

---

# Diplomarbeit

im Studiengang Audiovisuelle Medien  
Fachbereich Electronic Media

Titel:

## **Entwicklung einer Networking- und Vertriebsplattform für Recordlabels mit dem Schwerpunkt Public-Frontend-Funktionen**

vorgelegt von

**Gerold Scholte-Meyerink**

an der Fachhochschule Stuttgart – Hochschule der Medien

am 19. April 2006

1. Prüfer: Prof. Uwe Schulz

2. Prüfer: Dipl.-Ing. (BA) MBA Steffen Schachtler

---

# Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

---

Ort, Datum

---

Unterschrift

---

# Kurzfassung

Die vorliegende Arbeit behandelt die theoretischen Hintergründe zur Entwicklung einer mehrsprachigen Webanwendung mit Frontend-Funktionen für so genannte Independent Recordlabels. Die entwickelte Webanwendung vereinfacht die Marketing-Prozesse insbesondere hinsichtlich der Bereitstellung von Information für unterschiedliche Nutzergruppen.

Die entwickelte Flash-Anwendung ist dreifach dynamisch: mittels ihrer Hilfe werden dynamisch generierte Inhalte aus der zentralen Recordlabel-Datenbank in ansprechender dynamischer Form präsentiert, der Layoutaufbau erfolgt dabei während der Laufzeit nach Vorgaben einer leicht zu modifizierenden Layout-Klasse.

Im Speziellen wird in dieser theoretischen Ausarbeitung auf die Problematiken der Entwicklung einer sicheren Flash-Anwendung und eines sicheren Prelistening-Konzeptes sowie auf Usability-Aspekte eingegangen. Erläutert wird zudem, wie die unterschiedlichen Benutzergruppen in das Site-Konzept miteinbezogen wurden.

**Keywords:** Recordlabel-Software, Flash, Prelistening-Konzept, Rich Internet Application, sichere Flash-Anwendung, Usability

# Abstract

This elaboration treats the theoretical background for the development of a multilingual web-application with front-end functions for independent record labels. The developed web-application simplifies the marketing processes regarding the supply of information for different groups of users.

The developed Flash application is three-way dynamic: dynamically generated contents from the central record label data base are presented in an attractive dynamic way, thereby the structure of the application is built at run time based on a layout class which can be modified easily.

This theoretical elaboration especially deals on the problems of the development of a secure Flash application and a secure prelistening concept as well as on usability aspects. It describes, how the different user groups were referred into the site-concept.

**Keywords:** Record Label Software, Flash, Prelistening Concept, Rich Internet Application, Secure Flash Applications, Usability

---

# Abkürzungsverzeichnis

bzw.	beziehungsweise
CD	Compact Disc
CSV	Character Separated Values <i>oder</i> Comma Separated Values
DHTML	Dynamitc Hypertext Markup Language
DJ	Disc Jockey
E-Mail	Electronic Mail
f.	folgende
ff.	fortfolgende
FLA	Flash-Datei (bezieht sich auf die Dateieindung .fla)
FTP	File Transfer Protocol
GEMA	Gesellschaft für musikalische Aufführungs- und mechanische Vervielfältigungsrechte
GIF	Graphics Interchange Format
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	Identifikationsnummer
IP	Internet Protocol
JPG/JPEG	Joints Photographic Experts Group
MD5	Message Digest Algorithm, Version 5
MP3	ISO MPEG Audio Layer 3
MPEG	Motion Picture Experts Group
PHP	PHP Hypertext Preprocessor
PNG	Portable Network Graphics
RTP	RealTime Transfer Protocol
RTSP	RealTime Streaming Protocol
SDK	Software Development Kit
SHA1	Secure Hash Algorithm, Version 1
SQL	Structured Query Language
SWF	Small Web Format
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
vgl.	vergleiche
VUT	Verband unabhängiger Tonträgerhersteller
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

---

# Abbildungsverzeichnis

Abbildung 3.1	Schematische Darstellung der Online-Marketing-Abläufe in bisheriger Form und mit der entwickelten Recordlabel-Software .....	15
Abbildung 4.1	Schematische Darstellung des Zusammenspiels der vier eingesetzten Techniken .....	22
Abbildung 5.1	Die Linkstruktur des Frontends .....	34
Abbildung 5.2	Layout-Raster .....	35
Abbildung 5.3	Darstellung auf einem Monitor mit einer Auflösung von 800 x 600 Pixel .....	36
Abbildung 5.4	Der persönliche Bereich „MyPulver“ mit einer persönlichen Playlist .....	50
Abbildung 5.5	Der Bestellvorgang für ein White-Label .....	51
Abbildung 5.6	Self-Service: Neuanlegen eines Auftritt-Termins .....	52
Abbildung 5.7	Datei-Auswahl-Dialog beim Upload eines Demo-MP3s .....	53
Abbildung 5.8	Hilfestellungen zu den Formulareingaben bei der Registrierung .....	60
Abbildung 5.9	Aufistung der Suchergebnisse .....	61
Abbildung 6.1	Startseite .....	66
Abbildung 6.2	Musikabspiel-Modul mit Plattenspieler-Animation .....	67
Abbildung 6.3	Detailansicht eines Künstlerfotos .....	68
Abbildung 6.4	Das Künstler-Profil .....	69
Abbildung 6.5	Diskografie eines Künstlers .....	70
Abbildung 6.6	Informationen zu einem Release .....	71
Abbildung 6.7	Detailansicht für einen ausgewählten Titel eines Releases .....	72

---

# Vorwort

Am Anfang stand die Unzufriedenheit über die Leistungsmöglichkeiten der Software-Produkte für kleinere Recordlabels – die fehlende Schnittstelle für eine Website, das Gebundensein an einen bestimmten Rechner. Dies veranlasste Robin Hofmann vom Stuttgarter Recordlabel Pulver Records und die Firma topazmedia in Ludwigsburg dazu, die Entwicklung einer neuartigen Recordlabel-Software in Angriff zu nehmen.

In enger Kooperation mit Julian Servatius und seiner Diplomarbeit mit dem Titel „Entwicklung einer Networking- und Vertriebsplattform für Recordlabels mit dem Schwerpunkt Office-Funktionen“ entstand so eine völlig neuartige, wesentlich flexiblere Recordlabel-Software.

Die hier vorliegende Arbeit behandelt dabei den Teil mit der so genannten Public-Frontend-Funktion. Das Ergebnis ist vergleichbar mit einem komplexen Webauftritt, der auf ein mächtiges Backend zurückgreift. Die entwickelte Webanwendung ist jedoch mehr. Sie ist modular so aufgebaut, dass sie als Grundgerüst für viele andere Webauftritte mit völlig anderem Layout dienen kann. Sie kann jedoch mit recht wenig Aufwand auch in kleinerem Rahmen gestalterisch modifiziert werden. So ist es möglich die Recordlabel-Software als günstiges Gesamtpaket anzubieten, in dem ein umfangreicher Webauftritt gleich mitenthalten ist.

Diese theoretische Ausarbeitung soll die Rahmenbedingungen erläutern, unter der die Anwendung erstellt wurde, und die Probleme und Aufgaben, die bei der Umsetzung zu bewältigen waren.

Mit Robin Hofmann hatten wir bei der Umsetzung einen fachkundigen Helfer zur Hand, der uns die Arbeitsabläufe in einem so genannten Independent Recordlabel näher brachte und uns mit Rat und Tat zur Seite stand.

Der praktische Teil dieser Diplomarbeit ist vorübergehend im Internet unter der Adresse <http://labels.topazmedia.de/frontend/> einsehbar.

# Inhaltsverzeichnis

Erklärung.....	ii
Kurzfassung.....	iii
Abstract .....	iii
Abkürzungsverzeichnis .....	iv
Abbildungsverzeichnis .....	v
Vorwort .....	vi
Inhaltsverzeichnis .....	vii
<b>Kapitel 1 Überblick.....</b>	<b>1</b>
1.1 Das neue reichhaltige Internet.....	2
1.2 Abgrenzung des Themas .....	4
<b>Kapitel 2 Ziele.....</b>	<b>5</b>
2.1 Vereinfachung des Marketing-Prozesses.....	6
2.1.1 Kommunikation mit den Kunden .....	6
2.1.2 Komfortable Bereitstellung von Information .....	7
2.2 Direkte Brücke zwischen Recordlabel-Daten (Backend) und Frontend .....	9
2.3 Sicheres Prelistening-Konzept .....	10
<b>Kapitel 3 Bisheriger Stand der Technik.....</b>	<b>11</b>
3.1 Spezielle Recordlabel-Software .....	12
3.1.1 Officer.....	12
3.1.2 essCore .....	13
3.1.3 go_disko* .....	13
3.2 Individuelle Einzelanfertigungen für Recordlabels .....	14
3.3 Prelistening-Verfahren .....	16
3.3.1 Rechtliche Gegebenheiten.....	16
3.3.2 Technische Gegebenheiten.....	17
3.3.2.1 Einfache Verlinkung und Forced Download.....	17
3.3.2.2 Direkte Einbettung in HTML.....	18
3.3.2.3 Progressive Download.....	18
3.3.2.4 Streaming .....	18
3.3.2.5 Der Mittelweg: Progressive Download mit Datei-Header-Modifikation.....	20

<b>Kapitel 4</b>	<b>Eingeschlagener Realisierungsweg</b>	<b>21</b>
<b>4.1</b>	<b>Die magischen Vier</b>	<b>22</b>
<b>4.2</b>	<b>Clientseitig: Macromedia Flash</b>	<b>23</b>
4.2.1	Vor- und Nachteile	23
4.2.1.1	Vorteile	24
4.2.1.2	Nachteile	25
4.2.2	Gründe für die Verwendung von Macromedia Flash	27
4.2.2.1	Technische Gründe	27
4.2.2.2	Werbliche Gründe	28
<b>4.3</b>	<b>Serverseitig: PHP und MySQL</b>	<b>30</b>
<b>4.4</b>	<b>Die Brücke: XML</b>	<b>31</b>
<b>Kapitel 5</b>	<b>Die Umsetzung</b>	<b>32</b>
<b>5.1</b>	<b>Grundgedanken und Layout</b>	<b>33</b>
5.1.1	Linkstruktur	33
5.1.2	Layout	35
<b>5.2</b>	<b>Ein dynamisches Template in Flash (Layout-Klasse)</b>	<b>37</b>
5.2.1	Bedeutung für die Programmierung der Anwendung	38
<b>5.3</b>	<b>Sicherheitsaspekte in Flash-Anwendungen</b>	<b>39</b>
5.3.1	Möglichkeiten und Grenzen „sicherer“ Anwendungen in Macromedia Flash	39
5.3.1.1	Einwegverschlüsselung mit MD5 und SHA1	40
5.3.1.2	Grenzen und Risiken	41
5.3.2	Umsetzung im Projekt	42
5.3.2.1	Session-Verwaltung	42
5.3.2.2	Login	42
5.3.2.3	Beschränkungen bei Benutzer-Eingaben	43
5.3.2.4	Dateiupload	43
<b>5.4</b>	<b>Unechtes Streaming – Downloadschutz zur Wahrung der Urheberrechte</b>	<b>45</b>
5.4.1	Problematiken	45
5.4.2	Lösungsansatz und Umsetzung	46
<b>5.5</b>	<b>Miteinbeziehen der Benutzer</b>	<b>49</b>
5.5.1	Personalisierte Angebote	49
5.5.2	DJ-Previews	51
5.5.3	Self-Services	52
5.5.4	Demo-Upload	53
<b>5.6</b>	<b>Usability</b>	<b>56</b>
5.6.1	Flash-typische Usability-Problematiken	56
5.6.2	History	57

5.6.3	Strukturierung zur besseren Orientierung.....	58
5.6.4	Hilfestellungen.....	59
5.6.5	Suche und Suchergebnisse .....	60
<b>5.7</b>	<b>Die XML-Brücke .....</b>	<b>62</b>
<b>5.8</b>	<b>Mehrsprachigkeit.....</b>	<b>64</b>
<b>Kapitel 6</b>	<b>Präsentation der Anwendung.....</b>	<b>65</b>
<b>6.1</b>	<b>Startseite.....</b>	<b>66</b>
<b>6.2</b>	<b>Musikspieler .....</b>	<b>67</b>
<b>6.3</b>	<b>Informationsbereich.....</b>	<b>68</b>
6.3.1	Informationen zu einem Künstler .....	69
6.3.2	Informationen zu einem Release .....	70
<b>Kapitel 7</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>73</b>
<b>7.1</b>	<b>Zusammenfassung .....</b>	<b>74</b>
<b>7.2</b>	<b>Ausblick.....</b>	<b>75</b>
7.2.1	Erweiterungen der entwickelten Anwendung.....	75
7.2.2	Rich-Internet-Anwendungen mit Flash in der Zukunft .....	75
	<b>Anhang .....</b>	<b>77</b>
	<b>Inhalt der CD-ROM .....</b>	<b>78</b>
	<b>Glossar .....</b>	<b>80</b>
	<b>Literaturverzeichnis .....</b>	<b>83</b>

# Kapitel 1

## Überblick

---

Das neue reichhaltige Internet

Abgrenzung des Themas

# 1.1 Das neue reichhaltige Internet

*„Breitbandleitungen werden immer leistungsfähiger und Ideen werden fortschrittlicher... Die Zeit ist reif für das Web, aus seinen Kinderschuhen herauszuwachsen und ein höheres Entwicklungsstadium zu erreichen.“  
(Ford 2006)*

Große Worte, die Rob Ford, der Gründer der Favourite Website Awards, da von sich gibt. Aber tatsächlich ist momentan eine Bewegung im Internet festzustellen, die völlig neue Möglichkeiten zu entdecken scheint.

Zwei Begriffe werden in diesem Zusammenhang häufig genannt: Zum einen die von Dale Dougherty und Tim O'Reilly ins Leben gerufene Bezeichnung Web 2.0 (vgl. Kaul 2005) und auf der anderen Seite die so genannten Rich Internet Applications. Beide Begriffe beschreiben keine wirklich neue Technik an sich. Es sind vielmehr die neuartigen Nutzungsweisen vorhandener Technik.

Der Begriff Web 2.0 bezieht sich dabei vor allem auf die Bedeutung des Internets für den Benutzer. Das Jahr 2005 hat hier viel Neues ins Rollen gebracht. Der Internet-Nutzer von heute nutzt das Internet nicht mehr allein zu Informationszwecken (und selbst das tut er heute auf viel individuellere Art und Weise), sondern nutzt es vermehrt auch für seine privaten Belange. Er lässt andere Menschen in aller Welt an seinem Leben teilhaben durch Weblogs oder durch Foto- und Videoforen wie YouTube oder durch Teilnahme an der freien Enzyklopadie Wikipedia. Der Internet-Benutzer von heute und morgen partizipiert aktiv am Internet.

Web 2.0 steht auch für die Entwicklung, dass der eigene Computer-Desktop mehr und mehr vernetzt ist mit dem Internet. Apple war hier Vorreiter mit seinen so genannten Widgets, die Informationen aus dem Internet wie zum Beispiel aktuelle Börsenkurse oder Wetterdaten in kleinen Mini-Anwendungen auf dem Desktop anzeigen können.

Auf der anderen Seite gibt es die Rich Internet Applications, die ebenfalls auf Techniken beruhen, die es schon lange gibt, wie zum Beispiel Macromedia Flash, Java oder JavaScript. Im Gegensatz zum Web 2.0 gibt es hier jedoch einen einleuchtenderen Grund, warum sie erst jetzt zum Zuge kommen: die zunehmende Anzahl der Breitband-Verbindungen im Internet.

Rich Internet Applications sind Webanwendungen, die clientseitig Berechnungen durchführen können. Das spart Zugriffszeit und ermöglicht eine sehr schnelle Reaktion auf Benutzeranfragen. Allerdings sind diese Rich Internet Applications als solche größer als reine HTML-Lösungen und be-

nötigen daher längere Ladezeiten. Sie bieten jedoch völlig andere Interaktions- und Visualisierungsmöglichkeiten (vgl. wikipedia rich\_internet\_application 2006).

Da der Benutzer Rich Internet Applications nicht extra zu installieren braucht und sie überall auf jedem Rechner mit Internetverbindung und passender Technologie (zum Beispiel Browser und Flash-Plugin) sofort einsatzbereit sind, besitzen sie das Potential, klassische Software-Anwendungen in Zukunft zu ersetzen.

## 1.2 Abgrenzung des Themas

Die vorliegende Arbeit ist die theoretische Begleitschrift zur praktischen Umsetzung einer Networking- und Vertriebsplattform für Recordlabels mit Frontend-Funktion. Sie ist der zweite Teil einer Gesamtarbeit, die aus zwei Teilen besteht. Der erste Teil wurde zeitgleich unter dem Titel „Entwicklung einer Networking- und Vertriebsplattform für Recordlabels mit dem Schwerpunkt Office-Funktionen“ von Julian Servatius angefertigt.

Die hier vorliegende schriftliche Ausarbeitung soll die drei wesentlichen Ziele der Frontend-Anwendung und die theoretischen Hintergründe – Konzeption und Umsetzung betreffend – verdeutlichen.

Die drei wesentlichen Ziele der Frontend-Anwendung sind zum einen die Vereinfachung des Marketing-Prozesses für so genannte Independent Recordlabels, insbesondere die komfortable Bereitstellung von Release- und Künstler-Informationen, zum anderen die Möglichkeit, diese Informationen direkt aus den Arbeitsdaten des Recordlabels zu generieren und als weiteres wesentliches Ziel die Umsetzung eines sicheren aber günstigen Prelistening-Verfahrens. In Kapitel 2 werde ich diese Ziele genauer erläutern.

Kapitel 3 behandelt den heutigen Stand der Technik: In diesem Kapitel werde ich genauer auf vorhandene Software-Lösungen für Recordlabels eingehen, die Vorteile und Nachteile von individuell erstellten Einzelanfertigungen diskutieren und die rechtlichen und technischen Gegebenheiten für Prelistening-Verfahren erörtern.

Kapitel 4 beschreibt den eingesetzten Realisierungsweg und die dazu hauptsächlich verwendeten Techniken Macromedia Flash, MySQL, PHP und XML.

In Kapitel 5 werde ich darlegen, welche Problematiken bei der Umsetzung auftauchten und wie sie gelöst wurden. Zudem werden wichtige theoretische Hintergründe erläutert, die bei der Umsetzung einer solchen Anwendung zu beachten sind – wie zum Beispiel Sicherheits- und Usability-Aspekte.

Das Ergebnis der Arbeit werde ich Kapitel 6 in Teilen genauer präsentieren. Hier werde ich vor allem die Teile der Anwendung besprechen, die in den vorangegangenen Kapiteln noch nicht genauer beleuchtet wurden, wie zum Beispiel das Musik-Player-Modul in der Flash-Anwendung.

Kapitel 7 gibt eine abschließende Zusammenfassung und einen kurzen Ausblick auf die Zukunft der entwickelten Webanwendung und den zukünftigen Einsatz von so genannten Rich-Internet-Anwendungen mit Flash.

# Kapitel 2

## Ziele

---

Vereinfachung des Marketing-Prozesses

Direkte Brücke zwischen Label-Daten  
und Frontend

Sicheres Prelistening-Konzept

## 2.1 Vereinfachung des Marketing-Prozesses

*Ziel der entwickelten Webanwendung soll es sein, einen Großteil der Marketing-Aufgaben eines Recordlabels zu vereinfachen, im besonderen hinsichtlich der Bereitstellung von Produktinformationen und der Kommunikation zwischen Kunden und Recordlabel.*

Über die Webanwendung sollen alle Informationen zu einem Release und zu den Künstlern abrufbar sein, die für bestimmte Interessenten wichtig sein könnten.

Dabei soll die Möglichkeit bestehen, dass nicht jeder Nutzer des Webauftrittes jede Information lesen oder ändern kann. So soll zum Beispiel nur der Vertrieb auch Informationen darüber bekommen, welche Releases in den nächsten Monaten erscheinen. Der „normale Besucher“ bekommt jedoch nur die Releases aufgelistet, die tatsächlich schon erschienen sind. Der Webauftritt soll also nicht nur in erster Linie eine klassische Webpräsenz sein, sondern darüber hinaus auch Anlaufstelle für Presse und Rundfunk, für DJs, Musiker und deren Manager und für die direkten Kunden des Recordlabels (Vertrieb).

Künstler oder deren Manager sollen die Möglichkeit besitzen, über das Frontend ihre betreffenden Informationen selbst zu pflegen und zu ergänzen. So kann ein Künstler zum Beispiel seine nächsten Tour-Dates bekannt geben oder seine Biografie auf den neuesten Stand bringen, ohne dass sich ein Mitarbeiter des Recordlabels speziell darum kümmern muss.

**Unterschiedliche Rechte  
für die einzelnen Nutzer**

### 2.1.1 Kommunikation mit den Kunden

Ein entscheidender Punkt soll die Möglichkeit sein, jedem angemeldeten Benutzer persönliche Playlists zukommen und ihn an Previews mit Bewertungsfunktion („DJ-Tests“) teilnehmen zu lassen. Hier bietet sich ein großes Einsparpotenzial, da zurzeit noch bestimmte Tracks vor dem eigentlichen Erscheinungstermin in Form von so genannten White-Labels an bestimmte DJs auf dem Postweg verschickt werden. Der Versand ist entsprechend kostspielig, lohnt sich jedoch, wenn aussagekräftiges Feedback von den DJs zurückkommt. Genau hier liegt jedoch das Problem: Zu wenig DJs beteiligen sich aktiv am Feedback-Prozess. Dies soll durch die Webanwendung anders werden.

**Feedback**

Die Zuordnung der tatsächlich eingetroffenen Feedbacks zu den einzelnen White-Labels und wiederum den einzelnen Tracks stellt einen gewichtigen Organisationsaufwand dar.

Hier soll die entwickelte Webanwendung große Entlastung bringen. Die entsprechenden DJs werden per E-Mail darüber benachrichtigt, dass eine neue Preview für sie im zugangsgeschützten Bereich des Webauftrittes vorhanden ist. Nachdem sich die DJs eingeloggt haben, können sie hier schon mal alle vorhandenen Infos zu den einzelnen Tracks einsehen, ihn anhören und ein Statement und eine Bewertung abgeben. Zudem können sie ein klassisches White-Label anfordern, um die Musik auch im Club bei einem DJ-Set testen zu können.

Die Vorteile, die sich dadurch dem Recordlabel ergeben sind:

- ein direkteres Feedback, das jedem Track schon automatisch zugeordnet ist; also weniger Verwaltungsaufwand
- Einsparung von überflüssiger Produktion und unnötigem Versand von White-Labels an DJs, die an der in dieser Preview zusammengestellten Musik kein Interesse haben oder die sich kaum beziehungsweise gar nicht am Feedback-Prozess beteiligen.

Künstler und Künstleranwerter sollen über die Webanwendung Demo-Tracks hochladen können. Der Recordlabel-Mitarbeiter kann sich diese Tracks über das Backend dann sofort anhören und ein Urteil fällen. Der riesige Wust an Demo-Einsendungen (bisher zumeist über den Postweg) soll somit wesentlich übersichtlicher und besser handhabbar werden.

### Demo-Einsendungen

## 2.1.2 Komfortable Bereitstellung von Information

Ein großes Anliegen von Seiten des Auftraggebers<sup>1</sup> war es, auf dem Webauftritt wirklich alle Informationen zu den einzelnen Releases, den einzelnen Tracks und den Künstlern zugänglich zu machen. Bei eventuellen Nachfragen nach Release-Informationen oder Informationen zu bestimmten Künstlern kann so einfach auf den Webauftritt verwiesen werden.

Für Pressevertreter ist dabei kein eigener Pressebereich nötig. Journalisten können sich auf der Seite als Pressevertreter anmelden, wenn sie weitergehende Informationen wünschen, und bekommen dann in der schon gewohnten Umgebung für sie zusätzlich relevante Informationen und Materialien. Das

<sup>1</sup> Als Auftraggeber wird hier das Recordlabel bezeichnet, das die ursprüngliche Idee zur Entwicklung einer entsprechenden Webanwendung ins Spiel gebracht hat. Angedacht ist allerdings schon, die Webanwendung so allgemein wie möglich zu entwickeln, um sie auch weiteren Recordlabels anbieten zu können.

gleiche gilt für Nutzer aus dem Bereich des Vertriebs, des Künstlermanagements und für besondere Kundengruppen<sup>2</sup>.

