

DSDM

Analyse

Project Kaartenbalie²

"Browse your way through"

B3Partners

Contactpersoon Marc Vloemans

Door:

Chris Kramer

Utrecht, 27-09-2007

1 Inhoudsopgave

1	Inhoudsopgave	2
2	Algemeen	3
2.1	Inleiding	3
2.2	Bedrijfsgebied	3
3	Haalbaarheidsonderzoek	4
3.1	Beschrijving van het huidige systeem	4
3.2	Problemen	4
3.3	Doelstellingen en functionele eisen voor het systeem	4
3.4	Niet functionele eisen	5
3.5	Mogelijke alternatieven	5
3.6	Conclusie	5
4	Bedrijfsanalyse	6
4.1	Bedrijfsdoelstellingen	6
4.2	Te automatiseren gegevens	6
4.3	Scope	6
4.4	Essentiële gebieden	6
5	Use Cases	7
5.1	Demo gebruikers	7
5.2	Gebruikers als onderdeel van een bedrijf	8
5.3	Beheerders van het bedrijf	9
5.4	B3Partners beheer	10
6	Klassendiagram	11
6.1	Klassen Kaartenbalie	11
6.2	Klassen Datawarehousing	12
7	Bijlage 1: MoSCoW Analyse	13
7.1	Timebox I	13
7.2	Timebox II	14
7.3	Timebox III	15
7.4	Timebox IV	16
8	Bijlage 1: Gegevensfiltratie	17
9	Bijlage 3: Globale werking kaartenbalie	18

2 Algemeen

2.1 Inleiding

Voor u ligt het rapport Haalbaarheidsstudie en Bedrijfsanalyse. In dit rapport wordt nader ingegaan op het bedrijfsgebied en de oplossing die zal worden gemaakt.

2.2 Bedrijfsgebied

Dit project wordt gedaan in opdracht van B3Partners en het eindproduct zal worden gebruikt door verschillende bedrijven en (overheid)instellingen welke geografische informatiesystemen nodig hebben.

De bedrijven en (overheid)instellingen beschikken over geografische informatie in veel verschillende formaten waarin data staat opgeslagen met betrekking tot bijvoorbeeld waterwingebieden, grensgebieden, wijken, gemeenten, etc. Maar denk ook aan de locaties van bomen, verkeersborden en bouwplannen.

Zoals al misschien wel blijkt is dit een ongelofelijke hoeveelheid aan informatie die ook nog eens op een te groot aantal manieren is opgeslagen (in databases, in Excel, in Word, met of zonder geografische coördinaten, of gewoon nog op papier). Om al deze data inzichtelijk te maken is te zijner tijd de backend van Kaartenbalie ontwikkeld. Deze informatiefabriek combineert alle verschillende databronnen (waar mogelijk) in één of meer gestandaardiseerde WMS diensten. Hierdoor wordt informatie uitwisselbaar en toegankelijk.

Hier omheen is de Kaartenbalie ontwikkeld. Een WMS informatiedienst die kaartgegevens kan doorsturen naar een browser. Belangrijk om te vermelden is dat de Kaartenbalie voorziet in beveiliging die standaard niet in het WMS protocol zit. Door deze beveiliging is het in principe mogelijk kaartmateriaal af te schermen van de gebruiker en per klant een pakket van kaarten samen te stellen.

B3Partners en de klanten van B3Partners willen graag meer inzicht in het gebruik van de dienst. Het is wenselijk dat online overzichten kunnen worden weergegeven waarin staat wie wat heeft gebruikt voor zowel B3Partners zelf als de klanten. Ook wil B3Partners extra diensten gaan leveren zoals WFS.

3 Haalbaarheidsonderzoek

3.1 Beschrijving van het huidige systeem

“Kaartenbalie is een webapplicatie, aangeboden door B3Partners, waarmee u de mogelijkheid heeft om kaartmateriaal dat op verschillende locaties (verschillende webserver) wordt aangeboden, op een eenvoudige en snelle manier te combineren en te beveiligen tegen ongewenst gebruik. Dit kaartmateriaal dient daartoe wel aangeboden te worden als WMS 1.1.1 service.”

Bron: <http://www.kaartenbalie.nl>

Kaartenbalie is ontwikkeld in de programmeertaal Java. Hierop zijn een aantal uitbreidingen (frameworks) toegepast om de inhoud van het project beter vorm te geven. Als Servlet framework is Struts gebruikt om door middel van Model-View-Control een logische indeling aan de site te geven. Vervolgens is het Hibernate framework gebruikt om de objecten met de database te mappen.

Op dit moment worden alle WMS aanroepen via een servlet geregeld. Deze servlet zorgt er voor dat de gebruiker gecontroleerd wordt, dat de opgevraagde kaarten ook refereren naar het kaartmateriaal in de database en vraagt vervolgens aan de desbetreffende providers de gewenste informatie aan om deze uiteindelijk terug te sturen naar de cliënt.

3.2 Problemen

Op dit moment heeft de Kaartenbalie een aantal functionaliteiten tekort. Een uitbreiding hierop zou uitkomst bieden.

1. Het is bijvoorbeeld niet handig mogelijk om vast te leggen wat nu het werkelijke gebruik is van de kaartenbalie. Gegevens worden op dit moment door middel van csv bestanden en losse Access databases geanalyseerd. Deze manier van werken is niet altijd even handig en tijdrovend.
2. Verder heeft de kaartenbalie nog geen systeem om kosten te koppelen aan opgehaalde kaarten en gebruikers of bedrijven. De juiste informatie mist eigenlijk om een goed beeld te scheppen hoeveel het opvragen van kaarten nu per gebruiker kost.
3. Ook is het niet mogelijk om Kaartenbalie als WFS te gebruiken. Hierdoor mist het een belangrijk gedeelte extra dienstverlening welke eigenlijk prima ingebouwd zou kunnen worden. Wanneer WFS echter ingebouwd is, dan zijn de bovenstaande problemen met betrekking tot rapportage en kostenberekening ook hierop weer van toepassing.
4. Verder door deze problemen kan Kaartenbalie niet verder worden geëxploiteerd waardoor het zijn afzetmarkt op dit moment niet volledig kan benutten. Hierdoor is het belangrijk dat Kaartenbalie zo snel en goed mogelijk wordt uitgebreid.
5. Tevens is het op dit moment niet mogelijk om kaarten op bijvoorbeeld individueel of groepsniveau toe te staan of te blokkeren. Alles gaat nu nog op bedrijfsniveau.
6. Wijzigingen op kritische tabellen worden niet gezekeerd in een historie. Het verwijderen van gebruikers kan leiden tot destructieve resultaten in de toekomstige te ontwikkelen datamodellen en hun afhankelijkheden.

