|  |  |
| --- | --- |
|  |  |
| *Open Onderwijs API –Proof of Concept beschrijving* |
|  |
| *SURFnet* |



**Versiebeheer**

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Versie | Commentaar | Auteur |
| 16-12-2015 | 1.0 |  | Jan Blom |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| **Titel** | Proof of Concept beschrijving |
| **Status** | Definitief |
| **Project** | Open Onderwijs API |
| **Datum** | 16-12-2015 |
| **Auteurs** | Jan Blom, Ton Swieb |
| **Document eigenaar** | Ton Swieb |
| **Distributielijst** | Frans Ward (SURFnet) |
| **Goedkeuring** | Frans Ward (SURFnet) |

**Contactpersoon**

|  |  |
| --- | --- |
| **Naam** | Ton Swieb |
| **Functie** | Solution Expert Applicatie Integratie |
| **Kantooradres** | Finalist Rotterdam  Groot Handelsgebouw  Stationsplein 45, A4.205  3013 AK Rotterdam |
| **Telefoonnummer** | 088 217 08 15 |
| **E-mail adres** | [ton.swieb@finalist.nl](mailto:ton.swieb@finalist.nl) |

|  |  |
| --- | --- |
| **Naam** | Jan Blom |
| **Functie** | Software Engineer |
| **Kantooradres** | Finalist Eindhoven  Willemstraat 1L  5611 HA Eindhoven |
| **Telefoonnummer** | 088 217 08 00 |
| **E-mail adres** | jan.blom@finalist.nl |

**Inhoudsopgave**

1.1. Doelstelling 5

1.2. Componenten / Rollen 6

1.2.1. InHolland OpenOnderwijs API (Hogeschool Leiden) 6

1.2.2. SURFconext 6

1.2.3. Identity Provider (IDP, Hogeschool Leiden) 6

1.2.4. JBoss ApiMan 6

1.2.5. JBoss Keycloak 6

1.3. Technologie stack 6

1.4. Flows 7

1.4.1. Eerste gebruik van de API 7

1.4.2. Volgend gebruik van de API 7

1.5. 7

1.6. 8

1.7. URLs en toegangsgegevens 8

## Doelstelling

Doelstelling is middels een Proof of Concept (PoC) met een aan SURFconext gekoppelde omgeving waarin Apiman & Keycloak worden gebruikt, aan te kunnen tonen dat persoonlijke data op basis van de SURFconext ID van een gebruiker van de Hogeschool Leiden te ontsluiten is op basis van de Open Onderwijs API en te beveiligen is op basis van OAuth middels een standaard configuratie. Deze PoC zal inzicht geven in de mogelijkheden door het aanbieden van:

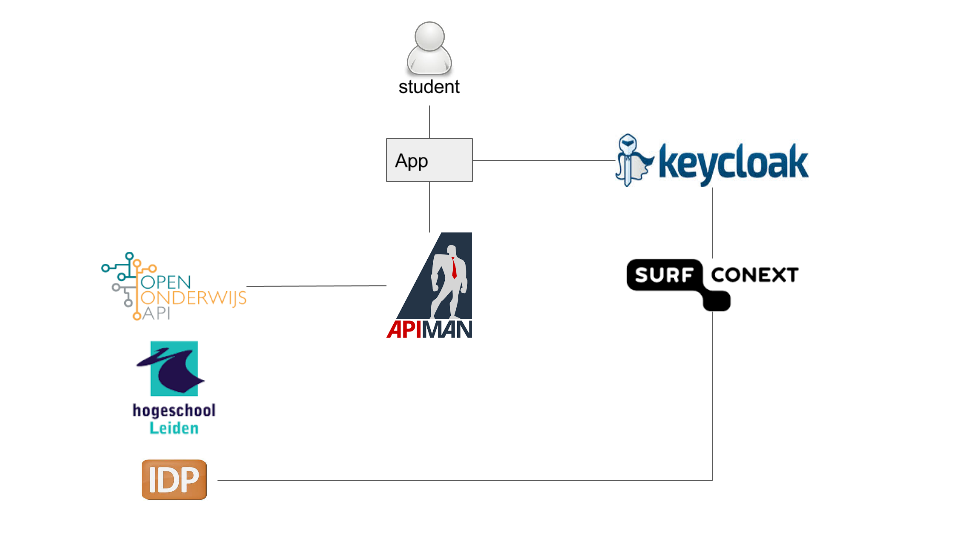
* een centrale omgeving waarin beschikbare API’s beheerd kunnen worden;
* SURFconext als federatieve authenticatie-omgeving, waardoor alle aangesloten onderwijsinstellingen van deze omgeving gebruik kunnen maken, en applicaties/apps makkelijk bruikbaar gemaakt kunnen worden voor meerdere onderwijsinstellingen;
* toegangstokens die een langere geldigheid kunnen bieden dan de sessie met SURFconext, zodat gebruikers niet genoodzaakt worden om frequent opnieuw in te loggen.

Binnen deze PoC zal een service voor het raadplegen van persoonlijke data, op basis van de OpenOnderwijs API, beschikbaar worden gesteld door Hogeschool Leiden, zal SURFnet SURFconext beschikbaar stellen als authenticatieplatform, en zal Finalist op basis van Apiman en Keycloak deze componenten integreren tot een demonstreerbaar werkende implementatie. Voor de implementatie wordt door SURFnet een server (VM) beschikbaar gesteld.

Het is de bedoeling dat zowel persoonlijke, niet publieke data kan worden opgevraagd waarop personalisatie op basis van de gebruikersidentiteit kan worden toegepast, als publieke data, waarvoor men niet ingelogd hoeft te zijn.

