

## Verslag leverancieroverleg Toepasbare regels 10 feb 2025

### STTR 3.0 – opsomming van geplande wijzigingen

17 feb 2025

#### A. Vooraf

Dit document is een combinatie van de wijzigingen m.b.t. STTR 3.0 en een beknopt verslag van het bespreken van deze wijzigingen op 10 feb 2025.

Groene tekst is een samenvatting van wat tijdens het document is besproken. Rode tekst na het overleg toegevoegd. Het bevat de volgende onderdelen

##### A. Vooraf

B. Planning STTR-versie (besproken tijdens het overleg)

C. Geplande wijzigingen: de lijst van wijzigingen die tijdens het overleg is besproken

D. N.a.v. het bespreken van de lijst: zijn er aanvullingen/opmerkingen?

E. Dynamische vragen 2.0: nieuwe versie van API uitvoeren services

F. Nieuwe verificatie maximum grootte STTR-bestand

#### B. Planning STTR-versie

Er zijn geen vragen over de getoonde planning van DSO.

Eén leverancier meldt expliciet geen problemen te voorzien.

DSO-LV heeft behoefte aan individuele feedback per leverancier.

#### C. Geplande wijzigingen: Inleiding

In de STTR standaard gaat in versie 3.0 het nodige wijzigen. Hier een lijst van alle zaken die worden aangepakt, met daarbij een korte toelichting. Een aantal van de voorgenomen wijzigingen zijn al behandeld in het CABS. Er is onderscheid gemaakt tussen wel en niet behandelde wijzigingen.

### Nog te behandelen wijzigingen

De lijst in deze paragraaf bevat wijzigingen die nog niet door het CABS zijn beoordeeld. Of wijzigingen die al wel zijn behandeld, maar waar in de tussentijd een aanpassing aan is gedaan. Als er al een CABS-verzoek is gerelateerd aan een wijziging, staat tussen haakjes het CABS-nummer erbij.

#### 1. (CABS-206): Splitsing document Specificatie STTR en verificaties afsplitsen van Specificatie IMTR.

Huidige STTR/IMTR-documentatie documentatie bevat naast de technisch-inhoudelijke beschrijving ook andere informatie (bijvoorbeeld over het aanleveren van de STTR-bestanden, de werking van het algoritme en content-gerelateerde richtlijnen) die strikt genomen geen onderdeel is van de standaard.

Op dit moment zijn de volgende documenten m.b.t. toepasbare regels beschikbaar:

- Specificatie STTR
- Specificatie IMTR (inclusief verificaties, exclusief verificaties Aanleveren)
- Het STTR Lagenmodel nader uitgelegd
- Koppelvlak aanleveren toepasbare regels (exclusief verificaties Aanleveren)
- Toepasbare activiteiten en locaties in relatie tot toepasbare regels

Het voorstel is om de niet aan de standaard gerelateerde informatie uit de Specificatie STTR te halen en deze in andere gebruikersdocumenten onder te brengen. Daarnaast zullen de verificaties uit de Specificatie IMTR afgesplitst worden en in een aparte verificatie-matrix worden opgenomen (zie punt 16). De documenten gerelateerd aan de standaard STTR worden dan:

- Specificatie STTR (focus op inhoud en structuur van STTR-bestanden; exclusief aanleveren, werking, adviezen m.b.t. content, etc.)
- Specificatie IMTR (exclusief verificaties)
- Verificatiematrix (uit IMTR, aangevuld met specifieke aanlever-verificaties), zie punt 16.

Geen opmerkingen.

## 2. STTR versie 3.0

In de STTR kan worden aangegeven van welke versie deze is. Hiervoor wordt gebruik gemaakt van een enumeratie. Die moet worden aangepast.

Wijziging	Omschrijving	Waar	Impact
STTR versie enumeratie aanpassen	Aan de enumeratie moet '3' als waarde aan worden toegevoegd. En voor STTR versie 3.0 zijn de waarden 1 en 2 niet meer toegestaan. Dus die worden verwijderd.	bedrijfsregel.xsd	Breaking change

```
<xs:simpleType name="sttrVersie">
  <xs:restriction base="xs:string">
    <del>xs:enumeration value="1"/>
    <del>xs:enumeration value="2"/>
    <xs:enumeration value="3"/>
  </xs:restriction>
</xs:simpleType>
```

Figuur 1 - aanpassing enumeratie sttrVersie

Nadere uitleg: Het is nog onduidelijk om de enumeratie te handhaven. Er wordt gekeken of het op een andere manier kan worden opgelost zodat er bij nieuwe versies geen sprake is van een breaking change.

Verder geen opmerkingen.

Dit is voor STTR 3.0 de definitieve implementatie; er komt dus geen andere manier.

