



## **Leidraad VISI-systematiek versie 1.4**

### **Bijlage 8**

#### **Richtlijn voor VISI-communicatie op basis van SOAP**

### **Normatief**

Documentversie: 1.2  
Datum: 17 oktober 2014  
Status: Definitief



VISI 2003, 2008, 2011, 2014.

Op deze uitgave is de Creative Commons Licentie – Naamsvermelding – NietCommercieel – GelijkDelen – van toepassing. (zie: <http://creativecommons.org/licenses/by-nc-sa/3.0/nl/>)

CROW en degenen die aan deze publicatie hebben meegewerkt, hebben de hierin opgenomen gegevens zorgvuldig verzameld naar de laatste stand van wetenschap en techniek. Desondanks kunnen er onjuistheden in deze publicatie voorkomen. Gebruikers aanvaarden het risico daarvan. CROW sluit, mede ten behoeve van degenen die aan deze publicatie hebben meegewerkt, iedere aansprakelijkheid uit voor schade die mocht voortvloeien uit het gebruik van de gegevens.



## Inhoud

Inleiding .....	4
Doel .....	5
1.    Protocollen .....	6
2.    Architectuur .....	7
3.    Scenario berichtuitwisseling .....	9
4.    Ontsluiten Raamwerk en update scenario .....	16
5.    Ontsluiten projectspecifiek bericht scenario .....	17
6.    Attachments, Ref's en Id's .....	18
7.    Encryptie .....	19
8.    SOAP function calls .....	20
9.    Project specifiek bericht .....	21



## Inleiding

Dit document bevat de richtlijn voor implementatie van VISI communicatie op basis van het SOAP protocol. Het doel van de richtlijn is om handreikingen te geven voor de implementatie van VISI communicatie waardoor gewaarborgd wordt dat het betreffende informatiesysteem in technische zin VISI-berichten met bijlagen kan uitwisselen met een ander informatiesysteem dat volgens de richtlijn is ingericht.

### Wijzigingen ten opzichte van eerdere versie

De voorliggende richtlijn is een licht aangepaste versie (1.2) van de eerste versie (1.0). Wijzigingen ten opzichte van versie 1.1 zijn geel gemarkeerd.

De maximale grootte van een VISI-bericht (inclusief attachments) is 120 MB. Deze oplossing is gebaseerd op de MTOM-SOAP<sup>1</sup> implementatie van XOP (XML-binary Optimized Packaging) Zie ook: <http://www.w3.org/TR/soap12-mtom/>. Dit wordt voor verdere optimalisatie gecombineerd met 'chunking', een methode om bestanden in kleine stukken te versturen en vervolgens weer samen te voegen.

De VISI Standaard dient af te dwingen dat de communicatie tussen SOAP Servers plaatsvindt via https en SSL, waardoor de inhoud van ontvangen berichten niet onderschept kan worden.

Het gebruik van HTTPS (TLS 1.0 of hoger, met minimaal 128 bits encryptie) voor het versturen van VISI berichten en bijlagen is verplicht. Het is niet toegestaan om de "parseMessage" SOAP service aan te bieden over een onbeveiligd http kanaal.

Verder dient alle communicatie (indien deze over internet verloopt) tussen een VISI gebruiker en een VISI applicatie gebruik te maken van HTTPS met dezelfde specificaties.

Overige beveiligingsmaatregelen kunnen desgewenst op infrastructuur niveau worden getroffen.

Verder kunnen SOAP Servers de techniek "whitelist" gebruiken, waardoor alleen geautoriseerde SOAP server berichten naar een SOAP Server kunnen sturen .

---

<sup>1</sup> MTOM staat voor 'Message Transmission Optimization Mechanism'. Dit is een W3C-standaardmethode om op efficiënte wijze binaire data naar en van webservices te versturen. Het maakt gebruik van XML-binary Optimized Packaging (XOP) om binaire data over te brengen en is bedoeld om zowel MIME- als DIME-bijlagen te vervangen (MIME staat voor 'Multipurpose Internet Mail Extensions', een internetstandaard voor e-mail die de structuur en codering van e-mailberichten vastlegt. DIME staat voor 'Direct Internet Message Encapsulation).

Binaire bestanden worden gewoonlijk gezien als een opeenvolging van bytes. Binaire bestanden bevatten typisch bytes die niet als tekstkarakters geïnterpreteerd moeten worden, maar die iets anders moeten voorstellen. In een SOAP-bericht wordt zo'n zogenaamde 'array of bytes' echter vaak gecodeerd als tekstdata. MTOM maakt het mogelijk om efficiënter de binaire data van grote bestanden in a SOAP-request (aanvraag) of response (reactie) te plaatsen.

