

## **VISI**

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

**Versie: 1.1**

**Datum: 17 juli 2007**

**Status: voorstel update van versie 1.0**

## **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**

Aangezien de voorliggende richtlijn de eerste versie is die gepubliceerd wordt, zijn er geen wijzigingen te beschrijven ten opzichte van een eerdere versie.

## **Geformuleerde doel**

De uitgangspunten van dit document zijn 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.

## **Richtlijn onderwerpen**

1. Protocollen
2. Architectuur / Servers
3. Scenario's berichtuitwisseling (header etc., serveraddress)
4. Ontsluiten Raamwerk en update scenario
5. Ontsluiten projectspecifiek bericht scenario
6. Attachments, Ref's en Id's
7. Encryptie
8. SOAP function calls
9. Project specifiek bericht

## 1. Protocollen

De volgende protocollen worden gebruikt:

Voor implementatie wordt het Base64 of het MTOM (SOAP Message Transmission Optimization Mechanism) protocol vereist. Laatste versie van MTOM op het moment van uitbrengen van deze notitie:

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

Het gebruikte protocol is te vinden in het projectspecifieke bericht. Dit bericht bevat onder projectinformatie een simpel element 'SOAPProtocol'. De huidige mogelijkheden zijn 'Base64' en 'MTOM', dit ziet er dus bijv. uit als voor het schema:

```
<ProjectType id="Project_xyz">
  <description>Standaard project</description>
  ...
  <complexContent>
    ...
    <ComplexElementTypeRef idref="AnderWillekeurigComplexElement"/>
    ...
  </complexContent>
</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>Base64</SOAPProtocol>
  ...
</AnderWillekeurigComplexElement>
```

Opmerking:

Base 64 is gekozen als uitgangspunt voor de tot nu uitgevoerde implementaties daar ten tijde van deze implementaties van deze richtlijn het protocol MTOM nog niet goed werd ondersteund door de ontwikkel omgeving.

Voor toekomstige versies van de richtlijn wordt MTOM geadviseerd, voor deze versie van de richtlijn Base64.

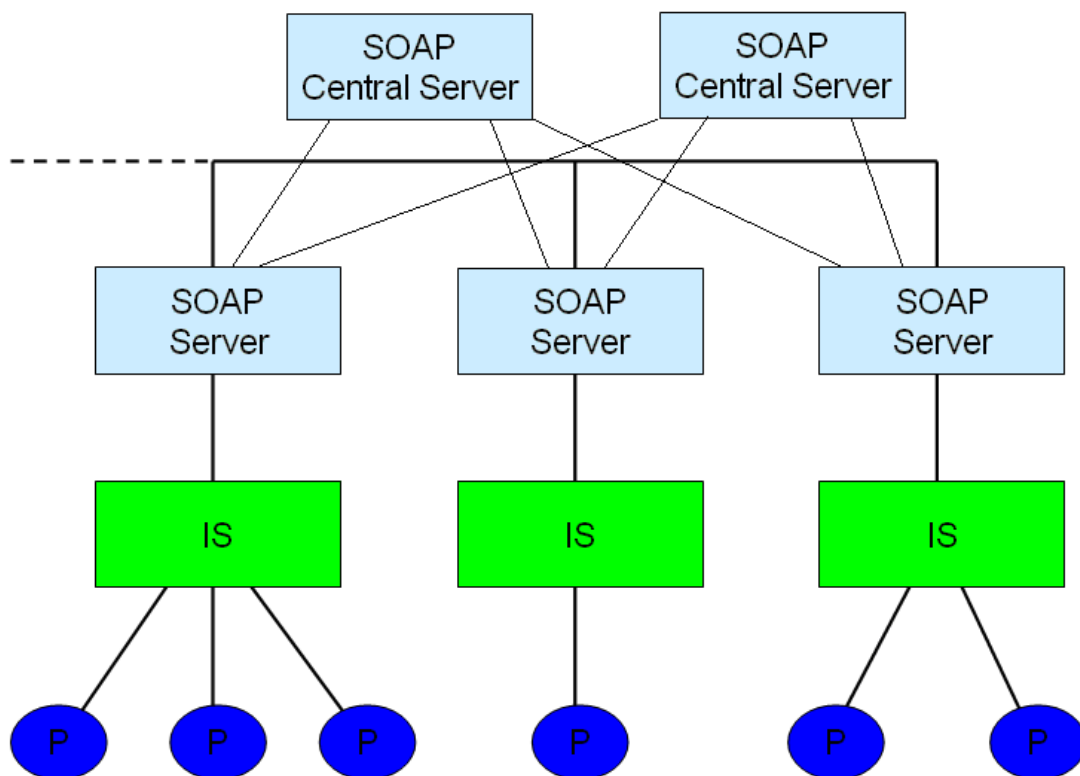
## 2. Architectuur

De architectuur is simpel gehouden door slechts één scenario te ondersteunen. Voorheen waren er meerder architecturen waarbij 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 1 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 1 of meerdere servers binnen een VISI project moeten 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 komen of een bericht goed is aangekomen
- elke server moet in staat zijn zonder extra kennis meerdere projecten te kunnen ondersteunen

