



Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Institut für Theoretische und Technische Informatik
Fachgebiet Rechnerarchitektur

DIPLOMARBEIT

Konzeption und Realisierung der Client-Komponenten
für ein P2P-File-Sharing-System mit Umsatzbeteiligung
für die Benutzer

Vorgelegt von: Jens Hasselbach
Verantwortlicher Hochschullehrer: Prof. Dr.-Ing. habil. W. Fengler
Betreuer: Dr.-Ing. J. Nützel
Datum der Ausgabe des Themas: 2. Mai 2002
Datum der Abgabe der Diplomarbeit: 4. November 2002

Inventarisierungsnummer: 2002-11-04/053/IN96/2231

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Diplomarbeit selbständig, nur unter Zuhilfenahme der aufgeführten Quellen und Hilfsmittel, verfasst zu haben.

Nordhausen, den 4. November 2002

Jens Hasselbach

Thesen zur Diplomarbeit

- Die Unterhaltungsindustrie verzeichnet Umsatzverluste durch das Verhalten der Konsumenten bezüglich Erwerb und Konsum von virtuellen Waren. Die Konsumenten betrachten virtuelle Waren als öffentliche Güter.
- Durch die Verbreitung von hochwertigen Kompressionsverfahren für virtuelle Waren (speziell: Multimediadaten) und P2P-File-Sharing-Systeme wird diese Entwicklung forciert.
- Die Unterhaltungsindustrie ist bestrebt, ihr Geschäftsmodell für reale Waren auf virtuelle Waren zu übertragen und die Kontrolle über ihre Inhalte zurück zu gewinnen. Aber der Erfolg, der auf Restriktion der Konsumenten basierenden Ansätze der Unterhaltungsindustrie, ist aus technischer und ökonomischer Sicht fragwürdig.
- Das „Potato System“ ist ein Multi-Level-Marketing-basiertes Geschäftsmodell zum Vertrieb von virtuellen Waren mit Umsatzbeteiligung für die Benutzer. Mit der Option für den Konsumenten, selbst Nutzen aus dem käuflichen Erwerb von virtuellen Waren zu ziehen, bietet sich eine Möglichkeit sowohl den Interessen der Nutzer, als auch den Interessen der Content-Provider gerecht zu werden.
- Die Client-Komponenten stellen die Funktionalitäten zur Verfügung, um die Authentizität und Integrität der angebotenen bzw. registrierten virtuellen Ware zu prüfen. Dies geschieht durch die Generierung von syntaktischen und semantischen Fingerabdrücken der Dateien und durch Extrahierung von Sample-Daten.
- Unter Berücksichtigung der Kriterien Portabilität, Benutzerfreundlichkeit und Sicherheit, stellen signierte Java-Applets die beste Wahl für die Realisierung der Prototypen der Client-Komponenten dar.

Nordhausen, den 4. November 2002

Jens Hasselbach

Inventarisierungsnummer: 2002-11-04/053/IN96/2231

Inhaltsverzeichnis

1	EINLEITUNG UND MOTIVATION	7
2	EINSCHÄTZUNG DER LAGE UND STAND DER TECHNIK	9
2.1	VIRTUELLE WAREN	9
	Was sind virtuelle Waren?	9
	Öffentliche Güter	10
	Eigentumsrechte und virtuelle Waren	10
2.2	DIE KRISE DER MUSIKINDUSTRIE.....	11
	Traditionelle Piraterie.....	13
	Internetpiraterie	13
	Privatkopien und Schulhofpiraterie.....	13
	Expertenmeinungen.....	14
2.3	KOPIERSCHUTZ- UND KONTROLLMECHANISMEN	16
2.3.1	<i>Kopierschutzverfahren für Audio-CDs</i>	<i>16</i>
	Cactus Data Shield (CDS100 + CDS200)	17
	SafeAudio.....	17
	Key2Audio	18
	MusicGuard.....	18
	MediaCloQ.....	19
	SafeAudioV3.....	19
2.3.2	<i>Video-DVD und DeCSS</i>	<i>20</i>
2.3.3	<i>Elektronische Bücher</i>	<i>21</i>
	Adobe Acrobat eBook Reader.....	21
	Microsoft Reader.....	21
	Glassbook Reader.....	22
	Gemstar Rocket-eBook	22
2.3.4	<i>DRM - Digital Rights Management</i>	<i>23</i>
	Sicherheitsziele und korrespondierende Techniken.....	23
	Spezielle DRM-Techniken: Digitale Wasserzeichen und Fingerabdrücke	24
	„Naive“ Sicherheitsmechanismen	26
	Microsoft DRM-2.....	26
	Sichere DRM-Systeme	27
	TCPA (Trusted Computing Platform Alliance).....	27
2.4	KOMPRESSIONSVERFAHREN FÜR AUDIO- UND VIDEODATEN	28
	MP3 (MPEG1 Layer 3).....	28