Wichtig ist in diesem Zusammenhang, dass trotz der Informationsfülle die Übersicht gewahrt bleiben soll. Detaillierte Informationen sollen erst dann sichtbar werden, wenn sie der Nutzer auch tatsächlich wünscht. Das heißt: Ein Nutzer, der schlicht nur daran interessiert ist, kurz in das Musikprogramm des Recordlabels hineinzuhören, soll gar nicht erst durch für ihn überflüssige und somit verwirrende Informationen abgelenkt werden. Sollte jedoch sein Interesse nach dem Anhören eines bestimmten Tracks geweckt worden sein, soll er trotzdem sehr schnell auf weitere Informationen zugreifen können. Das gilt durchgängig in der gesamten Webanwendung (also auch für die persönlichen Playlists und Previews im zugangsgeschützten Bereich).

Zudem soll die Möglichkeit bestehen, bestimmte Künstler und Releases besonders hervorzuheben. Künstler von besonderem Interesse können so speziell promotet werden. Besucher des Webauftrittes werden schon auf der Startseite auf neue Releases aufmerksam gemacht.

**Vermeidung von  
Unübersichtlichkeit**

<sup>2</sup> Besondere Kundengruppen sind zum Beispiel Interessenten, die die Musik für kommerzielle Zwecke (Filmmusik, Verwendung für Werbespots oder spezielle Events) nutzen wollen. Für sie kann eine eigene Playlist zusammengestellt werden.

(Vgl. Julian Servatius: „Entwicklung einer Networking- und Vertriebsplattform für Recordlabels mit dem Schwerpunkt Office-Funktionen“, Seiten 15 f.)

## 2.2 Direkte Brücke zwischen Recordlabel-Daten (Backend) und Frontend

*Durch die direkte Kopplung zwischen eigentlicher Labelsoftware und dem Webauftritt des Recordlabels ergeben sich enorme Potenziale hinsichtlich Aktualität der Daten, Fehlerreduktion und Einsparungen von Arbeitszeit.*

Da die Recordlabel-Software und das Frontend auf die gleiche Datenbank zurückgreifen, entfällt eine doppelte Datenführung für die täglichen Arbeitsdaten und das Marketing. Die Daten im Frontend entsprechen so stets dem aktuellsten Stand.

Der Recordlabel-Mitarbeiter, der die Daten für einen neuen Künstler anlegt, soll sich nicht mehr darum kümmern müssen, dass die Daten auch auf dem Webauftritt korrekt erscheinen und stets aktuell sind. Er braucht nicht mehr die Diskografie eines Künstlers auf dem Webauftritt ändern, wenn dieser ein neues Release herausgebracht hat.

Fehler beim Transfer von einer Arbeitsdatenbank auf den Webauftritt können so gar nicht erst entstehen.

Bei der Kommunikation mit dem Kunden ergeben sich zudem wesentliche Arbeitserleichterungen und Zeiteinsparungen.

## 2.3 Sicheres Prelistening-Konzept

*Ein weiteres bedeutendes Ziel ist die Verwendung einer sicheren Prelistening-Funktion, die für möglichst viele Nutzer verwendbar ist.*

Wenn man Musik im Internet zugänglich machen will, stößt man schnell auf einige Probleme. Diese sind zum einen rechtlicher und zum anderen technischer Natur. Dabei spielt die Art der eingesetzten Technik auch eine Rolle für die rechtlichen Aspekte.

Wichtig bei der Entwicklung der Webanwendung war es, ein Prelistening-Verfahren zu entwickeln, das einfach und schnell zu bedienen ist, keine großen technischen Herausforderungen an den Nutzer und den Anbieter stellt und trotzdem verhindert, dass die Musikdateien nach dem Abspielvorgang physikalisch auf dem Nutzer-Rechner als Dateien vorliegen.<sup>3</sup>

Ein wesentlicher Aspekt ist hierbei, dass die Anforderungen an den Server, auf dem die Webanwendung gehostet werden kann, möglichst niedrig sein sollen. Somit fällt die Möglichkeit aus, einen Streaming-Server einzusetzen. Trotzdem soll sich die Webanwendung ähnlich wie eine Streaming-Umgebung verhalten.<sup>4</sup>

<sup>3</sup> Physikalisch vorliegen bedeutet in diesem Zusammenhang das Vorhandensein des abgespielten Musikstücks als Datei auf dem Nutzer-Rechner, zum Beispiel im Browser-Cache des Rechners. Verhindern lässt sich dies zum Beispiel durch echtes Web-Streaming.

<sup>4</sup> In Kapitel 3.3 gehe ich genauer auf die rechtlichen und technischen Gegebenheiten von Prelistening-Verfahren ein. In Kapitel 5.4 wird eine entsprechende Umsetzungsmöglichkeit genauer präsentiert.

# Kapitel 3

## **Bisheriger Stand der Technik**

---

Spezielle Recordlabel-Software

Individuelle Einzelanfertigungen

Prelistening-Verfahren

## 3.1 Spezielle Recordlabel-Software

*Bisher gibt es drei bedeutende Software-Produkte, die sich speziell an Recordlabels richten:*<sup>5</sup>

- *Officer von der Officer Business Software Solutions GmbH, Wien (vor allem für den europäischen Markt)*<sup>6</sup>
- *essCore von TrendSoft, Franklin (vor allem für den amerikanischen Markt)*<sup>7</sup>
- *go\_disko\* von Pixelkrieg Database Design, Berlin*<sup>8</sup>

### 3.1.1 Officer

Officer ist im europäischen Gebiet die bekannteste Software-Lösung für so genannte Independent Recordlabels. Die Software funktioniert als Einzelplatz-Lösung und basiert auf der Datenbank FileMaker, die zu jeder Platzversion zusätzlich installiert sein muss.

Eine direkte Frontend-Anbindung ist in der Software Officer nicht implementiert. Generell gestaltet es sich schwierig, in Officer Daten mit anderen Anwendungen auszutauschen. Selbst ein Newsletter-Versand ist bei Officer nicht direkt möglich. Zunächst wird von der Software eine CSV-Datei mit den Empfänger-Adressen erstellt, an die mit einem herkömmlichen E-Mail-Programm die E-Mail verschickt werden kann, deren Text jedoch aus Officer per Zwischenablage herauskopiert werden muss. In größerem Maße individualisierte Daten sind mit dieser Arbeitsweise nur unter größeren Umständen möglich.

Auf Wunsch kann man sich jedoch von der Officer Business Software Solutions GmbH Erweiterungen programmieren lassen, wie zum Beispiel eine Verbindung zu einem Webshop. Diese

<sup>5</sup> Vgl. hierzu auch den Vergleich in der Arbeit von Julian Servatius: „Entwicklung einer Networking- und Vertriebsplattform für Record Labels mit dem Schwerpunkt Office Funktionen“, Seiten 26 ff.

<sup>6</sup> Officer von der Firma Officer Business Software Solutions GmbH, Wien, Österreich. Weitere Infos: [www.officer.de](http://www.officer.de)

<sup>7</sup> essCore von der Firma TrennSoft, Franklin, USA. Weitere Infos: [www.virtitel.com/products/core/esscore.aspx](http://www.virtitel.com/products/core/esscore.aspx)

<sup>8</sup> go\_disko\* von der Firma Pixelkrieg Database Design, Berlin, Deutschland. Weitere Infos: [www.go-disko.de](http://www.go-disko.de)

individuellen Anpassungen sind jedoch sehr viel teurer als die bloße Software und auf Grund der Einzelplatz-Anwendung immer noch sehr unflexibel.

### 3.1.2 essCore

Die Software essCore von der US-amerikanischen Firma TrennSoft funktioniert recht ähnlich wie Officer. Eine lokale Mehrplatzlösung mit einer SQL-Datenbank im Hintergrund bietet auch hier nur reine Verwaltungs-Funktionalitäten und keine Anbindung an ein Frontend-System.

Doch bietet TrennSoft individuell auf spezielle Bedürfnisse hin entwickelte Lösungen an, die jedoch sehr viel teurer sind als die eigentliche Software essCore. Sie sind so gesehen vergleichbar mit individuellen Einzelanfertigungen.<sup>9</sup>

### 3.1.3 go\_disko\*

Die Anwendung go\_disko\* ist ebenfalls eine reine Office-Anwendung. go\_disko\* greift dabei genauso wie Officer auf eine FileMaker-Datenbank zurück. Die Anwendung besteht aus einem Basis-Modul, für die fünf Erweiterungsbausteine verfügbar sind. Eine Frontend-Lösung, beziehungsweise eine Frontend-Schnittstelle, gibt es bei dieser Anwendung ebenfalls nicht.

<sup>9</sup>Vgl. hierzu auch Kapitel 3.2

## 3.2 Individuelle Einzelanfertigungen für Recordlabels

*Nahezu alle Internet-Auftritte von kleineren und mittelgroßen Recordlabels sind Einzelanfertigungen, die zu einem gewissen Teil mit einem Backend verbunden sind, das allerdings lediglich zum Pflegen der Daten für den Webauftritt dient.*

Es ist verständlich, dass die Webauftritte in der Musikszene in individuellen Einzelanfertigungen programmiert werden, da in dieser Szene viel Wert auf Individualität gelegt wird, gerade auch hinsichtlich der Präsentation nach außen. Ziel der entwickelten Webanwendung ist nicht, Einzelanfertigungen durch einen standardisierten Webauftritt zu ersetzen. Vielmehr dient sie als Interface zur zentralen Recordlabel-Datenbank und soll die Website-Pflege entscheidend vereinfachen. In den Kapiteln 4.3 und 4.4 werde ich genauer auf die Interface-Funktionalitäten eingehen.

Vollkommen individuell erstellte Webanwendungen haben zwei gewichtige Nachteile:

- relativ hohe Entwicklungskosten
- hoher Pflegeaufwand

Der Pflegeaufwand bei individuellen Einzelanfertigungen lässt sich durch die Anbindung eines Backends vermindern. Für solch ein Backend ist dann jedoch ein größerer Entwicklungsaufwand nötig. Die Daten für das Frontend sind dabei jedoch immer noch getrennt von den eigentlichen Arbeitsdaten. Verschiebt sich also zum Beispiel das Veröffentlichungsdatum eines Albums, muss auf beiden Seiten diese Information geändert werden. Nichts ist für den Besucher des Webauftrittes ärgerlicher, als sich auf Informationen des Webauftrittes zu verlassen und anschließend zum Beispiel im Plattengeschäft erfahren zu müssen, dass diese Informationen nicht aktuell waren. Jacobsen (2005: Seite 297) unterstreicht die Notwendigkeit des Aktualhaltens von Daten im Internet:

*„Auch inhaltlich muss die Site aktuell bleiben. Auf besondere Ereignisse sollten Sie so schnell wie möglich reagieren. Nutzen Sie den Vorteil des Internets, dass Sie weder Druckerei noch Post brauchen, um aktuelle Nachrichten zu verbreiten.“*

Dies ist umso bedeutender, da das Internet in der heutigen Zeit immer häufiger herangezogen wird, um die wirklich aktuellsten Informationen zu erhalten.

Einzelanfertigungen sind teuer. Erst recht dann, wenn eine Schnittstelle zu der Datenbank einer bestehenden Recordlabel-Software (wie zum Beispiel Officer) entwickelt werden soll. Aber auch ohne eine solche Anbindung muss – wenn man die Inhalte des Webauftrittes dynamisch erstellen möchte – zunächst ein Datenbank-Modell und ein Backend für das Frontend entwickelt werden.

Verzichtet man vollständig auf dynamische Inhalte aus einer Datenbank, wird die Pflege der Daten sehr aufwändig und die laufenden Kosten für den Webauftritt steigen.

Die für diese Arbeit entwickelte Webanwendung ist eine Kombination aus klassischem Webauftritt und Schnittstelle zur zentralen Datenbank des Recordlabels. Das Frontend kann in seinem Aussehen in einem bestimmten Rahmen schnell und einfach durch Modifikation einer Layout-Klasse verändert werden.

Die Kernteile der entwickelten Anwendung dienen als Schnittstelle zur Datenbank und können dazu verwendet werden, ein individuell gestaltetes Frontend mit den nötigen Daten aus der Datenbank zu versorgen. Da die Ausgabe dieser Schnittstelle als XML erfolgt, sind dafür vielfältige Umsetzungsmöglichkeiten gegeben. In der vorliegenden Arbeit wurde für die Ausgabe eine Website in Macromedia Flash erstellt.

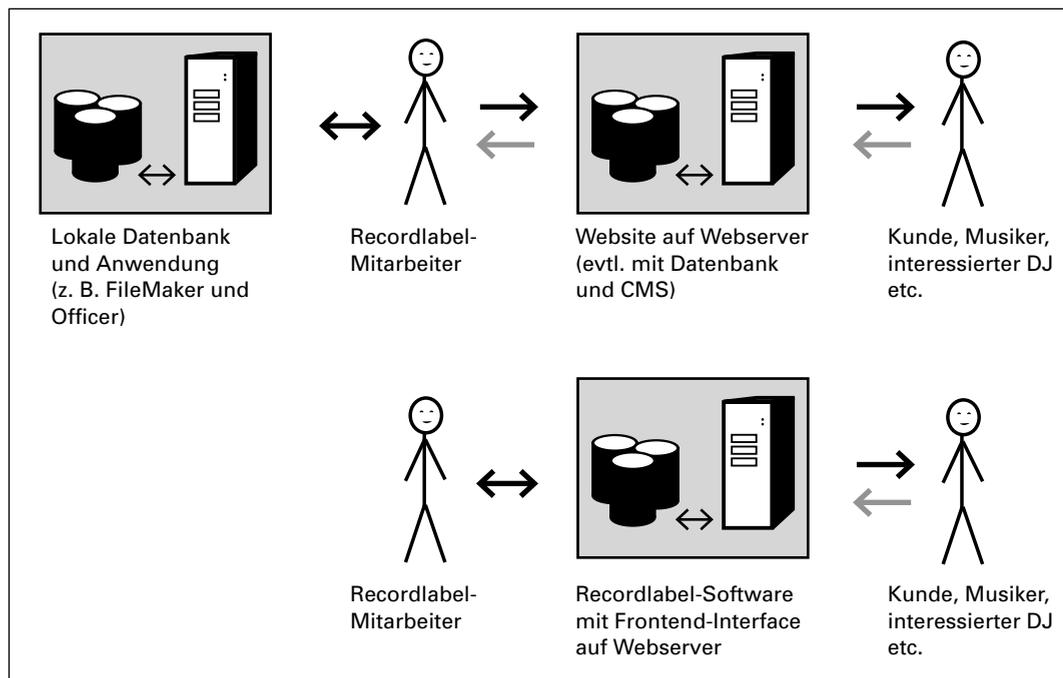


Abbildung 3.1:  
Schematische Darstellung der  
Online-Marketing-Abläufe  
in bisheriger Form (oben) und  
mit der entwickelten Record-  
label-Software (unten).

## 3.3 Prelistening-Verfahren

*Was liegt näher, als bei einem Webauftritt für Musik diese auch direkt zu präsentieren und zum Probehören anzubieten? Doch wie vermeidet man, dass diese Musikproben als Datei auf dem Rechner des Webauftrittbesuchers gespeichert werden können?*

Für jedes Plattenlabel stellt sich bei der Planung eines Webauftrittes die Frage nach der Handhabung des Prelistening, also einer „Probeanhör-Funktion“.

Zu beachten sind nicht nur technische, sondern auch rechtliche Belange.

### 3.3.1 Rechtliche Gegebenheiten

Eine einfache Zurverfügungstellung eines Musikstückes, das GEMA<sup>10</sup>-Rechten unterliegt, ist ohne weiteres nicht möglich.

Für Webauftritte von Recordlabels und Künstler-Websites gibt es eine spezielle Vereinbarung zwischen dem VUT<sup>11</sup> und der GEMA. Diese Vereinbarung gilt momentan nur interimistisch, da seit Anfang 2005 eine neue Regelung zwischen den beiden Institutionen ausgehandelt wird.

Diese Vereinbarung verpflichtet die Recordlabels zur Zahlung von 150,- Euro. Im Gegenzug kann das Recordlabel Prelistening und Streamings von seinen Tracks anbieten. Wobei zu beachten ist, dass sich diese Hörproben an die Endverbraucher richten und unentgeltlich sind. Pro Monat und Track sind zirka 10.000 Abrufe mit dieser Gebühr abgegolten.

Wichtig ist jedoch zusätzlich, dass, wenn es sich um vollständige Musikstücke handelt, die Tracks nicht als direkter Download angeboten werden dürfen. Das heißt, es ist sicherzustellen, dass die Musik-Tracks nach dem Anhör-Vorgang nicht mehr als Datei auf dem Benutzerrechner vorhanden sein dürfen. Dies gilt nicht für Werkteile, also kurze Ausschnitte aus den Musikstücken.

**Kein vollständiger  
Download**

<sup>10</sup> Gesellschaft für musikalische Aufführungs- und mechanische Vervielfältigungsrechte Stuttgart. Link: [www.gema.de](http://www.gema.de)

<sup>11</sup> Verband unabhängiger Tonträgerhersteller, Berlin. Link: [www.vut-online.de](http://www.vut-online.de)

### 3.3.2 Technische Gegebenheiten

Losgelöst von den rechtlichen Gegebenheiten gibt es viele technische Möglichkeiten und Wege, um Tondateien über das Internet zugänglich zu machen. Folgende Beispiele beschränken sich auf den Bereich der Verbreitung über HTTP und der Verbreitung durch Streamingverfahren.

#### 3.3.2.1 Einfache Verlinkung und Forced-Download

Man kann die entsprechenden Tondateien direkt zum Download anbieten, entweder als gewöhnliche Verlinkung (Anchor-Tag `<a href="audio/mysong.mp3">My Song</a>`) oder in Form eines Forced-Download-Links (erzwungenes Herunterladen, realisiert zum Beispiel durch Veränderung des Datei-Headers). Bei einem einfachen Klick auf den gewöhnlichen Link startet in der Regel ein Plugin zum Abspielen der Datei im Browser<sup>12</sup>. Bei Verwendung einiger Plugins wird dabei das Bedienfeld des Plugins in das Browserfenster geladen. Je nachdem, welche weiteren Attribute angegeben werden, verschwindet so der eigentliche Inhalt der Internetseite und ist nur über die Zurück-Schaltfläche des Browsers wieder erreichbar. Kann weder der Browser die Tondatei wiedergeben, noch ist ein entsprechendes Plugin vorhanden, öffnet der Browser eine externe Anwendung, die die Tondatei abspielen kann.

Beim Forced-Download-Link öffnet der Browser auch bei einem einfachen Klick (mit der linken Maustaste) ein Dialog-Fenster, in dem vom Benutzer der gewünschte Speicherort der Tondatei angegeben werden muss. Die danach heruntergeladene Datei kann/muss der Benutzer dann extra öffnen. Der Nutzer kann in diesem Fall die Datei erst dann anhören, wenn sie vollständig heruntergeladen ist. Ermöglicht wird ein solcher erzwungener Download durch die Veränderung des Datei-Headers. Die Angabe `Content-Disposition: attachment;` im Header der vom Browser angeforderten Datei veranlasst den Browser den Datei-speichern-Dialog zu öffnen.

In allen diesen Fällen besteht die Datei als direkte Kopie auf dem Benutzer-Rechner. Im Falle des direkten Anklickens eines einfachen Links im Browser-Cache; im Falle des gezielten Downloads (rechter Mausklick auf gewöhnlichen Link und der Angabe „Speichern unter...“) und im Falle des Forced-Downloads an einem vom Benutzer angegebenen Speicherort.

<sup>12</sup> Der Microsoft Internet Explorer ab Version 3 kann die gängigsten Audioformate eigenständig abspielen.

### 3.3.2.2 Direkte Einbettung in HTML

Tondateien lassen sich auch als so genannter Hintergrund-Sound in HTML-Code einbinden. Die dazu nötigen Tags sind jedoch teilweise browserspezifisch.<sup>13</sup>

```
<embed src="audio/mysong.mp3" autostart="true" />
<noembed><bgsound="audio/mysong.mp3" /></noembed>

<meta http-equiv="refresh" content="1;url=audio/mysong.mp3">
```

Generell ist jedoch von dieser Vorgehensweise abzuraten, da der Benutzer keine Steuerungsmöglichkeiten besitzt, außerdem startet die Wiedergabe erst nach vollständigem Download. Die Tondatei liegt auch in diesem Fall im Browser-Cache des Benutzers.

*Listing 3.1:  
Möglichkeiten, Sound direkt  
in HTML einzubetten.*

### 3.3.2.3 Progressive Download

Bestimmte Anwendungen und Plugins bieten die Möglichkeit des so genannten Progressive-Downloads. Jennifer Niederst bezeichnet diese Vorgehensweise auch als „Pseudo-Streaming“<sup>14</sup> (Niederst 2002: Seite 427). Vorreiter in dieser Technik ist die Firma Apple mit ihrem Quicktime-Player. Der Benutzer kann dabei die Wiedergabe der Tondatei schon während des Downloads starten, bevor die vollständige Datei auf den Benutzerrechner heruntergeladen ist. Dabei liegt auch hier nach dem Download eine identische Kopie des Originals auf dem Benutzerrechner.

### 3.3.2.4 Streaming

Bei den so genannten echten Streaming-Verfahren wird im Unterschied zu den bisher genannten Verfahren ein anderes Übertragungs-Protokoll verwendet. Dies sind laut Niederst (2002: Seite 426)

*„entweder UDP (User Datagram Protocol), RTSP (RealTime Streaming Protocol) oder RTP (RealTime Transfer Protocol) [...] Was diese Protokolle effizient macht, ist, daß die Datenübertragung weiterläuft, wenn ein Paket verlorenght.“*

<sup>13</sup>Vgl. hierzu auch Niederst 2002: Seite 436 f.

<sup>14</sup> Der Begriff Progressive-Download erscheint mir für diese Technik jedoch angemessener, da auch bei dieser Technik die Datei vollständig heruntergeladen wird, nach dem Download als identische Kopie auf dem Benutzerrechner vorliegt und keine echte Streaming-Technologie verwendet wird.

Ein großer Vorteil von Streaming ist also die Effizienz. Streams bieten die schnellste Möglichkeit, Audio- und Videodaten zu übermitteln, die sofort angehört beziehungsweise angesehen werden sollen. Die Schnelligkeit geht sogar so weit, dass die Streams nahezu in Echtzeit übermittelt werden können, also auch Live-Übertragungen möglich sind.

Die Serverlast ist für Streams relativ niedrig, so ist zum Beispiel der Zugriff auf ein und dasselbe Musikstück von vielen Rechnern aus möglich: „Kommerzielle Streaming-Software wie zum Beispiel RealServer kann Tausende von Daten-Streams gleichzeitig verarbeiten.“ (Niederst 2002: Seite 427)

Ein weiterer Vorteil bei der Streaming-Technologie ist die Möglichkeit, sofern es sich nicht um einen Live-Stream handelt, direkt an eine bestimmte Stelle zu springen. Also zum Beispiel mitten in einem Musikstück mit der Wiedergabe zu beginnen. Dabei muss nicht erst der gesamte vorherige Teil übertragen werden, wie es bei den bisher vorgestellten Techniken der Fall ist.

Heutige Streaming-Technologien ermöglichen es zudem, dass der Stream der zur Verfügung stehenden Bandbreite angepasst wird. Dabei wird die Kompressionsrate (und somit die Qualität) entsprechend angepasst.

Hinzu kommt, dass die übertragenen Pakete nicht auf dem Benutzerrechner gespeichert und auch gar nicht erst zu einer vollständigen Kopie zusammengefügt werden. Es besteht bei diesem Verfahren für den Benutzer also keinerlei Möglichkeit, eine vollwertige Kopie von dem übertragenen Audio- oder Videostream zu besitzen. Natürlich ist es weiterhin für den Benutzer möglich, durch spezielle Software (oder zum Beispiel durch Anschluss einer Musikanlage) den Stream aufzuzeichnen, doch die Vollständigkeit und Qualität des Originals können dabei nicht erreicht werden.

Es gibt bei Streaming jedoch auch gewaltige Nachteile: Benutzer mit langsamen Internetverbindungen haben keine Möglichkeit, ein Musikstück oder ein Video in sehr guter Qualität abzuspielen. Entweder wird vom Streaming-Server automatisch eine niedrigere Qualitätsstufe geliefert oder aber der Benutzer bekommt nur Bruchstücke zu hören beziehungsweise zu sehen.