## 3. (CABS-137) Aansluiten Informatieproducten

Tijdens workshops waarin met het DSO is samengewerkt om de registers van RCE, BAG en BeSi goed te kunnen bevragen, is gebleken dat aan de uitvoeringsregel van het type 'Registerbevraging API profiel' wat aanpassingen nodig waren. Deze aanpassingen zijn:

Wijziging	Omschrijving	Waar	Impact
afnamePunt: element endpointStatus is verwijderd	Dit element is ooit bedacht met als doel om te kunnen controleren of een endpoint daadwerkelijk bestaat. Maar de achterliggende	uitvoeringsregel.xsd	Breaking change

	functionaliteit is nooit gerealiseerd (en inmiddels ook achterhaald). Vandaar dat dit element kan worden verwijderd		
afnamePunt: element registerNaam is toegevoegd	Dit element zorgt ervoor dat in de STTR en in de DMN viewer de naam van het register wat geraadpleegd wordt met de registerbevraging zichtbaar is	uitvoeringsregel.xsd	Breaking change
registerKetenVariabele: waarden 'activiteitUrn', 'werkzaamheidUrn' en 'datum' zijn toegevoegd aan de enumeratie	in de werkplaats over aansluiten informatieproducten is gebleken dat er behoefte was om meerdere soorten informatie richting een register te sturen. Daarom is de lijst van typen gegevens in de registerKetenVariabele uitgebreid	uitvoeringsregel.xsd	Non-breaking change

**Let op:** CABS-137 is in het verleden al beoordeeld en goedgekeurd. Het verwijderen van het element endpointStatus is sinds die beoordeling echter toegevoegd aan de wijzigingen.

```
<xs:element name="afnamepunt">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="endpointAfnamepunt" type="dataTypes:nietLegeString"/>
      <xs:element name="endpointStatus" type="dataTypes:nietLegeString" minOccurs="0"/>
      <xs:element name="registerNaam" type="dataTypes:nietLegeString"/>
      <xs:element ref="uitv:query"/>
      <xs:element ref="uitv:criteria"/>
      <xs:element ref="uitv:projectie"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

*Figuur 2 - verwijdering van endpointStatus en toevoeging van registerNaam*

```
<xs:simpleType name="registerKetenVariabele">
  <xs:restriction base="xs:string">
    <xs:enumeration value="geometrie"/>
    <xs:enumeration value="activiteitUrn"/>
    <xs:enumeration value="werkzaamheidUrn"/>
    <xs:enumeration value="datum"/>
  </xs:restriction>
</xs:simpleType>
```

*Figuur 3 - toegevoegde waarden aan enumeratie registerKetenVariabele*

Nadere uitleg: Als een register is aangesloten moet deze geregistreerd zijn in de API store. Endpointstatus is dan niet meer nodig. Daar is ook nooit functionaliteit voor gebouwd. Verder geen opmerkingen.

#### 4. (CABS-205) Dynamische Vragen 2.0

Met de introductie van Dynamische Vragen 2.0 is een element, dat in het verleden aan de STTR standaard is toegevoegd om binnen Dynamische Vragen 1.0 bepaald gedrag mogelijk te maken, overbodig geworden. Daarnaast is geconstateerd dat een situatie die in DMN wel mogelijk is, niet werkt i.c.m. Dynamische Vragen 2.0. Er wordt een verificatie gebouwd om deze situatie niet meer toe te staan.

Wijziging	Omschrijving	Waar	Impact
Het element '<dynamischeVragenUit />' komt te vervallen	Dit element is binnen het Dynamische Vragen 1.0 mechanisme geïntroduceerd om het in DMN-decisions met hitPolicy 'COLLECT' mogelijk te maken om tóch aanvullende vragen te blijven stellen nadat er in die decision een uitkomst was bereikt. Dynamische Vragen 2.0 voorziet al in de juiste werking van dit type decisions. De uitzondering die eerder mogelijk is gemaakt in het oude mechanisme is nu dus niet meer nodig.	bedrijfsregel.xsd	Breaking change
Verificatie die FEEL expressies als input van beslissingen voorkomt	In DMN is het mogelijk om een FEEL expressie als input van een decision te definiëren. Dynamische Vragen 2.0 werkt niet in die situatie. Er wordt een verificatie ingebouwd die het niet mogelijk maakt om STTRs aan te leveren met een FEEL expressie als input.	Verificatie	Breaking change
Verificatie die gebruik van het element '<dynamischeVragenUit />' niet meer toestaat	Omdat '<dynamischeVragenUit />' in versie 2 van de STTR wel een functie heeft, is er voor gekozen om in versie 3 expliciet een verificatie toe te voegen. Meestal worden zaken die binnen <extensionElements> worden opgenomen en die geen functie hebben genegeerd in plaats van tegengehouden. In dit geval wekt de aanwezigheid van '<dynamischeVragenUit />' in een STTR van versie 3 mogelijk verwachtingen. Dat willen we voorkomen.	Verificatie	Breaking change

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:bedr="http://toepasbare-regels.omgevingswet.overheid.nl/v1.0/Bedrijfsregel"
  elementFormDefault="qualified"
  targetNamespace="http://toepasbare-regels.omgevingswet.overheid.nl/v1.0/Bedrijfsregel"
  version="1.0.0">
  ...
  <xs:simpleType name="sttrVersie">
    <xs:restriction base="xs:string">
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="sttr">
    <xs:complexType>
      <xs:attribute name="versie" type="bedr:sttrVersie" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="functioneleStructuurRef">
    <xs:complexType>
      <xs:attribute name="href" type="xs:anyURI" use="required"/>
    </xs:complexType>
  </xs:element>

  <del><xs:element name="dynamischeVragenUit"/></del>

</xs:schema>

