



Logius
*Ministerie van Binnenlandse Zaken en
Koninkrijksrelaties*

Software Architectuur Document Digimelding Portaal

Versie 1.3

Datum 1 april 2010
Status Definitief

Colofon

Projectnaam	Digimelding
Versienummer	1.3
Organisatie	Logius Postbus 84011 2508 AA Den Haag servicecentrum@logius.nl
Bijlage(n)	0

Inhoud

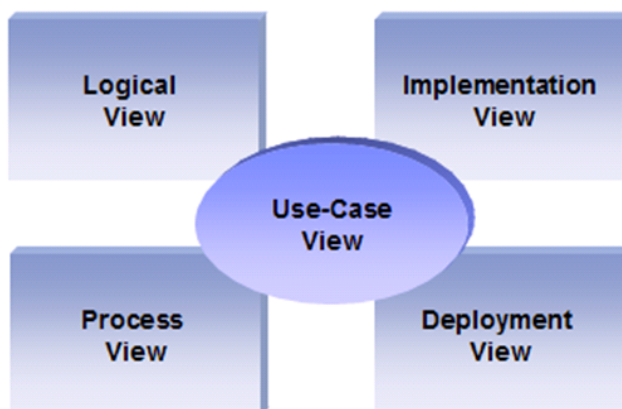
Colofon	2
Inhoud	3
Inleiding	5
1.1 <i>Doel van dit document</i>	5
1.2 <i>Referenties</i>	5
2 Architecturale eisen	7
2.1 <i>Niet-functionele requirements</i>	7
2.2 <i>Functionele requirements</i>	8
3 Architectuur doelen en voorwaarden	9
3.1 <i>Ontwerpbesluiten</i>	9
3.2 <i>Modulariteit / uitbreidbaarheid</i>	9
3.2.1 <i>Interfaces</i>	9
3.2.2 <i>Intern en extern datamodel</i>	9
3.2.3 <i>Dependency injection</i>	10
3.3 <i>Gebruikte open-source producten en frameworks</i>	12
4 Use-Cases	13
4.1 <i>Actoren</i>	13
4.2 <i>Use Cases</i>	13
4.2.1 <i>UC041S Meld gerede twijfel</i>	14
4.2.2 <i>UC044S Trek terugmelding in</i>	14
4.2.3 <i>UC045S Raadpleeg terugmelding</i>	14
4.2.4 <i>UC046S Raadpleeg terugmeldingen</i>	14
4.2.5 <i>Use Case realization – UC041S Meld gerede twijfel</i>	14
5 Logical view	15
5.1 <i>Overzicht</i>	15
5.2 <i>Model-View-Controller (MVC)</i>	16
5.3 <i>Klassediagrammen</i>	18
5.3.1 <i>Algemene opzet</i>	18
5.3.2 <i>Stelselcatalogus</i>	19
5.3.3 <i>DigimeldingCore</i>	20
6 Process view	21
7 Implementation view	22

7.1	<i>Digimelding Portaal</i>	22
7.1.1	Model (interne deel).....	22
7.1.2	Model (externe deel).....	23
7.1.3	View.....	24
7.1.4	Controller.....	24
7.2	<i>AntwoordService</i>	25
8	Deployment view	26
8.1	<i>Overzicht</i>	26
8.2	<i>Installatie</i>	27
8.3	<i>Beheer</i>	27
9	Kwaliteit	29
9.1	<i>Opbouw applicatie</i>	29
9.2	<i>Testplan en Acceptatieplan</i>	29
9.3	<i>Dekkingsgraad, Code checkstyle en Complexiteitsrapportage</i>	29
9.4	<i>Javadoc</i>	30

Inleiding

1.1 Doel van dit document

Dit Software Architectuur Document (SAD) beschrijft de architectuur van het Digimelding Portaal. Het SAD is opgebouwd aan de hand van het zogenaamde 4+1 view model [1].



Het 4+1 model bestaat uit vier separate views op het systeem, elk vanuit hun eigen perspectief. Zo is de implementation view bijvoorbeeld interessant voor de programmeur, terwijl de deployment view relevant is voor de beheerder. De Use-Case view is overkoepelend in de zin dat de samenkomst van de elementen uit de verschillende views de uitvoer van de use cases mogelijk maakt. Tezamen bieden al deze views aan alle betrokken partijen een goed overzicht van het ontwerp, de realisatie en de uitrol van de applicatie.

In het volgende hoofdstuk worden de architectuur doelen en voorwaarden besproken. De daarop volgende hoofdstukken gaan over bovengenoemde views uit het 4+1 model. Tenslotte volgt een hoofdstuk met betrekking tot de kwaliteit van de Digimelding Portaal applicatie.

1.2 Referenties

De volgende referenties worden gebruikt in dit document:

1. Philippe Kruchten, Architectural Blueprints—The “4+1” ViewModel of Software Architecture, 1995, <http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>
2. ICTU, Aanvraag ICTU - 800946 Webportal Terugmeldfaciliteit, 23-10-08
3. OVSoftware, 800496 Offerte aanvraag realisatie Digimelding Portaal, 02-10-08
4. Peter Schipperheijn, UC041S Meld gerede twijfel, 18-09-08
5. Peter Schipperheijn, UC044S Trek terugmelding in, 18-09-08
6. Peter Schipperheijn, UC045S Raadpleeg terugmelding, 18-09-08
7. Peter Schipperheijn, UC046S Raadpleeg terugmeldingen, 18-09-08
8. OVSoftware, Offerte ICTU - 800946 Webportal Terugmeldfaciliteit, 18-09-08
9. OVSoftware, UCM Digimelding Portal, 18-09-08

10. OVSoftware, Digimelding Portaal - Testplan, December 2008
11. Wikipedia, Model-View-Controller (MVC) pattern, 15-12-08,
12. OVSoftware, Use Case Realization - UC041S Meld gereede twijfel, December 2008
13. OVSoftware, Digimelding Portaal - Installatiehandleiding, December 2008
14. OVSoftware, Digimelding Portaal - Beheerhandleiding, December 2008
15. OVSoftware, Digimelding Portaal - Acceptatieplan, December 2008
16. OVSoftware, Digimelding Portaal - Kwaliteitsrapportage, December 2008
17. Wikipedia, Code Complexity (McCabe), 15-12-08,
http://en.wikipedia.org/wiki/Cyclomatic_complexity

2 Architecturele eisen

Onder architecturele eisen worden hier de functionele en niet-functionele requirements verstaan. De requirements zijn gebaseerd op zowel de aanvraag [2] als de afgegeven offerte [3]. Daarnaast zijn sommige eisen geëxtraheerd uit de aangeleverde use cases [4][5][6][7]. Overigens zijn de use cases ook de basis voor alle functionele requirements.

2.1 Niet-functionele requirements

De niet-functionele requirements vormen een wezenlijk onderdeel van de totale requirementsverzameling. Het zijn requirements die niet direct betrekking hebben op een benodigde functie maar die gelden voor de applicatie in zijn geheel. In onderstaande tabel zijn alle niet-functionele requirements voor het Digimelding Portaal opgenomen. Sommige niet-functionele requirements hebben betrekking op de gebruikte ontwikkelstraat bij OVSoftware.

Bron	Naam	Beschrijving	Uitwerking
[3][8]	[N1] Platform onafhankelijk	Het Digimelding Portaal wordt geïmplementeerd op basis van Java technologieën.	(zie code)
OVSoftware	[NF2] Eenvoudige installatie	Ontwerpbesluiten zullen een eenvoudige installatie zwaar laten meewegen.	8.2
OVSoftware	[NF3] Opensource / licentievrij	Alle benodigde software van derden dient opensource/licentievrij te zijn.	3.2
[8]	[NF4] Webbased	De applicatie zal een webbased GUI hebben.	3.1.4
[8]	[NF5] Browser-onafhankelijk	De applicatie zal in alle browsers werken.	3.1.4
[8]	[NF6] Servlet 2.4 en JSP 2.0	De applicatie wordt opgeleverd inclusief Tomcat 5.5 Servlet-container.	8.2
[3][8]	[NF7] Geen authenticatie eindgebruikers	Authenticatie van gebruikers vindt niet plaats door het Digimelding Portaal. De afnemersorganisatie is verantwoordelijk voor de toegang tot het Digimelding Portaal. Het Digimelding Portaal mag niet van buitenaf beschikbaar zijn.	3.1.3
[9]	[NF8] Geen opslag berichten en antwoorden	Er worden geen gegevens lokaal opgeslagen. Alle opslag van gegevens vindt plaats bij de Stelselcatalogus en DigimeldingCore.	7.1.2
[8]	[NF9] Uitbreiding op Digikoppeling Gateway	Alle communicatie met de Stelselcatalogus en DigimeldingCore vindt plaats via de Digikoppeling Gateway.	5.1