Inventarisierungsnummer: 2002-11-04/053/IN96/2231

Inhaltsverzeichnis

DivX.....	28
2.5 VERTRIEBSWEGE VIRTUELLER WAREN : METHODEN DES FILE-SHARINGS.....	29
2.5.1 „Klassische“ Methoden des File-Sharings.....	29
Usenet	29
Internet Relay Chat (IRC)	30
FTP (File Transfer Protocol)	30
WWW (World Wide Web).....	31
E-Mail	31
2.5.2 P2P-File-Sharing-Systeme	32
Ein kurzer Überblick über die Entwicklung von P2P	32
Koordiniertes P2P vs. Dezentrales P2P	33
P2P-File-Sharing-Systeme und die Unterhaltungsindustrie : Fallbeispiele	37
2.5.3 Die Angebote der Musikindustrie	39
2.5 ZUSAMMENFASSENDE BEURTEILUNG	40
3 KONZEPTION	41
3.1 DAS „POTATO SYSTEM“	41
Registrierungsbeleg im Dateinamen: Transaktionsnummern	44
Erweiterungen für das „Potato System“: Der Community-Ansatz	46
Dezentraler Ansatz vs. zentraler Ansatz.....	47
3.2 ANWENDUNGSFÄLLE	48
3.2.1 Als Provider registrieren	48
3.2.2 Als Vertriebspartner registrieren	49
3.2.3 Eine Datei registrieren	49
3.2.4 Eine Datei überprüfen	50
3.2.5 Eine Datei kaufen	51
3.2.6 Dateien verteilen.....	51
File Sharing	52
Dateidownload „on-demand“	52
3.3 DIE KOMPONENTEN DES „POTATO SYSTEMS“	54
3.3.1 Accounting Server.....	54
3.3.2 Creator	54
3.3.3 Redister.....	55
3.3.4 Zusammenfassung der Funktionalitäten der Client-Komponeten.....	57
3.3.5 Überlegungen zur Sicherheit	58
3.4 ENTWURF DER CLIENT - KOMPONENTEN	60

Inhaltsverzeichnis

4	IMPLEMENTIERUNG DER CLIENT-KOMPONENTEN.....	61
4.1	VERWENDUNG VON JAVA-APPLETS: VORÜBERLEGUNGEN	62
4.1.1	<i>Browser mit Java-Unterstützung</i>	63
4.1.2	<i>Verwendung des Sun Java Plug-ins.....</i>	65
4.1.3	<i>Sicherheitsrestriktionen für Applets.....</i>	67
4.1.4	<i>Signierte Applets.....</i>	68
4.1.5	<i>Fazit.....</i>	70
4.2	IMPLEMENTIERUNG DER PROTOTYPEN	70
4.2.1	<i>Dateien registrieren (Creator-Applet).....</i>	71
4.2.2	<i>Dateien überprüfen (Checker-Applet).....</i>	73
4.2.3	<i>Dateien umbenennen (Renamer-Applet).....</i>	74
5	ZUSAMMENFASSUNG UND AUSBLICK.....	76
	Anhang A: Init-Parameter für die Appletkonfiguration.....	77
	Anhang B: Struktur der XML-Dateien.....	79
	Quellenverzeichnis.....	81
	Abbildungsverzeichnis.....	85
	Tabellenverzeichnis.....	85

1 Einleitung und Motivation

Immaterielle Güter, insbesondere digitale Unterhaltungs- und Informationsprodukte, wie z.B. Musik- und Videoangebote, sind aus unserem täglichen Leben nicht mehr wegzudenken. Die Hersteller dieser Produkte (Musikindustrie, Filmindustrie, Softwareindustrie, Buchverlage) tätigen Investitionen, in der Erwartung, eine entsprechende Gegenleistung vom Konsumenten zu erhalten. Das Problem für die Hersteller besteht nun darin, dass sich digitale Produkte theoretisch von jedem Konsumenten ohne Qualitätsverlust kopieren und weiterverteilen lassen. Tatsächlich ist es so, dass sich viele Konsumenten hinsichtlich des Erwerbs und Konsums von immateriellen Gütern grundlegend anders verhalten als dies bei materiellen Gütern der Fall ist, d.h. sie kopieren und verteilen digitale Inhalte unabhängig vom geltenden Urheberrecht und teilweise unter Umgehung möglicher Kopierschutzmechanismen.

