

The logo for SIVI, featuring the word "SIVI" in a bold, black, sans-serif font. Above the letter "I" is a small graphic element consisting of a 2x2 grid of squares in shades of blue and grey.

SIVI

An orange circular badge with a soccer ball pattern. A white ribbon-like banner is wrapped around the middle of the circle, containing the text "NUMMER 2" and "JUNI 2021".

NUMMER 2
JUNI 2021

SIVI AFS Magazine

Voorwoord
customAttribute

SIVI AFS als standaard
koppelvlak voor ANVA

Handmatig berichtschema's
opstellen behoort tot verleden

AFS vanuit gezichtspunt ontwikkelaar

Webservices volgens SIVI
API-raamwerk

Proprietary APIs

5 vragen aan...
Paul Janssen

Vooruitblik

Feedbackcorner

Nieuwtjes

SIVI AFS

On the go

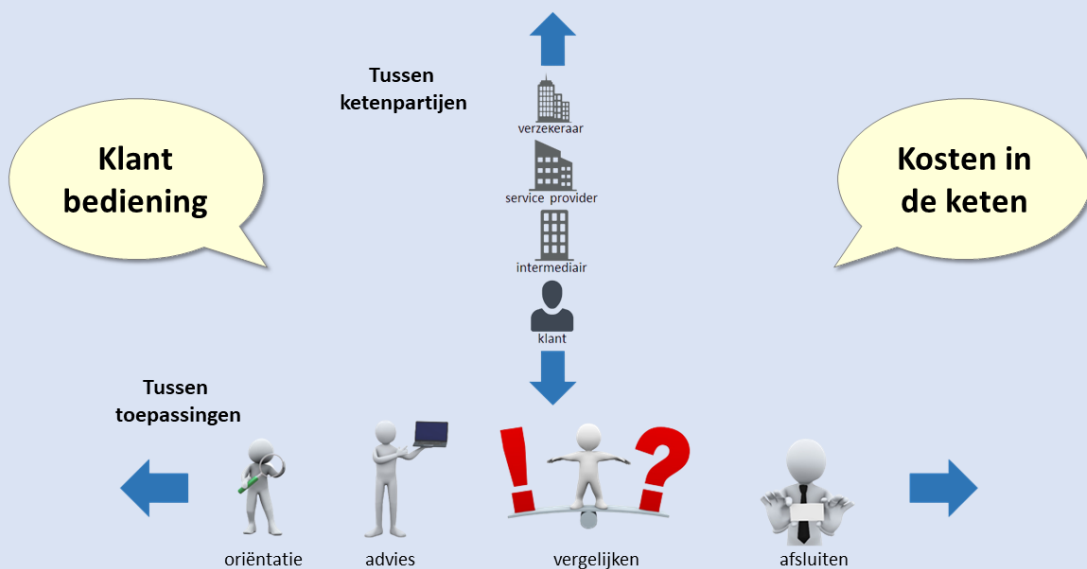
Herman Lenferink

De introductie van de SIVI All Finance Standard (SIVI AFS) is al weer een half jaar geleden. Inmiddels kennen we een vast maandelijks releaseschema met daarin, naast de uitbreidingen van AFD 1.0, de uitbreidingen aangevraagd door SIVI AFS-gebruikers. Een mooi teken dat SIVI AFS echt in gebruik is. Daarnaast hebben we recent het opschonen van de naamgeving van attributen afgerond. Dit heeft een veel eenduidiger AFD opgeleverd: een belangrijke stap in het verkleinen van de leercurve rond SIVI AFS. In het verlengde van feedback hebben we in dit magazine maar liefst drie externe bijdragen. Een mooie mijlpaal.

Met het SIVI AFS bedienen wij een agenda rond ketenoptimalisatie. In onderstaande figuur zie je de twee assen waarlangs wij dat doen. Verticaal (door de keten heen) gaat het vaak om concrete transacties rond bijv. offerte, polis, schade of boeking. Het product staat hier centraal. Horizontaal gaat het door de toepassingen

heen op bijvoorbeeld het bureau van een adviseur. Hier staat het klantdossier in combinatie met een processtap centraal, zoals het opstellen van een advies of het aanmaken van een vergelijking. Naast de focus op functies (zoals premieberekening of aanvraag) en dus de uitwisseling van gegevens, is er ook steeds meer aandacht

voor de vastlegging van gegevens in een klantdossier. In de komende maanden zullen een aantal uitbreidingen en verdiepingen volgen rond de AFD-structuur partyStructure, zodat men rond een klantdossier goed met zowel functies als de vastlegging van gegevens uit de voeten kan.





Elke ketenpartij moet uiteindelijk een eigen visie ontwikkelen rond de invulling of ondersteuning van ketenoptimalisatie. Dit levert o.a. een analyse op van welke functies men binnen een eigen API-omgeving wil invullen. In dit magazine geven wij daar een denkraam voor bij het gebruik van SIVI AFS. Bij de verdere uitwerking kan het vervolgens gebeuren dat zaken ontbreken binnen het API-raamwerk of het AFD.

Belangrijk hierbij op te merken is dat de doorontwikkeling van SIVI AFS vraaggedreven is. Iedereen (ketenpartijen, derden en leveranciers) kan

een uitbreiding aanvragen mits deze een relatie (in de meest ruime zin van het woord) heeft met financiële dienstverlening. Toch kan het voorkomen dat de verzochte uitbreiding niet past binnen het AFD of het API-raamwerk. Maar een verzekeraar bijvoorbeeld laat een uniek acceptatiecriterium niet vallen omdat het niet binnen de standaard past. Het niet ondersteunen van dit type verzoeken leidt uiteindelijk tot schaduwstandaarden, of zelfs het niet gebruiken van een standaard. Om deze reden is er binnen SIVI AFS nadrukkelijk ruimte voor het vastleggen van uitzonderingen op de standaard. In dit magazine

leggen we uit hoe: zowel voor attributen als functies. Binnenkort voegen we hier ook codelijsten aan toe. De centrale boodschap is dat als je onverhoopt niet de SIVI AFS-standaard kan volgen, je toch nog steeds binnen de context van die standaard kan blijven werken.

Rest mij om namens SIVI iedereen nadrukkelijk te bedanken voor zijn/haar bijdrage bij de doorontwikkeling van SIVI AFS. Veel leesplezier!

Herman

Zelf attributen definiëren in SIVI AFS customAttribute

Robin Oostrum

Stel: je bent bezig met het uitwerken van een webservice voor de acceptatie en premieberekening van een autoverzekering. In de [datacatalogus](#) van SIVI AFS heb je de `underwriting`-entiteit gevonden met entityType `underwritingQuestions`. Daarbinnen zie je twee attributen die je nodig hebt: `claimInPastThreeYears` (heeft de aanvrager van de polis in de voorbije drie jaar een schade gemeld?) en `drivingBan` (is het rijbewijs van de aanvrager (on)voorwaardelijk ontzegd?).

Maar de premie hangt van meer details af: zo hebben jullie vastgesteld dat het scheelt of de rij-ontzegging langer dan zes jaar geleden plaatsvond. In feite ben je op zoek naar het attribuut `drivingBanPastSixYears`. Helaas: die vind je niet terug in de datacatalogus. Maar je zoekt even verder, en dan vind je op de [landingspagina](#) van het SIVI AFS wel een aanvraagformulier voor uitbreidingen en wijzigingen. Zo werkt het voor AFD 1.0 al ruim twintig jaar, en zo werkt het ook voor AFD 2.0: mede dankzij deze aanvragen vanuit de branche blijft de datacatalogus actueel en alomvattend. Uitgangspunt van SIVI AFS is echter dat AFD 2.0 representatief moet zijn voor de hele keten. In dit geval betekent dat: attributen dienen algemeen toepasbaar te zijn, en niet toegespitst op één aanbieder en/of één product. Attributen als `drivingBanPastSixYears` horen dan niet in de brede AFD 2.0-standaard thuis.

Om de wildgroei van attributen binnen het AFD in te dammen, is het mogelijk binnen SIVI AFS zelf attributen te definiëren: de `customAttribute`-constructie. Deze constructie stelt een gebruiker in staat om op een gestandaardiseerde wijze gegevens te communiceren die niet zijn opgenomen binnen AFD 2.0. In de praktijk zijn dit in bijna alle gevallen acceptatievragen. Vaak zijn deze gegevens bedrijfs- of zelfs productspecifiek, en herziet een organisatie ze meermaals tijdens de levenscyclus van het betreffende product. Omdat deze gegevens zo situatiespecifiek zijn, zijn ze niet geschikt voor standaardisatie ten behoeve van de gehele sector. Daarom nemen we deze gegevens niet op in AFD 2.0.

Hoe ziet dit eruit? Binnen AFD 2.0 is voor dit soort specificaties de `afsElement`-entiteit opgenomen, met daarbinnen het `customAttribute` als entityType. `afsElement` kun je gebruiken binnen andere entiteiten, naast bestaande attributen. Bepalend voor de definitie van een `customAttribute` zijn twee verplichte attributen: de `customAttributeName` en het `valueType`. Dat laatste attribuut bepaalt welk datatype het zelf gedefinieerde attribuut krijgt. Met het (optionele) `customAttributeDescription`-attribuut is het daarnaast nog mogelijk om een omschrijving toe te voegen.