3.3 Doelstellingen en functionele eisen voor het systeem

De toevoeging aan de Kaartenbalie moet in staat zijn om een oplossing te bieden voor de omschreven problematiek. Hieronder volgt een lijst met vereisten welke nodig zijn om een oplossing te bieden.

1. Het verbruik moet meetbaar gemaakt worden, hiervoor moeten een aantal tabellen aan het datamodel worden toegevoegd. Vervolgens moet elke kaartaanvraag geregistreerd worden in deze tabellen zodat alle gegevens nadien eenvoudig uit te lezen zijn voor de beheerders. Hierbij moet het duidelijk zijn wanneer een kaart is opgevraagd, welke hoeveelheid data is verzonden, hoe lang het proces geduurd heeft (van zowel de cliënt naar de server, als van de server naar de WMS) en tot welke sessie de opvraag toebehoort. De sessie gegevens moeten tevens weer aan een gebruiker gekoppeld worden zodat precies na te gaan is bij welk bedrijf te kosten thuishoren. Voor het inzichtelijk maken moet een GUI ontwikkeld worden die voor beheerders per klant het verbruik kan weergeven. Tevens moet het uiteindelijk voor de klant mogelijk zijn om zelf ook deze gegevens online in te zien.
2. Er moet een prijs worden gekoppeld aan het verbruik of aan de Kaartenbalie dienst zelf. Het moet mogelijk zijn om een prijs per kaartaanvraag te registreren maar ook een abonnementsvorm te implementeren. In den beginsel is deze abonnementsvorm globaal (dat wil zeggen voor alle beschikbare kaarten van een klant) maar wanneer het systeem verder uitgebreid wordt moet het ook mogelijk worden om per kaart een abonnement af te sluiten.

3. Er moet tevens een prijs per periode kunnen worden vastgesteld en het moet te achterhalen zijn hoe deze prijs tot stand is gekomen. Ook moet er rekening mee worden gehouden dat er in de toekomst meerdere tariefplannen per kaart mogelijk zijn. Voor deze functionaliteit moet ten minste de beheerders GUI ontworpen worden waarmee de periode en de kosten in een totaaloverzicht, maar ook sessie specifiek kunnen worden bekeken. Ook moet er de beheerders GUI komen waarmee de kosten per kaart ingesteld kunnen worden. Hier dient rekening te worden gehouden met historie.
4. Het systeem dient te werken met een creditsysteem, waarbij het mogelijk zal moeten zijn credits te kopen door gebruik te maken van een betaalmethode als iDeal – hiervoor moet een conversieslag worden gemaakt van euro's naar credits - , credits bij te schrijven door de beheerders van B3Partners, credit af te schrijven op basis van een verbruik of tijdsperiode.
5. Naast deze functionaliteit moet ook WFS worden geïmplementeerd. WFS moet net als de WMS service via Kaartenbalie gaan verlopen zodat alle bovenstaande vereisten ook op WFS kunnen worden toegepast. Indien de planning het toestaat moet dus ook voor WFS het koppel, registreer en betalingstraject worden doorlopen. Hierbij valt ook weer te denken aan een abonnementsvorm of een prijs per mutatie. Het systeem dient dus zo gebouwd te worden dat het straks mogelijk is om voor zowel WMS als WFS abonnementen af te sluiten.
6. Ook is het wenselijk dat kaartmateriaal per gebruiker of groep als onderdeel van een bedrijf kan worden aangeboden. Op die manier kan het bedrijf zelf bepalen welke groepen en/of gebruikers toegang krijgen tot welke kaarten. Deze functionaliteit beschermt de klant en zorgt er voor dat de kosten voor bijvoorbeeld duurdere kaarten beter in de hand te houden zijn, wat bedrijven eerder zal overhalen om ook toegang tot deze duurdere kaarten aan te vragen.
7. Belangrijke data mogen niet meer verloren gaan. De entiteiten welke van groot belang zijn moeten in een 'datawarehouse' worden ondergebracht waar te allen tijde de historie van de gegevens terug te halen is.

3.4 Niet functionele eisen

- Het platform waar het systeem op draait blijft hetzelfde.
- De gebruikte technieken blijven Struts in combinatie met Hibernate
- Het systeem blijft even snel, er mag niet of nauwelijks verlies in responsetijd zijn met betrekking tot het ophalen van kaartmateriaal wanneer het registratiesysteem wordt geïmplementeerd.

3.5 Mogelijke alternatieven

- Het systeem en het proces zo laten als het nu is.
- Een losstaande rapportage module bouwen die aan de hand van de beschikbare gegevens zelf de berekeningen uitvoert, kosten toevoegt en achteraf facturen aanmaakt.

3.6 Conclusie

De uitbreidingen voor het systeem zoals omschreven zijn technisch haalbaar aangezien het een uitbreiding betreft waar met dezelfde technologieën en dezelfde frameworks gewerkt kan worden. Verder is een belangrijke stap voor het slagen van dit project al gezet; namelijk het bouwen van een WMS portaal. Dit portaal kan direct de verbruiksregistratie onder zijn hoede nemen. Er zijn wat dat betreft dus geen hindernissen. Aangezien het prototype de Kaartenbalie zelf is, is een nieuw prototype voor het registreren van gegevens niet noodzakelijk. Hierdoor kan veel tijd bespaard worden en wordt het haalbaarheidsonderzoek kort en bondig.

Ook het bouwen van de WFS servlet is geen onmogelijk taak. Deze lijkt qua structuur enorm veel op de WMS servlet logica, met als uitzondering dat de WFS servlet alleen textdata doorstuurt en geen fotomateriaal zoals WMS dat doet. De huidige WMS servlet is het 'proof of concept' voor WFS en daardoor is het ook niet nodig om hier een prototype voor te maken.

