DDAS API Koppelvlakspecificatie

Samenvatting

Dit document bevat een uitleg van de toepassing van Respec documentatie binnen VNG Realisatie.

- In hoofdstuk 1 wordt de werking van het template uitgelegd. Hoe je het kunt gebruiken voor je eigen specificatie, wat er precies gebeurd als je er mee aan de gang gaat en waar je daarbij op moet letten.
- Hoofdstuk 2 beschrijft hoe je binnen Imvertor (indien toegepast) een deel van de content van je specificatie kan genereren. Dit hoofdstuk is optioneel aangezien je Respec document niet per definitie betrekking hoeft te hebben op een Informatiemodel.
- Hoofdstuk 3 beschrijft waar je op moet letten als je het GitHub Respec template kopieert (zowel als admin als als gebruiker), hoe je de Respec documentatie, evt. naast de door Imvertor gegenereerde content, nog kan voorzien van andere content en hoe je door het aanpassen van configuration properties de stijl van het document kan veranderen daarbij rekening houdend met wat wij binnen VNG Realisatie met elkaar hebben afgesproken. In de flowchart van hoofdstuk 5 wordt dit hoofdstuk met de 'Generatie tak' geïllustreerd.
- Hoe de gegenereerde Respec bestanden gepubliceerd kunnen worden wordt in Hoofdstuk 4 beschreven. In de flowchart van hoofdstuk 5 wordt dit hoofdstuk met de 'Publicatie tak' geïllustreerd.
- In hoofdstuk 5 is een flowchart van het gehele proces voor het vervaardigen van Respec documentatie uitgewerkt. Dit is echter tevens een voorbeeld van het gebruik van Mermaid, een manier om flowcharts te vervaardigen.
- De hoofdstukken volgend op hoofdstuk 5 hebben slechts een illustratieve functie. Op basis daarvan moet je, samen met de uitleg in hoofdstuk 3, in staat zijn de getoonde functionaliteit te gebruiken.

Inhoudsopgave

1. ReSpec template instructies

ReSpec is een tool om html en pdf documenten te genereren op basis van markdown en html content.

Organisatie administrators dienen de knop *Use this template* te gebruiken om een kopie van de template repository aan te maken. Deze kan daarna door jouzelf aangepast en uitgebreid worden. Dit template is afgeleid van het Logius Respec template maar is op enkele details aangepast:

- Het organisation configuration bestand is in een aparte repository ondergebracht;
- Het lokale configuratie bestand is aangepast aan de behoeftes van VNG Realisatie;
- In de 'index.html' is een style element aangebracht waarmee de standaard instelling van de 'max-width' property op 'none' is gesteld;
- De uitleg van de toepassing van het VNG-R Respec profiel is als inhoud aan deze repository toegevoegd.

De dynamische pagina van het template document is hier te zien.

Deze repository bevat ook de GitHub Workflows om een statische HTML-pagina en PDF-document te genereren en enkele controles uit te voeren. Deze workflows worden automatisch gerund zodra er een aanpassing gedaan wordt aan de main branch. Een beschrijving van deze acties vind je onderaan dit hoofdstuk.

1.1 Vereiste voor gebruik

- Kennis van git/GitHub
- Kennis van Markdown en/of HTML
- Een webserver om de documentatie te hosten

Voor de laatste gebruiken wij GitHub. Kennis van de vorm van een Javascript object is handig om de in dit template voorkomende scripts aan te kunnen passen maar zonder die kennis kom je m.b.v. deze documentatie ook al heel ver.

1.2 Gebruikersinstructie

Om het gebruik van dit template makkelijker te maken raden we het aan om een IDE te gebruiken. Die geeft een voorbeeld van hoe de markdown eruit zal zien, kan laten zien of de config files nog

in de correcte vorm zijn en kan helpen in het gebruik van git.
Een gratis voorbeeld van een IDE is: Visual studio code. Een combinatie van GitHub desktop en je eigen favoriete Markdown editor is echter ook mogelijk.