Bron: [http://nl.wikipedia.org/wiki/Message\\_Transmission\\_Optimization\\_Mechanism](http://nl.wikipedia.org/wiki/Message_Transmission_Optimization_Mechanism).



## Doel

Het doel van dit document is om een protocol vast te stellen waarmee we communicatie van VISI berichten tussen alle VISI ondersteunende software partijen kunnen beschrijven. Dit dient op een zodanige manier beschreven te zijn dat het protocol voor alle partijen implementeerbaar is, geen ongewone eisen aan implementatie en/of hardware worden vereist en het strikt genoeg is om communicatie mogelijk te maken.

De onderliggende eisen van dit protocol:

- strikt en eenduidig genoeg om communicatie tussen alle softwarepartijen, welke implementaties van dit protocol uitvoeren, mogelijk te maken.
- duidelijk en eenvoudig genoeg voor implementatie.
- geen eisen aan VISI compatibiliteit voor de nieuwe/extra software/servers.
- mogelijkheid een kopie van alle bilaterale communicatie op 1 of meerdere servers te plaatsen.

Het doel van de SOAP Central Server is het opslaan van alle berichten (m.u.v. berichten binnen 1 organisatie) die binnen een project plaatsvinden. Wat er vervolgens met deze berichten gedaan wordt, welke partijen toegang tot welke berichten hebben en hoe de beveiliging en opslag van deze berichten is geregeld valt buiten dit protocol.



## 1. Protocollen

De volgende protocollen worden gebruikt:

Voor implementatie wordt het MTOM (SOAP Message Transmission Optimization Mechanism) protocol vereist. De laatste versie van MTOM op het moment van uitbrengen van deze notitie is:

<http://www.w3.org/TR/soap12-mtom/>

Het gebruikte protocol is te vinden in het projectspecifieke bericht. Dit bericht bevat onder project-informatie een simpel element 'SOAPProtocol'. De huidige mogelijkheid is alleen 'MTOM', dit ziet er dus bijv. uit als onderdeel van het raamwerk:

```
<ProjectType id="Project_xyz">
  ...
  <description>Standaard project</description>
  ...
  <complexElements>
    ...
    <ComplexElementTypeRef idref="AnderWillekeurigComplexElement"/>
    ...
  </complexElements>
</ProjectType>
<ComplexElementType id="AnderWillekeurigComplexElement">
  <description>Een ander willekeurig complex element</description>
  ...
  <simpleElements>
    ...
    <SimpleElementTypeRef idref="SOAPProtocol"/>
    ...
  </simpleElements>
</ComplexElementType>
<SimpleElementType id="SOAPProtocol">
  ...
</SimpleElementType>
```

En voor het projectspecifieke bericht:

```
<Project_xyz id="Project-000">
  ...
  <anderWillekeurigComplexElement>
    ...
    <AnderWillekeurigComplexElementRef idref="ProjectGegevens"/>
    ...
  </anderWillekeurigComplexElement>
</Project_xyz>
<AnderWillekeurigComplexElement id="ProjectGegevens">
  ...
  <SOAPProtocol>MTOM</SOAPProtocol>
  ...
</AnderWillekeurigComplexElement>
```