Ein gewichtiger Nachteil für den Anbieter hingegen ist die Notwendigkeit eines speziellen Streaming-Servers beziehungsweise einer Streaming-Serversoftware. Dies kann schnell zu einem bedeutenden Kostenfaktor werden (Niederst 2002: Seite 427):

*„Software und Hardware für einen dedizierten Streaming-Server können sehr kostspielig werden. Darüber hinaus fallen bei RealNetworks Lizenzgebühren auf Basis der Stream-Anzahl an.“*

**Keine vollwertige Kopie möglich**

**Nachteile**

Zudem gibt es bei vielen Paket-Produkten von Web-Hosting-Anbietern keine Möglichkeit, Streaming-Inhalte anzubieten.

#### **3.3.2.5 Der Mittelweg: Progressive Download mit Datei-Header-Modifikation**

Genauso wie beim Forced Download der Browser dazu veranlasst wird, den Speichern-unter-Dialog anzuzeigen, kann durch Modifikation des Datei-Headers der Browser und die Wiedergabe-Software dazu aufgefordert werden, die übermittelte Datei sofort nach der Benutzung aus dem Cache zu löschen. Die vollständige Kopie des Originals ist dabei in der Regel also nur so lange im Browser-Cache oder im Cache der Abspiel-Software vorhanden, so lange das Abspiel-Fenster geöffnet ist.

Dieses Verfahren wird in Kapitel 5.4 „Sicherheitsaspekte hinsichtlich Urheberrechte und Download-schutz“ genauer erörtert.

# Kapitel 4

## Eingeschlagener Realisierungsweg

---

Die magischen Vier

Clientseitig: Macromedia Flash

Serverseitig: PHP/MySQL

Die Brücke: XML

## 4.1 Die magischen Vier

Der praktische Teil der vorliegenden Diplomarbeit wurde in einer Kombination von vier Techniken umgesetzt:

1. MySQL als Datenbank
2. PHP als Middleware
3. XML als Marker-Sprache
4. Macromedia Flash als Ausgabe-Anwendung

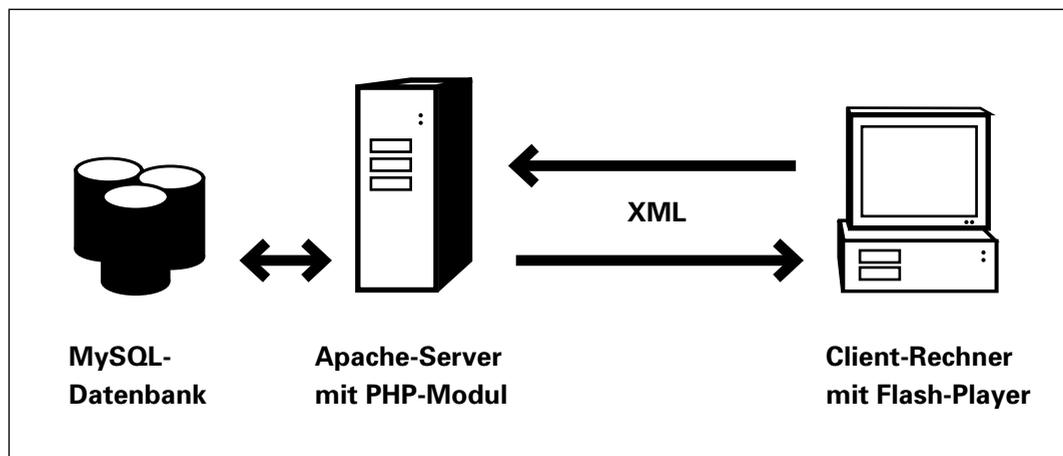


Abbildung 4.1:  
Schematische Darstellung  
des Zusammenspiels der vier  
eingesetzten Techniken

Abbildung 4.1 verdeutlicht im Groben den Zusammenhang der vier Bauteile. Die Flash-Anwendung auf dem Benutzerrechner erstellt eine Anfrage in Form einer XML-Datei, die an ein PHP-Skript auf dem Server geschickt wird. Das PHP-Skript wertet das XML aus und erstellt eine Datenbankanfrage für die MySQL-Datenbank. Diese liefert die gewünschten Ergebnisse, soweit vorhanden, an das PHP-Skript zurück. Aus den Ergebnissen erstellt das PHP-Skript wiederum eine XML-Datei, die der Server zurück zum Benutzerrechner schickt. Auf dem Benutzerrechner nimmt der Flash-Player die übermittelte XML-Datei entgegen, wertet es seinerseits aus und zeigt den Inhalt entsprechend der Layout-Vorgaben an.

## 4.2 Clientseitig: Macromedia Flash

*„We are seeing a new generation of Web applications being built with Flash as the front end.“  
(Swann/Caines 2002: Seite 149)*

Die von Swann und Caines angesprochene neue Generation der Webapplikationen, die in Flash erstellt sind, hat sich seither (der erste Druck des Buches stammt aus dem Jahr 2001) noch enorm weiterentwickelt. Mittlerweile sind wir bei der Version 8 des Flash-Players angelangt, die Programmiersprache ActionScript liegt in der Version 2 vor und ist damit objektorientiert geworden, und auch im Bereich Video-Einbettung, Bitmap- und Font-Rendering hat sich viel getan. Mit ActionScript 3 und dem nächsten Build von Flash wird sich weiterhin viel tun. Trotz der vielen Möglichkeiten mit Macromedia Flash sollte man jedoch gewisse Hürden und Problematiken nicht aus dem Auge verlieren, die sich mit der Verwendung dieses Formates ergeben.

Im folgenden möchte ich kurz auf diese Problematiken und die kreativen und technischen Möglichkeiten von Macromedia Flash eingehen. Da der praktische Teil der vorliegenden Diplomarbeit hauptsächlich in Macromedia Flash realisiert wurde, möchte ich danach die Gründe für den Einsatz im Speziellen erläutern.

### 4.2.1 Vor- und Nachteile

Schaut man sich in verschiedenen Internet-Foren über Webentwicklung um, so stößt man unweigerlich irgendwann auf Threads, in denen zwei Parteien unerbittlich über ein Thema diskutieren: Ist der Einsatz von Macromedia Flash auf Internet-Seiten wirklich sinnvoll oder nicht?

Um das Ergebnis einer sachlichen Auseinandersetzung mit dem Thema vorwegzunehmen: Geht es nicht primär um reine Information und soll der Benutzer emotional angesprochen werden, ist der Einsatz von Flash sinnvoller als ein reiner (D)HTML-Web-Auftritt. Zur schnellen und komfortablen Bereitstellung von Information ist eine (D)HTML-Lösung jedoch wesentlich besser geeignet als eine Flash-Lösung; es sei denn, es handelt sich um einen komplizierten Sachverhalt, der durch eine gelungene Animation besser veranschaulicht werden kann.

Wobei angemerkt sei, dass in den letzten Jahren viele Einschränkungen für den Einsatz von Macromedia Flash weggefallen sind. Das liegt zum einen darin begründet, dass die durchschnittliche Bandbreite der Internetverbindungen enorm zugenommen hat, als auch darin, dass Macromedia Flash neue Möglichkeiten bietet. Als Beispiel sei hier die `XML.sendAndLoad()`-Methode genannt, mit der es mittlerweile auch in Flash-Anwendungen möglich ist, mittels eines Serverskriptes dynamische Inhalte aus einer Datenbank anzuzeigen. Ein weiteres Beispiel ist die zunehmende Suchmaschinen-Freundlichkeit von Flash-Inhalten. Sascha Wolter geht auf einige dieser veralteten Problemfälle und den damit verbundenen Vorurteilen zum Einsatz von Flash genauer ein (Wolter 2005: Seite 24 ff.).

#### 4.2.1.1 Vorteile

Der größte Vorteil, der mittlerweile für die Verwendung von Macromedia Flash auf Webseiten spricht, ist der emotionale Faktor. Keine andere Technik vermag es momentan so leicht, Text, Ton, Bild, Video, Bewegung und Interaktion miteinander zu kombinieren. Dabei besteht die Möglichkeit, auch bei größeren Datenmengen durch intelligente Preloading-Verfahren die Wartezeiten für den Benutzer so kurz wie möglich oder zumindest so angenehm wie möglich zu machen.

In Macromedia Flash programmierte Webauftritte können dem Benutzer eine ganz andere Art von Interaktion bieten, die zum Beispiel spielerische Elemente aufweist.

Pascal Leroy, der Gründer der Agentur group94, vergleicht das Surfen auf einer gut gemachten Flash-Website mit einer Reise (Leroy: 2006):

*„Aufgrund technischer Einschränkungen ist das Surfen auf einer HTML-Website uni-dimensional. Im Gegensatz dazu kann das Surfen auf einer Flash-Website, wenn sie gut programmiert ist, mit einer Reise verglichen werden, die letztendlich zu einem bildhaften und ideenreichen Erlebnis wird.“*

Dabei spricht Leroy die Art von Webseiten an, die vor allem auf Imagebildung und Unterhaltung abzielen. Leroy schränkt jedoch auch ein, dass Webauftritte für große internationale Unternehmen nicht vollständig mittels Macromedia Flash realisiert werden sollten, der Einsatz von Flash-Elementen für Multimedia und Navigations-Elementen jedoch durchaus angebracht sei.

Ein verspieltes Navigationskonzept ist für eine reine Informationsseite sicherlich nicht angebracht. Geht es jedoch gar nicht darum, in erster Linie Information zu bieten, sondern den Benutzer emotional herauszufordern und zu fesseln, ist Macromedia Flash also die bessere Wahl.

#### Emotionen wecken

Ein weiterer entscheidender Vorteil für den Einsatz von Macromedia Flash ist die Tatsache, dass die Inhalte auf jeder Plattform identisch aussehen, für die ein Flash-Player verfügbar ist.

**Echte Crossplattform-Fähigkeit**

Der Einsatz von Macromedia Flash ermöglicht einen sehr großen typografischen Spielraum. Seit der Version 8 sorgt eine neue Render-Engine der Software zudem für eine optimierte Bildschirmdarstellung von Schriften, sodass Texte außerordentlich gut lesbar sind.

Macromedia Flash ist vektorbasiert. Dies ermöglicht wesentlich kleinere Dateigrößen für Illustrationen und einfache Animationen.

**Vektorbasiert**

Der Flash-Player selbst ist sehr schlank. Der Benutzer kann ihn somit schnell aus dem Internet laden und installieren. Die Installationsdatei der aktuellen Version des Flash-Players (Version 8) für Windows-PCs ist nicht einmal ein Megabyte groß und kann somit selbst mit einem 56K-Modem in etwa zwei Minuten heruntergeladen werden.

**Schlanker Player**

Bis zur Version 7 ist der Flash Player das am weitesten verbreitete Plugin: Im Dezember war der Flash-Player in der Version 7 laut Macromedia auf knapp 95 Prozent der Rechner aller europäischen Internetnutzer vorhanden (vgl. Adobe/Macromedia 2006, „Macromedia Flash Player Penetration“). Der Prozentsatz der Rechner mit einem Flash-Player ab der Version 2 von allen Rechnern im Internet wird von Adobe/Macromedia für Dezember 2005 mit 97,7 Prozent angegeben (vgl. Adobe/Macromedia 2006, „Macromedia Flash Player Penetration“).

**Weite Verbreitung**

Laut der selben Quelle ist der Microsoft Windows Media Player (ab der Version 7) nur auf 85,6 Prozent aller internetfähigen Rechner vorhanden beziehungsweise voll funktionsfähig. Bei anderen Media-Playern sieht es noch schlechter aus: Der Apple Quicktime Player kommt auf 65,9 Prozent, der RealOne Player auf 59,9 Prozent.

Zu beachten ist jedoch, dass der aktuellste Flash-Player (Version 8) längst nicht mehr so gut dasteht: Nur noch 55,1 Prozent aller Internet-Rechner in Europa besaßen im Dezember 2005 den neuesten Flash-Player. Seit der Version 8 gibt es zudem keine Linux-, Solaris- und MacOS-9-kompatible Version.

#### 4.2.1.2 Nachteile

Bei der Verwendung von Macromedia Flash gibt es auch Nachteile. Einige Nachteile, die noch vor ein paar Jahren sehr bedeutend waren, haben sich mittlerweile abgeschwächt oder sind gar nicht mehr vorhanden. So sind Macromedia-Flash-Anwendungen schon länger aus inhaltlicher Hinsicht keine

starrten Anwendungen mehr, da durch das XML-Objekt und der schon kurz angesprochenen `XML.sendAndLoad()`-Methode mittlerweile ein guter Datenaustausch stattfinden kann. Gegenüber (D)HTML-Lösungen mit dynamisch erstellten Inhalten bleibt jedoch das Problem des höheren Entwicklungsaufwandes.

Für die Erstellung von komplexen Flash-Anwendungen ist zudem die Macromedia-Flash-Software unentbehrlich, die im Gegensatz zu vielen Entwicklungs-Tools für den (D)HTML-Bereich aus der openSource-Community nicht frei zur Verfügung steht, sondern für einen recht hohen Preis erworben werden muss.

Macromedia-Flash-Webanwendungen sind nur bedingt für Suchmaschinen indexierbar. Zwar bietet Adobe/Macromedia ein Software-Development-Kit<sup>15</sup> für Suchmaschinen-Programmierer an, das ermöglicht, aus SWF-Dateien HTML-Dateien zu erstellen, die dann indexiert werden können, doch nutzen bisher nur wenige Suchmaschinen die Möglichkeit Flash-Anwendungen zu durchsuchen. Zudem ist die Erfolgsquote der Indexierung stark davon abhängig, wie der Entwickler die Flash-Anwendung aufbaut und programmiert.

Soll der Inhalt des Webauftrittes also sicher und ordentlich in Suchmaschinen indexiert, aber nicht auf Flash verzichtet werden, empfiehlt sich nach wie vor die Erstellung einer parallel vorhandenen XHTML-Seite.

Die Textanzeige einer Flash-Anwendung selbst lässt sich nicht durch die Suchfunktion des Browsers durchsuchen.

Flash-Anwendungen sind nicht immer für alle Benutzergruppen gut zugänglich; das Stichwort Accessibility sei hier genannt. Zwar bietet Macromedia Flash mittlerweile eine eigene Accessibility-Klasse und selbst die gängigsten Vorlese-Programme wie zum Beispiel Jaws von der Firma Freedom Scientific<sup>16</sup> können Inhalte aus Flash-Anwendungen vorlesen, doch gestaltet sich die Umsetzung einer wirklich gut zugänglichen Anwendung mitunter recht schwierig. Dies hat zur Folge, dass die Entwicklungskosten noch weiter steigen.

Flash-Anwendungen verlangen von den Entwicklern und Designern ein großes Maß an Verantwortung hinsichtlich der Usability. Nur zu häufig ist die Verlockung zu groß, alles auszureizen, was die Software Flash möglich macht. Die Benutzbarkeit des Webauftrittes kann dabei schnell nachlassen.

**Höherer Entwicklungsaufwand**

**Keine gute Indexierung bei Suchmaschinen**

**Accessibility**

<sup>15</sup> Mehr Informationen bietet Adobe/Macromedia unter <http://www.macromedia.com/licensing/developer/>

<sup>16</sup> Jaws von der Firma Freedom Scientific BLV Group, St. Petersburg. Link: [www.freedomscientific.com](http://www.freedomscientific.com)

Aber es ist auch das Gegenteil möglich: Flash-Anwendungen können mitunter sehr viel intuitiver Inhalte voneinander trennen oder miteinander in Verbindung setzen. Im Kapitel 5.6 werde ich noch genauer auf das Thema Usability eingehen.

Zum Usability-Aspekt gehört auch das Problem, dass eine Flash-Anwendung nicht auf bestimmte Browser-Bedienelemente reagiert, wie zum Beispiel die Zurück-Schaltfläche. Dies ist bedingt dadurch behoben, dass sich in der Zeitleiste von Flash-Filmen Anker setzen lassen, die dann direkt durch den Browser angesteuert werden können (erkennbar an der Raute mit dem darauffolgenden Anker-Namen am Ende der URL der Seite). Mit dieser Lösung lässt sich dann auch ein exakteres Lesezeichen setzen, doch erfordert sie einen häufig sehr viel komplizierteren Aufbau der Flash-Datei. Zudem führt diese Vorgehensweise vor allem bei dynamischen Inhaltsdaten nicht immer zum gewünschten Ergebnis oder ist schlicht nicht umsetzbar. Auf Macintosh-Betriebssystemen ist die Anker-Funktionalität nur eingeschränkt möglich.

**Back-Button und Bookmarks**

## 4.2.2 Gründe für die Verwendung von Macromedia Flash

Abgesehen davon, dass vom Auftraggeber der Gesamtsoftware ein Frontend gewünscht wurde, das in der aktuellsten Flash-Version erstellt ist (also erst ab dem Flash-Player 8 anzeigbar), gibt es verschiedene technische und das Marketing betreffende Gründe für die Verwendung von Macromedia Flash.

### 4.2.2.1 Technische Gründe

Aus technischer Sicht ist der Einsatz von Flash empfehlenswert, da der Flash-Player das am weitesten verbreitete und nebenbei im Vergleich auch das schlankste Plugin ist, das Audiodaten komfortabel wiedergeben kann. Da ein bedeutender Teil der Anwendung die Prelistening-Funktion ist, die nahtlos in den Gesamtauftritt miteinbezogen sein sollte, bot sich zudem die gesamte Umsetzung in Macromedia Flash an. So konnte zusätzlich sichergestellt werden, dass die Manipulation der Prelistening-Funktion deutlich erschwert ist, da Übergabe-Variablen an den Player nur mit größerem Aufwand herausgefunden werden können.

**Erschwertes „Hacken“**

Generell sind Hacks auf die Anwendungs-Schnittstelle zur Datenbank erschwert, da die Kommunikation über XML-Strings erfolgt, die aus einer geschlossenen Anwendung stammen. Was freilich nicht heißt, dass keine Hacks möglich wären oder gar Flash die sicherste Möglichkeit darstellte. Der Aufwand und somit die erste Hürde sind jedoch höher, als bei einer einfachen XHTML-Lösung, in der Variablen und Werte einfacher einsehbar sind.

Die Entscheidung, tatsächlich auch die Version 8 des Flash-Players vorauszusetzen, lag darin begründet, dass erst ab dieser Version die `FileReference`-Klasse verfügbar ist, die es erlaubt, direkt aus der Flash-Anwendung heraus Up- und Downloads auszuführen. Gerade für die Uploadmöglichkeit für Demo-MP3-Tracks ist dies aus Usability-Sicht zum Beispiel ein enormer Vorteil, da die `FileReference`-Klasse eine `onProgress()`-Methode bietet, mittels derer ohne großen Zusatzaufwand ein echter Status-Balken programmiert werden kann, der dem Benutzer mitteilt, wie weit der Upload schon fortgeschritten ist.

Für potentielle Hacker ist nicht ohne größeren Aufwand ersichtlich, wohin oder woher die Datei transferiert wird. Der Up- und Download ist nur mittels des Browser-Plugins von Flash möglich, Zugriffe aus dem Standalone-Player heraus sind nicht möglich.

Beim Upload werden vom Flash-Plugin nur die Dateien zugelassen, die dem gewünschten Typ entsprechen<sup>17</sup>.

#### 4.2.2.2 Werbliche Gründe

Die entscheidenderen Gründe für die Verwendung von Macromedia Flash sind jedoch marketing-technischer Natur. Die entwickelte Webanwendung richtet sich vor allem an so genannte Independent Recordlabels. Deren Kundenkreis besteht zumeist aus einem jugendlichen und trend-orientierten Publikum, das großen Wert auf ansprechendes Design und auf Unterhaltung legt.

Es gilt also, dieses Publikum emotional anzusprechen. Dazu ist bei Webauftritten Flash um einiges besser geeignet als zum Beispiel eine DHTML-Lösung (Swann/Caines 2002: Seite 149):

*„Flash allows you to visually and interactively display your content in ways that are not possible with the standard HTML interface and, on top of that, be totally cross-platform, cross-browser compatible.“*

Neben einer ansprechenderen Typografie sind mit Flash verspielte Elemente wie zum Beispiel die Animation eines Plattenspielers als Musikspieler möglich, die den Spaßfaktor des Webauftrittes erhöhen.

<sup>17</sup> Wobei angemerkt sei, dass dies allein nicht wirklich sicher ist! Es stellt lediglich ebenfalls eine Erschwerung für Angriffe dar. Ein potentieller Angreifer kann Dateien im Upload-Dialog nachträglich wieder umbenennen. Es ist somit serverseitig unbedingt noch einmal zu überprüfen, ob die übermittelte Datei tatsächlich dem zugelassenen Typ entspricht!

Durch spielerische Elemente wird der Entdeckergeist des Benutzers herausgefordert. Dies führt unter Umständen sogar dazu, dass der Benutzer zu einer Information gelangt, die für ihn interessant ist, die er jedoch vorher gar nicht explizit gesucht hat und auf die er somit auf einer rein funktional ausgelegten Informationsseite gar nicht gestoßen wäre.

Eine Flash-Seite bietet für ein Recordlabel einen sehr viel größeren Spielraum zur individuellen Abgrenzung gegenüber anderen Recordlabels. Ein abgrenzender, individueller Webauftritt ist in der Musikszene ein nicht zu vernachlässigender Positionierungs-Faktor auf dem Musikmarkt. Mit dem Webauftritt eines Recordlabels soll auch ein bestimmtes Image transportiert werden.

Musik bestimmt den Coolness-Faktor einer ganzen Generation. Das Auftreten der Musiker, aber auch des Recordlabels hat daran einen entscheidenden Anteil. Dabei hat es ein Recordlabel quasi gleich mit zwei Arten von potentiellen Kunden zu tun: Zum einen mit den Musik-Konsumenten, aber zum anderen auch mit guten Künstlern, die eventuell unter Vertrag genommen werden könnten. Macht das Label jedoch nach außen einen faden und uninteressanten Eindruck, wird sich der Künstler eher für ein anderes Label entscheiden.

**Entdeckergeist herausfordern**

## 4.3 Serverseitig: PHP und MySQL

*Flash allein kann nicht direkt auf Datenbanken zugreifen. Um in Flash-Anwendungen dynamisch erzeugte Inhalte aus einer Datenbank anzuzeigen, wird also eine so genannte Middleware auf dem Server benötigt.*

Im praktischen Teil der vorliegenden Diplomarbeit wurde dazu PHP eingesetzt. PHP (PHP Hypertext Preprocessor) ist eine Skriptsprache, die frei verfügbar und speziell für die Entwicklung von Webseiten entwickelt worden ist. PHP bietet umfangreiche Funktionalitäten und ist besonders für den Einsatz mit Datenbanken empfehlenswert (Krause 2003: Seite 25):

*„PHP besticht außerdem durch einen fast schon grandiosen Funktionsumfang. Hier kann keine andere Sprache mithalten. Wo immer ein Programmierer ein Problem zu lösen hatte, wurde der Funktionsumfang erweitert. Besonders deutlich wird dies bei den Datenbankfunktionen. [...] Den Erfolg macht nicht zuletzt auch die gute Unterstützung für die Datenbank MySQL aus.“*

Der Hauptgrund, warum PHP als Middleware für das Frontend gewählt wurde, liegt natürlich darin, dass auch die Office-Funktionalitäten der Gesamt-Anwendung in PHP programmiert wurden.<sup>18</sup> Hinzu kommt jedoch der Grund, dass die meisten Web-Hosting-Anbieter PHP und eine MySQL-Datenbank auf ihren Servern installiert haben. PHP bietet eine sehr gute Unterstützung von XML, das sozusagen als Kommunikationsbrücke zwischen Flash und PHP dient.

Der Einsatz von PHP bietet eine größere Flexibilität und Ausbaufähigkeit: So lässt sich zum Beispiel ohne großen Aufwand in Zukunft eine XHTML-Ausgabe hinzufügen.

<sup>18</sup> Für eine ausführlichere Darlegung der Gründe für den Einsatz von PHP und MySQL siehe Julian Servatius: „Entwicklung einer Networking- und Vertriebsplattform für Recordlabels mit dem Schwerpunkt Office-Funktionen“, Seiten 21 f. sowie Seiten 39 ff.

## 4.4 Die Brücke: XML

*„XML has started a new revolution in content structure and data exchange for the Web. Linked with the Web's most revolutionary interactive application – Flash – you have a wonderful world of new interactive opportunities. In its own right, XML is one of the hottest and fastest growing technologies on the Net today. It is shaping the next generation of Internet applications.“*

*(Swann/Caines 2002: Seite 9)*

XML (Extensible Markup Language) ist eine Metasprache, mit der sich beliebige Baumstrukturen darstellen lassen. Im Gegensatz zu HTML ist so der Inhalt völlig losgelöst von der Form (Gumm/Sommer 2002: Seite 586):

*„Die Bedeutung der Markierungen ist in HTML vordefiniert. In XML kann man dagegen eigene, problem-spezifische Markierungen benutzen, ohne ihre Bedeutung vorher definiert zu haben.“*

Durch die Trennung von Inhalt und Form lässt sich die Pflege der Inhalte schneller und effizienter durchführen, da keine Gestaltungsangaben als Ballast mitgeschickt werden müssen.