Het voorbeeld van `drivingBanPastSixYears` zou je dan als volgt definiëren binnen een `underwriting`-entiteit:

```
{
  "underwriting": [ {
    "entityType": "underwritingQuestions",
    "afsElement": [ {
      "entityType": "customAttribute",
      "customAttributeName": "drivingBanPastSixYears",
      "customAttributeDescription": "Is het rijbewijs (on)-
        voorwaardelijk ontzegd in de voorbij zes jaar?",
      "valueType": "boolean"
    } ]
  } ]
}
```

Op deze manier kun je maatschappij- en/of productspecifieke elementen eenvoudig binnen de context van SIVI AFS gebruiken, zonder de AFD 2.0-datacatalogus te overladen met overspecifieke attributen.

Vragen of meer informatie nodig over het gebruik van `customAttribute`? [Mail ons](#).

SIVI AFS als standaard koppelvlak voor ANVA

Thierry van Ekeren

Thierry van Ekeren



Thierry van Ekeren werkt al meer dan 10 jaar in software development vanuit verschillende functies. Het grootste deel daarvan heeft hij meegewerkt aan applicaties die de financiële dienstverlening ondersteunen. Op dit moment werkt hij bij ANVA als Lead Architect aan ANVA Hub, een platform dat vraag en aanbod met elkaar verbindt. In deze jaren is hij veelvuldig in aanraking gekomen met de standaardisatie vanuit SIVI. Dat SIVI bezig is met innovaties binnen de standaardisatie sluit naadloos aan op de ambities die hij binnen de ANVA-architectuur wilt doorvoeren.

ANVA hecht veel waarde aan connectiviteit en is daarom continu bezig om haar applicatielandschap open te stellen aan klanten en partners, om zo optimale oplossingen te kunnen realiseren. De behoefte dit te doen via marktstandaarden is een groeiende factor bij ANVA en de branche, vanwege het wegnemen van onduidelijkheid en het kunnen ontwikkelen van herbruikbare koppelvlakken. Om ANVA Hub completer te maken qua functionaliteit is er een sterke behoefte voor standaard interfaces voor het opvragen van o.a. klant- en contractgegevens. ANVA kiest voor de SIVI All Finance Standaard (SIVI AFS) om aan deze behoefte te voldoen.



ANVA lanceerde op 19 april 2019 ANVA Hub, het platform dat staat voor ultieme connectiviteit, efficiënte bedrijfsprocessen, toekomstbestendige technologie en vertrouwen op continuïteit in performance. Daarnaast is het platform gebruiksvriendelijk, auditeerbaar, schaalbaar, altijd up-to-date en voorzien van optimale security. ANVA ziet ANVA Hub als de verbindende schakel in het applicatielandschap van ANVA-klanten. Hét toegangspuntaal naar de ANVA-oplossingen en naar de oplossingen van ANVA-partners.

ANVA organiseert de ontwikkeling van ANVA Hub in domeinen. Het uitgangspunt is domeinen zo specifiek mogelijk te ontwikkelen voor het doel van het domein. Een voorbeeld hiervan is het

relatiedomein. ANVA gebruikt hiervoor Domain-Driven Design, een softwareontwikkelingsaanpak die de nadruk legt op het programmeren van een domeinmodel, dat veel kennis bevat van de processen en regels van een domein (Fowler, 2020). Deze aanpak leidt ertoe dat een domein zo goed mogelijk aansluit bij de behoeftes van de business. De data binnen een domein wordt ontsloten middels een API-First-aanpak (Wagner, 2021). De API-First-aanpak betekent dat voor alle ontwikkelingen binnen ANVA Hub de API (Application Programming Interface) beschouwd wordt als "first-class citizen". Alles draait om het idee dat de businesslogica geconsumeerd wordt middels een API die door meerdere applicaties gebruikt kan worden.

ANVA kiest SIVI AFS als basis om de domeinen naar externe partijen te ontsluiten. SIVI AFS geeft ANVA een aantal voordelen tegenover een eigen standaard of de AFD 1.0-standaard. SIVI AFS is logisch opgebouwd en getoetst in de markt, waardoor ANVA's API's ook direct voldoen aan deze criteria. Daarnaast is SIVI AFS volledig in het Engels beschikbaar. Hierdoor is het ook mogelijk voor niet-Nederlandssprekende ontwikkelaars te koppelen met ANVA-systemen en ontstaat er minder verwarring. De ANVA-ontwikkelaars kunnen het Engelse jargon gemakkelijk overnemen in ANVA's eigen domeinen, deze zijn ook voornamelijk in het Engels, wat foutieve vertalingen voorkomt.

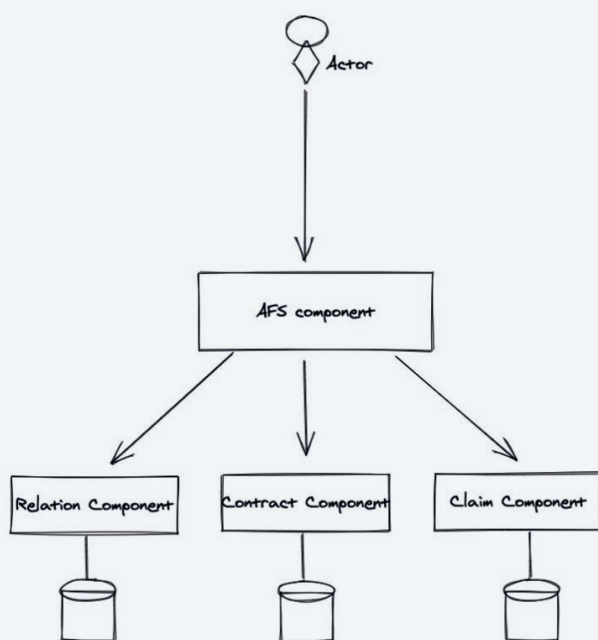
SIVI heeft met SIVI AFS veel voorbereidend werk gedaan en ondersteunt ANVA waar nodig bij het opzetten van de koppelvlakken en de structuur. Het gebruik van SIVI AFS geeft ANVA het voordeel om de interne datastructuur los te koppelen. Hierdoor komen er minder afhankelijkheden op de interne structuur, waardoor ANVA op dat gebied flexibel blijft en geen innovatiekracht verliest. De SIVI AFS-ontsluiting kan dan gecontroleerd uitgerold worden om op deze manier de impact te minimaliseren. Als laatste kan ANVA, door de uitgebreide documentatie van SIVI AFS, tijd en geld besparen. Partijen die willen koppelen kunnen worden verwezen naar de uitgebreide documentatie (SIVI, 2021). Hierdoor hoeft ANVA alleen basisgegevens en afwijkingen te documenteren.

Vanzelfsprekend ziet ANVA ook nadelen in het gebruik van SIVI AFS. De afhankelijkheid van SIVI AFS groeit en daarmee zal ANVA iets moeten inleveren op flexibiliteit en verwerkingstijd van vernieuwingen. Ontbrekende onderdelen in SIVI AFS moeten eerst uitgewerkt, goedgekeurd en doorgevoerd worden, voordat ze bruikbaar zijn vanuit de standaard. Het grootste nadeel dat nu gezien wordt is de mapping van de domeinen. Dit biedt zoals eerder uitgelegd voordelen maar kost ook tijd, daarbij worden mappingsfouten mogelijk minder snel gevonden doordat ze niet intern gebruikt worden.

Het ANVA-architectuurlandschap is gebaseerd op een microservice-architectuur (Lewis, 2014) die een groot palet van verschillende componenten kent. Al deze componenten hebben hun eigen dataopslag en zijn ook verantwoordelijk voor de gegevens. Deze gegevens worden vanwege security en privacy ook nog eens per ANVA-klant in een

aparte database opgeslagen. De componenten mogen alleen middels de gedefinieerde API's met elkaar communiceren en het is niet mogelijk om direct data uit de dataopslag van een andere microservice op te halen. Dit voorkomt vertroebeling van businesslogica en geeft componenten een sterke verantwoordelijkheid voor hun eigen gegevens. Om met deze beperking om te gaan wordt het SIVI AFS ingezet als een façade in het landschap (Figuur 1). De actor, ook wel partner van ANVA, koppelt haar softwaresysteem met een component op ANVA Hub die gedefinieerd is volgens SIVI AFS. Deze component verzamelt uit verschillende andere componenten de gegevens die nodig zijn om de gevraagde gegevens te leveren. Zo komt relatiedata uit de relatiecomponent en contractgegevens van die desbetreffende klant logischerwijs uit de contractcomponent. De businesslogica die verwerkt zit in de componenten wordt hergebruikt bij het ophalen van de gegevens.

Al deze endpoints zijn beveiligd middels HubID. Hierdoor dient de actor eerst in te loggen via één van de beschikbare OpenID flows. Door bij het aanroepen van het component gebruik te maken van de autorisatie van de gebruiker, kunnen de onderliggende componenten de autorisatie van de gebruiker afdwingen. Hierdoor kan de AFS-component nooit meer gegevens tonen dan het individuele onderliggende component. Bij wijziging van autorisaties wordt dit direct toegepast op alle koppelingen die gebruik maken van de AFS-component.



