

# MBG - Metamodel BerichtstructuurGegevens

VNG Realisatie Standaard  
Consultatieversie 21 februari 2024

**Deze versie:**

<https://vng-realisatie.github.io/publicatie/cim/mbg/0.9.0>

**Laatst gepubliceerde versie:**

<https://vng-realisatie.github.io/publicatie/cim/mbg>

**Laatste werkversie:**

<https://vng-realisatie.github.io/MBG-Werkomgeving/>

**Redacteurs:**

Robert Melskens ([VNG Realisatie](#))

Johan Boer ([VNG Realisatie](#))

**Auteur:**

Robert Melskens ([VNG Realisatie](#))

**Doe mee:**

[GitHub VNG-Realisatie/MBG-Werkomgeving](#)

[Dien een melding in](#)

[Revisiehistorie](#)

[Pull requests](#)

Dit document is ook beschikbaar in dit niet-normatieve formaat: [pdf](#)



Dit document valt onder de volgende licentie:  
[EUROPEAN UNION PUBLIC LICENCE v. 1.2](#)

## Samenvatting

Metamodel voor het beschrijven van Berichtgegevens (MBG), versie **0.9**.

Met MBG, het Metamodel voor BerichtstructuurGegevens, wordt een metamodel beschreven waarmee berichten kunnen worden gemodelleerd in de vorm van Berichtstructuurmodellen. Het beschrijft de metaklassen, metastructuur en metagegevens als grondslag voor alle onderdelen van een bericht. Doel hiervan is het vastleggen van de methode van het modelleren van berichten zodat dit op een eenduidige gebeurd en de gemeenschappelijke tooling deze modellen ook kan verwerken. Vooralsnog wordt ingezet op het definiëren van een metamodel voor het beschrijven

van OAS specificaties voor REST API's maar het is tevens bedoeld als basis voor verbetering en doorontwikkeling van dit metamodel.

## Status van dit document

Dit is een door het TO goedgekeurde consultatieversie. Commentaar over dit document kan gestuurd worden naar

## Inhoudsopgave

<b>1.</b>	<b>Inleiding</b>
1.1	Toepassingsgebied
1.2	Doelgroep
1.3	Leeswijzer
1.4	Gebruikswijzer
1.5	Wat is een BSM
1.6	Typering van modellen en scope van deze standaard
1.7	Wat is het metamodel voor Berichtstructuurmodellering
1.8	Uitdrukken in UML
1.9	Beheer
1.10	Normreferenties
<b>2.</b>	<b>Metamodel Algemeen</b>
2.1	Structuur metamodel
2.1.1	Kern
2.1.2	Groepering
2.2	Objecten
2.2.1	Berichttypes
2.2.1.1	Getberichttype
2.2.1.2	Postberichttype
2.2.1.3	Putberichttype
2.2.1.4	Patchberichttype
2.2.1.5	Deleteberichttype
2.2.2	Padtype
2.2.3	Koppelvlak
2.2.4	Domein
2.2.5	Bericht
2.3	Specificatie metagegevens
2.3.1	Specificatie metagegevens Packages

2.3.1.1	Metagegeven: Naam
2.3.1.2	Metagegeven: beheerder-email
2.3.1.3	Metagegeven: Koppelvlak-naam
2.3.1.4	Metagegeven: project_url
2.3.1.5	Metagegeven: release
2.3.1.6	Metagegeven: Serialisatie
2.3.2	Specificatie metagegevens modelelementen
2.3.2.1	Metagegeven: Naam
2.3.2.2	Metagegeven: berichtcode
2.3.2.3	Metagegeven: tag
2.3.2.4	Metagegeven: servicename
2.3.2.5	Metagegeven: custom_path_facet
2.3.2.6	Metagegeven: Positie
2.3.3	Specificatie metagegevens modelelement bindingen
2.3.3.1	Metagegeven: heeft padrelatie
2.3.3.2	Metagegeven: heeft entiteitrelatie
2.3.3.3	Metagegeven: verwijst naar supertype
2.4	Toegestane waarden voor (bepaalde) metagegevens
2.4.1	Waardebereik metagegevens
2.4.2	Defaultwaarden voor metagegevens modelelementen

### 3. Conformiteit

#### A. Index

- A.1 Begrippen gedefinieerd door deze specificatie
- A.2 Begrippen gedefinieerd door verwijzing

#### B. Referenties

- B.1 Normatieve referenties

## Colofon en Voorwoord

### Colofon

Aan de totstandkoming van deze versie van dit metamodel hebben de volgende inhoudelijke experts op het gebied van informatimodellering meegewerkt:

Naam	Organisatie	Rol en achtergrond
Robert Melskens	VNG	Invetor Gebruikersgroep. Invetor ontwikkelaar. MIM expertgroep.

Naam	Organisatie	Rol en achtergrond
Johan Boer	VNG	MIM expertgroep. Adviseur standaarden.

## Voorwoord

Bij de VNG en Kennisnet worden de Conceptuele InformatieModellen (CIM) die op basis van het MIM worden vervaardigd en die een beeld geven van de informatiebehoefte binnen een domein op conceptueel niveau omgezet naar Uitwisselings Gegevensmodellen (UGM). Een UGM is een vertaling van een CIM naar een technische structuur. Een UGM biedt echter onvoldoende informatie voor het genereren van berichtspecificaties (in yaml of xml) en daarom wordt een Bericht Structuur Model (BSM) vervaardigd waarbinnen het UGM gebruikt wordt om de inhoud van de berichten te definiëren.

Teneinde het gebruik van BSM's ook buiten de VNG en Kennisnet mogelijk te maken is besloten de structuur van de BSM's formeel te beschrijven. Onderliggend document is daarvan het resultaat. Het dient als basis voor verbeteringen en uitbreidingen waaraan binnen de Imvertor Gebruikersgroep gewerkt zal worden.