Het is wel belangrijk om binnen dit project goed te plannen en prioriteiten te stellen. Naar waarschijnlijkheid is er niet voldoende tijd om alles volledig te bouwen. Er zal veel tijd gaan zitten in het rapportagesysteem en de datawarehousing.

Juist door de eisen aan een strakke planning en het gebruik van prioriteiten is dit project geschikt voor DSDM. De oplevering gebeurt iteratief – er wordt immers eerst een verbruiksregistratie toegevoegd, daarna pas wordt een prijs toegevoegd en daarna volgt pas de afrekening. Ditzelfde traject geldt voor WFS met als uitzondering dat WFS servlet zelf nog gebouwd moet worden.

4 Bedrijfsanalyse

4.1 Bedrijfsdoelstellingen

De hoofddoelstelling van het bedrijf is het maken van winst in combinatie met een toekomstgerichte dienstverlening tegen een brede afzetmarkt. De uitvoering van dit project draagt direct bij aan deze doelstelling. Wanneer een oplossing wordt gebouwd voor de eerder omschreven problemen kan Kaartenbalie breder ingezet worden en is het service niveau een stuk hoger dan het nu is dankzij realtime rapportages.

4.2 Te automatiseren gegevens

De gegevens die uiteindelijk geautomatiseerd moeten worden zijn in te delen in drie deelprocessen. Allereerst is er het verbruiksregistratiesysteem, ten tweede moet het verbruik worden gekoppeld aan een prijsbepaling voor de kaarten en uiteindelijk moeten de gemaakte kosten worden geregistreerd en moet het bedrag van de lopende tegoeden afgehaald worden.

Hiernaast moet voor WFS een portaal ontwikkeld worden die als tussenstation kan dienen voor de cliënt en de services die achter de kaartenbalie draaien.

Verbruikregistratie

Nu wordt er periodiek een Apache Log uitgelezen naar een csv bestand welke vervolgens in een Access DB wordt geplaatst. Nadien worden er een x aantal bewerkingen over deze gegevens uitgevoerd om tot een verbruik per afnemer en snelheidsmeting te komen. Deze gegevens worden uiteindelijk gebruikt voor rapportage. In bijlage 1 staat een voorbeeld van de gegevens zoals deze op dit moment uit de log komen.

Het is de bedoeling dit proces in zijn geheel te automatiseren door de verbruiksdata direct op te slaan in de datastructuur van Kaartenbalie.

Prijsbepaling

Op dit moment wordt er afgerekend op basis van dataverbruik en responsesnelheid, waarbij het dataverbruik binnen een bepaalde bundel moet blijven en de responsetijden een voorwaarde zijn voor de dienstverlening.

Hiervoor is het de bedoeling de prijsbepaling voor kaartmateriaal ook op te slaan in de datastructuur van Kaartenbalie zodat abonnementen en dergelijke direct daarin kunnen worden aangepast.

Facturatie → Creditsysteem

De uiteindelijke facturatie voor de dienstverlening is op dit moment een afgesproken termijn bedrag. Er zijn nog geen situaties geweest waarbij het verbruik hoger was dan de toegestane limiet. Tevens zijn de responsetijden altijd binnen de afgesproken waarden geweest. Dit moet komen te vervallen en worden vervangen door een creditsysteem waarbij vooraf betaald kan worden voor de af te nemen dienst.

4.3 Scope

Dit project heeft tot doel uitbreidingen te bieden voor Kaartenbalie waarmee de genoemde doelstellingen vervuld kunnen worden. De uiteindelijke producten die tijdens dit project worden opgeleverd zijn een verbeterde webapplicatie met daarin uitbreidingen voor zowel de gebruiker als de beheerder.

Het bouwen van het systeem voor toegang, rapportage, betaling en WFS valt, wanneer de TimeBoxen dit toelaten, binnen de scope. Het geven van overzichten gebaseerd op de gegevens en het verbruik valt ook binnen de scope. De grafische weergave van deze gegevens valt echter weer buiten de scope en zal later opgepakt moeten worden.

4.4 Essentiële gebieden

Het moet in ieder geval mogelijk zijn om uitgebreid te kunnen rapporteren op basis van gegevens in de database. Tevens moet WFS tenminste zo minimaal mogelijk worden geïmplementeerd (Als doorgeef functionaliteit) en moet er tenminste een systeem worden gebouwd waarmee op abonnement niveau toegang kan worden gegeven en ontzegd.

Alle andere functionaliteiten zijn zeer gewenst en zullen wanneer de planning dit toelaat ook worden gebouwd, echter deze functionaliteiten worden niet essentieel geacht voor de werking van het systeem.

5 Use Cases

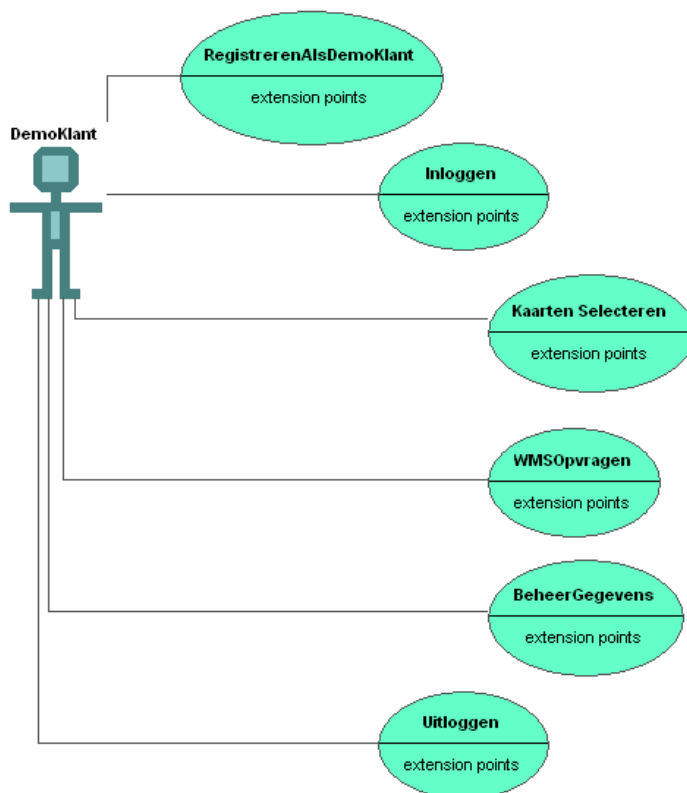
De use cases kunnen worden onderverdeeld in de vijf gebruikersrollen die het systeem op dit moment ondersteund of gaat ondersteunen.