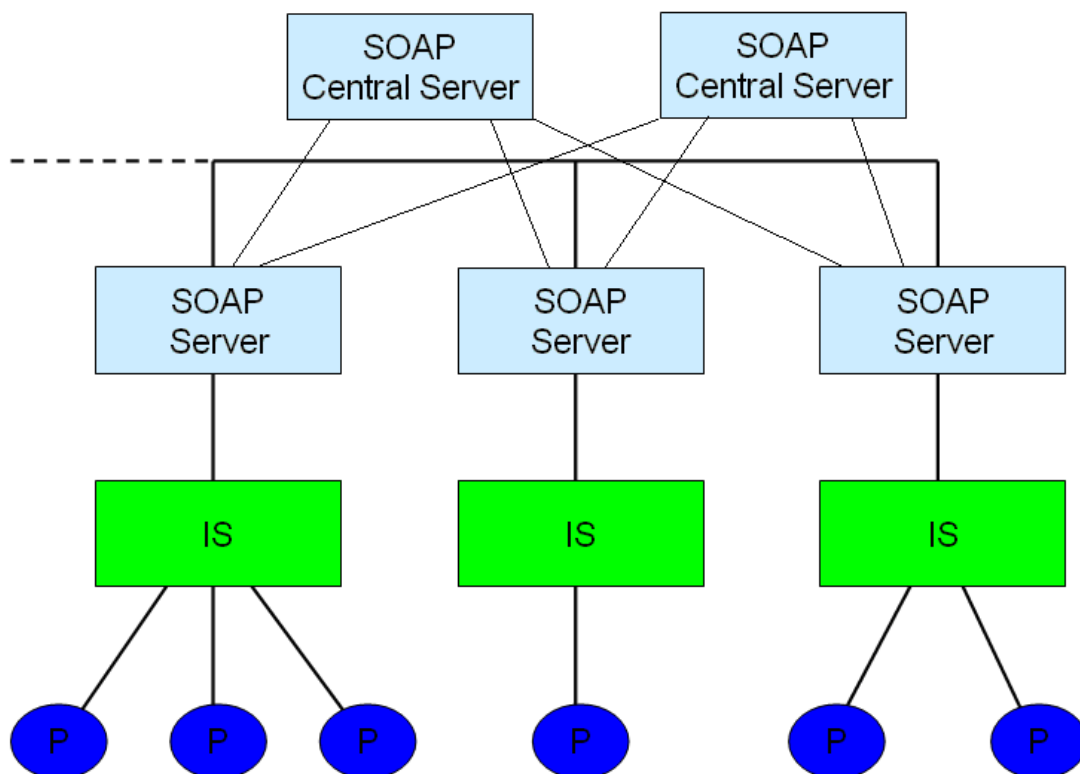
## 2. Architectuur

De architectuur is simpel gehouden door slechts één scenario te ondersteunen. Voorheen waren er meerdere architecturen waarbij de communicatie optioneel ook via een centrale server of gecombineerde server liep. Het scenario bevat onderlinge communicatie tussen SOAP servers met een kopie van deze berichten naar één of meerdere centrale servers indien aanwezig.

Het doel en de achtergrond van deze architectuur:

- de architectuur moet onderlinge communicatie tussen servers in een project kunnen ondersteunen zonder tussenkomst van andere servers,
- de architectuur moet opslag van alle berichten op één of meerdere servers binnen een VISI project kunnen ondersteunen,
- de gebruikte SOAP servers hoeven geen kennis van VISI te hebben,
- communicatie moet beveiligd uitgevoerd kunnen worden,
- berichten moeten eenduidig gecommuniceerd en opgeslagen kunnen worden,
- attachments moeten meegestuurd kunnen worden,
- de verzender moet te weten kunnen komen of een bericht goed is aangekomen,
- elke server moet in staat zijn zonder extra kennis meerdere projecten te kunnen ondersteunen.

**Architectuur:**



**Figuur 1 Architectuur**



## Scenario

In dit geschetste scenario zijn de volgende objecten te vinden:

- Het Informatie Systeem (**IS**), dit is de applicatie die VISI berichten verwerkt en genereert en daarnaast de user interface is naar de eindgebruiker .
- De eindgebruiker (**P** van persoon), dit is een persoon die voor een bepaalde organisatie een bepaalde rol vervuld.
- De SOAP Server (**SOAP Server**), dit is de server welke de afhandeling van VISI berichten volgens het in dit document beschreven pad afhandelt.
- Een SOAP Central Server (**SOAP Central Server**), dit is een SOAP server welke een kopie krijgt van al het berichtenverkeer. Deze server doet niets anders dan het opslaan van deze berichten en het vermelden aan de verzendende SOAP server dat het bericht in goede orde ontvangen is. Er kunnen in een project 0 tot n (meerdere) SOAP Central Servers aanwezig zijn. Elke SOAP server is verplicht alle gecommuniceerde berichten ook naar alle aanwezige SOAP Central Servers te sturen (exacte kopie van gecommuniceerd bericht).





