

Ontwikkeling van RESTful

Web Services

voor ICIS 4.0

door Wessel van Heerde

IC&S Service & Support B.V.

Begeleiding: B. Van De Garde

Haagse Hogeschool

Examinatoren:

E.M. Van Doorn

S.N. Mimoun

versie 1.0, 28 maart 2008

Afstudeerverslag met betrekking tot het afstuderen bij **IC&S B.V.** van **Wessel van Heerde (20051873)** student op de **Haagse Hogeschool** bij de opleiding **Informatica** afdeling **ICT & Media** in de periode **19 november 2007** tot **28 maart 2008** in het collegejaar **2007/2008**

Referaat

Van Heerde, W., Afstudeerverslag - 'Ontwikkeling van RESTful web services voor ICIS 4.0', Nieuwegein, IC&S Service & Support B.V., maart 2008.

Dit verslag beschrijft het proces rondom de werkzaamheden die in de periode van november 2007 tot maart 2008 in het kader van bovengenoemde afstudeeropdracht zijn uitgevoerd. In de genoemde periode is bovenstaande afstudeeropdracht uitgevoerd door een student aan de opleiding informatica van de Haagse Hogeschool.

De afstudeeropdracht betrof het ontwerpen en ontwikkelen van een tweetal RESTful web services voor de Hyperic monitoring software en de Bacula backup software met daarnaast het ontwikkelen van een web service consumer voor de twee web services. Het doel van de opdracht was een grafische interface te realiseren waarmee klanten en medewerkers van IC&S informatie kunnen inzien over availability, performance en backups van diensten die zij bij IC&S in beheer hebben.

Voorwoord

Het schrijven van een afstudeerverslag is niet makkelijk. Er zijn zoveel aspecten waar rekening mee gehouden dient te worden. Ik ben gelukkig erg goed geholpen waardoor ik terug mag kijken op een prettige afstudeerstage waarin ik veel geleerd heb en alleen nog maar meer zin heb gekregen om het bedrijfsleven binnen te stappen.

Hartelijk dank aan Bart van de Garde, intern begeleider bij IC&S, die me veel vrijheid gaf om mijn opdracht uit te voeren en altijd open stond voor vragen en feedback als dat nodig was. Ook veel dank aan dhr. E.M. van Doorn en mevr. S.N. Mimoun, begeleiders vanuit de Haagse Hogeschool, die altijd snel reageerden op e-mail correspondentie en me vaak konden voorzien van waardevolle feedback. Als laatste wil ik graag mijn collega's bedanken. Zij zorgden elke dag weer voor een onbetaalbaar goede sfeer op kantoor.

Wessel van Heerde, maart 2008

Bruikleen

”De auteur geeft toestemming dit document in zijn geheel of een deel van dit rapport te kopiëren of citeren voor persoonlijk gebruik. De auteur geeft ook toestemming dit document te archiveren op publiekelijk toegankelijke wijze.”

Wessel van Heerde, maart 2008

Samenvatting

Om haar algemene dienstverlening te verbeteren is IC&S in september 2007 begonnen met de ontwikkeling van ICIS 4.0, een web gebaseerd klantenportal. IC&S wil doormiddel van dit nieuwe portal de klanten van IC&S een verbeterd en dieper inzicht geven in hun lopende service requests en de performance van de diensten die klanten bij IC&S in beheer hebben. Klanten hebben kenbaar gemaakt dit wenselijk te vinden.

IC&S wil dit mogelijk maken door het monitoring systeem Hyperic en het backup systeem Bacula doormiddel van een web service te laten consumeren door ICIS 4.0, om zo de informatie die nodig is voor bovenstaand doel beschikbaar te maken voor de klanten van IC&S.

Dit afstudeerverslag beschrijft de ontwikkeling van een tweetal RESTful web services tijdens dit afstuderen, die beide het nieuwe ICIS 4.0 van dienst zullen zijn bij het benaderen van de informatie die nodig is bovenstaand doel te bereiken.

Trefwoorden

IC&S

ICIS

Hyperic

Bacula

Django

REST

RESTful

Web Application Framework

JpGraph

PHP

Python

XML

XPath

Ruby on Rails

Lijst van figuren

2.1	Organogram IC&S Service & Support B.V.	4
5.1	Typen entiteiten binnen Hyperic	16
7.1	Enkele voorbeelden van grafieken die JpGraph produceren kan	23
7.2	Enkele voorbeelden van grafieken die Biggles produceren kan	23
7.3	Enkele voorbeelden van grafieken die Ploticus produceren kan	24
7.4	Enkele voorbeelden van rapporten die JasperReports produceren kan	25
8.1	Architectuur diagram van de Hyperic- en Bacula web service	28
8.2	Architectuur diagram van de grafieken generator tool	29
8.3	Architectuur diagram van de ICIS 4.0 web service cliënt	30
9.1	Hyperic web service use-case diagram	35
9.2	Hyperic web service deployment diagram	38
9.3	Hyperic web service domein klassediagram	39
9.4	Hyperic web service klassediagram	40
9.5	Grafieken generator tool domein klassediagram	42
9.6	Grafieken generator tool klassediagram	43
9.7	Hyperic web service sequentiediagram 1	46
9.8	Hyperic web service sequentiediagram 2	47
9.9	Grafieken generator tool sequentiediagram	48
10.1	Bacula interactie tussen componenten	51
10.2	Bacula componenten architectuur en interactie	52
10.3	Bacula Statistiek ERD	53
11.1	Bacula web service use-case diagram	58
11.2	Bacula web service deployment diagram	59
11.3	Bacula web service domein klassediagram	60
11.4	Bacula web service klassediagram	61
11.5	Bacula web service sequentiediagram 1	64
11.6	Bacula web service sequentiediagram 2	65
13.1	Use-case diagram voor het gebruik van ICIS 4.0 met de Hyperic web service voor medewerkers van IC&S	72
13.2	ICIS 4.0 web service cliënt deployment diagram	74
14.1	Hyperic Paper Report combineert verschillende informatiebronnen tot een rapport	78
16.1	Hyperic web service use-case diagram	88
16.2	Bacula web service use-case diagram	94

16.3	Use-case diagram voor het gebruik van ICIS 4.0 met de Bacula web service voor medewerkers van IC&S	97
16.4	Use-case diagram voor het gebruik van ICIS 4.0 met de Hyperic web service voor klanten van IC&S	98
16.5	Use-case diagram voor het gebruik van ICIS 4.0 met de Bacula web service voor klanten van IC&S	99
16.6	Een schematische voorstelling van het samenkomen van ICIS 4.0, de Hyperic web service en de Bacula web service	99
16.7	Hyperic web service klassediagram	116
16.8	Grafieken generator tool klassediagram	120
16.9	Bacula web service klassediagram	125
16.10	Hyperic web service sequentiediagram 1	130
16.11	Hyperic web service sequentiediagram 2	131
16.12	Grafieken generator tool sequentiediagram	132
16.13	Bacula web service sequentiediagram 1	133
16.14	Bacula web service sequentiediagram 2	134

Lijst van tabellen

4.1	Beknopte planning iteratie 1	9
4.2	Beknopte planning iteratie 2	10
4.3	Beknopte planning iteratie 3	10
4.4	Door IC&S beschikbaar gestelde faciliteiten	11
5.1	Hyperic data retentie perioden	14
6.1	CRUD en corresponderende HTTP methods	18
9.1	Planning inceptie fase, 1e iteratie	32
9.2	Planning elaboration fase, 1e iteratie	33
9.3	Planning construction fase, 1e iteratie	33
9.4	Planning transition fase, 1e iteratie	33
9.5	Use-case Lijst met platformen opvragen	36
9.6	Dictionary van de Platform klasse	41
9.7	Dictionary van de HypericGraph klasse	45
11.1	Planning inceptie fase, 2e iteratie	56
11.2	Planning elaboration fase, 2e iteratie	56
11.3	Planning construction fase, 2e iteratie	56
11.4	Planning transition fase, 2e iteratie	57
11.5	Use-case Backup jobs overzicht opvragen	60
11.6	Dictionary van de JobStatistic klasse	62
13.1	Planning inceptie fase, 3e iteratie	70
13.2	Planning elaboration fase, 3e iteratie	70
13.3	Planning construction fase, 3e iteratie	70
13.4	Planning transition fase, 3e iteratie	71
13.5	Use-case Platform aanmaken	73

Verklarende woordenlijst

availability - Het beschikbaarheidsniveau van een object.

Bacula - Backup software die door IC&S gebruikt wordt.

GUI - Graphical User Interface. De grafische gebruikers interface.

HTTP - Hypertext Transfer Protocol. Een protocol dat over het algemeen wordt gebruikt voor het bezoeken van websites.

Hyperic - Monitoring software die door IC&S gebruikt wordt.

IC&S - IC&S staat voor Internet Consultancy & Support en is de naam van het bedrijf waar de afstudeerstage gelopen is.

ICIS - Klantenportaal dat door IC&S gebruikt wordt. ICIS staat voor IC&S Customer Information System.

ORM - Object Relational Mapping. Een representatie van een database in de vorm van een programmeerbaar object.

outsourcing - Het uitbesteden van werkzaamheden aan een externe partij.

performance - De prestaties van een object.

platform - Object in de Bacula web service, Hyperic web service en ICIS 4.0 web service client dat een fysieke server representeert.

platformservice - Object in Hyperic web service en ICIS 4.0 web service client dat een aan een platform gerelateerde service representeert. Voorbeelden van platformservices zijn fileserver mount en networkserver interface.

resource - Afhankelijk van de context wordt hiermee bedoeld op beschikbare bronnen.

REST - Representational State Transfer. Een verzameling principes.

RESTful - Een web service die voldoet aan de principes van REST.

RFC - Request For Change. Een RFC wordt door een klant ingediend op het moment dat de wens tot een wijziging bestaat.

RoR - Ruby on Rails. RoR is een in Ruby geschreven framework voor het ontwikkelen van web applicaties.

RPC - Remote Procedure Call. Het op afstand aanroepen van procedures. XML-RPC is een implementatie van RPC.

server - Afhankelijk van de context wordt hiermee bedoeld op een fysiek platform of een Hyperic server zoals bijvoorbeeld Apache, PostgreSQL of Postfix.

service - Object in Hyperic web service en ICIS 4.0 web service client dat een aan een server gerelateerde service representeert. Voorbeelden van services zijn Apache VHost, Postfix Queue of PostgreSQL database.

SLA - Service Level Agreements. In een SLA worden afspraken over het te leveren service niveau vastgelegd.

SMTP - Simple Mail Transfer Protocol. Een protocol dat wordt gebruikt voor het verzenden van e-mail. Hyperic is in staat SMTP servers te monitoren.

UML - Unified Modeling Language. Een methode om op een uniforme manier systeemmodellen te specificeren.

UP - Unified Processing. Een projectbeheersingsmethode die tijdens dit afstuderen toegepast is.

XML - Extented Markup Language. Een standaard waarmee gegevens op een gestructureerde manier vastgelegd kunnen worden.

Inhoudsopgave

Referaat	iii
Voorwoord	v
Bruikleen	vii
Samenvatting	ix
Trefwoorden	xi
Verklarende woordenlijst	xvii
1 Inleiding	1
2 IC&S	3
2.1 Bedrijfsprofiel	3
2.2 Positie binnen de organisatie	3
3 Probleembeschrijving	5
3.1 Doelstellingen	6
4 Werkwijze	7
4.1 Methoden en technieken	8
4.2 Planning en deliverables	9
4.3 Faciliteiten	11
4.4 Risicofactoren	12
5 Hyperic	13
5.1 Availability en performance	13
5.2 Hyperic HQ	14
5.3 Hyperic Agent	14
5.4 Metingen	14
5.5 Platform, server en service	15
6 Informatie benaderen met RESTful web services	17
6.1 Doelstelling	17
6.2 RESTful web services	17
6.3 Productselectie	18
6.4 Conclusie	20

7	Visualisatie van meetgegevens	21
7.1	Doelstelling	21
7.2	Aanpak	21
7.3	Productselectie	22
7.4	Conclusie	25
8	Architectuur	27
8.1	Web services	27
8.2	Grafieken generator tool	28
8.3	ICIS 4.0 web service cliënt	29
9	Hyperic web service	31
9.1	Requirements	31
9.2	Planning	32
9.3	Use-cases	34
9.4	Toegevoegde systeemeisen	36
9.5	Deployment	37
9.6	Klassediagrammen	37
9.7	Sequentiediagrammen	46
10	Bacula: open-source multi platform network backup systeem	49
10.1	3 componenten	49
10.2	Console	50
10.3	Catalog	50
10.4	Bacula Statistiek: openheid van (backup)zaken	50
11	Bacula web service	55
11.1	Requirements	55
11.2	Planning	56
11.3	Use-cases	58
11.4	Deployment	58
11.5	Klassediagrammen	60
11.6	Sequentiediagrammen	63
12	Transition van web services	67
12.1	Testen	67
12.2	Handleiding	67
12.3	Overdracht	68
13	ICIS 4.0 web service cliënt	69
13.1	Requirements	69
13.2	Planning	70
13.3	Use-cases	71
13.4	Deployment	71
13.5	Overdracht	75
14	Dynamisch rapporteren met L^AT_EX	77
14.1	L ^A T _E X	77
14.2	Hyperic Paper Report	77

15 Evaluatie	81
15.1 Proces evaluatie	81
15.2 Product evaluatie	82
16 Conclusie	85
Appendix A: Use-cases	87
16.1 Iteratie 1: Hyperic web service	88
16.2 Iteratie 2: Bacula web service	94
16.3 Iteratie 3: ICIS 4.0 web service cliënt	97
Appendix B: Klassediagrammen	115
16.4 Iteratie 1: Hyperic web service	116
16.5 Iteratie 2: Bacula web service	125
Appendix C: Sequentiediagrammen	129
16.6 Iteratie 1: Hyperic web service	130
16.7 Iteratie 2: Bacula web service	133
Appendix E: Referenties	135

Hoofdstuk 1

Inleiding

Dit verslag is het afstudeerverslag van Wessel van Heerde en beschrijft de ontwikkeling van een tweetal web services tijdens de afstudeerstage bij IC&S B.V. te Nieuwegein. Dit verslag is bedoeld voor de examinatoren, dhr. E.M. van Doorn en mevrouw S.N. Mimoun en de gecommitteerde.

De structuur van dit verslag is bepaald aan de hand van de iteraties die tijdens het afstuderen zijn doorlopen en maakt dat de inhoud ook chronologisch is weergegeven. Hoofdstuk 2 geeft een beschrijving van IC&S en mijn positie binnen IC&S weer. Hoofdstuk 3 geeft een omschrijving van het probleem weer en definieert de te behalen doelstellingen. Het daaropvolgende hoofdstuk beschrijft de werkwijze, waarin staat uitgelegd hoe het probleem aangepakt en behapbaar gemaakt wordt.

Hoofdstuk 5 geeft een inleiding op Hyperic, de monitoring software die door IC&S gebruikt wordt. Hoofdstuk 6 vervolgens geeft een inleiding op RESTful web services en doet een vergelijkend warenonderzoek tussen verschillende producten die ondersteuning bieden voor de implementatie van RESTful web services. Hoofdstuk 7 doet een vergelijkend warenonderzoek tussen verschillende graphing- en/of reporting engines. Het daaropvolgende hoofdstuk 8 beschrijft architectuur van de drie ontwikkelde onderdelen.

Hoofdstuk 9 beschrijft het ontwikkelproces van de Hyperic web service. Hoofdstuk 10 geeft een inleiding op Bacula, de backup software die IC&S gebruikt en hoofdstuk 11 beschrijft daarna de ontwikkeling van de Bacula web service. Hoofdstuk 11 beschrijft de ontwikkeling van de web service client voor de ICIS 4.0 klantenportal. Het daaropvolgende hoofdstuk, Hoofdstuk 12 beschrijft de transition voor de web services, de overgangsfase die aan het einde van elke iteratie voor komt. Het daaropvolgende hoofdstuk 13 beschrijft het ontwikkelproces van de ICIS 4.0 web service client en hoofdstuk 14 beschrijft een toepassing van de grafieken generator tool en laat daarmee een productieve toepassing zien van een van de opgeleverde producten.

Hoofdstuk 15 evalueert het proces en het opgeleverde product. Hoofdstuk 16 geeft een conclusie weer over het resultaat van de afstudeerstage.

Hoofdstuk 2

IC&S

2.1 Bedrijfsprofiel

IC&S is een bedrijf met zo'n 20 medewerkers, gevestigd in Nieuwegein en sinds 1999 actief in het aanbieden van service en support, development en consultancy diensten op gebied van hosting en netwerk infrastructures. Het huidige management begon toen met het aanbieden van kleinschalige hostingdiensten aan het midden en kleinbedrijf.

Service en support is het bedrijfsonderdeel waar de afstudeeropdracht zal worden uitgevoerd. Dagelijkse werkzaamheden bestaan voornamelijk uit te uitvoeren van RFC's, het monitoren van servers en diensten en het oplossen van incidenten, problemen en calamiteiten. Development is het bedrijfsonderdeel waar verschillende open-source software pakketten worden ontwikkeld en consultancy is het onderdeel dat klanten adviseert op het gebied van hardware en software oplossingen.

In de loop der jaren is IC&S gegroeid. Enkele websites die door IC&S gehost en/of beheerd worden zijn www.nu.nl, www.autoweek.nl en www.videotalent.nl.

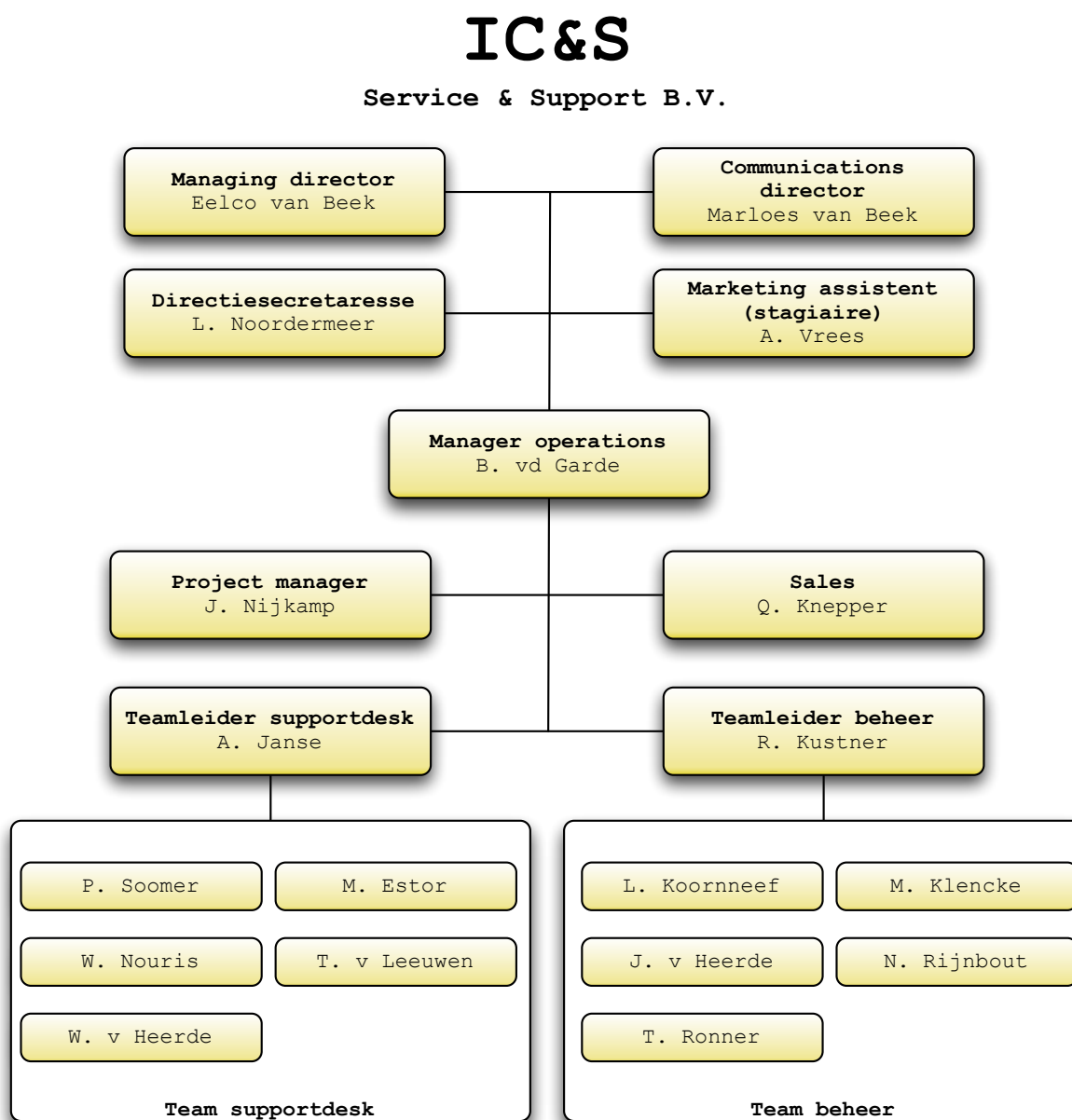
2.2 Positie binnen de organisatie

Voordat ik begon met deze afstudeerstage was ik al enkele jaren part-time werkzaam bij IC&S. Mijn werkzaamheden bestaan, mede vanwege de part-time aanwezigheid, voornamelijk uit het behandelen van 1e lijns support werkzaamheden, omdat de beperkt beschikbare tijd weinig tot geen mogelijkheid biedt bezig te zijn met projecten. Daarnaast beheer ik de backup omgeving, werk ik mee in het langlopende Hyperic project en doe ik ontwikkelwerkzaamheden aan interne applicaties.

De sfeer in het bedrijf is informeel. Als werknemer ben je vrij om de kleding te dragen die je wilt en tussen de werknemers is er weinig te zien van de verschillen in functie. Tijdens de afstudeerstage heb ik mijn bestaande werkplek behouden, maar heb ik de support werkzaamheden naast me neer gelegd en ben ik me gaan richten op de afstudeeropdracht.

2.2.1 Organogram

Figuur 2.1 toont het organogram van IC&S Service & Support B.V. Hieruit komt naar voren dat het bedrijf uit een tweetal teams, supportdesk en beheer, bestaat die elk een eigen teamleider hebben. De teamleider legt daarbij verantwoording af aan de Manager Operations die over de dagelijkse gang van zaken gaat.



Figuur 2.1: Organogram IC&S Service & Support B.V.

Hoofdstuk 3

Probleembeschrijving

IC&S heeft een aantal verschillende klanten in haar klantenbestand. Elk van deze klanten biedt weer verschillende diensten aan, zoals een vermogensbeheer applicatie, een informatieve website (www.autoweek.nl), een applicatie die je helpt bij het afvallen of een applicatie die tussenpersonen in het verzekeringswezen ondersteunt. Elk van deze diensten heeft zijn eigen omvang en verbruikt zijn eigen aandeel aan servers en resources.

Het is de verantwoordelijkheid van IC&S om al deze diensten draaiende te houden en te voldoen aan de eisen zoals deze in de afgesproken SLA zijn vastgelegd. Onderdeel van de SLA is dat er dagelijks backups worden gemaakt en dat de klant gefactureerd wordt voor het aantal gebruikte gigabytes.

Om er zeker van te zijn dat alle diensten die de klant aanbiedt draaien, worden de diensten periodiek op hun werking getest. Dit gebeurt niet handmatig, want dat zou veel te veel werk zijn. Hiervoor is zogenaamde monitoring software aanwezig met de naam Hyperic. Deze software is in staat om met een vaste regelmaat te controleren of een dienst nog beschikbaar (available) is en dat de servers waarop de dienst draait wel voldoende resources aanbieden (performance).

Om de klanten te factureren voor het aantal gigabytes die hun backups innemen, is er een web applicatie ontwikkeld voor intern gebruik die grafieken genereert waaruit het aantal gebruikte gigabytes op een bepaalde datum is af te lezen. Bij Hyperic, het monitoring pakket, was al een web applicatie aanwezig die grafieken toonde van de verzamelde availability en performance gegevens. Ook deze laatste web applicatie is alleen geschikt voor intern gebruik.

Klanten willen graag weten waar ze aan toe zijn, omdat ze op die manier kunnen inspelen op de toekomst. Zo kunnen ze aan de hand van het aantal gebruikte gigabytes in hun backup begrotingen maken, meer budget aanvragen voor het komende jaar of hun systemen opruimen om op die manier minder gigabytes te backupperen en dus lagere backup kosten te hebben. Hetzelfde geldt voor availability en performance grafieken en maakt het om die reden een hot item bij IT managers.

Omdat inzichtelijkheid in performance en availability zo'n hot item is, wordt IC&S steeds vaker gevraagd naar maandelijkse rapportages over backup, availability en performance en worden er steeds vaker afspraken over gemaakt in SLA's. Om aan die afspraken tegemoet te komen is IC&S enkele maanden geleden begonnen met het project ICIS 4.0, een software ontwikkeltraject dat direct geoutsourced is, omdat er veel facetten bij komen kijken die niet tot de core-business van IC&S behoren, zoals o.a. het ontwerpen van een GUI. ICIS 4.0 is een klantenportal in

web applicatie vorm die de gevraagde informatie uit de web applicatie voor backup en de web applicatie behorende bij Hyperic, voor medewerkers van IC&S en de klanten van IC&S samen moet gaan brengen.

3.1 Doelstellingen

De volgende doelstellingen zijn door de opdrachtgever gedefinieerd:

- *Aan het eind van de afstudeerperiode zullen klanten en medewerkers van IC&S met ICIS 4.0 de grootte van uitgevoerde backup jobs en de totaal verbruikte ruimte kunnen inzien. Deze informatie zal op een per server basis en op basis van datum kunnen worden gefilterd en zal bestaan uit één of meerdere grafieken en een statistisch overzicht.*
- *Aan het eind van de afstudeerperiode zullen klanten en medewerkers van IC&S met ICIS 4.0 één of meerdere grafieken kunnen inzien met informatie over availability en performance over door Hyperic gemonitorde diensten, zoals o.a. in de eerder beschreven voorbeelden genoemd worden.*

Door bovenstaande doelstellingen te voltooien wil IC&S het eerder beschreven probleem oplossen. De bestaande backup web applicatie en de bestaande Hyperic web applicatie zullen worden gekoppeld aan het in ontwikkeling zijnde ICIS 4.0.

Door IC&S is er voor gekozen een oplossing voor dit probleem te bieden doormiddel van web services die op REST gebaseerd zijn. REST is een verzameling begrippen die worden toegepast op een web service. Samen met het HTTP protocol wordt dit gebruikt om stateless client / server communicatie mogelijk te maken.

Hoofdstuk 4

Werkwijze

Het probleem dat in hoofdstuk 3 beschreven wordt is op te delen in een drietal verschillende onderdelen op basis van de bestaande producten zoals deze in de probleembeschrijving benoemd zijn en de keuze van IC&S om het algehele probleem op te lossen door middel van op REST gebaseerde web services.

1. Voor Hyperic, de monitoring software, zal een web service ontwikkeld worden die informatie kan geven over availability en performance.
2. Voor Bacula, de backup software, zal een web service ontwikkeld worden die informatie kan geven over uitgevoerde backup jobs met daaraan gekoppeld een weergave van aanwezige trends met betrekking tot in gebruik zijnde harddisk ruimte en grootte van backup jobs.
3. Voor ICIS 4.0, het klantenportal, zal een grafische interface worden ontwikkeld die gebruikers van het klantenportal in staat stelt informatie op te vragen over hun performance, availability en backups. Achter de schermen zal de grafische interface dan gebruik maken van de ontwikkelde web services voor Hyperic en Bacula.

Elk van de hierbovengenoemde onderdelen zal een losstaand product zijn. Het combineren van onderdelen is niet mogelijk omdat elk van de onderdelen zo verschillend is dat een combinatie (bijvoorbeeld een enkele gecombineerde web service voor Hyperic en Bacula) in zowel in ontwikkel- en beheer opzicht onnodig complex zou worden. Ook is IC&S van plan elk van de opgeleverde producten een lange tijd te gebruiken en voortdurend door te ontwikkelen.

Het bovenstaande in ogenschouw genomen is het toepasselijk om gebruik te maken van UP als projectbeheersingsmethode. Het gebruik van UP is niet door IC&S opgelegd (ik ben vrij te kiezen welke methode ik toepas), maar is toepasselijk omdat:

- IC&S heeft aangegeven de ontwikkelde producten lange tijd te willen gebruiken en voortdurend door te ontwikkelen. UP's iteratieve benadering en ondersteuning voor langlopende projecten is hier geschikt voor.
- Elk van de genoemde onderdelen is in een iteratie in te passen, op die manier worden drie iteraties gecreeerd, op te delen over 15 weken. Dat maakt de opdracht te opzichte van de beschikbare tijd voor deze afstudeeropdracht behapbaar en realistisch.
- De persoonlijke gebruikerservaring met UP is, in tegenstelling tot andere projectbeheersingsmethoden, goed.

4.1 Methoden en technieken

Binnen dit project is gebruik gemaakt van verschillende methoden en technieken. Elke gebruikte methode en daar eventueel bijbehorende technieken en motiveringen zullen apart worden beschreven.

UML is een verzameling van technieken waarmee een abstract model van een systeem kan worden gemodelleerd. Het gebruik van UML is niet verplicht gesteld door IC&S, maar wordt wel algemeen gebruikt voor het modelleren van systemen en biedt, zoals de naam ook stelt, een uniforme methode om systemen te modelleren. Dit zorgt ervoor dat bij een eventuele overdracht dezelfde taal wordt gesproken.

- Het **use-case diagram** is een UML techniek waar gebruiks cases in kunnen worden gemodelleerd. Use-cases zijn bepaald aan de hand van de gestelde requirements.
- Het **klassendiagram** is een UML techniek om model te geven aan de verschillende klassen die de web service zal bevatten. Het klassediagram is bepaald op basis van de use-cases en de data structuren die achter Hyperic, Bacula en ICIS 4.0 zullen zitten.
- Het **sequentiediagram** is een UML techniek die op een duidelijk manier de interactie tussen verschillende objecten weergeeft. Het sequentiediagram is gemaakt aan de hand van het klassediagram.
- Het **deploymentdiagram** is een UML techniek die gebruikt is om de hardware en software weer te geven waar het te implementeren systeem op zal gaan draaien.

XML is een standaard waarmee gegevens op een gestructureerde manier vastgelegd kunnen worden. XML zal toegepast worden in de respons die de web services zullen geven. XML is een veelgebruikte- en breed ondersteunde methode om gegevens over netwerken te sturen en vormt daarmee een geschikte methode om communicatie tussen web service en client plaats te laten vinden.

HTTP is het protocol dat toegepast zal worden om requests te doen naar een web service. HTTP is een veel gebruikt protocol voor deze doeleinden. Met het gebruik van HTTP als communicatieprotocol tussen web service en client wordt vastgehouden aan een standaard, iets dat essentieel is voor eventuele latere uitbreidingen van het product.

Ruby on Rails (RoR) is een framework voor de Ruby programmeertaal dat compleet geoptimaliseerd is voor het ontwikkelen van web applicaties. ICIS 4.0 wordt in RoR ontwikkeld en ook onderdeel 3 zal in RoR worden ontwikkeld om volledig met ICIS 4.0 te kunnen integreren. Het gebruik van RoR voor ICIS 4.0 is een keuze van IC&S geweest.

ORM is een van de grote bouwstenen van RoR, waar het geïmplementeerd wordt door het zogenaamde ActiveRecord. Ook in het later beschreven Django zal ORM gebruikt worden. ORM is een onderdeel van een product en is dus geen los gekozen techniek. Er zit dus geen losse motivatie aan vast.

REST staat voor Representational State Transfer en is een verzameling architectuur beschrijvende principes die toegepast kunnen worden op een web service. Op het moment dat een web service volledig voldoet aan de principes die REST onderschrijft mag de web service volledig RESTful worden genoemd. In veel gevallen is het zo dat een web service ook RESTful

wordt genoemd terwijl het eigenlijk een mix is van een web service die op de REST principes is gebaseerd en web service die op de XML-RPC techniek is gebaseerd. In zo'n geval is een web service naar officiële maatstaven niet RESTful maar een zogenaamde REST-RPC hybrid.

4.2 Planning en deliverables

In elk van de drie benoemde iteraties zullen de fases verwerkt worden die het UP proces beschrijft. Daarnaast heeft elke fase enkele deliverables. Hieronder een beknopte planning van de uitgevoerde werkzaamheden en de bijbehorende deliverables.

4.2.1 Iteratie 1: Hyperic web service

Iteratie 1		
Fase	Startdatum	Einddatum
Inception fase	19-11-2007	28-11-2007
Elaboration fase	29-11-2007	03-12-2007
Construction fase	04-12-2007	14-12-2007
Transition fase	17-12-2007	21-12-2007

Tabel 4.1: Beknopte planning iteratie 1

Deliverables

De volgende documentatie zal worden opgeleverd:

1. Inception rapport
2. Elaboration rapport
3. Construction rapport
4. Transition rapport
5. Handleiding voor de gebruiker
6. Handleiding voor de ontwikkelaar

Broncode van de volgende applicaties zal worden opgeleverd:

1. Hyperic web service
2. Grafieken generator tool

De volgende dienst zal worden opgeleverd:

1. Aanspreekbare Hyperic web service

De grafieken generator tool zal niet als aparte dienst worden opgeleverd, maar als losse software tool.

4.2.2 Iteratie 2: Bacula web service

Iteratie 2		
Fase	Startdatum	Einddatum
Inception fase	24-12-2007	28-12-2007
Elaboration fase	31-12-2007	04-01-2008
Construction fase	07-01-2008	18-01-2008
Transition fase	21-01-2008	25-01-2008

Tabel 4.2: Beknopte planning iteratie 2

Deliverables

De volgende documentatie zal worden opgeleverd:

1. Inception rapport
2. Elaboration rapport
3. Construction rapport
4. Transition rapport
5. Handleiding voor de gebruiker
6. Handleiding voor de ontwikkelaar

Broncode van de volgende applicaties zal worden opgeleverd:

1. Hyperic web service

De volgende dienst zal worden opgeleverd:

1. Aanspreekbare Bacula web service

4.2.3 Iteratie 3: ICIS 4.0 web service client

Iteratie 3		
Fase	Startdatum	Einddatum
Inception fase	28-01-2008	01-02-2008
Elaboration fase	04-02-2008	08-02-2008
Construction fase	11-02-2008	22-02-2008
Transition fase	25-02-2008	29-02-2008

Tabel 4.3: Beknopte planning iteratie 3

In bovenstaande planning komt naar voren dat voor iedere iteratie 5 weken is geplanned. Voor deze gelijke verdeling heb ik gekozen omdat het vooraf onduidelijk was of er onderdelen waren die meer ontwikkeltijd nodig hadden op basis van complexiteit.

4.3 Faciliteiten

Voor de uitvoer van het project heeft IC&S de volgende faciliteiten beschikbaar gesteld:

Faciliteit	Omschrijving
Hyperic HQ applicatie server	De Hyperic HQ applicatie server omvat toegang tot de Hyperic web applicatie en shell toegang tot de fysieke server waarop Hyperic HQ geïnstalleerd en operationeel is.
Hyperic HQ database server	De Hyperic HQ database server omvat toegang tot de database Hyperic HQ zijn o.a. zijn meetgegevens in opslaat en shell toegang tot de fysieke server waarop de Hyperic HQ database draait.
Bacula Director applicatie server	De Bacula Director applicatie server redigeert alle backup jobs en herbergt daarnaast de bestaande en in gebruik zijnde Bacula statistieken applicatie. Ook de database achter de Bacula statistieken applicatie is hier geïnstalleerd.
Bacula Storage Daemon applicatie server	De Bacula Storage Daemon applicatie server verzorgt het verkeer tussen de Bacula File Daemons die op de te backupper machines staat en slaat alle backups op.
Bacula Statistiek applicatie/database server	De Bacula Statistiek database server geeft toegang tot zowel de Bacula Statistiek database als de Bacula Catalog. De Bacula Statistiek applicatie, die fysiek op dezelfde server draait omvat toegang tot de Bacula Statistiek web applicatie.
ICIS 4.0 Development omgeving	De ICIS 4.0 Development omgeving omvat toegang tot de development versie van de ICIS 4.0 klantenportal en shell toegang tot de fysieke server waar de development omgeving op geïnstalleerd is.
ICIS 4.0 Broncode	De ICIS 4.0 Broncode omvat toegang tot de Subversion repository waar de broncode van ICIS 4.0 staat opgeslagen.
TextMate	TextMate is een tekst editor voor OSX met ondersteuning voor projecten en syntax highlighting voor verschillende programmeertalen. IC&S stelt een licentie beschikbaar voor TextMate.
Basecamp	Basecamp is het systeem dat IC&S gebruikt om haar projecten te beheren. Het biedt een web accessible, collaboratief platform met de mogelijkheid om o.a. ToDo lijsten maken en mijlpalen te stellen. Voor het afstuderen is een apart project binnen Basecamp aangemaakt, waarvan gebruik gemaakt dient te worden.
Subversion	Subversion is het versiebeheersysteem voor software dat IC&S gebruikt. Alle software die voor of door IC&S ontwikkeld wordt, wordt opgeslagen in een eigen Subversion repository.
Mantis	Mantis is de bugtracker die IC&S gebruikt om fouten in door IC&S beheerde broncode te registreren.

