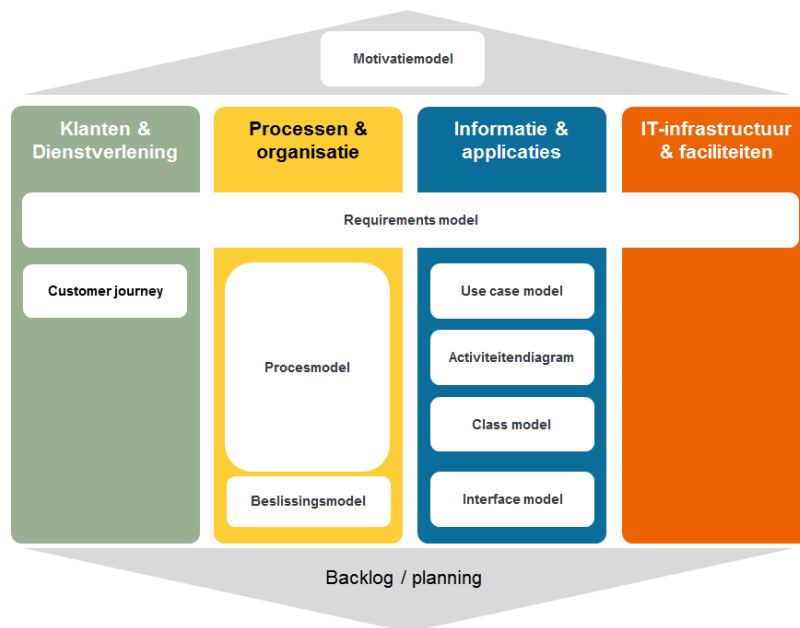


# STIJLGIDS

## Novius Bedrijfsontwerp



Plaats: Langbroek  
Datum: 6 maart 2018

Versie: 2.0  
Status: Definitief



# Inhoudsopgave

<b>Inhoudsopgave</b>	<b>2</b>
<b>1 Ontwikkeling Stijlgids</b>	<b>3</b>
1.1 Licentie	3
1.2 Wijzigingshistorie	3
<b>2 Inleiding</b>	<b>4</b>
2.1 Modellen, views en ontwerpdocumentatie	4
2.2 Gebruik van deze stijlgids	4
<b>3 Het Novius Bedrijfsontwerp Raamwerk</b>	<b>5</b>
3.1.1 Uitleg Novius Bedrijfsontwerp Raamwerk	5
3.1.2 Modellen	5
3.1.3 Metamodel bedrijfsontwerp	6
<b>4 Modellen</b>	<b>8</b>
4.1 <i>Motivatiemodel</i>	8
4.1.1 Metamodel	8
4.1.2 Standaard views	9
4.1.3 Business requirement motivatie view	10
4.2 <i>Requirements model</i>	11
4.2.1 Metamodel	11
4.2.2 Requirement decompositie view	13
4.2.3 Schrijfstijl: formulering van requirements	13
4.3 <i>Procesmodel</i>	14
4.3.1 Metamodel decompositie processen	14
4.3.2 Metamodel proces	15
4.3.3 Ketenproces view	20
4.3.4 Bedrijfsprocesview	20
4.3.5 Werkprocesview	21
4.4 <i>Relatie processen en requirements</i>	22
4.5 <i>Beslissingsmodel</i>	23
4.5.1 Relatie beslissingsmodel en procesmodel	24
4.6 <i>Applicatiefunctionaliteit</i>	25
4.7 <i>Use case model</i>	25
4.7.1 Metamodel	25
4.7.2 Relatie use case model en procesmodel	27
4.8 <i>Activiteiten diagram</i>	28
4.8.1 Relatie activiteitendiagram en user stories resp. systeem requirements	29
4.9 <i>Domein model</i>	31
4.9.1 Domein model	32
4.10 <i>Interfacemodel</i>	34
4.10.1 Requestbericht	34
4.10.2 Response bericht	35
<b>5 Aansluiting op Novius Architectuur Raamwerk</b>	<b>38</b>
5.1 <i>Trace relaties</i>	40
5.1.1 Motivatiemodel	40
5.1.2 Requirementsmodel	40
5.1.3 Procesmodel	40
5.1.4 Functioneel ontwerp	42

# 1 Ontwikkeling Stijlgids

Deze stijlgids is ontwikkeld door het competence centre Bedrijfsontwerp van Novius. Deze versie van de stijlgids beschrijft de correcte toepassing van modellen voor bedrijfsontwerp.

## 1.1 Licentie

Het document wordt beschikbaar gesteld onder de Creative Commons licentie BY-NC-SA 4.0. Dit houdt in dat iedereen vrij is om:

- **het werk te delen** — te kopiëren, te verspreiden en door te geven via elk medium of bestandsformaat
- **het werk te bewerken** — te remixen, te veranderen en afgeleide werken te maken

De licentiegever kan deze toestemming niet intrekken zolang aan de licentievoorwaarden voldaan wordt.

Licentievoorwaarden:

- **Naamsvermelding** — De gebruiker dient de maker van het werk te vermelden (zijnde Novius Bedrijfsontwerp), een link naar de licentie te plaatsen en aan te geven of het werk veranderd is. Je mag dat op redelijke wijze doen, maar niet zodanig dat de indruk gewekt wordt dat de licentiegever instemt met je werk of je gebruik van het werk.
- **NietCommercieel** — Je mag het werk niet gebruiken voor commerciële doeleinden.
- **GelijkDelen** — Als je het werk hebt geremixt, veranderd, of op het werk hebt voortgebouwd, moet je het veranderde materiaal verspreiden onder dezelfde licentie als het originele werk

## 1.2 Wijzigingshistorie

Versie	Datum	Auteurs	Toelichting
1.0	1-10-2015	Thea Kamps, Evelijn Tjarks, Peter Buijs	Novius BPMN styleguide
2.0	07-03-2018	Peter Buijs, Lars Krewinkel	Uitgebreid met: <ul style="list-style-type: none"><li>- Motivatiemodel</li><li>- Requirementsmodel</li><li>- Beslissingsmodel</li><li>- Use case model</li><li>- Activiteiten diagram</li><li>- Domein model</li><li>- Interface model</li><li>- Relatie met architectuur</li></ul> Naam gewijzigd in Stijlgids bedrijfsontwerp

## 2 Inleiding

### 2.1 Modellen, views en ontwerpdocumentatie

Deze stijlguide heeft als doel een toegankelijke en consistente schrijfwijze voor te schrijven voor het vastleggen van modellen en de daarop gebaseerde views volgens het Novius bedrijfsontwerp raamwerk (NBR). Deze modellen en views kunnen worden toegepast in agile-, waterval en hybride omgevingen. Bovendien sluit het NBR aan op het Novius Architectuur Raamwerk.

Een model bestaat uit concepten en de relaties tussen deze concepten en met de omgeving. Deze concepten en relaties vormen tezamen het gehele bedrijfsontwerp model. Het Novius bedrijfsontwerp model bestaat uit meerdere deelmodellen:

- Motivatiemodel
- Requirementsmodel
- Customer journey model [volgt in een volgende versie van de stijlguide]
- Procesmodel
- Use case model
- Class model
- Interface model

Een view bestaat uit een specifieke selectie van concepten en relaties. Een view dient een bepaald doel / belang die relevant is voor een (groep) stakeholder(s). Bedrijfsontwerp kent de volgende veelgebruikte standaard views:

- Business requirement realisatie view
- Requirement decompositie view
- Ketenproces view
- Bedrijfsproces view
- Werkproces view
- Use case view
- Domein view
- Interface view

Een ontwerp is een samenhangende specificatie van een concept bestaande uit één of meer views en toelichtingen. Bedrijfsontwerp kent de volgende veelgebruikte standaard ontwerpen:

- Requirementsontwerp
- Ketenprocesontwerp
- Bedrijfsprocesontwerp
- Werkprocesontwerp
- Use case ontwerp
- Interface ontwerp

### 2.2 Gebruik van deze stijlguide

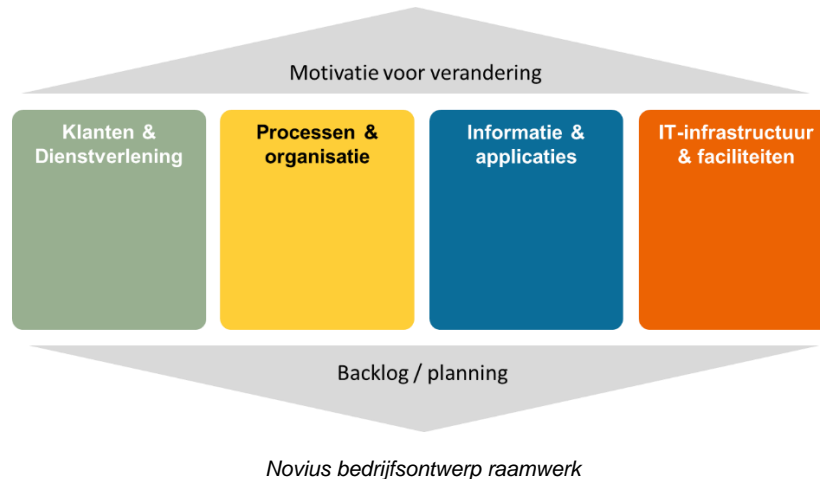
Deze stijlguide bevat een raamwerk met een aantal praktische modellen die toegepast kunnen worden om te ontdekken op welke manier een verandering het best gerealiseerd kan worden. Zowel in agile als waterval werkende organisaties wordt de gewenste verandering besproken en vastgelegd in modellen: grafisch in bijvoorbeeld een procesmodel of tekstueel in bijvoorbeeld een requirementsmodel.

**Deze stijlguide is een hulpmiddel voor organisaties om te bepalen met welke modellen men wil werken én welke afspraken hierover gemaakt worden. Het is dan ook aan te bevelen een eigen versie van deze stijlguide te maken met die modellen en views die het meest waardevol zijn.**

## 3 Het Novius Bedrijfsontwerp Raamwerk

### 3.1.1 Uitleg Novius Bedrijfsontwerp Raamwerk

Het Novius bedrijfsontwerp raamwerk (NBR) bestaat uit een aantal samenhangende modellen. Het NBR is gebaseerd op het Novius Business Transformation Framework dat tevens dient als basis voor het Novius Architectuur Raamwerk.



Een goed bedrijfsontwerp begint met een **motivering** voor de verandering. Wat is de werkelijke behoefte, wat is of zijn de businessdoelen die men wil bereiken, welke stakeholders zijn relevant.

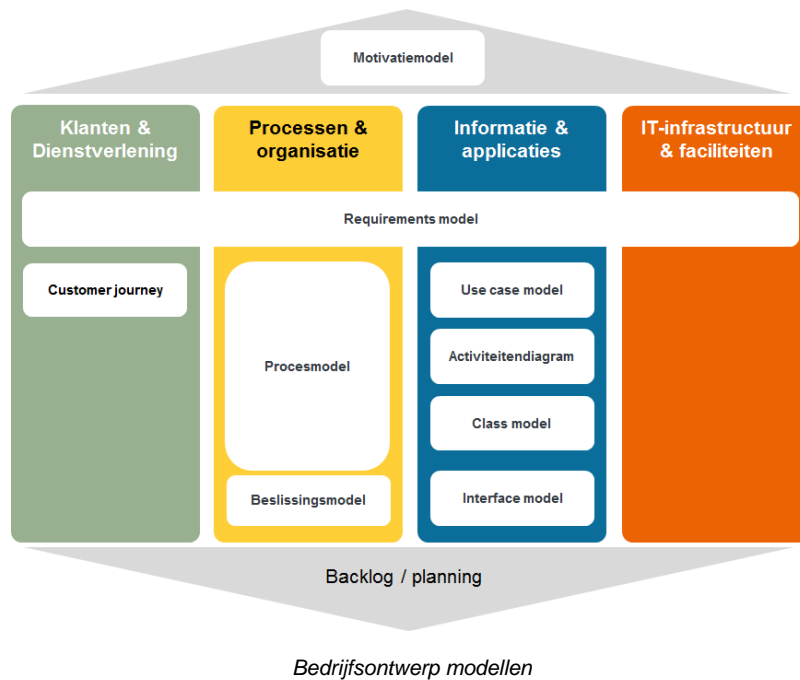
Vervolgens kan de bedrijfsontwerper met behulp van verschillende modellen een analyse en/of ontwerp maken van de huidige resp. gewenste situatie. De bedrijfsontwerper kan modellen uitwerken voor een viertal bedrijfsvoeringaspecten:

- **Klanten & dienstverlening**
- **Processen & organisatie**
- **Informatie & applicaties**
- **IT-infrastructuur & faciliteiten**

De veranderingen worden vervolgens opgenomen in een backlog of planning, zodat de veranderingen gerealiseerd kunnen worden. In agile werkende organisaties is dit een zeer iteratief proces, waarbij ook ervaringen uit de realisatiesprints input zijn om de modellen en backlog aan te passen. In waterval werkende organisaties worden de planning en de modellen gewijzigd via een changeprocedure. De backlog / planning is verder geen onderdeel van deze stijlgids.

### 3.1.2 Modellen

De meest gebruikte modellen zijn opgenomen in het NBR. In de toekomst kunnen hier nieuwe modellen aan toegevoegd worden.