Aanpassingen maken aan het document gaat op 2 manieren:

- De configuratie van het document aanpassing in de config files
- Markdown of html files toevoegen/veranderen

De **configuratie files** bevatten informatie over de organisatie en over de status van het document. Helemaal onderaan hoofdstuk 3 vind je meer informatie over de configuratie opties, daarnaast kun je ook de Logius ReSpec wiki bezoeken. De files zijn gesplitst in 2 files die weer in 2 verschillende repositories zijn ondergebracht: organisation-config.js en config.js.

De organisation_config (organisation-config.js) bevat configuratie properties die betrekking hebben op alle VNG-R Respec documentatie, de properties in deze file zullen zelden veranderen zoals bijv. de naam van de organisatie.

De document_config (config.js) bevat configuratie properties die alleen relevant is voor het betreffende Respec document en hoort dan ook in elke Respec renderende repository thuis.

Beide configuratie bestanden worden gelinkt in de `index.html` file waardoor ze beide bij het renderen van de Respec documentatie automatisch worden samengevoegd. Daardoor zijn de organisatie specifieke configuraties over alle Respec documentatie van VNG-R gelijk en hoeft deze niet steeds gekopieerd te worden. Op deze wijze zorgen we er voor dat alle VNG-R Respec documenten zo eenduidig mogelijk zijn en blijven.

In het volgende hoofdstuk staat beschreven hoe je de inhoud van het Respec document naar wens kunt aanpassen.

§ 1.3 Rendering, automatische controles en publicatie

Het bestand '.github/workflows/build.yml' bevat een action script waarmee automatisch een drietal acties worden uitgevoerd nadat een bestand in de repository wordt gewijzigd, toegevoegd of verwijderd:

- het renderen van het Respec document;
- het checken of de gerenderde Respec html wel correct is en voldoet aan de toegankelijkheidseisen;
- het publiceren van de gegenereerde statische html en pdf naar een centrale Respec publicatie repository.

We beschrijven de eerste 2 acties in het kort hieronder. Aangezien we de laatste actie nog niet werkende hebben wordt deze voorlopig nog handmatig uitgevoerd, dat beschrijven we in hoofdstuk 4. De log van deze acties is te vinden in het tabblad `Actions` in de GitHub repository.

1.3.1 Rendering

Deze actie start het renderen van de Respec html. Vervolgens wordt er op basis daarvan een statische html en een pdf bestand gegenereerd. Die gebruiken we uiteindelijk om te publiceren.

De PDF-versie wordt aangemaakt omdat de property `alternateFormats` in de organisation_config als volgt geconfigureerd staat:

```
alternateFormats: [
  {
      label: "pdf",
      uri: "snapshot.pdf",
  },
]
```

Er moet nog worden bepaald of we de document eigenaar zelf willen laten bepalen of hij/zij een pdf wil genereren. Indien we dat willen verhuisd deze property naar de document_config. De waarde van de property `uri` kan dan naar wens worden aangepast.

1.3.2 Checken

Na het renderen van de Respec html en pdf worden er via github actions 2 controles uitgevoerd op de html:

- een WCAG-check (Web Content Accessibility Guidelines), deze guidelines gemaakt door W3C zorgen voor een verbetering van de toegankelijkheid van webapplicaties verbeterd voor zowel verschillende apparaten als voor mensen met een beperking. Ook wordt de validiteit van het HTML bestand gecheckt, bijv.:
    - of er geen `<section>` elementen met 'id' attributen voorkomen die al voorkomen in het bestand;
    - of er geen `<a>` elementen voorkomen met 'href' attributen die verwijzen naar `<section>` elementen die helemaal niet bestaan.

Deze check moet eerst succesvol uitgevoerd zijn voordat wordt begonnen aan de volgende check. In de 'Action' die start met het woord 'Update' (zie het `Actions` tabblad) kun je in de actie 'Check/WCAG' de step 'Run pa11y snapshot.html' vinden. Daar kun je zien welke fouten geconstateerd zijn.