### 3. Scenario berichtuitwisseling

De berichtuitwisseling (header etc., server address) is opgezet op basis van drie randvoorwaarden:

- De SOAP Servers en de SOAP Central Server(s) zijn niet in staat VISI berichten te parsen (inhoudelijk te begrijpen).
- Het informatie systeem (IS) en de SOAP Server hebben geen (statische) kennis van het wel of niet aanwezig zijn van SOAP Central Servers, aanwezigheid van één of meerdere SOAP Central Servers moet dus ondersteund kunnen worden zonder iets aan het informatiesysteem of de SOAP server te wijzigen.
- Alle informatie over de aanwezige configuratie, URL adressen van personen in een bepaalde rol e.d. zijn bevat in VISI berichten volgens het raamwerk voor het uit te voeren project (natuurlijk mag deze randvoorwaarde de software niet van een specifiek VISI raamwerk afhankelijk maken).

Om aan bovenstaande randvoorwaarden te kunnen voldoen zullen we een nieuw concept binnen VISI lanceren.

**Ter info gebruikt nieuw VISI concept:**

- Enkele SimpleElementType objecten welk gedefinieerd kunnen worden in het raamwerk zullen op basis van hun naamgeving (attribuut: *id*) over alle raamwerken heen een specifiek gedrag/invulling toegekend krijgen.

We zullen aan de volgende SimpleElementTypes raamwerk-overschrijdend gedrag/invulling toekennen:

SimpleElementType	Behavior
SOAPServerURL	Bevat het URL van de SOAP server behorende bij deze Persoon, Rol of Organisatie
SOAPCentralServerURL	Bevat het URL van de SOAP central servers behorende bij dit project
SOAPProtocol	Bevat het gebruikte SOAP protocol

#### 3.1. Berichtuitwisseling vaste elementen op raamwerkniveau

Binnen deze notitie zullen we het volgende gebruik voorstellen (let wel: dit valt buiten het nieuwe VISI concept):

**Op raamwerkniveau:**

- Elk raamwerk zal het volgende stukje XML bevatten om de SOAPServerURL aan een organisatie toe te kunnen wijzen:

```
<OrganisationType id="Organisatie">
...
<description>Standaard organisatie</description>
...
<complexElements>
```



```

...
<ComplexElementTypeRef idref="WillekeurigComplexElement"/>
...
</complexElements>
</OrganisationType>
<ComplexElementType id="WillekeurigComplexElement">
  <description>Een willekeurig complex element</description>
  ...
  <simpleElements>
    ...
    <SimpleElementTypeRef idref="SOAPServerURL"/>
    ...
  </simpleElements>
</ComplexElementType>
<SimpleElementType id="SOAPServerURL">
  ...
</SimpleElementType>

```

- Elk raamwerk zal het volgende stukje XML bevatten om de SOAPCentralServerURL aan een project toe te kunnen wijzen:

```

<ProjectType id="Project_xyz">
  <description>Standaard project</description>
  ...
  <complexElements>
    ...
    <ComplexElementTypeRef idref="AnderWillekeurigComplexElement"/>
    ...
  </complexElements>
</ProjectType>
<ComplexElementType id="AnderWillekeurigComplexElement">
  <description>Een ander willekeurig complex element</description>
  ...
  <simpleElements>
    ...
    <SimpleElementTypeRef idref="SOAPCentralServerURL"/>
    ...
  </simpleElements>
</ComplexElementType>
<SimpleElementType id="SOAPCentralServerURL">
  ...
</SimpleElementType>

```

#### Alternatieven:

De gemaakte keuze lijkt de meest elegante, een alternatief zou kunnen zijn het veld SOAPServerURL aan PersonInRole toe te kennen bij de promotiestap. Deze keuze zou de VISI systematiek echter afhankelijk maken van SOAP en de conceptuele keuze om deze communicatie voor te schrijven.

### 3.2. Berichtuitwisseling vaste elementen op berichtniveau

**Op berichtniveau:** we gaan er vanuit dat er bij een project een projectspecifiek bericht aanwezig is zoals in 9. is beschreven.



- In dit projectspecifieke bericht is voor elke organisatie gedefinieerd wat zijn SOAPServerURL is:

```
<Organisatie id="Kraaijeveld">
  <name>Kraaijeveld's Aannemingsbedrijf BV</name>
  ...
  <willekeurigComplexElement>
    ...
    <WillekeurigComplexElementRef idref="KraaijeveldGegevens"/>
    ...
  </willekeurigComplexElement>
</Organisatie>
<WillekeurigComplexElement id="KraaijeveldGegevens">
  ...
  <SOAPServerURL>http://192.168.0.102/visi.wsdl</SOAPServerURL>
  ...
</WillekeurigComplexElement>
```