## 1. Inleiding

Het onderliggende document bevat het Metamodel voor BerichtstructuurGegevens (MBG), voor het beschrijven van Berichtstructuurmodellen. VNG Realisatie heeft er voor gekozen dit metamodel als een extensie op het MUG (Metamodel UitwisselingsGegevens) te definiëren. In theorie is het ook mogelijk om het MBG als een extensie op het MIM te definiëren. In beide gevallen vormen ze tezamen de basis voor een Bericht Structuur Model (BSM). Het metamodel beschrijft de modellering zoals toegepast bij VNG Realisatie en Kennisnet bij het modelleren van API-specificaties. Het dient als basis voor de doorontwikkeling daarvan binnen de Imvertor community zodat het ook voor andere organisaties en op mogelijke andersoortige specificaties toepasbaar wordt. Het model bevat duidelijke afspraken over het vastleggen van berichtgegevensspecificaties.

Dit document is opgesteld met kennis aanwezig bij VNG Realisatie.

### 1.1 Toepassingsgebied

Het metamodel biedt de modelleringstaal waarmee het bericht-technische deel van een BSM gemaakt, gelezen en begrepen kan worden. Het doel hiervan is: \* de leesbaarheid en eenduidigheid van BSM's te vergroten; \* een basis te leggen voor het aanbrengen van verbeteringen en uitbreidingen van dit metamodel en de gerelateerde BSM's; \* tooling te kunnen (door)ontwikkelen

en (her)gebruiken voor resp. door alle partijen die kiezen voor dit metamodel; \* en in het verlengde hiervan, om op een meer geautomatiseerde werkwijze sneller en beter voorspelbaar uitwisselingsstandaarden op te kunnen stellen.

Voor BSM's die op basis van dit metamodel zijn beschreven geldt: \* ze zijn eenduidig te interpreteren en goed te vergelijken; \* er kan documentatie mee worden opgesteld of gegenereerd welke geschikt is voor publicatie; \* er kunnen specificaties mee worden gegenereerd die direct geschikt zijn om bijv. als OAS voor een API-standaard in te zetten.

## 1.2 Doelgroep

Dit document is primair bestemd voor berichtontwerpers die BSM's maken, mensen die model-driven verder werken op basis van informatiemodellen die zijn vervaardigd op basis van [\[MIM\]](#). Kennis van informatiemodelleringen middels [\[UML\]](#) is een vereiste. Dit metamodel richt zich in het bijzonder op de informatievoorziening binnen het overheidsdomein, al is het ook in bredere context inzetbaar.

Daarnaast is dit document ook bruikbaar voor ontwikkelaars van tooling die de, op basis van de met dit metamodel vervaardigde, BSM's als input nemen.

## 1.3 Leeswijzer

Het metamodel beschrijven we in 3 hoofdstukken en enkele bijlages.

Lees de [Inleiding](#) verder voor inzicht in wat we onder een BSM en onder een metamodel verstaan, hoe deze modellen zich verhouden tot UML en de vier lagen metamodel architectuur van de Object Management Group [\[OMG\]](#),

Het hoofdstuk [Metamodel Algemeen](#) bevat de beschrijving van alle bouwstenen c.q. de modelementen van het MBG metamodel, in de vorm van definities en specificaties. De betekenis en toelichting van de modelementen van het metamodel vormt het materiaal waarmee een uitputtende modelspecificatie kan worden opgesteld.

Het hoofdstuk [Metamodel in UML](#) beschrijft hoe de implementatie van het MBG in [\[UML\]](#) er uit ziet. In dit hoofdstuk wordt beschreven hoe het MBG metamodel zich verhoudt tot het UML metamodel, welke uitbreidingen c.q. verbijzonderingen van het UML metamodel zijn aangebracht.

In het hoofdstuk [Afspraken & Regels](#) ..... In de bijlage [Diagrammen](#) vind je een aantal diagrammen die betrekking hebben op het MBG metamodel. Bijlage [Metamodel UitwisselingsGegevens \(MUG\)](#) beschrijft het MUG metamodel, een metamodel dat nagenoeg 1 op 1 te mappen is op het MIM en alleen qua naamgeving van de modelementen en qua metagegevens afwijkt van het MIM metamodel. Om die reden hebben we dit niet in een apart

document beschreven. Tenslotte vind je in de bijlagen [Specifieke MBG metagegevens voor Invertor](#) en [Specifieke MUG metagegevens voor Invertor](#) nog de specifiek voor Invertor gedefinieerde metagegevens.

## 1.4 Gebruikswijzer

Met de bouwstenen oftewel de modelelementen die in dit metamodel beschreven zijn is het bericht-technische deel van een BSM te maken. Om zo'n BSM te maken volstaat het veelal om het hoofdstuk [Metamodel Algemeen](#) en bijlage [Specifieke MBG metagegevens voor Invertor](#) door te nemen. Voor het maken van een UGM kun je terugvallen op de bijlagen [Metamodel UitwisselingsGegevens \(MUG\)](#) en [Specifieke MUG metagegevens voor Invertor](#).

De andere hoofdstukken kan je behandelen als naslagwerk, voor als er tijdens het modelleren vragen ontstaan. Neem vervolgens je favoriete modelleertool en ga aan de slag. Voor Enterprise Architect zijn er hulpmiddelen gemaakt waarmee de modelelementen aangemaakt kunnen worden door erop te klikken en bijvoorbeeld naar een diagram te slepen. Ook kan je daarmee valideren of je model correct het MBG volgt (wat automatisch gaat als je de hulpmiddelen gebruikt). Tot slot is het mogelijk om naar BSM's te kijken van organisaties die al een op het MBG gebaseerd BSM hebben gepubliceerd.

## 1.5 Wat is een BSM

Een BSM is een gegevensmodel waarin alle componenten van een bericht worden gemodelleerd, de bericht-technische objecten maar ook objecten die gaan over de inhoud van een domein. Het gaat dus om informatie over datgene wat uitgewisseld wordt maar ook om hoe het wordt uitgewisseld. De scope van een BSM is dus breder dan dat van het Metamodel voor Berichtgegevens (MBG).

In een BSM worden zoals gezegd ook objecten gemodelleerd die gaan over de inhoud van een domein. Van alle objecten wordt vastgelegd welke kenmerken/eigenschappen deze hebben, of deze kenmerken/eigenschappen in het beschouwde domein verplicht of optioneel zijn, hoe de objecten aan elkaar gerelateerd zijn, enzovoorts. Voor de vastlegging van deze objecten, kenmerken en eigenschappen wordt geen gebruik gemaakt van het MBG metamodel maar van het MUG metamodel. Wel is er in het MBG metamodel voorzien in een relatie met het MUG metamodel.