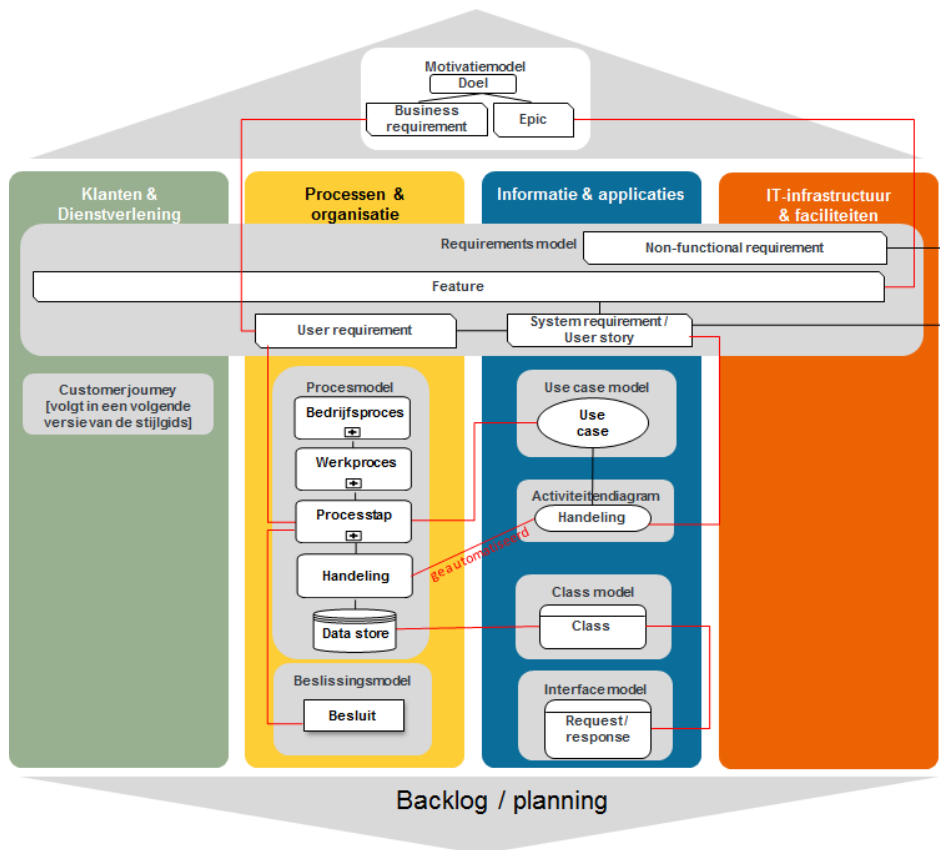


Voor het uitwerken van de verschillende modellen wordt gebruik gemaakt van internationale specificatie standaarden.

Model	Standaard taal en gehanteerde versie in deze stijlguide
Motivatiemodel	Archimate 2.1
Requirementsmodel	Archimate 2.1
Customer journey	[volgt in een volgende versie van de stijlguide]
Procesmodel	Business Process Model Notation (BPMN 2.0)
Beslissingsmodel	Decision Making Notation (DMN 1.1)
Use case model	Unified modelling language (UML 2.5)
Class model	Unified modelling language (UML 2.5)
Activiteitendiagram	Unified modelling language (UML 2.5)
Interface model	Unified modelling language (UML 2.5)

### 3.1.3 Metamodel bedrijfsontwerp

Ieder model bestaat uit een set concepten en relatie, zoals hieronder weergegeven. Tussen de modellen, of preciezer tussen de concepten uit de verschillende modellen, bestaan relaties. De relaties tussen de modellen wordt gelegd door een link (trace) te leggen tussen de concepten uit de verschillende modellen. Door deze modellen goed op elkaar aan te sluiten ontstaat een coherente inrichting van de organisatie die voor verschillende doeleinden gebruikt kan worden



*Bedrijfsontwerp modellen en concepten*

Zoals aangegeven in het metamodel hangen de verschillende modellen sterk met elkaar samen. Bij het uitwerken van de verschillende modellen is het daarom goed deze relaties expliciet te maken zodat de samenhang bewaakt wordt. Ook voor stakeholders is verhelderend als de link tussen de verschillende modellen uitgelegd wordt.

## 4 Modellen

De verschillende modellen worden hier nader toegelicht. Per model is beschreven:

- Wat en waarom van het model
- Het metamodel
- De definitie van de gebruikte concepten
- Het visuele element om het concept te modelleren
- De aard van de relatie tussen de concepten
- Een of meer standaard views

### 4.1 Motivatiemodel

#### Wat

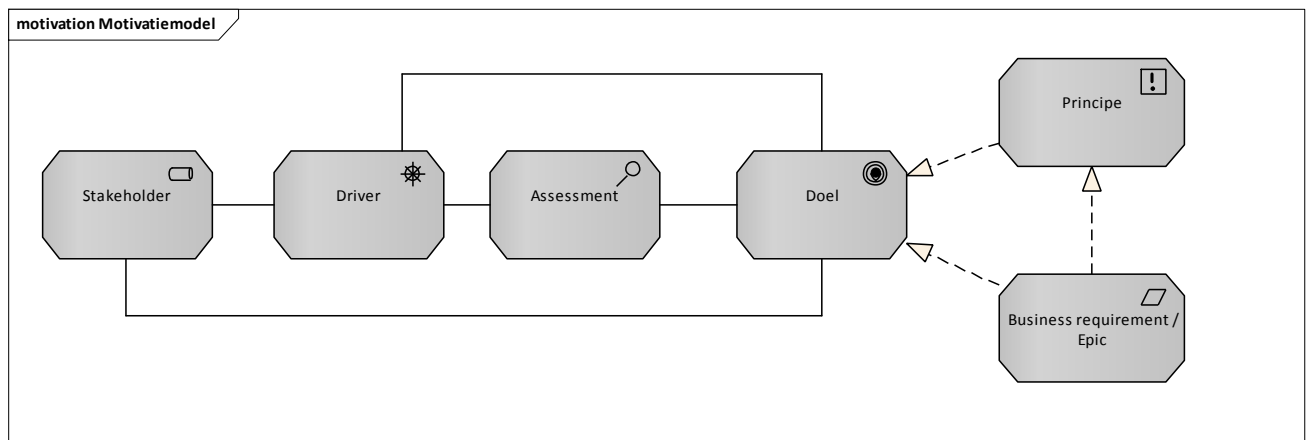
Het motivatiemodel structureert de motivatie waarom veranderingen uitgevoerd moeten worden. Waarom wil de organisatie veranderen, wat zijn de drijfveren (drivers) van relevante stakeholders, welke doelen willen de stakeholders bereiken? Vervolgens kan een relatie worden gelegd hoe de concrete veranderingen (business requirements of epics) helpen in het realiseren van de doelen.

#### Waarom

Door de motivatie van de stakeholder(s) voor de verandering expliciet te maken ontstaat een basis waaraan de ontwerpen en implementatie keuzes getoetst kunnen worden.

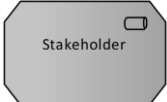
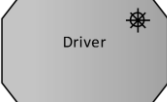
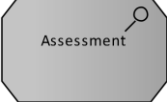
#### 4.1.1 Metamodel

Het hier gehanteerde motivatiemodel is gebaseerd op Archimate 2.1.


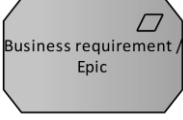
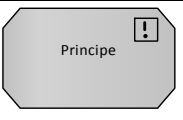


Motivatiemodel


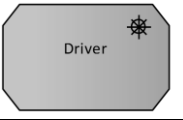

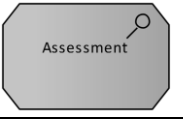
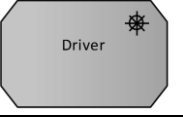
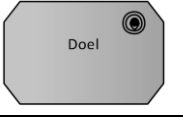
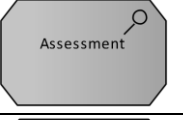

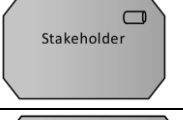
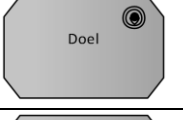
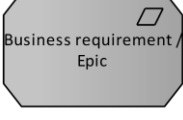

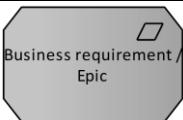
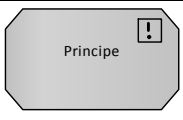
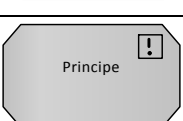
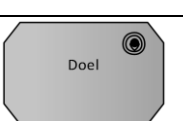
#### Toelichting concepten

Concept	Novius bedrijfsontwerp definitie
	Een stakeholder is de rol van een individu, team of organisatie die belang heeft bij de uitkomst van de verandering.
	Een driver is een interne of externe beweegreden die een verandering in de organisatie initieert, motiveert of voedt.
	Een assessment is een analyse resultaat m.b.t. de huidige situatie voor driver.



	Een doel is een concretisering van de richting of een gewenste resultaat die een stakeholder(s) wil bereiken.
	Een <i>business requirements</i> of <i>epic</i> is een omschrijving van een behoefte of middel die gerealiseerd moet worden om een doel(en) te bereiken.
	Een principe is een algemeen geldige (gedrags)regel, 'zo doen wij dat hier'. Het is een normatieve eigenschap van alle 'systemen' in een bepaalde context, of de wijze waarop deze worden gerealiseerd. Een principe is niet project of businessdomein specifiek, maar algemeen geldend.

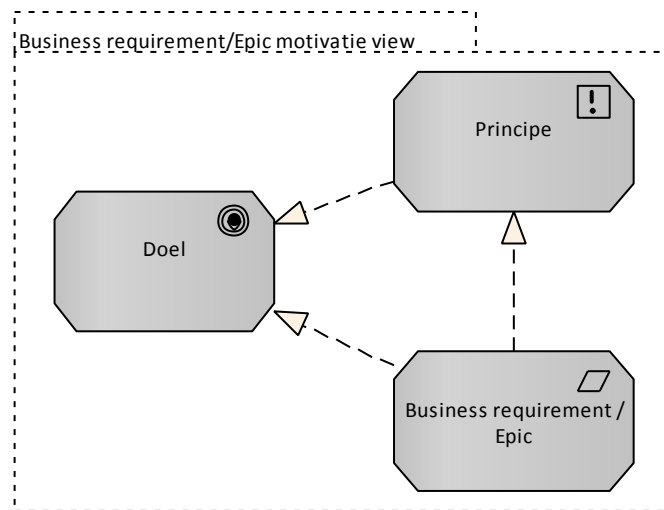
### Toelichting relaties

Van concept	Relatie	Naar concept	Toelichting
	Associatie _____		Een <i>stakeholder</i> heeft een <i>driver</i> . Een <i>driver</i> hoort bij een <i>stakeholder</i> .
	Associatie _____		Een <i>driver</i> heeft een <i>assessment</i> . Een <i>assessment</i> hoort bij een <i>driver</i> .
	Associatie _____		Een <i>driver</i> heeft een <i>doel</i> . Een <i>doel</i> hoort bij een <i>driver</i> .
	Associatie _____		Een <i>assessment</i> heeft een <i>doel</i> . Een <i>doel</i> hoort bij een <i>assessment</i> .
	Associatie _____		Een <i>stakeholder</i> heeft een <i>doel</i> . Een <i>doel</i> hoort bij een <i>stakeholder</i> .
	Realiseert ----->		Een <i>business requirement</i> of <i>epic</i> realiseert een <i>doel</i> . Een <i>doel</i> wordt gerealiseerd door een <i>business requirement</i> of <i>epic</i> .
	Realiseert ----->		Een <i>business requirement</i> of <i>epic</i> realiseert een <i>principe</i> . Een <i>principe</i> wordt gerealiseerd door een <i>business requirement</i> of <i>epic</i> .
	Realiseert ----->		Een <i>principe</i> realiseert een <i>doel</i> . Een <i>doel</i> wordt gerealiseerd door een <i>principe</i> .

#### 4.1.2 Standaard views

De motivatieview kan bestaan uit alle concepten en relaties uit het motivatiemodel of een deel daarvan. Afhankelijk van de doelgroep kan gekozen worden om een deel van de concepten te laten zien en andere weg te laten. Veel voorkomend is overigens dat het model niet als view (plaatje) wordt weergegeven maar in (gestructureerde) tekst wordt uitgeschreven.

### 4.1.3 Business requirement motivatie view



De business requirement of epic motivatie view geeft inzicht in de beoogde bijdrage van een business requirement of epic aan één of meer doelen. Het geeft de rationale weer voor de beoogde veranderingen zoals omschreven in het business requirement of epi. Tevens geeft deze view inzicht aan welke principe(s) een business requirement of epic moet voldoen en welke doelen de principes helpen realiseren.

Deze view wordt ook wel in een tabelstructuur weergegeven.

## 4.2 Requirements model

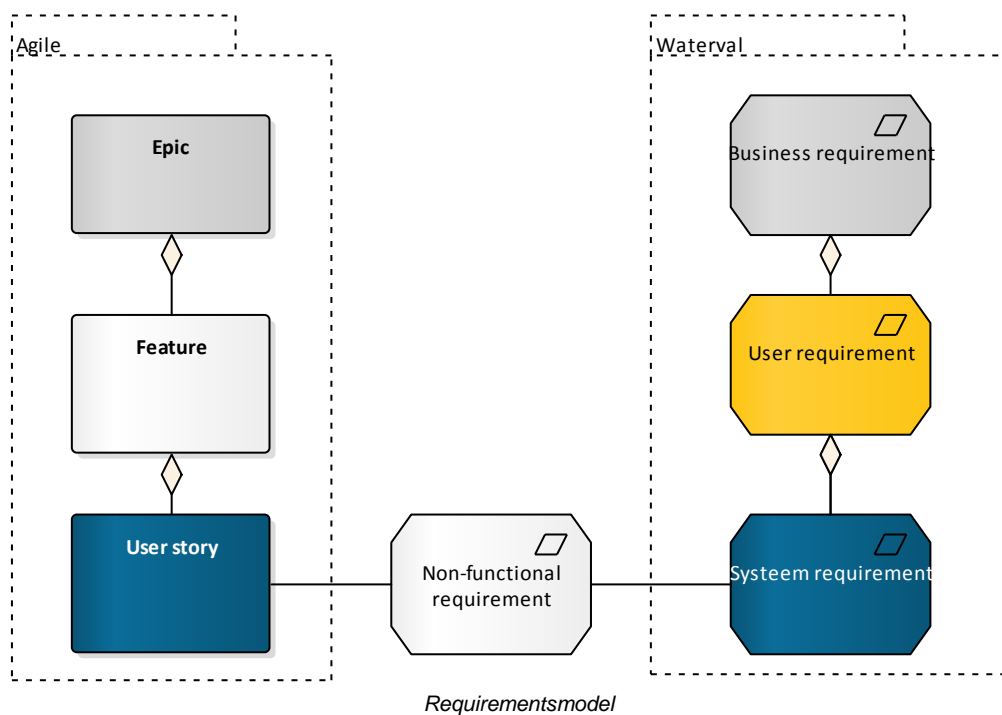
### Wat

Het requirementsmodel modelleert de veranderbehoefte van verschillende stakeholders in de vorm van doelen en verschillende typen requirements. Van grof (business requirement of epic) naar fijn (systeem requirement of user stories) kan een analyse gemaakt worden van de globale needs/behoefte van de organisatie naar gedetailleerde en specifieke eisen t.a.v. systemen.