- Bij de projectdefinitie in het projectspecifieke bericht zal de SOAPCentralServerURL te vinden zijn:

```
<Project_xyz id="Project-000">
  <name>VISI-Showcase</name>
  ...
  <anderWillekeurigComplexElement>
    ...
    <AnderWillekeurigComplexElementRef idref="ProjectGegevens"/>
    ...
  </anderWillekeurigComplexElement>
</Project_xyz>
<AnderWillekeurigComplexElement id="ProjectGegevens">
  ...
  <SOAPCentralServerURL>http://192.168.0.1/visi.wsdl</SOAPCentralServerURL>
  ...
</AnderWillekeurigComplexElement>
```

### 3.2.1. Gevolgen

De gevolgen van deze aanpak is dat het informatie systeem (IS) in staat is bij elk bericht binnen een transactie te achterhalen welke URL behoort tot de afzender en welke URL behoort tot de ontvanger. Daarnaast kan in het geval van aanwezigheid van Central Servers direct gezien worden welke URL behoort tot de centrale server. Deze gegevens (URL afzender, URL ontvanger en optioneel één of meerdere URL's centrale server(s)) zullen gecommuniceerd moeten worden met de SOAP server behorende bij dit informatiesysteem, natuurlijk samen met het VISI bericht zelf. Deze communicatie tussen het informatiesysteem en bijbehorende SOAP server valt buiten deze notitie en is aan de ontwikkelaar van het informatiesysteem en de bijbehorende SOAP server zelf.

### 3.3. Berichtuitwisseling initialisatie

Om een project te beginnen zullen alle partijen moeten weten met wie ze communiceren, welk raamwerk gebruikt wordt en alle andere projectspecifieke informatie beschikbaar moeten hebben. Om deze informatie altijd up-to-date te houden en te voorkomen dat ongeautoriseerde partijen deze informatie kunnen aanpassen is een projectspecifiek bericht gedefinieerd. Dit projectspecifieke



bericht bevindt zich op een door de projectleider aangegeven locatie (bestaande infrastructuur verzorgen vervolgens beveiliging en beschikbaarheid). Meer informatie over het projectspecifieke bericht is te vinden in hoofdstuk 5 en 9.

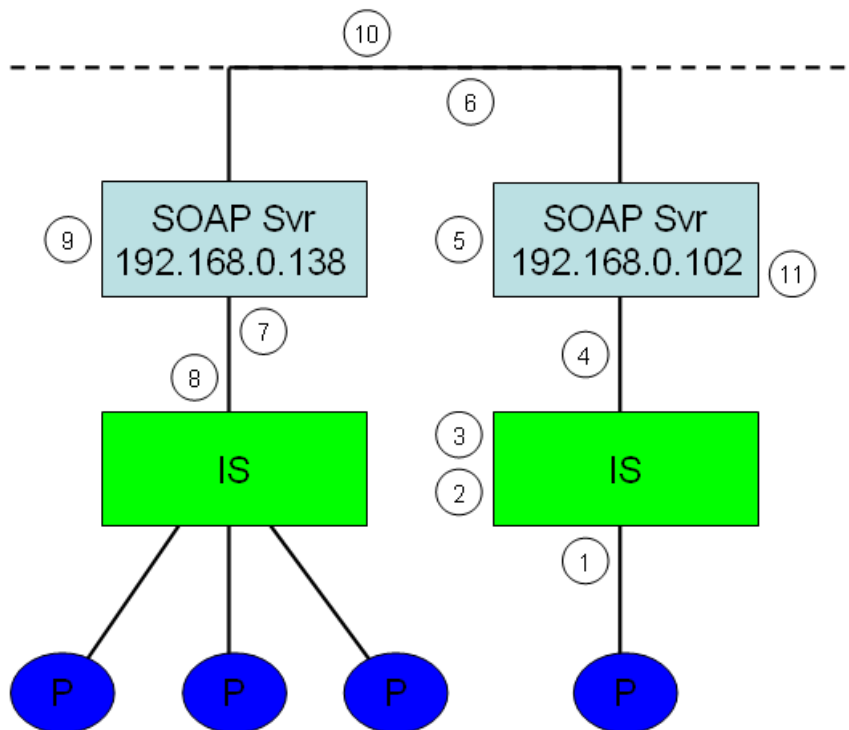
De initialisatie van elk informatiesysteem begint met het invoeren van het adres van het projectspecifieke bericht (dus bij ieder afzonderlijk informatiesysteem). Dit projectspecifieke bericht bevat de link naar het meest recente raamwerk (zoals ieder bericht naar een raamwerk verwijst) en bevat alle relevante projectinformatie waaronder de rollen, personen en organisaties die participeren (incl. URL adressen).