```

*Figuur 4 - verwijderen van element dynamischeVragenUit*

Nadere uitleg: FEEL (als input in beslissing): wordt nu niet gebruikt. Maar wordt voor de zekerheid dus wel onmogelijk gemaakt.

Een leverancier vraagt of FEEL wel in de inputs kan worden blijven gebruikt: dat wordt bevestigd, het enige wat niet meer kan is dat de definitie van een (nieuwe) kolom geen FEEL-expressie kan bevatten.

Vraag vanuit provincies: wordt de implementatie-strategie over Dynamische vragen 2.0 bij het laatste agendapunt behandeld? Antwoord is bevestigend.

Verder geen opmerkingen.

## 5. Verificatie op boolean waarden

In de STTR/IMTR documentatie zeggen we dat een boolean true of false moet zijn, maar in de praktijk controleren we dit niet. Daardoor kunnen 0 en 1 (waarden die binnen DMN ook zijn toegestaan als boolean waarden) ook worden aangeleverd. Dit resulteert in een onjuiste werking in het loket (1 wordt ten onrechte als false gezien).

Daarom wordt een verificatie toegevoegd die 0 en 1 als waarden niet meer toestaat.

Wijziging	Omschrijving	Waar	Impact
Verificatie op boolean waarden	<p>Alle plaatsen in de XSD waar gebruik wordt gemaakt van het standaard datatype xs:boolean moet deze vervangen worden door het nieuwe dataTypes:boolean (het nieuwe datatype dat onder punt 4 is toegevoegd)</p> <p>Om welke elementen gaat het?</p> <ul style="list-style-type: none"> <li>- Content.xsd <ul style="list-style-type: none"> <li>o Element 'belangrijk' in het element uitvoeringsregelToelichting</li> </ul> </li> <li>- Uitvoeringsregel.xsd <ul style="list-style-type: none"> <li>o Element 'publiceerbaar' in het element uitvoeringsregel</li> <li>o Element 'potentieelVertrouwelijk' in het element uitvoeringsregel</li> <li>o Element 'waarde' in het element implicietAntwoord</li> <li>o Element 'verborgenStuurvraag' in het element registerbevraging</li> <li>o Element 'verborgenStuurvraag' in het element registerbevragingAPIProfiel</li> <li>o Element 'optieGeenVanBovenstaande' in het element opties</li> </ul> </li> </ul>	content.xsd uitvoeringsregel.xsd	Breaking change

```

<xs:element name="uitvoeringsregelToelichting">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="toelichting" type="dataTypes:beperkteNietLegeString" minOccurs="0"/>
      <xs:element name="langeToelichting" type="dataTypes:nietLegeString" minOccurs="0"/>
      <xs:element name="belangrijk" type="dataTypes:boolean" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

*Figuur 5 - Voorbeeld van gebruik nieuwe datatype*

Nadere uitleg: Boolean: true en false, 0 en 1 mag niet. Dit komt overigens in de praktijk nauwelijks voor. Eigen datatype voor boolean; i.p.v. dmn-type voor boolean. Zie ook wijziging nr. 11.

Verder geen opmerkingen.

## 6. Verificatie gebroken getallen komt te vervallen

Er is een verschil tussen hoe in het DSO wordt omgegaan met gebroken getallen (aangegeven met een komma) en de manier waarop de IEEE 754 standaard vindt dat een gebroken getal eruit moet zien. Als in een STTR in een vaste waarde een gebroken getal "3,12" wordt gebruikt gaat de volgende verificatie af:

```
[STTR_00530] De waarde van een gegeven met type number moet een geldig getal zijn volgens de IEEE 754 standaard. Als het getal decimalen bevat moet een decimale punt gebruikt worden en geen komma. De waarde in dit veld is nu: 3,12.
```

Hierdoor zou een regelmaker het getal veranderen naar '3.12'. Maar als wij de string omzetten naar een getal gebruiken we de Nederlandse manier, waardoor de punt als duizendtal separator wordt gezien. Dus wordt '3.12' geïnterpreteerd als '312'.

In de productieomgeving bestaan er momenteel geen STTR-bestanden waarin gebruik wordt gemaakt van vaste waarden en vaste waarde voorbehouden van het type 'number'.

In de wijziging beschreven in punt 4 wordt de juiste notatie (met komma) voortaan afgedwongen. Daarom kan deze verificatie worden verwijderd.

Wijziging	Omschrijving	Waar	Impact
Verificatie gebroken getal verwijderen	Zie hierboven	Verificatie	Non-breaking change

Opmerking vanuit waterschappen: het gedrag is anders. Als je 0,12 wilt invullen, en je typt een '0' in, dan gaat de regel-executie al lopen en kunnen vervolgvragen al vervallen. Dit issue is al bekend als IPLO-melding. Het heeft geen directe relatie met deze verificatie.

Verder geen opmerkingen.

### 7. Lijst gebruiken in plaats van komma gescheiden string bij Vraag met OptieType "MeerdereAntwoorden"

Als in een STTR een uitvoeringsregel is waarbij de gebruiker meerdere antwoorden kan geven dan worden die antwoorden, als de gebruiker inderdaad meerdere antwoorden geeft, verwerkt als een string waarin de antwoorden door komma's gescheiden staan. Dit is een foutgevoelige manier van werken.

*Om te zien of een gebruiker een antwoord heeft geselecteerd, wordt door de regelanalist in de STTR een controle toegevoegd (door middel van de contains() functie van FEEL) of de komma-gescheiden string dat specifieke antwoord bevat. Stel je wilt als regelanalist controleren of de gebruiker het antwoord 'Suiker' heeft geselecteerd. Dan voeg je de controle 'contains("Suiker")' toe. Maar: ook als de gebruiker het antwoord 'Suikerwaren' heeft geselecteerd, zal 'contains("Suiker")' met een komma-gescheiden string een positief resultaat opleveren.*

In STTR3 wordt voortaan gebruik gemaakt van een echte lijst van strings in plaats van één string waar de waarden door komma's gescheiden in staan. Dit is geen aanpassing aan de STTR standaard. Maar regelanalisten/leveranciers zullen vanaf versie 3.0 wel een aanpassing moeten doen in de STTR-bestanden die zij genereren.

Wanneer regelmakers overgaan naar STTR3 moeten ze deze FEEL expressie:

```
if ?=null then false else contains(?, "optie1")
```

vervangen door:

```
if =null then false else list contains(?, "optie1")</pre
```

Nadere uitleg: Deze wijziging heeft alleen impact bij het aanmaken van STTR-bestanden (leveranciers). Geen impact op de STTR zelf.

Verder geen opmerkingen.

## 8. Bij 'registerbevragingAPIProfiel' responses met 'MeerdereAntwoorden' in een lijst plaatsen

Nadere uitleg: Voorbeeld: BAG kan voor een perceel meerdere gebruiksoppervlaktes van retourneren, dit wordt dan ook een 'list' van meerdere waarden in plaats van één string waarin alle waarden staan, gescheiden door een komma. Doordat er gebruik gemaakt wordt van een list, kun je in DMN gebruikmaken van de functies die een list biedt<sup>1</sup>. Met een 'list count' zou je bijvoorbeeld ook kunnen uitvragen hoeveel waardes er beschikbaar zijn.

Verder geen opmerkingen.

## 9. Element 'register' verwijderen uit registerbevraging

De uitvoeringsregel van het type 'registerbevraging' was ooit bedoeld om ook andere registers dan GGS te bevragen. Uiteindelijk is er voor gekozen om andere registers te ontsluiten via een nieuw uitvoeringsregeltype, genaamd 'registerbevragingAPIProfiel'.

In 'registerbevraging' zit nog steeds het element 'register', waarmee aangegeven kan worden welk register bevroegd moet worden. Hier is maar een optie voor mogelijk: 'GGS'. Daarom kan het element "register" verwijderd worden. De enige plek waar de registerbevraging wordt gebruikt, is in de Algemene Set.

Wijziging	Omschrijving	Waar	Impact
Element register verwijderen uit registerbevraging	Zie hierboven	Uitvoeringsregel.xsd	Breaking change

