

VERSLAG

Thematische Werkgroep 2

OSLO Voorwaarden Dienstverlening

Donderdag 15 juni 2023

INHOUD

Inhoud	2
Praktische Info	3
Aanwezig	3
Agenda	4
Inleiding	4
Samenvatting Thematische Werkgroep 1	4
Overzicht van de aanpassingen	6
Introductie tot uml: kardinaliteit & attributen	7
Onze aanpak	7
Datamodel	9
Raf gaat sporten met de Vrijtijdspas	9
De bibliotheek van Leuven voorziet gratis studieplaatsen	11
Emma en Loes tekenen hun contract om wettelijk samen te wonen	12
De Vlaamse Overheid laat 'inkomen' mee bepalen op korting voor De Lijn	15
Zonder waarde	16
Met waarde	16
Datatype	17
Serialisatie	18
Gebruik van InformatieConcept	21
Voorwaarden in het model: volgende stappen	22
Het datamodel 'OSLO Voorwaarden Dienstverlening'	22
Volgende stappen	23
Volgende thematische werkgroep	23

PRAKTISCHE INFO

Datum: 15/06/2023 (9u - 12u)

Locatie: Virtueel (Microsoft Teams)

AANWEZIGEN

Jef Liekens	Digitaal Vlaanderen
Yaron Dassonneville	Digitaal Vlaanderen
Geert Thijs	Digitaal Vlaanderen
Paul Bogaert	Digitaal Vlaanderen - Centrale redactie
Karl Vogels	Digitaal Vlaanderen - Redacteur Vlaanderen.be, IPDC, 1700
Seppe Gielen	Digitaal Vlaanderen - dev IPDC
Dieter Van Peer	Digitaal Vlaanderen - Architect IOM/IPDC
Tessa De Maesschalk	Digitaal Vlaanderen
Paul Jacxsens	Agentschap Binnenlands Bestuur - Functioneel analyst & PM
Stefanie Kerkhof	Agentschap Binnenlands Bestuur - Programmamanager SDG voor lokale besturen
Geertrui Timmers	Agentschap Binnenlands Bestuur - digiteam, PM LPDC
Boris De Vloed	Agentschap Binnenlands bestuur - extern analyst, dev
Petra van der Laag	Gemeente Schoten
Roos Bekaert	Coördinator dienstverlening Gemeente Evergem
Ann Mathieu	Cronos

AGENDA

09u00 - 09u10	Welkom en agenda
09u10 - 09u15	Samenvatting vorige werkgroep
09u15 - 09u25	Overzicht van aanpassingen
09u25 - 09u40	Definities klassen
09u40 - 09u50	Inleiding kardinaliteiten en attributen
09u50 - 10u00	Pauze
10u00 - 11u30	Overzicht model adhv storylines
11u30 - 11u45	Q&A en volgende stappen

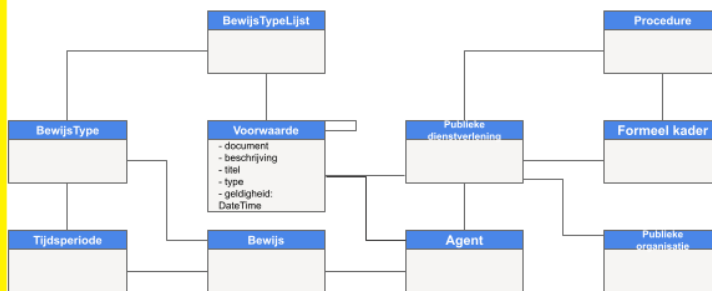
INLEIDING

SAMENVATTING THEMATISCHE WERKGROEP 1

In de eerste thematische werkgroep werden drie items behandeld. Als eerste kwam de introductie tot UML aan bod om aan te tonen hoe een datamodel in elkaar zit. Daarna werd de OSLO aanpak nog eens toegelicht. De use cases staan centraal, hieruit werd ook de scoping bepaald. De relevante datastandaarden kwamen ook aan bod bij de OSLO aanpak. Als laatste werd het sneuvelmodel toegelicht aan de hand van relevante storylines. Alle feedback die aan bod is gekomen werd verwerkt in de voorbereiding van deze thematische werkgroep.

Graag verwijzen we hiervoor naar slide 8.

Topics vorige werkgroep?



UML introductie

- Basisterminologie
 - Unified Modeling Language
 - Concepten
 - Relaties
 - Attribuering
- Voorbeeld asiel

Onze aanpak

- Starten van use cases (in scope <> feature/implementation)
- Bespreken van bestaande standaarden die we gebruiken bij OSLO Voorwaarden Dienstverlening

Sneuvemodel opbouwen adhv verzamelde use cases

- Use case storyline
- Opbouwen en voorstelling sneuvemodel
- Oefening: Laat dit model alle relevante use cases toe?

De scope van het project wordt gebruikt om een semantisch framework te ontwikkelen waarin alle data rond de voorwaarden van dienstverleningen in kaart wordt gebracht en gedeeld. Hiermee wordt een applicatieprofiel en vocabularium uitgewerkt en uiteindelijk volgt de publicatie op data.vlaanderen. Hiervoor wordt de OSLO-methodiek toegepast. Deze methodiek begint steeds met het oplijsten van diverse use cases. Daarna komt het aligneren met bestaande standaarden aan bod, zowel op Vlaams als Europees en internationaal niveau. Ontbrekende elementen worden gedefinieerd indien nodig, zo ontstaat er een nieuwe standaard.

Graag verwijzen we hiervoor naar slide 9.

Scope van het project

Ontwikkel een semantisch framework voor het in kaart brengen van Voorwaarden Dienstverlening en het delen van data

Ontwikkel een duurzaam **applicatieprofiel** en **vocabulary** voor Voorwaarden Dienstverlening.

We volgen de OSLO methodiek, wat betekent dat:



We starten van use cases



We definiëren zelf zaken waar nodig



We aligneren zoveel mogelijk met bestaande standaarden

OVERZICHT VAN DE AANPASSINGEN

Graag verwijzen we hiervoor naar slide 12.

