

Pro4all B.V.

Document Synchronisatie

Het synchroniseren van documenten tussen systemen voor documentbeheer.

Auteur: Klaas van Hagen 20066173
Datum: dinsdag 11 mei 2010
Versie: 1.0

Examinatoren: Dhr. E.M. van Doorn
Mevr. H.G.J. Bechet-Tjoonk

Periode: 8 februari 2010 t/m 4 juni 2010
Bedrijf: Pro4all B.V.
Bedrijfsmentor: Dhr. A.W.A. van Deenen

Referaat

Dit document bevat het procesverslag van de ontwikkeling van een synchronisatiesysteem voor documentmanagementsystemen. De ontwikkeling van dit systeem is uitgevoerd in de context van het afstudeerproject van student Klaas van Hagen aan de opleiding Informatica van De Haagse Hogeschool. De stage vond plaats bij Pro4all B.V. in Woerden gedurende de periode van 8 februari t/m 4 juni 2010.

Het document is primair geschreven voor de examinatoren en de gecomiteerde.

Kernwoorden:

- Docstream
- Microsoft SharePoint 2010
- Document management
- Synchronisatie
- Microsoft Sync Framework
- UML

Voorwoord

De afgelopen jaar hebben voor mij veel verandering gebracht. Ik heb een periode van groei doorgemaakt, niet alleen professioneel maar zeker ook op persoonlijk gebied.

Mijn opleiding heeft hier een groot aandeel in gehad, maar ook mijn intrede in de bergsport heeft aan de basis gelegen voor veel positieve ontwikkelingen in mijn leven tijdens de afgelopen vier jaar

Ik wil graag alle mensen bedanken die direct of indirect hebben bijgedragen aan het tot stand komen van dit verslag en de uitvoer van het project. In het bijzonder Antal van Deenen, Raymond Pijpers, Moniek de Groot en Pieter Dijker.

Samenvatting

In de periode van 8 februari tot 4 juni 2010 heb ik in een project uitgevoerd bij Pro4all B.V. te woerden. Ik heb het project uitgevoerd in het kader van het afronden van mijn opleiding Informatica aan de Haagse Hogeschool.

Ik heb in deze periode een architectuur ontwikkeld om documenten te synchroniseren tussen systemen voor documentbeheer. Voor deze architectuur is een proof-of-concept ontwikkeld.

Inhoudopgave

1	Inleiding	1
2	Pro4all B.V.	2
2.1	Organisatie	2
2.2	Positie binnen de organisatie	2
3	Opdracht.....	4
3.1	Aanleiding.....	4
3.2	Probleemstelling.....	4
3.3	Doelstelling.....	4
4	Plan van Aanpak	5
4.1	Uitgangssituatie.....	5
4.2	Werkzaamheden	5
4.3	Methoden en technieken	6
4.4	Producten	6
4.5	Risicofactoren	7
4.6	Planning	8
5	Oriënteren op Microsoft SharePoint 2010 ontwikkeling	9
5.1	Aanpak.....	9
5.2	Wat is Microsoft SharePoint 2010?	9
5.3	Wat kan Microsoft SharePoint 2010?	9
5.4	Hoe wordt Microsoft SharePoint 2010 gebruikt?	11
5.5	Hoe kan ik Microsoft SharePoint 2010 uitbreiden?	11
6	Onderzoeken van mogelijkheden tot synchronisatie.....	13
6.1	Ontwerpen van het onderzoek	13
6.2	Verzamelen en analyseren van literatuur	13
6.3	Bepalen van eisen en wensen aan synchronisatie.	14
6.4	Ontwikkeling van het advies	15
6.5	Rapporteren	20
7	Ontwikkelen van het proof of concept.....	21
7.1	Opstellen van het master-testplan.....	22
7.2	Opstellen van het systeemtestplan	23
7.3	Opstellen van het acceptatietestplan	24
7.4	Opstellen van requirements.....	24
7.5	Ontwikkelen van de SyncServer	25

7.6	Ontwikkelen van de SyncAgent implementaties.....	30
7.7	Ontwikkelen van een eigen implementatie van de MetadataStore	33
8	Uitvoeren analyse impact op Docstream	36
9	Evaluatie	37
9.1	Procesgang	37
9.2	Producten	37
9.3	Beroepstaken.....	37

1 Inleiding

In dit document breng ik verslag uit over mijn werkzaamheden tijdens een project dat ik in de periode van acht januari tot en met vier juni 2010 heb uitgevoerd in het kader van de afronding van mijn studie.

In de hoofdstukken twee tot en met vier een beschrijf ik de organisatie, de opdracht en de wijze waarop ik het project heb aangepakt.

In hoofdstuk vijf beschrijf ik de mogelijkheden van Microsoft SharePoint en hoe voor dit product nieuwe functionaliteit ontwikkeld kan worden.

In hoofdstuk zes beschrijf ik mijn werkzaamheden in het kader van een onderzoek naar de mogelijkheden om documenten te synchroniseren tussen Microsoft SharePoint en Docstream.

Het resultaat van dit onderzoek is een architectuur voor het synchroniseren van documenten tussen deze systemen. De ontwikkeling van een proof-of-concept voor deze architectuur vind u in hoofdstuk zeven.

Hoofdstuk acht beschrijft het uitvoeren van een analyse over de impact die het implementeren van de architectuur heeft op Docstream.

Tot slot vind u in hoofdstuk negen mijn evaluatie van het proces, de producten en een verantwoording ten aanzien van de beroepstaken.

2 Pro4all B.V.

2.1 Organisatie

Pro4all is een Nederlands bedrijf dat sinds 2001 actief is op de Nederlandse markt. Pro4all bestaat uit vier partners en acht medewerkers en heeft een totale omvang van twaalf FTE. Pro4all heeft haar wortels in de bouwnijverheid.

Pro4all is actief op het terrein van ICT dienstverlening en Construction management. In het licht van het afstudeerproject is alleen ICT dienstverlening van belang. De producten en diensten die Pro4all levert op het gebied van ICT zijn:

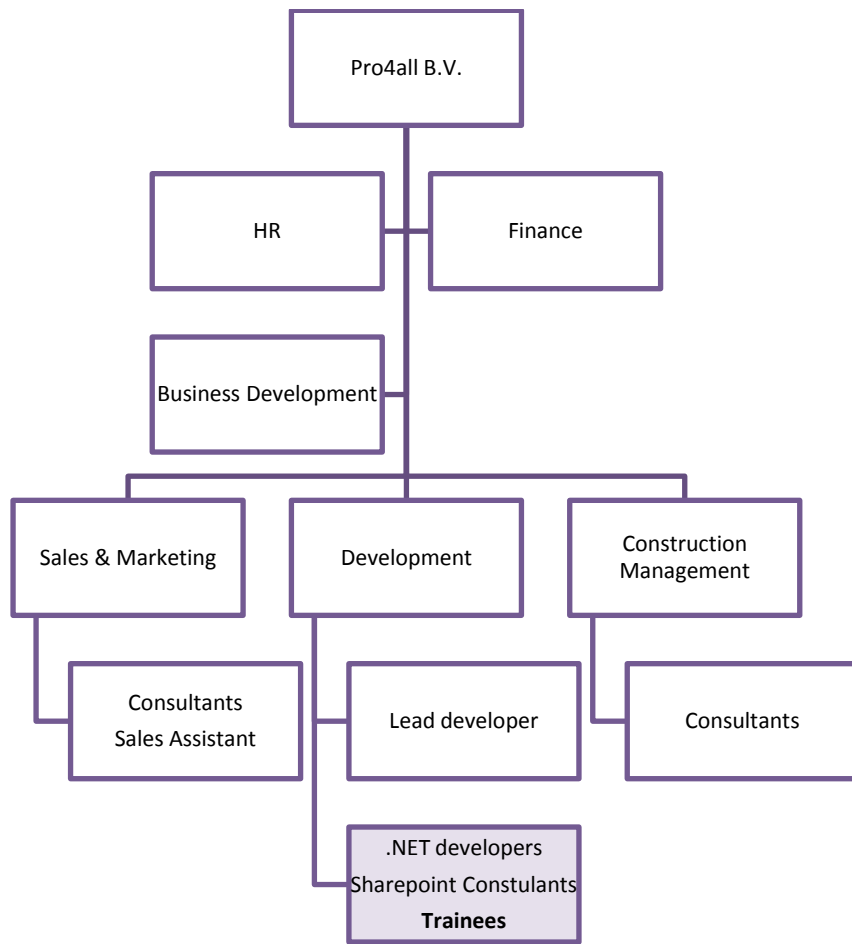
Docstream: Een applicatie die bedrijven in de bouwsector ondersteunt bij bouwprojecten. Het biedt uitgebreide functies voor het beheer van documenten, e-mail verkeer en procedures.

Docstream is vanuit de praktijk gegroeid. De oprichters van Pro4all hebben op basis van eigen ervaring in de bouwsector en een uitgebreid onderzoek de eerste versie van Docstream ontwikkeld. Deze eerste versie, een Windows applicatie, is op basis van feedback van klanten uitgegroeid tot een zeer gecompliceerde applicatie. Later is van Docstream een ASP.NET versie ontwikkeld. Bij deze versie lag de nadruk op de GUI van de applicatie. De ASP.NET versie heeft rijke GUI die het doet voorkomen alsof men met een Windows applicatie werkt in plaats van een webbrowser. Deze versie is in eigen beheer ontwikkeld en wordt met uitzondering van een paar klanten op ASP- basis (Application Service Provider) aangeboden aan klanten.

SharePoint consultancy: Sharepoint is een product van Microsoft en is gepositioneerd als een platform voor samenwerking binnen bedrijven. Pro4all levert consultancy op SharePoint implementaties en ontwikkelt indien nodig functionaliteit op maat.

2.2 Positie binnen de organisatie

De organisatiestructuur van Pro4all is ingericht volgens het onderstaande organogram. Mijn functie in de organisatie gedurende dit project is trainee.



Figuur 1: Organogram Pro4all B.V.

3 Opdracht

3.1 Aanleiding

In hoofdstuk twee heb ik verteld over de producten en diensten die Pro4all levert. In de tweede helft van 2009 is Pro4all op zoek gegaan naar een manier om haar dienstverlening rondom SharePoint uit te breiden. Tijdens mijn eerste gesprekken met Pro4all voor het opstellen van de opdracht heerste bij Pro4all het idee om de functionaliteit van Docstream te implementeren in SharePoint. Dit is de opdracht die ik heb geformuleerd in mijn afstudeerplan.

In de eerste weken na aanvang van het project bleek uit commercieel en technisch oogpunt een andere insteek noodzakelijk. Vanuit mijn eerste oriëntatie op SharePoint bleek dat de originele opdracht te weinig diepgang zou bieden om de beroepstaken op niveau uit te voeren. Na overleg met stakeholders vanuit Business Development, Development en Consultancy voor Docstream en SharePoint is gekozen om drie functionele scenario's voor samenwerking van Docstream en SharePoint verder uit te werken. Deze drie scenario's zijn:

1. Het eenmalig importeren van documenten uit Docstream naar SharePoint:
Door een eenmalige handmatige actie documenten uit Docstream in een SharePoint omgeving importeren.
2. Het geleidelijk importeren van documenten uit Docstream naar SharePoint:
Documenten worden geïmporteert in een SharePoint omgeving wanneer deze opgevraagd worden.
3. Het synchroniseren van documenten tussen Docstream en SharePoint:
Documenten worden tussen Docstream en een SharePoint omgeving consistent gehouden. Dit betekent dat nieuwe documenten die worden toegevoegd, gewijzigd of verwijderd in Docstream ook worden toegevoegd, gewijzigd en verwijderd in een SharePoint omgeving en vice versa.

Nadat de mogelijkheden van elk scenario technisch en functioneel uitgewerkt waren bleek de laatste optie (Het synchroniseren van documenten tussen Docstream en SharePoint) het meest geschikt. Ik heb hiervoor in een scorekaart de voorkeuren van de stakeholders genoteerd, deze scorekaart vind u in bijlage A (Scorekaart functionele scenario's).

3.2 Probleemstelling

Pro4all wil documenten kunnen synchroniseren tussen Docstream en SharePoint. Pro4all beschikt niet over een architectuur om documenten te synchroniseren tussen Docstream en SharePoint.

3.3 Doelstelling

De doelstelling van de opdracht is om een architectuur te ontwikkelen waarmee documenten tussen Docstream en SharePoint gesynchroniseerd kunnen worden. Pro4all wil dat door middel van een proof-of-concept de haalbaarheid van de geadviseerde architectuur wordt aangetoond.

4 Plan van Aanpak

4.1 Uitgangssituatie

Beschikbare software:

- Docstream
- Microsoft SharePoint 2010
- Microsoft Visual Studio 2010
- Microsoft Office 2010
- Microsoft SQL Server 2005 Express
- Microsoft SQL Server Management Studio
- Microsoft Visio 2010
- StarUML v5.0.2

Beschikbare hardware:

- Laptop met Intel Centrino T7500 processor en 3Gb RAM geheugen.

Beschikbare documenten:

- Handleiding Docstream.

Aanwezige ideeën:

- Het heeft de voorkeur om communicatie tussen systemen (indien nodig) d.m.v. webservices te realiseren.

4.2 Werkzaamheden

- **Oriënteren op SharePoint** : Ik heb niet eerder met Microsoft SharePoint gewerkt. Daarom begin ik met mijzelf op dit product te oriënteren. Hierbij zal ik mij hoofdzakelijk richten op ontwikkeling voor SharePoint omdat dit voor deze opdracht het meest relevant is.
- **Opstellen van plan van aanpak**: Om de uitvoer van het project op een gestructureerde wijze aan te pakken zal ik bij aanvang een plan van aanpak opstellen. Hierin beschrijf ik de probleem- en doelstelling, uitgangssituatie, uit te voeren werkzaamheden, te gebruiken methoden en technieken, op te leveren producten risicofactoren en de planning van de werkzaamheden.
- **Onderzoeken van de mogelijkheden voor synchronisatie tussen Docstream en SharePoint**: Om inzicht te krijgen in de verschillende mogelijkheden voor synchronisatie tussen Docstream en SharePoint zal ik hiernaar onderzoek verrichten. Tijdens dit onderzoek zal ik de volgende werkzaamheden uitvoeren:
 - **Opstellen onderzoeksplan**
 - **Zoeken van literatuur**
 - **Analyseren van literatuur**
 - **Ontwikkelen van een voorstel**
 - **Schrijven van een onderzoeksrapport**

- **Ontwikkelen van een proof-of-concept:** Om aan te tonen dat de in het onderzoek geadviseerde methode werkt zal ik een proof-of-concept ontwikkelen. Om het proof-of-concept te ontwikkelen zal ik de volgende werkzaamheden uitvoeren:
 - Opstellen requirements
 - Opstellen van een master testplan.
 - Opstellen use-cases
 - Opstellen van logische en fysieke testgevallen
 - Opstellen klassediagrammen
 - Opstellen relationeel representatiemodel
 - Opstellen relationeel implementatiemodel
 - Opstellen sequencediagrammen voor complexe functies
 - Programmeren van het proof-of-concept in C#
- **Uitvoeren van een impactanalyse voor Docstream:** Om inzicht te krijgen in de impact die het implementeren van het advies uit het onderzoeksrapport zal ik een impactanalyse maken van de wijzigingen die aan Docstream uitgevoerd zullen moeten worden.

4.3 Methoden en technieken

4.3.1 Methoden

Omdat het project uit verschillende onderdelen bestaat zal ik verschillende methoden toepassen. Ik zal het project in vijf fasen uitvoeren, deze fasen zijn:

- **F-1 Voorbereiding:** In deze fase zal ik voorbereidend werk verrichten om fasen F-2 en F-3 te kunnen uitvoeren.
- **F-2 Onderzoek:** In deze fase wordt onderzoek verricht.
- **F-3 Proof-of-Concept:** In deze fase wordt het proof-of-concept ontwikkeld.
- **F-4 Impactanalyse:** In deze fase wordt voor Docstream een impact analyse gemaakt voor het implementeren van de geadviseerde methode.
- **F-5 Afronding:** In deze fase zal ik mij richten op rapportages.

Voor het uitvoeren van het onderzoek baseer ik mijn werkwijze op de methode van Nel Verhoeven die zij beschrijft in haar boek: "Wat is onderzoek?".

4.3.2 Technieken

Voor het ontwerpen van het proof-of-concept heb ik gebruik gemaakt van enkele (modellerings)technieken beschreven in Praktisch UML van Warmer en Kleppe namelijk:

- Use-case en use-case-diagram.
- Klassediagram
- Sequencediagram

4.4 Producten

- Plan van aanpak
- Onderzoeksplan
- Onderzoeksrapport
- Proof-of-Concept:
 - Requirements

- Testplan
- Use-cases
- Use-case diagram
- Klassediagram
- Relatieve representatiemodel
- Relatieve implementatiemodel
- Sequencediagram
- Sourcecode
- Testrapport
- Impactanalyse

4.5 Risicofactoren

Het project is onderhevig aan een aantal risicofactoren:

- **Kennisgebrek**
 - Ik heb nog niet eerder met SharePoint gewerkt, dit houdt in dat het oplossen van eventuele problemen met SharePoint mij meer tijd zal kosten.
 - Om te voorkomen dat mijn gebrek aan kennis voor problemen zorgt ben ik de opdracht begonnen met een oriëntatie op SharePoint. Daarnaast kan ik indien nodig een collega raadplegen.
- **Kinderziektes**
 - De versie van SharePoint waarop ik ga ontwikkelen is SharePoint 2010. Deze versie is nog in beta-fase. Hierdoor bestaat het risico op kinderziektes en ongedocumenteerd gedrag.
 - Dit risico is niet geheel te vermijden, ik zal op dit punt echter zeer alert moeten zijn. Wanneer het vermoeden ontstaat dat een probleem hieraan te wijten is zal ik dit gedrag vergelijken met eventuele vergelijkbare functionaliteit op eerdere versies van SharePoint.
- **Onvolledige documentatie**
 - In het verlengde van de kinderziektes is de documentatie van de beta-versie van SharePoint nog niet volledig en aan wijzigingen onderhevig.
 - Documentatie van voorgaande versies kunnen uitkomst bieden alleen dient correcte werking hiervan wel geverifieerd te worden door middel van tests.

4.6 Planning

		Projectweek:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Fase	Taak	Kalenderweek:	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
F-1	Oriënteren SharePoint 2010																		
	Opstellen Plan van Aanpak																		
F-2	Opstellen Onderzoeksplan																		
	Interviewen																		
	Opstellen requirements																		
	Verzamelen literatuur																		
	Analyseren literatuur																		
	Schrijven onderzoeksrapport																		
F-3	Opstellen requirements (POC)																		
	Opstellen testplan																		
	Opstellen klassediagram (POC)																		
	Opstellen sequentiediagram(men) (POC)																		
	Programmeren en testen (POC)																		
	Testen & opstellen Testrapport (POC)																		
F-4	Uitvoeren impact analyse																		
F-5	Schrijven procesverslag																		

5 Oriënteren op Microsoft SharePoint 2010 ontwikkeling

5.1 Aanpak

Om het project uit te voeren heb ik kennis nodig van SharePoint. Om mijn kennis van SharePoint uit te breiden heb ik vier vragen geformuleerd waarmee ik verwacht voldoende kennis van SharePoint op te doen om het project uit te kunnen voeren. Deze vragen zijn:

1. Wat is Microsoft SharePoint 2010?
2. Wat kan Microsoft SharePoint 2010?
3. Hoe wordt Microsoft SharePoint 2010 gebruikt?
4. Hoe kan ik Microsoft SharePoint 2010 uitbreiden?

Om deze vragen te beantwoorden maak ik gebruik van online-literatuur in de van Microsoft. De bronnen die ik heb gebruikt zijn:

- De SharePoint 2010 homepage
- Het SharePoint Developer Center op MSDN
- Het SharePoint 2010 Tech Center op Microsoft TechNet

Na het beantwoorden van deze vragen heb ik een aantal simpele toepassingen ontwikkeld om “hands-on” ervaring op te doen.

5.2 Wat is Microsoft SharePoint 2010?

Microsoft SharePoint 2010 is een online platform voor samenwerking. Het biedt veel functionaliteit

Deze vraag blijkt erg lastig te beantwoorden. SharePoint wordt door microsoft zelf omschreven door de functionaliteit ervan te beschrijven. De reden dat het zo lastig is om te vertellen wat SharePoint nou eigenlijk is is omdat er geen eenduidig antwoord op te geven is. Technisch gezien is SharePoint een verzameling van websites, webservices en databases die samenwerken om een doel te bereiken. Dit doel is het ondersteunen van menselijke samenwerking door middel van techniek.

SharePoint 2010 staat nog in de kinderschoenen en is officieel gelanceerd op 12 mei j.l.

5.3 Wat kan Microsoft SharePoint 2010?

SharePoint is een platform voor het ondersteunen van menselijke samenwerking gericht op het bedrijfsleven. Een goed voorbeeld in het de toepassing hiervan is de teamsite.

Een teamsite is een website voor een team mensen, bijvoorbeeld een afdeling of een projectgroep. Op een teamsite staan alle zaken die voor dat team van belang zijn. Voorbeelden hiervan zijn documenten, takenlijsten en agenda's maar ook andere zaken zoals KPI's een weblog.

Al deze zaken worden gepersonaliseerd aangeboden waardoor de gebruiker alleen die informatie krijgt voorgeschotelt die relevant is voor zijn/haar functie binnen het team. Dit heet audience targeting en wordt op alle in SharePoint opgeslagen inhoud toegepast.

De mogelijkheden van SharePoint 2010 zijn in zes groepen te verdelen: Sites, Communities, Content, Search, Insights en Composites.

5.3.1 Sites

Sites vormen het uitgangspunt voor SharePoint functionaliteit. Vrijwel alle functionaliteit van SharePoint wordt door middel van een website ontsloten. SharePoint biedt met Sites een uniforme architectuur voor het aanbieden van websites voor intranet, extranet en internet toepassingen.

SharePoint biedt de mogelijkheid om zonder technische kennis websites te bouwen. Webparts zijn losse componenten die door een gebruiker op een webpart pagina geplaatst kunnen worden. Een webpart pagina is een pagina waarin webparts weergegeven kunnen worden. Door webparts op een webpart pagina te plaatsen kunnen gebruikers pagina's maken zonder enige kennis van technieken zoals HTML en ASP. Voorbeelden van webparts zijn een agenda-webpart, een takenlijst-webpart en een figuur-webpart.

Sites in SharePoint worden gepersonaliseerd weergegeven. Dit betekent onder andere dat een gebruiker webparts kan verbergen als hij/zij deze niet gebruikt of een site in zijn of haar eigen taal kan gebruiken. Dit is bijvoorbeeld erg handig voor bedrijven met werknemers van verschillende nationaliteiten.

Ook is het mogelijk om voor een site of een verzameling sites verschillende publieksklassen te definiëren. Op basis van deze klassen kan de weergave van een site worden aangepast. Dit gaat verder dan bijvoorbeeld het kleurenthema van de site maar ook heeft juist ook betrekking op gegevens die via de site beschikbaar worden gesteld. Zo kan een pagina worden geconfigureerd om op basis van een publiekklasse bepaalde webparts te tonen.

5.3.2 Communities

De functionaliteiten onder de noemer communities hebben betrekking op sociale netwerken. Gebruikers kunnen zich met een persoonlijke site, de MySite en een gebruikersprofiel profileren binnen de organisatie.

Met tagging kunnen gebruikers tags (labels) plaatsen op hun profiel en op site-inhoud. Social tagging heeft betrekking op site-inhoud, een tag kan aan site-inhoud toegevoegd worden om te beschrijven wat het is, bevat of doet. Expertise-tagging heeft betrekking op het gebruikersprofiel en kan gebruikt worden om eigenschappen als kennis en affiniteit mee aan te duiden.

Ook wikis en blogs zijn onderdeel van functionaliteit onder de noemer communities.

5.3.3 Content

SharePoint biedt uitgebreide mogelijkheden voor het beheer van documenten en records. Onderdelen hiervan zijn versiebeheer van documenten en verzamelingen van documenten, workflows, en functies voor compliance

Managed metadata maakt het mogelijk om centraal een taxonomie van hiërarchische metadata te beheren. Met de metadatabrowser is het mogelijk om te navigeren op basis van metadata in plaats van een mappenstructuur.

5.3.4 Search

SharePoint maakt het mogelijk om te zoeken naar gegevens binnen het systeem. Door de functionaliteiten uit de communities is het mogelijk om te zoeken op expertise. Zoekopdrachten

worden gepersonaliseerd. Er wordt rekening gehouden met sociale afstand. Zoekopdrachten kunnen gemakkelijk op basis van metadata worden verfijnd.