Genau genommen ist laut Niederst (2002: Seite 515) XML vielmehr „ein Regelwerk zur Erstellung anderer Markup-Sprachen. [...] Es ist ein mächtiges neues Tool für den Austausch bedeutungsvoller Informationen.“

XML lässt sich zum einen dazu nutzen, den gleichen Inhalt in verschiedenen Ausgabeanwendungen auszugeben, aber umgekehrt auch, durch Verwendung einer immer konstanten Dokumentstruktur unterschiedliche Inhalte in einer bestimmten Ausgabeanwendung immer auf die gleiche Art darzustellen.

XML eignet sich hervorragend dazu, Informationen zwischen verschiedenen Anwendungen auszutauschen. Und seit der Version 5 von Macromedia Flash ist XML auch für Flash-Anwendungen die bevorzugte *Verpackung* für dynamische Inhalte, die zwischen Server und Anwendung ausgetauscht werden sollen. Wie dies geschehen kann, zeige ich genauer im Kapitel 5.7.

**Ideal zum Informations-  
austausch**

# Kapitel 5

## Die Umsetzung

---

Grundgedanken und Layout

Dynamisches Template in Flash (Layout-Klasse)

Sicherheitsaspekte in Flash-Anwendungen

Sicherheitsaspekte hinsichtlich Downloadschutz

Miteinbezug der Site-Nutzer

Usability-Aspekte

XML-Brücke

## 5.1 Grundgedanken und Layout

*Ziel war es, eine Frontend-Anwendung zu erstellen, die möglichst flexibel, also einfach zu modifizieren ist. Sie sollte so modular wie möglich aufgebaut sein, damit der Entwicklungsaufwand für die Umsetzung eines neuen Layouts möglichst niedrig ausfällt.*

Auf der Schnittstellen-Ebene war dies recht einfach umsetzbar. Dadurch, dass die Schnittstelle in PHP programmiert wurde, ist auf Wunsch ohne großen Zusatzaufwand eine Ausgabemöglichkeit in HTML möglich. Zudem bietet die Ausgabe in XML die Möglichkeit, die Inhalte auf einer Vielzahl von Ausgabegeräten anzeigen zu können, also auch mit Anwendungen, die in der jetzigen Planung noch gar nicht vorgesehen sind.

Problematischer beziehungsweise komplexer gestaltete sich die Umsetzung der Flexibilität bei der Programmierung der Flash-Anwendung. Bei Flash-Anwendungen ist die Trennung von Form und Inhalt nicht so scharf und eindeutig möglich wie zum Beispiel bei XHTML und CSS (Cascading Style Sheets).

Um trotzdem eine möglichst weitgehende Trennung von Form und Inhalt zu ermöglichen, musste die Anwendung streng modular angelegt werden. Um darüber hinaus das Layout möglichst schnell in einem bestimmten Rahmen verändern zu können, wurde eine spezielle Layout-Klasse erstellt, in der die Rahmenbedingungen für das Layout vorgegeben sind und mittels der die gesamte Anwendung in ihrer Erscheinung mit wenigen Handgriffen entscheidend verändert werden kann. Dies setzt voraus, dass die gesamte Programmierung der Anwendung Rücksicht auf die in der Layout-Klasse gesetzten Parameter nimmt. Die entwickelte Anwendung ist also gleich dreifach dynamisch: Dynamische Inhalte, dynamischer Seitenaufbau durch Animationseinsatz und dynamisch generiertes Layoutraster.

**Dreifach dynamische  
Anwendung**

### 5.1.1 Linkstruktur

Die Linkstruktur des Projektes besitzt neben der Startseite fünf Hauptmenüpunkte: *Künstler*, *Releases*, *Labelinfos*, *Links* und *MyLabel*. Unter den Menüpunkten *Alle Künstler*, *Neue Releases* und *Alle Releases* befindet sich jeweils eine dynamisch angelegte Liste mit weitergehender Verlinkung auf die entsprechenden Künstler-Profile beziehungsweise auf die Informationsbereiche der einzelnen Releases. Abbildung 5.1 verdeutlicht die Linkstruktur der entwickelten Anwendung.

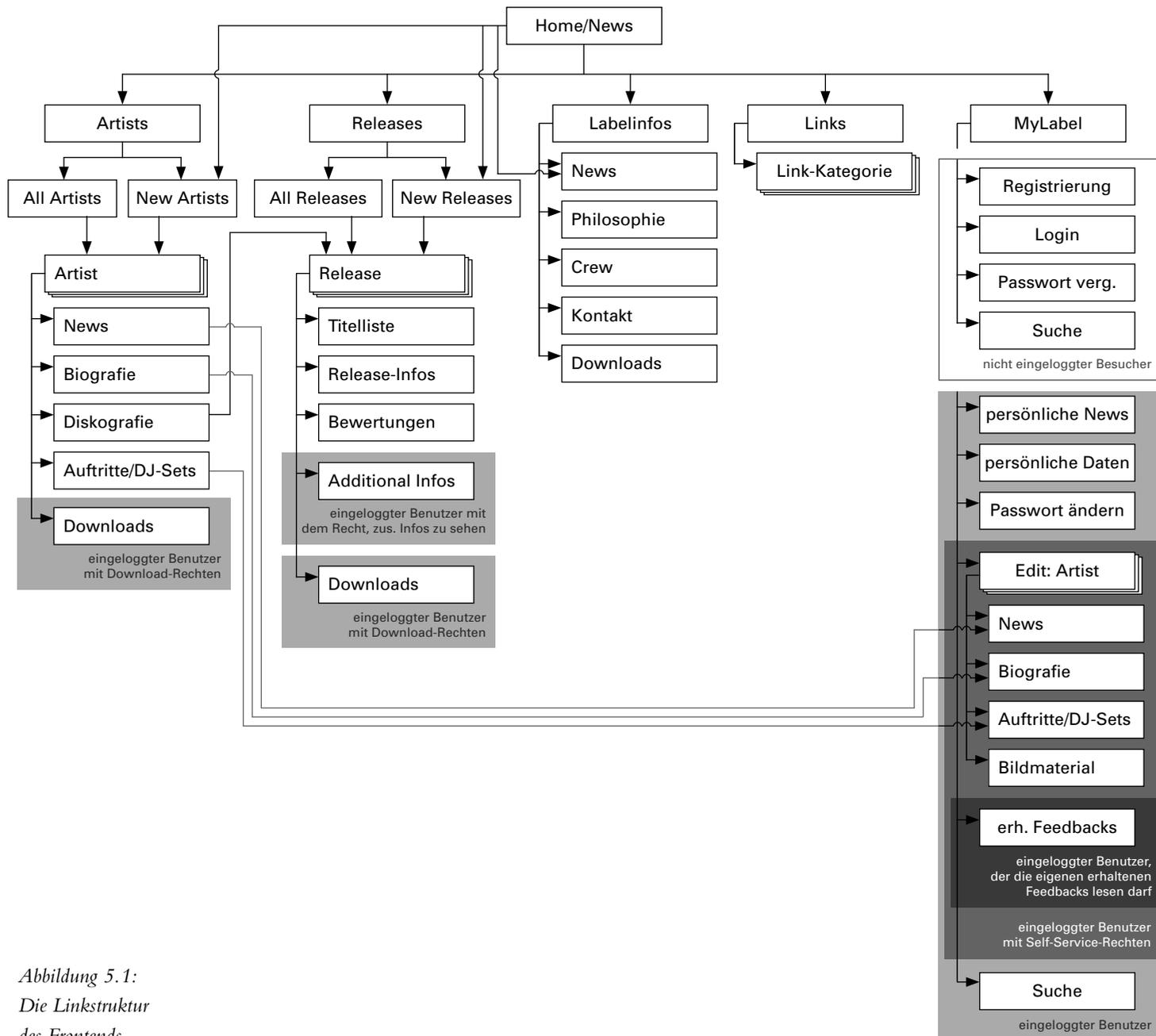


Abbildung 5.1:  
Die Linkstruktur  
des Frontends

### 5.1.2 Layout

Da die entwickelte Webanwendung ein sehr komplexes Bild über das Gesamtprogramm eines Recordlabels geben soll und einen sehr umfangreiches Informationsspektrum abdeckt, war es wichtig, ein Layout zu konzipieren, das dem Benutzer eine gute Orientierung ermöglicht. Der Benutzer soll ständig den Überblick darüber behalten können, wo er sich im Moment gerade befindet. Dazu wurde ein Layout-Raster erstellt, das aus acht Boxen besteht, die durch Stege voneinander getrennt sind. Die acht Boxen sind in vier Spalten und zwei Reihen aufgeteilt. Die obere Reihe bietet zum einen die Hauptnavigation, den Login-Bereich, Platz für das Logo und für Thumbnails des Bildmaterials, die zweite Reihe ist für die Visualisierung der Information und den Weg dorthin vorgesehen.

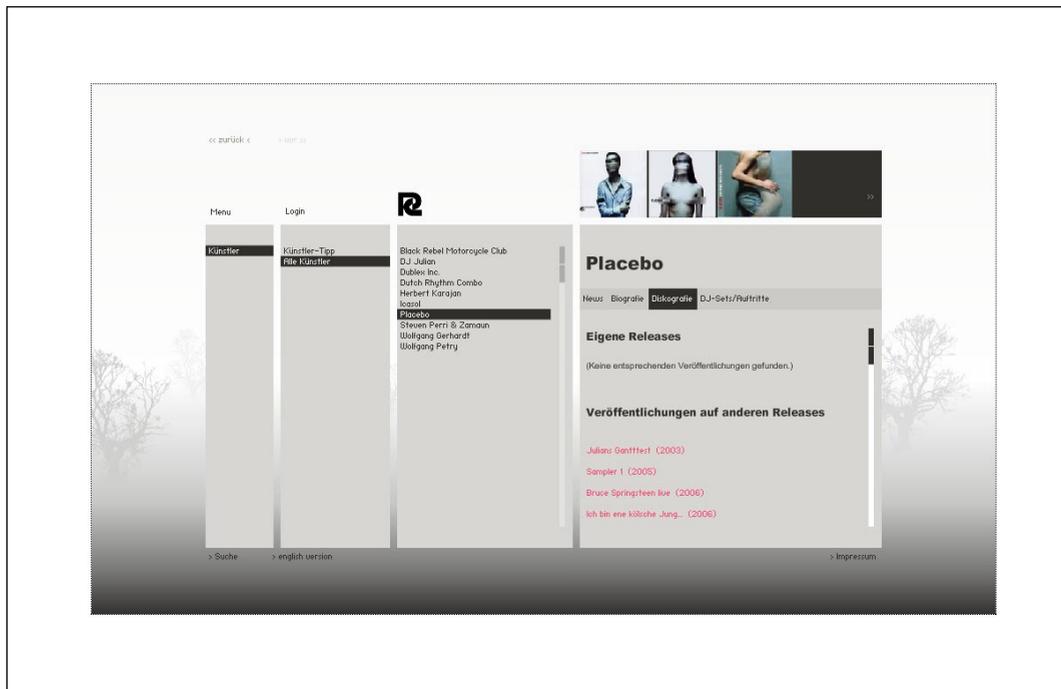


Abbildung 5.2:  
Layout-Raster. Der Benutzer  
kann immer schnell erfassen,  
wo er sich gerade befindet.

Der Benutzer kann so mit wenig Aufwand erfahren, wo er sich gerade befindet. Abbildung 5.2 zeigt einen exemplarischen Fall: Unter dem Hauptmenüpunkt „Künstler“ hat sich der Benutzer „Alle Künstler“ anzeigen lassen und sich dann für die Combo „Placebo“ entschieden. Auf der Informationsseite von Placebo wiederum interessiert er sich für die Diskografie und hat sich diese anzeigen lassen.

Von diesem Punkt aus kann er direkt zu einem der aufgelisteten Releases gelangen. Die Informationen in den ersten drei Boxen der unteren Reihe verändert sich dabei entsprechend, sodass der Benutzer sofort weiß, dass er sich nun in einem anderen Hauptmenüzweig befindet, und wie er demnächst auf direktem Wege zu dieser Information gelangen kann.

Die Farben sind bewusst gedeckt gehalten, um nicht vom eigentlichen Inhalt abzulenken. Außerdem sticht so das Bildmaterial besser ins Auge, das zudem einen angenehm auflockernden Charakter erhält.

Ein zu starker Einsatz von Farbe hätte im Zusammenhang mit dem animierten Seitenaufbau zudem bedeutet, dass auf den Benutzer zu viele Reize einwirken, die zum ersten von der Information ablenken, aber auch dazu führen könnten, dass sich der Benutzer überfordert fühlt.

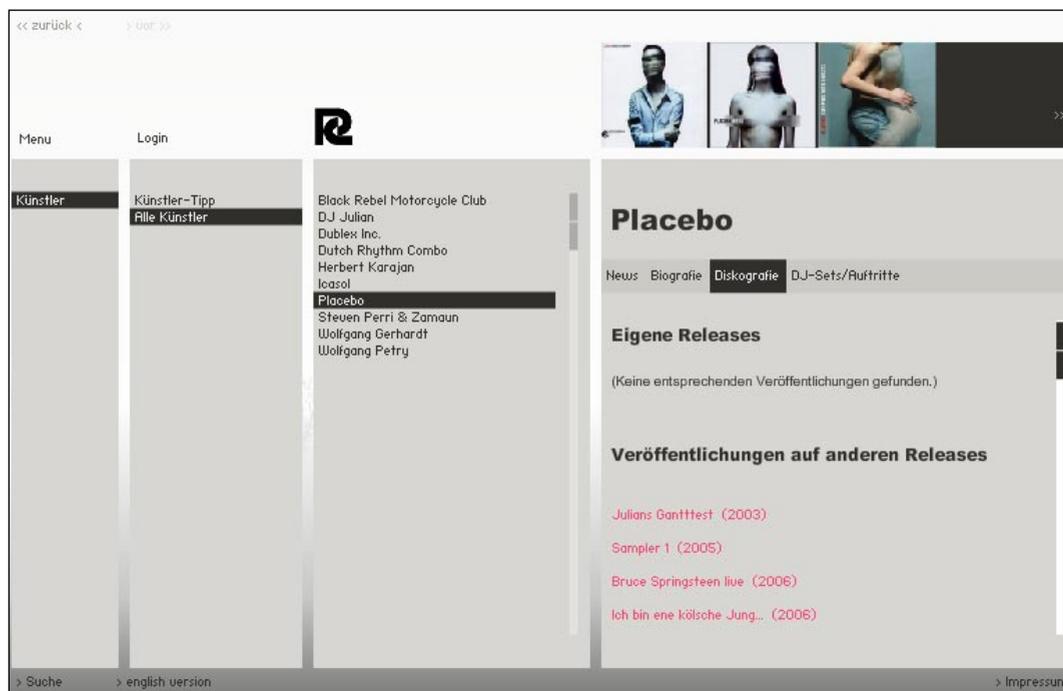


Abbildung 5.3:  
Ungefähre Darstellung auf  
einem Monitor mit einer Auf-  
lösung von 800 x 600 Pixel

Das Layout ist so angelegt, dass es mit einer Monitor-Auflösung von 1024 x 800 Pixel und darüber zwar am besten wirkt, aber auch mit einer Monitor-Auflösung von 800 x 600 Pixel noch bedienbar ist. In diesem Fall sind nur noch die Boxen selbst zu sehen.

## 5.2 Ein dynamisches Template in Flash (Layout-Klasse)

Wie im vorangegangenen Kapitel schon angesprochen, sollte die Frontend-Anwendung so entwickelt werden, dass mit wenig Aufwand ein neues Layout erstellt werden kann. So lässt sich die Gesamt-Anwendung leichter als Bundle an kleinere Recordlabels verkaufen, die sich die Entwicklung eines völlig eigenständigen individuellen Seitenlayouts nicht leisten können.

Mit wenigen Handgriffen lässt sich so vermeiden, dass jeder Kunde der Anwendung eine Standard-Website erhält. Eine gewisse Individualisierung ist somit möglich.

Problem bei der Entwicklung einer komplexen Macromedia-Flash-Anwendung ist jedoch, dass die Trennung von Form und Inhalt nicht so simpel zu trennen ist wie bei einer HTML-Lösung, da in der Anwendung selbst ja erst bestimmt wird, in welchem Gesamtrahmen die Information präsentiert werden soll.

Um die Anforderungen zu erfüllen, wurde eine eigene Layout-Klasse entwickelt. Diese Layout-Klasse ermöglicht es, die Größen aller acht Layout-Boxen anzupassen, sowohl in der Höhe wie auch in der Breite. Zudem sind die Farben für Schrift, Links, Mouseover-Effekte, Hintergrund und Linienelemente wählbar.

Das Layout lässt sich in der Hauptdatei mit wenig Aufwand noch weiter individualisieren: Hintergrundbild, Schriftart, Schriftschnitte und Schriftgröße sind mit wenigen Eingriffen geändert, da eine überschaubare Anzahl von MovieClips als Text- und Formularmodule eingesetzt werden. Gerade die Änderung der Schriftart lässt sich nicht anders regeln, da die Schriften nur über vorhandene MovieClips oder über eine in der Bibliothek abgelegte Schriftart eingebettet werden können, und danach der Hauptfilm erneut exportiert werden muss; es sei denn, es sollen die Standard-Systemschriften der Benutzerrechner verwendet werden.<sup>19</sup>

<sup>19</sup> Wobei dies nicht zu empfehlen ist, da die Kontrolle über die Schriften, insbesondere hinsichtlich der Laufweite, nicht gegeben ist.

### 5.2.1 Bedeutung für die Programmierung der Anwendung

Die Verwendung dieses dynamischen Template-Modells schließt mit ein, dass bei der weiteren Programmierung der Anwendung stetig Rücksicht auf die Layout-Klasse genommen werden muss. Jedes dynamisch erzeugte Content-Modul muss sich also in seiner Positionierung, seinem farblichen Erscheinungsbild und seinem Verhalten nach den Vorgaben der Layout-Klasse richten.

## 5.3 Sicherheitsaspekte in Flash-Anwendungen

*„In particular, developers are often unsure if Macromedia Flash applications can have the same level of security as a well-designed HTML-based application. The answer to that question is most definitely yes, if you design your application properly.“*

*(Hall 2002: oben)*

Branden Hall spricht hier ein wichtiges Thema an: Flash-Anwendungen sind nur so sicher, wie man sie programmiert. Prinzipiell ist der gleiche Sicherheitslevel mit Flash-Anwendungen möglich wie mit Anwendungen auf HTML-Basis.

### 5.3.1 Möglichkeiten und Grenzen „sicherer“ Anwendungen in Macromedia Flash

Zunächst sei angemerkt, dass es eine hundertprozentig sichere Webanwendung nicht geben kann. Man kann nur den Aufwand, der nötig ist, die Sicherheitsvorkehrungen zu knacken, so groß werden lassen, dass der Aufwand nicht mehr im Verhältnis zum Nutzen des Hackers steht (Fuhrberg/Häger/Wolf 2001: Seite 47):

*„Sie (die Gefährdungen, die beim Einsatz im Internet vorkommen können [Anm. d. Autors]) sind alle nur bedingt vermeidbar. Es können lediglich die Auswirkungen verringert und die Schwelle, die für ihre Ausnutzung überwunden werden muss, vergrößert werden.“*

Generell lässt sich sagen, dass mit Macromedia Flash durchaus sichere Webanwendungen erstellt werden können. Das Flash-Player-Plugin ist zum Beispiel auch in der Lage mittels des Browsers über SSL-verschlüsselte Verbindungen mit einem HTTPS-Server zu kommunizieren. Wichtig dabei ist jedoch zu beachten, dass sich die SWF-Datei dazu ebenfalls auf dem HTTPS-Server befinden muss. Eine SSL-verschlüsselte Verbindung stellt sicher, dass die übermittelten Daten von außen nicht eingesehen werden können.<sup>20</sup> Somit sind schon sehr weitreichende Sicherheitsvorkehrungen möglich.

<sup>20</sup> Sofern keine Hacker-Attacke in Form eines so genannten Man-In-The-Middle-Angriffs stattfindet.

Beim Einsatz von HTTPS und der `FileReference`-Klasse von Flash 8 ist jedoch zu beachten, das bei Dateiuploads auf einen HTTPS-Server unbedingt die Port-Nummer an die Server-URL anzuhängen ist, da diese bei der Macintosh-Version des Flash-Players standardmäßig mit 80 belegt ist.

Doch nicht immer besteht auch die Möglichkeit, einen HTTPS-Server zu nutzen. Für diesen Fall gibt es immer noch gute Möglichkeiten, Flash-Anwendung zumindest so sicher zu machen, dass Gelegenheits-Hackern zwar nicht völlig das Handwerk gelegt werden kann, aber die Schwelle zum Hacking-Erfolg genügend hoch angesetzt ist.

Leider wird immer noch in vielen Handbüchern zu Flash das Thema Sicherheit vor allem im Bereich Login viel zu leichtfertig behandelt. Dabei lässt sich in Flash-Anwendungen gerade ein enormer Sicherheitsvorteil nutzen: Die Versendung von sicherheitsrelevanter Information in verschlüsselter Form. Dies ist bei HTML-basierten Anwendungen nur durch den zusätzlichen Einsatz von clientseitigen Skripten (zum Beispiel JavaScript) oder den Einsatz von Java möglich. Java und JavaScript sind jedoch von vielen Benutzern wegen Sicherheitsbedenken deaktiviert. Wenn aber sowieso schon Flash für den Webauftritt eingesetzt wird, warum dann nicht gleich auf die vorhandenen Ressourcen zurückgreifen?

#### 5.3.1.1 Einweg-Verschlüsselung mit MD5 und SHA1

MD5 und SHA1 sind so genannte One-Way-Verschlüsselungen. Das heißt: Die Verschlüsselung ist nur unter sehr enormen Rechenanstrengungen umzukehren. Eine einmal mit MD5 oder SHA1 verschlüsselte Information ist also im Nachhinein quasi nicht mehr zurückzugewinnen. Was nützt also diese Verschlüsselung, bei der der eigentliche Inhalt nicht mehr zugänglich ist? MD5- und SHA1-Verschlüsselungen eignen sich sehr gut als Vergleichswerte zum Beispiel bei Login-Funktionen. Dabei wird das Passwort eines Benutzers nicht im Klartext gespeichert, sondern zum Beispiel als MD5-verschlüsselte Zeichenkette in der Datenbank hinterlegt. Meldet sich der Benutzer nun erneut an, wird sein von ihm eingegebenes Passwort zur Kontrolle ebenfalls mit MD5 verschlüsselt und die daraus gewonnene Zeichenkette mit der in der Datenbank abgelegten Zeichenkette verglichen. Stimmen die beiden Zeichenketten miteinander überein, war der Login erfolgreich.

Sollte in solch einem Fall ein Hacker an die Inhalte der Datenbank und somit auch an die verschlüsselten Passwörter gelangen, hat er davon keinen Vorteil. Er kann die ursprünglichen Passwörter nicht sehen.

Wenn das Passwort schon clientseitig verschlüsselt wird, lässt sich somit zusätzlich vermeiden, dass das Passwort mittels einer so genannte Sniffing-Attacke mitgelesen werden kann.

*„Mit Sniffing werden Angriffe bezeichnet, bei denen Informationen durch das Mitlesen von Daten bei einer TCP/IP-Datenübertragung gewonnen werden. Die Ursache für diese Angriffsmöglichkeit liegt in der offenen Übertragung von Daten im Internet und den dort eingesetzten Protokollen.“*  
(Fuhrberg/Häger/Wolf 2001: Seite 63)

Zwar könnte der Hacker in dieser Situation einen Login mit Benutzernamen und der beim Sniffing herausgefundenen MD5-Zeichenkette als Passwort in die Wege leiten. Jedoch ist ihm dies nicht über das Login-Modul der Client-Anwendung möglich (da dann ja die MD5-Zeichenkette erneut verschlüsselt würde). Er muss also quasi selbst eine Anwendung programmieren, die sich genauso verhält wie die Original-Anwendung, mit der Ausnahme, dass sie das eingegebene Passwort nicht verschlüsselt. Das ist ein ganzes Stück weit aufwändiger, als nur das im Klartext übermittelte Passwort mitzulesen und sich danach bequem mit Benutzernamen und Passwort über die vorhandene Anwendung einzuloggen.