Met het verwerken van de feedback uit de vorige sessie zijn er een aantal aanpassingen aan het model uitgevoerd. Een overzicht van de aanpassingen is hieronder weergegeven:

Overzicht van de aanpassingen

Nieuwe klassen

- InformatieConcept
- OndersteunendeWaarde
- Participatie

Verwijderde klasse

- TijdsPeriode

Verwijderde relatie

- Relatie 'Agent' en 'Bewijs'

Toegevoegde relaties

- Relatie 'Voorwaarde' en 'InformatieConcept'
- Relatie 'InformatieConcept' en 'Bewijs'
- Relatie 'InformatieConcept' en 'OndersteunendeWaarde'
- Relatie 'OndersteunendeWaarde' en 'Bewijs'

INTRODUCTIE TOT UML: KARDINALITEIT & ATTRIBUTEN

UML (Unified Modelling Language) is een modelleertaal om de OSLO-modellen op een gestandaardiseerde manier weer te geven. Aan de hand van het voorbeeld “Adoptie van een dier uit het asiel door een persoon” worden de basisconcepten van deze taal zichtbaar. Tijdens de werkgroep zijn de kardinaliteit en de attributen nog een keer uitgelicht, deze zijn cruciaal om het datamodel goed te interpreteren:

- **Kardinaliteit:** Via kardinaliteit kunnen de relaties tussen verschillende klassen extra duiding krijgen. Deze relaties hebben namelijk een bepaalde verstandhouding tegenover elkaar. Uit de kardinaliteit moet blijken hoeveel van de ene klasse er deel kunnen uitmaken van een andere klasse. Dit wordt dan toegepast op alle relaties tussen de klassen. Zo kan in dit voorbeeld een Dier opgevangen worden door 0 of 1 Asiel en kan een Asiel 0 tot meerdere Adressen hebben.
- **Attributen:** Een attribuut is een kenmerk van een klasse in een bepaalde dimensie, zoals bijvoorbeeld de naam van een persoon of de geboortedatum van een dier.

Het voorbeeld van OSLO Personen geeft weer hoe een afgewerkt UML model eruit ziet. Om dubbele interpretaties tegen te gaan wordt er ook een volledige HTML pagina opgericht op data.vlaanderen. Zo zijn de specificaties leesbaar voor iedereen zonder kennis van UML schema's. Hier staan ook de URI's van de attributen, de Unieke Resource Identificators.

Graag verwijzen we hiervoor naar slides 17-21

ONZE AANPAK

In de vorige werkgroep werd de scoping bepaald en toegelicht, meer bepaald wat binnen scope valt, wat niet en de aspecten die horen bij de implementatie. Tijdens de huidige werkgroep werd dit kort aangehaald:

Vertrekken van use cases

➤ Opdeling van use cases/concepten in verschillende categorieën:

Binnen de scope	Buiten de scope	Feature/implementation
Publieke dienstverlening	Informatieve stappen	Historische gegevens
Voorwaarden / criteria	Zakelijke dienstverlening	Simulatie
Bewijs	Overzicht beslissingen VR	Koppeling met register
Agent		
Doelgroepen		
Organisatie <ul style="list-style-type: none">• Klant• Aanbieder		

Tijdens de werkgroep kwam de dubbele interpretatie van Agent naar boven. Agent is de hoofdklasse, subklassen van deze klasse zijn zaken zoals Persoon, (Geregistreerde) Organisatie etc. De klasse Publieke Organisatie is apart opgenomen in het model aangezien deze klasse belangrijk is voor het model en bijkomende karakteristieken heeft dan de klasse Agent.

Verder werden een aantal bestaande modellen geïdentificeerd die bruikbaar zijn voor het opstellen van het datamodel. Twee uitgewerkte standaarden zijn gebruikt als inspiratie om het model op te bouwen.

De eerste standaard is het [Core Criterion and Core Evidence Vocabulary](#) (CCEV), uitgegeven door Semic. Dit model is ontworpen ter ondersteuning van de uitwisseling van informatie tussen organisaties of personen die eisen stellen of aan deze eisen voldoen. Deze standaard levert ook inspiratie over het modelleren van (on)gestructureerde bewijzen.

De tweede standaard is het [Core Public Service Vocabulary - Application Profile](#) (CPSV-AP), ook uitgegeven door Semic. Het is een eerste stap naar het creëren van een model dat overheidsdiensten beschrijft voor de burger en bedrijven. Als doel heeft het model ook een goed overzicht te geven van overheidsdiensten.

De standaarden waaruit inspiratie is gehaald, kunnen gevonden worden op slide 25 & 26.

DATAMODEL

In de eerste thematische werkgroep werd het sneuvelmodel voorgesteld aan de hand van storylines. Op basis van de gecapteerde feedback tijdens de eerste thematische werkgroep is het sneuvelmodel omgebouwd tot een datamodel. De attributen en relaties worden toegelicht aan de hand van dezelfde storylines:

1. Raf gaat sporten met de Vrijtijdspas
2. De bibliotheek van Leuven voorziet gratis studieplaatsen
3. Emma en Loes tekenen hun contract om wettelijk samen te wonen
4. De Vlaamse Overheid laat 'inkomen' mee bepalen op kortingen voor De Lijn

Bijkomend is er voor eigen gedefinieerde klassen en attributen een duidelijke definitie uitgewerkt. Tijdens de thematische werkgroep zijn deze aan bod gekomen, maar om het leesbaar te maken zullen ze nu per uitgewerkte storyline staan. De klassen, attributen, datatypes en enumeraties die niet aan bod komen zijn geërfd uit een ander datastandaarden. De definities van deze elementen zijn terug te vinden op de [Github pagina van OSLO Voorwaarden Dienstverlening](#).

RAF GAAT SPORTEN MET DE VRIJETIJDSPAS

In deze storyline komen drie klassen aan bod: **Agent**, **Voorwaarde** en **Publieke Dienstverlening**. De klasse **Agent** beschrijft alle vormen die een **Agent** kan hebben, het is een generiek concept. De klasse **Agent** is ook afgeleid uit de datastandaard OSLO Generiek. In het huidige voorbeeld is Raf een instantie van **Agent**. Als stad Hasselt de vrijetijdspas uitgeeft, dan is de stad ook een **Agent**. Zij geven dan de voorwaarde uit. De relatie tussen **Agent** en **Publieke Dienstverlening** heeft nog een hulpklasse genaamd **Participatie**. Hierin worden de verschillende rollen van **Agent** die deelneemt aan een **Publieke Dienstverlening** beschreven via een codelijst.

De klasse **Voorwaarde** is de klasse die centraal staat binnen dit model. Het wordt gedefinieerd als een voorwaarde of conditie die iets of iemand vraagt en waaraan moet worden voldaan. Deze klasse is de beslissende factor of een **Agent** al dan niet recht heeft op een **Publieke dienstverlening**. Er zijn een aantal attributen van de klasse **Voorwaarde** die ook bij andere klassen voorkomen. De naam van het attribuut wordt hieronder gegeven, met het datatype tussen haakjes gevolgd door de definitie:

- beschrijving (TaalString): Een korte uitleg die het begrip verder ondersteunt.
- naam (TaalString): Naam van de instantie, in dit geval naam van een **Voorwaarde**.
- identicator (Literal): Unieke en dubbelzijdige identicator van de **Voorwaarde**.
- type (Code): De categorie waartoe de **Voorwaarde** hoort, gekoppeld aan een codelijst ('Leeftijd', 'Woonplaats')

Uniek voor de klasse **Voorwaarde** is het attribuut beperking (Taalstring), dit attribuut geeft de tekstuele weergave van de beperking, zoals 'ouder dan 25 jaar'. Alle berekeningen en filters worden gedaan aan de hand van andere klassen die later nog aan bod komen, dit attribuut is louter voor de naamgeving.