- een link-check, deze check controleert of alle links die in het document staan ook bestaan. Het gaat dan bijv. om de links die worden vermeldt in:
  - 'Deze versie'
  - 'Laatst gepubliceerde versie'
  - 'Laatste werkversie'
  - 'Vorige versie'

  Deze links verwijzen naar specifieke locaties in de GitHub Pages interface van de 'publicatie' GitHub repository (zie de volgende subparagraaf voor meer uitleg). Om deze check goed te kunnen doorstaan moeten deze locaties dus al bestaan in die interface. Indien dat nog niet gedaan is moet daar de folder voor het nieuwe versienummer van de Respec documentatie al worden aangemaakt. Plaats in die folder dan ook een tijdelijk 'index.html' bestand. De inhoud van dat bestand is (nog) niet van belang.

> **LET OP!**
> Onderstaand tekst is slechts een voorstel. De definitieve url kan indien gewenst nog andere onderdelen bevatten zoals bijv. `publishDate`, `previousPublishDate`, `specStatus` en `previousMaturity`.

Bij het genereren van de links zijn op dit moment de volgende configuration properties van belang:

- nl_organisationPublishURL
  De basis url van de GitHub Pages interface van de 'publicatie' GitHub repository, op dit moment: `https://vng-realisatie.github.io/publicatie`. Deze is gedefinieerd in de organisation_config aangezien deze altijd gelijk blijft.
- pubDomain
  Het publicatie domein. Aangezien we vooralsnog slechts voor Conceptuele Modellen Respec documentatie genereren heeft deze de waarde `cim` en staat deze gedefinieerd in de organisation_config. Zo nodig kan deze overruled worden in de document_config. Vergeet in dat geval niet om ook de structuur in de 'publicatie' GitHub repository uit te breiden. Wordt gebruikt in 'latestVersion', 'thisVersion' en 'prevVersion'.
- specStatus
- latestVersion
  Wordt opgebouwd a.d.h.v. een aantal andere configuratie properties uit zowel de

organisation_config als de document_config en enkele vaste karakters. Deze is gedefinieerd in de organisation_config aangezien deze altijd gelijk blijft.

- thisVersion
  Wordt opgebouwd a.d.h.v. een aantal andere configuratie properties uit zowel de organisation_config als de document_config en enkele vaste karakters. Deze is gedefinieerd in de organisation_config aangezien deze altijd gelijk blijft.
- prevVersion
  Wordt opgebouwd a.d.h.v. een aantal andere configuratie properties uit zowel de organisation_config als de document_config en enkele vaste karakters. Deze is gedefinieerd in de organisation_config aangezien deze altijd gelijk blijft.
- shortName
  De project-mnemonic, een afkorting van het project. Zo wordt het project 'Open Raadsinformatie' wordt bijv. afgekort als 'ori'. Deze is gedefinieerd in de document_config aangezien deze natuurlijk afhankelijk is van het te genereren Respec document. Wordt gebruikt in 'latestVersion', 'thisVersion' en 'prevVersion'.
- publishVersion
  De versie van het te publiceren Respec document. Komt overeen met de Tagged Value 'Version' in het Enterprise Architect bestand van het model en heeft een waarde dat voldoet aan het formaat `x.x.x`, bijv. `2.0.0`. Deze is gedefinieerd in de document_config aangezien deze natuurlijk afhankelijk is van het te genereren Respec document. Wordt gebruikt in de titel van het Respec document maar ook in 'thisVersion'.
- previousVersion
  De voorgaande versie van het te publiceren Respec document. Komt overeen met de Tagged Value 'Version' in het Enterprise Architect bestand van het voorgaande versie van het model en heeft een waarde dat voldoet aan het formaat `x.x.x`, bijv. `2.0.0`. Deze is gedefinieerd in de document_config aangezien deze natuurlijk afhankelijk is van het te genereren Respec document. Wordt gebruikt in 'prevVersion'.