## 1.6 Typering van modellen en scope van deze standaard

Bij het modelleren van een domein zijn er een aantal beschouwingsniveaus, variërend van een zo getrouw mogelijke beschrijving van de betekenis en bedoeling van de woorden en termen die

mensen gebruiken als ze het ergens over hebben tot een specificatie van de wijze van registratie en uitwisseling van data. In het [\[MIM\]](#) vind je een uitgebreide beschrijving van deze niveau's.

Een BSM is een extensie op het logisch informatie- of gegevensmodel (niveau 3) en is dus een model op niveau 4. Het modelleert de bericht-technische componenten benodigd voor de uitwisseling van de met het logisch informatie- of gegevensmodel gedefinieerde informatie en koppelt deze daaraan. Waar de objecten, kenmerken en eigenschappen in een logisch informatie- of gegevensmodel techniek onafhankelijk zijn geldt dat voor de componenten in een BSM slechts deels. De componenten in een BSM kunnen nog op allerlei manieren geserialiseerd worden maar tegelijkertijd wordt er bij de modellering van de bericht-technische componenten al voorgesorteerd op de wijze waarop de specificatie zal worden gegenereerd, bijv. OAS. De bericht-technische componenten hebben dus elk een tegenhanger in de specificaties. Het component 'Padtype' bijvoorbeeld is specifiek in het leven geroepen voor het definiëren van een OAS specificatie. En hoewel niet uitgesloten mag worden dat dit component ook bij andersoortige specificaties een rol heeft of gaat krijgen, had het bij het modelleren van StUF berichten nooit een functie. Ook de enumeratie voor de naam van de 'EntiteitRelatie' hangt natuurlijk erg sterk samen met REST API's.

Een BSM kan, zoals hierboven reeds geschetst, geserialiseerd worden in verschillende soorten technische datamodellen en schema's (zoals XML, JSON of specifieke invullingen hiervan). Deze serialisaties zijn wat we in MBG niveau 4 noemen. De focus van het MBG ligt niet op dit niveau 4 en valt dan ook buiten scope. Wel staat het MBG een model-gedreven werkwijze voor waarbij de modellen van niveau 4 gegenereerd kunnen worden vanuit niveau 3. De aangegeven niveaus zijn vooral bedoeld om de scope van MBG duidelijk aan te geven, te weten niveau 3.

## 1.7 Wat is het metamodel voor Berichtstructuurmodellering

Een metamodel is een model van een model. Het definieert een verzameling van modelleerconstructies in de vorm van bouwstenen, oftewel modelelementen zoals een Berichttype, Entiteitrelatie en Padtype, met bijbehorende betekenis en met bijbehorende afspraken omtrent hoe deze toe te passen. Hiermee kan vervolgens een informatiemodel of in dit geval specifiek een BSM gemaakt worden. Het metamodel is daarmee de modelleertaal waarin een informatiemodel is uitgedrukt. Deze metataal beschrijft als het ware de grammatica en de syntax van de modelleertaal.

Vaak zie je dat het metamodel niet expliciet beschreven is en dat het metamodel een onderdeel van de domeinkennis is geworden. Bij domein overstijgende harmonisatie wordt het dan moeilijk om modellen met elkaar te vergelijken en op basis daarvan gegevens uit te wisselen. Bij een BSM is dat aspect echter van minder belang. Belangrijker is dat er duidelijkheid is over de betekenis van de modelelementen en de toepassing daarvan zodat het definiëren van een BSM eenvoudiger wordt en deze ook eenvoudiger door andere organisaties kan worden toegepast. Het is ook een voorwaarde voor de doorontwikkeling van dit metamodel, bijv. voor andere berichtsspecificatietalen.

Met de in een metamodel opgenomen set van modelleerconstructies worden informatiemodellen gemaakt. Door het schrijven van modelleertalen (zoals [\[UML\]](#)) in een metataal (zoals MOF) wordt gegarandeerd dat alle toepassingen van die talen op een standaard manier zijn opgebouwd en daardoor alom te begrijpen zijn. De metataal beschrijft als het ware de grammatica van de modelleertaal. Het metamodel in dit document is uitgewerkt voor modellering met UML.

## 1.8 Uitdrukken in UML

Zowel het metamodel als informatiemodellen zoals het BSM kunnen worden uitgedrukt in [\[UML\]](#). Aangezien het belangrijk is dat de lezer begrijpt wat we onder een informatiemodel en een metamodel verstaan verwijzen we naar paragraaf 1.8 van het [\[MIM\]](#). Daarin worden de modellen tevens in de juiste context geplaatst aan de hand van de vier lagen metamodel architectuur van de Object Management Group [\[OMG\]](#).

Verwar de daarin besproken vier lagen metamodel architectuur overigens niet met de in paragraaf 1.6 geschetste typering van modellen. Die typering van 4 lagen wijkt fundamenteel af van de in het [\[MIM\]](#) beschreven typering van 4 lagen (M0 t/m M3). De in paragraaf 1.6 geschetste lagen zijn geen instantie van de daarboven gelegen laag waar dat in het OMG model wel het geval is.

Het onderliggende Metamodel BerichtstructuurGegevens (MBG) is een voorbeeld van een model op de M2 laag. Een BSM is een instantie van het MBG en bevindt zich dan ook op de M1 laag.

## 1.9 Beheer

Het beheer van dit metamodel vindt plaats door ....

## 1.10 Normreferenties

#	Naam	Referentie
1.	Unified Modeling Language (UML)	<a href="#">[UML]</a>
2.	OMG Unified Modeling Language TM versie 2.5	<a href="#">[OMG]</a>

### NOOT: Versienummer normreferenties

De versies van de genoemde referenties zijn bewust niet opgenomen, zodat het mogelijk is om gebruik te maken van nieuwere versies van deze referenties. Wanneer er nieuwe versies uitkomen van de genoemde normreferenties dan wordt er gecheckt of MBG hier nog mee in lijn is. Wanneer dit niet zo is, dan wordt er aan gewerkt om dit weer met elkaar in lijn te brengen, zodat MBG 'upward compatible' is en blijft.