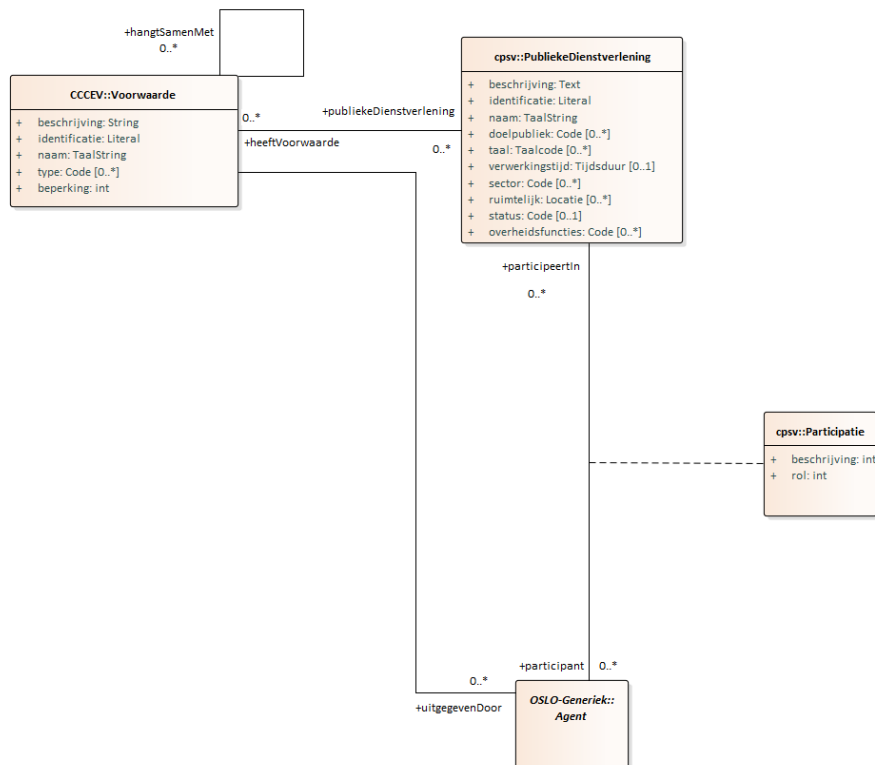
Publieke dienstverlening heeft als definitie: een geheel van verplichte of optioneel uitgevoerde of uitvoerbare acties door of in naam van een publieke organisatie. De dienstverlening is ten bate van een individu, een bedrijf, een andere publieke organisatie of groepen hiervan. De ontvangende partij van deze dienstverlening is een instantie van **Agent**. De klasse **Publieke dienstverlening** wordt overgenomen uit een andere standaard, namelijk het CPSV-AP. Specifieke attributen voor **Publieke Dienstverlening** zijn:

- doelpubliek (Code): Doelpubliek waarvoor de **Publieke Dienstverlening** bestemd is.
- verwerkingstijd (Tijdsduur): De geschatte tijd tot het uitvoeren van de **Publieke Dienstverlening**
- sector (Code): Sector van de **Publieke Dienstverlening**
- ruimtelijk (Locatie): Gebied waarin **Publieke Dienstverlening Geldig** is
- status (Code): Status van de **Publieke Dienstverlening** (actief, non-actief, aanmaak...)
- overheidsfuncties (Code): Verschillende overheidsfuncties.

De kardinaliteit van deze attributen is 0 tot 1 of meerdere, wat betekent dat ze semantisch gezien niet verplicht zijn om in te vullen. Bij de implementatie kunnen bepaalde attributen naar wens wel verplicht worden.

De datatypes die nog op 'int' staan worden aangepast richting de volgende werkgroep. Voor de definitie van de klasse **Bewijs** wordt verwezen naar de derde storyline.

Graag verwijzen we hiervoor naar slides 50 - 51.



DE BIBLIOTHEEK VAN LEUVEN VOORZIET GRATIS STUDIEPLAATSEN

In deze storyline worden er vier klassen behandeld. Één daarvan, **Publieke dienstverlening**, werd al besproken binnen de vorige storyline. De nog niet besproken klassen voor dit model zijn **Procedure**, **Formeel kader** en **Publieke organisatie**.

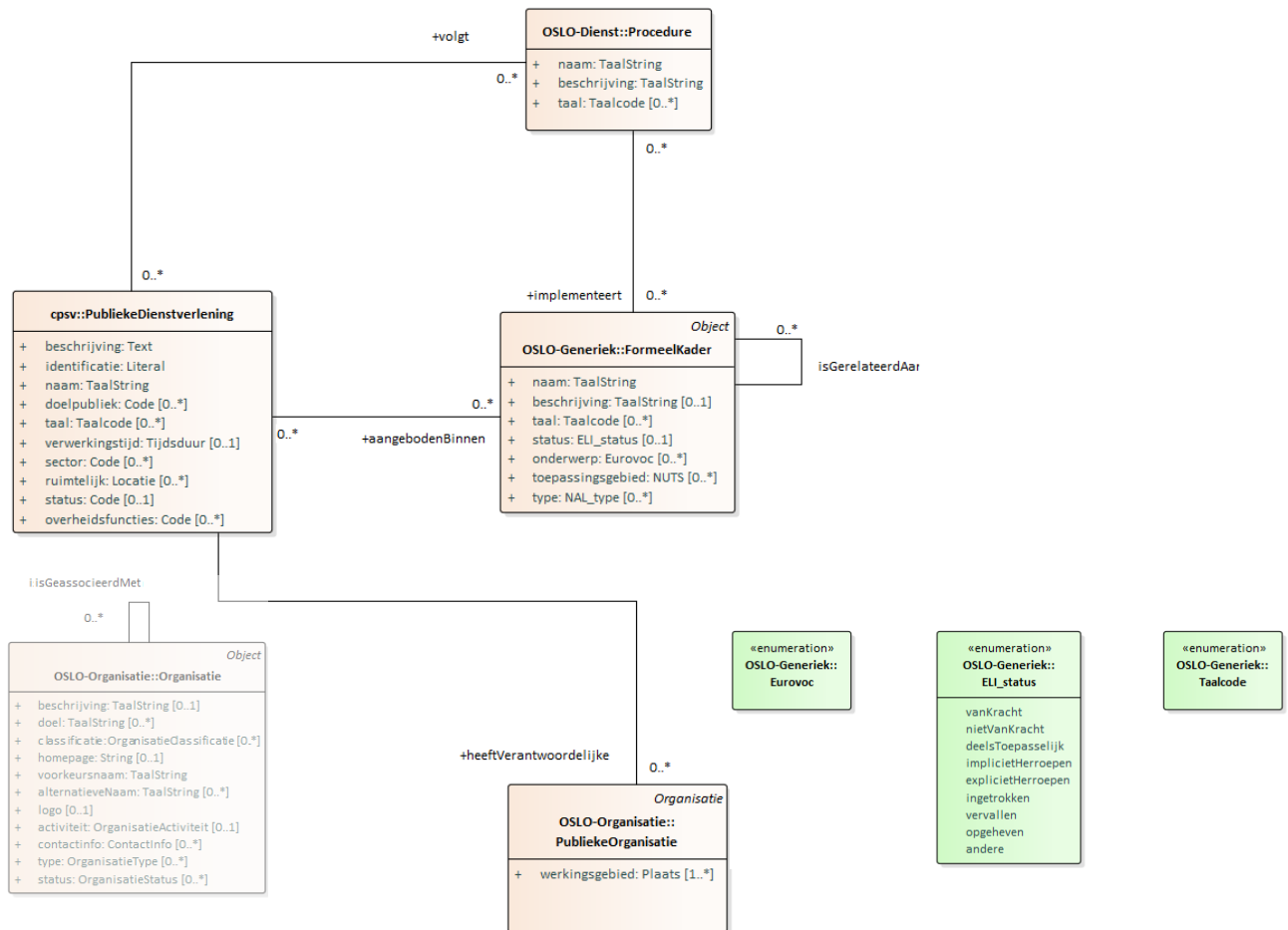
Een **Procedure** wordt gezien als de regels, richtlijnen of procedures van een **Publieke dienstverlening**. Deze definitie is afgeleid van CPSV-AP.