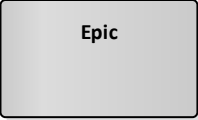
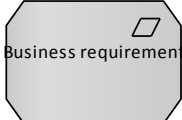
### Waarom


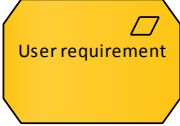
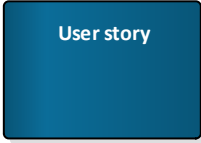
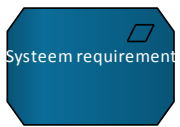
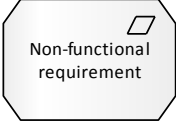
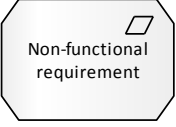
Het modelleren van requirements geeft een decompositie van de veranderbehoefte. Het maakt expliciet en traceerbaar welke requirements nodig zijn om gestelde doelen te bereiken.

### 4.2.1 Metamodel

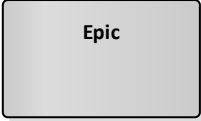
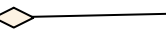


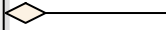

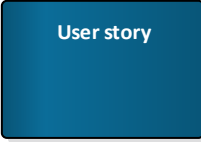

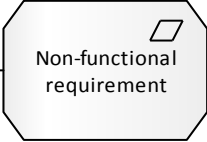
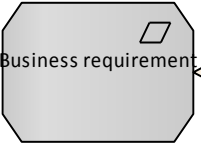
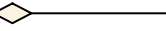
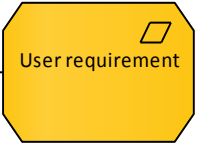
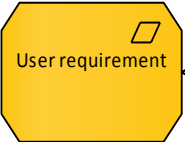
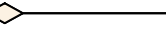
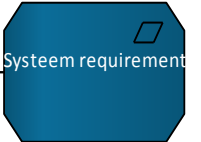


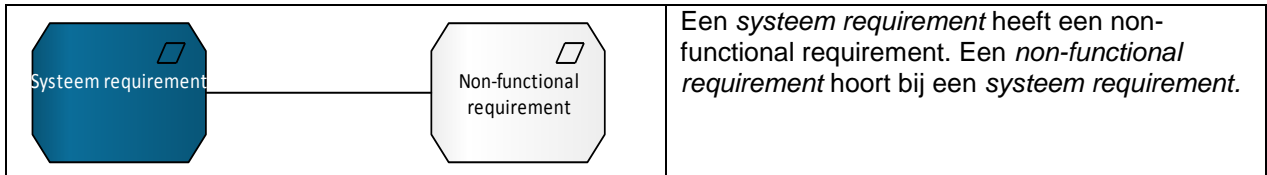
### Toelichting concepten

Concept agile	Novius bedrijfsontwerp definitie	Concept waterval	Novius bedrijfsontwerp definitie
	<p>Een <i>epic</i> is een beschrijving van een behoefte of middel die gerealiseerd moet worden (om een doel(en) te bereiken).</p> <p>Een epic vraagt nog nadere analyse en een uitwerking van een minimal viable product. Een epic kan verder uitgesplitst worden in features en user stories. Een epic kan niet in 1 sprint gerealiseerd worden en kan meerdere projecten of teams omvatten.</p>		<p>Een <i>business requirements</i> is een omschrijving van een behoefte of middel die gerealiseerd moet worden (om een doel(en) te bereiken).</p>

	removed to optimize the team's release time.		
	Een feature is een <i>service/functie</i> die voorziet in een behoefte van een stakeholder. Een feature kan impact hebben op meerdere componenten en kan verder worden uitgesplitst in user stories. Een feature wordt in meerdere sprints gerealiseerd.		Een <i>user requirement</i> is een behoefte van een medewerker of klant om een activiteit uit te kunnen voeren (voor het realiseren van een business requirement).
	Een <i>user story</i> is een korte beschrijving van een klein stukje gewenste functionaliteit, vanuit de gebruiker beschreven en heeft impact één component.  Een user story kan in één sprint worden gerealiseerd.		Een <i>systeem requirement</i> is gedrag dat een systeem systeem/product moet vertonen om te voorzien in een behoefte van een medewerker of klant (voor het realiseren van een user requirement)
	Een non-functional requirement is een kwaliteitseis of beperkingen waar het 'systeem' aan moet voldoen.		Een non-functional requirement is een kwaliteitseis of beperkingen waar het 'systeem' aan moet voldoen.

### Toelichting relaties

Van concept	Relatie	Naar concept	Toelichting
			Een <i>feature</i> hoort bij een <i>epic</i> . Een <i>epic</i> bestaat uit één of meer <i>features</i> .
			Een <i>user story</i> hoort bij een <i>feature</i> . Een <i>feature</i> heeft één of meer <i>user stories</i> .
			Een <i>user story</i> heeft een non-functional requirement.. Een <i>non-functional requirement</i> hoort bij een <i>user story</i> .
			Een <i>user requirement</i> hoort bij een <i>business requirement</i> . Een <i>business requirement</i> bestaat uit 1 of meer <i>user requirements</i> ..
			Een <i>systeem requirement</i> hoort bij een <i>user requirement</i> . Een <i>user requirement</i> heeft één of meer <i>systeem requirements</i> .

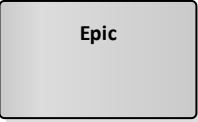


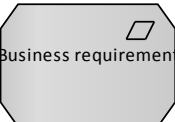
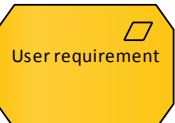
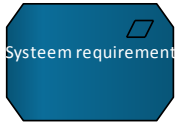
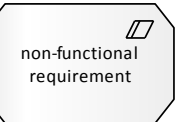


#### 4.2.2 Requirement decompositie view

De requirement decompositie view geeft inzicht in de hiërarchie van requirements.. Deze view wordt ook wel in een tabelstructuur weergegeven. In deze view kun je een deel van de requirements opnemen, bijv. alle features en user stories die bij één epic horen.

#### 4.2.3 Schrijfstijl: formulering van requirements

Een requirement is een in tekst geschreven behoefte die veelal een standaard zinsopbouw (sjabloon) volgt. Een tekstsjabloon is een standaardstructuur voor een individueel requirement en helpt de requirements eenduidig en compleet te formuleren. Ga hier verstandig mee om en gebruik het met name om op het juiste detailniveau het requirement te specificeren.

Concept	Novius bedrijfsontwerp standaard tekstsjabloon	
 <p>Epic</p>	<p>Epic</p> <ul style="list-style-type: none"> <li>• De &lt;stakeholder&gt; wil &lt;iets verbeteren&gt;</li> <li>• De &lt;stakeholder&gt; wil &lt;iets extra's kunnen&gt;</li> </ul> <p>NB: het komt ook wel voor dat epics worden geschreven met het user story tekstsjabloon.</p>	
 <p>Feature</p>	<p>Feature</p> <ul style="list-style-type: none"> <li>• &lt;zelfstandig naamwoord&gt; &lt;werkwoord&gt;</li> </ul>	
 <p>User story</p>	<p>User story</p> <ul style="list-style-type: none"> <li>• Als een &lt;gebruikersrol&gt; wil ik &lt;gewenste functionaliteit&gt; zodat ik &lt;waarde voor de business&gt;.</li> </ul>	
 <p>Business requirement</p>	<p>Business requirement</p> <p>De &lt;stakeholder&gt; wil &lt;iets verbeteren&gt;</p> <p>De &lt;stakeholder&gt; wil &lt;iets extra's kunnen&gt;</p>	
 <p>User requirement</p>	<p>User requirement eenvoudig tekstsjabloon:</p> <p>De &lt;actor&gt; kan &lt;iets&gt; &lt;werkwoordvorm&gt;</p>	<p>Uitgebreid tekstsjabloon (IREB: Pohl&amp;Rupp)</p> <ol style="list-style-type: none"> <li>1.&lt;wanneer/onder welke conditie&gt;</li> <li>2. &lt;zal, kan of mag&gt;</li> <li>3. &lt;het systeem of de actor&gt; een</li> <li>4. a. &lt;actie&gt;</li> <li style="padding-left: 20px;">b. de mogelijkheid bieden om &lt;actie&gt;</li> <li style="padding-left: 20px;">c. in staat zijn om &lt;actie&gt;</li> <li>5. &lt;object&gt;</li> <li>6. &lt;details over het object&gt;</li> </ol>
 <p>System requirement</p>	<p>System requirement eenvoudig tekstsjabloon:</p> <ul style="list-style-type: none"> <li>• Het systeem moet &lt;bepaald gedrag vertonen&gt;</li> </ul>	
 <p>non-functional requirement</p>	<p>Non-functional requirement</p> <ul style="list-style-type: none"> <li>• Het systeem moet &lt;bepaalde kwaliteiten bezitten&gt;</li> <li>• De &lt;gebruikersrol&gt; moet &lt;iets goed kunnen&gt;</li> </ul> <p>NB: binnen agile worden non-functional requirements ook wel opgenomen in de definition of done van een user story.</p>	

### 4.3 Procesmodel

#### Wat

Een procesmodel is een overzicht van activiteiten die in samenhang plaatsvinden, gericht op een beoogd resultaat dat toegevoegde waarde heeft voor een externe of interne klant.

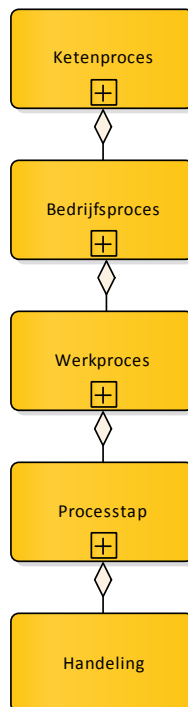
#### Waarom

Een procesmodel geeft inzicht in de huidige of gewenste situatie met als doel het proces:

- doelmatig en efficiënt uit te kunnen voeren
- te monitoren en bij te kunnen sturen
- te analyseren en te verbeteren (w.o. inzet van ICT)
- risico's en beheersmaatregelen te treffen



#### 4.3.1 Metamodel decompositie processen




Het metamodel geeft inzicht in de verschillende niveaus waarop processen gemodelleerd kunnen worden.



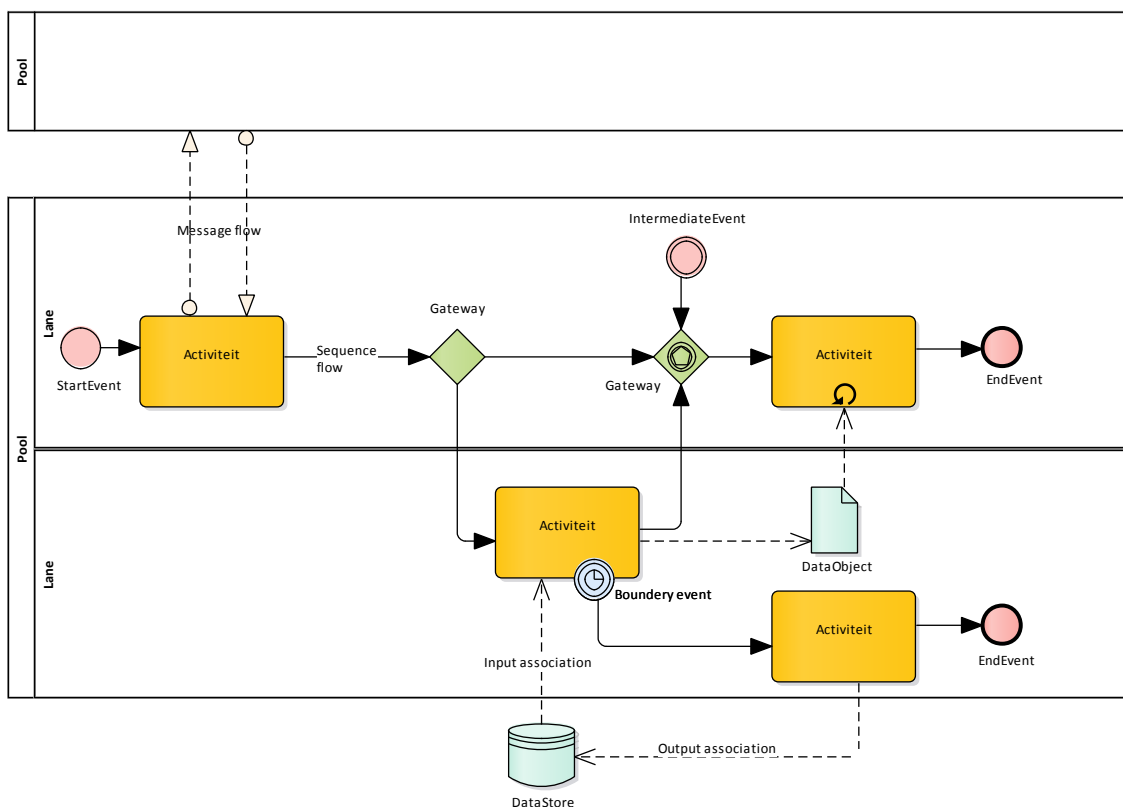
*Decompositie processen*

#### Toelichting concepten

Concept	Novius bedrijfsontwerp definitie
	<p>Een ketenproces is een geordende reeks bedrijfsprocessen die services aan elkaar leveren die in combinatie een dienst zijn voor de klant.</p> <p>Het ketenproces wordt met een Task met sub-proces marker gemodelleerd.</p>
	<p>Een bedrijfsproces bestaat uit een geordende reeks werkprocessen die binnen één organisatie wordt uitgevoerd met als doel een (combinatie van) bedrijfsservices te leveren aan een externe klant</p> <p>Het bedrijfsproces wordt met een Task met sub-proces marker gemodelleerd.</p>

 <p>Werkproces</p>	<p>Een werkproces bestaat uit een geordende reeks processtappen die binnen één bedrijfsfunctie worden uitgevoerd met als doel een specifieke bijdrage (prestatie) te leveren aan een bedrijfsservice die (uiteindelijk) zal worden geleverd aan een externe klant.</p>
 <p>Processtap</p>	<p>Het werkproces wordt met een Task met sub-proces marker gemodelleerd.</p> <p>Een processtap bestaat uit een geordende reeks handelingen die ononderbroken wordt uitgevoerd door één mens of systeem (eenheid van tijd, plaats en handelen).</p>
 <p>Handeling</p>	<p>Een processtap wordt gemodelleerd met een Task <b>zonder</b> sub-proces marker indien er <b>geen</b> handelingen worden gemodelleerd. Worden er wel handelingen gemodelleerd dan wordt de processtap met een Task met sub-proces marker gemodelleerd.</p> <p>Een handeling is het kleinst mogelijke eenheid van werk, uitgevoerd door één persoon of systeem op één plek op één moment.</p> <p>Handelingen kunnen in BPMN op dezelfde wijze gemodelleerd worden als de processtappen. Handelingen worden in de meeste procesmodellen echter niet gemodelleerd, maar in werkinstructies uitgeschreven. Wanneer inzicht in de opeenvolgende handelingen nodig is voor automatiseringsdoeleinden, dan wordt dit vaak een apart activiteitenmodel in UML gemodelleerd (zie elders in deze stijlgijs).</p> <p>Het handeling wordt altijd met een Task <b>zonder</b> sub-proces marker gemodelleerd.</p>

#### 4.3.2 Metamodel proces












Metamodel proces

Zoals in het metamodel proces weergegeven, maakt BPMN gebruik van verschillende concepten om processen te modelleren. De concepten in BPMN kunnen in een aantal categorieën worden onderverdeeld; flow object, swimlanes, connectors en artifacts. In deze stijlguides hebben wij de meest gebruikte flowobject, connectoren en artifacts opgenomen voor administratieve en dienstverlenende processen.