## 2. Metamodel Algemeen

Dit hoofdstuk beschrijft het metamodel in diagramvorm en in tekst.

Het metamodel beschrijft de modelelementen die worden gebruikt bij het maken van een BSM. Voorbeelden van modelelementen zijn: Berichttype, Entiteitrelatie en Padtype, maar denk ook datatypen of aan metagegevens. In de paragrafen hierna worden alle modelelementen beschreven en toegelicht.

- De eerste paragraaf bevat diagrammen, die kort een overzicht geven van de modelelementen die op metamodel niveau worden onderkend, alsmede hun onderlinge verhouding
- De paragrafen erna bevatten de beschrijvingen in tekst van de modelelementen die op metamodel niveau worden onderkend. Wanneer u liever de beschrijvingen eerst leest, kunt u ook met deze paragraaf starten. Dit begint bij 2.2.
- Paragraaf 2.5 bevat de metagegevens die worden bijgehouden over de modelelementen in een informatiemodel. Het MBG metamodel geeft aan welke metagegevens er zijn.

### **Uitgangspunten voor het metamodel**

- Elk modelelement heeft een naam en een eigen MBG metaclass, waaraan je het modelelement overal kan herkennen.
- De modelelementen worden eerst uitgelegd zonder een specifieke specificatietaal te gebruiken. Dit is zodat we hierna kunnen aangeven hoe je het modelelement uitdrukt per specifieke specificatietaal, te weten in UML of in W3C-specificatietechnieken.
- Een toolonafhankelijke beschrijving van het metamodel. Verder is er, omdat VNG Realisatie, Kadaster en Geonovum en veel andere organisaties Sparx EA gebruiken, aanvullend aangegeven hoe het metamodel in Enterprise Architect toegepast wordt. Hierdoor borgen we deze relatie. In een aparte bijlage zijn metagegevens beschreven die alleen op Invector van toepassing zijn.
- Uniforme toepassing van het metamodel in BSM's. Anders gezegd, uitbreiden mag, afwijken niet, maak voor hetzelfde doel geen alternatieve constructies.

*Toelichting metaclass* Alle modelelementen zijn een metaklasse in het metamodel. Hiermee wordt aangegeven dat het niet een klasse betreft in een informatiemodel, zoals de klasse Getfractie, maar dat het om de classificatie gaat dat de Getfractie een Getberichttype is, oftewel dat de klasse Getfractie van de metaklasse Getberichttype is. Vandaar de term metaclass.

De metaklassen worden ook gebruikt om aan te geven hoe deze zich verhouden tot de metaklassen van UML en W3C, in de volgende hoofdstukken.

Bij het maken van een BSM modelleer je in feite gewoon met de modelelementen, en geef je aan dat een 'Getfractie' een Getberichttype is en '/fracties' een Padtype.

Hierna volgen eerst diagrammen met de modelelementen, als overzicht. In de paragrafen erna wordt de betekenis van elk van deze modelelementen beschreven, met een definitie en een toelichting en een voorbeeld. Tot slot volgt een paragraaf met metadata die bijgehouden wordt, of kan worden, bij een modelelement.

## 2.1 Structuur metamodel

Deze paragraaf bevat een overzicht van het Metamodel voor BerichtGegevens, kortweg MBG, en geeft alle modelelementen weer. De beschrijving van de modelelementen staat in de volgende paragraaf.

De modelelementen zijn verdeeld over een tweetal diagrammen, die elk een eigen view op een deel van het metamodel tonen. Elk view toont een aantal van de modelelementen, inclusief hun onderlinge samenhang.

Alle views samen vormen het metamodel als geheel: \* [KERN](#), met de belangrijkste modelelementen in onderlinge samenhang; \* [GROEPERING](#) modelelementen.

Elk modelelement heeft een MBG metaclass met een naam. Hieraan is elk modelelement te herkennen in alle diagrammen en in de tekst en in elke specificatietaal die een uitdrukking is van dit metamodel.

Bij de modelelementen zijn in deze diagrammen geen beschrijvende kenmerken opgenomen, de metagegevens zoals naam, definitie enzovoorts. In de diagrammen in de bijlage zijn deze wel opgenomen.

### 2.1.1 Kern

View 1: De kern van een BSM. Deze bestaat uit de volgende modelelementen:

MBG metaclass

Getberichttype

Postberichttype

Putberichttype

Patchberichttype

Deleteberichttype

Padtype

Entiteitrelatie

Padrelatie

## MBG metaclass

Generalisatie

Interface

De betekenis van deze modelementen en de beschrijvingen ervan staat in [Objecten](#) en [Relaties](#).

In diagramvorm:

### Diagram toevoegen

Kern: Modelementen zonder Metagegevens

De verbindingen tussen de modelementen geven aan welke combinaties kunnen voorkomen op metamodelniveau, oftewel welke modelementen in een informatiemodel met elkaar gecombineerd kunnen worden. Bijvoorbeeld:

- Een Patchberichttype kan verbonden worden met een Padrelatie en deze Padrelatie kan weer verbonden worden met een Padtype. Dit geeft aan dat een Padrelatie een modelement is dat een Patchberichttype met een Padtype verbindt. Een Patchberichttype kan dus niet rechtstreeks verbonden worden met een Padtype.
- Een Deleteberichttype kan via een generalisatie verbonden worden met een Interface.

### 2.1.2 Groepering

View 2: Groepering. Deze bestaat uit de volgende modelementen:

## MBG metaclass

Koppelvlak

Domein

Bericht

De betekenis van deze modelementen en de beschrijvingen ervan staan in [Packages](#).

In diagramvorm:

### Diagram toevoegen

Diagram: groepering

## 2.2 Objecten

Diagram: [Kern](#)

### 2.2.1 Berichttypes

Het MBG metamodel kent een aantal berichttypes. Al deze berichttypes zijn afgeleid van de superclass Berichttype en overerven dus de metagegevens van die superclass. De class Berichttype is abstract en zal daarom zelf nooit in een BSM voorkomen. Alle berichten moeten één en niet meer dan één relatie van het type 'Padrelatie' en mogen één en niet meer dan één relatie van het type 'Generalisatie' hebben.

