

# **Architectuurpatronen en de IKEA metafoor**

Bert Dingemans  
December 2019

## Inhoudsopgave

<b>Wat is de IKEA metafoor</b>	<b>4</b>
Meubelmakers of IKEA	4
<b>Architectpatronen en -bouwblokken</b>	<b>7</b>
Definitie patronen	7
Definitie bouwblokken	8
<b>Voorbeelden en catalogi</b>	<b>9</b>
Datapatronen	9
Datamodel patronen	9
Semantische patronen	10
Canonieke data patronen	12
Integratie- en SOA patronen	13
Softwarepatronen	15
Interactiepatronen	15
Big Data Patterns	16
Infrastructuur patronen	16
Voorbeeld	17
<b>Bijlage 1 Voorbeeld toepassen bouwblokken</b>	<b>18</b>
Objectmodel Bouwblokken	18
Objecten en definities	18
Objecten, definities en eisen	23
ArchiMate Viewpoints voor Bouwblokken	24
Viewpoint Bouwblokken Basis Applicatielaag	25
Viewpoint Bouwblokken Technische laag	26
Viewpoint Bouwblokken en Eisen	27
Viewpoint bouwblok meerlagig applicatie- en infrastructuurlaag	27
Voorbeelden	28
<b>Catalogi</b>	<b>34</b>

## Samenvatting

*Veel organisaties hebben een applicatielandschap waarbij de introductie van maatwerkoplossingen wordt ontmoedigd en de introductie van hergebruik of het aanschaffen van standaardpakketten de voorkeur heeft. Vindt er al ontwikkeling van specifieke toepassingen plaats dan wordt er veelal gezocht naar bouwstenen of naar producten die configureerbaar zijn. Informatiearchitecten zijn in veel gevallen een belangrijke stakeholder/uitvoerder van de hierboven beschreven overgang. In dit whitepaper gaan we op basis van een metafoor uit de “constructiewereld” in op de uitdagingen en hulpmiddelen die ingezet kunnen worden om invulling te geven aan bovenstaande uitdagingen. We gebruiken hiervoor het voor iedereen bekende meubelwarenhuis IKEA. IKEA kenmerkt zich door de inzet van patronen en bouwblokken tot in vergaande vorm. Dat is daarmee voor ons als informatiearchitecten een mooi voorbeeld. Op basis van deze metafoor zullen we architectuur patronen en een aantal belangrijk patrooncatalogi behandelen.*

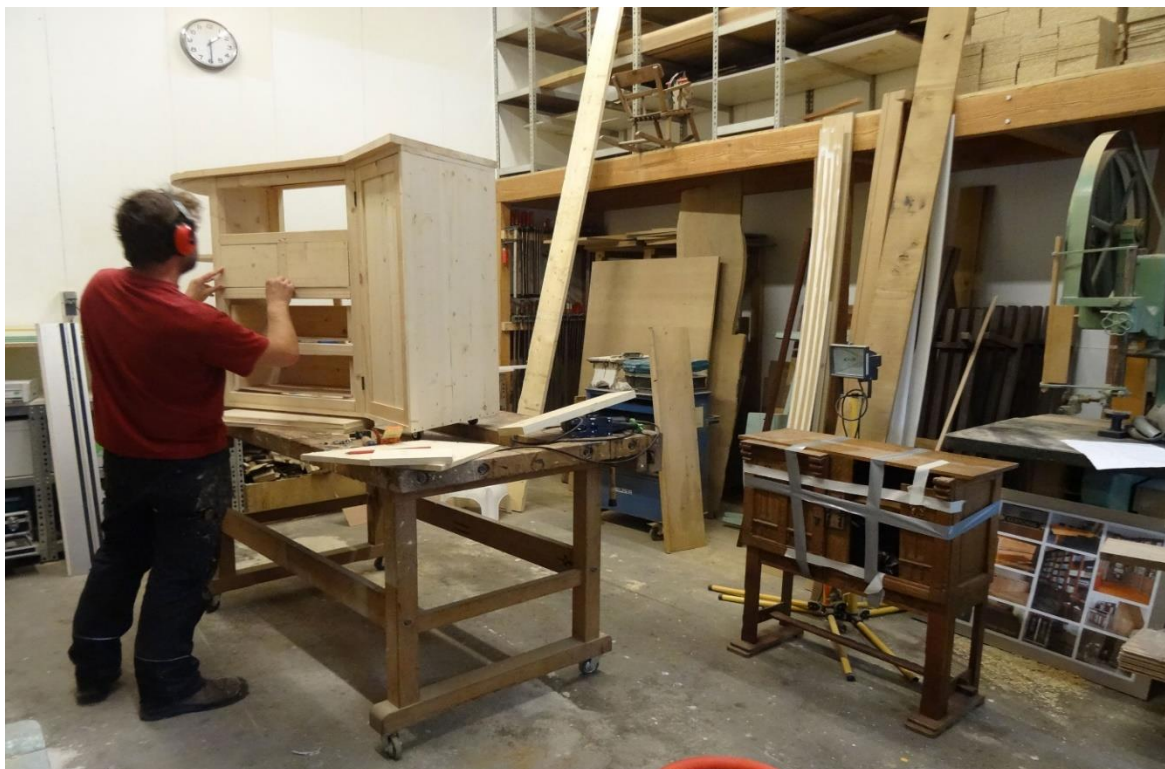
## Wat is de IKEA metafoor

Als je aan 100 mensen in je organisatie vraagt of ze ten minste een meubelstuk van IKEA in huis hebben staan dan zal waarschijnlijk 99 maal ja geantwoord worden. IKEA heeft de meubelindustrie enorm veranderd en die verandering is grotendeels gebaseerd op standaardisatie, hergebruik en het gebruik van bouwblokken en -stenen.

### Meubelmakers of IKEA

Stel dat ik tien nieuwe boekenkasten nodig heb dan kan ik twee scenario's volgen:

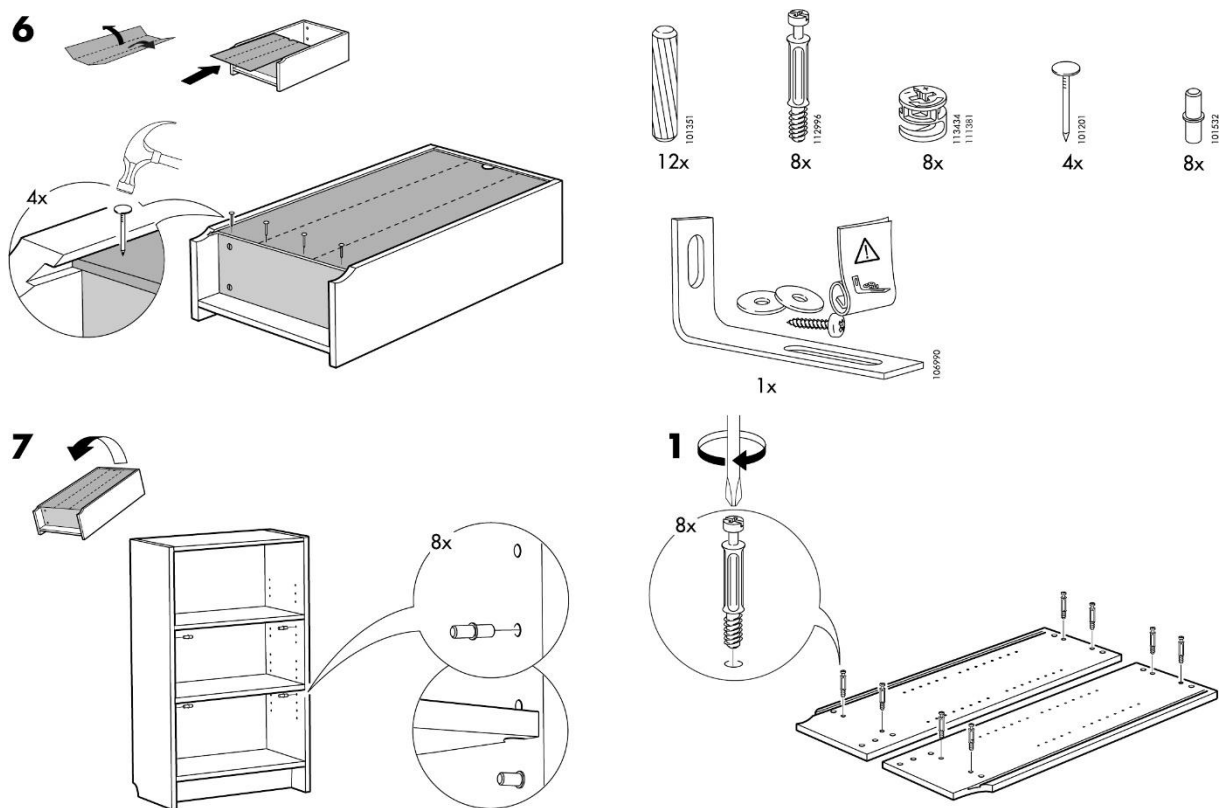
- Ik benader een of meer meubelmakers met de vraag of zij een aantal boekenkasten willen maken. Er ontstaat waarschijnlijk een dialoog waarin er functionele wensen worden uitgewisseld, afspraken worden gemaakt over financiën en levertijd en mogelijk een aantal non functionele zaken worden afgesproken. Vervolgens gaan de meubelmakers aan de slag en zullen na verloop van tijd de meubels bij mij afgeleverd worden.



Bij IKEA is de werkwijze compleet omgedraaid. Daar is door een ontwerper bedacht dat veel mensen behoefte hebben aan de mogelijkheid om boeken bij elkaar in een kast te zetten op een dusdanige wijze dat het past bij het interieur maar ook mogelijkheid biedt om een boekenkast te combineren met andere meubelstukken (al dan niet ontwikkeld door IKEA. In onderstaande afbeelding de Billy boekenkast van IKEA).

Echter deze bouwblokken werkwijze gaat bij IKEA veel verder. Als je meerdere producten koopt, ik spreek uit eigen ervaring, dan zie je op een gegeven moment dat er altijd een zakje schroefjes e.d. bij zit met daarin ook weer gestandaardiseerde onderdelen die naadloos op elkaar aansluiten. Blijkbaar heeft de meubelontwerper binnen IKEA een beperkte keuze uit soorten schroefjes e.d.. Hij/zij zal gebruik moeten maken uit bouwblokken en heeft niet een vrije keuze.

Als laatste is de handleiding gebaseerd op standaard (bouwblokken). De opbouw van het document is steeds hetzelfde, er worden dezelfde symbolen gebruikt en in een aantal gevallen zie je zelf dat plaatjes worden hergebruikt.



Zijn informatie architecten nu meubelmakers of IKEA ontwerpers? Momenteel zullen in veel organisaties genoodzaakt zijn om het werk in te vullen als meubelmaker. Dat is logisch, de bouwblokken zijn er niet, de patronen zijn er niet en veelal ontbreekt het aan de tijd om bouwblokken te gaan ontwikkelen. Is dat een probleem? Dat hangt van de organisatie af waarin de informatie architect dient te acteren. Zijn er geen problemen als opleveren langer duurt, als de oplossing mogelijk duurder is dan noodzakelijk of als het beheer complexer is. Dan is de meubelmaker aanpak geen enkel probleem.

Is de organisatie waarvoor gewerkt wordt dynamischer, of is de omgeving meer aan verandering onderhevig. Zijn kosten relevant? Dan wordt wellicht de IKEA aanpak interessant. Hierbij zal er wel een organisatieverandering plaats gaan vinden. Niet alleen voor de architecten, ook voor beheerorganisatie, de ontwikkelteams maar nog het meest voor de gebruikersorganisatie.

Zal de meubelmakersaanpak verdwijnen? Nee dat denk ik niet. Net als in de meubelwereld zijn er nog steeds meubelmakers nodig. Bijvoorbeeld als er bijzondere functionele eisen zijn, als een standaard oplossing niet volstaat of andere requirements gelden waardoor een IKEA aanpak niet past.

Is de introductie van een IKEA aanpak interessant binnen de informatie architectuur? Als het antwoord ja is dan kun je onderzoeken of een aanpak van architectuurpatronen en -bouwblokken een goede optie is. In dit whitepaper gaan we deze architectuurconcepten behandelen en behandelen we een aantal catalogi van specifieke architectuurpatronen die je op weg kunnen helpen om een set van bouwblokken samen te stellen voor je eigen context.

# Architectpatronen en -bouwblokken

Architectuurpatronen en -bouwblokken zijn niet nieuw. Al in 1977 heeft Christopher Alexander een boek geschreven over de inzet van patronen in de bouwwereld. Hij heeft een patroontaal geïntroduceerd uitgaande van steeds terugkerende functionele eisen en op basis hiervan een gestandaardiseerde oplossing te bieden.

Later is dit concept overgenomen voor de software ontwikkeling. Onder andere door Ward Cunningham en Erich Gamma (ook bekend als Gang of Four. Zie

[https://en.wikipedia.org/wiki/A\\_Pattern\\_Language](https://en.wikipedia.org/wiki/A_Pattern_Language).

In de IKEA metafoor kun je patronen zien als de gestandaardiseerde instructiefoldertjes die bij een bouwpakket zitten. Het zijn modellen hoe een bepaalde functionaliteit te introduceren is. De bouwblokken zijn vergelijkbaar met de gestandaardiseerde zakjes met schroeven en pluggen. Je kunt ze dus niet los van elkaar zien. Je kunt je ook hierbij goed voorstellen dat het werk van de beheerder behoorlijk gaat veranderen, vergelijkbaar met de bovenstaande meubelmakersmetafoor.