OVSoftware	[NF10] Maven	De projectinrichting vindt plaats op basis van Maven. Daarbij worden ook diverse plugins gebruikt voor kwaliteitsrapportages.	9
OVSoftware	[NF11] Spring framework	Het Spring-framework wordt gebruikt voor onder andere de mogelijkheid van dependency injection.	3.2
[8]	[NF12] Koppelvlak	[NF9] creëert een koppelvlak op basis van WUSLite. Antwoordberichten (ebMS) zullen worden opgevangen door een Antwoordservice.	7.2
UC041S	[NF13] Interne Stelselcatalogus	In de oude versie van Digimelding Portaal kende het systeem een interne opslag van basisregistraties, objecttypen en mogelijke statussen. Sinds Digimelding 1.1 is dit vervangen door operaties op de Stelselcatalogus web-service.	5.3.2 + 8.3
UC041S	[NF14] Statusloosheid	Uitgaande berichten worden niet vastgelegd, behalve eventueel in de logbestanden. Een Antwoordservice zal inkomende berichten opvangen en loggen, zie ook [NF12].	7.2 + 8.3

2.2 Functionele requirements

Bij functionele requirements gaat het om specifieke eisen aan de functionaliteit. De basis voor alle functionaliteit van het Digimelding Portaal zijn de aangeleverde use cases. De functionele requirements die voortvloeien uit deze use cases zijn hier niet nogmaals uitgewerkt, zie hiervoor de use case documenten [4][5][6][7] en hoofdstuk 4. Voor het testen van alle functionaliteit kan men in het testplan [10] scenario's vinden waarmee de gehele applicatie stap-voor-stap kan worden doorlopen.

3 Architectuur doelen en voorwaarden

3.1 **Ontwerpbesluiten**

Bij het ontwikkelen van het Digimelding Portaal zijn op basis van de opdracht en het toekomstig gebruik enkele belangrijke ontwerpbesluiten genomen. Deze keuzes hebben invloed op zowel de implementatie als het toekomstig gebruik. De ontwerpbesluiten zijn onder te verdelen in vier groepen:

1. Modulariteit / uitbreidbaarheid
2. Ondersteunbaarheid
3. Beveiliging
4. Gebruikersinterface

3.2 **Modulariteit / uitbreidbaarheid**

De modulariteit en de uitbreidbaarheid van het systeem hebben betrekking op de mogelijkheid om het systeem later te verbeteren of aan te passen, zonder dat grootschalige aanpassingen noodzakelijk zijn.

De mate van integratie tussen de verschillende objecten in het Digimelding Portaal is zo laag mogelijk gehouden. Objecten hebben een duidelijk doel en domein waarin ze gebruikt worden. Communicatie met andere delen van het systeem vindt zoveel mogelijk plaats via interfaces.

3.2.1 *Interfaces*

Een interface is een contract tussen een object en de gebruiker van een object. Indien een object een bepaalde interface implementeert dan zal het minimaal alle functies van de interface moeten implementeren. Door gebruik te maken van interfaces is het mogelijk om de achterliggende klasse te vervangen, zolang de vervangende klasse maar de interface implementeert.

In het Digimelding Portaal wordt uitgebreid gebruik gemaakt van interfaces zodat de verschillende lagen van elkaar gescheiden worden. Een voorbeeld hiervan zijn de verschillende Invoker-interfaces, welke vanuit de View (als in MVC [11], zie 5.2) toegang geven tot bijvoorbeeld de lijst met gedane terugmeldingen. De Invoker kan dan geïmplementeerd zijn door een Mock-klasse of door de echte webservice. Vanuit de View is het echter niet van belang welke klasse daar achter hangt, zolang deze maar de interface implementeert.

3.2.2 *Intern en extern datamodel*

De scheiding tussen de verschillende delen van de applicatie beperkt zich niet alleen tot de klassen. Ook het gebruik van een intern en extern datamodel zorgt voor een goede scheiding. Binnen de applicatie (de View en een deel van de Controller) wordt gebruik gemaakt van het interne datamodel, dat in twee delen uiteenvalt:

1. Interne Stelselcatalogus-gerelateerde datamodel
2. Interne DigimeldingCore-gerelateerde datamodel

De View spreekt dus een interface aan met zijn eigen datamodel zonder te weten welke specifieke implementatie achter de interface ligt.

Het interne datamodel wordt vertaald naar gelijkwaardige data-objecten in het externe datamodel. Ook het externe datamodel bestaat op zijn beurt uit meerdere delen, te weten:

1. Externe Stelselcatalogus-gerelateerde datamodel
2. Externe DigimeldingCore-gerelateerde datamodel met betrekking tot WUS-webservices
3. Externe DigimeldingCore-gerelateerde datamodel met betrekking tot EBMS-webservices

Door de vertaling wordt intern alleen gebruik gemaakt van het interne datamodel en extern, in de communicatie met de Digikoppeling Gateway, van het externe datamodel.

3.2.3 *Dependency injection*

Het gebruik van interfaces brengt met zich mee dat op een gegeven punt aan moet worden gegeven welke concrete implementatie van zo'n interface gebruikt moet worden. Een oplossing voor dit probleem is het gebruik van dependency injection. Bij deze oplossing worden de afhankelijkheden tussen de verschillende objecten pas achteraf bepaald door het instellen van een configuratiebestand. De afhankelijkheden zijn dus declaratief gedefinieerd in plaats van imperatief (dat wil zeggen in de code).

Het Digimelding Portaal gebruikt het Spring framework voor dependency injection. De configuratie vindt plaats door middel van een set XML-bestanden die de koppelingen beschrijven. Hierdoor is het vervangen van een of meerdere onderdelen erg eenvoudig, zolang de benodigde interface maar worden geïmplementeerd.

3.2.3.1 Onderhoudbaarheid en ondersteunbaarheid.

In het 4+1 view model[1] wordt duidelijk dat een systeem altijd vanuit meerdere perspectieven (views) kan worden bekeken. Vanuit het perspectief van de beheerder zijn met name de onderhoudbaarheid en de ondersteunbaarheid van groot belang. Met onderhoudbaarheid wordt hier nadrukkelijk het onderhoud van een draaiend systeem bedoeld bij de afnemer en dus niet het onderhoud van de code.

Om het onderhoud zo eenvoudig mogelijk te houden, wordt zoveel mogelijk gewerkt met standaarden en vrijelijk beschikbare open-source software. Zo wordt bijvoorbeeld gebruik gemaakt van Apache's Log4j, een open-source logging framework voor Java. Hiermee is het mogelijk voor de beheerder om het niveau waarop wordt gelogd eenvoudig aan te passen. Het lokaliseren van problemen wordt hierdoor versneld. Daarnaast is het Digimelding Portaal geïmplementeerd als een webapplicatie. De centralisering van de applicatie vereenvoudigt het onderhoud.

Verder is de ondersteunbaarheid van belang. Met ondersteunbaarheid wordt de mate van ondersteuning die beschikbaar dient te zijn binnen een afnemersorganisatie naar de eindgebruikers toe bedoeld. De interface van het Digimelding Portaal is zo gebruiksvriendelijk mogelijk opgebouwd, zie ook sectie 3.1.4. Daarnaast beschikt de applicatie over uitgebreide logfunctionaliteit, dit is verder toegelicht in sectie 8.3.

3.2.3.2 Beveiliging

De beveiliging is een belangrijk punt bij het Digimelding Portaal. De applicatie wordt immers gebruikt om (doorgaans) vertrouwelijke gegevens

te communiceren. De implementatie van het Digimelding Portaal is gebaseerd op het gebruik van de Digikoppeling Gateway. De Digikoppeling Gateway is een proxy-achtige service welke een vertaling tussen de protocollen WUS en EBMS aan de ene kant en anderzijds WUSlite en JMS mogelijk maakt. Daarnaast verzorgt de Gateway door middel van SSL-encryptie de communicatie met de Stelselcatalogus en DigimeldingCore.

Het Digimelding Portaal heeft geen persistente opslag in de zin van een database of andere lokale databestanden. De enige uitzonderingen hierop zijn de instellingsbestanden en de logging. Het is mogelijk om te loggen op een dusdanig niveau dat alle communicatie tussen het Digimelding Portaal en de Digikoppeling Gateway wordt opgeslagen. Geadviseerd wordt echter om de logging in een productie-omgeving op een lager niveau in te stellen (bijvoorbeeld waarschuwing-niveau). De verantwoordelijkheid voor het beveiligen van de toegang tot de server met daarop de log-bestanden alsmede de Digikoppeling Gateway zijn de verantwoordelijkheid van de afnemersorganisatie.

