

Uitwerking RFC0470: Plaats stuurgegevens in SOAP header ipv payload

De volgende stelling is als uitgangspunt genomen:

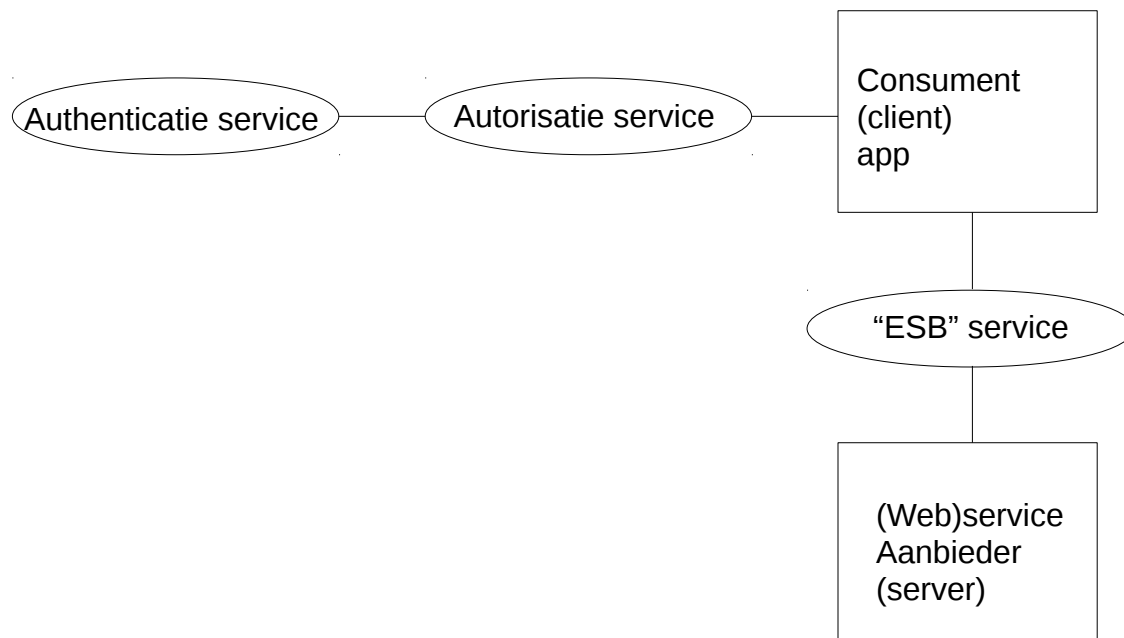
"Het gebruik van stuurgegevens in de "payload" van berichten is in strijd met de geest van de wereldwijde standaarden. Haal routeringsinformatie uit de payload."

De stelling

"Wel is gesteld dat een StUF bericht zelf moet vertellen wat er met hem moet gebeuren zonder dat er tussen de end-to-end points gerommeld wordt met de stuurgegevens." is niet altijd mogelijk.

Dit zou kunnen gelden als een consument applicatie zelf authenticatie/autorisatie regelt en daarna het StUF bericht samen stelt.

Steeds meer zien we dat er een authenticatie service en/of een autorisatie service wordt gebruikt. Deze services zullen de authenticatie/autorisatie "meegeven" aan de applicatie, die eventueel naast het bericht ook deze verkregen gegevens kan "doorgeven" als de aanbieder daar om vraagt. De veiligste manier is via signing en eventueel encryptie. Je wilt door de hele keten kunnen aantonen dat authenticatie gegevens alleen van de authenticatie service afkomstig kan zijn en de autorisatie gegevens van de autorisatie service. Als stuurgegevens onderdeel uitmaken van het bericht zijn de authenticatie service en autorisatie service onderdeel van de end-to-end points maar deze voorzieningen zijn niet in staat om het StUF bericht samen te stellen. Dit kan alleen de consument applicatie.



Dit "doorgeven" wil je niet in de payload. Als we uitgaan van de koppelvlakken die gebruik maken van HTTP/SOAP zijn er 2 locaties die beter geschikt zijn om stuurgegevens (authenticatie/autorisatie/routeringsgegevens) mee te geven:

- HTTP header
- SOAP header

Hier wordt ook volop gebruik van gemaakt. SOAP header met WS-* standaarden voor oa.

Digikoppeling-WUS en veel andere SOAP services. HTTP header als er bijvoorbeeld gebruik wordt gemaakt van een authenticatie service.

Er zijn andere koppelvlakken (REST/JMS) die bovenstaande gegevens op een andere manier zullen meegeven. Te denken valt aan JMS headers en HTTP headers.

Uit bovenstaande kan je concluderen dat je stuurgegevens niet voor alle koppelvlakken kan verplichten in de onderlaag.

Laat implementaties van koppelvlakken vrij in de keuze van stuurgegevens routing.

Als een implementatie van een HTTP/SOAP synchronisatie koppelvlak wordt gebruikt tussen 2 “client/server” systemen waarbij een client zelf het StUF bericht en stuurgegevens vult en er tussen de end-to-end points niet gerommeld hoeft te worden met de stuurgegevens kunnen stuurgegevens via een expliciete SOAP header als volgt meegegeven worden:

...

Voorbeeld uitwerking “Haal stuurgegevens weg uit de payload”:

Request:

```
<element name="npsSa04">
  <complexType>
    <sequence>
<!--           <element name="stuurgegevens" type="StUF:NPS-StuurgegevensSa04"/> -->
               <element name="object" type="BG:NPS-kerngegevens"/>
    </sequence>
  </complexType>
</element>
```

Response:

```
<complexType name="NPS-Lk02T">
  <sequence>
<!--           <element name="stuurgegevens" type="StUF:NPS-StuurgegevensLk02"/> -->
               <element name="parameters" type="StUF:ParametersLk02TInSx02"/>
               <element name="object" type="BG:NPS-kennisgeving_Sh"/>
  </sequence>
</complexType>
```

Voorbeeld uitwerking voor operatie npsSa04 in de implementatie van het koppelvlak

VerstrekSynchronisatieBericht voor het sectormodel Basisgegevens 0310 waarbij gebruik wordt gemaakt van stuurgegevens in de SOAP header:

...

```
<message name="npsSa04">
  <part name="body" element="BG:npsSa04"/>
  <part name="request_header" type="StUF:NPS-StuurgegevensSa04" />
</message>
```

en

```
<input>
  <soap:header use="literal" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
part="request_header" message="BG:npsSa04" />
  <soap:body use="literal" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
parts="body"/>
</input>
```

...

Voor de definities inclusief een request en response xml zie de bijlage.