ZGW API Authenticatie en Autorisaties

De standaard heeft een duidelijke scheiding in authenticatie en autorisaties:

- Authenticatie valt expliciet buiten de scope van de standaard. Het is volledig aan de gemeente om te bepalen hoe de authenticatie plaats vindt. Wel wordt beschreven wat het resultaat moet zijn na authenticatie. Het doel hiervan is dat na authenticatie de autorisaties gecontroleerd kunnen worden.
- Autorisaties worden in de AC geregistreerd en ontsloten.

De standaard beschrijft authenticatie en autorisaties op het gebied van systemen. Autorisaties op het niveau van gebruikers moeten in de taakapplicaties geregeld worden.

Het doel van authenticatie is om vast stellen wat de identiteit van de gebruiker is. Het is aan de gemeente om te bepalen hoe dit gedaan moet worden. Het eindresultaat staat wel vast. Er moet namelijk een JSON Web Token (JWT) aangemaakt worden dat in de communicatie met de AGW APIs gebruikt moet worden. De JWT bevat primair:

1. Client ID: een unieke identificatie van de client. Client ID kan aangevuld worden met client naam voor leesbaarheid.
2. Signature: een signature om te controleren dat JWT uitgegegeven is door een trusted partij. Bijv. de autorisatie systeem die de gemeente kiest.

Bij elke request die de afnemers sturen moet de JWT meekomen. Dit stelt de componenten uit de ZGW API in staat om de autorisaties te controleren.

Authenticatie

De BigIP F5 van de gemeente Rotterdam speelt een centrale rol in de communicatie met de ESB. Naast SSL offloading kan de F5 ook JWT's genereren naar de achterliggende ESB endpoints. Dat gaat als volgt:

1. De taakapplicatie krijgt een certificaat toegewezen.
2. De taakapplicatie moet dit certificaat aanbieden als het de ZGW API's aanspreekt. De ZGW API's op de ESB hangen achter BigIP F5.
3. F5 vang de request af en controleert het certificaat. Indien valide zal de F5 een JWT aanmaken met een clientID dat afgeleid wordt van het certificaat en een signature.
4. De F5 laat de request door richting de ESB met de JWT in de header.
5. De ESB ontvangt de request en leest de clientID uit de JWT en valideert de signature.

Vanaf dit punt is de identiteit van het aanroepende systeem bekend. Bovenstaande zaken zijn nog niet in de F5 aanwezig en moeten nog geïmplementeerd worden.

Autorisaties

Omdat de Client ID in elke request aanwezig is kan elk component controleren wat de rechten van die Client ID is. Dat gebeurt door de AC te bevragen met de Client ID als parameter. Een voorbeeld van het antwoord van AC is als volgt:


```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "url": "https://autorisaties-api.vng.cloud/api/v1/applicaties/fe7b0ee0-69a8-4bcd-bd7b-9238f64886be",
      "clientIds": [
        "test_id1",
        "test_id2"
      ],
      "label": "Test applicatie",
      "heeftAlleAutorisaties": false,
      "autorisaties": [
        {
          "component": "zrc",
          "componentWeergave": "Zaakregistratiecomponent",
          "scopes": [
            "zaken.lezen"
          ],
          "zaaktype": "https://catalogi-api.vng.cloud/api/v1/zaaktypen/f9e96031-bb25-4fd0-9b3f-41a3bcd2fc0a",
          "maxVertrouwelijkheidaanduiding": "openbaar"
        }
      ]
    }
  ]
}
```

Bovenstaande geeft aan dat deze client zaken van het type "f9e96031-bb25-4fd0-9b3f-41a3bcd2fc0a" (identificatie van het zaaktype) kan lezen. Zoals je ook kan zien heeft deze applicatie meerdere Client IDs.

eSuite implementatie

De eSuite implementatie bevat een implementatie van de AC component. We hebben hier echter 2 opties:

1. De JWT doorzetten naar de componenten van eSuite.
2. Een centrale AC component neerzetten voor gebruik en beheer door Rotterdam. Een JWT sturen naar eSuite die de ESB als clientID bevat.