### Flow objects

#### Activities

Een Task beschrijft de activiteit die uitgevoerd wordt. Met task types kan de aard van de activiteit gespecificeerd worden en met activity markers kan het uitvoergedrag van activiteit gespecificeerd worden..









Concept	Benaming	Omschrijving	Conventies
<b>Task types</b>			
	Activiteit	Een ketenproces, bedrijfsproces, werkproces, processtap of handeling,	Een activiteit bestaat altijd uit een werkwoord + zelfstandig naamwoord en eventueel een bijvoeglijk naamwoord
	Gebruiker stap / handeling	Processtap of handeling die wordt uitgevoerd door een gebruiker met ondersteuning van automatisering.	
	Automatische stap / handeling	Processtap of handeling die volledig geautomatiseerd wordt uitgevoerd.	
	Manuele stap / handeling	Processtap of handeling die wordt uitgevoerd zonder gebruik van automatisering.	
	Business Rule stap	Processtap waarbij een (complexe) business rule (*) wordt uitgevoerd.	
	Ontvangst stap of handeling	Processtap of handeling waarin informatie ontvangen wordt.	
	Verzend stap of handeling	Processtap of handeling waarin informatie verzonden wordt.	
<b>Activity markers</b>			
	Sub-proces marker	Met een sub-process marker wordt aangegeven dat er een verdieping van het proces gemodelleerd is.	
	Loop marker	Met een loop marker wordt aangegeven dat de activiteit met herhaling wordt uitgevoerd, totdat aan een eindconditie is voldaan.	



(\*) een business rule (bedrijfsregel) is een richtlijn over de actie die genomen moeten worden wanneer aan bepaalde condities voldaan wordt.

### Events

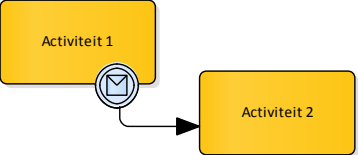
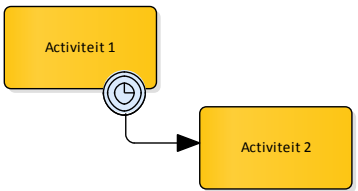
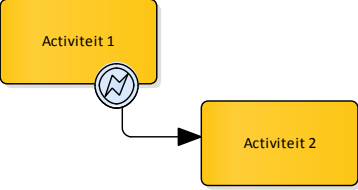
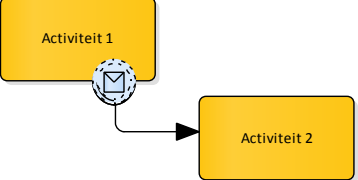
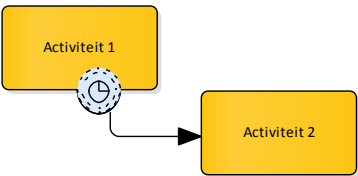
Events geven de startcondities en eindresultaten van een proces weer.

Concept	Benaming	Omschrijving
	Start Event	Start van het proces anders dan door ontvangst van een bericht / informatie of een timer.
	Message Start Event	Er wordt een bericht / informatie ontvangen waardoor het proces start.
	Timer Start Event	Het proces start op een specifiek punt in de tijd, tijdspanne of time-out.
	Intermediate message event (throw)	Gedurende het proces wordt er een bericht verzonden. Het proces gaat verder na versturen van het bericht..
	Intermediate message event (catch)	Het proces staat te wachten, zodra het bericht ontvangen wordt gaat het proces verder.
	Timer Intermediate Event	Het proces staat te wachten en gaat verder op een specifiek punt in de tijd, tijdspanne of time-out.
	Error Intermediate Event	Het initiëren van een foutafhandeling tijdens het proces, wanneer het proces verstoord raakt.
	End Event	Einde van het proces.
	Message End Event	Het proces eindigt met het versturen van een bericht / informatie.
	Terminate End Event	Alle activiteiten/stappen in het proces worden direct gestaakt.

### Boundary events




Boundary events geven aan dat er gedurende de uitvoer van een activiteit iets gebeurt waardoor er een alternatieve afhandeling plaats vindt. Boundary events kunnen interrupting zijn; de activiteit wordt gestopt. Boundary events kunnen ook non-interrupting zijn; de activiteit gaat verder.

Concept	Benaming	Omschrijving
---------	----------	--------------

	Boundary message - interrupting	Activiteit 1 wordt <b>tussentijds</b> gestopt en stuurt een bericht waardoor activiteit 2 start.
	Boundary timer - interrupting	Activiteit 1 wordt <b>tussentijds</b> gestopt doordat een timer afgaat voor einde van de activiteit 1. Activiteit 2 wordt gestart.
	Boundary error - interrupting	Activiteit 1 wordt <b>tussentijds</b> gestopt doordat er een (bekende) fout optreedt. Activiteit 2 wordt gestart.
	Boundary message – non interrupting	<b>Tijdens de uitvoer</b> van activiteit 1 wordt een bericht verstuurd waardoor activiteit 2 gestart wordt. Activiteit 1 gaat door totdat deze is afgerond.
	Boundary timer – non interrupting	<b>Tijdens de uitvoer</b> van activiteit 1 wordt activiteit 2 gestart doordat een timer tijdens de activiteit afgaat. Activiteit 1 gaat door totdat deze is afgerond.


### Gateways

Gateways worden gebruikt om aan te geven dat paden splitsen of samenkomen afhankelijk van bepaalde condities.

Concept	Benaming	Omschrijving
	Exclusive Gateway (XOR)	Op basis van informatie uit de voorafgaande activiteit wordt één van de <b>uitgaande</b> paden gevolgd, of het proces gaat verder met de opvolgende activiteit wanneer één van de <b>inkomende</b> paden gereed is.
	Parallel Gateway (AND)	Er worden twee of meer paden tegelijk uitgevoerd, of het proces gaat verder wanneer <b>alle inkomende</b> paden gereed zijn.
	Event (Based) Gateway	Het proces gaat verder zodra één van de uitgaande events optreedt.

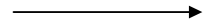
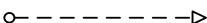
### Swimlanes

Swimlanes worden gebruikt om aan te geven wie verantwoordelijkheden hebben in het proces.

Concept	Benaming	Omschrijving
	Pool	Een pool is een organisatie, afdeling, proces of een partij buiten de organisatie die verantwoordelijk is voor de activiteiten die binnen de pool plaatsvinden.
	Lane	Een lane is een rol, systeem of afdeling binnen de organisatie die verantwoordelijk is voor de uitvoer van de in de lane geplaatste activiteiten, het nemen van een beslissing en waar de events plaatsvinden.



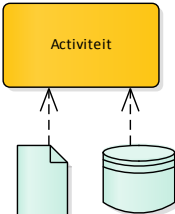
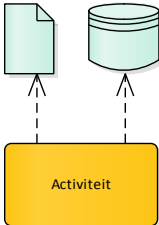
### Connectors

Connectors worden gebruikt om flow objects aan elkaar te verbinden.

Concept/element	Benaming	Omschrijving
	Sequence Flow	Deze connector geeft de volgorde aan van de activiteiten in het proces en binnen één pool.
	Message Flow	Deze connector geeft de communicatie weer tussen twee pools.

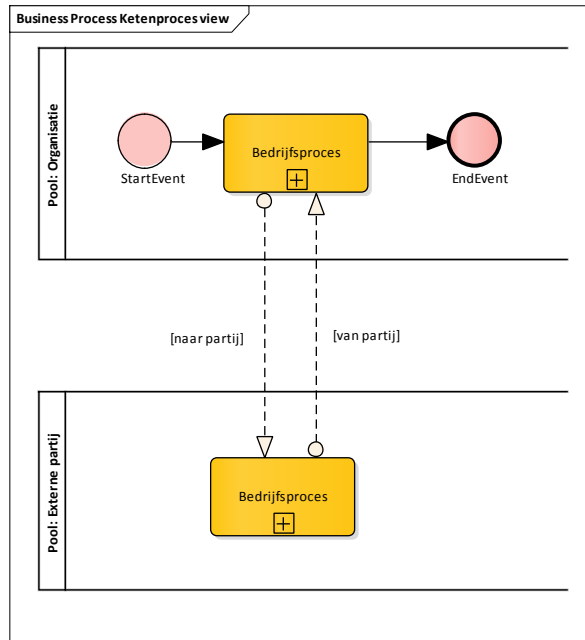
### Artifacts

Artifacts worden gebruikt om extra informatie toe te voegen aan het model.

Concept	Benaming	Omschrijving
	Data Object	Een data object is informatie die door het proces stroomt op dragers zoals documenten, brieven, e-mails en tijdens het proces wordt ontvangen, gemaakt of gewijzigd.
	Data Store	Een data store is een plaats waar het proces data kan schrijven en lezen. De data blijft ook na beëindigen van het proces bestaan. Data stores worden altijd <b>buiten</b> de pool geplaatst.
	Data input connector	Data object of data store die geraadpleegd wordt om een activiteit uit te voeren. De data object of data store wordt met een data input connector verbonden aan de activiteit.
	Data output connector	Data object of data store die aangemaakt/gewijzigd/verwijderd kan worden tijdens de activiteit. De data object of data store wordt met een data output connector verbonden aan de activiteit.

### 4.3.3 Ketenproces view

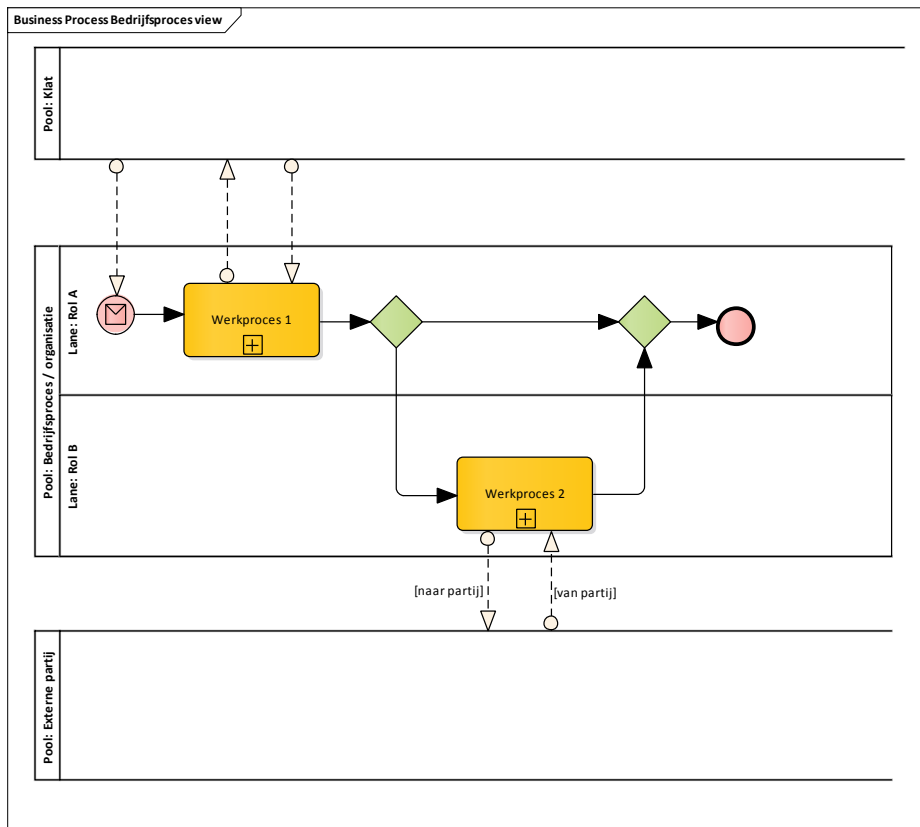
De ketenproces view geeft inzicht in de bedrijfsprocessen en services die de eigen organisatie uitwisselt met derde partijen die in combinatie een product/dienst leveren aan een klant. Deze view bevat alleen bedrijfsprocessen. De services worden weergegeven met message flows tussen de pools.



View Ketenproces in BPMN

### 4.3.4 Bedrijfsprocesview

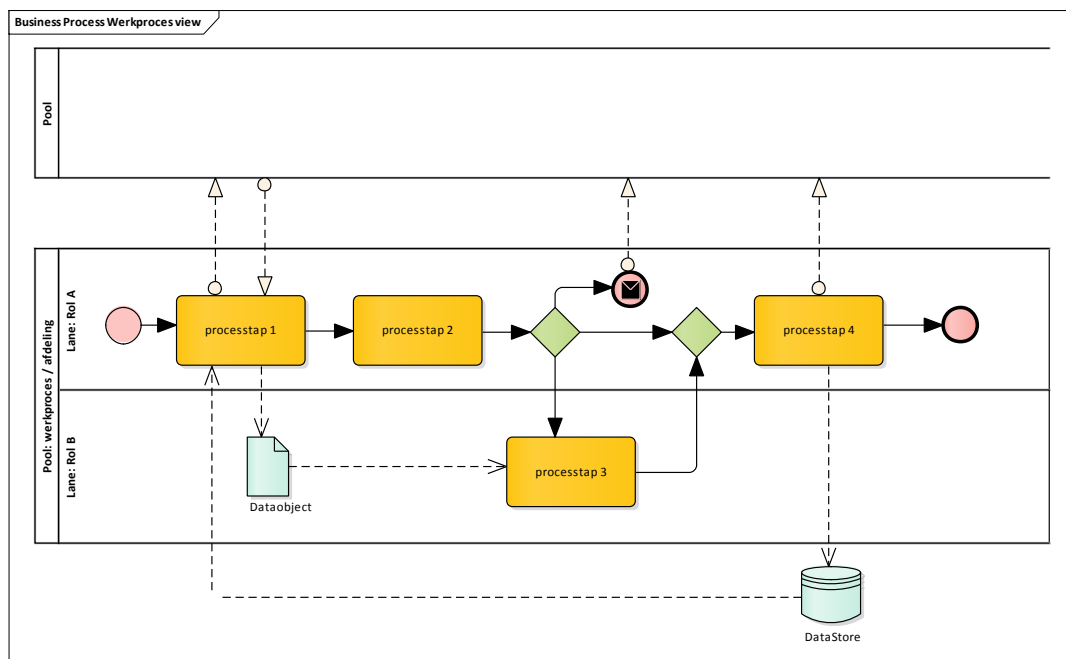
De bedrijfsproces view geeft een geordende reeks werkprocessen weer die binnen één organisatie wordt uitgevoerd met als doel een (combinatie van) bedrijfsservices te leveren aan een externe klant. De processen die buiten de organisatie uitgevoerd worden (in de andere pools) hoeven niet gemodelleerd te worden (zgn blackbox pools). Wanneer het zinvol is deze processen wel op te nemen in deze view dan kan mag wel.