Een **Formeel kader** daarentegen wordt gezien als alle formele documenten die regels, rechten en plichten ten aanzien van burgers, bedrijven, overheden, etc... beschrijven. De definitie van deze klasse is overgenomen uit OSLO Generiek.

Onder **Publieke organisatie** wordt een organisatie verstaan die via wettelijk kader op elk niveau als onderdeel van de publieke sector gedefinieerd wordt. Een **Publieke Organisatie** is een afgeleide van **Organisatie**, alle kenmerken van een **Organisatie** gelden dus ook voor een **Publieke Organisatie**. De klasse staat nu links onderaan, maar in het model wordt er verwezen naar het originele model van OSLO Organisatie en zal dus niet opgenomen worden in het huidige model.

Tijdens de werkgroep kwam naar boven dat de klassen **Procedure** en **FormeelKader** verouderd zijn en op een andere manier voorgesteld kunnen worden. Er is besloten om niet dieper op deze twee klassen in te gaan en het nut van deze klassen te bekijken in functie van het traject over Voorwaarden Dienstverlening.

Voor meer informatie verwijzen we graag naar slides 32 -33



EMMA EN LOES TEKENEN HUN CONTRACT OM WETTELIJK SAMEN TE WONEN

Graag verwijzen we hiervoor naar slides 34-36

In deze storyline komen de klassen **Bewijs**, **Bewijs Type** en **Bewijs Type Lijst** aan bod. De definities van deze klassen zijn:

- **Bewijs:** De gegevens die aantonen dat al dan niet aan een voorwaarde voldaan is.
- **BewijsType:** Informatie over de kenmerken en de nodige inhoud van een Bewijs.
- **BewijsTypeLijst:** Lijst van Bewijstypes die nodig zijn om aan een voorwaarde te voldoen.

De klasse **Bewijs** heeft als attributen:

- vertrouwelijkheidsLevelType (Code): Classificatie van de vertrouwelijkheid (Openbaar, geheim, vertrouwelijk...)
- afleveringsdatum (DateTime): Datum wanneer het **Bewijs** is afgeleverd.
- toepassingsgebied (NITS): Gebied waarin **Bewijs** is verkregen
- ondersteuntWaarde (OndersteunendeWaarde): Waarde op het **Bewijs** die bepaalt of aan de **Voorwaarde** voldaan is.
- geldigheidsperiode (Tijdsperiode): Periode waarin het **Bewijs** geldig is.

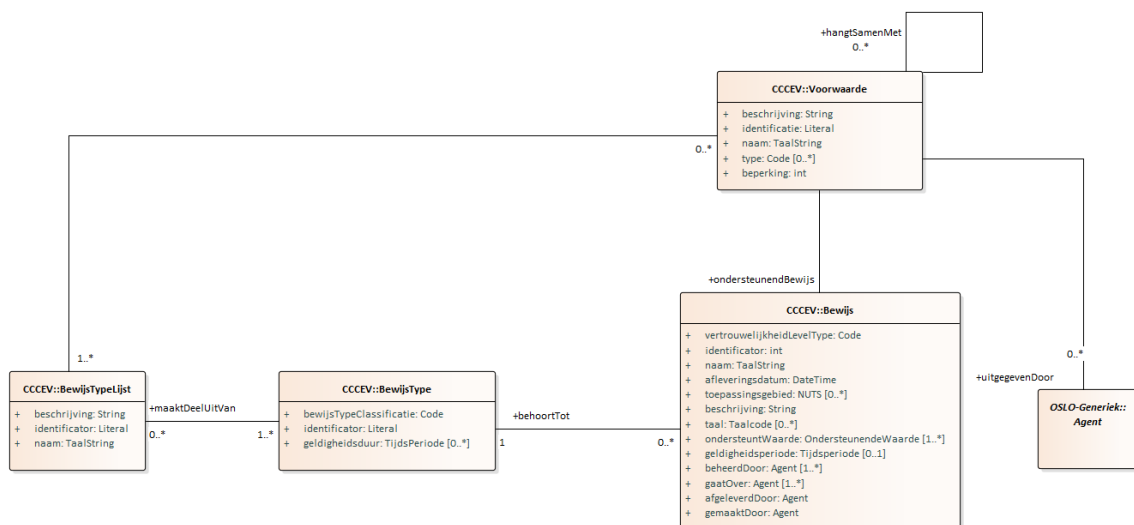
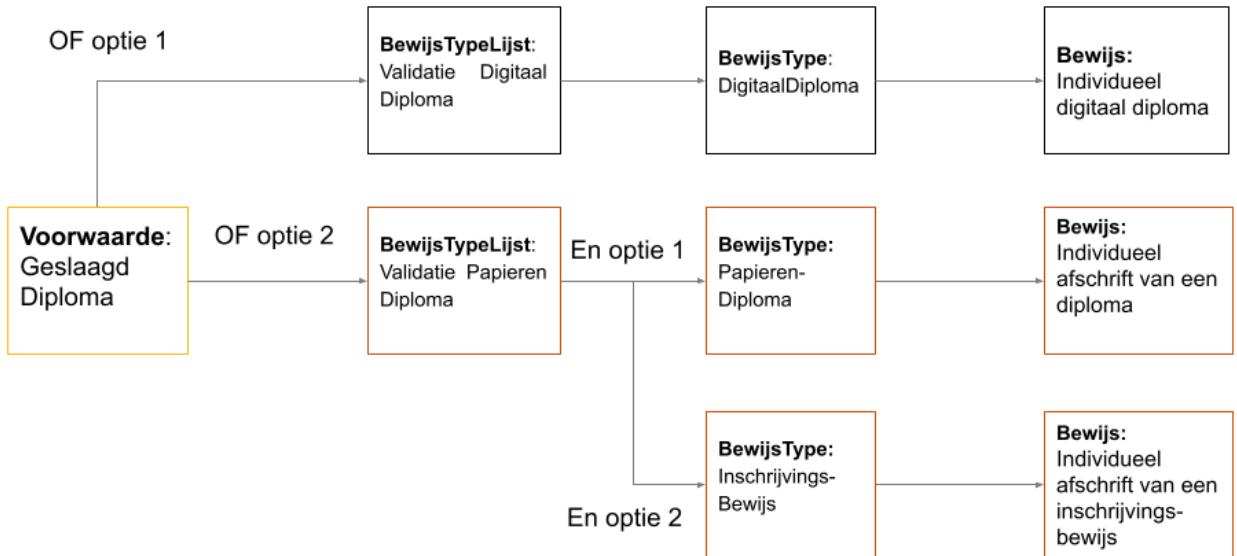
Een andere aanpassing is de relatie tussen **Agent** en **Bewijs**, deze werd verwijderd. Deze klassen zijn nog wel verbonden met elkaar. De relatie tussen **Bewijs** en **Agent** kan veel vormen aannemen. Om duidelijk weer te geven welke relaties er kunnen bestaan, zitten deze nu vervat onder de attributen van **Bewijs**. Een **Bewijs**:

- Wordt beheerd door een **Agent**
- Gaat over een **Agent**
- Wordt aangeleverd door een **Agent**
- Is gemaakt door een **Agent**

Hoewel er vier keer **Agent** staat, kunnen dit vier verschillende instanties van **Agent** zijn. Het is nu niet meer nodig om een relatie te tekenen tussen **Bewijs** en **Agent**.

Tijdens de werkgroep zijn de drie klassen voorgesteld met een data voorbeeld. Dit was voor de leden van de werkgroep niet helemaal duidelijk. Er werd afgesproken om richting de volgende thematische werkgroep een datavoorbeeld uit te werken. Tijdens de afgelopen werkgroep kwam volgende toelichting hier rond aan bod:

Een voorwaarde kan voldaan worden door middel van verschillende bewijzen zoals een rijbewijs, universiteitsdiploma etc. De klasse **Bewijs** duidt concrete instanties aan, zoals een Europees Rijbewijs, het papieren diploma van een arts uit 1974... Deze bewijzen kunnen tot een bepaald **BewijsType** horen, bijvoorbeeld het type 'PapierenDiploma', 'ElektronischRijbewijs'.. Het **BewijsType** specificeert de vorm en inhoud van het **Bewijs**. Zo zal er bij 'PapierenDiploma'(**BewijsType**) verplicht zijn om de instelling en het jaartal te vermelden op het papieren diploma van een arts uit 1974 (**Bewijs**). Per **Voorwaarde** hangt er 1 of meerdere **BewijsTypeLijst** aan vast. Deze lijst bevat 1 of meerdere **BewijsType** die zorgen dat het bewijs is voldaan. Zo is er de lijst 'Validatie Papieren Diploma' met meerdere **BewijsType**, namelijk 'PapierenDiploma' en het datatype 'InschrijvingsBewijs'. Beide **Bewijzen** moeten aangeleverd worden want ze zitten in de **BewijsTypeLijst**. Een andere **BewijsTypeLijst** kan zijn 'Validatie Digitaal Diploma', met slechts 1 **BewijsType** namelijk 'DigitaalDiploma'. Het **Bewijs** zal dan een digitaal diploma zijn. Om aan de **Voorwaarde** te voldoen moet maar 1 **BewijsTypeLijst** worden aangeleverd. Zo is er een combinatie van een AND/OR expressie.

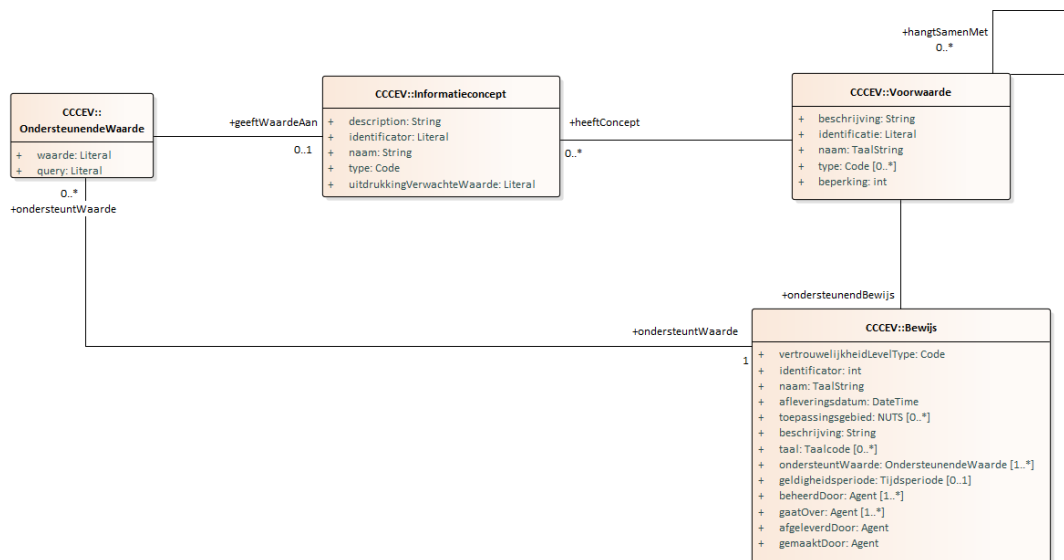


DE VLAAMSE OVERHEID LAAT 'INKOMEN' MEE BEPALEN OP KORTING VOOR DE LIJN

We verwijzen voor meer informatie graag naar slide 37.

In deze storyline komt het effectief modelleren van de **Voorwaarde** aan bod. Om de **Voorwaarde** machineleesbaar te maken, werden twee klassen gebruikt:

- **Informatieconcept:** De informatie die een **Bewijs** aanlevert en de **Voorwaarde** nodig heeft.
- **OndersteunendeWaarde:** Waarde aangeleverd door **Bewijs** die nodig is in **Informatieconcept**.