Tabel 4.4: Door IC&S beschikbaar gestelde faciliteiten

4.4 Risicofactoren

Bij elk project zijn factoren aanwezig die de projectvoortgang op enige wijze negatief kunnen beïnvloeden. Door faciliteiten als Basecamp, Subversion en een intranet Samba share beschikbaar te stellen, wordt in feite al redundantie aangeboden als het gaat om projectmanagement, broncode management en backups van laptops die door medewerkers worden gebruikt. De meeste risico factoren worden daardoor weggenomen. De volgende factoren kunnen nog wel een risico vormen voor de voortgang van het project:

- **Kennisgebrek** - Over enkele onderwerpen die tijdens dit project aan bod komen, heeft de afstudeerder beperkte of geen kennis. Hoewel dit doorgaans een uitdaging is, kan dit ook uitlopen op een vertraging. Vertraging door een risicofactor als deze is niet te voorkomen. De vertraging kan wel zoveel mogelijk ingekort worden door tijdig hulp te vragen.
- **Vertraging van afhankelijke software** - De laatste iteratie, de ontwikkeling van de ICIS 4.0 web service client interface, is afhankelijk van ICIS 4.0 omdat de interface uiteindelijk daarmee moet integreren. ICIS 4.0 wordt ontwikkeld in Argentinië en heeft enige vertraging opgelopen. Op het moment dat er geen versie van ICIS 4.0 beschikbaar is waar op gebouwd kan worden om de ICIS 4.0 web service client interface te ontwikkelen, kan dat vertraging voor het project opleveren. Het ontstaan van vertraging kan zoveel mogelijk worden ingeperkt door eventueel door te bouwen op basis van een eerdere build van ICIS 4.0 dan eigenlijk wenselijk zou zijn.

Hoofdstuk 5

Hyperic

Essentieel onderdeel van het server beheer is het monitoren van de servers en diensten die IC&S voor haar klanten in beheer heeft. Zonder monitoring zouden incidenten en problemen minder snel herkend worden terwijl het in het belang van zowel de klant als IC&S is dat die herkenning zo snel mogelijk plaats vind. Zonder herkenning van het incident kan er ook geen actie op worden ondernomen.

Het monitoren van de vele servers en diensten die IC&S beheert is geen handwerk, dat zou veel te arbeidsintensief zijn. Daarom is er door IC&S software in gebruik genomen die dat werk uit handen neemt en op systematische manier alle servers en diensten die IC&S in beheer heeft controleert op een juiste werking. De controle op juiste werking kan op verschillende manieren uitgevoerd worden, zo kan de bereikbaarheid van een server bijvoorbeeld worden getest door de server te pingen en kan de bereikbaarheid van een SMTP server worden getest door een connectie te maken op poort 25 (SMTP) van de betreffende server.

De huidige software (Nagios)¹ is niet meer toereikend omdat het eenodeloos ingewikkelde configuratie vereist en de ontwikkeling niet erg snel gaat. Daarbij wordt er door IC&S medewerkers te makkelijk vergeten bepaalde diensten of servers in te configureren zodat deze ook gemonitord kunnen worden en zijn de controles die door de monitoring worden uitgevoerd niet intelligent genoeg (bij een SMTP server wordt bijvoorbeeld alleen verbinding gemaakt op poort 25, terwijl bijvoorbeeld een HELO command uitgevoerd zou kunnen worden om te controleren of de SMTP server ook goed functioneert).

Om bovenstaande redenen heeft IC&S er voor gekozen de huidige monitoring software te vervangen door Hyperic: client-server monitoring software die een stuk meer in huis heeft dan de huidige software. Dit hoofdstuk geeft een inleiding op de Hyperic web service.

5.1 Availability en performance

Availability en performance monitoring zijn twee aspecten die altijd komen kijken in de monitoring wereld. Availability monitoring is de controle of een dienst of server beschikbaar is, performance monitoring is de controle op de prestaties van een server of dienst. De huidige monitoring software is niet in staat performance monitoring toe te passen op de servers en diensten die het monitort en dus was alleen availability monitoring een optie. De performance monitoring die Hyperic IC&S nu extra biedt wordt gebruikt om de prestaties van verschillen-

¹<http://www.nagios.org>

de servers en diensten te visualiseren en op papier te zetten. Op basis van die performance metingen kunnen aanbevelingen worden gedaan aan klanten.

5.2 Hyperic HQ

Hyperic HQ is het server gedeelte van Hyperic en is hoofdzakelijk verantwoordelijk voor het inroosteren en verwerken van de availability en performance checks (manier om te availability of performance van een gemonitort item op te halen), auto-inventory (automatische detectie van diensten die de server draait) en het verzenden van alerts (als een availability of performance meting bepaalde overtredingen begaat, dan wordt een medewerker van IC&S daarvan op de hoogte gesteld.) Daarnaast biedt Hyperic HQ een web interface die grafieken genereert van de servers en diensten die het monitort.

5.3 Hyperic Agent

Hyperic Agent is het client gedeelte van Hyperic en ontvangt opdrachten van Hyperic HQ. Die opdracht kan zijn dat bepaalde diensten moeten worden gecontroleerd op availability en/of performance of dat er een nieuwe auto-inventory scan moeten worden uitgevoerd. Het resultaat van de opdracht die de Hyperic Agent ontvangt wordt teruggestuurd naar de Hyperic HQ die het verder afhandeld.

5.4 Metingen

De resultaten die Hyperic HQ van de Hyperic Agent terug ontvangt worden door Hyperic HQ verwerkt in een meting en bewaard. Een meting is een verzameling van een of meerdere tijd(x)/waarde(y) combinaties. Periodiek voert Hyperic HQ onderhoud uit op de bewaarde meetgegevens: de gegevens worden geaggregeerd wat in het geval van Hyperic wil zeggen dat de gemiddelde waarden(y) worden genomen uit van tevoren bepaalde tijdframes. Het aggregeren van de meetgegevens is nodig, omdat er anders een enorme database aan meetgegevens zou ontstaan die niet meer te beheren valt. Hyperic aggregeert de 'live' gegevens naar blokken van respectievelijk 1 uur, 6 uur en 1 dag. Onderstaande tabel geeft meer duidelijkheid:

Aggregatie	Retentietijd	Omschrijving
live	3 uur	Bevat gegevens van 3 uur geleden tot heden
1 uur	2 weken	Bevat gegevens van 2 weken geleden tot heden
6 uur	4 weken	Bevat gegevens van 4 weken geleden tot heden
1 dag	oneindig	Bevat gegevens van begin tot heden

Tabel 5.1: Hyperic data retentie perioden

Door middel van bovenstaande tabel moet duidelijk worden dat een meting waar oudere gegevens van worden opgevraagd een lagere resolutie heeft. Dat wil zeggen dat een visualisatie van die meting waar oudere gegevens van worden opgevraagd veel minder pieken en dalen zal laten zien dan visualisatie van een meting die meer bij het heden ligt. Zie onderstaande voorbeelden voor meer duidelijkheid:

Voorbeeld 1

Starttijd	Eindtijd	Onderwerp
1 februari 2008 00:00:00	1 maart 2008 00:00:00	Load op een fictieve machine

Resultaat: een visualisatie van meetgegevens met bovenstaande parameters zal een grafiek opleveren met een zeer lage resolutie. Omdat de twee data meer dan 4 weken uit elkaar liggen zal gebruik worden gemaakt van 1 dags aggregatie. Gevolg hiervan is dat grote pieken en dalen die in werkelijkheid wel degelijk heeft plaats gevonden, niet meer te identificeren zullen zijn op de visualisatie.

Voorbeeld 2

Starttijd	Eindtijd	Onderwerp
1 maart 2008 00:00:00	2 maart 2008 00:00:00	Load op een fictieve machine

Resultaat: een visualisatie van meetgegevens met bovenstaande parameters zal een grafiek opleveren met een hoge resolutie. Omdat de twee data maar 1 dag uit elkaar liggen zal gebruik worden gemaakt van de 1 uurs aggregatie. Gevolg hiervan is dat grote pieken en dalen nog duidelijk zichtbaar zijn. Korte piekmomenten zullen echter niet meer waar te nemen zijn.

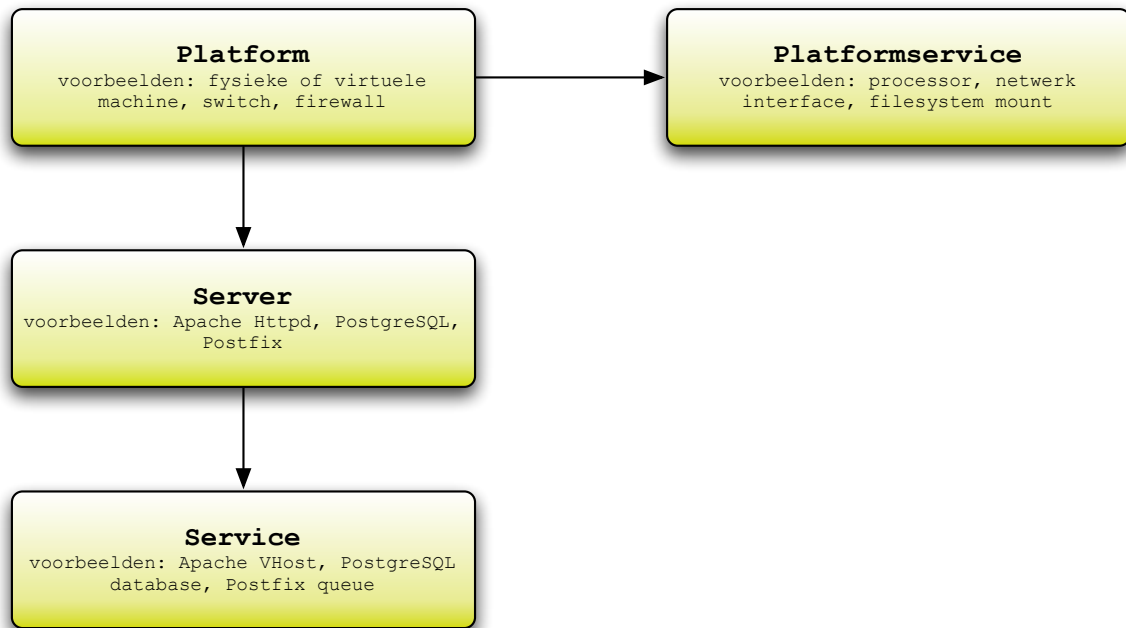
Voorbeeld 3

Starttijd	Eindtijd	Onderwerp
1 maart 2008 13:00:00	1 maart 2008 15:00	Load op een fictieve machine

Resultaat: een visualisatie van meetgegevens met bovenstaande parameters zal een grafiek opleveren met een zeer hoge resolutie. Omdat de twee data maar 2 uur uit elkaar liggen zal gebruik worden gemaakt van de live meetgegevens. Een aggregatie van meetgegevens heeft hierbij nog niet plaats gevonden. Gevolg hiervan is dat alle piek- en dal momenten zichtbaar zijn, ook die momenten die van korte duur zijn geweest.

5.5 Platform, server en service

Binnen Hyperic wordt een terminologie gebruikt die verwarrend kan zijn. Hyperic HQ maakt onderscheid tussen drie verschillende entiteiten, een platform, server en service. Een platform staat gelijk aan wat meestal een server wordt genoemd. Het is bijvoorbeeld een fysieke of virtuele computer, een switch of een firewall. Een server staat gelijk aan software die een dienst 'serveert' en op het platform draait. Voorbeelden van servers zijn Apache Httpd, PostgreSQL of Postfix. Een service kan twee betekenissen hebben. Of de service staat gelijk aan de dienst die de server (bijv. Apache Httpd, PostgreSQL of Postfix) levert. Voorbeelden van services zijn Apache VHost, PostgreSQL database of Postfix queue. Of de service staat gelijk aan een dienst die het platform levert. Voorbeelden van dat soort services zijn processor, netwerk interface of filesystem mount. Deze variant van een service wordt ook wel platformservice genoemd. Het schema in figuur 5.1 geeft hierover enige extra duidelijkheid.



Figuur 5.1: Typen entiteiten binnen Hyperic

Elke platform, server of service kan een of meerdere metingen onder zich hebben. Elke meting kan worden omgezet in een visuele representatie van die meting.

Hyperic HQ gebruikt een PostgreSQL database als backend om meetgegevens in op te slaan. De datastructuur achter Hyperic is vrij complex en voor de Hyperic web service is maar een klein deel van de beschikbare datastructuur benodigd. De overige tabellen binnen de Hyperic datastructuur zorgen voor authenticatie/autorizatie, auto-inventory, alerts, escalaties, etc.

Hoofdstuk 6

Informatie benaderen met RESTful web services

IC&S gebruikt enkele producten die in de eerste instantie bedoeld zijn voor de medewerkers van IC&S, bijvoorbeeld omdat het product toegang tot informatie verschaft die niet voor klanten bestemd is, of omdat de getoonde informatie niet of moeilijk te interpreteren is voor de gemiddelde klant. Hierbij moet dan vooral worden gedacht aan producten die omgaan met dynamische informatie en die informatie op een dynamische manier presenteren. Een voorbeeld van zo'n product is een CMDB (Configuration Management Database), een tool die bij het gebruik van ITIL Support Set een belangrijke rol speelt.

Om toch die informatie aan klanten beschikbaar te stellen die voor hen interessant is, is het, los van een papieren variant waarvan je de inhoud zelf controleert, een goede optie een interface te creëren die toegang geeft tot precies het gewenste niveau van informatie. IC&S heeft tijdens het beginstadium van de ontwikkeling van ICIS 4.0 besloten dit te willen doen doormiddel van RESTful web services.

Dit hoofdstuk geeft een inleiding op de Hyperic web service en de Bacula web service, waar een evaluatie naar enkele producten om RESTful web services te implementeren onderdeel van is en dient als beargumentatie van de keuze op welke manier RESTful web services ontwikkeld gaan worden.

6.1 Doelstelling

Aan het eind van de inception fase van iteratie 1, op 28 november 2007, zal er een keus gemaakt zijn voor een product om een RESTful web service te implementeren. Het is dan duidelijk welk product gekozen zal gaan worden.

6.2 RESTful web services

Een web service is een dienst die aanspreekbaar is over het internet. Web services kunnen gebruik maken van verschillende technieken op verschillende lagen in het OSI netwerkmodel om de informatie waartoe ze toegang geven beschikbaar te maken. Bestaande web service stijlen zijn RPC gebaseerde web services zoals XML-RPC en Service gebaseerde web services zoals SOAP.

Een alternatief, dat momenteel sterk groeit in populariteit, zijn RESTful web services. Een RESTful web service is een web service die voldoet aan de principes die samen REST vormen. Die principes zijn de volgende:

- Applicatie status en functionaliteit zijn onderverdeelt in resources.
- Elke resource is uniek aanspreekbaar doormiddel van hypermedia links.
- Alle resources delen een uniforme interface.
- Het protocol dient client-server, stateless, cachable en gelaagd te zijn.

Een RESTful web service werkt in principe net als een website. De informatie die middels de web service beschikbaar zal zijn is opgedeeld in resources, en wordt ook wel een resource oriented architecture (ROA) genoemd. Een resource staat gelijk aan een object, in taalkundige zin een zelfstandig naamwoord. Voorbeelden van resources zijn platformen, servers, services en platformservices. Een server met id 10009 dat onderdeel is van platform 457 wordt, met het oog op figuur 5.1, aangesproken met een HTTP GET request op de URI `/platforms/457/severs/10009/`.

De tegenhanger van ROA is SOA, wat staat voor service oriented architecture. SOA wordt gebruikt door een web service stijl als SOAP. Een SOAP request bestaat uit XML waar in de XML is aangegeven welke functie moet worden aangesproken met welke parameters. De SOAP request wordt gestuurd naar een specifieke URI, welke op zijn beurt de service herbergt.

Requests die naar een RESTful web service zijn gelijk aan een standaard HTTP request met als verschil dat de te gebruiken URI gebaseerd zal zijn op de resources die de web service aanbiedt. Elk HTTP request voert een actie uit, dat kan het opvragen van gegevens, aanmaken van gegevens, het verwijderen van gegevens of het updaten van gegevens zijn. De actie wordt bepaald aan de hand van de request method die met elke valide HTTP request meekomt. De request method kan goed vergeleken worden met de CRUD operaties die vaak bij relationele databases worden toegepast:

CRUD Actie	HTTP Method	URI	Actie
(C)reate	POST	/platforms	Maak een nieuwe machine aan
(R)ead	GET	/platforms/1	Retourneer het platform dat 1 als id heeft
(U)pdate	PUT	/platforms/1	Update het platform dat 1 als id heeft
(D)elele	DELETE	/platforms/1	Verwijder het platform dat 1 als id heeft

Tabel 6.1: CRUD en corresponderende HTTP methods

Zoals in de tabel hierboven te zien is, is het duidelijk dat met een enkele URI meerdere acties uitgevoerd kunnen worden, afhankelijk van de method die in de HTTP request wordt gedaan.

6.3 Productselectie

Momenteel is het aanbod van producten die RESTful web services kunnen serveren nog niet erg groot. Een logische eis voor een product is dat het stabiel, veilig en mature is. Het moet geschikt zijn te kunnen draaien in een productie omgeving. Daarbij wordt de keuze nog meer beperkt door de eis dat het product op Debian Linux moet kunnen draaien, omdat dat het

operating system waar alle systemen van IC&S op draaien. Het aanbod van producten die dat kunnen wordt met de gestelde criteria klein. Alleen Apache Axis2 en IBM's Project Zero lijken aan die eisen te kunnen voldoen.

Omdat in principe elke web applicatie het in zich heeft om RESTful te worden, is het ook een goede mogelijkheid om een RESTful web service te ontwikkelen in scripttalen als PHP, Perl, Ruby en Python. Dit zijn allemaal talen die (ook) prima toegespitst kunnen worden, of het van nature al zijn, om web applicaties te bouwen. Een project dat zich in die categorie sterk profileert als geschikt framework voor het ontwikkelen van web applicaties is Django: een web application framework voor perfectionisten met deadlines.

6.3.1 Evaluatie Apache Axis2

Apache Axis2 is een web service engine van de Apache Foundation die web services kan implementeren die gebaseerd zijn op SOAP, MTOM of REST. Apache Axis2 komt in twee uitvoeringen: een is gebouwd in Java en een ander is gebouwd in C. Speerpunten van Axis2 zijn een laag geheugengebruik, het asynchroon aanspreken van web services en hot web service deployment.

Voordelen

1. Keuze tussen Java of C.
2. Documentatie van hoge kwaliteit.
3. Axis2 is mature.

Nadelen

1. De implementatie van REST is nog vrij nieuw, en dus minder mature; de kans op bugs en kinderziektes is groter

6.3.2 Evaluatie IBM's Project Zero

Project Zero is een product van IBM. Ooit begonnen als een zogenaamd 'incubator project', een project dat begint met het ontdekken en onderzoeken van nieuwe ideeën. De bedoeling van een incubator is om de nieuwe ideeën in het echte leven te testen. Het project is inmiddels aangekomen in fase 2 van de evolutie vanaf 0 en probeert een omgeving te scheppen die het mogelijk maakt op een simpele manier applicaties te maken die gebaseerd zijn op populaire web technologieën, waaronder REST.

Voordelen

1. Uitgebreide documentatie.
2. Geoptimaliseerd voor RESTful web services .

Nadelen

1. Project is nog in een 'speelfase'. Ontwikkelaars worden uitgenodigd ermee aan de slag te gaan. De toekomst van dit product is daarom onzeker.

6.3.3 Evaluatie Django web application framework

Django omschrijft zichzelf als een web framework voor perfectionisten met deadlines en als een product dat het makkelijker maakt web applicaties sneller te ontwikkelen met minder code. Django is een compleet framework dat zich richt op meerdere aspecten van web applicaties (template system, internationalization, caching, authenticatie, autorisatie). Django bezit daarnaast een ingebouwde object-relational mapper; data modellen kunnen compleet in Python (de taal waar Django in geschreven is) worden gedefinieerd. Een compleet product.

Voordelen

1. Uitgebreide documentatie.
2. Compleet product.
3. De ontwikkeling van dit product loopt al enkele jaren en mag mature genoemd worden.

Nadelen

1. Dit product bevat veel toevoegingen die niet interessant zijn voor de doelen die voor dit afstuderen gedefinieerd zijn.

6.4 Conclusie

Op het eerste gezicht lijken Apache Axis2 en het Django web application framework de twee meest serieuze kandidaten. IBM's Project Zero is nog in een speelfase en kan nog niet mature genoemd worden, ondanks dat de nodige documentatie en ontwikkeltools aanwezig zijn. In ons achterhoofd houdend dat de web service geen honderden requests per seconde te verwerken zal krijgen maar we eerder moeten denken aan enkele honderden tot duizenden requests per dag, is het ook niet nodig een zwaar product te gebruiken dat daar mee om kan gaan.

Django en Axis2 zijn twee totaal verschillende producten met verschillende mogelijkheden over het geheel gezien, maar met dezelfde mogelijkheden wat betreft de implementatie van RESTful web services. Daarbij is natuurlijk een verschil dat Axis2 een web service engine is en Django een web service framework, wat in dit geval betekent dat Django een bepaalde ontwikkel architectuur verplicht stelt waar Axis2 meer vrijheid biedt. Django is op Python gebaseerd wat goed presteert in een Linux omgeving en is, relatief tot Axis2 gezien, lichtgewicht. De verplichte architectuur die Django je verplicht te gebruiken is een pluspunt, omdat je op die manier automatisch de juiste weg in wordt geleid en minder de weg hoeft te zoeken in de vrijheid die Axis2 biedt.

De verplichte architectuur en het gewicht van Django maken dat Django is gekozen boven Axis2 om de gebouwde web service mee te implementeren. De eigenschappen van Django maken dat Django in dit geval meer geschikt is dan Axis2.

Hoofdstuk 7

Visualisatie van meetgegevens

Meetgegevens zijn meestal een combinatie van twee of meer corresponderende gegevens. De gegevens horen bij elkaar en hebben vaak eenzelfde onderwerp. De meetgegevens helpen ons bij een betere interpretatie van gebeurtenissen en kunnen daarmee helpen tot het komen van inzichten die anders niet mogelijk waren, om vervolgens weer te komen tot oplossingen van problemen.

Bij IC&S worden doorgaans een combinatie van twee verschillende gegevens gebruikt, tijd(x) en waarde(y), omdat dit voldoet voor de manier waarop IC&S metingen wil interpreteren. Door de waarde aan een tijd te koppelen kunnen bepaalde gebeurtenissen worden bepaald en trends worden geanalyseerd. Losse meetgegevens echter laten nog niets zien; het is het naast elkaar laten zien en combineren van die meetgegevens die ons de inzichten geven waar vaak zo naar gezocht wordt. Het maken van een visuele representatie van diezelfde meetgegevens kan een hulpmiddel zijn hierbij.

Om te voldoen aan de requirements van de Hyperic web service (zie hoofdstuk 9.1) zullen er verschillende grafieken moeten kunnen worden gegenereerd die binnen een geselecteerd tijd-frame moeten vallen. Daarnaast moeten de grafieken er goed uitzien, goed aan te passen zijn en moet het genereren van de grafieken op basis van de meetgegevens niet te zwaar drukken op de prestaties van het systeem.

Er is een grote keuze aan graphing en reporting engines beschikbaar en de vrije keuze voor programmeertaal, graphing engine of reporting engine maakt de keuze alleen nog maar groter. Dit hoofdstuk geeft een inleiding op de Hyperic web service, waar een evaluatie naar enkele verschillende manieren om grafieken te genereren onderdeel van is en dient als beargumentatie van de keuze voor de grafieken generator tool (zie H9).

7.1 Doelstelling

Aan het eind van de inception fase van iteratie 1, op 28 november 2007, zal er een keus gemaakt zijn voor een graphing engine of reporting engine. Het is dan duidelijk welke graphing engine of reporting engine gekozen zal gaan worden en welke programmeertaal daar bij komt kijken.

7.2 Aanpak

Omdat er enorm veel graphing- en reporting engines in omloop zijn is de keuze te groot om zonder enige criteria aan het product een selectie te maken. Daarom is een selectie gemaakt van software die op het eerste gezicht voldoet aan onderstaande criteria:

1. Software is stabiel en veilig genoeg om in productie te draaien (d.w.z. de software is productieklaar of mature genoeg)
2. Software is gratis in gebruik; Open-source is niet verplicht, maar wel een pre
3. Website is in het Nederlands of Engels
4. Software wordt momenteel nog doorontwikkeld (d.w.z. de ontwikkeling van het product ligt niet stil)

De gevonden producten worden beoordeeld op basis van maturity (criteria 1), systeemimplicaties, grafische pracht en de verwachte leercurve, op basis van informatie op de website en andere (eventuele) community sources. Er is geen tijd om elk product te installeren en uitvoerig te testen.

7.3 Productselectie

De volgende producten voldoen op het eerste gezicht aan de criteria zoals die hierboven gesteld zijn en zullen daarom verder geëvalueerd worden.

- JpGraph
- Biggles
- Ploticus
- JasperReports (open-source editie)

7.3.1 Evaluatie JpGraph

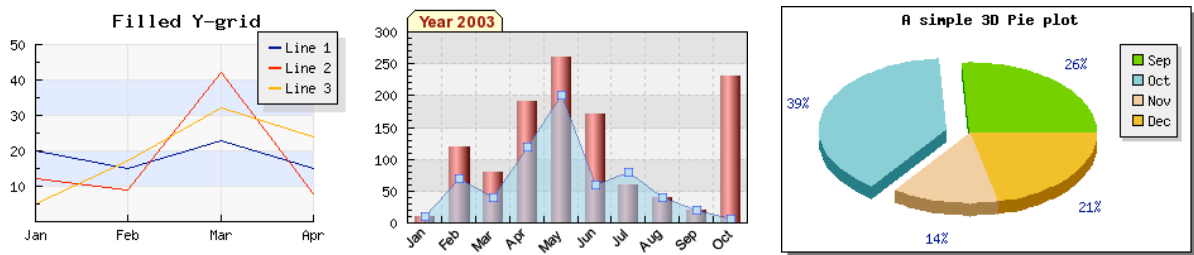
JpGraph is een object georiënteerde graphing library voor PHP en ook volledig geschreven in PHP. Het maakt gebruik van lib GD (library om grafische objecten te tekenen) en is in staat om veel verschillende soorten grafieken en diagrammen te tekenen, waaronder o.a. taartdiagrammen, lijngrafieken en staafdiagrammen. De opmaak van de getekende objecten is goed aan te passen. Zie figuur 7.1 voor voorbeelden van grafieken die JpGraph kan produceren.

Voordelen

1. Erg mature door vele functionaliteit en lange bestaan.
2. Voor IC&S een bewezen product.
3. Grafisch aantrekkelijk en goed aanpasbaar.
4. Ervaring met dit pakket is ruim aanwezig, dus eigenlijk geen leercurve.

Nadelen

1. PHP is niet standaard op elk systeem geïnstalleerd.
2. PHP is een interpreted language en, vooral bij grote datasets, niet zo snel als een compiled language.
3. PHP i.s.m. GD drukt relatief veel op de prestaties van het systeem waar het op draait.



Figuur 7.1: Enkele voorbeelden van grafieken die JpGraph produceren kan

7.3.2 Evaluatie Biggles

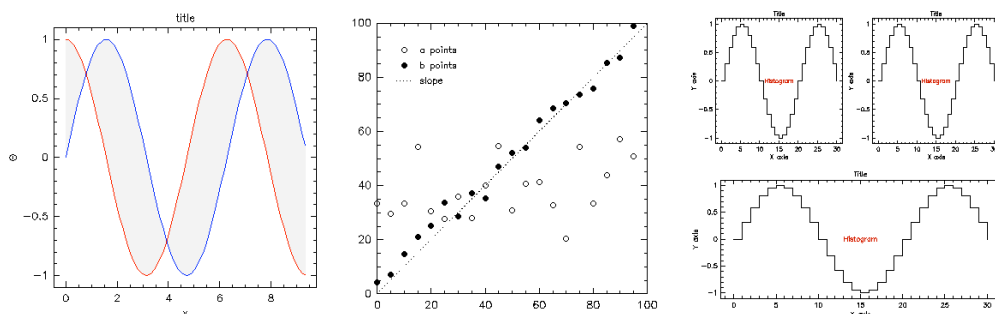
Biggles is een Python module die in staat is verschillende soorten grafieken en diagrammen te tekenen. De opmaak van de objecten die Biggles tekent is vrij aan te passen. Zie figuur 7.2 voor voorbeelden van grafieken die Biggles kan produceren.

Voordelen

1. In Python is snel te ontwikkelen, omdat er standaard al veel modules aanwezig zijn die veel werk uit handen kunnen nemen.
2. Op veel systemen is Python wel standaard aanwezig.
3. Ervaring met Python is wel aanwezig, dus geen stijle leercurve.
4. Grafieken zijn visueel goed aan te passen.

Nadelen

1. Nog niet zo mature.
2. Grafisch onaantrekkelijk.
3. Interpreted language, zie hiervoor nadeel 2 van JpGraph.



Figuur 7.2: Enkele voorbeelden van grafieken die Biggles produceren kan

7.3.3 Evaluatie Ploticus

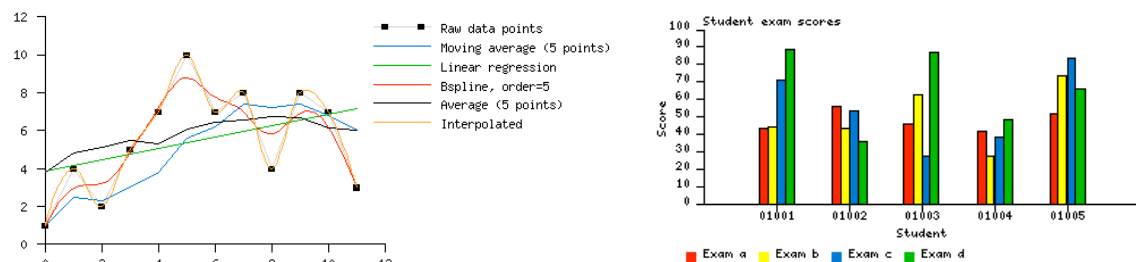
Ploticus is een in C geschreven library die in staat is verschillende soorten grafieken en diagrammen te tekenen, waaronder o.a. lijn- en staafdiagrammen en heatmaps. De opmaak van de getekende objecten is vrij aan te passen. Ploticus werkt met een eigen scripttaal om grafieken in hun opmaak en inhoud te definiëren. Zie figuur 7.3 voor voorbeelden van grafieken die Ploticus kan produceren.

Voordelen

1. De geproduceerde grafieken zijn goed aan te passen.
2. C is een compiled language en zal sneller en beter om kunnen gaan met grote datasets dan een interpreted language.
3. Kennis van C is niet nodig, vanwege de beschikbaarheid van een ploticus scripttaal.

Nadelen

1. Ervaring met de scripttaal van Ploticus is niet aanwezig, dat maakt de leercurve iets stijler.
2. De geproduceerde grafieken zijn grafisch niet erg aantrekkelijk.



Figuur 7.3: Enkele voorbeelden van grafieken die Ploticus produceren kan

7.3.4 Evaluatie JasperReports

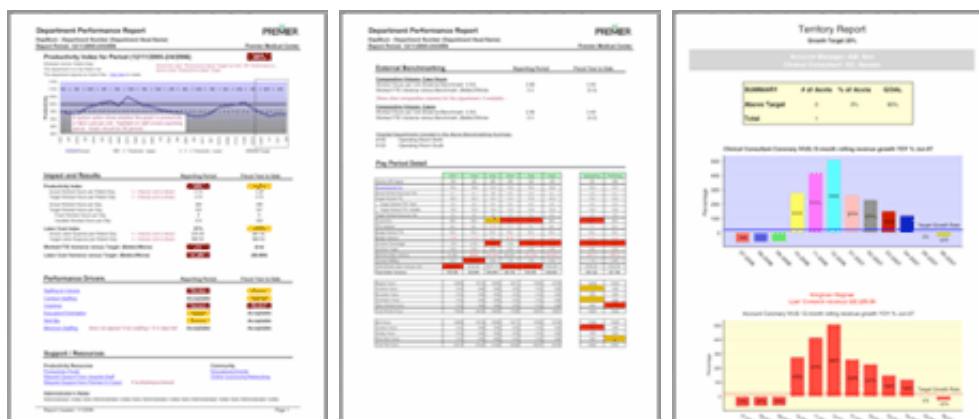
JasperReports is een in Java geschreven graphing en reporting engine. JasperReports heeft ontzettend veel features en is, in tegenstelling tot JpGraph en Ploticus die alleen grafieken en diagrammen genereren, ook in staat complete rapporten op te maken. Zie figuur 7.4 voor voorbeelden van rapporten die JasperReports kan produceren.

Voordelen

1. Enterprise Business Intelligence, dus wat features en stabiliteit betreft erg mature en productie klaar.
2. Erg uitgebreid door o.a. complete rapportage engine, subreports en crosstabs.

Nadelen

1. Geschreven in Java, maakt samen met de omvang van het pakket dat het erg veel resources nodig heeft.
2. Meer features dan nodig, wat het productodeloos ingewikkeld maakt.
3. Geen ervaring met dit product is aanwezig, wat een stijle leercurve creëert.



Figuur 7.4: Enkele voorbeelden van rapporten die JasperReports produceren kan

7.4 Conclusie

JasperReports is een mooi product: het is professioneel, heeft veel in huis en heeft een complete reporting engine aan boord. De omvang van het pakket en de niet aanwezige ervaring van het product, maakt dat het groter is dan nodig. Daarnaast maakt het feit dat het product in Java ontwikkeld is, dat het een flinke impact op het systeem zal hebben. Biggles is grafisch onaantrekkelijk. Het feit dat het in Python ontwikkeld is maakt de combinatie met Django wel aantrekkelijk, omdat een betere integratie tussen de twee producten dan mogelijk is. Ook Ploticus is grafisch onaantrekkelijk, hoewel het feit dat het in C geschreven is wel een hoop goed maakt.

Het mag duidelijk zijn dat grafische pracht zwaar weegt in deze evaluatie. Het is erg belangrijk dat de gegenereerde grafieken er grafisch aantrekkelijk en duidelijk uitzien. Ook de tijd die het kost om het product in gebruik te kunnen nemen is een belangrijke factor, omdat er deadlines aan dit project vastzitten. JpGraph biedt grafische pracht, een vlakke leercurve omdat ervaring al aanwezig is en is voor IC&S een bewezen product. De druk die PHP i.s.m. GD extra op het systeem zet, is daarbij verwaarloosbaar. Er wordt daarom voor gekozen grafieken te genereren met behulp van JpGraph.