3.2.3.3

Gebruikersinterface

Bij het ontwerp van de gebruikersinterface zijn zowel de gebruiksvriendelijkheid als de wijze van implementatie van belang. Het Digimelding Portaal is geïmplementeerd als een gebruiksvriendelijke webapplicatie. Bij het ontwerp is ervoor gekozen om een wizard-achtige interface te implementeren waarin de eindgebruiker voor de meeste voorkomende taak (het doen van een terugmelding) een simpel stappenplan kan doorlopen. De gebruiker kan op elk moment een stap terug gaan of het doen van een terugmelding beëindigen. Door de eenvoudige interface en de centralisatie van de applicatie wordt de ondersteuning van eindgebruikers versimpeld.

De implementatie is gebaseerd op Java Server Pages (JSP) en Servlet-technologie. Daarbij voldoen alle pagina's van de applicatie aan de HTML en CSS standaarden conform het W3C. Ook is de webapplicatie uitvoerig getest in de meest populaire browsers, te weten Firefox v3 en Internet Explorer v7. Het Digimelding Portaal maakt vrijwel geen gebruik van Javascript. Alle functionaliteit is ook beschikbaar voor gebruikers die geen Javascript geactiveerd hebben. Het voor de implementatie gebruikte pattern (MVC [11]) wordt besproken in sectie 5.2.

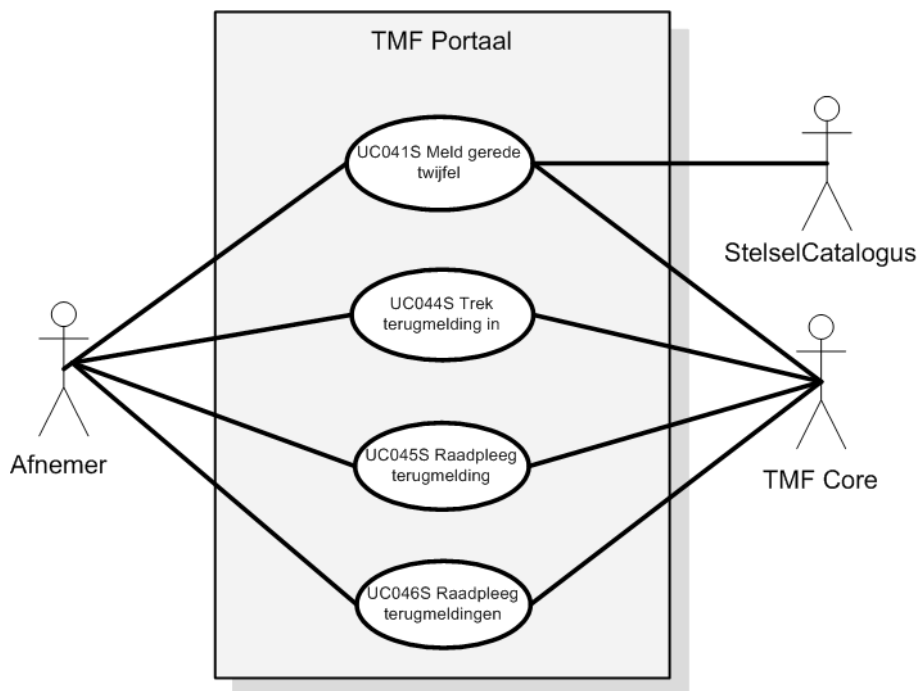
3.3 Gebruikte open-source producten en frameworks

In de volgende tabel wordt een kort overzicht gegeven van de belangrijkste open-source producten en frameworks die gebruikt worden in het Digimelding Portaal.

Product	Versie	Beschrijving en website
Spring framework	2.5	Veelgebruikt open-source application framework voor Java. Wordt hier gebruikt oa voor dependency injection. http://www.springframework.org/
Apache Log4j	1.2.15	Open-source logging framework voor Java. http://logging.apache.org/log4j/
JAX-WS + JAX-B	1.6	Open-source webservice framework voor Java. De hier gebruikte v2.1 is onderdeel van Java v1.6 update 11. http://java.sun.com/javase/6/docs/api/
Apache Tiles	2.0	Open-source framework voor het hergebruik van delen van een JSP-pagina. http://tiles.apache.org/
JUnit	3.8.1	Open-source test-framework voor unit-testen. http://www.junit.org/

4 Use-Cases

Het volgende diagram geeft een overzicht van de use cases en de actoren op hoog niveau. In de daadwerkelijke uitvoering wordt gebruik gemaakt van de Digikoppeling Gateway, zie ook 4.3. De getoonde use cases komen overeen met de lijst van te implementeren use cases [4][5][6][7] zoals aangeleverd door de opdrachtgever.



4.1 Actoren

Er zijn drie actoren te onderscheiden in het use case model, te weten:

1. Afnemer, dit is (een medewerker van) de afnemersorganisatie die gebruik maakt van een lokale installatie van het Digimelding Portaal.
2. Stelselcatalogus, dit is de (tijdelijke) catalogus van alle basisregistraties. Bij het opstellen van een terugmelding wordt bij deze dienst de lijst met basisregistraties, de lijst met objecttypen voor een bepaalde basisregistratie en de lijst met attributen voor een bepaald objecttype opgevraagd.
3. Digimelding Core, dit is de webservice waar nieuwe meldingen ingevoerd kunnen worden, gedane terugmeldingen kunnen worden opgevraagd, details van een terugmelding kunnen worden opgevraagd en een melding kan worden ingetrokken.

NB: In de daadwerkelijke uitwerking wordt er gebruik gemaakt van de Digikoppeling Gateway, waardoor er dus effectief een vierde actor is. In de ontwerp use cases is deze echter nog niet meegenomen. In de use case realization (zie 4.3) komt deze actor wel expliciet naar voren.

4.2 Use Cases

Voor de aanvang van het project waren de use cases [4][5][6][7] reeds uitgewerkt. Er hebben geen grote veranderingen plaatsgevonden in het

ontwerp waardoor hier wordt volstaan met een korte opsomming, met daarbij een verwijzing naar het relevante document.

4.2.1 *UC041S Meld gerede twijfel*

Bij het melden van gerede twijfel stelt een medewerker van een afnemersorganisatie een terugmelding op en verstuurt deze. Tijdens het opstellen wordt de lijst met basisregistraties, de lijst met objecttypen en de lijst attributen opgehaald bij de Stelselcatalogus. Vervolgens wordt de terugmelding, nadat deze correct is ingevuld, verstuurd naar de Digimelding Core. Deze use case is uitgewerkt in een separaat document [4].

Het plaatsen van nieuwe terugmeldingen kan worden beschouwd als het primaire doel van het Digimelding Portaal. Secundaire doelen zijn dan het opvragen van (details van) eerder gedane terugmeldingen en het intrekken van een eerder gedane terugmelding. Vanwege het belang van de use case "Meld gerede twijfel" is deze verder uitgewerkt in een zogenaamde use case realization in sectie 4.3.

4.2.2 *UC044S Trek terugmelding in*

Na het plaatsen van een terugmelding kan het zijn dat er reden is om deze terugmelding weer in te trekken. Bijvoorbeeld als er een fout is gemaakt tijdens de intrekking of als de gegevens naderhand toch correct blijken te zijn. De medewerker van de afnemersorganisatie kan een lijst van eerder gedane terugmeldingen opvragen. Van een specifieke terugmelding kan men vervolgens de details bekijken, met daarbij onder andere de status. Terugmeldingen met de status "gemeld" of "in onderzoek" kunnen vervolgens ingetrokken worden. Bij een intrekking dient altijd een toelichting gegeven te worden. De use case is uitgewerkt in een separaat document [5].

4.2.3 *UC045S Raadpleeg terugmelding*

Het kan voor een (medewerker van een) afnemersorganisatie interessant zijn om de voortgang van een terugmelding te volgen. De afnemer kan uit een lijst met gevonden terugmeldingen een specifieke terugmelding selecteren. Hij/zij zal dan een detailoverzicht krijgen van deze terugmelding met onder andere de huidige status. De use case is uitgewerkt in een separaat document [6].

4.2.4 *UC046S Raadpleeg terugmeldingen*

Een medewerker van een afnemersorganisatie kan een lijst met eerder gedane terugmeldingen opvragen via het Digimelding Portaal. Deze lijst zal altijd alleen terugmeldingen bevatten die gedaan zijn vanuit de eigen organisatie. De zoek-criteria zijn Digimelding-kenmerk, meldingskenmerk, van-datum, tot-datum en status. De use case is uitgewerkt in een separaat document [7].