We behandelen de opties hierna.

JWT doorzetten naar eSuite

Als de JWT van de taakapplicatie doorgezet wordt naar de eSuite componenten (bijv. ZRC, ZTC, DRC) moeten deze componenten volgens de standaard de autorisaties controleren. Dat betekent dat de ZRC van eSuite naar de AC van eSuite zal gaan om te controleren dat de clientID van de taakapplicatie de correcte autorisaties heeft. Deze opzet heeft de volgende eigenschappen:

1. eSuite zal de signature van de F5 moeten kunnen controleren. De JWT dat dorgezet is naar eSuite is immers daarmee ondertekend.
2. De clientID's van de taakapplicaties moeten opgevoerd worden in de eSuite AC.
3. Nieuwe aansluitingen op de ESB (clientID's) moeten in de eSuite AC opgevoerd worden.

Dit is een werkbare situatie maar het heeft wel wat gevolgen die onderzocht moeten worden zoals:

- Is het mogelijk om de signature van de F5 te "vertrouwen" in de AC van eSuite?
- Is het mogelijk voor beheerders om autorisaties te beheren in de AC van eSuite?
- In het geval dat we meerdere zakenmagazijnen zouden ontsluiten is het mogelijk om de AC van eSuite te gebruiken als "main" AC vanuit de andere zakenmagazijn?

Een centrale AC component

Door de AC centraal te stellen kunnen autorisaties over meerdere componenten (ook meerdere zakenmagazijnen) centraal beheerd worden. Rotterdam heeft volledige regie over de component en de inrichting hiervan.

De ESB hoeft in deze opzet geen JWT's door te sturen van taakapplicaties naar eSuite. De interactie met de ESB ziet er dan als volgt uit:

1. Taakapplicatie klopt aan met JWT bij ZRC.
2. De ESB extraheert de clientID en controleert autorisaties bij de centrale AC.
3. Als de autorisaties in orde zijn wordt de request doorgezet naar eSuite. Hierbij wordt de JWT die de taakapplicatie stuurde vervangen met een JWT van de ESB zelf.
4. eSuite ontvangt de request op de ZRC, extraheert de clientID (in dit geval de ESB zelf) en bevraagt de eigen AC omtrent de autorisaties van deze client. De ESB is geautoriseerd op alles en kan dus de actie uitvoeren.

Deze opzet betekent dat de AC component van eSuite alleen door eSuite gebruikt wordt en dat de enige autorisaties aan de ESB clientID uitgedeeld moeten worden op de bekende componenten: ZRC, ZTC en DRC.

Vanwege het modulaire karakter van de componenten in de ZGW API standaard is het mogelijk om de centrale AC apart af te nemen. Hiervoor bestaan er meerdere opties:

1. Inpassen van de Referentie Implementatie in Rotterdam
2. Zelfbouw
3. Aanschaffen kant en klare AC component

De Referentie Implementatie is een snelle manier om up en running te zijn maar zal waarschijnlijk een generiek karakter hebben en mogelijk zijn de beheerschermen moeilijk in te passen in de infrastructuur. Bovendien hebben we nog geen mogelijkheid gevonden om de broncode van de Referentie Implementatie te krijgen.

Zelfbouw is tegenwoordig zeer krachtig en kan op een snelle manier plaats vinden. Hierbij wordt voor de frontend Mendix toegepast en voor de backend Java services op basis van JSON/Rest. Dit biedt zeer veel flexibiliteit voor het bedienen van beheerders van de autorisaties.

Mogelijk bestaat er op de markt een kant en klare AC component waardoor zelfbouw niet nodig is. Mogelijk is de eSuite AC los te trekken en zelfstandig te gebruiken.