View Bedrijfsproces in BPMN

#### 4.3.5 Werkprocesview

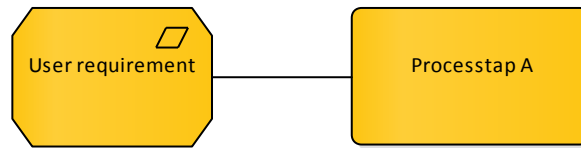
De werkproces view bestaat uit een geordende reeks processtappen die binnen één bedrijfsfunctie worden uitgevoerd met als doel een specifieke bijdrage (prestatie) te leveren aan een bedrijfsservice die (uiteindelijk) zal worden geleverd aan een externe klant.



View BPMN-werkproces

## 4.4 Relatie processen en requirements

Processtappen en user requirements beschrijven allebei gedrag dat in het bedrijf wordt uitgevoerd; grafische resp. in tekst. Een user requirement hoort bij een processtap en een processtap heeft 1 of meer user requirements. Deze relatie kan in een *'processtap requirements relatie view'* weergegeven worden. Deze view wordt veelal in tabelvorm gepresenteerd.



Relatie user requirement en processtap

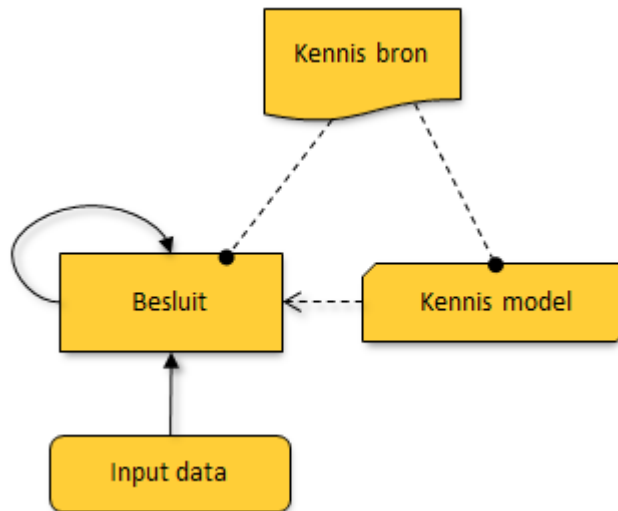
## 4.5 Beslissingsmodel

### Wat


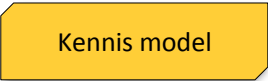
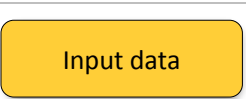



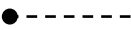
Een beslissingsmodel specificeert de actie die genomen moeten worden wanneer aan bepaalde condities voldaan wordt. Er zijn momenteel 2 scholen voor het modelleren van beslissingen: DMN (Decision Model & Notation) en TDM (The Decision Model). In deze stijlgeds hebben wij gekozen voor DMN omdat deze net als de andere talen in deze stijlgeds door de Object Management Group wordt beheerd.

### Waarom

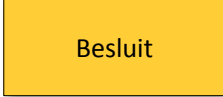


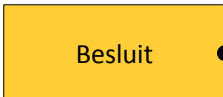


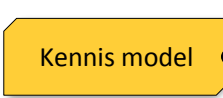

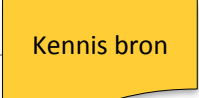
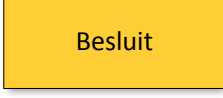

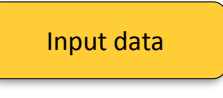
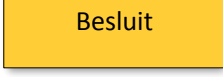

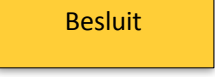
Besluiten sturen gedrag in de organisatie. Door de besluiten expliciet te maken worden deze objectiever en het resulterende gedrag voorspelbaarder.



Metamodel Decision model & Notation

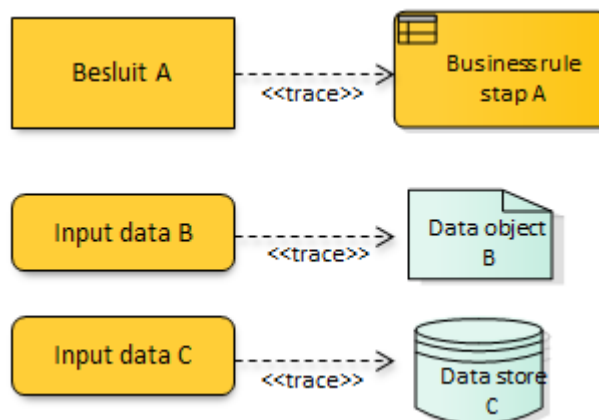
Concept	Benaming	Omschrijving
	Besluit	Een besluit is de activiteit om aan de hand van een kennis model en input data een beslissing te nemen.
	Kennis model	Een kennis model is een bedrijfsregel, beslissingstabel of analysemodel aan de hand waarvan een besluit gemaakt kan worden.
	Input data	Input data is de data die als input wordt gebruikt bij het nemen van een besluit.
	Kennis bron	Een kennis bron is de bron waar een besluit of het kennis model zich op baseert (zoals wet- en regelgeving, beleid, richtlijnen, productvoorwaarden, etc.)
	Kennis relatie	Een kennis relatie is de aanroep van een kennis model voor het nemen van een besluit.
	Informatie relatie	Een informatierelatie is het ophalen van inputdata, of output van een ander besluit, als input voor een besluit.
	Autoriteit relatie	Een autoriteit relatie is de referentie naar een kennisbron waarop een besluit of kennismodel gebaseerd is.

## Relaties

Van concept	Relatie	Naar concept	Toelichting
			Een besluit past een kennismodel toe. Een kennismodel wordt toegepast in een besluit
			Een besluit komt voort uit en is gebaseerd op een kennis bron. Een kennis bron is de basis voor een besluit.
			Een kennismodel is gebaseerd op een kennis bron. Een kennis bron is de basis voor een kennismodel.
			Een besluit gebruikt inputdata. Input data wordt gebruikt door een besluit.
			Een besluit gebruikt als input de uitkomst van een ander besluit. De output van een besluit is input voor een ander besluit.

### 4.5.1 Relatie beslissingsmodel en procesmodel

Een beslissingsmodel is een nadere uitwerking van een business rule processtep. Deze relatie wordt expliciet gemaakt door een trace te leggen van het *besluit* naar de bewuste *processtep*. Daarnaast komt de *input data* voor een besluit overeen met een *datastore* of een *dataobject* uit het procesmodel. Je kunt ervoor kiezen om alleen DMN objecten te gebruiken in het beslissingsmodel. In dat geval zul je onderstaande trace relaties moeten leggen.



Trace relatie besluitvorming en procesontwerp

Je mag ook de BPMN objecten hergebruiken in het beslissingsmodel, in dat geval mag je echter niets toevoegen aan de kenmerken of toelichtingen van de BPMN objecten. Je mag bijv. de dataobjecten en datastores gebruiken i.p.v. het input data concept.



## 4.6 Applicatiefunctieiteit

### 4.7 Use case model

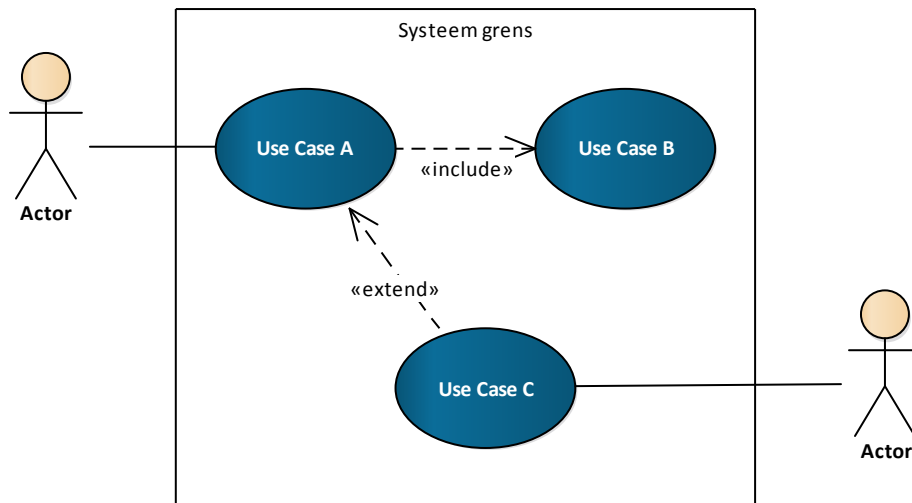
#### Wat

Een use case model is een weergave van de gedragingen (functies) die een systeem biedt aan haar omgeving zijnde gebruikers of andere systemen.

#### Waarom

Use cases zijn een middel om het gewenste gedrag van een systeem nader te specificeren. Een use case kent binnen het NBR dezelfde granulariteit als een processtap in het procesmodel: OTOPOP (one time, one place, one person). Dit betekent dat voor elke niet handmatige processtap in het werkprocesmodel, die gebruik maakt van het informatiesysteem in scope, een use case geïdentificeerd kan worden. De relatie tussen een use case en de gebruiker of een ander systeem geeft ook inzicht in de beoogde interfaces (human-to-system en system-to-system).

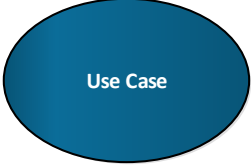


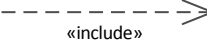
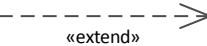
#### 4.7.1 Metamodel



Use case metamodel

#### Concepten

Concept/element	Benaming	Omschrijving	Conventies
	Actor	Een actor is een rol, organisatie, extern systeem of een tijdtrigger die betrokken is bij het uitvoeren van een use case of resultaten van een use case ontvangt.	De primaire actor(en) modelleren we links van het systeem, de secundaire actoren aan de rechterkant. Elke actor heeft minimaal één associatierelatie met een use case. Naamgevingsconventie = zelfstandig naamwoord.

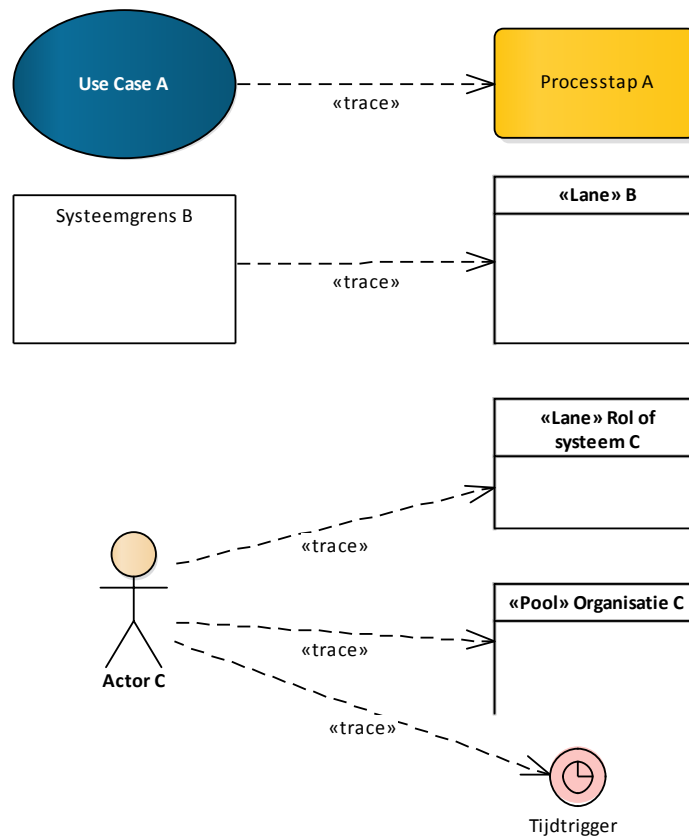
	Use case	Een use case is een eenheid van gedrag die door het systeem geboden wordt.	<p>Elke use case heeft minimaal één associatierelatie met een actor OF een include- of extendrelatie met een andere use case.</p> <p>Een use case wordt nader gespecificeerd m.b.v. een use case template eventueel aangevuld met een activity diagram.</p> <p>Naamgevingsconventie = werkwoord + zelfstandig naamwoord.</p>
	Associatie relatie	Geeft aan dat er een interactie of communicatie is tussen de actor en de use case.	
	Systeemgrens	De systeemgrens is het systeem(onderdeel) waar de use cases onderdeel van uitmaken.	.
	Include	Een include relatie geeft weer dat het gedrag van de ene use case (included) onderdeel uitmaakt van een andere use case (including). De pijl wijst naar de use case waarvan het gedrag overgenomen wordt.	Een included use case wordt aangemaakt om hergebruik van systeemgedrag binnen andere (including) use cases mogelijk te maken. Op deze manier kan het gedrag eenmaal gespecificeerd worden.
	Extend	Een extend relatie geeft weer dat het gedrag van de ene use case (extending), onder condities verwoord in de andere use case (extended), onderdeel uitmaakt van deze andere use case (extended). De pijl wijst naar de use case waarvan het gedrag, onder voorwaarden, in overgenomen wordt.	Een extended use case wordt gebruikt om aanvullende gedrag aan een use case toe te voegen, onder bepaalde voorwaarden. Ook wordt dit gebruikt om complex gedrag in een aparte (extending) use case te specificeren. De extended use case kan dan relatief eenvoudig blijven.

