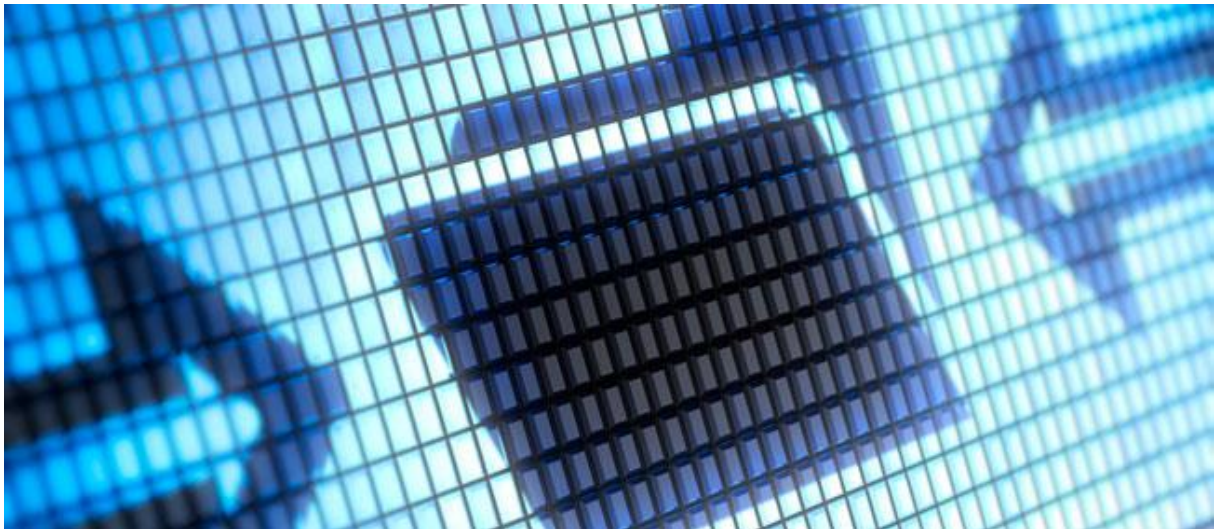


Messageservice webservice

Technische handleiding

Datum: 23 oktober 2017, versienummer 3.1



KETENSTANDAARD
BOUW EN INSTALLATIE

Versiebeheer

| Versie | Datum | Door | Toelichting wijzigingen |
|--------|------------|--|---|
| 1.00 | 11-05-2010 | K. Martens (ENK) | Eerste opzet |
| 1.10 | 21-05-2010 | R. Kessels (Galvano) | Verwerking diverse opmerkingen |
| 1.20 | 22-05-2010 | K. Martens (ENK) | Wijzigingen van Ron geaccepteerd en opmerkingen geantwoord. |
| 1.30 | 16-11-2010 | F. Van Mensel (Galvano) | Technische handleiding en MessageService voorstel samengevoegd en gereviseerd. |
| 1.40 | 26-11-2010 | K. Martens (ENK) | Splitsing basisdocument gebruik webservices en MessageService webservice. |
| 1.41 | 30-11-2010 | F. Van Mensel (Galvano) | Tekstuele aanpassingen doorgevoerd. |
| 1.42 | 03-01-2011 | F. Van Mensel (Galvano) | Enkele aanvullingen toegevoegd. |
| 1.43 | 23-03-2011 | A. Haeser (ETIM Nederland) | Standaarden ETIM, aanvullingen, aanpassingen rond de afspraken |
| 2.00 | 23-04-2013 | R. Kessels (SGBD-NL) | Opmerkingen en discussie n.a.v. werkgroep communicatieprotocollen en webservices van 12 april 2013 doorgevoerd. |
| 2.10 | 03-12-2013 | R. Kessels (SGBD-NL) en E. Geurts (Oosterberg) | Diverse aanpassingen na overleg, o.a. MsgFormat toegevoegd en paragraaf met afspraken geconcretiseerd. |
| 2.20 | 17-12-2013 | R. Kessels (SGBD-NL) | Opmerkingen A. Haeser, K. Martens en E. Geurts verwerkt |
| 2.25 | 18-12-2013 | R. Kessels (SGBD-NL) | Attachments toegevoegd. |



| | | | |
|------|----------------|---|---|
| 2.30 | 18-12-2013 | R. Kessels (SGBD-NL) | Aangepast dat een leverancier per message meerdere formaten/versies aan kan bieden. De klant kan vervolgens bepalen welk formaat hij ophaalt. |
| 2.35 | 19-12-2013 | R. Kessels (SGBD-NL) | Attachment gebruik van URL of AttachedData is verplicht (choice) |
| 2.40 | 03-02-2014 | E. Geurts Oosterberg | 'Request' en Response' aan elementen toegevoegd 'Fault' element aangepast. |
| 2.45 | 1 oktober 2015 | Jan Westerkamp Kyra Blankenstein Rachel van Rhijn | <ul style="list-style-type: none"> - Toevoegen alinea 'coproductie' in inleiding. - Sales in de Bouw vervangen voor Ketenstandaard Bouw en Installatie - Overzetten in nieuwe sjabloon |
| 3.0 | 18 mei 2017 | Cor Zijlstra | <ul style="list-style-type: none"> - Wijzigingen n.a.l.v. de 3.0 versie |
| 3.0 | 22 mei 2017 | Luuk d'Hooghe | <ul style="list-style-type: none"> - MessageResponse vervangen door MessageResult - Toegevoegd msgTypes (MTNINS en MTNSTA) - Verwijderen ETIM verwijzing |
| 3.0 | 29 mei 2017 | Luuk d'Hooghe | <ul style="list-style-type: none"> - "MsgDateTime" hoeft niet in ISO-8601 formaat |
| 3.0 | 6 juni 2017 | Luuk d'Hooghe Cor Zijlstra | <ul style="list-style-type: none"> - 1.4.1 Klant toepassing Soapaction |
| 3.0 | 18 juli 2017* | Luuk d'Hooghe | <ul style="list-style-type: none"> - Sample Authentication verwijderd - Toegevoegd oasis document en een link in de documentatie opgenomen - Voorbeeld 1.3.3.3 verwijderd (blz 11) |

| | | | |
|-----|-----------------|---------------|---|
| 3.1 | 23 oktober 2017 | Luuk d'Hooghe | <ul style="list-style-type: none">- Naamgeving 3.0 -> 3.1- 1.3.3.3 password in plain text |
|-----|-----------------|---------------|---|