5.3.5 Insights

Insights omvat de mogelijkheden van SharePoint voor Business Intelligence. Met diensten voor excel en visio kunnen interactieve rapporten en dashboards worden gepubliceerd. Deze kunnen worden gemaakt in de client versies van deze applicaties.

5.3.6 Composites

Met de functionaliteiten in de composites groep kunnen op snelle wijze toepassingen worden gerealiseerd. Functionaliteit kan hiermee worden samengesteld in plaats van ontwikkeld door onderdelen als business connectivity services, access services en silverlight te combineren.

5.4 Hoe wordt Microsoft SharePoint 2010 gebruikt?

SharePoint biedt erg veel mogelijkheden. Zoveel dat het gebruik ervan niet buiten de context van een specifieke implementatie beschreven kan worden. De manier waarop SharePoint binnen de context van een implementatie wordt gebruikt is een keuze die gemaakt wordt door de klant.

Veel voorkomende scenario's zijn intranet sites, team sites en documentbeheer. In beperkte mate worden SharePoint gebruikt voor het aanbieden van reguliere websites (bijvoorbeeld voor de homepage van een bedrijf) en als business intelligence platform.

5.5 Hoe kan ik Microsoft SharePoint 2010 uitbreiden?

SharePoint is uit te breiden door inhoudstypes, kolommen, lijsten, workflows en pagina's te maken. Dit kan men doen op drie mogelijkheden: met een webbrowser, met SharePoint Designer 2010 of met Visual Studio 2010.

Met een webbrowser is het mogelijk om eenvoudige uitbreidingen tot stand te brengen zoals het maken van een nieuw content type, een nieuwe kolom of nieuwe paginas.

Met SharePoint Designer 2010 zijn de mogelijkheden een stuk uitgebreider. Hiermee kunnen bijvoorbeeld ook workflows gemaakt worden en externe databronnen worden geconfigureerd waarna die binnen SharePoint als lijst benaderd kunnen worden. In SharePoint Designer 2010 kunnen naast standaard webpart pagina's ook eigen HTML en ASP.NET pagina's en master pages gemaakt worden.

Met Visual Studio 2010 kunnen alle bovengenoemde onderdelen ontwikkeld worden. Daarnaast kunnen met Visual Studio 2010 ook webparts, event receivers en workflows met programmacode gemaakt worden. Hiermee kan nieuwe functionaliteit worden toegevoegd aan SharePoint. Bij gebruik van een webbrowser of SharePoint Designer 2010 zijn de mogelijkheden voor complexe logica beperkt. In Visual Studio 2010 is deze beperking er niet. Ontwikkelaars kunnen door overerving het objectmodel van SharePoint 2010 uitbreiden en zo hierin eigen logica verwerken.

5.5.1 Experimenten

Omdat ik mij hoofdzakelijk heb geïnteresseerd in ontwikkeling voor SharePoint heb ik met behulp van Visual Studio 2010 een aantal experimenten uitgevoerd om "hands-on" ervaring op te doen. Ik heb drie componenten gemaakt: een webpart, een event receiver en een sequential workflow. Ik heb

voor deze componenten gekozen omdat ik verwachtte hiermee het meest in aanraking te komen. Daarnaast bieden ze de meeste mogelijkheden om functionaliteit uit te breiden.

Webpart

Een webpart is een herbruikbaar grafisch component dat in SharePoint pagina's opgenomen kan worden. Ik wilde met het maken van de webpart voornamelijk ervaren hoe ik het objectmodel van SharePoint kan gebruiken om gegevens in lijsten te bewerken. De webpart die ik heb gemaakt slaat de naam, start- en eindtijd van een taak op in een lijst.

Sinds SharePoint 2010 zijn er twee soorten webparts: een visueel webpart en een 'gewoon' webpart. Het visuele webpart biedt de mogelijkheid om de grafische kant van het webpart te 'tekenen' met behulp van een ontwerptool (onderdeel van Visual Studio 2010). In een 'gewoon' webpart moet alle inhoud vanuit code gegenereerd worden. Deze laatste optie is minder overzichtelijk maar biedt meer flexibiliteit. Ik heb voor mijn experiment-webpart een 'gewoon' webpart gemaakt. Ik had al wat voorkennis m.b.t. het ontwikkelen van ASP.NET webcontrols die grotendeels op dezelfde manier ontwikkeld worden.

Event receiver

Een event receiver is vergelijkbaar met een database-trigger. Ze bieden de mogelijkheid te reageren op gebeurtenissen in het systeem. Op het moment dat ik deze event receiver heb ontwikkeld was nog sprake van de originele opdracht zoals beschreven in het afstudeerplan. Één verschil tussen SharePoint en Docstream is de manier waarop rechten worden toegewezen. In Docstream worden per map de rechten aangegeven voor alle documenten in die map. In SharePoint worden rechten toegekend per document. Ik heb een event receiver ontwikkeld die de manier van rechtentoewijzing van Docstream toepast op SharePoint.

Workflow

Workflows kunnen in SharePoint gebruikt worden om standaard procedures te automatiseren. Ze kunnen betrekking hebben op bestanden, lijsten maar ook gebruikers in het systeem. Een voorbeeld van een workflow is de request feedback workflow. Deze kan een gebruiker starten voor een document en daarmee gebruikers om feedback te vragen aan collega's. Workflows kunnen automatisch en handmatig worden gestart. Een workflow wordt asynchroon uitgevoerd, zo kan het voorkomen dat een workflow pas enkele minuten nadat de opdracht hiervoor is gegeven uitgevoerd wordt. Ik heb een workflow gemaakt met dezelfde functionaliteit als de event receiver die ik eerder heb ontwikkeld om te ervaren wat de verschillen zijn.

6 Onderzoeken van mogelijkheden tot synchronisatie

Om inzicht te krijgen in de mogelijkheden voor het synchroniseren van documenten tussen SharePoint en Docstream heb ik een onderzoek uitgevoerd.

Voor het uitvoeren van het onderzoek heb ik een methode gehanteert die gebaseerd is op de methode die Nel Verhoeven presenteert in haar boek *Wat is onderzoek?*. Deze aanpak heb ik eerder naar tevredenheid gebruikt.

De methode verdeelt het onderzoek in vier fasen: ontwerpen, verzamelen, analyseren en evalueren. Ik heb de fasen verzamelen en analyseren per deelvraag uitgevoerd.

6.1 Ontwerpen van het onderzoek

In de ontwerpfase van het onderzoek voert de onderzoeker een probleemanalyse uit, maakt een onderzoeksvoorstel en stelt een onderzoeksplan op. Ik heb geen specifieke probleemanalyse en onderzoeksvoorstel gemaakt omdat ik het probleem al in het plan van aanpak heb beschreven. In plaats hiervan ben ik begonnen met het onderzoeksplan. Dit heb ik ter goedkeuring aangeboden aan de opdrachtgever.

6.1.1 Hoofd- en deelvragen

In het onderzoeksplan heb ik de hoofd- en deelvragen geformuleerd. Dit heb ik naar eigen inzicht gedaan. De hoofdvraag van het onderzoek is:

Hoe kan synchronisatie van documenten en metadata tussen Microsoft SharePoint en Docstream gerealiseerd worden?

Om deze vraag te beantwoorden heb ik de volgende deelvragen gedefiniëerd:

1. Welke eisen worden er vanuit Pro4all aan de synchronisatie gesteld?
2. Welke best-practices zijn er m.b.t. data synchronisatie?
3. Welke mogelijkheden biedt SharePoint 2010 om met andere pakketten te synchroniseren?
4. Welke mogelijkheden biedt Docstream om met andere pakketten te synchroniseren?

6.1.2 Onderzoeksontwerp

Zowel de hoofdvraag als alle deelvragen zijn kwalitatief van aard. Er is echter wel een verschil in de manier van gegevens verzamelen. Deelvragen 2 en 3 kunnen met een literatuuronderzoek beantwoord worden maar deelvragen 1 en 4 niet. Die heb ik beantwoord door het afnemen van interviews.

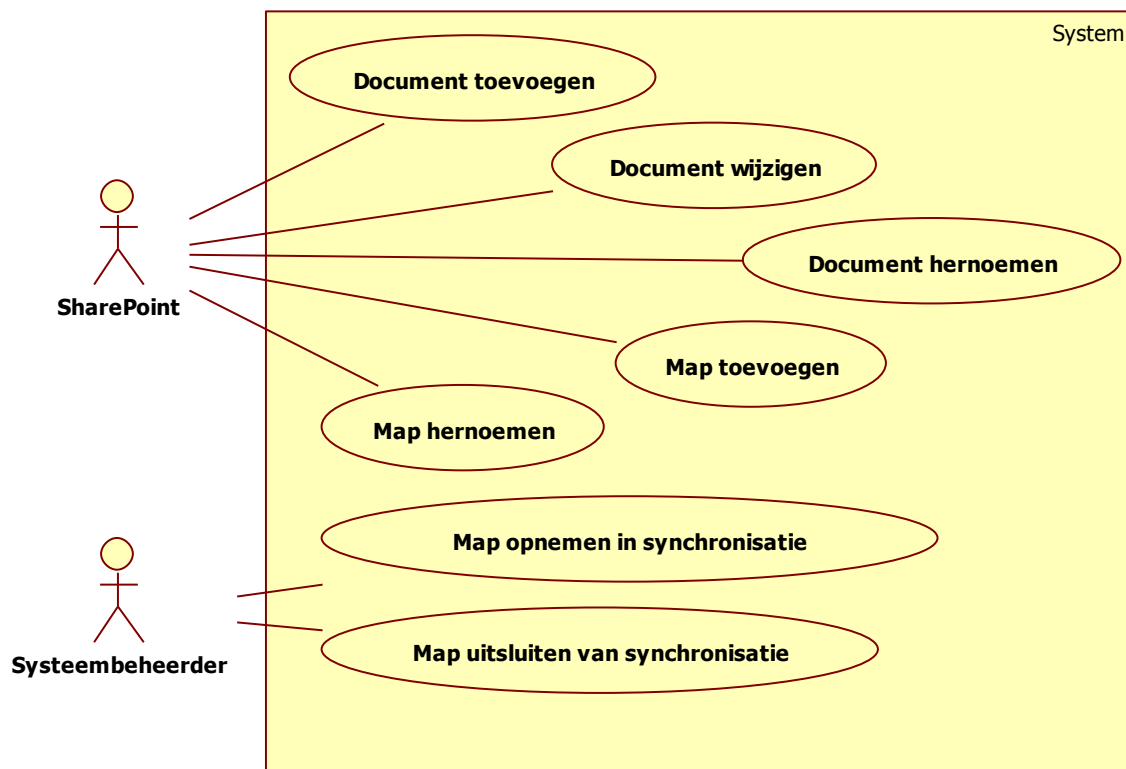
6.2 Verzamelen en analyseren van literatuur

Voor het literatuur onderdeel van het onderzoek heb ik veelvuldig gebruik gemaakt van digitale bronnen. Dit heeft mogelijk een nadelige invloed hebben op de herhaalbaarheid van het onderzoek. Met name doordat de documentatie van SharePoint 2010 op het moment van onderzoeken net als het product zelf nog in beta-toestand verkeerde. Op het moment van onderzoeken was er echter geen andere literatuur beschikbaar. Ik heb geprobeerd de impact hiervan te beperken door de gevonden documentatie te vergelijken met die van SharePoint 2007. Deze strategie is succesvol gebleken.

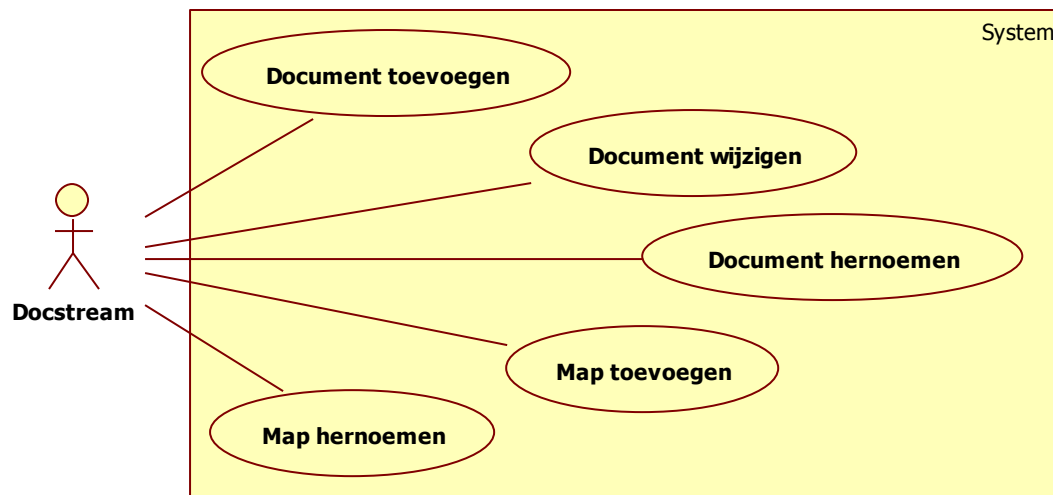
6.3 Bepalen van eisen en wensen aan synchronisatie.

Om de requirements van de synchronisatie te bepalen heb ik veel gesprekken gevoerd met verschillende stakeholders. Omdat ik gelijktijdig met het opstellen van requirements ook literatuur heb verzameld en geanalyseerd is tijdens het opstellen van de requirements in grote mate sprake geweest van voortschrijdend inzicht.

Om de functionele requirements op te kunnen stellen ben ik begonnen met het opstellen van use-cases naar aanleiding van gesprekken met de stakeholders. Op dit punt tijdens het proces was er nog maar weinig bekend van de mogelijkheden van SharePoint om te synchroniseren. Ik vermoedde dat in één of beide producten (Docstream en SharePoint) componenten geïntegreerd zouden moeten worden om de synchronisatie tot stand te laten komen. In deze opzet zouden beide systemen rechtstreeks met elkaar communiceren en zodoende actor van elkaar zijn. De use cases die hieruit zijn ontstaan zijn vind u in figuur 2 en 3. Figuur 2 bevat het initiële use-case diagram voor een uitbreiding op Docstream. Figuur 3 bevat het initiële use-case diagram voor een uitbreiding op SharePoint.



Figuur 2: Initiële use-case diagram voor een uitbreiding op Docstream.



Figuur 3: Initiële use-case diagram voor een uitbreiding op SharePoint.

Tijdens het verzamelen en analyseren van literatuur over de mogelijkheden van SharePoint om met andere software te synchroniseren ben ik met het Microsoft Sync Framework in aanraking gekomen. Dit is erg belangrijk gebleken voor het verdere verloop van het project. Omdat ik meer wilde weten over de mogelijkheden van deze technologie heb ik een aantal prototypes gemaakt. Kleine programmatjes om te bepalen of deze technologie te gebruiken is.

Ik heb Dhr. Van Deenen vrijwel dagelijks mondeling op de hoogte gehouden rondom de vorderingen van het onderzoek en daarmee de mogelijkheden van het Microsoft Sync Framework. In veel gevallen heeft dit geleid tot discussies waarmee door het nieuw verkregen inzicht de requirements aan het systeem veranderde. Om te controleren dat ik de wijzigingen goed heb begrepen heb ik deze na verwerking in de requirements met de stakeholders overlegd.

De ontwikkeling die tijdens het bepalen van de requirements heeft plaatsgevonden is het verbreden van de horizon met betrekking tot de systemen waarmee gesynchroniseerd wordt. De eerste gedachten waren beperkt tot het synchroniseren van Docstream en SharePoint. De huidige visie omvat synchronisatie van systemen voor documentbeheer in het algemeen. Mijn bijdrage aan dit groeiproces is adviserend met betrekking tot de techniek en creatief met betrekking tot de toepassing hiervan.

6.4 Ontwikkeling van het advies

In paragraaf (§6.3) heb ik beschreven hoe ik door de toename van inzicht in de technologische mogelijkheden een proces heb doorlopen waardoor de visie op het resultaat is geëvolueerd tot een systeem waarmee documenten tussen verschillende DMS-en gesynchroniseerd kunnen worden. In deze paragraaf beschrijf ik het verloop van dit proces en de keuzes die hierbij gemaakt zijn.

6.4.1 Terug naar het begin

Zoals ik in de aanleiding voor de opdracht (paragraaf §3.1) heb beschreven was Pro4all zoekende naar een manier om de activiteiten rondom SharePoint ontwikkeling uit te breiden. Dit heeft geleid tot de originele opdracht, het implementeren van functionaliteit van Docstream in SharePoint 2010.

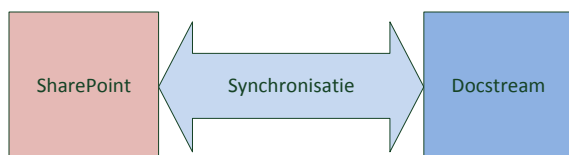
In de eerste weken na aanvang van het project bleek vanuit commercieel en technisch oogpunt een andere insteek noodzakelijk. Vanuit mijn eerste oriëntatie op SharePoint bleek dat de originele opdracht te weinig diepgang zou bieden om de beroepstaken op niveau uit te voeren. Na overleg met stakeholders vanuit Business Development, Development en Consultancy voor Docstream en SharePoint is gekozen om drie functionele scenario's voor samenwerking van Docstream en SharePoint verder uit te werken. Deze drie scenario's zijn.

1. Het eenmalig importeren van documenten en –metadata van Docstream naar SharePoint.
2. Het geleidelijk importeren van documenten en –metadata van Docstream naar SharePoint, documenten en –metadata worden geïmporteert naar SharePoint wanneer deze opgevraagd worden.
3. Het synchroniseren van documenten en –metadata tussen Docstream en SharePoint.

Nadat de mogelijkheden van elk scenario technisch en functioneel uitgewerkt waren bleek de laatste optie (Het synchroniseren van documenten tussen Docstream en SharePoint) het meest geschikt. Ik heb hiervoor in een scorekaart de voorkeuren van de stakeholders genoteerd, deze scorekaart vind u in bijlage A (Scorekaart functionele scenario's)

6.4.2 Rechtstreekse synchronisatie

Na de beslissing om te proberen synchronisatie van documenten tussen Docstream en SharePoint tot stand te brengen ben ik een onderzoek gestart naar de mogelijkheden hiervoor. In eerste instantie ging ik er vanuit dat de synchronisatie rechtstreeks uitgevoerd zou worden, zoals weergegeven in het onderstaande figuur.



Figuur 4: Rechtstreekse synchronisatie tussen SharePoint en Docstream.

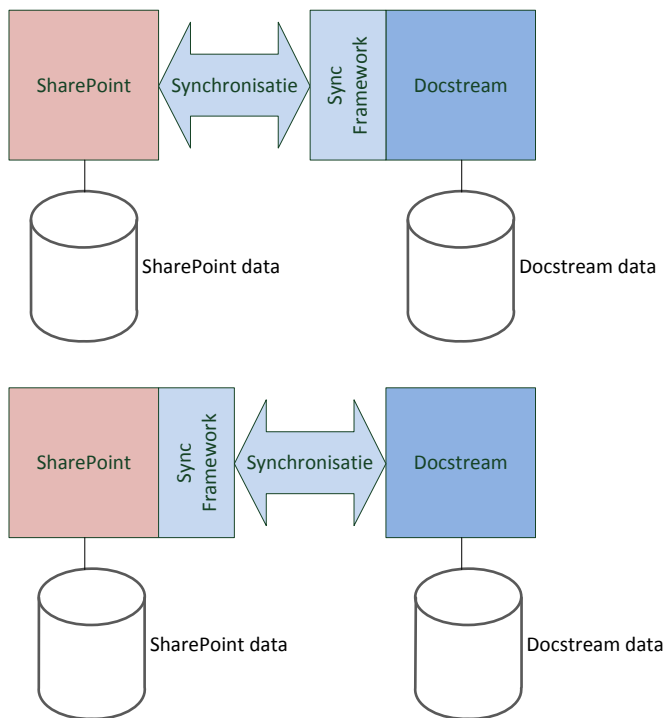
Hierdoor zou SharePoint, Docstream of beiden uitgebreid moeten worden. Tijdens mijn oriëntatie op SharePoint heb ik kennis gemaakt met verschillende technieken om gegevens en documenten in SharePoint te bewerken. Mijn eerste idee voor een oplossing was om met een event receiver op wijzigingen in documentbibliotheken te reageren. Een probleem hiermee is dat event receivers synchroon worden uitgevoerd. Hierdoor kan synchronisatie de reactiesnelheid van SharePoint ernstig schaden.

6.4.3 Rechtstreekse synchronisatie met het Microsoft Sync Framework

Tijdens het verzamelen van literatuur ben ik met het Microsoft Sync Framework in aanraking gekomen. Om de bruikbaarheid van deze technologie in te schatten heb ik een aantal voorbeeld-programma's onderzocht en zelf een prototype ontwikkeld.

De voor- en nadelen van de verschillende methoden van synchronisatie tussen Docstream en SharePoint heb ik in een scorekaart gevat, deze vind u in bijlage B (Scorekaart synchronisatiemogelijkheden SharePoint en Docstream). Hieruit blijkt dat gebruik van het Microsoft SyncFramework het beste aansluit bij de requirements.

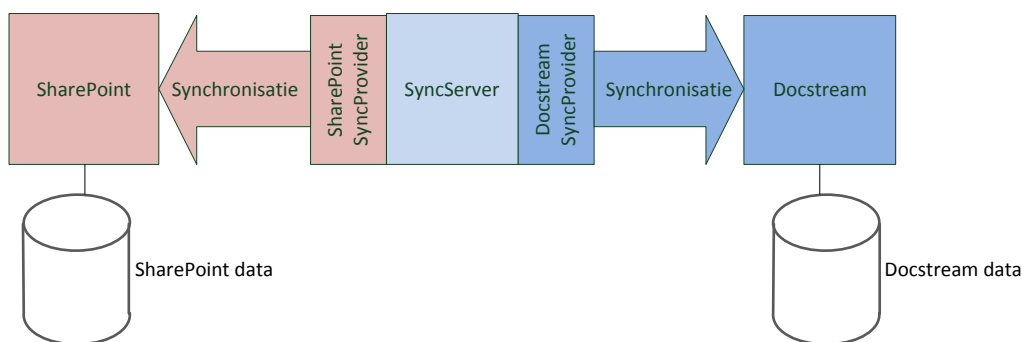
Ik ben in eerste instantie nog steeds uitgegaan van directe synchronisatie tussen Docstream en SharePoint. Hierbij kan het Microsoft Sync Framework ingezet worden aan SharePoint of Docstream zijde.



Figuur 5: Rechtstreekse synchronisatie tussen SharePoint en Docstream met het Microsoft Sync Framework.

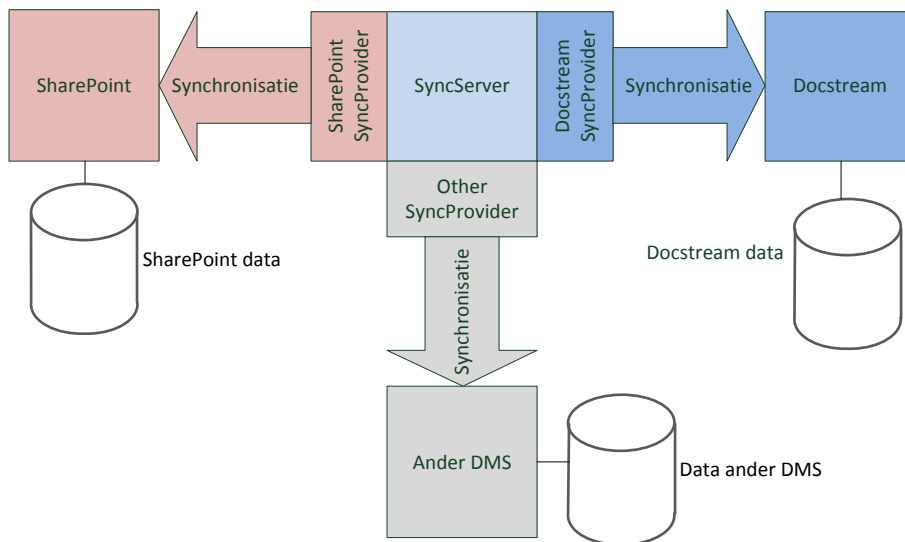
6.4.4 Synchronisatie via de SyncServer

Een nieuwe eis aan het systeem is dat Docstream met meerdere systemen moet kunnen synchroniseren. Dit maakt de synchronisatie complex. Om deze complexere situatie beheersbaar te maken heb ik de SyncServer geïntroduceerd. In dit stadium werd de SyncServer nog gezien als een proxy om synchronisatie tussen twee systemen op te zetten.