Besonders die Musikbranche muss seit der wachsenden Verbreitung von CD-Brennern und der nahezu unbegrenzten Verfügbarkeit von MP3-Raubkopien im Internet mit steigenden Umsatzrückgängen leben. Der weltweite Umsatz der Musikindustrie sank von 35,5 Milliarden US-Dollar im Jahr 2000 auf 33,7 Milliarden US-Dollar im Jahr 2001. Dies entspricht einem Umsatzrückgang von ca. 5%. Zwar spielen auch die ökonomische Flaute eine gewisse Rolle, die Hauptschuldigen sind aber nach Meinung der Musikindustrie die Kopierer und Raubkopierer. [Musik-02]

Aber nicht nur die Musikindustrie ist von dieser Entwicklung stark betroffen, sondern auch die Filmindustrie wird in zunehmendem Ausmaß mit den Auswirkungen der Verbreitung von Raubkopien konfrontiert. Als 1999 eine Raubkopie von „Star Wars – Episode I“ als 1,2 GB großes MPEG-Video im Internet zum kostenlosen Download angeboten wurde, war das Interesse der Filmindustrie, wohl aufgrund der Dateigröße und der schlechten Qualität, eher gering. Aber mit der stetigen Weiterentwicklung des DivX-Videocodex (welcher eine hohe Komprimierungsrate in Verbindung mit hoher Bildqualität erlaubt) und der Verbreitung von Breitband-Internetzugängen, erfolgte der Durchbruch für den illegalen Tausch von Filmkopien. Alleine für das Jahr 2002 rechnet die US-Filmindustrie deshalb mit Einnahmeverlusten in Milliardenhöhe. [PCWelt] & [Zota-01]

Die Reaktion der „Unterhaltungsbranche“ ist der Versuch, mittels Kopierschutz- und Kontrollmechanismen und Prozessen gegen Raubkopierer ihre Rechte durchzusetzen und so einem drohenden „Kontrollverlust“ über ihre Inhalte entgegenzuwirken. Die oben genannten Umsatzzahlen sprechen allerdings für sich und auch die Einschätzungen von Marktforschungsinstituten wie Forrester Research sagen diesen Versuchen keine großen Erfolgchancen voraus. Eric Scheirer von Forrester Research: „Die Kunden haben gesprochen. Sie fordern den Zugang zu Inhalten mit allen dazu nötigen Mitteln. Weder digitale Sicherheit noch Prozesse werden den Diebstahl von Inhalten im Internet beenden können. [...Die Verleger] müssen attraktive Dienste mit den Inhalten, in den Formaten und mit den Geschäftsmodellen anbieten, die von den Kunden gewünscht werden.“ [RötzerCOC]

Wie könnte nun solch ein Modell aussehen, dass sowohl den Interessen der Konsumenten als auch den Interessen der „Unterhaltungsbranche“ gerecht wird? Während die Hersteller ihre Eigentumsrechte und damit ihre Preisvorstellungen durchsetzen wollen, kann man auf der Konsumentenseite von dem Streben nach individueller Nutzenmaximierung ausgehen. Neben Güterversorgung und Gewinn können auch „Güter“ wie Prestige, Macht und Wohlergehen anderer das Ziel dieser individuellen Maximierung sein.

Das angestrebte Modell müsste also für den Konsumenten einen Anreiz schaffen, für das Produkt zu bezahlen, der stärker ist als alle sonstigen Anreize. Infolge dieser dann vorhandenen Zahlungsbereitschaft, sollte der Hersteller die Verteilung seines Produktes durch die Konsumenten als in seinem eigenen Interesse liegend erkennen. Genau diesen Ansatz verfolgt das „Potato System“, welches im Folgenden hinsichtlich verschiedener Aspekte, wie Sicherheit und technischer Realisierbarkeit untersucht werden wird.

Das in dieser Arbeit behandelte „Potato System“ ist kein weiterer „Napster“-Clone oder eine P2P-Applikation im Sinne der Definition unter 2.3.2. Es handelt sich hierbei vielmehr um ein Geschäftsmodell mit der Besonderheit der Umsatzbeteiligung für die Benutzer. Diese Umsatzbeteiligung soll die Grundlage für die Zahlungsbereitschaft der Konsumenten von digitalen Gütern schaffen. Im Kontext des „Potato Systems“ wird der Begriff „Peer-to-Peer“ in seiner allgemeinen Bedeutung benutzt, d.h. es bleibt völlig offen auf welchem Weg die Dateien getauscht bzw. vertrieben werden. Denkbar ist also der Download von speziellen Servern, der Erwerb auf Medien (CD, DVD etc.) oder aber eben der Tausch mittels P2P-File-Sharing-Systemen im Stil von „Napster“ oder „Gnutella“.

2 Einschätzung der Lage und Stand der Technik

2.1 Virtuelle Waren

Was sind virtuelle Waren?