Het consistent en nauwgezet invullen van de configuratie properties in de document_config is essentieel voor een goede werking van de links.

De bovenstaande properties hebben invloed op de wijze waarop het eerste deel van de Respec documentatie wordt gegenereerd. Hieronder sommen we de regels op. Indien wordt besloten de properties 'latestVersion', 'thisVersion' en 'prevVersion' een andere inhoud te geven dan zullen onderstaande regels herzien moeten worden.

- Als de parameter 'specStatus' de waarde 'WV' heeft dan wordt de waarde van de parameter 'thisVersion' niet gebruikt voor het bepalen van 'Deze versie' maar wordt daar dezelfde waarde neergezet als bij 'Laatste werkversie'.
- Als de parameter 'specStatus' de waarde 'WV' heeft dan wordt de waarde van 'Subtitel 2' niet gebaseerd op de parameter 'publishDate' maar op de datum waarop de Respec documentatie

door GitHub wordt gegenereerd.

- Als de parameter 'specStatus' de waarde 'WV' heeft dan wordt het versienummer niet in de titel van het document opgenomen.
- Als de parameter 'previousPublishVersion' niet bestaat dan kan 'prevVersion' niet bepaald worden en wordt 'Vorige versie' niet gegenereerd.
- Als de parameter 'publishVersion' niet bestaat dan kan 'thisVersion' niet bepaald worden en wordt 'Deze versie' niet gegenereerd.

§ 2. Uitgangpunten

Het koppelvlak moet voldoen aan de volgende wetten, afspraken en standaarden:

- NORA
- BIO
- Digikoppeling – REST-API profiel
- Nederlandse API strategie
- NL Gov REST-API Design Rules
- Algemene verordening gegevensbescherming
- Wet op het Centraal Bureau voor de Statistiek
- FSC

De volgende keuzes zijn gemaakt:

- Gebruik Digikoppeling REST-API standaard
- JSON formaat voor berichten
- Gebruik Diginetwerk voor transport
- Gebruik PKIo certificaten
- Gebruik JWS voor signen
- Gebruik JWE voor encryptie
- Beveiligingsniveau BBN2 (zou uit DPIA moeten volgen)

3. Beheer van de specificatie

De koppelvlakspecificatie is onderhevig aan wijzigingen: de technologie ontwikkelt zich, er zijn mogelijk andere gegevens nodig, de samenwerking tussen de betrokken partijen kan wijzigen, etc. Om deze wijzigingen op een betrouwbare en juiste manier te verwerken in de specificatie, is een wijzigingsproces ingericht. Dit wijzigingsproces valt onder verantwoordelijkheid van de stuurgroep DDAS en wordt uitgevoerd door het programma DDAS, zolang het programma DDAS actief is. Daarna wordt het overgedragen aan een nog aan te wijzen organisatie. Voor het beoordelen van wijzigingsverzoeken wordt een beheeroverleg samengesteld, met afgevaardigden van de betrokken partijen, onder voorzitterschap van het programma DDAS. Dit beheeroverleg komt periodiek bijeen om wijzigingsverzoeken te beoordelen en eventueel verder uit te werken.

Het streven is om maximaal eenmaal per jaar een nieuw release van de koppelvlakspecificatie uit te brengen.

3.1 Indienen wijzigingsverzoek

Wijzigingsverzoeken worden verzameld via [nog in te vullen]. Alle betrokken partijen mogen wijzigingsverzoeken indienen. Er is geen template voor het indienen van een wijzigingsverzoek, maar het verzoek moet in elk geval de volgende informatie bevatten:

- Indiener (inclusief contactgegevens)
- Datum indienen
- Beschrijving gewenste wijziging (bondig, maar voldoende specifiek om in te kunnen schatten wat de impact is)
- Onderbouwing/ aanleiding gewenste wijziging
- Prioriteit volgens de indiener (hoe snel moet de wijziging doorgevoerd worden)

Als het aantal wijzigingsverzoeken groot wordt of de afhandeling daarvan complex, dan wordt een systeem gebruikt om een en ander in te administreren.