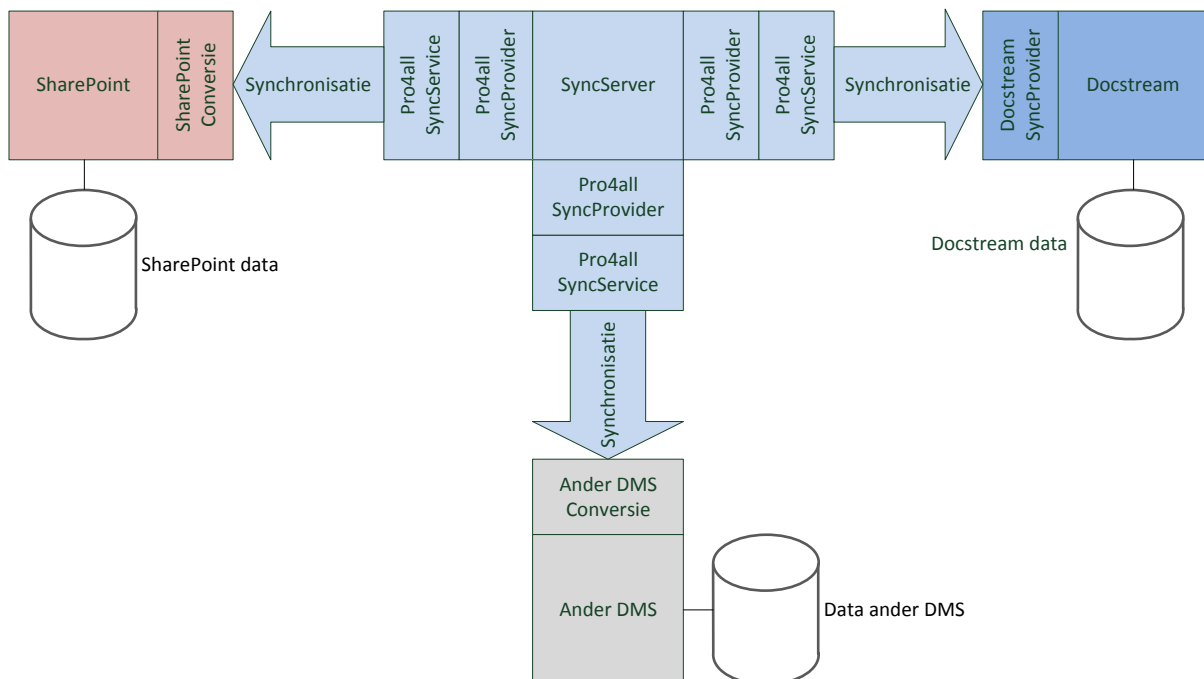
Figuur 6: Synchronisatie van SharePoint en Docstream met een SyncServer.

De introductie van de SyncServer heeft de mogelijkheid geboden om niet alleen Docstream met SharePoint te synchroniseren maar ook Docstream met Docstream of SharePoint met SharePoint. En ieder willekeurig ander DMS wanneer daarvoor een provider wordt ontwikkeld.



Figuur 7: Synchronisatie van SharePoint, Docstream en andere DMS-en meteen SyncServer

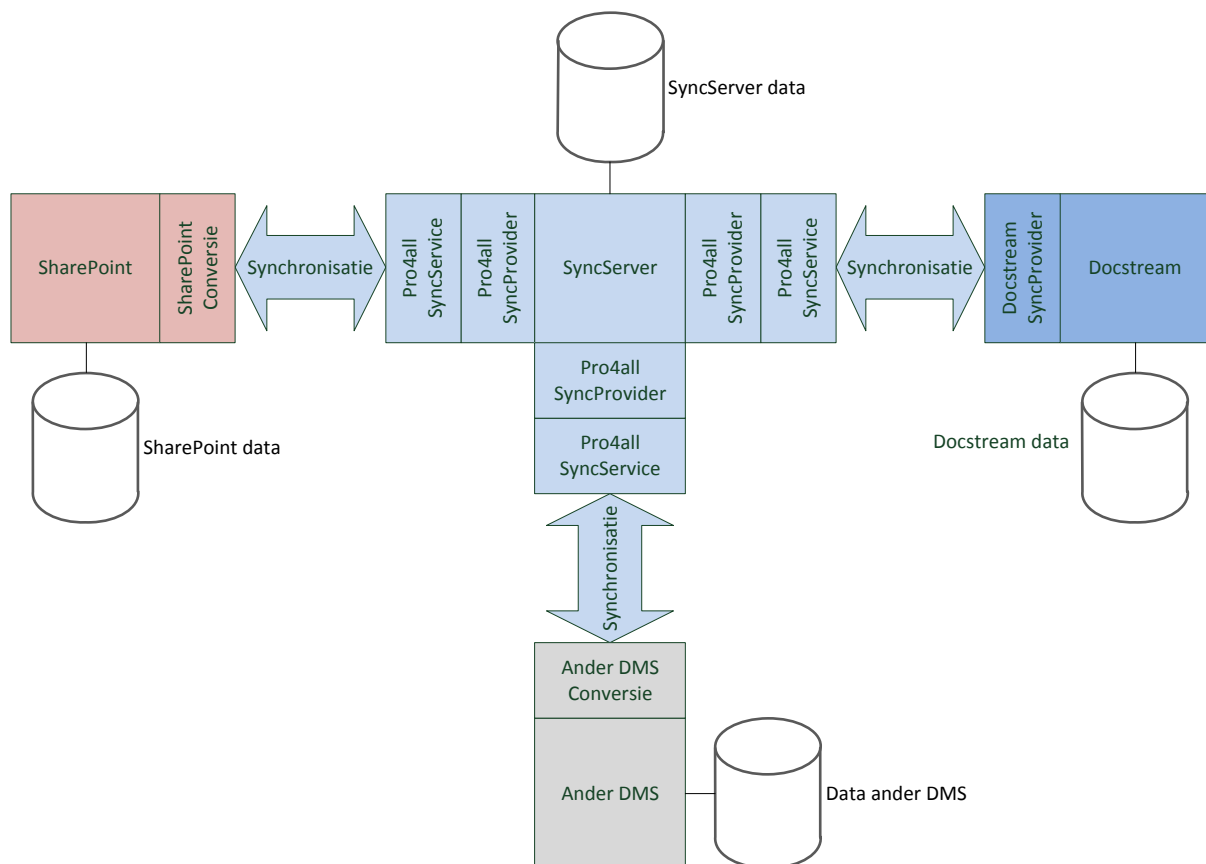
Door deze ontwikkeling zijn we verder na gaan denken over het gemak waarmee andere systemen gekoppeld kunnen worden. Ideaal is het wanneer een koppeling van een ander systeem gemaakt kan worden zonder enige wijziging aan de SyncServer. De SyncServer kan gegevens alleen synchroniseren als deze hetzelfde formaat hebben. Conversie van het formaat van de synchroniserende DMS-en naar het SyncServer formaat gebeurde tot nu toe in de SyncProvider voor dat DMS. Door de verantwoordelijkheid voor conversie van het gegevensformaat te verplaatsen van de SyncProvider naar het DMS wordt het mogelijk om met één type SyncProvider te volstaan. Er moet dan wel een generieke vorm van communicatie zijn tussen de SyncServer en een DMS. Daarom hebben Dhr. Van Deenen en ik besloten om gebruik te maken van webservices. Dit stelt de producent van een DMS om een eigen koppeling te ontwikkelen zonder wijzigingen aan het systeem.



Figuur 8: Synchronisatie met de SyncService.

6.4.5 Meer dan een proxy

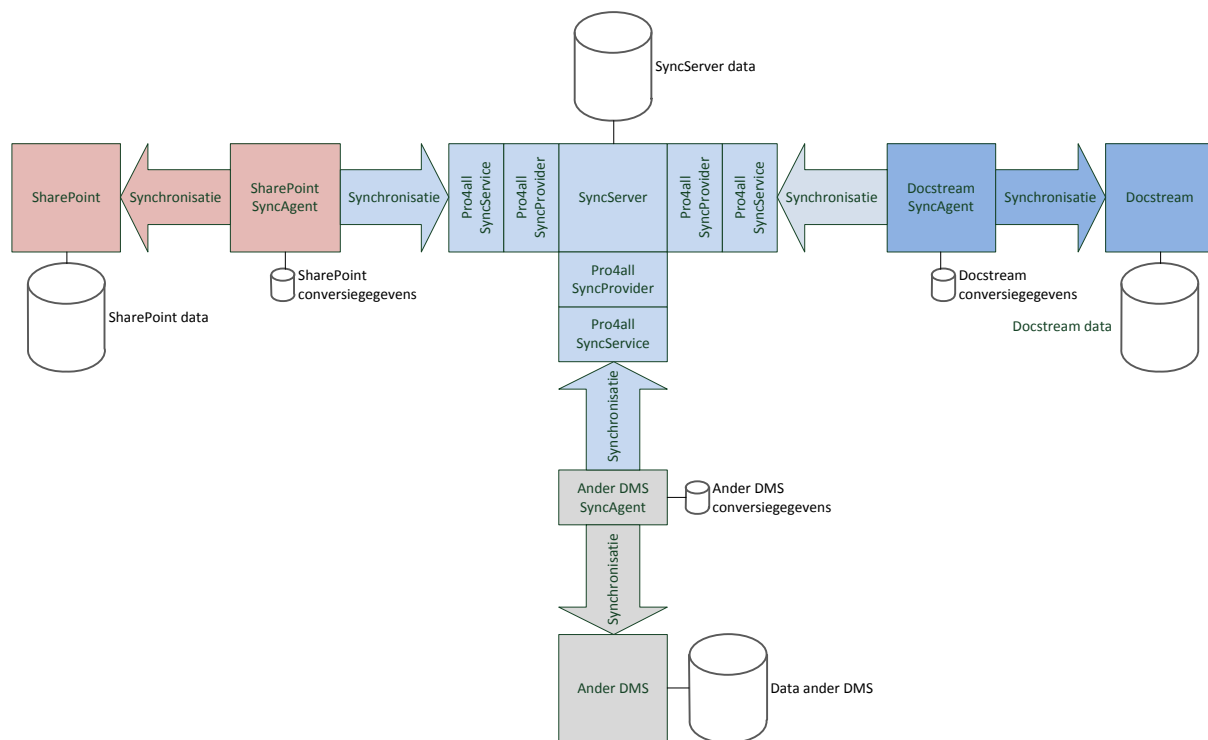
Als de SyncServer slechts als proxy gebruikt wordt is het niet mogelijk om garanties te bieden met betrekking tot de prestaties van de dienst. De snelheid van een synchronisatie is sterk afhankelijk van de prestaties van beide DMS-en. Niet alleen reactiesnelheid van een DMS maar ook de bandbreedte van de verbinding kunnen voor problemen zorgen. Om dit probleem te voorkomen worden alle documenten en –metadata op de SyncServer opgeslagen. Hiermee is de SyncServer een master-replica geworden waarmee ieder gekoppeld DMS onafhankelijk van andere gekoppelde DMS-en kan synchroniseren. Hierdoor heeft Pro4all grip op de kwaliteit van de geleverde dienstverlening en wordt de impact van problemen met één gekoppeld DMS beperkt tot dat DMS.



Figuur 9: SyncServer als master-replica.

6.4.6 Communiceren met de SyncServer via SyncAgent.

SharePoint biedt via een webservice een functie om wijzigingen aan lijsten (en daarmee documentbibliotheken) te enumereren. Om deze functionaliteit te benutten is een applicatie nodig die deze webservice aanspreekt. Deze applicatie, de SyncAgent communiceert met SharePoint en met de SyncServer en heeft daarmee een goede plaats in de architectuur om conversie tussen het gegevensformaat van SharePoint en dat van de SyncServer uit te voeren. De introductie van deze SyncAgent heeft daarnaast het bijkomende voordeel dat deze wederom op een locatie in de technische infrastructuur van de klant geplaatst kan worden van waaruit deze toegang heeft tot zowel SharePoint als de SyncServer. Dit voordeel heeft mij doen besluiten het als algemeen component in de architectuur op te nemen. Per DMS wordt een SyncAgent ontwikkeld die verantwoordelijk is voor het initiëren van het synchronisatieproces en het converteren van het gegeven sformaat van dat DMS naar dat van de SyncServer.



Figuur 10: Synchronisatie met de SyncServer via een SyncAgent.

6.5 Rapporteren

6.5.1 Voortgang

Zoals ik in paragraaf §6.3 heb aangegeven heb ik gedurende het onderzoek met zeer grote regelmaat mondeling rapport uitgebracht aan de stakeholders. Naast deze rapportage heb ik tijdens het wekelijkse SharePoint overleg de leden van de SharePoint projectgroep op de hoogte gehouden door mijn bevindingen te presenteren op een flapover en mondeling toe te lichten.

6.5.2 Onderzoeksrapport

Als afsluiting van het onderzoek naar de mogelijkheden voor synchronisatie heb ik een rapport geschreven met een aanbeveling over de meest geschikte manier van synchronisatie van documenten.

7 Ontwikkelen van het proof of concept

Het proof-of-concept is ontwikkeld om de haalbaarheid van de architectuur aan te tonen. Om dit te bereiken hoeft niet aan alle requirements van de architectuur voldaan te worden. Ik heb daarom een selectie gemaakt uit deze requirements om in het proof-of-concept te implementeren. Deze selectie heb ik gebaseerd op een inschatting van mogelijke knelpunten in de architectuur. Ik heb hierbij rekening gehouden met de beschikbare tijd voor implementatie. De geselecteerde requirements heb ik met de stakeholders overlegd.

Om de architectuur aan te kunnen tonen is het van belang dat alle elementen van de architectuur in het proof-of-concept worden vertegenwoordigd. Binnen de architectuur kunnen van het SyncAgent element verschillende implementaties tegelijkertijd bestaan. Om dit aan te kunnen tonen heb ik hiervan twee implementaties gemaakt. Omdat het idee voor de architectuur is voortgekomen uit de wens om documenten te synchroniseren tussen Docstream en SharePoint. Daarom heb ik voor deze systemen een SyncAgent met beperkte functionaliteit ontwikkeld.

De geïmplementeerde architectuur-elementen zijn:

- SyncServer
 - Syncprovider
 - SyncService
- SharePoint SyncAgent
- Docstream SyncAgent

In mijn originele aanpak en planning ben ik uitgegaan van het ontwikkelen van één applicatie. Nu blijkt dat er drie verschillende applicaties ontwikkeld moeten worden heb mijn planning en aanpak aangepast. Het ontwikkelen van van het proof-of-concept heb ik in drie iteraties gedaan:

1. Ontwikkelen van de SyncServer.
2. Ontwikkelen van de SharePoint SyncAgent.
3. Ontwikkelen van de Docstream SyncAgent.

De volgorde waarin in deze onderdelen heb ontwikkeld is gebaseerd op een inschatting van de hoeveelheid werk en de kans op problemen. SharePoint was tijdens deze opzet nog in beta fase. Tijdens het onderzoek de te gebruiken functionaliteit al heb toegepast in prototype. Hierdoor is de ontwikkeling van de SharePoint SyncAgent voornamelijk het samenvoegen en testen van eerder ontwikkelde code. De ontwikkeling van de Docstream SyncAgent is minder risicovol omdat ik rechtstreeks contact kan onderhouden met de ontwikkelaars van Docstream.

Hiervoor heb ik een aangepaste planning gemaakt. Ik heb besloten een week langer de tijd te nemen voor het ontwikkelen van het proof-of-concept en het uitvoeren van de impact-analyse tegelijk uit te voeren met het programmeren van de Docstream SyncAgent-implementatie.

		Projectweek:	12	13	14	15	16
Iteratie	Taak	Kalenderweek:	17	18	19	20	21
Master testplan	Opstellen master testplan						
SyncServer	Opstellen use-cases en use-case diagrammen						
	Opstellen requirements						
	Opstellen testplan						
	Opstellen klassediagram						
	Opstellen sequentiediagram(men)						
	Programmeren en testen						
	Testen & opstellen Testrapport						
SharePoint SyncAgent	Opstellen use-cases en use-case diagrammen						
	Opstellen requirements						
	Opstellen testplan						
	Opstellen klassediagram						
	Opstellen sequentiediagram(men)						
	Programmeren en testen						
	Testen en opstellen Testrapport						
Docstream SyncAgent	Opstellen use-cases en use-case diagrammen						
	Opstellen requirements						
	Opstellen testplan						
	Opstellen klassediagram						
	Opstellen sequentiediagram(men)						
	Programmeren en testen						
	Testen en opstellen Testrapport						

Figuur 11: Aangepaste planning voor het ontwikkelen van het proof-of-concept.

7.1 Opstellen van het master-testplan.

Om te zorgen dat onderdelen van de architectuur niet dubbel of juist niet worden getest heb ik een master testplan opgesteld. Ik heb hiervoor gebruik gemaakt van de methode die wordt beschreven in Testen volgens Tmap. Ik heb gekozen voor deze methode omdat ik hiermee tijdens mijn studie mee in aanraking ben gekomen. De methode is te gebruiken in combinatie met alle ontwikkelmethoden dus dit vormt geen belemmeringen.

T-Map beschrijft een aantal activiteiten voor het opstellen van een mastertestplan:

1. Opdrachtformulering
2. Oriëntatie
3. Bepalen teststrategie
4. Definiëren organisatie
5. Definiëren infrastructuur
6. Bepalen planning
7. Fixeren mastertestplan

Tijdens de activiteit Oriëntatie oriënteert de auteur van het mastertestplan zich op de (project)organisatie, de doelstelling van het systeemontwikkelpocess, het te ontwikkelen informatiesysteem en de eisen waaraan het systeem moet voldoen. Omdat ik deze documentatie zelf heb geschreven heb ik deze activiteit niet uitgevoerd.

7.1.1 Opdrachtformulering

De opdrachtformulering heb ik aan de hand van de beschreven werkwijze uitgevoerd. De beschrijving van de opdrachtgever en opdrachtnemer heb ik zeer sumier gehouden omdat het een interne opdracht betreft.

Bij het bepalen van de testsoorten waarop het mastertestplan betrekking heeft heb ik besloten het toetsen niet in het mastertestplan op te nemen. Toetsen ofwel het inspecteren van tussenproducten en ontwikkelingsprocessen wordt onder andere gedaan door middel van reviews, inspecties en audits. Door de eenvoud van de projectorganisatie (één persoon) acht ik het niet nodig om deze gestructureerd uit te voeren. In plaats hiervan bespreek ik mijn producten regelmatig met de verschillende stakeholders.

7.1.2 Bepalen teststrategie

De teststrategie heb ik in vier stappen bepaald. Deze stappen zijn: Risicotaxatie, bepalen van kwaliteitsattributen, Bepalen relatief belang kwaliteitsattributen en Toewijzen van kwaliteitsattributen aan testsoorten.

TMap hanteert een eigen set kwaliteitskenmerken. Om verwarring te voorkomen heb ik geen gebruik gemaakt van de TMap kwaliteitskenmerken maar de ISO 9126 standaard voor kwaliteitskenmerken van software te hanteren.

7.1.3 Definiëren organisatie

Omdat het team slechts uit een persoon bestaat is het onzinnig om een uitgebreide testorganisatie met meerdere functies op te stellen. Ik heb gekozen voor de functies Testen en Testmanagement wel in de organisatie op te nemen.

7.1.4 Definiëren infrastructuur

De infrastructuur heb ik gedefinieerd aan de hand van de hard- en software die nodig is om het proof-of-concept te kunnen testen.

7.1.5 Bepalen planning

Omdat de ontwikkeling het proof-of-concept iteratief is heb ik ervoor gekozen om het uitvoeren van de activiteiten parallel aan de ontwikkeling te plannen.

7.1.6 Fixeren mastertestplan

Ik heb het mastertestplan voorgedragen en de inhoud ervan uitgelegd aan de stakeholders. Deze hebben daarop goedkeuring verleend, hiermee is het mastertestplan gefixeerd.

7.2 Opstellen van het systeemtestplan

Om het systeemtestplan op te stellen heb ik gebruik gemaakt van de activiteiten die staan beschreven in de fase planning en beheer van black box testen. Deze activiteiten zijn:

1. Opdrachtformulering
2. Globale intake en studie
3. Vaststellen testbasis
4. Bepalen teststrategie
5. Inrichten organisatie
6. Inrichten testproducten

7. Inrichten beheer
8. Bepalen planning
9. Fixeren testplan

Enkele van deze activiteiten zijn reeds in het mastertestplan aan de orde gekomen. Ik heb deze onderwerpen hierop gebaseerd en waar nodig aangepast.

7.3 Opstellen van het acceptatietestplan

Het acceptatietestplan is vrijwel gelijk aan het systeemplan met uitzondering van de planning, uitvoerenden en de infrastructuur.

Omdat voor de Docstream SyncAgent wijzigingen in Docstream uitgevoerd moeten worden is het helaas niet mogelijk om dit component in de acceptatietests te betrekken.

De werking van de architectuur kan echter nog steeds aangetoond worden door documentbibliotheken van twee verschillende SharePoint omgevingen te synchroniseren. Hiervoor heb ik gebruik gemaakt van een Microsoft BPOS omgeving. Microsoft BPOS is een verzameling van online diensten aangeboden door Microsoft. Een van deze diensten is SharePoint 2010. Deze SharePoint omgeving is niet in beheer van Pro4all en daarom geschikt als omgeving voor de acceptatietest.

De acceptatietest heb ik uitgevoerd in samenwerking met een vertegenwoordiger van de stakeholders.

7.4 Opstellen van requirements

De requirements van het proof-of-concept heb ik gebaseerd op een selectie van de requirements van de architectuur. Ik heb de selectie zo gekozen dat het hoofddoel van de architectuur gedekt is: het synchroniseren van documenten tussen DMS-en..

Aan de requirements heb ik een prioriteit toegekend volgens de MoSCoW methode. Ik had op basis daarvan met de stakeholders kunnen overleggen requirements met een lage prioriteit niet te implementeren als daarvoor te weinig tijd zou zijn.

De geselecteerde geprioriteerde requirements zijn:

Code	Ref	Omschrijving	Prioriteit
F1	F1.3.1	In DS gemaakte documentversies worden in SP toegevoegd.	M
F2	F1.3.2	In SP gemaakte documentversies worden in SP toegevoegd.	M
F3	F1.3.3	In DS gewijzigde documentversies worden in SP bijgewerkt.	M
F4	F1.3.4	In SP gewijzigde documentversies worden in SP bijgewerkt.	M
F5		In DS verwijderde documentversies worden in SP verwijderd.	M
F6		In SP verwijderde documentversies worden in DS verwijderd.	M
F7	F3.1	Profiel 1: Alle documentversies worden gesynchroniseerd.	M
F8	F4	Metadata van documenten wordt gesynchroniseerd.	M
F9	F5	Rechten op documenten worden niet gesynchroniseerd.	-
F10	F6	Gebruikers worden niet gesynchroniseerd.	-
F10.1	F6.1	Gebruikersnamen en email adressen worden meegezonden in document metadata.	M
F12		Het plaatsen van documenten in DS gebeurt in een op de SyncAgent ingestelde map.	M
F13		Synchronisatie van SP gebeurt met een op de SyncAgent ingestelde	M

		document-library.	
F14	F9	Er kunnen meer dan 2 replica's gesynchroniseerd worden.	S

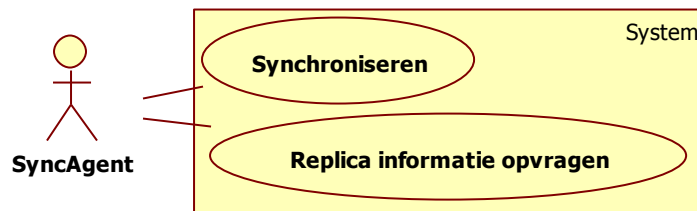
Code	Ref	Omschrijving	Prioriteit
N1	O1	Communicatie tussen SP en DS kan vanaf één locatie geïnitieerd worden om firewalls te omzeilen, dit kan zowel de SP- als de DS locatie zijn.	M
N2	O2	De synchronisatie is bestand tegen onderbrekingen.	M
N3	O3	Het is mogelijk alle communicatie beveiligd plaats te laten vinden.	M
N4	D2	Metadata van synchronisatie wordt opgeslagen in een database.	M

7.5 Ontwikkelen van de SyncServer

De SyncServer heeft een centrale rol in de geadviseerde architectuur. De SyncServer is verantwoordelijk voor de centrale opslag van documenten, het bijhouden van metadata over het synchronisatieproces en uiteraard de directie over het synchronisatieproces zelf.

7.5.1 Opstellen use-cases en use-case beschrijvingen

Voor het proof-of-concept heeft de SyncServer geen menselijke actoren, maar wordt geactiveerd door een SyncAgent. De SyncServer biedt de SyncAgent twee mogelijke acties: Synchroniseren en informatie ophalen over een andere SyncAgent die met de SyncServer heeft gesynchroniseerd. Op basis van dit gegeven heb ik use-case beschrijvingen en een use-case diagram opgesteld.