Waren sind Produkte oder Dienstleistungen, welche auf einem Markt einen bestimmten Preis erzielen. Im Gegensatz zu anderen Produkten haben Medienprodukte (Musik, Film, Printmedien) die Eigenschaft, dass sie nicht nur in realer Form (d.h. in Verbindung mit einem physischen Medium, z.B. Musik auf einer CD, ein Film auf einer DVD, ein Buch) existieren können, sondern auch in einen virtuellen Zustand überführt werden können.

Die Überführung aus der realen Form in die virtuelle Form wird durch das Verfahren des Digitalisierens bzw. durch die Loslösung, des bereits in digitaler Form auf einem physischen Datenträger vorliegenden Inhaltes, erreicht (siehe Abbildung 1).

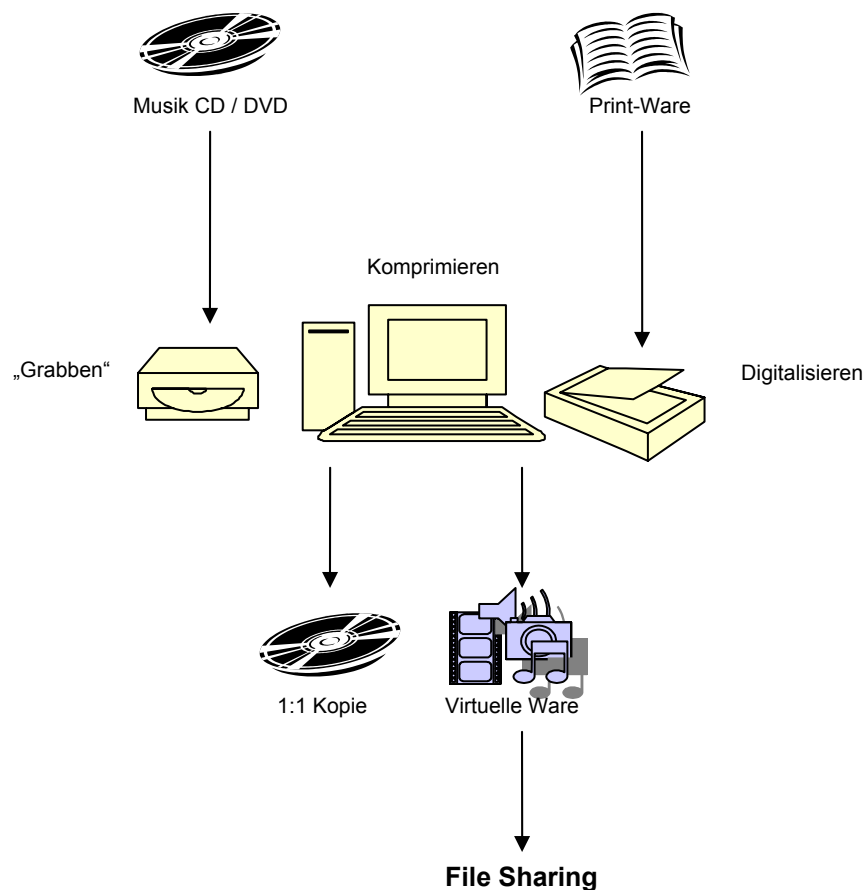


Abbildung 1: Die Überführung in den virtuellen Zustand

Die Digitalisierung und der Prozess der Trennung von digitalem Inhalt und Datenträger sind also Vorgänge um den Zustand der Virtualität zu erreichen („Virtualisierung“). Wenn im Folgenden von „virtuellen Waren“ gesprochen wird, sind damit alle Medienprodukte gemeint, welche in virtueller Form vorliegen. [GriNütz-02]

Auch Software-Produkte jeglicher Art (Spiele, Anwendungen) sind virtueller Natur. Im Rahmen dieser Arbeit werden jedoch ausschließlich verbreitete „Multimedia-Formate“ (konkret: MP3, PDF, diverse Bildformate, DivX) betrachtet. Das Konzept, des in dieser Arbeit vorgestellten „Potato Systems“, lässt sich aber prinzipiell auf alle virtuellen Waren anwenden.

Öffentliche Güter

Der Charakter eines öffentlichen Gutes wird durch zwei Merkmale gekennzeichnet (siehe [Heinrich-94]):

- **Nicht-Rivalität:** Ein Gut kann konsumiert werden, ohne dass der Konsum einer Person den Konsum anderer Personen beeinträchtigt.
- **Nicht-Ausschluss:** Nicht zahlungsbereite Konsumenten können nicht vom Konsum ausgeschlossen werden.

Virtuelle Güter besitzen offensichtlich das Merkmal der „Nicht-Rivalität“, beispielsweise nutzt sich ein MP3-Musikstück durch tausendfaches kopieren und konsumieren nicht ab. Die Anwendbarkeit des Ausschlussprinzips hingegen ist in der Regel eine Frage der Technik (siehe Kapitel „Kopierschutz- und Kontrollmechanismen“) und des Aufwandes (und damit der Kosten) und bildet die Voraussetzung zur Durchsetzung von Eigentumsrechten.