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 naar de eindgebruiker toe is.
- De eindgebruiker (**P** van persoon), dit is een persoon welke 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 zijn alle gecommuniceerde berichten ook naar alle aanwezige SOAP Central Servers te sturen (exacte kopie van gecommuniceerd bericht).

### 3. Scenario berichtuitwisseling (header etc., serveraddress)

De berichtuitwisseling is opgezet op basis van drie randvoorwaarden:

- De SOAP Servers en de SOAP Central Server 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 informatie systeem of de SOAP server te wijzigen.
- Alle informatie over de aanwezig 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 specifieke behavior (gedrag/invulling) toegekend krijgen.

We zullen aan de volgende SimpleElementTypes raamwerk-overschrijdend behavior toekennen:

<b>SimpleElementType</b>	<b>Behavior</b>
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>
  ...
  <complexType>
    ...
    <ComplexElementRef idref="WillekeurigComplexElement"/>
    ...
  </complexType>
</OrganisationType>
<ComplexElement id="WillekeurigComplexElement">
  <description>Een willekeurig complex element</description>
  ...
  <simpleContent>
    ...
    <SimpleElementRef idref="SOAPServerURL"/>
    ...
  </simpleContent>
</ComplexElement>
<SimpleElement id="SOAPServerURL">
  ...
</SimpleElement>
<SimpleElement id="SOAPServerURL">
  ...
</SimpleElement>
```

- 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>
  ...
  <complexType>
    ...
    <ComplexElementRef idref="AnderWillekeurigComplexElement"/>
    ...
  </complexType>
</ProjectType>
<ComplexElement id="AnderWillekeurigComplexElement">
  <description>Een ander willekeurig complex element</description>
  ...
  <simpleContent>
    ...
    <SimpleElementRef idref="SOAPCentralServerURL"/>
    ...
  </simpleContent>
</ComplexElement>
<SimpleElement id="SOAPCentralServerURL">
  ...
</SimpleElement>
<SimpleElement id="SOAPCentralServerURL">
  ...
</SimpleElement>
```

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 1 of meerdere URL's centrale server) zullen gecommuniceerd moeten worden naar de SOAP server behorende bij dit informatie systeem, natuurlijk samen met het VISI bericht zelf. Deze communicatie tussen het informatie systeem en bijbehorende SOAP server valt buiten deze notitie en is aan de ontwikkelaar van het informatie systeem 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 uit infrastructuur, verzorging, beveiliging en beschikbaarheid). Meer informatie over het projectspecifieke bericht is te vinden in **5.** en **9.**

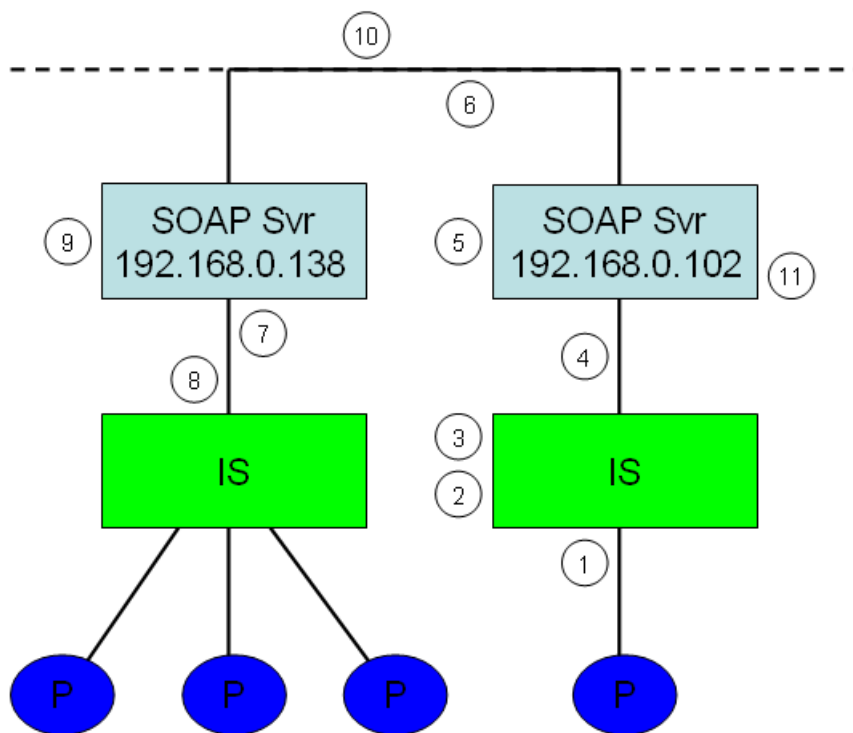
De initialisatie van elk informatie systeem begint met het invoeren van het adres van het projectspecifieke bericht (dus bij ieder afzonderlijk informatie systeem). Dit projectspecifieke bericht bevat de link naar het meest recente raamwerk (zoals ieder bericht naar een raamwerk verwijst) en bevat alle 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 betekend ook dat deze berichten niet op de central 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>`