Figuur 1 Hub Architectuur met SIVI AFS

Om de performance te optimaliseren worden de API's van de onderliggende componenten aangesproken middels interne routing. Ook worden de verschillende componenten asynchroon aangesproken om het wachten op voorgaande calls te beperken. Nadat een response is ontvangen, wordt de data via een vertaling omgezet naar de AFS-datastructuur; hiervoor zijn slimme mechanismen gemaakt om dit zo onderhoudsvriendelijk mogelijk te doen. Uiteindelijk wordt een gecombineerd resultaat van alle vertalingen teruggeven aan de client.

Dit is de eerste opzet voor het gebruik van SIVI AFS binnen het architectuurlandschap van ANVA. Vanuit deze ervaring wordt

onderzocht hoe de standaardisatie verder kan worden doorgevoerd binnen het ANVA landschap. Hierbij wordt ook gedacht aan het aanpassen van de onderliggende componenten, om zo geen mapping meer te hoeven doen tussen de interne componenten en de externe interfaces. Deze keuze zal gebaseerd worden op de ervaringen met de eerste opzet.

Als aanvulling op dit onderwerp kijkt ANVA naar de mogelijkheid om GraphQL (GraphQL, 2021) toe te voegen aan de SIVI AFS-component. Met GraphQL stelt ANVA de partner in staat om zelf de data te beschrijven die zij wenst te ontvangen. Hiermee krijgen partners de mogelijkheid om beter te voldoen aan de AVG-richtlijnen. In de use case waar we

een overzicht willen tonen van de polissen met beperkte velden (bv. omschrijving en polisnummer) kan de client deze specifieke gegevens vragen middels de beschrijving zoals aangegeven in figuur 2. Deze toevoeging is vrij gemakkelijk te maken door de standaardisatie van SIVI AFS, daarnaast geeft het ANVA de mogelijk flexibiliteit te geven aan de afnemers van de API.

```
{
  party(id: "1") {
    policy {
      afdDefinitionName
      contractNumber
    }
  }
}
```

Figuur 2 GraphQL voorbeeld

Referenties

Fowler, M. (2020, 04 22). *DomainDrivenDesign*. Retrieved from martinFowler.com: <https://martinfowler.com/bliki/DomainDrivenDesign.html>

SIVI. (2021, 02 10). *SIVI All Finance Standaard (SIVI AFS)*. Retrieved from SIVI: <https://www.sivi.org/standaarden/sivi-all-finance-standaard/>

Wagner, J. (2021, 01 25). *Understanding the API-First Approach to Building Products*. Retrieved from Swagger: <https://swagger.io/resources/articles/adopting-an-api-first-approach/>

Lewis, J. (2014, 03 25). *Microservices*. Retrieved from martinFowler.com: <https://martinfowler.com/articles/microservices.html>

SIVI. (2021, 01 25). *JSON*. Retrieved from SIVI: <https://www.manula.com/manuals/sivi/sivi-all-finance-standard/1/en/topic/json>

GraphQL. (2021, 02 10). *GraphQL*. Retrieved from GraphQL: <https://graphql.org/>

Over ANVA

ANVA ontstond in 1975 als maatschappijafhankelijk automatiseringsproject van NVA (Nederlandse Vereniging van Makelaars in Assurantiën en Assurantieadviseurs. Nu Adfiz).

ANVA is uitgegroeid tot een totaalleverancier op het gebied van software en ondersteuning voor de gehele verzekeringsmarkt, van verzekeraars en serviceproviders tot beursmakelaars, volmachten en financieel advieskantoren. Meer dan tienduizend financieel professionals werken met de software om miljoenen consumenten en duizenden ondernemers een goede verzekering te kunnen aanbieden. Dankzij de jarenlange ervaring in de verzekeringsbranche, kent ANVA het werkveld en bijbehorende uitdagingen als geen ander.

Handmatig berichtschema's opstellen behoort tot verleden

Gebruik AFD Online Samenstellen

Danny Schaapherder

Al weer bijna een jaar is de tool AFD Online Samenstellen (AOS) op de SIVI-website beschikbaar. Met deze SIVI-tool voor het samenstellen, beheren, distribueren én certificeren van AFD-definities is het sinds kort ook mogelijk AFD 2.0-definities in JSON samen te stellen. In deze bijdrage een overzicht.

AFD Online Samenstellen (AOS)

AOS is ontwikkeld om AFD-definities op te stellen (zie kader) zonder inhoudelijke kennis rond XML-schema's, XPATH (AFD 1.0) of JSON Schema en JSONPath (AFD 2.0).

De inzet is dat je op basis van een AFD-definitie voor de afhandeling van een bericht, de aanroep van een service of de specificatie van een registratie precies weet wat je moet doen. Ontwikkelaars kunnen op basis van een AFD-definitie in moderne ontwikkelomgevingen zelf een deel van hun code generen.

Met behulp van het gebruik van AFD-definities halen we veel kosten uit de keten rond afstemming en testen.

Opstellen AFD-definitie

Het opstellen van een AFD-definitie gaat via de tool AFD Online Samenstellen (AOS) op de SIVI-website. Op de SIVI-website zie je hoe je hier een account voor kan krijgen. Na het opstarten maak je eerst een keuze voor domein of branche. Dit maakt het veel makkelijker een AFD-definitie op te stellen. In plaats van het hele AFD zie je vervolgens alleen de entiteiten en attributen die je kan toepassen binnen dat betreffende domein. Tevens zie je direct welke beperkingen er zijn: bijvoorbeeld rondom herhalingen van entiteiten en waardebereiken binnen codelijsten.



Contact | SIVI

BEHEER ▾ MENU ▾

AFD Online Samenstellen

Terug

Samenstellen AFD Definitie
SIVI - AFD 2.0-Contractdocument-Motorrijtuigen - Motorrijtuigen product A

Entiteiten	Attributen
<p>Wijzig entiteitsselectie</p> <ul style="list-style-type: none">▶ commonFunctional (*)▼ policy - policyDetails (*)<ul style="list-style-type: none">▶ party - regularDriver▶ coverage - hullVehicle▼ coverage - thirdPartyLiability<ul style="list-style-type: none">premium - premiumDetails▼ object - motorVehicle	<p>Wijzig attribuutselectie</p> <ul style="list-style-type: none">coverageCode (string)collectivityFactor (number)deductibleAmount (number)grossPremiumInstallment (number)costsInstallmentAmount (number)insuredAmountMaterialDamage (number)insuredAmountPersonalInjury (number)

Wat is de AFD-definitiestandaard?

De AFD-definitiestandaard biedt een raamwerk voor specificatie en validatie van berichten in twee delen, gebaseerd op industriestandaarden:

- JSON Schema voor het specificeren van berichten met betrekking tot de structuur en het gebruik en samenhang van entiteiten en attributen. Het gebruik van JSON Schema bevordert het opstellen van eenduidige specificaties. Daarnaast kan JSON Schema worden gebruikt als basis voor geautomatiseerde controle van berichten, het inrichten van databases en het inrichten van dialogen.
- JSONPath voor het specificeren van verbandscontroles tussen attributen en hun veldwaarden, wat niet kan met JSON Schema. Het gebruik van JSONPath bevordert het creëren van eenduidige specificaties.

De combinatie van JSON Schema en JSONPath levert een nauwkeurige AFD-definitie op die kan worden ondersteund door verschillende (vaak vrij verkrijgbare) softwarebibliotheken. Het gebruik ervan door ontwikkelaars is eenvoudig en resultaten kunnen snel worden bereikt. De introductie van de AFD-definitiestandaard is een belangrijke stap voorwaarts voor de sector als het gaat om het afhandelen van diensten en het vastleggen van gegevens. De standaard bevordert de datakwaliteit en efficiëntie van distributieketenprocessen.

Effecten van het gebruik van de AFD-definitiestandaard:

- Vereenvoudiging van het beheer van productspecificaties.
- Door het gebruik van JSON-standaarden (met veel vrij verkrijgbare open source tooling) krijgen ontwikkelaars ondersteuning bij softwareontwikkeling.
- Verbetering van de datakwaliteit in berichten en databases.
- De controle van gegevens tijdens opslag en berichtenuitwisseling voorkomt onjuiste en onvolledige berichten.
- Meer gebruiksvriendelijke dialogen.
- Het volgen van de gedefinieerde productstructuur en controles bij schermdialogen zorgt voor een betere interactie met de gebruiker en dus ook voor een betere invoerkwaliteit van gegevens.
- Efficiëntere uitvoering van processen.
- De ontwikkeltijd zal afnemen dankzij de flexibele configuratie van producten in software.
- Verkorte time-to-market voor implementaties en wijzigingen.

Meer informatie over de AFD-definitiestandaard is via deze [link](#) te vinden op onze website.