- Demo gebruikers
- Gebruikers als onderdeel van een bedrijf
- Beheerders van het bedrijf
- B3Partners beheer

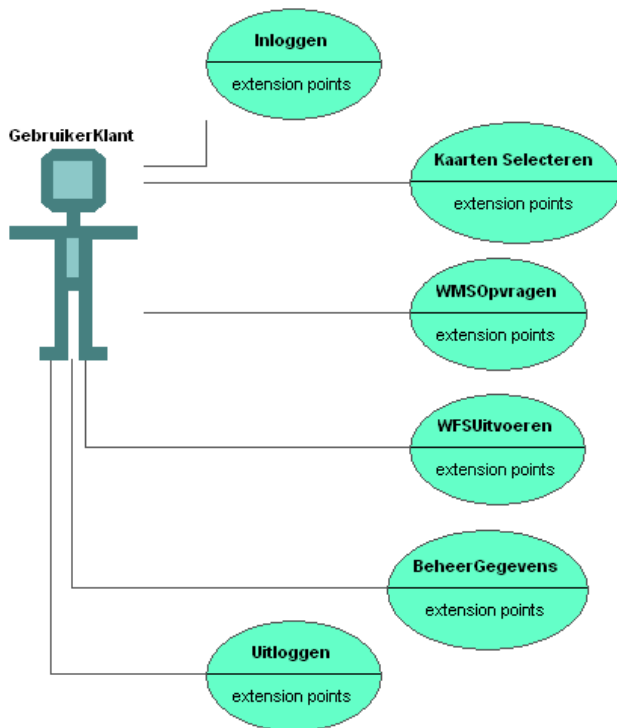
Een deel van de UseCases bestaat al. Welke dit zijn, staat in de omschrijving van de desbetreffende UseCase.

5.1 Demo gebruikers

Aan deze Use Case worden geen wijzigingen gemaakt. Het bestaande model is voldoende.