## Definitie patronen

Volgens de definitie van ontwerppatronen, "Een ontwerppatroon of patroon in de informatica is een generiek opgezette softwarestructuur, die een bepaald veelvoorkomend type software-ontwerpprobleem oplost." zijn het oplossingen voor veelvoorkomende ontwerpproblemen. Daarmee worden patronen vanuit het oogpunt van stakeholders en concerns interessant, problemen in het algemeen en ontwerpproblemen in het bijzonder hebben altijd de aandacht van één of meerdere betrokkenen.

Toch hebben patronen een ander karakter dan concepten als principes. Met name de oplossingsgerichtheid maakt ze interessant. Door inzet van patronen is de sturing van de architectuur een veel natuurlijker proces. Er is een probleem in het werkveld en door inzet van één of meerdere patronen wordt hieraan invulling gegeven. Uitwerken van verschillende patroonregisters is daarmee vanuit de rol van de architect een krachtig sturingsmiddel.

Ontwerp- en architectuurpatronen zijn er voor meerdere werkgebieden, denk aan software, integratie of datapatronen. In onderstaande afbeelding is een indeling gemaakt op basis van de data functies met een aantal relevante patroonvormen. Deze worden vervolgens in onderstaande paragrafen toegelicht.

Een patroon heeft veelal een gestandaardiseerde opbouw. Dit kan variëren per aspectgebied maar onderstaande tabel geeft een algemene structuur.

Template patronen	
Naam	Naam is een korte krachtige samenvattende term die gebruikt kan worden in discussies tussen betrokkenen
Probleem	Beschrijving van het probleem dat kan worden opgelost met dit patroon
Context	Conditie waarin het patroon wordt ingezet
Oplossing	Beschrijving van de (technische) oplossing
Voorbeeld	Een of meer voorbeelden van problemen en contexten waarin dit patroon is toegepast

Rationale	Een verklaring of toelichting waarom dit patroon nodig is
Gerelateerde patronen	Verwijzing naar vergelijkbare patronen
Bestaande toepassingen	Verwijzing naar voorbeelden waar dit patroon is toegepast.

## Definitie bouwblokken

Bouwblokken zijn gedefinieerd binnen Togaf en hebben de volgende kenmerken [Togaf]:

- Een bouwsteen is een functionaliteitspakket dat is gedefinieerd om te voldoen aan de zakelijke behoeften binnen een organisatie.
- Een bouwsteen heeft interfaces gepubliceerd om toegang te krijgen tot de functionaliteit.
- Een bouwsteen kan samenwerken met andere, onderling afhankelijke, bouwstenen.
- Een goede bouwsteen heeft de volgende kenmerken:
- Het houdt rekening met de implementatie en het gebruik en ontwikkelt technologie en standaarden.
- Het kan worden samengesteld uit andere bouwstenen.
- Het kan een subsamenstelling van andere bouwstenen zijn.
- Idealiter is een bouwsteen herbruikbaar en vervangbaar.
- Een bouwsteen kan meerdere implementaties hebben, maar met verschillende onderling afhankelijke bouwstenen.

Een bouwsteen is daarom gewoon een pakket van functionaliteit dat is gedefinieerd om te voldoen aan functionele behoeften. De manier waarop functionaliteit, producten en aangepaste ontwikkelingen worden samengevoegd tot bouwstenen zal sterk verschillen tussen organisaties afhankelijk van de indeling van de organisatie maar ook door de indeling van de organisatie architectuur. Elke organisatie moet zelf beslissen welke indeling van bouwstenen voor hen het beste werkt. Een goede keuze aan bouwstenen kan leiden tot verbeteringen in legacy-systeemintegratie, interoperabiliteit en flexibiliteit bij het creëren van nieuwe systemen en applicaties.

Systemen zijn opgebouwd uit verzamelingen bouwstenen, dus de meeste bouwstenen moeten samenwerken met andere bouwstenen. Waar dat waar is, is het belangrijk dat de interfaces met een bouwsteen worden gepubliceerd en redelijk stabiel zijn.

Bouwstenen kunnen op verschillende detailniveaus worden gedefinieerd, afhankelijk van het stadium van architectuurontwikkeling.

In een vroeg stadium kan een bouwsteen bijvoorbeeld eenvoudigweg bestaan uit een groep functionaliteit zoals een standaardpakket met een aantal interfaces. Bouwstenen op dit functionele niveau van definitie worden in TOGAF beschreven als Architecture Building Blocks (ABB's). Later vervangen echte producten of specifieke aangepaste ontwikkelingen deze eenvoudige definities van functionaliteit, en de bouwstenen worden vervolgens beschreven als Solution Building Blocks (SBB's).

Kenmerkend is dat bouwblokken samenstellingen kunnen zijn. Hierdoor ontstaat een structurering die als aanbod vanuit de architecten aan de organisatie kan dienen. Zie bijvoorbeeld de bijlage met de voorbeelduitwerking in de bijlage.

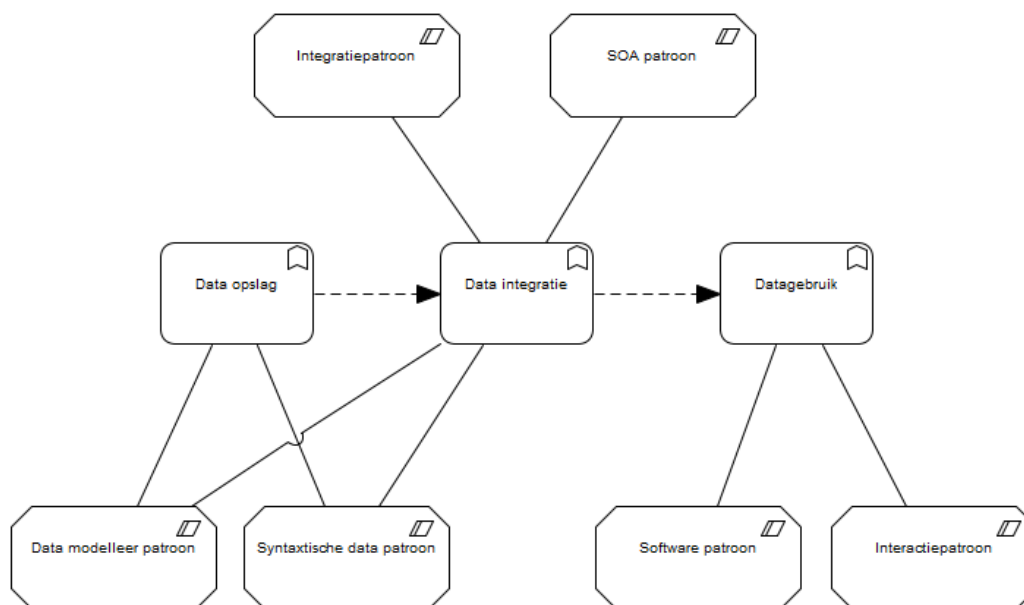


## Voorbeelden en catalogi

Patronen en bouwblokken komen in meerdere deelgebieden van de architectuur voor. Denk hierbij aan integratie patronen of aan data patronen. Zelfs zijn er architecturen waarbij de patronen de belangrijkste producten zijn. Denk hierbij aan SOA patronen. In een aantal gevallen zijn deze patronen in een catalogus of register opgenomen en worden daarbinnen op een gestandaardiseerde wijze beschreven. In dit hoofdstuk gaan we een aantal van deze voorbeelden en catalogi beschrijven.

### Datapatronen

Datamodellering is al tientallen jaren aanwezig binnen de informatie architectuur. Er zijn daardoor al heel veel patronen en bouwblokken aanwezig in dit werkveld. Dat kan zowel syntactisch, als technisch als semantisch zijn en hebben verschillende abstractieniveaus. Hieronder een indeling van de belangrijkste patronen waarbij data een rol speelt.



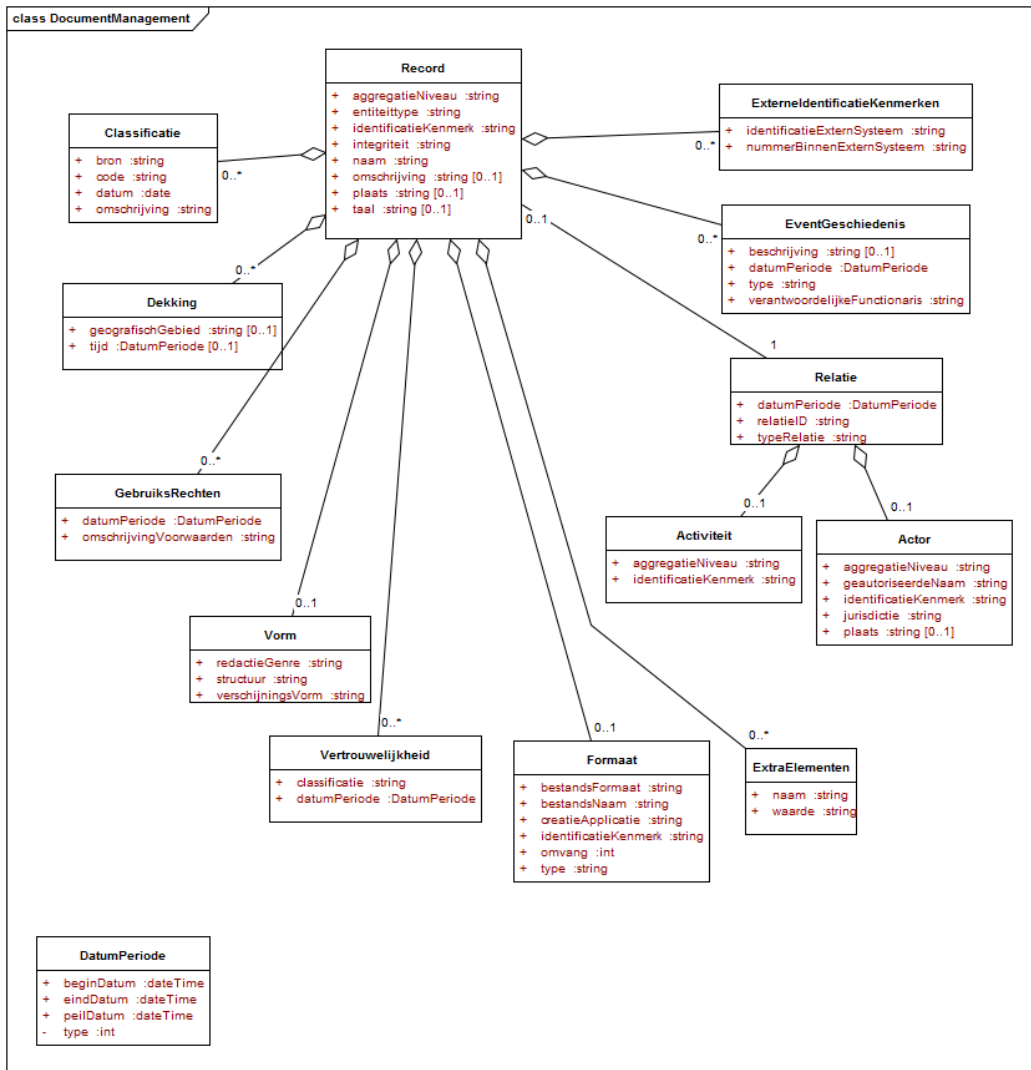
Afbeelding: Data functies en patronen

### Datamodel patronen

Datamodelpatronen zijn een beschrijving van best practices voor het uitwerken van datamodellen. Ze zijn met name interessant voor de data opslag maar komen ook voor binnen integratie en SOA patroontalen.

Idee achter deze patronen is dat bij het modelleren van data er grote delen van het datamodel generiek zijn. Bijvoorbeeld bij het modelleren van een persoon zijn bijna altijd voornaam, tussenvoegsel en achternaam relevant. Door modellen op te stellen van deze generieke entiteiten, eigenschappen en onderlinge relaties ontstaat een register van initiële modellen die ingezet kunnen worden bij het uitwerken van het eigen datamodel. Het is hierbij niet ongebruikelijk om meerdere van deze patronen met elkaar te combineren.

Datamodelpatronen bestaan al vele tientallen jaren, daardoor is literatuur rond dit onderwerp veelal al enige jaren oud. Dat is echter geen enkel bezwaar zie bijvoorbeeld de “Data model patterns, convention of thoughts” van David Hay wat nog steeds waardevolle modellen bevat. Een andere interessante bron zijn catalogie van registers zoals de stelselcatalogus van de Nederlandse overheid. Ook vermeldenswaardig zijn diverse open standaarden die voor de uitwisseling van gegevens in diverse branches datamodellen hebben ontwikkeld. Bijvoorbeeld UBL voor business entiteiten, Dublin Core voor document management of HRXML voor HRM modellen. In onderstaande afbeelding een voorbeeld van een patroon voor document management



Afbeelding: Patroon voor toepassingsprofiel document management.