*betreft een bug-fix, de wsdl is niet aangepast

Inhoud

| | |
|--|-----------|
| Inleiding | 5 |
| 1. MessageService webservice | 7 |
| 1.1 Conceptuele uitwerking | 7 |
| 1.2 Gebruik webservice..... | 9 |
| 1.3 Afspraken..... | 10 |
| 1.4 Implicaties klant en leverancier | 15 |
| 2. Webmethods..... | 16 |
| 2.1 GetAvailableMessages..... | 16 |
| 2.2 GetMessage..... | 17 |
| 2.3 DeleteMessage | 18 |
| 2.4 PostMessage | 19 |
| 2.5 Beschrijving elementen gebruikt in de input/output parameters..... | 20 |

Inleiding

Binnen de bouw- en installatiebranche heeft Ketenstandaard Bouw en Installatie de afgelopen jaren gewerkt aan standaarden voor primaire transactieberichten zoals bestelling, orderbevestiging, pakbon en de factuur. Voor de uitwisseling van de betreffende berichten is geen standaard communicatieprotocol gedefinieerd met als gevolg een wildgroei aan communicatievormen. Vanuit de commissie communicatie standaarden (CCS) is verzocht te werken aan een standaard communicatieprotocol voor de uitwisseling van berichten. Daarvoor is vanuit de CCS de werkgroep 'communicatieprotocollen en webservices' opgericht. Een van de communicatieprotocollen die door de werkgroep is gedefinieerd betreft webservices. Dit document beschrijft een standaard voor de uitwisseling van berichten middels webservices. Inmiddels is dit de 3^e versie van deze service.

Eisen en randvoorwaarden

Door de werkgroep zijn een aantal eisen en randvoorwaarden benoemd waaraan het communicatieprotocol moet voldoen. Dit zijn:

1. Het communicatieprotocol moet laagdrempelig zijn voor de klant (aannemer/installateur). Het moet minimale eisen stellen aan de (IT-)omgeving/organisatie van de klant. Concreet betekent dit dat:
 - Het protocol moet standaard werken op een eenvoudige pc/laptop. Om te communiceren hoeft de klant geen (web)servers in te richten om de communicatie te faciliteren.
 - De communicatie via het internet verlopen. Dus geen aparte netwerken.
 - De communicatie moet gebruikmaken van standaard internetpoorten (80, 8080, 443). Dit omdat bij veel klanten bepaalde poorten geblokkeerd zijn.
 - Er is geen additionele software nodig om de communicatie te faciliteren, de communicatie moet rechtstreeks vanuit de software van de klant opgestart kunnen worden.
De communicatie en het te gebruiken protocol zijn vrij van kosten. Er is geen abonnement nodig voor bijvoorbeeld het gebruik van standaarden, platformen of licenties.
2. Het communicatieprotocol moet zowel synchrone als asynchrone communicatie faciliteren:
 - Met synchrone communicatie wordt een 'vraag-antwoord-spel' bedoeld tussen klant en leverancier. Uitgangspunt is dat de klant de vraag stelt en de leverancier het antwoord geeft (andersom is niet van toepassing). Bij synchrone communicatie stelt de klant een vraag aan de leverancier en in de response op de vraag (die binnen enkele seconden (max. 10) volgt) zit het antwoord van de leverancier. Denk daarbij aan het opvragen van de beschikbaarheid van een artikel.
 - Met asynchrone communicatie worden berichten bedoeld die van de ene partij naar de andere partij verstuurd worden, waarop niet direct antwoord/reactie verwacht wordt (de ontvangende partij dient wel aan te geven dat hij het bericht succesvol ontvangen heeft). Dit kunnen zowel berichten betreffen die van klant aan leverancier gestuurd worden als andersom van leverancier naar klant. Denk daarbij aan orders, order(her)bevestigingen, leverbonnen en facturen.

3. Het te kiezen protocol moet geen bijzondere eisen stellen aan het te gebruiken device, het protocol moet aan te roepen zijn vanuit servers, desktops, laptops, tablets en smartphones.
- Het protocol moet een dusdanige beveiliging hebben dat de inhoud van berichten via een beveiligde verbinding tussen partijen uitgewisseld kunnen worden.
 - Bij gebruik van het protocol moet de klant zich identificeren, daarvoor zijn een tweetal mogelijkheden:
 - Middels een gebruikersnaam en wachtwoord (eventueel aangevuld met een gebruikerscode) die door de leverancier verstrekt zijn.
 - Het protocol moet geen eisen stellen aan het bestandsformaat van het bericht dat via het protocol uitgewisseld wordt. Het moet dus mogelijk zijn zowel XML, JSON als ASCII berichten via het protocol uit te wisselen. Enige beperking die het protocol mag opwerken is een maximale berichtgrootte.
 - Het protocol moet ruimte bieden om de aanroepende applicatie van de klant te identificeren. Eventueel kan zo tussen de aanroepende applicatie van de klant en de ontvangende applicatie van de leverancier bilaterale implementaties gemaakt worden.
 - Het protocol moet zo ingericht worden dat het mogelijk is de communicatie naast rechtstreeks tussen klant en leverancier ook via een 3^e partij (HUB) te laten verlopen. In het geval van een 3^e partij communiceert de klant met de 3^e partij. De 3^e partij communiceert vervolgens met de leverancier, voor de communicatie tussen leverancier en 3^e partij zou in dat geval een ander protocol gebruikt kunnen worden.

