

Conceptontwerp van een

Flash Content Systeem voor Op Flash gebaseerde websites.



Door: Clenn ten Hove
Student Mediatechnologie
ID: 1182500
TROS Internet & Nieuwe Media 2007

1. Voorwoord

Het onderwerp van deze afstudeerscriptie is tot stand gekomen door mijn behoefte aan een Flash Systeem waarmee Flash sites online kunnen worden aangepast, beheerd en geüpdate. Ik heb geruime tijd rond gelopen met de vraag naar een dergelijk systeem. In mijn zoektocht naar een afstudeer stage kwam ik snel bij de TROS uit, waarna ik dit onderwerp voorstelde aan mijn huidige begeleider. Gelukkig was de reactie positief en ben ik eind januari 2007 begonnen met afstuderen bij de TROS om zo te werken aan dit project.

Ik heb als project naam en tevens als naam van het systeem gekozen voor 'FlashLive'. De naam geeft aan dat het over Flash gaat en dat er Live veranderingen in Flash plaats kunnen vinden. Het live aanpassen van Flash bestanden kan gezien worden als de voornaamste functie van een content systeem voor Flash sites.

Er wordt in deze scriptie ervan uitgegaan dat de lezer mediatechnologische basiskennis bezit en bekend is met begrippen als Flash, PHP XML en enkele Actionscript principes kent.

Graag wil ik de volgende mensen bedanken voor het mogelijk maken van mijn stage en afstudeer opdracht:

- Marcel Douwstra – stagebegeleider en senior developer TROS
- Ronald Kools – senior developer TROS
- Niels Berkens – technisch project leider TROS
- Steven van der Linden & Eelco Tienstra – stage docenten
- Alle collega's van de afdeling Internet & nieuwe Media
- Iedereen van Sterren.nl
- Valérie Sankatsing
- HU - utrecht

2. Inhoudsopgave

1.	Voorwoord	2
2.	Inhoudsopgave	3
3.	Samenvatting	5
4.	Inleiding	6
5.	Probleemstelling	7
5.1.	<i>Opdrachtomschrijving</i>	7
5.2.	<i>Product</i>	7
5.3.	<i>Opdracht</i>	7
6.	Orientatie	8
6.1.	<i>De meerwaarde van een Flash content systeem</i>	8
6.2.	<i>Bruikbaarheid & meerwaarde voor TROS</i>	8
6.3.	<i>Doelgroep</i>	8
6.4.	<i>FlashLive Mogelijkheden</i>	9
7.	Onderzoek Flash Functionaliteiten en Toepassingen	10
7.1.	<i>Technieken:</i>	10
7.1.1.	Ontwikkel omgevingen	10
7.1.1.1.	Flash IDE	10
7.1.1.2.	Flex 2.0.	10
7.1.1.3.	FAMES	11
7.1.1.4.	AMES	12
7.1.1.5.	Eclipse	12
7.1.1.6.	FlashDevelop	12
7.1.1.7.	Sejpy	12
7.1.1.8.	Weborb for PHP	12
7.1.2.	Compilers	13
7.1.2.1.	Ming	13
7.1.2.2.	MTASC	13
7.1.2.3.	SwfMill	14
7.1.3.	Actionscript toepassingen	15
7.1.3.1.	AsWing	15
7.1.3.2.	Xleff	15
7.1.4.	Flash uitwissel technieken	17
7.1.4.1.	Flash Remoting	17
7.1.4.2.	AMFPHP	19
7.1.4.3.	XML	20
7.1.4.4.	Name/value pairs	21
7.1.4.5.	PHPObject	22
7.1.4.6.	WDDX	22
7.1.4.7.	SWX	23
7.2.	<i>Bestaande CMS sytemen</i>	23
7.2.1.	Flashblocks	23
7.2.2.	Fcmspro	23
7.3.	<i>Socket servers</i>	24
7.4.	<i>Media servers</i>	24
7.4.1.	Flash Media Server2	24
7.4.2.	RED5	25
7.4.3.	Haxevideo	26
7.5.	<i>Toekomstige Technieken</i>	26
7.5.1.	ActionScript 3.	26
7.5.2.	Silverlight	27
8.	Veiligheid	28
8.1.	<i>Flash Player Beveiliging</i>	28
8.2.	<i>f_TEA</i>	28

9.	Snelheid en belasting	29
9.1.	<i>Niet binaire data</i>	29
9.2.	<i>Binaire data.....</i>	29
10.	Resultaat Analyse.....	30
10.1.	<i>Gemaakte technische keuzes en methodes.....</i>	30
10.1.1.	Ontwikkelomgeving	30
10.1.2.	Alternatieve Compilers / tools	31
10.1.3.	Actionscript.....	31
10.1.4.	Uitwissel techniek	32
10.2.	<i>Analyse Schaalbaarheid.....</i>	32
10.3.	<i>Analyse Beveiliging.....</i>	33
10.4.	<i>Analyse Snelheid.....</i>	33
11.	Conceptontwerp	34
11.1.	<i>Inleiding</i>	34
11.2.	<i>Structuur</i>	34
11.3.	<i>FlashLive Opbouw.....</i>	35
11.4.	<i>Functioneel Ontwerp.....</i>	36
11.4.1.	FlashLive Structuur	36
11.4.2.	FlashLive Onderdelen	37
11.4.2.1.	Site	38
11.4.2.2.	Pages.....	39
11.4.2.3.	Layout	40
11.4.2.4.	Content.....	42
11.4.3.	Logoff	43
11.5.	<i>Technisch Ontwerp</i>	44
11.5.1.	Dataflow FlashLive.....	44
11.5.1.1.	Main(frontEnd) - Technische beschrijving	44
11.5.1.2.	Pages - Technische beschrijving	46
11.5.1.3.	LayOut - Technische beschrijving	48
11.5.1.4.	Content – Technische beschrijving.....	52
11.5.1.5.	Site – Technische beschrijving	55
11.5.2.	Componenten	57
12.	Ontwikkeling	59
13.	Conclusies en Aanbevelingen	60
13.1.	<i>Conclusie Probleemstelling.....</i>	60
13.2.	<i>Bruikbare mogelijkheden voor de Tros</i>	60
14.	Evaluatie	61
15.	Bronnen	62
16.	Bijlagen.....	63
16.1.	<i>Bijlage - Plan van aanpak</i>	63
16.1.1.	Activiteiten en Fasering	63
16.1.2.	Tijdsplanning	64
16.1.3.	Schema.....	64
16.2.	<i>Bijlage – Ming Objecten</i>	65
16.3.	<i>Bijlage – Ming Installatie en voorbeeld Code</i>	67
16.4.	<i>Bijlage – Flashvar voorbeeld.....</i>	68
16.5.	<i>Bijlage – Flashvars laden uit database met een server side script ...</i>	69
16.6.	<i>Bijlage – RED 5 Actionsript code</i>	70
16.7.	<i>Bijlage - RTMP</i>	71
16.8.	<i>Bijlage – Voorbeeld Crossdomain.xml.....</i>	72
16.9.	<i>Bijlage – FlashLive pagina beschrijving voor gebruik met SWFMill</i>	73

3. Samenvatting

In deze scriptie wordt een concept ontwikkeld voor een Flash content systeem voor op Flash gebaseerde websites, genaamd FlashLive. Er wordt een probleemstelling gedefinieerd die door middel van onderzoek en het uit werken van een concept, in de conclusie beantwoord zal worden. Na het definiëren van de probleemstelling komt de oriëntatiefase aan bod, hier wordt vooruit gekeken naar het FlashLive systeem en wordt een doelgroep aangegeven. Het hierop volgende onderzoek naar verschillende Flash technieken en functionaliteiten zal een groot deel van de scriptie in beslag nemen. In de resultaat analyse worden de bevindingen van het onderzoek geanalyseerd waarna met de gemaakte keuzes een conceptontwerp is uitgedacht. Dit ontwerp bestaat uit een functioneel ontwerp, waarin het FlashLive systeem wordt beschreven en per onderdeel wordt uitgelegd, en het technisch ontwerp, waarin processen worden uit gelegd en beschreven aan de hand van tekst en dataflow diagrammen. De realisatie van het concept valt buiten deze scriptie. De conclusie wordt dan ook gebaseerd op het gedane onderzoek en het ontwikkelde concept met als doel de probleemstelling te beantwoorden.

4. Inleiding

Tegenwoordig is het gebruik van Flash voor websites bijna niet meer weg te denken. Ook de TROS maakt hier veelvuldig gebruik van.

Toch bestaan er op dit moment weinig tot geen Flash Content Systemen voor op Flash gebaseerde websites. Gezien het feit dat het gebruik van Flash de laatste jaren sterk is toegenomen, is het ontwikkelen en bedenken van nieuwe concepten en technieken, gebaseerd op Flash, een logisch gevolg. In deze scriptie zullen verschillende van deze technieken en concepten worden onderzocht. Ook zal er worden getracht een concept te ontwikkelen waarmee op Flash gebaseerde websites online kunnen worden gecreëerd en onderhouden.

5. Probleemstelling

Zijn op Flash gebaseerde websites en onderdelen via een Flash content systeem te beheren? En kan dit even snel en makkelijk werken als het beheren van reguliere websites?

5.1. Opdrachtomschrijving

De opdrachtomschrijving voor dit project luidt: het onderzoek doen naar de mogelijkheden om een Flash content systeem te ontwikkelen en het opstellen van een conceptontwerp voor dit systeem.

5.2. Product

Een Flash Content Systeem werkend met XML voor het beheren, aanpassen en updaten van op Flash gebaseerde websites.

Het systeem zal worden geïmplementeerd met nieuwe Flash technieken en er zal PHP server side scripting worden gebruikt. Dit systeem zal worden ontwikkeld met in gedachten uitbreiding naar andere omgevingen, als Typo3.

5.3. Opdracht

Onderzoek doen door middel van eindgebruikersfeedback welke mogelijkheden, onderdelen en componenten het Flash Content Systeem moet bezitten.

Achtereenvolgens zullen ook, doormiddel van het opzetten en uitvoeren van testen, de veiligheid, schaalbaarheid, meerwaarde, en de verschillende toepassingen van het Flash content systeem voor de TROS worden onderzocht.

Aan de hand van onderzoeksresultaten zal een conceptontwerp voor het Flash content Systeem worden gemaakt. Dit kan worden gebruikt voor toekomstige TROS sites, site-onderdelen en projecten.

Het uitdagende aan deze opdracht is dat er nog weinig systemen bestaan waarbij Flashsites kunnen worden beheerd, aangepast en geupdate. Tevens is het als mediatechnoloog zijnde interessant om te leren over flash, xml en de nieuwe mogelijkheden die hiermee gepaard gaan. Deze gegevens leveren de motivatie om hier meer over te willen weten.

6. Orientatie

6.1. *De meerwaarde van een Flash content systeem*

Een Flash content systeem biedt in het algemeen verschillende voordelen. Door het grafische karakter van flash is het mogelijk interactieve en duidelijke interfaces te maken.

Een Flash Content systeem kan gebruik maken van deze grafische voordelen en zo voor een grotere doelgroep aantrekkelijk zijn. Voor mensen die geen ervaring hebben met een content systeem lijkt het logischerwijs gemakkelijker te zijn met een Flash content systeem te werken, dan met andere content systemen. Dit is te danken aan het grafische en visuele karakter van Flash.

Flash draait clientside en kan daardoor de hoeveelheid server belasting verkleinen. Een Flash content systeem kan data preloaden waardoor er minder laad acties uitgevoerd moeten worden.

6.2. *Bruikbaarheid & meerwaarde voor TROS*

De TROS maakt veel gebruik van Flash in hun sites, maar heeft nog geen volledig flash gestuurde websites. Enkele jaren geleden is er een experiment geweest met een volledig op Flash gebaseerde site, genaamd cyberchannel.nl (is niet meer actief). Tot op heden word er alleen gebruik gemaakt van Flash voor in menu's en banners.

Na ondervraging is gebleken dat de TROS toekomst ziet in flash, en geïnteresseerd is in de mogelijkheden. Met de komst van nieuwe technieken wordt flash voor de TROS steeds interessanter. Hier speelt deze scriptie op in door onderzoek te doen naar nieuwe technieken met als resultaat het uitdenken van een Flash content systeem, met de projectnaam FlashLive. Dit project kan later dienst doen als voorbeeld systeem voor de TROS.

6.3. *Doelgroep*

Om goed in te spelen op het karakter van dit project is een doelgroep gedefinieerd.

De doelgroep varieert van TROS werknemers, die geen ervaring hebben met het maken van op Flash gebaseerde websites, tot TROS flash ontwikkelaars.

6.4. FlashLive Mogelijkheden

Na verscheidene gesprekken met internet redactieleden, webontwikkelaars en designers, waarvan sommigen werkzaam bij de TROS en anderen niet, is de volgende lijst tot stand gekomen. Deze lijst geeft de eisen weer waaraan een dergelijk Flash content systeem zou moeten voldoen.

- Direct updaten van content op een live Flash website
- Zoekmachines indexeren
- Zoek mogelijkheden in flash
- Template systeem (css of xml)
- Verwijderen, toevoegen en creëren van pagina's en Flash bestanden
- MySQL database ondersteuning
- Een Flash text editor
- Bewerkbare content blokken
- Een fileadmin
- Verborgten login functie
- Rechten
- Veiligheid (secure xml en flash encryption)
- Creëren van rss feeds
- Tween effecten
- Xml menu's
- Site overzicht / site diagram
- (live)Audio/Video stream mogelijkheden

7. Onderzoek Flash Functionaliteiten en Toepassingen

Om een goed beeld te krijgen van de hedendaagse Flash mogelijkheden is er een onderzoek gedaan naar verschillende technieken en methodes. Dit onderzoek is bedoeld om inzicht te krijgen voor het ontwikkelen van het FlashLive systeem concept. Er zijn literaire bronnen gebruikt en is er op internet gezocht naar veel verschillende open source flash projecten (zie hoofdstuk 6.2). Alle onderzochte technieken zijn geïnstalleerd en getest om een goed beeld te krijgen. In dit hoofdstuk wordt er bij elke onderzochte techniek en methode, een uitleg gegeven.

7.1. Technieken:

7.1.1. Ontwikkel omgevingen

7.1.1.1. *Flash IDE*

Met het begrip Flash kunnen twee dingen worden omschreven: De Adobe Flash Player en de Adobe Flash Professional programmeeromgeving, een IDE genoemd. De Flash Player is een 'virtual machine' (vm) die Flash bestanden , SWF files genaamd, verwerkt en uitvoert. Flash Professional is de standaard programmeeromgeving die Adobe levert om Flash applicaties te ontwikkelen.

7.1.1.2. *Flex 2.0*

Flex 2.0 is een ontwikkelomgeving van Adobe, bestaande uit verschillende applicaties om RIAs, Rich Internet Applications, te kunnen maken. Deze RIAs worden gemaakt in Flash.

Een Flex applicatie wordt ontwikkeld in MXML en actionscript 3.0. MXML wordt ook wel Flex Markup Language of Multimedia eXtensible Markup Language genoemd. Dit is een op XML gebaseerde taal van Adobe, om interface elementen te beschrijven. Actionscript 3.0, de opvolger van Actionscript 2.0, wordt gebruikt om de procedurele code voor de RIAs te schrijven.

De Flex 2.0 ontwikkelomgeving bestaat uit 4 producten:

Flex Builder 2

Flex Builder is het voornaamste ontwikkelplatform van de Flex 2.0 ontwikkelomgeving. Door gebruik te maken van de Eclipse IDE (voor uitleg zie 7.1.1.5) en Flex classes, beschikt een ontwikkelaar of designer over verschillende layout en interface tools.

Flex Data Services 2

Met behulp van Flex Data Services binnen de Flex 2.0 ontwikkelomgeving, kunnen krachtige applicaties gemaakt worden. Deze applicaties zijn bijvoorbeeld in staat grote hoeveelheden data te verwerken. Flex Data services bestaan uit 4 onderdelen:

- **Web Tier Flex Compiler:**
een MXML Compiler
- **RCP Services:**
voor uitwisseling van objecten tussen verschillende web services
- **Messaging:**
Events versturen tussen clients en services via RTMP (relatime messaging protocol) *(voor uitleg zie bijlage 14.7)*
- **Data Managment:**
Object data editing en synchronisatie met multi-user mogelijkheden.

Flex SDK

De Flex Software Development Kit, een SDK genoemd, verzorgt technieken en compilers. Hiermee kunnen RIA's in flash gemaakt worden, met een ontwikkelomgeving naar keuze.

Kortgeleden heeft Adobe bekend gemaakt dat de Flex SDK een open source project zal worden. Dit zal uitgegeven worden onder de Mozilla Public License. Het gevolg hiervan is dat niet alleen de Actionscript componenten van de Flex SDK als source code beschikbaar worden, maar ook de Java broncode voor de ActionScript Virtual Machine, de MXML compilers, de Scionscript Debugger en de core Actionscript Libraries.

Flex Charting 2

Flex Charting is een programeer library met interactieve grafieken en diagrammen, die gebruikt kunnen worden in de Flex 2 SDK.

7.1.1.3. FAMES

FAMES bestaat net als Flex uit een set tools om, zonder de standaard Flash IDE, flashmovies te ontwikkelen. FAMES staat voor **F**lashout+**A**STD+**M**TASC +**E**clipse IDE+**S**wfMill. Hieronder volgt een korte uitleg van de verschillende programma's.

- *Eclipse*: een universeel ontwikkelplatform met uitbreidingsmogelijkheden *(voor uitleg zie 7.1.1.5)*
- *Flashout*: een Eclipse plugin die ervoor zorgt dat men gecompileerde flashmovies binnen Eclipse kan draaien.
- *ASTD*: een Eclipse plugin die een editor incl color highlighting en syntax checking levert waarmee je Actionscript 2.0 code kan creëren.
- *MTASC*: Motion-Twin ActionScript 2 Compiler om ActionScript 2.0 code te compileren. *(voor uitleg zie 7.1.2.2)*

- *SwfMill*: een XML naar SWF en SWF naar XML processor om een lege Flash movies met alleen een stage te maken. (voor uitleg zie 7.1.2.3)

Aangezien Flashout niet helemaal opensource is, wordt aangeraden om de AMES tool set te gebruiken.

7.1.1.4. AMES

Een volledige open source toolset die bestaat uit de volgende programma's:

- ASDT
- Apache Ant: Een Java compiler die met behulp van XML, projecten automatiseert
- MTASC
- Eclipse IDE

7.1.1.5. Eclipse

Eclipse is een universeel open source ontwikkelplatform, dat erg veel uitbreidingsmogelijkheden bezit. Deze mogelijkheden variëren van het bieden van een ontwikkelomgeving voor verschillende talen tot het maken van een Rich Client omgeving voor RIAs. Dit is tevens het geval bij de Flex ontwikkelomgeving.