Hoofdstuk 8

Architectuur

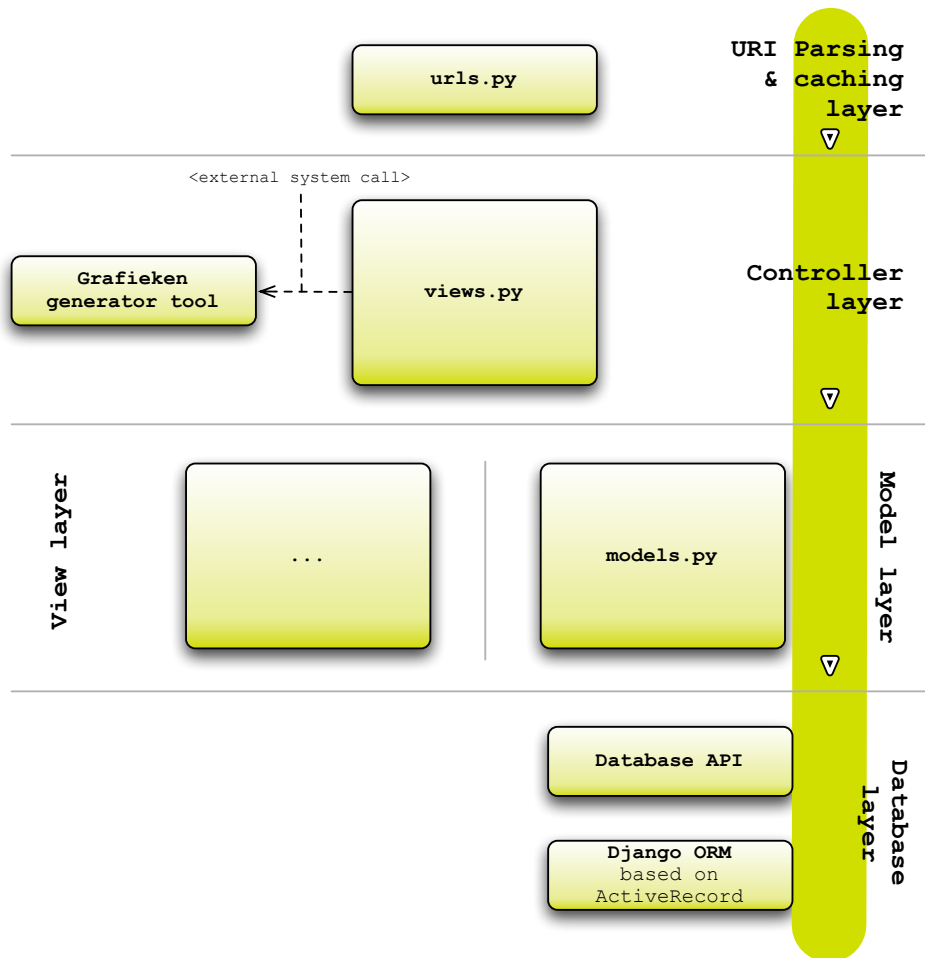
8.1 Web services

Figuur 8.1 toont de architectuur van de Hyperic web service en de Bacula web service zoals deze in Django geïmplementeerd is. Django werkt hoofdzakelijk op de volgende vijftal lagen: URI parsing en caching, controller, view, model en database. Elke laag heeft zijn eigen taken welke hieronder beschreven worden. De architectuur beschrijving van de Bacula web service en de Hyperic web service is gecombineerd omdat deze, op de voorbeeldnamen van de controllers na, volledig overeen komt.

- De **URL parsing en caching layer** heeft als doel de binnengekomen requests naar de juiste controller door te redigeren of eventuele items in de Django cache direct te retourneren. Het redigeren van een request naar de juiste controller wordt gedaan op basis van een reguliere expressie op het URI path. De reguliere expressie extraheert parameters, zoals bijvoorbeeld platform id of service id uit het URI path en passed deze door aan de controller.
- De **controller layer** wordt met de juiste parameters aangesproken op het moment dat Django geen gecached resultaat kan retourneren. Op basis van de parameters doet de controller het werk waarvoor het geprogrammeerd is. In het geval van Hyperic haalt de view controller (view.py) een lijst op met bestaande platformen en retourneert deze. Op het moment dat er een grafiek opgevraagd wordt doet de controller in kwestie een system call naar de grafieken generator tool.

In het geval van Bacula wordt een lijst met backup client geretourneerd en wordt het bestaande grafieken script aangesproken om een grafiek te genereren en retourneren.

- De **view layer** is wel een laag in Django maar wordt niet gebruikt door de Hyperic- of Bacula web service omdat de te retourneren informatie direct buiten een view om worden geserialiseerd naar een XML document. De reden hiervoor is dat de consumer van de Hyperic- of Bacula web service, de ICIS 4.0 klantenportal, meer heeft aan een gestructureerd XML document dan een opgemaakte (x)html pagina.
- De **model layer** bevat een definitie van het database model en past de ORM techniek die Django bevat toe. Elk model representeert een tabel in de database.
- De **database layer** verzorgt de relatie en vertaling tussen programma model en database model en implementeert ORM in Django. De ORM implementatie in Django is gebaseerd op ActiveRecord¹. Met deze laag heb je als ontwikkelaar niet direct te maken.



Figuur 8.1: Architectuur diagram van de Hyperic- en Bacula web service

8.2 Grafieken generator tool

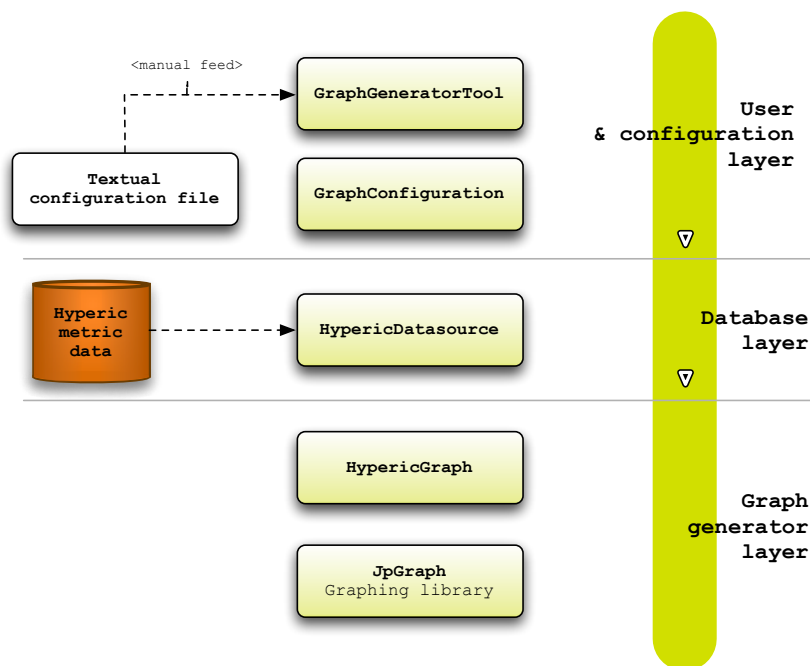
Figuur 8.2 toont de architectuur van de grafieken generator tool. De grafieken generator tool is onderverdeelt op de volgende drie lagen: user en configuration, database en graph generator. Elke laag heeft zijn eigen functie.

- De **user en configuration layer** ontvangt de gewenste programma parameters vanaf de command line. Daarnaast leest het een configuratie bestand uit dat enkele parameters bevat die niet via de command line worden meegegeven. De parameters die via de command line worden meegegeven zijn onder andere de start- en eindtijd en de meting waar de te produceren grafiek uiteindelijk een visualisatie van zal zijn.
- De **database layer** heeft als taak op basis van de opgegeven meting de database aan te spreken en de metrische data voor de gewenste periode op te vragen. Ook worden er

¹Active Record is een steeds vaker toegepast ontwerp patroon. Zie ook http://en.wikipedia.org/wiki/Active_record_pattern.

algemene gegevens over de meting opgevraagd, zoals de naam van de meting en de eenheid waarin de meting wordt uitgedrukt.

- De **graph generator layer** heeft de taak een visualisatie (grafiek) te genereren op basis van de gegevens die de database layer heeft verzameld. De gegenereerde grafiek komt in de vorm van een plaatje dat wordt opgeslagen op een plek op het filesysteem dat in het configuratie bestand gedefinieerd is.



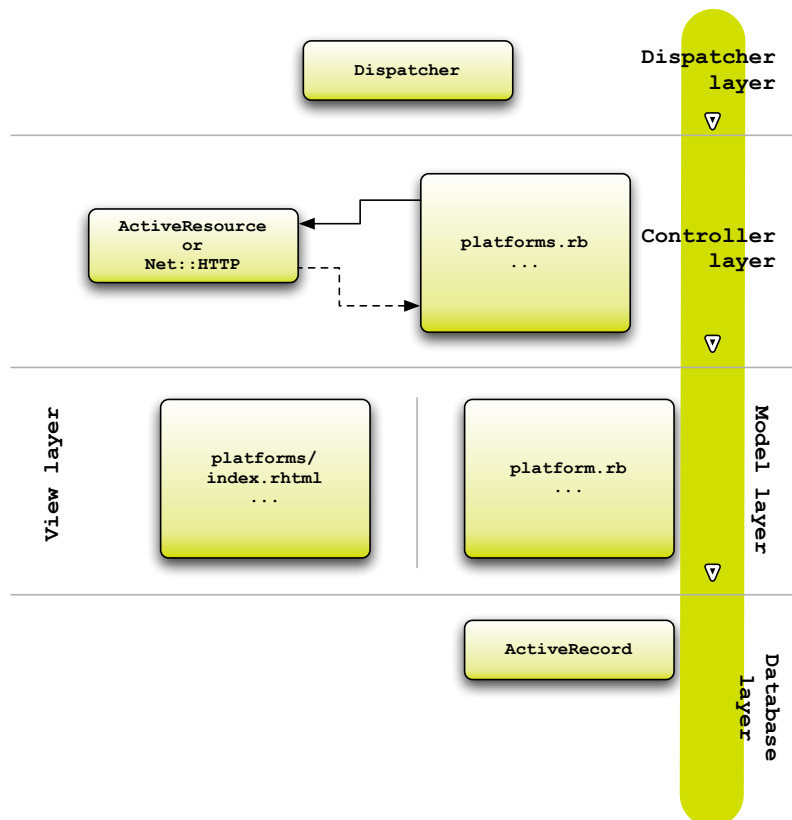
Figuur 8.2: Architectuur diagram van de grafieken generator tool

8.3 ICIS 4.0 web service client

Figuur 8.3 toont de architectuur van de ICIS 4.0 web service client. De ICIS 4.0 web service client is onderverdeeld in de 5 lagen: dispatcher, controller, model, view en database. De functie van elke laag wordt hieronder beschreven.

- De **dispatcher layer** heeft als functie binnengekomen HTTP requests naar de juiste controller door te sturen. De dispatcher maakt een keuze voor de juiste controller op basis van zogenaamde routes die in de configuratie van een RoR applicatie gedefinieerd staan.
- De **controller layer** heeft als functie het juiste model aan te spreken als dat nodig is en de gewenste view aan te spreken. Waarden die in de view getoond dienen te worden worden door de controller geïnitieerd. De controller is daarnaast verantwoordelijk voor de afhandeling van formulieren.

- De **model layer** heeft als functie relaties met andere models te definiëren, constraints voor database kolommen te definiëren en database specifieke functies zoals het samenvoegen van database kolomwaarden te definiëren.
- De **view layer** bevat de markup van de grafische gebruikersinterface en bestaat uit een verzameling zogenaamde rhtml bestanden die als template fungeren. In een template kan gebruik worden gemaakt van Ruby code om daarvoor in de controller aangewezen variabelen aan te spreken. Op die manier is het mogelijk een dynamisch element aan de gebruikersinterface toe te voegen.
- De **Database layer** gebruikt het ActiveRecord ontwerp patroon in samenwerking met ORM om de database aan te spreken en een objectmatige representatie van database tabellen toegankelijk te maken voor model en controller. op die manier kunnen de standaard CRUD operaties op het object uitgevoerd worden in plaats van direct op de database.



Figuur 8.3: Architectuur diagram van de ICIS 4.0 web service client

Hoofdstuk 9

Hyperic web service

9.1 Requirements

Tijdens overleg met Bart van de Garde dat tijdens de inception fase is uitgevoerd en uit het Engelstalige Basecamp (tool voor projectmanagement) zijn onderstaande requirements naar voren gekomen. Deze requirements zijn nadrukkelijk alleen van toepassing op de 1e iteratie van het afstudeer project en opgedeeld in de producten Hyperic web service en grafieken generator tool, zoals ook gesteld in de deliverables voor deze iteratie.

Hyperic web service

1. Gebaseerd op REST.
2. Moet grafieken kunnen leveren voor metingen gerelateerd aan platformen, servers en services.
3. Moet kunnen omgaan met parameters als startdate, enddate, warningvalue en criticalvalue.
4. Moet een grafiek als plaatje retourneren, dus geen metrische data; al het 'rekenwerk' moet al uitgevoerd zijn.
5. Moet in staat zijn grafieken te cachen om de load op de grafieken generator tool te verlagen.
6. Keuze voor software om de web service mee te implementeren alsmede de keuze voor programmeertaal staat vrij.
7. De software die de web service implementeert alsmede de te gebruiken programmeertaal moet kunnen werken op de Debian Linux 3.1 distributie.

Een andere eis die gesteld was, was de eis om later de mogelijkheid te hebben om authenticatie op de web service gebruik te laten maken van de access control lists (ACL's) die in Hyperic gespecificeerd staan. Deze eis is komen te vervallen in verband met de complexiteit van de implementatie hiervan. De tijd die de implementatie zou kosten weegt niet op tegen het relatief kleine voordeel dat er mee zou kunnen worden behaald.

Grafieken generator tool

1. Keuze voor zowel programmeertaal als graphing library staat vrij.
2. Moet in staat zijn stand-alone, dus afhankelijk van de Hyperic web service te opereren.
3. Moet in staat zijn grafisch aantrekkelijke grafieken te genereren.
4. Moet efficient zijn (een aantrekkelijke systemload:snelheid verhouding hebben)¹.
5. Moet in staat zijn verschillende soorten grafieken te genereren.
6. Moet in staat zijn een dataset op te bouwen binnen een bepaald tijdsframe, zie ook het derde requirement van de Hyperic web service.
7. Zowel de programmeertaal als de graphing library moeten kunnen draaien op de Debian 3.1 Linux distributie.

9.2 Planning

Onderstaande tabel toont een uitgebreide planning voor de Hyperic web service. De planning is opgedeeld in de vier fasen die een UP iteratie bevat en voor elke fase zijn de specifieke taken en hun geplande doorlooptijd beschreven. Een standaard werkweek bij IC&S bestaat uit 5 dagen van 8 uur.

Inception fase (19-11-2007 t/m 28-11-2007)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Onderzoek REST specificaties, toepassingsmogelijkheden en implementatie	19-11-2007	19-11-2007	8
Onderzoek Hyperic datastructuur	20-11-2007	20-11-2007	6
Onderzoek en vaststellen requirements	21-11-2007	21-11-2007	5
Onderzoek reporting en graphing engines	21-11-2007	22-11-2007	5
Modelleren use-cases	22-11-2007	22-11-2007	5
Opstellen inception rapport	27-11-2007	28-11-2007	16
Kennismaken met ICIS 4.0 en Ruby on Rails	19-11-2007	28-11-2007	8
		subtotaal	54
<i>Opmerking: vrijdag 23 november 2007 zal een vrije dag zijn. Accoord hiervoor is ruim van tevoren door de opdrachtgever gegeven.</i>			

Tabel 9.1: Planning inceptie fase, 1e iteratie

¹De aantrekkelijkheid van de systemload:snelheid verhouding is subjectief. Belangrijk is dat de verhouding redelijk blijft naar maatstaven van de server en de situatie.

Ontwikkeling van RESTful web services voor ICIS 4.0

Elaboration fase (29-11-2007 t/m 03-12-2007)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Modelleren Hyperic web service in UML klassediagram	29-11-2007	29-11-2007	6
Modelleren Hyperic web service in UML sequentie-diagram	29-11-2007	30-11-2007	6
Modelleren Hyperic web service in UML deploymentdiagram	30-11-2007	30-11-2007	4
Opmaken elaboration rapport	03-12-2007	03-12-2007	8
		subtotaal	24

Tabel 9.2: Planning elaboration fase, 1e iteratie

Construction fase (04-12-2007 t/m 14-12-2007)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Constructie Hyperic web service + grafieken generator tool	04-12-2007	12-12-2007	56
Constructie simpele REST web service client	13-12-2007	13-12-2007	8
Opmaken construction rapport	14-12-2007	14-12-2007	8
		subtotaal	72

Tabel 9.3: Planning construction fase, 1e iteratie

Transition fase (17-12-2007 t/m 21-12-2007)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Uitvoeren ad-hoc tests door opdrachtnemer	17-12-2007	18-12-2007	10
Opmaken handleiding gebruiker	18-12-2007	18-12-2007	6
Opmaken handleiding ontwikkelaar	19-12-2007	19-12-2007	8
Opmaken transitie rapport	20-12-2007	20-12-2007	8
Oplevering producten + overleg	21-12-2007	21-12-2007	2
		subtotaal	34

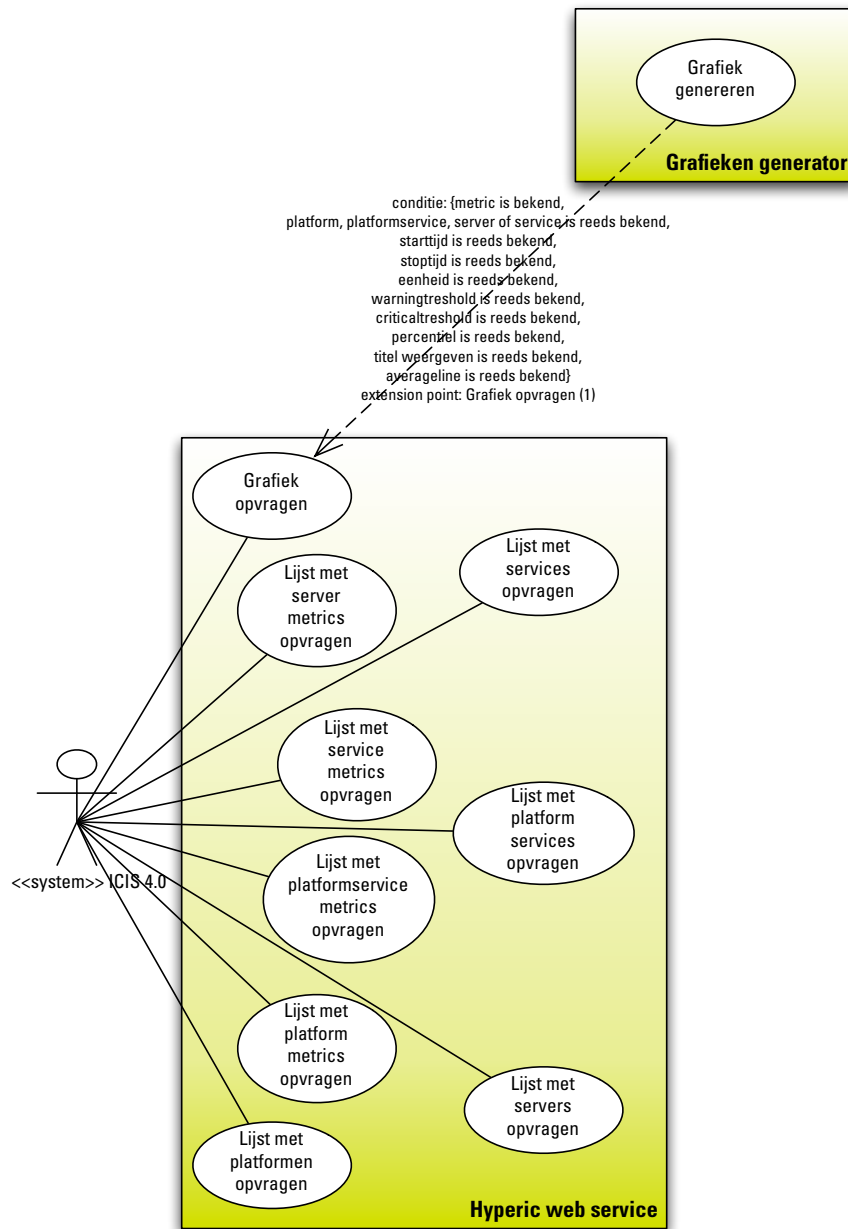
Tabel 9.4: Planning transition fase, 1e iteratie

Totaal aantal uur iteratie 1	184
-------------------------------------	------------

9.3 Use-cases

Het diagram in figuur 9.1 toont het use-case diagram voor de Hyperic web service. In het diagram zijn een tweetal systemen gedefinieerd, de Hyperic web service en de grafieken generator tool, waarbij de enig gedefinieerde actor alleen doormiddel van de Hyperic web service de grafieken generator tool kan aanspreken. De condities voor het aanspreken van de grafieken generator tool door de Hyperic web service zijn indirect vastgesteld door de requirements van de grafieken generator tool. Het zijn de parameters die essentieel zijn voor een juiste werking van de grafieken generator tool die opgenomen zijn in de lijst met condities.

De use-cases binnen het Hyperic web service systeem zijn bepaald op basis van de requirements van de Hyperic web service en de datastructuur die Hyperic heeft. Zoals te zien in figuur 5.1 maakt Hyperic gebruik van een bepaalde hiërarchie voor zijn entiteiten. Omdat daar niet vanaf kan worden geweken is diezelfde hiërarchie ook doorgevoerd in de Hyperic web service. Dat houdt in dat voor elke entiteit een aparte lijst moet worden opgehaald. Omdat elke entiteit een of meerdere metingen bevat, kan ook voor elke entiteit een lijst met beschikbare metingen worden opgehaald. Hiervoor zijn de 'Lijst met * metrics opvragen' use-cases.



Figuur 9.1: Hyperic web service use-case diagram

De tabel in tabel 8.2 beschrijft de use-case lijst met platformen opvragen. De overige use-case beschrijvingen zijn als appendix bijgevoegd in appendix A.

Naam	Lijst met platformen opvragen
Samenvatting	Er wordt een lijst met platformen opgevraagd. Het resultaat zal hetzelfde zijn als de lijst met platformen in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Geen
Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met platformen in Hyperic. 2. De Hyperic web service haalt een lijst met platformen op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platformen] Als er geen platformen in de Hyperic database zitten, dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met platformen die in Hyperic zitten.

Tabel 9.5: Use-case Lijst met platformen opvragen

9.4 Toegevoegde systeemeisen

Tijdens de elaboration fase zijn de volgende systeemeisen aan de bestaande systeemeisen toegevoegd:

Hyperic web service

1. **Mogelijkheid om percentiel parameters te accepteren** - wanneer op bandbeedte gefactureerd wordt, is het vaak gebruikelijk 5% van de hoogste waarden weg te laten in de factuur. Dit wordt ook wel het 95e percentiel genoemd. De Hyperic web service moet in staat zijn om te gaan met een parameter die het percentiel bepaald.
2. **Mogelijkheid om een eenheid parameter te accepteren** - Metingen binnen de Hyperic monitoring applicatie worden altijd uitgevoerd in de kleinst mogelijke eenheid. Het inkomende dataverkeer op een netwerk interface wordt bijvoorbeeld gemeten in bytes. Soms gaat het om zule grote hoeveelheden dat er omgerekend moet worden om een grafiek leesbaar te houden. Daarom moet de Hyperic web service in staat zijn om te gaan met een parameter die de eenheid bepaald.
3. **Communicatie tussen de Hyperic web service en haar client moet over de Secure Socket Layer (SSL) gaan** - De communicatie tussen de Hyperic web service en de client die met de Hyperic web service communiceert is gevoelig. Het zou erg onprofessioneel overkomen mocht de informatie die de Hyperic web service toegankelijk maakt in verkeerde handen vallen. Communicatie tussen Hyperic web service en client moet daarom over een secure socket verlopen, om het op die manier onmogelijk te maken het verkeer tussentijds te onderscheppen.

Grafieken generator tool

1. **Mogelijkheid om een percentiel waarde te berekenen en af te tekenen** - Wanneer er een percentiel waarde is bepaald moeten de waardes in de meting gesorteerd worden op grootte, om vervolgens $100 - \langle \text{percentiel waarde} \rangle$ in procenten van deze waarden hier vanaf te halen. De hoogste waarde die dan nog overblijft is de gedefinieerde percentiel waarde. In de grafiek moet een horizontale lijn getekend worden op de betreffende waarde.
2. **Mogelijkheid om de waarden binnen een meting om te rekenen** - Wanneer er een alternatieve eenheid is bepaald, moeten de waarden in de meting worden omgerekend om aan de eenheid te voldoen. Een voorbeeld van een omrekening is bytes naar gigabytes. De eenheid die gebruikt wordt moet worden weergegeven in de grafiek, om onduidelijkheid te voorkomen.

9.5 Deployment

Het deploymentdiagram verdeelt losse omgevingen in zogenaamde nodes en de systemen die daarop draaien in artefacten. Op die manier is de verdeling tussen de verschillende artefacten goed te zien en wordt duidelijker wat de relaties zijn tussen elk van de systemen. Het diagram is te zien in figuur 9.2.

Elk blok met ronde hoeken in bovenstaand deploymentdiagram geeft weer in welke iteratie de node die daar in zit in behandeling wordt genomen. Het blok met de groene rand is de iteratie die op dat moment in behandeling is. Elke kubus is een node en elk rechthoek dat daar in zit is een artefact.

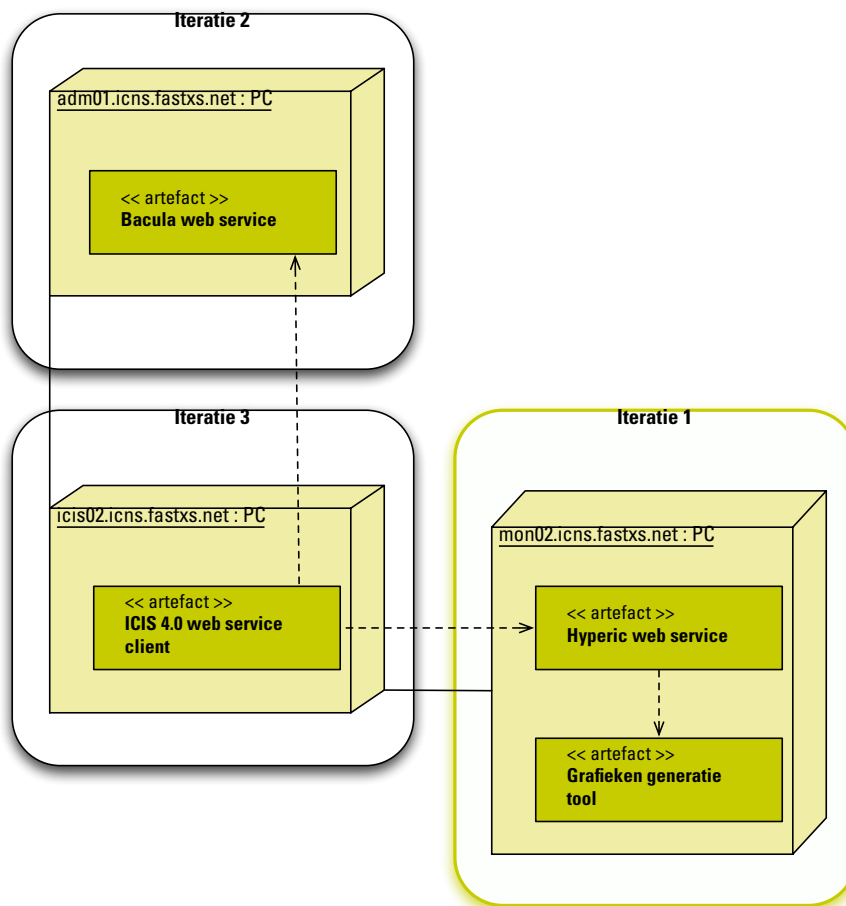
Te zien is dat de Hyperic web service en de grafieken generator tool allebei op dezelfde node komen te draaien. De grafieken generator tool is van buitenaf alleen toegankelijk via de Hyperic web service. Vanaf de node zelf zal de tool ook toegankelijk zijn vanaf de CLI, maar dat valt buiten de scope van de opdracht. Binnen de scope van dit afstuderen wordt de Hyperic web service wordt aangesproken door een web service client die in ICIS 4.0 zal worden gebouwd, maar in theorie is het mogelijk de Hyperic web service aan te spreken met een HTTP request vanuit elke denkbare bron.

9.6 Klassediagrammen

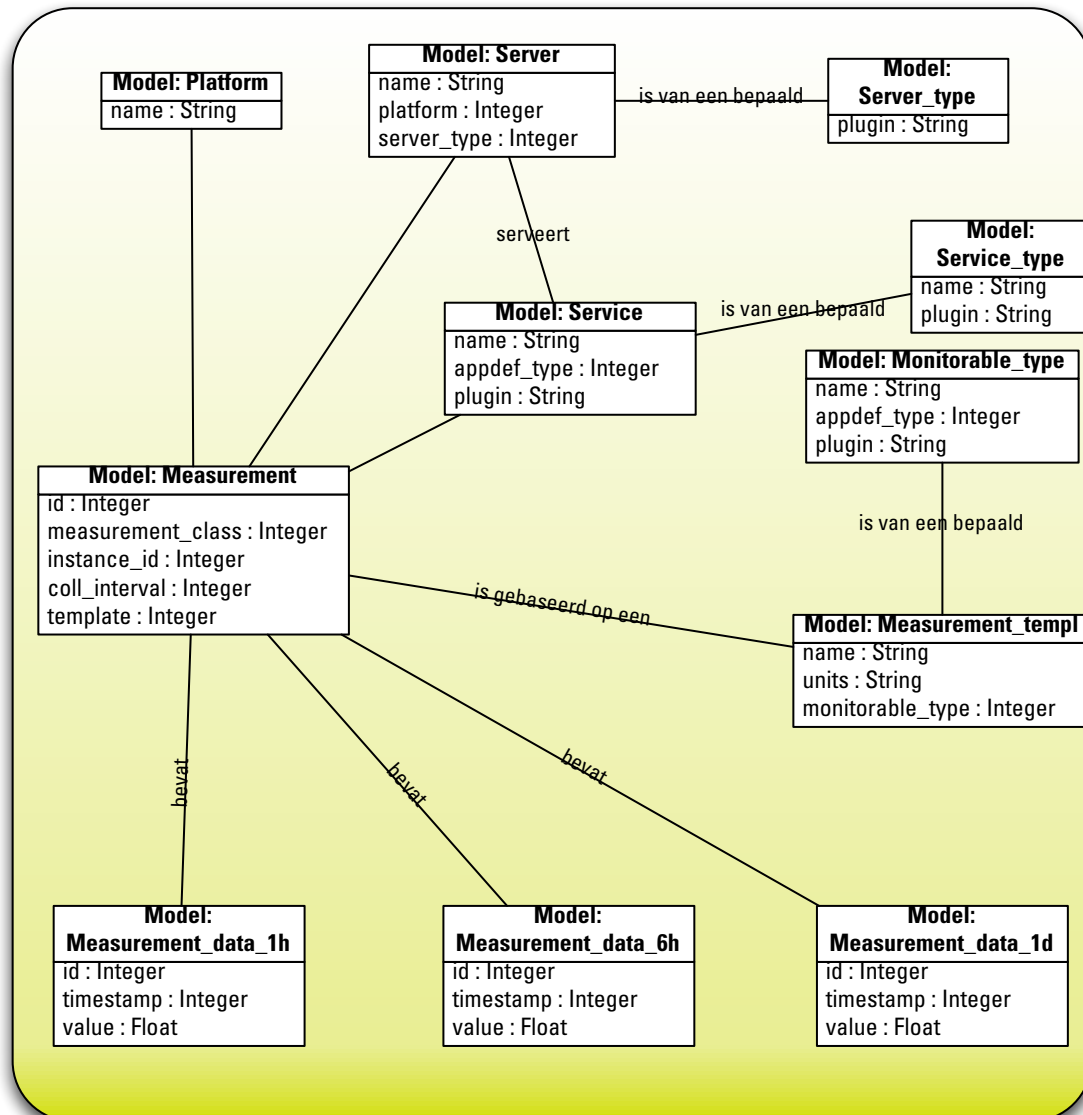
Dit hoofdstuk bevat klassediagrammen voor de Hyperic web service en de grafieken generator tool. Voor zowel de web service als de grafieken generator tool geldt dat er een domeinmodel en een uitgebouwd model aanwezig is. Elk uitgebouwd model bevat een deel van het bijbehorende model dictionary die in zijn geheel terug te vinden is in appendix B.

9.6.1 Hyperic web service domein klassediagram

In figuur 9.3 is een domein klassediagram te zien behorende bij de Hyperic web service. Het domein klassediagram is toegespitst op het gebruik van Django, waarbij gebruik wordt gemaakt van het MVC ontwerp patroon en is gebaseerd op de achter Hyperic HQ liggende data structuur, omdat daar niet van af te wijken is.



Figuur 9.2: Hyperic web service deployment diagram

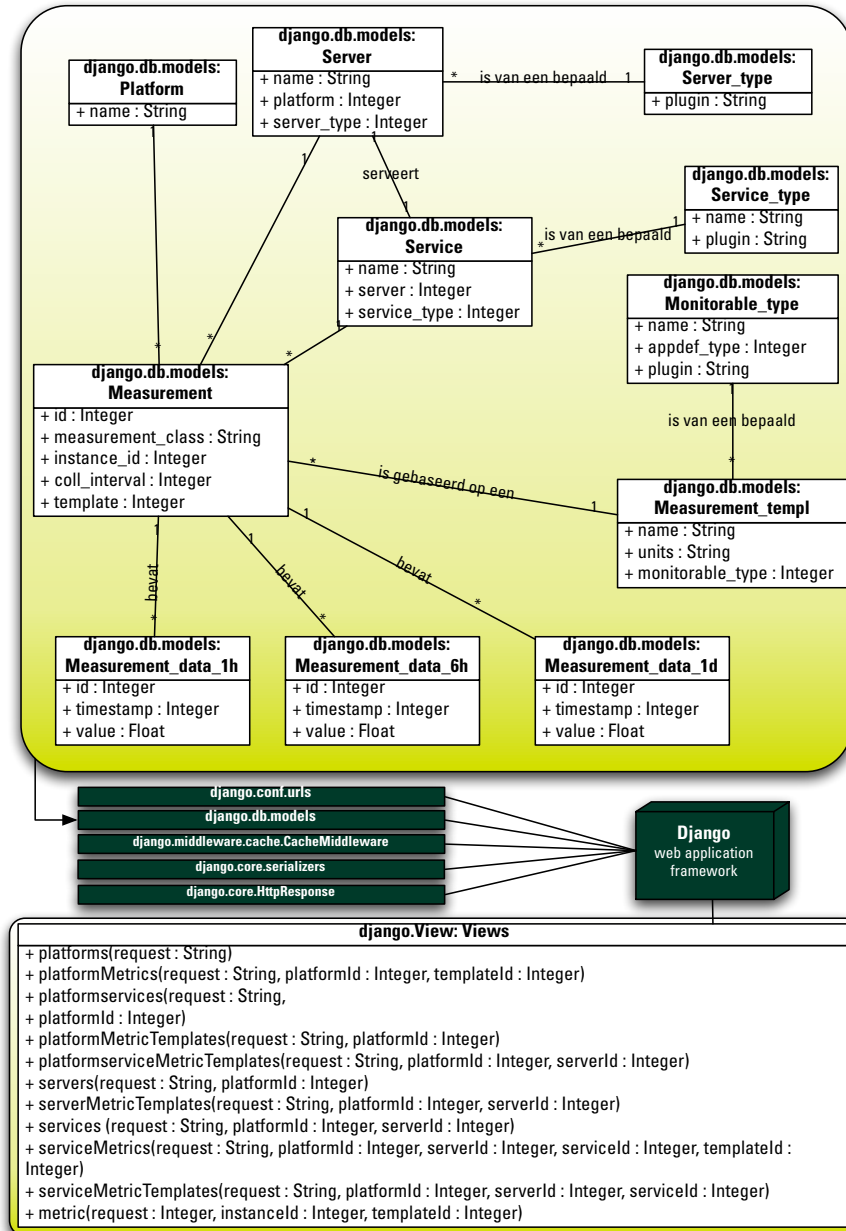


Figuur 9.3: Hyperic web service domein klassediagram

Dit domein klassediagram is sterk gerelateerd aan de data structuur die achter Hyperic HQ ligt en bevat de benodigde modelklassen die nodig zijn om aan de gestelde requirements te voldoen. De Measurement_data_x klassen bevatten geaggregeerde meetgegevens van de specifieke meting waaraan ze gerelateerd zijn. Elke meting, een instantie van de Measurement klasse, is verbonden aan een platform, service of server. Een service is, wanneer het niet verbonden is aan een server, een platformservice.