Leeswijzer

Hoofdstuk 1 beschrijft de aanleiding van dit document en de eisen en randvoorwaarden die vanuit de CCS gesteld zijn aan de oplossingsrichting. In hoofdstuk 2 wordt het concept van de webservice besproken en wat de implicaties zijn voor zowel de klant als de leverancier.

Coproductie

Dit document wordt gezamenlijk onderhouden door Ketenstandaard Bouw en Installatie en GS1 Nederland. De stichting Ketenstandaard Bouw en Installatie heeft als doel het verbeteren van de processen tussen bedrijven die werkzaam zijn in, leveren aan of afnemer zijn van de bouw- en installatiebranche. Hierbij staat het gebruik van ICT-toepassingen in de communicatie over inkoop/verkoop transacties centraal.

Voor vragen of opmerkingen over dit document kunt u contact opnemen met info@ketenstandaard.nl

1. MessageService webservice

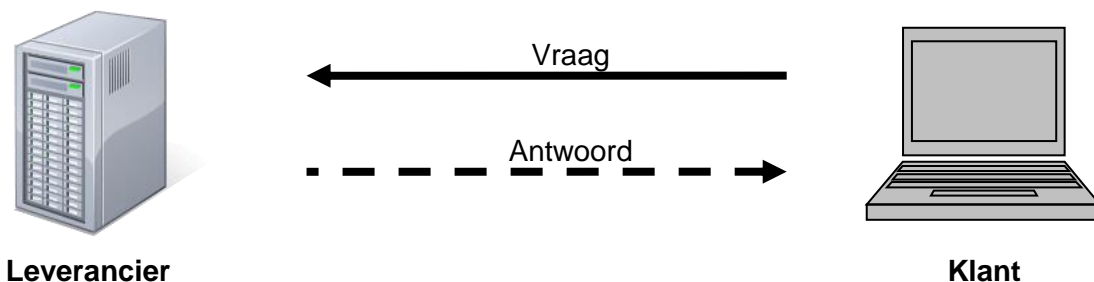
In dit hoofdstuk wordt de gekozen oplossingsrichting beschreven, daarvoor wordt in de eerste paragraaf in hoofdlijnen uitgelegd hoe de webservice werkt. Vervolgens wordt functioneel in meer detail de webservice beschreven waarna vervolgd wordt in een set met afspraken. Als laatste worden de implicaties voor zowel de klant als de leverancier beschreven.

1.1 Conceptuele uitwerking

De communicatie tussen de klant en de leverancier kan getriggerd worden vanuit beide partijen. Gezien het feit dat de CCS bepaald heeft geen eisen aan de omgeving van de klant te willen stellen moet afhankelijk van de partij die de communicatie wil opstarten een andere methode gekozen worden voor de uitwisseling van de berichten. Dit wordt in onderstaande twee subparagrafen verder uitgewerkt.

1.1.1 Toepassing

Wanneer de klant een bericht aan de leverancier stuurt is dit eenvoudig in te richten. Uitgangspunt is dat de leverancier (of hub) een webserver ingericht heeft met daarop de webservice waar de klant tegenaan kan 'praten'. Dit ziet er schematisch als volgt uit:



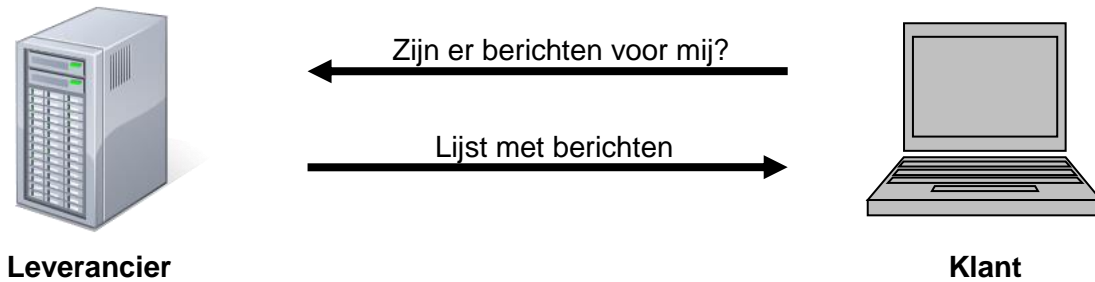
Figuur 1.1: schema communicatie

Bovenstaand schema kan zowel voor synchrone (vraag-antwoord spel) alsook voor asynchrone communicatie gebruikt worden. Bovenstaand schema kan ook gebruikt worden voor communicatie via een hub.