Elk hieronder toegelicht Berichttype is een abstractie van de werkelijkheid, nl. de berichtspecificatie. In die werkelijkheid heeft de keuze voor een berichttype gevolgen voor de wijze waarop dit bericht in de specificatie moet worden gedefinieerd. Die gevolgen hebben slechts ten dele invloed op de wijze waarop het bericht gemodelleerd moet worden in een BSM. Hieronder wordt per bericht aangegeven wat die gevolgen zijn.

#### 2.2.1.1 Getberichttype

##### **DEFINITIE : Getberichttype**

De typering van het berichttype waarmee resources kunnen worden opgehaald.

*Toelichting:*

Classes van dit type hebben verplicht 2 relaties van het type EntiteitRelatie hebben, één relatie met de naam 'request' en één met de naam 'response'. Alleen bij de eerste is het toegestaan een leeg Entiteittype aan de EntiteitRelatie te koppelen.

#### 2.2.1.2 Postberichttype

##### **DEFINITIE : Postberichttype**

De typering van het berichttype waarmee resources kunnen worden aangemaakt.

*Toelichting:*

Classes van dit type hebben verplicht 2 relaties van het type EntiteitRelatie, één relatie met de naam 'requestbody' en één met de naam 'response'. Een Entiteittype dat m.b.v. de eerste verbonden wordt met het Postberichttype bevat altijd minimaal 1 property. Een Entiteittype dat m.b.v. de tweede verbonden wordt met het Postberichttype mag leeg blijven.

### 2.2.1.3 Putberichttype

#### **DEFINITIE : Putberichttype**

De typering van het berichttype waarmee bestaande resources kunnen worden vervangen of nieuwe kunnen worden aangemaakt.

#### *Toelichting:*

Classes van dit type hebben verplicht 2 relaties van het type EntiteitRelatie, één met de naam 'requestbody' en één met de naam 'response'. Een Entiteittype dat m.b.v. de eerste verbonden wordt met het Postberichttype bevat altijd minimaal 1 property. Een Entiteittype dat m.b.v. de tweede verbonden wordt met het Postberichttype mag leeg blijven.

### 2.2.1.4 Patchberichttype

#### **DEFINITIE : Patchberichttype**

De typering van het berichttype waarmee properties van resources kunnen worden aangepast.

#### *Toelichting:*

Classes van dit type hebben verplicht 2 relaties van het type EntiteitRelatie, één met de naam 'requestbody' en één met de naam 'response'. Een Entiteittype dat m.b.v. de eerste verbonden wordt met het Postberichttype bevat altijd minimaal 1 property. Een Entiteittype dat m.b.v. de tweede verbonden wordt met het Postberichttype mag leeg blijven.

### 2.2.1.5 Deleteberichttype

#### **DEFINITIE : Deleteberichttype**

De typering van het berichttype waarmee resources kunnen worden verwijderd.

#### *Toelichting:*

Classes van dit type hebben verplicht 2 relaties van het type EntiteitRelatie, één relatie met de naam 'request' en één met de naam 'response'. Alleen bij de tweede is het toegestaan een leeg Entiteittype aan de EntiteitRelatie te koppelen.

## 2.2.2 Padtype

### DEFINITIE : Padtype

Het Padtype definieert een relatief path naar een individueel endpoint.

#### *Toelichting:*

Tezamen met een Server Object url (het deel van de url dat definieert waar de API wordt gehost) vormt het path een uniek adres. De naam van een Padtype start altijd met een slash (/) en mag templates bevatten. Dit zijn strings voorafgegaan door een slash (/), startend met een open accolade ({} en eindigend met een sluit accolade (}). Binnen een model mogen niet meerdere Padtypes voorkomen met dezelfde naam. Meerdere berichten mogen gebruik maken van hetzelfde Padtype zolang deze maar niet van hetzelfde Berichttype zijn.

Classes van dit type hebben 1 verplichte relatie van het type 'Padrelatie'.

## 2.2.3 Interface

### DEFINITIE : Interface

Een Interface definieert een berichttype op een hoger abstractie niveau.

#### *Toelichting:*

Een interface definieert eigenschappen (metagegevens, attribuutsoorten en/of relaties) die een groep van berichttypes gemeen hebben. Een interface kan d.m.v. een generalisatie relatie gekoppeld worden aan een specifiek berichttype waarmee deze de eigenschappen van die interface overerft.

## 2.3 Relaties

Verbanden met betekenis, die gelegd zijn tussen modelementen van het type Berichttype naar het type Entiteittype of van het type Berichttype naar het type Padtype.

Diagram: [Kern](#)

### 2.3.1 Generalisatie

#### DEFINITIE : Generalisatie tussen Berichttype en Interface

De typering van het hiërarchische verband tussen een meer generiek en een meer specifiek modelement van hetzelfde soort, waarbij het meer specifieke modelement eigenschappen van het meer generieke modelement overerft. Dit verband is alleen gedefinieerd voor generalisaties tussen Berichttypes en Interfaces.

#### *Toelichting:*

De generalisatie relatie geeft aan dat bepaalde eigenschappen van een interface (metagegevens) ook gelden voor de gerelateerde berichttypen. 2.3.2 EntiteitRelatie

**DEFINITIE : EntiteitRelatie**

Modelelement waarmee verschillende relaties tussen de berichtdefinitie en de resource waar het bericht betrekking op heeft kunnen worden gemodelleerd.

*Toelichting:*

Berichttypes danken hun bestaan aan een specifieke resource, de resource waarop de methode, die de berichttype vertegenwoordigd, moet worden uitgevoerd. Het modelleren van de relatie tussen het Berichttype en het Entiteittype dat de resource vertegenwoordigd gebeurt met de metaklasse EntiteitRelatie. Een Berichttype kan, afhankelijk van het type, 1 of 2 EntiteitRelaties hebben. Een EntiteitRelatie kan maar met 1 Berichttype en 1 Entiteittype verbonden zijn maar een Entiteittype kan wel meerdere EntiteitRelaties hebben. De naam van deze relatie kan alleen 'request', 'requestbody' of 'response' zijn.

