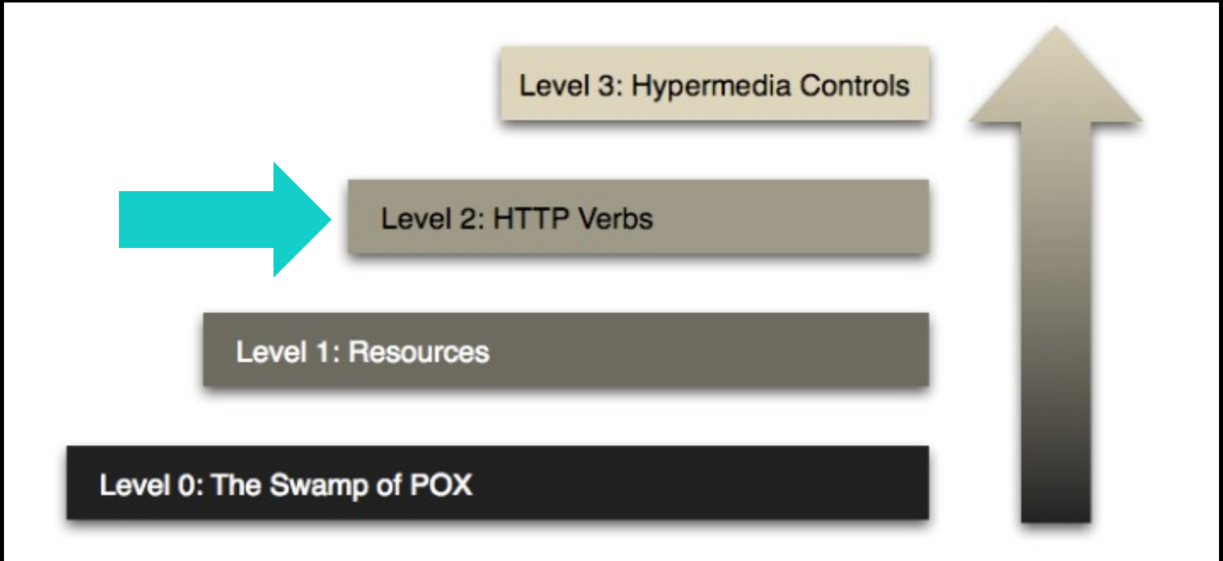
- We aim for an API layer as a **clearly defined boundary** to our platform to be accessed from various front-ends (Web/AEM, mobile, devices, …) and external applications
- The APIs that we expose **MUST** follow **common agreed standards**
- **Vendor-specific APIs MUST NOT be exposed** (outside their own application portfolio), i.e., all APIs of commercial-of-the-shelf applications can only be reached through an anti-corruption-layer/wrapper so that we are more flexible to exchange backend systems but still can fulfil our API contract
- APIs **MUST NOT** break the contract with their clients with a new version
- All APIs that are exposed to the outside (incl. 1st-party clients) **MUST** be reviewed and approved by RAQN **API Governance** Body

API Standards

- **REST-APIs** follow the OpenAPI Standard (link)
- REST-APIs SHOULD follow Microsoft's Open Source API Guidelines: link
- **GraphQL-APIs** follow a **federated approach** so that all relevant query APIs (e. g. Search, Products, Recommendations) can be reached through an enterprise graph
- RAQN Event Hub implements primarily two standards:
  - **Cloud Events (link)**
  - **AsyncAPI (link)**
- **Data Elements are English, lowerCamelCase and UTF-8 encoded**
- **Multi-tenancy** is handled with header variable "X-Site-ID"
  i.e. every API request should have the "X-Site-ID" header included. A central API is provided to register and resolve hostnames to site-IDs: GET /site-id?hostname=frag-team-clean.de

REST-API – Maturity – We aim für Level 2

More details: (link, and link)



Level 2 Key-takeaways:

- Use GET only for operations that do not change state
- To change state, use e.g., POST/PUT
- Use status codes to help communicate errors
- Use query parameters for paging, searching, filtering, and querying
- Use HTTP headers to exchange additional non-payload information regarding the request itself

REST-API Standards - Methods

Method	Description	Is Idempotent
GET	Return the current value of an object	True
PUT	Replace an object, or create a named object, when applicable	True
DELETE	Delete an object	True
POST	Create a new object based on the data provided, or submit a command	False
HEAD	Return metadata of an object for a GET response. Resources that support the GET method MAY support the HEAD method as well	True
PATCH	Apply a partial update to an object	False
OPTIONS	Get information about a request; see below for details.	True

REST-API Standards

- **Versioning**
  - embedded in the path of the request URL, at the end of the service root and should only expose major versions, e.g., v1
  - minor versions MUST not break the API-contract but e.g., could expose more data
- **URI-paths should follow this pattern:**

  /<service>/<version>/<list resources>/<id> e.g., /community/v1/articles/
- **Verbs** (if necessary) should be expressed as …

API Services

- **All APIs are exposed through:**
  - Sandbox-environment: **lab-api.raqn.io**
  - Development-environment (internal): **dev-api.raqn.io**
  - Test-environment (demo): **test-api.raqn.io**
  - Production-environment: **api.raqn.io**

- **Developer portal should be exposed at: developer.raqn.io**

Authentication and Authorization

- All clients that access RAQN APIs MUST **register with API Management**

- **1st parties** MUST use **token-based authentication** with JWT tokens.
  - JWT token validation will be done centrally by the API Management for **inbound traffic**
  - Subsequent API calls must carry the JWT-Token in the „Authorization"-header so that in the future a side-car proxy in the RAQN **service meshes** can validate it too.