5.2 Gebruikers als onderdeel van een bedrijf



Al bestaande functionaliteiten

- Inloggen
- Kaarten Selecteren
- BeheerGegevens
- Uitloggen

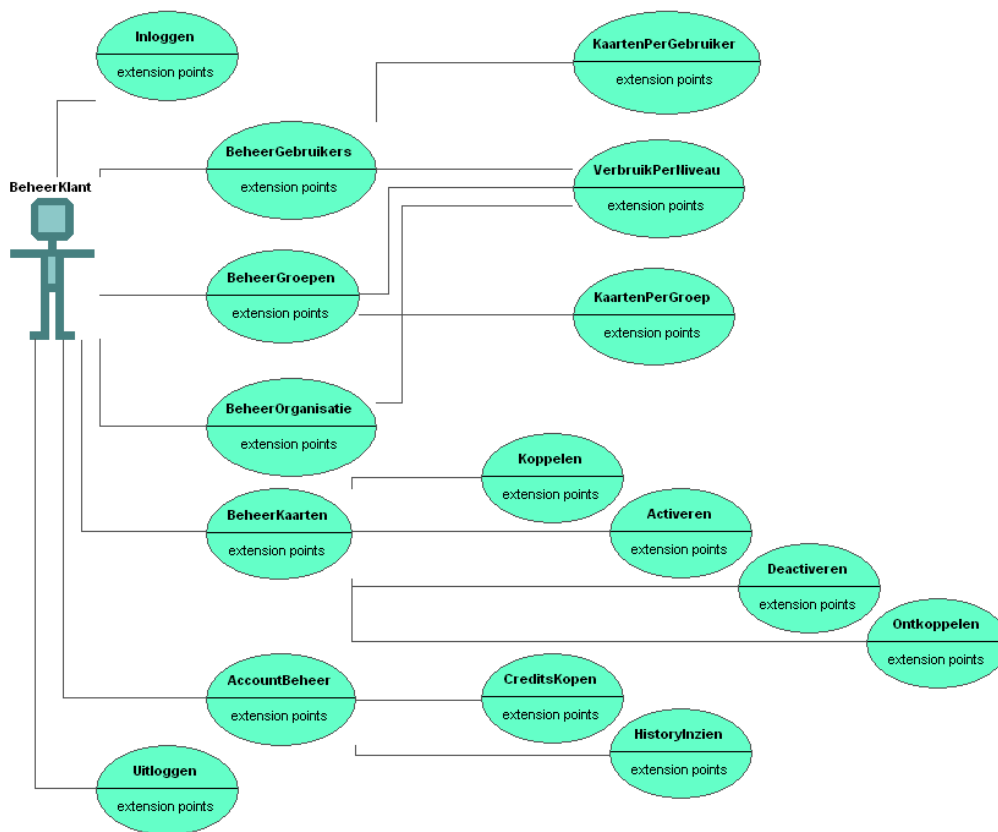
Te wijzigen functionaliteiten

- WMS Opvragen

Te ontwikkelen functionaliteiten

- WFSUitvoeren

5.3 Beheerders van het bedrijf



Al bestaande functionaliteiten

- Inloggen
- Uitloggen

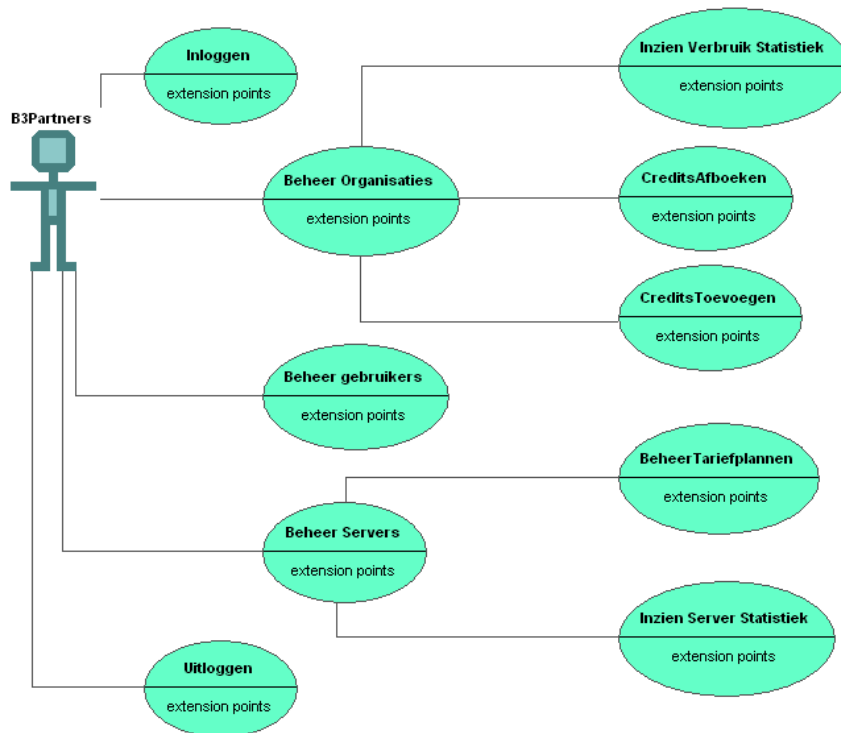
Te wijzigen functionaliteiten

- Beheer Gebruikers
- Beheer Groepen

Te ontwikkelen functionaliteiten

- Beheer Gebruikers
 - KaartenPerGebruiker
 - Verbruik op gebruikersniveau
- Beheer Groepen
 - KaartenPerGroep
 - Verbruik op groepsniveau
- BeheerOrganisatie
 - GlobaleKaarttoegang
- Beheer Kaarten
 - Koppelen
 - Activeren
 - Deactiveren
 - Ontkoppelen
- AccountBeheer
 - CreditsKopen
 - HistoryInzien

5.4 B3Partners beheer



Al bestaande functionaliteiten

- Inloggen
- Uitloggen
- Beheer Gebruikers

Te wijzigen functionaliteiten

- Beheer Organisaties
 - Inzien Verbruik Statistiek
 - Credits Afboeken
 - Credits Toevoegen
- Beheer Servers