Figuur 12: Use-case diagram van de SyncServer

7.5.2 Opstellen klassediagram

De SyncServer omvat drie elementen uit de architectuur: de SyncService, de SyncProvider en natuurlijk de SyncServer zelf. De SyncServer communiceert met de SyncAgent door middel van een webservice, de SyncService. Ik heb gekozen om deze webservice te implementeren met behulp van WCF (Windows Communication Foundation). WCF is sinds versie 3.5 onderdeel van het .NET Framework en biedt een basis voor het ontwikkelen van webservices. Door de toepassing van WCF in de architectuur en het proof-of-concept kan aan toekomstige eisen omtrent schaalbaarheid en beveiliging worden voldaan door slechts de configuratie te wijzigen.

Ik heb de SyncServer verdeeld in drie componenten:

1. **Data:** Het component data bevat klassen die door de SyncService met de SyncAgent worden gecommuniceerd. Deze klassen definiëren de metadata van een documenten.
2. **DataStore:** Dit component bevat klassen voor het opslaan van objecten van de klassen uit het Data component.

3. **SyncServer**: Bevat de kern van het systeem waaronder de SyncProvider en SyncService.

Ik heb voor deze opdeling gekozen door functionaliteit te groeperen. Deze opzet maakt het mogelijk om het DataStore component te vervangen met een andere implementatie hiervan zonder dat hiervoor wijzigingen aan het SyncServer component plaats hoeven te vinden. Dit komt de onderhoudbaarheid ten goede (maintainability-changeability).

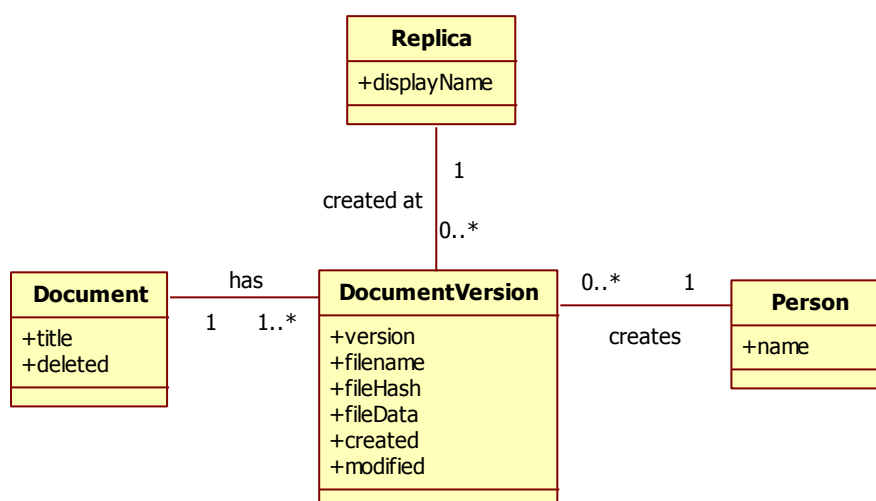
Het component Data wijkt hiervan af, dit component is apart omdat het ook door de SyncAgent gebruikt wordt.

Van deze componenten heb ik per stuk eerst een eenvoudig model gemaakt waarin slechts klassen met relaties en eigenschappen zijn gemodelleerd.

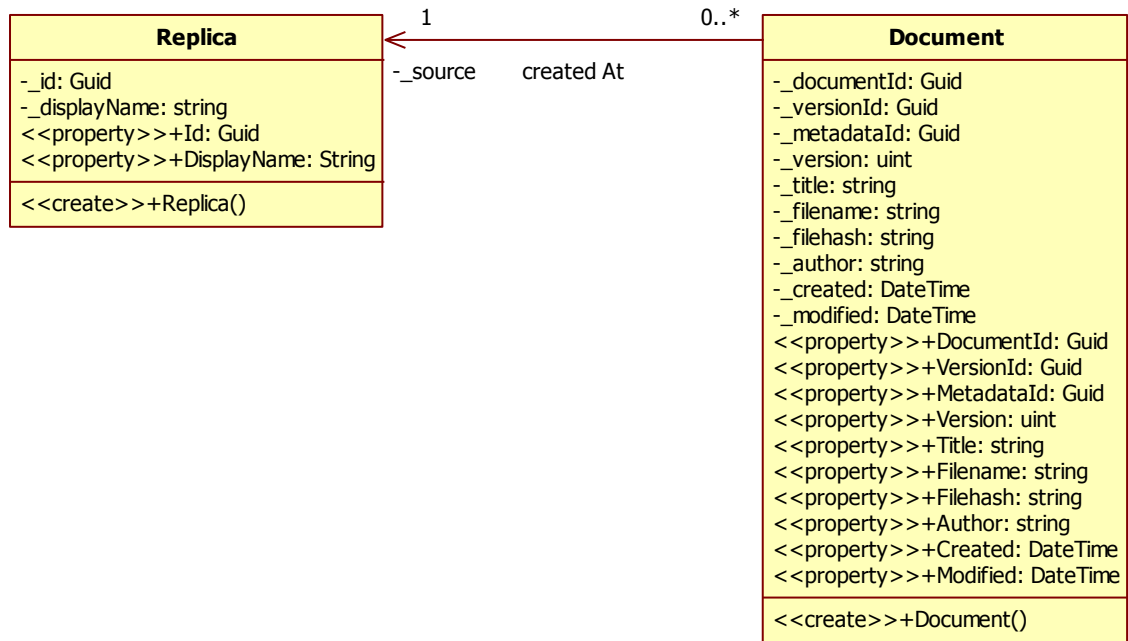
Vanuit dit model ben ik begonnen met implementeren. Het implementeren van deze modellen leidde tot nieuwe inzichten. Op basis van deze inzichten heb ik de modellen uitgebreid tot een implementatie model. Op deze manier zijn modellen en implementatie samen gegroeid. Omdat ik na nieuwe inzichten vanuit het model wijzigingen heb uitgevoerd heb ik grip kunnen houden op het ontwerp. Hierdoor is er geen wildgroei aan klassen, methoden en eigenschappen ontstaan.

Data

Om een WCF service te maken moet een ServiceContract worden opgesteld, dit ServiceContract beschrijft de functies die de service aanbied en de structuur van de gegevens die gecommuniceerd worden. Als de gegevensstructuur in een apart component wordt geïmplementeerd kan dit component door client-applicaties (in dit geval een SyncAgent) gebruikt worden en maken zowel client als server gebruik van dezelfde klassen. Het onderstaande figuur (Figuur 13) toont het domein model voor dit component. In het implementatiemodel heb ik de klassen Document en DocumentVersion samengevoegd omdat er voor de synchronisatie alleen documentversies worden gesynchroniseerd. De entiteit Person heb ik tevens in deze samengevoegde klasse als attribuut opgenomen. De uitwerking vind u in figuur 14 Implementatiemodel data



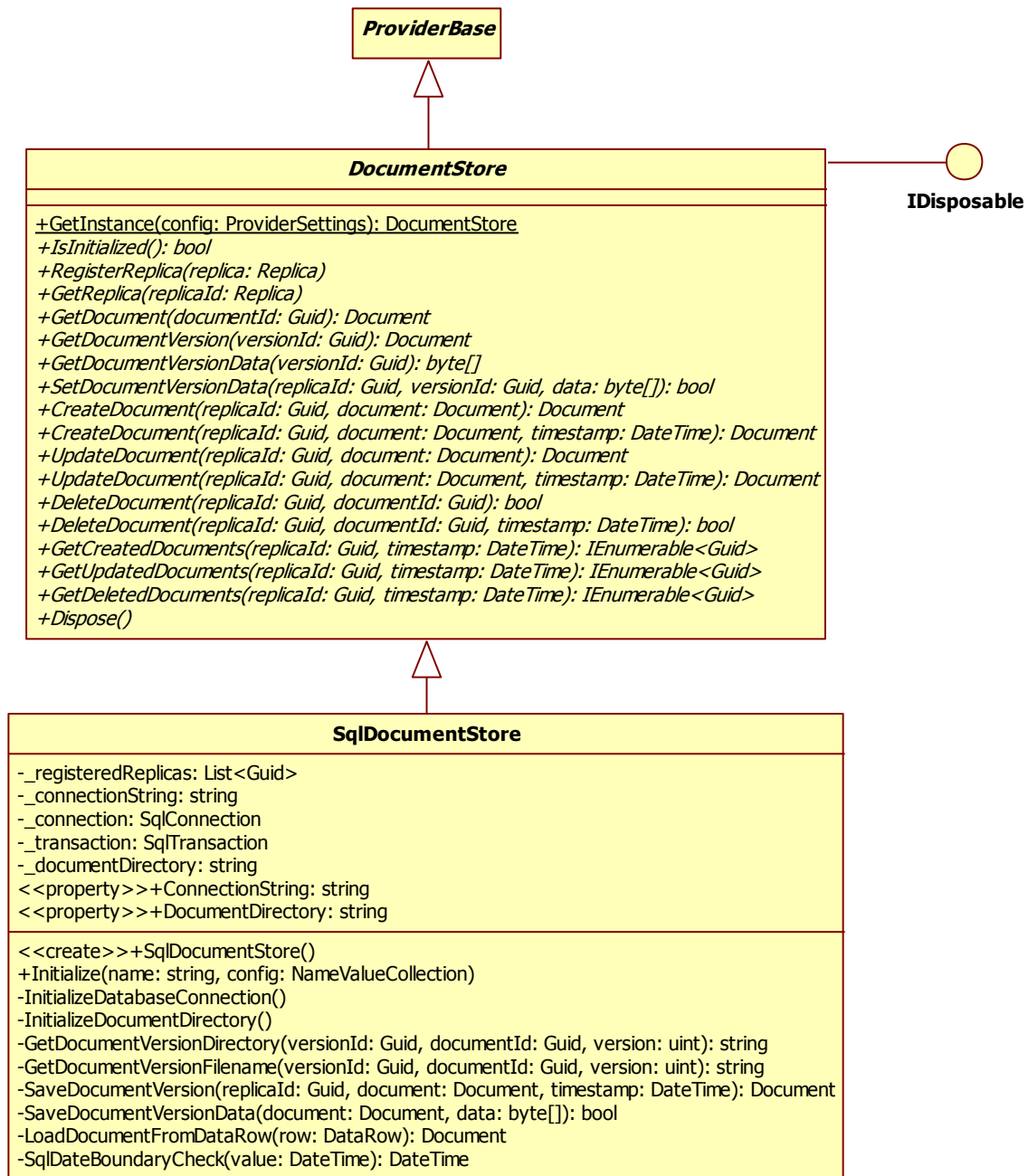
Figuur 13: Domeinmodel data.



Figuur 14: Implementatiemodel data.

DataStore

De data die door de SyncService van een SyncAgent wordt ontvangen moet worden opgeslagen. Voor de opslag hiervan heb ik een apart component ontwikkeld. Ik hiervoor gekozen omdat ik verwacht dat deze functionaliteit in toekomstige onderdelen toe te kunnen passen. Om de software onafhankelijk te maken van een databaseplatform heb ik gekozen om Microsoft's Provider design pattern toe te passen. De toepassing van dit design-pattern op de DataStore maakt het mogelijk om zonder wijzigingen aan bestaande code een andere implementatie van de DataStore te gebruiken. Bijvoorbeeld voor een ander database-platform. Ik heb voor het proof-of-concept een eenvoudige implementatie van de DocumentStore provider gemaakt voor SQL Server. In figuur 15 vind u het klassediagram van het DataStore component.



Figuur 15: Implementatiemodel DataStore

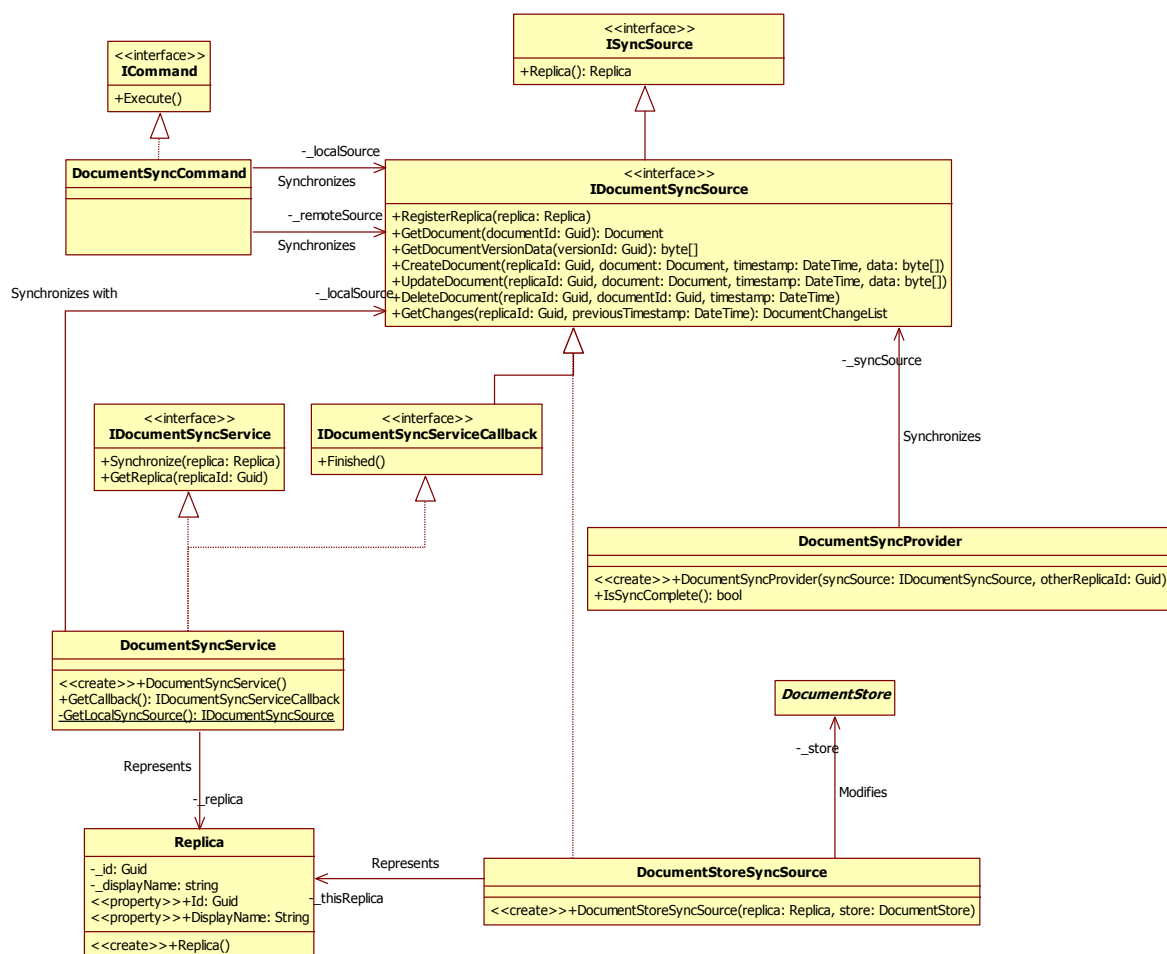
SyncServer

Om het synchronisatieproces te dirigeren maak de SyncServer gebruik van componenten van het Microsoft Sync Framework. De belangrijkste componenten van het Sync Framework die door de SyncServer gebruikt worden zijn de SyncOrchestrator en de KnowledgeSyncProvider. De SyncOrchestrator is een klasse van het Microsoft Sync Framework die het synchroniseren van twee SyncProviders dirigeert. De KnowledgeSyncProvider is het aanknopingspunt voor applicaties om te synchroniseren met het Microsoft Sync Framework.

In de SyncServer wordt de KnowledgeSyncProvider geïmplementeerd door de DocumentSyncProvider. Deze gebruikt een IDocumentSyncSource als databron. Hierdoor kunnen

met slechts één implementatie van de complexe logica van de DocumentSyncProvider meerdere databronnen worden gesynchroniseerd. Er zijn twee klassen die de IDocumentSyncSource interface implementeren, DocumentStoreSyncSource met een DocumentStore object als back-end en de DocumentSyncService de SyncAgent als back-end gebruikt door middel van het callback mechanisme van WCF.

De synchronisatie wordt gestart door een instantie van DocumentSyncCommand, objecten van dit type maken een SyncOrchestrator, koppelen een lokale en externe SyncProvider en starten het synchronisatieproces. Dit had ook direct in de SyncService uitgevoerd kunnen worden. Ik heb gekozen voor deze oplossing omdat deze opzet de mogelijkheid biedt een synchronisatieproces te starten vanuit een andere context. Een voorbeeld hiervan is een management interface van waaruit synchronisatie naar een backup-medium gestart kan worden.



Figuur 16: Klassediagram SyncServer

7.5.3 Opstellen relationeel representatie- en implementatiemodel

De SyncServer slaat gesynchroniseerde documenten op. Dit wordt gedaan door een DocumentStore implementatie. Voor het proof-of-concept heb ik een simpele implementatie hiervan gemaakt in Microsoft SQL Server. Omdat dit een relationeel database management systeem is heb ik hiervoor een relationeel representatie- en implementatiemodel gemaakt.

Replica(<u>replicaId</u> , displayName) replicaId is primaire sleutel Document(<u>documentId</u> , created, deleted) documentId is primaire sleutel DocumentVersion(<u>versionId</u> , <i>documentId</i> , version, title, filename, filehash, author, created, modified, <i>replicaId</i> , timestamp) versionId is primaire sleutel documentId is vreemde sleutel, verwijst naar Document (not null) replicaId is vreemde sleutel, verwijst naar Replica (not null) - documentId, timestamp is kandidaatsleutel
--

7.5.4 Opstellen sequencediagrammen

Op basis van de use-cases en inzichten die ik heb verworven tijdens het onderzoek heb ik voor complexe methoden in het systeem sequence diagrammen opgesteld. Ik heb hierin alleen berichten tussen objecten opgenomen die ik zelf heb ontwikkeld.

7.5.5 Testen

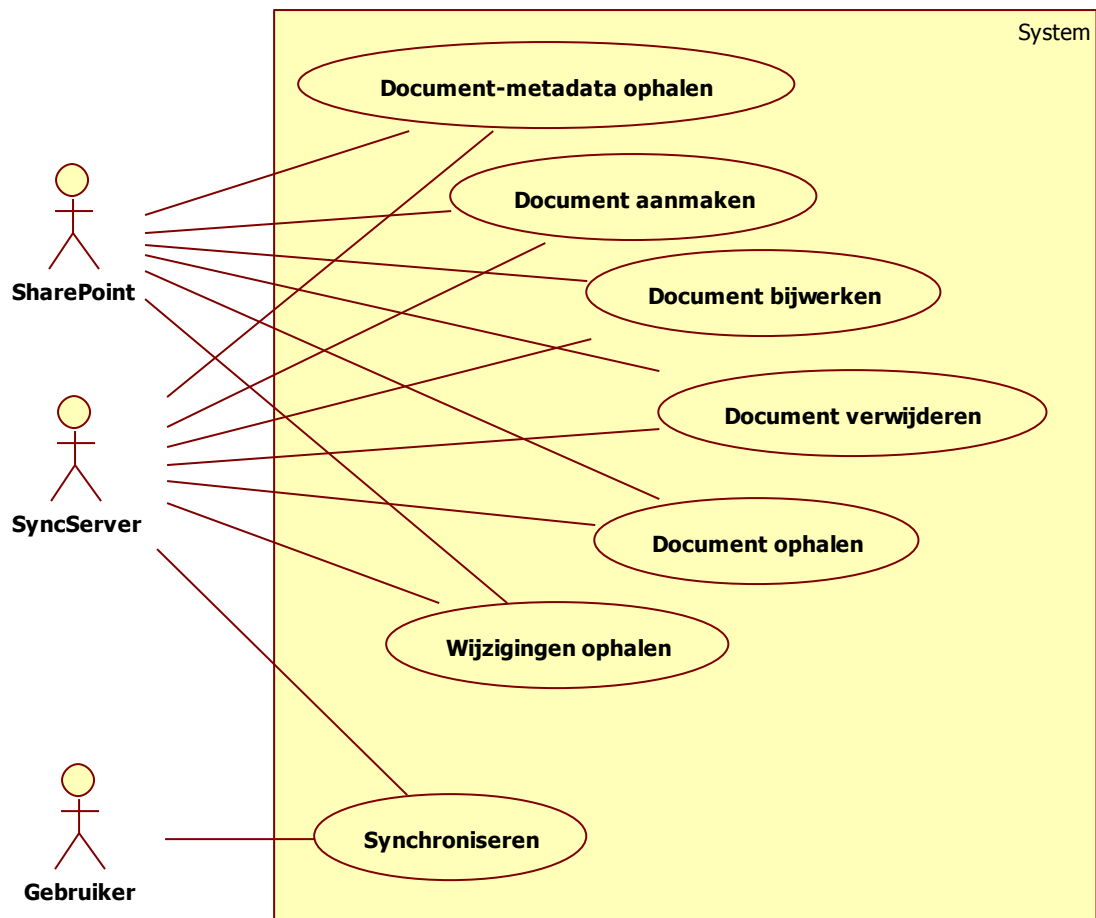
Om de SyncServer te kunnen testen heb ik een simpele SyncAgent implementatie gemaakt met een DataStore component als backend. Dit heeft het mogelijk gemaakt om de SyncServer onafhankelijk van de SharePoint en Docstream implementaties van de SyncAgents te testen.

7.6 Ontwikkelen van de SyncAgent implementaties

De SyncAgent implementaties van SharePoint zijn op veel vlakken gelijk, zo ook de ontwikkeling van deze applicaties. Ik heb er echter voor gekozen de gezamenlijke functionaliteit niet in één component te vatten. De reden hiervoor is dat in de implementatie van de SyncAgent voor SharePoint een klasse vervuld is met betrekking tot de samenhang van de methoden in die klasse. Het gaat hierbij om de klasse SPClient waarin zowel wordt gecommuniceert met SharePoint als met de SyncServer. In de Docstream implementatie van deze applicatie heb ik ervoor gewaakt dat deze situatie zich weer voor zou doen.

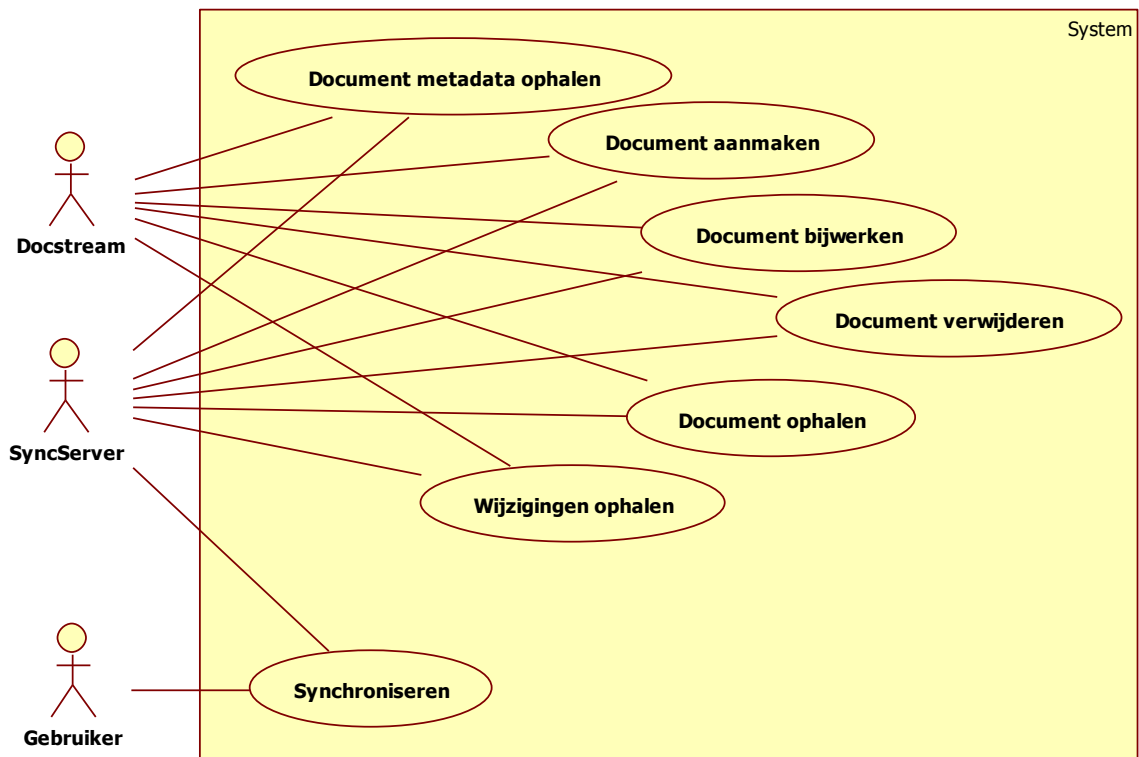
7.6.1 Opstellen van use-cases en use-case diagrammen

De SyncAgent heeft drie actoren, de gebruiker en de SyncServer en SharePoint. De gebruiker start het synchronisatieproces en de SyncServer.