Dit systeem moet zo openbaar mogelijk zijn, om zo transparant mogelijk te zijn over de afhandeling van verzoeken en om te voorkomen dat dezelfde wijzigingsverzoeken meerdere malen ingediend worden.

§ 3.2 Afhandelen wijzigingsverzoek

Het wijzigingsverzoek wordt door het programma DDAS geanalyseerd, waarbij vastgesteld wordt welke onderdelen van de specificatie geraakt worden en wat de geschatte impact is op de specificatie, de techniek, de processen en de betrokken partijen. Tevens wordt ingeschat wat de randvoorwaarden, kosten en doorlooptijd van de gewenste wijziging zouden zijn. Dit leidt tot een voorstel voor de verdere afhandeling: of, hoe en wanneer dit wijzigingsverzoek doorgevoerd wordt.

Het wijzigingsverzoek met de analyse van het programma DDAS worden besproken in het (nog in te richten) beheeroverleg DDAS. Als alle betrokken partijen akkoord gaan met de voorgestelde afhandeling, wordt deze afhandeling gevolgd (d.w.z. inplannen voor een release, via noodprocedure eerder doorvoeren, of afwijzen van het verzoek).

§ 3.3 Releaseproces

Wijzigingen die doorgevoerd moeten worden, worden zoveel mogelijk via een release in productie gebracht. Het streven is om maximaal eenmaal per jaar een release door te voeren. De stappen die hiervoor doorlopen worden zijn:

- Vaststellen scope van de release door de stuurgroep DDAS, op basis van advies van beheeroverleg [6 maanden voor productiedatum]
- Publiceren aangepaste specificatie door programma DDAS [5 maanden voor productiedatum]
- Doorvoeren noodzakelijke wijzigingen in de testomgeving door deelnemers (gegevensleveranciers en CBS) [tot 1 maand voor productiedatum]
- Testen nieuwe release in testomgeving [in laatste maand voor productiedatum]
- Livegang nieuwe release

§ 3.4 Releasenummering

Ieder release wordt aangeduid met een releasenummer. Deze krijgt de vorm X.Y, waarbij X het "major" nummer is en Y het "minor" nummer. Voor testreleases kan een derde nummer toegevoegd

worden; het zogenaamde "patch" nummer. In de productieomgeving wordt geen patch nummer gebruikt.

Als een release via het reguliere releaseproces naar productie gaat, dan krijgt deze een nieuw major nummer en het minor nummer 0 (bv. "1.0"). Als er via de noodprocedure een release doorgevoerd wordt, dan blijft het major nummer hetzelfde, maar wordt het minor nummer opgehoogd (bv. "1.1").

§ 3.5 Vrijgaveprocedure en afwijkingsverzoeken

Er is geen "vrijgave" van deelnemers voor een release nodig. Als de specificatie complexer wordt kan de stuurgroep DDAS besluiten om een vrijgaveprocedure in te richten. De deelnemer moet dan aan de hand van een set testscenario's aantonen te voldoen aan de nieuwe specificaties. Als dit succesvol is, dan krijgt de deelnemer vrijgave voor de nieuwe release. Als dit niet succesvol is, dan kan de deelnemer een afwijkingsverzoek indienen bij het programma DDAS en toch gegevens blijven aanbieden. Een afwijkingsverzoek wordt alleen geaccepteerd als dit de rapporten van CBS niet compromitteert. In het afwijkingsverzoek wordt altijd aangegeven hoe lang de afwijking geldig mag blijven.

§ 3.6 Noodprocedure

Het kan gebeuren dat een wijziging niet kan wachten op een gepland release, maar sneller doorgevoerd moet worden. De stuurgoep DDAS kan dan op advies van het beheeroverleg, een noodprocedure aanroepen.

