Referentie & solution architectuur (WIP)

Nuts SSIBAC

Een solution architectuur voor het implementeren van ABAC gebasseerd op attributen verkregen via SSI. Maakt gebruik van OpenID4VP, DIDs, VCs, FHIR en OPA.

NOTE: vanwege limitaties van aanroepen OPA vanuit andere sources dan nginx en aanroep naar OPA container anders dan nuts-pxp, kiezen we nu (may change) voor een enkele Rego package als root (/v1/data/nuts/allow) die alle andere regels importeert en 1-voor-1 afgaat.

NOTE2: het datamodel moet wellicht volledig JSON object map worden zodat het gedrag van de standaard OPA container hetzelfde is als voor nuts-pxp. Dus scope: {verifier:{client:}}

Introductie

Voor succesvolle gegevens uitwisseling in de zorg zijn een heel aantal (technische) stappen nodig. Deze stappen zijn afhankelijk van afspraken, standaarden en technology. Veel van deze afhankelijkheden vallen onder de generieke functies. Dit document gaat over het autoriseren van een gegevens vraag bij een FHIR endpoint.

Het Nuts initiatief promoot SSI als authenticatie oplossing binnen de zorg. Credentials en attributen verkregen via SSI hebben een hoog betrouwbaarheidsniveau en zijn daardoor uitermate geschikt voor gebruik in Attribute Based Access Control. Voor gegevensuitwisseling in de zorg is het van groot belang dat er hergebruik plaats vindt van technologie en standaarden over meerdere use cases. Als er uitgegaan wordt van authenticatie via SSI en het beschikbaar stellen van medische gegevens via een FHIR API, dan is er een autorisatie & accountability model nodig die de twee met elkaar verbindt. Dit document beschrijft zo'n model in de vorm van een referentie architectuur en geeft een voorbeeld van een implementatie mbv Open Source software.

Access control policy

De basis van autorisatie is een autorisatievraag die door een access control policy (kort: policy) beantwoord moet worden. Zo'n policy is specifiek voor een use case. Verschillende use cases stellen verschillende eisen aan een policy. Een policy is afhankelijk van één of meerdere input parameters (kort: input). We beschrijven hier mogelijke input, niet elke policy maakt altijd gebruik van alle input.

Assertions

Bij het gebruik van SSI zullen er één of meerdere assertions vrijgegeven zijn door de gebruiker. Het access token dat afgegeven is door de autorisatie server is gekoppeld aan deze assertions. Bij de introspectie van het access token in de "authn" stap wordt het access token ingewisseld voor een lijst van attributen. Deze attributen kunnen als input gebruikt worden.

Request

Dit document beperkt zich tot het beschrijven van autorisaties van REST calls. De vraag naar een resource gaat middels een HTTP request (GET /fhir/Patient/4 HTTP/2.0). Deze informatie kan gebruikt worden in de policy.