### 3.4. Berichtuitwisseling scenario

Deze beschrijving is geschikt voor alle type berichten binnen transacties, zowel een eerste bericht binnen een transactie als reacties op ontvangen berichten binnen een transactie. Belangrijk is dat berichten binnen een organisatie niet via dit pad hoeven te lopen, dit betekent ook dat deze berichten niet op de centrale server komen te staan.

Bij de voorbeelden:

- URL SOAP server versturende partij: <http://192.168.0.102>
- URL SOAP server ontvangende partij: <http://192.168.0.138>
- URL SOAP central server: <empty>



Figuur 2

De volgorde van berichtuitwisseling:

- 1 Het VISI bericht wordt opgemaakt door het versturende IS op basis van het ontvangen bericht (in geval van een nieuwe transactie wordt de informatie uit het projectspecifieke bericht gehaald).



- 2 De URL adressen van het versturende en het ontvangende IS worden door het versturende IS uit het opgemaakte bericht gehaald.
- 3 Het URL adres van de SOAP central server wordt gezocht maar niet gevonden in het project-specifieke bericht (anders bevonden we ons in scenario II).
- 4 Het opgestelde VISI bericht + URL adres van versturende SOAP server + URL adres van ontvangende SOAP server + URL adres van SOAP central server (deze is in dit scenario leeg) worden naar de bijbehorende server gecommuniceerd (wijze van communicatie wordt aan de softwarepartijen zelf overgelaten).
- 5 De SOAP server van de versturende partij bouwt een SOAP bericht als volgt op:

```
<SOAP-ENV:Envelope ...>
  <SOAP-ENV:Header>
    <SOAPServerURL ...>
      <sender>http://192.168.0.102</sender>
      <receiver>http://192.168.0.138</reciever>
    </SOAPServerURL>
    <SOAPCentralServerURL ...>
      <server/>
    </SOAPCentralServerURL>
    <UniqueID ...>
      <ID>UniqueIDonMessageInitiatingSOAPServer_XYZ</ID>
    </UniqueID>
    <Attachments ...>
      <count>2</count>
    </Attachments>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <parseMessage ...>
      <Data ...>
        ... // Attachment I
      </Data>
      <Data ...>
        ... // Attachment II
      </Data>
      <visiXML_MessageSchema ...>
        ... // VISI Bericht
      </visiXML_MessageSchema>
    </parseMessage>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**note:** de id van elke attachment is gelijk aan de id die in het VISI bericht wordt gebruikt om de metadata van een attachment te beschrijven. (<Data id="abc">)

- 6 De SOAP server van de versturende partij verstuurt dit bericht naar de SOAP Central Server of indien dit niet mogelijk is (bij dit scenario niet mogelijk) naar de SOAP server van de ontvangende partij.
- 7 De SOAP server van de ontvangende partij stuurt een standaard SOAP exception error bericht naar de versturende partij.
  - a. De SOAP server van de ontvangende partij stuurt het VISI message deel van het bericht door naar het IS van de ontvangende partij (wijze van communicatie wordt



aan de softwarepartijen zelf overgelaten).

- 8 Het IS van de ontvangende partij interpreteert het VISI bericht en indien akkoord verstuurt hij dit bericht + URL adres van versturende SOAP server + URL adres van ontvangende SOAP server + URL adres van SOAP central server (deze is in dit scenario leeg) naar zijn eigen SOAP server (dus van de ontvangende partij). De wijze van communicatie wordt aan de softwarepartijen zelf overgelaten.
- 9 De SOAP server van de ontvangende partij bouwt bij succes een SOAP reactiebericht op als volgt:

```
<SOAP-ENV:Envelope ...>
  <SOAP-ENV:Header>
    <SOAPServerURL ...>
      <sender>http://192.168.0.102</sender>
      <reciever>http://192.168.0.138</reciever>
    </SOAPServerURL>
    <SOAPCentralServerURL ...>
      <server/>
    </SOAPCentralServerURL>
    <UniqueID ...>
      <ID>UniqueIDonMessageInitiatingSOAPServer_XYZ</ID>
    </UniqueID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ERRORS>
      <ERROR CODE="0"></ERROR>
    </ERRORS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

bij een enkelvoudige fout:

```
<SOAP-ENV:Envelope ...>
  <SOAP-ENV:Header>
    <SOAPServerURL ...>
      <sender>http://192.168.0.102</sender>
      <receiver>http://192.168.0.138</reciever>
    </SOAPServerURL>
    <SOAPCentralServerURL ...>
      <server/>
    </SOAPCentralServerURL>
    <UniqueID ...>
      <ID>UniqueIDonMessageInitiatingSOAPServer_XYZ</ID>
    </UniqueID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ERRORS>
      <ERROR CODE="1"> Er is een fout opgetreden bij ...</ERROR>
    </ERRORS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Bij meerdere fouten (bijvoorbeeld bij validatie van de xsd):