Het beheeroverleg adviseert de stuurgroep welke stappen genomen moeten worden en in welk tempo deze doorlopen moeten worden. Als de stuurgroep hiermee akkoord gaat, voert het programma DDAS de regie op de uitvoering van de stappen.

De release die hiermee ontstaat krijgt geen nieuw "major" versienummer, maar een nieuw "minor" nummer (zie ook "releasenummering").

§ 3.7 Escalatie

Als de partijen het niet eens worden, wordt het verzoek geëscaleerd naar de stuurgroep DDAS. Als het behandelen van het verzoek niet kan wachten tot het eerstvolgende overleg van de stuurgroep, worden de stuurgroepleden schriftelijk om hun oordeel gevraagd.

§ 4. Transportlaag

Hoe ziet de technische uitwisseling van berichten eruit.

Vragen:

- Gebruik van Diginetwerk? Kunnen alle organisaties die gegevens gaan leveren hierop aansluiten? Wordt waarschijnlijk niet mogelijk... Dan via “open” internet: vereist mogelijk extra maatregelen, zoals versleuteling van de gegevens. Dit hangt af van de DPIA.
- Dubbelzijdig TLS (wordt voorgeschreven in Digikoppeling en FSC standaard) NB: Dit vereist een certificaat dat door alle betrokken partijen vertrouwd wordt.
- Gebruik van PKIo certificaten voor authenticatie op basis van het Nederlandse profiel van OAuth? Het is de vraag of alle partijen een PKIo certificaat mogen aanvragen. Als dit niet mogelijk is, moet een “trust anchor” gevonden worden: de autoriteit die certificaten kan uitgeven, die door alle betrokken partijen vertrouwd worden.

§ 4.1 Identificatie, Authenticatie en Autorisatie

Hoe worden de schuldhulpverleners (gegevensleveranciers) geïdentificeerd? (o.b.v. (sub)OIN?) Als niet alle betrokken partijen een (sub)OIN kunnen krijgen, moet een systematiek gevonden worden om alle partijen uniek te kunnen identificeren.

Hoe worden systemen geauthenticeerd? (obv PKIo certificaten? Als dat niet kan: wie wordt de “Trust Anchor” – de autoriteit die door alle partijen vertrouwd wordt?)

Autorisatie lijkt niet heel spannend: er zal waarschijnlijk maar één service komen met een vaste set gegevens, waar maar één partij (CBS) toegang toe zal krijgen. Als fijnmaziger autorisatie nodig is, dan bestaat er een voorkeur voor PBAC (Policy Base Authorisation Control). De autorisatie wordt dan bepaald op basis van beleidsregels, zoals “organisatie X krijgt toegang tot gegeven G als de organisatie overeenkomst O getekend heeft en het gegeven is vrijgegeven door autoriteit A”. Dan is de vraag wie deze beleidsregels vaststelt en wie ze beheert.

4.2 Signing en Versleuteling

NB: De Digikoppeling standaard voor REST-API heeft (nog) geen standaard voor signing en encryptie vastgesteld. Daarom voorstel om JWT te gebruiken en dus eerst een JWT aan te vragen, die daarna bij het request wordt meegestuurd. Eventueel kunnen hierbij protocollen van de FSC standaard toegepast worden.

4.2.1 Signing

Voorstel: Signing op basis van JWS in een JWT conform de FSC standaard.

4.2.2 Versleuteling (Encryptie)

Is versleuteling nodig? (zou uit DPIA moeten komen – ik vermoed dat het nodig is)

Voorstel: Versleuteling op basis van JWE in een JWT met PKIo certificaten. NB: ook hier geldt dat als niet alle betrokken partijen PKIo certificaten kunnen aanvragen, er een Trust Anchor nodig is die vertrouwde certificaten kan uitgeven.