Personalisatie in de resultaten van API calls is afhankelijk van de achterliggende systemen. Zodra de achterliggende systemen ook overweg kunnen met personalisatie, kunnen deze ook worden getoond in de demo applicatie.

## Componenten / Rollen



### InHolland OpenOnderwijs API (Hogeschool Leiden)

Deze API biedt toegang tot persoonlijke gegevens op basis van de OpenOnderwijs API.

### SURFconext

Federatieve Identity Provider. Authentiseert gebruikers voor toegang tot de InHolland API door middel van authenticatie bij een aangesloten instelling.

### Identity Provider (IDP, Hogeschool Leiden)

Identity provider van onderwijsinstelling.

### JBoss Apiman

JBoss Apiman is de API management omgeving. Dient als proxy voor de InHolland API, en als Identity Broker. Gebruikt SURFconext als Identity Provider. Schermt de InHolland API af, en geeft alleen toegang aan deze API voor gebruikers die een geldig token hebben. Delegeert toegangsverzoeken zonder geldig token aan Keycloak.

Apiman vormt de gateway tussen de client applicatie en de beschikbare API’s.

In Apiman zijn een aantal gepubliceerde services beschikbaar. Deze services bevatten policies ten aanzien van beveiliging. Een van deze policies controleert of de API aanroep een bearer token bevat, en valideert dat token. Als het bearer token valide is dan wordt het userid eruit gehaald en met de API call doorgegeven naar het backend systeem.

In Apiman is het backend systeem zodanig geconfigureerd dat er wordt ingelogd met basic authentication, en dat er een subscription key wordt meegegeven voor Microsoft Azure.

Apiman roept dus feitelijk de Microsoft Azure omgeving aan met basic authentication en een extra header met daarin de userid van de geauthentiseerde gebruiker, en een header met daarin de subscription key vereist door Microsoft Azure.

### JBoss Keycloak

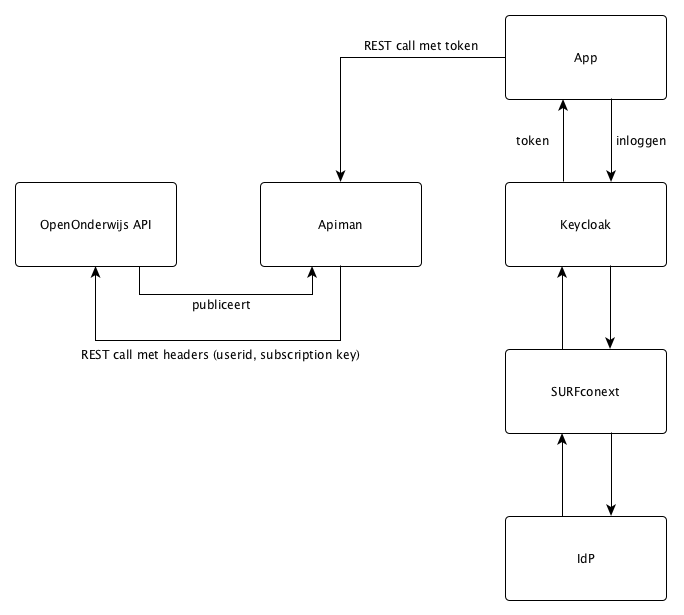
Fungeert als identity broker voor JBoss Apiman. Wanneer een afgeschermde service wordt aangeroepen en geen geldig token wordt meegestuurd, zal Apiman laten authentiseren via Keycloak. Op basis van de authenticatie verschaft Keycloak dan een geldig token.

In Keycloak worden realms gedefinieerd. Instellingen zoals de geldigheidsduur van een token worden per realm geconfigureerd.

## Technologie stack

Apiman en Keycloak draaien in een Wildfly applicatie server. Deze server is ondergebracht in een Docker container. Het bouwen en configureren van deze container is volledig gescript, en omgevingsonafhankelijk gemaakt, waardoor dezelfde set van scripts gebruikt kan worden om de container via een beperkte set parameters te bouwen voor de gewenste omgeving (development, test, acceptatie, productie). De container kan gedeployed worden op platforms die Docker ondersteunen.

## Flows



### Eerste gebruik van de API

Voor toegang tot de API is een access token vereist. De toegang tot de API verloopt via Apiman, die naar Keycloak zal verwijzen voor authenticatie. Keycloak laat de authenticatie door SURFconext uitvoeren, die op zijn beurt de gebruiker de keuze biedt uit een aantal identity providers. Na authenticatie zal Keycloak een access token uitgeven. Dit bevat onder meer de userid van de geauthentiseerde gebruiker.

### Volgend gebruik van de API

De API wordt benaderd met een access token uitgeven. Op basis van een geldig access token zal Apiman toegang verlenen tot de API. Daarbij worden headers meegestuurd die de userid bevatten, en de voor de API benodigde subscription key.

## URL’s en toegangsgegevens

De wachtwoorden die hier staan zijn de wachtwoorden die worden gebruikt bij het bouwen/deployen van de Docker image. Waarschijnlijk wijken de wachtwoorden op de acceptatieomgeving hiervan af, deze worden dan afzonderlijk gedeeld met degenen die toegang nodig hebben.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | **URL** | **Gebruikers naam** | **Wacht woord** | **Toelichting** |
| Keycloak | <https://apiman.openonderwijsapi.nl:7443/auth/admin/> | admin | admin123! | Keycloak is standaard gebundeld met de Apiman distributie, maar kan ook los geïnstalleerd worden. |
| Apiman UI | <https://apiman.openonderwijsapi.nl:7443/apimanui/> | admin | admin123! |  |
| JS-Console | <https://apiman.openonderwijsapi.nl:7443/js-console/> | n.v.t. | n.v.t. | Javascript Client voor het testen van de JBoss Apiman / JBoss Keycloak setup. |
|  |  |  |  |  |