Flash selbst bietet in seiner aktuellen Version noch keine eigene Verschlüsselungsfunktion an, es gibt im Internet jedoch genug Ressourcen, wo der Code für eine SHA1- oder eine MD5-Verschlüsselung frei verfügbar ist.<sup>21</sup> Hier ist auch von Vorteil, dass sich JavaScript und ActionScript zum Verwechseln ähnlich sind: Zur Not kann eine MD5-JavaScript-Funktion ohne große Mühe an ActionScript angepasst werden. Der Schritt zu einer MD5-Verschlüsselungs-Klasse in ActionScript-2-Code ist dann auch nicht mehr weit.<sup>22</sup>

### 5.3.1.2 Grenzen und Risiken

Manch unerfahrener Flash-Programmierer ist leider immer noch der Ansicht, dass es ja vollkommen ausreichen würde, Benutzerkennung und Passwort in der SWF-Datei selbst abzuspeichern, da ja im Gegensatz zu HTML und JavaScript der Scriptcode nicht einsehbar sei. Dies ist ein fataler Irrtum! Es gibt mittlerweile genügend so genannte Decompiler-Tools, die aus einer SWF-Datei wieder eine FLA-Datei erstellen können, inklusive aller MovieClips, Grafiken, Texte und auch ActionScript-Codes! Es reicht also vollkommen aus, die übertragene SWF aus dem Browser-Cache zu holen, zu dekompileieren, sich den Code genauer anzuschauen und schon ist man im Besitz der Zugangsdaten. Auch mit einer extern geladenen SWF-Datei ist da nicht viel Abhilfe getan.

**SWFs sind leicht dekom-  
pilierbar!**

<sup>21</sup> Eine gute Adresse für MD4-, MD5- und SHA1-JavaScript-Funktionen ist zum Beispiel die Seite von Paul Johnston, erreichbar unter: <http://pajhome.org.uk/crypt/md5>

<sup>22</sup> Der Code für eine ActionScript-2-MD5-Klasse befindet sich im Anhang dieser Arbeit

Für einen sicheren Login kommt man um einen Datenbank-gestützten Login am besten in Kombination mit einer Sessionverwaltung nicht herum.

Die Tatsache, dass sich SWF-Dateien recht leicht decompilieren lassen, ist auch der Grund dafür, dass Anwendungen mit einem sehr hohen Sicherheitsanspruch nicht mit Macromedia Flash umgesetzt werden sollten.

Fast unüberwindbare Probleme bekommt man auch dann, wenn man eine sichere Highscore-Übertragung entwickeln will. Das Problem ist hier nicht die offene also einsehbare Übertragung per TCP/IP, sondern der Schutz vor der Übermittlung von gefälschten Highscore-Daten. Dies ist nicht im Speziellen ein Problem von Macromedia-Flash-Anwendungen. Da aber viele Internet-Spiele mit Macromedia Flash entwickelt werden, sollten diese Risiken jedoch bei der Spiele-Entwicklung und einer eventuellen Preisvergabe für den Höchstplatzierten unbedingt mitbedacht werden. Wirklich sichere Lösungen lassen sich in diesem Fall nur mit einem Flash Media Server<sup>23</sup> umsetzen.

### Highscore-Übertragung

## 5.3.2 Umsetzung im Projekt

Für den praktischen Teil der vorliegenden Diplomarbeit galt es, eine möglichst sichere Anwendung zu entwickeln, die jedoch für den Benutzer ohne großen Aufwand bedienbar ist.

### 5.3.2.1 Session-Verwaltung

Der gesamte Webauftritt ist Sitzungs-gestützt, das heißt, jeder Besucher des Webauftrittes bekommt vom Server eine individuelle Session-ID zugeteilt. Die serverseitige Sessionverwaltung übernimmt PHP. In der Flash-Anwendung muss dazu nichts weiter veranlasst werden. Im Gegenteil: Würde auch hier die Session-ID gespeichert und bei der Kommunikation mit den Server-Skripten mitgeschickt, machte man die Anwendung leichter angreifbar.

### 5.3.2.2 Login

Für den zugangsgeschützten Bereich muss sich der Benutzer zunächst mit einem Benutzernamen und einem Passwort einloggen. Das Passwort wird dabei noch in der Flash-Anwendung per MD5

<sup>23</sup> vormals Flash Communication Server

verschlüsselt, so ist sichergestellt, dass das Passwort nicht im Klartext zum Server übermittelt wird. Bei einem erfolgreichen Login wird dem Benutzer eine neue Session-ID zugeteilt, über die der Benutzer ab diesem Zeitpunkt in Zusammenhang mit seinem Benutzernamen identifiziert wird. Die neue Session-ID wird dazu in der Datenbank abgelegt. Bei jeder Anfrage wird nun nur noch der Benutzername übermittelt. Serverseitig wird geprüft, ob eine Session-ID zu diesem Benutzer vorhanden ist, und ob sie mit der laufenden Sitzung übereinstimmt. Zudem wird die Sitzung auf ihre zeitliche Gültigkeit hin überprüft.

Bei Ablauf der Sitzung oder bei einem Logout wird die in der Datenbank gespeicherte Session-ID zurückgesetzt und die Sitzung serverseitig zerstört.

### 5.3.2.3 Beschränkungen bei Benutzer-Eingaben

Alle eingegebenen Texte, die der Benutzer in Eingabefeldern machen kann, werden auf Gültigkeit hin überprüft. So wird in einem Eingabefeld für eine Telefonnummer auch nur eine Telefonnummer anerkannt.

Bedauerlich ist in diesem Zusammenhang, dass Macromedia Flash in der aktuellen Version noch keine regulären Ausdrücke verarbeiten kann. Kontrollmechanismen für die korrekte Dateneingabe mussten also in eigens angefertigten Kontroll-Funktionen entwickelt werden.

Serverskriptseitig wird nochmals sichergestellt, dass keine ungültigen Zeichen oder gar sicherheitsbedenklichen Zeichenfolgen an die Datenbank weitergegeben werden.

### 5.3.2.4 Dateiupload

Ein weiterer sicherheitstechnisch bedenklicher Punkt ist der Dateiupload. Hier muss sichergestellt werden, dass die hochgeladenen Dateien auch tatsächlich dem gewünschten Typ entsprechen und nicht etwa ausführbare sicherheitsbedenkliche Skripten sind.

Zwar lässt sich mit der `FileReference`-Klasse die Auswahlmöglichkeit der zu übertragenden Dateitypen einschränken, jedoch ist diese Vorgehensweise allein noch nicht ausreichend. Die `browse()`-Methode der `FileReference`-Klasse erwartet dazu ein Objekt mit der `description` (Dateityp-Beschreibung) und der `extension` (Dateiendung) der im Auswahldialog auswählbaren Dateien. Diese Vorauswahl lässt sich jedoch austricksen, indem die Dateiendung einer schädlichen Datei vorher in eine zulässige Endung umgewandelt wird (aus `schaden.exe` wird somit zum Beispiel `schaden.jpg`).

Zu diesem Zeitpunkt wäre die Datei nicht zwingend gefährlich, so lange sie niemand mit einem Doppelklick öffnen würde. Im Dateiauswahl-Dialog des Betriebssystems lässt sich jedoch zumeist der Dateiname nachträglich wieder ändern (aus `schaden.jpg` würde so wieder `schaden.exe`). Genau dieser Dateiauswahl-Dialog wird vom Flash-Player bei einem Upload angefordert. So lässt sich also ohne große Mühe ein schädliches Skript auf den Server laden. Um dort einen Schaden zu vermeiden, muss also serverseitig vor dem Abspeichern der übertragenen Daten unbedingt noch einmal überprüft werden, ob der Dateityp der übertragenen Datei mit dem gewünschten Dateityp tatsächlich übereinstimmt.

Dies ist leider in manchen Fällen nur über die Kontrolle der Dateiendung möglich, da der Flash-Player den MIME-Type von hochzuladenden Dateien zumeist als `application/octet-stream` angibt.

## 5.4 Unechtes Streaming – Downloadschutz zur Wahrung der Urheberrechte

*„Das Urheberrecht schützt persönliche geistige Schöpfungen auf dem Gebiet der Musik, Wissenschaft, Kunst und Literatur. [...] Der auf künstlerisch und technischem Gebiet kreativ Schaffende wird vor fremdem Zugriff auf seine Arbeitsergebnisse geschützt. Damit sichert ihm das Urheberrecht eine Beteiligung am wirtschaftlichen Nutzen, der aus seinen kreativen Leistungen gezogen werden kann.“*

*(Böhringer/Bühler/Schlaich/Ziegler 2000: Seite 454)*

### 5.4.1 Problematiken

*„Im Bereich der juristisch so bezeichneten ‚unkörperlichen Verwertung‘ von Musikstücken sollen die Urheber angemessen am wirtschaftlichen Nutzen ihrer Werke beteiligt werden. Etwa dadurch, dass die Urheber bzw. Komponisten prozentual an den vom Anbieter und Verbreiter eines Musiktitels erzielten Einnahmen beteiligt werden. Ausschlaggebend ist der geldwerte Vorteil, den der Anbieter durch die Nutzung der Musik erreicht. Geldwerte Vorteile sind dabei in erster Linie Abonnements- oder Einzelnutzungsgebühren, aber auch etwaige Werbe- und Sponsorengelder.“*

*(Böhringer/Bühler/Schlaich/Ziegler 2000: Seite 475)*

Zur Wahrung der Rechte der Musikschaftenden arbeitet die GEMA, die deutsche Gesellschaft für musikalische Aufführungs- und mechanische Vervielfältigungsrechte. Leider geraten gerade kleinere Independent Recordlabels durch das Urheberrecht und die Regelungen der GEMA in eine Zwickmühle: Zum einen soll das Recordlabel die Musik der Künstler, die bei ihm unter Vertrag stehen, promoten und erfolgreich verkaufen. Auf der anderen Seite aber darf das Recordlabel selbst zu verkaufsfördernden Zwecken die Musik nicht ohne weiteres öffentlich abspielen oder vollständig als Hörprobe anbieten. Das Recordlabel ist in solch einem Fall dazu verpflichtet, selbst GEMA-Gebühren für die Musik der bei ihm unter Vertrag stehenden Künstler zu zahlen.

Wie in Kapitel 3.3.1 schon erläutert, ist bei Hörproben auf einem Recordlabel-Webauftritt sicherzustellen, dass die Musikdateien nach dem Anhören nicht als identische Kopie auf dem Benutzerrechner

vorliegen.<sup>24</sup> Ansonsten wird das Urheberrecht verletzt. Zudem ist ein Pauschale an die GEMA in Höhe von 150,- Euro zu entrichten.

Als Lösung wird von der GEMA Streaming vorgeschlagen. Echte Streaming-Lösungen sind jedoch teuer in der Umsetzung und für viele kleinere Recordlabels finanziell nicht leistbar. Ziel war es also, ein Anhör-Konzept zu entwickeln, das zwar vollständige Musikstücke zur Hörprobe anbietet (vor allem wichtig für die Presse und für die DJ-Previews), zugleich aber verhindert, dass die Musikdatei danach als Kopie auf dem Benutzerrechner vorliegt. Zudem sollte es wesentlich günstiger und bequemer umzusetzen sein als der Einsatz einer echten Streaming-Technologie.

### 5.4.2 Lösungsansatz und Umsetzung

Die hier vorgestellte Lösung lässt sich prinzipiell mit jeder Progressive-Download-fähigen Abspielsoftware umsetzen. Am einfachsten und sichersten gelingt dies jedoch mit Macromedia Flash. Bei einer anderen Abspielsoftware muss sichergestellt sein, dass diese auch tatsächlich bei der Musikdatei-Übermittlung startet, ansonsten öffnet sich der Speichern-unter-Dialog des Browsers mittels dessen eine \*.php-Datei auf die Festplatte gespeichert werden kann.<sup>25</sup>

Angemerkt sei hier noch einmal, dass diese Umsetzung keine echte Streaming-Lösung darstellt. Es ist ein Progressive-Download, bei dem die Datei nach dem Anhören sofort automatisch gelöscht wird.

Zunächst einmal muss sichergestellt werden, dass das Verzeichnis, in dem die Original-Musikdateien liegen, zugriffsgeschützt ist. So wird vermieden, dass die Musikdateien direkt aus dem Verzeichnis geladen werden können, sollte der Pfad dorthin auf irgendeinem Weg bekannt werden. Dies ist am einfachsten und sichersten mit einer .htaccess-Datei im entsprechenden Verzeichnis zu erreichen.<sup>26</sup> Ein Zugriffsschutz durch eine .htaccess-Datei bewirkt, dass bei allen Zugriffen, die über HTTP stattfinden, eine Identifizierung mittels Benutzername und Passwort vom Server verlangt wird. Serverskripten, die sich auf dem selben Server befinden, können jedoch ungehindert auf das Verzeichnis zugreifen.

<sup>24</sup> Es sei denn, es handelt sich nur um kurze Ausschnitte eines Musikstückes.

<sup>25</sup> Was der Benutzer eventuell nicht weiß, ist, dass sich hinter dieser \*.php-Datei eigentlich eine \*.mp3-Datei versteckt. Das Problem, dass die Musikdatei nicht als Kopie auf dem Benutzer-Rechner vorhanden sein soll, wäre damit jedoch alles andere als gelöst.

<sup>26</sup> Voraussetzung ist, dass der Server zum NCSA-Server kompatibel ist. Dies ist bei Apache-Servern zum Beispiel der Fall. Auf anderen, nicht NCSA-Server-kompatiblen Servern gibt es jedoch zumeist ein vergleichbares Konzept.

Für weitere Infos siehe <http://de.selfhtml.org/servercgi/server/htaccess.htm>

Clientseitig reicht ein einfaches in Macromedia Flash erstelltes Musikwiedergabe-Modul oder eine andere Progressive-Download-fähige Abspielsoftware mit den oben genannten Einschränkungen. In Macromedia-Flash muss dazu nur die `loadSound()`-Methode eingesetzt werden, bei der als Parameter die URL zu dem Serverskript angegeben wird. Die URL enthält zusätzlich als angehängte GET-Parameter eine Track-ID und idealerweise die Identifizierungs-Daten des Benutzers.

Das Serverskript überprüft nun anhand der Identifizierungs-Daten das Zugriffsrecht auf das angeforderte Musikstück.<sup>27</sup> Besitzt der Benutzer das entsprechende Zugriffsrecht, und ist das Musikstück tatsächlich auf dem Server vorhanden, wird vom Serverskript eine MP3-Ausgabe generiert. Dazu wird zunächst ein entsprechender Datei-Header aufgebaut, der als zusätzliche Informationen Angaben über das Caching-Verhalten beinhaltet. Diese Caching-Angaben veranlassen die Abspielsoftware auf dem Benutzerrechner, die gelieferte Datei nur so lange im Cache vorzuhalten, wie es für das Abspielen der Datei nötig ist. Nachdem der erste Titel abgespielt worden ist und zum Beispiel ein nächster Titel startet, wird also die Datei vom ersten Titel aus dem Cache gelöscht.

```
// Header erzeugen
header("Pragma: public");
header("Expires: Tue, 11 Oct 1977 00:54:00 GMT");
header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
header("Cache-Control: no-store, no-cache, must-revalidate");
header("Content-Type: audio/x-mpeg, audio/x-mpeg-3, audio/mpeg3");
header("Content-Length: ".$entityParser(@urlencode(@filesize($mp3Track))));
```

Die wichtigen Angaben in diesem Zusammenhang sind das Ablaufdatum (`Expires`), das in der Vergangenheit angesiedelt sein muss und die Angaben zum Caching-Verhalten (`Cache-Control`).

Nachdem der Header erstellt worden ist, wird nun die Original-Datei Stück für Stück im Bytecode ausgelesen und ausgegeben.

Der Benutzer kann nun das Musikstück schon hören, bevor es vollständig heruntergeladen ist. Nach dem Abspielen liegt jedoch keine Kopie der Originaldatei auf seinem Rechner vor. Angemerkt sei jedoch, dass der Benutzer während der Wiedergabe auf den Cache des Flash-Players zugreifen könnte, um eine Kopie von der übermittelten Datei zu erstellen. Dieses System ist also nicht hundertprozentig sicher, erschwert das Herankommen an eine Kopie jedoch deutlich. Außerdem dürfte einem Benutzer, der diesen Schutz hintergeht, klar sein, dass sein Vorgehen nicht im Sinne des Anbieters ist,

*Listing 5.1:  
PHP-Skript zum Aufbau  
eines Headers für eine MP3-  
Datei, die nicht im Browser-  
Cache gespeichert bleiben soll*

<sup>27</sup> Um Missbrauch zu vermeiden, sollte die Überprüfung des Zugriffsrechtes die laufende Session natürlich miteinbeziehen.

da die Tatsache, dass die Musikdatei nur zur Laufzeit und nicht darüber hinaus auf seinem System gespeichert wird, deutlich macht, dass es nicht gewünscht ist, dass die Musikdatei auf seinem System verbleibt.

## 5.5 Miteinbeziehen der Benutzer

Um die Benutzer dazu zu bewegen, möglichst häufig seinen Webauftritt zu besuchen, muss der Anbieter entweder ständig wechselnde Neuigkeiten bieten oder er muss die Benutzer mit in den Webauftritt miteinbeziehen (Jacobsen 2005: Seite 72): „Virtuelle Gemeinschaften oder *Communities* sind die beste Möglichkeit, Benutzer an die Site zu binden.“ Eine Community im eigentlichen Sinne wird mit dem vorliegenden Projekt nicht umgesetzt, jedoch können registrierte Benutzer an mehreren Stellen aktiv an dem Webauftritt partizipieren.

### 5.5.1 Personalisierte Angebote

Gefällt dem Benutzer die Musik des Recordlabels hat er beim vorliegenden Projekt die Möglichkeit, sich zu für den zugangsgeschützten Bereich zu registrieren. Sofern gewünscht, kann sich der Benutzer zudem dafür bewerben, in das Adressbuch des Recordlabels mitaufgenommen zu werden. Dies muss er jedoch begründen und zusätzlich seine Adressdaten mit angeben. Sinnvoll ist diese Funktion zum Beispiel für Presse-Akkreditierungen, interessierte DJs und potentielle neue Künstler. Backend-seitig kann so jeder angemeldete Benutzer einer bestimmten Kontaktgruppe zugewiesen werden, die spezielle Zugriffsrechte besitzt.

Auf Wunsch bekommt der Benutzer einen Newsletter, der ihn über die Aktivitäten des Recordlabels und dessen Künstler auf dem Laufenden hält. Die Newsletter sind Backend-seitig individualisierbar, sodass jede Kontaktgruppe einen speziell zugeschnittenen Newsletter erhalten kann. Gut gemachte Newsletter sind eine gute Möglichkeit, die Benutzer an die Site zu binden (Jacobsen 2005: Seite 294):

*„Newsletter sind eine Möglichkeit, Ihre Besucher regelmäßig an Ihre Site zu erinnern. Sie entscheiden sich freiwillig dafür, immer wieder Werbung von Ihnen zu bekommen. Dies ist für Sie kostengünstig in Herstellung und Versand.“*

Aber trotzdem ist Vorsicht angebracht (Jacobsen 2005: Seite 295):

*„Wird ihr Newsletter als Spam empfunden, bringt er mehr Schaden als Nutzen. [...] Überlegen Sie, was Ihre Zielgruppe interessiert und richten Sie die Inhalte danach aus.  
Im Newsletter können Sie zum Beispiel Neuigkeiten, auf Sonderangebote bzw. -aktionen hinweisen“*

Das Recordlabel kann den registrierten Nutzern so noch weitere kleine Zusatzfeatures zukommen lassen. So kann für jeden Benutzer eine persönliche Playlist erstellt werden, die der Benutzer exklusiv in seinem zugangsgeschützten Bereich einsehen und anhören kann. Der Benutzer kann weitere Informationen zu jedem Track direkt in dieser Playlist abrufen, in der gewohnten Form.

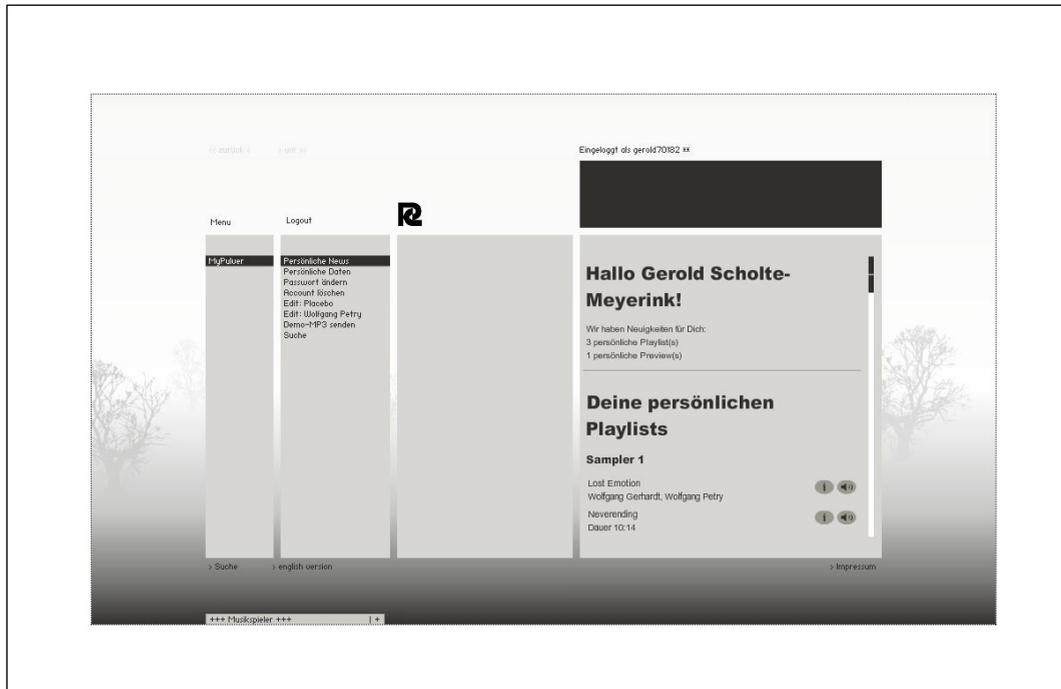


Abbildung 5.4:  
Der persönliche Bereich  
„MyPulver“ mit einer  
persönlichen Playlist

So könnte man zum Beispiel exklusiv einen noch nicht veröffentlichten Remix präsentieren. Der Benutzer wird nach dem Login am Kopf der Seite darüber informiert, dass neue Playlists oder DJ-Previews (siehe Kapitel 5.5.2) für ihn vorliegen. Auch wäre es denkbar, Benutzern, die freiwillig ihr Geburtsdatum angegeben haben, zu ihrem Geburtstag als Geschenk eine besondere Playlist mit Raritäten zum Anhören anzubieten. Solche kleinen Aufmerksamkeiten werden vom Benutzer positiv registriert und binden ihn weiter an den Webauftritt.

Je nachdem, zu welcher Kontaktgruppe der Nutzer gehört, kann er im gesamten Webauftritt zusätzliche Informationen zu den Künstlern, den Releases, den einzelnen Tracks und zum Recordlabel selbst erhalten. So bekommen zum Beispiel Pressevertreter zu jedem Release, zu jedem Künstler und zum Recordlabel selbst eine Pressemappe zum Download angeboten. Pressevertreter müssen sich also nicht extra in einen separaten Pressebereich einloggen, sondern bekommen die zusätzlichen Informationen

genau an Ort und Stelle.<sup>28</sup> Die Navigation und der Seitenaufbau bleiben in der gewohnten Art und Weise bestehen.

### 5.5.2 DJ-Previews

Benutzer, die in die Kontaktgruppe der DJs eingetragen sind und das entsprechende Zugriffsrecht besitzen, bekommen die Möglichkeit, Tracks und Releases kurz vor Erscheinungsdatum anzuhören und zu bewerten.

Diese als DJ-Preview oder DJ-Test bezeichnete Funktion nimmt den Mitarbeitern des Recordlabels viel Arbeit ab. So ist zentral hinterlegt, wer welche Feedbacks gegeben hat, wem welche Releases gut gefallen haben und wer aktiv am Feedback teilnimmt. Das Feedback kann direkt im Backend redigiert und für die Veröffentlichung freigegeben werden.

