

---

<Naam bedrijf opdrachtgever>

---

<Projectnaam>  
**Software Architectuur Document**

**Versie 0.1**

*[Om de automatische velden in Word bij te werken (die welke een grijze achtergrond tonen bij selecteren), moet u Bestand > Eigenschappen > Samenvatting selecteren en Titel, Onderwerp en Bedrijf vervangen door de juiste informatie. Doe hetzelfde met Eigenschappen > Aangepast; vervang versie en datum waarde door de juiste informatie. Sluit het dialoogvenster en update de automatische velden in het gehele document door Bewerken > Alles selecteren te kiezen (of Ctrl-A) en op F9 te drukken, of klik het veld aan en druk op F9. Dit moet nog een keer apart gebeuren voor de koptekst en de voettekst. Met Alt-F9 wisselt u tussen de weergave van de veldnamen en hun inhoud. Kies Help > Microsoft Word help voor meer informatie over het werken met velden.]*

<Projectnaam>	Versie: 0.1
Software Architectuur Document	Datum: 1-1-2011

## Documenthistorie

Datum	Versie	Beschrijving	Auteur
	0.1	Initiële versie	

## Distributie

Naam	0.1													

## Accordering document

Namens <Naam bedrijf opdrachtgever>

*[Vul hier de naam van de ondertekenaar in]*

.....



<Projectnaam>	Versie: 0.1
Software Architectuur Document	Datum: 1-1-2011

## Inhoudsopgave

1.Inleiding	5
1.1.Doel van dit document	5
1.2.Referenties	5
1.3.Documentoverzicht	5
2.Architecturele eisen	7
2.1.Niet-functionele requirements	7
2.2.Use Case View (functionele requirements)	7
3.Logical View	8
3.1.Lagen	8
3.2.Deelsystemen	8
3.3.Use Case Realizations	8
4.Implementation View	8
4.1.Packagestructuur	8
4.2.Invulling lagenstructuur	8
4.3.(Her)gebruik van componenten en frameworks	9
5.Deployment View	9

<Projectnaam>	Versie: 0.1
Software Architectuur Document	Datum: 1-1-2011

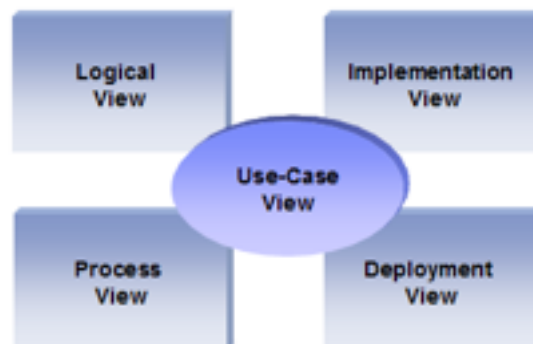
## 1. Inleiding

### 1.1. Doel van dit document

*[Geef hier het doel van dit Software Architectuur Document aan. Meestal formuleer je zo iets als de onderstaande tekst:]*

Het Software Architectuur Document (SAD) bevat een uitgebreide architecturale kijk op het systeem <Projectnaam> ontwikkeld door <Naam bedrijf opdrachtgever>. Het beschrijft een aantal verschillende architecturale views van het systeem om zo verschillende aspecten van het systeem te belichten.

Dit document beschrijft de verschillende RUP views op de software architectuur volgens het 4+1 view model.



Het 4+1 view model stelt de verschillende belanghebbenden in staat vanuit hun eigen perspectief de invloed van de gekozen architectuur te bepalen. De Process View (communicatie van processen) is niet als los hoofdstuk uitgewerkt maar ondergebracht bij de hoofdstukken 3.3 en 5.

### 1.2. Referenties

*[Geef hier een volledige lijst van alle documenten op basis waarvan dit Software Architectuur Document tot stand is gekomen. Vermeld eventueel waar het document te vinden is.]*

Titel	Versie	Auteur	Vindplaats
<i>[Geef hier de titel of documentnaam van de referentie]</i>			<i>[Geef de vindplaats, indien mogelijk ook als hyperlink]</i>

### 1.3. Documentoverzicht

*[Maak een leeswijzer waarin je beschrijf wat de rest van het Software Architectuur Document bevat en hoe het is georganiseerd.]*

Hoofdstuk	Belanghebbende	Doel
2 Architecturale eisen	Software Architect	Overzicht van architectureel relevante requirements.
3 Logical View	Programmeur (t.b.v. technisch ontwerp)	Inzicht in de conceptuele structuur van de applicatie.
4 Implementation View	Programmeur (t.b.v. bouw)	Inzicht in de technische structuur van de applicatie.

Vertrouwelijk

©<Naam bedrijf opdrachtgever>

Pagina 5 van 9

2011

<Projectnaam>		Versie:	0.1
Software Architectuur Document		Datum:	1-1-2011
Hoofdstuk	Belanghebbende	Doel	
5 Deployment View	Beheer rollen	Inzicht in de manier waarop de applicatie wordt gedeployed en de manier waarop de (interne en externe) communicatie plaatsvindt.	