4.2.5 *Use Case realization – UC041S Meld gerede twijfel*

De belangrijkste use case heeft betrekking op het opstellen en versturen van een nieuwe terugmelding. Daarom is besloten om UC041S Meld gerede Twijfel verder uit te werken in een zogenaamde use case realization. De use case is uitgewerkt in een afzonderlijk document [12].

5 Logical view

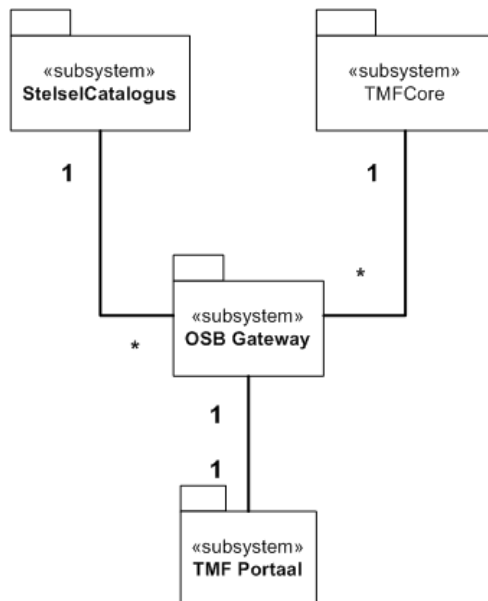
In dit hoofdstuk wordt de logische opbouw van de Digimelding Portaal applicatie besproken. De eerste sectie geeft een overzicht van de applicatie op subsystemen niveau, inclusief externe systemen. Vervolgens wordt het MVC pattern besproken in de tweede sectie. In de laatste sectie worden de belangrijkste delen van de applicatie besproken aan de hand van klassediagrammen.

5.1 Overzicht

Het Digimelding-systeem bestaat uit enkele subsystemen, te weten:

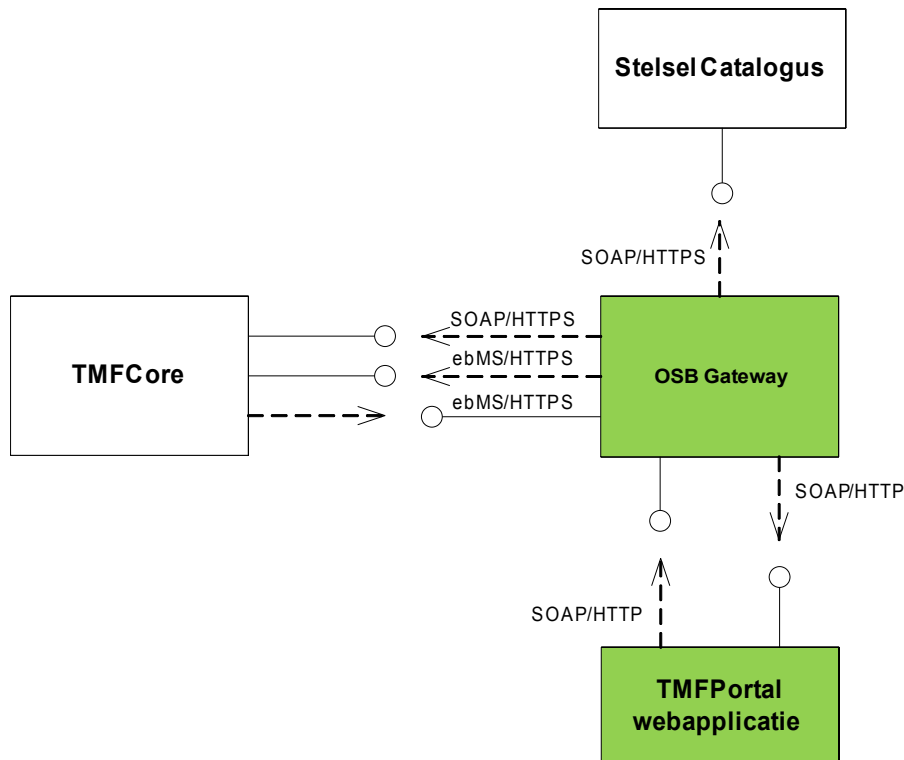
1. Digimelding Portaal, de webapplicatie waarmee een afnemersorganisatie terugmeldingen kan doen.
2. Digikoppeling Gateway, de dienst waar via het Digimelding Portaal communiceert met de buitenwereld. De gateway vertaalt berichten in het WUSLite protocol naar berichten in het ebMS- of WUS-protocol en vice versa.
3. Stelselcatalogus, webservice-dienst waarbij een lijst met basisregistraties, een lijst met objecttypen voor een bepaalde basisregistratie en een lijst met attributen voor een bepaald objecttype bevroegbaar zijn.
4. DigimeldingCore, webservice-dienst waarbij een lijst met terugmeldingen is op te vragen, alsmede de details van een specifieke terugmelding. Daarnaast is het ook mogelijk om een eerder gedane melding in te trekken of om een nieuwe terugmelding toe te voegen.

De samenhang tussen de subsystemen wordt in onderstaand diagram weergegeven.



Binnen het Digimelding-systeem zijn verschillende communicatie-protocollen in gebruik. In onderstaand component diagram zijn de volgende subsystemen te onderscheiden:

1. Stelselcatalogus, dit subsysteem is een service provider die communiceert via SOAP/HTTPS, ook wel WUS genoemd in de Digikoppeling standaarden.
2. Digimelding Core, dit subsysteem communiceert ook volledig via HTTPS. Daarbij wordt ebMS gebruikt voor asynchrone communicatie, zoals het toevoegen van een nieuwe terugmelding of het intrekken van een eerder gedane terugmelding.
3. Digikoppeling Gateway, dient als proxy voor de communicatie tussen de Stelselcatalogus (SOAP/HTTPS uitgaand) en Digimelding Core (SOAP/HTTPS uitgaand, ebMS/HTTPS uitgaand+inkomend) enerzijds en het Digimelding Portaal (SOAP/HTTP uitgaand en inkomend) anderzijds.
4. Digimelding Portaal, verstuurt alle berichten via SOAP/HTTP. De Antwoordservice welke deel is van het Digimelding Portaal ontvangt ook antwoordberichten via SOAP/HTTP. Dit zijn ebMS/HTTPS antwoordberichten van Digimelding Core welke door de Digikoppeling Gateway vertaald zijn.



5.2 Model-View-Controller (MVC)

Bij de implementatie van het Digimelding Portaal is ervoor gekozen om een webapplicatie te implementeren op basis van het Model-View-Controller (MVC) patroon [11]. Het MVC patroon zorgt voor een nette scheiding tussen de verschillende lagen. Door het afscheiden van de View was het mogelijk om hieraan een software ontwikkelaar zelfstandig te laten werken, terwijl anderen zich concentreerden op het Model en de

Controller. De keuze voor het MVC patroon heeft dan ook meerdere voordelen:

1. Duidelijke scheiding van verschillende lagen waardoor de verantwoordelijkheden duidelijk worden verdeeld binnen de applicatie.
2. Mogelijkheid tot het (semi-)separaat ontwikkelen van delen van de applicatie.
3. Betere testbaarheid doordat het gemakkelijk is om mock-objecten te gebruiken, welke voldoen aan de te implementeren interfaces.

De view is verantwoordelijk voor de interactie met de eindgebruiker. Het betreft een web-based presentatie die compatibel is met Internet Explorer en Firefox. De werking van het MVC-patroon en de rol van de verschillende componenten binnen het Digimelding Systeem wordt hier duidelijk gemaakt aan de hand van een praktijkvoorbeeld. De eindgebruiker volgt het volgende scenario:

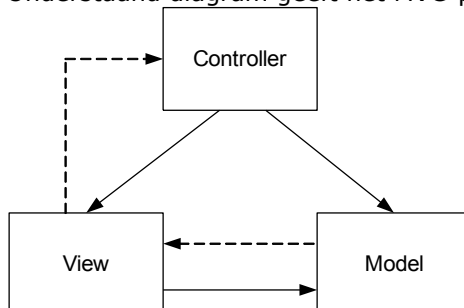
1. Gebruiker opent een webbrowser en surft naar de Digimelding Portaal applicatie.
2. Men kiest voor "Overzicht" om een lijst met eerder gedane terugmeldingen op te vragen.
3. Op het eerste scherm voert men enkele zoekcriteria in en start de zoekopdracht.
4. Vervolgens wordt een overzicht weergegeven van eerder gedane terugmeldingen die voldoen aan de zoekcriteria.