De eerste stap om een AFD-definitie aan te maken het invoeren van een aantal eigenschappen, zoals een naam, versienummer etc. Een AFD-definitie specificeer je voor een functie en een situatie, bijvoorbeeld de aanroep van een premieberekening of het resultaat van de aanvraag van een polis. Het begrip 'functie' moet je hier overigens breed opvatten: ook de registratie van bijvoorbeeld een polis wordt als een functie gezien. In AOS kun je een grote hoeveelheid standaardfuncties selecteren, waaronder functies die in het SIVI API-raamwerk gedefinieerd zijn. Het is ook mogelijk zelf een functie(-naam) op te geven.

De volgende stap is het per functie selecteren van de gewenste entiteiten en attributen. Per entiteit kun je hierbij het minimale en maximale aantal voorkomens instellen, voor ieder attribuut geef je aan of het verplicht of optioneel is. Daarnaast is het mogelijk diverse voorwaarden toe te voegen aan de eigenschappen van attributen:

- Zo is het mogelijk bij attributen met een codelijst een subset van de codelijst te selecteren. Voor grotere codelijsten is het mogelijk om hiervoor een bestand met codewaarden te importeren.
- Bij attributen van het type String (zonder codelijst) is het mogelijk de minimale en maximale lengte van het attribuut in te stellen.
- Bij attributen van het type Number is de minimale en maximale waarde en het aantal decimalen in te stellen. In plaats hiervan is het ook mogelijk een optielijst aan te maken. Bijvoorbeeld: bij attribuut `paymentTermInMonths` een optielijst aanmaken met 1, 3, 6 en 12.
- Nieuw in AFD 2.0 is de ondersteuning van het gebruik van attributen van het type array. Hiermee is het mogelijk een attribuut te definiëren met meerdere waarden in het bericht. Ook is het bij deze attributen mogelijk om voorwaarden te stellen over de array zelf, zoals het aantal (verschillende) waarden dat een attribuut mag bevatten. Welke attributen van het type array zijn ligt vast in het domein dat je gekozen hebt.

Eigenschappen

Naam
coverage - thirdPartyLiability

Toegestane herhalingen entiteit

Functie	min Occurs	max Occurs
Baseline	0	2147483647
calculatePremium_offer_request	0	1
calculatePremium_offer_response	0	1

Eigenschappen

Naam Type

Omschrijving

Codelijst

Codelijst

Gebruik binnen functie

Functie	Conditie
calculatePremium_offer_request	<input type="text" value="Verplicht"/>
calculatePremium_offer_response	<input type="text" value="Verplicht"/>

Verbandscontroles

De verbandscontroles binnen een AFD-definitie hebben als doel verbanden tussen verschillende attributen en hun waarden te controleren. Bijvoorbeeld als de verzekerde som groter is dan €50.000, dan mag het eigen risico niet €0 zijn. Deze verbandscontroles zorgen voor een snellere en/of betere kwaliteit van invoer op schermen, aanroep van services of de registratie van gegevens. Voor AFD 2.0 heeft een verbandscontrole de vorm van een conditionele expressie in JSONPath.

Output

AOS maakt na het vrijgeven voor elke functie binnen een AFD-definitie een zip-file met daarin drie JSON-bestanden. Twee JSON Schema's waarvan een (1) functieschema met alle entiteiten en attributen en een (2) codelijstschema waarin alle codewaarden zijn opgenomen. Daarnaast bevat het een JSON-bestand met de JSONPath-expressies voor de verbandscontroles. AOS plaatst deze zip-files in een register van waaruit men ze kan downloaden. Overigens is het ook mogelijk AFD-definities aan te maken die niet zichtbaar zijn voor derden, of wel zichtbaar maar niet te downloaden.

Releases AOS in 2021

Op dit moment biedt AOS ondersteuning voor het samenstellen van AFD 2.0-definities in de branches Motorrijtuigen, Reis en Inkomen. In de komende maanden zullen we stapsgewijs het aantal branches uitbreiden en een generieke branche toevoegen. Op dit moment zijn voor AOS de onderstaande releases voorzien.

Release voorjaar 2021

- AOS-gebruikers krijgen de mogelijkheid zich aan te melden voor de notificatieservice om via e-mail op de hoogte te blijven van nieuwe of gewijzigde AFD-definities in het register.
- Een AOS-API komt beschikbaar om (semi-)automatische verwerking van AFD-definities mogelijk te maken. Door middel van de API kun je opvragen welke nieuwe gepubliceerde AFD-definities beschikbaar zijn en kun je AFD-definities ophalen. Dit maakt het mogelijk geautomatiseerd te controleren op updates en (deels) geautomatiseerd deze updates door te voeren. Het feit dat AFD-definities gebruik maken van de gangbare XML- en JSON-standaarden biedt hier veel mogelijkheden.
- Er komt een verbeterde navigatie binnen het publieke register van gecertificeerde en gepubliceerde AFD-definities.
- De AFD-definitieschema's van alle functies komen automatisch in één zip-container. Het certificeren en publiceren gebeurt vanaf dat moment ook per groep van functies en niet meer per losse functie binnen een AFD-definitie.

Release najaar 2021

In het najaar van 2021 voorzien we een verdere uitbreiding van de ondersteuning van AFD 2.0 in AOS. We anticiperen op de vraag van gebruikers: suggesties zijn welkom! Op dit moment staan het gebruik van customAttributes en de mogelijkheid om van pseudocode gebruik te maken bij het invoeren van verbandscontroles op de lijst.

Draag bij aan kostenbesparingen binnen de keten!

Met AOS biedt SIVI de markt een makkelijk te gebruiken en geavanceerde tool waarmee je eenduidige berichtspecificaties kunt opstellen, beheren en te distribueren. Door certificering zijn gebruikers van AFD-definities er van verzekerd dat de AFD-definities voldoen aan de branchestandaard AFD. Daarnaast kan de gebruiker door de inzet van JSON-standaarden kosten besparen in het ontwikkeltraject. Draag bij aan kostenbesparingen binnen de keten en start met het gebruik van AFD-definities. Heb je nog geen account? Maak deze dan snel aan via deze [link](#) en bekijk hoe AOS jou kan helpen bij het snel en eenvoudig opstellen van AFD-definities.

Een eerste indruk van AFD 2.0

Antoine Boot

Antoine Boot



Antoine Boot is Integratie Specialist / Cloud Engineer bij de ICT-organisatie van De Goudse en een regelmatige gebruiker van AFD 1.0. Wij hebben Antoine gevraagd wat zijn eerste indruk is van AFD 2.0 en welke vragen hij

voor SIVI heeft. Hieronder zijn eerste observaties en vragen. We sluiten af met een korte reactie van SIVI op de vragen.

AFD 2.0 vanuit het gezichtsveld van een (integratie-)ontwikkelaar

Antoine Boot

AFD is al jaren een fundamentele bouwsteen om informatie op een uniforme wijze uit te wisselen in het Nederlandse verzekeringslandschap tussen de diverse partijen in de keten. Midden jaren tachtig begonnen met standaardisatie-afspraken die leidden tot de introductie van ADN, en die in de jaren negentig verder evolueerden tot AFD 1.0.

Inmiddels is het 2021 en is de wereld qua inzichten om berichtuitwisseling vorm te geven knap veranderd. REST heeft het stokje overgenomen van SOAP voor de implementatie van web services (API). XML wordt tegenwoordig als "zwaar" beschouwd en is daarom opgevolgd door JSON als notatiewijze voor berichten.

Gedurende deze periode is ook ons vocabulaire om onze verzekeringsberichten op te stellen veranderd door o.a. productontwikkelingen en andere veranderingen in de markt. Aangezien het gemakkelijker is om dingen toe te voegen aan een standaard dan er uit te halen heeft dit geleid tot een fors gegroeide datacatalogus. Een catalogus die vol zit met keuzes uit het verleden, onduidelijkheden, verkeerd gekozen namen en synoniemen die op momenten het opstellen van eenduidige berichten knap in de weg kunnen zitten.

Eigen ervaringen AFD 1.0

Binnen onze organisatie hebben we jaren geleden als (integratie)ontwikkelaars ervoor gekozen om AFD te omarmen als ons interne canonieke datamodel (CDM) voor onze Enterprise Service Bus (ESB), ondanks dat SIVI destijds het AFD vooral positioneerde als berichtenformaat tussen partijen in. De definitie van Forrester voor een CDM is: "Een Canoniek Data Model is een model van de definities, samenhang en structuur van informatie die refereert aan een regelset die vastgelegd is binnen een gedefinieerde context, voor communicatie tussen een set van applicaties en/of partijen."

Aangezien we een verzekeraar zijn, lijkt het gebruik van AFD als CDM een logische keuze. Het AFD is immers een model dat gebaseerd is op standaardterminologie en betekenis. De praktijk is echter weerbarstiger. Een van de uitdagingen die je in de praktijk hebt is dat het de nodige vaardigheid kost om een goed bericht op te stellen dat zowel functioneel je product beschrijft als ook technisch verwerkt kan worden door ICT-systemen.