7.1.1.6. FlashDevelop

FlashDevelop is een open source Actionscript2.0 editor en integreert technieken, als SWFMill en MTASC, om flash applicaties te generen. Voordeel van FlashDevelop is dat het als één van de eersten open source ontwikkelomgevingen ActionScript 3.0 en MXML 'code completion' en 'syntax checking' heeft. Ook kan er gebruik worden gemaakt van de Flex2 compiler, die wordt meegeleverd bij de Flash SDK. Een groot nadeel van gebruik van FlashDevelop bij de TROS, is dat er alleen Pc versies bestaan en geen versies voor de Macintosh.

7.1.1.7. Se|py

Se|py is evenals FlashDevelop een gratis open source Actionscript Editor en maakt ook gebruik van de MTASC compiler.

Ook Se|py bevat vele handige functies als "code completion" en "syntax checking". Een groot voordeel ten opzichte van FlashDevelop is dat Se|py ook beschikbaar is voor de Macintosh, hierdoor is het breder inzetbaar bij de TROS.

7.1.1.8. Weborb for PHP

Weborb bestaat uit een ontwikkelplatform om RIA's te maken in Flash of Ajax. Met aan de server kant .net of java en sinds kort ook Ruby on Rails en PHP. Deze laatste twee zijn open source en gratis. Ze missen echter mogelijkheden als 'Flex Data Management' en 'Flex Messaging' die de commerciële .net en java varianten wel bezitten.

7.1.2. Compilers

7.1.2.1. Ming

Ming is een c library om Flash movies te maken. Bij Ming worden een aantal wrappers geleverd om de library te kunnen gebruiken in c++ en scripttalen als PHP, Python, Ruby en Perl.

Aangezien er bij de TROS vooral met PHP gewerkt wordt, is het gebruik van de Ming library met PHP voor dit onderzoek de meest logische keuze.

Na de installatie voegt Ming 13 nieuwe objecten toe aan PHP, inclusief attributen en methodes, om Flash movies te genereren.

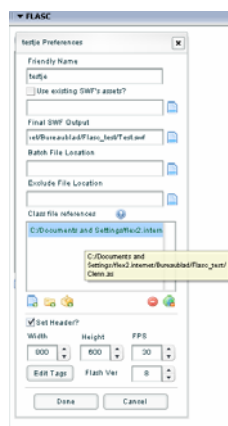
De meegeleverde bijlage bevat een uitleg van de 13 beschikbare objecten en een ming voorbeeld code (zie bijlage 16.2 en 16.3).

7.1.2.2. MTASC

MTASC staat voor 'Motion-Twin ActionScript 2 Compiler'.

MTASC is een 'commandline' compiler, die kan worden gebruikt in verschillende editors als compiler voor .as (Actionscript class) bestanden. MTASC kan vele .as bestanden in een korte tijd compileren en direct de bijhorende SWF bytecode genereren. Dit vindt plaats zonder tussenkomst van de Adobe Flash IDE of andere programma's. MTASC heeft als voordeel zeer snel te zijn.

In de situatie waarin een class A code gebruikt uit een class B, is het alleen nodig class A te compileren. MTASC ziet namelijk automatisch dat er code uit class B gebruikt moet worden. Als men een aantal classes standaard wil mee laten compileren kan er voor worden gekozen om een main class te maken waarin alle nodige classes worden geïmporteerd. Deze class kan als een startpunt van de applicatie werken. Er is tevens een MTASC versie die speciaal binnen de Adobe Flash IDE kan worden gebruikt, genaamd FIASC. Deze is als extensie te installeren.



Flasc – MTASC extensie voor de Adobe Flash IDE

7.1.2.3. SwfMill

Zoals nu gebleken is, zijn verschillende programma in staat om Flash bestanden te genereren. Om ook losse onderdelen als fonts, plaatjes en componenten toe te kunnen voegen en te compileren in een nieuw Flash bestand, is SWFMill bedacht.

SWFMill is een programma dat instaat is om binaire Flash bestanden en SWFML in elkaar om te zetten. SWFMill is een XML dialect gebaseerd op SWF, een Flash movie bestand. Aangezien SWFML grotendeels op het officiële SWF formaat van Adobe is gebaseerd, is SWFMill in staat om illegale SWF bestanden te creëren. Dit kan enkele risico's met zich meebrengen voor de werking van de applicatie en client, wanneer er foute SWFML code wordt omgezet naar SWF.

SWFMill kan vandaag de dag de volgende onderdelen in een SWF bestand compileren:

- SWF bestanden
- JPG bestanden
- PNG bestanden (24 en 32 bit)
- TrueType lettertypes

Tevens kan SwfMill onderdelen op een stage plaatsen en bepalen waar deze worden toegevoegd en in welk frame. Ook kan SWFMill movieclips maken die vanuit de code kunnen worden aangeroepen.



7.1.3. Actionscript toepassingen

7.1.3.1. AsWing

AsWing is een Open Source Flash ActionScript Grapical User Interface (GUI) framework, die ervoor zorgt dat programmeurs gemakkelijk een user interface kunnen maken. In het gebruik lijkt asWing op Java Swing. Het levert een set GUI componenten, welke men in object georiënteerde ActionScript 2 projecten kan gebruiken.

7.1.3.2. Xleff

Steeds vaker is te zien dat XML wordt gebruikt voor grafische interfaces. De XLEFF layout engine is speciaal ontwikkeld voor flash, en om deze reden voor dit project interessant om te onderzoeken. Met Xleff kan men layout gegevens extern opslaan in een XML opmaak, en op deze manier de layout van Flash movies aanpassen. Dit is tevens één van de meest belangrijke eisen waaraan het FlashLive systeem moet voldoen.

XLEFF staat voor XML Layout Engine For Flash, hieruit blijkt wat de functie van XLEFF is: het toevoegen van een layout manager aan een Flash movie. Een beter begrip van XLEFF is benodigd om te onderzoeken of XLEFF te gebruiken is voor het FlashLive systeem.

XLEFF laat het toe een component-based user interface creëren vanuit een XML beschrijving.

Wanneer er een user interface moet worden gecreëerd, met één instantie van het button component op de posities x: 100 en y: 89, dan zijn er twee manieren waarop dit kan worden gedaan:

- de button instantie kan gecreëerd worden door in de flash IDE een button component op stage naar de juiste positie te slepen
- de button instantie kan gecreëerd worden door Actionscript

Er is echter nog een derde manier, door gebruik van de XML Layout Engine, om dezelfde button op stage te krijgen:

```
<button name="myButton" x="100" y="89" />
```

Het is bij dit project van belang naast de mogelijkheid de content van de flash site online te beheren en aan te passen, ook de mogelijkheid te bieden de site layout (gebruikers interface) online te veranderen.

De eerste manier waarop handmatig een button gecreëerd kan worden, door deze op de stage te slepen, is te statisch. De interface kan na het compileren niet meer worden aangepast. Bij de tweede manier wordt de button geplaatst via Actionscript. Dit geeft enkele dynamische mogelijkheden. Zo kan aan de hand van voorwaardes een specifieke button positie worden gegenereerd. Dit loopt

echter nog steeds volgens een stramien en is niet volledig dynamisch. Tevens zorgt het creëren van een interface door Actionscript voor veel code, die des te complexer de site of applicatie wordt, des te moeilijker te begrijpen is.

Bij de derde manier wordt de button geplaatst via de XML layout. Dit geeft de mogelijkheid om een volledig dynamische site interface op te bouwen. Tevens zorgt de XML notatie voor een duidelijk beeld van de interface en voorkomt het op deze manier veel complexe Actionscript code. Dit zorgt ervoor dat ontwikkelaars zich beter op de Actionscript code kunnen focussen en minder rekening hoeven te houden met de interface. Men kan het gebruik van de XML layout engine zien als het scheiden van php en html code met het gebruik van templates.

Enkele voordelen van een XML Layout Engine zijn:

- Interfaces met veel genestelde componenten worden overzichtelijker
- De interface is makelijker bij te houden en uit te breiden
- Interfaces kunnen sneller gemaakt worden
- XML's herbruiken
- Een effectieve en nette scheiding van interface en code

De vraag blijft waarom het gunstig is met een XML beschrijving de flash layout te creëren. Het antwoord bestaat voor een deel uit het feit, dat een gescheiden gebruikers interface van de programma code ervoor zorgt dat het programma modulair kan worden opgebouwd. Ook kan de interface op gebruikers rechten worden aangepast en is het geheel beter beheerbaar. Wanneer dit concept goed geïmplementeerd wordt kan dit vele voordelen bieden.

7.1.4. Flash uitwissel technieken

Steeds vaker worden Flash-applicaties ontwikkeld als deel van een uitgebreide applicatie. De gegevens van de applicatie zijn vaak extern opgeslagen in tekstbestanden, databases of XML bestanden. Er zijn verschillende manieren om deze gegevens tussen een Flash-applicatie en een niet flash applicatie uit te wisselen. Dit kan op een gemakkelijke manier via *XML* of *name/value pairs*, maar beter via *Flash Remoting* of *phpObject*. Ook bestaan er nieuwere technieken en concepten als *WDDX* en *SWX*. Al deze technieken zullen in het volgende hoofdstuk worden toegelicht. In het hoofdstuk 'Resultaat analyse' (*hoofdstuk 8*) zullen de voor en nadelen van de verschillende technieken besproken worden en hieruit zal een conclusie voortvloeien.

7.1.4.1. Flash Remoting

Flash remoting is ontstaan als een Macromedia client/server framework bestaande uit een gateway, componenten, Actionscript classes en een debugger. Dit zorgt voor een snelle, gemakkelijke en betrouwbare data uitwisseling tussen een flash client en een server. Aan de Flash kant zijn er enkele Actionscript classes die het encoden en decoden van data voor hun rekening nemen. En aan de server kant is er een gateway die ook data encodeert en decodeert, en werkt als een soort regelaar om de juiste services aan te kunnen roepen. Flash remoting maakt voor het uitwisselen van data gebruik van het binaire AMF formaat (ActionScript Message Format), geïnspireerd op Soap. Het AMF formaat heeft twee versies. De initiële Flash Remoting versie AMF0 en de nieuwe Flex 2.0 Data services versie, die gebruik maakt van AMF3. Deze laatste zorgt onder anderen voor snelheidsverbeteringen. De gateway en de Actionscript classes vertalen data naar dit neutrale formaat, en zorgt ervoor dat data transparant kan worden uitgewisseld. Omdat AMF een open formaat betreft en het simpel is opgebouwd, zijn er, nadat macromedia het AMF0 formaat uitbracht, veel verschillende ontwikkelaars begonnen met producten hiervoor te ontwikkelen. Eén van de eerste, en tevens basis voor vele anderen, is ontwikkeld door Wolfgang Hamann. Deze zorgde niet alleen voor de meest bekende en meest gebruikte AMF gateway, genaamd AMFPHP, maar opende hiermee ook de weg voor anderen. Hierdoor is flash remoting in bijna elke programmeertaal gratis te gebruiken. Verderop in dit hoofdstuk wordt ingegaan op AMFPHP en enkele andere Remoting services.

Waarom Remoting?

Als men data wilt uitwisselen tussen een Flash client en een server is een veel gebruikte methode XML. Deze is te gebruiken als uitwissel formaat. Actionscript heeft een XML parser, die XML bestanden direct van de server kan inladen en de data uit de XML kan lezen. Maar wat te doen als men data uit een database wilt

halen? Men kan dan een PHP script maken die, verbinding maakt met een database, een query vormt, alle data uitleest, een XML string van de data maakt, en uiteindelijk de XML string naar flash terugstuurt. In Flash word de XML weer ingeladen en uitgelezen, waarna flash beschikking heeft over de data.

Deze manier van data uitwisseling is natuurlijk iets wat omslachtig omdat de data, afkomstig van de database, eerst in XML formaat wordt omgezet. Hierna moet de XML data in Flash worden geladen en nogmaals worden omgezet (geparseerd), om uiteindelijk gebruikt te kunnen worden.

Hoe werkt een AMF gateway?

Technisch gezien werkt een AMF gateway in drie stappen. Ten eerste wordt de AMF data ontvangen door de gateway en gedecodeerd. Als tweede worden de juiste methodes uitgevoerd, afhankelijk van de ontvangen AMF data. Als derde en laatste stap wordt de verwerkte data weer gecodeerd en terug naar de flash movie gestuurd.

Decoderen

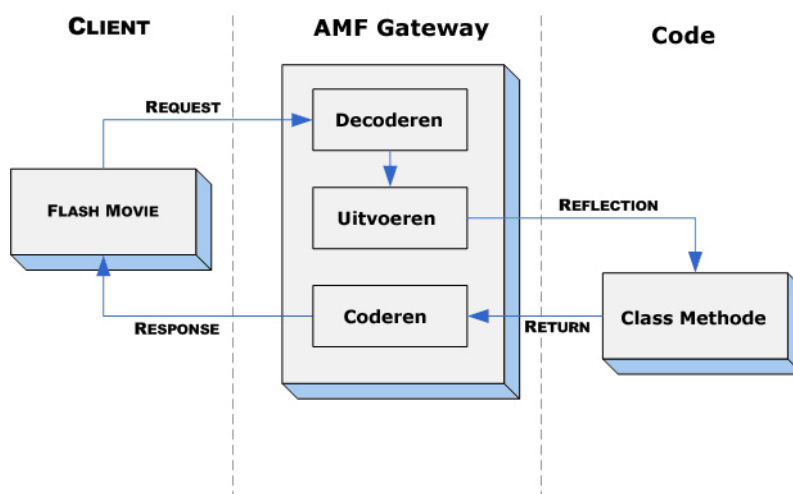
Decoderen bestaat uit het uitlezen van het binaire AMF formaat uit de 'Response.InputStream' en het omzetten in de juiste data. Een groot gedeelte bestaat uit het mappen van flash data types naar bijvoorbeeld PHP data types.

Uitvoeren

Tijdens het uitvoeren wordt de gedecodeerde data uit de eerste stap gebruikt om de gevraagde methode via het reflectie principe uit te voeren. Elke uitzondering tijdens het uitvoeren wordt, bij wijze van reflectie, terug naar flash gestuurd. Deze is in de 'onStatus handler' binnen flash op te vragen.

Coderen

De laatste coderingsstap gebruikt de uitkomst van de uitgevoerde methode en verpakt dit terug in een 'AMF response' formaat. Dit pakket wordt weggeschreven in de 'OutputStream'.



AMF -flowdiagram

Verschillende Remoting services/gateways

Er zijn verschillende Remoting services voor flash te vinden in veel verschillende variaties. Om niet te veel uit te wijken en omdat men bij de TROS hoofdzakelijk PHP gebruikt worden alleen de open source PHP implementaties van Flash Remoting behandeld.

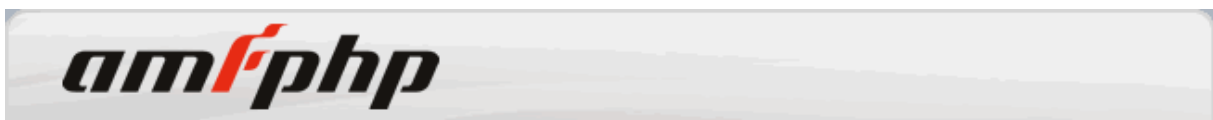
7.1.4.2. AMFPHP

AMFPHP is waarschijnlijk de meest bekende en gebruikte open source 'Flash Remoting' gateway. AMFPHP laat applicaties, door middel van 'Remote Procedure Calls', communiceren tussen clients en servers. Voordeel van AMFPHP ten opzichte van de vele andere Flash Remoting implementaties is dat AMFPHP geen specifieke objecten en structuren gebruikt aan de server kant. Objecten kunnen worden aangeroepen alsof ze lokaal beschikbaar zijn. Een tweede voordeel van AMFPHP is de service browser waar te zien is welke classes zijn geladen.

Om AMFPHP te kunnen gebruiken moet de MX Remoting Components worden geïnstalleerd. Deze zijn te downloaden van adobe.com. AMFPHP werkt met een zogenaamde methodeTable. Hierin wordt aangegeven welke functies (methodes) in een eigen gemaakte php class er voor remoting beschikbaar mogen zijn. Deze methodeTable moet in de 'constructor' functie de eigen gemaakte class worden gezet. Een voorbeeld hiervan is:

```
$this->methodTable = array{
    "mijnfunctie" => array{
        "description" => "dit is mijn functie die een string terug stuurd"
        "access" => "remote",
        "arguments" => array ("mijnArgument"),
        "returntype" => "String"
    }
}
```

Bovenstaande code, zorgt ervoor dat "mijnfunctie" beschikbaar is voor remoting. Dit komt doordat "access" is gezet op "remote".



7.1.4.3. XML

Eén van de tegenwoordig meest gebruikte technieken in combinatie met flash is XML. Een complete uitleg van het begrip XML valt buiten de content van deze scriptie. Het is algemeen bekend, onder ontwikkelaars, dat XML bestaat uit tags met informatie. Deze worden tezamen nodes genoemd. Deze nodes kunnen attributen bezitten die simpele waardes bevatten. Nodes, met of zonder attributen, vormen de basis van een XML opmaak.

Het is gebruikelijk om in een XML de mime-type, de DOCTYPE en xml versie te declareren. Wanneer XML met een andere applicatie zou moeten worden gedeeld, kunnen deze gegevens nodig zijn. In deze scriptie wordt echter alleen de relatie tussen flash en XML behandeld. Omdat Flash deze gegevens niet kan parsen zijn ze niet van belang. XML word vaak gebruikt met Flash om databases te vervangen als het om kleine hoeveelheden data gaat. Flash heeft ingebouwde XML ondersteuning via het XML object. Om data uit het XML object te halen moet door de XML Structuur heen worden gegaan met loop.

XPath

Om data uit een XML object iets makkelijker te verkrijgen bestaat XPath. Dit staat voor XML Path Language, en is een taal om XML waardes te extraheren. Sinds Flash MX 2004 op de markt kwam is XPath als API class te gebruiken. XPath kan vanuit de DataBinding componenten worden toegevoegd om zo via een XPath (een pad met slashes naar de juiste node) data te verkrijgen. XPath kent de volgende twee methodes: 'selectNodelist' en 'selectSingleNode'. Omdat de XPath api uit de DataBinding componenten geen volledige implementatie is, lijkt het beter te zijn om XPathStudio te gebruiken.

XPathStudio

Een alternatief voor de Xpath class van Adobe is XPathStudio, ook wel XPath4AS genoemd. XPathStudio heeft een meer complete XML implementatie dan de XPhath Class van Adobe.