### Use case view

De use case view is gelijk aan het metamodel.

#### 4.7.2 Relatie use case model en procesmodel

Een use case representeert een eenheid van gedrag van het systeem en komt qua granulariteit overeen met een processtap uit het procesontwerp. Deze relatie wordt expliciet gemaakt door een trace te leggen van de *use case* naar de bewuste *processtap*. Daarnaast komt de *systeemgrens* overeen met een *lane* uit het procesmodel. Een actor is een *lane* (rol of systeem), een *pool* (*Organisatie*) of een timerevent (*tijdtrigger*).



Tracing use case model naar procesmodel

## 4.8 Activiteiten diagram

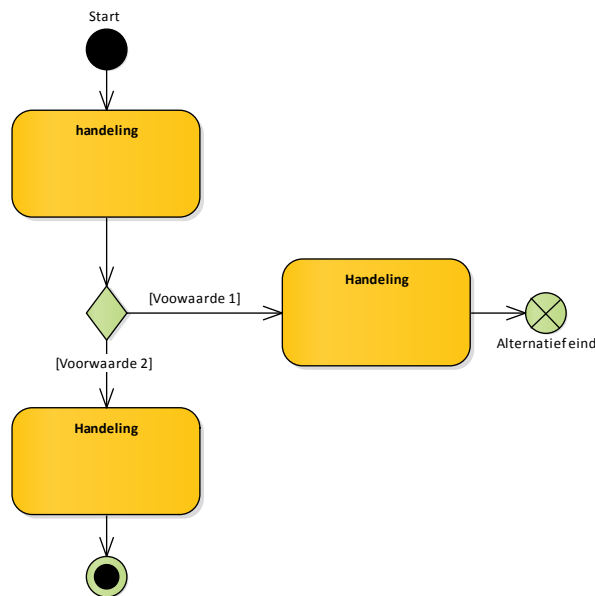
### Wat

Een activiteiten diagram (AD) is een grafische weergave van chronologische handelingen die binnen een use case (of processtap) worden uitgevoerd.

### Waarom



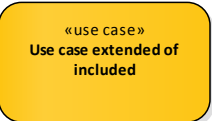

Het AD helpt inzicht te verkrijgen in de (functionele) systeem requirements, biedt houvast bij de controle op compleetheid en kan ondersteunen bij het testen van het systeem.




### Metamodel



Activity diagram metamodel

### Concepten

Concept/element	Benaming	Omschrijving
Start 	Startpunt	Een startpunt markeert de start van een reeks handelingen binnen één processtap.
	Handeling (activity)	Een handeling is de kleinst mogelijke eenheid van werk, uitgevoerd door één persoon of één systeem op één plek op één moment
	Extended of included use case	Een included of extended use case wordt opgenomen in een activity diagram als handeling met stereotype <<use case>>. Op deze wijze modelleer je het hergebruik van deze elders uitgewerkte use case.
	Beslissing	Een beslissing is een splitsing of merge om verschillende scenario's weer te geven. Vanuit een beslissing komen twee of meer paden voort. Een merge wordt geplaatst voor de handeling waar 2 of meer paden weer samenkomen.
[voorwaarde]	Voorwaarde / conditie	De voorwaardetekst wordt toegevoegd naast een beslissing om de voorwaarde/conditie te specificeren welk scenario wanneer genomen moet worden. De voorwaardeteksten bij één beslissing moeten elkaar uitsluiten.

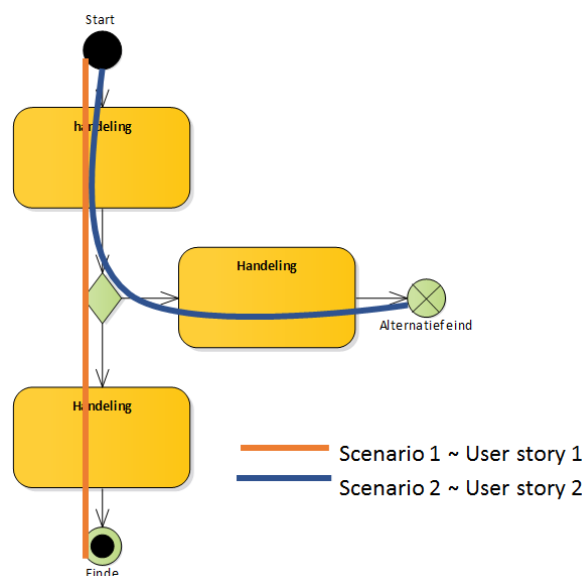
 Einde	Eindpunt	Een eindpunt is het einde van één of meer handelingen waarmee het beoogde doel van de use case (processtap) gehaald is.
 Alternatief eindpunt	Alternatief eindpunt	Een alternatief eindpunt is het einde van één of meer handelingen waarbij het beoogde doel van de use case (processtap) niet wordt gehaald.
	Control flow	Deze geven de stroomrichting van het diagram aan. De pijl wijst in de richting van de opvolgende activiteiten.

## Conventies

- Modelleer de happy flow op een verticale lijn en de alternatieve flows links of rechts daarvan.
- Geef op een beslissing aan wat de keuze is, in de vorm van een vraag (bv. update geslaagd?).
- Geef op de uitgaande pijl uit een beslissing aan wanneer welk scenario gevolgd wordt, in de vorm van een voorwaarde/conditie (bv. ja, nee). Deze voorwaarde/condities zijn volledig en sluiten elkaar uit. 'Anders' of 'overig' is toegestaan als de condities niet volledig zijn, maar geen betekenisvolle/onderscheidende condities meer zijn te definiëren.
- Gebruik altijd een merge om alternatieve paden weer bij elkaar te laten komen; dit vergemakkelijkt het ontdekken van scenario's.
- Eventueel kan een merge met een direct opvolgende beslissing gecombineerd worden t.b.v. het overzicht.
- Het is mogelijk om meerdere eindpunten te hebben. Geef bij elk van de eindpunten de postconditie aan, de staat waarin het systeem zich bevindt.
- Gebruik geen swimlanes (of andere partitionering).
- Vermijd kruisende lijnen waar mogelijk.

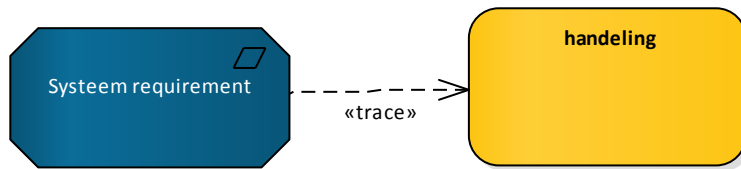
### 4.8.1 Relatie activiteitendiagram en user stories resp. systeem requirements

Een activiteitendiagram geeft de handelingen weer die een actor met het systeem uitvoert. Een user story komt qua granulariteit ongeveer overeen met één use case scenario.



Relatie scenario vs user story

Een systeem requirement komt qua granulariteit overeen met één handeling. Werk je met systeem requirements dan leg je een trace relatie naar de bewuste handeling.



Trace systeem requirement naar handeling

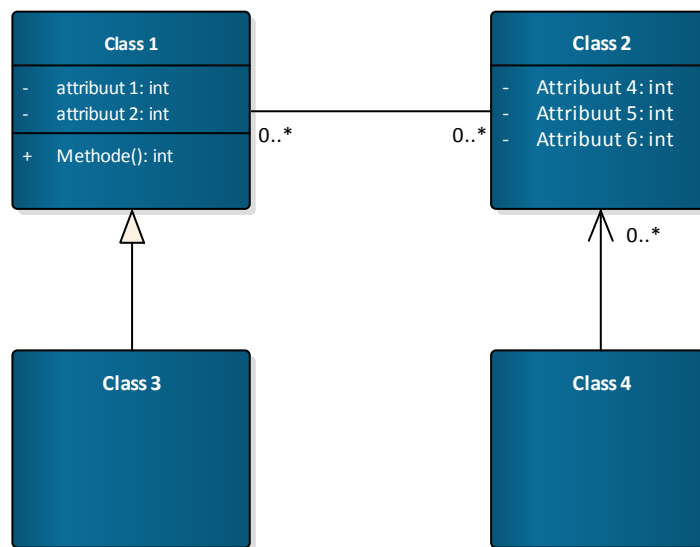
## 4.9 Domein model

### Wat

Een domein model (of class model) is een model van de statische structuur van een systeem. Het is een model van 'mensen, dingen, plaatsen, gebeurtenissen of concepten' uit de echte wereld, de relaties tussen deze objecten en de gegevens die over deze objecten vastgelegd worden.

### Waarom

Met een domein model (of class model) worden eenduidige definitie van de objecten (classes), de relaties tussen objecten en de gegevens (attributen en methoden) vastgelegd. Het laat zien welke gegevens voor het systeem van belang zijn en hoe die samenhangen.

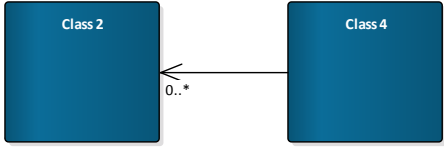
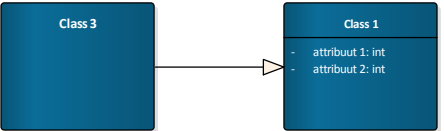


Class model metamodel

### Concepten

Het domein model kan met een UML class model gemodelleerd worden.

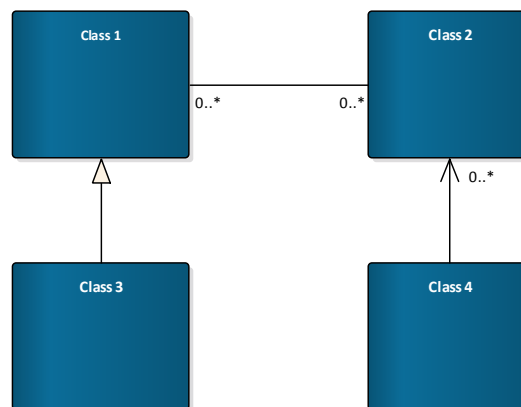
Concept/element	Benaming	Omschrijving
	Class	<p>Een class is een collectie van personen, plaatsen, dingen gebeurtenissen of concepten uit de werkelijkheid waar attributen over worden vastgelegd in een systeem. Een class heeft een naam (zelfstandig naamwoord): bovenste deel van de class</p> <p>Een class kan attribuuttypen bevatten: middelste deel van de class.</p> <p>Een class kan methoden bevatten die gebaseerd zijn op de attributen uit de class: onderste deel van de class.</p>
	Bidirectionele relatie	<p>Een bi-directionele relatie is een twee-weg relatie tussen twee classes. Beide classes weten van elkaars bestaan af.</p>

	Unidirectionele relatie	Een unidirectionele relatie is een één-weg relatie. Slechts 1 van beide classes weet van het bestaan van de andere. De class waar de pijl heen wijst (class 2) heeft geen weet van de andere class (class 4).
	Erving relatie	Een erving relatie geeft aan dat de eigenschappen van de ene class (1 = moeder) ook gelden voor de andere class (3= kind).  De class waar de pijl uit vertrekt (3) erft de eigenschappen van de class waar de pijl heen wijst (1).
0..* (0 of meer) 0..3 (0 t/m 3) 1 (exact 1) 1...* (1 of meer) 2,4,6 (exact 2,4 of 6)	Multipliciteit / cardinaliteit	De multipliciteit / cardinaliteit geeft aan hoeveel instanties van een class een relatie kunnen hebben met één instantie van de betreffende class.

#### 4.9.1 Domein model

Er zijn drie views die veel gebruikt worden:

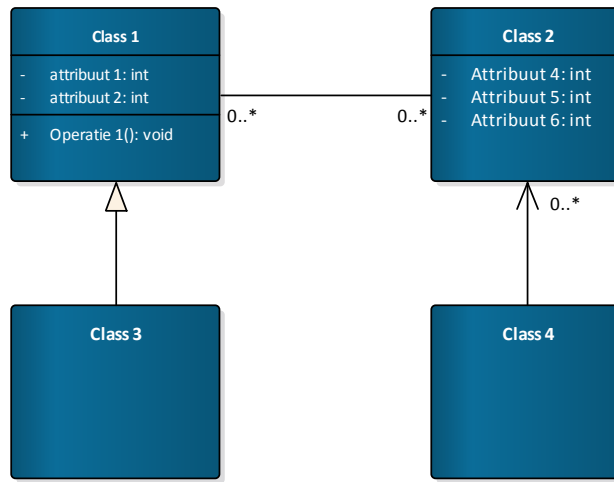
- 1) Domein relatie model: alleen de classes en de relaties tussen de classes die door een systeem onderhouden worden. Met deze view wordt gefocused op de globale structuur van de classes. Welke class heeft wat voor relatie met een andere class.



Domein relatie model

- 2) Domein model inclusief attributen en methoden: als boven inclusief de attributtypen, die door het systeem onderhouden worden. Met deze view wordt gefocused op gegevens en methoden die over de classes worden vastgelegd.

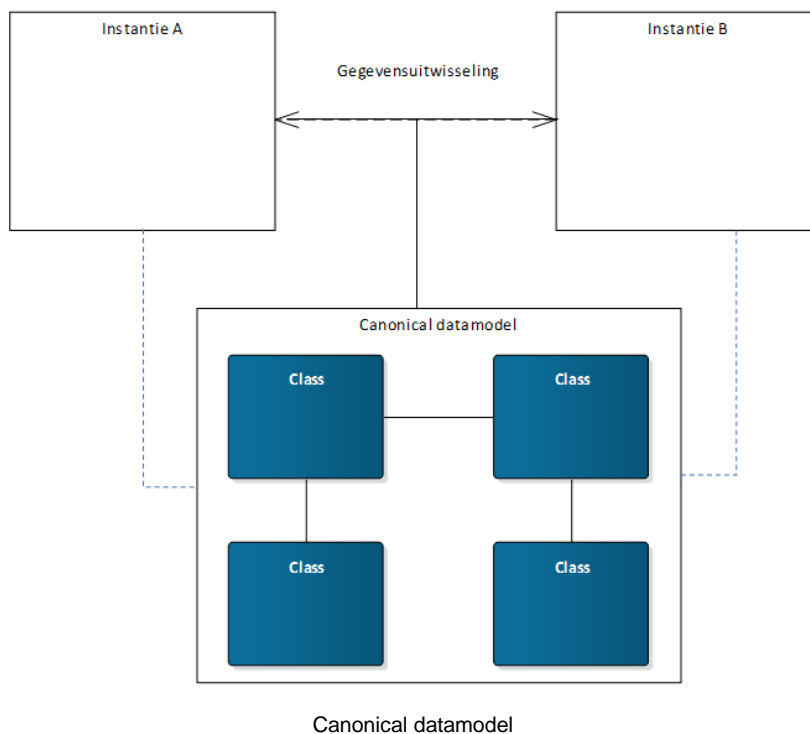




*Domein model inclusief aantal attributen en methoden*

### 3) Canonical datamodel: als bij view 2

Het canonical datamodel geeft de classes, relaties en attribuuttypen weer zoals deze in de berichtuitwisseling tussen gegevens leverende instanties gebruikt worden. Iedere gegevens leverende instantie zorgt voor een vertaling tussen het eigen datamodel van/naar het canonical datamodel.