In **Informatieconcept** zit het attribuut **uitdrukkingVerwachteWaarde**. Hierin wordt de regel machineleesbaar gemaakt. Er zijn een aantal vormen aangehaald in CCCEV en ook onderzocht om dit voor te stellen, namelijk:

1. **Zonder waarde:** tekstvorm die de regel weergeven
2. **Met waarde:** een waarde die de regel valideert
3. **Datatype:** aparte datatypes voor regels
4. **Serialisatie:** volledig omzetten van regels in een machineleesbare taal

Deze vier vormen worden via een data voorbeeld voorgesteld. De gebruikte taal is JSON-LD, deze wordt vaak gebruikt in het kader van Linked Data. Het voorbeeld gaat

telkens over een Seniorenbiljet, de **Publieke Dienstverlening** hiervan heet Seniorenbiljet. Het wordt aangeboden door de **NMBS**, er is een leeftijdsvereiste als **Voorwaarde**. In het **InformatieConcept** zit uiteindelijk deze **Voorwaarde**. Deze voorwaarde kan op de vier eerder vermelde vormen worden weergegeven op de volgende manier:

Zonder waarde

In de eerste vorm wordt in de description van **InformatieConcept** gezet 'Gebruiker is 65 of ouder'. Dit is niet machine leesbaar, deze methode is dus niet wenselijk.

```
{
  "@context": {
    "dcterms": "http://purl.org/dc/terms/",
    "skos": "http://www.w3.org/2004/02/skos/core#",
    "cv": "http://data.europa.eu/m8g/",
    "cpsv": "http://purl.org/vocab/cpsv#",
    "cccev": "http://data.europa.eu/m8g/cccev#",
    "Requirement.forPublicService": {
      "@reverse": "cpsv:holdsRequirement"
    }
  },
  "@type": "cv:Requirement",
  "Requirement.forPublicService": {
    "@type": "cpsv:PublicService",
    "dcterms:title": "Seniorenbiljet",
    "cpsv:hasCompetentAuthority": {
      "@type": "PublicOrganisation",
      "skos:prefLabel": "NMBS"
    },
    "dcterms:title": "Leeftijdsvereiste",
    "cccev:hasConcept": {
      "@type": "cccev:InformationConcept",
      "dcterms:description": "Gebruiker is 65 of ouder."
    }
  }
}
```

Met waarde

De klasse **InformatieConcept** laat nog andere vormen toe om de voorwaarde weer te geven. De tweede vorm geeft een waarde weer van het label. Het label is 'MinimumLeeftijd' met waarde '65' in het attribuut **UitdrukkingVerwachteWaarde**. Voor simpele regels is dit handig, maar complexere regels zullen niet machineleesbaar zijn.


```
{
  "@context": {
    "dcterms": "http://purl.org/dc/terms/",
    "skos": "http://www.w3.org/2004/02/skos/core#",
    "cv": "http://data.europa.eu/m8g/",
    "cpsv": "http://purl.org/vocab/cpsv#",
    "cccev": "http://data.europa.eu/m8g/cccev#",
    "xml-schema": "http://www.w3.org/2001/XMLSchema#",
    "Requirement.forPublicService": {
      "@reverse": "cpsv:holdsRequirement"
    }
  },
  "@type": "cv:Requirement",
  "Requirement.forPublicService": {
    "@type": "cpsv:PublicService",
    "dcterms:title": "Seniorenbiljet",
    "cpsv:hasCompetentAuthority": {
      "@type": "PublicOrganisation",
      "skos:prefLabel": "NMBS"
    },
    "dcterms:title": "Leeftijdsvereiste",
    "cccev:hasConcept": {
      "@type": "cccev:InformationConcept",
      "skos:preflabel": "MinimumLeeftijd",
      "cccev:expressionOfExpectedValue": {
        "@value": "65",
        "@type": "xml-schema:integer"
      }
    }
  }
}
```

Datatype

De derde methode is een extra datatype ontwerpen om de regel voor te stellen. In dit voorbeeld is het nieuwe datatype dan 'minimumLeeftijd'. De titel is 'Leeftijdsvereise' in het **InformatieConcept** met type 'Leeftijd'. Het gaat niet om elke leeftijd, maar het gaat over een minimumleeftijd. De 'c-evt' is een codelijst met concepten (zoals minimumleeftijd) waarmee de voorwaarden kunnen worden gemaakt. Deze optie heeft als voordeel om een complexere waarde te lezen, zoals 65+. De codelijst moet wel bij de implementatie gemaakt worden waar ook duidelijke afspraken rond de waarden bij moeten komen. Deze codelijsten zijn van de vorm 'skos concept' en zijn via bepaalde tools eenvoudig op te stellen. Deze concepten hebben een eigen pagina en een URI, waarin de formele beschrijving, definitie, taal etc. staat van het concept.

```
{
  "@context": {
    "dcterms": "http://purl.org/dc/terms/",
    "skos": "http://www.w3.org/2004/02/skos/core#",
    "cv": "http://data.europa.eu/m8g/",
    "cpsv": "http://purl.org/vocab/cpsv#",
    "cccev": "http://data.europa.eu/m8g/cccev#",
    "cl-evt": "https://example.com/concept/expectedValueType/",
    "Requirement.forPublicService": {
      "@reverse": "cpsv:holdsRequirement"
    }
  },
  "@type": "cv:Requirement",
  "Requirement.forPublicService": {
    "@type": "cpsv:PublicService",
    "dcterms:title": "Seniorenbiljet",
    "cpsv:hasCompetentAuthority": {
      "@type": "PublicOrganisation",
      "skos:prefLabel": "NMBS"
    },
    "dcterms:title": "Leeftijdsvereiste",
    "cccev:hasConcept": {
      "@type": "cccev:InformationConcept",
      "skos:preflabel": "Leeftijd",
      "cccev:expressionOfExpectedValue": {
        "@value": "65+",
        "@type": "c-evt:minimumLeeftijd"
      }
    }
  }
}
```