7.1.4.4. Name/value pairs

Flash heeft de mogelijkheid om data als een grote 'String' in te laden. Deze data moet vervolgens in de vorm van een Name/value (naam / waarde) formaat worden gepresenteerd:

bedrijf=TROS

Om meer data in een string te kunnen stoppen kan met het '&' teken de data worden verlengd:

bedrijf=TROS&stagiair=Clenn

Met dit principe kan zowel uit een normaal tekst bestand als uit een Database data in Flash worden geladen. Dit geschiedt met de LoadVars class.

Als een tekstbestand wordt gebruikt kan deze, via LoadVars, direct in Flash worden geladen. (zie bijlage 14.4)



Wanneer met Flash iets uit een Database gehaald moet worden en de Name/Value techniek moet worden toegepast, is een server side script benodigd. Een voorbeeld hiervan is een PHP script dat voor een DB connectie zorgt en een query op de Database los laat. Hierna haalt Flash de variabelen op uit het PHP script (zie bijlage 14.5).



7.1.4.5. PHPObject

PHPObject is een opensource project dat als alternatief voor Flash Remoting kan worden gezien. Met PHPObject kan je PHP functies in een PHP class binnen flash aanroepen alsof het eigen functies zijn. Het is in Flash mogelijk om naast objecten als movieclips ook generieke objecten te maken. Een voorbeeld hiervan is:

```
var mijnScriptie = new Object;
```

Met PHPObject is het mogelijk om een PHPObject te maken binnen flash en deze aan een PHP Class, bijvoorbeeld scriptie.php, te koppelen.

```
var mijnScriptie = new PHPObject ("scriptie");
```

PHPObject verzorgt de client –server connectie en maakt het op deze manier mogelijk om object variabelen tussen flash en php uit te wisselen.

PHPObject zit relatief eenvoudiger in elkaar dan bijvoorbeeld AMFPHP. De installatie bestaat uit de PHPObject class voor Flash, die via de extensie manager is toe te voegen. Aan de server kant moet de 'gateway.php' en de 'config.php' neer worden gezet. In de 'condif.php' kan worden aangegeven in welke map de gateway moet zoeken naar gemaakte PHP classes. Ook kan een geheime key als beveiliging worden ingevoerd. Het enige wat nog rest is het correct aanroepen van de gateway binnen de Flash applicatie:

```
import.com.ghostwire.PHPObject.*
```

```
PHPObject.defaultGatewayKey = "geheim"
```

```
PHPObject.defaultGatewayUrl = http://localhost/<een_map>/gateway.php
```

Hierna kunnen, zoals hierboven beschreven, de PHP classes worden aanroepen als ware het flash classes.

7.1.4.6. WDDX

WDDX is een op XML gebaseerde technologie waarmee complexe data kan worden uitgewisseld tussen verschillende talen. De data wordt serieel omgezet naar het WDDX formaat, waarna het tussen bijvoorbeeld PHP en Flash kan worden verstuurd. Een WDDX parser kan, doordat in het WDDX formaat precies staat beschreven om wat voor data het gaat, de seriële data terugzetten in data, die gebruikt kan worden in de juiste omgeving. Flash heeft, anders dan PHP, van nature geen WDDX functies. Deze kunnen echter worden toegevoegd door gebruik te maken van de wddx_mx class, die in de Flash exchange op adobe.com beschikbaar is.

7.1.4.7. SWX

Een recente techniek om gegevens uit te wisselen met Flash is SWX. SWX staat voor SWF Data Exchange Format en gebruikt uitgekilde Flas bestanden om data uit te wisselen. SWX bestaat uit een gateway, die de requests controleert, en een script dat Flash bestanden genereert. Deze bestanden zijn uitgekilde Flash bestanden die uit één frame bestaan en alleen data bevatten. Een voordeel van dit systeem is dat, via een simpele 'loadMovie()' functie, data binnen Flash kan worden geladen. Deze methode van data uitwisseling is nog in ontwikkeling, maar biedt interessante mogelijkheden voor de toekomst.

7.2. Bestaande CMS sytemen

Het is voor het onderzoek van belang om naar al bestaande oplossingen te zoeken, om te gebruiken als referentie materiaal. Er zijn slechts twee applicaties die hier enigszins voor in aanspraak komen. Deze applicaties worden hieronder besproken.

7.2.1. Flashblocks

Flashblocks is een commercieel systeem om op een snelle manier Flash websites te creëren en content management mogelijkheden aan Flash websites toe te voegen. Flashblocks werkt met een methode waarbij men in de Flash IDE een website template maakt en daarop editable Flashblocks componenten sleept. Eenmaal gecompileerd kan de inhoud van deze editable componeten online veranderd worden. Deze inhoud word door het 'include.php' bestand weggeschreven op server. Dit systeem heeft enkele nadelen. Eenmaal gecompileerd zijn het design en de layout namelijk niet meer aan te passen. Men heeft altijd de Flash IDE nodig om templates te veranderen. Verder is het nodig om bij elke kleine interface verandering of toevoeging alles op nieuw te compileren. Data word via een PHP script weggeschreven naar een bestand, waardoor dit systeem niet te gebruiken is om complexere systemen met veel data te maken.

7.2.2. Fcmspro

Fcmspro staat voor Flash Cms Professional en is tevens een commercieel systeem wat content managment mogelijkheden voor Flash biedt. Fcmspro heeft vele interessante mogelijkheden. De demo op de site laat echter alleen het editen van tekstbestanden toe. Uit de feature lijst op de site komt niet naar voren of dit systeem in staat is online de interface aan te passen. Dit wordt overigens wel in een grote reclame banner op de site beweerd. Helaas is er geen trial van dit product te downloaden om het systeem nader te kunnen bekijken.

Wel wordt duidelijk dat dit systeem gebruik maakt van een database en dus complexere websites kan beheren.

7.3. **Socket servers**

Socket servers voor Flash vallen eigenlijk buiten de content van dit project, aangezien er geen technieken als een socket servers benodigd zijn om een Flash content systeem te kunnen ontwikkelen.

Socket servers worden interessant wanneer men games en chat applicaties wil creëren. Er kan in de toekomst vraag naar het tegelijkertijd online samenwerken, binnen het FlashLive systeem, ontstaan. Om deze samenwerking te realiseren is een Socket Server benodigd, die een realtime verbinding met de clients behoudt. Om deze reden worden enkele Socket servers genoemd die in aanmerking kunnen komen om gebruikt te worden in het FlashLive systeem.

Een eerste voorbeeld is de **SmartFoxServer**, dit is een multi-omgeving socket server gemaakt om in Flash multi-user applicaties te ontwikkelen. Ten tweede is er de **ElectroServer**, ontwikkeld door electrotank, een high preformance server die duizenden gebruikers tegelijk aankan. Als laatste is er de **Unity 2 socket Server** die leverbaar is met een complete multi-user ontwikkelomgeving

7.4. **Media servers**

Hoewel het voor de hoofdzaak van dit project niet nodig is dat FlashLive media stream mogelijkheden bezit, is het voor de volledigheid toch verstandig om bij de twee belangrijkste servers voor het streamen van media stil te staan. Het kan immers in de toekomst voorkomen dat het FlashLive systeem media streams moet kunnen laden.

7.4.1. **Flash Media Server2**

Flash Media Server, voorheen Flash Communication Server genoemd, levert streaming media mogelijkheden en biedt een ontwikkelomgeving om RIAs te ontwikkelen. De Flash Media Server biedt drie verschillende services.

Als eerste kan de server data van en naar clients versturen via het RTMP protocol. Clients kunnen remote procedure calls, RPCs genoemd, naar de server sturen en de server kan op zijn beurt functies aanroepen aan de client kant. Een zogenaamd SharedObject kan gebruikt worden om data in AMF formaat te versturen naar alle clients. Dit berust op het zelfde principe als het eerder besproken PHPObject. De data overdracht vindt in dit geval echter plaats tussen de flash clients onderling in plaats van tussen Flash en PHP functies. Er kunnen meerdere clients tegelijk luisteren naar een call.

Ten tweede kan de Flash Media server verbindingen maken met verschillende (web)services. Hierdoor is het mogelijk complexe applicaties, berustend op (web)services, te maken met de Flash Media Server.

Als laatste en tevens belangrijkste service kan de Flash Media Server media streams ontvangen en verzenden. Zo kunnen clients niet alleen Flash video bestanden vanaf de server streamen maar bijvoorbeeld ook webcam streams naar de server toesturen.

De Flash Media Server developer edition is gratis beschikbaar om applicaties mee te ontwikkelen en testen. Deze kan echter slechts een gelimiteerd aantal connecties aan. Om over meerdere connecties te kunnen beschikken zullen licenties gekocht moeten worden. Helaas zijn deze op dit moment nog erg kostbaar. Een licentie voor circa 150 connecties bedraagt rond de 4500 euro. Voor dit project en een omroep als de TROS, met duizenden website bezoekers per dag, is dit geen oplossing. Gelukkig zijn er sinds kort enkele open source alternatieven voor de Flash Media server beschikbaar. Voorbeelden hiervan zijn de RED5 server en de HaxeVideo server.

7.4.2. RED5

RED5 is een project begonnen als open-source implementatie voor het RTMP protocol. De man hierachter is John Grden, die tevens het eerste boek over de Flash Media Server, destijds Flash Communications Server genaamd, heeft geschreven. Het idee van Jhon Grden om als eerste een open source alternatief voor de Flash Media Server te maken werd sceptisch ontvangen. Volgens kenners zou het project te groot worden en niet lang stand houden. Dit was echter niet het geval. Begin oktober 2005 zag RED5 het licht.

Het RED5 project bestaat grotendeels uit het hacken en implementeren van het RTMP protocol. In het beginstadium van het project waren ook de opensource ontwikkelaars zelf sceptisch over de ondersteuning voor video en audio streaming. Sinds versie 0.3 behoort dit echter tot de mogelijkheden.

Men wil met de uiteindelijke release van RED5 server niet alleen een alternatief voor de Flash Media Server 2 bieden, maar ook veel functies van Flex 2.0 ontwikkelomgeving leveren. De RED5 server biedt meer mogelijkheden in vergelijking met Flash Media Server, doordat deze volledig gebruik kan maken van de Java programmeer taal. Aangezien Flash Media Server slechts één bepaalde versie van Javascript toe laat, lijkt deze gelimiteerd te zijn. RED5 daarentegen kan gebruikmaken van meerdere talen als Java/JRuby/Groovy en Python.

7.4.3. Haxevideo

Haxevideo is een open source streaming server voor Flash video, geschreven in de Haxe programmeertaal. De server ondersteunt video streaming en -recording en maakt tevens gebruik van het AMF protocol. Om de haxevideo server te kunnen draaien moet Haxe worden geïnstalleerd. Haxe is een web georiënteerde taal waarmee websites en web applicaties ontwikkeld kunnen worden. Het levert tevens de mogelijkheid om SWF files te compileren. Tijdens dit onderzoek is het niet gelukt deze video server aan de praat te krijgen om een nog onbekende reden.

7.5. Toekomstige Technieken

Twee Flash gerelateerde ontwikkelingen die binnenkort op het toneel zullen verschijnen zijn Actionscript 3.0, onderdeel van Flash 9 opvolger van Flash 8, en Silverlight, het antwoord van microsoft op flash. In de volgende twee paragrafen wordt hier kort op ingegaan.

7.5.1. ActionScript 3

Actionscript 3.0, de opvolger van Actionscript 2.0, is een nieuwe krachtige objectgeoriënteerde programmeertaal voor flash waarmee we RIAs kunnen maken. Hoewel Actionscript 2.0 en 3.0 op het eerste gezicht veel van elkaar lijken te verschillen is dit niet helemaal het geval. De overgang van 2.0 naar 3.0 is kleiner dan de overgang van 1.0 naar 2.0.

Actionscript 3.0 voldoet volledig aan de derde editie van de 'ECMAScript language Specificatie' en bevat zelfs al onderdelen uit de vierde editie.

Om Actionscript 3.0 Applicaties te kunnen draaien hebben is de nieuwe Flash Player 9.0 of hoger benodigd. Binnen de Flash Player word Actionscript code uitgevoerd door de ActionScript Virtual Machine (AVM). Met de komst van de nieuwe Flash Player 9.0 is er ook een nieuwe versie van de ActionScript Virtual machine 2, AMV2. Flash Player 9.0 ondersteunt tevens de oudere virtual machine 1 om zo compatible met Actionscript 2.0 te blijven.

Nieuwe eigenschappen van Actionscript 3.0 zijn onder anderen:

- Een verbeterde foutafhandeling tijdens de run-time. Met speciale classes kan men nu duidelijke foutrapporten maken.
- Gegevens types worden tijdens de run-time gecontroleerd waardoor de ontwikkelaar verplicht is de types juist te declareren. Dit word 'strong typing' genoemd en zorgt voor zogenaamde 'type safety' waardoor er prestatieverbeteringen in de AVM zijn en het geheugen beter gebruikt wordt.

- Nieuwe ondersteuning voor XML dankzij de volledige implementatie van de derde ECMAScript editie. Actionscript 3.0 hoeft XML niet meer te parsen en ziet het als een gewoon datatype binnen Flash.
- Gebruik van reguliere expressies, dit zijn ingewikkelde regels voor het doorzoeken van strings.
- Nieuwe gegevenstypen 'int' en 'uint' om getallen in op te slaan. In ActionScript 2.0 is alleen het gegevenstype 'Number' beschikbaar die veel geheugen gebruikt .

Ook zijn er veel functies en klassen veranderd waarbij opvalt dat bekende Actionscript 2.0 functies als '_global', '_root' en '_level' helemaal zijn verdwenen. De functies '_global' en '_root' worden anders aangeroepen en het level principe is totaal verwijderd uit Actionscript 3.0.

7.5.2. Silverlight

Silverlight, voorheen Windows Presentation Foundation genoemd, kan worden beschouwd als een bedreiging voor Flash omdat deze op hetzelfde vlak opereert. Silverlight bestaat uit een crossplatform plugin en een ontwikkelmodel voor het creëren van RIAs, met ondersteuning voor AJAX, C#, VB, Python, en Ruby.

Silverlight bestaat uit twee onderdelen:

- Een 'Core Presentation Framework' dat bestaat uit een user interface en interactieve componenten voor onder anderen de mediaplayer, webapplicaties en presentatie componenten als vector plaatjes, teksten, animaties en bitmaps. Ook is er een DOM API om Silverlight vanuit scripts (bijvoorbeeld javascript) aan te roepen en maakt Silverlight gebruik van XAML, dat staat voor eXtensible Application Markup Language. Hiermee kunnen vector plaatjes en animaties in 2d en 3d worden gedefinieerd.
- Een '.net Framework' subset bestaande uit verschillende libraries en componenten.



8. Veiligheid

Belangrijke eis vanuit de TROS waaraan een Flash applicatie moet voldoen is dat deze veilig is. Hierbij is van belang op welke manier data wordt ingeladen en opgeslagen.

8.1. Flash Player Beveiliging

De Flash Player bevat een ingebouwde beveiliging die werkt op vier niveaus. De eerste twee niveaus hebben betrekking op de Flash Player instellingen op de gebruikers computer. Het is mogelijk om als administrator, van een lokale computer, een globale beveiliging in te stellen. Deze beveiliging is van toepassing op de webcam, audio inputs, lokale disk download en upload instellingen, flash player schijf gebruik en als laatst de auto update instellingen. De beveiligingsinstellingen zijn ook op gebruikersniveau in aan te passen. De laatste twee niveaus worden extra toegelicht omdat deze van toepassing zijn op het FlashLive project en er tevens bij de TROS mee wordt gewerkt. Ten eerste bestaan er zogenaamde 'Cross-Domain policy' bestanden. Deze bestanden bestaan uit regels xml code waarin kan worden aangegeven tot welke data, een flash movie afkomstig uit een ander domein, toegang mag hebben op het betreffende domein. Sinds Flash Player 7 bestaat zoekt Flash, als er externe data van een bepaald domein word ingeladen, op de root van dit domein naar een '*crossdomain.xml*'. Indien de Flash Player geen '*crossdomain.xml*' kan vinden of geen toegang krijgt, zal externe data niet worden ingeladen. (*voor een voorbeeld crossdomain.xml zie bijlage 14.8*) Als laatste bestaan er cross-script APIs, die ontwikkelaars de mogelijkheid geven om vanuit Actionscript enkele beveiligingsopties voor de Flash applicatie in te stellen. Deze beveiligingsopties stellen onder anderen script toegang en toegang tot lokale connecties in. Tevens is er de mogelijkheid om de Flash Player op een andere lokatie naar een crossdomain.xml te laten zoeken .

8.2. f_TEA

f_TEA is een 128-bit encryptie class die gebaseerd is op het XTEA voor Flash en PHP. Met de f_TEA class kan gevoelige tekst data met een 128 bit sleutel worden versleuteld en weer teruggehaald. f_TEA versleutelt niet de Flash movie maar alleen de uitgewisselde data tussen Flash en de buitenwereld. Ook binaire data, zoals plaatjes, kunnen door f_TEA niet worden versleuteld. f_TEA kan via de Extension Manager worden toegevoegd aan flash. De F_TEA classes zijn hierna in de standaard Class map te vinden: <pad naar de classes>..\Classes\com\bnm\fc . Met 'import com.bnm.fc.f_TEA;' kan de F_TEA encryptie worden toegevoegd aan een flash project. Vervolgens kunnen de encodeString en decodeString functies respectievelijk 'strings' coderen en decoderen.

9. Snelheid en belasting

Een belangrijke eis vanuit de TROS waar een applicatie aan moet voldoen is een hoge laadsnelheid en een lage server belasting. Nu zal een Flash applicatie zelf niet voor een grote server belasting zorgen omdat de bewerkingen grotendeels aan de client kant worden uitgevoerd. Om deze reden wordt hier geen verder onderzoek naar verricht. Aangezien de laadsnelheid van verschillende data en laad technieken wel een bepaalde drempelwaarde kan overschrijden, is hier verder onderzoek naar gedaan.

Er kunnen op twee manieren in Flash data worden ingeladen: binair en niet binair. Dit blijkt voor de laadsnelheid gevolgen te hebben.

9.1. *Niet binaire data*

XML en tekst bestanden worden niet binair overgestuurd en zullen aan de client kan worden ingelezen waardoor de server belasting minimaal blijft. De effectieve snelheid kan bij het inlezen aan de client wel laag zijn. De flash client moet bijvoorbeeld XML bestanden eerst parsen voordat de data gebruikt kan worden. Dit kan met grote hoeveelheden data lang duren en in oude Flash players eventueel tot time-outs leiden, wanneer het laden van een XML langer dan 15 seconden duurt. Voor kleine applicaties waar kleine hoeveelheden data moet worden opgeslagen is XML handig, voor grotere applicaties met veel data is een database een betere oplossing. Een XML bestand is gemakkelijk te maken en heeft geen extra services nodig om te kunnen worden gebruikt in Flash. Wanneer de applicatie uit verschillende onderdelen bestaat, die data los van elkaar aanroepen, is het beter om verschillende kleine XML bestanden te maken. Op deze manier is er sprake van een hogere laadsnelheid dan wanneer er één groot bestand van zou worden gemaakt.