Aangenomen wordt dat de gebruiker een valide zoekcriteria invoert en dat er terugmeldingen zijn die aan de zoekcriteria voldoen.

Hetzelfde scenario verloopt aan de server kant volgens het MVC-patroon als volgt:

1. De Tomcat Server krijgt een verzoek voor de Digimelding Portaal applicatie; een nieuwe thread wordt gestart.
2. De controller (Controller) behorende bij de startpagina krijgt het verzoek binnen en bouwd de startpagina (View).
3. Als de gebruiker op de link "overzicht" drukt dan zal de bijbehorende controller (Controller) de zoekcriteria-pagina opbouwen.
4. Nadat de gebruiker de zoekcriteria heeft ingevoerd zal de controller (Controller) eerst die criteria valideren. Aangezien de validatie slaagt wordt vervolgens een lijst met terugmeldingen (Model) opgehaald. Deze lijst wordt verwerkt door de Controller en meegegeven aan de View alwaar de lijst wordt weergegeven.

Onderstaand diagram geeft het MVC-patroon schematisch weer.



De verantwoordelijkheden van de drie lagen zijn als volgt binnen de Digimelding Portaal applicatie:

View, rendering van informatie uit het Model, alsmede toevoegingen hieraan. Communicatie met het Model verloopt via de Controller.

Controller, handelt verzoeken uit de View af, stuurt informatie naar en haalt informatie uit het Model en geeft deze door aan de View.

Model, opslag van en toegang verlenen tot gegevens zoals bijvoorbeeld terugmeldingen.

Het model heeft hier nog enige verdere uitleg. Het Digimelding Portaal heeft zelf geen persistente opslag van gegevens (uitgezonderd logbestanden), maar maakt hiervoor gebruik van twee sets webservices die worden aangeboden door de Stelselcatalogus en Digimelding Core. Communicatie met deze twee diensten vindt plaats via de Digikoppeling Gateway die als een soort van proxy gezien kan worden. Data-toegang is dus mogelijk door de combinatie van:

1. De webservice providers aan de Stelselcatalogus en de DigimeldingCore kant.
2. Digikoppeling Gateway, als proxy zijnde.
3. Het Model-gedeelte van de Digimelding Portaal applicatie waarin de services op de Digikoppeling Gateway worden aangeroepen en het resultaat wordt vertaald naar het interne datamodel.

Alle persistente opslag van data vindt plaats op de Stelselcatalogus en de Digimelding Core servers.

5.3 Klassediagrammen

Een belangrijk deel van de implementation view is het geven van een overzicht van de belangrijkste klassen in het ontwerp. In de volgende sectie zal eerst de algemene opzet van de applicatie worden besproken, gevolgd door klassediagrammen voor respectievelijk de Stelselcatalogus- en Digimelding Core-kant van het Digimelding Portaal.

5.3.1 Algemene opzet

Het logisch ontwerp van het Digimelding Portaal berust op enkele belangrijke ontwerpbesluiten. Zo is er voor gekozen om daar waar mogelijk gebruik te maken van interfaces. Dit heeft als voordeel dat een achterliggende klasse makkelijk kan worden vervangen indien nodig. De configuratie vindt plaats door middel van dependency injection met behulp van het Spring framework.

Een andere belangrijke ontwerpkeuze is het gebruik van een intern en een extern datamodel. Deze keuze versterkt de scheiding tussen de lagen (MVC), doordat de View (als in MVC) uitsluitend met het interne datamodel werkt. De belangrijkste data-objecten, te weten BasisRegistratie, BROBject, BRObjectAttribuut (Stelselcatalogus-kant) en Terugmelding (Digimelding Core-kant), alsmede hun vertaling naar het externe datamodel zijn reeds besproken in de use case realization van UC041S Meld gereede twijfel [12].

In de volgende secties zullen afzonderlijk de Stelselcatalogus- en de DigimeldingCore-kant van de applicatie worden besproken. Deze beide kanten komen pas in de View (als in MVC) samen, waar ze in samenhang worden gebruikt voor bijvoorbeeld het doen van een nieuwe terugmelding.

Bekijk eventueel de use case realization van UC041S Meld gerede twijfel [12] om te zien hoe beide kanten in de praktijk gebruikt worden.

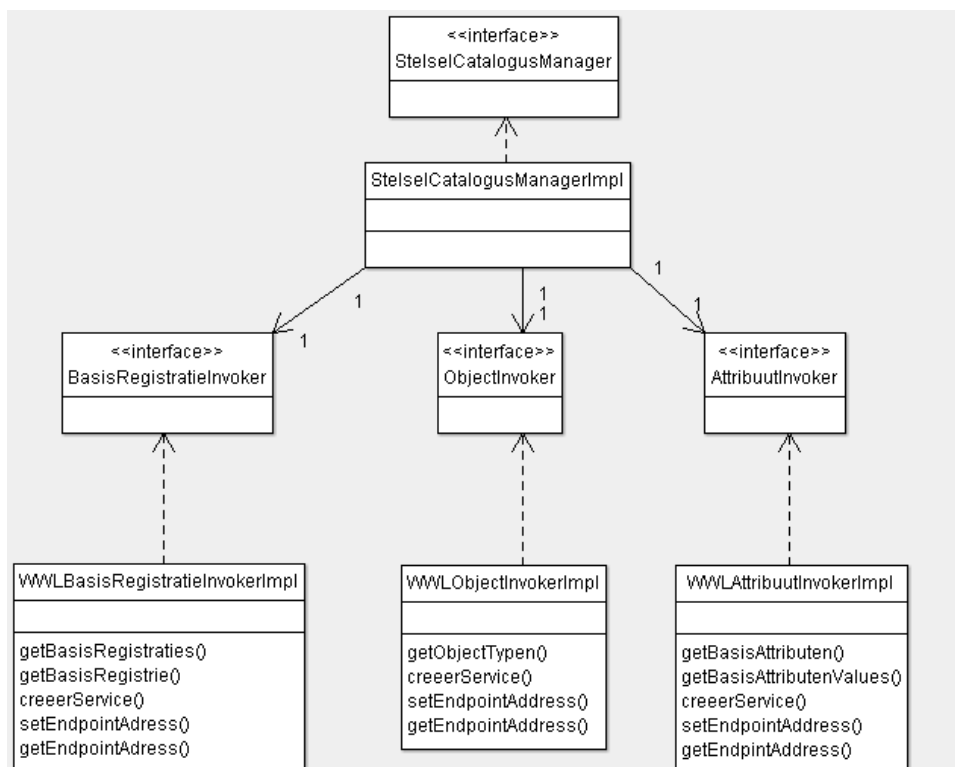
5.3.2

Stelselcatalogus

Onderstaand klassediagram geeft een logisch overzicht van de belangrijkste klassen aan de Stelselcatalogus-kant van de Digimelding Portaal applicatie. Het Digimelding Portaal heeft een `StelselcatalogusManagerImpl` welke de `StelselcatalogusManager`-interface implementeert. De functie van deze klasse is het bijhouden van een set Invokers welke gerelateerd zijn aan de Stelselcatalogus. Deze Invokers zijn beschikbaar via een interface. Een Invoker wordt gebruikt om verschillende functies aan te roepen, bijvoorbeeld het opvragen van een lijst van basisregistraties.

Het klassediagram geeft drie concrete Invokers weer:

1. `WWLBasisRegistratieInvokerImpl`, gebruikt de implementatie aan de Stelselcatalogus-kant in de zin dat deze echt de Stelselcatalogus-webservice gebruikt. De webservice kan gebruikt worden om een lijst met basisregistraties op te halen. De afkorting WWL staat voor WUS-WUSLite, wat slaat op de verbinding van de Digikoppeling Gateway naar de Stelselcatalogus (WUS) en de verbinding van de Digimelding Portaal applicatie naar de Digikoppeling Gateway (WUSLite).
2. `WWLObjectInvokerImpl`, idem aan voorgaande maar dan voor basisregistratie-objecten.
3. `WWLAttribuutInvokerImpl`, idem aan voorgaande maar dan voor attribuuttype-objecten.