Externe data

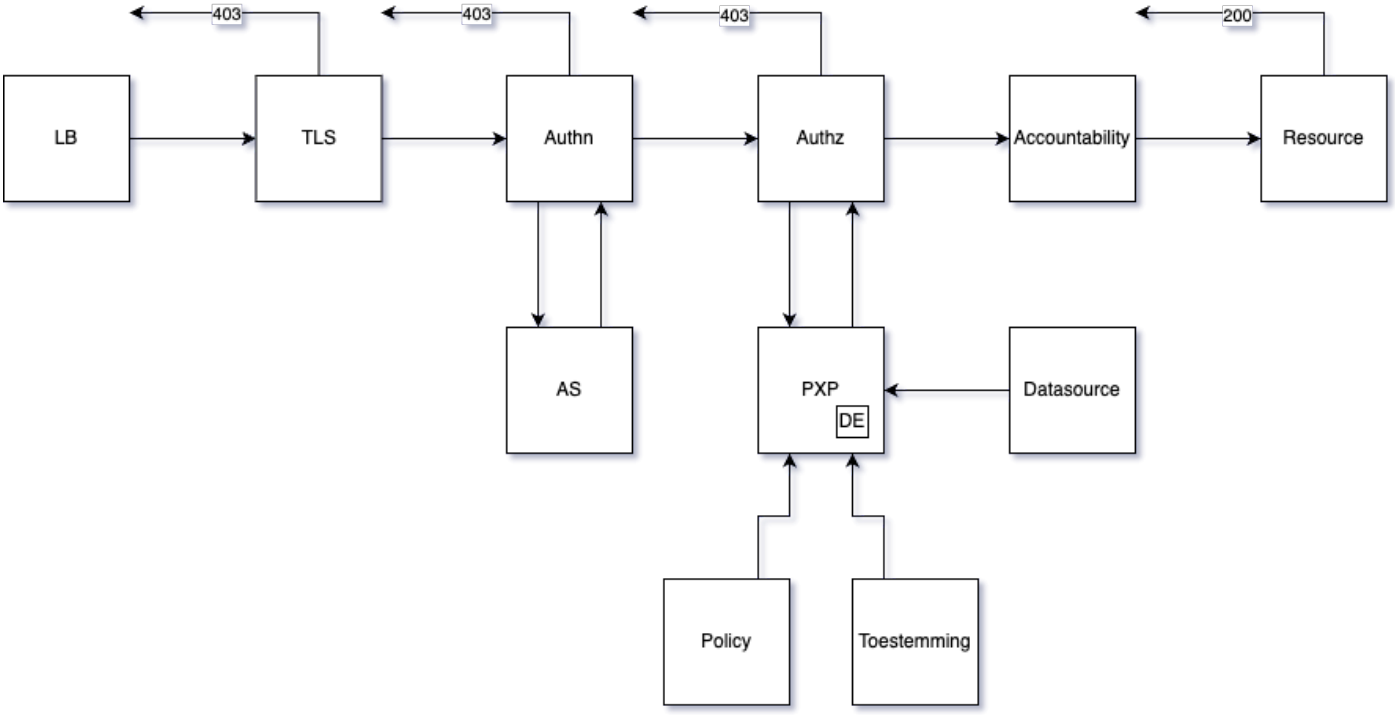
Indien de VC attributen en het request niet voldoende input is om tot een besluit te komen, kan er een afhankelijkheid zijn naar externe data.

Toestemming

In een use case staat beschreven wat de wettelijke grondslag is tot uitwisseling. Indien dit uitdrukkelijke toestemming is dan is deze toestemming input voor de policy.

SSIBAC referentiearchitectuur

Onderstaand model geeft de referentiearchitectuur weer voor SSIBAC.



Een uitleg van de verschillende componenten:

- **LB** - L4 Loadbalancer, verdeelt het verkeer over meerdere machines.
- **TLS** - TLS termination point. Het versleutelde kanaal gaat vanaf hier onversleuteld verder. Eventueel mTLS voor extra authenticatie.
- **Authn** - Controle van access token. Access token wordt bij Authorization Server (AS) ingewisseld voor assertions. Bij ongeldig access token stopt de flow.
- **AS** - Authorization server als uitgever van access tokens.
- **Authz** - Effectueren van autorisatie beslissing. Roept PXP aan voor autorisatiebeslissing. De flow stopt indien niet succesvol.
- **PXP** - Autorisatie component. Verzamelt de benodigde input en maakt een beslissing obv een policy. **DE** is de descision engine voor de policy.
- **Policy** - Getructureerde policy die door de DE uitgevoerd kan worden.
- **Toestemming** - Bron van toestemmingen.
- **Datasource** - Bron voor additionele input voor de PXP.
- **Accountability** - Bij positieve beslissing door PXP zorgt dit component voor de logging van het request.
- **Resource** - Database of service die de gevraagde data oplevert.

Het PXP component is weergegeven als centraal component dat gebruik maakt van meerdere input. Hoe het component toegang heeft tot de input is onderdeel van de solution architectuur.

Use case afspraken

In het kader van herbruikbaarheid is het belangrijk dat alle use cases dezelfde standaarden gebruiken. Zie ook de keuzes van de Nuts SSIBAC solution architecture.

Per use case moet er afgesproken worden welke VCs/assertions er vanuit het authn component komen en welke input er gebruikt wordt.

Relatie tot XACML rollen

In de referentie architectuur worden de PAP, PDP en PIP samengevoegd tot PXP. De authn en authz componenten uitafbeelding de referentie architectuur vervullen samen de PEP rol.

Nuts SSIBAC solution architectuur

Hieronder volgen de keuzes van de Nuts SSIBAC solution architectuur die invulling geven aan de referentie architectuur.