9.2. *Binaire data*

Door gebruik van Flash Remoting, PHPAMF of PHPObjekt kan data binair via het AMF formaat worden overgestuurd en zal deze data niet meer door Flash moeten worden omgezet. Als gevolg hiervan is de laad snelheid hoger. Wanneer binaire data overdracht wordt gebruikt is het ook mogelijk om data op te slaan in een database. Indien er veel data moet worden opgeslagen is het gebruik van een relationele database, een database met tabellen die gerelateerd zijn aan elkaar, aan te raden. Het gebruik van indexen, query's en Caching kan er voor zorgen dat data snel opgezocht wordt.

10. Resultaat Analyse

Na het onderzoek volgt nu een technische analyse per onderwerp en wordt er vooruit gekeken naar de schaalbaarheid van de gekozen applicaties.

10.1. Gemaakte technische keuzes en methodes

10.1.1. Ontwikkelomgeving

Uit het testen van verschillende ontwikkelomgevingen is gebleken dat de standaard Flash IDE qua workflow enkele tekortkomingen bevat. In de standaard Flash IDE lijkt de nadruk nog steeds op de grafische interface te liggen. Ontwikkelomgevingen als Flex, FlashDevelop, Eclipse en Se|py bieden voor het schrijven van Actionscript classes betere tools en verbeterde "code completion" en "syntax checking". Ook is het enigszins verouderde 'timeline' principe in deze ontwikkelomgevingen steeds minder belangrijk.

Het gebruik van MTASC als compiler in ontwikkelomgevingen als FlashDevelop, Eclipse en Se|py heeft een groot voordeel ten opzichte van de standaard Flash IDE. De MTASC compiler is vele malen sneller dan de standaard Flash compiler. Tevens biedt MTASC meer uitgebreide 'trace' mogelijkheden.

De keuze van de ontwikkelomgeving is altijd een persoonsgebonden overweging. Wel lijkt het een goede zet te zijn om van de standaard Flash IDE af te stappen. Adobe geeft met het uitbrengen van de FLEX ontwikkelomgeving zelf ook aan dat het programmeren in Actionscript een meer serieuze omgeving nodig heeft, dan de standaard IDE. Eclipse en Flex, waarvan de laatst genoemde gebaseerd is op de eerst genoemde, bieden de gebruiker de meeste mogelijkheden. Echter, door de complexiteit en overbodige luxe van deze twee ontwikkelomgevingen, voor dit project, vallen deze af. Voor de ontwikkeling van het FlashLive systeem lijkt FlashDevelop de beste keuze te zijn, aangezien het een compacte omgeving is waar weinig instellingen voor nodig zijn.

10.1.2. Alternatieve Compilers / tools

Van de drie alternatieve compilers die zijn onderzocht, is na lang testen naar voren gekomen dat het gebruik van een combinatie van MTASC en SWFMILL de beste keuze is. Voor het creëren van het FlashLive systeem wordt MTASC als compiler gekozen, omdat deze momenteel de beste compiler is.

FlashLive moet echter ook in staat zijn om Flash movie bestanden 'server side' te genereren, voor bijvoorbeeld nieuwe pagina's. Om deze reden is een tweede compiler benodigd. Deze moet vanuit Flash een beschrijving kunnen ontvangen, om daar nieuwe Flash movie bestanden mee te genereren. Hoewel Ming deze eigenschap bezit door uit PHP Flash movie bestanden te kunnen maken, is er gekozen voor SWFMill. Ming heeft een grotere overhead, is complexer in gebruik en wordt niet vaak geupdate. SWFMill is relatief nieuw en gebruikt xml files, dat handig kan zijn om pagina's als referentie op te slaan zonder ze te compileren. Tevens bevat het een snellere compiler.

10.1.3. Actionscript

ASwing en XLeff zijn interessante nieuwe concepten. Beiden kunnen voordeel bieden in het creëren van user interfaces.

ASwing geeft ontwikkelaars de mogelijkheid om kant en klare user interface componenten te gebruiken. Deze componenten bevatten veel eigenschappen en functies. FlashLive zal in eerste instantie alleen layout panels gebruiken voor het opbouwen van een user interface. Het gebruik van speciale ASwing interface componenten is hierdoor overbodig.

XLeff lijkt een perfect concept te zijn voor dit project. De ontwikkeling gaat echter langzaam en er is weinig bruikbare documentatie beschikbaar. Er is mede om deze reden gekozen om geen gebruik te maken van XLeff. Wel kan het dynamisch opbouwen van een user interface met XML als concept dienen voor het opbouwen van een pagina layout. Het is bijvoorbeeld goed mogelijk om, met het hiervoor besproken SWFMill, Flash movies te creëren gebaseerd op XML. Hieruit vloeit de mogelijkheid om een pagina layout virtueel te creëren, binnen Flash, en als SWFMill XML beschrijving weg te schrijven.

10.1.4. Uitwissel techniek

De data uitwisseling met Flash heeft een belangrijk aandeel in dit project.

Uit de bevindingen blijkt dat bij grote hoeveelheden data, tekstbestanden en XML bestanden als uitwissel techniek niet erg geschikt zijn. Voor tekstbestanden is er meer code nodig wanneer er meer data is om in te laden. XML heeft als voordeel een gestructureerd te zijn, maar moet door flash worden geparsed alvorens de data gebruikt kan worden.

Flash Remoting heeft als voordeel data binair in het AMF formaat te kunnen versturen. Dit levert een hoge laadsnelheid op. Het is echter een nadeel dat er een gateway benodigd is en server side scripting, om het AMF formaat te kunnen omzetten.

Er is besloten om een combinatie van XML en Flash Remoting voor het FlashLive systeem te gebruiken. XML zal worden gebruikt voor de opbouw en layout van de pagina's. Dit zijn kleine hoeveelheden data die direct zonder tussenkomst van server side scripting in flash geladen kunnen worden. De eerder gemaakte keuze voor het gebruik van SWFMill ondersteunt dit. Voor het inladen van content zal gebruik kunnen worden gemaakt van Flash Remoting, omdat het een grote hoeveelheid data kan betreffen. Het gebruik van Flash Remoting kan in dit geval een hogere snelheid opleveren. Om geen dure Flash oplossing van Adobe te kiezen zal, indien Flash Remoting aan bod komt, voor AMFPHP gekozen worden. AMFPHP is open source, gratis te verkrijgen en heeft een grote gebruikersgroep.

10.2. Analyse Schaalbaarheid

De schaalbaarheid van applicaties is voor de TROS een belangrijke voorwaarde. Een applicatie moet makkelijk aanpasbaar zijn om zo in verschillende omgevingen, met verschillende omstandigheden, inzetbaar te zijn. Flash applicaties zijn van nature statisch. Met de komst van verschillende uitwisselingstechnieken kan content zich, binnen een flash applicatie, echter aanpassen. Het gedane onderzoek heeft enkele technieken uitgelicht, die naast content, ook de mogelijkheid geven interface aanpassingen mogelijk te maken. Met het inzetten van SWFMill, die Flash movies vanuit een XML beschrijving kan genereren, ontstaat de optie om online Flash movies aan te passen aan verschillende omgevingen en omstandigheden.

10.3. Analyse Beveiliging

In het onderzoek is gekeken naar een tweetal beveiligingsmethodes. Eén vanuit Flash Player zelf en één actionscript class, die data kan versleutelen. De Flash Player beveiliging is standaard goed ingesteld. De Flash Player is bijvoorbeeld niet in staat om zonder specifiek 'crossdomain.xml' bestand, data in te laden. Ook kan binnen de Flash applicatie worden aangegeven met welke externe scripts de applicatie mag samenwerken. Dit zorgt voor een relatief goede beveiliging en zal daarom ook worden toegepast in dit project.

Het versleutelen van data met de f_TEA class blijkt na onderzoek soms fout te gaan. In sommige situaties is het überhaupt niet handig om opgeslagen data te versleutelen. Systemen als typo3, dat bij de TROS wordt gebruikt, kunnen dan namelijk de data niet meer lezen. Mede hierdoor is er voor gekozen om geen gebruik te maken van de f_TEA class.

10.4. Analyse Snelheid

Er is aan de hand van het snelheidsonderzoek gebleken dat voor grote hoeveelheden data het gebruik van binaire dataoverdracht de hoogste laadsnelheid geeft, en daarom de beste keuze hiervoor is. Aangezien FlashLive gebruik zal maken van SWFMill voor het genereren van Flash movies, en deze met XML werkt, zal hiervoor ook met niet binaire dataoverdracht gewerkt worden. De omvang van deze XML bestanden is zo klein dat dit niet van invloed zal zijn op de snelheid van de applicatie.

11. Conceptontwerp

11.1. Inleiding

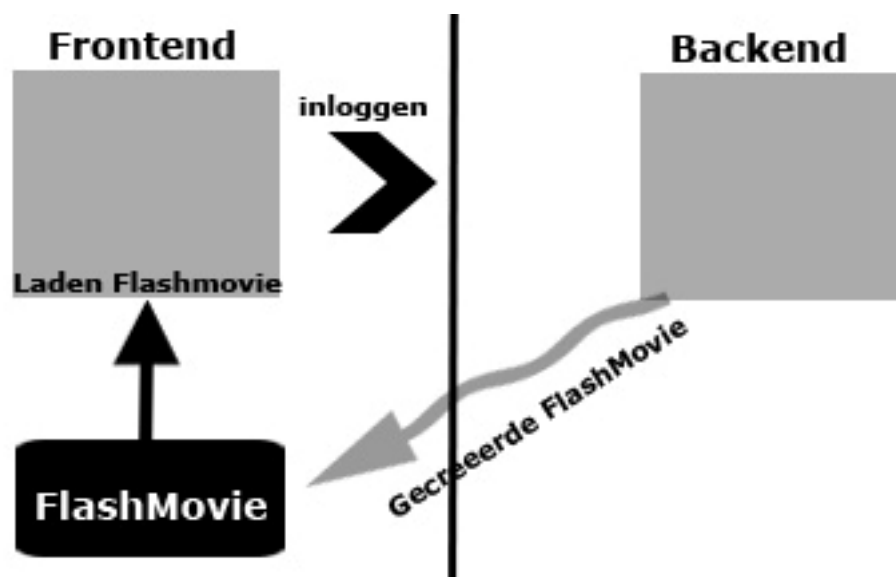
Na het onderzoek en de analyse van de beschikbare technieken is er een concept tot stand gekomen. Bij de totstandkoming van dit concept is rekening gehouden met de eisenlijst voor een Flash content syteem.

Er is een functioneel ontwerp gemaakt waarin de voornaamste eigenschappen zijn verwerkt. Het maken van een design, rekeninghoudend met de eisenlijst, is een goed beginpunt om een overzicht te krijgen over hoe FlashLive in elkaar moet steken.

Na het functioneel ontwerp is er voor elk onderdeel van het FlashLive systeem een technische beschrijving en dataflow gemaakt. Op deze manier is er getracht een sluitend systeem te maken, dat rekening houdt met snelheid, belasting en beveiliging.

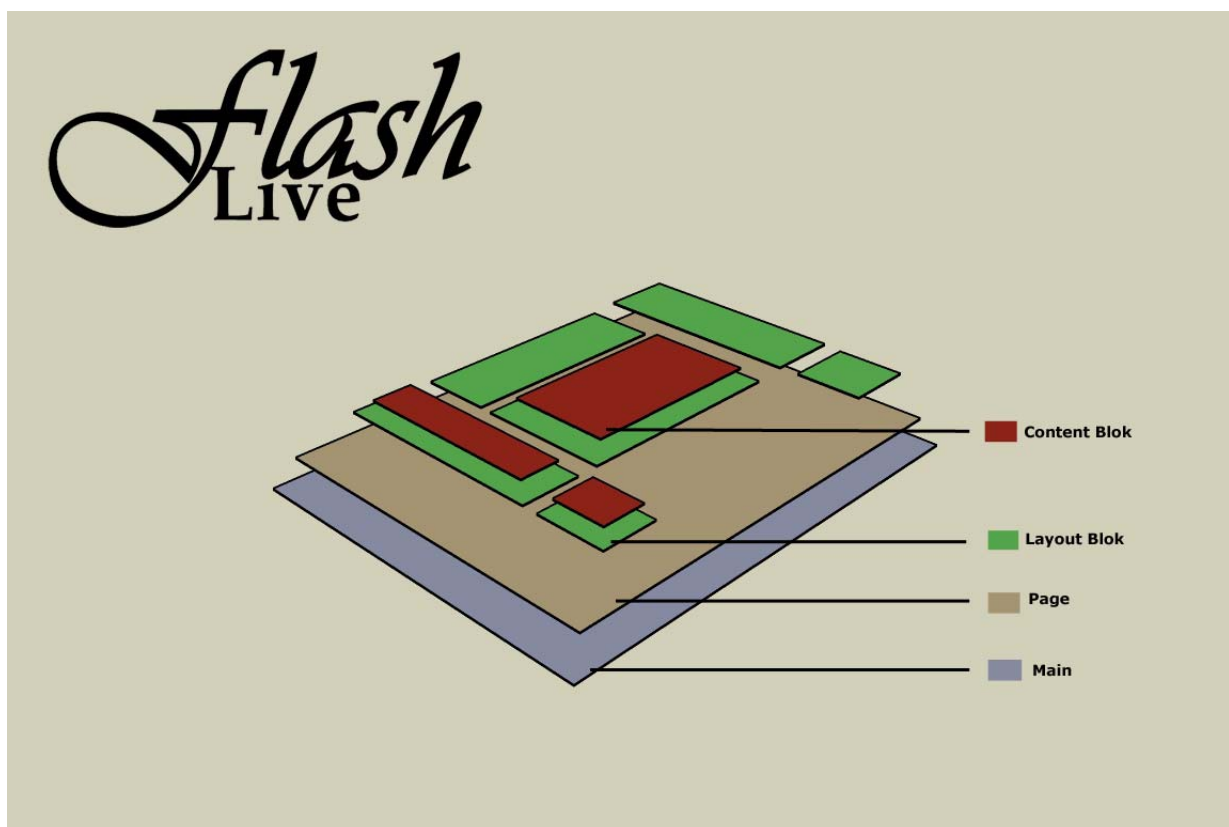
11.2. Structuur

Het FlashLive systeem heeft twee kanten. De een frontend kant en een backend kant. Aan de frontend kant krijgt de bezoeker de gecreëerde site te zien die met de FlashLive backend in elkaar is gezet. De frontend kant en backend kant zijn om veiligheidsredenen van elkaar gescheiden door een .Htaccess beveiligde directory. Omdat de frontend kant alleen het inladen van gecreëerde pagina's en inloggen als functie heeft, wordt in het functioneel ontwerp alleen ingegaan op de backend kant, het voornaamste onderdeel van het FlashLive systeem. In het technisch ontwerp zullen de twee functies van de frontend wel worden beschreven.



11.3. FlashLive Opbouw

Het FlashLive Systeem is opgebouwd uit vier lagen. De eerste laag bestaat uit de site holder, genaamd 'Main', en wordt op de frontend aangeroepen. In de Main worden, de in de backend gecreëerde, pagina's geladen. In het onderstaande overzicht wordt een dergelijke pagina met 'Page' aangeduid. Elke Page bezit layout blokken die door de eindgebruiker zijn gecreëerd. Op elk layout blok kan de gebruiker vervolgens content blokken creëren. Het creëren van layout blokken en content blokken geschiedt middels het verslepen van de gewenste blokken op de desbetreffende 'Page'.



11.4. Functioneel Ontwerp

11.4.1. FlashLive Structuur

Het FlashLive systeem is aan de backend zijde eenvoudig doch doeltreffend opgebouwd en heeft de volgende structuur:

- Site
 - Save
 - Settings
 - Preview
 - Help

- Pages
 - New
 - Help

- Layout
 - Page
 - Preset
 - Pannel Manager

- Content
 - Page
 - Component Manager

- Log Off

11.4.2. FlashLive Onderdelen

Wanneer men in de backend inlogt zal men een lege pagina te zien krijgen met bovenin een menubalk en links daarvan een dropdown menu.



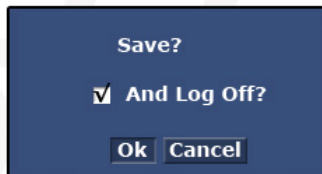
Het dropdown menu zorgt voor de hoofdnavigatie door de FlashLive backend en bestaat uit de volgende onderdelen: *Site*, *Pages*, *Layout*, *Content*, *Logoff*. Deze worden per onderdeel besproken.

11.4.2.1. Site

Het *site* onderdeel geeft algemene site opties weer en bevat de volgende sub pagina's:

Save

Hier worden alle wijzigingen site breed opgeslagen en kan men kiezen om uit te loggen



Settings

Site settings als Titel, Meta Data, on /off line kunnen hier worden opgegeven.



Preview

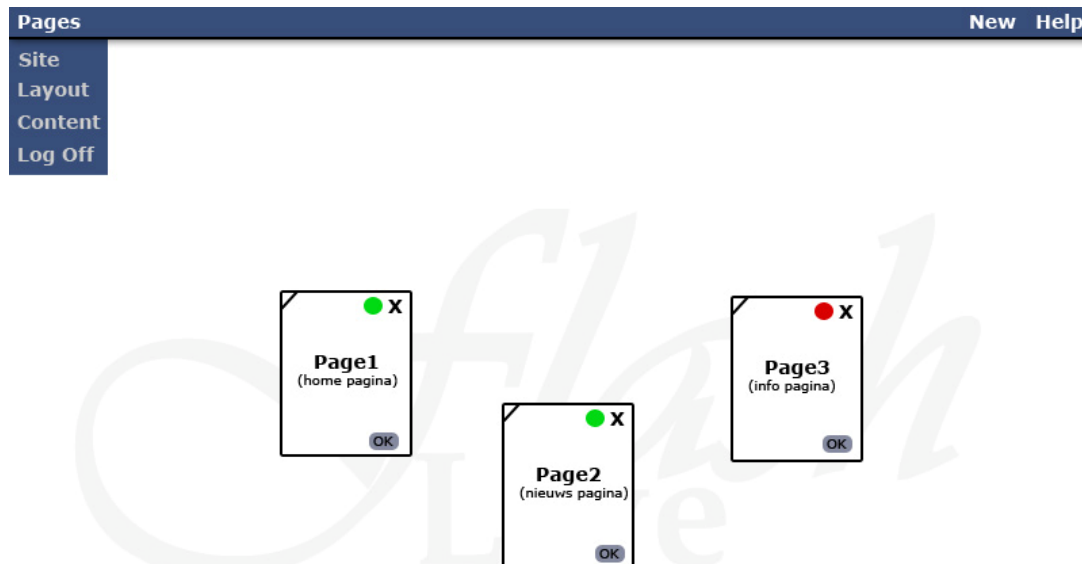
Laat een voor gecompileerde versie van de site zien.