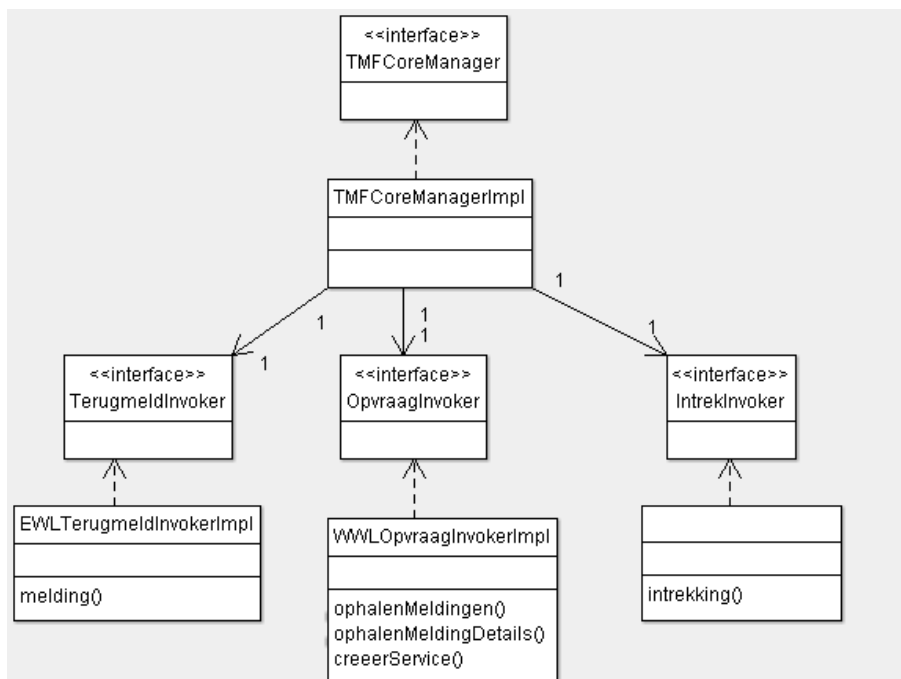
5.3.3

DigimeldingCore

De opzet van de DigimeldingCore-kant van de Digimelding Portaal applicatie lijkt sterk op de implementatie van de Stelselcatalogus-kant. Ook hier wordt er gebruik gemaakt van een Manager-implementatie die een interface implementeert. De Manager heeft als taak het bijhouden van een set Invokers. De concrete Invokers implementeren alle een interface zodat de concrete klasse makkelijk kan worden vervangen indien nodig.

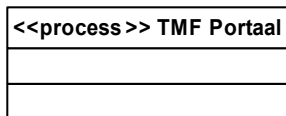
Het klassediagram geeft drie concrete Invokers weer:

1. EWLTerugmeldInvokerImpl, deze implementatie roept de ebMS webservice aan voor het doen van een nieuwe terugmelding. De afkorting EWL staat voor ebMS-WUSLite, wat wil zeggen dat de communicatie tussen de Digikoppeling Gateway en Digimelding Core verloopt via ebMS, terwijl de communicatie van het Digimelding Portaal naar de Digikoppeling Gateway verloopt via WUSLite.
2. WWLOp vraagInvokerImpl, deze implementatie maakt het mogelijk om een lijst met eerder gedane terugmeldingen op te vragen alsmede de details van een specifieke terugmelding. De afkorting WWL staat voor WUS-WUSLite.
3. EWLIntrekInvokerImpl, deze implementatie maakt het mogelijk om een intrekking te doen.



6 Process view

Kruchten [1] definieert een proces als een groep van taken die tezamen een uitvoerbare eenheid vormen. Zo'n uitvoerbare eenheid kan bestuurd worden in de zin van operaties als starten, herstellen, configureren en beëindigen. De enige uitvoerbare eenheid in die zin die hier van toepassing is, is het Digimelding Portaal.



Het Digimelding Portaal communiceert door middel van WUSlite met de Digikoppeling Gateway. WUSlite is een synchroon communicatieprotocol. Echter een deel van de communicatie tussen de Digikoppeling Gateway en Digimelding Core verloopt via een asynchroon protocol genaamd EBMS. De antwoorden die via EBMS bij de Digikoppeling Gateway belanden worden doorgestuurd naar de Antwoordservice binnen het Digimelding Portaal. De Antwoordservice heeft in zijn geheel geen communicatie met de eindgebruiker.

7 Implementation view

In de volgende sectie wordt de opdeling van de Digimelding Portaal applicatie volgens het MVC pattern[11] besproken. Daarbij worden alle voorkomende packages gemapt naar een laag. In de tweede sectie wordt de Antwoordservice besproken.

7.1 Digimelding Portaal

Het Digimelding Portaal maakt gebruik van het MVC design pattern [11] om de applicatie te scheiden in duidelijk afzonderlijke delen. Daarbij zijn de volgende delen te onderscheiden:

- Model, het model in MVC wordt doorgaans ruim opgevat, niet alleen het datamodel maar ook de toegang tot en bewerkingen op de data worden gezien als onderdeel van het Model. Bij het Digimelding Portaal valt deze laag uiteen in twee delen, te weten een intern en een extern deel. Hierbij wordt het interne deel gebruikt in de View en in de Controller, terwijl het externe deel uitsluitend gebruikt wordt in het Model.
- View, de interface waarmee de gebruiker met het systeem communiceert. Het Digimelding Portaal maakt gebruik van JSP- en Servlet technologie voor de View.
- Controller, deze laag handelt events uit de View af en roept functies in het Model aan om gegevens op te halen of te bewerken.

7.1.1 Model (interne deel)

Het interne deel van het model heeft betrekking op datatoegang en het datamodel vanuit de View en de Controller. Het interne deel maakt op zijn beurt weer gebruik van het externe deel om te communiceren met de webservices.

7.1.1.1 Datatoegang

Alle toegang tot het model verloopt via een Manager en een aantal Invokers. Er is een StelselcatalogusManager welke toegang biedt tot Invokers voor basisregistraties, objecten en attributen. Aan de andere kant is er ook een DigimeldingCoreManager welke Invokers biedt voor het doen van een terugmelding, het opvragen van (details van) eerder gedane terugmeldingen en het intrekken van een terugmelding. Vanuit het interne deel worden functies aangeroepen uit het externe deel van het Model. Deze externe functies communiceren op hun beurt weer met de webservices. Zie ook sectie 5.3 voor een schematisch overzicht van deze klassen. De volgende tabel biedt een overzicht van de packages die de Managers en Invokers bevatten.

Package	Beschrijving
com.ovsoftware.ictu.osb.Digimeldingportal.service.common	Instellingen, conversie van interne naar externe datamodel en vice versa, bericht handlers voor nabewerking en loggen berichten.
com.ovsoftware.ictu.osb.Digimeldingportal.service.stelselcatalogus	Interface voor de Manager en Invokers.

com.ovsoftware.ictu.osb.Digimeldingportal.service.stelselcatalogus.impl	Implementatie van de Manager.
com.ovsoftware.ictu.osb.Digimeldingportal.service.stelselcatalogus.impl.mock	Mockversies van alle Invokers. De BasisRegistratieInvoker en de ObjectInvoker zijn nog geïmplementeerd dmv mockobjecten.
com.ovsoftware.ictu.osb.Digimeldingportal.service.stelselcatalogus.impl.wus.wuslite	WUS-WUSLite implementatie van de AttribuutInvoker.
com.ovsoftware.ictu.osb.Digimeldingportal.service.tmfcore	Interface voor de Manager en de Invokers.
com.ovsoftware.ictu.osb.Digimeldingportal.service.tmfcore.impl	Implementatie van de Manager.
com.ovsoftware.ictu.osb.Digimeldingportal.service.tmfcore.impl.ebms.wuslite	De ebMS-WUSLite implementatie van de Intrek- en TerugmeldInvokers.
com.ovsoftware.ictu.osb.Digimeldingportal.service.tmfcore.impl.mock	Mockversies van alle Invokers. Geen enkele wordt nog actief gebruikt.
com.ovsoftware.ictu.osb.Digimeldingportal.service.tmfcore.impl.wus.wuslite	WUS-WUSLite implementatie van de OpvraagInvoker.

7.1.1.2

Datamodel

Het datamodel van het Digimelding Portaal bestaat uit twee delen, te weten een intern datamodel en een extern datamodel. Het interne datamodel wordt door de View gebruikt. Het externe datamodel wordt gebruikt bij het aanroepen van services door de Invokers in de Controller.

Package	Beschrijving
com.ovsoftware.ictu.osb.Digimeldingportal.service.stelselcatalogus.datatypes	Data-objecten voor een basisregistratie, object, attribuut en objectinfo inclusief bijbehorende lijst attributen.
com.ovsoftware.ictu.osb.Digimeldingportal.service.Digimeldingcore.datatypes	Terugmelding en Terugmeldingdetails datatypes inclusief alle daarin gebruikte types zoals Bijlage, ContactGegevens, ObjectAttribuut en TerugmeldingKern.
com.ovsoftware.ictu.osb.Digimeldingportal.service.Digimeldingcore.datatypes.mock	TerugmeldingCompleet-object, welke bestaat uit een Terugmelding en TerugmeldingDetails paar voor de MockInvokers.