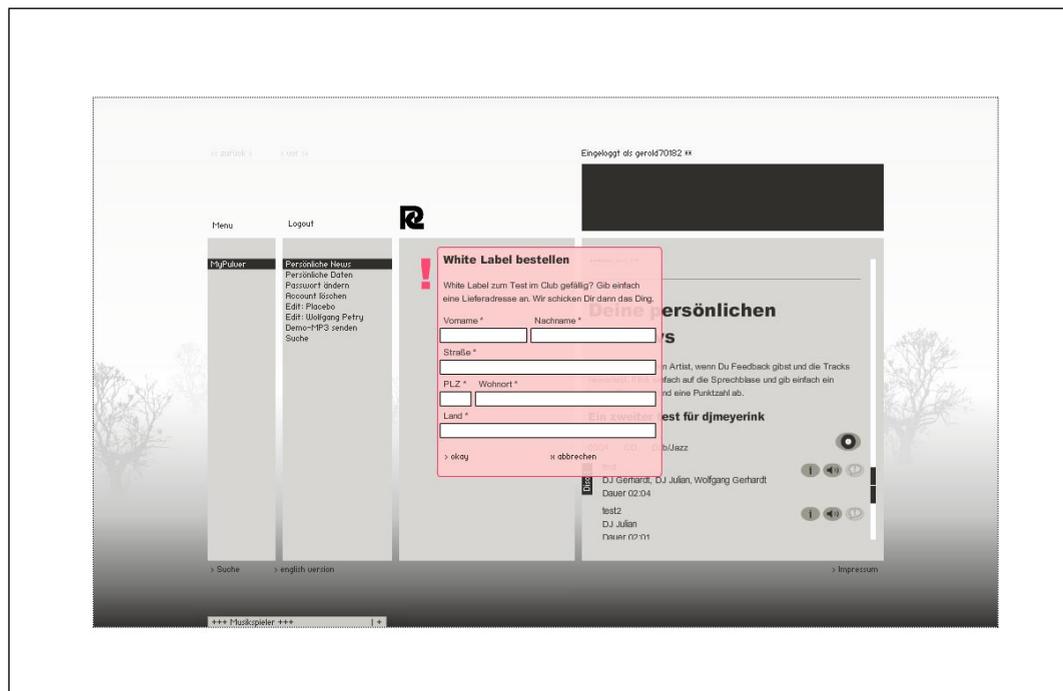


Abbildung 5.5:  
Der Bestellvorgang für ein  
White-Label

<sup>28</sup> Auch ein nachträglicher Login kann direkt an Ort und Stelle erfolgen. Der Seitenaufbau bleibt dabei genau gleich, er wird lediglich um die erweiterten Informationen ergänzt.

Für den eingeloggt DJ ist der Ansporn, eine gute konstruktive Kritik abzugeben, natürlich höher, wenn er weiß, dass seine Kritik unter Umständen auch auf dem Webauftritt veröffentlicht wird.

Über die White-Label-Bestellfunktion kann der DJ zudem eine Ausführung als Vinyl-Platte bestellen, um den Release schon vorab einmal im Club, quasi unter Live-Bedingungen, testen zu können.

### 5.5.3 Self-Services

Künstler, die bei dem Recordlabel schon unter Vertrag sind, können die Informationen, die sie selbst betreffen, über das Frontend direkt ändern und ergänzen. Dies betrifft Neuigkeiten allgemein, die Biografie, Angaben zu Auftritten und Bildmaterial. Alle diese Informationen sind jedoch auch von Seiten des Recordlabels über das Backend editierbar. Änderungen, die Frontend-seitig gemacht werden, können so auch im Backend kontrolliert werden.

Die Möglichkeit, dass der Künstler seine eigenen Informationen ändern kann, stärkt das Vertrauen und die Bindung zu seinem Recordlabel.

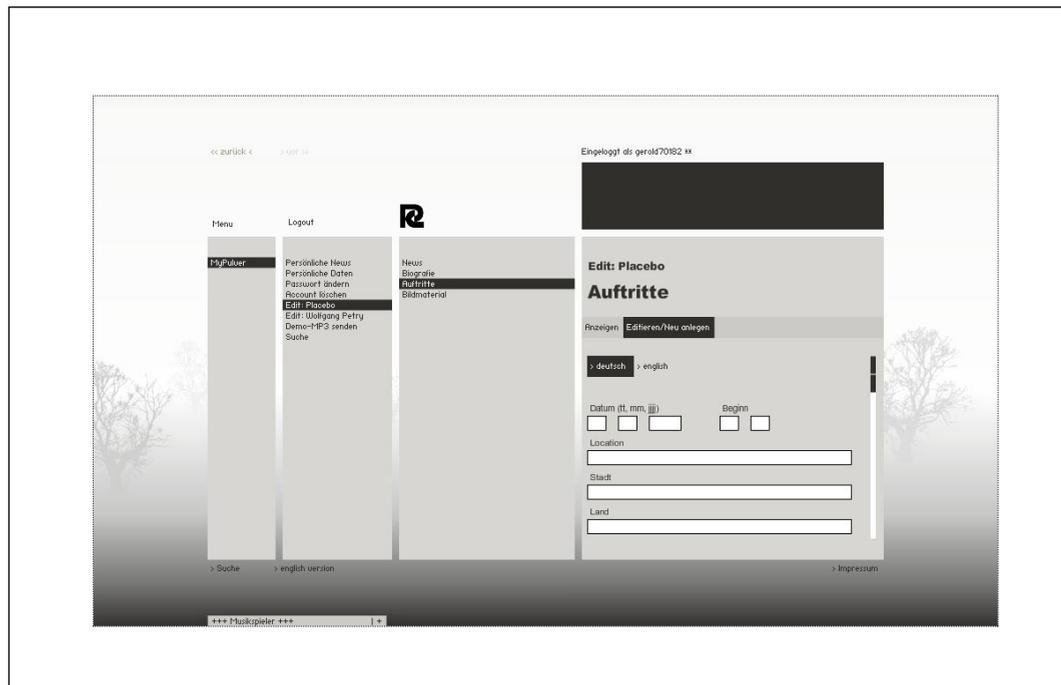


Abbildung 5.6:  
Self-Service: Neuanlegen  
eines Auftritt-Termins

### 5.5.4 Demo-Upload

Alle angemeldeten Benutzer besitzen zusätzlich die Möglichkeit, sich als Künstler mit einem Demo-Track zu bewerben. Die Hoffnung ist, dass die große Zahl von CD- und Tape-Einsendungen beim Recordlabel so deutlich reduziert werden kann. Die Auswahl von guter Musik ist so für die Mitarbeiter des Recordlabels stark vereinfacht. Vor allem die Möglichkeit, sofort nach dem Anhören ein Feedback zu geben (zum Beispiel eine personalisierte Standard-Absage) hat eine große Zeitersparnis zur Folge.<sup>29</sup>

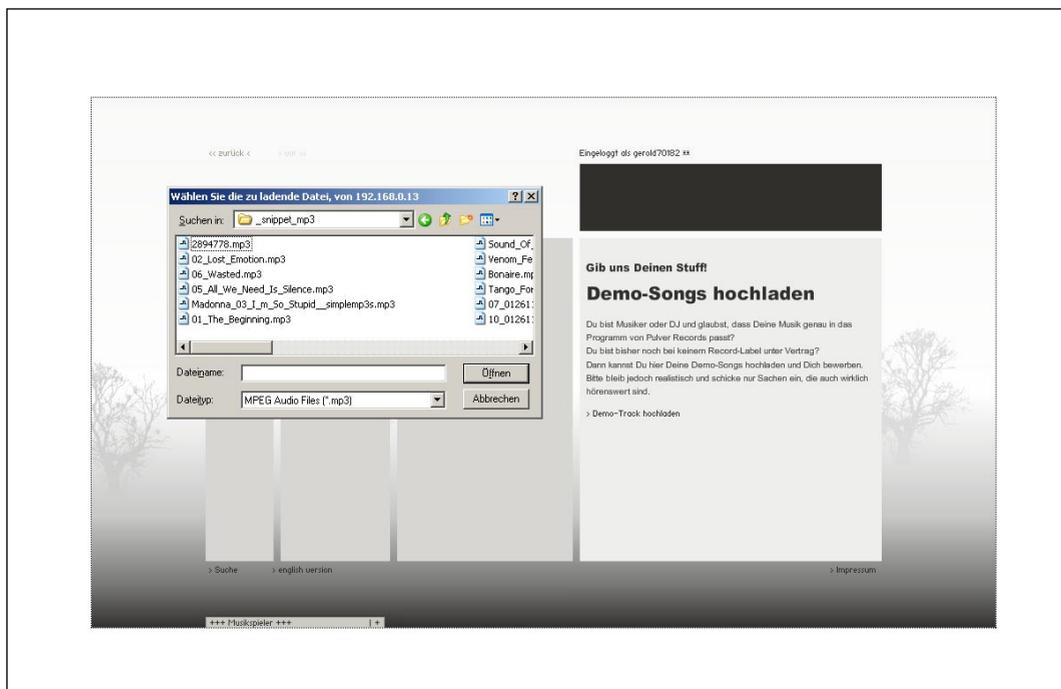


Abbildung 5.7:  
Datei-Auswahl-Dialog beim  
Upload eines Demo-MP3s

Der Demo-Upload ist mit der neuen FileReference-Klasse von Macromedia Flash 8 entwickelt worden. Dazu muss gleich angemerkt werden, dass die Umsetzung dieser neuen Klasse in manchen Teilen nicht ganz ausgereift scheint und es in mancherlei Hinsicht nicht einfach macht, wirklich sichere und komfortable Anwendungen zu erstellen. Die Verwendung der FileReference-Klasse

<sup>29</sup> Für weitere Informationen zur Backend-Realisierung siehe Julian Servatius: „Entwicklung einer Networking- und Vertriebsplattform für Recordlabels mit dem Schwerpunkt Office-Funktionen“, Seiten 16 f. und Seite 63

bietet jedoch auch Vorteile, vor allem hinsichtlich des Komforts für den Benutzer. Denn im Gegensatz zu HTML-basierten Umsetzungen mit PHP als Serverskript-Sprache ist es mittels der Upload-Funktionen der `FileReference`-Klasse in Flash möglich, eine echte Status-Anzeige zu ermöglichen. Dies ist vor allem bei großen Upload-Dateien (um die es sich ja bei Demo-MP3-Dateien handelt) sehr hilfreich, da der Benutzer eine klare Rückmeldung bekommt, wie weit der Upload fortgeschritten ist und wichtiger noch, dass der Upload auch tatsächlich Fortschritte macht.

An sich ist die Entwicklung eines Uploads in Macromedia Flash nicht wirklich schwer. Man muss nur das entsprechende Package (`flash.net.FileReference`) importieren, ein Objekt der `FileReference`-Klasse erzeugen und diesem ein Listener-Objekt hinzufügen, das die Funktionen `onSelect`, `onProgress`, `onComplete` und `onHTTPError` bereithält. Darüber hinaus können noch Funktionen wie `onOpen` und `onCancel` definiert werden, diese sind jedoch nicht unbedingt nötig. Die `onSelect`-Methode wird benötigt, um nach der Auswahl der Datei einen sicheren Dateinamen generieren zu können (ohne Sonder- und Leerzeichen, um Probleme auf dem Server zu vermeiden). Der Dateiname der zu übertragenden Datei kann jedoch noch nicht vom Flash-Player verändert werden. Dies lässt das Sicherheitskonzept des Flash-Players nicht zu. Drum schickt man am besten den bereinigten Dateinamen per angehängten GET-Parameter über die URL des Serverskriptes mit. Erst das Serverskript ändert nun den Dateinamen der übermittelten Datei und legt diese, nachdem sie auf Sicherheitsrisiken überprüft wurde, in das Zielverzeichnis ab.

Und nun beginnt das große Dilemma: Die `FileReference`-Klasse von Macromedia Flash bietet keine Möglichkeit, eine detaillierte Response-Meldung vom Serverskript zurückzuerhalten. Der Listener der `FileReference`-Klasse achtet lediglich auf die HTTP-Statusmeldung, die vom Server zurückgeschickt wird, wenn die Datei vollständig übermittelt wurde.<sup>30</sup> Und auch die wird beim Flash-Player auf Macintosh-Systemen nur dann erkannt, wenn das Serverskript tatsächlich eine Ausgabe macht.<sup>31</sup>

Wie kann man nun also sicherstellen, dass das Serverskript die Datei ordentlich abgespeichert hat und keine Fehler auf dem Server stattgefunden haben? Denn bis zu diesem Punkt wäre für die Flash-An-

<sup>30</sup> Oder wenn ein Fehler aufgetaucht ist, wie zum Beispiel *404 – Nicht gefunden*

<sup>31</sup> Viele Dateiuploads in Flash *scheinen* mit MacOS X nicht zu funktionieren. Die Wahrheit ist: Der Upload wird tatsächlich ausgeführt, jedoch erkennt der Flash-Player anscheinend die HTTP-Status-Meldung vom Server nicht. Das hat zur Folge, dass weder die `onProgress`-Methode noch die `onComplete`-Methode des Listener-Objektes reagiert. Dies ist ziemlich ärgerlich, da so keine Rückmeldung an den Benutzer gegeben werden kann, ob der Upload in die Wege geleitet wurde oder nicht. Die Lösung des Problems ist recht simpel: Das Serverskript muss nach dem erfolgreichen Upload der Datei eine Rückmeldung geben. Zum Beispiel reicht eine echo-Ausgabe mit der Information „Upload erfolgreich“. Die Textinformation selbst kann weder der Listener noch die `FileReference`-Klasse auswerten, doch ist so der Flash-Player anscheinend in der Lage, die HTTP-Status-Meldung vom Server zu erkennen.

wendung der Upload erfolgreich abgeschlossen (da sie eine HTTP-Status-Meldung 200 erhalten hat), obwohl das Serverskript den Upload zum Beispiel wegen einer zu großen Dateigröße abgewiesen hat. Hier hilft nur ein wenig Rumgetrickse, sozusagen nicht wirklich feine Programmierung: Zunächst muss sichergestellt werden, dass das Serverskript selbst keinerlei Fehlerausgaben macht (denn diese werden ja mit einer HTTP-Status-200-Meldung versendet). Stellt nun das Serverskript fest, dass die Datei zu groß ist, wird zum Beispiel einfach ein Fehler 400 (Bad Request/Syntaxfehler) fungiert, indem in den Header der Antwort-Ausgabe der HTTP-Status auf 400 gesetzt wird. In PHP ist dies zum Beispiel einfach durch die Angabe `header('HTTP/1.1 400 Bad Request')` möglich. Wie gesagt: Programmiertechnisch kein schöner Stil, verhilft hier jedoch zu einem wesentlich benutzerfreundlicherem Verhalten der Anwendung.

Nachdem die `onComplete`-Funktion gerufen wurde, wird eine Eingabemaske angezeigt, in der der Bewerber zusätzliche Angaben zu seinem Demo machen kann. Die weiteren Daten werden nun in Verbindung mit dem vorher in Flash bereinigten Dateinamen (hier liegt also der Grund für die Bereinigung schon in Flash und nicht erst auf dem Server) an ein zweites Serverskript übermittelt, das nun zur Sicherheit noch einmal überprüft, ob die Datei tatsächlich auf dem Server vorhanden ist, und die weiteren Eintragungen wie zum Beispiel Titel, Version und Künstlername in die Datenbank übernimmt.

Ebenfalls von Adobe/Macromedia undokumentiert ist die Tatsache, dass zur Verwendung der `FileReference`-Klasse bei einer Flash-Anwendung, die über eine HTTPS-Verbindung aufgerufen wird, bei der URL-Angabe für die `browse()`-Methode die Portnummer angehängt werden muss. Ansonsten wird vom Flash-Player auf MacOS X immer standardmäßig die Portnummer 80 angehängt.

## 5.6 Usability

*„Benutzerfreundlichkeit ist im Web entscheidend. Einfach ausgedrückt: Wenn ein Kunde ein Produkt nicht findet, wird er es nicht kaufen.“*

*Das Internet ist das perfekte Werkzeug, um dem Kunden mehr Macht zu geben. Die Person an der Maus hat jegliche Entscheidungsfreiheit. Es ist höchst einfach, andere Anbieter zu finden – sämtliche Mitbewerber in der ganzen Welt sind nur einen Mausklick entfernt.“*

*(Nielsen 2001: Seite 9)*

Die Erkenntnis, dass Benutzerfreundlichkeit ein wichtiger Aspekt von Webseiten ist, hat in den letzten Jahren erfreulicherweise mehr und mehr Anhänger gewonnen. Wenngleich die Ansichten von Jakob Nielsen in mancherlei Hinsicht sehr streng sind und dem Designer von Webauftritten nicht mehr viel Gestaltungsspielraum übrig lassen, sind seine Erkenntnisse jedoch durchaus wichtig und richtig. Die Praxis verlangt in häufigen Fällen jedoch einen goldenen Mittelweg zwischen emotional fesselnden Auftritten und der möglichst standardisierten und klar gegliederten Präsentation von Information.

Wichtig bei der Entwicklung von Webanwendungen ist jedoch in jedem Fall, immer die Benutzer der Webanwendung und deren Bedürfnisse im Auge zu behalten. Nach deren Bedürfnissen muss die Gestaltung und Gliederung des Webauftrittes angepasst werden. Problem ist nur, dass Entwickler und Designer (wie auch jeder andere Mensch) jeweils dazu tendieren, zu glauben, dass andere die gleichen Ansichten haben, die man selbst hat, gerade in gestalterischen Fragen (Krug 2006: Seite 125):

*„there's a natural tendency to project these likes and dislikes onto Web users in general: to think that most Web users like the same things we like. We tend to think that most Web users are like us.“*

### 5.6.1 Flash-typische Usability-Problematiken

*„Durch Fortschritte bei Animations-, Video- und Audio-Techniken wird Multimedia zur Unterstützung von traditionellen Präsentationen im Internet immer populärer. Diese neuen Medien bieten zwar weitaus mehr Gestaltungsalternativen, verlangen andererseits aber auch mehr Disziplin.“*

*(Nielsen 2001: Seite 131)*

Macromedia Flash wird vor allem deswegen eingesetzt, weil dadurch mehr Gestaltungsspielraum besteht. Und Nielsen hat völlig recht: Diese neuen Möglichkeiten verlangen mehr Disziplin. Nicht alles,

was mit Macromedia Flash technisch möglich ist, sollte auch eingesetzt werden. Aber in Macromedia Flash umgesetzte Webanwendungen bieten hinsichtlich der Usability noch ein paar andere Schwächen, die zum Teil im Kapitel 4.1.1 schon besprochen wurden.

Dazu gehören

- die nicht ganz einfach umsetzbare Accessibility
- die Tatsache, dass gerade bei Anwendungen mit dynamischen Inhalten keine Bookmarks gesetzt werden können
- die eingeschränkte Funktionalität des Zurück-Schalters des Browsers
- bei der Gestaltung eines neuen Interfaces, dass der Benutzer dessen Verhalten erst lernen muss

Manche dieser Nachteile lassen sich jedoch auch innerhalb der Anwendung lösen. So kann mit relativ wenig Aufwand eine zusätzliche History integriert werden.

### 5.6.2 History

Gerade bei komplexeren Anwendungen wie dem praktischen Teil der vorliegenden Diplomarbeit ist ein so genannter View-Controller<sup>32</sup> nahezu unverzichtbar, da in Hauptmenü-Punkten wie *Künstler* oder *Releases* mit einer nicht voraussehbaren Anzahl von Untermenü-Punkten umgegangen werden muss. Bei einer unbekanntem Anzahl von Menüpunkten kann man in Macromedia Flash keine Anker-Navigation über die Zeitleiste verwenden, da nachträglich zur Laufzeit keine Anker auf der Zeitleiste gesetzt werden können.

Wenn solch ein View-Controller entwickelt wird, ist es nicht mehr schwierig zusätzlich eine History-Funktion miteinzubeziehen. Eine eigens in der Anwendung integrierte History-Funktion ersetzt quasi die Vor- und Zurück-Schalter des Browsers. Ein Usability-Problem von Flash-Anwendungen wäre somit behoben.

Alles, was dazu benötigt wird, ist in erster Linie ein mehrdimensionales Array. Es speichert die aktuelle Kategorie als Breadcrumb-Pfad<sup>33</sup> und in einer festgelegten Maximalanzahl die Breadcrumb-Pfade der

<sup>32</sup> Ein View-Controller managet den Aufbau einer Internetseite oder einer Anwendung. Ausgehend von einer möglichst eindeutigen Angabe, welcher Informationsbereich angezeigt werden soll (zum Beispiel durch eine URL), baut der View-Controller entsprechend die gewünschte Seite auf.

<sup>33</sup> Ein Breadcrumb-Pfad oder auch Brotkrumen-Pfad ist in Anlehnung an das Märchen von Hänsel und Gretel ein Orientierungspfad, der Schritt für Schritt ausgehend von der absoluten Startposition festhält, wie man zu dem momentanen Aufenthaltspunkt gelangt ist. Er ist somit quasi mit einer schriftlichen Wegbeschreibung vergleichbar.

vorher aufgerufenen Seiten als jeweils einzelnes Array. Zusätzlich benötigt man einen Zeiger, der die momentane History-Position angibt.

Der View-Controller baut die Seite entsprechend des aktuellen Breadcrumb-Pfades (also dem Array entsprechend, auf das der Zeiger weist) auf. Er muss um eine Funktion erweitert werden, die den Aufbau der History managet. Diese Funktion muss bei jedem neuen Seitenaufruf überprüfen, ob es sich eventuell um einen History-Zugriff handelt oder um einen normalen Seitenaufruf.

Handelt es sich um einen History-Zugriff wird die Seite entsprechend dem Breadcrumb-Array an der Position des History-Zeigers aufgebaut.

Ist es ein normaler neuer Seitenaufruf, muss das History-Array direkt nach der Position des History-Zeigers um ein Array mit dem neuen Breadcrumb-Pfad ergänzt werden. Alle nachfolgenden Breadcrumb-Arrays werden, sofern vorhanden, gelöscht. Ist zudem die Maximalanzahl der History-Einträge erreicht, müssen am Anfang des History-Arrays entsprechend viele Einträge gelöscht werden.

Ein gewisses Usability-Problem bleibt jedoch bestehen: Der Benutzer muss erst lernen, dass er nur über die History-Schaltflächen der Flash-Anwendung hin- und hernavigieren kann, und nicht über die Browser-Schaltflächen.

### 5.6.3 Strukturierung zur besseren Orientierung

*„Die Daten müssen in irgendeiner Form strukturiert sein. Diese Struktur muss für den Anwender nachvollziehbar, d.h. intuitiv verstehbar sein.“*

*(Schweibenz/Thissen 2003: Seite 25)*

Wichtig war bei der Konzeption des praktischen Teils der vorliegenden Diplomarbeit, dass bei der Fülle und Detailliertheit der Daten die Übersichtlichkeit gewahrt bleibt.

*„Die wichtigste Frage bei der Navigation lautet wahrscheinlich: ‚Wo bin ich?‘. Denn die Anwender werden die Struktur einer Website niemals verstehen, wenn sie nicht wissen, wo sie sich befinden. Wenn sie nicht wissen, wo Sie sind, können Sie auch die Bedeutung des gerade angeklickten Links nicht herausfinden.“*

*(Nielsen 2001: Seite 188)*

Der Benutzer soll jederzeit die Orientierung behalten, wo er sich gerade befindet, und wie er zur nächsten gewünschten Information gelangt. Die Gliederung durch das Layout-Raster bietet dazu gute Möglichkeiten. Der Weg zur angezeigten Information ist an jeder Position nachvollziehbar.

Es wurde dabei bewusst relativ viel Raum für diese Visualisierung in Kauf genommen.<sup>34</sup> Nur beim Aufruf wirklich detaillierter Information (zum Beispiel Informationen zu jedem einzelnen Track) wird ein Teil des Gesamtrahmens ausgeblendet, da der Bezug dieser Detailinformation direkt beim Track und dem zugehörigen Release zu suchen ist.

Steve Krug (Krug 2006: Seite 76) hält fest, dass eine Visualisierung des momentanen Standortes nach der Vorgehensweise „*You are here*“ besser ist als zum Beispiel die reine Wegbeschreibung zum Beispiel in Form von Breadcrumbs:

*„Unlike ‚You are here‘ indicators, which show you where you are in the context of the site’s hierarchy, Breadcrumbs only show you the path from the Home page to where you are. [...] kind of like the difference between looking at a road map and looking at a set of turn-by-turn directions. The directions can be very useful, but you can learn more from a map.“*

### 5.6.4 Hilfestellungen

Um den Benutzer das Bedienen der Webanwendung möglichst leicht zu machen, wurden zusätzliche Hilfestellungen (zumeist in Form von Sprechblasen) eingefügt.