Help

Help onderdeel.

11.4.2.2. Pages

Het *pages* onderdeel laat de op de stage, de blanco achterkant, eerder gemaakte pagina's zien, en biedt de mogelijkheid om elke pagina direct te verwijderen en aan of uit te zetten.



Het *pages* onderdeel heeft de volgende sub pagina's:

New

Het creëren van een nieuwe pagina en het invoeren van een pagina titel , omschrijving en online status.

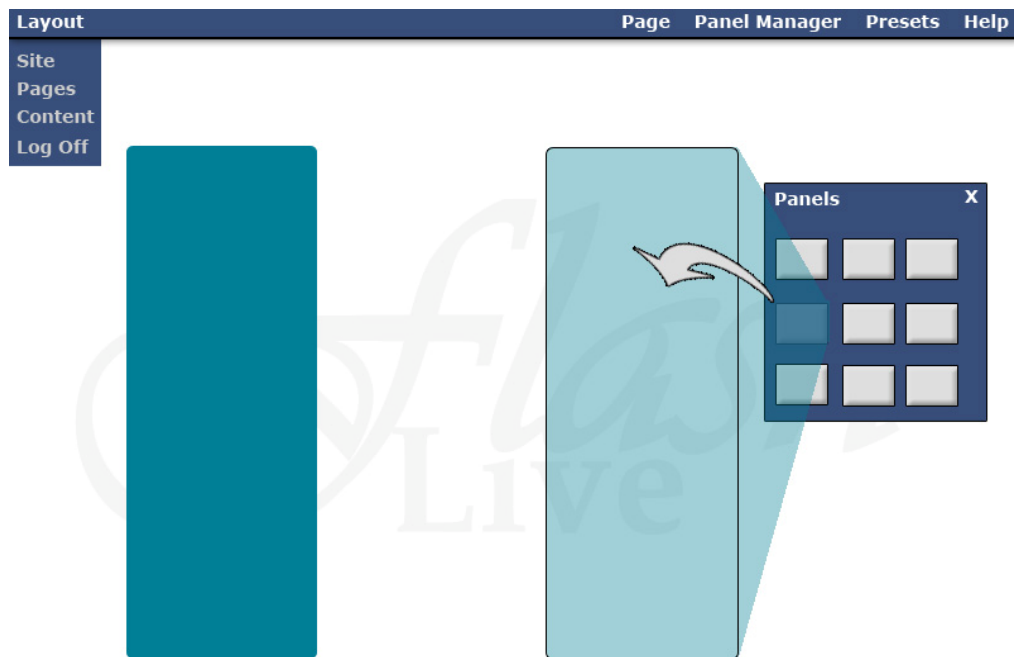


Help

Help onderdeel.

11.4.2.3. Layout

In het *layout* onderdeel kan men een pagina selecteren om daar zogenaamde 'layout panels' via een 'drag and drop' manier op te slepen en zo gemakkelijk een layout in elkaar te zetten.



Het *layout* onderdeel bestaat uit de volgende sub pagina's:

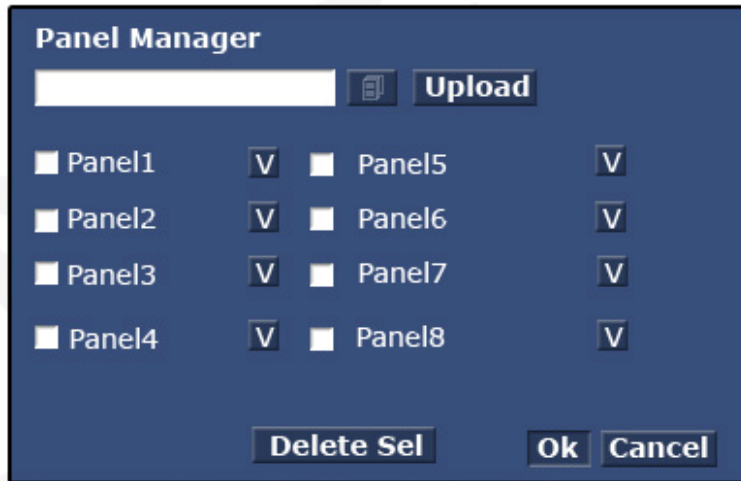
Pages

Laat een overzicht van reeds gecreëerde pagina's zien met de mogelijkheid om deze te selecteren voor de layout toepassing



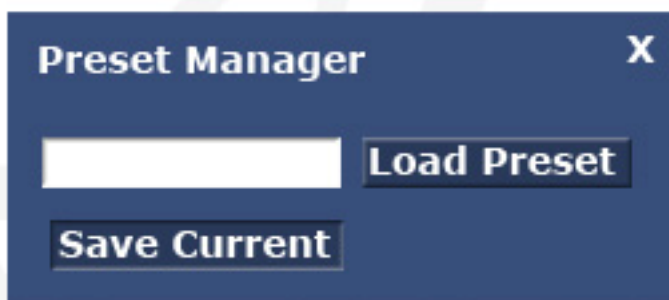
Pannel Manager

Hier kunnen Layout Pannels worden toegevoegd en worden verwijderd.



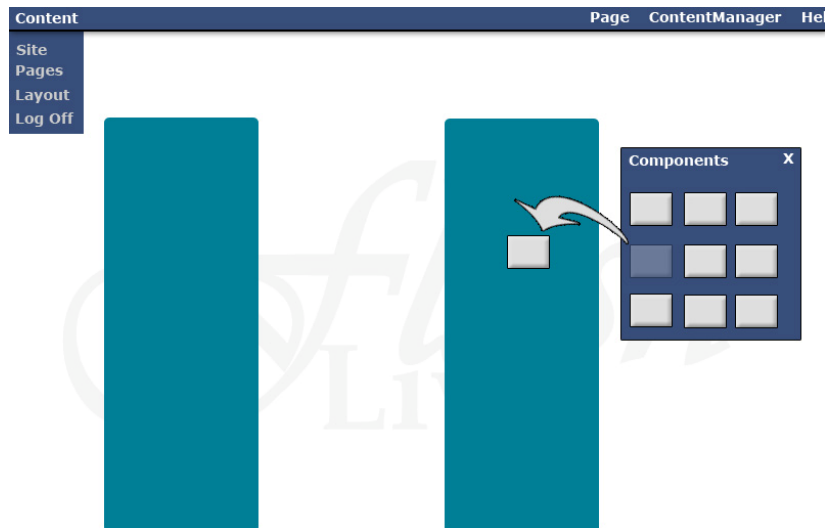
Presets

Hier kunnen gemaakte layouts als XML worden opgeslagen en nieuwe layouts worden ingeladen.



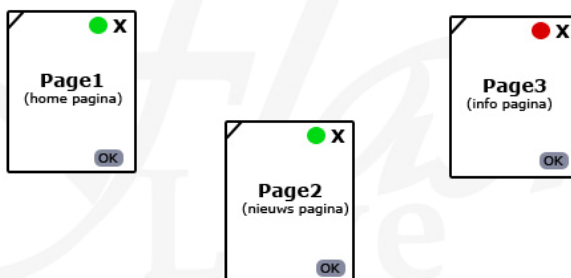
11.4.2.4. Content

Het *content* onderdeel laat het toe dat de gebruiker een lege of layout pagina selecteert om daar content componenten op te slepen via een 'drag and drop' manier, en zo mogelijk te bewerken.



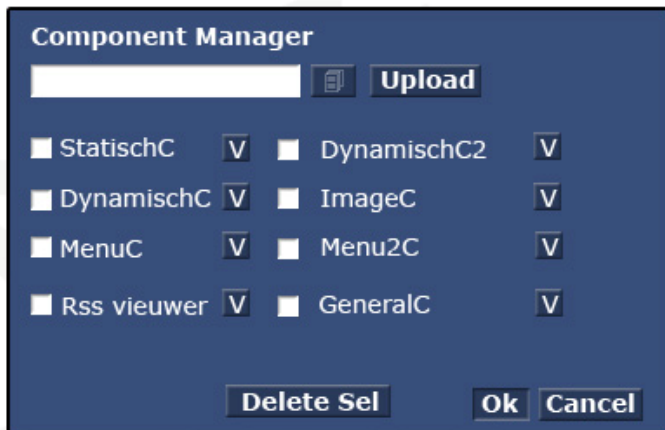
Pages

Laat een overzicht van reeds gecreëerde pagina's zien met de mogelijkheid om deze te selecteren voor de content toepassing.



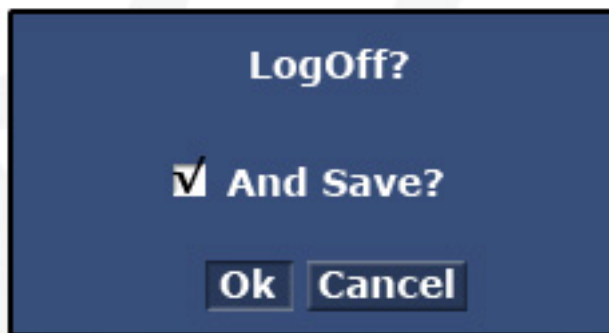
Component Manager

In de component manager kan men content componenten toevoegen en verwijderen.



11.4.3. Logoff

LogOff laat het uitlog venster zien met een optie om alle wijzigingen op te slaan.



11.5. Technisch Ontwerp

11.5.1. Dataflow FlashLive

Hier volgt per onderdeel een technische beschrijving met daarna een grafische dataflow van de belangrijkste functies. Er wordt in deze beschrijving gesproken over, XML bestanden voor SWFMill, een uitleg over dit type bestand is te vinden in bijlage 16.9. Om een logische volgorde aan te houden is er voor gekozen om het site onderdeel als laatste te beschrijven.

11.5.1.1. Main(frontEnd) - Technische beschrijving

De Main pagina heeft twee functies, het inladen van gecreëerde pagina's en het aanroepen van de backend.

Het inladen van gecreëerde pagina's geschiedt door het uitlezen van de 'pageRef.xml'. In deze xml wordt bijgehouden welke pagina's er zijn gecreëerd en welke eigenschappen deze pagina's bezitten. De eigenschappen bestaan uit een titel, beschrijving en status van de pagina. Ook wordt er in deze XML aangegeven welke pagina de Main als eerste moet aanroepen. Na het inlezen weet de Main welke Flash movie moet worden geladen.

Functies:

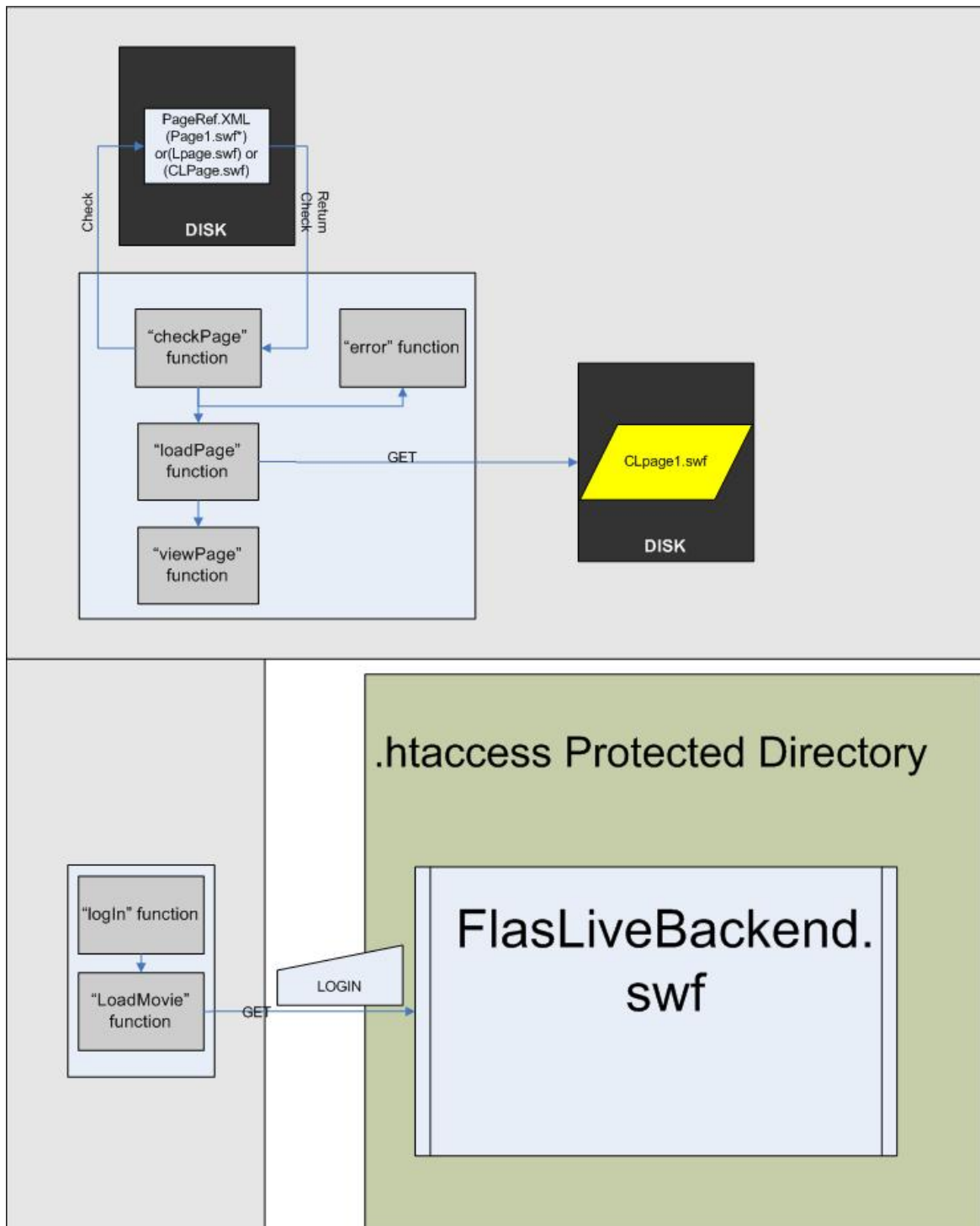
- **checkPage:** controleert pageRef.xml en geeft de lokatie van de te laden pagina door aan de loadPage functie.
- **loadPage:** laadt de doorgekregen pagina in de Main
- **errorPage:** geeft een foutmelding wanneer pagina ontbreekt of nog niet is aangemaakt
- **viewPage :** geeft de pagina op de Main stage weer

Het inloggen op de backend geschiedt door het laden van de backend uit een .htaccess beveiligde directory door middel van de loadMovie functie.

Functies:

- **login:** wordt actief na een geheimen toetscombinatie en roept de loadMovie functie aan
- **loadMovie:** laadt backend uit de beveiligde directory

Dataflow: Main (frontEnd)



11.5.1.2. Pages - Technische beschrijving

Het Pages onderdeel heeft twee functionaliteiten:

- Het creëren van een nieuwe lege pagina
- Het bewerken of verwijderen van reeds bestaande pagina's

Het creëren van nieuwe pagina's geschiedt door het invoeren van gegevens zoals in de functionele specificatie staat beschreven. Na het invoeren van de gegevens worden deze verstuurd naar, 'create_SWFMill_XML.php', welke een XML beschrijving, genaamd 'page.xml', van de nieuw te creëren pagina maakt. Deze pagina kan door SWFMill worden gecompileerd. 'Create_SWFMill_XML.php' stuurt een event terug naar de onResult functie van flash om te laten weten of het creëren van de XML is gelukt.

Wanneer de onResult functie een positieve melding krijgt zal deze 'Excecuteswfmill.php' aan roepen. Excecuteswfmill roept SWFMill aan, verstuurt een melding naar Flash en update de 'pageRef.xml' met gegevens van de nieuwe pagina. SWFMill gebruikt de gecreëerde XML beschrijving om een nieuwe pagina te genereren.

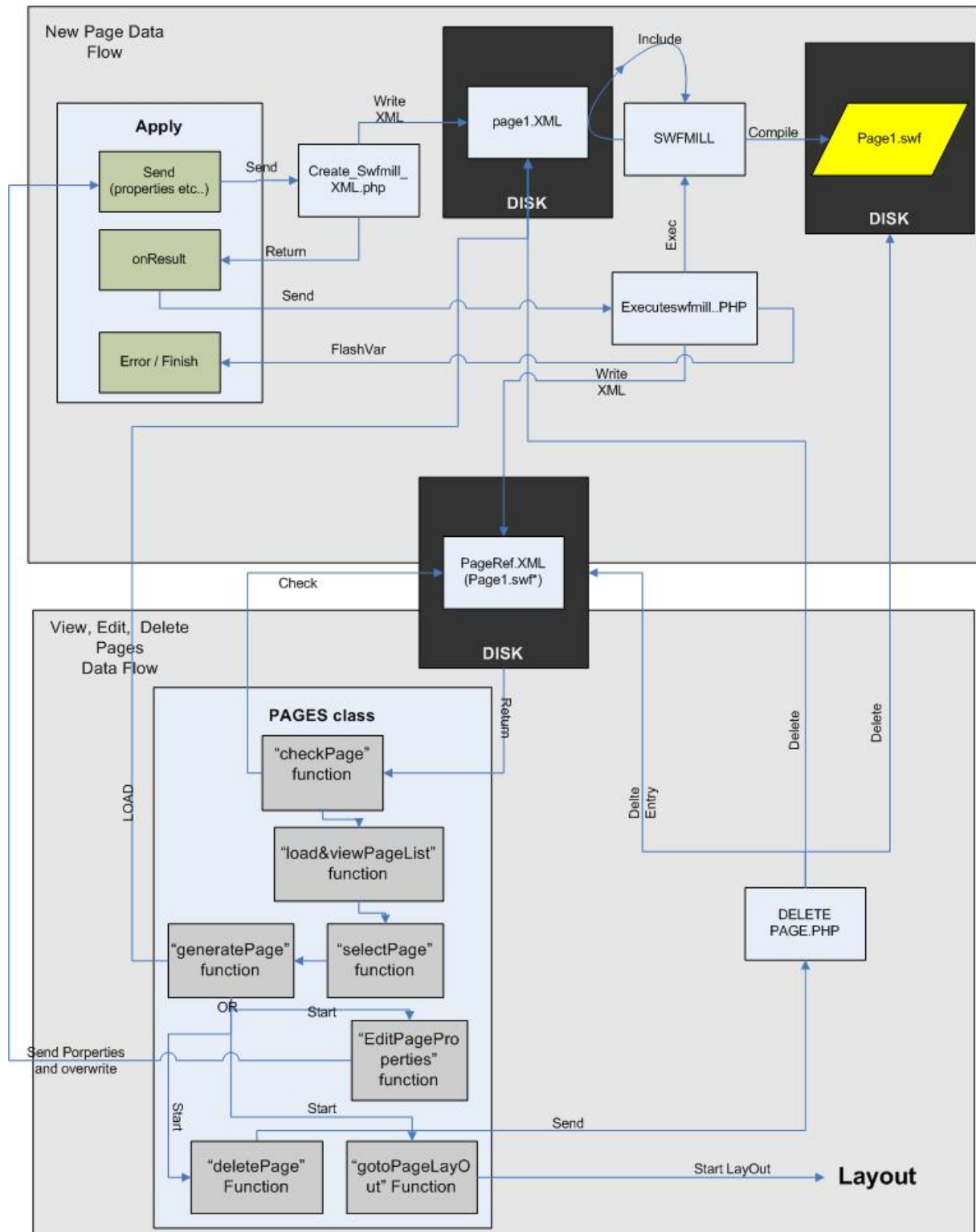
Wanneer er reeds pagina's zijn gecreëerd, zullen deze bij het openen van het Pages onderdeel zichtbaar zijn. De 'checkPage' functie leest 'pageRef.xml' uit om reeds beschikbare pagina's door te kunnen geven aan de 'loadandviewPageList' functie, die de beschikbare pagina's op de stage laat zien.