### 2.3.2 PadRelatie

**DEFINITIE : PadRelatie**

Modelelement waarmee de relatie tussen de berichtdefinitie en het Padtype dat het bericht identificeert kan worden gemodelleerd.

*Toelichting:*

Elk Berichttype moet aan een relatief path worden gekoppeld. Het modelleren van de relatie tussen een Berichttype en het PadType gebeurt m.b.v. de metaklasse PadRelatie. Elk berichttype kan maar aan 1 PadType worden gekoppeld maar meerdere Berichttypes kunnen wel hetzelfde PadType hebben. Deze relatie heeft altijd de naam 'pad'.

### 2.4 Packages

**DEFINITIE : Package**

Een package is een benoemde en begrensde verzameling/groepering van modelementen.

Diagram: [Groepering](#)

Het MBG metamodel kent een aantal packages. Al deze packages zijn afgeleid van de superclass Package. De class Package is abstract en zal daarom zelf nooit in een BSM voorkomen.

De verschillende modelementen van het type package zijn:

- Koppelvlak
- Domein

- Bericht

In een Koppelvlak package komen Bericht packages voor en kunnen Domein packages opgenomen worden.

In een Bericht package mogen de volgende MBG-elementen worden opgenomen:

- Getberichttype
- Postberichttype
- Putberichttype
- Patchberichttype
- Deleteberichttype
- Padtype

en alle modelementen van het Metamodel Uitwisselings Gegevens (MUG).

In een Domein package mogen alleen modelementen van het Metamodel Uitwisselings Gegevens (MUG) worden opgenomen.

In beide gevallen is de volgorde daarbij niet van belang.

De drie package-typen worden hieronder uitgelegd.

### 2.2.3 Koppelvlak

#### **DEFINITIE : Koppelvlak**

De groepering van alle modelementen waaruit het Koppelvlak is opgebouwd. Het Koppelvlak als geheel.

*Toelichting:* Het Koppelvlak is een package, te weten het hoofdpakage van het koppelvlak, waaronder alle subpackages die een koppelvlak beschrijven, zoals Domein en Bericht, vallen. Het koppelvlak wordt verder beschreven met metadata, zoals de serialisatie waarin het koppelvlak moet worden uitgedrukt ('json' of 'hal+json') en de naam van het koppelvlak.

### 2.2.4 Domein

Een koppelvlak kan optioneel een aantal Domein packages hebben als subpackage, waarbij aangegeven wordt dat deze de modellering van de gegevens van het domein bevatten.

#### **DEFINITIE : Domein**

Een groepering van constructies die een semantisch samenhangend gedeelte van een



koppelvlak beschrijven welke met de in het koppelvlak beschreven berichten kunnen worden uitgewisseld.

*Toelichting:* Een Domein package bevat die modelementen die met het koppelvlak uitgewisseld worden, zoals het Entiteitstype 'Persoon' en het Entiteitstype 'Adres' en impliciet ook relaties als 'gehuwdMet'. In ieder geval alleen modelementen uit het Metamodel Uitwisselings Gegevens (MUG). Deze modelementen kunnen 1 op 1 overgenomen zijn van een Uitwisselingsgegevensmodel (UGM) maar kunnen ook bewerkingen daarvan zijn.

### 2.2.5 Bericht

Een koppelvlak bevat een of meer Bericht packages als subpackage.

#### **DEFINITIE : Bericht**

Een groepering van constructies die de modelementen van een of meerdere gerelateerde berichten clustert.

*Toelichting:* Berichten worden doorgaans geclusterd op basis van een in een UGM gemodelleerd Entiteitstype. Alle berichten die dezelfde Entiteitstype als top level delen kunnen bij elkaar in één Bericht package worden geplaatst. Met 'dezelfde Entiteitstype' bedoelen we hier niet 'dezelfde instantie van een entiteitstype klasse'. In een bericht mogen dus meerdere entiteitstype klassen van dezelfde Entiteitstype worden opgenomen zolang de naam maar uniek blijft binnen het koppelvlak.

Een Bericht package bevat i.i.g. de voor de berichten benodigde modelementen van een set van gerelateerde berichten maar kan ook alle modelementen bevatten die in een 'Domein' package mogen worden opgenomen.

## 2.3 Specificatie metagegevens

Bij de modelementen in een koppelvlak kunnen metagegevens, zoals 'naam' of 'berichtcode' van een modelement, worden bijgehouden. Dit zijn geen eigenschappen van een object en worden daarom niet als bijvoorbeeld een EntiteitRelatie van een Berichttype gemodelleerd.

In de volgende paragrafen worden de metagegevens in tekst beschreven. Bij elk metagegeven is de definitie opgenomen, een toelichting en de toepassing ervan bij modelementen en soms een voorbeeld.

### 2.3.1 Specificatie metagegevens Packages

We onderkennen een aantal specifieke metagegevens op het niveau van de packages. Deze staan beschreven in deze paragraaf.

#### 2.3.1.1 Metagegeven: Naam

##### **DEFINITIE : Naam**

Korte identificatie van een koppelvlak of beschrijving van een package als definitie van diens functie of inhoud.

*Toelichting:* Korte identificerende naam van een Koppelvlak package of definitie van een Domein of Bericht package zodat duidelijk is waartoe het dient en/of wat voor inhoud het heeft of kan worden opgenomen.

*Bijvoorbeeld:* Enumeraties.

*Toepassing:* Alle packages (verplicht)

#### 2.3.1.2 Metagegeven: beheerder-email

##### **DEFINITIE : beheerder-email**

Het e-Mail adres van de persoon of organisatie die het koppelvlak beheert.

*Toelichting:*

Het e-Mailadres dat in de berichtenspecificatie en/of de documentatie daarvan kan worden opgenomen zodat voor de gebruikers van de specificatie duidelijk is hoe ze middels e-Mail contact op kunnen nemen met de beherende persoon of organisatie.

*Toepassing:* Koppelvlak (optioneel)

#### 2.3.1.3 Metagegeven: Koppelvlak-naam

##### **DEFINITIE : Koppelvlak-naam**

Volledige naam van een koppelvlak.

*Toelichting:* Bedoelt om kernachtig de functie/domein van het koppelvlak dat gemodelleerd wordt aan te geven.