9.6.2 Hyperic web service uitgebouwd klassediagram

Figuur 9.4 laat een uitgebouwd klassediagram van de Hyperic web service zien. Dit uitgebouwde klassediagram is gebaseerd op het domein klassediagram.



Figuur 9.4: Hyperic web service klassediagram

In het uitgebouwde klassediagram is het domein klassediagram uitgewerkt tot een implementatie in Django. De klassen uit het domein klassediagram zijn opgenomen als model en een enkele view wordt gebruikt voor het aanspreken van de models, het serialiseren van gegevens die via een model en ORM uit de database en het retourneren van de geserialiseerde gegevens aan de actor of eindgebruiker. De reden dat er voor een enkele view is gekozen en niet voor een splitsing van views op entiteit niveau is dat de meeste functies in de view dezelfde methode gebruiken

om models te benaderen waardoor code makkelijk hergebruikt kon worden en op die manier een tijdswinst behaald kon worden.

9.6.3 Hyperic web service model dictionary

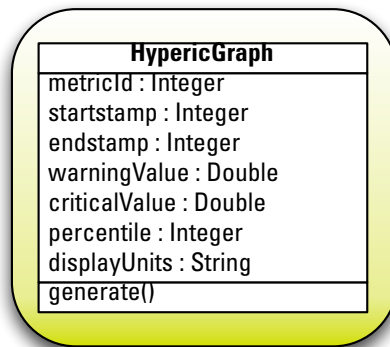
Klasse: Platform	
Een platform staat gelijk aan een fysieke machine. De fysieke machine kan alles zijn, zolang het maar via een netwerk kan worden aangesproken. Hierbij kan gedacht worden aan bijvoorbeeld routers, switches, PDUs of apparatuur draaiend op Linux, Windows, Solaris of OS X.	
Attribuut naam	Omschrijving
<i>+ name: String</i>	De naam van het platform binnen Hyperic. De naam is nodig om duidelijk te maken wat het onderwerp van de grafiek is en is onderdeel van de requirement dat de grafieken uiterlijk aantrekkelijk moeten zijn. De naam van het platform heeft een waarde van het type string en is uniek.

Tabel 9.6: Dictionary van de Platform klasse

Verdere uitwerking van de modeldictionary is te vinden in appendix B.

9.6.4 Grafieken generator tool domein klassediagram

In figuur 9.5 is een domein klassediagram te zien behorende bij de Grafieken generator tool. Dit domein klassediagram is gebaseerd op de gestelde requirements voor de grafieken generator tool, zoals die ook te terug te lezen zijn in de hoofdstukken 8.4 en 8.6.1.

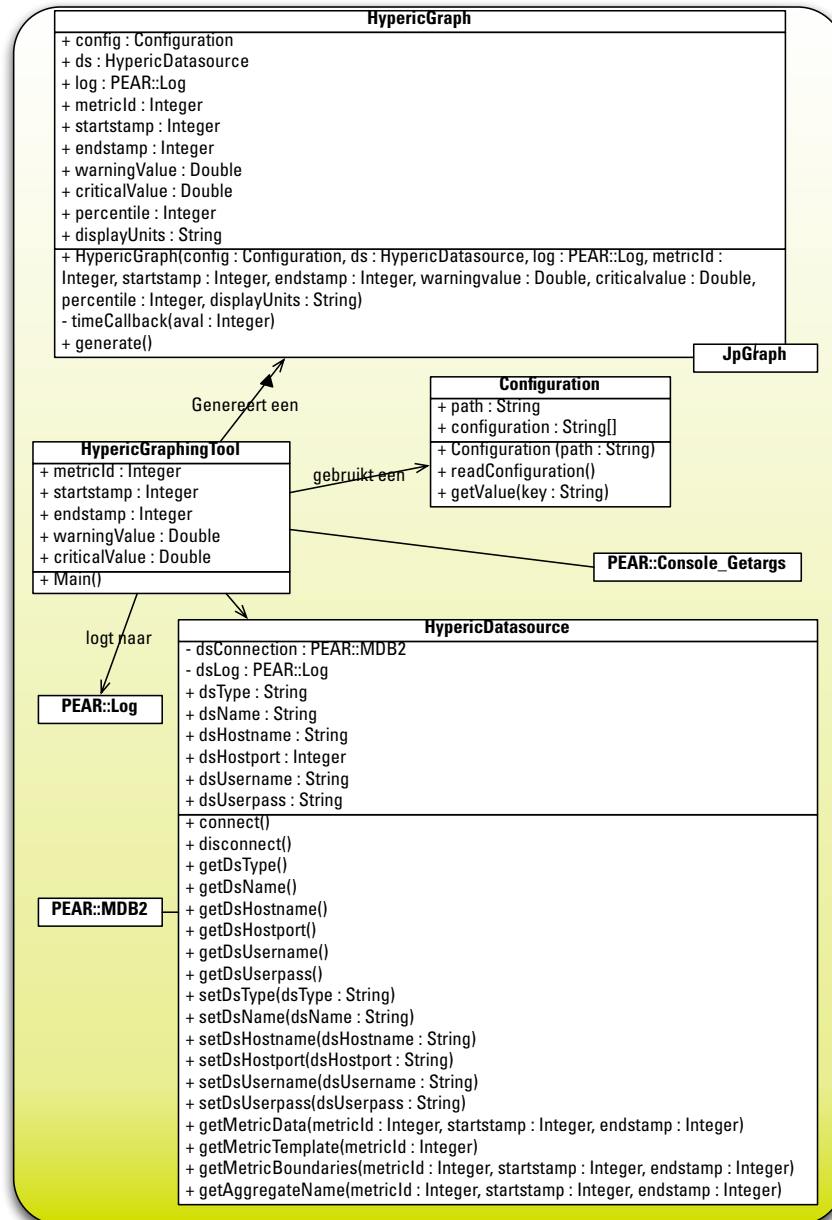


Figuur 9.5: Grafieken generator tool domein klassediagram

De klasse die bovenstaand domeinmodel weergeeft laat precies zien wat de grafieken generator tool doet: het genereert grafieken. De reden dat het uitgewerkte model uitgebreider is, is dat de omliggende klassen alleen functionaliteit bieden die meewerken aan de uiteindelijke te genereren grafiek. De `HypericGraph` klasse gebruikt de klassen in het uitgewerkte model om configuraties uit te lezen, algemene gegevens over een meting te verzamelen en metrische gegevens van een meting te verzamelen. Het uiteindelijke interpreteren van de metrische gegevens en het genereren van een visuele weergave in een grafiek gebeurt door de `HypericGraph` klasse.

9.6.5 Grafieken generator tool uitgebouwd klassediagram

Figuur 9.6 laat een uitgebouwd klassediagram van de grafieken generator tool zien. Dit uitgebouwde klassediagram is gebaseerd op het domein klassediagram.



Figuur 9.6: Grafieken generator tool klassediagram

9.6.6 Grafieken generator tool model dictionary

Klasse: HypericGraph	
Een instantie van de HypericGraph klasse representeert een grafiek.	
Attribuut naam	Omschrijving
<i>+ config: Configuration</i>	Een instantie van de Configuratie klasse. Geeft toegang tot de uitgelezen waarden uit het configuratiebestand om zo benodigde configuratie waarden op de halen.
<i>+ ds: HypericDatasource</i>	Een instantie van de HypericDatasource klasse. Geeft de grafiek toegang tot de datasource om zo de benodigde data uit de datasource op te halen.
<i>+ log: PEAR::Log</i>	Een instantie van de PEAR::Log klasse. Geeft de mogelijkheid om messages te loggen naar een bepaald target.
<i>+ metricId: Integer</i>	Het id van de gewenste meting. De waarde van dit attribuut wordt als argument op de command line meegegeven en komt vanuit de HypericGraphingTool klasse in de HypericGraph klasse terecht.
<i>+ startstamp: Integer</i>	De starttijd van de grafiek. Elke grafiek plot de metrische data binnen een bepaald tijdframe (tussen twee data). De startstamp geeft het begin van dat tijdframe aan.
<i>+ endstamp: Integer</i>	De eindtijd van de grafiek. Zie ook startstamp hierboven. De endstamp geeft het eind van die tijdsspanne aan.
<i>+ warningValue: Double</i>	De waarde waarop de lijn op de grafiek moet worden getekend die aangeeft wanneer een waarschuwings waarde wordt overschreden. De lijn wordt altijd getekend, ook als dat betekent dat de zichtbaarheid van de plotlijn verminderd.
<i>+ criticalValue: Double</i>	De waarde waarop de lijn op de grafiek moet worden getekend die aangeeft wanneer een kritieke waarde wordt overschreden. De lijn wordt altijd getekend, ook als dat betekent dat de zichtbaarheid van de plotlijn verminderd.
<i>+ percentile: Integer</i>	De waarde voor het <i>x</i> -de percentiel.
<i>+ displayUnits : String</i>	De eenheid die op de assen van de grafiek gebruikt zullen worden en waarin de waarden van de grafiek geconverteerd zullen worden.
Methode naam	Omschrijving
<i>+ HypericGraph(config : Configuration, ds : HypericDatasource, log : PEAR::Log, metricId : Integer, startstamp : Integer, endstamp : Integer, warningValue : Double, criticalValue : Double, percentile : Integer, displayUnits : String)</i>	De constructor van de HypericGraph klasse. Neemt een instantie van Configuration, HypericDatasource en PEAR::Log klassen als argument. Daarnaast worden het metricId, de startstamp, endstamp, warningvalue, criticalvalue, percentile en displayUnits meegegeven.

Methode naam	Omschrijving
- <i>timeCallback(aval : Integer)</i>	Deze methode neemt een UNIX timestamp als argument en zet deze om naar een voor mensen leesbare datum in een dd-mm-jjjj uu:mm of een uu:mm formaat, afhankelijk van de tijdsperiode. Deze functie wordt alleen intern aangeroepen door de generate() methode.
+ <i>generate()</i>	Deze methode genereert een grafiek op basis van het metricId, de startstamp en endstamp, de warning- en criticalValue en de percentile en displayUnits waarden.

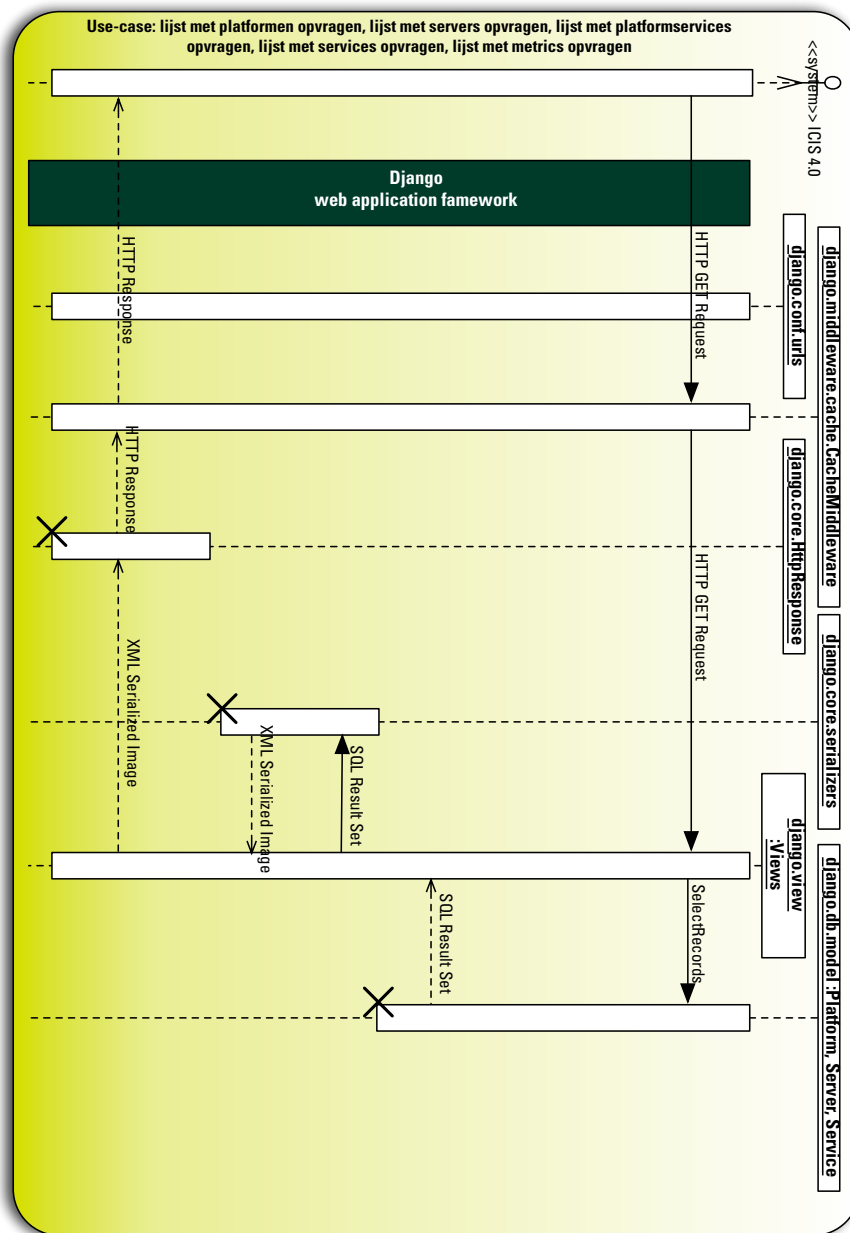
Tabel 9.7: Dictionary van de HypericGraph klasse

Verdere uitwerking van de modeldictionary is te vinden in appendix B.

9.7 Sequentiediagrammen

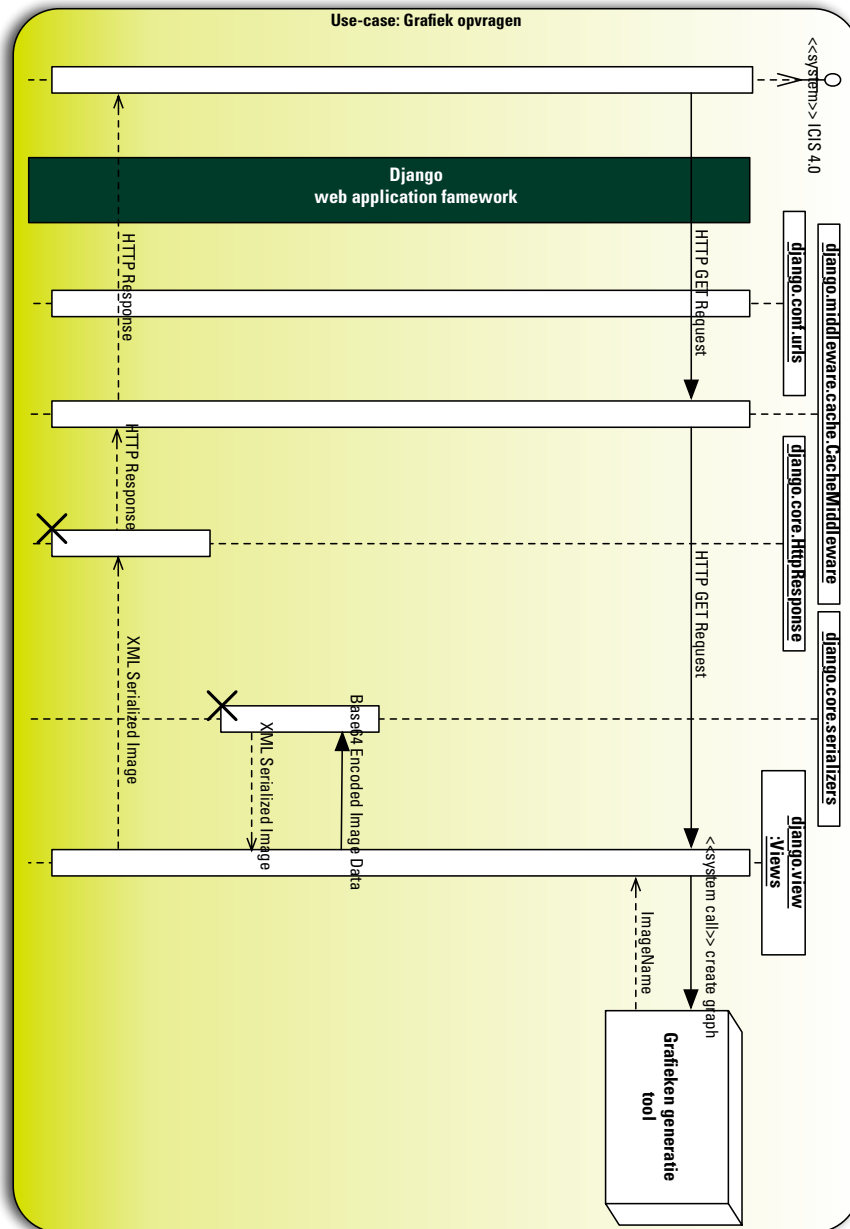
9.7.1 Hyperic web service sequentiediagram

Figuur 9.7 geeft het sequentiediagram weer van alle 'Lijst met * opvragen' use-cases. In het diagram wordt duidelijk gemaakt hoe een request van de actor naar de Hyperic web service verloopt en welke objecten er worden aangemaakt en destroyed. Te zien is dat er voor het ophalen van een lijst geen aanroep naar de grafieken generator tool gedaan hoeft te worden. Dit diagram is opgebouwd aan de hand van use-cases en het klassediagram en is toegespitst op het framework van Django. De meeste klassen in onderstaand diagram zijn onderdeel van Django. De Views klasse en alle model klassen zijn toegevoegd voor de Hyperic web service.



Figuur 9.7: Hyperic web service sequentiediagram 1

Figuur 9.8 geeft een sequentiediagram weer van de 'Grafiek opvragen' use-case. Te zien is dat er een system call wordt gedaan naar de grafieken generator tool en dat er geen aanroepen worden gedaan naar models en er dus geen gegevens uit de Hyperic database worden verzameld.



Figuur 9.8: Hyperic web service sequentiediagram 2

Hoofdstuk 10

Bacula: open-source multi platform network backup systeem

"It comes by night and sucks the vital essence from your computers". En dat is precies wat Bacula doet. Bacula is de backup software die door IC&S in gebruik is om elke nacht de vitale gegevens van ongeveer 150 servers te backupperen. Het is geschikt voor vele platformen, kan meerdere backups tegelijk maken naar tape, dvd, cdrom en harddisk, is open-source en gratis in gebruik. Het is een goed alternatief voor dure commerciële backup oplossingen. Dit hoofdstuk leidt in op iteratie 2: de Bacula web service.

10.1 3 componenten

Bacula bestaat uit drie componenten, te weten de director, de storage daemon en de file daemon. Elk component heeft zijn eigen rol in het backup proces. Per component staat hieronder beschreven wat zijn rol is.

- **Director** - de director redigeert het complete backup proces. De director houdt de configuratie bij, roostert de te draaien jobs in en geeft de opdracht aan de file daemon om een job te starten. Daarnaast houdt de director bij wanneer en met wat voor status een job eindigt.
- **Storage daemon** - de storage daemon is het component waar de te backupperen data heen wordt gestuurd. De file daemon slaat de te backupperen data op de juiste manier op. In het geval van een restore verstuurt de storage daemon de te herstellen gegevens naar de juiste file daemon.
- **File daemon** - de file daemon is het component dat de opdracht om een backup te starten ontvangt. In dat geval begint de file daemon met indexeren welke bestanden verstuurt zullen gaan worden, om vervolgens de geïndexeerde bestanden naar de storage daemon te sturen. In het geval van een restore ontvangt de file daemon de te herstellen bestanden en plaatst ze op een vooraf gekozen locatie. De file daemon wordt op elke server geïnstalleerd die deel neemt aan het backup proces.

10.2 Console

De console biedt een manier om interactie te hebben met het director onderdeel van Bacula. Middels de console kan o.a. het rooster worden aangepast, kunnen backups handmatig (d.w.z. buiten het rooster om) worden gestart en kunnen restores worden gestart. De console hoort niet bij de drie hoofdcomponenten van Bacula, omdat Bacula zonder console ook zou kunnen opereren. De console is voor een beheerder een onmisbare tool om de controle over Bacula te behouden. De console biedt een alternatieve shell met een prompt, dat gebruikt kan worden allerlei commando's uit te voeren.

10.3 Catalog

Zoals de meeste backup systemen heeft ook Bacula een zogenaamde catalog. In de catalog worden allerlei gegevens bijgehouden over backup jobs die gedraaid zijn of die op dit moment nog draaien. Daarnaast wordt bijgehouden wat er gebackuppeld is en waar het gebackuppeld is. De gegevens zijn nodig op het moment dat er een restore moet worden uitgevoerd. Op basis van de catalog gegevens weet Bacula precies waar (op welk backup volume) het een te herstellen bestand heeft gelaten en waar het dus moet zoeken om een succesvol herstel uit te voeren. Daarnaast kan de catalog gebruikt worden om statistische overzichten te creëren. Zie hiervoor ook hoofdstuk 10.

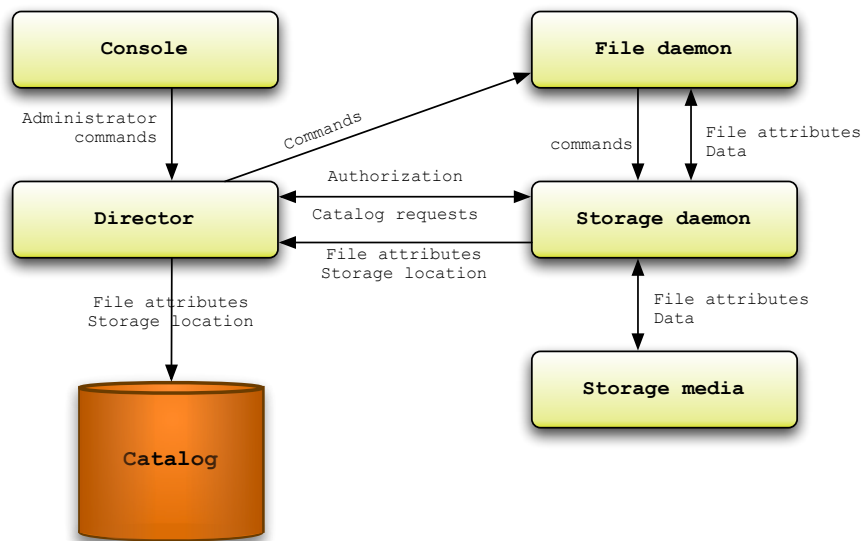
De catalog waar Bacula gebruik van maakt komt in de vorm van een relationele database. Bacula kan daarvoor gebruik maken van drie verschillende open-source DBMS'en: sqlite, MySQL of PostgreSQL, waarbij de eerste alleen wordt aanbevolen voor gebruik in kleinere omgevingen (<15 backup clients).

De gegevens in de catalog worden geen onbeperkte tijd bewaard. Na een van tevoren vastgestelde retentietijd worden de bewaarde gegevens uit de catalog verwijderd. Reden hiervoor is dat de catalog te traag zou worden en te snel in grootte zou groeien als de bewaarde gegevens nooit uit de catalog zouden worden verwijderd. Figuur 10.1 laat de mogelijke vormen van interactie tussen verschillende Bacula componenten zien. Figuur 10.2 geeft een overzicht van de aanwezige componenten binnen de Bacula architectuur.

10.4 Bacula Statistiek: openheid van (backup)zaken

Zoals is aangegeven maakt Bacula gebruik van een catalog om allerlei gegevens bij te houden over backup jobs die gedraaid zijn of op dit moment nog draaien en wordt de catalog op basis van retentietijden periodiek opgeschoond en daarmee ontdaan van gegevens die niet meer valide zijn. Bacula statistiek slaat een deel van de gegevens op in een andere relationele database, de statistieken database, voordat ze uit de catalog verwijderd worden en creëert daarmee een backup historie die alleen relevante informatie bevat. De overgenomen gegevens beperken zich tot de volgende:

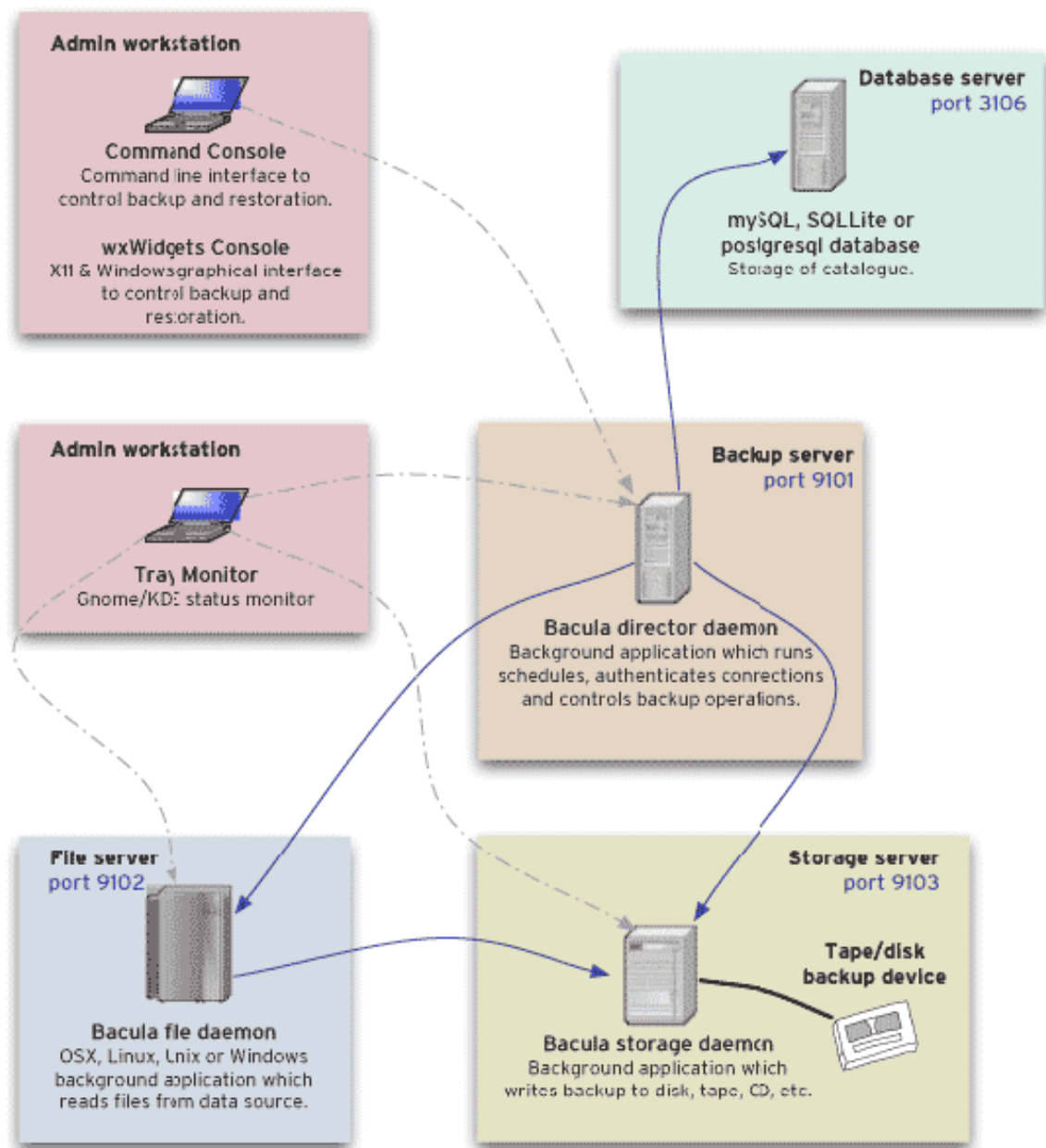
- **Clients** - van elke server die door Bacula gebackuppeld wordt, wordt een record bewaard in de statistieken database om elke gedraaide backup job aan een server of client te kunnen koppelen.
- **Backup jobs** - van elke backup job, of deze nu gefaalt of mislukt is, wordt een record bewaard. Door een md5 hash te maken van de backup job kan hoge garantie gegeven worden dat elk record uniek is.



Figuur 10.1: Bacula interactie tussen componenten

- **Logmessages** - Gebeurtenissen die tijdens een backup job plaatsvinden en een relatie hebben met een backup job worden als zodanig opgeslagen.

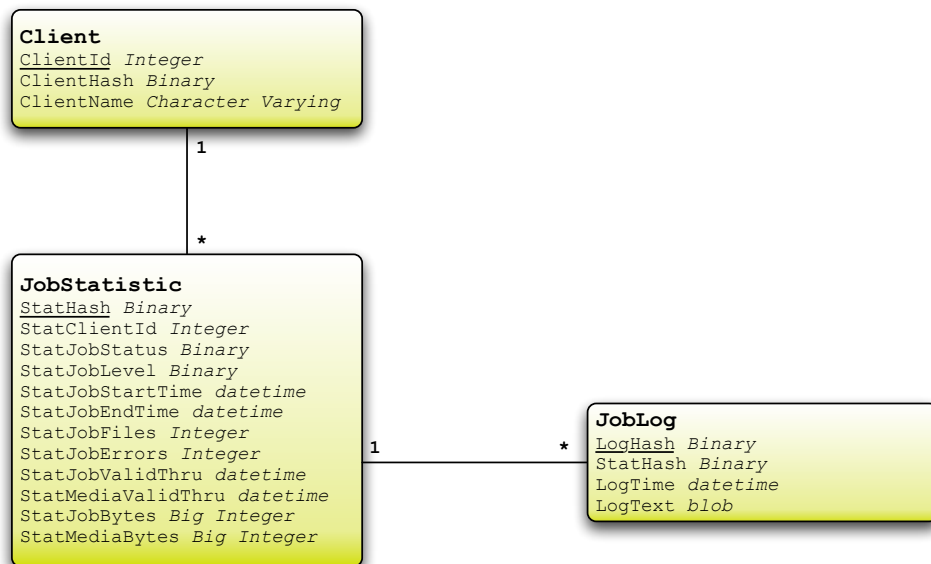
De gegevens in de statistieken database kunnen vervolgens worden aangesproken middels Bacula Statistiek: een web interface die grafieken en statistische informatie genereert op basis van de gegevens in de statistieken database. Per backup client kan worden ingezien hoeveel data de client in beslag nam binnen een tijdspanne en hoe groot de backup job(s) waren binnen een tijdspanne. Doordat de gegevens direct vanaf Bacula afkomstig zijn, zijn de gegenereerde overzichten betrouwbaar. Figuur 10.3 laat het entiteit relatie diagram zien dat bij Bacula Statistiek hoort.



Note that these applications may actually run on fewer machines than shown here. You could run everything on one machine if you only wanted to back up a local disk to a local tape or disk.

Port numbers are the defaults and can be changed.

Figuur 10.2: Bacula componenten architectuur en interactie



Figuur 10.3: Bacula Statistiek ERD

Hoofdstuk 11

Bacula web service

11.1 Requirements

Uit gesprekken met de Bart van de Garde die tijdens de inception fase zijn gevoerd zijn onderstaande requirements naar voren gekomen. Deze requirements zijn nadrukkelijk alleen van toepassing op de 2e iteratie van het afstudeer project.

Bacula web service

1. Gebaseerd op REST.
2. Moet dezelfde grafieken kunnen leveren als de huidige Bacula Statistiek applicatie.
3. Moet dezelfde job overzichten en job details kunnen weergeven als de huidige Bacula Statistiek applicatie.
4. Moet kunnen omgaan met parameters als startdate en enddate.
5. Moet een grafiek als plaatje retourneren, dus geen metrische data.
6. Moet in staat zijn grafieken te cachen om de load op de database laag te houden.
7. Keuze voor software om de web service te implementeren, alsmede de keuze voor programmeertaal staat vrij.
8. De software die de web service implementeert moet kunnen draaien op de Debian Linux distributie.

11.2 Planning

Onderstaande tabel toont een uitgebreide planning voor de Bacula web service. Ook deze planning is opgedeeld in de vier fases die een UP iteratie bevat en voor elke fase zijn de specifieke taken en hun geplande doorlooptijd beschreven.

Inception fase (24-12-2007 t/m 28-12-2007)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Beschrijven Bacula statistieken applicatie data-structuur	24-12-2007	24-12-2007	5
Onderzoek en vaststellen requirements	24-12-2007	24-12-2007	3
Modelleren use-cases	27-12-2007	27-12-2007	8
Opstellen inception rapport	28-12-2007	28-12-2007	10
		subtotaal	26
<i>Opmerking: 1e en 2e kerstdag vallen op 25 en 26 december 2007. Deze dagen zijn niet ingeplanned als werkdagen.</i>			

Tabel 11.1: Planning inceptie fase, 2e iteratie

Elaboration fase (31-12-2007 t/m 04-01-2008)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Modelleren Bacula web service in UML klassediagram	31-12-2007	31-12-2007	4
Modelleren Bacula web service in UML sequentie-diagram	31-12-2007	02-01-2008	8
Modelleren Bacula web service in UML deploymentdiagram	02-01-2008	02-01-2008	4
Opmaken elaboration rapport	03-01-2008	04-01-2008	16
		subtotaal	32
<i>Opmerking: op 1 januari valt nieuwjaarsdag. Deze dag is niet ingeplanned als werkdag.</i>			

Tabel 11.2: Planning elaboration fase, 2e iteratie

Construction fase (07-01-2008 t/m 18-01-2008)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Constructie Bacula web service	07-01-2008	15-01-2008	56
Constructie simpele REST web service client	16-01-2008	17-01-2008	12
Opmaken construction rapport	17-01-2008	18-01-2008	12
		subtotaal	80

Tabel 11.3: Planning construction fase, 2e iteratie

Ontwikkeling van RESTful web services voor ICIS 4.0

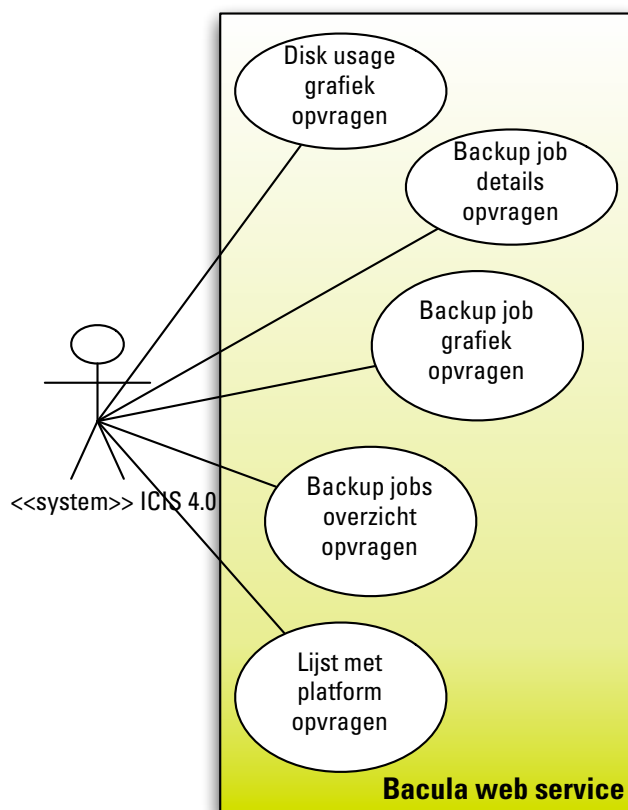
Transition fase (21-01-2008 t/m 25-01-2008)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Uitvoeren ad-hoc tests door opdrachtnemer	21-01-2008	21-01-2008	6
Opmaken handleiding gebruiker	21-01-2008	22-01-2008	6
Opmaken handleiding ontwikkelaar	22-01-2008	23-01-2008	6
Opmaken transitie rapport	24-01-2008	25-01-2008	8
Oplevering producten + overleg	25-01-2008	25-01-2008	1
		subtotaal	27

Tabel 11.4: Planning transition fase, 2e iteratie

Totaal aantal uur iteratie 2	165
-------------------------------------	------------

11.3 Use-cases

Het diagram in figuur 11.1 toont het use-case diagram voor de Bacula web service.