Wanneer een pagina wordt geselecteerd voor bewerkingen, zal de 'selectPage' functie, de 'generatePage' functie aanroepen. Deze genereert een virtuele representatie van de pagina met behulp van de bijhorende SWFMill XML beschrijving. Na wijzigingen zullen de bovenstaande stappen voor het creëren van pagina's worden herhaald en worden de output bestanden overschreven.

Het is mogelijk om een pagina te verwijderen. Bij het verwijderen wordt de 'deletePage' functie aangeroepen. De 'deletePage' functie zal 'delete.php' aanroepen, die de gecreëerde Flash movie en XML beschrijving zal verwijderen. Tevens wordt de pagina referentie uit 'pageRef.xml' verwijderd.

De 'gotoLayout' functie stuurt de gebruiker door naar het layout gedeelte.

Dataflow: Pages



11.5.1.3. LayOut - Technische beschrijving

Het Layout onderdeel heeft drie functionaliteiten:

- Het toevoegen van een layout aan een pagina en het updaten daarvan.
- Het beheren van layout panels
- Het opslaan en laden van layout presets

Wanneer het Layout onderdeel wordt aangeroepen zal de 'checkPage' functie uit de 'pageRef.xml' de beschibare pagina's tonen. De 'loadandview' functie laadt de paginalijst en laat deze zien. Wanneer via de 'selectPage' functie een lege of layout pagina wordt geselecteerd, wordt 'generatePage' aangeroepen om uit de juiste XML een virtuele representatie van de pagina te maken.

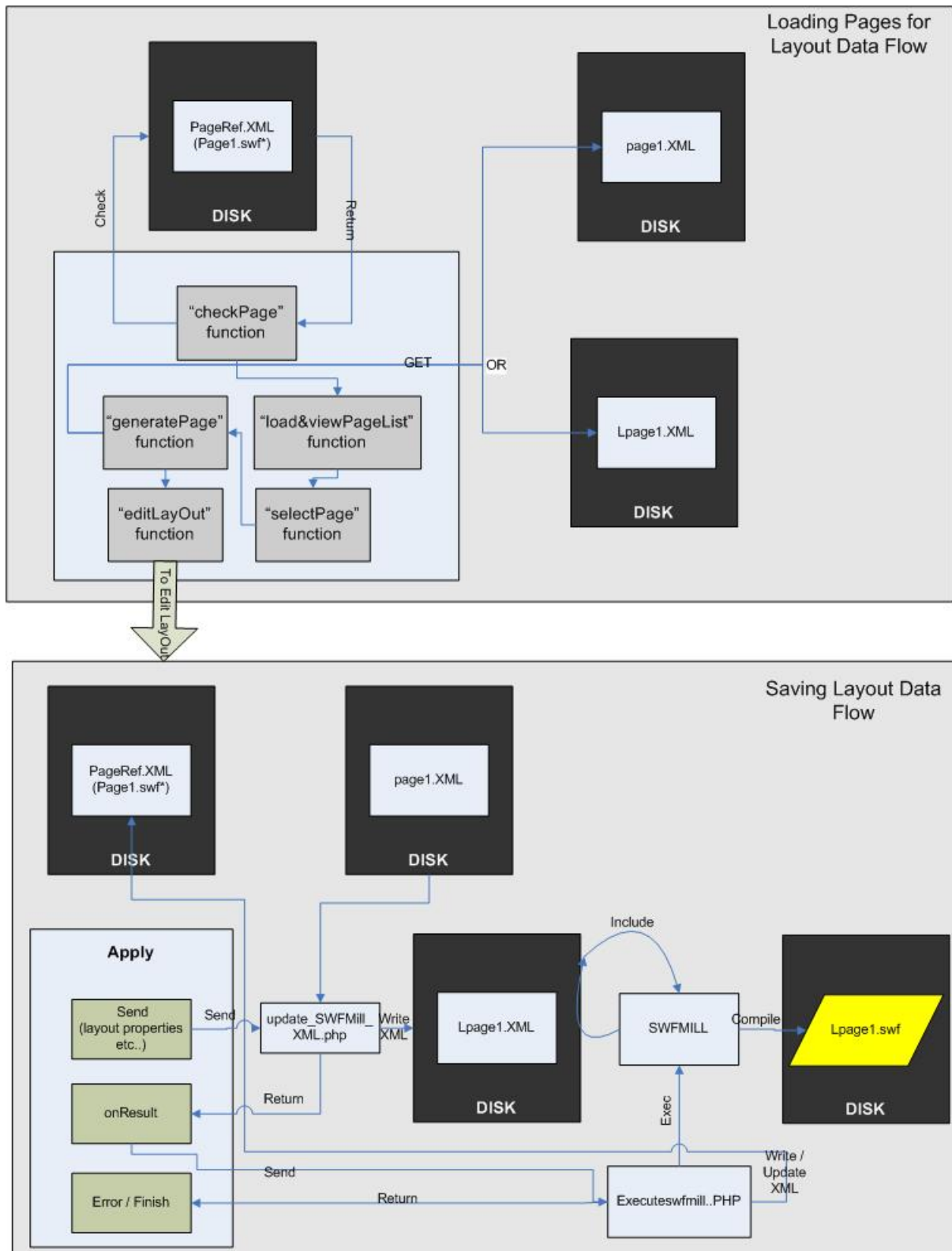
De gebruiker is nu in staat een layout op de virtuele pagina te maken zoals staat beschreven in de functionele specificatie.

Na het plaatsen van de layout panels worden de gegevens verstuurd naar 'create_SWFMill_XML.php', die een XML beschrijving, in de dataflow 'Lpage.xml' genoemd, van de nieuw te creëren pagina met layout maakt.

'Create_SWFMill_XML.php' haalt de eerder gemaakte page.xml op en gebruikt deze om layout gegevens toe te voegen. Wanneer het creëren van de XML beschrijving is geslaagd, stuurt 'Create_SWFMill.php' een event terug naar de onResult functie van flash. Deze laat weten of het creëren van de XML is gelukt. Wanneer de onResult functie een positieve melding krijgt zal deze 'Excecuteswfmill.php' aan roepen. Excecuteswfmill roept SWFMill aan, verstuurt een melding naar Flash en update de 'pageRef.xml' met nieuwe statusgegevens van de pagina. SWFMill gebruikt de gecreëerde XML beschrijving om een nieuwe pagina te genereren.

Layout wijzigingen worden via bovengenoemde stappen uitgevoerd.

Dataflow: Layout

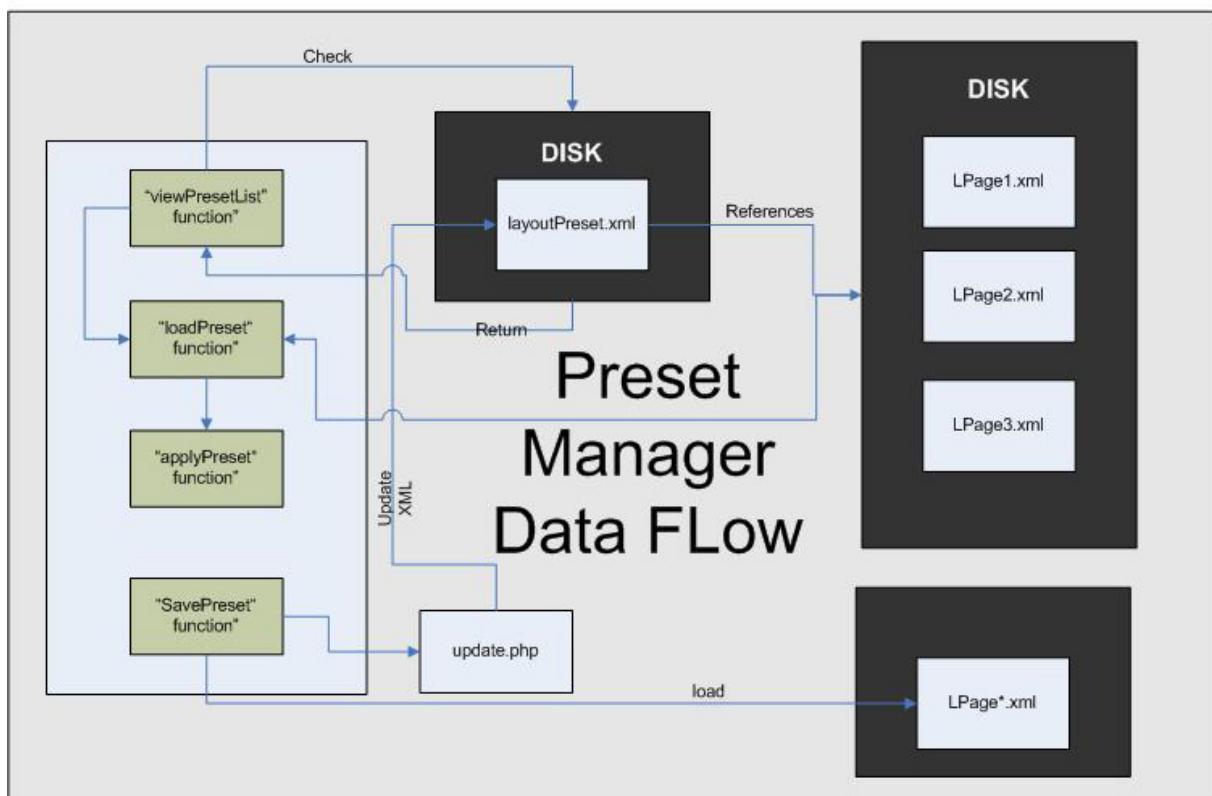


Preset Manager – Technische beschrijving

De Layout Preset Manager laat de gebruiker gemaakte layout beschrijvingen opslaan en inladen.

De 'viewPresetList' functie haalt uit 'presetRef.xml' de lokatie van de beschikbare layout beschrijvingen. De 'loadPreset' functie laadt de juiste 'Lpage.xml' in en roept de 'applyPreset' functie aan. 'applyPreset' stuurt de XML beschrijving naar de 'generatePage' functie van het Layout onderdeel.

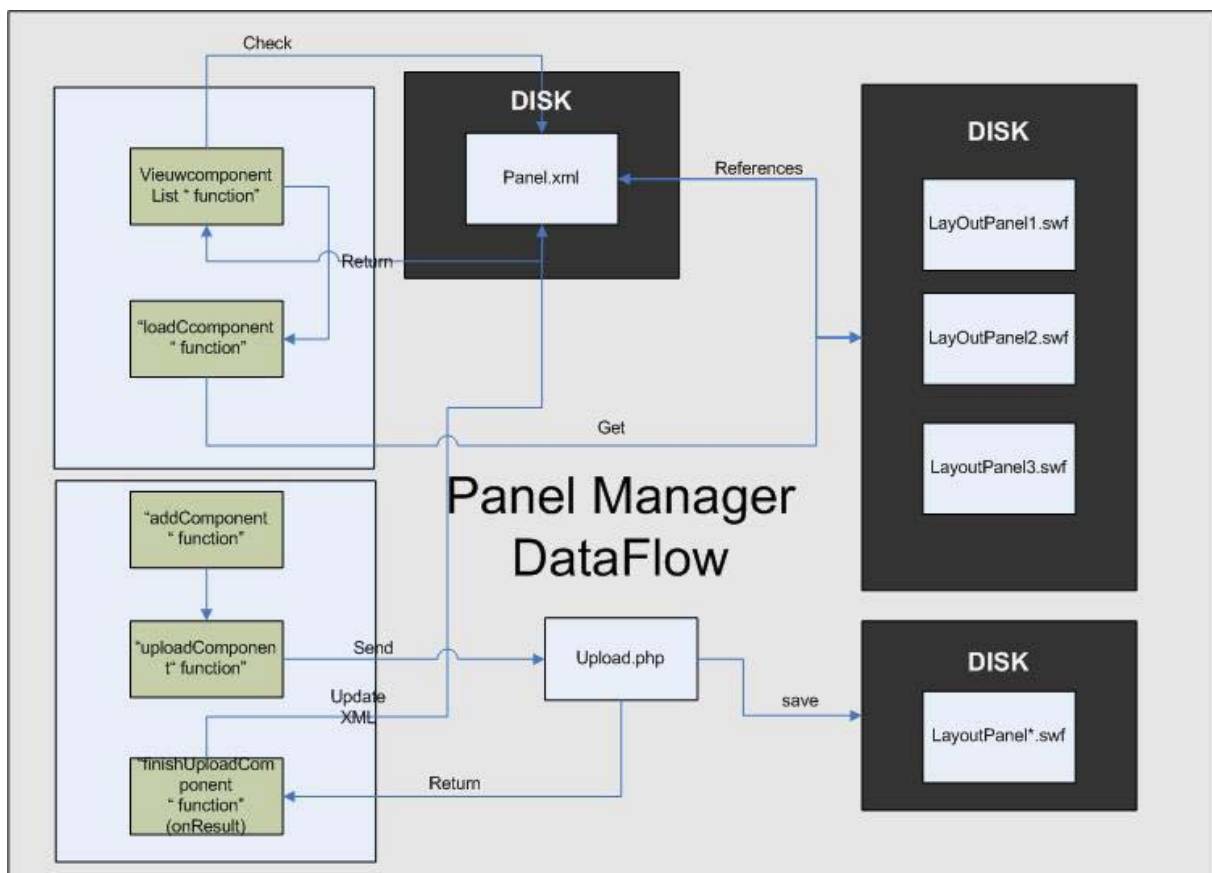
Dataflow: Layout Preset Manager



Panel Manager – Technische beschrijving

De Panel Manager geeft de gebruiker de mogelijkheid layout Panels te importeren. De 'VieuwcomponentList' functie haalt uit 'Panel.xml' een lijst met beschikbare layoutpanels. Deze worden doorgegeven aan de 'loadComponent' function voor gebruik. De 'addComponent' function zorgt voor het lokaal selecteren van layout componenten. De bestandsreferentie wordt doorgegeven aan de 'uploadComponent' functie die 'Upload.php' aanroept om de nieuwe layout component online op te slaan. Als het uploaden is geslaagd zal 'upload.php' de verwijzing toevoegen aan 'Panel.xml'.

Dataflow: Layout Panel Manager



11.5.1.4. Content – Technische beschrijving

Het Content onderdeel heeft twee functionaliteiten:

- Het toevoegen van content aan een layout pagina en het updaten daarvan
- Het beheren van content componenten

Wanneer het Content onderdeel wordt aangeroepen zal de 'checkPage' functie uit de 'pageRef.xml' de beschikbare pagina's tonen. De 'loadandview' functie laadt de paginalijst en laat deze zien. Wanneer de 'selectPage' functie een layout pagina 'Lpage.xml' of layout+content pagina 'CLpage.xml' selecteert, wordt de 'generatePage' functie aangeroepen om uit de bijhorende XML een virtuele representatie van de pagina temaken.

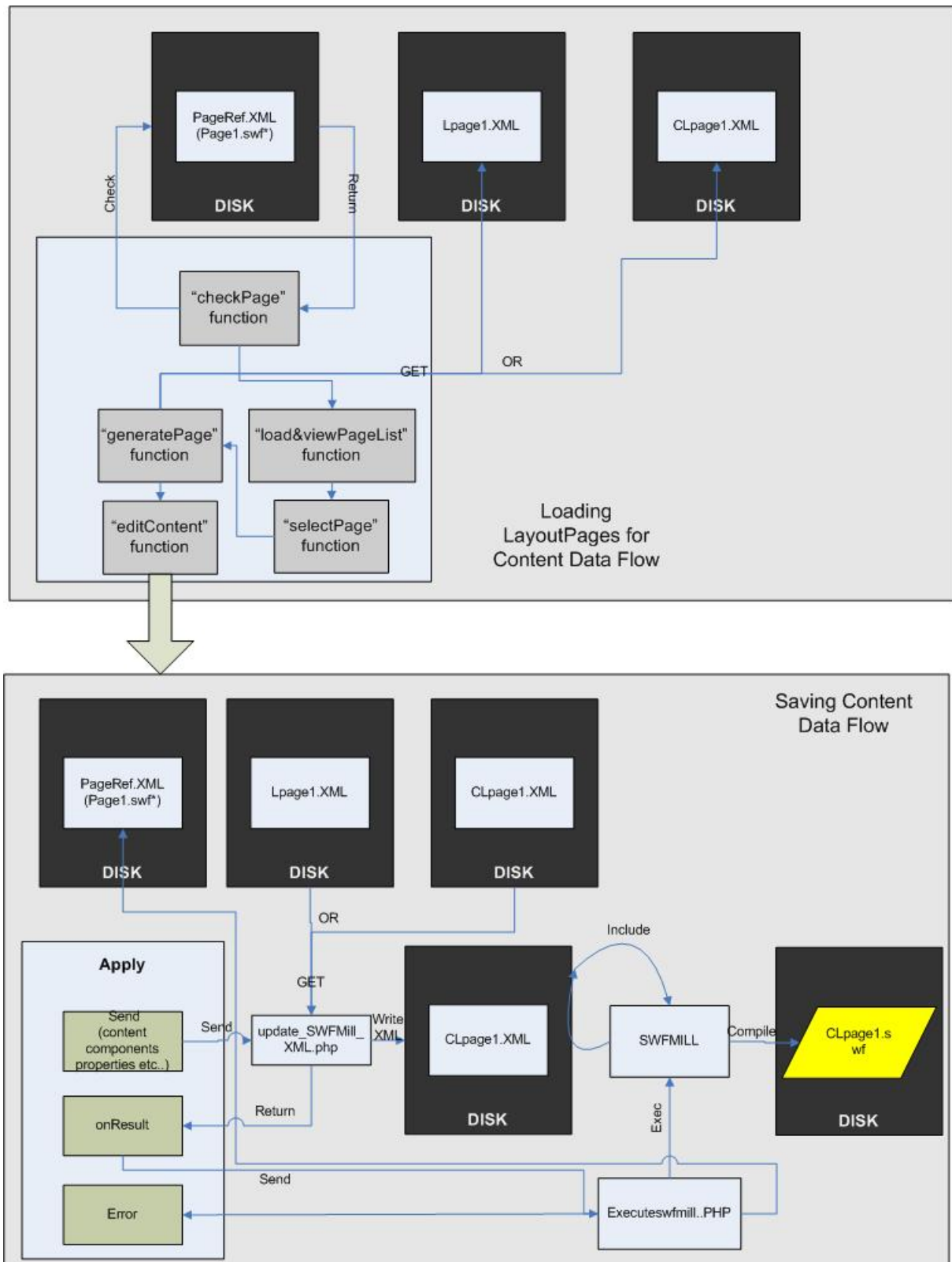
De gebruiker is nu instaat om content op de virtuele pagina te slepen en aan te aanpassen. Dit staat beschreven in de functionele specificatie.