*Bijvoorbeeld:* ‘Open Raads- en StatenInformatie’ of ‘Regels bij activiteiten’.

*Toepassing:* Koppelvlak (verplicht)

#### 2.3.1.4 Metagegeven: *project\_url*

**DEFINITIE : project\_url**

Url van de bij het koppelvlak horende project repository.

*Toelichting:* De url van de repository waarin de al dan niet met het koppelvlakmodel te genereren of gegenereerde specificatie en documentatie wordt opgeslagen. Dit kan bijv. een GitHub repository zijn maar ook een GitLab of andersoortige repository.

*Bijvoorbeeld:* <https://github.com/VNG-Realisatie/ODS-Open-Raadsinformatie>

*Toepassing:* Koppelvlak (optioneel)

#### 2.3.1.5 Metagegeven: *release*

**DEFINITIE : release**

Datum waarop begonnen is met het realiseren van de betreffende versie van het koppelvlak.

*Toelichting:* De datum wordt uitgedrukt in het volgende formaat ‘jjjjmmdd’.

*Bijvoorbeeld:* 20230404

*Toepassing:* Koppelvlak (verplicht)

#### 2.3.1.6 Metagegeven: *Serialisatie*

**DEFINITIE : Serialisatie**

Definieert de vorm waarnaar het koppelvlak geserialiseerd moet worden.

*Toelichting:* BSM's op basis van het MBG worden opgesteld teneinde omgezet te worden naar een technische specificatie. De vorm waarin de technische specificatie wordt uitgedrukt is de serialisatie. Voor REST OAS API's gaat het om de serialisatietypes, ‘json’ en ‘hal+json’. In principe zijn ook andere serialisaties mogelijk zoals ‘xml’. Dit metagegeven definieert de gewenste serialisatievorm.

*Toepassing:* Koppelvlak (optioneel)

### 2.3.2 Specificatie metagegevens modelelementen

De specifieke metagegevens op het niveau van de modelelementen waarmee een BSM wordt samengesteld staan beschreven in deze paragraaf.

Zo is er de naam van het modelelement, bijvoorbeeld het Getberichttype met als naam 'Getagendapunt' en een bijbehorende berichtcode, bijvoorbeeld Gr01.

Welke metagegevens verplicht zijn per modelelement en welke niet staat beschreven in het diagram in Metagegevens per modelelement. Dit diagram is een onderdeel van deze specificatie.

Elk modelelement kent een eigen set van metagegevens, die bepaalde aspecten van het modelelement specificeren. Metagegevens kunnen dus verplicht of optioneel zijn. Zo is een berichtcode altijd verplicht voor de berichttype modelelementen.

#### 2.3.2.1 Metagegeven: Naam

##### **DEFINITIE : Naam**

De naam van een modelelement.

*Toelichting:* Een korte beschrijving die het modelelement benoemt. In de berichttype modelelementen en 'Interface' heeft dit metagegeven alleen een functie bij het kunnen interpreteren van het model. De in de modelelementen 'EntiteitRelatie' en 'Padtype' gedefinieerde waarden voor dit metagegeven 'Naam' komen terug in de berichtspecificatie'. Bij de modelelementen 'EntiteitRelatie' en 'PadRelatie' wordt de naam gekozen uit een beperkte set van waarden. In het geval van de laatste heeft het zelfs altijd slechts 1 mogelijke waarde, te weten 'pad'.

Binnen een BSM mogen niet meerdere Berichttype, Padtype, Domein en Bericht modelelementen met dezelfde naam voorkomen.

*Bijvoorbeeld:* '/stemmingen/{stemmingsidentificatie}' is de naam van het modelelement 'Padtype' en 'Getactiviteit' is de naam van het modelelement 'Getberichttype'.

*Toepassing:* alle modelelementen behalve Generalisatie (verplicht).

#### 2.3.2.2 Metagegeven: berichtcode

##### **DEFINITIE : berichtcode**

Code ter typering van een bericht.

*Toelichting:* De code bestaat uit een limitatieve lijst van waarden en typeert een bericht. Aan de hand van deze code worden door Invertor aan properties/kenmerken toegekend aan een bericht in een berichtspecificatie standaard.

*Toepassing:* Alle van Berichttype afgeleide modelementen (verplicht).

#### 2.3.2.3 Metagegeven: tag

##### **DEFINITIE : tag**

Naam waarmee een aantal berichten kan worden gegroepeerd.

*Toelichting:* Dit metagegeven heeft een ordenende functie. De wijze van ordenen is niet voorgeschreven maar over het algemeen krijgen berichten die een (deel van een pad) met elkaar gemeen hebben dezelfde waarde voor dit metagegeven.

*Toepassing:* Alle van Berichttype afgeleide modelementen (optioneel).

#### 2.3.2.4 Metagegeven: servicename

##### **DEFINITIE : servicename**

OperationId van een OAS3 service.

*Toelichting:* Unieke identificatie van een operation in een OAS berichtspecificatie. Een operation is een combinatie van een path en een HTTP method. Binnen een OAS berichtspecificatie is het niet toegestaan om een path en een HTTP method meerdere keren met elkaar te combineren.

*Bijvoorbeeld:* 'getUserById'.

*Toepassing:* Alle van Berichttype afgeleide modelementen (verplicht).

#### 2.3.2.5 Metagegeven: custom\_path\_facet

##### **DEFINITIE : custom\_path\_facet**

Niet af te leiden deel van het path.

*Toelichting:* Het deel van het path dat niet uit het berichtenmodel kan worden afgeleid. Om het path toch te kunnen verifiëren tegen het model wordt daar eerst de custom\_path\_facet uit verwijderd. Deze tv mag '/' (slashes) bevatten maar niet aan het begin en eind.

*Bijvoorbeeld:* ‘actuelestatus’ en ‘nawgegevens’ in ‘/zaken/{id}/actuelestatus’ resp. ‘/ingeschrevennatuurlijkpersonen/{burgerservicenummer}/nawgegevens’.

*Toepassing:* Padtype (optioneel).

#### 2.3.2.6 Metagegeven: Positie

##### **DEFINITIE : Positie**

Positie die een construct inneemt t.o.v. constructs op hetzelfde niveau.