Serialisatie

In de Linked Data wereld is er een aparte manier om regels weer te geven. Dit vervangt het specificeren van datatypes zoals in de vorige vorm, er zijn al verschillende concepten gedefinieerd. Het type van het attribuut 'uitdrukkingVerwachteWaarde' is nu 'rif:Document', de regelvorming via de RIF Rules. Deze regels komen onder de waarde van 'uitdrukkingVerwachteWaarde' te staan.

```

"@context": {
  "dcterms": "http://purl.org/dc/terms/",
  "skos": "http://www.w3.org/2004/02/skos/core#",
  "cv": "http://data.europa.eu/m8g/",
  "cpsv": "http://purl.org/vocab/cpsv#",
  "cccev": "http://data.europa.eu/m8g/cccev#",
  "rif": "http://www.w3.org/2007/rif#",
  "Requirement.forPublicService": {
    "@reverse": "cpsv:holdsRequirement"
  }
},
"@type": "cv:Requirement",
"Requirement.forPublicService": {
  "@type": "cpsv:PublicService",
  "dcterms:title": "Seniorenbiljet",

  "cpsv:hasCompetentAuthority": {
    "@type": "PublicOrganisation",
    "skos:prefLabel": "NMBS"
  },
  "dcterms:title": "Leeftijdsvereiste",
  "cccev:hasConcept": {
    "@type": "cccev:InformationConcept",
    "skos:prefLabel": "Leeftijd",
    "cccev:expressionOfExpectedValue": {
      "@value":
        Document(
          Prefix (pred <http://www.w3.org/2007/rif-builtin-predicate#>)
          Prefix (pers <http://example.com/ns/person#>)
          Prefix (serv <http://example.com/ns/service#>)
          Prefix (cpt <http://example.com/concept/actiontype#>)
          Group
          (
            Forall ?pers:Person ?pers:age ?serv:Service (
              cpt:entitles(?pers:Person ?serv:Service):-
                And(External(pred:numeric-greater-than-or-equal(?
                  pers:age 65)))
            )
          )
        )
      ,
      "@type": "rif:Document"
    }
  }
}

```

De werkwijze van RIF wordt toegelicht aan de hand van twee kleuren voorbeelden onderaan. **RIF** staat voor **Rule Interchange Format** en wordt gebruikt op het semantisch web voor het weergeven van regels. Alle benodigdheden zitten in een document, vandaar de start met 'Document('. Eerst wordt er verwezen naar de URI's van de gebruikte data. Daarna staat er bij 'Group' de groepering van regels. De 'Forall' geeft alle parameters weer, in dit geval persoon, leeftijd en dienst. De 'cpt' is de actie die uit de regel komt, in dit geval recht hebben op. Het 'pred' is het predicaat, wat de handeling of rekenwerk bepaalt. Zowel de acties als de predicaten komen uit een reeds gedefinieerde lijst, maar er kunnen ook eigen conceptlijsten worden opgesteld. De regel zelf is dus de 'And', daar staat de leeftijd hoger of gelijk aan 65 moet zijn.

Voorbeeld met 1 voorwaarde

```
Document(  
  Prefix (pred <http://www.w3.org/2007/rif-builtin-predicate#>)  
  Prefix (pers <http://example.com/ns/person#>)  
  Prefix (serv <http://example.com/ns/service#>)  
  Prefix (cpt <http://example.com/concept/actiontype#>)  
  Group  
  (  
    Forall ?pers:Person ?pers:age ?serv:Service (  
      cpt:entitles(?pers:Person ?serv:Service) :-  
        And(External(pred:numeric-greater-than-or-equal(?pers:age 65)))  
      )  
    )  
  )  
)
```

De regel stelt dat voor alle combinaties van een persoon, zijn leeftijd en een dienst, als de leeftijd van de persoon hoger is dan of gelijk is aan 65, de persoon recht heeft op de dienst.

Het **getal** is de OndersteunendeWaarde, aangeleverd door Bewijs

Voorbeeld met 2 voorwaarden

```
Document(  
  Prefix (pred <http://www.w3.org/2007/rif-builtin-predicate#>)  
  Prefix (pers <http://example.com/ns/person#>)  
  Prefix (serv <http://example.com/ns/service#>)  
  Prefix (cpt <http://example.com/concept/actiontype#>)  
  Prefix (house <http://example.com/ns/house#>)  
  Group  
  (  
    Forall ?pers:Person ?pers:age ?serv:Service ?pers:house ?house:nrBathroom (  
      cpt:entitles(?pers:Person ?serv:Service) :-  
        And(  
          External(pred:numeric-greater-than-or-equal(?pers:age 65))  
          External(pred:numeric-less-than(?house:nrBathroom 4)))  
      )  
    )  
  )  
)
```

De regel stelt dat voor alle combinaties van een persoon, zijn leeftijd, een huis en een dienst, als de leeftijd van de persoon hoger is dan of gelijk is aan 65 EN de persoon heeft huis met minder dan 4 badkamers dan heeft hij recht heeft op de dienst.

Het **getal** is wederom de OndersteunendeWaarde, aangeleverd door Bewijs.

Er zijn ook andere vormen van serialisatie onderzocht om te gebruiken. Het semantisch web voorziet een aantal structuren of specifieke talen om dit te doen, maar RIF is de beste kandidaat. Andere vormen zijn:

Waarom RIF?

De bedoeling is om regels te maken in het datamodel.
Hiervoor zijn een aantal opties onderzocht:

1. **SWRL**: Oude set van regels, niet meer relevant
2. **SPIN**: Deze regelset beschrijft beperkingen op de klassen in het model, zoals disjoints
3. **R2RML**: Vertaalt relationele databanken naar RDF-Statements
4. **SHACL**: Valideert de Linked Data (LD) in het model
5. **XPATH**: Beschrijft de XML specificatie
6. **OCL**: Object *Constraint* Language, vergelijkbaar met SPIN & SHACL dus ongeschikt.
7. **RIF**: Gaat over regels ivm het onderwerp, niet over regels van het model

Rule Interface Format

Doel:

- Regels weergeven in een machine leesbaar formaat.
- Regels uitwisselen en interoperabiliteit vergemakkelijken.

Syntaxis:

- Gebruikt XML of RDF als het serialisatie formaat.
- Staat verschillende dialecten toe om regels uit te drukken.

Structuur:

- Regels bestaan uit een combinatie van voorwaarden en acties.
- Voorwaarden definiëren de criteria voor het toepassen van regels.
- Uitkomst is simpel: dienst wordt verleend of niet.

Dialecten:

- RIF staat verschillende dialecten om regels mee te bouwen.
- Wij baseren op RIF-BLD (Basic Logic Dialect) en RIF-Core, de basis van RIF.
- Indien gewenst kan er zelf een dialect opgesteld worden

Interoperabiliteit:

- RIF wisselt regels uit tussen regel-gebaseerde systemen.
- Regels die in het ene systeem zijn geschreven, worden begrepen door een ander systeem.
- Ondersteunt naadloze integratie en samenwerking tussen verschillende systemen.

GEbruik van InformatieConcept

In het vorige deel zijn de vier vormen van **InformatieConcept** besproken. De eerste vorm was eenvoudig te maken, maar niet machineleesbaar. De laatste vorm was volledig machineleesbaar, maar vergt wat werk om te implementeren. Uit de werkgroep kwam naar boven dat NRules ook een mogelijke vorm was om de regels op te bouwen. In NRules kan er ook RIF gebruikt worden. Het handige aan NRules is dat er een validator bestaat, deze

tool kan dus bepalen of regels geldig zijn opgebouwd. Dit bestaat nog niet voor RIF, dus N3 wordt verder onderzocht richting de volgende thematische werkgroep. Als N3 de regels formeel kan voorstellen en ook de optie heeft om te linken met de Linked Data dan kan er overgegaan worden naar N3.