- **Mobile apps and 3rd parties** MUST use
  - **OpenIDConnect** for Authentication
  - **OAuth2** for Authorization

1st Party token-based authentication

Login (exceptions are SAP CDC Screen Sets)

1. The request for the user’s authentication is done from the browser to the API Management.
2. API Management will call an Azure function which is already authenticated as a service against CDC to validate the login by API call.
3. The Azure function connects to CDC and validates the authentication
4. This Azure function creates a new JWT token (TTL 10h), including proper scopes and returns it via the API Management to the user where it is stored in the browser as cookie (SameSite-policy).
   Tokens should be stored in REDIS cache to later be able to revoke it.
5. The JWT will be included as Authorization Bearer header in all further requests. Meaning, we don’t use CDC sessions for API calls. CDC is just used for verification of the user’s credentials.

1st Party token-based authentication

Backend Service Calls

1. JWT Bearer is sent as Authorization header for any request to the API Gateway
2. API Gateway validates the JWT token (encryption/signature)
3. API Gateway validates the scopes (optional)
4. API Gateway trims the scopes to the only necessary for the called service
5. API Gateway forwards the request to the Backend Service including the trimmed JWT
6. Backend Service validates the scopes and executes request based on the User-Id & scopes
7. Backend Service might call another backend service with the same JWT token it received (service mesh)
8. Backend Service Return of the response via API Management

Token-revocation

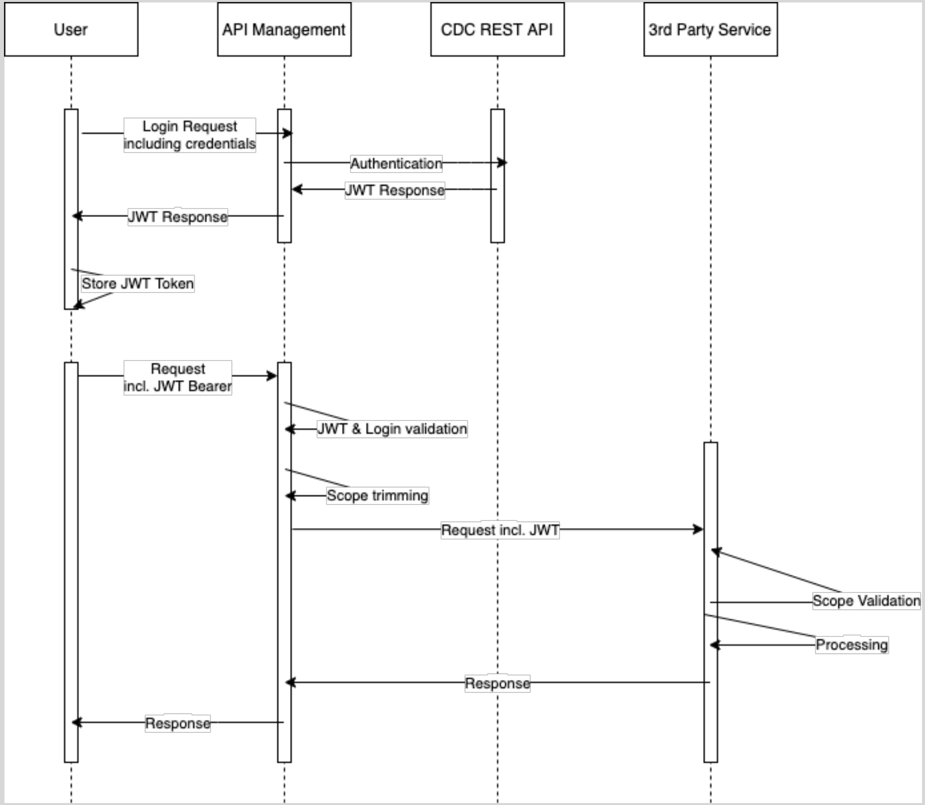
- User should be enabled to revoke token. This should be handled from a UX perspective as a "user log out" (later: w/ " … from all devices" a la Netflix) → submitted token are set invalid in REDIS cache

1st Party token-based authentication

LOGIN

Not implemented yet due to usage of „SAP CDC screen sets"

BACKEND SERVICE CALLS



[CX Works | SAP Customer Data Cloud: Client Side vs. Server Side Integrati