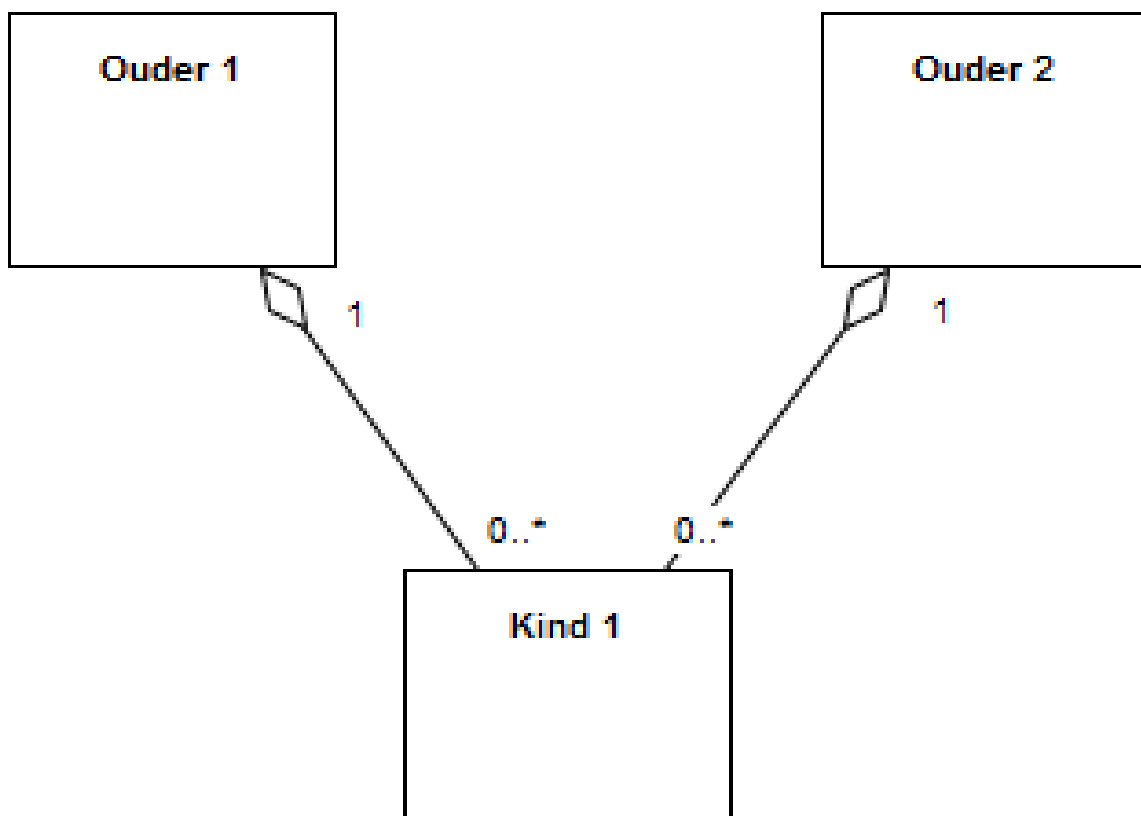
### Semantische patronen

In de voorgaande paragraaf zijn de concrete datapatronen aan de orde gekomen. Deze patronen geven een gedetailleerde uitwerking van de entiteiten en veelal ook de eigenschappen. Er is echter ook een meer abstract niveau van modelleren waarbij je minder over datamodelpatronen maar meer over data-ontwerppatronen spreekt.

Bij data-ontwerppatronen worden meer abstracte patronen gebruikt die een oplossing geven voor een bepaald veelvoorkomend modelleerprobleem zonder dat dit concreet is toe te wijzen aan een werkveld. In onderstaande opsomming een tweetal voorbeelden

- **Veel op veel relatie implementatie**, bij conceptuele modellering kunnen entiteiten op basis van veel op veel relaties aan elkaar gerelateerd zijn. Bijvoorbeeld een leraar heeft veel studenten en een student veel leraren. Dit wordt bij het uitwerken van een fysiek model omgezet in een koppeling van twee entiteiten vanwege het feit dat data opslag niet met veel op veel relaties kan omgaan.
- **Ouder-kind of bestaansafhankelijkheid relatie**, hierbij modelleer je een entiteit die gerelateerd is aan een andere entiteit op een dergelijke wijze dat de eerste niet kan bestaan zonder de andere. Denk bijvoorbeeld aan een orderregel die niet kan bestaan zonder een order maar veelal ook niet zonder een artikel of een dienst.

Deze conceptuele patronen zijn minder zichtbaar in een model maar bieden veelal een krachtige oplossing voor problemen, echter registers rond dit onderwerp zijn schaars. In onderstaand voorbeeld een eenvoudige ouder-kind en bestaansafhankelijkheid relatie in UML notatie. In de uitwerking is te zien dat de finesse in de details zit. Namelijk in de oudercardinaliteiten van de aggregaties. Deze is één en daarmee kan het kind niet bestaan zonder beide ouders.



Afbeelding: Bestaansafhankelijkheid patroon

## Canonieke data patronen

Datapatronen kunnen helpen bij het inrichten van de data-architectuur. Vandaar dat voor het uitwerken van een canonieke data-architectuur reeds aanwezige patronen een kader vormen. In de literatuur zijn een aantal relevante patronen te vinden [Hohpe] [Erl]:

- Canonical data model
- Canonical schema
- Canonical protocol
- Contract centralisation

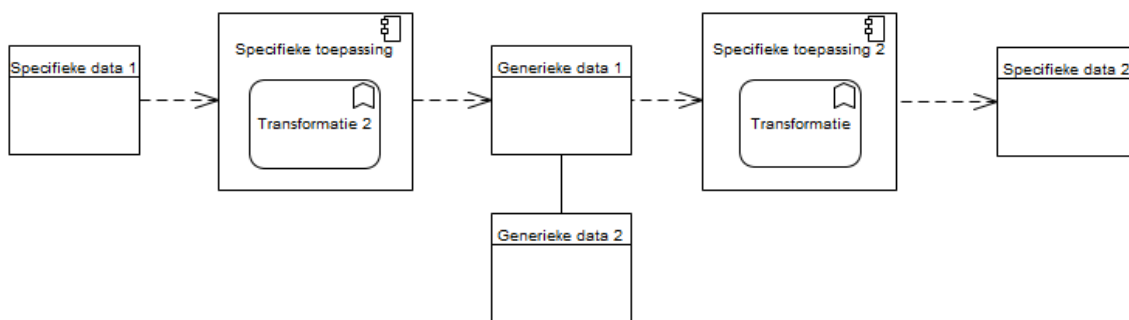
Canonical data model en canonical schema worden hieronder kort toegelicht.

### Canoniek data model

Dit patroon beschrijft een integratieoplossing waarbij voor interactie tussen twee of meer applicaties een gezamenlijk gestandaardiseerd datamodel geïntroduceerd wordt. Hierbij wordt het specifieke datamodel binnen de applicatie omgezet naar het gezamenlijke model. Gevolg hiervan is dan ook dat bij een integratie implementatie twee keer in plaats van eenmalig en conversie van datamodel plaats moet vinden. Dit is dan ook een veelgenoemd kritiekpunt voor dit patroon. Echter, zodra er meerdere applicaties behoefte hebben aan de data-entiteiten binnen deze implementatie, dan wordt dit patroon al snel interessant vanuit het oogpunt van efficiëntie.

In de uitwerking wordt ingegaan op de complexiteit van het introduceren van een canoniek datamodel. De kans op succes verkleint met name als gestreefd wordt naar een complete uitwerking van het datamodel voor een organisatie. Een stapsgewijze introductie waarbij de entiteiten uitgewerkt worden binnen het kader van integratieprojecten verhoogt de kans op succes.

Onderstaande afbeelding is een weergave van het patroon op basis van een integratie met twee bron- en twee doelsystemen.

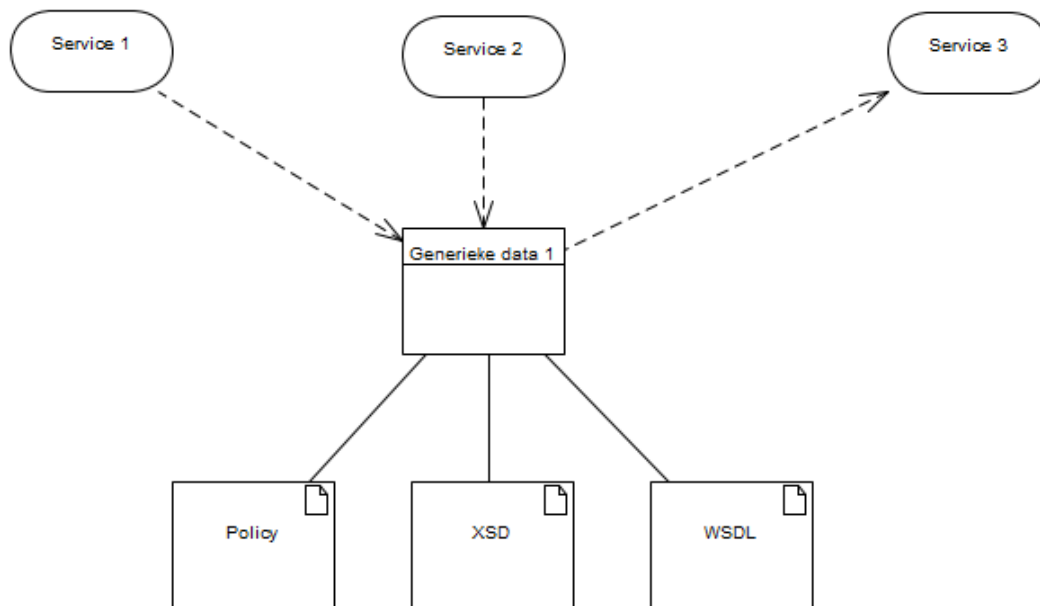


Afbeelding: Canoniek data model

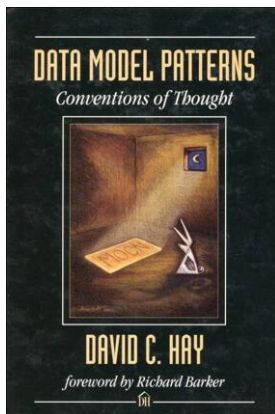
### Canoniek schema

Dit patroon is afkomstig uit de servicegeoriënteerde architectuur en richt zich dan ook op een aantal implementatieaspecten van service-oriëntatie. Het patroon is gericht op het uitwerken van een datamodel binnen de SOA-implementatie en gaat ervan uit dat het datamodel reeds aanwezig is. Zie voor meer details over SOA patronen de uitwerking in de volgende paragraaf

Echter, in de praktijk maak ik regelmatig mee dat het canonieke datamodel nog niet aanwezig is en dus gelijktijdig ontwikkeld wordt met de inrichting van de service-architectuur. Zeker bij generieke gegevensverzamelingen zoals relatiegegevens, producten en ondersteunende sets zal deze werkwijze in meerdere iteraties verijnd worden. Bij het inrichten van de schema's dient hier rekening mee gehouden te worden.



Bouwblokken zijn binnen de data modellering eveneens veelvuldig toegepast. Denk hierbij bijvoorbeeld aan uitgewerkte bouwblokken voor bepaalde sectoren zoals banken en verzekeringsmaatschappijen. Ander bekend voorbeeld is het Common Information Model van de energiesector. Dit zijn de meer concrete modellen die als startpunt kunnen dienen voor specifieke implementaties binnen de eigen organisatie, Ook zijn er meer abstracte bouwblokken zie hiervoor bijvoorbeeld het boek van David Hay met daarin het metamodel voor data modellering uitgewerkt (<https://epdf.pub/data-model-patterns-a-metadata-map.html>)



### Catalogi

Op het vlak van data modellering zijn er een groot aantal data modelleer catalogi en publicaties. Een belangrijke catalogus is het boek van David Hay Data Model Patterns.