Figuur 17: Use-case diagram SharePoint SyncAgent

De Docstream versie is vrijwel gelijk.



Figuur 18: Use-case diagram Docstream SyncAgent

7.6.2 Opstellen Requirements

Het opstellen van de requirements van de SharePoint SyncAgent heb ik gedaan aan de hand van de requirements aan het proof-of-concept en de verantwoordelijkheden van een SyncAgent binnen de architectuur.

Code	Omschrijving	Prioriteit
SP1	Het systeem kan een document afkomstig van de SyncServer aan een SharePoint document library toevoegen	M
SP2	Het systeem kan een nieuwe versie van een document afkomstig van de SyncServer aan een SharePoint document library toevoegen.	M
SP3	Het systeem kan op basis van gegevens afkomstig van de SyncServer een document uit een SharePoint document library verwijderen.	M
SP4	Het systeem kan een document uit een SharePoint document library ophalen en naar de SyncServer zenden in het gegevensformaat SyncServer.	M
SP5	Het systeem kan metadata over een document in een SharePoint document library ophalen in het gegevensformaat van de SyncServer.	M
SP6	Het systeem kan een lijst met gewijzigde documenten in een SharePoint document library ophalen en naar de SyncServer sturen.	M
SP7	Een gebruiker kan het synchronisatieproces met de de SyncServer starten.	M

De requirements van de Docstream SyncAgent zijn wederop vrijwel identiek.

Code	Omschrijving	Prioriteit
DS1	Het systeem kan een document afkomstig van de SyncServer aan een Docstream document library toevoegen	M
DS2	Het systeem kan een nieuwe versie van een document afkomstig van de SyncServer aan een Docstream document library toevoegen.	M
DS3	Het systeem kan op basis van gegevens afkomstig van de SyncServer een document uit een Docstream document library verwijderen.	M
DS4	Het systeem kan een document uit een Docstream document library ophalen en naar de SyncServer zenden in het gegevensformaat SyncServer.	M
DS5	Het systeem kan metadata over een document in een Docstream document library ophalen in het gegevensformaat van de SyncServer.	M
DS6	Het systeem kan een lijst met gewijzigde documenten in een Docstream document library ophalen en naar de SyncServer sturen.	M
DS7	Een gebruiker kan het synchronisatieproces met de de SyncServer starten.	M

7.6.3 Opstellen klassediagram relationeel

Net als bij het ontwikkelen van de SyncServer ben ik begonnen met met een eenvoudig klassediagram om dit vervolgens uit te breiden op basis van inzichten verkregen door het implementeren hiervan.

7.6.4 Opstellen sequence diagrammen

Ik heb voor het opstellen van de sequence diagrammen dezelfde werkwijze gehanteert als voor de SyncServer.

7.6.5 Testen

Ik heb beide SyncAgent implementaties getest volgens de systeem- en acceptatietestplannen.

7.7 Ontwikkelen van een eigen implementatie van de MetadataStore

Voor het opslaan van synchronisatie metadata biedt het Microsoft Sync Framework een standaard component. Dit component is de SqlMetadataStore. Dit component slaat gegevens op in een bestand op het bestandssysteem. Het is echter niet mogelijk om de gegevens in dit bestand te benaderen zonder gebruik van dit component.

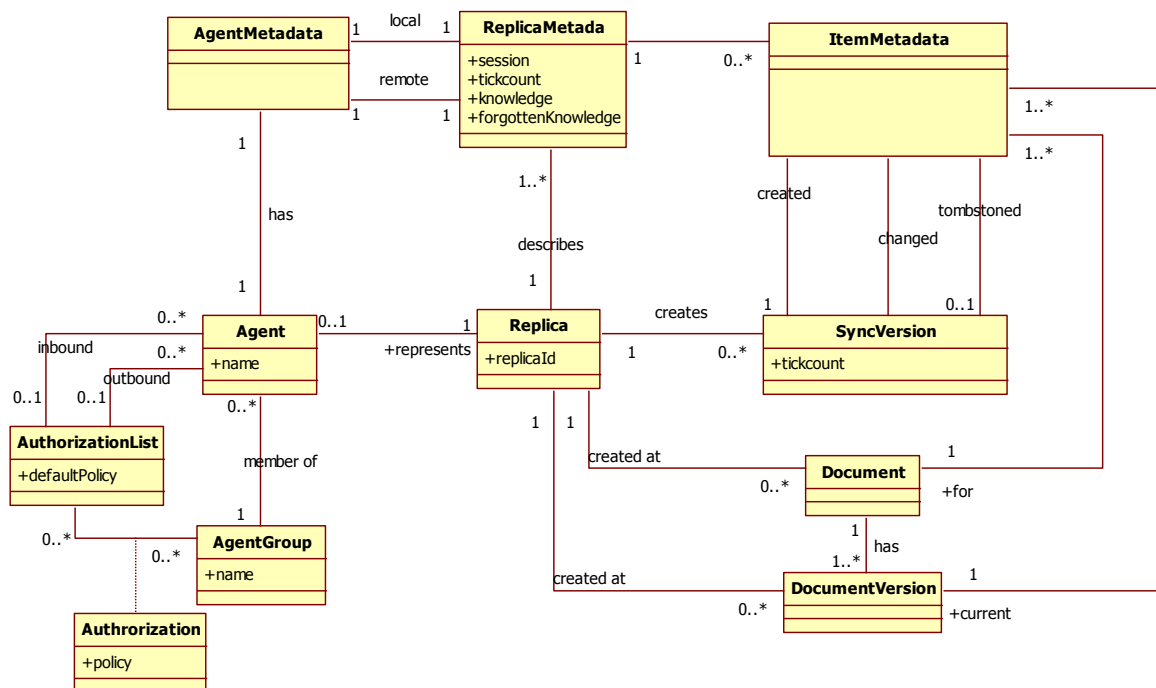
Een van de requirements aan de architectuur is dat de synchronisatie metadata door middel van een ADO.NET koppeling aangesproken kan worden. Om deze reden heb ik geprobeerd om een eigen implementatie van deze functionaliteit in SQL server te implementeren. Ik heb hiervoor wederom gekozen om het Provider design pattern toe te passen en een implementatie hiervan voor SQL server te maken. Helaas is de tijd te kort gebleken om van dit component een werkende implementatie te maken. Het probleem vindt zijn oorzaak in het berekenen van wijzigingen. Deze berekening wordt voor een deel uitgevoerd door componenten van het Microsoft Sync Framework, dit maakt het erg lastig om de specifieke oorzaak te achterhalen.

De database die ik heb ontwikkeld voor dit doel is echter wel functioneel.

7.7.1 Opstellen klassediagram

Ik heb voor de MetadataStore een domein model opgesteld op basis van inzichten die ik heb verkregen tijdens het ontwikkelen van de SyncServer. Om te bepalen welke gegevens opgeslagen moeten worden heb ik de SqlMetadataStore, ReplicaMetadata en ItemMetadata klassen van het SyncFramework geanalyseerd. De bevindingen van deze analyse heb ik verwerkt in het domein klassediagram van een aangepaste MetadataStore.

Voor het opstellen van het klassediagram heb ik de SqlMetadatstore van het Microsoft Sync Framework geanalyseerd. De kennis die ik uit deze analyse heb opgedaan heb ik gebruikt om het onderstaande model te ontwikkelen. Ik heb in het model bovendien klassen toegevoegd die ht in de toekomst mogelijk maken om gegevens met betrekking tot de beveiliging van gesynchroniseerde data op te slaan.



Figuur 19:Klassediagram van MetadataStore zonder methoden en implementatie-specifieke gegevens.

Ik heb op basis van dit model een relationeel representatiemodel opgesteld:

AgentGroup(agentGroupId, name)

agentGroupId is PK

AuthorizationList(authorizationListId, name, defaultPolicy)

authorizationListId is PK

AuthorizationListItem(authorizationListId, agentGroupId, policy)

authorizationListId, agentGroupId is PK

authorizationListId is FK verwijst naar AuthorizationList NOT NULL

agentGroupId is FK verwijst naar AgentGroup NOT NULL

Agent(agentId, name, *agentGroupId*, *inbound*, *outbound*)

agentId is PK

agentGroupId is FK verwijst naar AgentGroup NULL

inbound is FK verwijst naar AuthorizationList NULL

outbound is FK verwijst naar AuthorizationList NULL

Replica(replicaId, *agentId*)

replicaId is PK

agentId is FK verwijst naar Agent NOT NULL

Document(documentId, replicaId)

documentId is PK

replicaId is FK verwijst naar Replica NOT NULL

DocumentVersion(documentVersionId, *documentId*, *replicaId*)

documentVersionId is PK

documentId is FK verwijst naar Document NOT NULL

replicaId is FK verwijst naar Replica NOT NULL

ReplicaMetadata(*agentId*, *replicaId*, local, session, tickcount, knowledge, forgottenKnowledge)

agentId, replicaId is PK

agentId, local is kandidaatsleutel

agentId is FK verwijst naar Agent NOT NULL

replicaId is FK verwijst naar Replica NOT NULL

ItemMetadata(*agentId*, *replicaId*, *documentId*, *documentVersionId*, *createdReplica*, createdTickcount, *changedReplica*, changedTickcount, *tombstoneReplica*, tombstoneTickcount)

agentId, replicaId, documentId is PK

agentId is FK verwijst naar Agent

replicaId is FK verwijst naar Replica

documentId is FK verwijst naar Document

documentVersionId is FK verwijst naar DocumentVersion

createdReplica is FK verwijst naar Replica

changedReplica is FK verwijst naar Replica

tombstoneReplica is FK verwijst naar Replica

8 Uitvoeren analyse impact op Docstream

Om een inschatting te maken van de impact die de implementatie van de architectuur heeft op Docstream heb ik een impact analyse gemaakt. Ik heb dit gedaan door de Lead Developer van Docstream te interviewen. In dit interview heb ik hem de requirements van de architectuur voorgelegd en met hem de consequenties hiervan voor Docstream besproken.

Door de aard van de architectuur heeft deze in principe geen impact op Docstream. Om Docstream te integreren in de architectuur moeten echter een aantal zaken aangepakt worden.

Er moet een uitbreiding komen op de GUI van de Docstream voor:

- Het visualiseren van deelname aan synchronisatie van een map. (56 uur)
- Configuratie van map-deelname aan synchronisatie. (48)
- Het tonen van de synchronisatiestatus van documentversies binnen mappen die deelnemen aan synchronisatie. (24)
- Het tonen van de status van een synchronisatie. (40)

Hiervoor moeten in de database van Docstream ondersteunende tabellen en store-procedures toegevoegd worden. De uren hiervoor zijn reeds opgenomen in de bovenstaande schatting.

Er moet een servicelaag voor Docstream ontwikkeld worden. De ontwikkeling hiervan gaat ongeveer 120 uur kosten.

In totaal zal het 288 uur kosten om te implementeren.

9 Evaluatie

9.1 Procesgang

Voortschrijdend inzicht heeft een grote rol gespeeld in het gedurende de uitvoer van het project. Dit heeft geleid tot resultaten die noch ik noch de opdrachtgever van tevoren in heb kunnen inschatten. Desondanks ben ik steeds planmatig tewerk gegaan en heb ik naar tevredenheid van de opdrachtgever gehandeld. Ik ben dan ook van mening dat de procesgang over het geheel goed is verlopen.

Verbeterpunten ten aanzien van het proces zie ik hoofdzakelijk in de toepassing van tools en andere hulpmiddelen ter ondersteuning hiervan. Bijvoorbeeld door gebruik te maken van SharePoint. Hoewel ik vroeg tijdens het project met SharePoint in aanraking ben gekomen tijdens mijn oriëntatie op dit product heb ik op dat moment niet de mogelijkheden herkent.

9.2 Producten

Het plan van aanpak is van redelijke kwaliteit. Het onderdeel planning had achteraf strakker gekunt. Ik had hierbij meer tijd in moeten plannen voor verslaglegging.

De oriëntatie op SharePoint was in de originele opdracht belangrijker dan in de opdracht die uiteindelijk is uitgevoerd. Ik heb hiermee echter wel voldoende kennis opgedaan om de opdracht uit voeren.

Het onderzoeksplan sloot goed aan bij de uitvoering van het onderzoek, het onderzoek is binnen de gestelde planning uitgevoerd. Het onderzoeksrapport is door de stakeholders geaccepteerd en de aanbevelingen in het rapport zijn overgenomen.

Het proof-of-concept heeft zijn taak volbracht, namelijk het aantonen van de werking van de geadviseerde architectuur. De documentatie en implementatie zijn beiden van voldoende kwaliteit.

De impact analyse heb ik samen met de lead developer van Docstream uitgevoerd, de kwaliteit hiervan staat niet ter discussie.

9.3 Beroepstaken

9.3.1 Uitvoeren analyse door definitie van requirements

Het achterhalen van de requirements was een lastige klus. Vooral omdat de stakeholders zelf ook niet wisten waar ze heen wilden. De requirements aan de architectuur zijn gedurende het project blijven groeien. In de bijlagen vindt u naast de requirements aan de architectuur naar aanleiding van het onderzoek ook een document met requirements die ik heb opgesteld op basis van de laatste stand van zaken.

9.3.2 Opstellen gegevensmodel voor een database

De gegevens die worden opgeslagen worden door de SyncServer en SyncAgents zijn niet erg complex van aard. De gegevens die opgeslagen worden in de MetadataStore voldoen echter wel aan de complexiteitscriteria, ik heb deze activiteit ongeleid en daarmee op niveau vier uitgevoerd.

9.3.3 Ontwerpen systeemdeel

De architectuur van het proof-of-concept heb ik beschreven met visuele hulpmiddelen. De applicaties en elemente in de architectuur heb ik zodanig in detail gemodelleerd dat het implementeren hiervan mogelijk is. Het betreft applicaties in een gedistribueerde architectuur.

9.3.4 Bouwen applicatie

Door de SyncServer in verschillende componenten te verdelen en waar mogelijk gebruik te maken van het Provider design pattern is deze applicatie gemakkelijk te wijzigen. Componenten als de DataStore zijn opnieuw te gebruiken in andere systemen. Waar toepasbaar heb ik geavanceerde eigenschappen van C# zoals LinQ en generics gebruikt.

9.3.5 Initiëren en plannen van het testproces

Doordat de focus van de opdracht is veranderd is deze beroepstaak minder relevant geworden. Het product is voor de opdrachtgever een proof-of-concept. Hierdoor wordt er minder belang gehecht aan de kwaliteit van de software er wordt puur functionaliteit gevraagd.

Referenties

Literatuur

- Nel Verhoeven, Wat is onderzoek?
2^e herziene druk 2007 (1^e druk 2004), Boom Onderzijs
- Jos Warmer & Anneke Kleppe, Praktisch UML 3de editie
3^e druk 2006 (1e druk 2004), Pearson Addison Wesley
- Martin Fowler, Patterns of Enterprise Application Architecture
10e druk 2006, Pearson Education, Inc.
- Martin Pol, Ruud Teunissen en Erik van Veenendaal, Testen volgens TMap
2^e druk

Websites

- <http://sharepoint.microsoft.com>
- <http://msdn.microsoft.com/en-us/sharepoint/default.aspx>
- <http://technet.microsoft.com/en-us/sharepoint/default.aspx>

Begrippenlijst

Begrip	Omschrijving
ASP.NET	Een technologie van Microsoft om webpagina's te genereren.
Dashboard	Een pagina waarop meerdere KPI's worden weergegeven.
DMS	Document Management Systeem, een systeem voor het beheren van digitale documenten.
Docstream	Een gebruiksvriendelijke applicatie voor documentbeheer ontwikkeld door Pro4all B.V
Firewall	Apparaat voor het beveiligen van netwerkverkeer.
FTE	Full Time Employee, een maat voor bedrijfsomvang.
GUI	Graphical User Interface, grafische gebruikers interface
KPI	Key Performance Indicator, een variabele om prestaties van ondernemingen te analyseren.
Master page	Een pagina die als template dient voor het weergeven van andere pagina's.
SharePoint	Een product van Microsoft, biedt een platform voor samenwerking. Een onderdeel hiervan is documentbeheer.
SyncAgent	Element in de ontwikkelde architectuur.
Synchronisatie	Het gelijk maken van twee gegevensbronnen.
SyncServer	Element in de ontwikkelde architectuur.
Trigger	Een mechanisme dat een proces in gang zet.
Visual Studio	Een interactieve ontwikkelomgeving ontwikkeld door Microsoft.
Webpart	Een onderdeel van een SharePoint webpagina
Workflow	Werkstroom

Bijlagen

A Scorekaart functionele scenario's

Scenario's:

Scenario	Beschrijving
Het eenmalig importeren van documenten uit Docstream naar SharePoint:	Door een eenmalige handmatige actie documenten uit Docstream in een SharePoint omgeving importeren.
Het geleidelijk importeren van documenten uit Docstream naar SharePoint	Documenten worden geïmporteert in een SharePoint omgeving wanneer deze opgevraagd worden.
Het synchroniseren van documenten tussen Docstream en SharePoint	Documenten worden tussen Docstream en een SharePoint omgeving consistent gehouden. Dit betekent dat nieuwe documenten die worden toegevoegd, gewijzigd of verwijderd in Docstream ook worden toegevoegd, gewijzigd en verwijderde in een SharePoint omgeving en vice versa.

Scores per stakeholder:

Stakeholder	Eenmalig importeren	Importeren op aanvraag	Synchroniseren
Business Development	+	++	+++
Consultants	+	++	+++
Development	+++	++	+
Totaal	+++++	++++++	+++++++

B Scorekaart synchronisatiemogelijkheden SharePoint en Docstream

SharePoint:

	Objectmodel	Client objectmodel	SyncFramework
Voordelen	Direct reageren op wijzigingen mogelijk (+)	Weinig invloed op SharePoint performance. (++)	Weinig invloed op SharePoint performance (+++) Mogelijkheid om wijzigingen te detecteren. (+)
Nadelen	Beïnvloed SharePoint performance (---)	Geen mogelijkheid om wijzigingen te detecteren. (-)	
Score	--	+	++++

Docstream:

	Direct op database	Met een webservice
Voordelen	Direct reageren op wijzigingen d.m.v. triggers. (+) Snel te implementeren. (+)	Schaalbaar.(+++) Business logica blijft intact.(++)
Nadelen	Beveiligingsrisico. (-) Omzeilen business logica.(--)	Niet direct reageren op wijzigingen. (-)
Score	0	++++

C Plan van aanpak

Pro4all

Plan van aanpak

Het synchroniseren van documenten en metadata tussen Docstream en SharePoint

Inhoudsopgave

Opdrachtomschrijving	2
Opdrachtgever.....	2
Uitvoerenden.....	2
Kader	2
Probleemstelling.....	2
Doelstelling.....	2
Uitgangssituatie.....	2
Werkzaamheden	2
Methoden en technieken	3
Producten	3
Afbakening.....	4
Randvoorwaarden	4
Risicofactoren.....	4
Wijze van rapporteren.....	4
Benodigde middelen en mensen.....	4
Globale planning.....	6
Detail planning	7
Mijlpaalproducten	8

Opdrachtomschrijving

Opdrachtgever

Pro4all is een commercieel bedrijf dat software ontwikkelt. Pro4all is opgericht in 2001 en gespecialiseerd in projectsoftware en informatiemanagement voor de bouwsector. Het bedrijf bestaat uit 16 medewerkers en is gevestigd in Woerden.

Uitvoerenden

De opdracht wordt uitgevoerd door Dh. K. van Hagen, student aan de Haagse Hogeschool en stagiair bij Pro4all.

Kader

De opdracht wordt uitgevoerd in het kader van technologische vernieuwing: “met de tijd meegaan”.

Probleemstelling

Docstream is een applicatie die bedrijven in de bouwsector ondersteunt bij bouwprojecten. Het biedt uitgebreide functies voor het beheer van documenten, e-mail verkeer en procedures.

Docstream is vanuit de praktijk gegroeid. De oprichters van Pro4all hebben op basis van eigen ervaring in de bouwsector en een uitgebreid onderzoek de eerste versie van Docstream ontwikkeld. Deze eerste versie, een Windows applicatie, is op basis van feedback van klanten uitgegroeid tot een zeer gecompliceerde applicatie. Later is van Docstream een ASP.NET versie ontwikkeld. Bij deze versie lag de nadruk op de GUI van de applicatie. De ASP.NET versie heeft rijke GUI die het doet voorkomen alsof men met een Windows applicatie werkt in plaats van een webbrowser. Deze versie is in eigen beheer ontwikkeld en wordt met uitzondering van een paar klanten op ASP- basis (Application Service Provider) aangeboden aan klanten.

Naast Docstream levert Pro4all ook consultancy op Microsoft SharePoint. SharePoint is een platform voor documentbeheer en samenwerking voor bedrijven. SharePoint staat in de belangstelling en veel klanten van Pro4all hebben een SharePoint omgeving. In tegenstelling tot Docstream bevindt SharePoint zich in de kantooromgeving van de klant. Momenteel bestaat er geen koppeling tussen SharePoint en Docstream. Het is dus niet mogelijk voor klanten om beide pakketten naast elkaar te gebruiken met dezelfde documenten.

Doelstelling

Pro4all wil in 2010 synchronisatie van documenten tussen Docstream en Microsoft SharePoint realiseren.

Uitgangssituatie

Er bestaan binnen Pro4all nog geen documenten die betrekking hebben op dit onderwerp.

Werkzaamheden

- Opstellen plan van aanpak
- Oriënteren op SharePoint ontwikkeling.
- Technische oplossingen synchronisatie onderzoeken:
 - Opstellen onderzoeksplan.
 - Uitvoeren onderzoek.
 - Opstellen onderzoeksrapport.

- Ontwikkelen Proof-Of-Concept
 - Opstellen requirements
 - Opstellen klassediagram
 - Opstellen sequencediagram(men)
 - Programmeren
 - Testen
- Uitvoeren van een Impact Analyse voor de voorstelsoplossing.

Methoden en technieken

Onderzoek

Voor het uitvoeren van het onderzoek maak ik gebruik van een methode die gebaseerd is op de methode die Nel Verhoeven aandraagt in haar boek “Wat is onderzoek?”. Technieken die toegepast worden tijdens het uitvoeren van het onderzoek zijn:

- Interview
- Literatuuronderzoek
- Experiment

Proof-of-Concept

Voor de ontwikkeling van het Proof-of-Concept wordt geen specifieke ontwikkelmethodiek gebruikt. Op basis van de resultaten van het onderzoek worden in overleg met de opdrachtgever requirements en een functioneel ontwerp opgesteld. Dit functioneel ontwerp wordt uitgewerkt tot een technisch ontwerp. Op basis van dit ontwerp wordt de sourcecode geschreven. De geproduceerde applicatie wordt getest:

- Requirements worden een prioriteit toegewezen op basis van de MoSCoW-methode.
- Het functioneel ontwerp bestaat uit: een use-case diagram (UML) en een beschrijving van de hierin gemodelleerde use-cases.
- Het technisch ontwerp bestaat uit: een klassediagram, sequentiediagrammen, relationeel representatiemodel.
- Het testen van het Proof-of-Concept wordt beschreven in een testplan. De resultaten van de test(s) worden vastgelegd in een testrapport.

Technieken

- Microsoft SharePoint 2010
- Microsoft SQL Server
- Microsoft .NET Framework
- ASP.NET
- C#

Producten

- Plan van aanpak
- Onderzoeksplan
- Onderzoeksrapport
- Proof-of-Concept:
 - Requirements

- Klassediagram
- Sequencediagram
- Sourcecode
- Testplan
- Testrapport
- Impactanalyse

Afbakening

Synchronisatie tussen Docstream en Microsoft SharePoint heeft in dit project betrekking op documenten en metadata hierover. Toewijzing van gebruikersrechten op gesynchroniseerde documenten aan zijde van SharePoint valt buiten de scope van dit project. Wel wordt rekening gehouden met rechten aan zijde van Docstream, er worden alleen documenten gesynchroniseerd die de gebruiker of organisatie aan SharePoint zijde mag bekijken/bewerken.

Randvoorwaarden

Het is noodzakelijk dat gegevens en documenten uit Docstream via een externe interface beschikbaar worden gesteld. Achter deze interface kan een test- of dummyomgeving gekoppeld zijn die gebruikt kan worden om externe toegang te testen. Deze interface moet gelijk zijn aan de interface die in productie gebruikt wordt/gaat worden.