Canonical datamodel

## 4.10 Interfacemodel

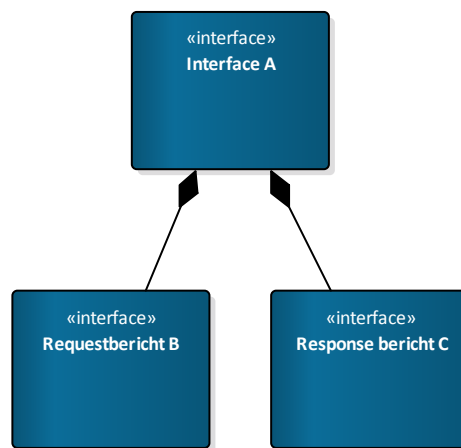
### Wat

Een interfacemodel specificeert de berichten van gegevens leverende instanties. Een interfacespecificatie bestaat uit een request- en een response bericht. Het request bericht specificeert de attributen (en bijbehorende classes) waarmee een selectie van (instanties van) objecten uit de database opgevraagd kan worden. Het response bericht specificeert de attributen en bijbehorende classes die per (instantie van een) object opgevraagd kunnen worden.

### Waarom

Een eenduidig begrip van de gegevens die uitgewisseld kunnen worden is essentieel om gegevensuitwisseling tussen systemen of organisaties goed te laten verlopen. De gegevens die in een interface bericht opgenomen zijn, zijn bij voorkeur gebaseerd op een canonical datamodel waar de definities helder en eenduidig vastgelegd zijn.

### Metamodel

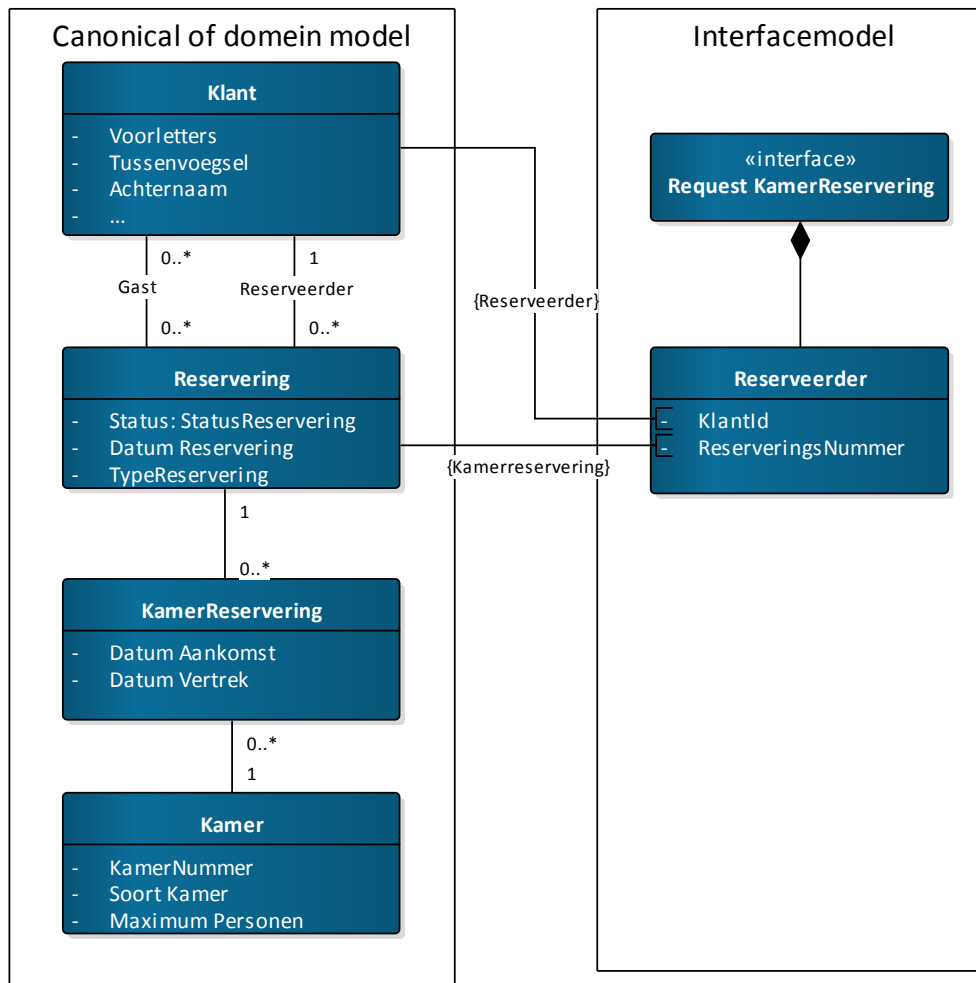


Interfaces metamodel

### 4.10.1 Requestbericht

Het requestbericht specificeert de objecttypen (classes) waarover gegevens opgevraagd kunnen worden en welke deelverzamelingen van objecten (instanties van classes) opgevraagd kunnen worden. Door het leggen van associatie relaties tussen de classes in het canonical (of domein) model en de classes in het requestbericht wordt gemodelleerd welke selectie van objecten opgevraagd kan worden.

In onderstaand voorbeeld kan een KAMERRESERVERING opgevraagd worden van een specifieke RESERVEERDER aan de hand van een klantid en/of een reserveringsnummer.

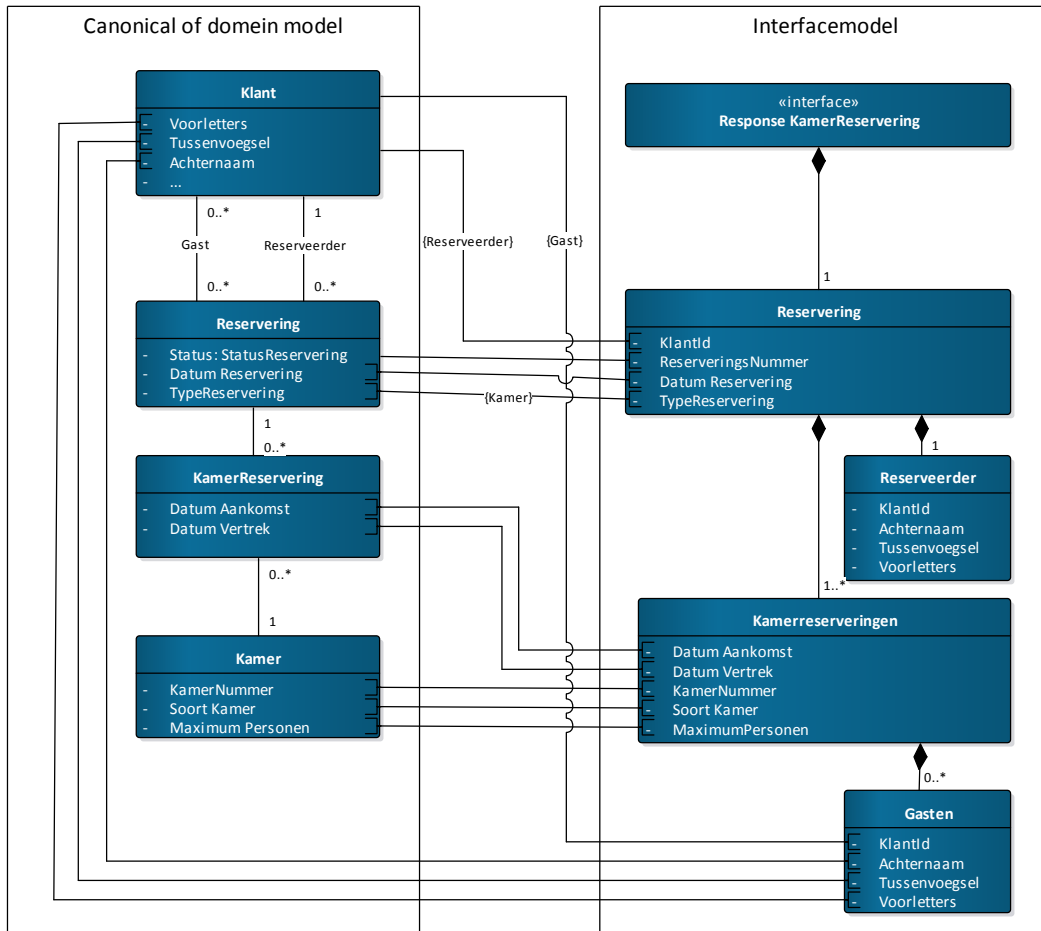


Request bericht voorbeeld

#### 4.10.2 Response bericht

Het responsebericht specificeert de objecttypen en gegevens die door de leverende instantie geleverd kunnen worden n.a.v. een requestbericht.

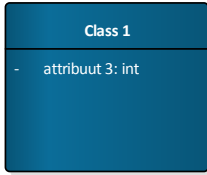


In onderstaand voorbeeld worden de gegevens van de opgevraagde RESERVERING doorgegeven inclusief gegevens over de RESERVEERDER, de KAMERRESERVERINGen en de GASTen. Met associatie relaties is aangegeven wat de herkomst van de attributen is op basis van het canonical (of domein) model. Per associatie relatie kan een beperking aangegeven worden.



Response bericht voorbeeld

## Concepten

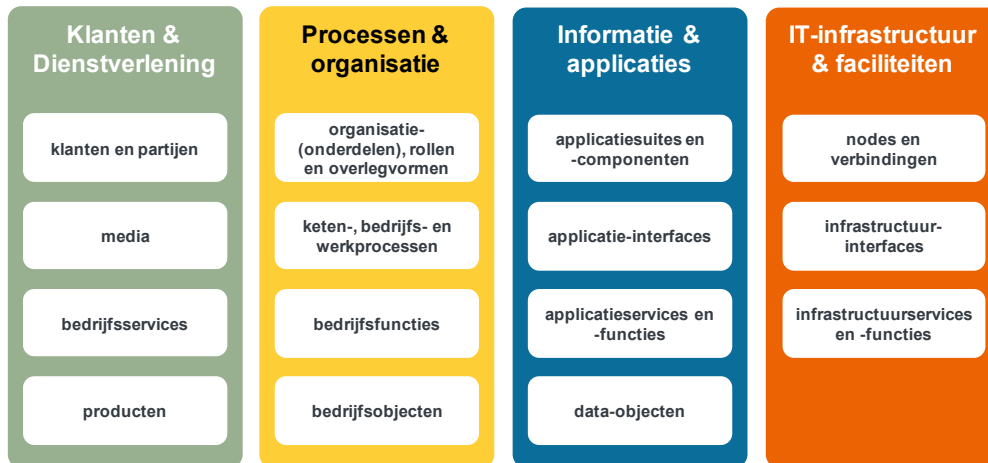
Concept/element	Benaming	Omschrijving
«interface» Interface A	Interface	Een interface is een specificatie van request- en response berichten.  Een interface is een class met <<interface>> als classificatie.
«interface» Requestbericht B	Request bericht	Een requestbericht is een specificatie van objecttypen (classes) waarover gegevens opgevraagd kunnen worden en welke deelverzamelingen van objecten opgevraagd kunnen worden.  Een requestbericht is een class met <<interface>> als classificatie. Naamgeving = 'Request' + zelfstandig naamwoord.
«interface» Response bericht C	Response bericht	Een Responsebericht is een specificatie van objecttypen (classes) en attributen die door de leverende instantie geleverd kunnen worden n.a.v. een requestbericht

		Een responsebericht is een class met <<interface>> als classificatie. Naamgeving = 'Response' + zelfstandig naamwoord.
	Class	Een class is een collectie van personen, plaatsen, dingen gebeurtenissen of concepten uit de werkelijkheid waar attributen over worden vastgelegd in het systeem. Een class heeft een naam (zelfstandig naamwoord). Een class kan attribuuttypen bevatten.
	Compositie relatie	Een compositie relatie geeft aan dat een class onderdeel is van (het bestaan van) een andere class. Wanneer een instantie van een class verwijderd wordt dan worden de bijbehorende instanties van de andere class ook verwijderd.
	Associatie relatie	Met een associatie relatie wordt aangegeven wat de herkomst is van een attribuut in interface bericht.  Optioneel kan een beperking opgenomen worden. Met een beperking worden alleen instanties van objecten geselecteerd die aan de beperking voldoen. Een beperking wordt tussen {...} geplaatst.  Met een associatie relatie kan ook aangegeven worden dat een identificerend gegeven van een class als identificerend attribuut van een interface class wordt gebruikt.
<p>0..* (0 of meer) 0..3 (0 t/m 3) 1 (exact 1) 1..* (1 of meer) 2,4,6 (exact 2,4 of 6)</p>	Multipliciteit / cardinaliteit	De multipliciteit / cardinaliteit geeft aan hoeveel instanties van een class een relatie kunnen hebben met één instantie van de betreffende class.

Hieronder is een view van een interface weergegeven. Alle elementen vanuit het metamodel worden hierin gebruikt.