Figuur 11.1: Bacula web service use-case diagram

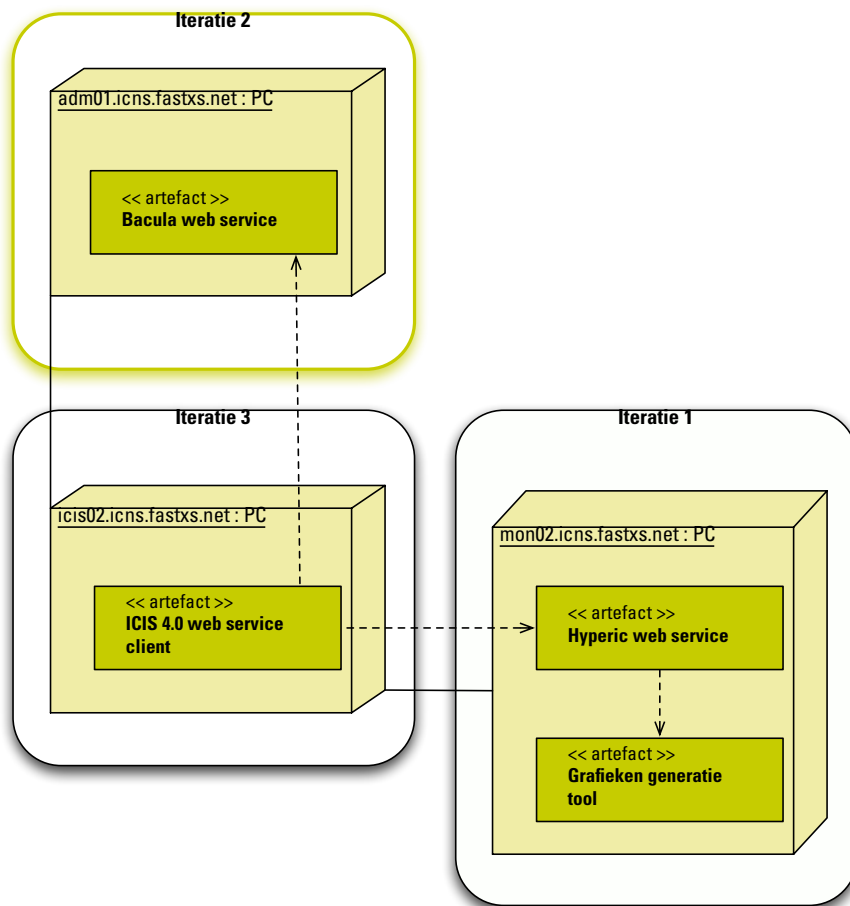
De table in tabel 10.2 beschrijft de use-case backup jobs overzicht opvragen. Verdere use-case beschrijvingen zijn als appendix bijgevoegd in appendix A.

11.4 Deployment

Het deploymentdiagram verdeelt losse omgevingen in zogenaamde nodes en de systemen die daarop draaien in artefacten. Op die manier is de verdeling tussen de verschillende artefacten goed te zien en wordt duidelijker wat de relaties zijn tussen elk van de systemen. Het diagram is te zien in figuur 11.2.

Elk blok met ronde hoeken in bovenstaand deploymentdiagram geeft weer in welke iteratie de node die daar in zit in behandeling wordt genomen. Het blok met de groene rand is het blok dat nu in behandeling is. Elke kubus is een node en elk rechthoek dat daar in zit is een artefact.

Te zien is dat de Bacula web service op de adm01.icns.fastxs.net node komt te draaien, dezelfde node als waar de Bacula statistieken applicatie op draait. De Bacula statistieken applicatie is alleen toegankelijk voor IC&S medewerkers; de Bacula web service zal een interface bieden voor toegang vanaf de ICIS 4.0 klantenportal. In dit project is het zo dat de Bacula web



Figuur 11.2: Bacula web service deployment diagram

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Backup jobs overzicht opvragen
Samenvatting	Een overzicht wordt opgevraagd van de gedraaide backup jobs in een bepaalde periode. De periode wordt bepaald aan de hand van user input of aan de hand van een standaardberekening in het geval dat er geen user input aanwezig is.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform is bekend. Daarnaast is de gewenste startdatum, starttijd, stopdatum en stopdatum bekend.
Beschrijving	1. ICIS 4.0 geeft via een URI de opdracht een overzicht van gedraaide backup jobs in een bepaalde periode te genereren 2. De Bacula web service genereert een backup job overzicht vanuit de statistieken database aan de hand van de binnengekomen parameters
Uitzonderingen	[Geen backups] Als er geen backups gedraaid zijn in de opgegeven periode, dan zal daar een melding van worden geretourneert.
Resultaat	Een overzicht van de gedraaide backup jobs in de aangegeven periode.

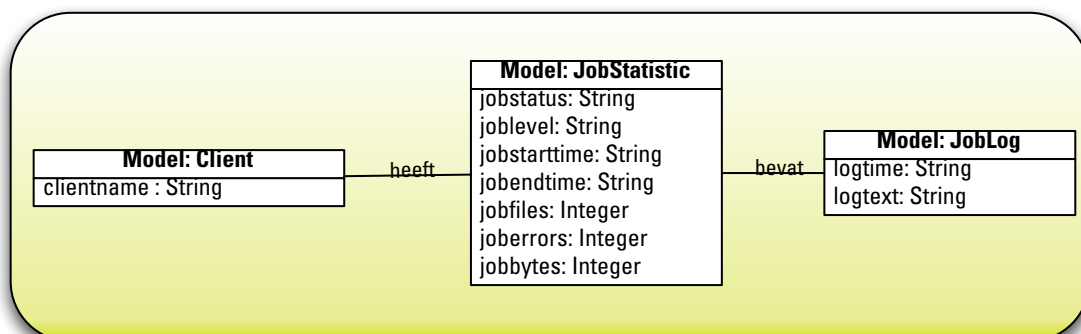
Tabel 11.5: Use-case Backup jobs overzicht opvragen

service wordt aangesproken door een web service client die in ICIS 4.0 zal worden gebouwd, maar in theorie is het mogelijk de Bacula web service aan te spreken met een HTTP request vanuit elke denkbare bron.

11.5 Klassediagrammen

11.5.1 Bacula web service domein klassediagram

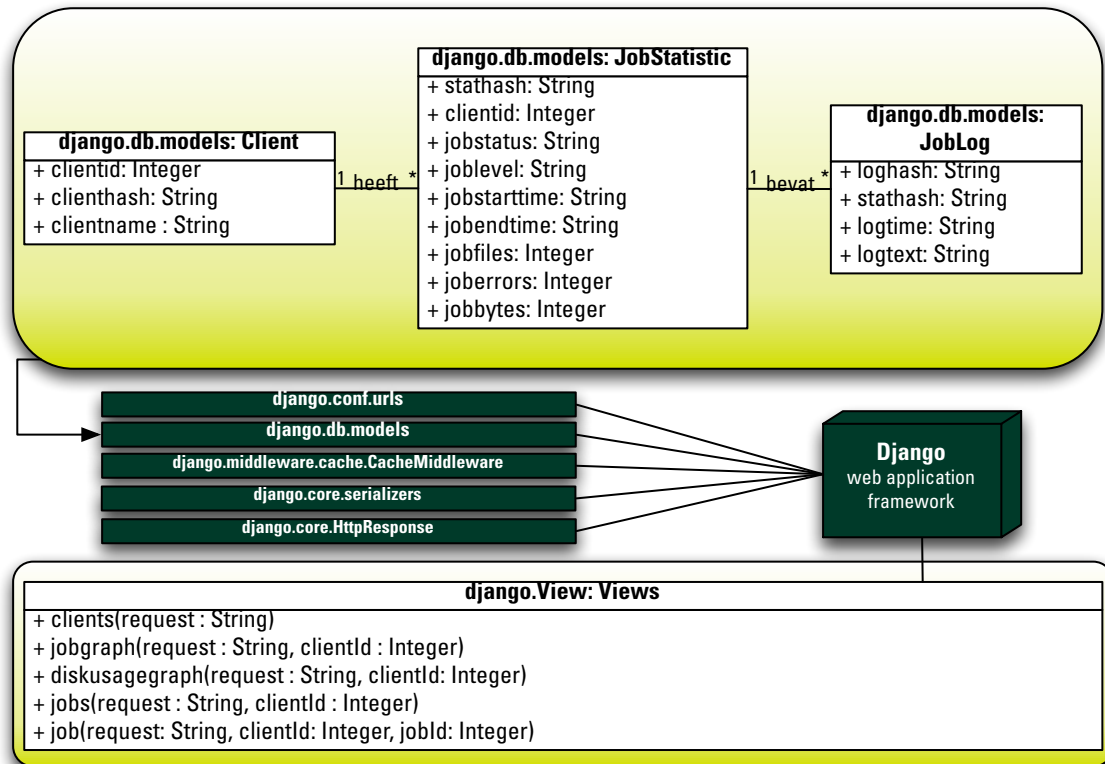
In figuur 11.3 is een domein klassediagram te zien behorende bij de Bacula web service. Het domein klassediagram is toegespitst op het gebruik van Django, waarbij gebruik wordt gemaakt van het MVC ontwerp patroon.



Figuur 11.3: Bacula web service domein klassediagram

11.5.2 Bacula web service klassediagram

In figuur 11.4 is een uitgebreud klassediagram te zien. Het uitgebreude diagram is gebaseerd op het diagram in figuur 11.3



Figuur 11.4: Bacula web service klassediagram

11.5.3 Bacula web service model dictionary

Klasse: JobStatistic	
Een job statistic staat gelijk aan een uitgevoerde backup job, onafhankelijk van het resultaat van de job. De naam JobStatistic is een beetje ongelukkig gekozen ten tijde van de ontwikkeling van de Bacula statistieken applicatie. Elke job statistic bevat een aantal feiten over een backup job die gebruikt kunnen worden bij het maken van grafieken en statistieken.	
Attribuut naam	Omschrijving
<i>+ stathash: String</i>	Het identificerende attribuut van de job statistic. In plaats van een oplopende integer waarde is gekozen voor een hash als identificerende waarde. De hash wordt gemaakt op basis van variabelen als job naam en tijd, om te kunnen garanderen dat elke job statistic los te identificeren valt.
<i>+ clientid: Integer</i>	Een verwijzing naar een instantie van de Client klasse.
<i>+ jobstatus: String</i>	De uiteindelijke status van de backup job. De status is uitgedukt in een enkel karakter waarbij er 5 mogelijke statussen zijn. De betekenis van elke mogelijkheid is uitgelegd in de handleiding van de Bacula web service en in de handleiding van Bacula zelf.
<i>+ joblevel: String</i>	De level / niveau waerop de backup job gedraait heeft. De level is uitgedrukt in een enkel karakter waarbij er 3 mogelijkheden zijn. Hierbij moet dan worden gedacht aan levels als: Full, Incremental en Differential. De precieze uitleg van elke level is uitgelegd in de handleiding van de Bacula web service en in de handleiding van Bacula zelf.
<i>+ jobstarttime: String</i>	De tijd waarop de job gestart werd.
<i>+ jobendtime: String</i>	De tijd waarop de job eindigde.
<i>+ jobfiles: Integer</i>	Het aantal files dat door de backup job gebackuppeld werd.
<i>+ joberrors: Integer</i>	Het aantal errors die tijdens de backup job naar voren kwamen.
<i>+ jobbytes: Integer</i>	Het aantal bytes dat de backup job heeft verbruikt.

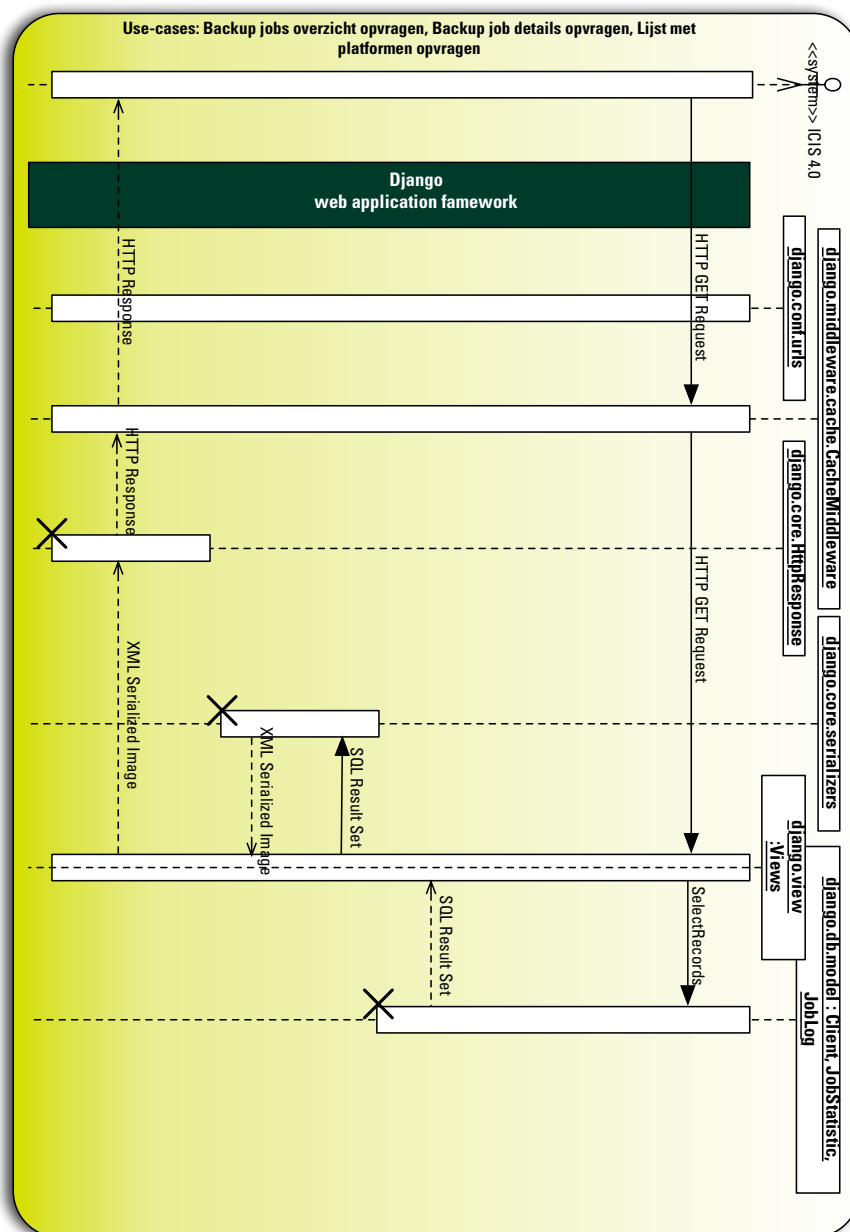
Tabel 11.6: Dictionary van de JobStatistic klasse

Verdere uitwerking van de model dictionary is te vinden in appendix B.

11.6 Sequentiediagrammen

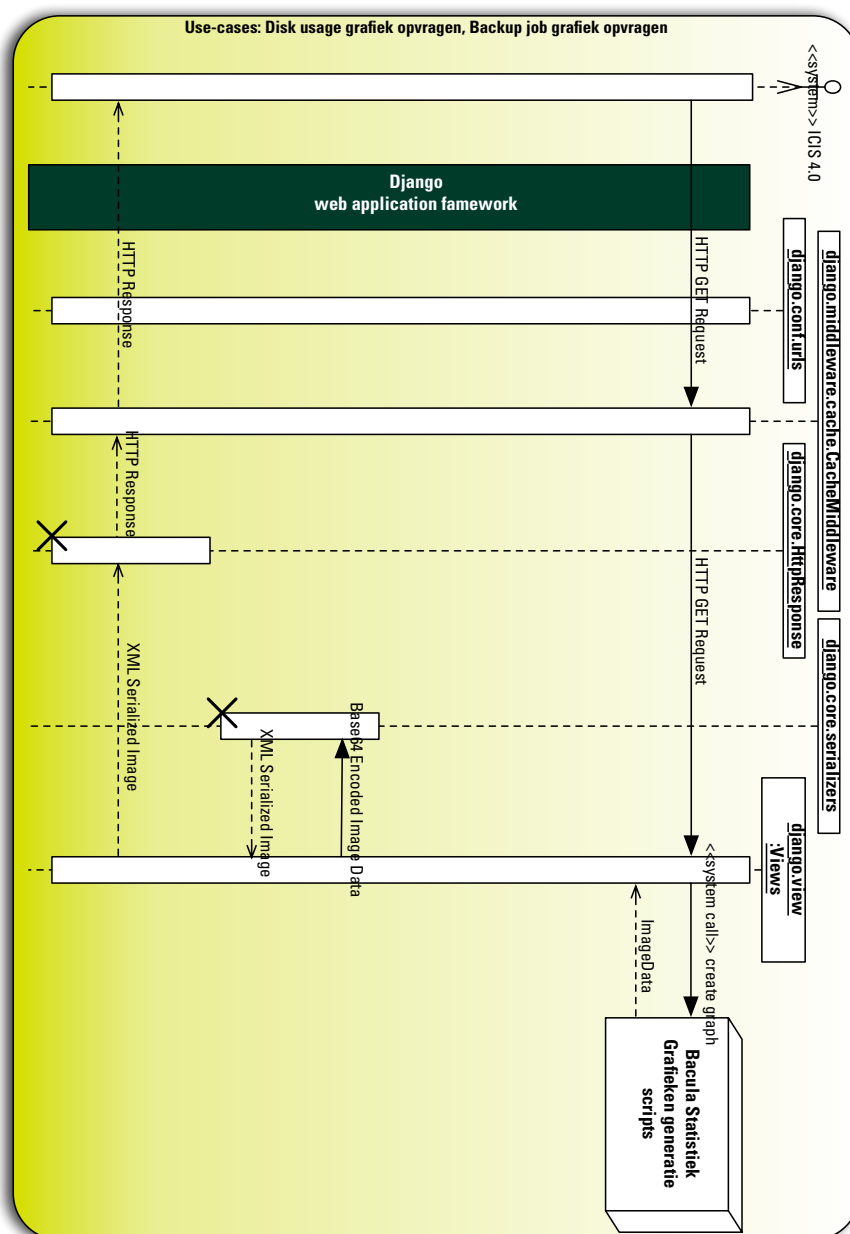
Dit hoofdstuk beschrijft het sequentiediagram van de Bacula web service. Het sequentiediagram geeft de interactie weer tussen de verschillende klassen binnen een applicatie. Op die manier wordt er een bepaalde hiërarchie binnen de klassen duidelijk en worden de onderlinge verbanden tussen de verschillende klassen verduidelijkt.

De use-cases disk usage grafiek opvragen en backup job grafiek opvragen zijn samengevoegd in een enkel sequentie diagram, omdat de operatie achter de schermen voor elk van de use cases hetzelfde behelst. Figuur 11.5 laat dit sequentie diagram zien. De meeste klassen in onderstaand diagram zijn onderdeel van Django. De Views klasse en alle model klassen zijn toegevoegd voor de Bacula web service. De Bacula Statistiek grafieken generatie scripts waren al bestaande items.



Figuur 11.5: Bacula web service sequentiediagram 1

De use cases backup jobs overzicht opvragen, backup job details opvragen en lijst met platformen opvragen zijn ook in een enkel sequentie diagram verwerkt, omdat ook hier geldt dat de uiteindelijke interactie tussen klassen voor elke use case hetzelfde is. Alleen de Views klasse is toegevoegd voor de Bacula web service. Figuur 11.6 laat dit diagram zien.



Figuur 11.6: Bacula web service sequentiediagram 2

Hoofdstuk 12

Transition van web services

12.1 Testen

De binnen IC&S gebruikelijke manier van testen van intern ontwikkelde applicaties die op zichzelf geen commerciële doeleinden hebben, is doormiddel van het uitvoeren van een functionele test volgens de ad hoc test methode. Dat wil zeggen dat de software zonder enig testplan wordt getest. De tester zoekt daarbij op basis van improvisatie naar bugs in de software via alle mogelijke en aannemelijke wegen. Belangrijke bugs kunnen op die manier snel worden geïdentificeerd.

Om de test uit te voeren is gebruik gemaakt van een sumiere web service client geschreven in Java. De client is in staat alle geïmplementeerde functionaliteit van de web service aan te spreken en het resultaat ervan weer te geven. Eventuele problemen kunnen zo direct worden aangepakt en in de meeste gevallen worden opgelost.

Een probleem dat bij het testen naar voren kwam deed zich voor bij het opbouwen van een tree met daarin een hiërarchische opbouw van alle platformen, platformservices, servers, services en de daarbij behorende metingen. Omdat RESTful web services stateless zijn en elke request dus als losse transactie wordt gezien, moet je er rekening mee houden dat de consumerende applicatie ook elke request als losse transactie moet zien. Bij het opbouwen van de tree met de hiërarchie van platformen, servers, metrics etc. werd dus voor elk platform een lijst opgevraagd met metrics, server en platformservices, werd voor elke server een lijst met metrics en services opgevraagd etc. etc. De client deed er gemiddeld 15 minuten over om de tree te vullen met objecten.

Er is geen echte oplossing voor dit probleem. De web service werkt goed, alleen de consumerende applicatie maakte er geen juist gebruik van. Door de navigatie zo aan te passen dat een gebruiker zelf zijn weg door de hiërarchie kiest kan veel overbodige load vermeden worden. Bij elke keuze wordt bepaald welke requests aan de web service uitgevoerd dienen te worden. Op die manier wordt alleen relevante data opgevraagd en worden de requests ook over een groter tijdsbestek verdeelt waardoor wachttijden acceptabel worden.

12.2 Handleiding

Om overdracht en gebruik makkelijker te maken zijn er een handleiding voor gebruikers en een handleiding voor ontwikkelaars geschreven. Verschil tussen de twee handleidingen is dat de handleiding voor gebruikers in gaat op de installatie en het aanspreken van de web service, terwijl de handleiding voor ontwikkelaars in gaat op de gebruikte techniek en een uitwerking

van de aangeboden functionaliteit biedt. Daarnaast toont de handleiding voor ontwikkelaars ook voorbeeldcode die laat zien hoe de web service geconsumeerd kan worden en is de handleiding vooral bedoelt voor het ontwikkelen van consumerende applicaties en niet zo zeer voor uitbreiding van de web service zelf. De documentatie die daarvoor gebruikt kan worden wordt bijgehouden tijdens de diverse iteraties die nog zouden kunnen volgen.

12.3 Overdracht

Omdat dit mijn project is binnen IC&S en ik na de afstudeer werkzaamheden werkzaam zal blijven bij IC&S is het hoogst waarschijnlijk dat wijzigingen of doorontwikkeling aan de web services ook door mij opgepakt zullen worden. De documentatie die opgeleverd is tijdens de verschillende fases binnen de iteraties kan gebruikt worden bij wijzigingen of doorontwikkeling door derden.

Hoofdstuk 13

ICIS 4.0 web service client

13.1 Requirements

Tijdens overleg met Bart v.d. Garde voor de aanvang van het afstuderen en tijdens de inception fase zijn onderstaande requirements naar voren gekomen. Deze requirements zijn alleen van toepassing op de 3e iteratie van het afstudeer project.

1. Moet geïmplementeerd worden in Ruby on Rails, omdat de rest van ICIS 4.0 daar ook in geïmplementeerd is.
2. Moet op een veilige manier kunnen communiceren met de ontwikkelde Hyperic web service en de ontwikkelde Bacula web service.
3. De bijbehorende grafische interface moet aantrekkelijk zijn en passen bij de rest van ICIS 4.0
4. Grafieken moeten alleen op aanvraag gegenereerd worden. Alleen bij de uitdrukkelijke opdracht een grafiek te genereren moet de opdracht daartoe verzonden worden. Op die manier wordt voorkomen dat er grafieken gegenereerd worden op het moment dat de wens daartoe eigenlijk niet bestaat, met als positief gevolg dat de laadtijden van ICIS 4.0 en de load op de Hyperic- en Bacula web service binnen de perken wordt gehouden.

De aantrekkelijkheid van de bij de web service client behorende grafische interface is subjectief. Bij IC&S is het de gewoonte doormiddel van trial & error de grafische interface in kleine stappen aan te passen en zodoende tot een geheel te komen dat acceptabel is. De reden voor het gebruik van de trial & error methode is dat het ontwerpen en optimaliseren van grafische interfaces buiten de core-business van IC&S valt en de kennis daarover dus beperkt is. Trial & error biedt dan een methode waarmee toch acceptabel product ontwikkeld kan worden.

13.2 Planning

Onderstaande tabellen tonen een uitgebreide planning voor de ICIS 4.0 web service client. De planning is opgedeeld in de vier fases die een UP iteratie bevat en voor elke fase zijn de specifieke taken en hun geplande doorlooptijd beschreven.

Inception fase (28-01-2008 t/m 01-02-2008)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Onderzoek en vaststellen requirements	28-01-2008	28-01-2008	5
Modelleren use-cases	28-01-2008	29-01-2008	10
Opstellen inception rapport	29-01-2008	31-01-2008	16
Verdiepen in Ruby on Rails	28-01-2008	01-02-2008	9
		subtotaal	40

Tabel 13.1: Planning inceptie fase, 3e iteratie

Elaboration fase (04-02-2008 t/m 08-02-2008)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Modelleren ICIS 4.0 web service client in UML klas-sediagram	04-02-2008	04-02-2008	8
Modelleren ICIS 4.0 web service client in UML sequentiediagram	05-02-2008	05-02-2008	8
Modelleren ICIS 4.0 web service client in UML deploymentdiagram	06-02-2008	06-02-2008	4
Opmaken elaboration rapport	06-02-2008	07-02-2008	8
Verdiepen in Ruby on Rails	04-02-2008	08-02-2008	12
		subtotaal	40

Tabel 13.2: Planning elaboration fase, 3e iteratie

Construction fase (11-02-2008 t/m 22-02-2008)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Constructie ICIS 4.0 web service client	11-02-2008	21-02-2008	72
Opmaken construction rapport	22-02-2008	22-02-2008	8
		subtotaal	80

Tabel 13.3: Planning construction fase, 3e iteratie

Transition fase (25-02-2008 t/m 29-02-2008)			
Omschrijving	Startdatum	Einddatum	Aantal uur
Uitvoeren ad-hoc tests door opdrachtnemer	25-02-2008	26-02-2008	10
Opmaken handleiding gebruiker	26-02-2008	26-02-2008	6
Opmaken handleiding ontwikkelaar	27-02-2008	27-02-2008	8
Opmaken transitie rapport	28-02-2008	28-02-2008	8
Oplevering producten + overleg	29-02-2008	29-02-2008	2
		subtotaal	34

Tabel 13.4: Planning transition fase, 3e iteratie

Totaal aantal uur iteratie 3	194
-------------------------------------	------------

13.3 Use-cases

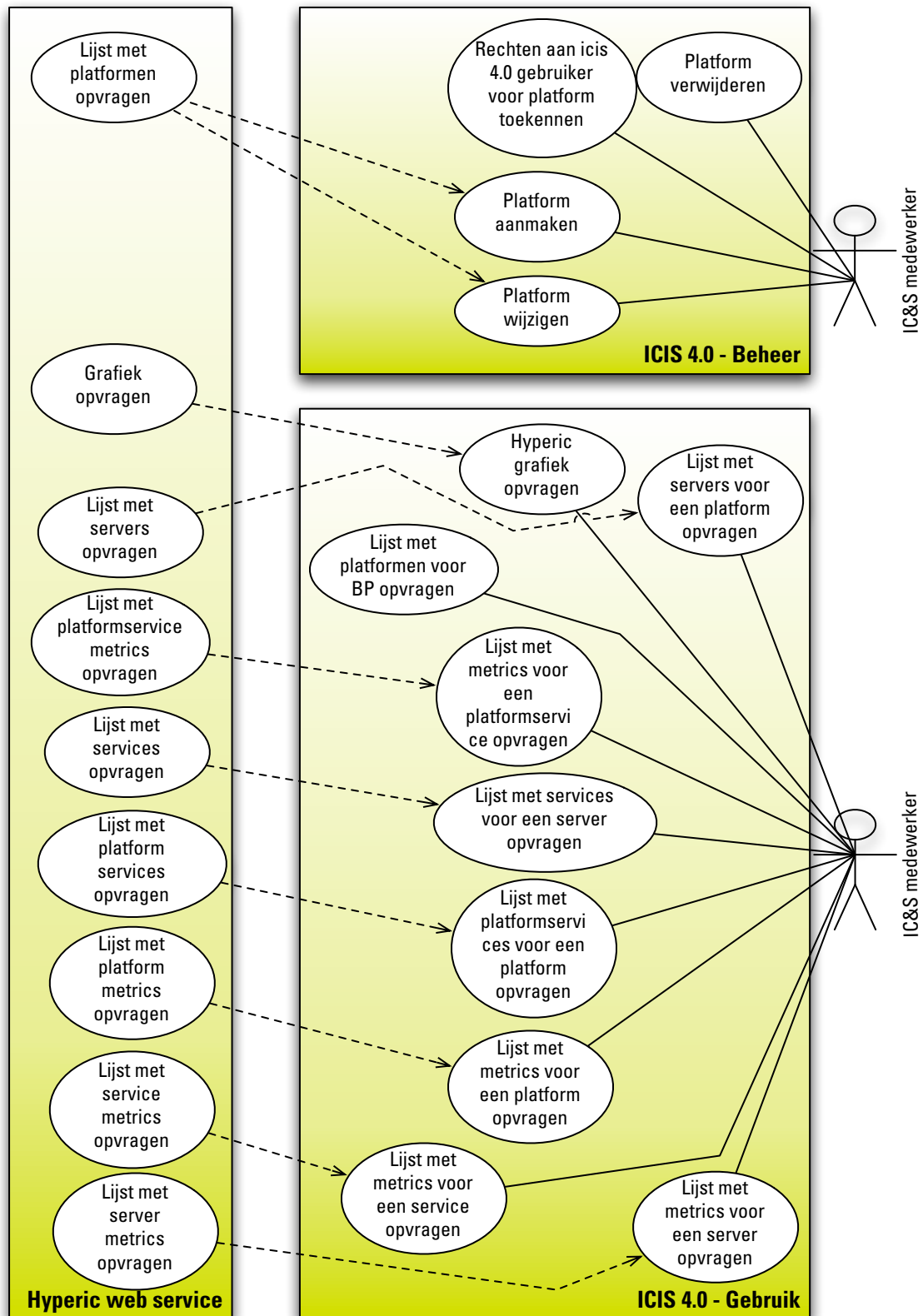
Figuur 13.1 toont een van de use-case diagrammen voor de ICIS 4.0 web service client. Om het overzicht te bewaren en het verschil tussen IC&S medewerkers en IC&S klanten en daarnaast Hyperic web service en Bacula web service te benadrukken is het use-case diagram opgedeeld in vier use-case diagrammen. De use-cases die te zien zijn, zijn tot stand gekomen op basis van de requirements van de 3e iteratie, de Hyperic web service en de Bacula web service. De overige use-case diagrammen zijn te vinden in appendix A.

Een volledige uitwerking van bovenstaand use-case diagram is te vinden in appendix A.

13.4 Deployment

Het deploymentdiagram verdeelt losse omgevingen in zogenaamde nodes en de systemen die daarop draaien in artefacten. Op die manier is de verdeling tussen de verschillende artefacten goed te zien en wordt duidelijker wat de relaties zijn tussen elk van de systemen. Het diagram is te zien in figuur 13.2.

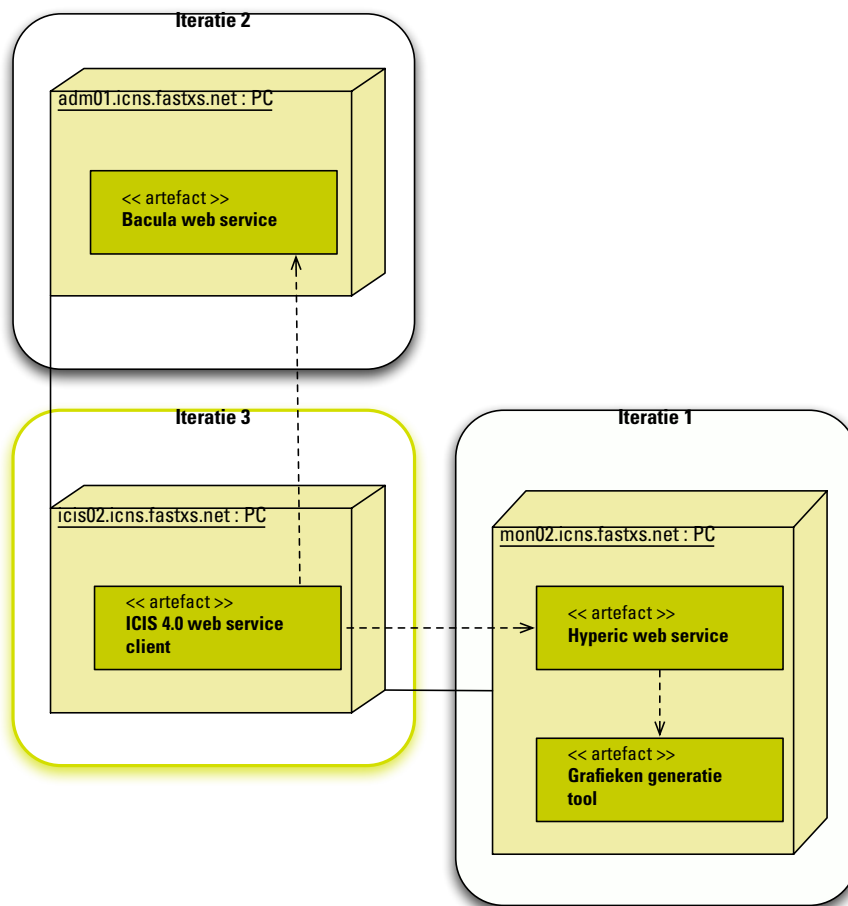
Te zien is dat de ICIS 4.0 web service client zal opereren vanaf de `icis02.icns.fastxs.net` node en van daaruit de nodes zal aanspreken die de Hyperic web service en de Bacula web service herbergen. De verbinding tussen de nodes wordt op fysiek niveau doormiddel van een internet verbinding tot stand gezet. Door middel van de internet verbinding zal de ICIS 4.0 web service client artefact in staat zijn te communiceren met de web service artefacten op de nodes om de `icis02.icns.fastxs.net` node heen.



Figuur 13.1: Use-case diagram voor het gebruik van ICIS 4.0 met de Hyperic web service voor medewerkers van IC&S

Naam	Platform aanmaken
Positie in diagram	ICIS 4.0 ->Beheer ->Hyperic web service
Samenvatting	Er wordt een nieuw platform aangemaakt. Het nieuwe platform is een losstaand object dat een referentie heeft naar een Hyperic platform en/of een Bacula platform of geen van beide omdat een platform niet per definitie in Hyperic of Bacula hoeft te bestaan.
Actoren	IC&S medewerker
Aannamen	IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. Medewerker opent de 'settings' pagina, zoals deze pagina in ICIS 4.0 genoemd wordt. 3. Medewerker kiest ervoor een nieuw platform aan te maken. 4. Medewerker vult verplichte gegevens (platform naam) en de niet verplichte gegevens (Hyperic platform, Bacula platform) in. De niet verplichte gegevens (Hyperic platform en Bacula platform) worden aangeleverd door de Hyperic web service en de Bacula web service. 5. Medewerker bevestigt het nieuwe platform doormiddel van het inklikken van de bevestigings knop.
Uitzonderingen	[Platform bestaat al] Als er al een platform bestaat dat dezelfde naam draagt, dan zal daarvan een melding worden geretourneert en zal de medewerker stap 4 moeten aanpassen.
Resultaat	Een nieuw toegevoegd platform in ICIS 4.0.