<Projectnaam>	Versie: 0.1
Software Architectuur Document	Datum: 1-1-2011

## 2. Architecturele eisen

*[Deze sectie beschrijft de software eisen die voor het ontwikkelen van de software architectuur van belang zijn.]*

### 2.1. Niet-functionele requirements

*[Beschrijf hier de architectureel relevante niet-functionele requirements die voor het ontwikkelen van de software architectuur van belang zijn. Denk hierbij aan beveiliging, privacy, gebruik van producten van derden, systeem(on)afhankelijkheid, distributie en hergebruik. Ook omgevingsfactoren zoals context, design en implementatie strategie, ontwikkeltools, teamsamenstelling, doorlooptijd, gebruik van legacy code kunnen hier worden meegenomen. Geef per requirement ook de bron aan en de plaats in dit document waar hij geadresseerd wordt.]*

Bron	Naam	Architecturele relevantie	Geadresseerd in:
<i>[De bron is de Vision, het Acceptatie Plan of een belanghebbende.]</i>	<i>[Geef hier de naam van het requirement]</i>	<i>[Beschrijf hier wat het genoemde requirement relevant maakt bij het opstellen van de software architectuur.]</i>	<i>[Geef hier een paragraafnummer waar het requirement wordt behandeld.]</i>

### 2.2. Use Case View (functionele requirements)

*[Verwijs hier naar Use Cases of Use Case scenario's uit het Use Case Model die voor het ontwikkelen van de software architectuur van belang zijn. De opgenomen Use Cases moeten centrale functionaliteit in het uiteindelijke systeem herbergen, veel architecturele elementen bevatten of een specifiek delicaat punt van de architectuur benadrukken.]*

Onderstaand overzicht verwijst naar architectureel relevante Use Cases uit het Use Case Model (zie referenties).

Bron	Naam	Architecturele relevantie	Geadresseerd in:
<i>[De bron is een UC of scenario.]</i>	<i>[Geef hier de naam van een Use Case of Use Case scenario]</i>	<i>[Beschrijf hier wat de genoemde Use Case of het genoemde scenario relevant maakt bij het opstellen van de software architectuur.]</i>	<i>[Geef hier een paragraafnummer waar de UC of scenario wordt behandeld.]</i>
	UC01: Aanmelden		

<Projectnaam>	Versie: 0.1
Software Architectuur Document	Datum: 1-1-2011

### 3.Logical View

*[Beschrijf hier de architectureel significante logische opbouw van het systeem. Denk hierbij aan de decompositie in lagen en deelsystemen. Beschrijf ook de manier waarop Use Cases, rekening houdend met deze logische decompositie, technisch worden vertaald naar Use Case Realizations.]*

#### 3.1.Lagen

*[Beschrijf hier de onderkende lagen (layers) en hun verantwoordelijkheid binnen het systeem. Ga hierbij bijvoorbeeld uit van het 4-lagen-model:*

- *Presentatie*
- *Service*
- *Domein*
- *Data*

*Geef van iedere laag duidelijk aan wat zijn verantwoordelijkheid binnen het systeem is en hoe op logisch niveau de communicatie met andere lagen plaats zal vinden. (De technische invulling van de lagen wordt beschreven in de Implementation View)]*

#### 3.2.Deelsystemen

*[Beschrijf hier de decompositie van het systeem in deelsystemen en laat de samenhang hiertussen zien.]*

#### 3.3.Use Case Realizations

*[Illustreer hier de wijze waarop de eerder genoemde Use Case Specifications technisch vertaald worden naar Use Case Realizations. Werk voor één of enkele in de Use Case View vermelde Use Cases (of scenario's) de technische vertaling uit (bijvoorbeeld in een sequentie-diagram). Besteed hierbij aandacht aan de communicatie door de onderkende lagen in de applicatie en verklaar hoe de gebruikte componenten/objecten bijdragen aan de functionaliteit.]*

### 4.Implementation View

*[Deze sectie beschrijft de technische invulling van de logical view.]*

#### 4.1.Packagestructuur

*[Geef hier een overzicht van de packagestructuur in de ontwikkelomgeving. Probeer bij de package structuur de logische lagen scheiding terug te laten komen in de naamgeving van de packages, zoals in het voorbeeld hieronder:*

- *nl.bedrijf.afdeling.project.subsysteem.ui*
- *nl.bedrijf.afdeling.project.subsysteem.service*
- *nl.bedrijf.afdeling.project.subsysteem.domain*
- *nl.bedrijf.afdeling.project.subsysteem.data*

*Illustreer de samenhang van de package structuur dmv een package-diagram.]*

#### 4.2.Invulling lagenstructuur

*[Beschrijf hier de technische invulling van de in de Logical View onderscheiden lagen. Benoem ook de regels voor een component om in een bepaalde laag opgenomen te worden.]*

vertrouwelijk

©<Naam bedrijf opdrachtgever>,

Pagina 8 van 9

2011



<Projectnaam>	Versie: 0.1
Software Architectuur Document	Datum: 1-1-2011

#### 4.3.(Her)gebruik van componenten en frameworks

*[Beschrijf hier de bij de bouw te (her)gebruiken componenten en frameworks (intern en van derden). Dit voor zover ze niet bij de invulling van de lagenstructuur zijn behandeld. Indien er bij de eisen bepaalde frameworks zijn genoemd, dienen deze hier terug te komen.]*

### 5.Deployment View

*[Beschrijf hier de fysieke netwerk(hardware) configuraties waarop de software gaat draaien. Beschrijf minimaal de configuraties van de verschillende fysieke nodes (computers, CPUs), de interactie tussen (deel)systemen en de connecties tussen deze nodes (bus, LAN, point-to-point, messaging, SOAP, http, https). Maak gebruik van een deployment-diagram.]*

Naam	Type	Omschrijving
<i>[Naam node of deployment-werkproduct]</i>	<i>[Node / werkproduct]</i>	<i>[Omschrijving met technische gegevens]</i>