Na het plaatsen van content componenten, worden de statische content data en component eigenschappen verstuurd naar 'create_SWFMill_XML.php'. Deze maakt een XML beschrijving ,CLpage.xml genaamd, van de nieuw te creëren pagina met layout en content opmaak. 'Create_SWFMill_XML.php' haalt de eerder gemaakte Lpage.xml op en gebruikt deze om layout gegevens toe te voegen. Wanneer het creëren van de XML beschrijving is geslaagd, stuurt 'Create_SWFMill.php' een event terug naar de onResult functie van flash. Deze laat weten of het creëren van de XML is gelukt.

Wanneer de onResult functie een positieve melding krijgt zal deze 'Excecuteswfmill.php' aan roepen. Excecuteswfmill roept SWFMill aan, verstuurt een melding naar Flash en update de pageRef.xml met nieuwe statusgegevens van de pagina. SWFMill gebruikt de gecreëerde XML beschrijving om een nieuwe pagina te genereren.

Content wijzigingen worden via bovengenoemde stappen uitgevoerd.

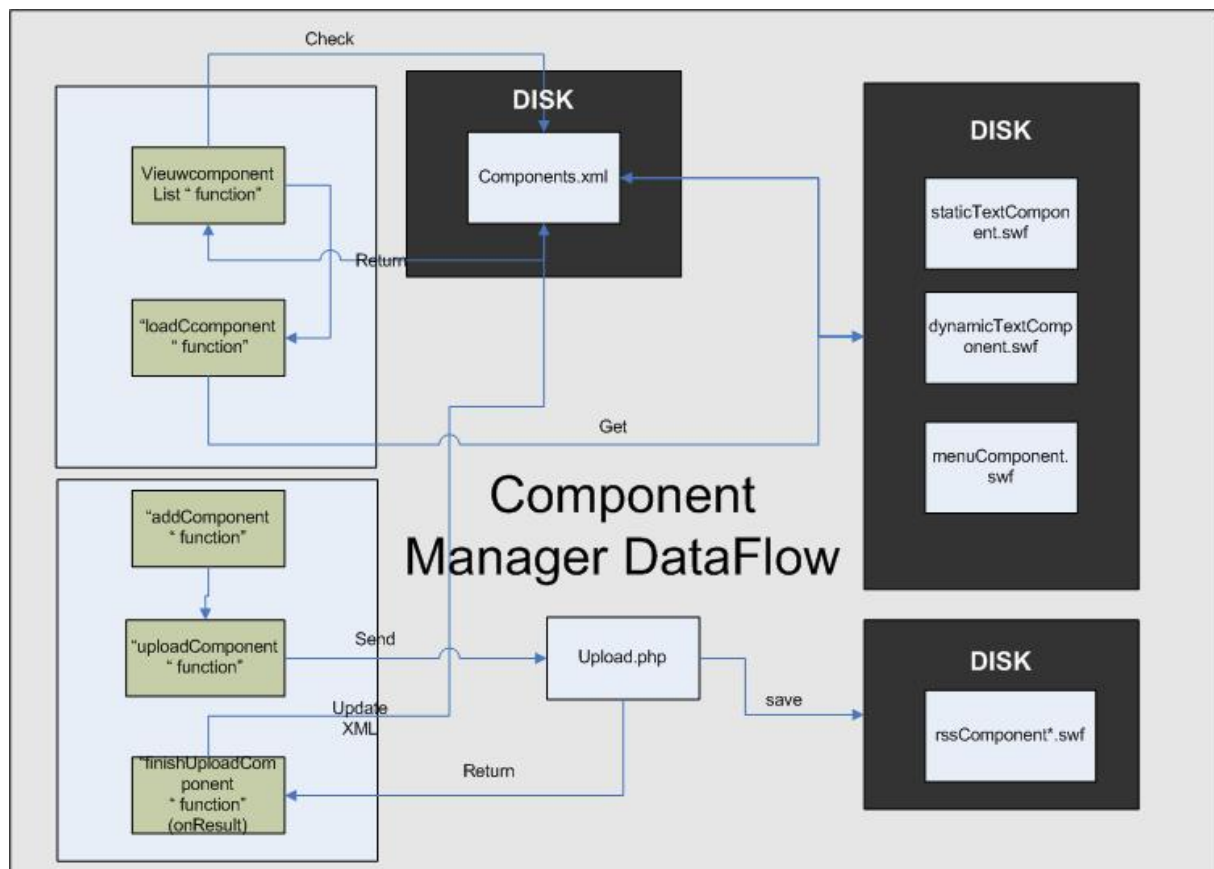
Dataflow: Content



Content Component Manager – Technische beschrijving

De Content Component Manager geeft de gebruiker de mogelijkheid content components te importeren. De 'VieuwcomponentList' functie haalt uit 'Component.xml' een lijst met beschikbare componenten. Deze worden doorgegeven gegeven aan de 'loadComponent' function voor gebruik. De 'addComponent' function zorgt voor het lokaal selecteren van componenten. De bestands lokatie wordt doorgegeven aan de 'uploadComponent' functie die 'Upload.php' aanroept om het nieuwe component te uploaden en online op te slaan. Als het uploaden is geslaagd zal 'upload.php' de verwijzing toevoegen aan 'Components.xml'.

Dataflow: Content Component Manager



11.5.1.5. Site – Technische beschrijving

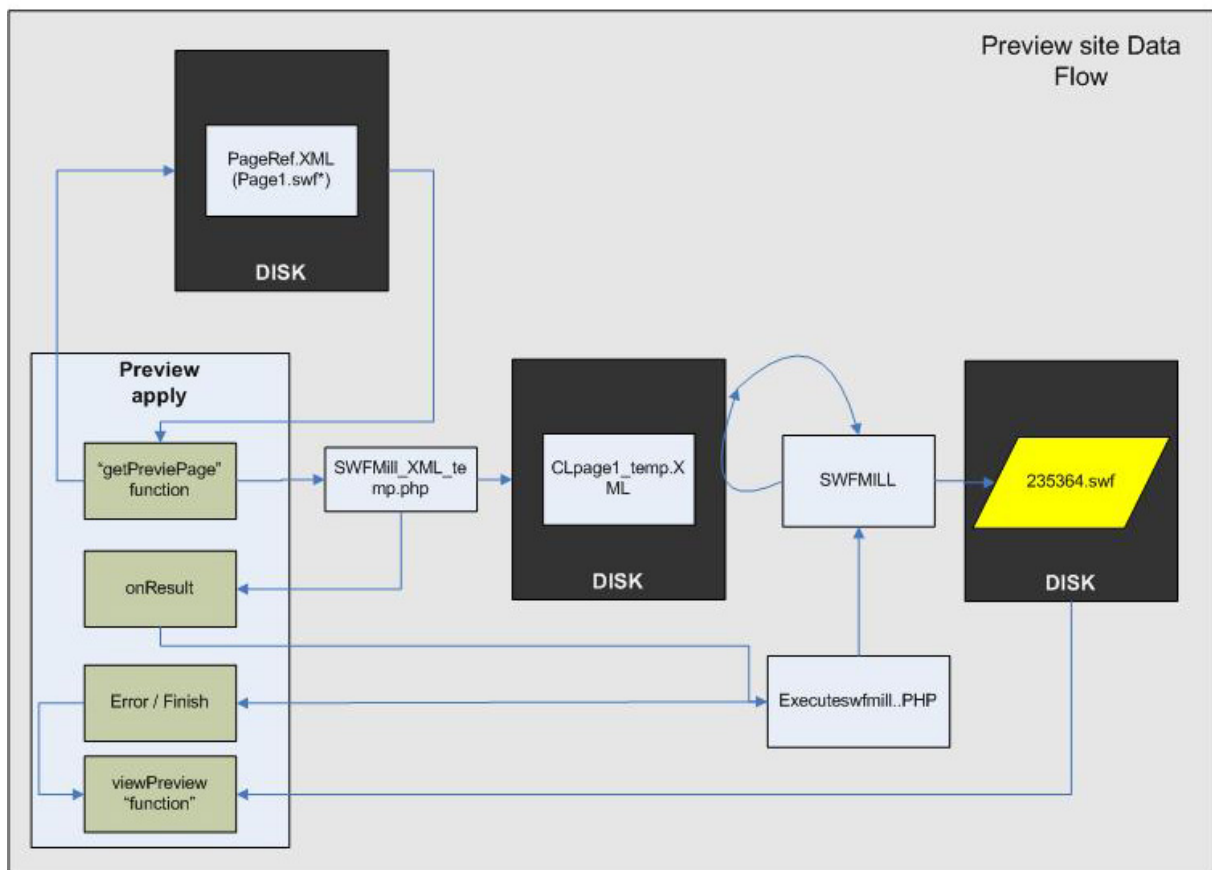
Het Site onderdeel heeft twee functionaliteiten:

- Een preview van de gecreëerde site tonen
- Instellen van site eigenschappen

Wanneer de preview functie wordt aangeroepen zal de 'getPreviewPage' functie de startpagina uit 'pageRef.xml' lezen. Deze wordt aan 'SWFMill_XML_Temp.php' door gegeven om met de XML beschrijving van de startpagina een tijdelijke SWFMill xml te creëren. Wanneer het creëren van de XML beschrijving is geslaagd, stuurt 'Create_SWFMill.php' een event terug naar de onResult functie van flash. Deze laat weten of het creëren van de XML is gelukt.

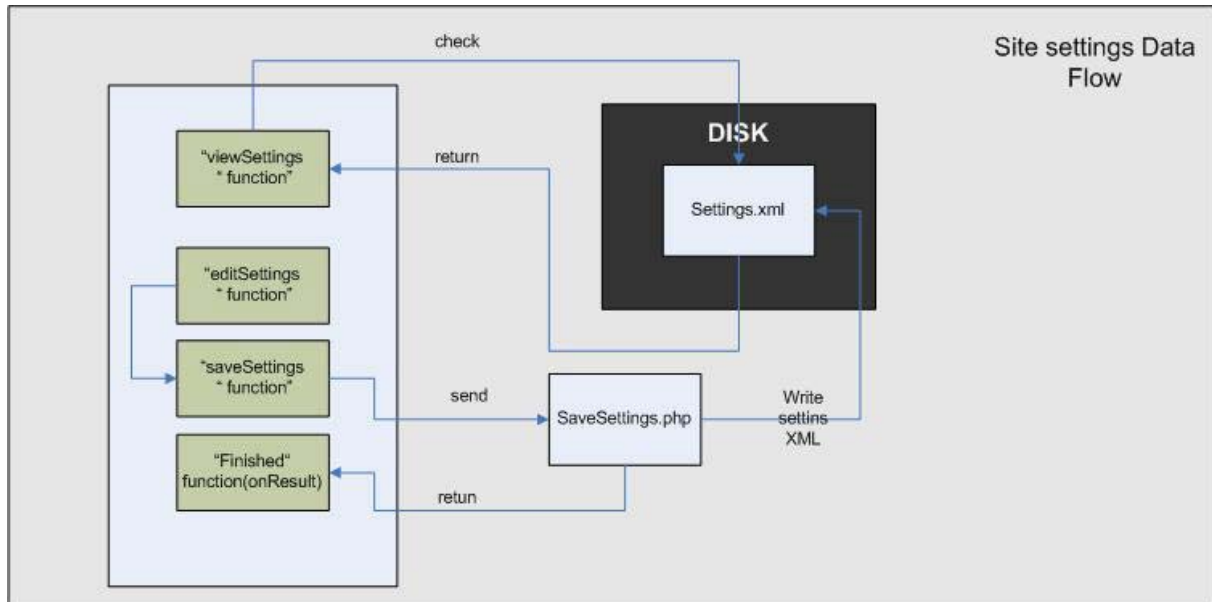
Wanneer de onResult functie een positieve melding krijgt zal deze 'Executeswfmill.php' aan roepen. Executeswfmill roept SWFMill aan en verstuurt een melding naar de 'error/Finish' functie in Flash. SWFMill gebruikt de gecreëerde XML beschrijving om een tijdelijke pagina te genereren. De tijdelijke pagina wordt na het compileren door de 'viewPreview' functie weergegeven.

Dataflow: Site



De site settings functie haalt via de 'viewSettings' functie setting gegevens uit de 'Settings.xml'. Wanneer er aanpassingen gemaakt worden in de site settings, stuurt de 'editSettings' functie de nieuwe waardes naar de 'saveSettings' functie, die SaveSettings.php aanroept. Deze update de settings.xml.

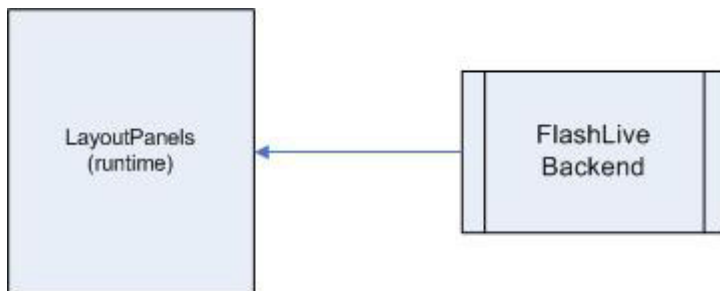
Dataflow: site settings



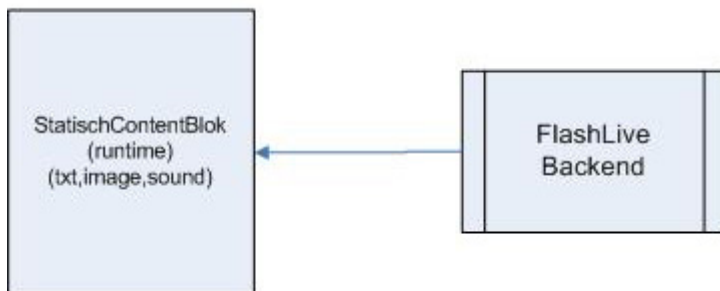
11.5.2. Componenten

Eén van de voornaamste functionaliteit van FlashLive is de mogelijkheid om met componenten te werken. Deze componenten kunnen op pagina's worden geplaatst. De volgende componenten types komen in FlashLive voor:

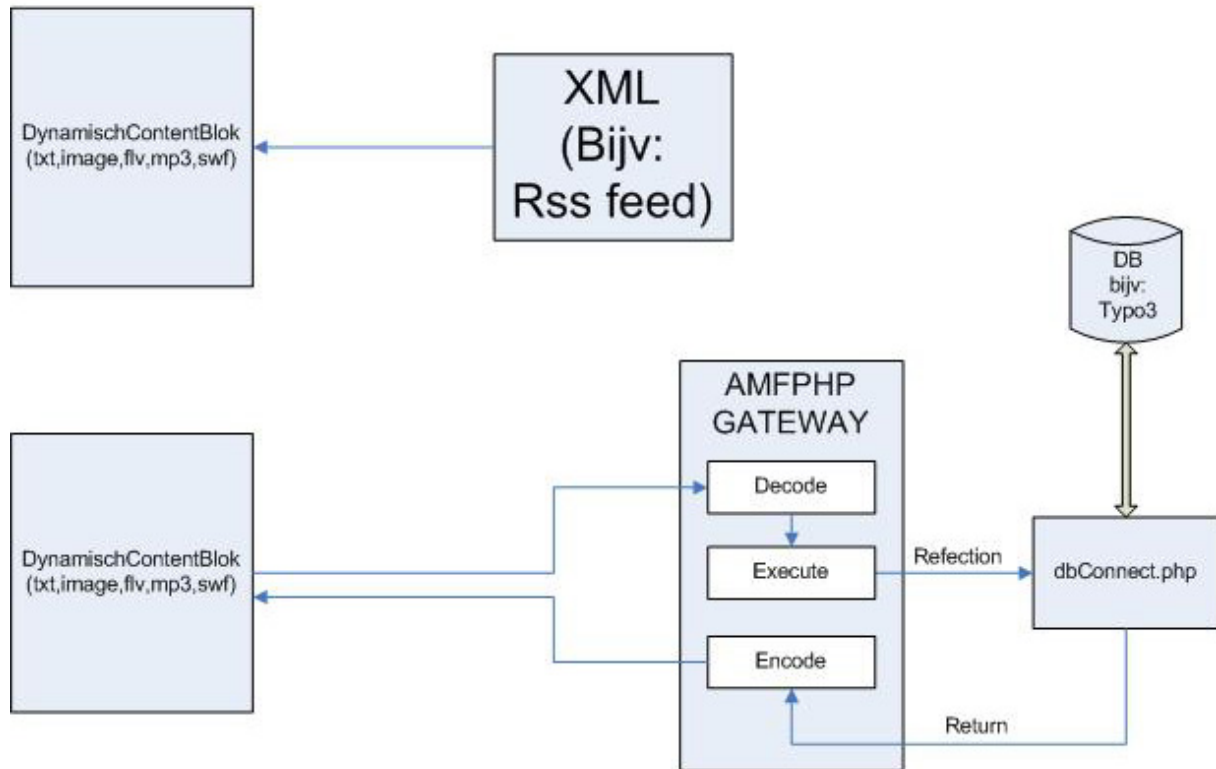
LayoutPanels: een Flash movie bestaande uit grafische elementen. Deze wordt tijdens het genereren in de pagina verwerkt.