Tabel 13.5: Use-case Platform aanmaken



Figuur 13.2: ICIS 4.0 web service client deployment diagram

13.5 Overdracht

Omdat dit mijn project is binnen IC&S en ik na de afstudeer werkzaamheden werkzaam zal blijven bij IC&S zal ik na de afstudeer periode de werkzaamheden hervatten op het punt waar ik gestopt ben. De documentatie die opgeleverd is tijdens de verschillende fases binnen de iteraties zal als leidraad gebruikt blijven worden bij de verdere ontwikkeling van de ICIS 4.0 web service client.

Hoofdstuk 14

Dynamisch rapporteren met L^AT_EX

De grafieken generator tool die beschreven wordt in hoofdstuk 8 wordt door IC&S ook gebruikt voor het opstellen van papieren versies van performance en availability rapporten. Door het combineren van de grafieken generator tool en L^AT_EX als markup language kan op een krachtige manier een dynamisch rapport in elkaar gezet worden. Dit hoofdstuk gaat in op de manier waarop de rapporten door IC&S geproduceerd en gebruikt worden. Het product dat tijdens deze afstudeerstage opgeleverd is speelt een grote rol in de productie van deze performance rapporten en is dus een toepassing van het opgeleverde product en kon worden gerealiseerd naar aanleiding van requirement 2 van de grafieken generator tool.

14.1 L^AT_EX

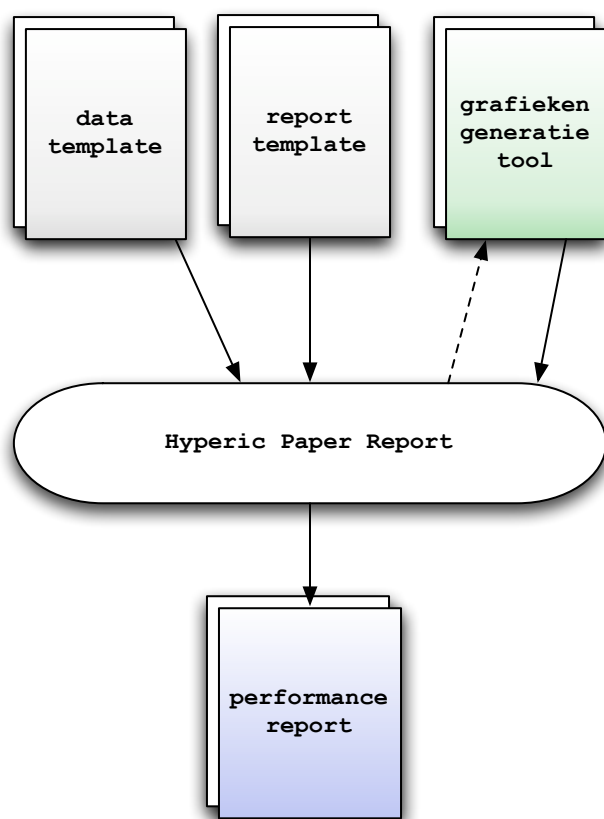
L^AT_EX (spreek uit als la-tech) is een document markup language en document preparation systeem voor het TeX typesetting systeem. Simpel gezegd komt het erop neer dat de auteur van een L^AT_EX document zich meer over de inhoud dan over de layout druk hoeft te maken, omdat juist L^AT_EX de layout op een consistente manier regelt. L^AT_EX wordt veel gebruikt voor het schrijven van wetenschappelijke documentatie omdat het goed met de weergave van formules om kan gaan.

In tegenstelling tot de meeste tekst editors, zoals MS Word, Open Office of Apple Pages, die allemaal WYSIWYG (What You See Is What You Get) editors zijn, is L^AT_EX een zogenaamde WYMIWYG (What You Mean Is What You Get) editor. Als auteur van een L^AT_EX document type je zowel de tekst als de layout van een document.

L^AT_EX documenten kunnen worden omgezet naar verschillende output formaten, zoals PDF, DVI of PostScript. Omdat zowel de inhoud als de layout van een document getypt worden, is het met L^AT_EX ook goed mogelijk dynamische rapporten te schrijven.

14.2 Hyperic Paper Report

Deze drie ingredienten vormen samen een tool die dynamisch rapporten kan creëren, binnen IC&S Hyperic Paper Report genoemd. Door het samenvoegen van een template waar verwijzingen naar metingen in Hyperic in staan en een template die de layout van een rapport bepaalt, wordt door een Python script de grafieken generator tool aangesproken en gecombineerd tot een performance rapport. Zie figuur 14.1 voor een diagram dat laat zien hoe de verschillende informatiebronnen worden samengevoegd.



Figuur 14.1: Hyperic Paper Report combineert verschillende informatiebronnen tot een rapport

Het data template is een xml document dat de grafieken, toelichtingen en de volgorde ervan aangeeft voor een performance en availability rapport. Het data template zorgt voor de inhoud en de structuur van het document. Per grafiek kan worden aangegeven welke opties zichtbaar gemaakt dienen te worden in het rapport en per grafiek kan optioneel een toelichting worden gegeven. Een voorbeeld van de inhoud van een data template is hieronder te zien:

```
<?xml version="1.0" encoding="utf-8"?>

<datatemplate>
  <title>Autoweek</title>
  <subtitle>performance en availability rapport</subtitle>
  <author>W. van Heerde</author>
  <company><![CDATA[IC\&S Service \& Support B.V.]]></company>

  <documentfill>
    <group>
      <name>Autoweek proxy omgeving</name>

      <subgroup>
        <name>Server Load</name>

        <description>
          De load grafiek laat de gemiddelde load van een server
          over 5 minuten zien. Met de load van een server kan een
          goede indicatie worden verkregen over de performance van
          een server. Een load kleiner dan 3, met sporadisch enkele
          pieken is normaal. Een load die structureel hoger ligt dan
          3 is een actiepoint. Bij een load hoger dan 7 is er een
          serieus risico voor de diensten die de server levert.
        </description>

        <graph metric="77712" warning="3.0" critical="7.0" average="yes"/>
        <graph metric="222402" warning="3.0" critical="7.0" average="yes"/>
      </subgroup>
    </group>
  </documentfill>
</datatemplate>
```

Het report template levert de layout van het document aan. Het report template bestaat uit een standaard $\text{\LaTeX}$ document met daarin enkele variabele waarden. Elke variabele waarde wordt in het $\text{\LaTeX}$ document tussen twee dollartekens weergegeven, bijvoorbeeld: $\text{\$author\$}$. Tijdens het samenvoegen van de het data template en het report template door het Python script worden de variabele waarden vervangen door de corresponderende waarden uit het data template. Sommige elementen uit het data template kunnen niet zomaar geplaatst worden, zoals het graph element. Het graph element is een referentie naar een grafiek die gegenereert zal worden, waarbij de attributen van het element beschrijven wat de opties van de te genereren grafiek zijn.

Het resultaat van het samenvoegen van het data- en rapportage template is een nieuw $\text{\LaTeX}$ document dat vervolgens zal worden omgezet naar een PDF document. Dezelfde manier van samenstellen zal toegepast gaan worden op periodieke security scans.

Hoofdstuk 15

Evaluatie

Evaluatie is een belangrijk onderdeel binnen de meeste projecten. Op die manier kunnen behaalde resultaten en eventuele knelpunten inzichtelijk worden gemaakt. De evaluatie is opgedeeld in een tweetal onderdelen: de proces evaluatie waarin het proces dat tijdens het project doorlopen is wordt geevalueerd en de product evaluatie waarin de ontwikkelde producten geevalueerd worden.

15.1 Proces evaluatie

15.1.1 Planning

De planning die voor dit project van tevoren is opgesteld heeft voor iteratie 1 en iteratie 2 goed stand kunnen houden. De fases die doorlopen zouden worden zijn ook allemaal doorlopen en afgerond op of rond de daarvoor geplande tijdstippen. Door een vertraging in de oplevering van de nieuwe ICIS 4.0 klantenportal heeft iteratie 3 niet volgens planning kunnen plaatsvinden waardoor iteratie 3 in de elaboration fase is blijven steken.

Wat betreft de planning zou ik het niet anders doen. De tijd die ingeplanned stond voor elke fase en haar tussenproducten was over het algemeen genoeg. Op sommige momenten was er tijd over waar op andere momenten iets te weinig tijd beschikbaar was. Over het algemeen gezien was de planning in orde. De opgelopen vertraging en het blijven steken in de 3e iteratie is niet veroorzaakt door een lakende planning.

15.1.2 Afspraken en overlegmomenten

Afspraken over de te ontwikkelen producten zijn gemaakt tijdens overlegmomenten die op willekeurige momenten tijdens het proces plaatsvonden. Op het moment dat ik of de opdrachtgever het nodig achtte was het het altijd mogelijk een overlegmoment te plannen of direct aan te gaan en kon worden voorkomen dat er overbodige overlegmomenten ingeplanned werden.

Het maken van afspraken verliep op een duidelijke manier. Zo werden afspraken altijd vastgelegd op papier en/of in Basecamp zodat altijd terug kon worden gekeken naar de gemaakte afspraken. Het doen van overleg-on-demand is ook een prettige manier van werken omdat op die manier informatie verkregen kan worden op het moment dat het nodig is en hoeft er niet gewacht te worden tot geplande overlegmomenten. Wat ik de volgende keer wel anders zou willen is het inplannen van een 2 wekelijks overleg waarin de status van het project kan worden besproken zodat duidelijk is wat de vooruitgang van betrokken partijen is en welke nieuwe ontwikkelingen er eventueel zijn.

15.1.3 Keuze van REST implementatie software

De keuze voor Django om de web services in te implementeren is uiteindelijk een keuze geweest die tijd heeft opgeleverd. Het framework dat Django biedt neemt veel werk uit handen. De aanwezige documentatie is zo georganiseerd dat het makkelijk is de onduidelijkheden die het ontwikkelen in voor de ontwikkelaar nieuwe producten te verduidelijken. Als ik een van de andere software opties had gekozen om de web services te implementeren was ik langer bezig geweest met het installeren en configureren van de software. Of ik ook langer of korter bezig zou zijn geweest met de ontwikkeling van de web services zelf is niet te zeggen. Als ik de keuze opnieuw zou moeten maken dan zou het resultaat niet anders zijn geweest.

15.1.4 Keuze graphing- of reporting engine

De keuze voor JpGraph als graphing engine heeft zowel zijn voor- als nadelen gehad. Voordelen waren de grafische pracht die JpGraph te bieden had en de aanwezige ervaring met het product, iets dat heeft meegeholpen de ontwikkeltijd binnen de perken van tijd te plannen. Een nadeel kwam naar voren bij de ontwikkeling van de dynamische rapporten zoals beschreven in hoofdstuk 14. Daar bleek dat het handig zou zijn geweest als er ook rapporten konden worden gedefinieerd door grafieken volgens een bepaalde volgorde te groeperen. Een product als JasperReports zou daar ondersteuning voor hebben gehad.

Als ik de keuze opnieuw zou moeten maken dan zou het resultaat misschien wel anders zijn geweest. JasperReports heeft toch wel erg interessante features te bieden die IC&S zeker van pas zouden komen. De stijle leercurve van JasperReports en lange implementatietijd die door diezelfde leercurve veroorzaakt zou worden zou echter niet compatible zijn met de beschikbare tijd voor het afstuderen.

15.2 Product evaluatie

15.2.1 Behaalde doelstellingen

Onderstaande doelstelling waren door de opdrachtgever gedefinieerd:

- *Aan het eind van de afstudeerperiode zullen klanten en medewerkers van IC&S met ICIS 4.0 de grootte van uitgevoerde backup jobs en de totaal verbruikte ruimte kunnen inzien. Deze informatie zal op een per server basis en op basis van datum kunnen worden gefilterd en zal bestaan uit één of meerdere grafieken en een statistisch overzicht.*
- *Aan het eind van de afstudeerperiode zullen klanten en medewerkers van IC&S met ICIS 4.0 één of meerdere grafieken kunnen inzien met informatie over availability en performance over door Hyperic gemonitorde diensten, zoals o.a. in de eerder beschreven voorbeelden genoemd worden.*

De eerste van de twee doelstellingen is gedeeltelijk behaald. Technisch gezien is het zeker mogelijk de grootte van uitgevoerde backup jobs en de totaal gebruikte ruimte in te zien doormiddel van de Bacula web service. De gewenste informatie kan geselecteerd worden op een per server basis en op een per datum basis kunnen worden gefilterd en er kunnen grafieken en statistische informatie opgehaald worden. Het deel van de doelstelling dat nog niet behaald is, is het deel dat stelt dat klanten en medewerkers van IC&S met ICIS 4.0 de gewenste informatie kunnen benaderen.

De tweede van de twee doelstellingen is ook gedeeltelijk behaald. Technisch gezien is het mogelijk een of meerdere grafieken in te zien met informatie over performance en availability over door Hyperic gemonitorde diensten doormiddel van de Hyperic web service. Het deel van de doelstelling dat niet behaald is, is het deel dat klanten en medewerkers van IC&S met ICIS 4.0 de gewenste informatie kunnen benaderen.

15.2.2 Requirements

Aan elk van de in de werkwijze benoemde onderdelen waren een aantal requirements toegekend die in samenwerking met de opdrachtgever tot stand waren gekomen. Deze zijn terug te vinden in de hoofdstukken 9.1, 11.1 en 13.1. Aan de requirements van de Hyperic web service is allemaal voldaan, net als de requirements van de grafieken generator tool. De 4e requirement van de grafieken generator tool is dat de tool efficient moet zijn en een aantrekkelijke systemload:snelheid verhouding hebben. Het genereren van een grafiek duurt gemiddeld 1 seconde waarbij de load van het systeem geen tekenen vertoont die aanwijzen dat de verhoudingen scheef liggen.

De requirements die aan de Bacula web service waren toegekend zijn allemaal voldaan. De ontwikkeling van de ICIS 4.0 web service client was nog niet in een ver genoeg ontwikkeld stadium om een van de gestelde requirements als voldaan te benoemen.

15.2.3 Opgeleverde producten

De volgende documentatie is opgeleverd in de 1e iteratie:

- Inception rapport
- Elaboration rapport
- Construction rapport
- Transition rapport
- Handleiding voor de gebruiker
- Handleiding voor de ontwikkelaar

Werkende en geteste versies van de volgende applicaties is opgeleverd:

- Hyperic web service
- Grafieken generator tool

De volgende dienst is over internet aanspreekbaar:

- Hyperic web service

De volgende documentatie is opgeleverd in de 2e iteratie:

- Inception rapport
- Elaboration rapport
- Construction rapport

- Transition rapport
- Handleiding voor de gebruiker
- Handleiding voor de ontwikkelaar

Werkende en geteste versies van de volgende applicaties is opgeleverd:

- Bacula web service

De volgende dienst is aanspreekbaar over internet:

- Bacula web service

In de 3e iteratie is alleen de volgende documentatie opgeleverd:

- Inception rapport

Hoofdstuk 16

Conclusie

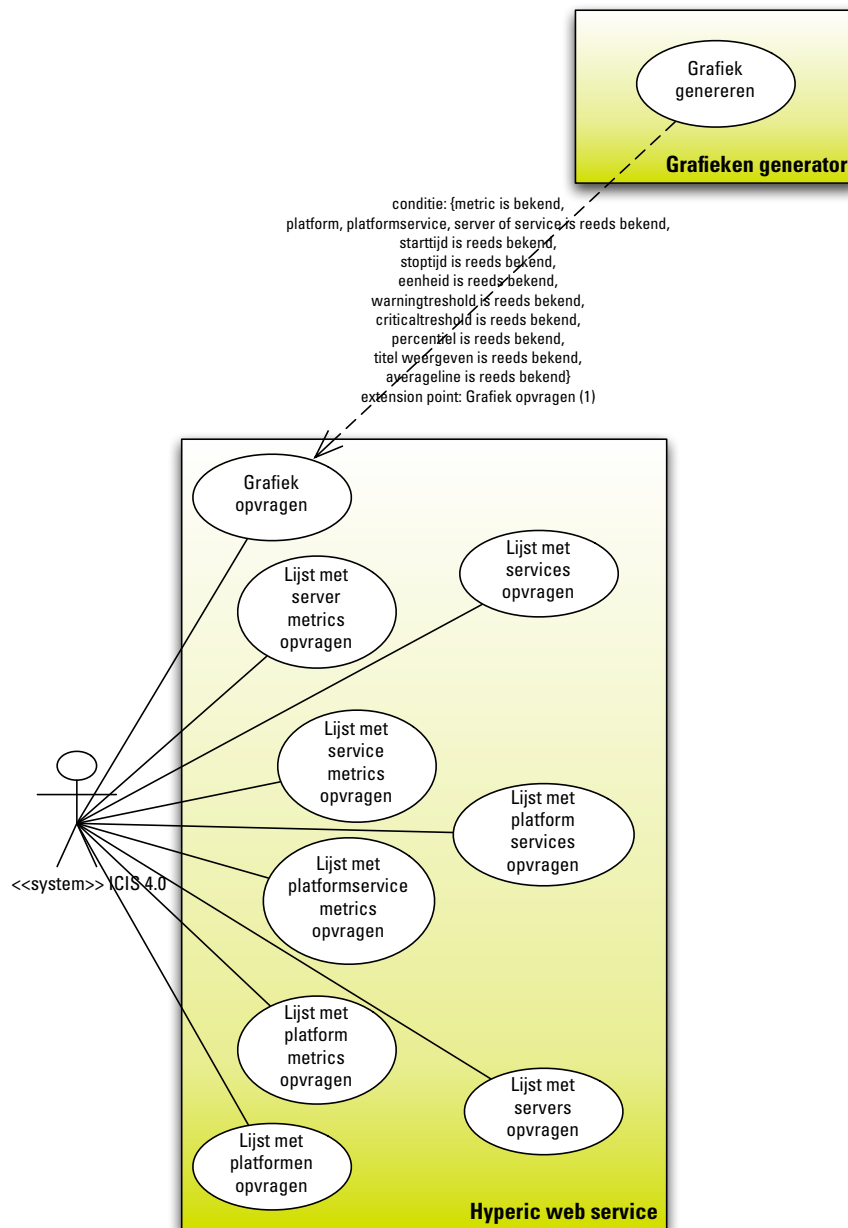
Tijdens het afstuderen is voldaan aan het grootste deel van de requirements die door de opdrachtgever gesteld zijn en zijn de gestelde doelstellingen voor het grootste deel behaald. De beloofde deliverables zijn ook voor het grootste deel opgeleverd aan de opdrachtgever. De situatie waarin de 3e iteratie zich verkeerd zou op basis van de opgeleverde documentatie binnen die iteratie goed te hervatten zijn voor derden. Hiervan is echte geen sprake omdat de 3e iteratie door mijzelf zal worden opgepakt op het punt waar ik was gebleven.

De opdrachtgever heeft aangegeven tevreden te zijn met de opgeleverde producten en het feit dat de producten ook daadwerkelijk in gebruik zijn genomen of nog worden genomen getuigt daar ook van. De Haagse Hogeschool heeft aangegeven dat de opdracht genoeg diepgang bevat om de afstudeeropdracht uit te laten stijgen boven een normale stage. Ondanks de opgelopen vertraging kan ik niet anders concluderen dat het afstuderen overwegend positief is verlopen en dat ook het opgeleverde resultaat positief is voor zowel opdrachtgever als opdrachtnemer.

Appendix A: Use-cases

Dit hoofdstuk bevat alle use-cases uit de drie iteraties die tijdens dit afstuderen behandeld zijn. Per iteratie zal opnieuw het use-case diagram worden getoond en zullen de uitwerkingen van alle drie de bijbehorende use-cases worden weergegeven.

16.1 Iteratie 1: Hyperic web service



Figuur 16.1: Hyperic web service use-case diagram

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met platformen opvragen
Samenvatting	Er wordt een lijst met platformen opgevraagd. Het resultaat zal hetzelfde zijn als de lijst met platformen in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Geen

Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met platformen in Hyperic. 2. De Hyperic web service haalt een lijst met platformen op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platformen] Als er geen platformen in de Hyperic database zitten, dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met platformen die in Hyperic zitten.

Naam	Lijst met platform services opvragen
Samenvatting	Er wordt een lijst met services voor een platform opgevraagd. Het resultaat zal hetzelfde zijn als de lijst met platformservices in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform is bij ICIS 4.0 bekend.

Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met services die bij het gewenste platform horen. 2. De Hyperic web service haalt een lijst met services op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platform] Als blijkt dat het gewenste platform niet bestaat dan wordt daarvan een melding geretourneert. [Geen services] Als er geen services aan het gewenste platform gekoppelt zijn, dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met services die aan het gewenste platform gekoppelt zijn.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met servers opvragen
Samenvatting	Er wordt een lijst met servers voor een platform opgevraagd. Het resultaat zal hetzelfde zijn als de lijst met platformen in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform is bij ICIS 4.0 bekend.
Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met servers die bij het gewenste platform horen. 2. De Hyperic web service haalt een lijst met servers op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platform] Als blijkt dat het gewenste platform niet bestaat dan wordt daarvan een melding geretourneert. [Geen servers] Als er geen servers aan het gewenste platform gekoppelt zijn, dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met servers die aan het gewenste platform gekoppelt zijn.

Naam	Lijst met services opvragen
Samenvatting	Er wordt een lijst met services voor een server opgevraagd. Het resultaat zal hetzelfde zijn als de lijst met services in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform en de gewenste server is bij ICIS 4.0 bekend.
Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met services die bij de gewenste server horen. 2. De Hyperic web service haalt een lijst met services op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platform] Als blijkt dat het gewenste platform niet bestaat dan wordt daarvan een melding geretourneert. [Geen server] Als blijkt dat de gewenste server niet (meer) bekend is bij Hyperic dan wordt hiervan een melding geretourneert. [Geen services] Als er geen services aan de gewenste server gekoppelt zijn, dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met services die aan het gewenste platform gekoppelt zijn.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met platform metrics opvragen
Samenvatting	Er wordt een lijst met metrics voor een platform opgevraagd. Het resultaat zal hetzelfde zijn als de lijst in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform is bij zowel ICIS 4.0 als Hyperic bekend.
Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met de metrics die bij het gewenste platform horen. 2. De Hyperic web service haalt een lijst met metrics op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platform] Als blijkt dat het gewenste platform niet bestaat dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met metrics die aan het gewenste platform gekoppelt zijn.

Naam	Lijst met platformservice metrics opvragen
Samenvatting	Er wordt een lijst met metrics voor een platformservice opgevraagd. Het resultaat zal hetzelfde zijn als de lijst in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform en de gewenste platformservice zijn bij zowel ICIS 4.0 als Hyperic bekend.
Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met de metrics die bij de gewenste platformservice horen. 2. De Hyperic web service haalt een lijst met metrics op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platform] Als blijkt dat het gewenste platform niet bestaat dan wordt daarvan een melding geretourneert. [Geen service] Als blijkt dat de gewenste platformservice niet bestaat dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met metrics die aan de gewenste platformservice gekoppelt zijn.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met server metrics opvragen
Samenvatting	Er wordt een lijst met metrics voor een server opgevraagd. Het resultaat zal hetzelfde zijn als de lijst in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform en de gewenste server zijn bij zowel ICIS 4.0 als Hyperic bekend.

Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met de metrics die bij de gewenste server horen. 2. De Hyperic web service haalt een lijst met metrics op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platform] Als blijkt dat het gewenste platform niet bestaat dan wordt daarvan een melding geretourneert. [Geen server] Als blijkt dat de gewenste server niet bestaat dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met metrics die aan de gewenste server gekoppelt zijn.

Naam	Lijst met service metrics opvragen
Samenvatting	Er wordt een lijst met metrics voor een service opgevraagd. Het resultaat zal hetzelfde zijn als de lijst in de Hyperic web interface.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform, de gewenste server en de gewenste service zijn bij zowel ICIS 4.0 als Hyperic bekend.

Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met de metrics die bij de gewenste service horen. 2. De Hyperic web service haalt een lijst met metrics op uit de Hyperic database en retourneert deze.
Uitzonderingen	[Geen platform] Als blijkt dat het gewenste platform niet bestaat dan wordt daarvan een melding geretourneert. [Geen server] Als blijkt dat de gewenste server niet bestaat dan wordt daarvan een melding geretourneert. [Geen service] Als blijkt dat de gewenste service niet bestaat dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met metrics die aan de gewenste service gekoppelt zijn.

Ontwikkeling van RESTful web services voor ICIS 4.0

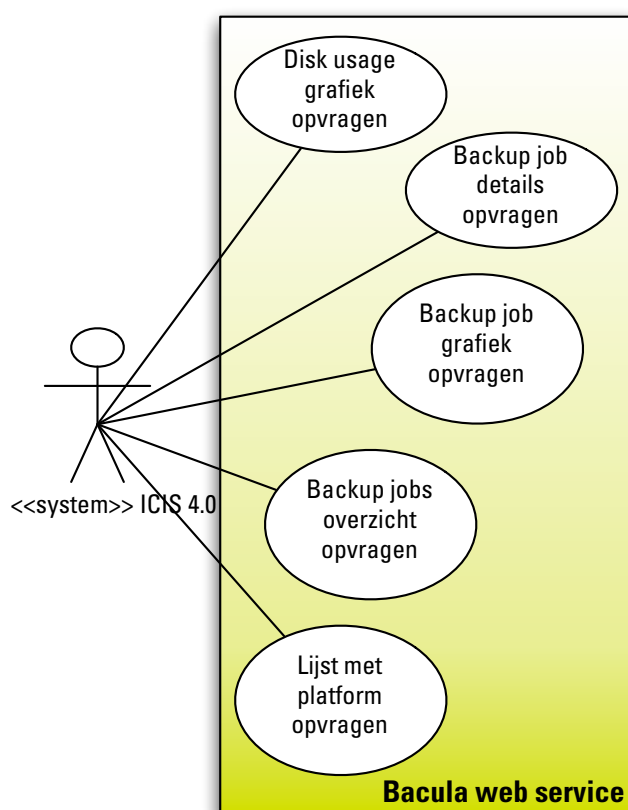
Naam	Grafiek opvragen
Samenvatting	Aan de hand van enkele parameters wordt de opdracht gegeven een nieuwe grafiek te genereren.
Actoren	<<system>>ICIS 4.0
Aannamen	Naast het gewenste platform, platformservice, server of service, is de gewenste metric, de gewenste starttijd, stoptijd, eenheid, warningtreshold, criticaltreshold en percentiel waarde bekend. Ook is bekend of er een titel boven de grafiek zal worden gezet en of er een lijn getekend moet worden die het gemiddelde aangeeft.

Beschrijving	1. ICIS 4.0 geeft via een URI de opdracht een nieuwe grafiek te genereren op basis van de ingevoerde parameters.
Uitzonderingen	-
Resultaat	Melding dat de opdracht tot het maken van de grafiek verzonden is.

Naam	Grafiek genereren
Samenvatting	Aan de hand van enkele parameters wordt een nieuwe grafiek gegenereerd.
Actoren	<<system>>ICIS 4.0
Aannamen	Naast het gewenste platform, platformservice, server of service, is de gewenste metric, de gewenste starttijd, stoptijd, eenheid, warningtreshold, criticaltreshold en percentiel waarde bekend. Ook is bekend of er een titel boven de grafiek zal worden gezet en of er een lijn getekend moet worden die het gemiddelde aangeeft.

Beschrijving	1. De Hyperic web service speelt de argumenten door aan de grafieken generator tool. 2. De grafieken generator genereert een grafiek aan de hand van de binnengekomen argumenten.
Uitzonderingen	[Metric data niet beschikbaar] Komt voor op het moment dat door de Hyperic web service een grafiek wordt aangevraagd met een startdatum en einddatum waarin er nog geen meet gegevens zijn verzameld voor de gewenste service. Deze uitzondering kan ook voorkomen op het moment dat de gewenste metric niet bestaat.
Resultaat	Een visuele representatie van de gewenste meting met de gewenste parameters in de vorm van een grafiek. De grafiek wordt geretourneerd als een plaatje.

16.2 Iteratie 2: Bacula web service



Figuur 16.2: Bacula web service use-case diagram

Naam	Disk usage grafiek opvragen
Samenvatting	Er wordt een nieuwe graek aangemaakt die weergeeft hoe hoog disk usage van een opgegeven platform is in een bepaalde periode. De periode wordt bepaald aan de hand van user input of aan de hand van een standaardberekening in het geval dat er geen user input aanwezig is.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform is bekend. Daarnaast is de gewenste startdatum, starttijd, stopdatum en stopdatum bekend.
Beschrijving	1. ICIS 4.0 geeft via een URI de opdracht ene nieuwe disk usage graek te genereren op basis van de ingevoerde parameters. 2. De Bacula web service genereert een graek vanuit de statistieken database aan de hand van de binnengekomen parameters.
Uitzonderingen	-
Resultaat	Een gegenereerde grafiek in de vorm van een plaatje

Naam	Backup job grafiek opvragen
Samenvatting	Er wordt een nieuwe graek aangemaakt die weergeeft hoe groot de backup-jobs waren die in een bepaalde periode gedraaid hebben. In de graek is ook te zien of er jobs gedraaid hebben. De periode wordt bepaald aan de hand van user input of aan de hand van een standaardberekening in het geval dat er geen user input aanwezig is.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform is bekend. Daarnaast is de gewenste startdatum, starttijd, stopdatum en stopdatum bekend.

Beschrijving	1. ICIS 4.0 geeft via een URI de opdracht ene nieuwe backup job graek te genereren op basis van de ingevoerde parameters. 2. De Bacula web service genereert een graek vanuit de statistieken database aan de hand van de binnengekomen parameters.
Uitzonderingen	-
Resultaat	Een gegenereerde grafiek in de vorm van een plaatje

Naam	Backup jobs overzicht opvragen
Samenvatting	Een overzicht wordt opgevraagd van de gedraaide backup jobs in een bepaalde periode. De periode wordt bepaald aan de hand van user input of aan de hand van een standaardberekening in het geval dat er geen user input aanwezig is.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform is bekend. Daarnaast is de gewenste startdatum, starttijd, stopdatum en stopdatum bekend.

Beschrijving	1. ICIS 4.0 geeft via een URI de opdracht een overzicht van gedraaide backup jobs in een bepaalde periode te genereren. 2. De Bacula web service genereert een backup job overzicht vanuit de statistieken database aan de hand van de binnengekomen parameters.
Uitzonderingen	[Geen backups] Als er geen backups gedraaid zijn in de opgegeven periode, dan zal daar een melding van worden geretourneert.
Resultaat	Een overzicht van de gedraaide backup jobs in de aangegeven periode.

Ontwikkeling van RESTful web services voor ICIS 4.0

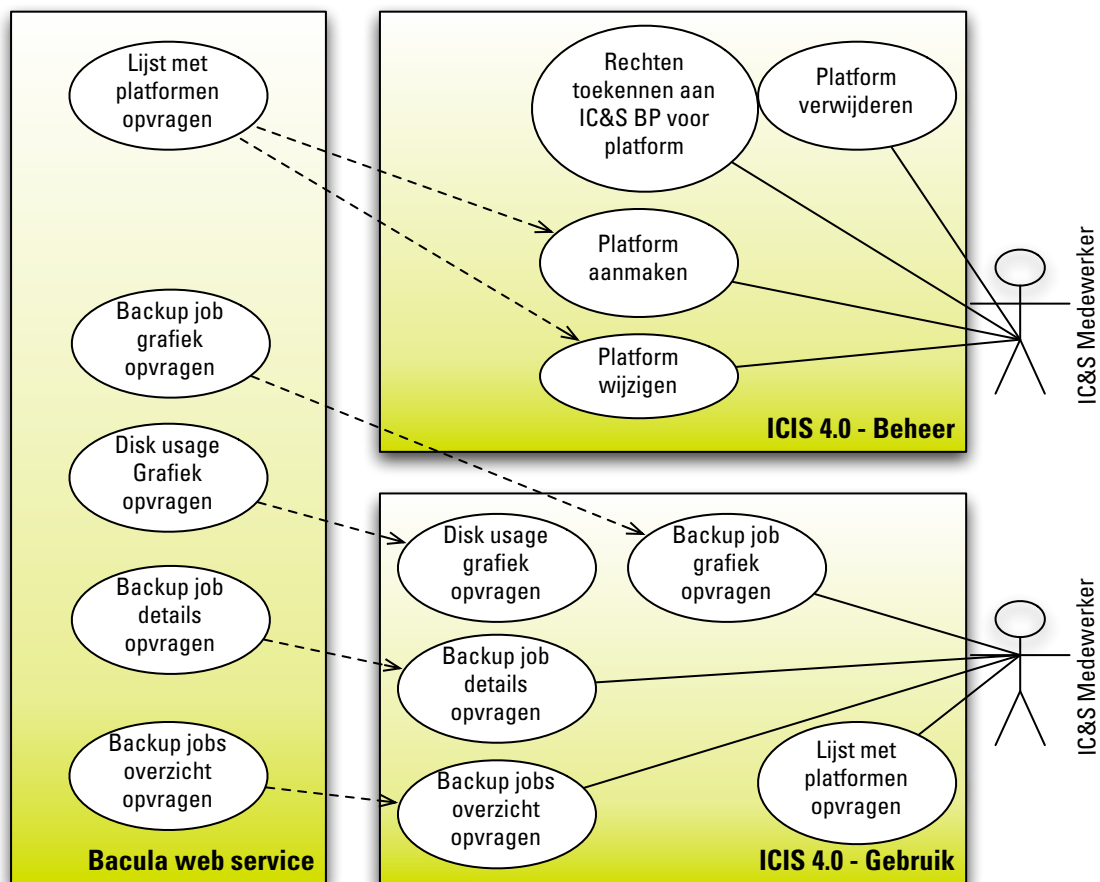
Naam	Backup jobs details opvragen
Samenvatting	Details van een specifieke job worden opgevraagd. De details omvatten o.a. de grootte van de backup job en de messages die door de Backup software zijn gegenereerd tijdens het back-uppen. De selectie voor een specifieke backup job wordt gemaakt aan de hand van de Backup jobs overzicht opvragen use-case.
Actoren	<<system>>ICIS 4.0
Aannamen	Het gewenste platform is bekend. Daarnaast is uit het overzicht een backup job gekozen.

Beschrijving	1. ICIS 4.0 geeft via een URI de opdracht details op te halen van een specifieke backup job. 2. De Bacula web service retourneert de details van een specifieke backup job vanuit de statistieken database aan de hand van de binnengekomen parameters.
Uitzonderingen	[Geen job] Als de gekozen job niet valide is, dan zal daarvan een melding worden geretourneert.
Resultaat	Details van de geselecteerde backup job.