Uitdagingen waar wij tegenaan liepen is dat de syntax van AFD 1.0 niet erg makkelijk leesbaar is door zijn vele cryptische afkortingen. Daarnaast kan je de interne databehoeftes niet altijd goed kwijt en heeft de gebruikersorganisatie moeite met het mappen van zijn eigen vocabulaire (programmatuur/output) naar die van de gestandaardiseerde set in AFD. Vanuit de technische kant loop je vaak tegen de gekozen dataformaten (en bijbehorende codelijsten) en lengtebeperkingen aan binnen de standaard, ten opzichte van die van je eigen systemen.

Eerste blik op SIVI AFS en AFD 2.0

Vorig jaar heb ik een eerste blik geworpen op de bètaversie van SIVI AFS en AFD 2.0. Zoals in de inleiding geschetst zijn er in de tijd de nodige veranderingen geweest in de markt en vind ik het een goede stap dat het SIVI op basis van de feedback van de markt heeft besloten een vernieuwde standaard in de markt te zetten. Het geeft de ruimte om te vernieuwen en (deels) afscheid te nemen van een stuk historie. Het indikken van de data catalogus is een voorbeeld hiervan.

Een opvallende wijziging is de overgang naar het Engels in AFD 2.0. De argumentatie hierbij is dat er steeds meer met buitenlandse ontwikkelaars wordt gewerkt. Op zich een begrijpelijke keuze maar mij is altijd geleerd dat je bij het modelleren vooral de taal van de gebruiker moet volgen en niet andere begrippen moet introduceren. Vanuit die gedachte is de overgang naar het Engels voor de begrippen uit het AFD niet even logisch. Wat ik wel fijn vind is dat alles een stuk leesbaarder is omdat attributen en entiteiten netjes worden uitgeschreven. Dit zal zeker helpen als AFD-berichtstructuren worden omgezet naar (transformatie)code. Gelukkig heeft het SIVI ook geluisterd naar adviezen op het gebied van uniforme naamgeving en aandachtspunten inzake omzetbaarheid naar code.

In AFD 1.0 was het default dataformaat XML. In AFD 2.0 is dit het technisch efficiëntere JSON, waardoor berichten kleiner worden tijdens de uitwisseling. Conversie tussen JSON en XML is nog steeds mogelijk maar dat zal een API-aanbieder dan wel moeten faciliteren. Een goed ontworpen API zal de aanroepende partij de mogelijkheid geven om zelf aan te geven in welk formaat de data-uitwisseling zal plaatsvinden. Wat ik vanuit mijn programmeerachtergrond met object-oriëntatie ook fijn vind van de JSON-notatie is dat er nu met generieke entiteiten als party en object wordt gewerkt. Dat biedt perspectieven voor de softwareontwikkelaars op het gebied van hergebruik van code.

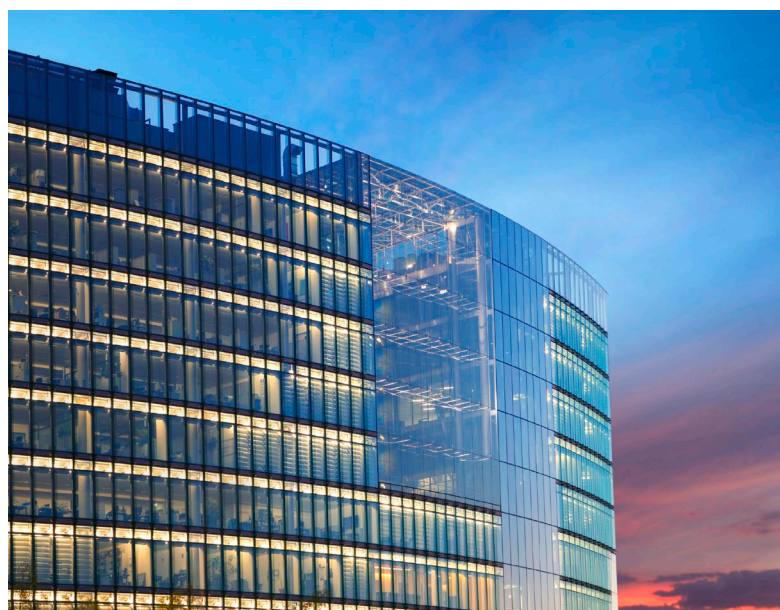
Een aandachtspunt bij de overgang naar JSON is mijns inziens het loslaten van de lengtelimieten. Het moet niet zo zijn dat er een mismatch gaat ontstaan tussen de implementaties van de diverse partijen op dit gebied, want dat kan tot dataverlies gaan leiden of uitval van berichten in de keten. Ik ben benieuwd

wat de gedachten vanuit het SIVI hierover is. De REST/JSON-community zette zich in het verleden nogal krampachtig af tegen de schema/versionering-wereld van SOAP/XML. De behoefte aan een eenduidig contract was echter groter dan de aversie en inmiddels kent de REST/JSON-community ook zijn eigen schema-technieken; ik ben ervan overtuigd dat hier een oplossing voor komt.

Het laatste aspect is dat de standaard ruimte biedt voor eigen attribuutuitbreidingen. Een oplossing die kan helpen bij bepaalde uitdagingen bij berichtuitwisselingen binnen het eigen ICT-landschap. Waar wel voor gewaakt moet worden is dat dit soort uitbreidingen onterecht gebruikt worden voor de communicatie buiten de eigen organisatie. Communicatie over de bedrijfsgrenzen heen vergt in de basis standaardisatie: maatwerk moet een uitzondering blijven.

Toekomst

Ik ben benieuwd hoe de markt AFS en AFD 2.0 gaat adopteren. Zeker voor de partijen die de afgelopen jaren hun landschap hebben ingericht op AFD 1.0 zal dit een aanvullende investering vergen. Minimaal in een conversie-module tussen AFD 2.0 en AFD 1.0. De papieren van SIVI AFS en AFD 2.0 zien er in ieder geval goed uit om op een moderne manier met elkaar te integreren in de keten.



Reactie SIVI

■ Gebruik van Engels

Binnen een gestandaardiseerd datamodel is het volgen van de taal van de gebruiker altijd een uitdaging. Voor het gebruik van AFD 2.0 is dat niet anders. Waar je in AFD 1.0 een cryptisch attribuutnaam kunt koppelen aan je eigen interne uitleg van een attribuut is dat in AFD 2.0 wel minder makkelijk door de uitgeschreven attribuutnaam.

Wij zien dit vraagstuk overigens het meest terug bij het gebruik van codelijsten. Deze zijn overigens nog steeds gewoon in het Nederlands. Een voorbeeld is de codelijst ADNBMU met code 'R' voor 'Riet'. Er zijn partijen die liever 'Rietendak' zien, en anderen die liever 'Rietenkap' zien omdat dit beter aansluit bij de communicatie rond het product. Bij de beroepenlijst zien we dit soort vragen ook. Voor dit soort situaties zal het binnenkort mogelijk zijn in het verlengde van een AFD-codelijst een eigen omschrijving toe te voegen. Op deze wijze kunnen gestandaardiseerde codelijsten toch aansluiten bij de terminologie van de eigen organisatie.

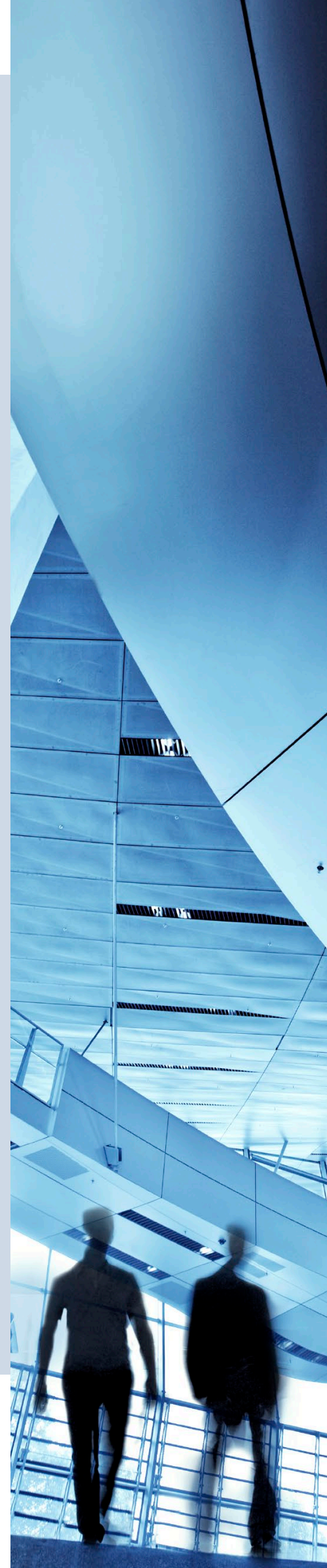
■ Lengtelimieten JSON

Het omgaan met datalimieten of beperkingen binnen AFD 2.0 is zeker een onderwerp waar SIVI nog meer duiding kan geven. Beperkingen kan men goed aangegeven in JSON Schema's: deze gebruiken we op dit moment bij het aanmaken van AFD-definities voor AFD 2.0 in AOS op de SIVI-website. De vraag voor ons is op welke plaatsen restrictie vanuit de standaard anno 2021 en hierna nog echt noodzakelijk is. In een wereld waar men data steeds vaker opslaat in NoSQL-databases en de velden op schermen scrollable zijn. Op dit moment kijken we nog naar reacties uit de markt. De lijn die we nu volgen is dat in het algemene AFD 2.0 er geen wezenlijke beperkingen zijn. Wel kan het zijn dat binnen een specifiek domein (bijv. premieberekening volmachten) beperkingen gaan gelden, voor een overgangperiode of voor onbepaalde tijd. We komen hier zeker op terug.