So wird beim Aktivieren des Login-Formulars darauf hingewiesen, wo sich der Benutzer für einen Login-Zugang registrieren kann, sofern er noch keinen besitzt.

Bei einigen Formularen wie zum Beispiel beim Registrierungsformular wird bei Falschangabe in einem Formularfeld nicht nur mit einer entsprechenden Nachricht darauf hingewiesen; das Formular scrollt zusätzlich automatisch an die entsprechende Position.

Bei wichtigen Eingaben und Entscheidungen (wenn zum Beispiel ein Eintrag oder ein Bild aus der Datenbank gelöscht werden soll) wird der Benutzer mittels einer Warnhinweis-Box auf die Folgen dieses Vorgangs hingewiesen und nochmals gefragt, ob er diese Aktion auch wirklich durchführen will.

<sup>34</sup> Was zugegebenermaßen im Widerspruch zu Jakob Nielsens Forderungen steht, dass der Inhalt König sei und entsprechend die Navigationselemente so wenig Raum wie möglich einnehmen sollten. Doch nur so ergibt sich die Möglichkeit, die Informationen von allen Künstlern oder Releases sehr schnell durchstöbern zu können. So kann der Benutzer zum Beispiel in der Release-Ansicht jederzeit sehr einfach zum nächsten aber auch direkt zum übernächsten Release gehen und sich die entsprechende Tracklist anhören.

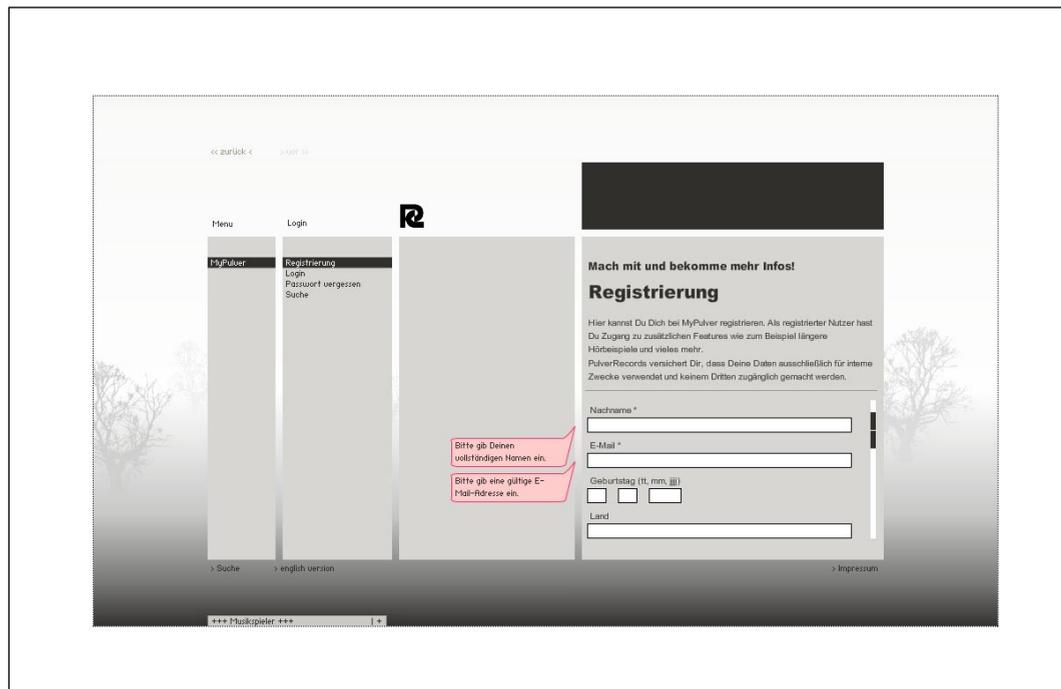


Abbildung 5.8:  
Hilfestellungen zu den  
Formulareingaben bei der  
Registrierung

### 5.6.5 Suche und Suchergebnisse

Eine Suchfunktion ist bei umfangreichen Seiten, die zudem noch einen gewissen Datenbank-Charakter besitzen, nahezu unentbehrlich.

Um dem Benutzer bei den ermittelten Suchergebnissen schon möglichst viel Information und Orientierung zu bieten, werden die Suchergebnisse zum einen vorsortiert nach den Rubriken *Künstlerprofile*, *Releases/Alben* und *Einzelne Titel/Tracks*. Zu den Releases werden dabei schon grundlegende Informationen wie Veröffentlichungsdatum und Tonträger-Format mitangegeben. Zum anderen werden die Ergebnisse in Bezug zueinander gesetzt, so ist bei jedem einzelnen Track auch immer gleich zu sehen, zu welchem Release er gehört.

Um den Komfort der Suche weiter zu steigern, werden die Suchergebnisse temporär mittels eines so genannten SharedObjects (Flash-Cookie) auf dem Benutzerrechner gespeichert. So kann sich der Benutzer jederzeit die bisher bei diesem Besuch ermittelten Suchergebnisse nochmals ohne längere Wartezeit anzeigen lassen und verschiedene Suchergebnisse ohne großen Aufwand miteinander vergleichen.

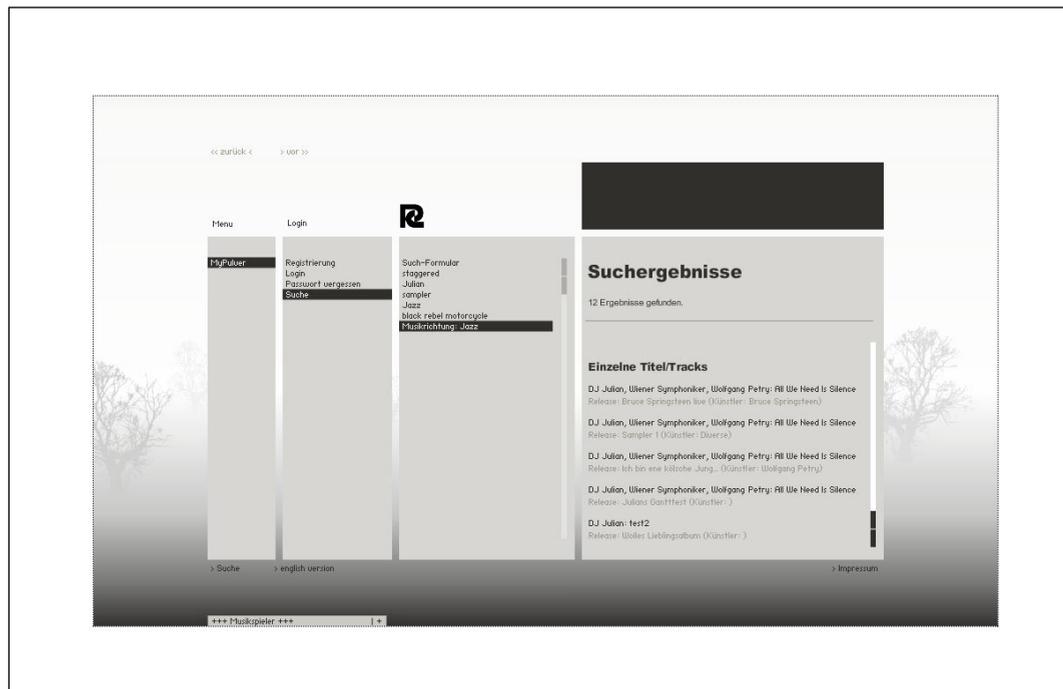


Abbildung 5.9:  
Auflistung der Suchergebnisse

## 5.7 Die XML-Brücke

Es gibt drei wesentliche Möglichkeiten, eine Flash-Anwendung mit Serverskripten kommunizieren zu lassen:

- mittels der allgemeinen Funktion `loadVariables()`
- mittels des `LoadVars`-Objektes
- mittels des `XML`-Objektes

Um es gleich vorweg zu nehmen: Die Kommunikation mittels des `XML`-Objektes ist die flexibelste und wahrscheinlich auch komfortabelste Art.

Die Funktion `loadVariables()` ist schon sehr alt und auf Grund der Tatsache, dass mit ihr nur bedingt zweiseitiger Informationsaustausch möglich ist<sup>35</sup>, empfiehlt sich deren Verwendung nicht für komplexe Anwendungen. Um zum Beispiel lediglich die aktuelle Besucherzahl der Website von einem Serverskript abzurufen, ist sie jedoch nach wie vor gut geeignet, da wenig Aufwand für die Übermittlung der Information getrieben werden muss, und die Funktion schneller ist, als zum Beispiel die Übermittlung in `XML`-Elementen über die `load()`-Funktion des `XML`-Objektes.<sup>36</sup>

Soll in beiden Richtungen kommuniziert werden, kommen also nur noch die Funktionen des `LoadVars`-Objektes und des `XML`-Objektes in Frage.

Da der Einsatz von `XML` im Allgemeinen jedoch flexibler, `XML` ein Standard ist und auch die `XML`-Funktionalitäten in Flash und PHP sehr gut entwickelt sind, empfiehlt es sich, auf das `XML`-Objekt in Macromedia Flash zurückzugreifen.

Im vorliegenden Projekt wurde zudem noch aus einem weiteren Punkt die Übermittlung im `XML`-Format gewählt: Durch die Verwendung von `XML` ist sichergestellt, dass auch andere (zukünftige)

<sup>35</sup> Es kann in erster Linie nur Information geladen werden. Das Senden von Information ist nur dadurch möglich, die Information als angehängte GET-Parameter im URL-String zu transportieren. Komplexe und umfangreiche Informationen lassen sich so nicht zum Server übermitteln.

<sup>36</sup> Vgl. Swann/Caines 2001: Seite 189. Die Übermittlung mittels `loadVariables` ist nahezu doppelt so schnell wie die Übermittlung der Information in `XML-Elementen` über das `XML`-Objekt. Die Übermittlung der Information in `XML-Attributen` mittels des `XML`-Objektes ist jedoch noch schneller (ungefähr doppelt so schnell wie die Übertragung mittels `loadVariables`). Allerdings ist der Aufwand, der getrieben werden muss, um die Information für den Transport vorzubereiten (Erstellen eines `XML`-Dokumentes) im Vergleich zu groß. Auch steigt die Gesamt-Datenmenge der übermittelten Daten durch die Kapselung der reinen Information in ein `XML`-Dokument an (Für das Beispiel der schlichten Übertragung eines Zählerstandes wäre so der Geschwindigkeitsvorteil schnell wieder zunichte gemacht, da bei der `loadVariables`-Methode eine reine String-Ausgabe nach dem Muster „meineVariable=meineInformation“ ausreicht).

Technologien ohne großen Aufwand für die Ausgabe verwendet werden können. So kann nachträglich zum Beispiel eine parallele Präsentation in XHTML angegliedert werden.

Angemerkt sei noch, dass die Kommunikations-Module innerhalb der Flash-Anwendung vollkommen von den Modulen, die das Layout betreffen, getrennt sind. So ist es leicht möglich, auch ein völlig anderes Layout umzusetzen.

Zur genauen Vorgehensweise beim Datenaustausch mittels XML bei Flash-Anwendungen sei hier auf die zahlreiche Fachliteratur verwiesen.<sup>37</sup>

<sup>37</sup> Zum Beispiel Swann/Caines 2001  
oder Sas Jacobs (2006): Foundation XML for Flash, Berkeley: Friends of Ed – Apress

## 5.8 Mehrsprachigkeit

Da viele Recordlabels nicht nur regional oder national von Bedeutung sind, sondern auch über die Landesgrenzen hinweg Kunden, Fans und Freunde haben, ist es wichtig, dass die Möglichkeit besteht, die Informationen auch in mehreren Sprachen anbieten zu können.

Der praktische Teil zur vorliegenden Diplomarbeit ist deswegen für den Einsatz von mehreren Sprachen entwickelt worden. Die Anwendung wurde dafür mit mehreren Sprachmodulen ausgestattet. Um sicherzustellen, dass die Webanwendung möglichst schnell geladen wird, wird jedoch empfohlen, möglichst nur so wenig Sprachmodule wie nötig zu installieren. Da so die nötige Anzahl eingebetteter Schriftzeichen sowie die Zeichenanzahl für die verschiedenen Übersetzungen der Kernelemente und somit insgesamt die Dateigröße verringert werden.

Zu den Besonderheiten der Mehr-Sprachen-Funktionalität gehört, dass der Benutzer an jeder Stelle der Anwendung die Sprache wechseln kann ohne auf die Startseite zurückgeführt zu werden. Dazu wird lediglich die neu ausgewählte Sprache in einer entsprechenden Variable hinterlegt und der View-Controller angewiesen, den Inhalt neu anzuzeigen.

Sollte eine Information in der Datenbank nicht in der gewünschten Sprache vorliegen, wird der Benutzer darauf hingewiesen. Er erhält jedoch die Nachricht in der Standard-Sprache der Anwendung (die vom Recordlabel in den Einstellungen festgelegt worden ist).

**Sprachwechsel an jeder  
Stelle möglich**

# Kapitel 6

## **Präsentation der Anwendung**

---

Startseite

Musikspieler

Informations-Bereich

*Im folgenden möchte ich nun noch einige Teile der entwickelten Anwendung präsentieren, die in den vorangegangenen Kapiteln noch nicht im Detail besprochen wurden. Dies sind vor allem der Musikspieler und der Aufbau der Informationsseiten zu den Releases und den Künstlern.*

## 6.1 Startseite

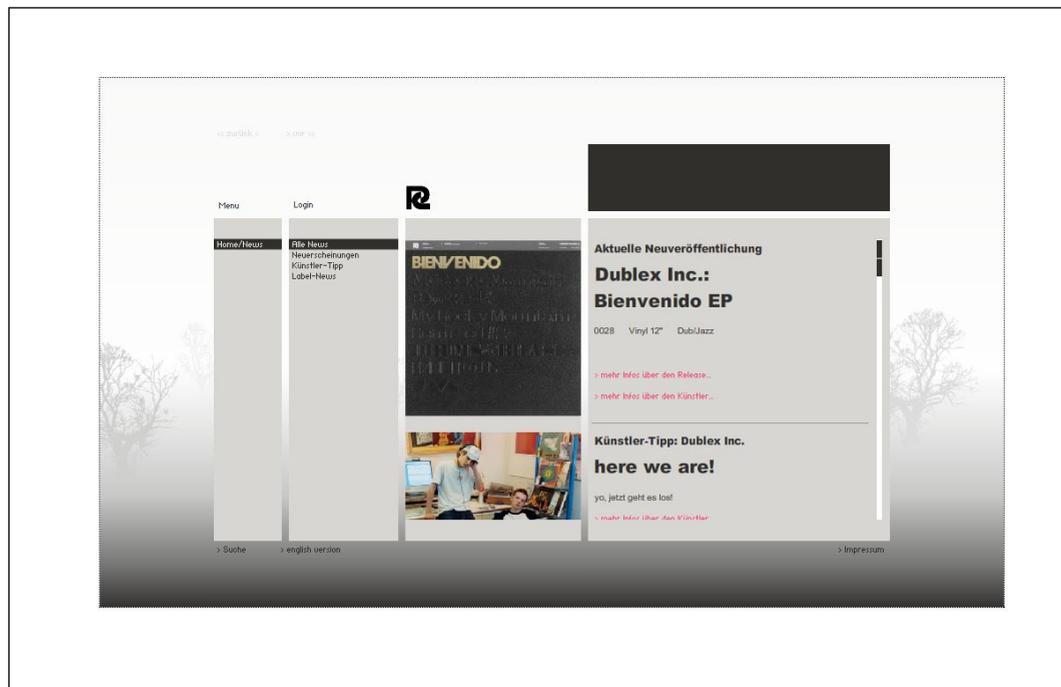


Abbildung 6.1:  
Startseite

Auf der Startseite wird der Benutzer auf die Neuigkeiten des Recordlabels aufmerksam gemacht. Zudem wird hier der neueste Release kurz vorgestellt und ein Künstler-Tipp gezeigt. Auf Wunsch kann der Benutzer über Verlinkungen zu den detaillierten Release- oder Künstlerinfos gelangen.

## 6.2 Musikspieler

An Anlehnung an die bei DJs gebräuchlichen Plattenspieler wurde im MP3-Player-Modul eine Animation eines Plattenspielers miteingebaut.

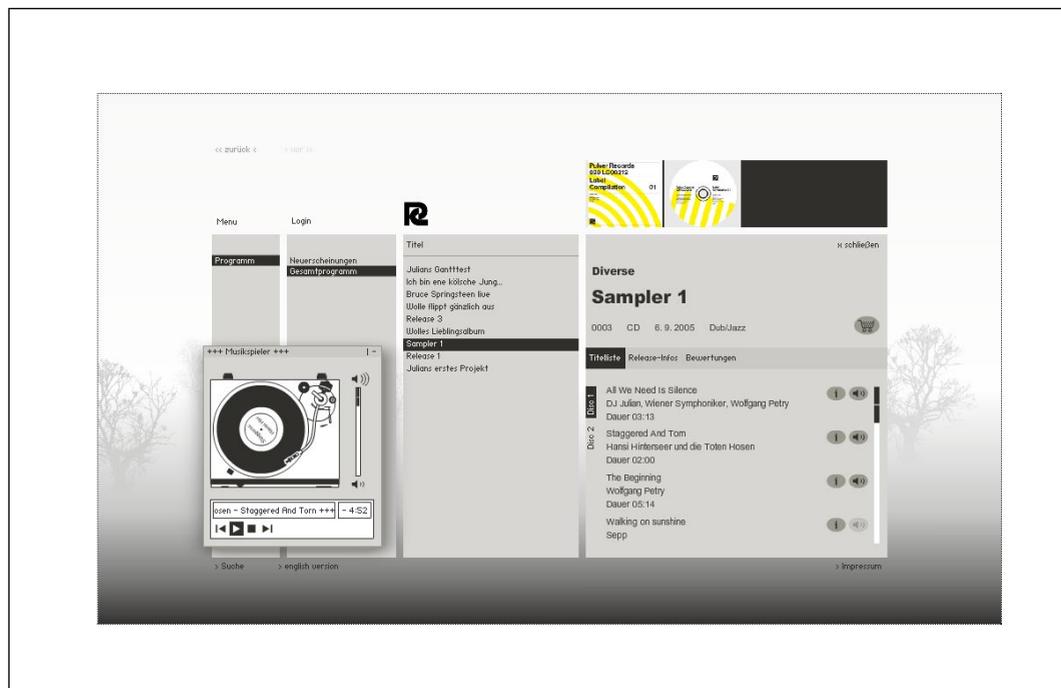


Abbildung 6.2:  
Musikabspiel-Modul mit  
Plattenspieler-Animation

Die Plattenspieler-Animation verhält sich während der Wiedergabe wie ein realer Plattenspieler. Der Tonarm wird zunächst auf die Platte aufgesetzt und bewegt sich dann entsprechend dem Abspiel-Fortschritt auf der Platte vor. Am Ende des Tracks fährt der Tonarm auf die Ausgangsposition zurück.

Je nach Zugriffsrecht des Benutzers und Grundeinstellung vom Recordlabel kann der Benutzer entweder nur Ausschnitte (Snippets) eines Titels oder den vollständigen Titel (Full Audio) anhören.

Das Player-Modul liegt auf einer separaten Ebene über dem eigentlichen Seitenaufbau und kann frei über die gesamte Fläche verschoben werden. Zudem besteht die Möglichkeit, das Player-Modul zu minimieren.

## 6.3 Informations-Bereich

Die Darstellung der Informationen zu den Künstlern und den Releases folgt dem gleichen Layout-Raster: Verfügbares Bildmaterial wird immer in der oberen rechten Box angezeigt. Wenn mehr Bilder vorhanden sind, als in der Box Platz haben, werden die Bilder in Bildfolgen aufgeteilt, die der Benutzer durchblättern kann. Jedes Bild kann außerdem in einer größeren Ansicht angezeigt werden. Dazu schiebt sich ein Reiter über den Inhaltsbereich. Innerhalb dieses Reiters kann der Benutzer Bild für Bild vorwärts oder rückwärts das Bildmaterial durchblättern.

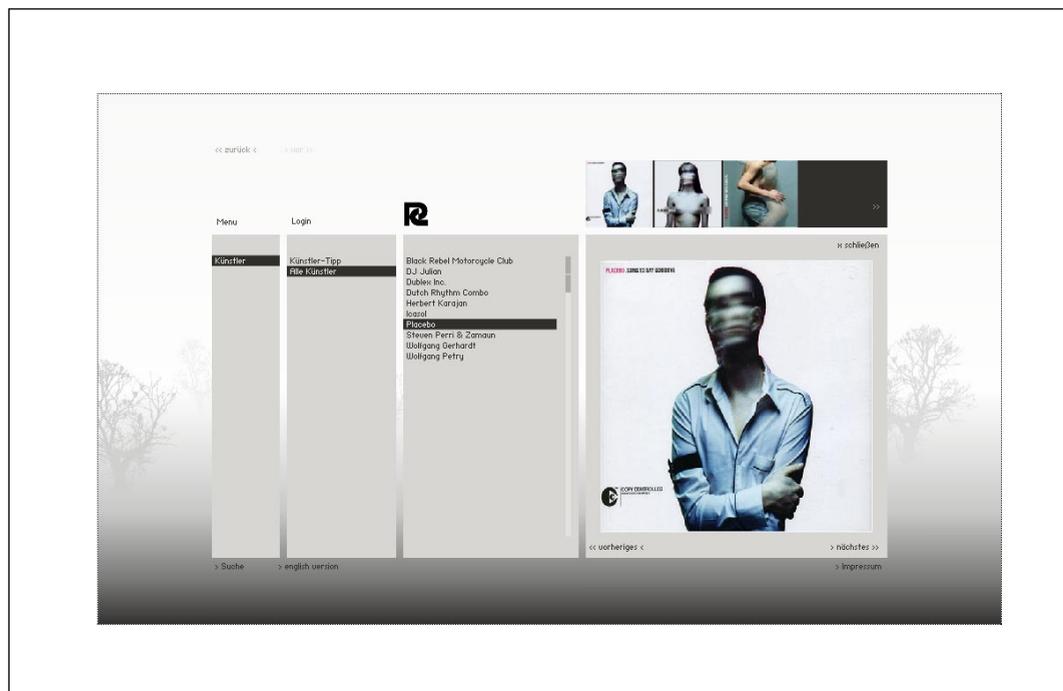


Abbildung 6.3:  
Detailansicht eines  
Künstlerfotos

Sofern angegeben werden zusätzliche Informationen zu dem angezeigten Bild geboten, wie zum Beispiel Angaben zum Fotografen oder bei Cover-Abbildungen Angaben zum Artwork und Konzept.

### 6.3.1 Informationen zu einem Künstler

Zu jedem Künstler können individuelle News, eine Biografie und Termin-Angaben zu Auftritten abgerufen werden. Diese Menüpunkte kann der betreffende Künstler selbst pflegen. Ist er eingeloggt und schaut auf sein eigenes Künstler-Profil, kann er mit einem direkten Link zum Edit-Bereich gelangen, um die betreffende Information zu ändern oder zu ergänzen.

Zudem gibt es einen Menüpunkt Diskografie, unter dem alle vom Künstler produzierten Releases, alle anderen Releases, an denen der Künstler beteiligt war, und alle Remixes, die der Künstler gemacht hat, aufgelistet sind. Die entsprechenden Ergebnisse sind als Verlinkung angelegt, die direkt zum entsprechenden Release führt.