```
<SOAP-ENV:Envelope ...>
  <SOAP-ENV:Header>
    <SOAPServerURL ...>
      <sender>http://192.168.0.102</sender>
      <receiver>http://192.168.0.138</reciever>
    </SOAPServerURL>
    <SOAPCentralServerURL ...>
      <server/>
    </SOAPCentralServerURL>
    <UniqueID ...>
      <ID>UniqueIDonMessageInitiatingSOAPServer_XYZ</ID>
    </UniqueID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ERRORS>
      <ERROR CODE="1">Waarden van simpel element1 is niet volgens definitie</ERROR>
      <ERROR CODE="1">Waarden van simpel element2 is niet volgens definitie</ERROR>
    </ERRORS>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In geval van geen fout is de code 0, bij een onbekende fout is de code 1, deze foutmeldingen zullen moeten worden opgelost door de programmeurs. Alle foutmeldingen met een hogere code betreffen fouten die door de software begrepen kunnen worden.

- 10 De SOAP server van de ontvangende partij stuurt dit reactiebericht naar de SOAP server van de versturende partij en naar elke SOAP central server.
  - a. De SOAP server van de versturende partij stuurt een standaard SOAP exception error bericht naar de ontvangende partij als standaard reactie op dit reactiebericht.
- 11 Reactiebericht terug bij af, controleer of de informatie overeenkomt met het verstuurd bericht en indien akkoord notificeer het IS van de ontvangende partij dat het bericht volgens afspraak is verstuurd en ontvangen.

**Opmerking:**

- Bovengenoemde proces blijft hanteerbaar indien beide partijen dezelfde IS gebruiken. In dat geval is het URL adres van de ontvangende en de versturende SOAP server identiek.
- Tijdens het uitkomen van dit protocol zijn er geen standaard fout codes hoger dan 1 gedefinieerd. Zodra een dergelijke fout code is vastgelegd wordt duidelijk aangegeven wat de fout code is, een omschrijving van de fout en een eenduidige omschrijving van hoe de software zou moeten reageren.
- Indien er niet ('op tijd') gereageerd wordt op een bericht wordt niet nogmaals hetzelfde bericht verstuurd. We blijven wachten op het antwoord of vinden een oplossing buiten VISI om.
- In stap 7 en stap 10 is een bericht altijd eerst naar de andere partij gestuurd. Pas als via het standaard SOAP protocol een goede reactie is ontvangen (7a en 10a) wordt een kopie naar de SOAP Central Server(s) gestuurd.



## 4. Ontsluiten Raamwerk en update scenario

Raamwerken zijn voor een informatiesysteem via een URI bereikbaar. Elk VISI compatible bericht binnen een specifiek project verwijst naar dit raamwerk (dus ook het projectspecifieke bericht).

Het binnen de VISI gebruikte XML/XSD concept dwingt af dat ieder bericht een referentie naar zijn schema bevat via een URI. De VISI systematiek schrijft vervolgens voor dat op deze URI (welke altijd de extensie XSD bevat) echter met extensie xml het raamwerk is opgeslagen.

Verder schrijft VISI voor eens gebruikte raamwerken voor altijd te bevriezen. Bij wijzigingen in het raamwerk zal een nieuwe URI voor dit nieuwe raamwerk moet worden bedacht. De applicaties worden automatisch op de hoogte gesteld door het projectspecifieke bericht te laten verwijzen naar de XSD behorende bij dit nieuwe raamwerk.

Hiermee garandeert gebruik van VISI dat alle raamwerken en alle berichten eenduidig blijven in de tijd ondanks aanpassingen aan de raamwerken, rollen, personen of organisaties of zelfs de systematiek.