Eigentumsrechte und virtuelle Waren

Nach der Theorie der „Property Rights“ [Picot], in deren Zentrum das individuelle Eigentumsrecht an Gütern steht, können folgende Einzelrechte unterschieden werden:

- Das Recht, ein Gut zu benutzen
- Das Recht, ein Gut zu verändern
- Das Recht auf die Gewinne (aber auch die Verluste), die das Gut einbringt
- Das Recht, das Gut zu verkaufen

Die Zuweisung von Eigentumsrechten spielt eine zentrale Rolle im wirtschaftlichen Leben. Ohne die Möglichkeit, der Durchsetzung von Eigentumsrechten, ist keine kommerzielle Verwertung von Produkten möglich, da das Produkt zu einem „öffentlichen Gut“ werden würde. Die ursprünglichen Eigentumsrechte wären praktisch nicht mehr vorhanden.

Die Erfahrungen mit File-Sharing-Systemen wie „Napster“ haben gezeigt, dass sehr viele Konsumenten virtuelle Waren als öffentliche Güter wahrnehmen und sich dementsprechend anders verhalten, als bei realen Gütern.

Dieses Verhalten der Konsumenten in der Praxis stimmt mit dem Charakter des „homo oeconomicus“ überein, wie er in der „Property Rights“-Theorie beschrieben wird. Hierbei wird dem Konsumenten unterstellt, dass er stets bestrebt ist, seinen persönlichen (wirtschaftlichen) Nutzen zu optimieren und seine Handlungen dementsprechend auszurichten. [Picot]

Virtuelle Waren lassen sich mittels handelsüblicher Computertechnik mit geringem Aufwand reproduzieren und verteilen. Die Frage, die sich nun stellt ist: Kann man virtuelle Waren besitzen, d.h. ist die Durchsetzung von Eigentumsrechten für virtuelle Waren überhaupt möglich?

In den folgenden Kapiteln wird untersucht werden, welche Ansätze es in der Vergangenheit und in der Gegenwart gab und gibt, um Eigentumsrechte für virtuelle Waren durchzusetzen und wie erfolgreich deren Umsetzung in der Praxis war bzw. ist.

2.2 Die Krise der Musikindustrie

Wie schwer ist die Krise der Unterhaltungsindustrie (speziell der Musikindustrie) wirklich? Wenn man den Stimmen von Vertretern der Branche glauben kann, gehen Künstler und Musikindustrie schweren Zeiten entgegen.

Die wichtigsten Daten des Jahres 2001 aus der Sicht der deutschen Musikindustrie werden im Jahreswirtschaftsbericht 2001 des IFPI (Bundesverband der Phonographischen Wirtschaft e.V.) wie folgt zusammengefasst [IFPI2002]:

- drastischer Umsatzrückgang um 10,2%
- Absatz 2001: 244,1 Millionen Tonträger (2000: 266,4)
- Erstmals mehr mit Musik kopierte CD-Rohlinge (182 Mio.) als verkaufte CD-Alben (173,4 Mio.)
- Fast 500 Millionen Downloads aus zumeist illegalen Musikangeboten im Internet
- Private Vervielfältigung bedroht Investitionskraft und kulturelle Vielfalt des deutschen Musikmarkts
- Tonträgerhersteller setzen Kopierschutz ein
- Umgehung von Kopierschutz muss verboten werden: EU-Urheberrechtsrichtlinie muss schnell in deutsches Recht umgesetzt werden

Und in der Rede von Gerd Gebhardt, Vorsitzender der deutschen Phonoverbände, zur Jahrespressekonferenz am 21.3.2002 heißt es:

„[...] Die Musikwirtschaft ist in einer schwierigen Lage, weil die Rahmenbedingungen für ein funktionierendes Geschäftsmodell zurzeit akut bedroht sind. Im Ernst: Wer heute ganz legal kopiert oder sogar Kopien für Dritte herstellt, verhält sich doch aus seiner Sicht vernünftig, wenn er Musik lieber kopiert als kauft und das gesparte Geld für DVDs oder ein Handy ausgibt. Aber die Künstler und die Musikwirtschaft gehen dabei leer aus.“ [GebhJPK-02]

Der Umsatzrückgang der deutschen Musikbranche wird laut IFPI auf verschiedenen Arten der Musikpiraterie zurückgeführt. Vor allem seien Downloads von illegalen Musikangeboten im Internet sowie die „Schulhofpiraterie“, also das verbotene Verkaufen von kopierten CDs im Freundes- und Bekanntenkreis, für einen Großteil der Verluste verantwortlich. Als weitere Ursachen für die Verluste werden „Traditionelle (gewerbsmäßige) Piraterie“ und der Bereich der (legalen) Privatkopien genannt. Wäre die kopierte Musik gekauft worden, hätte sie einen Umsatzwert von etwa 3,5 Milliarden Euro gehabt. Der Bericht räumt allerdings ein, dass zwar nicht jede Kopie ein Kaufverlust sei, aber anhand der Zahlen doch die Größenordnung des Problems zu erkennen sei.