7.1.2

Model (externe deel)

Het externe deel van het model behelst alle datatoegang en datatypes die nodig zijn voor communicatie met de webservices van zowel de Stelselcatalogus als DigimeldingCore. Voor het aanroepen van webservices wordt gebruik gemaakt van gegenereerde klassen. Deze klassen zijn met behulp van de WSImport-tool van JAX-WS v2.1 (welke onderdeel is van Java v1.6 update 11) aangemaakt. Daarbij zijn de door de Digikoppeling Gateway getransformeerde WSDLs als invoer gebruikt. De gegenereerde klassen zijn onderverdeeld in drie subprojecten, te weten:

1. Stelselcatalogus-WUS-WUSlite, met daarbij de packagenaam `com.ovsoftware.ictu.osb.gateway.stelselcatalogus.wus.wuslite`
2. `tmfCore-EBMS-WUSlite`, met daarbij de packagenaam `com.ovsoftware.ictu.osb.gateway.tmfcore.ebms.wuslite`
3. `DigimeldingCore-WUS-WUSlite`, met daarbij de packagenaam `com.ovsoftware.ictu.osb.gateway.tmfcore.wus.wuslite`

De subprojecten worden in de vorm van bibliotheken (JAR-files) toegevoegd aan het Digimelding Portaal project zelf. Hierdoor is het mogelijk om vanuit het interne deel van het Model, dat wil zeggen vanuit de Invokers, de webservices en de datatypes te gebruiken.

7.1.2.1 Datatoegang en datamodel

Zowel de klassen voor datatoegang als het datamodel bevinden zich in de hierboven genoemde packages. Het opvragen van data gebeurt via de klassen eindigend op `Service`. Deze klassen steunen daarbij zwaar op het JAX-WS deel van het Java framework. De andere klassen in de package zijn allemaal datatypes op basis van de WSDL.

De service-functies worden aangeroepen door de Invokers vanuit het interne deel van het Model. Daarbij krijgt de aanroepende Invoker een datatype terug uit het externe deel van het Model, welke om wordt gezet naar een intern datatype met behulp van de Converter-klasse uit de `common-package`. De handlers uit diezelfde package worden gebruikt om het bericht te loggen en om het aan te passen (namespace, soapaction) waar nodig.

7.1.3 View

De view bestaat uit een webbased interface die gemaakt is door middel van JSP- en servlet-technologie. Gebeurtenissen in de View worden door de Controller afgehandeld. Een overzicht van alle bestanden in de map "`src/main/webapp`" is te vinden in de volgende tabel.

Bestanden	Beschrijving
<code>css/*.css</code>	De CSS-stylesheets welke worden gebruikt voor de opmaak van de GUI.
<code>images/*.png</code> en <code>*.gif</code>	Afbeeldingen die gebruikt worden in de GUI.
<code>WEB-INF/*.xml</code> en <code>*.xsd</code>	Configuratiebestanden voor de webapplicatie.
<code>WEB-INF/defs/definitions.xml</code>	Definities van tiles (zie laatste regel tabel).
<code>WEB-INF/jsp/*.jsp</code>	De afzonderlijke JSP-pagina's die tezamen de applicatie vormen.
<code>WEB-INF/jsp/tiles/*.jsp</code>	Delen van pagina's die meerdere malen voorkomen zoals het menu. Werking gebaseerd op Apache Tiles v2.

7.1.4 Controller

De controller bestaat uit een verzameling controller-classes die invoer krijgen vanuit de View en vervolgens het Model bewerken. Daarbij worden de volgende packages gebruikt:

Package	Beschrijving
com.ovsoftware.ictu.osb.tmfportal.web	Controller voor het start-scherm en het algemene gedeelte.
com.ovsoftware.ictu.osb.tmfportal.web.intrekken	Controller, validator en data object voor een intrekking.
com.ovsoftware.ictu.osb.tmfportal.web.overzicht	Controllers voor het afhandelen van een zoekopdracht.
com.ovsoftware.ictu.osb.tmfportal.web.terugmelding	Controllers voor het afhandelen van een nieuwe terugmelding.

7.2

AntwoordService

Het Digimelding Portaal systeem bestaat naast de Digimelding Portaal applicatie zelf nog uit een Antwoordservice applicatie. Deze Antwoordservice is verantwoordelijk voor het ontvangen en loggen van inkomende berichten. Het betreft een zeer simpele applicatie welke instelbaar is via een Spring XML-bestand en de Apache Log4j configuratie. Een inkomend bericht is ofwel een foutmelding of een ontvangstbevestiging. Alle inkomende berichten worden uiteindelijk via Apache's Log4j weggeschreven.

8 Deployment view

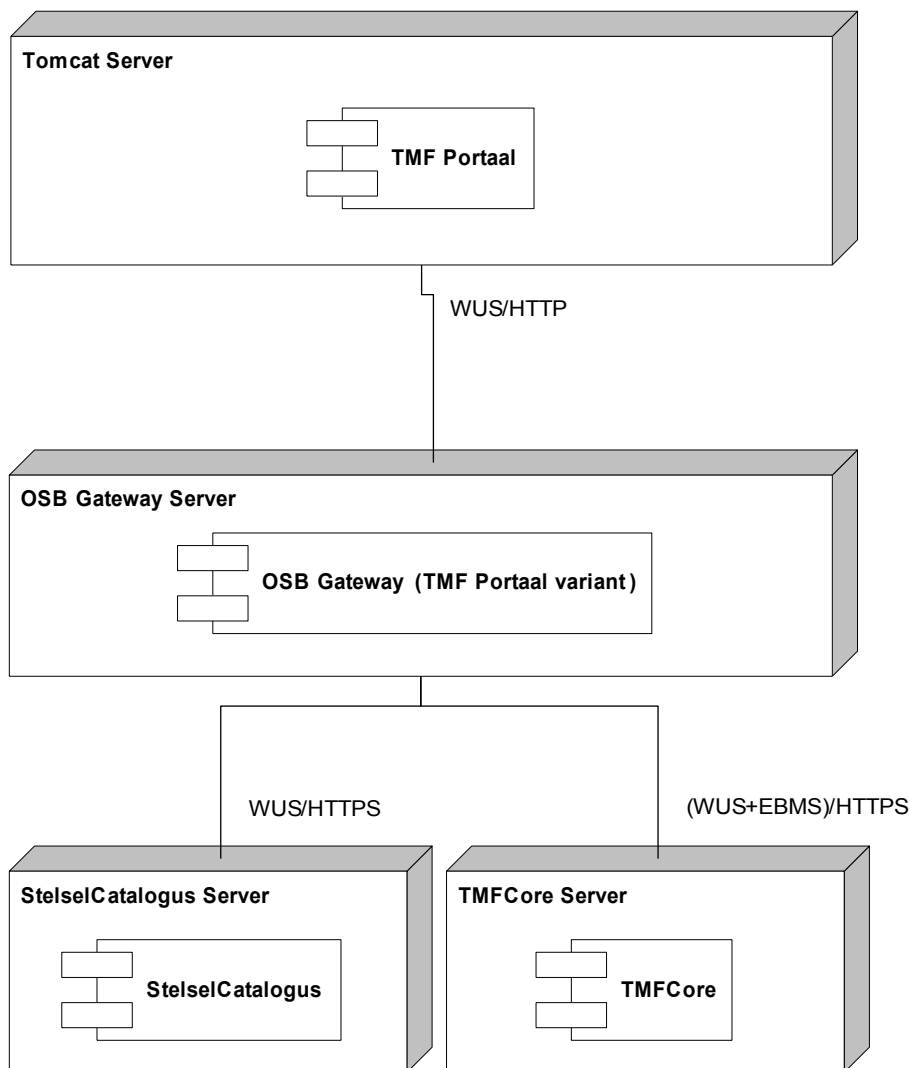
Binnen het 4+1 model van Kruger [1] kan de deployment view worden beschouwd als het laagste, meest concrete niveau. In de deployment view wordt hier gekeken naar de fysieke hardware topologie van de onderdelen die tezamen het Digimelding systeem vormen. Daarnaast wordt kort stilgestaan bij de installatie en het onderhoud van de applicatie.