1.1.2 Communicatie van leverancier → klant

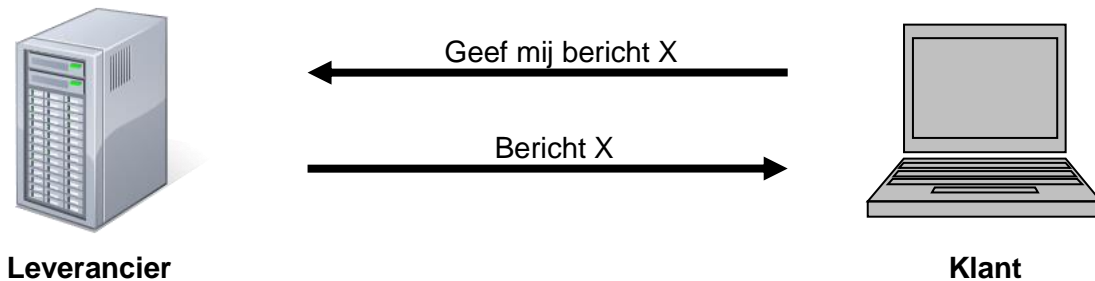
Wanneer de leverancier een bericht wil sturen aan de klant (a-synchroon) dan is de moeilijkheid dat in de randvoorwaarden staat dat aan de (IT) omgeving van de klant geen eisen gesteld mogen worden. Derhalve kan het schema zoals dat in de vorige paragraaf staat niet eenvoudigweg omgedraaid worden omdat de klant dan zou moeten beschikken over een webserver. Daarom is gekozen voor een andere oplossing waarbij de leverancier de berichten cached op zijn webserver en de klant periodiek de berichten ophaalt. Geadviseerd wordt om minimaal 1 maal per dag te controleren of er berichten zijn, daarnaast moet de gebruiker in de software zelf het initiatief kunnen nemen om te bekijken of er nog berichten klaarstaan. Schematisch ziet dat er dan als volgt uit:

Allereerst vraagt de klant een lijst met berichten op die klaarstaan bij de leverancier. De leverancier geeft terug welke berichten er klaar staan.



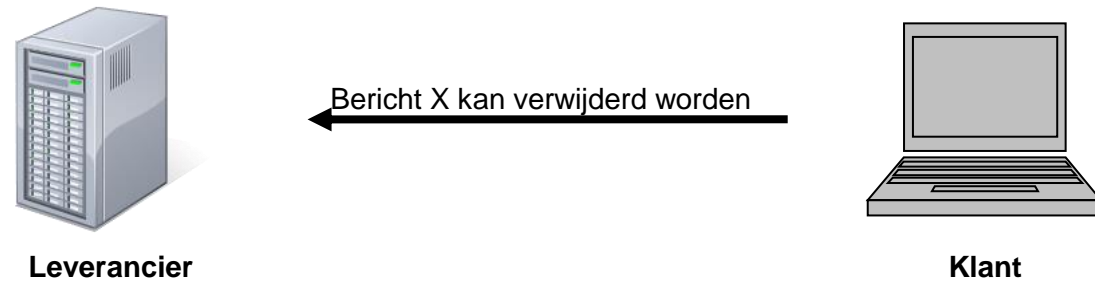
Figuur 1.2: schema berichten die klaar staan

Vervolgens vraagt de klant elk bericht (een voor een) op, de leverancier geeft in het antwoord het bericht terug.



Figuur 1.3: schema opvragen berichten

Als de klant het bericht succesvol ontvangen heeft geeft hij dat aan bij de leverancier zodat de leverancier het bericht kan verwijderen (of markeren als opgehaald).



Figuur 1.4: schema ontvangen, markeren of verwijderen bericht

1.2 Gebruik webservice

De leverancier dient zelf de klanten te registreren en bij te houden wat voor afspraken er zijn gemaakt met deze klant (m.b.t. type berichten e.d.). Daarnaast kan er bilateraal geschakeld worden voor de verwerking met behulp van de meegestuurde **ApplicationId** en **VersionId** van de applicatie.

De klant zal met een bepaalde interval de functie 'GetAvailableMessages' aanroepen. Hierbij kan de klant specificeren welk berichttype hij wil ontvangen. De leverancier zal een lijst terugsturen van de gevraagde of alle berichten die klaarstaan. Per bericht zal de leverancier aangeven in welke **MsgFormat** en in welke **MsgVersion** hij het betreffende bericht kan aanleveren, eventueel kan de leverancier aangeven dat hij in meerdere formaten/versies kan aanleveren.

Voor ieder bericht dat de klant wenst te verwerken zal hij de 'GetMessage' functie aanroepen met het **MsgId** van het bericht dat hij wil ontvangen en het **MsgFormat** en de **MsgVersion** waarin hij het bericht wenst te ontvangen. Als het bericht goed ontvangen is zal de klant vervolgens het bericht verwijderen door de functie 'DeleteMessage' aan te roepen. De klant dient een bericht dat hij goed heeft ontvangen altijd te verwijderen, ongeacht of het bericht ook goed door de klant is te verwerken. Daarmee komt het bericht niet meer voor bij de volgende opvraging van de lijst van klaarstaande berichten. De klant is zelf verantwoordelijk voor de foutafhandeling van de verwerking en het bewaren en signaleren van niet goed verwerkte berichten.

Indien de klant een bericht wil versturen aan de leverancier dan gebruikt hij hiervoor de functie 'PostMessage'. De klant bepaalt hier zelf een **MsgId** (bijvoorbeeld een guid/uuid) die door de leverancier gebruikt wordt. Indien de leverancier op basis van het ontvangen bericht direct (synchroon) een bericht terug wil en kan sturen dan kan hij hiervoor het **Message** element gebruiken waarin meteen een bericht kan worden teruggezonden. Anders zal een eventueel antwoord van de leverancier (asynchroon) moeten worden opgehaald door een aanroep van de functie 'GetAvailableMessages' met het **MsgType** van het antwoord als parameter bv. "ORDRSP". De klant kan dan zijn specifieke bericht matchen met de **MsgId** die hij met het bericht verstuurd heeft.

Een voorbeeld is een orderbericht. Als de klant een orderbericht via de functie 'PostMessage' verzendt aan de leverancier, dan kan de leverancier de orderbevestiging in de **Message** response element zetten. Let op: zodra de klant de response van de 'PostMessage' correct heeft verwerkt moet hij een 'DeleteMessage' uitvoeren, anders zal de Response ook aangeboden worden in de volgende 'GetAvailableMessages' aanroep. Op deze manier wordt ondersteund dat klanten die niet om kunnen gaan met synchrone verwerking deze berichten missen.