Bij het aanpassen van een raamwerk zijn de volgende stappen te nemen:

- Pas het raamwerk aan, bijvoorbeeld met behulp van een VISI raamwerkeditor.
- Distribueer de voorgestelde wijzigingen met behulp van het metaraamwerk en wacht op instemming van alle betrokken VISI partners in het project.
- Promote het raamwerk en plaats zowel het raamwerk als het gepromote raamwerk (met dezelfde naam, maar andere extensie) in de gewenste directory.
- Pas het projectspecifieke bericht aan En distribueer dit over alle betrokken VISI partners en stel de ingangsdatum en tijd vast.

Op de afgesproken tijd wordt het gewijzigde raamwerk en het bijbehorende projectspecifieke bericht actief voor nieuwe transacties.

De applicaties zullen bestaande transacties en nieuwe 'subtransacties' via het oude raamwerk laten lopen. Nieuwe transacties zullen via het nieuwe raamwerk opgestart worden.





## 5. Ontsluiten projectspecifiek bericht scenario

De exacte inhoud van een projectspecifiek bericht is elders beschreven. Hier is beschreven wat er met updates moet gebeuren.

Er kunnen verschillende redenen zijn om een projectspecifiek bericht te updaten, dit kan zijn omdat personen, organisaties en/of rollen zijn aangepast, meer informatie bekend is omtrent personen, organisatie, rollen of het project dan wel omdat het raamwerk is aangepast.

In principe is de informatie van een projectspecifiek bericht vluchtig, het kan gezien worden als een 'vergaarbak' van belangrijke projectspecifieke informatie welke altijd up-to-date beschikbaar is voor de verschillende informatie systemen die een rol spelen in dit project. De informatie is in tegenstelling tot raamwerken dan ook dynamisch waarbij hier de locatie (URI) statisch is, i.e. elk informatiesysteem weet waar hij de laatste meest up-to-date informatie van het project kan vinden (deze URI wordt gebruikt om een project mee op te starten, zie ook **3.3.**). Om de verschillende versies van het projectspecifieke bericht te kunnen achterhalen schrijft VISI voor als volgt een update uit te voeren:

- Maak een kopie van het bestaande projectspecifieke bericht, dit bericht krijgt dezelfde naam met een volgnummer direct achter de naam.
- Verzorg de update van het projectspecifieke bericht.
- Distribueer het projectspecifieke bericht over de VISI projectpartners en stel de ingangsdatum plus tijd vast.
- Op de afgesproken datum en tijd wordt het nieuwe projectspecifieke bericht geactiveerd.



## 6. Attachments, Ref's en Id's

De attachments en elementen in de header file worden zoals in 3. beschreven verwerkt. Gebruik van het MTOM concept verzorgt de afhandeling van attachments en het VISI bericht zelf. Tevens zal bij encryptie van de body van een SOAP bericht automatisch deze encryptie op de attachments aanwezig zijn. Voor de laatste info over MTOM zie ook: <http://www.w3.org/TR/soap12-mtom/>



## 7. Encryptie

Het gekozen protocol MTOM heeft in de Microsoft Web Services Enhancements (WSE) 3.0 een integratie van MTOM en security voor de implementeur. Mijn voorstel zou zijn deze door Microsoft gekozen bij MTOM passende security aan te houden en op basis van ervaringen van eerste implementaties uitspraken voor deze VISI notitie vast te stellen.

**NB:**

- alleen geldig indien deze encryptie ondersteund wordt door alle systemen.



## 8. SOAP function calls

De gekozen oplossing vereist slechts twee function calls.

functioncall **parseMessage**

*(1 inputvariabele van het type string, bevat een XML file, zie ook 3.4)*

functioncall **parseMessageConfirmation**

*(2 inputvariabelen van het type string, t.w.*

*UniqueIDonMessageInitiatingSOAPServer\_XYZ en de gevonden errors, geeft wel een XML file terug, zie ook 3.4)*

De inhoud is in beide gevallen gelijk, namelijk het VISI bericht voorafgegaan aan de attachments. De vorm van het SOAP bericht behoort te voldoen aan het protocol welke gekozen is in het project-specifieke bericht.

‘Meta’ informatie over de verschillende URL adressen en het unieke ID voor de versturende SOAP server worden in de header van het SOAP bericht opgenomen.

De wijze van implementatie en gebruik is beschreven in paragraaf 3.4.



## 9. Project specifiek bericht

De inhoud projectspecifiek bericht is reeds in een eerder document beschreven. De wijze van handelen bij updates van een projectspecifiek bericht is te vinden in hoofdstuk 5.

< einde Bijlage 8 >