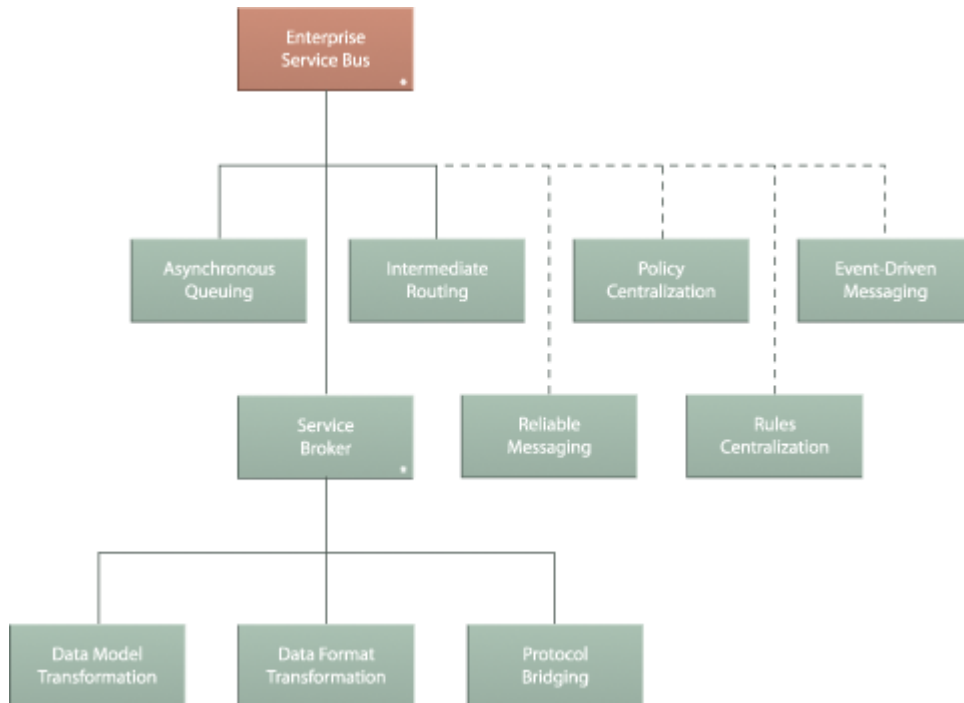
Daarnaast zijn er een aantal sectorale catalogi:

- [Energiesector:](#)
- [Nederlandse Overheid](#)
- [Verzekeringen](#)
- [Data warehousing](#)

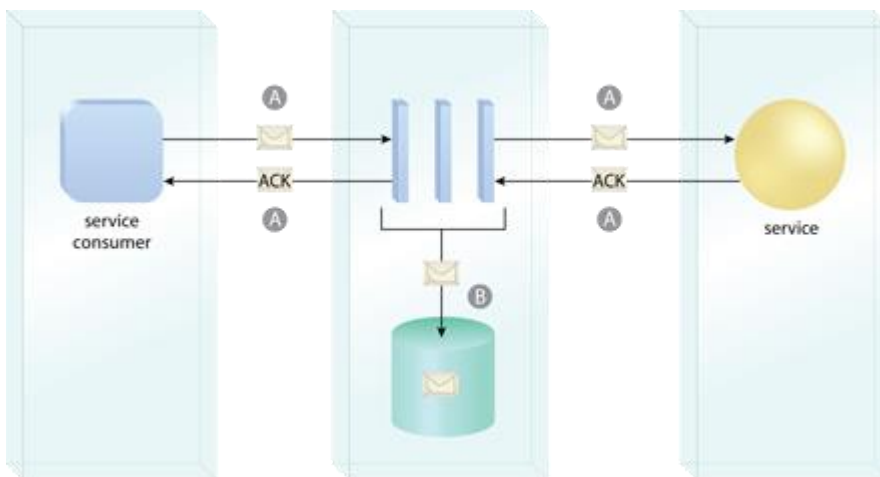
### Integratie- en SOA patronen

Integratie en SOA patronen richten zich onder andere op data integratie, veelal zijn deze patronen gericht op het transformeren van data. Daarnaast zijn er patronen gericht op verschillende transport- en opslagmechanismen. Een goed voorbeeld is de Enterprise Service Bus, een concept dat bij veel organisaties conceptueel of fysiek wordt ingezet. Een ESB is de implementatie van samengestelde SOA patronen (zie: [http://soapatterns.org/compound\\_patterns/enterprise\\_service\\_bus](http://soapatterns.org/compound_patterns/enterprise_service_bus))

In onderstaande afbeelding een voorbeeld van dit samengestelde patroon. Het is opgebouwd uit meerdere patronen die zelf ook weer samengesteld kunnen zijn. In onderstaande afbeeldingen een voorbeeld van een samengesteld en een basispatroon.



Afbeelding: Enterprise Service Bus (Bron: <http://www.soapatterns.org>)



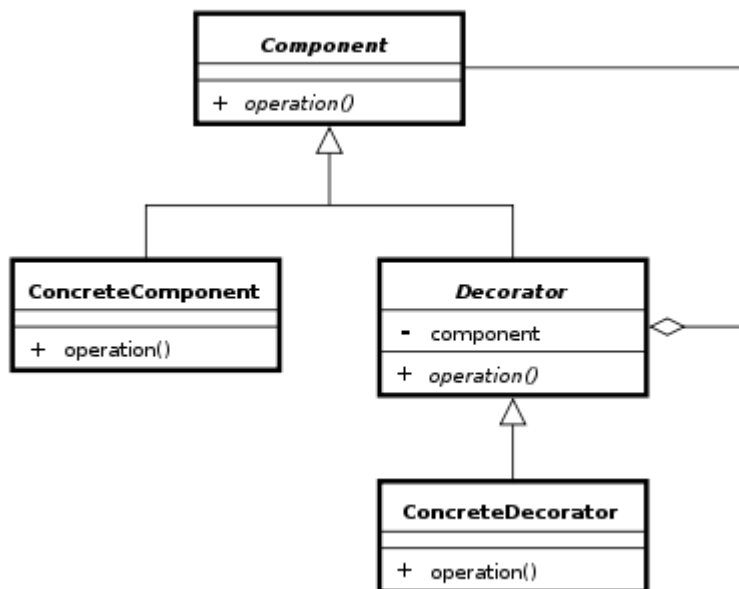
Afbeelding: Reliable messaging (Bron: <http://www.soapatterns.org>)

Bijzonder aan deze patronen is dat het zowel een ABB of functioneel bouwblok is maar ook als Solution Building Block aanwezig is. Zo zijn er meerdere implementaties van dit patroon met eenzelfde naam ESB. Denk bijvoorbeeld aan producten van Oracle, Tibco en Mulesoft.

Op het vlak van applicatie integratie en SOA zijn meerdere patronen en patroonregisters aanwezig. Zie bijvoorbeeld <http://www.eaipatterns.com> of <http://www.soapatterns.org>. Deze registers kennen patronen op meerdere integratiewerkvelden waaronder data transformatie en – representatie.

## Softwarepatronen

Dit zijn de meest bekende patronen op dit moment, worden veelvuldig in software toegepast. In een aantal gevallen is niet eens meer bekend dat het origineel een patroon is. Binnen dit werkveld zijn de patronen van de Gang of Four de meest bekend. Hieronder een voorbeeld van een van deze patronen:



Afbeelding: Decorator (Bron: [Wikipedia](http://Wikipedia))

Ook hierbij een mooi voorbeeld van een combinatie van ABB en SBB. De ABB zijn de beschrijvingen van de patronen (veelal in catalogi zoals de hierboven benoemde Gang of Four Patterns). De SBB zijn beschikbaar als functionaliteiten in ontwikkeltools als Visual Studio en kunnen een programmeur snel op weg helpen doordat grote delen van de broncode op basis van deze bouwblokken gegenereerd kunnen worden (zie bijvoorbeeld <https://www.dofactory.com/net/design-patterns>)

Op het vlak van software patronen zijn er een veelheid aan catalogi te vinden. Zie bijvoorbeeld voor de GoF patterns: <http://www.blackwasp.co.uk/gofpatterns.aspx> of [https://www.tutorialspoint.com/design\\_pattern/design\\_pattern\\_overview.htm](https://www.tutorialspoint.com/design_pattern/design_pattern_overview.htm) waar je op basis van een tutorial de patronen kunt leren.

## Interactiepatronen

Interessant en minder bekend zijn interactiepatronen. Dit zijn patronen voor de grafische gebruikersinterface en voor Human Computer Interaction.

Op het vlak van interactiepatronen is voor een architect veel te bereiken. Het uitwerken van een beperkte set van interactiepatronen die ingezet mogen worden voor de gebruikersinterface van informatiesystemen en websites e.d. vergroot het gebruiksgemak.

In de afbeelding hieronder zie je een voorbeeld van een dergelijk patroon uitgewerkt



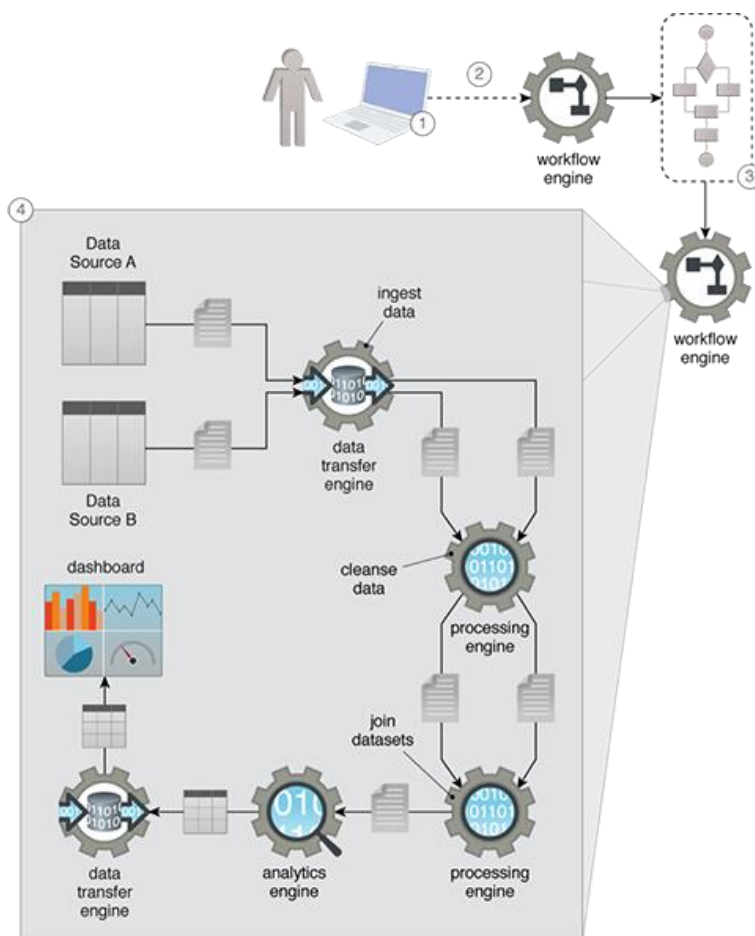
(<http://designinginterfaces.com/patterns/two-panel-selector/>)

Bron: <http://designinginterfaces.com/patterns/>.

Op het vlak van interactiepatronen zijn een aantal goede registers (en literatuur) te vinden. Zie bijvoorbeeld het register <http://designinginterfaces.com/patterns/>.

## Big Data Patterns

Ook binnen modern werkvelden van de informatica hebben patronen voordelen. Zo zijn er patroon catalogic voor Big Data, IoT en Machine Learning en Artificial Intelligence. Arcitura is een partij die deze catalogic actief beheerd en doorontwikkeld. Hieronder een voorbeeld van een Big Data Patroon.



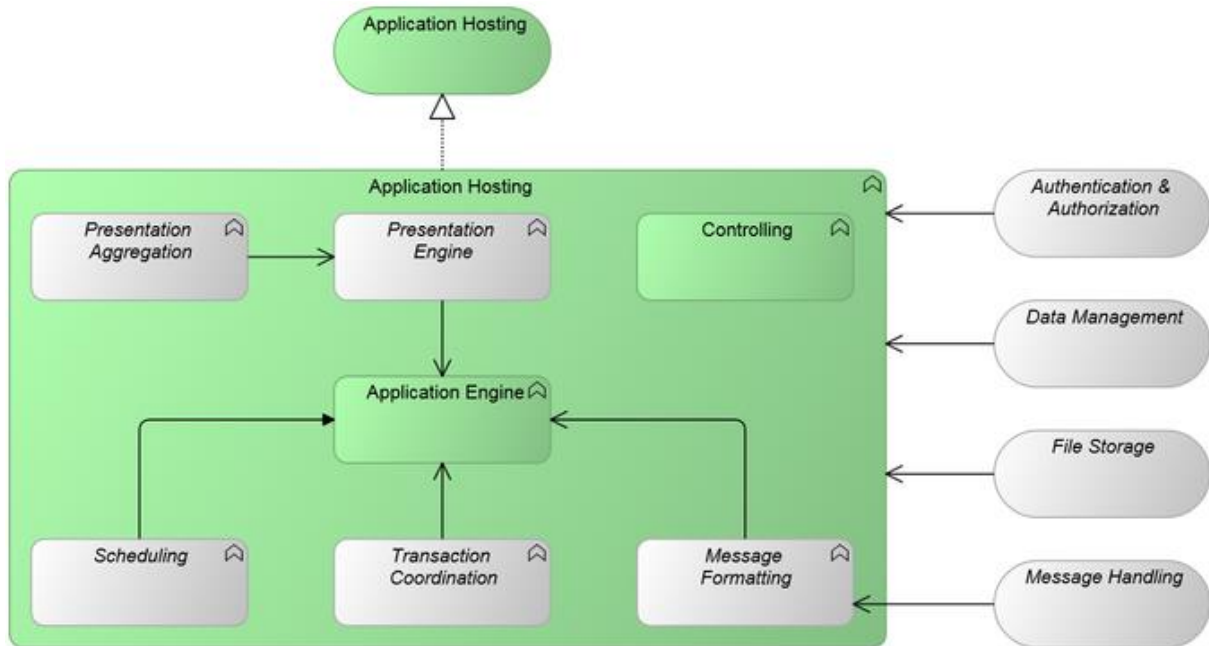
Afbeelding: Automated dataset execution (Bron: [Arcitura](http://Arcitura))

## Infrastructuur patronen

Laatste patronen die we gaan behandelen in dit white paper zijn infrastructuur patronen. Opnieuw een mooi voorbeeld van een ABB opzet waarbij je op eenvoudige wijze de eigen SBB's kunt relateren aan het uitgewerkte landschap.



In de Open Infrastructure Architecture Repository kun je een aantal uitwerken vinden vergelijkbaar met onderstaande afbeelding



Afbeelding: Applicatie hosting (Bron [OJAR](#))

## Voorbeeld

In de hierna volgende bijlage wordt een voorbeeld van een uitwerking van bouwblokken binnen een architectuur uitgewerkt. Dit is een voorbeeld van een uitwerking voor een ABB en SBB modellen voor een ICT organisatie

# Bijlage 1 Voorbeeld toepassen bouwblokken

## Inleiding

Dit is een voorbeeld van een uitwerking van een bouwblokken model opgesteld door een overheidsorganisatie. Het is gebaseerd op het model van ABB en SBB zoals beschreven in Togaf. Naast de definitie van de bouwblokken is er een uitwerking van een voorbeeld opgenomen.