1.3 Afspraken

Deze webservice is samengesteld volgens de richtlijnen opgesteld door Ketenstandaard Bouw en Installatie.

Een aantal afspraken worden hieronder nog expliciet toegelicht of verscherpt ten behoeve van deze webservice.

1.3.1 Hoofdlettergebruik

De in deze webservice gebruikte webmethods, nodes en elementen dienen in UpperCamelCase (ook wel Pascal case genoemd) te worden geschreven, zodat de SoapAction, nodig voor Soap 1.1 tussen de verschillende leveranciers vergelijkbaar is. Concreet betekent dit dat er altijd gestart wordt met een hoofdletter en dat elk volgend woord ook start met een hoofdletter. Afkortingen zoals ID worden gezien als een woord, derhalve in UpperCamelCase geschreven als Id.

```
<ApplicationId>
```

Figuur 1.1: voorbeeld

1.3.2 Beveiliging van de communicatie

De Ketenstandaard Bouw en Installatie richtlijnen schrijven voor dat communicatie over HTTP of HTTPS moet geschieden. Vanwege het toepassingsgebied (bijv. uitwisseling van ORDERS, FACTUREN) wordt bij deze service verplicht gesteld dat communicatie over HTTPS loopt. Aanvullende motivatie om HTTPS te gebruiken is vanwege het feit dat inloggegevens worden verstuurd in de webservice.

1.3.3 Gebruik namespaces

Conform de richtlijnen moet voor de Soap-header elementen gebruik gemaakt worden van de namespace.

Hierbij dient opgemerkt te worden dat als het **MsgContent** element een XML-document bevat er vanuit dit document een verwijzing naar een geldige namespace opgegeven is. Een dergelijk document dient tevens gevalideerd te zijn tegen de bijbehorende XSD.

1.3.3.1 Authenticatie

De service bevat een policy voor WS-I Basic Security Profile, hiermee wordt dmv username / password de authenticatie geregeld. Een SOAP header fragment met een voorbeeld is in de documentatieset beschikbaar.

1.3.3.2 Custom Info

Conform de richtlijnen maakt deze service gebruik van het **CustomInfo** element. Voor deze webservice is het gebruik van het **CustomInfo** element verplicht.

Voor deze webservice is het ApplicationId element verplicht. De ApplicationId is relevant, omdat de leverancier op basis daarvan de berichten (en versies) die voor de betreffende applicatie klaarstaan terug kan sturen. Het VersionId is losgetrokken van de ApplicationId, in verband met updates en het eventueel bilateraal kunnen schakelen op de inhoud/afhandeling van de calls/berichtenset.

Voor deze webservice is het **IsTestMessage** element verplicht. Dit element geeft aan of dit bericht een testbericht is ("true") of een productiebericht ("false"). Overigens wordt geadviseerd voor test-, acceptatie- en productieomgevingen aparte URL's aan te houden.

RelationId: dit betreft de inlogcode van de aanroepende partij. Dit kan bijvoorbeeld een gebruikersnaam, debiteurnummer, crediteurnummer of een GLN zijn. Wanneer gebruik gemaakt wordt van authenticatie op basis van credentials is dit veld verplicht.

Het **LanguageCode** element is optioneel. De code is uitgedrukt in een string (lengte 2, ISO-formaat:3166-1). Dit element geeft de client de mogelijkheid om aan te geven in welke taal er gecorrespondeerd moet worden, zoals artikel info, functionele foutmeldingen, etc. Indien dit element niet meegegeven wordt is de default 'NL'.

Het **IsContentCompressed** element is optioneel. Dit element geeft aan of de inhoud van de body tag gecomprimeerd is ("true") of niet ("false"). Er kan gekozen worden voor een opzet waarbij de berichten gezip worden en de zip als BASE64 string in de <Soap:Body> response bericht wordt geplaatst. Voorstel om alleen te zippen bij artikel- en conditiebestanden, rest van de berichten loont het zippen niet, hiervoor kan het gzip formaat gebruikt worden. Deze techniek kan bilateraal worden afgesproken en is enkel van toepassing indien er geen gebruik wordt gemaakt van http-compressie (gzip). Wanneer dit element niet meegegeven is kan aangenomen worden dat de content niet gecomprimeerd is.

Het element **IsProcessedSuccessful** wordt niet gebruikt. Dit element is overbodig, omdat indien een bericht onjuist is verwerkt er altijd een Soap Fault teruggegeven dient te worden.

1.3.3.3 Voorbeeld

Hier wordt verwezen naar:

Web Services Security UsernameToken Profile 1.0.pdf

<https://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>

(Toelichting: password versturen in plain tekst)

1.3.4 XML en vreemde tekens in MsgContent

De inhoud van het **MsgContent** element kan bestaan uit XML document of een Base64 encoded string. De XML moet daarvoor XML encoded worden opgenomen in de SOAP body.

```
<Soap:Body>
  <MsgProperties>
    <MsgId>?</MsgId>
    <MsgDateTime>?</MsgDateTime>
    <MsgFormat>?</MsgFormat>
    <MsgVersion>?</MsgVersion>
    <MsgType>?</MsgType>
  </MsgProperties>
  <Message>
    <MsgContent>
      &lt;?xml version=&quot;1.0&quot;?&gt;
      &lt;xml-bericht&gt;
      &lt;/xml-bericht&gt;
    </MsgContent>
  </Message>
</Soap:Body>
```

Figuur 1.6: voorbeeld

1.3.5 Gebruik logging

Ketenstandaard Bouw en Installatie adviseert om zoveel mogelijk te loggen. Vanwege het toepassingsgebied van deze webservice wordt dit bij deze service uitdrukkelijk geadviseerd. Bij voorkeur de complete Soap requests en responses.