```
<xs:complexType name="registerbevraging">
  <xs:sequence>
    <xs:element name="verborgenStuurvraag" type="dataTypes:boolean" default="false" minOccurs="0"/>
    <del><xs:element ref="uitv:register"/></del>
    <xs:element name="registerGegeven" type="xs:string"/>
    <xs:element name="registerKoppelveld" type="xs:string" minOccurs="0"/>
    <xs:element ref="uitv:vraag"/>
  </xs:sequence>
</xs:complexType>
```

*Figuur 6 - verwijderen van het element register uit de registerbevraging*

<sup>1</sup> Dit is niet besproken in het overleg, maar dit geldt natuurlijk ook voor de aanpassing van punt 7.



Nadere uitleg: dit element is overbodig, kan dus worden weggehaald. Het heeft geen impact op leveranciers, omdat zij het register (GGS) niet kunnen bevragen.

Verder geen opmerkingen.

## Reeds behandelde wijzigingen

De lijst in deze paragraaf bevat wijzigingen die nog al door het CABS zijn behandeld en goedgekeurd.

### **10. (CABS-187) Veldlengtes van velden die aan STAM worden doorgegeven beperken**

Enkele gegevens die hun oorsprong vinden in STTR worden in een verzoek doorgegeven via het STAM-bericht. Op dit moment zit op die velden in het STAM-bericht wel een maximale veldlengte, maar in de STTR nog niet. In de STTR gaat nu dezelfde beperking qua lengte van deze velden toegepast worden als in het STAM-bericht.

<b>Wijziging</b>	<b>Omschrijving</b>	<b>Waar</b>	<b>Impact</b>
Naam van een regelgroep beperken tot 80 karakters	Het element 'naam' krijgt een maximale lengte van 80	Interactieregel.xsd	Breaking change
BijlageType beperken tot 80 karakters	Het element 'bijlageType' in het element 'bijlage' krijgt een maximale lengte van 80	Uitvoeringsregel.xsd	Breaking change
vraagReferentie beperken tot 512 karakters	Het element 'vraagReferentie' krijgt een maximale lengte van 512	Uitvoeringsregel.xsd	Breaking change
vraagTekst beperken tot 512 karakters	Het element 'vraagTekst' krijgt een maximale lengte van 512	Uitvoeringsregel.xsd	Breaking change

```

<xs:element name="naam">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <!--
        De waarden van dit veld komen uiteindelijk uit in de STAM berichten, daar mag
        dit veld maximaal 80 lang zijn.
      -->
      <xs:maxLength value="80"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

*Figuur 7 - beperken van de lengte van het element naam*

```

<xs:element name="bijlage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="bijlageType">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
            <!--
              De waarden van dit veld komen uiteindelijk uit in de STAM berichten, daar mag
              dit veld maximaal 80 lang zijn.
            -->
            <xs:maxLength value="80"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="documenttypeGeneriek" type="uitv:documenttypeGeneriek" default="Overig" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

*Figuur 8 - beperken van de lengte van het element bijlageType*

```

<xs:element name="vraagReferentie">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <!--
        De waarden van dit veld komen uiteindelijk uit in de STAM berichten, daar mag
        dit veld maximaal 512 lang zijn.
      -->
      <xs:maxLength value="512"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

*Figuur 9 - beperken van de lengte van het element vraagReferentie*

```

<xs:element name="vraagTekst">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <!--
      De waarden van dit veld komen uiteindelijk uit in de STAM berichten, daar mag
      dit veld maximaal 512 lang zijn.
      -->
      <xs:maxLength value="512"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

Figuur 10 - beperken van de lengte van het element vraagTekst

Nadere uitleg: Dit is geen aparte verificatie, maar wordt rechtstreeks via de XSD afgedwongen.

Verder geen opmerkingen.

### 11. (CABS-194) Verificatie op of de waarde van een vaste waarde overeenkomt met het gegevenstype

Bij een vaste waarde kan een regelanalist opgeven van welk gegevenstype deze vaste waarde is (boolean, string, number etc.). Vervolgens geeft de regelanalist de waarde op. Er vindt geen controle plaats of de opgegeven waarde ook daadwerkelijk van het juiste gegevenstype is. Met andere woorden: een vaste waarde met gegevenstype boolean kan als waarde de string 'Ja' bevatten. Dit kan resulteren in onverwacht gedrag.

**Let op:** Dit is een wijziging naar aanleiding van een binnengekomen issue (CABS-194). In het issue werd gesproken over een probleem met een vaste waarde van het type boolean. In de gebouwde oplossing worden vaste waarden van alle dataTypes gecontroleerd op correctheid.

Wijziging	Omschrijving	Waar	Impact
Specifieke datatypes definiëren	Een drietal nieuwe datatypes is gedefinieerd: boolean, number en date. Daarin wordt vastgelegd waar de inhoud van die waarden aan moeten voldoen.	dataTypes.xsd	Non-breaking change
Specifieke datatypes gebruiken	Vervolgens moeten deze definities worden gebruikt in de vaste waarde. Het element gegevensType wordt vervangen door een keuzelijst van elementen. In die elementen plaats je vervolgens de vaste waarde die je wilt gebruiken. Daarom is ook het element 'waarde' niet meer nodig.  Voorbeelden: <booleanWaarde>true</booleanWaarde> <dateWaarde>27-01-2025</dateWaarde>	uitvoeringsregel.xsd	Breaking change

Ondanks dat we wel het gegevensType 'list' kennen, komt hier in de nieuwe keuzelijst geen optie voor terug.  
Voorheen kon een VasteWaarde (al dan niet in een VasteWaardeVoorbehoud) aangeleverd worden met GegevensType 'List'. Omdat een lijst niet meer is dan een string, waarin de waarden door komma's gescheiden zijn opgenomen, behandelen we dit gegevensType precies hetzelfde als een normale string. Dit is verwarrend, daarom willen we het niet langer toestaan om een VasteWaarde aan te leveren met GegevensType 'list'.