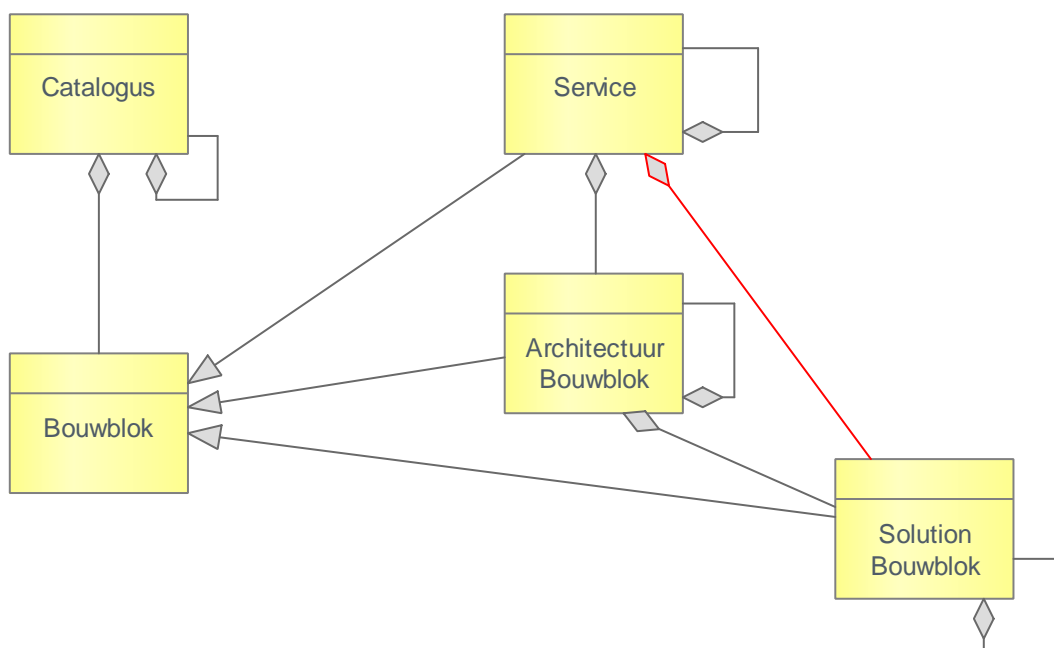
## Objectmodel Bouwblokken

Het objectmodel beschrijft het concept bouwblok zoals dat gedefinieerd is binnen het architectuurproces van XXX. Bouwblokken zijn communicatieve concepten tussen architecten onderling en tussen architectuur en de verschillende stakeholders binnen XXX, de diensten en eventueel externe stakeholders zoals leveranciers of ketenpartners.

Het model bestaat uit een beperkte set aan concepten met onderlinge relaties. Dit model is uitgewerkt in een ArchiMate business objecten diagram.

De concepten in het objecten en definitie diagram zijn vervolgens in detail uitgewerkt en beschrijven hiermee de kaders van de bouwblokken.

## Objecten en definities



Het bouwblokken model bestaat uit de generieke entiteiten bouwblok en catalogus. Deze vormen de basis van een referentie architectuur. Catalogi zijn groeperingen van bouwblokken binnen een

bepaald domein. Er zullen meerdere catalogi ontstaan die aan elkaar gerelateerd zijn en elkaar overlappen binnen de visualisaties.

Het voordeel van de opzet van het werken met catalogi en bouwblokken is:

- Er ontstaan registers van herbruikbare architecturale onderdelen gericht op een bepaald werkveld
- Inzet van bouwblokken brengt standaardisatie met zich met en stimuleert hergebruik van architecturale configuraties
- Bouwblokken maakt het architecturale- en het ontwikkelproces eenvoudiger.
- Er ontwikkelen zich architecturale product catalogi gericht op specifieke werkvelden. Dit heeft een positieve invloed op de dienstverlening van XXX.

De bouwblokken kennen drie specialisaties waarvan de definities in detail zijn uitgewerkt. Deze beschrijvingen zijn hieronder in de paragrafen uitgewerkt. Indien relevant is aan deze uitwerking extra informatie toegevoegd zoals links naar Togaf en voorbeelden van de implementatie van deze architecturale concepten

In het model wordt gewerkt met een pragmatisch model voor wat betreft de associaties tussen Service en SBB. Vanuit ArchiMate perspectief is de route van service via ABB naar SBB. Dit heeft als kenmerk dat het functionele aspect goed ingebed is in het bouwblokken model. Bij het publiceren van deze modellen wordt gezocht naar een mogelijkheid om rapporten en webpagina's te genereren die voor niet architectuur stakeholders geen gegevens bevatten niet relevant voor het model, dat kan betekenen dat in een aantal situaties de ABBs niet uitgewerkt hoeven te worden. Het in de modellering toepassen van de directe associatie tussen service en SBB voldoet niet aan de viewpoints.

## Architectuur Bouwblok

### Statement

*Een architectuur bouwblok is de logische definitie van een functionaliteit*

### Omschrijving

Voor een architectuur bouwblok wordt de afkorting ABB gebruikt. Een architectuur bouwblok beschrijft de functionaliteiten die aangeboden worden aan een hogerliggende entiteit.

Een ABB beschrijft WAT er nodig is, zonder te schrijven naar een specifieke oplossing. De hoger liggende entiteit kan een service of een samengestelde ABB zijn.

Een ABB kan samengesteld zijn uit één of meerdere SBB. Deze SBB zijn de implementatie van de functionaliteit. Met andere woorden de SBB realiseren de ABB.

### Kenmerken

- Beschrijving van een functionaliteit
- Beschrijving van het gedrag van informatievoorziening elementen zonder kenmerken van fysieke implementatie
- ABBs zijn logisch, zonder technische specificatie of merknamen
- Infrastructurele- en applicatielaag zijn in de huidige fase van dit model het belangrijkste toepassingsgebied.
- Architectuur bouwblokken zijn gerelateerd aan kwaliteiten, constraints en principes.
- Dit is het kader waarbinnen bijv. een productmanager een product kan selecteren.
- Wanneer een product aan het einde van de LCM is, kunnen de kaders in het ABB opnieuw worden gebruikt om een nieuw product te selecteren.
- Uitgangspunt is het voorkomen dat een ABB wordt geschreven naar een beschikbare oplossing. Deze dient daarom oplossing- en technologieneutraal te zijn.

Meer informatie	
Omschrijving	Hyperlink

Meer informatie	
Omschrijving	Hyperlink
Definitie van een ABB binnen Togaf	<a href="https://pubs.opengroup.org/architecture/toga-f9-doc/arch/chap03.html#tag_03_08">https://pubs.opengroup.org/architecture/toga-f9-doc/arch/chap03.html#tag_03_08</a>

## Bouwblok

### Statement

*Een bouwblok is een afgebakend en herkenbaar architectureel concept dat invulling geeft aan een informatievoorzienings behoefte.*

### Omschrijving

Een bouwblok is in dit document een abstract concept dat in drie concrete specialisaties is uitgewerkt, service, ABB en SBB. Bouwblokken worden binnen onze definities als synoniem gezien van een architectuur patroon.

De drie specialisaties kennen een hiërarchie. Een service is in ons model het hoogste abstractieniveau, daarbinnen worden Architectuur bouwblokken onderkend welke geïmplementeerd worden door een of meerdere solutions bouwblokken.

Een belangrijke karakteristiek van bouwblokken is dat ze samengesteld kunnen zijn. Deze samenstelling kan op twee manieren:

- Een bouwblok is een samenstelling van bouwblokken van dezelfde specialisatie (bijvoorbeeld een service is opgebouwd uit een of meerdere subservices)
- Een bouwblok wordt bediend door een samenstelling van een of meerdere bouwblokken uit de onderliggende laag (bijvoorbeeld een service wordt gerealiseerd door meerdere gedrags-elementen in een ABB.
- Samengestelde bouwblokken worden binnen onze definities als synoniem gezien van een architectuur patroon.

Bij de specialisaties van de bouwblokken wordt de samenstelling voor de specialisaties uitgewerkt. De samenstelling binnen bouwblokken kan uit meerdere lagen bestaan. Echter het is wenselijk dat het aantal niveaus van bouwblokken binnen een specialisatie (Service, ABB of SBB) beperkt blijft. Wordt een catalogus te complex door het aantal lagen en bouwblokken dan kan beter een splitsing van een catalogus toegepast worden.

### Kenmerken

- Een bouwblok heeft een afgebakende grens en is herkenbaar als een specifiek architectureel concept.
- Een bouwblok is herbruikbaar
- Een bouwblok is loosely coupled.
- Een bouwblok kan interacteren met een of meerdere andere bouwblokken van verschillende typen.
- Een bouwblok is onderdeel van een catalogus.
- Template voor een oplevering die aan een combinatie van eisen en wensen voldoet.
- Een bouwblok kan bestaan uit andere bouwblokken en wordt daarmee een **samengesteld bouwblok**.
- Een bouwblok kan onderdeel zijn van een compositie (**samengesteld bouwblok**).
- Bouwblokken kunnen recursief zijn en worden in dat geval samengesteld.

Meer informatie	
Omschrijving	Hyperlink
Definitie van een bouwblok binnen Togaf	<a href="https://pubs.opengroup.org/architecture/toga-f9-doc/arch/chap03.html#tag_03_23">https://pubs.opengroup.org/architecture/toga-f9-doc/arch/chap03.html#tag_03_23</a>

## Catalogus

## Statement

*Een collectie van logisch gerelateerde bouwblokken van dezelfde specialisatie (service, ABB of SBB).*

## Omschrijving

Een catalogus is een collectie of register van bouwblokken van eenzelfde type. Het is veelal gericht op een specifiek werkveld, denk hierbij bijvoorbeeld aan infrastructuur, geo, integratie.

Binnen een catalogus zitten veelal bouwblokken van dezelfde specialisatie dus services, architectuur - of solution bouwblokken. Echter de bouwblokken hierbinnen zullen veelal eveneens samengesteld zijn.

Een catalogus kan gezien worden als een etalage van generieke en herbruikbare architecturale producten. Wanneer deze bouwblokken door een project worden ingezet wordt voldaan aan een aantal architecturale eisen, principes en requirements. Voordeel voor architectuur is dat deze bouwblokken worden hergebruikt. Voordeel vanuit projectperspectief is dat er voldaan wordt aan de architecturale principes en dat implementatie gestandaardiseerd wordt en waarschijnlijk sneller kan.

Vanuit veranderingen in de omgeving (projecten, LCM, innovaties) zal de inhoud van een catalogus regelmatig worden aangepast, uitgebreid of meer gedetailleerd worden uitgewerkt. Een catalogus en de daarin opgenomen entiteiten wordt daarmee een "levend" ecosysteem.

In eerste instantie wordt gewerkt met een aanbod gestuurd catalogus model. Met andere woorden. Iedere domeinarchitect maakt voor zijn domein een bouwblokken catalogus. In een later stadium wordt dit aangepast naar een vraaggerichte uitwerking, het zogenaamde etalagemodel.

## Kenmerken

- Collectie van bouwblokken
- Bouwblokken van hetzelfde (architectuur) concept kunnen opgenomen worden in een catalogus
- Catalogi worden gecategoriseerd op basis van een scope. (bijvoorbeeld, infrastructuur, integratie, geo)
- Catalogi binnen een scope hebben een eigenaar
- Catalogi worden beschreven in een register (beheerd in Sparx Enterprise Architect en gepubliceerd naar KIS en PDF documenten)
- Catalogi zijn veelal hiërarchisch cq gelaagd van opzet. Enerzijds door de indeling in Service, ABB en SBB, anderzijds door de opzet met samengestelde bouwblokken

Meer informatie	
Omschrijving	Hyperlink
Service portfolio definitie binnen Togaf	<a href="https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html#tag_03_68">https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html#tag_03_68</a>
Voorbeeld van een breed toegepaste patroon catalogus	<a href="https://patterns.arcitura.com/big-data-patterns">https://patterns.arcitura.com/big-data-patterns</a>

## Service

### Statement

*Services zijn de beschrijving van een combinatie van functionaliteiten en diensten tussen aanbieder(s) en afnemer(s).*

### Omschrijving

Services zijn een vorm van inkapseling van de functionaliteit en implementatie van een samenstelling van bouwblokken. Services worden, net als ABB en SBB, gebruikt als communicatiemiddel om tussen aanbieder (XXX) en afnemer (Diensten) aan te geven welke dienst door de aanbieder aan de afnemer geleverd gaat worden. Services kunnen ook intern binnen XXX

gedefinieerd zijn (ook in in aanbieder en afnemer verband), bijvoorbeeld infrastructurele services voor een applicatieve service of afnemer.

Een service kan een samenstelling zijn van een bouwblok dat functionaliteit implementeert binnen het ICT landschap. Daarnaast kan een service bestaan uit het leveren van een meer ingerichte (ICT) werkprocessen in relatie met de bovengenoemde ICT landschappen, bijvoorbeeld een servicedesk. In dit document hebben we de scope beperkt gehouden tot die van de ICT architectuur, ICT werkprocessen worden hier niet uitgewerkt maar zijn binnen andere delen van XXX zeker relevant (service management).

Dit bouwblokken model kan desgewenst op meerdere manieren toegepast worden en niet alleen in het ICT werkveld. Hier beperken we ons tot ICT architectuur.