Risicofactoren

- **Kennisgebrek:** Ik heb nog niet eerder met SharePoint gewerkt, dit houdt in dat het oplossen van eventuele problemen met SharePoint mij meer tijd kost dan gepland. Om te voorkomen dat mijn gebrek aan kennis voor problemen zorgt ben ik de opdracht begonnen met een oriëntatie op SharePoint. Daarnaast kan ik indien nodig een collega raadplegen.
- **Kinderziektes:** Ik heb ontwikkeld op de nieuwste versie van SharePoint, SharePoint 2010. Deze versie was een groot deel van de opdracht nog in beta-versie. Hierdoor bestaat het risico op kinderziektes en ongedocumenteerd gedrag.
- **Onvolledige documentatie:** In het verlengde van de kinderziektes is de documentatie van de beta-versie van SharePoint nog niet volledig en aan wijzigingen onderhevig. Documentatie van voorgaande versies kunnen uitkomst bieden alleen dient correcte werking hiervan wel geverifiëerd te worden middels tests.

Wijze van rapporteren

Rapportage over de voortgang geschied op basis van een voortgangsrapport dat tijdens het wekelijkse SharePoint overleg mondeling wordt toegelicht.

Benodigde middelen en mensen

Voor de uitvoering van dit project zijn de volgende middelen nodig:

- PC met:
 - Windows 7

- SharePoint 2010
 - Internettoegang
- Interface voor extern benaderen van documenten en metadata over documenten.

En de volgende mensen:

- Dhr. K. van Hagen (SharePoint developer)
- Dhr. A.W.A. van Deenen (teamleider ontwikkeling, stakeholder Docstream development)
- Dhr. A.C. Bleij (SharePoint expert)
- Dhr. P. Beuker (Stakeholder sales)
- Dhr. R. Pijpers (Stakeholder directie)

Globale planning

[illegible]

Detail planning

		Projectweek:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Fase	Taak	Kalenderweek:	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
F-1	Oriënteren SharePoint 2010																		
	Opstellen Plan van Aanpak																		
F-2	Opstellen Onderzoeksplan																		
	Interviewen																		
	Opstellen requirements																		
	Verzamelen literatuur																		
	Analyseren literatuur																		
	Schrijven onderzoeksrapport																		
F-3	Opstellen requirements (POC)																		
	Opstellen testplan																		
	Opstellen klassediagram (POC)																		
	Opstellen sequentiediagram(men) (POC)																		
	Programmeren en testen (POC)																		
	Testen & opstellen Testrapport (POC)																		
F-4	Uitvoeren impact analyse																		
F-5	Schrijven procesverslag																		

Mijlpaalproducten

- Plan van aanpak
- Onderzoeksplan
- Requirements
- Onderzoeksrapport
- Proof of Concept
- Testrapport Proof of Concept
- Impact Analyse

D Onderzoeksplan

Pro4all B.V.

Onderzoekplan synchronisatie

Beschrijving van de onderzoeksmethode voor het onderzoek naar
synchronisatie van documenten tussen SharePoint en Docstream

Auteur:	Klaas van Hagen
Datum:	5 maart 2010
Bedrijfsmentor:	Dhr. A.W.A. van Deenen
Begeleider/examinator:	Dhr. E.M. van Doorn
Expert/examinator:	Mevr. H.G.J. Bechet-Tjoonk

Inhoudsopgave

Inleiding	2
Doelstelling	2
Onderzoeksvraag	2
Producten	2
Afbakening	2
Aanpak	2
Methoden en technieken	3
Werkzaamheden	3
Randvoorwaarden	3
Aannames	3
Risico's	3
Projectorganisatie	3
Planning	4

Inleiding

Pro4all heeft Docstream ontwikkeld. Docstream is een applicatie voor documentbeheer. Docstream onderscheidt zich van andere pakketten voor documentbeheer door haar gebruikersvriendelijkheid. Naast Docstream ontwikkelt Pro4all ook toepassingen op basis van Microsoft SharePoint. Microsoft SharePoint bevat net als Docstream functionaliteiten voor documentbeheer. Pro4all wil beide producten aan haar klanten kunnen leveren zonder dat deze producten onderling concurreren. Om dit te realiseren wil Pro4all de mogelijkheid om documenten en gegevens hierover tussen beide producten te synchroniseren.

Doelstelling

Pro4all wil in 2010 synchronisatie van documenten tussen Docstream en Microsoft SharePoint realiseren. Het doel van dit onderzoek in dit geheel is adviseren over de methode van synchroniseren.

Onderzoeksvraag

De hoofdvraag van het onderzoek is:

Hoe kan synchronisatie van documenten en metadata tussen Microsoft SharePoint en Docstream gerealiseerd worden?

Om deze vraag te kunnen beantwoorden moet antwoord worden gegeven op de volgende deelvragen:

1. Welke eisen worden er vanuit Pro4all aan de synchronisatie gesteld?
2. Welke best-practices zijn er m.b.t. data synchronisatie?
3. Welke mogelijkheden biedt SharePoint 2010 om met andere pakketten te synchroniseren?
4. Welke mogelijkheden biedt Docstream om met andere pakketten te synchroniseren?

Producten

- Onderzoeksplan
- Onderzoeksrapport

Afbakening

Dit onderzoek heeft betrekking op de synchronisatie van documenten en metadata hiervan. Er wordt niet gekeken naar andere gegevens in Docstream of SharePoint. De SharePoint versie die wordt onderzocht is Microsoft SharePoint 2010. De versie van Docstream die wordt onderzocht is Docstream 3. Oudere of nieuwere versies van beide pakketten vallen buiten het domein van dit onderzoek.

Aanpak

Een beschrijving van de aanpak die gebruikt zal worden om het onderzoek uit te voeren. Op basis van:

Methoden en technieken

Voor uitvoeren van het onderzoek maak ik gebruik van een methode die gebaseerd is op de methode die Nel Verhoeven aandraagt in haar boek “Wat is onderzoek?”. Technieken die toegepast worden tijdens het uitvoeren van het onderzoek zijn:

- Interview
- Literatuuronderzoek

Werkzaamheden

De werkzaamheden per deelvraag:

1. Welke eisen worden er vanuit Pro4all aan de synchronisatie gesteld?
 - Opstellen use-cases voor de synchronisatie tussen DS en SP.
 - Interview met interne stakeholders.
 - Requirements opstellen op basis van de use-cases en interview resultaten.
2. Welke best-practices voor architectuur zijn er m.b.t. data synchronisatie?
 - Verzamelen van literatuur.
 - Analyseren van literatuur.
3. Welke mogelijkheden biedt SharePoint 2010 om met andere pakketten te synchroniseren?
 - Verzamelen literatuur.
 - Analyseren literatuur.
4. Welke mogelijkheden biedt Docstream om met andere pakketten te synchroniseren?
 - Interview met dhr. Van Deenen.

Randvoorwaarden

Het is noodzakelijk dat gegevens en documenten uit Docstream via een externe interface beschikbaar worden gesteld. Achter deze interface kan een test- of dummyomgeving gekoppeld zijn die gebruikt kan worden om externe toegang te testen. Deze interface moet gelijk zijn aan de interface die in productie gebruikt wordt/gaat worden.

Aannames

Bij aanvang van het onderzoek doe ik de aanname dat het mogelijk is om synchronisatie tussen SharePoint en Docstream te verwezelijken.

Risico's

Een risico voor dit onderzoek is de “nieuwheid” van SharePoint technologie, hierdoor is het mogelijk dat er geen of weinig documentatie te vinden is. Of dat de documentatie er wel is maar niet volledig blijkt te zijn.

Projectorganisatie

Alle activiteiten worden uitgevoerd door K. van Hagen. Rapportage over de voortgang geschied mondeling tijdens het wekelijkse SharePoint overleg.

Planning

Taak	10	11	12	13	14	15
Opstellen Onderzoeksplan						
Interviewen						
Opstellen requirements						
Verzamelen literatuur						
Analyseren literatuur						
Schrijven onderzoeksrapport						

E Onderzoeksrapport

Pro4all B.V.

Synchronisatie tussen SharePoint en Docstream

Een onderzoek naar de mogelijkheden voor synchronisatie van documenten en metadata tussen SharePoint 2010 en Docstream.

Auteur:	Klaas van Hagen
Datum:	15 april 2010
Bedrijfsmentor:	Dhr. A.W.A. van Deenen
Begeleider/examinator:	Dhr. E.M. van Doorn
Expert/examinator:	Mevr. H.G.J. Bechet-Tjoonk

Inhoudsopgave

1	Samenvatting.....	3
2	Inleiding	3
3	Methode.....	3
3.1	Onderzoeksgroep	3
3.2	Onderzoeksonderwerp.....	3
4	Resultaten.....	4
4.1	Eisen van Pro4all aan synchronisatie	4
4.2	Best-practices voor datasynchronisatie	4
4.3	Mogelijkheden van pakketten tot synchronisatie.....	5
4.3.1	SharePoint 2010	5
4.3.2	Docstream	6
4.4	Architectuur.....	6
4.4.1	Overzicht.....	6
4.4.2	Data en Metadata.....	8
4.4.3	SyncServer	10
4.4.4	Communicatie tussen SyncServer en replica	10
4.4.5	SharePoint	11
4.4.6	Docstream	12
5	Conclusie	13
6	Bronnen	15
6.1	Literatuur.....	15
6.2	Websites.....	15

Samenvatting

In dit rapport wordt verslag gelegd over het onderzoek door Klaas van Hagen naar de mogelijkheden voor synchronisatie van documenten en metadata tussen SharePoint en Docstream. Het onderzoek is uitgevoerd in opdracht van Pro4all B.V.

Aan de hand van interviews met de opdrachtgever is onderzocht welke eisen aan synchronisatie worden gesteld en wat de mogelijkheden van Docstream voor synchronisatie zijn. De mogelijkheden van SharePoint voor synchronisatie en bestaande best-practices zijn onderzocht in een literatuuronderzoek en een verkennend experiment naar de mogelijkheden van het Microsoft Sync Framework.

Het resultaat van het onderzoek is een advies in de vorm van een architectuur waarmee synchronisatie van documenten en metadata tussen SharePoint en Docstream gerealiseerd kan worden en tegelijk de mogelijkheid biedt tot uitbreiding naar andere pakketten.

1 Inleiding

Docstream en SharePoint zijn beide systemen voor documentbeheer. Documentbeheer is slechts één van vele onderdelen van SharePoint. SharePoint is een product van Microsoft dat door middel van portal-websites een platform aanbiedt voor samenwerking binnen organisaties. Docstream is een systeem voor documentbeheer dat ontwikkeld is met het oog op de bouwsector. Docstream onderscheidt zich van andere documentbeheer systemen door de gebruiksvriendelijkheid.

Gebruikers op verschillende niveaus in een (project)organisatie stellen verschillende eisen aan systemen. Een directielid heeft behoefte aan rapportages maar zal weinig bouwtekeningen bewerken. SharePoint biedt uitgebreide rapportagemogelijkheden en biedt daarnaast functionaliteit voor documentbeheer. Een architect zal echter veel wijzigingen aanbrengen in bouwtekeningen. Daarvoor is het gebruiksvriendelijke Docstream beter geschikt dan SharePoint.

Om beide gebruikers te bedienen en toch over dezelfde documenten te laten beschikken wil Pro4all documenten tussen SharePoint en Docstream synchroniseren. In dit onderzoek worden de mogelijkheden hiervoor nagegaan.

Methode

Onderzoeksgroep

De onderdelen van dit onderzoek waarbij mensen bevraagd zijn voor informatie, de interviews, hebben binnen Pro4all plaatsgevonden. Hierin zijn twee groepen mensen te benoemen, het SharePoint team en het Docstream team.

Het SharePoint team bestaat uit: Antal van Deenen, Arthur Bleij, Raymond Pijpers, Rob Suurland en Pieter Beuker.

Uit het Docstream team heb ik gesproken met: Antal van Deenen en Daniël Bennekens.

Onderzoeksontwerp

De hoofdvraag waarop met dit onderzoek een antwoord is gezocht is:

Hoe kan synchronisatie van documenten en metadata tussen Microsoft SharePoint en Docstream gerealiseerd worden?

Om deze vraag te beantwoorden zijn een aantal deelvragen gedefiniëerd. Om deze deelvragen te beantwoorden zijn verschillende methoden voor het verzamelen van gegevens gebruikt.

1. Welke eisen worden er vanuit Pro4all aan de synchronisatie gesteld?

Op basis van gesprekken met de opdrachtgever is een lijst opgesteld die ter review is aangeboden aan het SharePoint team. De lijst is herhaalde malen bijgewerkt gedurende het onderzoek op basis van feedback en voortschrijdend inzicht.

2. Welke best-practices zijn er m.b.t. data synchronisatie?

Deze deelvraag is onderzocht in een literatuuronderzoek. Er is gebruik gemaakt van zowel digitale al analoge literatuur. De gebruikte analoge literatuur bestaat uit de literatuur die betrekking heeft op design patterns. Hiervoor is alleen literatuur onderzocht die ten tijde van het onderzoek binnen Pro4all beschikbaar was. Digitale literatuur is gezocht op internet via google en in de patterns & practices bibliotheek van MSDN. Op google is gezocht met de zoektermen: synchronization design pattern. Tot slot zijn publiekelijk toegankelijke digitale publicaties gebruikt waarnaar wordt verwezen op de engelstalige wikipedia pagina over software design patterns.

3. Welke mogelijkheden bied SharePoint 2010 om met andere pakketten te synchroniseren?

Deze deelvraag is in twee delen onderzocht, een literatuuronderzoek en een aantal experimenten. Voor het literatuuronderzoek is gebruik gemaakt van de online bibliotheek van Microsoft, de producent van SharePoint 2010. In de experimenten is de theorie uit het literatuuronderzoek in de praktijk gebracht.

4. Welke mogelijkheden bied Docstream om met andere pakketten te synchroniseren?

De mogelijkheden van Docstream om te synchroniseren zijn onderzocht in een aantal interviews met dhr. Van Deenen.

Resultaten

Eisen van Pro4all aan synchronisatie

De eisen die pro4all stelt aan het synchronisatiesysteem zijn vastgelegd in het document Requirements, dit document vind u als bijlage in dit rapport.

Best-practices voor datasynchronisatie

Dit onderdeel van het onderzoek heeft helaas maar weinig bruikbare informatie opgeleverd. In de onderzochte literatuur worden een aantal Design Patterns behandeld die betrekking hebben op concurrency. Problemen die optreden bij synchronisatie zijn nauw verwant aan die van concurrency. Wanneer men 2 systemen synchroniseert is dit eigenlijk niet veel anders dan dat beide systemen een virtuele database bijwerken. Deze virtuele database representeert bij aanvang van synchronisatie de status van beide systemen aan het eind van de voorgaande synchronisatie. Vervolgens wordt de virtuele database door beide systemen (ieder systeem een eigen sessie) bijgewerkt.

In zijn boek *Patterns of Enterprise Application Architecture* beschrijft Martin Fowler 4 Design Patterns gebaseerd op locking: Optimistic Offline Lock, Pessimistic Offline Lock, Coarse Grained Lock en Implicit Lock.^[1]

Optimistic Offline Lock werkt op basis van pre-commit validatie. Dit wil zeggen dat er voordat gegevens opgeslagen worden gecontroleerd wordt of de te wijzigen gegevens onveranderd zijn t.o.v. het moment waarop deze door de applicatie zijn opgehaald. Als de gegevens wel veranderd zijn is er sprake van een conflict en wordt de wijziging niet geaccepteerd.

Pessimistic Offline Lock maakt het onmogelijk om conflicten te veroorzaken. Gegevens worden bij aanvraag gelocked en pas nadat de gebruiker/sessie klaar is met wijzigen van de gegevens wordt de lock vrijgegeven. Er zijn een aantal lock-types die gebruikt kunnen worden met deze Design Pattern: exclusive write lock, exclusive read lock en read/write lock. Bij gebruik van de exclusive (exclusief) locks is de gebruiker/sessie met de lock de enige die de gegevens kan bewerken (exclusive write) en/of bekijken (exclusive read). Wanneer read/write lock gebruikt wordt kan slechts een gebruiker/sessie de gegevens bewerken, meerdere gebruikers kunnen gegevens bekijken. Een gebruiker/sessie kan geen write lock krijgen wanneer een of meer andere gebruikers een read lock op die gegevens heeft/hebben.

Coarse-Grained Lock biedt de mogelijkheden om groepen objecten die aan elkaar gerelateert zijn als geheel te locken. Het design pattern biedt 2 variaties: shared version lock en root lock. Bij shared version lock wordt een lock geplaatst op een object waarmee alle objecten in de groep een relatie hebben. Zo kan elk object in de groep via het gedeelde (shared) object controleren of het gelocked is. Bij root lock wordt uitgegaan van een hiërarchische structuur waarbij alleen het hoogste object in de structuur (root) een lock heeft. Alle onderliggende objecten kunnen via de hiërarchie de lock controleren of plaatsen. Dit pattern kan gebruikt worden in samenhang met Optimistic Offline Lock en Pessimistic Offline Lock.

Implicit Lock stelt dat het locken van objecten/gegevens niet aan de programmeur zelf overgelaten mag worden maar door het "Framework" (bijvoorbeeld gegenereerde code) wordt geïmplementeerd. Hierdoor kan niet meer voorkomen dat een programmeur de locking code vergeet of er een fout in maakt die grote gevolgen kan hebben voor de werking van de applicatie op de langere termijn.

Het pattern dat bij elke vorm synchronisatie toegepast wordt is Optimistic Offline Lock.

Mogelijkheden van pakketten tot synchronisatie

1.1.1 SharePoint 2010

Wie op MSDN zoekt naar Library Documentation & Articles met de zoektermen: "SharePoint" "2010" "synchronization" vind direct een artikel met de titel: "SharePoint Synchronization". Op MSDN heeft Microsoft een sectie met documentatie over synchronisatie voor SharePoint 2010. Uit artikels in deze sectie blijkt dat SharePoint gebruik maakt van synchronisatie om gebruikers offline toegang te bieden tot lijsten en documenten. Deze synchronisatie is wordt door SharePoint gerealiseerd m.b.v. het Microsoft Sync Framework. SharePoint biedt via de Lists.asmx webservice een aantal methoden om veranderingen te enumereren, en lijsten (en bibliotheken) bij te werken. Het HTTP/DAV protocol kan gebruikt worden om documenten bij te werken.^[3]

Een andere manier om lijsten te bewerken in SharePoint 2010 is het gebruik van het Client Object Model. Het Client Object Model is nieuw in SharePoint 2010 en biedt een client-API om de SharePoint Foundation webservices aan te spreken. De methoden voor het enumereren van wijzigingen zijn via het Client Object Model echter niet beschikbaar.

Het SharePoint 2010 Object Model kan gebruikt worden wanneer programmatuur wordt uitgevoerd in de context van de SharePoint server, bijvoorbeeld in een WebPart of Service Application.

Documenten bijwerken moet zowel in het geval van het Client Object Model als bij gebruik van het Object Model net als bij gebruik van de Lists webservice d.m.v. het HTTP/DAV protocol.

Het Microsoft Sync Framework is zoals de naam al suggereert een framework voor synchronisatie. Applicaties implementeren een applicatie-specifieke SyncProvider die zorgt voor registratie van metadata (versie en status van objecten op die replica) en CRUD (Create, Read, Update, Delete) operaties van objecten die gesynchroniseerd moeten worden) functionaliteit. De applicatie maakt een SyncOrchestrator object aan, koppelt een local en een remote SyncProvider en roept de methode Synchronize() aan op de SyncOrchestrator instantie. De SyncOrchestrator biedt 4 mogelijke methoden van conflicthantering: de locale replica wint, de remote replica wint, de wijzigingen worden samengevoegd (deze functionaliteit moet in de SyncProvider worden geïmplementeerd) of de wijziging wordt uitgesteld. Ook is het mogelijk om in het geval van een conflict terugkoppeling te laten vragen en d.m.v. programmatuur (bijvoorbeeld het tonen van een keuze-dialoog aan de gebruiker) een beslissing te nemen over de actie in een specifiek geval.^[4]

Testen met verschillende typen replicas (SharePoint, SQL Server en een XML Database) met daarvoor specifiek geïmplementeerde SyncProviders hebben aangetoond dat het Sync Framework door de provider architectuur snel kan reageren op koppelingen met nieuwe systemen, ook wanneer deze een ander datamodel hebben.

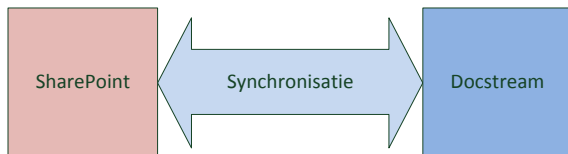
1.1.2 Docstream

Momenteel is er in Docstream geen mogelijkheid om te synchroniseren met andere pakketten. De synchronisatie kan wel rechtstreeks op de database uitgevoerd worden maar daarmee wordt business logica die in Docstream zit omzeilt. Een nieuwe versie van Docstream waarin gebruik wordt gemaakt van een z.g.n. servicelaag moet hiervoor een oplossing bieden. Op deze servicelaag kan rechtstreeks gekoppeld worden. Het is ook mogelijk om op deze laag een webservice te koppelen en via deze webservice te communiceren.

Architectuur

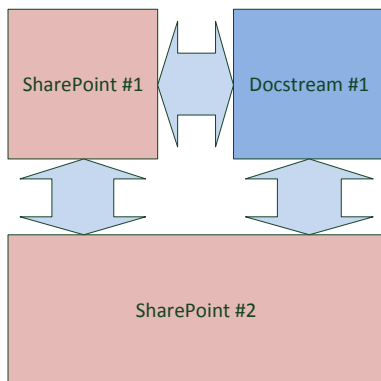
1.1.3 Overzicht

Op basis van de requirements die bepaald zijn in deelvraag 1 “Welke eisen worden er vanuit Pro4all aan de synchronisatie gesteld?” is het mogelijk om tot een globale architectuur (Figuur 1) te komen. Deze architectuur bestaat uit 2 componenten: SharePoint en Docstream.



Figuur 1

Wanneer meer dan 2 replica's met elkaar gesynchroniseerd moeten worden (requirement F9) wordt deze opzet van rechtstreekse communicatie een stuk ingewikkelder. In dit geval zullen òf alle replica's kennis moeten hebben van alle andere replica's (figuur 2) òf met maximaal 2 replica's kunnen synchroniseren (figuur 2).

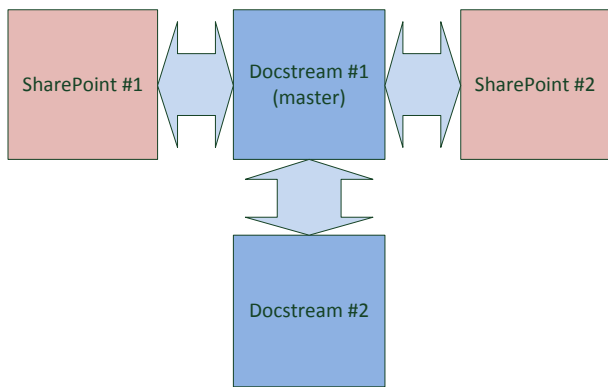


Figuur 2: Meerdere replica's onderling verbonden.

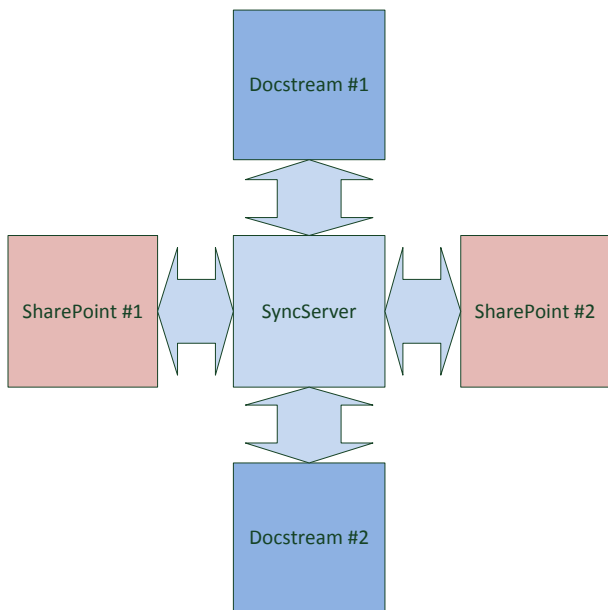


Figuur 3: Meerdere replica's serieel verbonden.

Om het synchroniseren van meerdere replica's overzichtelijk en beheer(s)baar te houden kan gebruik gemaakt worden van een ster topologie. Dit kan gerealiseerd worden door één replica als master aan te wijzen (figuur 4). In een topologie met veel replica's die verschillende synchronisatieprofielen gebruiken (requirement F3.4) kan op de master een zware werklast komen te liggen, dit kan de prestaties van die replica negatief beïnvloeden. In het slechtste geval kan dit het functioneren replica zelfs volledig stoppen. Niet alleen stopt de synchronisatie van alle replica's dan maar kunnen ook de gebruikers van die replica niet meer werken. Om dit te voorkomen kan een speciale master replica worden gebruikt, de SyncServer. Deze replica is nog steeds gevoelig voor de zware werklast maar wordt niet door gebruikers gebruikt. Gebruikers worden dus niet direct beïnvloed door een zware werklast op de SyncServer, ze kunnen gewoon doorwerken. Daarnaast wordt de SyncServer specifiek voor dit doel ontwikkeld, er kan dus bij de ontwikkeling rekening gehouden worden met deze problematiek.