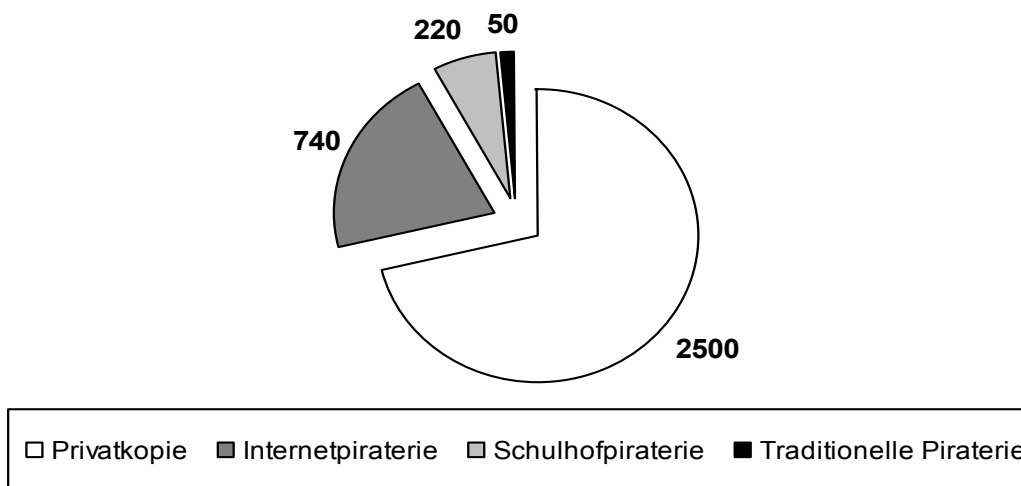


Abbildung 2: Umsatzverluste der deutschen Musikindustrie durch Raubkopien (in Mio. Euro) [IFPI2002]

Während die Musikindustrie die Schuldigen an ihrer Misere also klar zu erkennen glaubt – Raubkopierer und File-Sharing-User – sieht die Frankfurter Finance die überholte Preispolitik und monopolartige Strukturen als eigentliche Ursache: „Seit der Einführung der Compact Disc (CD) Mitte bzw. Ende der 80er Jahre haben sich die Preise der neuen Form von Tonträgern kaum reduziert. Als die CD Mitte bzw. Ende der 80er kam, zahlte man anfangs 35 DM für einen Tonträger – ein deutlicher Aufschlag zur vormaligen Schallplatte. Heute aber ist die Industrie fast komplett auf CD umgestellt und ein neuer Tonträger kostet überwiegend immer noch 35 DM (bzw. 17 Euro), obwohl die Produktionskosten deutlich geringer sind als für Vinyl. Sicher könnte man als Argument die höheren Marketingkosten gegenüber früher anführen.“

Jedoch sieht man in der Musikindustrie alle Anzeichen, die mangelnden Wettbewerb und monopolartige Strukturen charakterisieren: Ineffizienz, hohe Kosten, Bürokratie, hohe Preise, geringe Innovationsneigung!“ [FFinance-01]

Mit welchen Mitteln begegnet nun die Musikindustrie den verschiedenen Formen der Piraterie?

Traditionelle Piraterie

Die Umsatzverluste durch gewerbsmäßige Piraterie werden als relativ gering eingeschätzt. Dies ist wohl darauf zurückzuführen, dass „durch optimale Zusammenarbeit mit Polizeibehörden, Staatsanwaltschaften und Zoll ein hoher Ermittlungsdruck besteht, der die Piraterieszene eingrenzt und den Aufbau großer Lager unterbindet“.

Internetpiraterie

Im dem Bericht wird festgestellt, dass sich die Verluste durch Onlinepiraterie gegenüber dem Vorjahr verdoppelt hätten und die illegalen Erben von „Napster“ „dramatisch an Zahl und Volumen der über sie getauschten Musik-Downloads“ gewachsen seien. Allerdings wird ebenfalls festgestellt dass man „sehr erfolgreich“ gegen illegale Musikangebote im Internet vorgegangen sei und im Jahr 2001 weit über 1.200 Anbieter abgemahnt habe, die daraufhin ihre Angebote vom Netz nahmen.