8.1 Overzicht

Het volgende diagram geeft een globaal overzicht van de servers en de applicaties die deel uit maken van het gehele Digimelding Systeem. Van boven naar beneden zijn de volgende servers weergegeven:

- Tomcat Server: een server bij de afnemersorganisatie waarop de Digimelding Portaal applicatie geïnstalleerd staat.
- Digikoppeling Gateway Server: een server bij de afnemersorganisatie waarop de Digimelding Portaal variant van de Digikoppeling Gateway geïnstalleerd staat. Deze variant heeft enkele lichte aanpassingen ondergaan ten opzichte van de standaard Digikoppeling Gateway. Het is heden nog niet mogelijk een bestaande Digikoppeling Gateway te gebruiken in combinatie met het Digimelding Portaal.
- Stelselcatalogus Server: een server buiten de afnemersorganisatie waarmee de Digikoppeling Gateway communiceert. De Stelselcatalogus voorziet in het ophalen van lijsten met attributen en eventueel de huidige waarden van deze attributen.
- Digimelding Core Server: een server buiten de afnemersorganisatie waarmee de Digikoppeling Gateway communiceert, zowel synchroon (via WUS) als asynchroon (via EBMS). Digimelding Core kan overzichten van meldingen en details met betrekking tot een specifieke melding verschaffen. Daarnaast is het mogelijk om nieuwe terugmeldingen te plaatsen en oude terugmeldingen in te trekken.

In onderstaand deployment diagram zijn de basisregistraties die achter de Stelselcatalogus en Digimelding Core liggen met reden niet opgenomen. Vanuit de Digimelding Portaal applicatie gezien worden deze namelijk alleen benaderd via de Stelselcatalogus en Digimelding Core. Binnen het Digimelding Portaal is geen kennis nodig van de communicatie tussen deze en de basisregistraties. De Stelselcatalogus en Digimelding Core zijn om dezelfde reden juist wél opgenomen in het deployment diagram. Hoewel alle communicatie plaatsvindt via de Digikoppeling Gateway is de implementatie van het Digimelding Portaal wel afhankelijk van de door de Stelselcatalogus en Digimelding Core aangeboden webservices.



8.2 Installatie

De installatie van het Digimelding Portaal bestaat voor een afnemersorganisatie uit twee delen, te weten het Digimelding Portaal zelf en de Digikoppeling Gateway. Het is aan te bevelen om de Digikoppeling Gateway en het Digimelding Portaal op twee fysiek gescheiden systemen te installeren. De Digikoppeling Gateway is namelijk via internet verbonden met de Stelselcatalogus en DigimeldingCore. Ook zal toegang tot het Digimelding Portaal door de afnemersorganisatie beperkt dienen te worden tot geautoriseerde gebruikers. Als onderdeel van de set met op te leveren documenten is een uitgebreide installatiehandleiding [13] beschikbaar waarin de installatie van de Digikoppeling Gateway, Tomcat en het Digimelding Portaal wordt beschreven.

8.3 Beheer

De configuratie van het Digimelding Portaal zal waarschijnlijk slechts eenmalig plaats hoeven te vinden tijdens het installatieproces. Verwacht wordt dat wijzigingen in instellingen als de locatie van de Digikoppeling Gateway, alsmede de Stelselcatalogus en Digimelding Core slechts zelden voorkomen. De belangrijkste taak van de beheerder zal dan ook ondersteuning van gebruikers zijn in het geval dat problemen met het gebruik van het Digimelding Portaal optreden. De beheerder heeft hierbij

de beschikking over de log-functionaliteit van het Apache Log4j framework. Dit wordt nader toegelicht in de beheerdershandleiding [14] welke onderdeel vormt van de set van opgeleverde documenten.

9 Kwaliteit

De kwaliteit van de opgeleverde applicatie is afhankelijk van meerdere factoren, zoals de opbouw van de applicatie, dekkingsgraad van de testen en de bijgeleverde documentatie. In de volgende secties zullen deze factoren kort worden toegelicht.

9.1 **Opbouw applicatie**

Bij de opbouw van de applicatie is rekening gehouden met toekomstige aanpassingen en/of uitbreidingen. Er wordt daar waar mogelijk gebruik gemaakt van interfaces die het mogelijk maken achterliggende klassen aan te passen. Met behulp van het Spring framework wordt door middel van dependency injection de applicatie geconfigureerd om gebruik te maken van een specifieke achterliggende implementatie van een dergelijke interface. Daarnaast is er een duidelijke scheiding van verantwoordelijkheden door het gebruik van het MVC-patroon. Dit alles tezamen zorgt voor een duidelijke en flexibele opzet die toekomstgericht is.

Een ander belangrijk aspect van de opbouw is de integratie van voldoende logging-mogelijkheden. Het Digimelding Portaal maakt hiervoor gebruik van Apache's Log4j, dit is een uitgebreid logging-framework voor de Java programmeertaal. Door middel van een XML-bestand is het niveau waarop gelogt wordt alsmede de locaties waarnaar de uitvoer optreedt in te stellen. Zo is het bijvoorbeeld mogelijk om alleen logberichten op error-niveau te loggen. Uitgebreide logging-mogelijkheden zijn zowel nuttig voor beheer als onderhoud van de applicatie. Zie voor meer informatie de beheerhandleiding [14].

9.2 **Testplan en Acceptatieplan**

Bij een applicatie als het Digimelding Portaal is het belangrijk om alle functionaliteit grondig te testen. Deze testen dienen echter geautomatiseerd plaats te vinden, zodat ze regelmatig herhaald kunnen worden. Testen kunnen helpen bij het opsporen van fouten in de implementatie of in de configuratie. In het Digimelding Portaal project wordt gebruik gemaakt van zogenaamde unit-testen met behulp van JUnit. Door middel van dit framework, kunnen losse klassen en functies getest worden, en ook uitgebreidere test-scenario's worden uitgevoerd. Daarnaast is in het Digimelding Portaal project gebruik gemaakt van zogenaamde Mock-objecten. Door het gebruik van interfaces en dependency injection is het mogelijk om de echte Webservice-klassen (Invoker-implementaties) te verwisselen met Mock-klassen. Hiermee kan de interface volledig getest worden zonder dat de echte webservices benodigd zijn. Een uitgebreide beschrijving van de testen [10], alsmede het acceptatieplan [15] zijn te vinden in afzonderlijke documenten.

9.3 **Dekkingsgraad, Code checkstyle en Complexiteitsrapportage**

Om de kwaliteit van de code te kunnen garanderen zijn de dekkingsgraad van de testen, de style van de code en de complexiteit gecontroleerd. De dekkingsgraad van de testen bedraagt op de kern-onderdelen minimaal 80% en loopt op tot zelfs 100%. Een onderdeel dat hier expliciet buiten valt is de View. Voor deze JSP-pagina's is het niet mogelijk om geautomatiseerde JUnit testen te maken. Hiervoor zijn testscenario's opgesteld, welke terug zijn te vinden in het testplan [10].

De style van de code is daarnaast ook van belang. Daarom wordt gebruik gemaakt van een subset van de SUN code conventie waarin gecontroleerd wordt op belangrijke en veel voorkomende fouten. De controles die daarbij uit staan zijn gelimiteerd tot whitespace controles en dergelijke die geen schadelijke gevolgen hebben. Het code checkstyle rapport [16] wordt meegeleverd in de vorm van een website.

Naast de code coverage en de checkstyle kan het ook interessant zijn om de complexiteit van de code te kunnen bekijken. Binnen dit project wordt de complexiteit beschreven door het bepalen van de zogenaamde "Cyclomatic Complexity" als ontwikkeld door McCabe [17]. Hierbij wordt het aantal lineair onafhankelijke paden door de code gemeten. De complexiteit van zowel klassen als complete packages is opgenomen in het rapport [16] dat gegenereerd is door Cobertura.

9.4 Javadoc

Voor toekomstig onderhoud en/of uitbreidingen is het belangrijk dat het project wordt opgeleverd met voldoende documentatie. Dit document speelt daarin een belangrijke rol; het geeft een overzicht van de applicatie en het project in zijn geheel. Daarnaast is het echter ook belangrijk om te beschikken over documentatie op technisch niveau. De gebruikte programmeertaal (Java) kent hiervoor de Javadoc generator welke op basis van commentaar in de code de API documentatie kan genereren.

De Digimelding Portaal applicatie is volledig voorzien van Javadoc commentaar zodat alle klassen en de functies daarin gedocumenteerd zijn. Daarnaast zijn de complexere functies ook voorzien van commentaar binnen de functie, zodat overal de werking duidelijk beschreven is. De gegenereerde documentatie [16] is bijgesloten als onderdeel van de set van opgeleverde documenten.