Voor het gebruik van parameters in een API call zullen we wel duidelijk gaan verwijzen naar de noodzaak van sterke typering, met behulp van bijvoorbeeld regular expressions. Hier is sterke typering een onderdeel van de afscherming tegen aanvallen van buiten (bijv. creëren van overbelasting van server door te lange input strings voor parameters).

■ Definitie van eigen attribuut

Het opstellen van een AFD-definitie voor AFD 2.0-berichten sluit af met certificeren. Op het moment dat AOS het customAttribute ondersteunt zullen wij in het verlengde van de certificering toetsen of partijen geen 'misbruik' maken van de customAttribute-constructie. Als dat het geval is zal SIVI er op wijzen. Op dit moment is onze indruk dat partijen zich goed bewust zijn van de noodzaak om het gebruik van deze constructie te beperken tot echt noodzakelijk.



Webservices ontwikkelen volgens het SIVI API-raamwerk

Wat te doen en waar te beginnen

Robin Oostrum

Jouw organisatie wil met haar software aansluiten op het SIVI API-raamwerk. Maar waar begin je? Hoe pak je dit raamwerk op en plot je dit op je eigen diensten? In deze bijdrage een denkraam om je op gang te helpen. Het hieronder uitgewerkte voorbeeld gaat over een verzekeraar, maar dit kan natuurlijk ook een softwareleverancier, service provider of dienstenaanbieder zijn.

In dit voorbeeld wil een verzekeraar een API aanbieden bij de vernieuwing van zijn Zakelijk Schade Product voor aansprakelijkheid (Bedrijfsaansprakelijkheid Flex). Je hoeft als verzekeraar natuurlijk niet alle 89 functies van het API-raamwerk te implementeren: je maakt een afweging en selectie van functies die bijvoorbeeld commercieel (in functie van verkoopkansen) en operationeel (in functie van kostenbesparingen) interessant zijn. Een randvoorwaarde als de systeem-infrastructuur kan daarbij zomaar een beperkende factor zijn: als bijvoorbeeld het product geadmistreerd is in een minder toegankelijke toepassing is een realtime webservice voor het ophalen van gegevens – retrievePolicy – minder voor de hand liggend. Naast de afweging vanuit je eigen agenda maak je ook een afweging vanuit het type bedrijfsvoering van de gebruikers van de API. Bijvoorbeeld: is de gebruiker een volmacht of een groot provinciaal kantoor? Per type bedrijfsvoering inventariseer je vervolgens de software die gebruik gaat maken van deze API. Gebruikt men specifieke koppelvlakken (als VPI) of past men de software specifiek voor het gebruik van deze API aan?

Bedrijfsaansprakelijkheid Flex		API-functies							
Doelgroep	Doel software	calculatePremium (quickQuote)	calculatePremium (policyApplication)	underwrite (new)	underwrite (change)	submitPolicy (new)	changePolicy (update)	retrievePolicy (single)	retrievePolicy (status)
Provinciaal (klein)	Klantmap (light)							✓	
	Vergelijkingssoftware	✓							
	Adviessoftware	✓							
Provinciaal (groot)	Online verkoop	✓				✓	✓		
	Klantmap							✓	
	Mid office					✓	✓	✓	✓
	Vergelijkingssoftware	✓							
	Adviessoftware	✓							
Volmacht	Klantmap	✓							
	Extranet (service provider)	✓	✓	✓	✓				
	Mid office		✓	✓	✓				
	Back office		✓	✓	✓				

- Doelgroep:** in deze kolom definiëren we de groepen gebruikers die de API gaan implementeren voor het betreffende product. In dit geval wil men een ketenbrede ondersteuning inrichten. De bediening van kleine provinciale kantoren is primair ingericht via het extranet (niet in deze tabel). Maar binnen klantmappen die in gebruik zijn wil men graag het raadplegen van een polis ondersteunen. Daarnaast is het van belang dat de advies- en vergelijkingssoftware een indicatieve premie kan berekenen. De verzekeraar werkt met een aantal grote adviesorganisaties samen die provinciaal willen werken, maar de ondersteuning voor het product wel willen opnemen in de zelf ingerichte sluitstraat (Mid office) en binnen een website waar een klant zelf kan sluiten (Klantmap). Voor de volmachten gaat de verzekeraar de API aansluiten op het VPI-platform. Twee functies zijn voorzien: premieberekening en acceptatie.

- **Doel software:** het is van belang goed te begrijpen welk type software de API-services gaat gebruiken. Dit speelt een rol bij de functionaliteit die de functies moeten bieden, maar ook bij het samenstellen van de shortlist van leveranciers die op de hoogte moeten zijn van het bestaan van de API. Het kan ook gewenst zijn met specifieke leveranciers afspraken te maken rond implementatie. Het voordeel van het volgen van standaarden is dat de software een generiek karakter heeft. Op die wijze kan men maatwerk voor specifieke partijen zo veel mogelijk vermijden. In de tabel hierboven staan algemene namen: je kunt hier natuurlijk ook de namen van specifieke leveranciers en omgevingen neerzetten, dat maakt het inzicht nog concreter. In het Overzicht software en services financiële dienstverlening vind je een actueel overzicht.

Geen microservices

Het SIVI API-raamwerk is ingericht om services te faciliteren, niet microservices. Microservices zijn van toepassing wanneer je bijvoorbeeld als verzekeraar voor het opmaken van een polis aparte webservices hebt voor achtereenvolgens het uitvragen van klantdata, het opstellen van de dekking en het toevoegen van objecten. Services daarentegen faciliteren een hele processtap, in bovenstaand voorbeeld het opmaken van de polis.

De belangrijkste reden voor deze keuze is dat microservices het best tot hun recht komen binnen het gericht ontwikkelen van een specifieke toepassing. Zo kan het voor partijen binnen verschillende processen of op verschillende momenten relevant zijn om inzicht te hebben in contractpartijen. Dit is ook zeer interessant als bijvoorbeeld het doel is een detailintegratie met de software van een specifieke leverancier vorm te geven. Op dat moment is het kunnen uitvoeren van detailoperaties makkelijk en maakt dit het ontwikkeltraject wendbaar. Maar als men voor het inrichten van een proces een tiental verschillende services moet gebruiken, brengt dit ook meer complexiteit en kosten met zich mee. Het ligt niet voor de hand dat elke klant of leverancier met een eigen toepassing een fijnmazige koppeling zal inrichten. De agenda van SIVI is om de kosten bij gegevensuitwisseling binnen de keten te beperken: daarom ligt de focus van het API-raamwerk nu op het definiëren van services die een hele operatie omvatten.

Het kan zijn dat niet alle benodigde functies passen binnen het SIVI AFS API-raamwerk. Soms is er geen andere optie dan een bedrijfseigen functie te implementeren. In het artikel over Proprietary APIs op pagina 16 gaan we hier uitgebreid op in.

De impactbepaling van aansluiting op het SIVI API-raamwerk zal een aantal iteraties vergen. SIVI begeleidt hierin graag: zoals ondersteuning bij het invullen van het denkraam hierboven of ondersteuning bij het maken van keuzes/afwegingen.

Vragen of meer informatie nodig over aansluiting op het API-raamwerk? [Mail ons.](#)

Wat als het SIVI API-raamwerk niet past?

SIVI AFS voor bedrijfseigen API-implementaties

John Boots

Het SIVI AFS API-raamwerk voorziet binnen de financiële dienstverlening in een standaard voor gegevensuitwisseling, zoals het afsluiten van een polis of het berekenen van een premie. Het doel van het SIVI AFS API-raamwerk is de verschillende processen binnen de keten te ondersteunen. Het SIVI AFS API-raamwerk omvat alles rond de voortbrenging en de uitvoering van financiële producten. Het SIVI AFS API-raamwerk richt zich op ketenpartijen, derden (herstellers, experts, etc.) en leveranciers. De standaard opbouw binnen het SIVI AFS API-raamwerk zorgt voor eenduidige uitgangspunten en uniformiteit. Partijen die software ontwikkelen en gebruik maken van APIs gebaseerd op het SIVI AFS API-raamwerk kunnen hierop anticiperen. Zo beperken we gezamenlijk de kosten voor de implementatie van toepassingen.