4.3 Vraagbericht (request)

Request zoals dat door CBS naar de schuldhulpverlener gestuurd wordt. Alleen een GET en/ of POST request: alleen opvragen gegevens, geen mutaties. Bij GET zitten de parameters in de URL, waardoor mogelijk cache gegevens gebruikt worden, als de parameters niet wijzigen.

NB: Met een JWT voor authenticatie, signen en versleutelen (als versleutelen nodig is – met een BSN in het bericht zou dat waarschijnlijk moeten). Mogelijk kan dit afgevangen worden door het FSC-concept van inward- en outward-services?

Voorstel voor parameters die meegestuurd kunnen worden (allemaal optioneel):

- Startdatum (default vandaag)
- Einddatum (default vandaag)

- Gemeente (default alle – alleen relevant als over meer dan 1 gemeente gegevens aangeleverd worden)
- BSN? (of ander gegeven waarmee een inwoner geïdentificeerd kan worden – default alle)
- SHV-traject? (default alle)

Technische uitwerking in OAS3 (YAML/ JSON bestand op Github?)

4.4 Antwoordbericht (response)

Response van de schuldhulpverlener met de gewenste gegevens in JSON formaat.

Als versleutelen nodig is, in een JWE vorm (eventueel gezipt, versleuteld in een token).

Payload zoals gedefinieerd door Arjen!

Responses: opnemen in OAS3 beschrijving. Bv:

- 200: bericht goed verwerkt (met payload)
- Welke foutberichten? (FSC standaard volgen?)

4.5 Niet functionele eisen

4.5.1 Beschikbaarheid

Niet kritische toepassing: geen hoge beschikbaarheid vereist.

Afstemmen met CBS: wanneer willen zij gegevens verzamelen? Dan zou de beschikbaarheid wat hoger moeten zijn. BV: tijdens kantooruren

4.5.2 Performance

Geen afhankelijkheden in het primaire proces: geen hoge performance vereist.

Wordt gebruik van cache toegestaan (volgens mij moet dat kunnen)? Onder welke voorwaarden?

4.5.3 Monitoring

Verantwoordelijkheid voor monitoring ligt bij partij die verantwoording hierover moet afleggen. Welke verantwoording verwacht het programma of SZW?

Voor gemeenten (suggestie):

- Aantal bevragingen naar datum en afzender (altijd CBS?)
- Aantal en soort foute bevragingen
- Aantal en soort meegestuurde parameters

Voor CBS:

- Aantal bevragingen naar datum en schuldhulpverlener
- Aantal en soort responses

5. Niet functionele eisen

5.1 Beschikbaarheid

Niet kritische toepassing: geen hoge beschikbaarheid vereist.

Afstemmen met CBS: wanneer willen zij gegevens verzamelen? Dan zou de beschikbaarheid wat hoger moeten zijn. BV: tijdens kantooruren

5.2 Performance

Geen afhankelijkheden in het primaire proces: geen hoge performance vereist.

Wordt gebruik van cache toegestaan (volgens mij moet dat kunnen)? Onder welke voorwaarden?

5.3 Monitoring

Verantwoordelijkheid voor monitoring ligt bij partij die verantwoording hierover moet afleggen. Welke verantwoording verwacht het programma of SZW?

Voor gemeenten (suggestie):

- Aantal bevragingen naar datum en afzender (altijd CBS?)
- Aantal en soort foute bevragingen
- Aantal en soort meegestuurde parameters

Voor CBS:

- Aantal bevragingen naar datum en schuldhulpverlener
- Aantal en soort responses

6. Aanleverprotocol

Stappen bij het aanleveren van gegevens:

- CBS roept systeem van schuldhulpverlener (gegevensleverancier) aan om JWT voor signing en encryptie te krijgen
- CBS roept API van de gegevensleverancier aan (eventueel met parameters)
- CBS controleert response technisch (signing, viruscontrole, berichtformaat)
- CBS controleert response functioneel (verplichte velden, vreemde waarden, etc.)
- CBS stuurt een verwerkingsverslag naar de gegevensleverancier
- Indien OK, dan worden de gegevens ingelezen in de database