Figuur 4: Stertopologie met een replica als master.



Figuur 5: Stertopologie met de SyncServer als master.

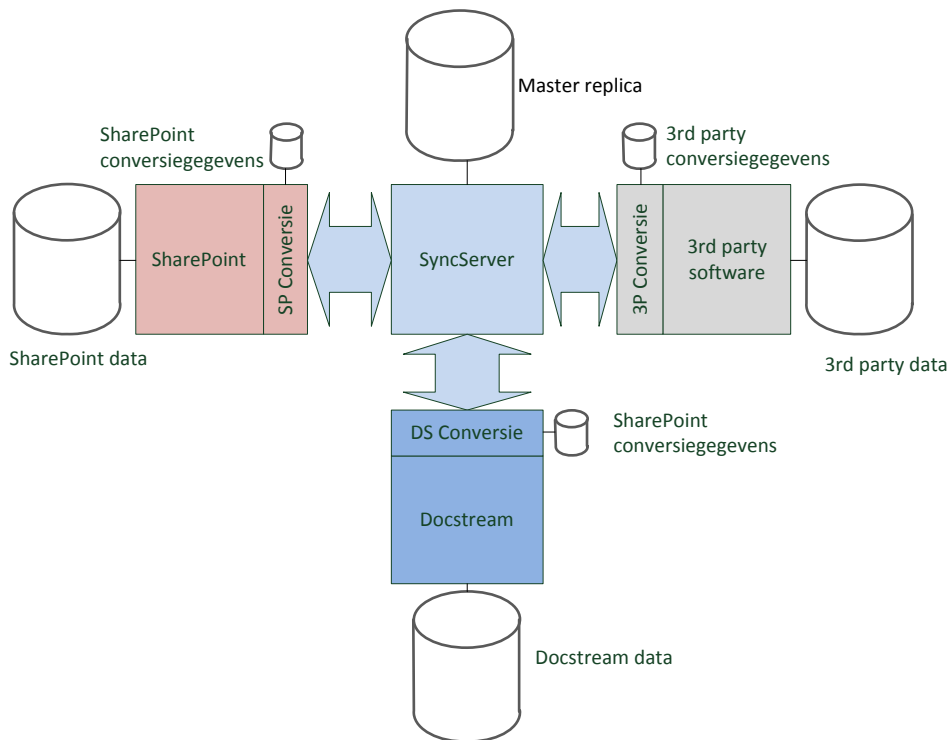
Een bijkomend voordeel van het gebruik van de SyncServer is dat dit component onafhankelijk van de andere componenten geïnstalleerd kan worden, hiermee voldoet deze architectuur aan requirement O1.

1.1.4 Data en Metadata

Verschillende systemen hebben verschillende dataformaten. Om te kunnen synchroniseren is het noodzakelijk dat er een vertaalslag gemaakt wordt tussen de te synchroniseren systemen. Omdat elke replica met de master-replica synchroniseert hoeft er per systeem slechts één vertaalslag gemaakt te worden: de vertaling van het systeemformaat naar het SyncServer formaat.

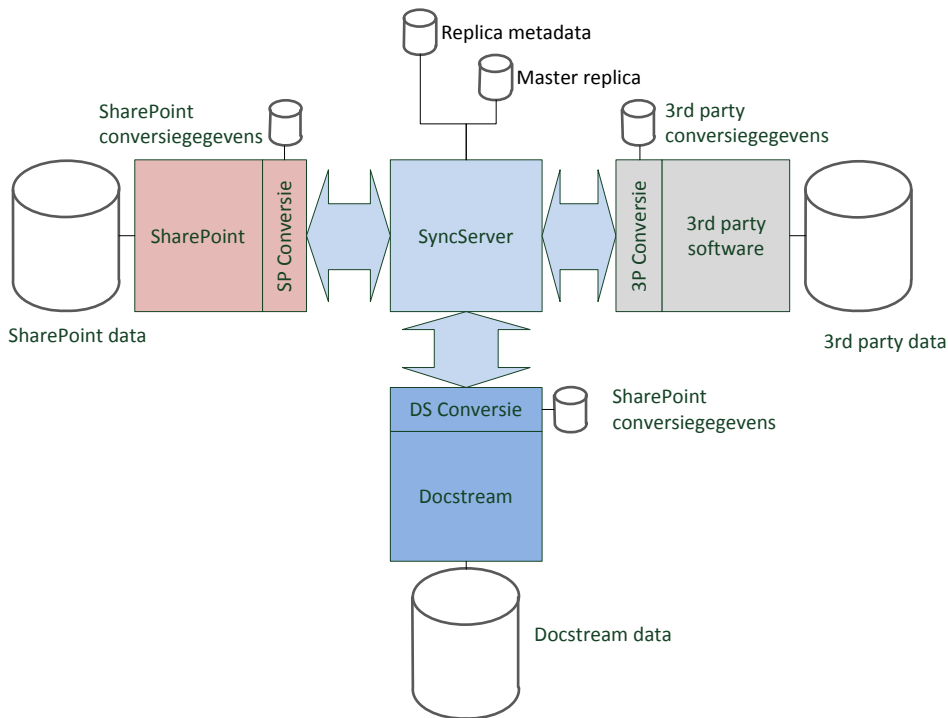
Voor het vertalen van gegevens van het ene formaat naar het andere is het soms noodzakelijk dat over die vertaling metadata opgeslagen wordt. Een voorbeeld hiervan is de vertaling van unieke waarden. In het ene systeem wordt een object uniek geïdentificeerd a.d.h.v. een numerieke waarde terwijl het in het andere systeem gebruik wordt gemaakt van een code die samengesteld is uit zowel cijfers als letters. Naast mogelijke verschillen in het formaat van de unieke identificatie verschilt de waarde hiervan vrijwel zeker omdat objecten in de master replica niet in alle replica's bestaan (requirement F3). Per systeem is een conversie-component noodzakelijk. Conversiegegevens moeten

per replica opgeslagen worden. Het conversiecomponent en het formaat van de conversiegegevens zijn specifiek per systeem (figuur 6).



Figuur 6: Conversiecomponenten met conversiegegevens.

Om een replica te kunnen synchroniseren moet de SyncServer op de hoogte zijn van de op die replica aanwezige documenten en hun versies. Op basis hiervan kan de SyncServer berekenen welke objecten op een replica gemaakt, bijgewerkt of verwijderd moeten worden. De SyncServer houdt deze gegevens bij in een database (requirement D2). Dit is een andere database dan de master-replica zodat deze apart geoptimaliseerd kunnen worden en elkaars prestaties zo min mogelijk beïnvloeden (figuur 7).



Figuur 7: Replica metadata wordt door de SyncServer opgeslagen.

1.1.5 SyncServer

De SyncServer is verantwoordelijk voor het synchroniseren van de master-replica met alle gekoppelde replica's. Uit het literatuuronderzoek en de verkennende experimenten rondom het Microsoft Sync Framework is gebleken dat dit framework zeer geschikt is voor deze toepassing. Dit framework biedt functionaliteit voor het berekenen van wijzigingen en is erg flexibel te gebruiken met verschillende systemen door het gebruik van het Provider design pattern. De SyncOrchestrator dirigeert het synchronisatieproces tussen twee SyncProvider objecten.

1.1.6 Communicatie tussen SyncServer en replica

De communicatie tussen de SyncServer en de replica's zal vrijwel altijd via een netwerk verlopen. Om flexibel met derde partijen te kunnen koppelen wordt gebruik gemaakt van SOAP webservices. Een probleem met communicatie via webservices is dat deze altijd vanuit één zijde wordt geïnitieerd. Het werkt op basis van verzoek en antwoord. Voor synchronisatie is echter bi-directionele communicatie nodig. Microsoft's Windows Communication Foundation (WCF) biedt de mogelijkheid om SOAP services over verschillende transport-typen aan te bieden. Hierdoor is het mogelijk om SOAP berichten direct over TCP te sturen. Daarnaast heeft WCF de wsDualHttpBinding, deze methode van berichttransport ondersteund bidirectionele communicatie over HTTP.

WCF- services communiceren met een vooraf afgesproken formaat, het service contract. Om de service te gebruiken implementeert een systeem dit contract.

Om de koppeling met systemen zoveel mogelijk te standaardiseren zal zoveel mogelijk gebruik worden gemaakt van één servicecontract. Door één SyncProvider implementatie te maken die wordt aangestuurd door een instantie van de WCF-service zorgt ervoor dat de SyncProvider functionaliteit gelijk (en getest) blijft over verschillende systemen. Wanneer een 3^e partij een koppeling wil realiseren hoeft deze partij hiervoor geen kennis te hebben van het Microsoft Sync Framework en de

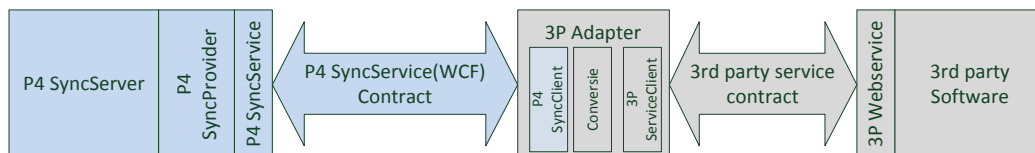
werking van de SyncProvider. Daarnaast is het de verantwoordelijkheid van de client-implementatie om zorg te dragen voor conversie naar het gezamenlijke dataformaat (figuur 8).



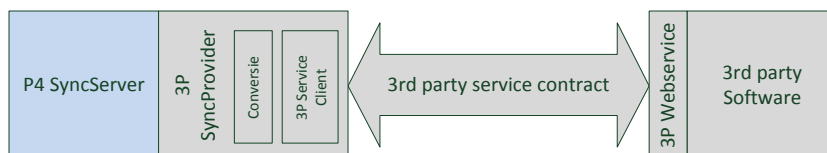
Figuur 8: Implementatie van het Service Contract en conversie door 3e partij.

Wanneer een 3^e partij een webservice aanbied waarmee gekoppeld moet worden kan dat op 2 manieren:

1. Pro4all implementeert een adapter, dit component bestaat uit een client voor de P4 SyncService, een conversie-component en een client voor de webservice v/d 3^e partij (figuur 9).
2. Pro4all implementeert een SyncProvider met daarin een adapter-component en een client voor webserivce v/d 3^e partij (figuur 10).



Figuur 9: Pro4all ontwikkelt een adapter om met webservices van 3^{den} te communiceren.



Figuur 10: Pro4all ontwikkelt een SyncProvider om rechtstreeks met webservices van 3^{den} te communiceren.

De 1^e optie, het gebruik van een apart adapter component biedt grote flexibiliteit. De adapter kan bijvoorbeeld in een aparte applicatie geïmplementeerd worden en op een server in een DMZ geïnstalleerd worden zodat het component veilig met beide systemen kan communiceren (requirement O1). Daarnaast hoeven er voor deze manier slechts configuratie wijzigingen aangebracht te worden. De basis van het systeem (SyncServer, SyncProvider, SyncService) blijft ongewijzigd (requirement D1).

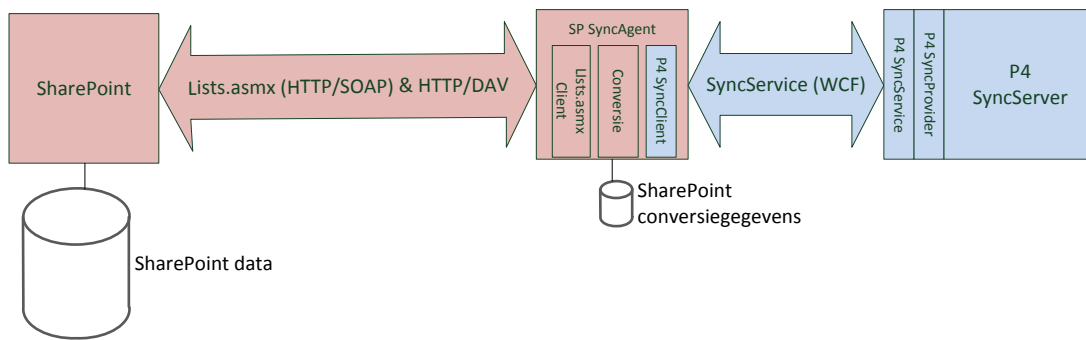
De bovenstaande indelingen zijn tevens te hanteren wanneer de software in kwestie met een protocol communiceert.

1.1.7 SharePoint

Zoals eerder besproken biedt SharePoint 3 verschillende manieren om lijsten te bewerken: middels het Client Object Model, middels het Object Model en middels webservices. Het bijwerken van bestanden kan alleen via HTTP/DAV. Alleen door het gebruik van webservices is het mogelijk om de slechts gewijzigde items van een lijst op te vragen. Dit heeft grote voordelen t.o.v. het enumereren van alle synchroniseerbare items in een lijst m.b.t. de hoeveelheid communicatie tussen de

SyncServer en SharePoint alsmede de belasting op de SyncServer omdat deze bij iedere synchronisatie de status van alle items moet vergelijken met die na de vorige synchronisatie.

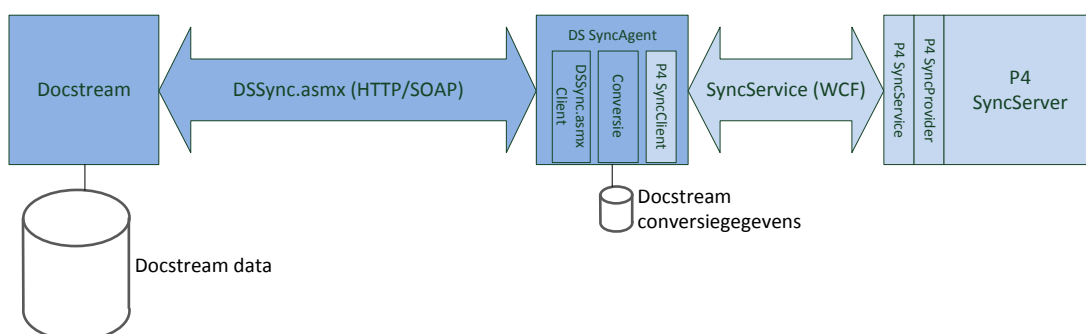
Omdat Pro4all gebonden is aan de webservices die SharePoint aanbiedt en niet in een onderhandelingspositie is zal één van de indelingen van figuren 9 en 10 gebruikt moeten worden om SharePoint in de synchronisatie op te kunnen nemen. Zoals eerder geconcludeerd heeft de indeling van figuur 9 de voorkeur. De implementatie van SharePoint in het synchronisatiesysteem is weergegeven in figuur 11.



Figuur 11: SharePoint implementatie in het Synchronisatiesysteem.

1.1.8 Docstream

Net als SharePoint verbindt Docstream met de SyncServer via een SyncAgent. De SyncAgent koppelt met de service-laag van Docstream. Dit kan op 2 manieren: via een daarvoor ontwikkelde webservice of rechtstreeks. In het geval dat dit rechtstreeks gebeurt is de SyncAgent eigenlijk een instantie van Docstream zonder GUI (Graphical User Interface). Het gebruik van een webservice biedt echter de mogelijkheid om Docstream in de omgeving te gebruiken waarvoor deze ontwikkeld is (hosted op IIS/ASP.NET). Daarnaast is het koppelen met een webservice flexibeler omdat de SyncAgent geen verbinding hoeft te kunnen maken met de Docstream database. Hierdoor kan de SyncAgent geïnstalleerd worden op een locatie die wél toegang heeft tot de (beveiligde) webservice van Docstream maar niet de database. Deze webservice is verantwoordelijk voor het leveren van zowel document-metadata als de documenten zelf. De implementatie van Docstream in het synchronisatiesysteem is weergegeven in figuur 12.



Figuur 12: Docstream implementatie in het synchronisatiesysteem.

Conclusie

In dit onderzoek is onderzocht aan welke eisen de synchronisatie van documenten en documentmetadata tussen SharePoint en Docstream moeten voldoen en hoe dit te realiseren is.

In de voorgaande hoofdstukken heeft u de antwoorden kunnen lezen op de deelvragen op basis waarvan het antwoord op de hoofdvraag is geformuleerd. In dit deel van dit rapport wordt een architectuur gepresenteert in de vorm van een architectuur.

De hoofdvraag luidt: **Hoe kan synchronisatie van documenten en metadata tussen Microsoft SharePoint en Docstream gerealiseerd worden?**

De aanbevolen architectuur van het systeem voor synchronisatie van documenten en documentmetadata is weergegeven in figuur 13. Een gedetailleerde weergave vind u in de bijlage Architectuur. Hieronder volgt een korte beschrijving van de componenten.

Replica:

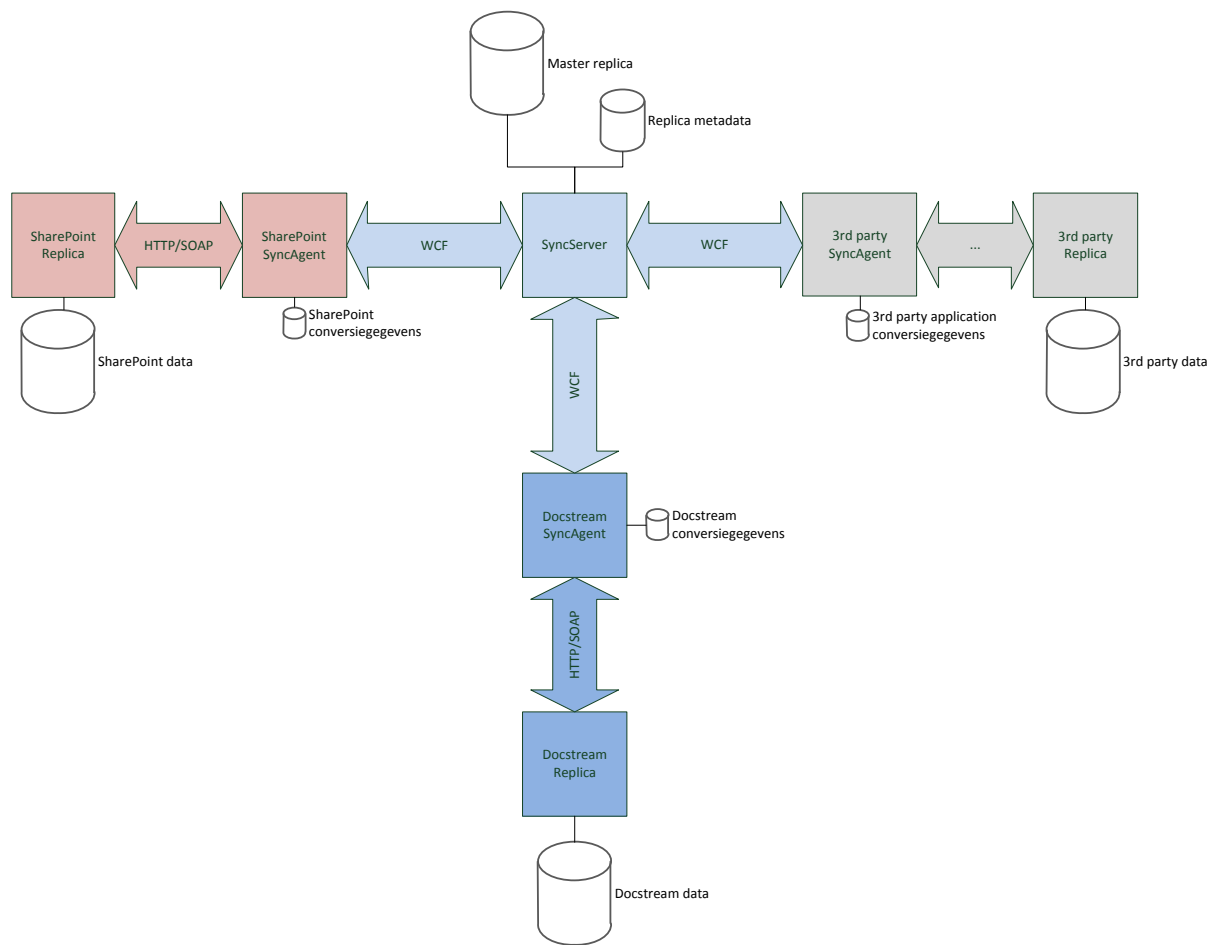
De replica is het te synchroniseren systeem, dit kan een SharePoint of Docstream installatie zijn maar ook andere pakketten voor documentbeheer. Ontwikkeling en onderhoud aan de replica valt (met uitzondering van Docstream) buiten het bereik van Pro4all.

SyncAgent:

De SyncAgent koppelt Replica aan de SyncServer. Dit is het component dat voor ieder type replica opnieuw ontwikkeld moet worden. Het communiceert middels een WCF service met de SyncServer volgens een vooraf afgesproken formaat. De SyncAgent is verantwoordelijk voor communicatie met de Replica middels een replica-specifiek protocol (voor SharePoint HTTP/SOAP en HTTP/DAV, voor Docstream met HTTP/SOAP) en *vertaalt* gegevens van het formaat v/d replica naar het formaat v/d SyncServer. Op de SyncAgent wordt tevens bepaalt welke gegevens andere replica's met die replica mogen synchroniseren.

SyncServer:

De SyncServer is het centrale punt waarmee alle replica's synchroniseren. Hiervoor wordt gebruik gemaakt van het Microsoft Sync Framework. Alle gesynchroniseerde documenten worden met bijbehorende metadata opgeslagen in een database op de SyncServer: de Master replica. Gegevens en statistieken over de synchronisatie zelf worden door de SyncServer opgeslagen in de Replica metadata. De SyncServer is tevens verantwoordelijk voor het afschermen van gegevens uit de Master replica voor specifieke Replica's gebaseerd op de regels die op de SyncAgent van de bron-replica zijn ingesteld.



Figuur 13: Overzicht architectuur, globale componenten en communicatie

Bronnen

Literatuur

[1] Martin Fowler, Patterns of Enterprise Application Architecture
10e druk 2006, Pearson Education, Inc.

Websites

[2] Microsoft SharePoint 2010 beta Development Center:

<http://msdn.microsoft.com/en-us/sharepoint/ee514561.aspx>

[3] SharePoint Synchronization:

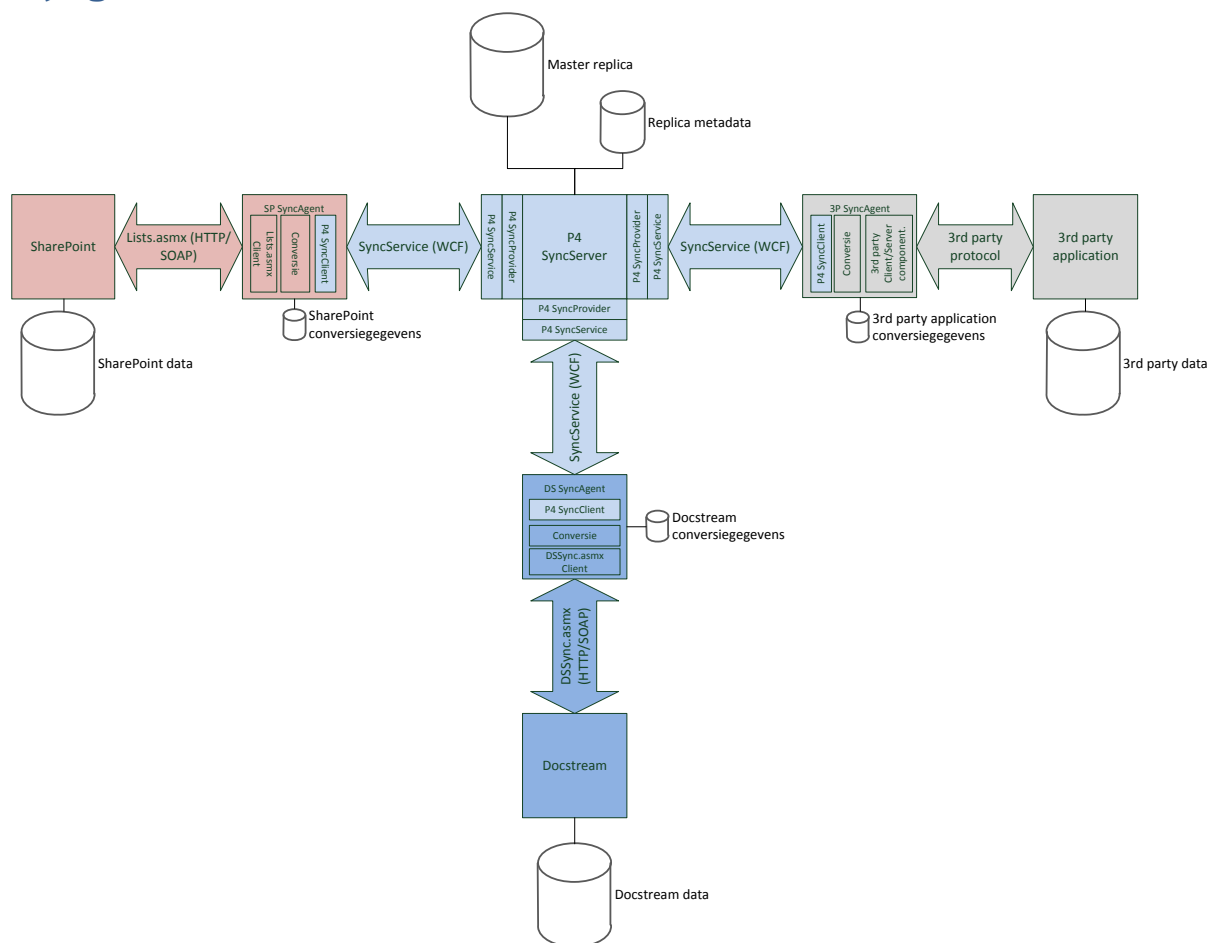
<http://msdn.microsoft.com/en-us/library/ee537897%28office.14%29.aspx>

[4] Microsoft Sync Framework Development Center:

<http://msdn.microsoft.com/en-us/sync/default.aspx>

Bijlage I Requirements

Bijlage II Architectuur



Figuur 14: Gedetailleerde weergave van de architectuur voor synchronisatie.