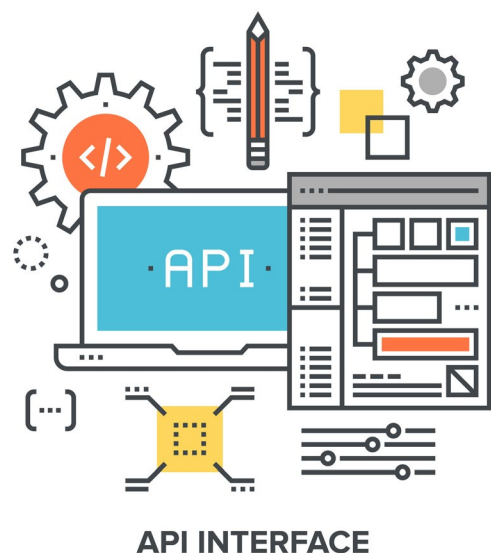
SIVI neemt functies binnen het SIVI AFS API-raamwerk op als ze een min of meer generiek doel dienen binnen de keten (ze zijn relevant voor meerdere ketenpartijen). Is een functie niet beschikbaar, dan kan men hem aanvragen, analoog als hoe dat nu voor AFD is ingericht. Tegelijk zijn er altijd bedrijfseigen situaties waarbij er geen sprake is van brede relevantie.

Hiernaast twee voorbeelden ▶

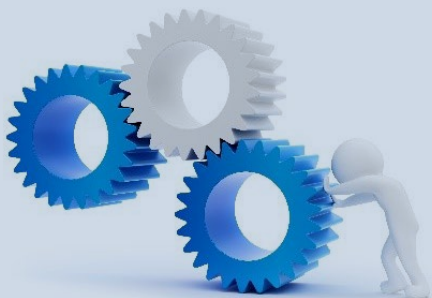
- Leverancier biedt een dienst aan voor risico analyse rond werkmaterieel. Met een aantal webservices kan de gebruiker deze analyse in stappen bevragen. De functie calls en attributen zijn zo dienst-specifiek dat ze niet in het SIVI AFS API-raamwerk passen.
- Verzekeraar wil webservice voor aanvraag verzuimverzekering ontwikkelen. De informatie rond de polis staat in een ander systeem dan de informatie rond en onderbouwing van de te verzekeren loonsom. Systeemtechnisch moet een aanvraag voor de korte termijn over twee services (endpoints) verdeeld worden. Dit is een variant voor de API-functie submitPolicy die zo situatie specifiek is, dat hij niet in het SIVI AFS API-raamwerk past.

Toch is het belangrijk zo veel als mogelijk aansluiting te houden bij de voordelen van gestandaardiseerde gegevensuitwisseling binnen het SIVI AFS API-raamwerk. Hiervoor zijn richtlijnen uitgewerkt, waaraan alle functies binnen het SIVI AFS API-raamwerk voldoen.

Deze richtlijnen kan een organisatie ook inzetten voor het ontwikkelen van bedrijfseigen APIs. Dit borgt dat een bedrijfseigen API aansluit bij de ontwerpcriteria en op de juiste wijze gebruik maakt van AFD-structuren en het AFD-model. Merk op dat men ook voor bedrijfseigen APIs verzoeken kan indienen voor entiteiten, attributen en codelijsten. Op het moment dat SIVI deze verzoeken niet kan invullen, kan men met de customAttribute constructie op een generieke wijze eigen attributen toevoegen.



Het SIVI AFS Handboek geeft ontwerper/analist en ontwikkelaar informatie en uitleg bij het opzetten van bedrijfseigen APIs (7.2 Proprietary API). Onderwerpen die aan de orde komen zijn onder andere:



- Naamgevingconventies
- Definiëren van API eindpunten (URI)
- Definiëren van de structuur van de data uitwisseling (payload)
- Gebruik van http-operaties (GET, POST, PUT, PATCH)
- Gebruik van pad-parameters (/resources/{resource})
- Gebruik van query-parameters (/resources/lijst&item=item)

Daarnaast is er een toelichting rond de indeling van het antwoordbericht. Het antwoordbericht kent een standaard indeling en is bij foutsituaties aanvullend op de 4xx http code (RFC 7807 specificatie) voorzien van extra informatie.

Uniformiteit van API-ontwerp helpt analisten en ontwikkelaars bij het opzetten, implementeren en onderhouden van APIs. Bij de afstemming tussen partijen bij implementatie trajecten is uniformiteit wenselijk maar ook bij uitwisseling tussen en opvolging van analisten en ontwikkelaars.

In de komende periode wordt de opzet en gebruik van de "Proprietary API" breder uitgewerkt in het online SIVI AFS Handboek. [Vragen, opmerkingen en tips](#) zijn zeer welkom; hiermee kunnen wij het SIVI AFS API-raamwerk en het gebruik van "Proprietary API" zo optimaal en helder mogelijk maken.

5 vragen aan...



Naam: Paul Janssen

Rol: Product Director

Organisatie:



1 Wat is Ockto?

Ockto is een platform waarmee individuen makkelijk, snel en veilig hun persoonlijke gegevens kunnen ophalen uit bronsystemen en deze gegevens kunnen delen met aangesloten organisaties die hen een verzoek tot het delen van gegevens doen. Hierbij worden alleen de gegevens gedeeld die de dataverzoeker nodig heeft.

2 Jullie hebben samen met SIVI de mapping naar SIVI AFS (AFD 2.0) opgesteld. Wat brengt dit jullie?

Ockto sluit graag aan op een standaard als het AFS, zodat het aansluiten op Ockto eenvoudiger wordt. Bijvoorbeeld voor partijen die deze standaard al gebruiken of een softwarepakket gebruiken dat dit al doet.

Ockto levert het Ockto Data Model (ODM) waarin de door het individu gedeelde gegevens zijn opgenomen. Dit model is al door veel instellingen ontsloten. Daarnaast levert Ockto optioneel ook importbestanden voor de meest gebruikte adviessystemen. Dit vergt veel onderhoud en specifieke kennis. Onze verwachting is dat veel adviessystemen het AFS zullen gaan ontsluiten en zodoende ook het Ockto Data Model kunnen verwerken.

3 Wat voor reacties kregen jullie op het persbericht rond de mapping?

Om eerlijk te zijn nog opvallend weinig. Onze verwachting is dat dit alsnog snel zal ontstaan. Met name in de schadeverzekeringensector is Ockto nog beperkt geïmplementeerd. Die sector zou via het AFS met beperkte inspanning ook het

ODM kunnen ontsluiten. Waardoor ze hun klanten dus in staat stellen om snel, makkelijk en veilig geverifieerde gegevens door te geven.

4 Wat zijn jouw belangrijkste drie tips voor organisaties die overwegen om een traject met SIVI AFS te starten?

- Reken eens uit hoeveel moeite het verkrijgen van geverifieerde gegevens van jouw klanten nu kost; voor jezelf en voor je klant. Zet dat af tegen de kosten van Ockto. Is er een business case?
- Bedenk hoe jouw (toekomstige) klant zijn/haar gegevens zou willen delen. Nu en over twee jaar. Is er noodzaak om deze gegevens digitaal te gaan ontsluiten?
- Bedenk dat productleveranciers hun acceptatieprocessen steeds verder automatiseren en daarbij vertrouwen op het verkrijgen van gestructureerde en geverifieerde gegevens. Zijn dat ook de leveranciers waar jij zaken mee doet?

5 Heb je nog een tip voor SIVI?

Laten we snel zorgen voor een mooie case die we kunnen delen, waarmee we de kracht van AFS i.c.m. Ockto aantonen.



Vooruitblik

Irene Deen-Tai

Vanuit SIVI bieden we steeds meer en betere ondersteuning bij het ontwikkelen van toepassingen op basis van SIVI AFS. Met de lancering in maart van de online tool *AFD Online Samenstellen (AOS)* voor AFD 2.0 hebben we een belangrijke nieuwe stap gezet. Ook op andere vlakken maken we flinke vorderingen. In deze vooruitblik geven we meer details over waar we staan en wat we de komende periode gaan doen.

AOS

Met de online tool AOS voor AFD 2.0 is het mogelijk AFD-definities op te stellen, met JSON Schema en JSONPath als output. Analisten kunnen zonder detailkennis van JSON Schema en JSONPath specificaties uitwerken. In ons artikel over AOS op pagina 8 vind je hier meer informatie over. De tooling is op dit moment al beschikbaar voor de domeinen Reis, Motorrijtuigen en Inkomen. De komende periode zullen we het aantal domeinen verder uitbreiden. Ook nieuwe AFD 2.0-features als `customAttribute` zullen we dit jaar in AOS implementeren.

Mappings

Omdat je AFD 2.0 zeer goed kan inzetten voor een eenduidige registratie van gegevens, is het belangrijk om duidelijk te maken hoe je andere datamodellen kunt mappen naar het AFD. Inmiddels hebben we ook concrete vragen hier over. We zijn begonnen met mappings vanuit o.a. Ockto, VNAB en HDN. De mapping van Ockto naar SIVI AFS is gereed en nu beschikbaar voor geïnteresseerden, de API voor deze mapping volgt rond de zomer.

Omdat het voor ons niet haalbaar is een mapping uit te voeren voor alle mogelijke datamodellen, stellen we nu richtlijnen op hoe zo'n mapping uit te werken. Zo kunnen partijen ook zelfstandig mapping naar SIVI AFS ontwikkelen. Natuurlijk levert SIVI ondersteuning bij de uitwerking hiervan.