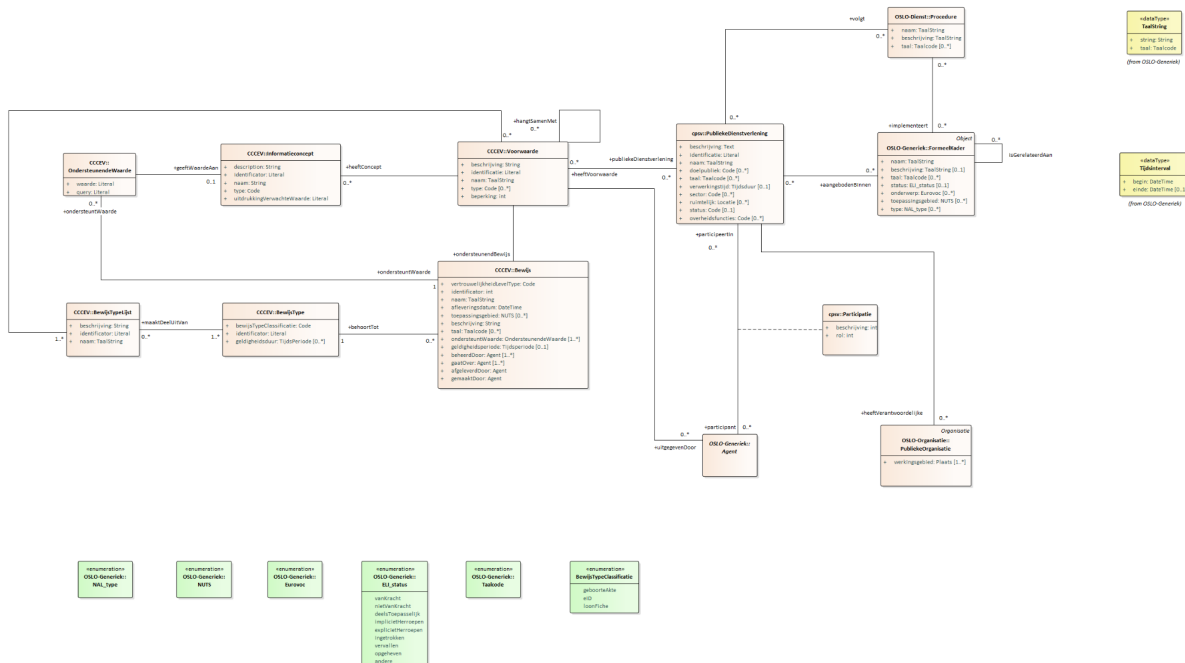
Alle vier vormen blijven steeds open voor gebruik binnen de uitgewerkte standaard en de uiteindelijke keuze hangt af van de effectieve implementatie. Richting de volgende thematische werkgroep zal N3 met RIF uitgewerkt zijn, alsook een reëel datavoorbeeld van alle andere vormen. Bij de implementatie zal er dan bepaald worden op welke manier het **InformatieConcept** ingevuld wordt. Dit kan ook in de gebruikersnota van **InformatieConcept**, als er gekozen wordt om het datamodel aan te passen. Om het datamodel generiek te houden, komt er best een implementatiemodel waarin het datamodel wordt klaargestoomd voor implementatie door de verschillende overheidsinstanties.

VOORWAARDEN IN HET MODEL: VOLGENDE STAPPEN

HET DATAMODEL 'OSLO VOORWAARDEN DIENSTVERLENING'

Hieronder is het volledige datamodel terug te vinden. Met de verkregen feedback zullen aanpassingen doorgevoerd worden om tot een eerste concreet model te komen

Deze is ook terug te vinden op slide 49.



VOLGENDE STAPPEN

Richting de volgende thematische werkgroep zullen een aantal stappen genomen worden. Op de slide hieronder is zichtbaar wat deze stappen zijn.

Voor de volgende stappen verwijzen we ook graag naar slides 52 - 55

Volgende stappen



Verwerken van alle input uit de thematische werkgroep.



Rondsturen van een verslag van deze werkgroep. Feedback is zeker welkom.



Feedback capteren via GitHub.



Aangepaste versie van semantisch model publiceren op GitHub. Hier is feedback ook zeker welkom.

VOLGENDE THEMATISCHE WERKGROEP

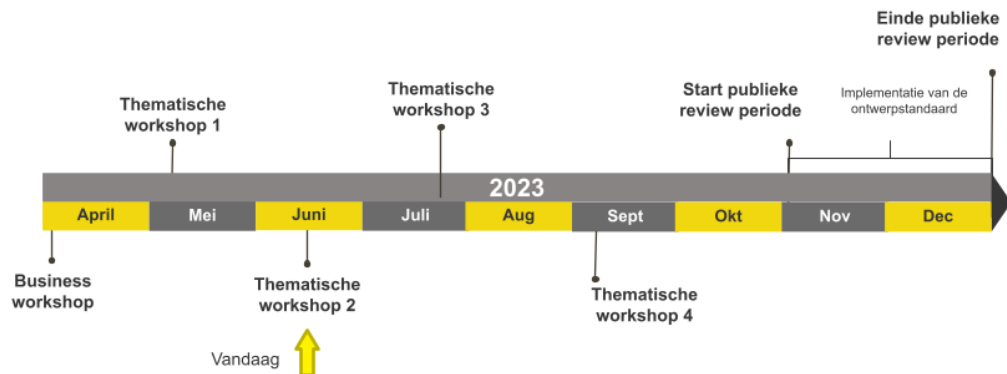
De volgende thematische werkgroep zal plaatsvinden op **donderdag 27 juli van 9u00 tot 12u00**. Deze zal online doorgaan via Microsoft Teams, inschrijven kan via [deze link](#).

Om een overzicht te geven van wat er nog binnen het traject 'OSLO Voorwaarden dienstverlening' gepland staat, verwijzen we graag naar slide 53 of naar de [landingspagina](#).

OSLO tijdslijn

Thematische werkgroep 3 op **donderdag 27 juli: 09u00 - 12u00**

Schrijf u in via volgende link: [3de thematische werkgroep](#)



Indien er feedback is of vragen zijn, kunnen deze gedeeld worden op de GitHub van Digitaal Vlaanderen of op onderstaande e-mailadressen.

E-mailadressen:

- digitaal.vlaanderen@vlaanderen.be
- laurens.vercauteren@vlaanderen.be
- yaron.dassonneville@vlaanderen.be
- jef.lieken@vlaanderen.be

[GitHub](#)