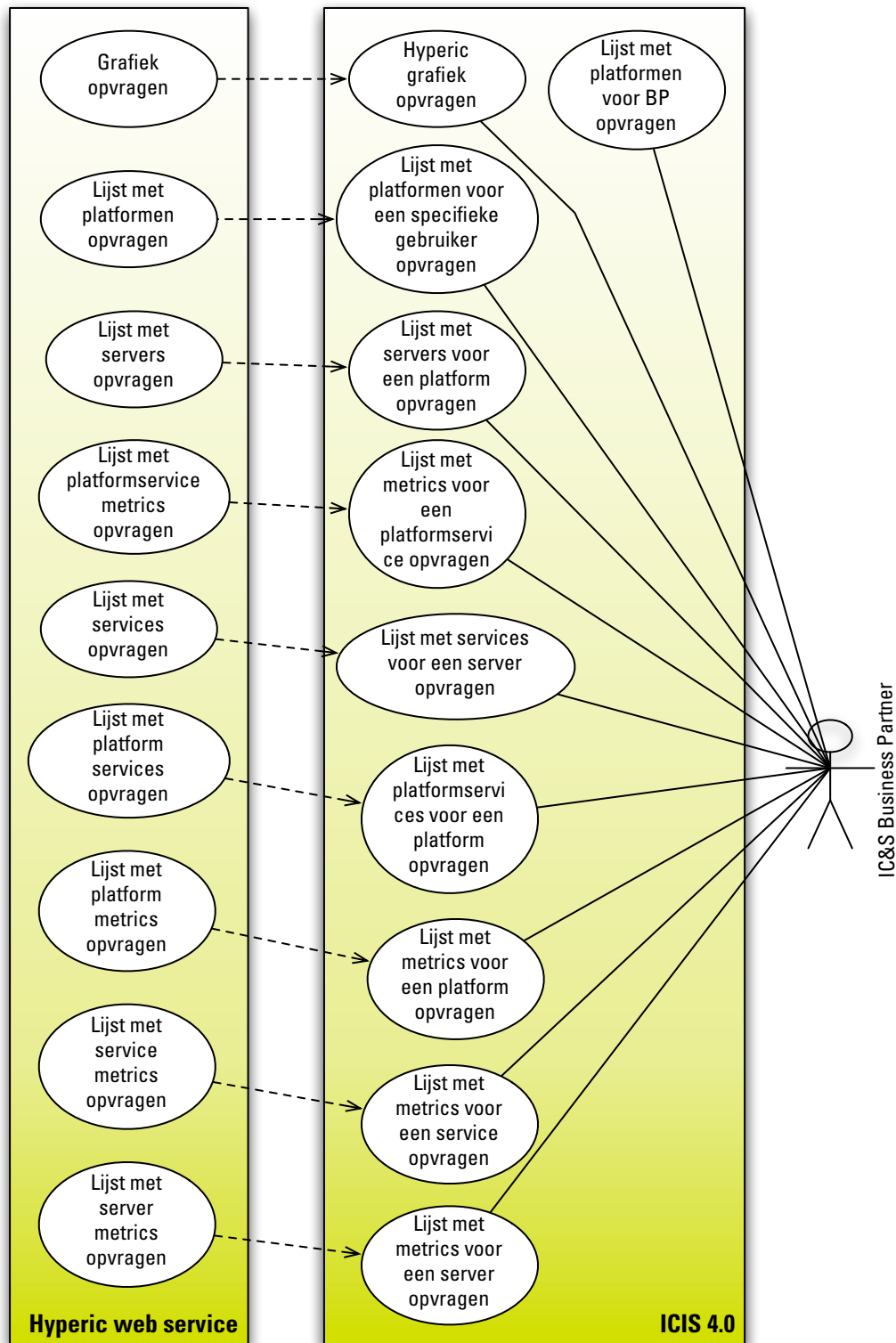
Naam	Lijst met platformen opvragen
Samenvatting	Er wordt een lijst met platformen opgevraagd, zoals deze ook in de Bacula (backup software) configuratie te vinden is.
Actoren	<<system>>ICIS 4.0
Aannamen	Geen

Beschrijving	1. ICIS 4.0 vraagt via een URI een lijst op met de platformen in Bacula. 2. De Bacula web service haalt een lijst met platformen op uit de Bacula statistieken database en retourneert deze.
Uitzonderingen	[Geen platformen] Als er geen platformen in de Bacula database zitten, dan wordt daarvan een melding geretourneert.
Resultaat	Een lijst met platformen die in Bacula zitten.

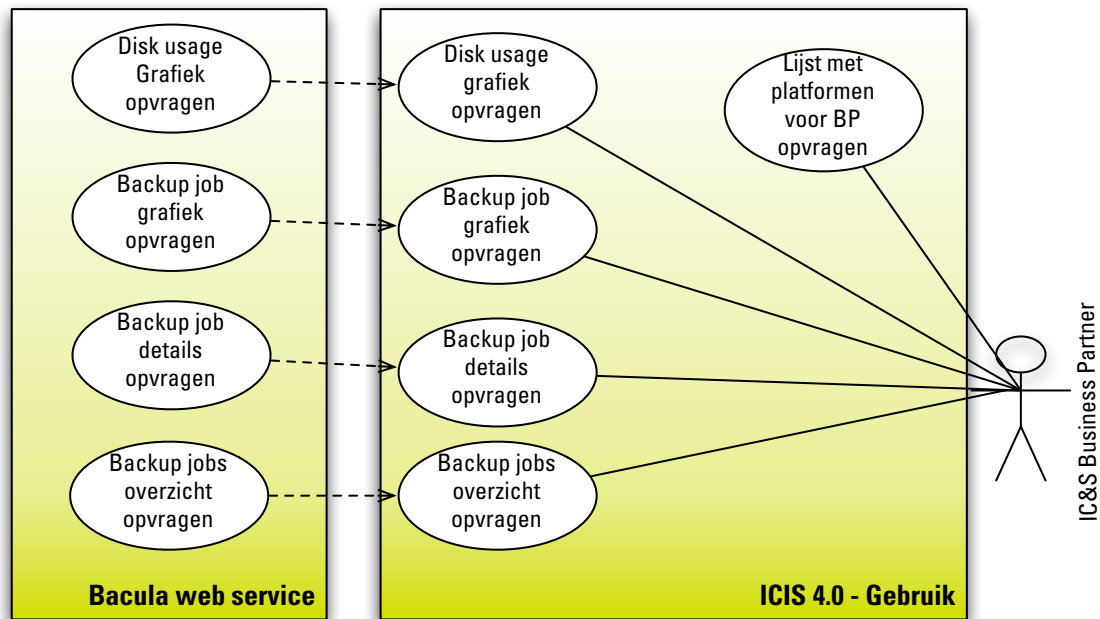
16.3 Iteratie 3: ICIS 4.0 web service client



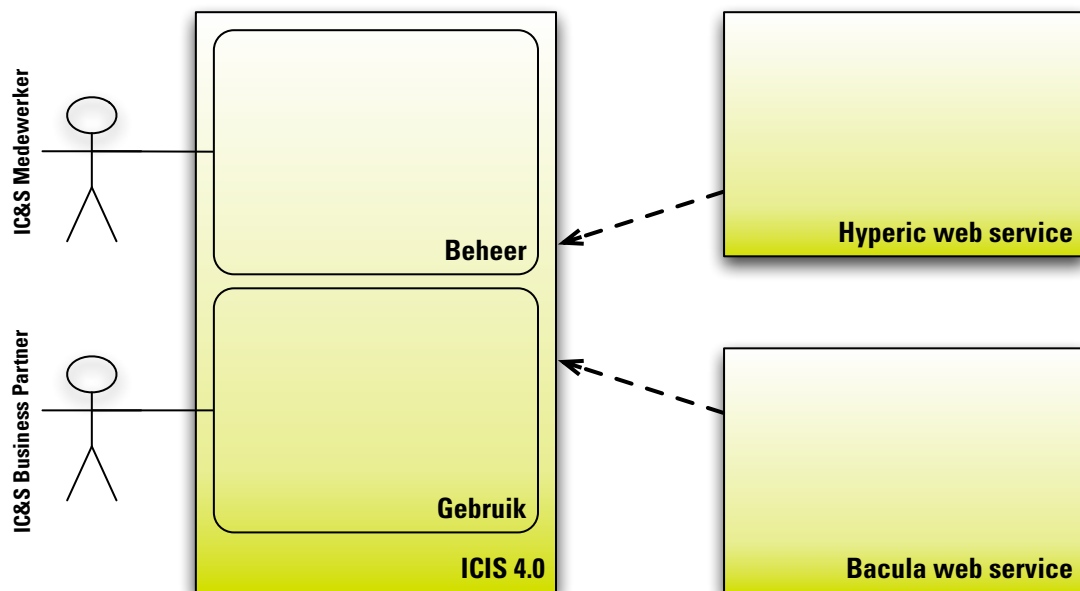
Figuur 16.3: Use-case diagram voor het gebruik van ICIS 4.0 met de Bacula web service voor medewerkers van IC&S



Figuur 16.4: Use-case diagram voor het gebruik van ICIS 4.0 met de Hyperic web service voor klanten van IC&S



Figuur 16.5: Use-case diagram voor het gebruik van ICIS 4.0 met de Bacula web service voor klanten van IC&S



Figuur 16.6: Een schematische voorstelling van het samenkomen van ICIS 4.0, de Hyperic web service en de Bacula web service

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met platformen voor BP opvragen
Positie in diagram	ICIS 4.0 ->Gebruik ->Hyperic web service
Samenvatting	Er wordt een lijst met platformen opgevraagd zoals deze door een IC&S medewerker aan een specifieke business partner is gekoppeld.
Actoren	IC&S Business Partner
Aannamen	De IC&S Business Partner is ingelogd in ICIS 4.0.
Beschrijving	1. BP navigeert middels de tab CMDB naar het platform overzicht. 2. De platformen die door een IC&S medewerker aan de ingelogde BP gekoppeld zijn en waar de BP een view rol op heeft worden in een dropdown list getoont.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt.
Resultaat	Een lijst met platformen die in ICIS 4.0 zitten en daarnaast aan de BP gekoppeld zijn en waar de BP een view rol op heeft.

Naam	Lijst met platformen voor BP opvragen
Positie in diagram	ICIS 4.0 ->Beheer ->Hyperic web service
Samenvatting	Er wordt een lijst met platformen opgevraagd zoals deze door een IC&S medewerker aan een specifieke business partner is gekoppeld.
Actoren	IC&S medewerker
Aannamen	De IC&S medewerker is ingelogd in ICIS 4.0
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. Medewerker kiest uit een lijst met beschikbare Business Partners een BP naar keuze. 3. Medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare metrics. 4. De platformen die door een IC&S medewerker aan de ingelogde BP gekoppeld zijn drop-down list getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de geselecteerde BP gekoppeld zijn wordt daar naar de IC&S medewerker toe melding van gemaakt.
Resultaat	Een lijst met platformen die in ICIS 4.0 zitten en daarnaast aan de geselecteerde BP gekoppeld zijn.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met metrics voor een platform opvragen
Positie in diagram	ICIS 4.0 ->Hyperic web service
Samenvatting	Er wordt een lijst met metrics voor een geselecteerd platform weergegeven.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. BP of medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 3. BP of medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare metrics. 4. De metrics die bij het geselecteerde platform horen worden getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt. [Geen platformmetrics] Als er geen metingen aan het platform gekoppeld zijn.
Resultaat	Een lijst met metrics die bij het geselecteerde platform horen.

Naam	Lijst met servers voor een platform opvragen
Positie in diagram	ICIS 4.0 ->Hyperic web service
Samenvatting	Er wordt een lijst met servers voor een geselecteerd platform weergegeven.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. BP of medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 3. BP of medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare metrics. 4. De servers die bij het geselecteerde platform horen worden getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt. [Geen servers] Als er geen servers aan het platform gekoppeld zijn.
Resultaat	Een lijst met servers die bij het geselecteerde platform horen.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met metrics voor een server opvragen
Positie in diagram	ICIS 4.0 ->Hyperic web service
Samenvatting	Er wordt een lijst met metrics voor een geselecteerde server weergegeven.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. BP of medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 3. BP of medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare servers. 4. BP of medewerker selecteert een server uit de lijst met beschikbare servers. De inhoud van de lijst is afhankelijk van het resultaat van de 'lijst met servers voor een platform opvragen' use-case. 5. De metrics die bij de geselecteerde server horen worden getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt. [Geen servers] Als er geen servers aan het platform gekoppeld zijn. [Geen servermetrics] Als er wel servers aan het platform gekoppeld zijn, maar er geen metrics aan de server gekoppeld zijn.
Resultaat	Een lijst met metrics die bij de geselecteerde server horen.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met services voor een server opvragen
Positie in diagram	ICIS 4.0 ->Hyperic web service
Samenvatting	Er wordt een lijst met services voor een geselecteerde server weergegeven.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. BP of medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 3. BP of medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare servers. 4. BP of medewerker selecteert een server uit de lijst met beschikbare servers. De inhoud van de lijst is afhankelijk van het resultaat van de 'lijst met servers voor een platform opvragen' use-case. 5. De services die bij de geselecteerde server horen worden getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt. [Geen servers] Als er geen servers aan het platform gekoppeld zijn. [Geen services] Als er wel servers aan het platform gekoppeld zijn, maar er geen services aan de server gekoppeld zijn.
Resultaat	Een lijst met services die bij de geselecteerde server horen.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met metrics voor een service opvragen
Positie in diagram	ICIS 4.0 ->Hyperic web service
Samenvatting	Er wordt een lijst met metrics voor een geselecteerde service weergegeven.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. BP of medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 3. BP of medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare servers. 4. BP of medewerker selecteert een server uit de lijst met beschikbare servers. De inhoud van de lijst is afhankelijk van het resultaat van de 'lijst met servers voor een platform opvragen' use-case. 5. BP of medewerker selecteert een service uit de lijst met beschikbare services. De inhoud van de lijst wordt is afhankelijk van het resultaat van de 'lijst met services voor een server opvragen' use-case. 6. De metrics die bij de geselecteerde service horen worden getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt. [Geen servers] Als er geen servers aan het platform gekoppeld zijn. [Geen services] Als er wel servers aan het platform gekoppeld zijn, maar er geen services aan de server gekoppeld zijn. [Geen servicemetrics] Als er wel services aan de server gekoppeld zijn, maar er geen metrics aan de service gekoppeld zijn.
Resultaat	Een lijst met metrics die bij de geselecteerde service horen.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met platformservices voor een platform opvragen
Positie in diagram	ICIS 4.0 ->Hyperic web service
Samenvatting	Er wordt een lijst met platformservices voor een geselecteerd platform weergegeven.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. BP of medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 3. BP of medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare servers. 4. De platformservices die bij het geselecteerde platform horen worden getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt. [Geen platformservices] Als er geen platformservices aan het platform gekoppeld zijn.
Resultaat	Een lijst met platformservices die bij het geselecteerde platform horen.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Lijst met metrics voor een platformservice opvragen
Positie in diagram	ICIS 4.0 ->Hyperic web service
Samenvatting	Er wordt een lijst met metrics voor een geselecteerde platformservice weergegeven.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. BP of medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 3. BP of medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare servers. 4. BP of medewerker selecteert een server uit de lijst met beschikbare servers. De inhoud van de lijst is afhankelijk van het resultaat van de 'lijst met servers voor een platform opvragen' use-case. 5. BP of medewerker selecteert een service uit de lijst met beschikbare platformservices. De inhoud van de lijst wordt is afhankelijk van het resultaat van de 'lijst met platformservices voor een platform opvragen' use-case. 6. De metrics die bij de geselecteerde platformservice horen worden getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt. [Geen platformservices] Als er geen platformservices aan het platform gekoppeld zijn. [Geen platformservicemetrics] Als er wel platformservices aan het platform gekoppeld zijn, maar er geen metrics aan de platformservice gekoppeld zijn.
Resultaat	Een lijst met metrics die bij de geselecteerde platformservice horen.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Hyperic grafiek opvragen
Positie in diagram	ICIS 4.0 → Hyperic web service
Samenvatting	Er wordt een graek opgevraagd van de Hyperic web service.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. BP of medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 3. BP of medewerker navigeert middels het 'performance' menu item naar het overzicht met beschikbare servers. 4. De BP of medewerker selecteert een metric uit het totaaloverzicht aan beschikbare metrics. Dit totaaloverzicht wordt gecreëerd doormiddel van alle bovenstaande use-cases. 5. De BP of medewerker bepaalt de overige opties (startdatum, starttijd, einddatum, eindtijd, warning threshold, critical threshold, eenheid, graektitel, gemiddelde lijn en percentiel) 6. Op basis van de geselecteerde entiteit (platform, server, service of platformservice), de geselecteerde metric en de aangegeven opties wordt een graek opgevraagd bij de Hyperic web service. 7. De opgevraagde graek wordt getoond.
Uitzonderingen	[Geen platformen] Als er geen platformen aan de BP gekoppeld zijn of wanneer de BP geen view rol heeft op een enkel platform wordt daar naar de BP toe melding van gemaakt. [Metric data niet beschikbaar] Als de graek niet opgevraagd kan worden.
Resultaat	Een graek die voldoet aan de opgegeven parameters.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Platform aanmaken
Positie in diagram	ICIS 4.0 ->Beheer ->Hyperic web service
Samenvatting	Er wordt een nieuw platform aangemaakt. Het nieuwe platform is een losstaand object dat een referentie heeft naar een Hyperic platform en/of een Bacula platform of geen van beide omdat een platform niet per definitie in Hyperic of Bacula hoeft te bestaan.
Actoren	IC&S medewerker
Aannamen	IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. Medewerker opent de 'settings' pagina, zoals deze pagina in ICIS 4.0 genoemd wordt. 3. Medewerker kiest ervoor een nieuw platform aan te maken. 4. Medewerker vult verplichte gegevens (platform naam) en de niet verplichte gegevens (Hyperic platform, Bacula platform) in. De niet verplichte gegevens (Hyperic platform en Bacula platform) worden aangeleverd door de Hyperic web service en de Bacula web service. 5. Medewerker bevestigt het nieuwe platform doormiddel van het inklikken van de bevestigings knop.
Uitzonderingen	[Platform bestaat al] Als er al een platform bestaat dat dezelfde naam draagt, dan zal daarvan een melding worden geretourneert en zal de medewerker stap 4 moeten aanpassen.
Resultaat	Een nieuw toegevoegd platform in ICIS 4.0.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Platform verwijderen
Positie in diagram	ICIS 4.0 ->Beheer
Samenvatting	Een bestaand platform wordt verwijderd. De verwijdering is alleen van toepassing op ICIS 4.0 en niet op de platformen in de Hyperic web service en de Bacula web service die aan het platform gekoppelt zijn.
Actoren	IC&S medewerker
Aannamen	IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. Medewerker opent de 'settings' pagina, zoals deze pagina in ICIS 4.0 genoemd wordt. 3. Medewerker kiest ervoor een bestaand platform te verwijderen. 4. Medewerker selecteert het te verwijderen platform. 5. Medewerker bevestigt de verwijdering van het platform doormiddel van het inklikken van de bevestigings knop.
Uitzonderingen	-
Resultaat	Het platform is uit ICIS 4.0 verwijderd en niet meer opvraagbaar voor BP's of IC&S medewerkers. Alle koppelingen met een of meerdere BP's is daarmee ook verwijderd. Ook zijn de koppelingen met de Hyperic web service en de Bacula web service verwijderd.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Platform wijzigen
Positie in diagram	ICIS 4.0 ->Beheer
Samenvatting	Een bestaand platform wordt gewijzigd. Een platform kan op gewijzigd worden op platform naam, Hyperic platform koppeling (Hyperic web interface) en Bacula platform koppeling (Bacula web interface).
Actoren	IC&S medewerker
Aannamen	IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. Medewerker opent de 'settings' pagina, zoals deze pagina in ICIS 4.0 genoemd wordt. 3. Medewerker kiest ervoor een bestaand platform te wijzigen. 4. Medewerker selecteert het te wijzigen platform. 5. Medewerker maakt de gewenste wijzigingen. 6. Medewerker bevestigt de wijzigingen van het platform doormiddel van het inklikken van de bevestigings knop.
Uitzonderingen	[Platform bestaat al] Als er al een platform bestaat dat dezelfde naam draagt als de gewenste naam, dan zal daarvan een melding worden gere- tourneert en zal de medewerker stap 5 moeten aanpassen.
Resultaat	Het platform is in ICIS 4.0 gewijzigd. Er zijn geen wijzigingen gemaakt aan het platform in Hyperic of Bacula. De wijziging behelst een verandering in platform naam, koppeling met Hyperic (Hyperic web service) of koppeling met Bacula (Bacula web service).

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Rechten wijzigen aan IC&S BP voor platform
Positie in diagram	ICIS 4.0 ->Beheer ->Hyperic web service
Samenvatting	Aan een bestaand platform worden de rechten gewijzigd. De gewijzigde rechten geeft de BP waar de rechten op van toepassing zijn de mogelijkheid het platform op te vragen middels de lijst die gecreerd wordt in de 'Lijst met platformen voor BP opvragen' use-case of juist een platform onzichtbaar te maken voor een BP.
Actoren	IC&S medewerker
Aannamen	IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. Medewerker opent de 'settings' pagina, zoals deze pagina in ICIS 4.0 genoemd wordt. 3. Medewerker kiest ervoor rechten te wijzigen voor een bestaand platform. 4. Medewerker selecteert het platform, de BP en de rechten die toegekend dienen te worden. 5. Medewerker bevestigt de rechten van het platform doormiddel van het inklikken van de bevestigings knop.
Uitzonderingen	-
Resultaat	De rechten op het platform zijn gewijzigd.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Disk usage graek opvragen, backup job graek opvragen
Positie in diagram	ICIS 4.0 ->Bacula web service
Samenvatting	Er wordt een disk usage graek of een backup job grafiek opgevraagd van de Bacula web service.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker navigeert middels het 'backup' menu item naar het overzicht met beschikbare metrics. 3. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de lijst met platformen voor BP opvragen use-case. 4. De BP of medewerker selecteert een van de twee beschikbare metrics (disk usage of backup job). De BP of medewerker bepaalt de overige opties (startdatum, starttijd, einddatum, eindtijd, showvalues). 5. Op basis van het geselecteerde platform, de geselecteerde metric en de aangegeven opties wordt een graek opgevraagd bij de Bacula web service. 6. De opgevraagde graek wordt getoond.
Uitzonderingen	-
Resultaat	Een graek die voldoet aan de aangegeven parameters.

Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Backup jobs overzicht opvragen
Positie in diagram	ICIS 4.0 -> Bacula web service
Samenvatting	Er wordt een overzicht van uitgevoerde backup jobs opgevraagd. Dit overzicht bevat zowel jobs die mislukt zijn als jobs die wel succesvol zijn.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker navigeert middels het 'backup' menu item naar het overzicht met beschikbare metrics. 3. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 4. Het overzicht van uitgevoerde backup jobs wordt getoond.
Uitzonderingen	[Geen backup jobs] Als er geen backup jobs zijn uitgevoerd, wordt daarvan melding gemaakt aan de BP of IC&S medewerker.
Resultaat	Een overzicht van de uitgevoerde backup jobs.

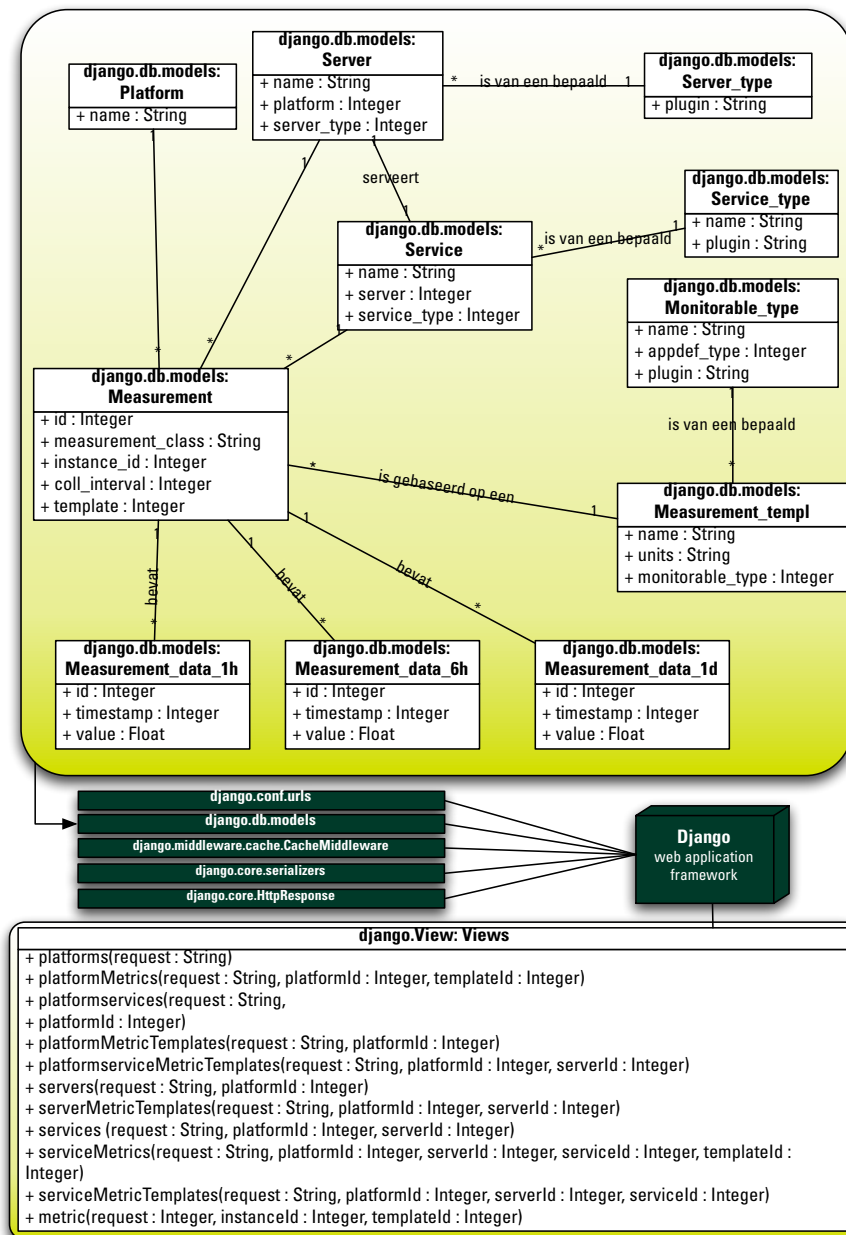
Ontwikkeling van RESTful web services voor ICIS 4.0

Naam	Backup job details opvragen
Positie in diagram	ICIS 4.0 ->Bacula web service
Samenvatting	Van een geselecteerde backup job worden alle beschikbare details weergegeven. Deze details bestaan uit de starttijd, stoptijd, executie tijd, grootte, level, status en log messages van de backup job.
Actoren	IC&S Business Partner of IC&S medewerker
Aannamen	De IC&S Business Partner of IC&S medewerker is ingelogd in ICIS 4.0.
Beschrijving	1. Medewerker navigeert middels de tab CMDB naar het platform overzicht. 2. BP of medewerker navigeert middels het 'backup' menu item naar het overzicht met beschikbare metrics. 3. BP of medewerker selecteert een platform uit de lijst met beschikbare platformen. De inhoud van deze lijst is afhankelijk van het resultaat van de 'lijst met platformen voor BP opvragen' use-case. 4. BP of medewerker selecteert een job uit het getoonde overzicht van uitgevoerde backup jobs; het resultaat van de Backup jobs overzicht opvragen use-case. 5. De backup job details worden weergegeven.
Uitzonderingen	-
Resultaat	Details van de geselecteerde backup job.

Appendix B: Klassediagrammen

Dit hoofdstuk bevat alle klassediagrammen uit de drie iteraties die tijdens dit afstuderen behandeld zijn. Per iteratie zal opnieuw het uitgewerkte klassediagram worden getoond en zal de bijbehorende model dictionary compleet beschreven zijn.

16.4 Iteratie 1: Hyperic web service



Figuur 16.7: Hyperic web service klassediagram

Klasse: Platform	
Een platform staat gelijk aan een fysieke machine. De fysieke machine kan alles zijn, zolang het maar via een netwerk kan worden aangesproken. Hierbij kan gedacht worden aan bijvoorbeeld routers, switches, PDUs of apparatuur draaiend op Linux, Windows, Solaris of OS X.	
Attribuut naam	Omschrijving
<i>+ name: String</i>	De naam van het platform binnen Hyperic. De naam is nodig om duidelijk te maken wat het onderwerp van de grafiek is en is onderdeel van de requirement dat de grafieken uiterlijk aantrekkelijk moeten zijn. De naam van het platform heeft een waarde van het type string en is uniek.

Klasse: Server	
Een server is een stuk software dat draait op een platform. Een server kan verschillende doelen en functies kennen en kan per platform of server instantie anders geconfigureerd zijn. Voorbeelden van servers zijn Apache web server, MySQL of PostgreSQL.	
Attribuut naam	Omschrijving
<i>+ name: String</i>	De naam van de server binnen Hyperic. De naam is nodig om duidelijk te maken wat het onderwerp van de grafiek is en is onderdeel van de requirement dat de grafieken uiterlijk aantrekkelijk moeten zijn. De naam van de server heeft een waarde van het type string.
<i>+ platform: Integer</i>	Een verwijzing naar een instantie van de Platform klasse
<i>+ server_type: Integer</i>	Een verwijzing naar een instantie van de Server_type klasse

Klasse: Server_type	
Een server_type bepaalt van wat voor type de gerelateerde server is.	
Attribuut naam	Omschrijving
<i>+ plugin: String</i>	De naam van de plugin die gebruikt wordt om de server van metrische gegevens te voorzien. Voorbeelden van plugin namen zijn netdevice, apache, db2 en iis.

Klasse: Service	
Een service is een onderdeel van een server of platform. Een service kan verschillende doelen en functies kennen en kan per platform of server instantie anders geconfigureerd zijn. Voorbeelden van services zijn Apache VHost of FileServer Mount.	
Attribuut naam	Omschrijving
<i>+ name: String</i>	De naam van de service binnen Hyperic. De naam is nodig om duidelijk te maken wat het onderwerp van de grafiek is en is onderdeel van de requirement dat de grafieken uiterlijk aantrekkelijk moeten zijn. De naam van de service heeft een waarde van het type string.
<i>+ server: Integer</i>	Een verwijzing naar een instantie van de Server klasse
<i>+ service_type: Integer</i>	Een verwijzing naar een instantie van de Service_type klasse

Klasse: Service_type	
Een service_type bepaalt van wat voor type de gerelateerde service is.	
Attribuut naam	Omschrijving
<i>+ name: String</i>	De naam van het type service. Voorbeelden van namen zijn DB2 9.x Table, Exchange 2000 IMAP4 en JBoss 4.0 Hibernate Session Factory.
<i>+ plugin: String</i>	De naam van de plugin die gebruikt wordt om de server van metrische gegevens te voorzien. Voorbeelden van plugin namen zijn netdevice, apache, db2 en iis.

Klasse: Measurement	
Een measurement is een meting. Een instantie van deze klasse representeert een meting. Een meting is een verzameling waarden die aan een bepaalde tijd gekoppeld zijn.	
Attribuut naam	Omschrijving
<i>+ id: Integer</i>	De identificerende kolom voor de meting.
<i>+ measurement_class: String</i>	Metingen zijn er in de 'D' vorm en de 'R' variant. Om redenen die Hyperic, de fabrikant van de Hyperic monitoring applicatie, niet vrij wou geven, is het belangrijk te werken met metingen van het type 'D'. Metingen van het type 'R' bevatten namelijk geen meetgegevens.
<i>+ instance_id: Integer</i>	Een verwijzing naar het platform, server of service die aan de meting gerelateerd is.
<i>+ coll_interval: Integer</i>	De collectie interval. Deze waarde bepaalt het aantal seconden tussen twee meetwaarden.
<i>+ template: Integer</i>	Een verwijzing naar een instantie van de Measurement_templ klasse.

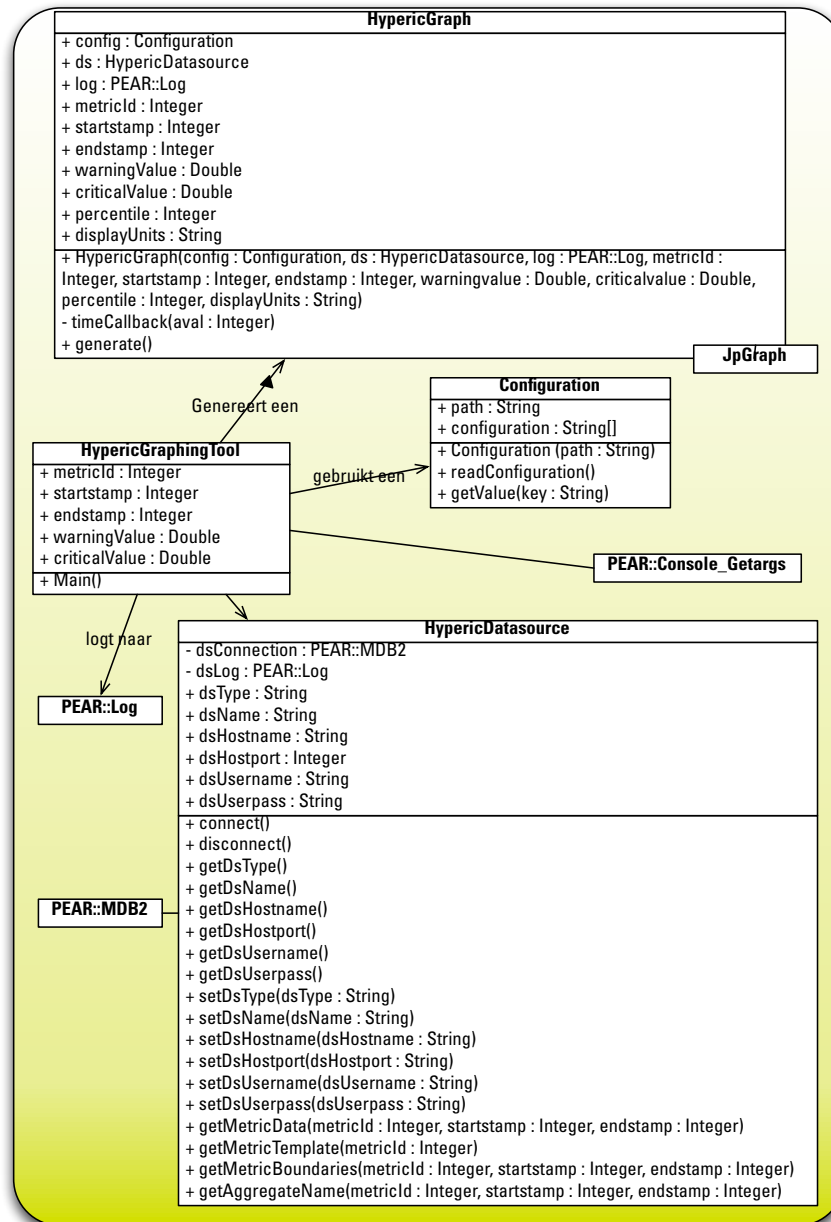
Klasse: Measurement_templ	
Een measurement_templ bevat bepaalde standaardwaarden voor een meting, zoals naam van de meting en de eenheid waarin de meting standaard wordt uitgevoerd.	
Attribuut naam	Omschrijving
<i>+ name</i>	De naam van de gerelateerde meting.
<i>+ measurement_class: String</i>	Metingen zijn er in de 'D' vorm en de 'R' variant. Om redenen die Hyperic, de fabrikant van de Hyperic monitoring applicatie, niet vrij wou geven, is het belangrijk te werken met metingen van het type 'D'. Metingen van het type 'R' bevatten namelijk geen meetgegevens.
<i>+ units: String</i>	De eenheid waarin de meting uitgevoerd wordt.
<i>+ monitorable_type: Integer</i>	Een verwijzing naar het Monitorable_type model.

Klasse: Monitorable_type	
Een monitorable_type bepaalt of een meting op platform, server of service niveau gedaan wordt. Het is nodig dat te weten om de juiste meetgegevens bij de juiste meting te achterhalen.	
Attribuut naam	Omschrijving
+ name: String	De naam van de monitorable_type.
+ appdef_type: Integer	Het niveau waarop de meting wordt uitgevoerd (platform, server of service).
+ plugin: String	De plugin die aan de meting gerelateerd is.

Klasse: Measurement_data_1h	
Een instantie van de Measurement_data_1h klasse representeert de data in diezelfde tabel op database niveau en geeft daarmee toegang tot meet gegevens die tot 1 uur geaggregeerd zijn.	
Attribuut naam	Omschrijving
+ id : Integer	De identificerende waarde van een meet gegeven.
+ timestamp: Integer	De tijd waarop het meet gegeven is verzameld. Uitdrukkelijk niet de tijd waarop het meetgegeven in de database is ingevoerd.
+ value: Float	De verzamelde waarde die bij de timestamp hoort.

Klasse: Measurement_data_6h	
Een instantie van de Measurement_data_6h klasse representeert de data in diezelfde tabel op database niveau en geeft daarmee toegang tot meet gegevens die tot 6 uur geaggregeerd zijn.	
Attribuut naam	Omschrijving
+ id : Integer	De identificerende waarde van een meet gegeven.
+ timestamp: Integer	De tijd waarop het meet gegeven is verzameld. Uitdrukkelijk niet de tijd waarop het meetgegeven in de database is ingevoerd.
+ value: Float	De verzamelde waarde die bij de timestamp hoort.