1.3.6 Foutafhandeling

Als tijdens de verwerking van het bericht een probleem is opgetreden, wordt een SOAP bericht met een <Fault> element teruggezonden. Dit geldt zowel voor synchrone als asynchrone communicatie. Tevens geldt dit voor alle soorten fouten, technisch en functioneel.

De werking van het Soap Fault element is beschreven op <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>. Het Soap Fault element bevat twee verplichte subelementen: **FaultCode** en **FaultString**. Het **FaultCode** element bevat een (gestandaardiseerde) foutcode: "VersionMismatch", "MustUnderstand", "Client" of "Server". Het **FaultString** element bevat een beschrijving van de fout, bedoeld voor de gebruiker. Voor de MessageService wordt geadviseerd om ook het optionele **Detail** element te gebruiken en dit te vullen met een **FaultDetails** element. Dit element bestaat dan verplicht uit een **ErrorCode** dat een id bevat

waarmee de foutmelding te herkennen is (zie tabel hieronder). Deze id kan afgevangen worden door de ontvangende software. Tevens bevat de **FaultDetails** ook verplicht een **Message** element dat een tekstuele toelichting op de foutmelding bevat. Hierbij wordt geadviseerd gebruik te maken van het “xml:lang” attribuut. Voor de vulling van de **FaultDetails** onderkennen we in deze webservice de volgende foutmeldingen:

| ErrorCode | Message |
|------------------|---|
| 110 | Aanroep mislukt, de backoffice is momenteel niet beschikbaar. |
| 120 | Inloggen mislukt |
| 121 | Authenticatie error, credentials onjuist. |
| 130 | Autorisatie mislukt, u heeft niet de juiste bevoegdheid. |
| 140 | Aanroep mislukt, ApplicationId ongeldig. |
| 141 | Aanroep mislukt, ApplicationId verplicht. |
| 160 | Aanroep mislukt, MsgType onbekend. |
| 161 | Aanroep mislukt, MsgType verplicht. |
| 170 | Aanroep mislukt, MsgId onbekend. |
| 171 | Aanroep mislukt, MsgId verplicht. |
| 172 | Aanroep mislukt, MsgId is nog niet opgehaald. |
| 173 | Aanroep mislukt, MsgId is al eerder gewist. |
| 180 | Aanroep mislukt, MsgFormat onbekend. |
| 181 | Aanroep mislukt, MsgFormat verplicht. |
| 182 | Aanroep mislukt, Gevraagd MsgFormat niet beschikbaar |
| 190 | Aanroep mislukt, MsgVersion onbekend. |
| ErrorCode | Message |
| 191 | Aanroep mislukt, MsgVersion verplicht. |
| 192 | Aanroep mislukt, Gevraagd MsgVersion niet beschikbaar |
| 199 | Ongedefinieerde fout |
| 9xx | Range voor vrij te definiëren foutmeldingen |

Tabel 1.1: overzicht foutmeldingen

```

<Soap:Body>
  <Soap:Fault>
    <faultcode>Soap:Server</faultcode>
    <faultstring>Server error</faultstring>
    <detail>
      <FaultDetails>
        <ErrorCode>199</ErrorCode>
        <Message xml:lang='nl'>Er is een ongedefinieerde fout opgetreden,
neem contact op met 000-0000000.</Message>
      </FaultDetails>
    </detail>
  </Soap:Fault>
</Soap:Body>

```

Figuur 1.7: voorbeeld foutbericht

Met uitzondering van bovengenoemde 'header' errorcodes geldt het onderstaande. Het detail element is bestemd voor toepassing specifieke foutinformatie over het Body element. Het moet aanwezig zijn indien de inhoud van de Body elementen niet met succes kon worden verwerkt. Het mag niet worden gebruikt voor error(s) van Header elementen. Gedetailleerde error informatie die behoort tot de Header moeten binnen de Header afgehandeld worden.

De afwezigheid van het detail element in het Fault element geeft aan dat de storing niet is gerelateerd aan de verwerking van het Body element. Dit kan worden gebruikt om onderscheid te maken of het Body element kon worden verwerkt of niet in het geval bij een foutsituatie.

Alle direct onderliggende elementen van het detail element worden detail data genoemd en elk detail item wordt gecodeerd als een zelfstandig onderdeel binnen de detail element.

1.4 Implicaties klant en leverancier

De gekozen oplossingsrichting faciliteert de communicatie tussen twee partijen. Echter zullen leveranciers zakendoen met meerdere klanten en klanten zakendoen met meerdere leveranciers. Derhalve dienen beide partijen in hun software hier rekening mee te houden. In deze paragraaf wordt voor beide partijen de consequenties beschreven.

1.4.1 Klant

- Een uniek ApplicationId hebben dat gebruikt wordt in de communicatie zodat de leverancier weet met welke softwareapplicatie 'gepraat' wordt en op basis hiervan kan vastleggen welke berichtenversies uitgewisseld moeten worden. Geadviseerd wordt om voor het ApplicationId een GLN te gebruiken.
- Per leverancier moet vastgelegd worden:
 - URL waarop de webservice van de leverancier bereikbaar is, het zogenaamde end point.
 - Gebruikersnaam (eventueel aangevuld met een relatie code) en een wachtwoord waarmee de klant kan inloggen bij de leverancier.

Alle partijen die gebruik maken van deze standaard webservice kunnen de verbinding oproepen met dezelfde client, slechts de hierboven genoemde gegevens verschillen en dienen instelbaar te zijn per leverancier.

1.4.2 Leverancier

De leverancier kan:

- Per ApplicationId (en eventueel versie) vastleggen:
 - Welke berichttypen het softwarepakket kan ontvangen.
 - In welk berichtformaat de software de berichten wil ontvangen.
 - Welk versienummer van dat berichtenformaat de software wil ontvangen.
- Per klant vastleggen welke berichten de klant wil ontvangen in welk ApplicationId.
- De berichten vastleggen in een database per klant/ApplicationId.
- Bij berichten vastleggen wanneer het bericht aangemaakt is in de database, monitoren of klanten het opgevraagd hebben en wanneer na een bepaalde periode het bericht nog niet opgevraagd is door de klant kan er een actie ondernomen worden.