```
<xs:element name="vasteWaarde">
  <xs:complexType>
    <xs:sequence>
      <!-- Het veld waardeVraag dient enkel als documentatie voor de regelmaker. Het wordt verder niet
      <xs:element name="waardeVraag" type="dataTypes:nietLegeString" minOccurs="0"/>
      <del>xs:element name="waarde" type="dataTypes:nietLegeString"/>
      <del>xs:element name="gegevensType" type="uitv:gegevensType"/>
      <xs:choice>
        <xs:element name="booleanWaarde" type="dataTypes:boolean"/>
        <xs:element name="numberWaarde" type="dataTypes:number"/>
        <xs:element name="stringWaarde" type="dataTypes:nietLegeString"/>
        <xs:element name="dateWaarde" type="dataTypes:date"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

*Figuur 11 - Gebruik van nieuwe dataTypes in vasteWaarde*

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
In deze XSD zijn de zelfgedefinieerde datatypes gedefinieerd.
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  targetNamespace="http://toepasbare-regels.omgevingswet.overheid.nl/v1.0/DataTypes"
  xmlns="http://toepasbare-regels.omgevingswet.overheid.nl/v1.0/DataTypes"
  version="1.0.0">

  <xs:simpleType name="boolean">
    <xs:restriction base="xs:string">
      <xs:enumeration value="true"/>
      <xs:enumeration value="false"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="number">
    <xs:restriction base="xs:string">
      <!-- gebruik komma voor gebroken getallen, duizendtallen mogen niet gescheiden worden -->
      <xs:pattern value="[0-9]+(,[0-9]+)?"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="date">
    <xs:restriction base="xs:string">
      <!-- formaat: dd-mm-yyyy -->
      <xs:pattern value="(0?[1-9]|[12][0-9]|3[01])-(0?[1-9]|1[012])-[0-9]{4}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="nietLegeString">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="beperkteNietLegeString">
    <xs:restriction base="nietLegeString">
      <xs:maxLength value="2048"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>

```

*Figuur 12 - toevoegen datatype-definities*

Nadere uitleg: is in de praktijk één keer voorgekomen.

Impact bij leveranciers: aanleveren van vasteWaarde: rekening houden met nieuwe manier van datatypes (zie ook wijziging nr. 5).

Verder geen opmerkingen.

## 12. (CABS-187) Markdown toestaan in tekstVoorbehoud van vaste waarde onder voorbehoud

Het element 'tekstVoorbehoud' is een attribuut (string) van vasteWaardeVoorbehoud. De tekst wordt, wanneer relevant, getoond op de conclusie-pagina van de check. In de gebruikerstoepassing wordt dit veld al als Markdown geïnterpreteerd. Alleen wordt in de STTR dit veld niet op Markdown gevalideerd zoals we dit bijvoorbeeld wel doen bij toelichtingen op vragen.

Wijziging	Omschrijving	Waar	Impact
Verificatie tekstVoorbehoud toevoegen	De inhoud van het element tekstVoorbehoud wordt gevalideerd of dit volgens de STTR standaard valide Markdown is.	Verificatie	Breaking change
tekstVoorbehoud krijgt maximale lengte	tekstVoorbehoud gaat ook gevalideerd worden op dat de tekst niet langer is dan 2048 tekens.	Uitvoeringsregel.xsd	Breaking change

```
<xs:element name="vasteWaardeVoorbehoud">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="uitv:vasteWaarde"/>
      <xs:element name="tekstVoorbehoud" type="dataTypes:beperkteNietLegeString"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figuur 13 - aanpassen van het type van tekstVoorbehoud, om lengte te controleren

Nadere uitleg: Nog niet voorgekomen.

Verder geen opmerkingen.

## Ter Info

Daarnaast is er nog een aantal zaken die niet direct impact hebben op de STTR standaard, maar die wel van invloed zijn op de manier waarop gegevens door bevoegd gezagen moeten worden aangeleverd en de manier waarop zij informatie terugkrijgen.

### **13. (CABS-192) Introductie van warnings bij het aanleveren van STTR bestanden**

In de STTR kennen we nu alleen ERRORS die teruggegeven kunnen worden in het response-bericht bij het aanleveren van een STTR door een bevoegd gezag. Daar worden WARNINGS aan toegevoegd. Via deze warnings kunnen verificaties worden teruggekoppeld die in een toekomstige versie van STTR mogelijk als ERROR zullen worden geïntroduceerd. Op deze manier zijn bevoegd gezagen en leveranciers hier sneller van op de hoogte, en kunnen zij daar op inspelen.