- CBS loopt alle gerapporteerde trajecten af en combineert trajecten van dezelfde BSN tot één “traject”
- Bij het combineren wordt de volledigheid en kwaliteit van de gegevens gecontroleerd – op basis daarvan krijgt het traject een "betrouwbaarheidsindicator""
- CBS genereert de gewenste rapporten

NB: Als er bij deze stappen algoritmen gebruikt worden, moeten deze voldoen aan de Europese AI-verordening (definitieve tekst nog niet gevonden) en aangemeld worden bij het Algoritmeregister van de Nederlandse overheid.

§ 7. Aansluitprotocol

Iedere schuldhulpverleningsorganisatie of gemeente (hierna: "deelnemer") die gegevens beschikbaar gaat stellen aan CBS, moet het aansluitprotocol doorlopen. Dit protocol valt onder verantwoordelijkheid van het programma DDAS. Voor vragen hierover kan altijd contact opgenomen worden met [contactadres].

Het protocol kan aangepast worden als hiervoor aanleiding is. Na aanpassingen wordt de meest recente versie met versienummer en versiedatum gepubliceerd op [documentatiewebsite].

De stappen die de deelnemer moet doorlopen, zijn:

- De deelnemer meldt zich bij de stelselbeheerder (CBS of DDAS?) via het aanmeldformulier [waar staat dit? wie beheert dit?], waarin in elk geval het volgende ingevuld:
    - Naam van de deelnemer + contactgegevens
    - Naam van de gegevensleverancier + contactgegevens
    - Endpoint waar de productiegegevens beschikbaar komen
    - Endpoint waar de testgegevens beschikbaar komen
    - Akkoord met de aansluitvoorwaarden, waaronder de verwerkersovereenkomst met CBS
    - Eventuele verzoeken om de aansluiting tot stand te krijgen, zoals een gewenste publicatiedatum, specifieke testdata of specifieke beschikbaarheid
- *Indien PKIo certificaten niet mogelijk zijn: de stelselbeheerder (DDAS of CBS?) genereert een certificaat voor de TLS verbinding, signing en encryptie, en levert deze aan de deelnemer.*

- De deelnemer richt in de testomgeving de API, conform de AOS documentatie [yaml-bestand, nog toe te voegen] in. Voor de installatie van FSC komt een handleiding en een referentie implementatie beschikbaar.
- De deelnemer voert CBS op in de management module van FSC, om toegang te verlenen.
- CBS voert enkele bevragingen uit in de testomgeving en beoordeelt de kwaliteit van de gegevens. Op basis van de bevindingen wordt de API aangepast.
- Indien er geen blokkerende bevindingen zijn, krijgt de deelnemer vrijgave van de stelselbeheer (DDAS of CBS?) en wordt de API in de productieomgeving ingericht en beschikbaar gesteld.
-
- CBS voert de deelnemer op in de management module van FSC, zodat de API bevraagd wordt bij het ophalen van alle gegevens.

Ten behoeve van de testen stelt DDAS een set testgegevens beschikbaar [wie maakt deze set? waar komt dit te staan?].

Voor ondersteuning bij de aansluiting is een referentie implementatie en documentatie beschikbaar [waar?] en kan contact opgenomen worden met [contactadres]. Als er bij de aansluiting bevindingen zijn, die niet door de deelnemer opgelost kunnen worden, kan een wijzigingsverzoek ingediend worden.

8. Conformiteit

Naast onderdelen die als niet normatief gemarkeerd zijn, zijn ook alle diagrammen, voorbeelden, en noten in dit document niet normatief. Verder is alles in dit document normatief.

A. Index

A.1 Begrippen gedefinieerd door deze specificatie

§ A.2 Begrippen gedefinieerd door verwijzing