2. Webmethods

Dit zijn de daadwerkelijke functies die de verwerking van het bericht afhandelen. Hiervoor gebruiken we een specifieke naamgeving die duidelijk het type functie weergeeft:

- **Post:** geeft aan dat de input moet verwerkt/opgeslagen worden in de backoffice.
- **Get:** geeft aan dat er enkel informatie wordt opgevraagd uit de backoffice.
- **Delete:** geeft aan dat de betreffende input moet worden verwijderd in de backoffice.

2.1 GetAvailableMessages

Geeft een lijst terug van alle berichten van het opgegeven **MsgType**. Indien geen **MsgType** wordt meegegeven dienen alle berichten (ongeacht berichttype) teruggegeven te worden. De return value van de functie is een array met **AvailableMessagesRequest** elementen.

| <u>Parameters</u> | <u>Type</u> |
|-------------------|--|
| Input | MsgType (string) optioneel |
| Output | AvailableMessagesResponse [] |
| Faults | 110 Aanroep mislukt, de backoffice is momenteel niet beschikbaar. 120 Inloggen mislukt 121 Authenticatie error, credentials onjuist. 130 Autorisatie mislukt, u heeft niet de juiste bevoegdheid. 140 Aanroep mislukt, ApplicationId ongeldig. 141 Aanroep mislukt, ApplicationId verplicht. 160 Aanroep mislukt, MsgType onbekend. 161 Aanroep mislukt, MsgType verplicht. |

Tabel 2.1: overzicht

2.2 GetMessage

Deze functie wordt aangeroepen met een **MsgId** van het bericht dat men wil ophalen aangevuld met het **MsgFormat** en de **MsgVersion** waarin de klant het bericht wenst te ontvangen. Je kunt per aanroep dus maximaal een message opvragen. De functie geeft een **MessageRequestResponse** element terug met daarin het bericht.

| <u>Parameters</u> | <u>Type</u> |
|-------------------|--|
| Input | MsgId (string) verplicht MsgFormat (string) verplicht MsgVersion (string) verplicht |
| Output | MessageRequestResponse [] |
| Faults | 110 Aanroep mislukt, de backoffice is momenteel niet beschikbaar. 120 Inloggen mislukt 121 Authenticatie error, credentials onjuist. 130 Autorisatie mislukt, u heeft niet de juiste bevoegdheid. 140 Aanroep mislukt, ApplicationId ongeldig. 141 Aanroep mislukt, ApplicationId verplicht. 170 Aanroep mislukt, MsgId onbekend. 171 Aanroep mislukt, MsgId verplicht. 173 Aanroep mislukt, MsgId is al eerder gewist. 180 Aanroep mislukt, MsgFormat onbekend. 181 Aanroep mislukt, MsgFormat verplicht. 182 Aanroep mislukt, Gevraagd MsgFormat niet beschikbaar 190 Aanroep mislukt, MsgVersion onbekend. 191 Aanroep mislukt, MsgVersion verplicht. 192 Aanroep mislukt, Gevraagd MsgVersion niet beschikbaar |

Tabel 2.2: overzicht

2.3 DeleteMessage

De functie wordt aangeroepen met het **MsgId** dat moet worden verwijderd. Per aanroep kan maximaal 1 Message verwijderd worden. Deze functie moet worden aangeroepen nadat een bericht is opgehaald via de 'GetMessage' functie ongeacht of het bericht ook succesvol kan worden verwerkt. De service geeft een **MessageResult** terug waarin aangegeven is of het bericht succesvol verwijderd is. Wanneer een **MsgId** niet verwijderd kan worden wordt een foutmelding (Soap Fault) teruggegeven.

| <u>Parameters</u> | <u>Type</u> |
|-------------------|--|
| Input | MsgId (string) verplicht |
| Output | MessageResult (boolean) |
| Faults | 110 Aanroep mislukt, de backoffice is momenteel niet beschikbaar. 120 Inloggen mislukt 121 Authenticatie error, credentials onjuist. 130 Autorisatie mislukt, u heeft niet de juiste bevoegdheid. 140 Aanroep mislukt, ApplicationId ongeldig. 141 Aanroep mislukt, ApplicationId verplicht. 170 Aanroep mislukt, MsgId onbekend. 171 Aanroep mislukt, MsgId verplicht. 172 Aanroep mislukt, MsgId is nog niet opgehaald. 173 Aanroep mislukt, MsgId is al eerder gewist. |

Tabel 2.3: overzicht

2.4 PostMessage

Biedt een bericht aan dat door de webservice moet worden opgeslagen en verwerkt. Dit wordt gedaan middels het **Message** element met daarin het bericht. De service geeft middels **MessageResult** aan of het succesvol ontvangen is, optioneel kan direct een antwoord gegeven worden via het **Message** element met daarin het bericht. Het **MsgFormat** en **MsgVersion** van de response is altijd gelijk aan die van de aanroep.

| <u>Parameters</u> | <u>Type</u> |
|-------------------|--|
| Input | MessageServiceMessage[] |
| Output | MessageResult (boolean) Message[] optioneel |
| Faults | 110 Aanroep mislukt, de backoffice is momenteel niet beschikbaar. 120 Inloggen mislukt 121 Authenticatie error, credentials onjuist. 130 Autorisatie mislukt, u heeft niet de juiste bevoegdheid. 140 Aanroep mislukt, ApplicationId ongeldig. 141 Aanroep mislukt, ApplicationId verplicht. 170 Aanroep mislukt, MsgId onbekend. 171 Aanroep mislukt, MsgId verplicht. |

Tabel 2.4: overzicht

Wanneer er door de leverancier niet direct een antwoordbericht teruggestuurd kan worden, dan wordt enkel het **MessageResult** element teruggegeven.