Privatkopien und Schulhofpiraterie

Es ist leicht zu erkennen, dass der weitaus größte Anteil auf Privatkopien zurückzuführen ist. Als Ursache hierfür wird die stetig wachsende Verbreitung von CD-Brennern angesehen, welche dann in der Folge als Hauptursache für die Umsatzverluste erkannt wird. Der Bericht beruft sich dabei auf eine Studie der Gesellschaft für Konsumforschung (GfK).

„Aktuelle Marktforschungsergebnisse für 2001 liegen inzwischen von der GfK vor. Danach steigt die Zahl der CD-Brenner in Deutschland rapide. [...] Im Jahr 2001 sind insgesamt 332 Millionen CD-Rs/CD-RWs verkauft worden, etwa 45% mehr als im Vorjahr. Etwa 47% davon werden legal mit Musik bespielt (Privatkopien), weitere 8% fallen unter „Schulhofpiraterie“. Die GfK hat auch ermittelt, dass 17,5% aller Personen, die CDs brennen, weniger Alben kaufen, nur 4,8% kaufen mehr. Per Saldo kaufen damit rund 10% dieser Tonträger-Kunden weniger Alben. Hierin liegt die Hauptursache für die deutlichen Umsatzverluste der Phonowirtschaft im vergangenen Jahr.“ [IFPI2002]

Wie die Musikindustrie die Verluste im Sektor der privaten Kopien verringern will, wird in der bereits erwähnten Rede des Vorsitzenden der deutschen Phonoverbände deutlich:

„Kopierschutz ist die eine Antwort auf die Frage nach den Lösungen. Um hier effektiv zu sein, brauchen wir aber die Unterstützung des Gesetzgebers. [...] Ebenso wollen wir, dass das Umgehen von Kopierschutz gesetzlich untersagt wird. Wir können und wollen niemandem in sein Arbeitszimmer schauen – aber wir können dann die Anleitung zum Knacken und das Angebot entsprechender Geräte und Programme unterbinden.“ [GebhJPK-02]

Konkret soll die Forderung nach staatlicher Unterstützung in der geplanten Urheberrechtsnovelle umgesetzt werden. Demnach dürfen technische Maßnahmen, die „im normalen Betrieb dazu bestimmt sind“, Werke oder andere Schutzgegenstände vor nicht von den Urhebern genehmigten Handlungen zu schützen, „ohne Zustimmung des Rechtsinhabers nicht umgangen werden“. Zudem sollen die Herstellung, die Einfuhr, die Verbreitung, der Verkauf, die Vermietung sowie die Werbung für und der Besitz von Werkzeugen oder Vorrichtungen zur Umgehung der Schutzmaßnahmen zu „gewerbsmäßigen Zwecken“ verboten werden. [KrempIURH-02]

Da durch gesetzliche Regelungen kein Schutz „de facto“ gegeben ist, soll dieser durch den Einsatz von DRM-Systeme (Digital Rights Management) erreicht werden. DRM-Systeme verbinden Techniken wie Wasserzeichen, Verschlüsselung und Systeme zur Verwaltung von Benutzerrechten, um digitale Inhalte zu schützen. Experten bezweifeln allerdings die Wirksamkeit und Sicherheit von DRM-Systemen im praktischen Einsatz.

Expertenmeinungen

Gene Kan, Unterstützer eines der zahlreichen „Gnutella“-Projekte, über die Filmindustrie und DRM: „Die Verschlüsselung ist ein Witz“. Alle Kryptografie-Bemühungen der Filmindustrie seien reine Zeit- und Geldverschwendung, da die Mechanismen bereits geknackt wären, wenn die Videos ins Netz gestellt würden. Als Beispiel führt er Sightsounds an, bei deren Methode es ausreiche, wenn ein Hacker einen Film kaufe und ihn dann in ein anderes Format (etwa DivX) umwandle. Seiner Meinung nach gehört die Piraterie fremden intellektuellen Eigentums im digitalen Zeitalter zur Tagesordnung. [KrempIDRM-00]

Das Marktforschungsinstitut Forrester Research kommt in seinem Bericht „Content out of control“ zu dem Schluss, dass das digitale geistige Eigentum technisch nicht wirksam geschützt werden könne – und auch juristisches Vorgehen keine Lösung sei. Auch Vertriebsformen, die auf der Kontrolle der Inhalte beruhen, wird kein Erfolg vorausgesagt. Die Konsumenten werden sich keinen Kaufregeln oder einer restriktiven Technik unterwerfen. „Und es genügt, wenn nur ein Mensch die Sicherheitsmaßnahmen durchbricht und die Inhalte im Netz weitergibt.“ [RötzerCOC-00]