*Het toevoegen van deze functionaliteit is niet (goed) mogelijk binnen de huidige technische kaders. Op dit moment wordt gebruik gemaakt van het SOAP protocol. Het DSO is voornemens om communicatie via REST mogelijk te maken. Wanneer dit wordt geïmplementeerd, is het een stuk gemakkelijker om deze functionaliteit toe te voegen, en zal deze worden meegenomen.*

Nadere uitleg: Niet STTR3 gerelateerd.

Verder geen opmerkingen.

### **14. Verbetering aansluitpunten**

De manier waarop Aansluitpunten werken, gaat worden aangepast. Binnen de huidige manier van werken liepen in de praktijk leveranciers tegen problemen aan die nu geadresseerd gaan worden.

Wat gaat er veranderen?

- De constructie waarbij twee STTR bestanden samen één geheel vormen (het aansluitpunt en de aansluiting, via de INVOKE-constructie) komt te vervallen
- Wanneer een bevoegd gezag een aansluitpunt wil creëren, levert het – net zoals bij een 'gewone' STTR - één STTR bestand aan
- Via de opdracht waarmee het STTR bestand wordt aangeleverd krijgt het bevoegd gezag de mogelijkheid om aan te geven dat het om een aansluitpunt gaat
- Wanneer een lokaal bevoegd gezag wil aansluiten op het aansluitpunt, levert ook dit bevoegd gezag één STTR bestand aan
- Dit lokale STTR bestand vervangt in zijn geheel het bestand van het hogere bevoegd gezag

Deze aanpassing heeft geen impact op de STTR standaard; er verandert in technische zin niets aan de STTRs die worden aangeleverd. Maar er verandert wel het een en ander aan de manier waarop bestanden rondom aansluitpunten moeten worden aangeleverd. Deze wijzigingen zullen landen in het document 'Koppelvlak aanleveren toepasbare regels'.

Eén leverancier ziet dit als een ingrijpende verandering. Hij vraagt wat is de status is, omdat de voorgestelde oplossing nog niet breed is gedeeld. Hij heeft Miel dan ook aanvullende vragen: bijvoorbeeld hoe kan je zien of het een aansluitpunt of aansluiting is, wanneer je een STTR-bestand importeert? Antwoord: deze wijziging heeft vooral impact in de aanlevering, en zal worden gecombineerd met versionering 2.0 (wijziging nr. 15). Over deze wijziging zal nog worden overlegd met leveranciers en BLMers. Deze wijziging zal ook niet voor juli 2025 worden geïmplementeerd.

## 15. Versionering 2.0

Versionering 2.0 adresseert enkele problemen waar bevoegd gezagen en leveranciers tegenaan lopen bij het toepasbaar maken van activiteiten en aanleveren van STTR-bestanden. Denk aan zaken als:

- Het is nu niet mogelijk om een bestaande, in werking zijnde STTR bestand een einddatum te geven
- Het is nu niet mogelijk om meer dan één STTR bestand in de toekomst klaar te zetten

Alle wijzigingen die in het kader van versionering 2.0 worden gedaan, hebben geen impact op de STTR standaard. Maar wel op de manier waarop bestanden (STTR bestanden en toepasbaar maken bestanden) moeten worden aangeleverd. Deze wijzigingen zullen landen in het document 'Koppelvlak aanleveren toepasbare regels'.

Nadere uitleg: Ook hier wordt nog op teruggekomen. Hierover heeft al een sessie plaatsgevonden (o.a. mogelijk maken om meerdere bestanden te kunnen aanleveren met een toekomstige geldig vanaf datum). Deze wijziging moet nog in detail worden gespecificeerd (samen met 13, 14).

## 16. Verificaties voor minimaal sttr-bestand nakijken

Op dit moment bestaan er verschillende verificaties die controleren of een STTR bestand voldoet aan de minimale eisen om een STTR bestand te kunnen zijn. Is nog in onderzoek, lijken we niets mee te willen gaan doen. Zo wordt er bijvoorbeeld een controle gedaan die bekijkt of een STTR bestand minimaal één uitvoeringsregel bevat.

Er loopt een onderzoek om te bekijken of deze verificaties nog toegevoegde waarde hebben. Mocht dat niet het geval zijn, dan kunnen deze verwijderd worden. Op dit moment lijkt het erop dat er geen aanpassingen gaan plaatsvinden. Maar het onderzoek is nog niet volledig afgerond. Mogelijk heeft het onderzoek dus nog impact op enkele verificaties.

Nadere uitleg: Nog onzeker of het in STTR 3 komt of niet. Bijv. een bestaande verificatie is dat een STTR-bestand minstens één uitvoeringsregel moet bevatten. Deze zou nog kunnen worden aangepast of worden verwijderd. Het zal nooit een verstrenging betekenen, dus voor leveranciers en bestuursorgaan wordt geen impact verwacht.