2.5 Toelichting op gebruik van bepaalde elementen

In deze paragraaf worden bepaalde elementen die in de diverse webmethods gehanteerd worden toegelicht.

2.5.1 MsgType

Het **MsgType** is een string die het type bericht aanduidt. Het idee is dat hier ook de mogelijkheid blijft bestaan om eigen **MsgType** waarden te definiëren. Door alle eigen **MsgType** met 'CST_' en een unieke code van de leverancier te prefixen voorkomen we dat eigen MsgTypes gaan conflicteren met mogelijk nieuw gedefinieerde standaard berichten. Door het toestaan van eigen MsgTypes ontstaat de mogelijkheid om ook berichten te versturen welke bilateraal met elkaar zijn afgestemd.

De statische lijst van types kan het best door de leverancier als een string van enumeraties worden weergegeven.

Standaard: (string, 50)

- PRICAT Artikelgegevens (van leverancier naar klant).
- PRODAT Productgegevens (van leverancier naar klant).
- TRMMSG Terms Message (conditiebestand) (van leverancier naar klant).
- REQOTE Offerteaanvraag (van klant naar leveranciers).
- QUOTES Offerte (van leverancier naar klant).
- ORDERS Inkooporder, bestelling (van klant naar leverancier).
- ORDCHG Orderwijziging (van klant naar leverancier).
- ORDRSP Orderbevestiging (van leverancier naar klant).
- ORSSTA Order status opvraag (van klant naar leverancier).
- ORDREP Order status rapport (van leverancier naar klant).
- DESADV Paklijst (van leverancier naar klant).
- INVOIC Factuur (van leverancier naar klant).
- COLINV Verzamelfactuur (van leverancier naar klant).
- INVRPT Inventory report (van klant naar leverancier).
- MTNINS MaintenanceInstruction (Onderhoudsopdracht)
- MTNSTA MaintenanceStatus (Onderhoudsstatus)

Custom voorbeelden: (string, 50)

- CST_SELL_GETARTICLEINFO Opvragen artikelinformatie bij Sell.
- CST_SELL_GETBASKET Opvragen stuklijsten bij Sell.

2.5.2 MsgFormat

Het **MsgFormat** is een string die de standaard van het bericht benoemt, bijvoorbeeld INSBOU. Wanneer er geen sprake is van een standaard dan wordt een custom formaat benoemd.

De statische lijst van formaten kan het best door de leverancier als een string van enumeraties worden weergegeven.

Binnen de MessageService worden de volgende formaten gedefinieerd:

Standaard: (string, 50)

- SALES De XML-standaard van Ketenstandaard Bouw en Installatie (INSBOU004).
- INSBOU Geharmoniseerde standaard Bouw- en Installatiebranche (INSBOU003).
- ETIM De classificatie standaard (ASCII en XML).
- D96A De EDI standaard D96A.
- CUSTOM Eigen benoemd berichtformaat, wordt verder uitgewerkt in MsgType.

2.5.3 MsgVersion

Het **MsgVersion** is een string van max. 50 posities die het versienummer van het bericht aanduidt. Het versienummer wordt exact volgens de schrijfwijze zoals deze in de standaard gedefinieerd is opgenomen.

2.5.4 MsgId

Het **MsgId** is een string van max. 50 posities die de message binnen de leverancier uniek aanduidt. De voorkeur gaat naar het gebruik van guids/uuids. De **MsgId**'s mogen in principe wel hergebruikt worden zolang het **MsgId** een bericht bij het aanroepen van 'GetMessage' en 'DeleteMessage' uniek kan identificeren.

2.5.5 MsgDateTime

Het **MsgDateTime** is een date element. Dit betreft altijd de datum en tijd waarop het originele bericht is opgesteld.

2.5.6 MsgContent

Het **MsgContent** element is een string met een onbeperkte lengte. De **MsgContent** bevat de daadwerkelijke inhoud van het bericht. Indien hier een XML bericht in wordt geplaatst dan dient er geen SOAP-envelop gebruikt te worden in dit XML bericht en dient de XML-inhoud XML encoded te zijn met karakter encoding UTF-8. ASCII, JSON en andere berichten moeten als BASE64 string worden gecodeerd.

2.5.7 Attachment

Bij sommige berichten kunnen ook attachments meegestuurd worden, hiervoor wordt het **Attachment** element gebruikt. Het **Attachment** element is een optioneel element en kan meerdere keren voorkomen. De attachment(s) horen 1 op 1 bij het document in de **MsgContent**. Bijvoorbeeld een pdf-bestand bij een ICF of XML-factuur, of een gescand werkbriefje bij een order. De bijlagen moeten als BASE64 string zijn opgenomen in het **AttachedData** element of er moet verwezen worden middels een **URL** naar een locatie op het internet. Een **URL** heeft niet de voorkeur als hierin (leesbare) inloggegevens voorkomen.

Let op: in de INSBOU003 XML-factuur berichten is het Attachment element reeds in het bericht zelf opgenomen. Derhalve dient de attachment dan in het bericht zelf opgenomen te worden, omdat die voorziening daar al aanwezig is.

Het **DocumentType** betreft een code die het soort document beschrijft. Voor deze code is de volgende enumeratielijst vastgesteld:

- DOC Bestand.
- FAC Afbeelding factuur.
- OTH Anders.
- PIC Afbeelding.

Het **FileType** betreft een code die het bestandstype beschrijft. Voor deze code is de volgende enumeratielijst vastgesteld:

- DOC
- GIF
- JPG
- PDF
- PNG
- XLS
- XML
- ZIP