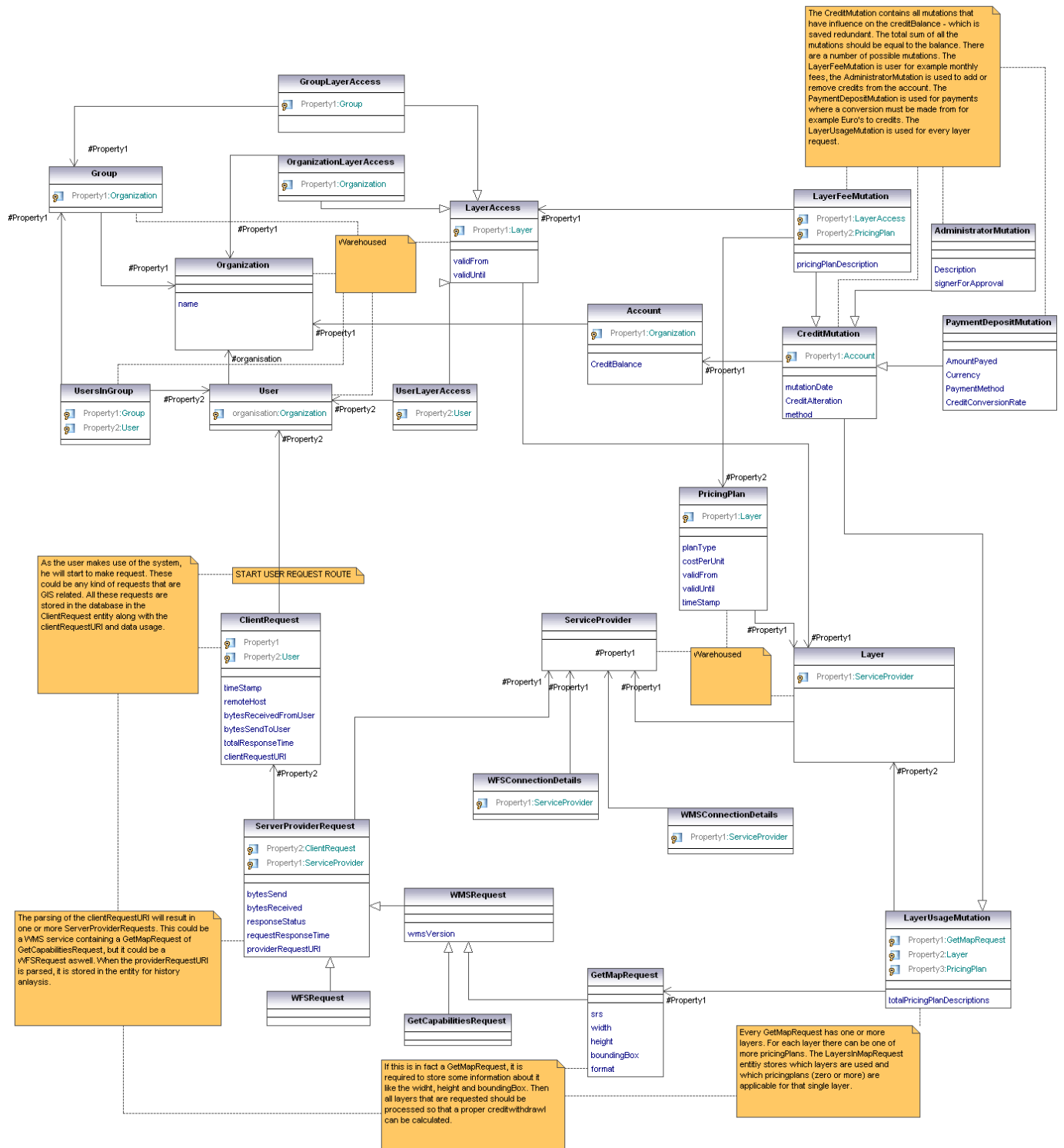
Te ontwikkelen functionaliteiten

- Beheer Servers
 - Beheer Tariefplannen
 - Inzien Server Statistiek

6 Klassendiagram

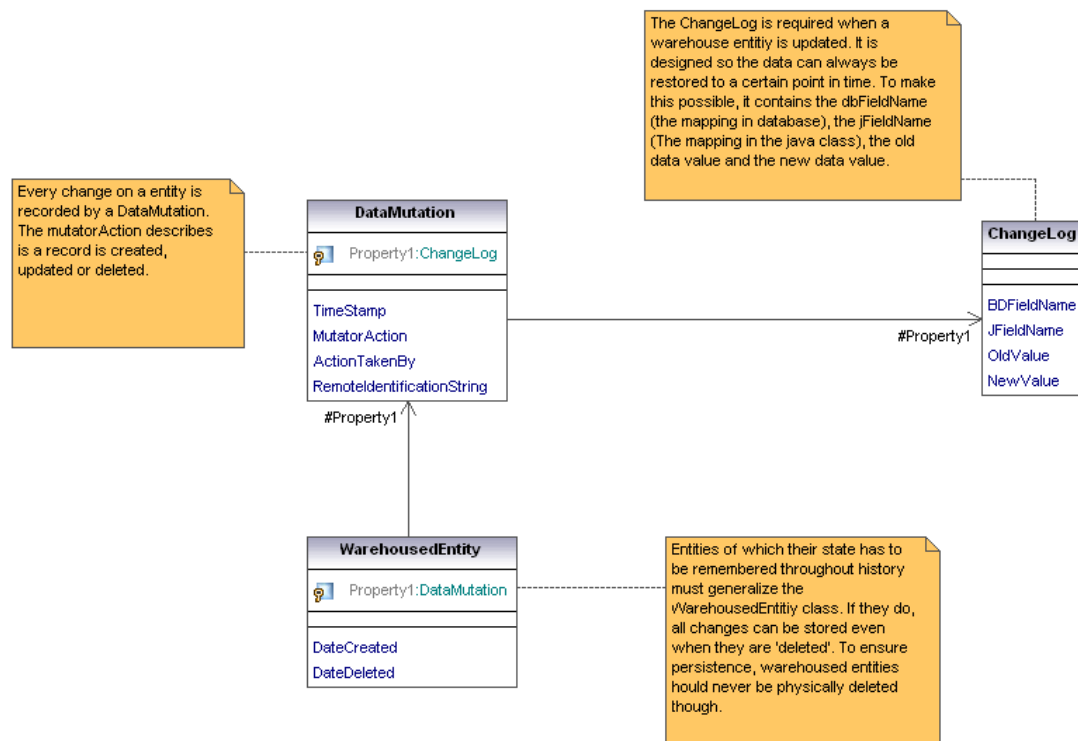
Het volgende domein klassendiagram is opgesteld om als aanzet te dienen voor verdere uitwerking van het functioneel- en technisch-ontwerp. Niet alle klassen van de Kaartenbalie zijn opgenomen, alleen die klassen die relevant zijn voor de uitvoering van dit project zijn toegevoegd.

6.1 Klassen Kaartenbalie



6.2 Klassen Datawarehousing

Voor het project bestaat er de eis dat bepaalde data nooit verloren kan gaan bij het updaten en verwijderen van entiteiten. Hiervoor wordt een soort van Datawarehouse in het leven geroepen waarbij data gezekeerd wordt. Alle handelingen op een entiteit worden opgeslagen in de DataMutation tabel. Voor een update wordt er tevens een ChangeLog gebruikt waarin de waarden de veranderingen worden vastgelegd. Bij een delete wordt een entiteit niet werkelijk weggehaald, maar wordt een entiteit simpelweg op verwijderd gezet. Alle entiteiten die van datawarehousing worden voorzien moeten de WarehousedEntity extenden.



7 Bijlage 1: MoSCoW Analyse

7.1 Timebox I

Timebox I

Periode 08-Oct-07 - 04-Nov-07

Category	SubCategory	UseCase	Type	Description	MoSCoW	Time/hours
Algemeen	Hibernate		Onderzoek	Het leren omgaan met Hibernate en xbm bestanden	Must Have	8
Algemeen	Hibernate		Onderzoek	Hibernate EntityManager vergelijken met de SessionFactory	Could Have	4
ClientRequest						
Registratie			Klassen	Het maken van de klasse voor ClientRequest	Must Have	4
Registratie			Automatiseren	Het registreren van ClientRequests in het systeem	Must Have	16
ServiceProviderRequest I						
Registratie	ServiceRequest		Klassen	Het maken van de klasse voor ServiceProviderRequest	Must Have	4
Registratie	WMS		Klassen	Het maken van de klassen WMSRequest & GetMapRequest	Must Have	4
Registratie	WMS		Automatiseren	Het registreren van ServiceProviderRequest in de vorm van GetMapRequests	Must Have	16
Registratie	WMS		Klassen	Het maken van de klasse voor GetCapabilitiesRequest voor WMS	Wont have	2
Registratie	WMS		Automatiseren	Het registreren van GetCapabilitiesRequests voor WMS	Wont have	4
Registratie	WMS		Model	Het modeleren van GetFeatureInfo voor WMS	Could Have	4
Registratie	WMS		Klassen	Het maken van de klassen voor GetFeatureInfo voor WMS	Could Have	4
Reporting I						
Reporting			Onderzoek	De mogelijkheden van objecten naar XML omzetten bestuderen. Hiervoor zijn diverse standaard mogelijkheden beschikbaar. (Castor of JAXB)	Should Have	16
Reporting			Model	Een XSD opstellen waarmee rapportage mogelijk moet gaan worden.	Should Have	12
Reporting			Automatiseren	De klasse voor ClientRequest , en ServiceProviderRequest exporteerbaar maken naar XML volgens de XSD	Should Have	8
Reporting			Automatiseren	De export verder uitbreiden met de extending klassen van ServiceProviderRequest volgens de XSD	Should Have	6
Reporting			Onderzoek	Experimenteren en testcases maken met XSLT.	Could Have	10

ttM	ttS	ttC	ttW
8			
		4	
4			
16			
4			
4			
16			
			2
			4
		4	
		4	
	16		
	12		
	8		
	6		
		10	
52	42	22	6