Op dit moment is het noodzakelijk dat een STTR-bestand minimaal één uitvoeringsregel bevat. Er zijn echter situaties denkbaar waarbij dit niet nodig is, bijvoorbeeld bij aansluitpunten. Daarom worden de volgende verificaties verwijderd:

- Een sttr-bestand moet minimaal 1 uitvoeringsregel hebben (verificatie bij aanleveren): "Het is niet gelukt om de gebruikte STTR versie af te leiden uit de namespace van Uitvoeringsregel".
- STTR\_00060: Een sttr-bestand moet een extensionsElement attribuut met een uitvoeringsregels attribuut bevatten. (Uitleg: Als de afwezigheid van uitvoeringsregels is toegestaan voegt deze verificatie niet veel toe.)
- STTR\_00090: De namespace voor uitvoeringsregels moet aanwezig zijn. (Uitleg: Als de bovenstaande verificaties verwijderd zijn is een sttr-bestand dat geen gebruik maakt van deze xsd mogelijk. Daarmee lijkt het gek om wel te eisen dat de xsd geïmporteerd wordt.)

We verwachten dat het verwijderen van deze verificaties geen impact heeft, omdat dit niet leidt tot beperkingen, eerder tot meer mogelijkheden.

## D. N.a.v. het bespreken van de lijst: zijn er aanvullingen/opmerkingen?

Vertegenwoordiger van de VNG is blij dat de wijzigingen op deze manier worden gedeeld en besproken. Hij vraagt zich wel af of alles duidelijk is voor leveranciers.



Twee leveranciers reageren:

1. Het is allemaal wel duidelijk en in het algemeen prima. Vooral voorstander van opsplitsing van de documentatie, wat het makkelijker maakt om dingen op te zoeken. Wel zorg over wijzigingen die de STTR niet raken, die zijn dan misschien makkelijker te wijzigen, maar er kan bij leveranciers wel veel impact zijn. Hij verwacht geen grote problemen met het implementeren van STTR 3 en het niet meer kunnen aanleveren STTR 1 en 2 medio 2026.  
Reactie DSO-LV: ook over ontwikkelingen die niet aan STTR zijn gekoppeld blijven we in gesprek, o.a. d.m.v. kennissessies zoals deze. Naast de afspraken rondom wijzigingen van standaarden blijven de afspraken t.a.v. de API-strategie ook gelden.
2. De wijzigingen zijn duidelijk. Maar er is geen garantie dat er zich bij de implementatie geen problemen gaan voordoen.  
Reactie DSO-LV: In overleg treden als er knelpunten ontstaan. M.b.t. implementatie van STTR 3 verwacht DSO-LV weinig impact bij leveranciers; het betreft voornamelijk het toevoegen van verificaties die ongewenste situaties t.a.v. dynamische vragen 2.0 onmogelijk maken.

Verder geen opmerkingen.

### **E. Dynamische vragen 2.0: nieuwe versie van API uitvoeren services**

Eén leverancier reageert. Hij stelt dat de plannen zijn besproken en dat de afspraak was dat je met dynamische vragen 1.0 moet kunnen blijven werken. Nu wordt gezegd dat er vanaf 1 juli 2026 alleen kan worden aangeleverd i.r.t. dynamische vragen 2.0. Zo is het destijds niet gecommuniceerd. Er waren destijds veel vragen en onduidelijkheden  
STTRs versie 1 blijven draaien.

Reactie DSO-LV: STTRs versie 1 die in de RTR zijn opgeslagen blijven draaien, ook na 1 juli 2026. Na die datum kunnen echter geen STTRs met versie 1 worden aangeleverd. DSO-LV heeft STTR-bestanden van alle leveranciers getest. Ook is er onlangs met de waterschappen getest, die geen knelpunten zien en voorstander zijn voor de overstap naar dynamische vragen 2.0. Er zal nog één testsessie worden ingepland. Bij één leverancier is bekend dat dynamische vragen 2.0 problemen kan opleveren. Zoals de zaken er nu voorstaan zullen alle leveranciers uiterlijk 1 juli 2026 moeten zijn overgestapt naar STTR versie 3 om daarna ook nog te kunnen blijven aanleveren.

### **F. Nieuwe verificatie maximum grootte STTR-bestand (niet besproken tijdens overleg)**

Vanuit security wordt aanbevolen om aan aan te leveren bestanden een beperking op te leggen t.a.v. bestandsgrootte. Bij het aanleveren zijn de zip-opdracht-bestanden al gelimiteerd op 20MB. Bij deze wijziging (verificatie) wordt ook het STTR-bestand gelimiteerd op maximaal 20MB. Het grootste STTR-bestand op productie is momenteel 13MB. DSO-LV verwacht daarom geen impact bij leveranciers. Deze maximum bestandsgrootte zal gelden voor alle STTR-bestanden, ongeacht de STTR-versie.