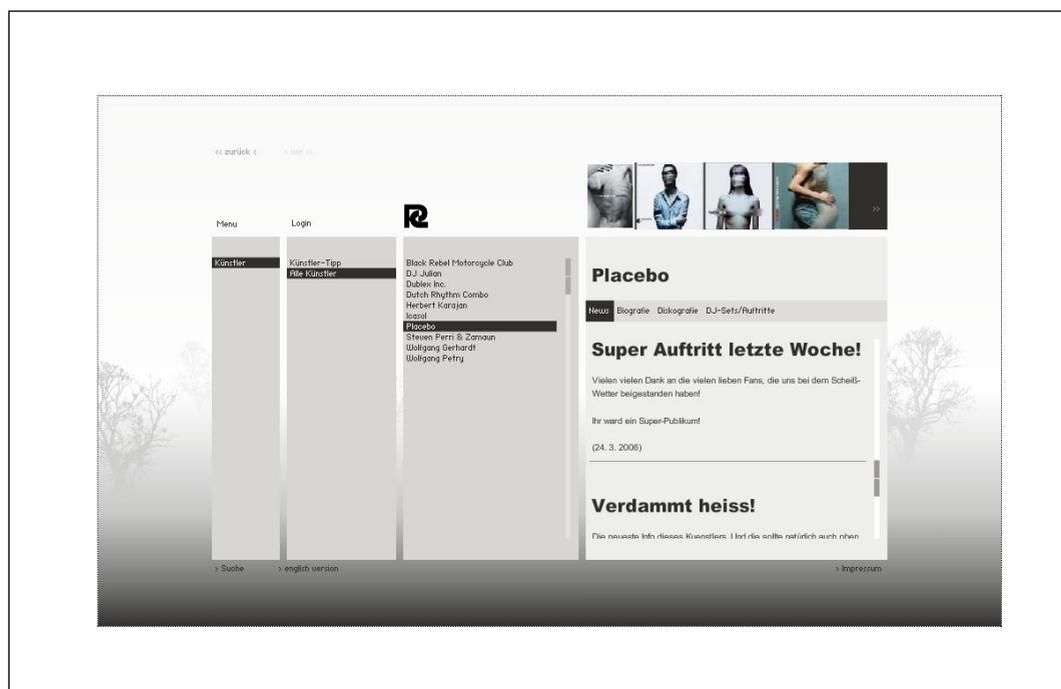


Abbildung 6.4:  
Das Künstler-Profil

Für Benutzer aus der Kontaktgruppe Presse gibt es zusätzlich einen Menüpunkt Downloads, unter dem Pressematerialien (Bilder, Texte und ein ZIP-Archiv mit allen gesammelten Informationen) über den Künstler zum Download zur Verfügung stehen.

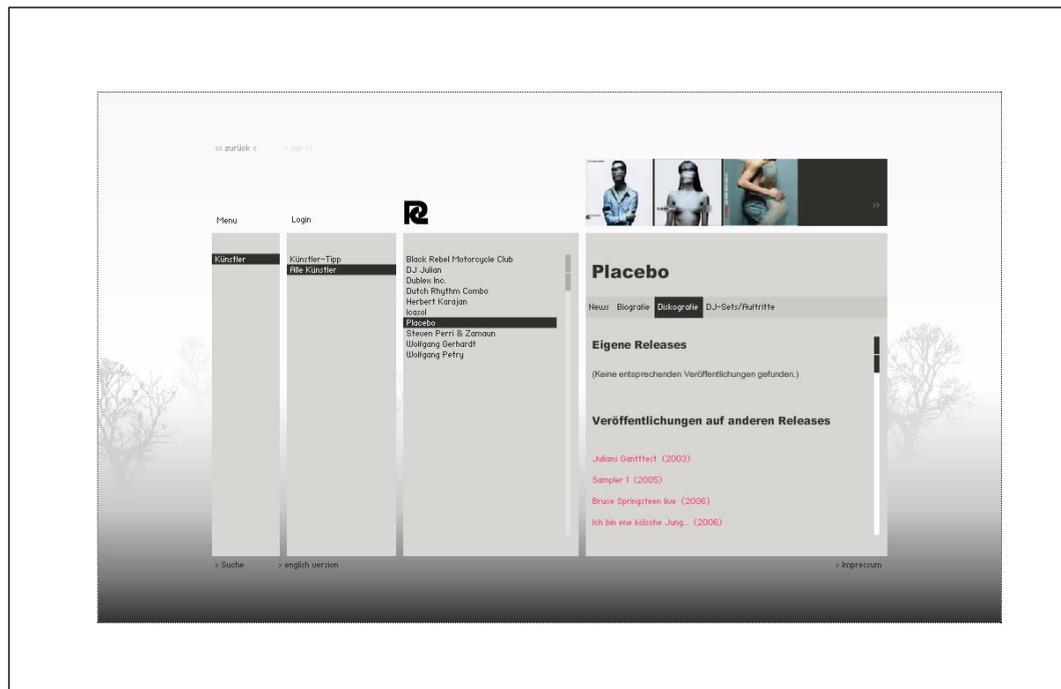


Abbildung 6.5:  
Diskografie eines Künstlers

### 6.3.2 Informationen zu einem Release

Zu einem Release werden standardmäßig die Titelliste, allgemeine und weiterführende Informationen sowie Kritiken angezeigt.

Die Titelliste ist entsprechend der vorhandenen Tonträger aufgeteilt. Bei Doppel-CDs gibt es somit also zwei Titellisten. Zu jedem Titel gibt es die Möglichkeit, weitere detaillierte Informationen anzeigen zu lassen. Diese Informationen werden in einem eigenen Reiter angezeigt, der sich links vom Info-Bereich aufzieht (vgl. Abb. 6.7), und beinhalten zum Beispiel Angaben über den Produzenten des Titels, die Angabe der Beats per Minute (sofern angegeben), aber auch Feedbacks speziell zu diesem Titel von anderen DJs und Pressestimmen.

Sofern eine entsprechende Musikdatei auf dem Server vorliegt und der Benutzer das entsprechende Zugriffsrecht besitzt, gibt es die Möglichkeit, den Titel über das Abspielmodul anzuhören. Je nach Zugriffsrecht und Grundeinstellung ist dies nur ein so genanntes Snippet, also nur ein Ausschnitt, oder der vollständige Titel (Full Audio).

Zu jedem Release gibt es zudem einen so genannten Shop-Link, der entweder auf einen später integrierten Shop verweisen könnte oder aber per so genanntem Deep-Link auf eine andere Bestellmöglichkeit (zum Beispiel zu Vertriebspartnern) verweist.

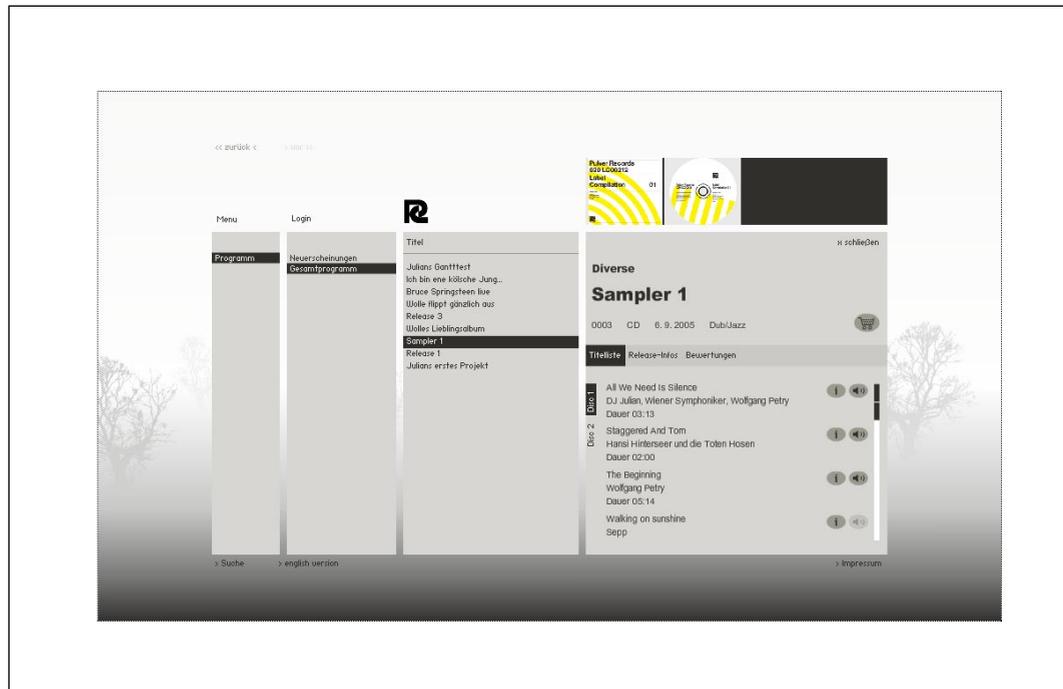


Abbildung 6.6:  
Informationen zu einem  
Release

Unter dem Menüpunkt Bewertungen kann sich der Benutzer Feedbacks von DJs und Pressestimmen zu dem gesamten Release ansehen.

Für eingeloggte Benutzer der Kontaktgruppe Vertrieb besteht zusätzlich die Möglichkeit spezielle weitergehende Informationen zu dem Release zu bekommen. Dazu gehören zum Beispiel Angaben über den Projektstatus (Vertriebs-Kontakte haben die Möglichkeit auch schon Projekte aufgelistet zu bekommen, die noch in Produktion sind) und das Gewicht des Produktes.

Wie auch bei den Künstler-Infos können Benutzer der Kontaktgruppe Presse unter einem Extra-Menüpunkt Downloads Materialien direkt zu diesem Release herunterladen.

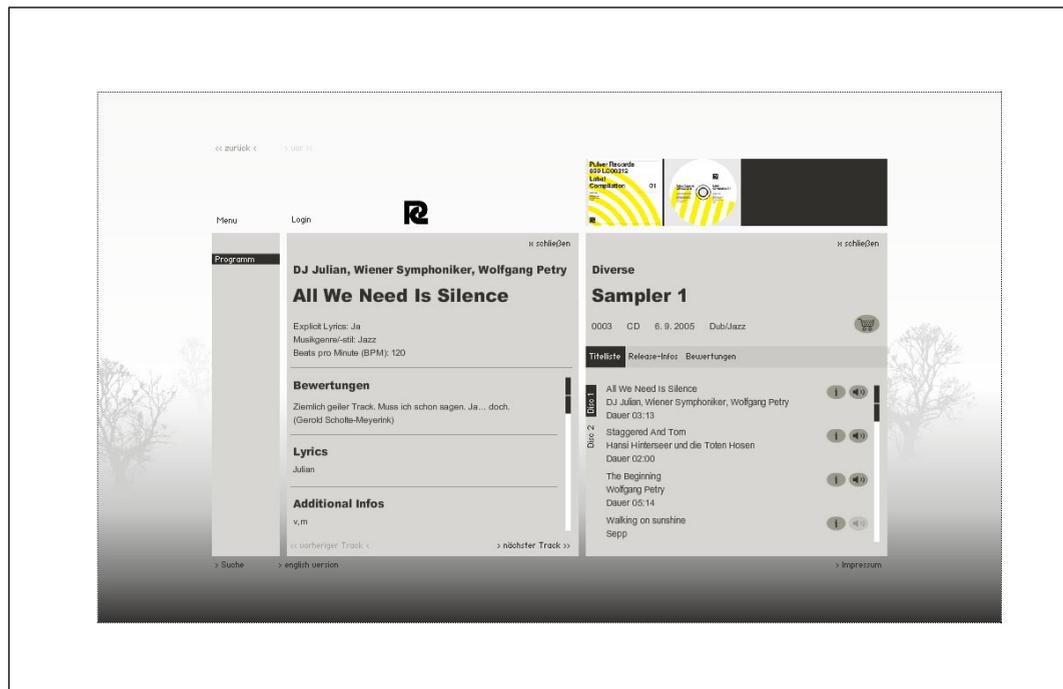


Abbildung 6.7:  
Detailansicht für einen ausge-  
wählten Titel eines Releases

# Kapitel 7

## **Zusammenfassung und Ausblick**

---

## 7.1 Zusammenfassung

Die entwickelte Webanwendung ist in ihrer Art völlig neu. Sie bietet für so genannte Independent Recordlabels eine sehr viel komfortablere Möglichkeit, sich, seine Künstler und seine Produkte zu präsentieren.

Entwickelt wurde ein Interface, das die Daten der Recordlabel-Software für eine Frontend-Anwendung zugänglich macht.

Zugleich wurde eine ebensolche Frontend-Anwendung mittels Macromedia Flash erstellt. Diese ist mit wenig Aufwand mittels einer Layout-Klasse in einem bestimmten Rahmen individualisierbar. Durch den modularen Aufbau ist es leicht möglich, auch ein völlig anderes Layout zu verwenden.

Die Darstellung der Information wird zur Laufzeit mittels eines View-Controllers gesteuert. Dieser View-Controller beinhaltet zur Steigerung der Usability eine History-Funktion, zudem ist die Darstellung der Information in mehreren Sprachen möglich. Die Sprache lässt sich an jeder beliebigen Stelle der Anwendung ändern.

Über einen zugangsgeschützten Bereich kann der Benutzer der Webanwendung individuelle Inhalte, Playlists und Previews erhalten. Die Previews sind kombiniert mit einem Feedback-System, das dem Recordlabel eine enorme Zeit- und Arbeitersparnis ermöglicht. Der Einsatz von Previews und persönlichen Playlists macht eine direkte Kommunikation mit dem Kunden über den Webauftritt möglich.

Es wurde ein sicheres Prelistening-Verfahren umgesetzt, das verhindert, dass die Musikdateien nach dem Anhörvorgang als Kopie auf dem Benutzerrechner gespeichert sind, und das ohne den Einsatz teurer Streaming-Verfahren auskommt.

## 7.2 Ausblick

Zum Abschluss möchte ich noch einen kurzen Ausblick darauf geben, was zum einen in nächster Zeit für die entwickelte Webanwendung angedacht ist, und zum anderen, wie sich der Einsatz von komplexen Flash-Anwendungen im Internet entwickeln wird.

### 7.2.1 Erweiterungen der entwickelten Anwendung

In einer weiteren Version der Recordlabel-Software soll es möglich sein, dass auch an der Produktion der Tonträger beteiligte Unternehmen über das Frontend die nötigen Daten eines Projektes (zum Beispiel Druckdaten für die Booklets und Cover) abrufen können. Die an dem Projekt beteiligten Künstler wiederum können dann über dieser Erweiterung alle erforderlichen Angaben für die Produktion (zum Beispiel die Texte für das Booklet) über das Frontend eingeben.

Für die Zukunft sind weitere Ausbaustufen denkbar. So könnte in Zukunft die Musik direkt in digitaler Form an die Endkunden über eine eingebundene Shop-Funktionalität verkauft werden. Bisher wird an den entsprechenden Stellen zu externen Vertriebspartnern verlinkt.

Die entwickelte Anwendung wurde bewusst flexibel und modular aufgebaut, um zum einen Erweiterungen ohne großen Aufwand vornehmen zu können, aber vor allem auch aus dem Grund, die Anwendung auch an andere interessierte Recordlabels verkaufen und dabei leicht individuelle Anpassungen durchführen zu können.

### 7.2.2 Rich-Internet-Anwendungen mit Flash in der Zukunft

Flash-Anwendungen im Internet werden in Zukunft dank der immer größeren Möglichkeiten (vor allem hinsichtlich der Übertragungs-Bandbreite) vermehrt im Informations- und Infotainment-Bereich Einzug halten. So genannte Hybrid-Seiten<sup>37</sup> wie der aktuelle Webauftritt von Panasonic<sup>38</sup> und

<sup>37</sup> Als Hybrid-Seiten werden Internet-Auftritte bezeichnet, die sowohl Flash- als auch HTML-Elemente besitzen.

<sup>38</sup> [www.panasonic.de](http://www.panasonic.de) (Datum des Zugriffs: 14. April 2006)

viele Auftritte von Automarken zeigen, was für Möglichkeiten der Einsatz von Flash eröffnet; Möglichkeiten, Information mit Emotion und Image zu verbinden – und das interaktiv. Bei der zunehmenden Verschmelzung von Internet und Fernsehen werden Flash-Anwendungen zudem eine große Rolle spielen.

Flash-Sites werden in Zukunft immer mehr in Form von eigenständigen Anwendungen programmiert werden, wie auch Rob Ford festhält (Ford 2006): „Meiner Meinung nach werden wir jetzt verstärkt anwendungsbetonte Websites erleben, wie etwa die neue Website von FWA ([www.theFWA.com](http://www.theFWA.com) [Anm. d. Autors]), die so läuft und sich so anfühlt wie die Software auf dem Desktop.“

# Anhang

---

Inhalt der CD-ROM

# Inhalt der CD-ROM

Auf der beigefügten CD-ROM liegt diese theoretische Ausarbeitung in elektronischer Form als PDF vor.

Alle angegebenen Internet-Quellen sind zudem als HTML-Archiv auf der CD-ROM abgespeichert.

Desweiteren enthält die beigefügte CD-ROM Klassen und Funktionen, die für viele Projekte nützlich sein können. Bis auf die Encoding-Funktionen von Paul Johnston wurden sie alle eigenständig für das vorliegende Projekt entwickelt.

## Encoding-Klasse

Die Klasse *Encoding.as* enthält Funktionen, um einen String mittels MD5 oder SHA1 zu verschlüsseln. Die Programmierung der Funktionen stammt von Paul Johnston und war ursprünglich als JavaScript angelegt. Diese Funktionen sind auf der Internetseite <http://pajhome.org.uk/crypt/md5/> unter der BSD-Lizenz veröffentlicht.

Um die Funktionen auch bei zukünftigen Projekten bequem einsetzen zu können, habe ich die Funktionen in eine ActionScript-2-Klasse eingebettet.

## Layout-Klasse

Die Klasse *LayoutBox.as* ist der Grundstock für den dynamischen Aufbau der entwickelten Flash-Anwendung.

## Klasse zur Kontrolle von Eingabedaten

Die Klasse *Zeichen.as* bietet mehrere Funktionen zur Überprüfung von Daten, die ein Benutzer in ein Eingabe-Textfeld machen kann, auf Gültigkeit. So lässt sich zum Beispiel über die Funktion `isEmailString()` testen, ob die eingegebene Zeichenfolge eine formell korrekt geschriebene E-Mail-Adresse ist.

## **Bildupload in Flash mit PHP als Serverskript**

Ein Beispiel für die Umsetzung eines Dateiuploads mit Response-Möglichkeit über den HTTP-Status.

# Glossar

---

## **Accessibility**

Barrierefreiheit. Barrierefreie Webauftritte und Anwendungen können auch von Menschen mit Behinderungen ohne große Hürden bedient werden.

## **Backend**

Im Gegensatz zum →Frontend der Teil der Anwendung, der auf dem Server läuft. Häufig wird jedoch auch der gesamte Teil einer Webanwendung Backend genannt, der für den normalen Besucher des Webauftrittes nicht sichtbar/zugänglich ist und zum Pflegen der Daten dient, die im allgemein zugänglichen Teil zu sehen sind.

## **Bookmark**

→Lesezeichen

## **FileMaker o. FileMaker Pro**

Datenbank-Programm für Windows- und Macintosh-Systeme.

## **Flash-Anwendung**

Eine Anwendung, die mittels der Software Flash erstellt worden ist; vor allem eingesetzt für Animationen und interaktiven Webanwendungen; für die Verwendung im Internet zumeist als SWF-Datei (Small Web Format). Die Software Flash wird von Adobe/Macromedia vertrieben. Zur Wiedergabe der SWF-Dateien ist ein eigener Player nötig: der Flash-Player. Er ist als Stand-alone-Player und als Browser-→Plugin verfügbar.

## **Hosting**

Zur Verfügungstellung von so genanntem Web-Space, also Speicherplatz im World Wide Web.

## **HTTP**

HyperText Transfer Protocol. Protokoll, das festlegt, wie die Webseiten und deren Medien zwischen Webserver und Browser des Benutzerrechners ausgetauscht werden sollen.

## **HTTPS**

HyperText Transfer Protocol Secure. Ermöglicht eine gesicherte Verbindung zwischen Server und Browser des Benutzerrechners. Die übermittelten Daten werden dabei mit einem vorher ausgemakelten Schlüssel verschlüsselt.

## **Independent Recordlabel**

Unabhängiges Platten-Label. Quasi ein Verlag für Musik-Tonträger, der bestimmte Musik-Künstler unter Vertrag hat und für deren Musik die Vermarktung übernimmt. Independent Recordlabels sind unabhängig von großen Konzernen und Institutionen (daher der Name).

## **Lesezeichen**

Favorit oder →Bookmark. Vom Benutzer im Browser abgespeicherte URL zum schnellen Zugriff auf die Seite.

## **Plugin**

Erweiterungen für einen Browser um bestimmte Inhalte wie zum Beispiel Video, Animationen oder Audio wiedergeben zu können.

## **Prelistening-Funktion**

Die Möglichkeit, bestimmte Musikstücke über das Internet anhören bzw. probenhören zu können.

## **Preview**

Eine Preview ist die Möglichkeit, ein (oder mehrere) →Track(s) vor ihrem Veröffentlichungstermin (Release-Date) anhören zu können. Vor allem wichtig für die Presse und DJs.

## **Release**

Veröffentlichung. Ein Release ist ein veröffentlichtes Projekt. In der Regel ein Tonträger mit mehreren Musikstücken (→Tracks), zum Beispiel Album oder EP, seit einiger Zeit jedoch auch eine bestimmte feste Zusammenstellung von digitalen Musikdateien.

## **Session-ID**

Individuell vom Web-Server vergebene ID zur Identifizierung von Benutzern. Wird benötigt, da HTTP ein verbindungsloses Protokoll ist.

## **Track**

Musiktitel. Ein Track ist die kleinste Einheit in der entwickelten Recordlabel-Software. Mehrere Tracks können in einem Projekt zusammengefasst werden, aus dem dann ein →Release hervorgehen kann.

# Literaturverzeichnis

---

# Literaturverzeichnis

**Adobe/Macromedia** (2006):

Macromedia Flash Player Statistics.

[http://www.macromedia.com/software/player\\_census/flashplayer/](http://www.macromedia.com/software/player_census/flashplayer/)

(Datum des Zugriffs: 13. April 2006).

**Böhringer, Joachim/Bühler, Peter/Schlaich, Patrick/Ziegler, Hanns-Jürgen** (2000):

Kompendium der Mediengestaltung für Digital- und Printmedien.

Berlin; Heidelberg; New York: Springer-Verlag (X.media.press)

**Ford, Rob** (2006):

in Wiedemann, Ed. Julius:

Web Design: Flash Sites.

Köln; London; Los Angeles; Madrid; Paris; Tokyo: Taschen (Icons)

Seiten 12 bis 15.

**Fuhrberg, Kai/Häger, Dirk/Wolf, Stefan** (2001):

Internet Sicherheit – Browser, Firewalls und Verschlüsselung.

München; Wien: Carl Hanser Verlag

**Gumm, Heinz-Peter/Sommer, Manfred** (2002):

Einführung in die Informatik.

München; Wien: Oldenbourg

**Hall, Branden** (2002):

Locking down Macromedia Flash.

<http://www.macromedia.com/devnet/flash/extreme/extreme003.html>

(Datum des Zugriffs: 13. April 2006)

**Jacobsen, Jens** (2005):

Website-Konzeption – Erfolgreich Web- und Multimedia-Anwendungen entwickeln.

München: Addison-Wesley Verlag

**Johnston, Paul** (2006):

Cryptography: JavaScript MD5.

<http://pajhome.org.uk/crypt/md5/>

(Datum des Zugriffs: 22. November 2005)

**Kaul, Kristina** (2005):

Web 2.0: Phantom oder Phänomen?

<http://www.dw-world.de/dw/article/0,2144,1790308,00.html>

(Datum des Zugriffs: 14. April 2006)

**Krause, Jörg** (2003):

Grundlagen und Profiwissen PHP 4 – Webserver-Programmierung unter Windows und Linux.

München; Wien: Carl Hanser Verlag

**Krug, Steve** (2006):

Don't Make Me Think! A Common Sense Approach to Web Usability.

Berkeley, USA: New Riders

**Leroy, Pascal** (2006):

in Wiedemann, Ed. Julius:

Web Design: Flash Sites.

Köln; London; Los Angeles; Madrid; Paris; Tokyo: Taschen (Icons)

Seiten 38 bis 40.

**Niederst, Jennifer** (2002):

Webdesign in a Nutshell.

Köln: O'Reilly Verlag

**Nielsen, Jakob** (2001):

Designing Web Usability.

München: Markt + Technik Verlag

**Schweibenz, Werner/Thissen, Frank** (2003):

Qualität im Web – Benutzerfreundliche Webseiten durch Usability Evaluation.

Berlin; Heidelberg; New York: Springer Verlag (X.media.press)

**Swann, Craig/Caines, Gregg** (2002):

XML in Flash.

Indianapolis, USA: Que Publishing

**wikipedia rich\_internet\_application** (2006):

[http://de.wikipedia.org/wiki/Rich\\_Internet\\_Application](http://de.wikipedia.org/wiki/Rich_Internet_Application)

(Datum des Zugriffs: 14. April 2006)

**Wolter, Sascha** (2004):

ActionScript 1 und 2 – Objektorientierung und Codedesign mit Flash MX 2004.

Bonn: Galileo Press