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 projectspecifieke 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 op als volgt:

```

<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 attachement is gelijk aan de id die in het VISI bericht wordt gebruikt om de metadata van een attachement 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 verstuurd 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), (wijze van communicatie wordt aan de softwarepartijen zelf overgelaten)
- 9 De SOAP server van de ontvangende partij bouwt een SOAP reactiebericht op als volgt bij succes:

```

<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 enkele error:

```
<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>
    <UniquelD ...>
      <ID>UniquelDonMessageInitiatingSOAPServer_XYZ</ID>
    </UniquelD>
  </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 errors (bijvoorbeeld bij validatie 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>
    <UniquelD ...>
      <ID>UniquelDonMessageInitiatingSOAPServer_XYZ</ID>
    </UniquelD>
  </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 error de code is 0, bij een onbekende error de code is 1, deze errors zullen moeten worden opgelost door de programmeurs. Alle errors met een hogere code zijn errors die door de software begrepen kunnen worden.

- 10 De SOAP server van de ontvangende partij verstuurt 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 error codes hoger dan 1 gedefinieerd. Zodra een dergelijke error code is vastgelegd wordt duidelijk aangegeven wat de error code is, een omschrijving van de error 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 een bericht is 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 de VRE
- 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, veelal zal het aanpassen van de referentie naar het nieuwe raamwerk voldoende zijn, slechts in uitzonderlijke gevallen zal het projectspecifieke bericht niet meer valide zijn en verder aangepast moeten worden
- Klaar, de applicaties doen de rest

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

#### **Note:**

- De beschreven werkwijze is in de huidige implementatie niet gehanteerd. De problematiek bij het aanpassen van een raamwerk dient eerst goed uitgezocht te worden aan de hand van een praktijkprobleem teneinde juist implementeerbaar te zijn door de softwarepartijen.
- subtransacties zijn een specialisatie van de standaard transactie. Een transactie is een subtransactie indien hij wordt gerefereerd door een ander TransactieType via het attribuut 'subTransaction' in het raamwerk. Een subtransactie is een transactie die kan worden gestart vanuit een transactie welke vanuit zijn TransactieType refereerd naar het TransactieType van de subtransactie.
- het updaten van raamwerken en projectspecifieke berichten is nieuw, de beste wijze van afhandeling wordt pas duidelijk bij echt gebruik van deze functionaliteit. Zodra er meer praktische ervaringen zijn zullen we het protocol aanpassen op basis van de kennis opgedaan tijdens deze ervaringen.

## 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 informatie systeem 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 copy van het bestaande projectspecifieke bericht, dit bericht krijgt dezelfde naam met een volgnummer direct achter de naam.
- Verzorg de update van het projectspecifieke bericht.



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

De attachments en elementen in de header file worden zoals in 3. beschreven verwerkt. Gebruik van MTOM concept verzorgt de afhandeling van attachments en het VSI 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/>

Note: de huidige implementatie gebruiken nu nog Base 64, naar verwachting zal deze software ook aangepast worden voor communicatie volgens MTOM.

## **7. Encryptie**

Het gekozen protocol MTOM heeft in de Microsoft Web Services Enhancements (WSE) 3.0 beta een integratie van MTOM en security voor de implementor. 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.

**Note:**

- alleen geldig indien deze encryptie ondersteund wordt door alle systemen
- voor deze versie van het protocol is nog gekozen voor Base64, hiermee is dit hoofdstuk niet relevant.

## 8. SOAP function calls

De gekozen oplossing vereist slechts 2 functiecalls.

functiecall **parseMessage**

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

functiecall **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 projectspecifieke 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 **3.4**.

.

## **9. Project specifiek bericht**

De inhoud project specifiek bericht is reeds in een eerder document beschreven. De wijze van handelen bij updates van een project specifiek bericht is te vinden in **5.**

De rol van het projectspecifieke bericht bij initialisatie van een project is beschreven in **3.3.**