Services kunnen samengesteld zijn uit onderliggende services. Daarnaast kunnen zij opgebouwd zijn uit een of meerdere architectuur bouwblokken. Door deze samenstellingen kunnen constellaties ontstaan van bouwblokken die zorgdragen voor standaardisatie van herbruikbare toegepaste services. Denk hierbij aan een standaard ingerichte applicatieserver met services (zoals backup en restore) en ABB (bijvoorbeeld relationele opslag)

Services zijn gerelateerd aan requirements, constraints en principes. Dit is bij voorkeur uitgewerkt om aan te geven aan welke behoeften vanuit afnemersperspectief een invulling wordt gegeven en aan welke niet.

Rond de term service bestaat veel verwarring. Om te voorkomen dat bij iedere samenstelling van personen een discussie over de definitie ontstaat wordt gekozen voor de term service welke gebaseerd is op onderstaande kenmerken. Daarnaast is er een lijst van synoniemen geformuleerd die verwijzen naar dezelfde onderstaande kenmerken.

### Kenmerken

- Een herhaalbare activiteit of gedrag die wordt gevraagd te worden uitgevoerd
- Een service biedt een of meerdere voor de afnemer begrijpelijke invulling van ICT behoeften
- Combinatie van een implementatie van een functionaliteit in een of meerdere ABBS
- Eventueel in combinatie met een of meer ICT werkprocessen
- Services worden aangeboden aan afnemers vanuit aanbieders
- Services hebben een commercieel en een financieel (kosten) aspect
- Services kennen voorwaarden voor gebruik cq implementatie.

Meer informatie	
Omschrijving	Hyperlink
Definitie van een service conform Togaf	<a href="https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html#tag_03_65">https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html#tag_03_65</a>

## Solution Bouwblok

### Statement

*Een solution bouwblok is de fysieke implementatie van een functionaliteit in één of meerdere ABB.*

### Omschrijving

Voor een solution bouwblok wordt de afkorting SBB gebruikt. Een SBB beschrijft de implementatie waarmee een functionaliteit gerealiseerd wordt. Een SBB biedt deze implementatie aan aan een hogerliggende entiteit. In ons model is een SBB implementatie van ABB of van een samengestelde SBB.

Hierbij is het relevant dat er een onderscheid gemaakt wordt in architecturale lagen. Voor ons zijn de infrastructurele- en de applicatielaag het belangrijkste toepassingsgebied. Een SBB op applicatieniveau kan hiermee een samenstelling zijn van SBBs op zowel de applicatie- als de infrastructurele laag.

SBB zijn gerelateerd aan kwaliteiten, constraints en principes. Dit is bij voorkeur uitgewerkt om aan te geven aan welke vereisten vanuit implementatie perspectief een invulling wordt gegeven en

aan welke niet. Dit in combinatie met het model van kwaliteiten binnen de ABB en de requirements zoals op serviceniveau zijn uitgewerkt biedt een complete beschrijving van de kenmerken die door een service worden aangeboden.

### Kenmerken

- SBB is een fysieke implementatie van een (deel van) of meerdere ABBs cq functies
- Technische en productspecificatie zijn bekend
- Merk- en leveranciersnamen zijn bekend.
- Een SBB is veelal vervangbaar door een ander product of implementatie

Meer informatie	
Omschrijving	Hyperlink
Definitie van een SBB binnen Togaf	<a href="https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html#tag_03_70">https://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html#tag_03_70</a>

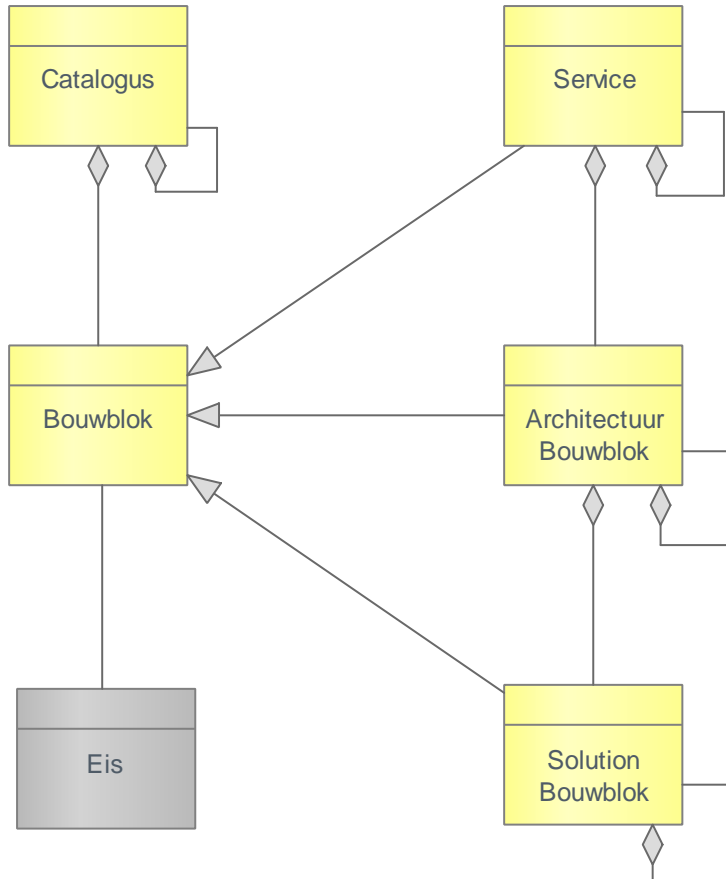
### Objecten, definities en eisen

Bouwblokken met een uitbreiding van een secundaire scope rond de eisen en karakteristieken die aan de verschillende bouwblokken kunnen worden gesteld. Feitelijk de invulling van de vraag- en aanbodzijde voor de bouwblokken.

In de viewpoints uitwerking is een voorstel uitgewerkt welke ArchiMate concepten gebruikt worden voor welk bouwblok specialisatie

## ArchiMate Viewpoints voor Bouwblokken

In dit hoofdstuk wordt een voorstel gedaan voor een aantal ArchiMate viewpoints voor het



modelleren van de verschillende bouwblokken en hun onderlinge relaties.

In de diagrammen binnen dit hoofdstuk worden de viewpoints alleen uitgewerkt op basis van de elementen en de relevante onderlinge associaties. Een beschrijving van de concepten zelf wordt niet gedaan. Hierbij nemen we definities en mogelijke associaties over zoals die gedefinieerd zijn binnen de modelleertaal ArchiMate zelf.

ArchiMate kent in het core model drie lagen, namelijk Business, Application en Technology layer. xBB kunnen toegepast worden in de drie hierboven genoemde lagen. Echter omdat het perspectief in dit document voor de xBB voornamelijk ligt op applicatie en infrastructuur zijn de viewpoints alleen voor deze twee lagen uitgewerkt.

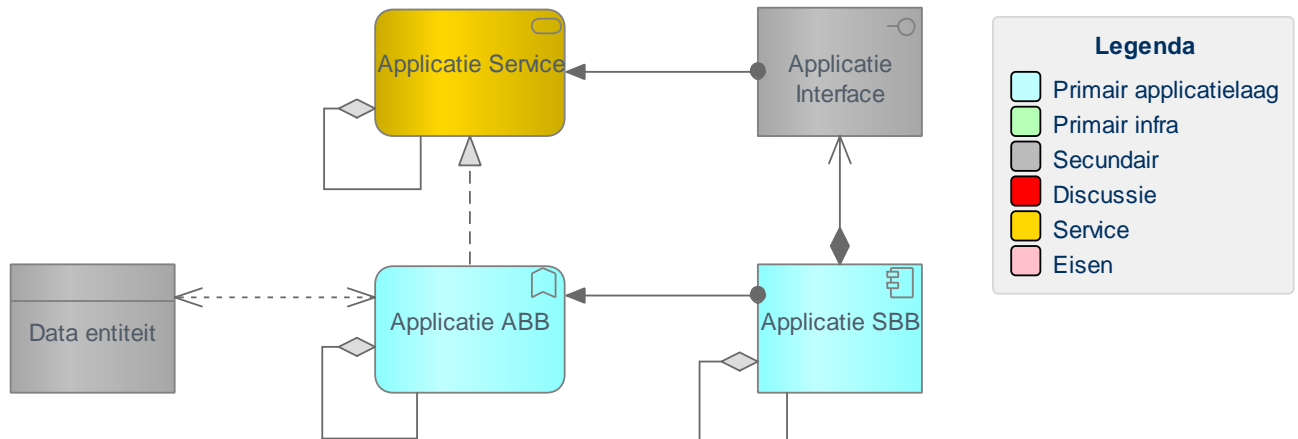
Voor de ABB wordt in ArchiMate gebruik gemaakt van de Behaviour kolom. Sinds ArchiMate 3 bestaan binnen deze kolom meerdere elementen. Bij de uitwerking in de viewpoints worden alleen de Application\_Function en Technology\_Function gebruikt. Is een ander concept bijvoorbeeld een Application\_Process of Technology\_Process relevant bij een uitwerking dan kan dit vanzelfsprekend ook toegepast worden.

Voor de SBB wordt in ArchiMate gebruik gemaakt van de Active Structure kolom. Met name binnen de technology layer zijn veel verschillende concepten beschikbaar. Bij de uitwerking in de viewpoints wordt alleen de System\_Software gebruikt. Is een ander concept relevant bij een uitwerking in deze dimensie dan kan dit vanzelfsprekend ook toegepast worden.



## Viewpoint Bouwblokken Basis Applicatielaag

### Primaire concepten



In de applicatielaag zijn de drie specialisaties van de bouwblokken relatief eenvoudig te relateren aan een ArchiMate element, namelijk:

- Service <-> Application\_Service
- ABB <-> Application\_Function, zoals reeds genoemd kan hier ook een ander behavioural element gebruikt worden
- SBB <-> Application\_Component

Tussen de elementen kunnen ArchiMate relaties gedefinieerd worden:

- Service <-> ABB: Realisation
- SBB <-> ABB: Assignment
- xBB <-> xBB: Aggregation

Met name de laatste associatie de aggregatie is van belang omdat hiermee samengestelde bouwblokken samengesteld kunnen worden.

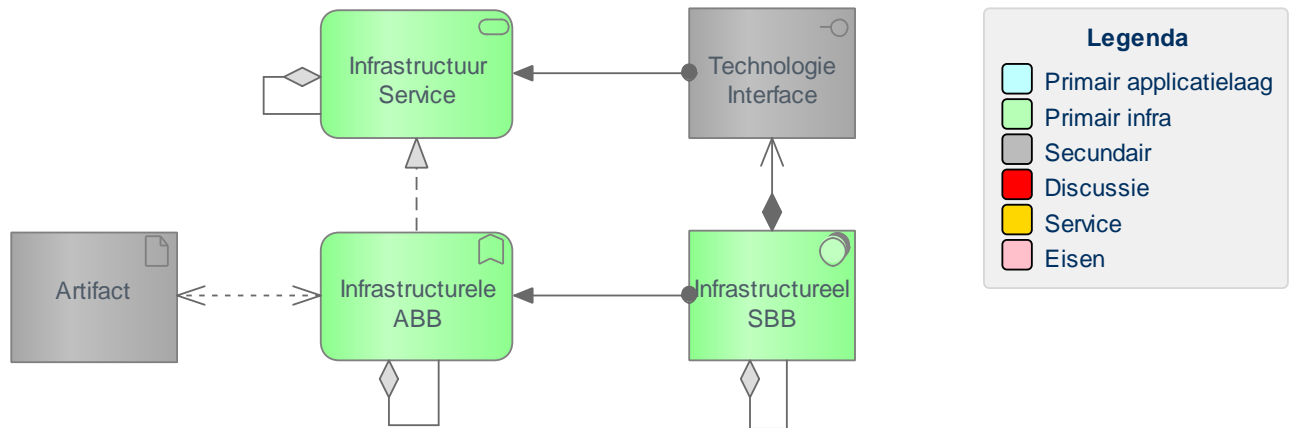
Naast de genoemde associaties zijn meerdere typen associaties te kiezen zoals de dynamische associaties. Bij het uitwerken van de views binnen dit viewpoint staat het je vrij deze extra associaties toe te passen, mits uitgewerkt in de reeds aanwezige algemene XXX viewpoints.

### Secundaire concepten

Naast de primaire elementen en associaties zijn er een tweetal elementen en associaties relevant, echter niet in alle architectuurdomeinen. Dit zijn:

- Data\_Object, binnen bijvoorbeeld de integratie architectuur zijn data objecten noodzakelijk voor het beschrijven van bijvoorbeeld herbruikbare berichtdefinities binnen een bouwblok
- Application\_Interface, eveneens binnen de integratie architectuur is voor de implementatie van bijvoorbeeld een webservice dit concept noodzakelijk als extra ArchiMate element binnen de xBB modellering.

## Viewpoint Bouwblokken Technische laag



### Primaire concepten

In de technische laag zijn de drie specialisaties van de bouwblokken relatief eenvoudig te relateren aan een ArchiMate element, namelijk:

- Service <-> Technology\_Service
- ABB <-> Technology\_Function, zoals reeds genoemd kan hier ook een ander behavioural element gebruikt worden
- SBB <-> System\_Software, Node, Network, Device, Path en andere technische actieve structuur elementen.

Tussen de elementen kunnen ArchiMate relaties gedefinieerd worden:

- Service <-> ABB: Realisation
- SBB <-> ABB: Assignment
- xBB <-> xBB: Aggregation

Met name de laatste associatie de aggregatie is van belang omdat hiermee samengestelde bouwblokken samengesteld kunnen worden.