*Toelichting:* De positie van de construct binnen producten waarin deze opeenvolging een rol speelt. Dit metagegeven had in het verleden een functie maar is inmiddels obsolete.

*Toepassing:* Entiteitrelatie (optioneel).

### 2.3.3 Specificatie metagegevens modelelement bindingen

Bindingen geven aan hoe modelelementen met elkaar verbonden kunnen en mogen worden.

Voorbeeld: de binding tussen een Berichttype en een Padtype, om aan te geven dat een Berichttype aan een pad moet worden gekoppeld. Een voorbeeld van wat niet mag is de verbinding van een Padtype aan een EntiteitRelatie.

Deze metagegevens zijn alleen nodig voor de binding van modelelementen aan elkaar en zijn vrijwel altijd een onderdeel van een modelleertaal (waarmee een BSM gemaakt kan worden).

De bindingen zijn ook in diagram vorm te lezen aan het begin van dit hoofdstuk, in Diagrammen van modelelementen.

#### 2.3.3.1 Metagegeven: heeft padrelatie

Verkorte schrijfwijze: PadRelatie

##### **DEFINITIE : heeft padrelatie**

De binding van een Berichttype aan een Padtype middels een Padrelatie.

*Toelichting:* Een Berichttype heeft een relatie naar een Padtype.

*Toepassing:* Alle Berichttype modelelementen en Padtype.

### 2.3.3.2 Metagegeven: heeft entiteitrelatie

Verkorte schrijfwijze: EntiteitRelatie

**DEFINITIE : heeft entiteitrelatie**

De binding van een Berichttype aan een Entiteittype uit het MUG middels een EntiteitRelatie.

*Toelichting:* Een Berichttype heeft een relatie naar een Entiteittype, de top level entiteit.

*Toepassing:* Alle Entiteittype modelementen uit het MUG en alle Berichttype modelementen.

### 2.3.3.3 Metagegeven: verwijst naar supertype

Verkorte schrijfwijze: supertype

**DEFINITIE : verwijst naar supertype**

De binding van een supertype aan een subtype middels een generalisatie.

*Toelichting:* Een subtype verwijst met een generalisatie naar een supertype en subtype verwijst met een generalisatie naar een supertype.

*Toepassing:* Alle Berichttype en Interface modelementen.

## 2.4 Toegestane waarden voor (bepaalde) metagegevens

### 2.4.1 Waardebereik metagegevens

Een aantal metagegevens hebben als datatype `CharacterString`. Aanvullend geldt: \* Voor datums dat deze het volgende patroon volgen: `jjjjmmdd`

Voor de volgende metagegevens geldt een specifiek waardebereik.

Metagegeven	Waardenbereik
berichtcode	Gr01, Gr02, Gc01, Po01, Pu01, Pa01, De01
Grouping	resource, collection
Naam (bij EntiteitRelatie)	resource, request, requestbody
Naam (bij PadRelatie)	Pad
Is afgeleid (bij Koppelvlak)	Ja, Nee

Metagegeven	Waardenbereik
Is afgeleid (bij andere elementen)	Ja, Nee, Zie package
Serialisatie	json, hal+json
Supplier-project	UGM
toelichting opnemen vanaf	SIM, UGM, BSM

## 2.4.2 Defaultwaarden voor metagegevens modelementen

Er zijn metagegevens die een defaultwaarde hebben. Het is echter niet nodig om deze defaultwaarde expliciet aan te geven in het informatiemodel. De default staat hier aangegeven. Alleen wanneer er afgeweken wordt van deze default wordt dit in een BSM aangegeven.

Metagegeven	Defaultwaarde
Is afgeleid (bij Koppelvlak)	Ja
Is afgeleid (bij andere elementen)	Ja
Serialisatie	hal+json
toelichting opnemen vanaf	SIM

Opmerking met betrekking tot de kardinaliteit van relaties: deze staat niet in de tabel. Deze kennen geen defaultwaarde. De kardinaliteit aan de doel kant moet altijd worden aangegeven. De kardinaliteit aan de bron/eigenaar kant van een relatie is optioneel om in te vullen, wanneer er niets is ingevuld dan wordt er niets over de kardinaliteit gezegd en kent deze geen default waarde (in de praktijk betekent dit dat een kardinaliteit aan de bron kant als 0..\* geïmplementeerd wordt).

## 3. 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

[beheerder-email](#) §2.3.1.2

[Bericht](#) §2.2.5

[berichtcode](#) §2.3.2.2

[custom\\_path\\_facet](#) §2.3.2.5



[Deleteberichttype](#) §2.2.1.5

[Domein](#) §2.2.4

[EntiteitRelatie](#) §2.2.2

[Generalisatie tussen Berichttype en Interface](#)  
§2.2.2

[Getberichttype](#) §2.2.1.1

[heeft entiteitrelatie](#) §2.3.3.2

[heeft padrelatie](#) §2.3.3.1

[Interface](#) §2.2.2

[Koppelvlak](#) §2.2.3

[Koppelvlak-naam](#) §2.3.1.3

Naam

[definitie van](#) §2.3.1.1

[definitie van](#) §2.3.2.1

[Package](#) §2.2.2

[PadRelatie](#) §2.2.2

[Padtype](#) §2.2.2

[Patchberichttype](#) §2.2.1.4

[Positie](#) §2.3.2.6

[Postberichttype](#) §2.2.1.2

[project\\_url](#) §2.3.1.4

[Putberichttype](#) §2.2.1.3

[release](#) §2.3.1.5

[Serialisatie](#) §2.3.1.6

[servicename](#) §2.3.2.4

[tag](#) §2.3.2.3

[verwijst naar supertype](#) §2.3.3.3

## A.2 Begrippen gedefinieerd door verwijzing

## B. Referenties

### B.1 Normatieve referenties

#### [MIM]

*MIM - Metamodel Informatie Modelling*. Geonovum. 17 Februari 2022. URL:  
<https://docs.geostandaarden.nl/mim/def-st-mim-20220217/>

#### [OMG]

*Object Management Group Unified Modeling Language TM*. Object Management Group (OMG). BG-FINAL. URL: <http://www.omg.org/spec/UML/2.5>

#### [UML]

*Unified Modeling Language (UML)*.. URL: <http://uml.org>