Ondertussen is ook het ontwikkeltraject gestart voor het converteren van AFD 1.0 (XML-berichten) naar AFD 2.0 (JSON of XML) en andersom. In samenwerking met ABN AMRO zijn we daarnaast begonnen met de eerste mappings van ADN-berichten (EDIFACT) naar AFD 2.0.

Zodra mappings en richtlijnen voor mapping beschikbaar zijn, zullen we dit via de SIVI-nieuwsbrief laten weten. Natuurlijk kun je voor vragen over deze onderwerpen altijd bij ons terecht.

Eigen attributen en tabellen

De komende periode werken we de `afsElement`-entiteit en `afsTable`-entiteit uit. Deze entiteiten geven de gebruiker binnen SIVI AFS de mogelijkheid om in specifieke gevallen eigen attributen en tabellen te definiëren. Dit is een belangrijke nieuwe AFD-functionaliteit, die maakt dat je ook bij echte uitzonderingen nog steeds van het AFD gebruik kunt blijven maken. Een eerste toelichting hiervan vind je in het artikel over `customAttributes` op pagina 4.

SIVI AFS-handboek

Gestaag werken we verder aan het opnemen van oude SIVI-handleidingen binnen het online SIVI AFS-handboek. In de komende periode werken we aan het handboek 'AFD-berichten'. De AFD-berichten zijn onderverdeeld in de berichtsoorten contractbericht, pakketbericht, relatiebericht, batchbericht, groepbericht,

schadebericht en schadefactuur. De algemene en specifieke spelregels die gelden voor de AFD-berichtsoorten gaan we overbrengen naar het SIVI AFS-handboek.

SIVI API-raamwerk

Gebruikers hebben ons laten weten dat ze, naast de geboden functies in het SIVI API-raamwerk die meer gericht zijn op het proces, ook graag gebruik willen maken van APIs rond kern-entiteiten van het AFD. De eerste invulling hiervan is het definiëren van functies rond de object-entiteit, zoals het ophalen van objectdata op basis van specifieke sleutels. Hiervoor is in het AFD de `objectStructure` toegevoegd en zijn API-endpoints voor de bijbehorende functies uitgewerkt. De volgende stap in dit proces is het definiëren van functies voor de party-entiteit.

Onze doelstelling blijft jullie optimaal te ondersteunen in het werken met SIVI AFS en ontwikkelen van toepassingen. Feedback is daarvoor erg belangrijk. We zijn heel benieuwd naar jullie reacties op de SIVI AFS-releases. Voor tips om SIVI AFS nóg beter en toegankelijker te maken, zijn we bereikbaar via support@SIVI.org of via **030-6988096.**



Feedbackcorner

1 Geen punt

Bij SIVI zijn we gewend zorgvuldig met taal om te gaan. Een van de dingen waar op we letten in teksten is de interpunctie: staan er komma's en punten op de plaatsen waar ze horen? Een zin netjes eindigen met een punt is een tweede natuur. Uit de reacties die we kregen op de AFD 2.0-datacatalogus van AFS bleek echter dat die tweede natuur ons ook parten kan spelen. De nieuwe Engelse attribuutnamen waren een aantal keren afgesloten met een punt, terwijl dat niet de bedoeling is. Gemeld werden onder andere `amountOccupationalHealthService.` en `categoryDescription.` De gemelde gevallen zijn gecorrigeerd en het complete bestand is hier nog een keer op gecontroleerd.

Voor correctie:

```
{
  "financialTransaction": {
    "entityType": "booking",
    "amountOccupationalHealthService.": 78.95
  },
  "object": {
    "entityType": "parts",
    "categoryDescription.": "losse onderdelen"
  }
}
```

Na correctie:

```
{
  "financialTransaction": {
    "entityType": "booking",
    "amountOccupationalHealthService": 78.95
  },
  "object": {
    "entityType": "parts",
    "categoryDescription": "losse onderdelen"
  }
}
```

2 Nieuw! Datacatalogus nu met korting

Soms leiden aanvragen voor uitbreiding tot nieuw inzichten. De datacatalogus van SIVI AFS kent bijvoorbeeld een aantal attributen die betrekking hebben op kortingen en toeslagen. In de meeste gevallen gaat het om het verhogen of verlagen van de premie als gevolg van het in- of uitsluiten van (delen van) het risico. Maar er is ook een mogelijkheid om korting te geven, bijvoorbeeld voor meerdere producten (pakketpolis) of groepen (collectiviteit), of in het kader van een promotionele actie.

Naar aanleiding van een aanvraag voor extra attributen op kortingsgebied bleek de datacatalogus geen vaste set van kortingsattributen te bevatten. Ook de mogelijkheid om korting-gerelateerde attributen te groeperen per kortingssoort ontbrak. Daarom introduceren we in AFD 2.0 de `discountOrSurcharge`-entiteit. Hierin is de soort korting of toeslag vast te leggen op basis van een codelijst. Ook kan (onder meer) een begin- en einddatum voor een bepaalde korting worden opgegeven. Meer hierover is te vinden in het AFD 2.0 online raadplegen.

Online raadplegen – AFD 2.0

Versie 1 mei 2021 – Neem [contact](#) op als je een label niet kunt vinden of een vraag hebt. Bekijk de [korte toelichting](#) bij de zoekfunctie en zoektips.

Entities	Attributes	Cir	discountOrSurcharge	i
Entity	Description			
discountOrSurcharge	Information about a discount or surcharge. (NL: Informatie over een korting of toeslag.)			

3 Gecorrigeerde attribuutnamen binnen AFD 2.0

In de voorbije maanden heeft SIVI de attribuutnamen grondig gecheckt op consistentie. Daarbij zijn de namen van 2000 attributen beter in lijn gebracht met de uitgangspunten van SIVI AFS.

Een gebruiker gaf bijvoorbeeld aan dat de attributen *surcharge1Calculated*, *surcharge2Calculated* en *surcharge3Calculated* niet juist waren gedefinieerd. Deze losse attributen zijn in de nieuwste versie van AFD 2.0 omgezet naar één array-attribuut (*surchargesCalculated*).

Voor correctie:

```
{
  "underwriting": {
    "entityType": "valuePreventionMeter",
    "surchargeAmount": [231.21, 98.88, 25],
    "surcharge1Calculated": true,
    "surcharge2Calculated": false,
    "surcharge3Calculated": false
  }
}
```

Na correctie:

```
{
  "underwriting": {
    "entityType": "valuePreventionMeter",
    "surchargeAmount": [231.21, 98.88, 25],
    "surchargesCalculated": [true, false, false]
  }
}
```

Meer details over de gecorrigeerde naamgevingen vind je in de release notes op de landingspagina van SIVI AFS.

Nieuwtjes

- Vanaf 1 maart 2021 is het SIVI-team versterkt met een nieuwe collega: John Boots. John gaat zich binnen SIVI AFS richten op het SIVI AFS API-raamwerk en alle technische vraagstukken.
 - Voor AFD 2.0 zijn ruim 31.000 woorden in 8046 attributenamen door elkaar gehusseld en 'ver-camelCaset', aan elkaar geplakt en weer gesplitst, gecontroleerd en her-gecontroleerd. De maximale attribuutlengte is 50 tekens, de kortste 2 (id) en de gemiddelde 25.
 - OpenApi 3.1 werd afgelopen maart geïntroduceerd. SIVI analyseert de nieuwe mogelijkheden en de impact daarvan op de reeds gedefinieerde functies binnen het SIVI AFS API-raamwerk. SIVI houdt de ontwikkelingen in de markt nauwlettend in de gaten en zorgt zo voor aansluiting op de voortdurend evoluerende markt.
 - Het aantal SIVI AFS-baselines in AOS is verder uitgebreid. Met de toevoeging van de masterAgreementStructure-baseline in het domein Motorrijtuigen staat het totaal aantal baselines dat gebruikmaakt van AFD 2.0 nu op vier.
- Sinds 1 januari 2021 is de SIVI AFS-overzichtspagina druk bezocht door geïnteresseerden en gebruikers. Hieronder de getallen op een rijtje:
 - 1.077 bezoekers bezochten de overzichtspagina;
 - 1.070 bezoekers bekeken AFD 2.0 Online Raadplegen;
 - 1.967 bezoekers raadpleegden het SIVI AFS Online Handboek;
 - 234 bezoekers downloadden één van de beschikbare downloads:
 - Paper - Toelichting SIVI AFS
 - AFD 2.0 Master table
 - Mappingtable AFD 1.0 - AFD 2.0
 - Datacatalogus
 - AFS Release Note
- De afgelopen 15 SIVI AFS-webinars zijn bezocht door 160 deelnemers.
 - Een kristallen jubileum: op dinsdag 8 juni vond alweer het 15e SIVI AFS-webinar plaats. In deze webinars introduceert managing consultant Herman Lenferink in vogelvlucht van een uur het hele SIVI AFS. Inschrijven voor volgende edities kan via [link](#).



Wil jij een bijdrage leveren aan het SIVI AFS Magazine?

Werk jij met SIVI AFS en wil je jouw inzichten rondom het gebruik van SIVI AFS delen, dan maken we graag plaats in SIVI AFS Magazine. Neem contact op met Herman Lenferink via herman.lenferink@sivi.org om dit verder af te stemmen.