Statische Content Blokken: een intern blok met input veld. Content wordt tijdens het genereren in de pagina verwerkt.



Dynamische Content Blokken: een extern component met eigen functies. Content wordt tijdens runtime ingeladen. Dynamische Content Blokken kunnen data via externe bronnen inladen, mits de crossdomain policy file (voor uitleg zie 7.6.1.1) op het externe domein aanwezig is.



Het Menu Blok: creëert tijdens runtime dynamisch een menu door gebruik te maken van pageRef.xml



12. Ontwikkeling

De nadruk van deze scriptie ligt op het onderzoek naar, en de ontwikkeling van een concept voor het FlashLive systeem. Het uitvoeren van dit concept valt buiten de content van deze scriptie omdat vanaf het begin, mede door een voorzien tijdgebrek, hier niet voor gekozen is. Het concept is echter klaar om uitgewerkt te worden door de TROS.

13. Conclusies en Aanbevelingen

13.1. Conclusie Probleemstelling

Na onderzoek te hebben gedaan naar verschillende Flash technieken en de beveiliging, snelheid en belasting van Flash applicaties en een concept te hebben ontwikkeld voor het FlashLive systeem, is het tijd om een conclusie te trekken. Er is gebleken dat op Flash gebaseerde websites te beheren zijn via een Flash content systeem, in dit geval FlashLive. Deze geeft de gebruiker namelijk de mogelijkheid om zowel de content als de visuele aspecten van de op Flash gebaseerde website online te kunnen veranderen. Dit wordt bewerkstelligd door het plaatsen van layout pannels, interface onderdelen en content componenten, die men online in het Flashlive systeem kan aanpassen, zoals in het concept staat beschreven.

Ook is gebleken dat een Flash content systeem net zo gemakkelijk en snel te beheren is als een reguliere website. Dit blijkt uit het feit dat bijvoorbeeld het Flashlive systeem de gebruiker dezelfde functionaliteit kan bieden als met reguliere websites, met een content systeem. De gebruiker is in staat om in te loggen op een backend, website eigenschappen aan te passen, nieuwe onderdelen toe te voegen en content te beheren op een gemakkelijke manier. Dit kan mogelijk gemaakt worden door layout panels op de stage te slepen, en hierop weer content blokken te slepen, die van inhoud kunnen worden voorzien. Aangezien er gebruik gemaakt kan worden van het 'drag and drop' principe kan men relatief snel websites opbouwen.

13.2. Bruikbare mogelijkheden voor de Tros

Er zijn verschillende doelen waarvoor de TROS het FlashLive kan inzetten. Er kan hierbij gedacht worden aan websites waar kinderen zelf hun eigen pagina of weblog kunnen maken. Het FlashLive systeem is gemakkelijk en snel in gebruik en dus ideaal voor kinderen. Ook banners ,op reeds bestaande sites, kunnen met behulp van het FlashLive concept beheerbaar worden, zodat ze snel kunnen worden aangepast. Door het gebruik van dynamische content blokken, die data extern kunnen ophalen, kan ook data uit het, door de TROS gebruikte, typo3 systeem gehaald worden.

14. Evaluatie

Als eerst wil ik kwijt dat mijn afstudeerstage bij de TROS een erg fijne periode is geweest. Er heerst een goede open sfeer en er zijn leuke collega's. Dit alles zorgt ervoor dat ik op een fijne manier aan mijn afstudeerproject heb kunnen werken. Ik heb tijdens mijn stage goede ondersteuning gekregen als het gaat om het uitdenken van mijn gemaakte concept.

Ik ben erg tevreden over het gedane onderzoek en vooral over het uiteindelijke concept.

Tijdens de ontwikkeling van het concept waren er echter enkele kleine struikelblokken. Veel van de bedachte oplossingen en dataflow diagrammen waren in het begin erg complex en niet sluitend. Ik had bijvoorbeeld moeite met het bedenken hoe meerdere contentblokken, in het systeem, ingevoerd konden worden, door gebruik te maken van componenten. Ik probeerde een oplossing te vinden voor het plaatsen van unieke contentblokken die meerdere keren als referentie van het generieke component 'statischcontentblok', op de stage geplaatst konden worden. Het idee was om een kopie van dit 'statischcontentblok' component te maken en deze een unieke naam te geven. Hierdoor zou het systeem tijdens runtime de locatie waardes uit een XML moeten halen waar ieder component, met een unieke naam, geplaatst moet worden. Dit werd al gauw complex. Gelukkig kwam de oplossing, dankzij de tool SWFMill. Bij elke aanpassing worden nu de contentblokken hard in de Flash movie verwerkt zodat daarvan nieuwe Flash movies gegenereerd kunnen worden, of oudere Flash movies overschreven worden. Dit was uiteindelijk een veel simpelere, betere maar vooral werkende oplossing.

Al met al was het afgelopen halfjaar, bij de TROS, fijn, interessant en leerzaam. Ik was, voor na mijn opleiding, opzoek naar een parttime baan die ik kon combineren met mijn bedrijf, om zo mijzelf te verzekeren van een vast inkomen. Tot mijn blijdschap heeft nu de TROS mij een aanbod gegeven. Na mijn afstuderen zal de TROS mij voor 3 dagen in de week als bedrijf in te huren. Een mooiere afsluiting van mijn studie Mediatechnologie kan ik mij niet bedenken.

15. Bronnen

Geraadpleegde Sites:

<http://www.swfmill.org>
<http://www.xleff.org>
<http://www.adobe.com>
<http://labs.adobe.com>
<http://ant.apache.org>
<http://www.eclipse.org>
<http://www.flashdevelop.org>
<http://sourceforge.net/projects/sepy>
<http://ming.sourceforge.net>
<http://www.mtasc.org>
<http://www.aswing.org>
<http://www.flash-remoting.com>
<http://amfphp.sourceforge.net>
<http://www.themidnightcoders.com>
<http://www.ghostwire.com>
<http://www.openwddx.org>
<http://www.swxformat.org>
<http://www.flashblocks.com>
<http://www.fcmspro.com>
<http://www.red5.nl>
<http://www.haxevideo.org>
<http://www.silverlight.net>

Geraadpleegde Literatuur:

Object-Oriented ActionScript for Flash 8
Peter Elst and Todd Yard
ISBN: 1-59059-619-6
friendsofed.com

Actionscript 2.0 en 3.0 handboek
Peter Kassenaar
ISBN: 90-5940-243-x
van DUUREN media

Dynamische websites met PHP
Ward van der Put
ISBN: 90-456-3763-4

Flash Player 8 Security White Paper
Adrian Ludwig
http://www.macromedia.com/devnet/flashplayer/articles/flash_player_8_security

16. Bijlagen.

16.1. Bijlage - Plan van aanpak

16.1.1. Activiteiten en Fasering

Orientatie fase:

- Wat kan een flash content systeem voor meerwaarde brengen?
- Wat kan het beheren van op flash gebaseerde websites voor meerwaarden hebben?

Onderzoekfase:

- Onderzoek naar (nieuwe) flash ontwikkel omgevingen, tools methodes en technieken.
- Onderzoeken naar bruikbaarheid van nieuwe technieken?
- Onderzoek naar veiligheid, schaalbaarheid, snelheid en belasting??
- Onderzoek naar meerwaarde voor de TROS?
- Onderzoek naar bruikbare mogelijkheden voor de TROS?

Analyse:

- Analyse Onderzoekfase
- Keuzes maken
- Doornemen gemaakte keuzes

Ontwerpfase:

- Maken functionele Specificatie aan de hand van de analyse uitkomst
- Makken Technische Specificatie aan de hand van de analyse uitkomst
- Maken benodigde dataflow diagrammen

Ontwikkelfase:

- Ontwikkelen van Flash Content Systeem Prototype voor Flash gebaseerde websites.
- Ontwikkelen van Beheerbare Flash website
- Toepassen van nieuwe technieken

Testfase:

- Stabiliteit
- Snelheid
- Veiligheid
- Schaalbaarheid
- Computer belasting
- Bandbreedte

16.1.2. Tijdsplanning

- Introductie
Week 5 (29 Jan t/m 2 Feb)
- Oriëntatiefase
Week 6 t/m 7 (5 Feb t/m 16 Feb)
- Onderzoekfase
Week 8 t/m 10 (19 Feb t/m 30 Mrt)
- Analysefase
Week 14 t/m 15 (2 Apr t/m 13 Apr)
- Ontwerpfase
Week 16 t/m 19 (16 Apr t/m 11 Mei)
- Ontwikkelfase
Week 11 t/m 20 (14 Mei t/m 8 Juni)
- Testfase
Week 22 t/m 25 (11 juni t/m 22 juni)

16.1.3. Schema

	MA	DI	WO	DO	VRIJ	Fase:	Periode:	Uren
Week 5						Introductie	29 Jan t/m 2 Feb	40
Week 6						Oriëntatie	5 Feb t/m 16 Feb	80
Week 7								
Week 8						Onderzoek	19 Feb t/m 30 Mrt	240
Week 9								
Week 10								
Week 11								
Week 12								
Week 13								
Week 14						Analyse	2 Apr t/m 13 Apr	80
Week 15								
Week 16						Ontwerp	16 Apr t/m 11 Mei	160
Week 17								
Week 18								
Week 19								
Week 20						Ontwikkeling	14 Mei t/m 8 Juni	160
Week 21								
Week 22								
Week 23								
Week 24						Testen	11 Juni t/m 22 Juni	80
Week 25								
Totaal:								840

16.2. Bijlage – Ming Objecten

SWFMovie

Maakt een nieuwe flash movie

SWFShape

Maakt een nieuw shape object waarmee men lijnen kan teken en vormen mee kan inkleuren.

SWFDisplayItem

Maakt een nieuw displayitem object . Nadat je een shape, text object een sprite of een button hebt gedefinieerd kan je het aan de flashmovie toevoegen en zichtbaar maken via het SWFDisplayItem object.

SWFGradient

Maakt een SWFGradient object waarmee shape fills met een gradient (kleurverloop) kunnen worden gevuld

SWFBitmap

Maakt een SWFBitmap object waarmee een bitmap fill aan een shape kan worden toegevoegd. Je kan jpg's en dbl files toevoegen. Dbl files zijn png paatjes die met de png2dbl tools zijn omgezet. Dit omdat Ming nog niet overweg kan met png.

SWFFill

Maakt een SWFFill object waarmee shapes kunnen worden gevuld met solid ,gradient en bitmap fills.

SWFMorph

Maakt een SWFMorph object aan waarmee je shape tweens tussen 2 shapes kan maken.

SWFText

Maakt een SWFText object aan waarmee text aan de movie kan worden toegevoegd en worden gemanipuleerd.

SWFFont

Maakt een SWFFont object waarmee je een font kan definiëren via een FDB ("font definition block") of browser afhankelijke font referentie.

SWFTextField

Maakt een SWFTextField object, het SWFTextField object is minder flexibel als het SWFText object, maar kan wel worden gebruikt in formulieren.

SWFSprite

Maakt een SWFSprite object. Dit komt overeen met een MovieClip op de stage in flash. Elke Sprite heeft zn eigen tijdslijn en veel methodes die een MovieClip in

flash ook bezit.

SWFButton

Maakt een SWFButton object inclusief rollover.

SWFAction

Maakt een SWFAction object waarin scripts kunnen worden uitgevoerd welke weer een waarde terug geeft aan het SWFAction object.

16.3. Bijlage – Ming Installatie en voorbeeld Code

De installatie van Ming bestaat uit 2 stappen:

- 1: de Ming Library installeren (de dll's)
- 2: de php.ini-dist uit de distributie hernoemen naar php.ini en de `;extension=php_ming.dll` regel uitcommentariëren.

Met het volgende stukje actionscript wordt een blokje in de flashmovie getoont:

```
_root.createEmptyMovieClip( 'triangle', 1 );  
with ( _root.triangle ) {  
  lineStyle( 5, 0xff00ff, 100 );  
  moveTo( 200, 200 );  
  lineTo( 300,300 );  
  lineTo( 100, 300 );  
  lineTo( 200, 200 );  
}
```

Dit zelfde blokje kan door een Ming gecreëerde flash movie met PHP worden gemaakt door de volgende code:

```
<?ming_setScale(20.00000000);  
ming_useswfversion(6);  
$movie = new SWFMovie();  
$movie->setDimension(550,400);  
$movie->setBackground(0xcc, 0xcc, 0xcc );  
$movie->setRate(31);  
$strAction = "  
_root.createEmptyMovieClip( 'triangle', 1 );  
with ( _root.triangle ) {  
  lineStyle( 5, 0xff00ff, 100 );  
  moveTo( 200, 200 );  
  lineTo( 300,300 );  
  lineTo( 100, 300 );  
  lineTo( 200, 200 );}  
";  
$movie->add(new SWFAction(str_replace("\r", "", $strAction)));  
$movie->save("trianglemx.swf");  
? >
```

16.4. Bijlage – Flashvar voorbeeld

```
var mijnBedrijf:String;

mijnData = new LoadVars();
mijnData.load("Tekst.txt");
mijnData.onLoad = function (succes){
    if(succes){
        mijnBedrijf = this.bedrijf;
    }else{
        Trace("fout tijdens laden");
    }
}
```

Dit stukje Actionscript zal de waarde 'TROS' die bij de naam 'bedrijf' hoort uit het 'Tekst.txt' bestand halen en in de variabele 'mijnBedrijf' zetten. Mocht er tijdens het laden van de 'Tekst.txt' bestand een fout optreden, dan komt er een melding in de output window van de Flash IDE.

De PHP code voor database connectie:

```
<?php
    $connect = mysql_connect("localhost","root","root");
    mysql_select_db("tros", $connect);
    $result = mysql_query("SELECT Naam, Plaatje FROM stagiaires");
    $teller = 0;
    While ($row=mysql_fetch_array($result)){
        echo "Naam$teller=$row[Naam]&Plaatje$teller=$row[Plaatje]&";
        $teller++;
    }
    Echo "teller=$teller";
?>
```

Wat opvalt is dat er een teller wordt gebruikt die achter elke naam wordt gezet. Deze teller loopt in de lus op. Wanneer er meerdere rijen in de database tabel 'stagiaires' staan, kan Flash dankzij de teller onderscheid maken.

16.5. Bijlage – Flashvars laden uit database met een server side script

Aan de Flash kant lijkt de code erg op de code om een tekst bestand in te lezen.:

```
mijnData = new LoadVars();
mijnData.ref = this;
mijnData.load("flash.php");
mijnData.onLoad = function (succes){
    if(succes){
        for(i=0; i<this.teller; i++){
            this.ref["mijnNaam"+i] = this["Naam"+i];
            this.ref["mijnPlaatje"+i] = this["Plaatje"+i];
        }else{
            Trace("fout tijdens laden");
        }
    }
}
```

Het enige verschil is dat er nu een paar keer met een lus door de data heen wordt geloopt. De teller wordt gebruikt om het aantal loops aan te geven. Er ontstaat nu een dynamisch aantal variabelen, beginnend bij mijnNaam0, mijnPlaatje0, mijnNaam1, etc.

Het opslaan van data zal altijd via een server script moeten gebeuren omdat Flash niet direct een tekstbestand of database kan benaderen voor updates.

Met de 'sendAndLoad()' functie in flash kunnen name/value waardes terug worden gestuurd naar een PHP bestand. Deze waardes worden vervolgens weggeschreven in een tekstbestand of database.

```
mijnData.sendAndLoad("save.php",mijnData,"post");
```

16.6. Bijlage – RED 5 Actionscript code

Voorbeeld van stukje Actionscript dat een stream afspeelt:

```
var mijnVideo:Video;  
var mijn_nc:NetConnection = new NetConnection();  
var connected:Boolean = mijn_nc.connect("rtmp://localhost/myapp");  
var mijn_ns:NetStream = new NetStream(mijn_nc);  
mijnVideo.attachVideo(mijn_ns);  
mijn_ns.play("flvName");
```

Localhost is hier de RED5 server.

16.7. Bijlage - RTMP

Real Time Messaging Protocol (RTMP) Is een protocol ontwikkeld door Adobe Systems (voorheen Macromedia) dat voornamelijk word gebruikt door Macromedia Flash Media Server om audio en video over het internet te streamen naar een Flash Player client, maar het protocol kan ook worden gebruikt voor remote procedure calls (RPC). Het RTMP protocol maakt gebruik van een binaire tcp connectie of een polling http tunnel. Het protocol is een container voor data pakketjes in amf en raw audio / video data zoals in een flv (flashvideo). Een enkele connectie kan vele net streams multiplexen doorverschillende kanalen te gebruiken. In deze kanalen zijn de pakketjes opgedeeld in delen van vaste grote.

De standaard RTMP poort is 1935. Omdat veel bedrijfsnetwerken en mensen thuis een firewall hebben en deze poort dicht hebben staan is het aanbevolen om te streamen via HTTP, port 80. Afhankelijk van de soort applicatie kan hier wel of niet voor worden gekozen.

16.8. Bijlage – Voorbeeld Crossdomain.xml

Voorbeeld van een crossdomain.xml:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy
  SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="www.tros.nl" />
  <allow-access-from domain="tros.nl" />
  <allow-access-from domain="eenvandaag.nl" />
  <allow-access-from domain="www.eenvandaag.nl" />
  <allow-access-from domain="ad.nl.doubleclick.net" />
</cross-domain-policy>
```


16.9. Bijlage – FlashLive pagina beschrijving voor gebruik met SWFMill

Een voorbeeld van een XML pagina beschrijving voor SWFMill, gecreëerd in de FlashLive backend:

```
//xml versie en encoding
<?xml version="1.0" encoding="iso-8859-1" ?>

// Page eigen schappen
<movie width="800" height="600" framerate="31">
<background color="#ffffff"/>

<frame name="clenn">
// Library met referentie van de geplaatste componenten
  <library>
// Er wordt één layout panel en één statischcontentblok gebruikt
    <clip id="panel1" import="panels/panel1.swf"/>
    <textfield id="statischcontentBlok" width="300" height="60"
      size="12" font="vera" text="Clenn is een koning!"/>
  </library>

//panel1 en statischcontentBlok kunnen nu meerdere keren op de stage worden
//geplaatst met een oplopen de variabele in de naam

    <place id="panel1" name="newpanel_1" x="10" y="10" depth="1"/>
    <place id="panel1" name="newpanel_2" x="120" y="10" depth="2"/>
    <place id="statischcontentBlok" name="contentblok_1" x="130" y="20"
      depth="100"/>

  </frame>
</movie>
```

SWFMill zal na het compilen van deze XML beschrijving een witte pagina creëren met daarop 2 pannels en één contentblok die de inhoud 'Clenn is koning!' bezit.