F Requirements

Pro4all B.V.

Requirements DMES

Eisen en wensen aan de architectuur van DMES.

Klaas van Hagen
26 mei 2010

Functionele requirements

1. Het systeem kan wijzigingen aan documenten synchroniseren met verbonden systemen.
2. Er zijn 3 soorten wijzigingen aan documenten mogelijk.
 - 2.1. Het betreft nieuwe versie van een document.
 - 2.2. Het betreft bijgewerkte metadata van een documentversie.
 - 2.3. Het betreft het verwijderen van een bestaand document.
3. De eigenschappen van synchronisatie worden per verbonden systeem ingesteld.
 - 3.1. Er zijn 3 typen synchronisatie mogelijk.
 - 3.1.1. Externe wijzigingen aan documenten worden door het systeem doorgevoerd.
 - 3.1.2. Lokale wijzigingen aan documenten worden aan externe systemen aangeboden.
 - 3.1.3. Externe wijzigingen aan documenten worden door het systeem doorgevoerd en Lokale wijzigingen aan documenten worden aan externe systemen aangeboden.
 - 3.2. Wijzigingen aan documenten worden gefilterd aan verbonden systemen aangeboden.
 - 3.2.1. Wijzigingen aan documenten worden gefilterd op basis van document-metadata.
 - 3.2.1.1. Alleen goedgekeurde documentversies worden gesynchroniseerd.
 - 3.2.1.2. Alleen definitieve documentversies worden gesynchroniseerd.
 - 3.2.2. Wijzigingen aan documenten worden gefilterd op basis van oorsprong van de wijziging.
4. Wijzigingen aan documenten worden door het systeem opgeslagen.
 - 4.1. Een document heeft een of meer documentversies.
 - 4.1.1. Een documentversie bevat data.
 - 4.1.2. Een documentversie bevat document-metadata
 - 4.1.2.1. Document-metadata kan onafhankelijk van een documentversie gewijzigd worden.
 - 4.1.2.2. Het systeem bouwt historie op van gewijzigde document-metadata
 - 4.1.3. Het systeem bouwt historie op van gemaakte documentversies.
 - 4.2. Extern verwijderde documenten worden door het systeem als verwijderd gemarkeerd.
5. Het systeem slaat historische gegevens op van wijzigingen aan documenten.
 - 5.1. Een document wordt niet door het systeem verwijderd.
 - 5.2. Een documentversie wordt niet door het systeem verwijderd.
 - 5.3. Document-metadata wordt niet door het systeem verwijderd.
6. Het systeem slaat metadata van wijzigingen op in een database.
7. Het systeem kan rapporten genereren.
 - 7.1. Het systeem kan een rapport genereren over de synchronisatie status van een document.
 - 7.2. Het systeem kan een rapport genereren over de synchronisatie status van een verbonden systeem.
 - 7.3. Het systeem kan een rapport genereren over de gedeelde mappen en documenten van een verbonden systeem.
8. Het systeem houdt een logboek bij.
 - 8.1. Uitgevoerde acties worden in het logboek geregistreerd.
9. Wijzigingen aan autorisaties voor propagatie van wijzigingen aan documenten moeten door een daartoe bevoegd persoon worden bevestigd.

Niet-functionele requirements

1. Communicatie tussen het systeem en verbonden systemen zowel lokaal als extern geïnitieerd worden.
2. Het systeem is veilig.
 - 2.1. Wijzigingen aan documenten worden enkel aan daartoe geautoriseerde verbonden systemen aangeboden.
 - 2.1.1. Per verbonden systeem wordt aangegeven welke andere verbonden systemen wijzigingen aan documenten van dat systeem mogen ontvangen.
 - 2.1.2. Per verbonden systeem wordt aangegeven van welke andere verbonden systemen gewijzigde documenten ontvangen worden.
 - 2.2. Communicatie met verbonden systemen gebeurt over een met SSL-encryptie beveiligde verbinding.
 - 2.3. Verbonden systemen moeten zich identificeren voordat wijzigingen aan documenten of andere gegevens ermee worden overlegd.
 - 2.3.1. Identificatie geschied op basis van identificatiecode, wachtwoord en client-certificaat.
3. Het systeem is betrouwbaar.
 - 3.1. Het systeem is bestand tegen onderbreken van de synchronisatie door:
 - 3.1.1. Verlies van verbinding met het verbonden systeem.
 - 3.1.2. Verlies van verbinding met de database.
 - 3.1.3. Software crash.
 - 3.1.4. Hardware crash.
4. Het systeem is efficiënt.
 - 4.1. Het systeem kan met meerdere verbonden systemen tegelijk een synchronisatieproces uitvoeren met acceptabele performance.
 - 4.2. Tijdens het synchronisatieproces gebruikte bronnen worden na beëindiging hiervan direct vrijgegeven.
 - 4.2.1. Gebruikt computergeheugen.
 - 4.2.2. Gebruikte tijdelijke schijfruimte.
 - 4.2.3. Database verbindingen.
 - 4.2.4. Verbinding met het verbonden systeem.
 - 4.3. Alle communicatie met externe systemen verloopt asynchroon.
5. Het systeem is onderhoudbaar.
 - 5.1. Het systeem kan zonder wijzigingen aan de basis uitgebreid worden om te verbinden met nieuwe typen systemen.

G Requirements Proof-of-concept

H Requirements DMES

I Use-Cases

J Klassediagrammen

K Sequencediagrammen

L Master testplan

Pro4all B.V.

Master testplan Proof-of-Concept

Master testplan voor het proof-of-concept van de DMES architectuur.

Inhoudsopgave

1	Opdracht.....	1
1.1	Opdrachtgever.....	1
1.2	Opdrachtnemer	1
1.3	Testsoorten.....	1
1.4	Beschouwingsgebied	1
1.5	Doel	1
1.6	Randvoorwaarden	1
2	Teststrategie.....	2
2.1	Kwaliteitsattributen.....	2
2.2	Testsoorten per kwaliteitsattribuut.	2
	Testorganisatie	3
2.3	Funcities en taken	3
3	Infrastructuur	3
3.1	Testomgeving	3
3.2	Testtools	4
3.3	Kantoorinrichting.....	4
4	Planning	4

1 Opdracht

1.1 Opdrachtgever

De opdrachtgever is Pro4all B.V.

1.2 Opdrachtnemer

De opdrachtnemer is Klaas van Hagen, trainee bij Pro4all B.V.

1.3 Testsoorten

Dit mastertestplan heeft betrekking op toetsen, de systeemtest en de functionele acceptatietest. De programma-, en productie acceptatietests wordt buiten beschouwing gelaten.

1.4 Beschouwingsgebied

Het testproces heeft betrekking op het proof-of-concept van deDMES (Document Management Enterprise Synchronization) architectuur. Hieronder vallen de SyncServer, de Docstream SyncAgent en de SharePoint SyncAgent met onderliggende componenten en onderlinge interfaces.

De functionaliteit van Microsoft SharePoint en Docstream valt buiten de beschouwing van de testopdracht.

1.5 Doel

Het doel van het testproces is vaststellen of de kwaliteit van het proof-of-concept voldoende is ten aanzien van de daaraan gestelde functionele en niet-functionele eisen.

1.6 Randvoorwaarden

Het testproces dient uiterlijk op 3 juni te zijn voltooid.

Oplevering van de testbasis en testobjecten volgens de onderstaande planning.

		Projectweek:	12	13	14	15	16
Iteratie	Taak	Kalenderweek:	17	18	19	20	21
Master testplan	Opstellen master testplan						
SyncServer	Opstellen use-cases en use-case diagrammen						
	Opstellen requirements						
	Opstellen testplan						
	Opstellen klassediagram						
	Opstellen sequentiediagram(men)						
	Programmeren en testen						
	Testen & opstellen Testrapport						
SharePoint SyncAgent	Opstellen use-cases en use-case diagrammen						
	Opstellen requirements						
	Opstellen testplan						
	Opstellen klassediagram						
	Opstellen sequentiediagram(men)						
	Programmeren en testen						
	Testen en opstellen Testrapport						
Docstream SyncAgent	Opstellen use-cases en use-case diagrammen						
	Opstellen requirements						
	Opstellen testplan						
	Opstellen klassediagram						
	Opstellen sequentiediagram(men)						
	Programmeren en testen						
	Testen en opstellen Testrapport						

Figuur 1: Planning testbasis en testobjecten.

Het testproces wordt met uitzondering van de acceptatietests geheel door Klaas van Hagen uitgevoerd.

2 Teststrategie

2.1 Kwaliteitsattributen

ISO9126 Kwaliteitsattribuut	Rel. belang
Accuracy	50%
Interoperability	40%
Changeability	10%
Totaal	100%

Tabel 1: Kwaliteitsattributen volgens ISO9126 met hun relatief belang.

2.2 Testsoorten per kwaliteitsattribuut.

In de onderstaande tabel vind u een overzicht van de testsoorten die per ISO 9126 kwaliteitsattribuut worden uitgevoerd.

In de kolomkoppen worden de volgende afkortingen gebruikt:

- TDS: Toest Definitiestudie
- TFO: Toets functioneel ontwerp
- TTO: Toets technisch ontwerp
- ST: Systeemtest
- FAT: Functionele acceptatietest

	TDS	TFO	TTO	ST	FAT	Rel. belang
Accuracy				+	+	50%
Interoperability				+		40%
Changeability			+			10%
Totaal						100%

Tabel 2: Testsoorten per ISO9126 kwaliteitsattribuut.

Testorganisatie

2.3 Functies en taken

De in dit project gebruikte functies zijn Testen en Testmanagement.

2.3.1 Testen

De functie testen heeft de volgende taken:

- Specificatie van logische & fysieke testgevallen en uitgangsbestanden.
- Vullen van uitgangsbestanden.
- Uitvoeren van testgevallen.
- Uitvoeren van controles en onderzoeken.
- Registreren van bevindingen.

2.3.2 Testmanagement

De functie testmanager heeft de volgende taken:

- Opstellen, verkrijgen van goedkeuring en onderhouden van het testplan.
- Uitvoeren van het testplan binnen planning en budget.
- Rapporteren over de voortgang van het testproces en de kwaliteit van het testobject.

3 Infrastructuur

3.1 Testomgeving

Om de tests uit te kunnen voeren is de onderstaande omgeving noodzakelijk:

- Hardware
 - Een laptop met de volgende minimale systeemeisen:
 - Processorarchitectuur: x64
 - Geheugen: >= 2 GB
 - Harde schijf: >= 40 GB
- Software
 - 64bit Windows (vista/7/2008/2008 R2)
 - Microsoft Office 2007 of nieuwer
 - Microsoft SharePoint 2010
 - Microsoft SQL Server 2008 R2
 - Microsoft SQL Server Management Studio 2008 R2
 - Microsoft Sync Framework 2.0 redistributable
 - Docstream 3

3.2 Testtools

De testtools die voor het testen gebruikt zullen worden zijn:

- Microsoft SQL Server Management Studio 2008 R2:
 - Wordt gebruikt om bestanden te herstellen bij aanvang van tests.
- Microsoft SharePoint 2010:
 - Wordt gebruikt bij het vullen van bestanden tijdens het uitvoeren van tests.

3.3 Kantoorinrichting

Het uitvoeren van de tests wordt gedaan op een van de flexplekken van het kantoor.

4 Planning

Activiteit	TTO	ST	FAT
Opstellen testscripts	WK 12-15	WK 12-15	WK 12-15
Testen	WK 12-15	WK 12-15	WK 12-15

M Systeem testplan

Pro4all B.V.

Systeemtestplan Proof-of-Concept

Systeemtestplan voor het proof-of-concept van de DMES architectuur.

Inhoudsopgave

1	Opdracht.....	1
1.1	Opdrachtgever.....	1
1.2	Opdrachtnemer	1
1.3	Beschouwingsgebied	1
1.4	Doel	1
1.5	Randvoorwaarden	1
2	Testbasis	1
3	Teststrategie.....	1
3.1	Kwaliteitsattributen.....	1
3.2	Strategiematrix	2
3.3	Testtechnieken per deelsysteem	2
4	Testorganisatie	2
4.1	Functies en taken	2
4.2	Functie invulling	2
5	Infrastructuur	2
5.1	Testomgeving	2
5.2	Testtools	3
5.3	Kantoorinrichting.....	3
6	Beheer	3
7	Planning.....	3
8	Risicos	3

1 Opdracht

1.1 Opdrachtgever

De opdrachtgever is Pro4all B.V.

1.2 Opdrachtnemer

De opdrachtnemer is Klaas van Hagen, trainee bij Pro4all B.V.

1.3 Beschouwingsgebied

Het testproces heeft betrekking op het proof-of-concept van deDMES (Document Management Enterprise Synchronization) architectuur. Hieronder vallen de SyncServer, de Docstream SyncAgent en de SharePoint SyncAgent met onderliggende componenten en onderlinge interfaces.

De functionaliteit van Microsoft SharePoint en Docstream valt buiten de beschouwing van de testopdracht.

1.4 Doel

Het doel van het testproces is vaststellen of de kwaliteit van het proof-of-concept voldoende is ten aanzien van de daaraan gestelde functionele en niet-functionele eisen.

1.5 Randvoorwaarden

Het testproces dient uiterlijk op 3 juni te zijn voltooid.

2 Testbasis

De testbasis is gevormd door :

Productdocumentatie:

- Het functioneel ontwerp van het proof-of-concept.
- De requirements van het proof-of-concept

Normen en standards:

- ISO 9126 kwaliteitskenmerken voor software.

3 Teststrategie

3.1 Kwaliteitsattributen

ISO9126 Kwaliteitsattribuut	Rel. belang
Accuracy	80%
Interoperability	20%
Totaal	100%

Tabel 1: Kwaliteitsattributen volgens ISO9126 met hun relatief belang.

3.2 Strategiematrix

In de onderstaande tabel vind u een overzicht van de ISO 9126 kwaliteitsattributen, deelsystemen en de mate waarop deelsystemen

	SyncServer	DSSyncAgent	SPSyncAgent	Totaal systeem	Rel. belang
Accuracy	+	+	+		80%
Interoperability				+	20%
Totaal	40%	20%	20%	20%	100%

Tabel 2: Testsoorten per ISO9126 kwaliteitsattribuut.

3.3 Testtechnieken per deelsysteem

Deelsysteem	Testtechnieken
SyncServer	Zie Systeemtest_POC
DSSyncAgent	Zie Systeemtest_POC
SPSyncAgent	Zie Systeemtest_POC
Totaal systeem	Zie Systeemtest_POC

4 Testorganisatie

4.1 Functies en taken

4.1.1 Tester

De functie tester heeft de volgende taken:

- Specificatie van logische & fysieke testgevallen en uitgangsbestanden.
- Vullen van uitgangsbestanden.
- Uitvoeren van testgevallen.
- Uitvoeren van controles en onderzoeken.
- Registreren van bevindingen.

4.1.2 Testmanagement

De functie testmanager heeft de volgende taken:

- Opstellen, verkrijgen van goedkeuring en onderhouden van het testplan.
- Uitvoeren van het testplan binnen planning en budget.
- Rapporteren over de voortgang van het testproces en de kwaliteit van het testobject.

4.2 Functie invulling

Functie	Uitvoerende persoon
Tester	K. van Hagen
Testmanagement	K. van hagen

5 Infrastructuur

5.1 Testomgeving

Om de tests uit te kunnen voeren is de onderstaande omgeving noodzakelijk:

- Hardware
 - Een laptop met de volgende minimale systeemeisen:
 - Processorarchitectuur: x64
 - Geheugen: >= 2 GB
 - Harde schijf: >= 40 GB
- Software
 - 64bit Windows (vista/7/2008/2008 R2)
 - Microsoft Office 2007 of nieuwer
 - Microsoft SharePoint 2010
 - Microsoft SQL Server 2008 R2
 - Microsoft SQL Server Management Studio 2008 R2
 - Microsoft Sync Framework 2.0 redistributable
 - Docstream 3

5.2 Testtools

De testtools die voor het testen gebruikt zullen worden zijn:

- Microsoft SQL Server Management Studio 2008 R2:
 - Wordt gebruikt om bestanden te herstellen bij aanvang van tests.
- Microsoft SharePoint 2010:
 - Wordt gebruikt bij het vullen van bestanden tijdens het uitvoeren van tests.

5.3 Kantoorinrichting

Het uitvoeren van de tests wordt gedaan op een van de flexplekken van het kantoor.

6 Beheer

Gezien de omvang van het testproject wordt er alleen beheer uitgevoerd op de testbasis en de testdocumentatie

7 Planning

Activiteiten	Start	Eind
Opstellen testgevallen	WK 12	WK 15
Testen	WK 12	WK 16

8 Risicos

Het uitvoeren van de systeemtests is onderhevig aan de volgende risico's:

Software is slecht testbaar: Indien de software slecht te testen is kan dit gevolgen hebben voor de kwaliteit van de testen en het uitvoeren van het testen volgens de planning.

Afhankelijkheden van de software zijn niet stabiel: Indien het testobject afhankelijk is van software buiten zijn invloedssfeer vormt dit een bedreiging voor de validiteit van het testproces.

Problemen met de infrastructuur maken het testen onmogelijk: Problemen met de infrastructuur kunnen verhinderen dat het testproces uitgevoerd kan worden.

De tester is tevens ontwikkelaar van het product: Dit vormt het risico dat het uitvoeren van testen wordt vervuld door voorkennis van het testobject.

N Acceptatie testplan

Pro4all B.V.

Acceptatietestplan Proof-of-Concept

Acceptatietestplan voor het proof-of-concept van de DMES architectuur.

Inhoudsopgave

1	Opdracht.....	1
1.1	Opdrachtgever.....	1
1.2	Opdrachtnemer	1
1.3	Beschouwingsgebied	1
1.4	Doel	1
1.5	Randvoorwaarden	1
2	Testbasis	1
3	Teststrategie.....	1
3.1	Kwaliteitsattributen.....	1
3.2	Strategiematrix	2
3.3	Testtechnieken per deelsysteem	2
4	Testorganisatie	2
4.1	Functies en taken	2
4.2	Functie invulling	2
5	Infrastructuur	2
5.1	Testomgeving	2
5.2	Testtools	3
5.3	Kantoorinrichting.....	3
6	Beheer	3
7	Planning.....	3
8	Risicos	3

1 Opdracht

1.1 Opdrachtgever

De opdrachtgever is Pro4all B.V.

1.2 Opdrachtnemer

De opdrachtnemer is Klaas van Hagen, trainee bij Pro4all B.V.

1.3 Beschouwingsgebied

Het testproces heeft betrekking op het proof-of-concept van deDMES (Document Management Enterprise Synchronization) architectuur. Hieronder vallen de SyncServer, de Docstream SyncAgent en de SharePoint SyncAgent met onderliggende componenten en onderlinge interfaces.

De functionaliteit van Microsoft SharePoint en Docstream valt buiten de beschouwing van de testopdracht.

1.4 Doel

Het doel van het testproces is vaststellen of de kwaliteit van het proof-of-concept voldoende is ten aanzien van de daaraan gestelde functionele en niet-functionele eisen.

1.5 Randvoorwaarden

Het testproces dient uiterlijk op 3 juni te zijn voltooid.

2 Testbasis

De testbasis is gevormd door :

Productdocumentatie:

- Het functioneel ontwerp van het proof-of-concept.
- De requirements van het proof-of-concept

Normen en standards:

- ISO 9126 kwaliteitskenmerken voor software.

3 Teststrategie

3.1 Kwaliteitsattributen

ISO9126 Kwaliteitsattribuut	Rel. belang
Accuracy	80%
Interoperability	20%
Totaal	100%

Tabel 1: Kwaliteitsattributen volgens ISO9126 met hun relatief belang.

3.2 Strategiematrix

In de onderstaande tabel vind u een overzicht van de ISO 9126 kwaliteitsattributen, deelsystemen en de mate waarop deelsystemen

	SyncServer	DSSyncAgent	SPSyncAgent	Totaal systeem	Rel. belang
Accuracy	+	+	+		80%
Interoperability				+	20%
Totaal	40%	20%	20%	20%	100%

Tabel 2: Testsoorten per ISO9126 kwaliteitsattribuut.

3.3 Testtechnieken per deelsysteem

Deelsysteem	Testtechnieken
SyncServer	Zie Acceptatietest_POC
DSSyncAgent	Zie Acceptatietest_POC
SPSyncAgent	Zie Acceptatietest_POC
Totaal systeem	Zie Acceptatietest_POC

4 Testorganisatie

4.1 Functies en taken

4.1.1 Tester

De functie tester heeft de volgende taken:

- Specificatie van logische & fysieke testgevallen en uitgangsbestanden.
- Vullen van uitgangsbestanden.
- Uitvoeren van testgevallen.
- Uitvoeren van controles en onderzoeken.
- Registreren van bevindingen.

4.1.2 Testmanagement

De functie testmanager heeft de volgende taken:

- Opstellen, verkrijgen van goedkeuring en onderhouden van het testplan.
- Uitvoeren van het testplan binnen planning en budget.
- Rapporteren over de voortgang van het testproces en de kwaliteit van het testobject.

4.2 Functie invulling

Functie	Uitvoerende persoon
Tester	K. van Hagen
Testmanagement	K. van hagen

5 Infrastructuur

5.1 Testomgeving

Om de tests uit te kunnen voeren is de onderstaande omgeving noodzakelijk:

- Hardware
 - Een laptop met de volgende minimale systeemeisen:
 - Processorarchitectuur: x64
 - Geheugen: >= 2 GB
 - Harde schijf: >= 40 GB
- Software
 - 64bit Windows (vista/7/2008/2008 R2)
 - Microsoft Office 2007 of nieuwer
 - Microsoft SharePoint 2010
 - Microsoft SQL Server 2008 R2
 - Microsoft SQL Server Management Studio 2008 R2
 - Microsoft Sync Framework 2.0 redistributable
 - Docstream 3
 - Een werkende Microsoft BPOS account.

5.2 Testtools

De testtools die voor het testen gebruikt zullen worden zijn:

- Microsoft SQL Server Management Studio 2008 R2:
 - Wordt gebruikt om bestanden te herstellen bij aanvang van tests.
- Microsoft SharePoint 2010:
 - Wordt gebruikt bij het vullen van bestanden tijdens het uitvoeren van tests.

5.3 Kantoorinrichting

Het uitvoeren van de tests wordt gedaan op een van de flexplekken van het kantoor.

6 Beheer

Gezien de omvang van het testproject wordt er alleen beheer uitgevoerd op de testbasis en de testdocumentatie

7 Planning

Activiteiten	Start	Eind
Opstellen testgevallen	WK 12	WK 15
Testen	WK 12	WK 16

8 Risicos

Het uitvoeren van de systeemtests is onderhevig aan de volgende risico's:

Software is slecht testbaar: Indien de software slecht te testen is kan dit gevolgen hebben voor de kwaliteit van de testen en het uitvoeren van het testen volgens de planning.

Afhankelijkheden van de software zijn niet stabiel: Indien het testobject afhankelijk is van software buiten zijn invloedssfeer vormt dit een bedreiging voor de validiteit van het testproces.

Problemen met de infrastructuur maken het testen onmogelijk: Problemen met de infrastructuur kunnen verhinderen dat het testproces uitgevoerd kan worden.

De tester is tevens ontwikkelaar van het product: Dit vormt het risico dat het uitvoeren van testen wordt vervuild door voorkennis van het testobject.