Klasse: Measurement_data_1d	
Een instantie van de Measurement_data_1d klasse representeert de data in diezelfde tabel op database niveau en geeft daarmee toegang tot meet gegevens die tot 1 dag geaggregeerd zijn.	
Attribuut naam	Omschrijving
+ id : Integer	De identificerende waarde van een meet gegeven.
+ timestamp: Integer	De tijd waarop het meet gegeven is verzameld. Uitdrukkelijk niet de tijd waarop het meetgegeven in de database is ingevoerd.
+ value: Float	De verzamelde waarde die bij de timestamp hoort.



Figuur 16.8: Grafieken generator tool klassediagram

Klasse: HypericGraph	
Een instantie van de HypericGraph klasse representeert een grafiek.	
Attribuut naam	Omschrijving
<i>+ config: Configuration</i>	Een instantie van de Configuratie klasse. Geeft toegang tot de uitgelezen waarden uit het configuratiebestand om zo benodigde configuratie waarden op de halen.
<i>+ ds: HypericDatasource</i>	Een instantie van de HypericDatasource klasse. Geeft de grafiek toegang tot de datasource om zo de benodigde data uit de datasource op te halen.
<i>+ log: PEAR::Log</i>	Een instantie van de PEAR::Log klasse. Geeft de mogelijkheid om messages te loggen naar een bepaald target.
<i>+ metricId: Integer</i>	Het id van de gewenste meting. De waarde van dit attribuut wordt als argument op de command line meegegeven en komt vanuit de HypericGraphingTool klasse in de HypericGraph klasse terecht.
<i>+ startstamp: Integer</i>	De starttijd van de grafiek. Elke grafiek plot de metrische data binnen een bepaald tijdframe (tussen twee data). De startstamp geeft het begin van dat tijdframe aan.
<i>+ endstamp: Integer</i>	De eindtijd van de grafiek. Zie ook startstamp hierboven. De endstamp geeft het eind van die tijdsspanne aan.
<i>+ warningValue: Double</i>	De waarde waarop de lijn op de grafiek moet worden getekend die aangeeft wanneer een waarschuwings waarde wordt overschreden. De lijn wordt altijd getekend, ook als dat betekent dat de zichtbaarheid van de plotlijn verminderd.
<i>+ criticalValue: Double</i>	De waarde waarop de lijn op de grafiek moet worden getekend die aangeeft wanneer een kritieke waarde wordt overschreden. De lijn wordt altijd getekend, ook als dat betekent dat de zichtbaarheid van de plotlijn verminderd.
<i>+ percentile: Integer</i>	De waarde voor het x -de percentiel.
<i>+ displayUnits : String</i>	De eenheid die op de assen van de grafiek gebruikt zullen worden en waarin de waarden van de grafiek geconverteerd zullen worden.
Methode naam	Omschrijving
<i>+ HypericGraph(config : Configuration, ds : HypericDatasource, log : PEAR::Log, metricId : Integer, startstamp : Integer, endstamp : Integer, warningValue : Double, criticalValue : Double, percentile : Integer, displayUnits : String)</i>	De constructor van de HypericGraph klasse. Neemt een instantie van Configuration, HypericDatasource en PEAR::Log klassen als argument. Daarnaast worden het metricId, de startstamp, endstamp, warningvalue, criticalvalue, percentile en displayUnits meegegeven.

Methode naam	Omschrijving
- <i>timeCallback(aval : Integer)</i>	Deze methode neemt een UNIX timestamp als argument en zet deze om naar een voor mensen leesbare datum in een dd-mm-jjjj uu:mm of een uu:mm formaat, afhankelijk van de tijdspanne. Deze functie wordt alleen intern aangeroepen door de generate() methode.
+ <i>generate()</i>	Deze methode genereert een grafiek op basis van het metricId, de startstamp en endstamp, de warning- en criticalValue en de percentile en displayUnits waarden.

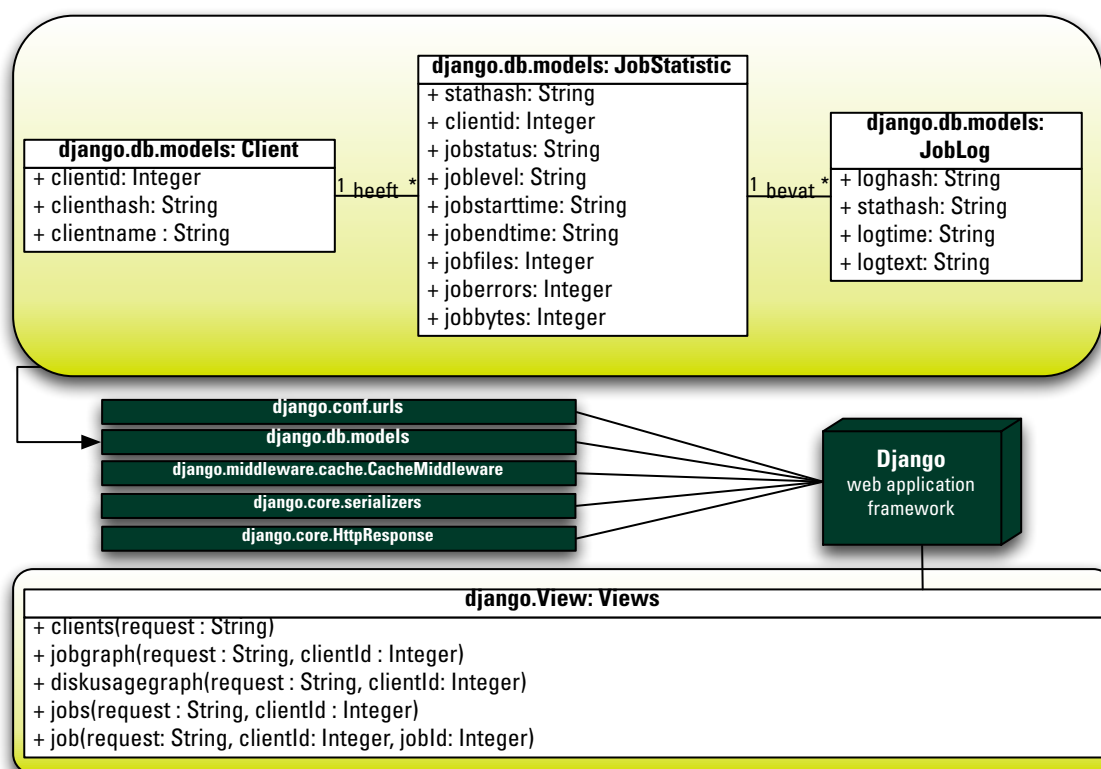
Klasse: HypericDatasource	
Een instantie van deze klasse representeert de datasource. De datasource is e bron waar alle metrische data vandaan zal komen. Dit is in alle gevallen een database. Hyperic biedt de mogelijkheid gebruik te maken van DBMS'en van verschillende fabrikanten, zoals MySQL, PostgreSQL en Oracle. De grafieken generator tool kan ook gebruik maken van deze verschillende DBMS'en.	
Attribuut naam	Omschrijving
- <i>dsConnection</i> : <i>PEAR::MDB2</i>	Een instantie van de MDB2 klasse van PEAR. Door middel van deze klasse kan een verbinding worden gemaakt met een scala aan verschillende DBMSen, waaronder MySQL, PostgreSQL en Oracle.
- <i>dsLog</i> : <i>PEAR::Log</i>	Een instantie van de Log klasse van PEAR. Door middel van deze klasse kunnen log messages worden weggeschreven naar verschillende targets, zoals een bestand, syslog, database e.d.
+ <i>dsType</i> : <i>String</i>	Het type datasource dat gebruikt dient te worden. Het type moet bepaald worden omdat MDB2 die nodig heeft om te bepalen hoe de verbinding met de datasource tot stand moet worden gebracht.
+ <i>dsName</i> : <i>String</i>	De naam van de datasource. In dit geval de naam van de database die aangesproken dient te worden.
+ <i>dsHostname</i> : <i>String</i>	De naam van de host waar de datasource te vinden is en waarop de datasource via TCP te bereiken is.
+ <i>dsHostport</i> : <i>Integer</i>	Het portnummer waarop de connectie naar de host gemaakt dient te worden.
+ <i>dsUsername</i> : <i>String</i>	De naam van de gebruiker waarmee authenticatie bij de datasource plaats dient te vinden
+ <i>dsUserpass</i> : <i>String</i>	Het wachtwoord waarmee, samen met de naam van de gebruiker, authenticatie bij de datasource plaats dient te vinden.

Methode naam	Omschrijving
<i>+ connect()</i>	Methode die connectie met de datasource legt aan de hand van de dsType, dsName, dsHostname, dsHostport, dsUsername en dsUserpass attributen. Het verkregen connectie object wordt toegewezen aan het dsConnecion attribuut.
<i>+ disconnect()</i>	Methode die de connectie met de datasource verbreekt.
<i>+ getMetricData(metricId : Integer, startstamp : Integer, endstamp : Integer)</i>	Methode die aan de hand van de meegekregen attributen metricId, startstamp en endstamp bij de datasource de metrische data opvraagt. De metrische data valt binnen de aangegeven tijdspanne en is allemaal aan een en dezelfde meting gerelateerd.
<i>+ getMetricTemplate(metricId : Integer)</i>	Methode die aan de hand van het meegekregen attribuut metricId, enkele eigenschappen van de opgevraagde metric opvraagt. Dit is noodzakelijk om duidelijkheid te verschaffen over het onderwerp van de grafiek. Voorbeelden van eigenschappen zijn de naam van de metric en de eenheid die de metrische data representeert.
<i>+ getMetricBoundaries(metricId : Integer, startstamp : Integer, endstamp : Integer)</i>	Methode die aan de hand van de meegekregen attributen metricId, startstamp en endstamp bij de datasource enkele eigenschappen van de metrische data opvraagt. De eigenschappen die opgevraagd worden zijn de hoogste waarde, laagste waarde en gemiddelde waarde van de meting binnen de startstamp en endstamp.
<i>+ getAggregateName(metricId : Integer, startstamp : Integer, endstamp : Integer)</i>	Methode die aan de hand van de meegekregen attributen metricId, startstamp en endstamp bepaald welke aggregate gebruikt moet worden. Om efficiënt te blijven aggregaat Hyperic metrische data over een drietal grafieken. De tijdspanne die de grafiek representeert bepaalt uit welke aggregatie tabel de metrische data moet worden opgevraagd. Dit bepaalt de resolutie van de grafiek.

Klasse: Configuration	
Een instantie van deze klasse representeert de configuration zoals deze is uitgelezen uit het configuratiebestand dat als parameter is meegegeven bij het starten van de applicatie. Een instantie van deze klasse kan door de hele applicatie heen worden aangesproken zodat de ingelezen configuratie door de gehele applicatie heen benaderbaar is.	
Attribuut naam	Omschrijving
<i>+ path : String</i>	Het pad op het filesystem dat naar het uit te lezen configuratie bestand wijst.
<i>+ configuration : String[]</i>	Een array gevuld met de key:value waarden. De key:value waarden worden uitgelezen uit het bestand dat in het path attribuut gedefinieerd is.
Methode naam	Omschrijving
<i>+ Configuration(path : String)</i>	Constructor van de Configuration klasse. Neemt het pad naar het uit te lezen configuratie bestand als parameter.
<i>+ readConfiguration()</i>	Methode die aan de hand van de path klassevariabele het aangewezen configuratie bestand uitleest en de uitgelezen waarde in de configuration array plaatst die als klassevariabele gedefinieerd is.
<i>+ getValue(key : String)</i>	Methode die aan de hand van de parameter key de corresponderende value uit de configuration array retourneert.

Klasse: HypericGraphingTool	
Een instantie van deze klasse representeert de Hyperic graphing tool. Vanuit deze klasse worden ondersteunende klassen geïnstantieert en processen in gang gezet.	
Attribuut naam	Omschrijving
<i>+ metricId : String</i>	De meting waarvan een grafiek gegenereerd zal worden.
<i>+ startstamp : Integer</i>	De starttijd van de grafiek. Elke grafiek plot de metrische data binnen een bepaald tijdframe (tussen twee data). De startstamp geeft het begin van dat tijdframe aan.
<i>+ endstamp : Integer</i>	De eindtijd van de grafiek. Zie ook startstamp hierboven. De endstamp geeft het eind van die tijdsspanne aan.
<i>+ warningvalue : Double</i>	De waarde waarop de lijn op de grafiek moet worden getekend die aangeeft wanneer een waarschuwings waarde wordt overschreden. De lijn wordt altijd getekend, ook als dat betekent dat de zichtbaarheid van de plotlijn verminderd.
<i>+ criticalvalue : Double</i>	De waarde waarop de lijn op de grafiek moet worden getekend die aangeeft wanneer een kritieke waarde wordt overschreden. De lijn wordt altijd getekend, ook als dat betekent dat de zichtbaarheid van de plotlijn verminderd.
Methode naam	Omschrijving
<i>+ Main()</i>	De basis routine van de grafieken generator tool. Vanuit deze routine worden ondersteunende klassen geïnstantieert en ondersteunende routines aangeroepen.

16.5 Iteratie 2: Bacula web service



Figuur 16.9: Bacula web service klassediagram

Klasse: Client	
Een client staat gelijk aan een client in Bacula. Dat wil niets meer zeggen dan dat de client gedefinieerd is in de Bacula configuratie en dat er naar alle waarschijnlijkheid backups van de client gemaakt worden.	
Attribuut naam	Omschrijving
+ <i>clientid</i> : Integer	Het identificerende attribuut van een client. Dit attribuut wordt alleen intern binnen de Bacula statistiek applicatie gebruikt. Voor het importeren van statistische gegevens wordt gebruik gemaakt van het clienthash attribuut.
+ <i>clienthash</i> : String	Het identificerende attribuut van de client. In plaats van een oplopende integer waarde is gekozen voor een hash als identificerende waarde. De hash wordt gemaakt op basis van variabelen als client naam en tijd, om te kunnen garanderen dat elke client los te identificeren valt bij het importeren van gegevens.
+ <i>clientname</i> : Integer	De naam van de client.

Klasse: JobStatistic	
Een job statistic staat gelijk aan een uitgevoerde backup job, onafhankelijk van het resultaat van de job. De naam JobStatistic is een beetje ongelukkig gekozen ten tijde van de ontwikkeling van de Bacula Statistiek applicatie. Elke job statistic bevat een aantal feiten over een backup job die gebruikt kunnen worden bij het maken van grafieken en statistieken.	
Attribuut naam	Omschrijving
<i>+ stathash: String</i>	Het identificerende attribuut van de job statistic. In plaats van een oplopende integer waarde is gekozen voor een hash als identificerende waarde. De hash wordt gemaakt op basis van variabelen als job naam en tijd, om te kunnen garanderen dat elke job statistic los te identificeren valt.
<i>+ clientid : Integer</i>	Een verwijzing naar een instantie van de Client klasse.
<i>+ jobstatus : String</i>	De uiteindelijke status van de backup job. De status is uitgedukt in een enkel karakter waarbij er 5 mogelijke statussen zijn. De betekenis van elke mogelijkheid is uitgelegd in de handleiding van de Bacula web service en in de handleiding van Bacula zelf.
<i>+ joblevel: String</i>	De level / niveau waerop de backup job gedraait heeft. De level is uitgedrukt in een enkel karakter waarbij er 3 mogelijkheden zijn. Hierbij moet dan worden gedacht aan levels als: Full, Incremental en Differential. De precieze uitleg van elke level is uitgelegd in de handleiding van de Bacula web service en in de handleiding van Bacula zelf.
<i>+ jobstarttime: String</i>	De tijd waarop de job gestart werd.
<i>+ jobendtime: Sting</i>	De tijd waarop de job eindigde.
<i>+ jobfiles: Integer</i>	Het aantal files dat door de backup job gebackuppeld werd.
<i>+ joberrors: Integer</i>	Het aantal errors die tijdens de backup job naar voren kwamen.
<i>+ jobbytes: Integer</i>	Het aantal bytes dat de backup job heeft verbruikt.

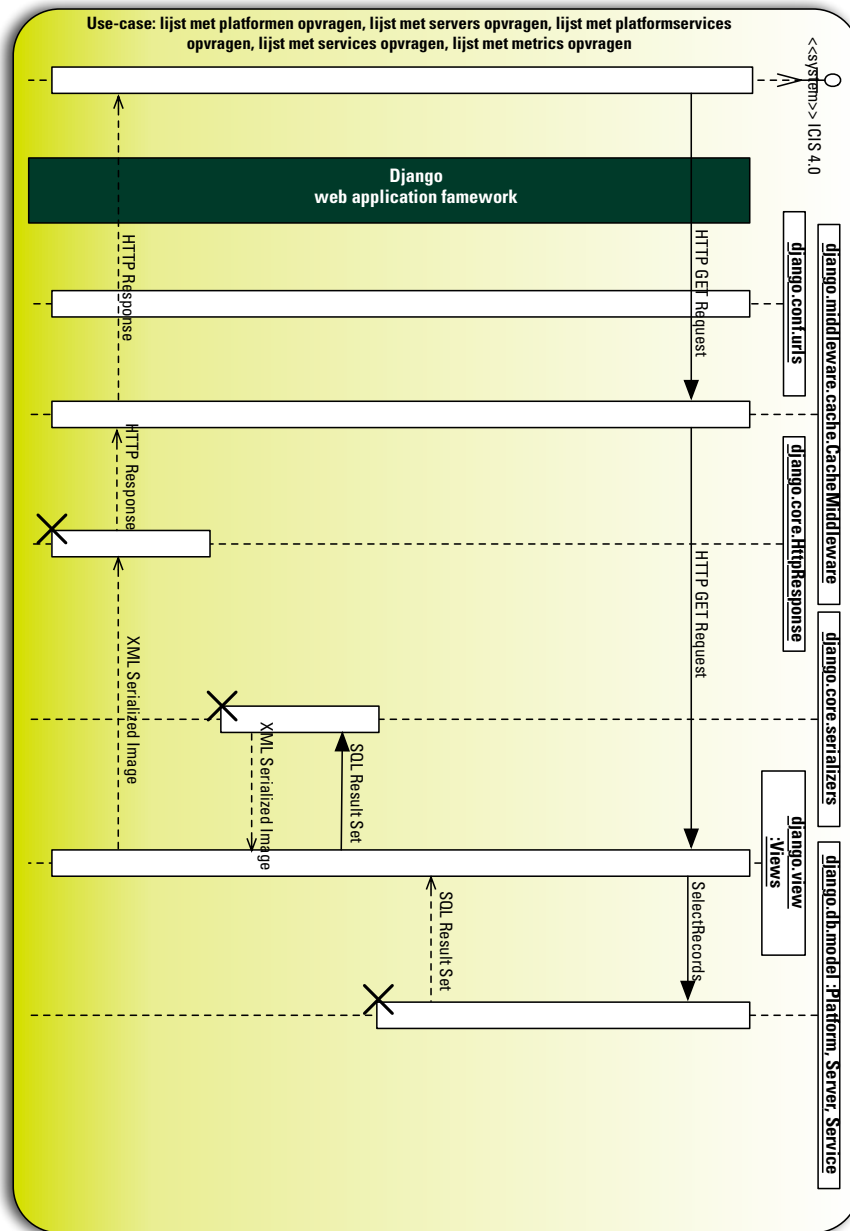
Klasse: JobLog	
Een JobLog staat gelijk aan een log bericht van Bacula. Zo'n bericht bestaat over het algemeen uit een Bacula status bericht.	
Attribuut naam	Omschrijving
<i>+ loghash : String</i>	Het identificerende attribuut van een JobLog. Dit attribuut wordt gebruikt voor het importeren van statistische gegevens.
<i>+ stathash: String</i>	Een verwijzing naar een instantie van JobStatistic.
<i>+ logtime : String</i>	De tijd waarop het log bericht binnen Bacula heeft plaatsgevonden.
<i>+ logtext : String</i>	Het bericht dat Bacula naar de log heeft weggeschreven.

Klasse: View	
Een instantie van de View klasse biedt toegang tot de achterliggende models.	
Method naam	Omschrijving
<i>+ clients(request : String)</i>	Deze methode retourneert een lijst met geconfigureerde Bacula clients.
<i>+ job-graph(request: String, clientid : Integer)</i>	Deze methode retourneert een grafiek waarin de grootte van uitgevoerde jobs in een gewenste periode gevisualiseerd zijn. De benodigde parameters voor het genereren van de grafiek worden uit de request parameter gehaald en kunnen als querystring aan de HTTP requeststring worden meegegeven.
<i>+ diskusage-graph(request : String, clientid : Integer)</i>	Deze methode retourneert een grafiek waarin de grootte van de gebruikte disk space in een gewenste periode gevisualiseerd zijn. De benodigde parameters voor het genereren van de grafiek worden uit de request parameter gehaald en kunnen als querystring aan de HTTP requeststring worden meegegeven.
<i>+ jobs(request : String, clientid : Integer)</i>	Deze methode retourneert een lijst met uitgevoerde backup jobs voor een client.
<i>+ job(request : String, clientid : Integer, jobid : Integer)</i>	Deze methode retourneert een lijst met gedetailleerde informatie over een job met o.a. de bij een job behorende job logs.

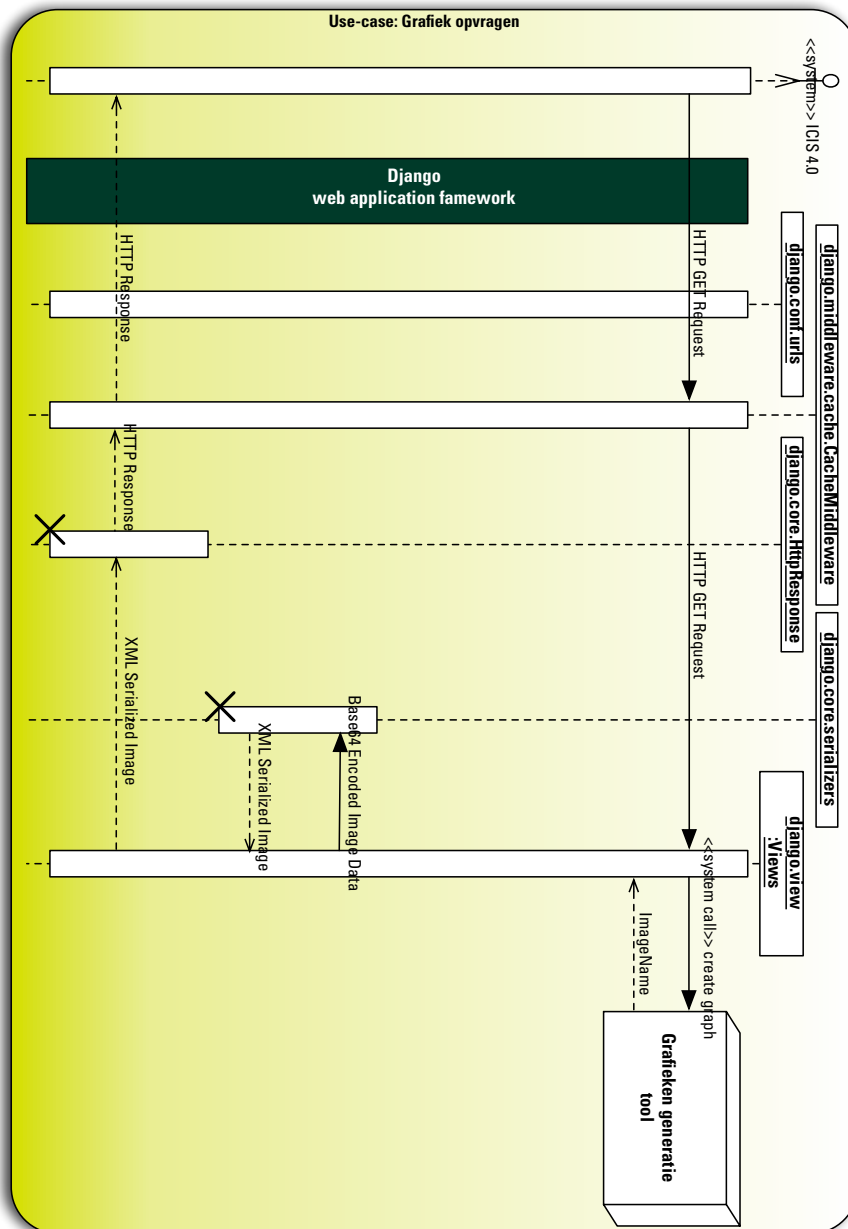
Appendix C: Sequentiediagrammen

Dit hoofdstuk bevat alle sequentiediagrammen uit de drie iteraties die tijdens dit afstuderen aan bod zijn gekomen. Per iteratie zullen opnieuw de bijbehorende sequentiediagrammen worden getoond.

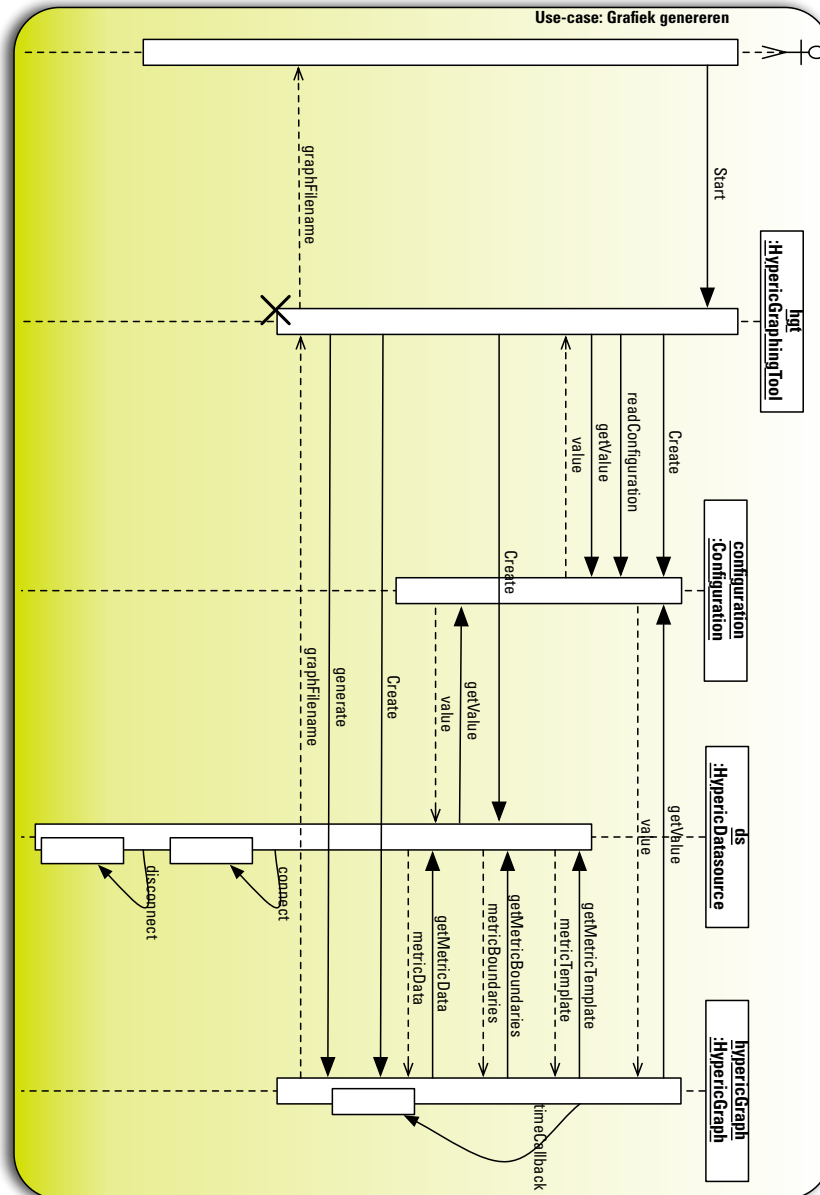
16.6 Iteratie 1: Hyperic web service



Figuur 16.10: Hyperic web service sequentiedigram 1

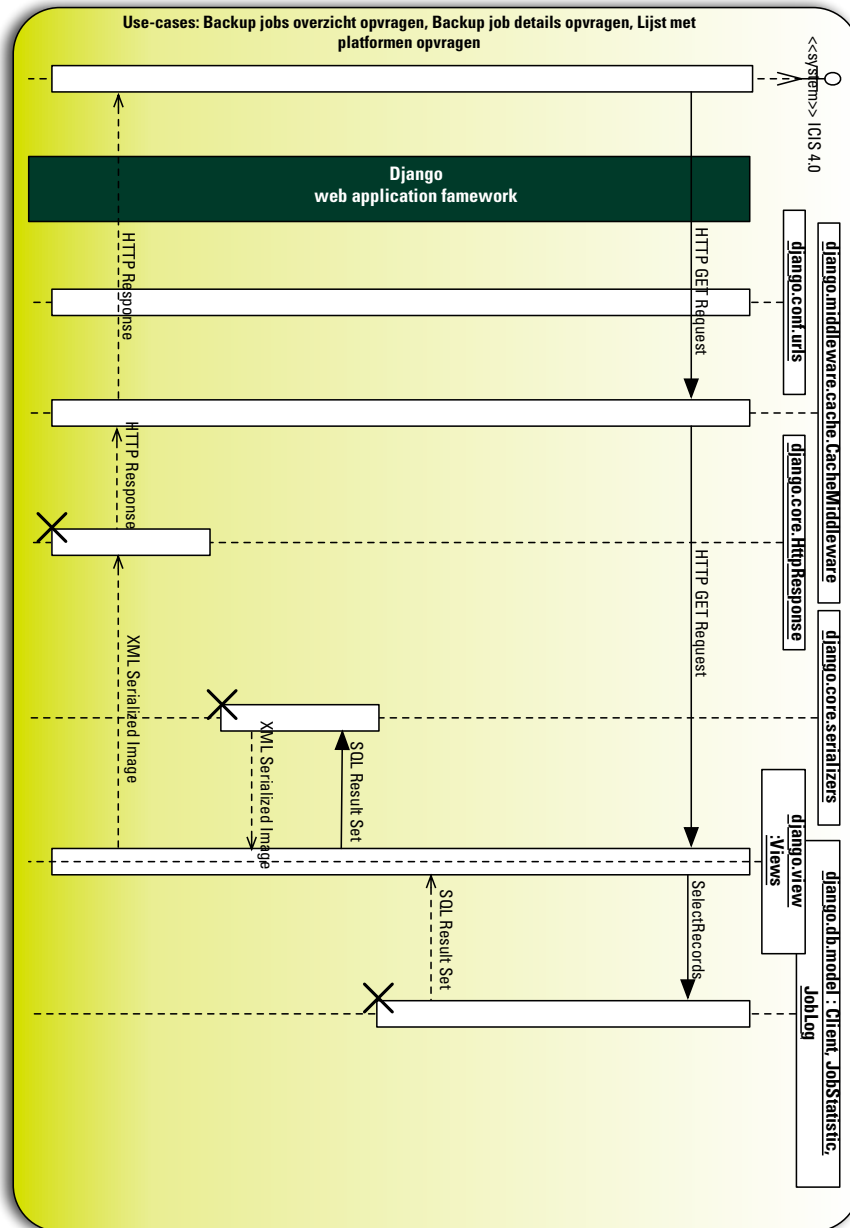


Figuur 16.11: Hyperic web service sequentiediagram 2

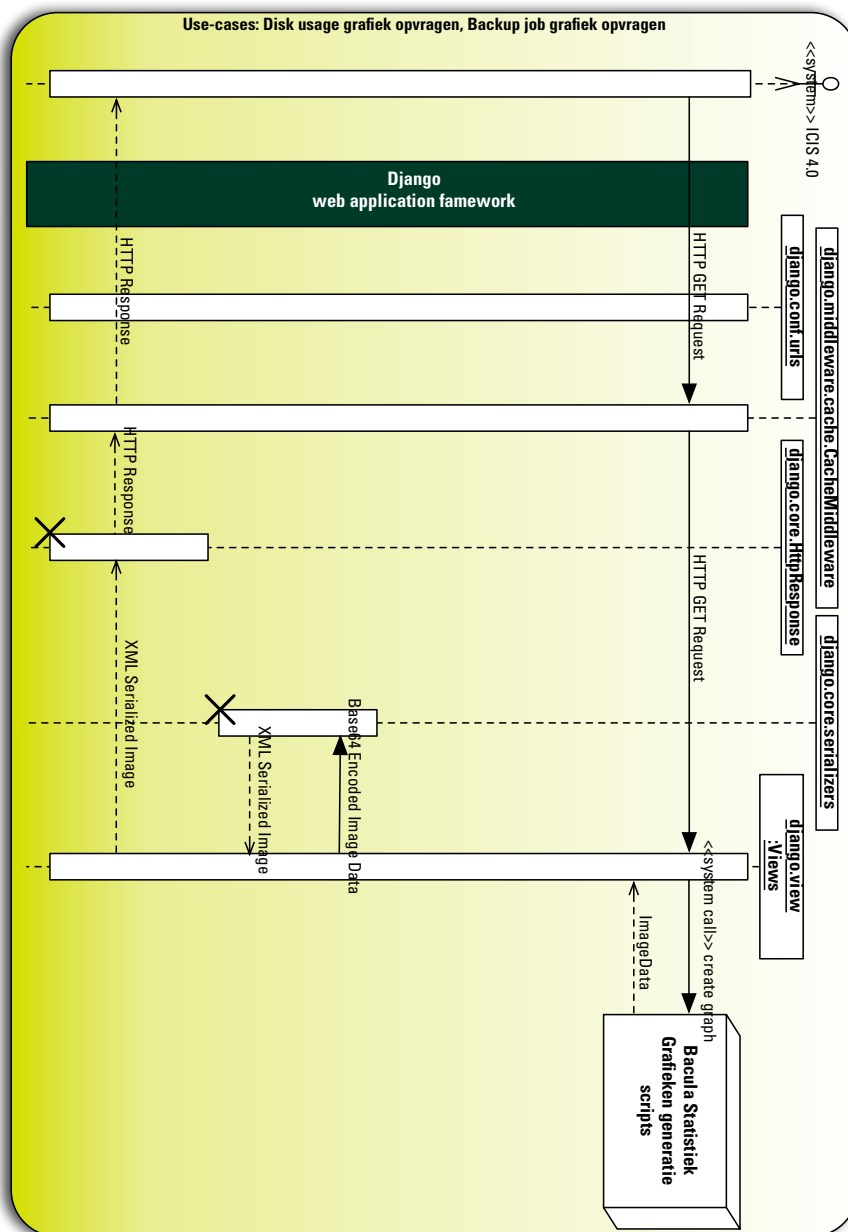


Figuur 16.12: Grafieken generator tool sequentiediagram

16.7 Iteratie 2: Bacula web service



Figuur 16.13: Bacula web service sequentiedigram 1



Figuur 16.14: Bacula web service sequentiedigram 2

Appendix E: Referenties

Apache Axis2

Apache Axis2, Apache Software Foundation, <http://ws.apache.org/axis2/>

Availability reporting using JasperReports

Availability reporting using JasperReports, <http://forums.hyperic.com/jiveforums/thread.jspa?threadID=1845&start=0>

Bacula

Bacula, <http://www.bacula.org>

Biggles

Biggles, Mike Noltam <http://biggles.sourceforge.net>

Building web service the REST way

Building web services the REST way, Roger L. Costello, <http://www.xfront.com/REST-Web-Services.html>

Django

Django, The Django Project, <http://www.djangoproject.com>

JasperReports OpenSource

JasperReports OpenSource, JasperSoft Corporation, <http://www.jasperforge.org>

JpGraph

JpGraph, Aditus Consulting, <http://www.aditus.nu/jpgraph/>

Ploticus

Ploticus, Steve Grubb, <http://ploticus.sourceforge.net>

Praktisch UML, 3e editie

Praktisch UML, 3e editie, Jos Warmer, Anneke Kleppe, Published by Pearson Education Benelux BV, 2005

Project Zero

Project Zero, IBM, <http://www.projectzero.org>

RESTful Web Services

RESTful web services, Leonard Richardson, Sam Ruby, Published by O'Reilly Media, 2007

Ruby on Rails documentation

Ruby on Rails documentation, David Heinemeier Hansson & Community, <http://doc.rubyonrails.org/>