Naast de genoemde associaties zijn meerdere typen associaties te kiezen zoals de dynamische associaties. Bij het uitwerken van de views binnen dit viewpoint staat het je vrij deze extra associaties toe te passen, mits uitgewerkt in de reeds aanwezige algemene XXX viewpoints.

### Secundaire concepten

Naast de primaire elementen en associaties is er element en associatie relevant, echter niet in alle architectuurdomeinen. Dit zijn:

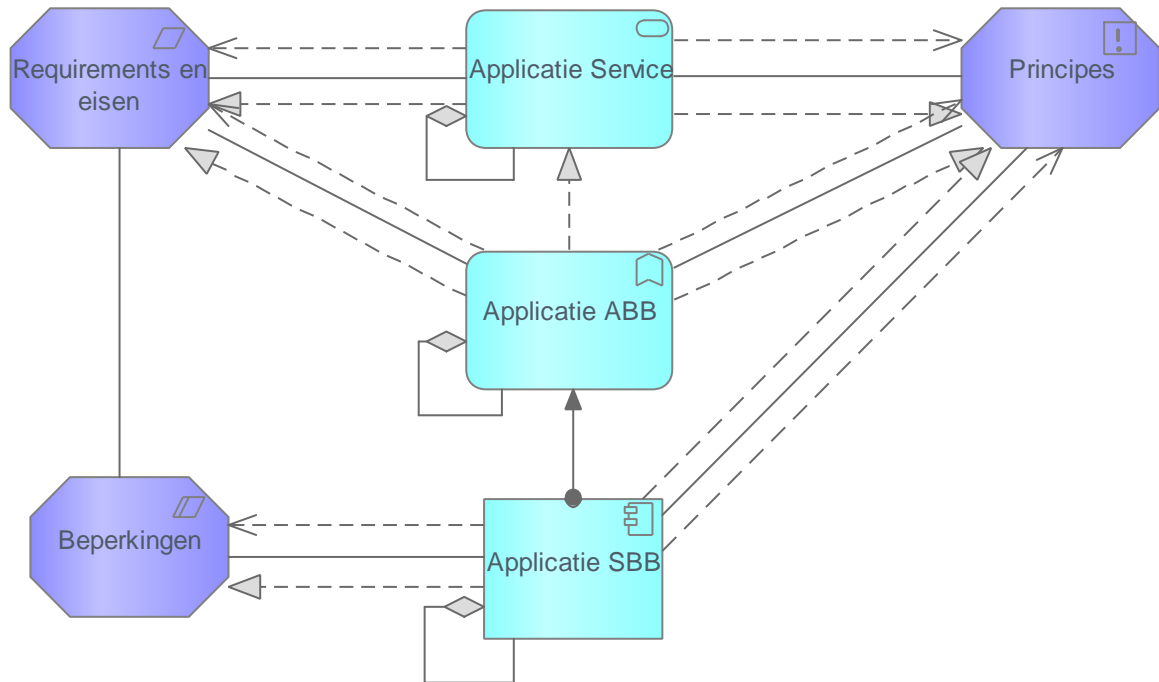
- Technology\_Interface, eveneens binnen de integratie architectuur is voor de implementatie van bijvoorbeeld een webservice dit concept noodzakelijk als extra ArchiMate element binnen de xBB modellering.
- Een introductie van een artifact is in deze alleen in bepaalde deelgebieden relevant (geo). In andere gevallen zal worden uitgeweken naar een taal met meer detail zoals UML klassediagram of XSD modellen. In dat laatste geval wordt er via een trace associatie gelegd tussen de modelleertaal concepten.

## Artifact

### Naamgevingsconventie

Gebruik: Fysieke Naam van het bestand. Bijvoorbeeld: Order.jar, vraag.doc, klant.xls, Naam aangevuld met de suffix

## Viewpoint Bouwblokken en Eisen



Belangrijke extra dimensie bij het beschrijven van de xBB zijn de kenmerken die bij de communicatie tussen aanbieder van het xBB en de afnemer relevant zijn. Deze extra dimensie richt zich met name op de kenmerken waaraan een xBB wel of niet voldoet. Dit kunnen beperkingen, regels of principes zijn.

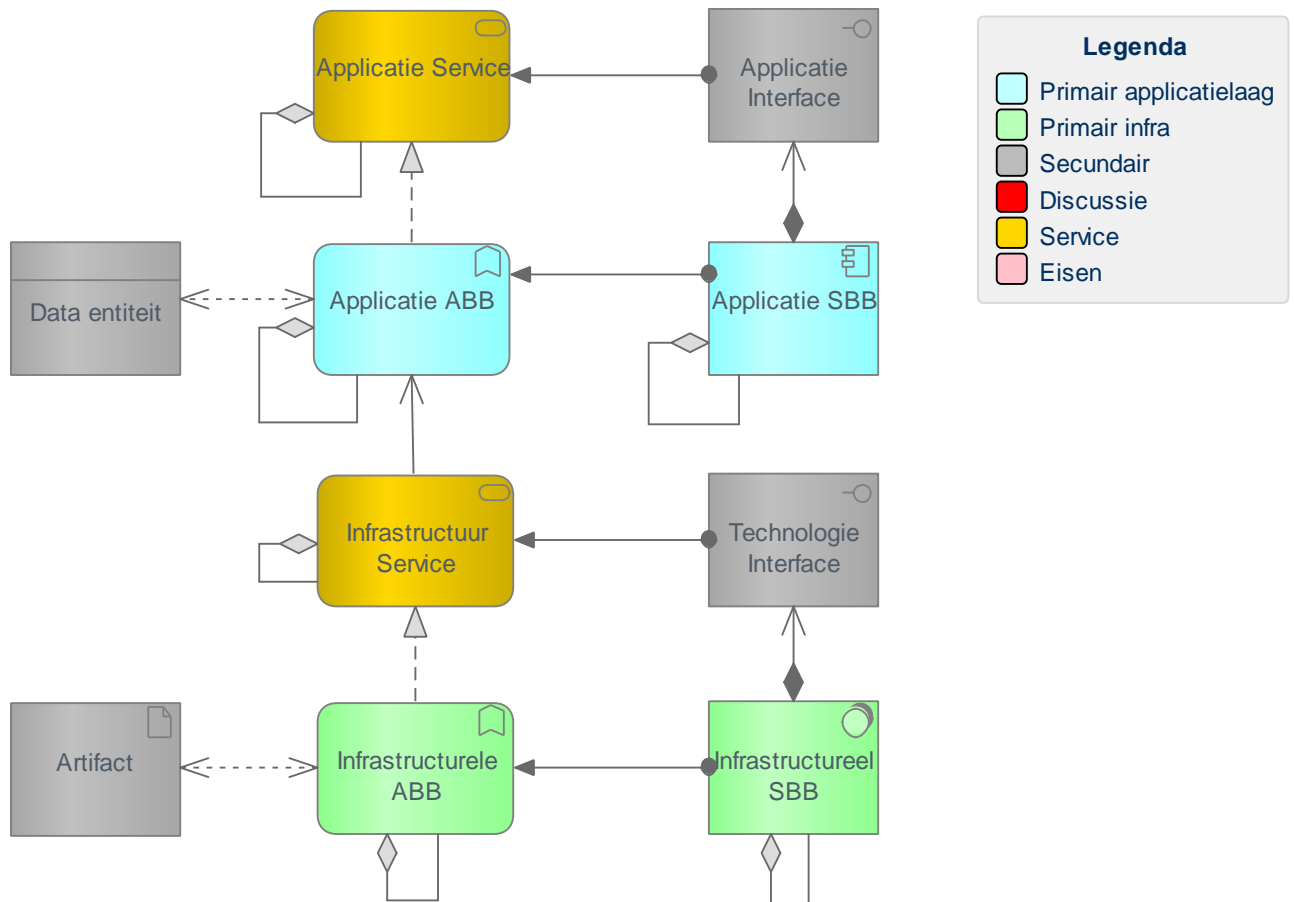
In het viewpoint worden een drietal ArchiMate concepten gebruikt uit de motivation extensie:

- Principe, veelal afkomstig van de landelijke overheidsreferentie architecturen
- Requirements, eisen en (non functionele) kwaliteitsaspecten
- Beperkingen, zijn vereisten welke meestal gelden voor de SBB, bijvoorbeeld gericht op bepaalde programmeerparadigma's of talen.

Voor de associaties tussen de motivation elements en de andere elementen kan gebruikt gemaakt worden van de associatie, influence en realisation

In dit model is alleen een uitwerking gemaakt van het bouwblokken viewpoint binnen de applicatielaag. Vanzelfsprekend geldt hetzelfde als hierboven beschreven voor de viewpoint op de technische laag.

## Viewpoint bouwblok meerlagig applicatie- en infrastructuurlaag



Dit is een discussie plaat voor de situatie waarbij een service op de applicatielaag is uitgewerkt in een aantal ABB en SBB op de applicatielaag. Vervolgens wordt het applicatie ABB ingevuld door de functionaliteit en implementatie en ABB en SBB binnen een infrastructurale service.

In deze afbeelding is op basis van de ArchiMate notatiewijze een voorbeeld uitgewerkt waarbij de infrastructurale bouwblokken via een service de applicatie ABB ondersteund. Hiermee wordt het model relatief omvangrijk maar wel gebaseerd op de ArchiMate viewpoint

## Artifact

### Naamgevingsconventie

Gebruik: Fysieke Naam van het bestand. Bijvoorbeeld: Order.jar, vraag.doc, klant.xls, Naam aangevuld met de suffix

## Voorbeelden

In deze voorbeelden worden een aantal aspecten van het modelleren op basis van bouwblokken uitgewerkt. Beschouw dit voorbeeld zonder dat er is nagedacht over de inrichting van de catalogi en de granulariteit van de bouwblokken.

Tevens is dit nog niet besproken in de werkgroep en daarmee een voorstel van mijn kant en is dit onderwerp van discussie voor de volgende bijeenkomst.

Daarnaast wordt alleen een toelichting gegeven bij de diagrammen, niet bij de daarin uitgewerkte concepten.

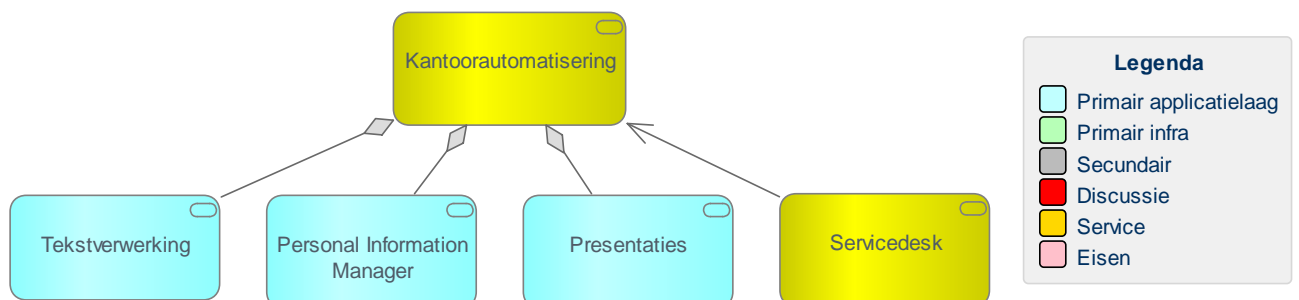
### Voorbeeld Service Basis



Dit eerste voorbeeld een uitwerking van de services. Kenmerkend hierin is dat de service een communicatiemiddel is tussen aanbieder en afnemer. Het is daarbij van belang dat er een begrijpelijk portfolio van services gemaakt wordt.

De portfolio aanpak is nog een onderwerp van discussie, wel opgenomen in het voorbeeld diagram. Discussiepunt is daarbij wat is de indeling van de portfolio en de services.

### Voorbeeld Service Samengesteld

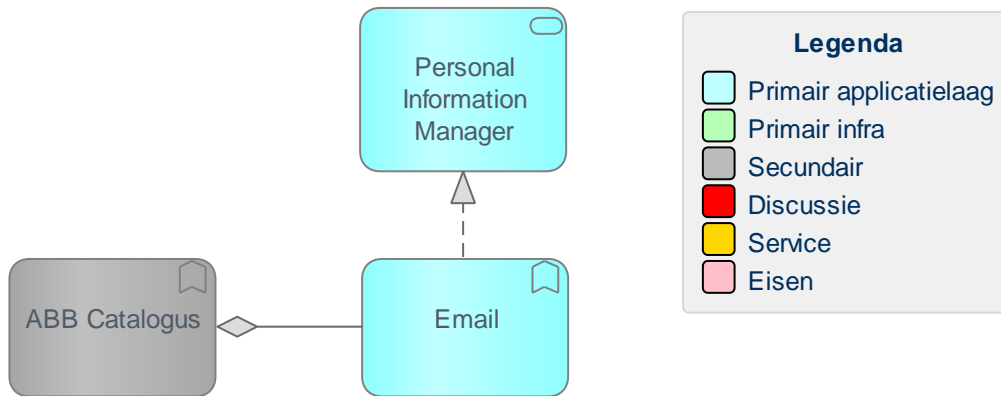


In het samengestelde servicemodel is te zien hoe een service op een hoger abstractieniveau is opgebouwd uit kleinere services met een meer specifiek karakter. In dit model kantoorautomatisering wordt een tussenlaag van services opgenomen, dat hoeft niet perse, je zou ook direct de koppeling kunnen leggen naar Microsoft Office dat is afhankelijk van de context.