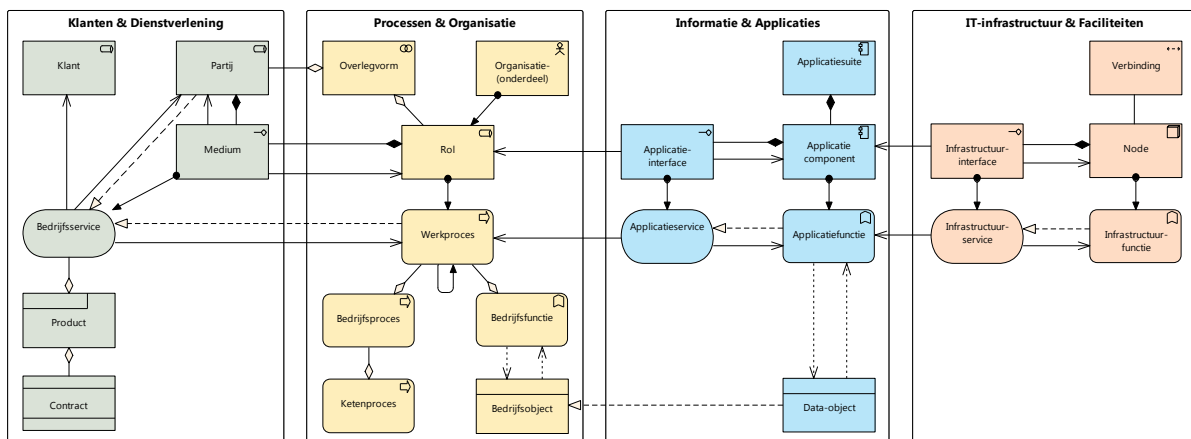
## 5 Aansluiting op Novius Architectuur Raamwerk

het Novius architectuurraamwerk is afgeleid van het Novius business transformation framework. Kenmerkend voor het Novius raamwerk zijn de vier kolommen waarvan de eerste twee betrekking hebben op de businessaspecten van een bedrijf en de laatste twee op de informatievoorziening en infrastructuur van een bedrijf.



Novius Architectuurraamwerk

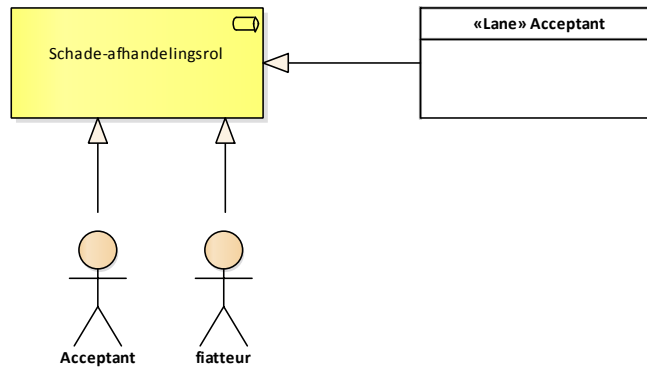
Het Novius architectuurraamwerk heeft een metamodel gebaseerd op Archimate.



Novius Architectuurraamwerk – metamodel Archimate

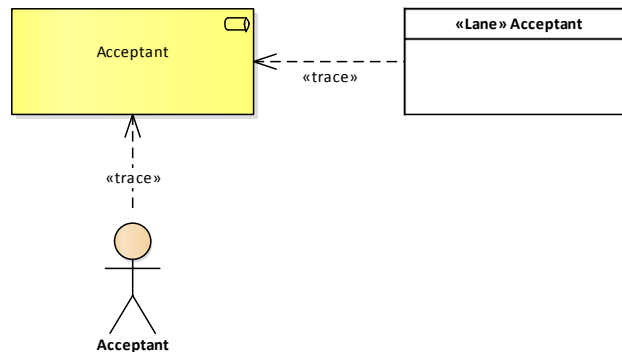
Het Novius bedrijfsontwerp raamwerk sluit aan op het Novius architectuurraamwerk. Dit betekent dat bedrijfsontwerp een specificatie maakt van een instantie van een architectuurconcept. Er zijn drie manieren om de relatie tussen architectuurconcepten en bedrijfsontwerpconcepten te leggen:

- 1) Specialisatie relatie: bedrijfsontwerp maakt gebruik van eigen concepten die een verbijzondering zijn van een architectuurconcept. De bedrijfsontwerper legt een specialisatiere relatie tussen het bedrijfsontwerpconcept en het architectuurconcept.



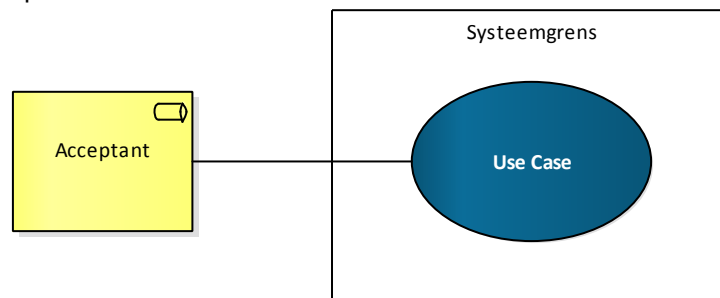
Voorbeeld specialisatiere relatie

- 2) Trace relatie: bedrijfsontwerp maakt gebruik van eigen concepten die een substituuat zijn voor architectuurconcepten. De bedrijfsontwerper legt een trace relatie tussen het bedrijfsontwerpconcept en het architectuurconcept.



Voorbeeld tracerelatie

- 3) Hergebruik architectuurconcepten in bedrijfsontwerp modellen. De bedrijfsontwerper maakt gebruik van architectuurconcepten in bedrijfsontwerp modellen. De beperking is wel dat de bedrijfsontwerper geen kenmerken of toelichtingen mag wijzigen van de architectuurconcepten.



Voorbeeld hergebruik architectuurconcept in bedrijfsontwerp model

## 5.1 Trace relaties

Een aantal concepten binnen bedrijfsontwerp hebben vergelijkbare concepten binnen de architectuur, het concept bestaat m.a.w. zowel in het bedrijfsontwerpmodel als in het architectuurmodel. Wanneer in beide modelwerelden met hetzelfde concept gemodelleerd wordt maar wel in de modelwereld eigen concepten, dan zal de relatie tussen deze vergelijkbare concepten onderhouden moeten worden. De bedrijfsontwerper zal een trace relatie moeten onderhouden tussen het bedrijfsontwerpconcept en het architectuurconcept. Op deze manier zorg je ervoor dat de concepten uit beide werelden los van elkaar gebruikt kunnen worden en toch goed op elkaar (blijven) aansluiten. In deze paragraaf zijn de mogelijke trace relaties toegelicht.

### 5.1.1 Motivatiemodel

Het motivatiemodel kan zowel door bedrijfsontwerpers als door architecten opgesteld worden. In deze stijlguide wordt hiervoor het metamodel uit Archimate gehanteerd. Afhankelijk van de situatie wordt de motivatie voor de verandering uitgewerkt door de bedrijfsontwerper, de architect of het is een collaboratie tussen bedrijfsontwerper en architect. Voor zover er geen extra verdieping gemaakt wordt door bedrijfsontwerp is tracing hier niet aan de orde.

### 5.1.2 Requirementsmodel

Het requirementsmodel modelleert de veranderbehoefte van verschillende stakeholders in de vorm van doelen en verschillende typen requirements. Van grof (business requirements en epics) naar fijn (systeem requirements en user stories) kan een analyse gemaakt worden van de globale needs/behoefte van de organisatie naar gedetailleerde en specifieke eisen t.a.v. systemen. Wanneer architectuur de business requirements / epics opstelt, dan kan de bedrijfsontwerper deze 1-op-1 dupliceren om de volgende verdiepingsslag hierop te maken (user requirements, systeemrequirements, features en/of user stories). Dit lijkt wat dubbelop, maar hiermee wordt geborgd dat architectuur en bedrijfsontwerp hun eigen model kunnen beheren en wijzigingen in de architectuur gecontroleerd doorgezet kunnen worden in het bedrijfsontwerp.

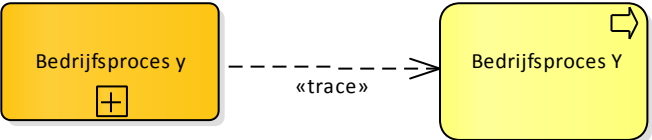
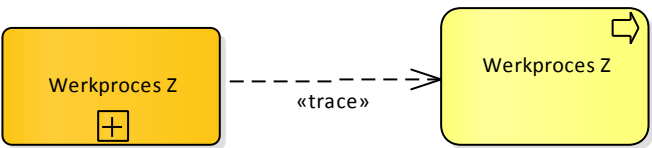
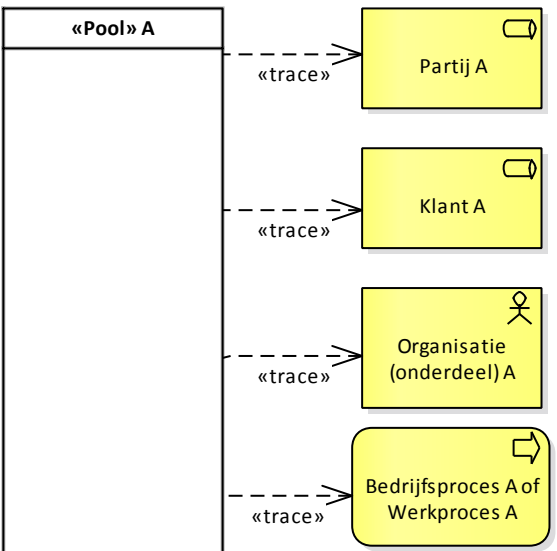
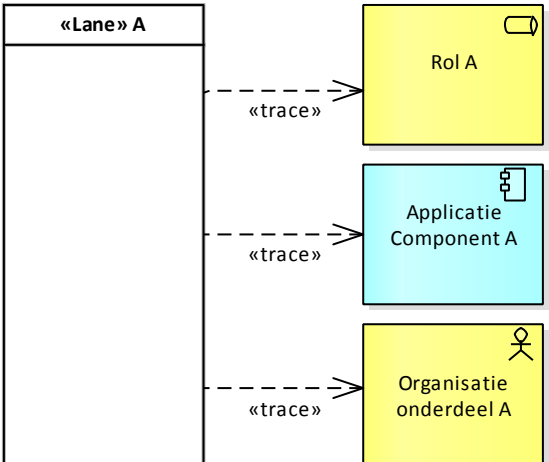
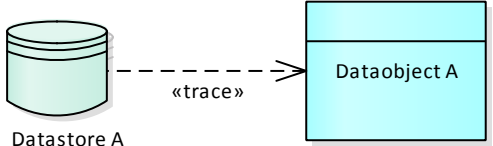
Trace relatie van bedrijfsontwerp naar architectuur	Bedrijfsontwerp	Architectuur
Een business requirement / epic X in bedrijfsontwerp tracen naar een business requirement / epic X in architectuur.	<p>The diagram shows two elements in the Business Design layer: a Business requirement (bedrijfsontwerp) represented by a grey octagon with a small square icon, and an Epic (bedrijfsontwerp) represented by a grey rounded rectangle. Dashed arrows labeled «trace» point from the Business requirement to the Epic.</p>	<p>The diagram shows two elements in the Architecture layer: a Business requirement (architectuur) represented by a grey octagon with a small square icon, and an Epic (architectuur) represented by a grey rounded rectangle. Dashed arrows labeled «trace» point from the Business requirement to the Epic.</p>

### 5.1.3 Procesmodel

In de volgende tabel is opgenomen welke mogelijke traces er kunnen bestaan tussen procesontwerp concepten en architectuur concepten




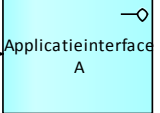
Trace relatie van procesontwerp naar architectuur	Procesontwerp	Architectuur
Een ketenproces X in bedrijfsontwerp tracen naar een ketenproces X in architectuur.	<p>The diagram shows a Chain Process X (bedrijfsontwerp) in the Business Design layer, represented by a yellow rounded rectangle with a plus sign icon. A dashed arrow labeled «trace» points to a Chain Process X (architectuur) in the Architecture layer, represented by a yellow rounded rectangle with a square icon.</p>	



<p>Een bedrijfsproces Y in bedrijfsontwerp tracen naar een bedrijfsproces Y in architectuur.</p>	 <p>The diagram shows a yellow box on the left labeled 'Bedrijfsproces y' with a plus sign icon. A dashed arrow labeled '«trace»' points to a yellow box on the right labeled 'Bedrijfsproces Y' with a right-pointing arrow icon.</p>
<p>Een werkproces Z in bedrijfsontwerp tracen naar een werkproces Z in architectuur.</p>	 <p>The diagram shows a yellow box on the left labeled 'Werkproces Z' with a plus sign icon. A dashed arrow labeled '«trace»' points to a yellow box on the right labeled 'Werkproces Z' with a right-pointing arrow icon.</p>
<p>Een pool A in bedrijfsontwerp tracen naar één instantie in de architectuur:</p> <ul style="list-style-type: none"> <li>- Partij A of,</li> <li>- Klant A of,</li> <li>- Organisatie(onderdeel) A of,</li> <li>- Bedrijfsproces A of,</li> <li>- Werkproces A.</li> </ul>	 <p>The diagram shows a vertical box on the left labeled '«Pool» A'. Four dashed arrows labeled '«trace»' point to four yellow boxes on the right: 'Partij A' (with a right-pointing arrow icon), 'Klant A' (with a right-pointing arrow icon), 'Organisatie (onderdeel) A' (with a stick figure icon), and 'Bedrijfsproces A of Werkproces A' (with a right-pointing arrow icon).</p>
<p>Een lane A in bedrijfsontwerp tracen naar één instantie in de architectuur:</p> <ul style="list-style-type: none"> <li>- (personele) Rol of,</li> <li>- Applicatie component A of</li> <li>- Organisatieonderdeel A.</li> </ul>	 <p>The diagram shows a vertical box on the left labeled '«Lane» A'. Three dashed arrows labeled '«trace»' point to three boxes on the right: 'Rol A' (yellow, with a right-pointing arrow icon), 'Applicatie Component A' (light blue, with a plug icon), and 'Organisatie onderdeel A' (yellow, with a stick figure icon).</p>
<p>Een datastore A in bedrijfsontwerp tracen naar een Data object A in architectuur.</p>	 <p>The diagram shows a cylinder icon on the left labeled 'Datastore A'. A dashed arrow labeled '«trace»' points to a light blue box on the right labeled 'Dataobject A'.</p>

### 5.1.4 Functioneel ontwerp

In de volgende tabel is opgenomen welke mogelijke traces er kunnen bestaan tussen functioneel ontwerp concepten en architectuur concepten.

Trace relatie van functioneel ontwerp naar architectuur	Functioneel ontwerp	Architectuur
<b>Domein model</b>		
Een class in UML wordt getraced naar één dataobject in Archimate.	 <pre> classDiagram     class ClassA     </pre>	 <pre> classDiagram     class DataobjectA     </pre>
<b>Interface model</b>		
Een interface class in UML wordt getraced naar één applicatie interface in Archimate	 <pre> classDiagram     class ApplicatieInterfaceA["«interface»\nApplicatie interface\nA"]     </pre>	 <pre> classDiagram     class ApplicatieinterfaceA["Applicatieinterface\nA"]     </pre>