Authenticatie

Voor het verkrijgen van een access token wordt RFC021 of OAuth2 en OpenID4VP gebruikt. De scope parameter in het autorisatie request is per use case gemapt op een Presentation Definition (kort: PD). Deze PD wordt gebruikt in de flows om de juiste Verifiable Presentation te genereren door de client/wallet. Obv de VP geeft de autorisatie server een access token af.

De client voegt het access token toe aan het request via de Authorization http header. Het authn component stuurt de waarde van de header naar de autorisatie server. De AS geeft een introspection result terug volgens RFC7662. In dit resultaat staan iig de volgende attributen: `valid`, `client_id`, `iss` en `scope`. Daarnaast staan er key/value paren in waarbij de keys afkomstig zijn van de PD en de values de bijbehorende waardes uit de gestuurde VCs. De PD en de key/value paren moeten per use case vastgelegd worden in een Scope to Presentation Definition mapping. Alle waarden uit het introspection result kunnen gebruikt worden in de policy.

Het authn component stuurt het introspection result en request data door naar het authz component.

Autorisatie

Het autorisatie component roept het PXP component aan met als input een JSON body:

```
{
  "input": {
    "type": "http",
    "request": {
```

```
      "scheme": "http",
      "method": "GET",
      "host": "pep-right",
      "query": {},
      "path": "/fhir/Patient/4",
      "headers": {
        "X-Userinfo":
"eyJvcmdhbml6YXRpb25fbmFtZSI6IkxlZnQiLCJzY29wZSI6ImVPdmVyZHJhY2h0LXJlY2VpdmVyIiwic3ViIjoiZGlkOndlYjpub2RlLnJpZ2h0LmxvY2FsOmlhbTpyaWdodCIsImV4cCI6MTcxODI5MDE4NiwiaWF0IjoxNzE4Mjg5Mjg2LCJpc3MiOiJkaWQ6d2ViOm5vZGUucmlnaHQubG9jYWw6aWFtOnJpZ2h0IiwiYWN0aXZlIjp0cnVlLCJjbGllbnRfaWQiOiJkaWQ6d2ViOm5vZGUubGVmdC5sb2NhbDppYW06bGVmdCIsIm9yZ2FuaXphdGlvbl9jaXR5IjoiR3JvZW5sbyJ9",
        "host": "pep-right:9080",
        "authorization": "Bearer TonUNXLwVn2UgJgVfpVDNa7WaXAlE2W-mS6CfqDzeP0",
        "content-length": "0",
        "user-agent": "go-resty/2.13.1 (https://github.com/go-resty/resty)",
        "X-Access-Token": "TonUNXLwVn2UgJgVfpVDNa7WaXAlE2W-mS6CfqDzeP0",
        "accept-encoding": "gzip",
        "content-type": "text/plain; charset=utf-8",
        "connection": "close"
      },
      "port": 9080
    }
  }
}
```


Hierbij is het introspection result base64 encoded opgenomen onder `input.request.headers.X-Userinfo`.

Het PXP component antwoord met een JSON object met een enkele waarde `allow` (boolean). Op basis van dit antwoord stuurt het autorisatie component het request door of geeft het een 403 terug.


```
{
  "result": {
    "allow": true
  }
}
```


Het pxp component combineert input met data uit een datasource. Vanwege performance redenen is het verstandig om deze datasource "dicht" bij het pxp component te hebben staan. Daarnaast moet de datasource over alle beschikbare gegevens beschikken voor de policy.

Hoe dit efficient, schaalbaar en eenvoudig te doen is nog onderwerp van discussie.

Accountability

Om te kunnen voldoen aan de NEN7513 (logging) zijn er organisatie en gebruiker gegevens nodig. Een use case moet erop toezien dat die via de Verifiable Credentials beschiklbaar worden gemaakt door de gebruiker. Dit kan via de Presentation Definition. Gevolg is dat de benodigde gegevens in te token introspection beschikbaar zijn. Het authz componenent moet dan ook na de beslissing het token introspection result meesturen naar het accountability component.

Bijlage I - voorbeelden

eOverdracht

Bijlage II - OS ref impl

nginx, nuts-pxp

Refs

https://www.dpss.inesc-id.pt/~mpc/pubs/SSIBAC__Self_Sovereign_Identity_Based_Access_Control.pdf

---

Revision #6
Created 31 May 2024 07:56:39 by Wout Slakhorst
Updated 25 September 2024 12:05:21 by Wout Slakhorst