In deze uitwerking worden alleen applicatie services gemodelleerd en in de samenstelling opgenomen. Echter naast applicatie services kun je hier ook bedrijfsservices definiëren. Denk hierbij aan de combinatie van de implementatie van Office en een servicedesk voor vragen bij problemen. Dat ligt nu buiten scope maar wordt op zeker moment relevant. Relevant hierin is dat

er dan een knelpunt in de ArchiMate modellering ontstaat. ICT business services inbedden in de lagere architectuurlagen.

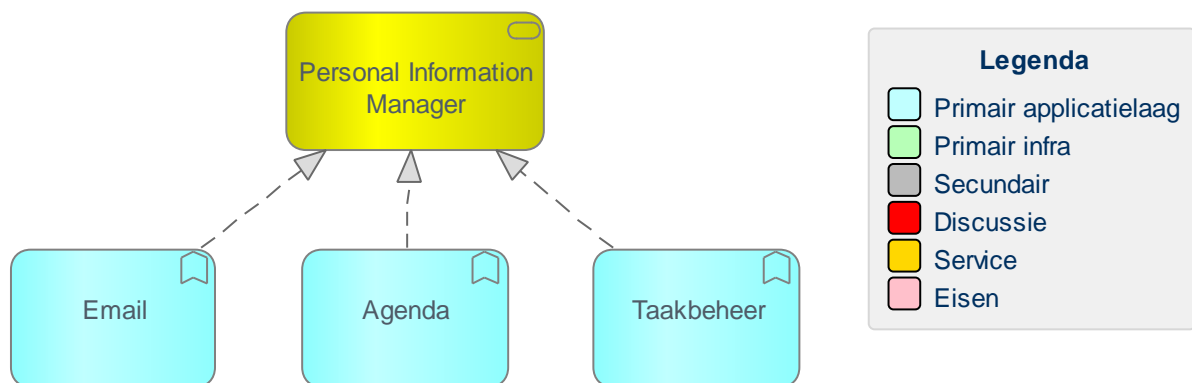
### Voorbeeld ABB Basis



Dit model is relatief eenvoudig van opzet, er is een applicatie service die wordt ingevuld door één logische applicatie functie. Echter dit kan in andere situaties een meer complexe samenstelling zijn.

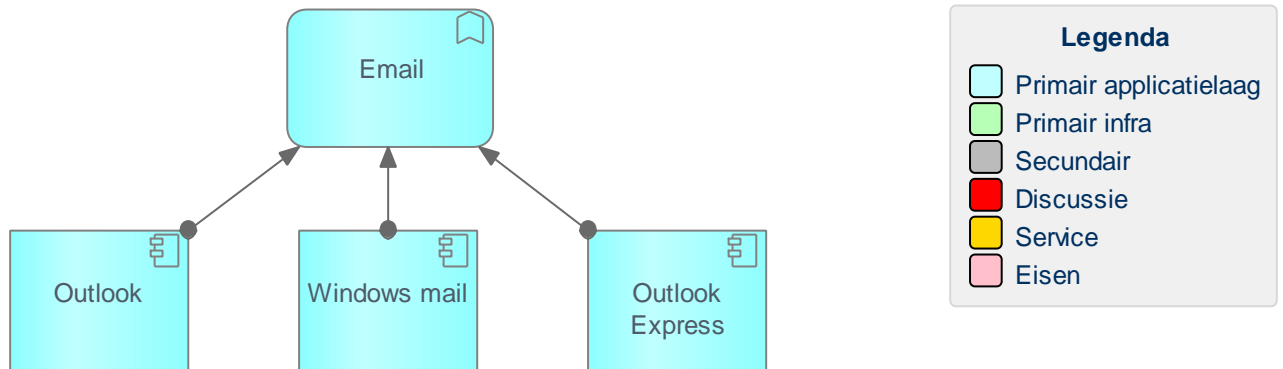
Wordt er voor de architectuur bouwblokken een register of portfolio opgesteld dan is dit een extra vorm van aggregatie. Alternatief is om via het service portfolio te aggregeren en groeperen. Is een discussiepunt.

### Voorbeeld ABB Samengesteld



Een service kan opgebouwd zijn uit meerdere ABB. In dit voorbeeld een service die ingevuld wordt door een aantal ABB. Ook hierbij is het punt van de samenstelling een punt van aandacht. Er kan een aggregatie toegevoegd worden van een samengestelde ABB die de andere ABB aggregereert en dit als samengestelde service aanbiedt. Is afhankelijk van de context van de bouwblokken, echter wel een punt om een werkinstructie van te maken. Desgewenst kan ook een join worden toegepast.

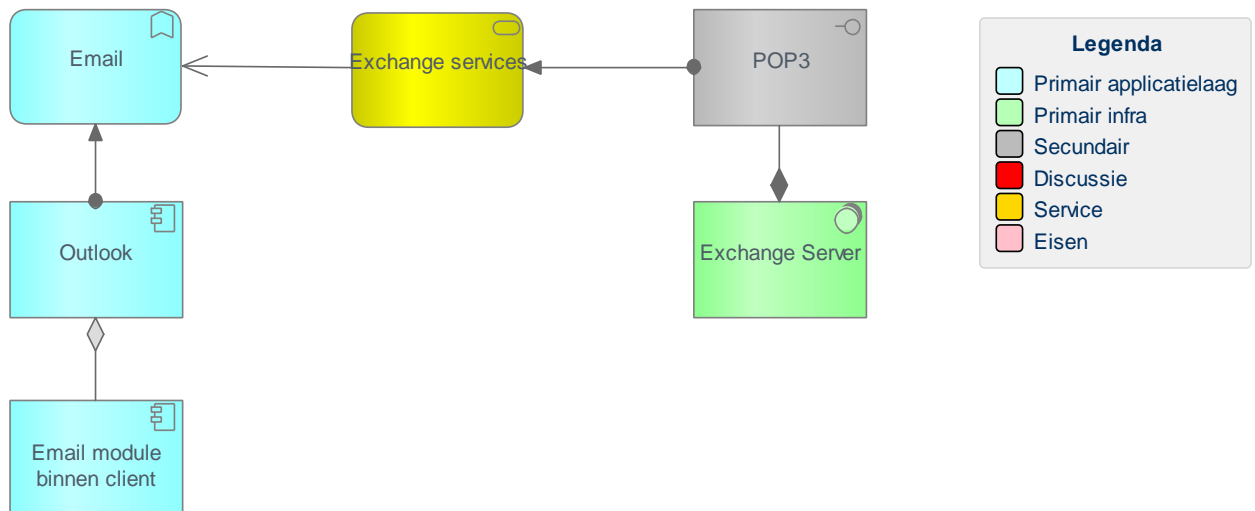
### Voorbeeld SBB Basis



Een ABB kan zijn opgebouwd uit een of meerdere SBB. Daarnaast is het mogelijk dat een ABB ingevuld kan worden door een van meerdere SBB, er ontstaat dus een keuzemogelijkheid. Dat wordt in dit voorbeeld getoond.

Hierbij wordt het van belang dat inzichtelijk gemaakt wordt wat de verschillen zijn tussen de verschillende SBB. Zie hiervoor het uitgebreide voorbeeld diagram.

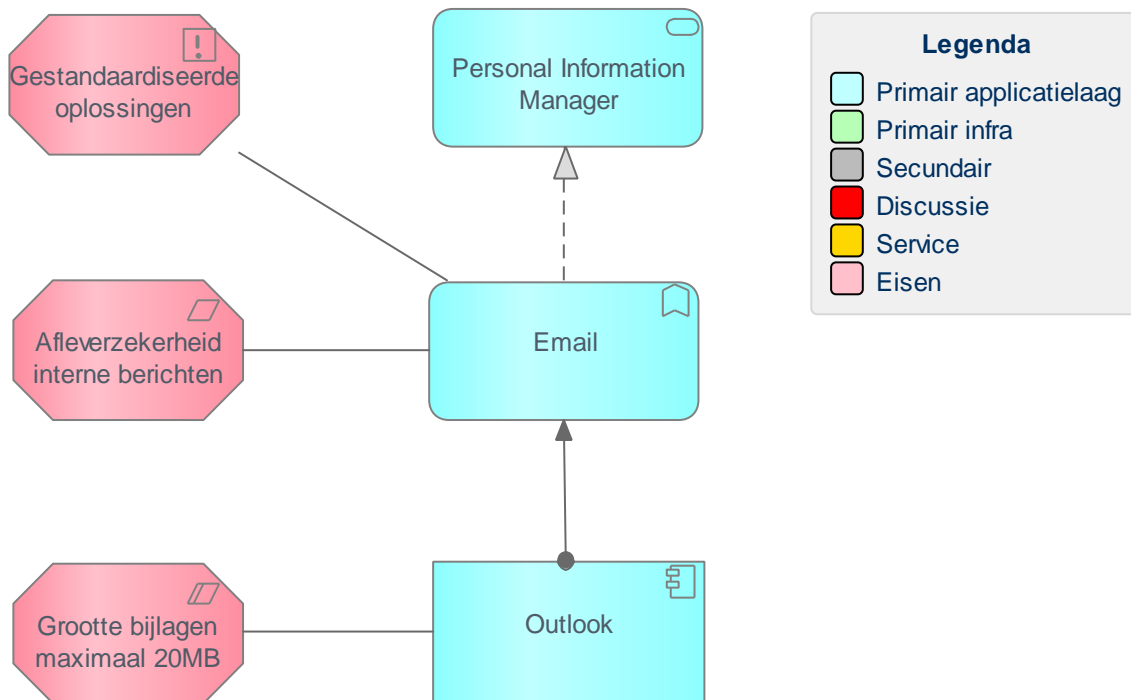
### Voorbeeld SBB Samengesteld



In dit samengestelde SBB model komen een aantal zaken samen:

- Het email ABB wordt enerzijds ingevuld door een Email component in de applicatielaag
- Ten tweede worden de email ABB ingevuld door een technische functie en - service binnen de infrastructurele ABB en SBB
- Het voorbeeld toont hoe een service de verbinding legt tussen de lagen.
- In het model is een secundair element opgenomen als voorbeeld een interface obv een mail protocol.

### Voorbeeld XBB Requirements en eisen



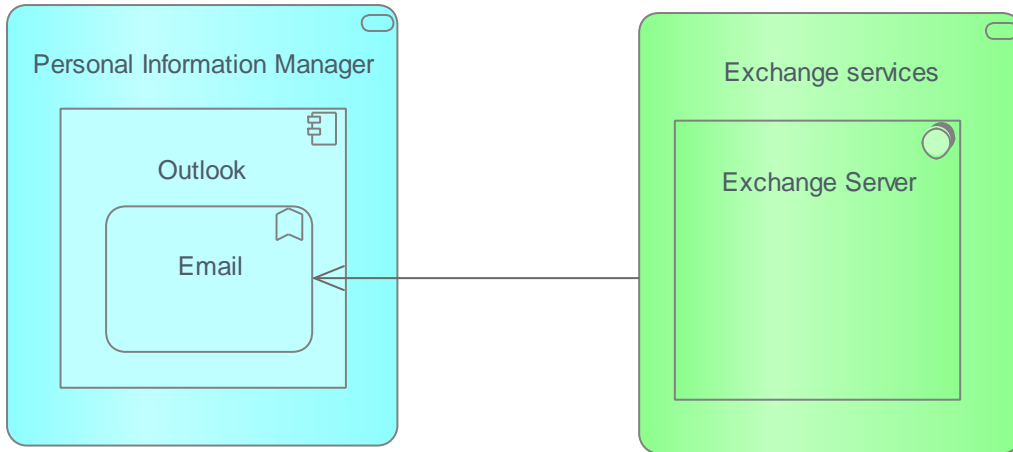
Dit voorbeeld laat op eenvoudige wijze zien hoe de kenmerken/eisen op basis van requirements, constraints, kwaliteiten en principes inzichtelijk gemaakt kunnen worden.

In dit voorbeeld is te zien hoe de kenmerken van de verschillende xBB op basis van ArchiMate concepten in kaart gebracht kunnen worden. Dit wordt straks een belangrijk mechanisme in de verschillende xBB catalogi.

Naast deze aanpak kunnen de kenmerken ook uitgewerkt worden via de interne kenmerken van de entiteiten in EA. Zoals requirements, constraint en scenario. Als laatste is er de mogelijkheid om tagged values te gebruiken. In de werkgroep is bepaald dat hierbij het leggen van associaties tussen ArchiMate concepten de voorkeur verdient. Is een uitwerking met ArchiMate concepten onvoldoende en wil men uitwijken naar interne eigenschappen of tagged values dan dient dit kortgesloten te worden



### **Voorbeeld Visualisatie Bouwblok**



Andere visualisatie van een viewpoint op basis van een nesting van elementen. De modelleur heeft hierbij de nodige vrijheden bij het gebruik van verschillende nestingen. Houdt rekening met het feit dat door het verbergen van associaties er onduidelijkheden kunnen ontstaan. Dus de associaties mogen onzichtbaar gemaakt worden maar mogen niet worden verwijderd uit het modell.

# Catalogi

- [Analysis patterns](#)
- [Arcitura Pattern catalogi](#)
- [Enterprise Application Integration](#)
- [Enterprise Architecture Patterns](#)
- [Infrastructuur patronen](#)
- [Interactiepatronen](#)
- [SOA patterns](#)
- [Software patronen](#)
- [Togaf Architecture Patterns](#)