7.2 Timebox II

Timebox II

Periode 05-Nov-07 - 02-Dec-07

Category	SubCategory	UseCase	Type	Description	MoSCoW	Time/hours	ttM	ttS	ttC	ttW
Algemeen	Struts		UserInterfaces	Bekend worden met de Struts omgeving van de Kaartenbalie	Must Have	12	12			
Warehousing										
Warehousing			Klassen	Het maken van de klasse voor WarehousedEntity , DataMutation en Changelog .	Must Have	8	8			
Warehousing			Automatiseren	Een framework opzetten waarmee entiteiten die gewarehoused zijn gemakkelijk kunnen worden opgeslagen en bijgewerkt zonder kennis van dat de entiteiten gewarehoused zijn.	Must Have	20	20			
Warehousing			Automatiseren	ServiceProvider en Layer in het warehouse bewaren.	Must Have	4	4			
Warehousing			Automatiseren	PricingPlan in het warehouse bewaren.	Must Have	2	2			
Warehousing			Automatiseren	LayerAccess in het warehouse bewaren.	Must Have	2	2			
Warehousing			Automatiseren	Organisation in het warehouse bewaren.	Should Have	2		2		
Warehousing			Automatiseren	Group , UsersInGroup , User , UserLayerAccess in warehouse bewaren.	Could Have	8			8	
Warehousing		B3Partners	UserInterfaces	Een systeem ontwikkelen waarmee precies kan worden gezien wanneer welk record bewerkt is en wat de status op een bepaald moment was ipv dit met een query te realiseren	Wont have	40				40
Accounting & Pricing I										
Account			Klassen	Het maken van de klassen voor Account en CreditMutation	Must Have	6	6			
Pricing			Klassen	Het maken van de klasse voor PricingPlan	Must Have	4	4			
Mutation			Klassen	Het maken van de klasse voor AdministratorMutation	Must Have	2	2			
Mutation			Klassen	Het maken van de klasse voor PaymentDepositMutation	Should Have	2		2		
Mutation			Klassen	Het maken van de klasse voor LayerFeeMutation	Should Have	2		2		
Pricing		B3Partners	UserInterfaces	Het mogelijk maken om op Layers verschillende prijzen te registreren met verschillende perioden waarbij een afrekening per kaart mogelijk is.	Must Have	20	20			
Pricing		B3Partners	UserInterfaces	Het mogelijk maken om op Layers verschillende prijzen te registreren met verschillende perioden waarbij een afrekening per periode mogelijk is.	Must Have	8	8			
Mutation			Automatiseren	Een soort kleine bank bouwen waarin mutaties kunnen worden verwerkt en waarmee het resterende credit saldo kan worden opgevraagd.	Must Have	16	16			
							104	6	8	40

7.3 Timebox III

Timebox III

Periode	03-Dec-07	-	30-Dec-07
---------	-----------	---	-----------

[illegible]

7.4 Timebox IV

Timebox IV

Periode 31-Dec-07 - 27-Jan-08

Category	SubCategory	UseCase	Type	Description	MoSCoW	Time/hours	ttM	ttS	ttC	ttW
WFS										
Service	WFS		Onderzoek	Onderzoek doen naar de werking van WFS en welke functionaliteiten van toepassing zijn. Testen met client programma's zoals UDIG.	Must Have	16	16			
Service	WFS		Model	Uitzoeken in hoeverre het bestaande datamodel gebruik kan gaan worden voor het bouwen van de WFS dienst.	Must Have	8	8			
Service	WFS		Klassen	Het bestaande datamodel zodanig aanpassen dat WFS bruikbaar gemaakt kan worden. Hierbij valt te denken aan een splitsing van de ServiceProvider tabel.	Must Have	16	16			
Service	WFS		Automatiseren	Een servlet bouwen die er voor zorg dat WFS informatie data wordt doorgestuurd naar de achterliggende servers. Gebruik makend van geringe logica en een URL per serviceprovider logica.	Must Have	24	24			
Service	WFS									
Service	WFS		Onderzoek	Onderzoek doen naar hoeverre het mogelijk is om WFS als dienst zoals WMS aan te bieden, dus weer een URL per gebruiker ipv serviceprovider.	Could Have	32			32	
Service	WFS		Model	Het bouwen van een systeem waar slechts een URL per gebruiker werkzaam is. Hierbij moet aan bijvoorbeeld somatielogica gedacht worden over alle verschillende lagen heen.	Wont have	64				64
ServiceProviderRequest II										
Registratie	WFS		Klassen	Het maken van de klasse WFSRequest	Could Have	4			4	
Registratie	WFS		Automatiseren	Het registreren van WFSRequests in het systeem	Should Have	4		4		
							64	4	36	64

8 Bijlage 1: Gegevensfiltratie

Door middel van een Apache Log worden gegevens verzameld met betrekking tot het verbruik. Hieronder volgt een regel welke al door de logfile parser omgezet is van de apache log naar het csv formaat die vervolgens gebruikt wordt om statistiek mee te genereren.

213.84.161.200,2007-06-19

16:09:38,GET,"/kaartenbalie_mnp/wms/27dba405cc87840e1cc1ffedb5ce5534?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=1065&HEIGHT=867&LAYERS=91_aerogrid2&TRANSPARENT=TRUE&FORMAT=image/png&BBOX=50694.832756632044,366502.49134948105,255538.43137254898,533262.491349481&SRS=EPSG:28992&STYLES=",200,2319949,430,2320254,7176,3419,beheerder,EPSG:28992,1.1.1,GetMap,1065,867,"50694.832756632044,366502.49134948105,255538.43137254898,533262.491349481",image/png

De volgende gegevens worden hier uit gefiltreerd:

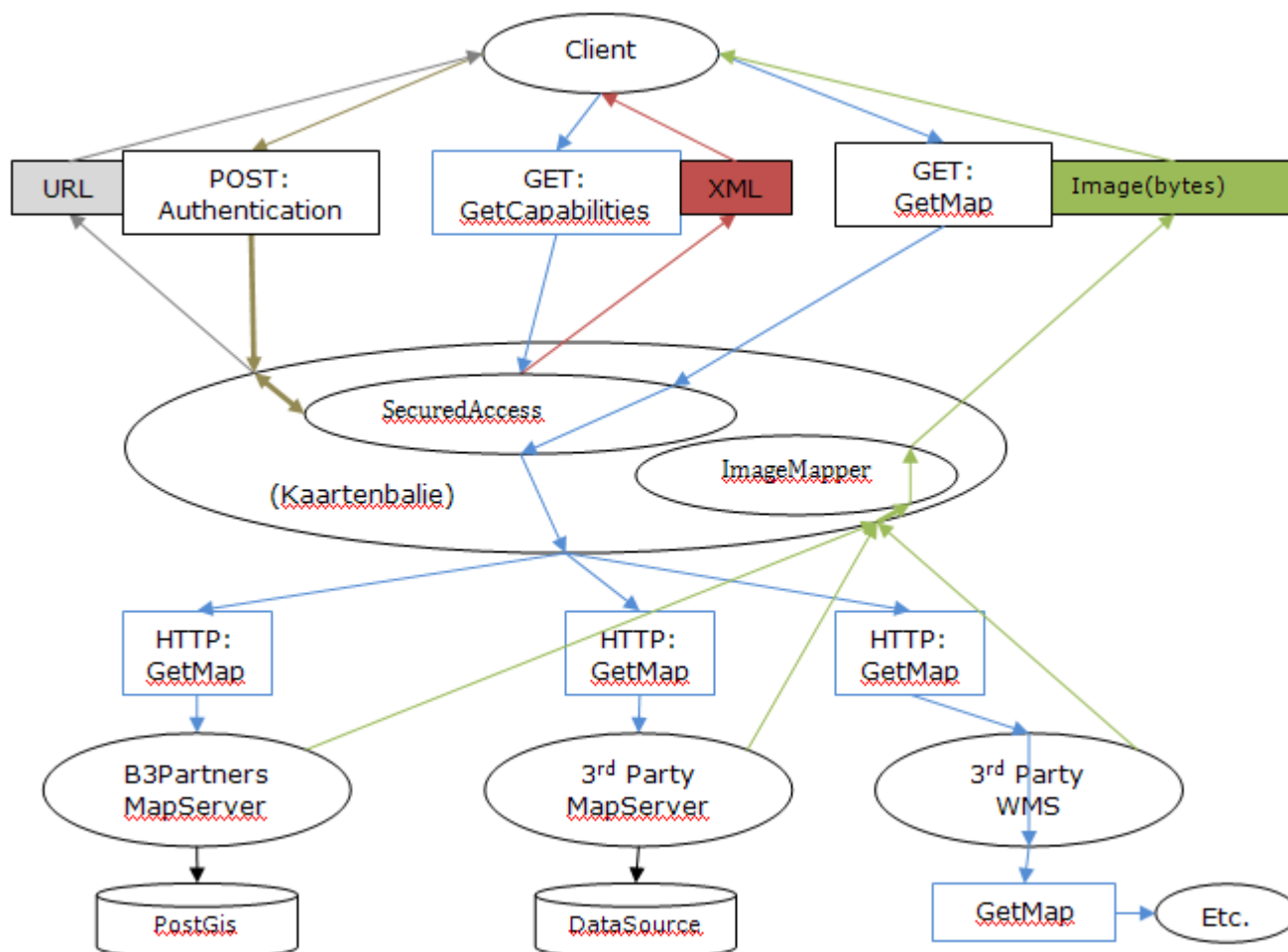
Gegeven	Omschrijving
213.84.161.200	Remote Host
2007-06-19 16:09:38,	Datum/tijd
GET	Method
"/kaartenbalie_mnp/wms/27dba405cc87840e1cc1ffedb5ce5534?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&WIDTH=1065&HEIGHT=867&LAYERS=91_aerogrid2&TRANSPARENT=TRUE&FORMAT=image/png&BBOX=50694.832756632044,366502.49134948105,255538.43137254898,533262.491349481&SRS=EPSG:28992&STYLES="	URI
200	Response Status
2319949	Size of the response in bytes
430	Bytes received
2320254	Bytes sent
7176	The time taken to serve the request, in microseconds
3419	The time it took for the Mapserver to respond in microseconds.
Beheerder	User
EPSG:28992	WMS SRS
1.1.1	WMS Version
GetMap	WMS Request
1065	WMS Width
867	WMS Height
"50694.832756632044,366502.49134948105,255538.43137254898,533262.491349481"	WMS BoundingBox
image/png	WMF Format

9 Bijlage 3: Globale werking kaartenbalie

Hieronder volgt een korte beschrijving van de werking van Kaartenbalie en WMS. WMS, of WFS is een stateless protocol en heeft tot resultaat XML, HTML of grafische data. Voorop staat dat zowel WMS en WFS niet gebaseerd op SOAP zijn en dus ook niet zo te benaderen zijn – er wordt immers gebruik gemaakt van een URL om de parameters mee te geven en het resultaat is niet altijd hetzelfde. Alle standaarden zijn opgesteld door het OpenGIS Consortium.

De kaartenbalie werkt als volgt:

- Een cliënt meldt zich aan bij de Kaartenbalie doormiddel van een login en wachtwoord op de kaartenbalie site.
- De kaartenbalie geeft een unieke URL terug waarmee elke willekeurige WMS viewer contact kan maken met de Kaartenbalie. Dit kan een losstaande viewer zijn, maar ook de geïntegreerde viewer op de Kaartenbalie website.
- De viewer van de cliënt vraagt vervolgens aan Kaartenbalie een GetCapabilities aan door middel van een set van parameters in de URL.
- De Kaartenbalie heeft deze informatie in zijn eigen database staan en verstuurd vervolgens na controle een XML met deze informatie terug. Deze XML is geformatteerd document volgens de standaarden Open Geospatial Consortium.
- De cliënt extraheert informatie uit deze XML. Bijvoorbeeld welke lagen Kaartenbalie aanbiedt.
- Vervolgens vraagt de cliënt doormiddel van nog een set aan parameters in de URL een GetMap aan.
- Kaartenbalie controleert of de cliënt toegang heeft tot de gevraagde gegevens en stuurt het verzoek door naar de bijbehorende MapServers of andere WMS portalen.
- Deze MapServers of Portalen verwerken het verzoek en leveren een plaatje terug.
- Kaartenbalie verwerkt al deze plaatjes tot één plaatje en stuurt dit weer door naar de cliënt.



NB: De kaartenbalie kan zelf worden onderzocht en uitgetprobeerd op <http://www.kaartenbalie.nl/>