Die Hersteller von DRM-Systemen (Adobe, IBM, Intel, Intertrust, Microsoft) halten ihre Produkte für ausgereift und einsatzfähig. Zu einer anderen Einschätzung kommt Hannes Federrath, Ingenieur im Bereich Informations- und Kodierungstheorie an der TU Dresden auf der „Digital Rights Management“-Konferenz: Nach seiner Analyse bieten alle Verfahren zum digitalen Rechtemanagement zahlreiche Schwachpunkte, die ein „ernsthafter Angreifer“ leicht ausnutzen könne. Gerade die wegen ihrer geringen Kosten bei der Industrie beliebten Softwarelösungen bieten nach Federrath „fast keinen Schutz“ vor Angriffen. Unerwünschtes Kopieren verhindern könnte nur die Verlagerung der Öffnung der Inhalte in eine „vertrauenswürdige Hardware“. [HeiseDRM-02]

Der Informationswissenschaftler Rainer Kuhlen (Universität Konstanz) vertritt die Meinung, dass die geplante flächendeckende Einführung von Kontrolltechnologien wie dem DRM keine Lösung für den Konflikt zwischen Industrie und Konsumenten bietet. Schließlich sei die künstliche Verknappung von Informationsgütern noch nie eine wirtschaftlich tragfähige Lösung gewesen: „Märkte werden umso größer, je offener und freizügiger die Nutzung und Verbreitung von Wissen betrieben werden kann.“ Frei muss dabei nicht immer kostenlos heißen, aber frei von Kontrolle. [Krempf-02]

2.3 Kopierschutz- und Kontrollmechanismen

Im Folgenden werden verschiedenen zurzeit gebräuchlichen Kopierschutzverfahren und Kontrollmechanismen vorgestellt und (bekannte) Wege gezeigt, diese zu umgehen. Dies soll keine Anleitung zum Erstellen von Raubkopien sein. Es soll lediglich gezeigt werden, dass es nur eine Frage der Zeit ist, bis jedes (bei seinem Erscheinen als sicher angesehene) Verfahren „geknackt“ wird.

2.3.1 Kopierschutzverfahren für Audio-CDs

Miniaturisierung und Digitalisierung haben den Musikmarkt in den letzten Jahrzehnten entscheidend beeinflusst und verändert. Die Miniaturisierung von Tonbandgeräten war der erste Schritt in Richtung Mobilität des individuellen Musikkonsums. Der Digitalisierung stand man zunächst (1980) mit Skepsis gegenüber. Es wurde bezweifelt, ob diese Technologie wirklich so wesentlich besser sei, als die bisherige Schallplatte. 1979, nach 10jähriger Entwicklungszeit von Ingenieuren von Sony und Philips, konnte Philips den ersten CD-Spieler vorführen. Die CD konnte Dank ihrer außergewöhnlichen Tonqualität überzeugen und löste nach einer Umsatzkrise auf dem gesamten Tonträgermarkt zu Beginn der 80er Jahre ein enormes Wachstum aus.

Jahr	CD-Verkäufe (Weltweit in Mio. Stück)
1983	4,8
1985	5,5
1990	770,0
1998	1.200,0

Tabelle 1: Weltweite CD-Verkäufe (1983 – 1998) [KrögerMM-99]

Bei Einführung der CD war das private Kopieren einer CD (ohne Qualitätsverlust) noch undenkbar, folglich spielte der Gedanke des Schutzes der Eigentumsrechte noch keine so große Rolle. Aber seit CD-Brenner zur Standardausstattung eines PCs gehören, hat sich die Situation drastisch geändert. Laut einer aktuellen Studie (2002) der Gesellschaft für Konsumforschung (GfK), haben rund 19 Millionen Menschen Zugang zu einem CD-Brenner. 17,1 Millionen davon brennen auch Musik. 23,4% aller Personen über 10 Jahre können zu Hause, 27,3% im Büro auf einen CD-Brenner zugreifen – und nutzen ihn hauptsächlich für Musikkopien. [GebhJPK-02]

Mittels verschiedener Kopierschutzverfahren soll verhindert werden, dass 1:1-Kopien von Audio-CDs erstellt werden. Um zu verhindern, dass Audio-Tracks ausgelesen bzw. überhaupt von PC-CDROM-Laufwerken erkannt werden können, manipulieren einige Verfahren die TOC (Table Of Contents) der zu schützenden CD oder verschleiern, welche Art von Track (Audio oder Daten) sich auf der CD befindet.

Einige Verfahren benutzen auch Mechanismen (z.B. manipulierte Fehlerkorrekturdaten), um die Qualität einer Kopie negativ zu beeinflussen. [ChipK2A]

In den folgenden Abschnitten werden die gebräuchlichsten Verfahren - und Angriffstechniken um sie zu umgehen - vorgestellt. Die zahlreichen erfolgreichen Angriffe sind als Indiz für die Verwundbarkeit der (derzeitigen) Kopierschutzverfahren zu werten (siehe auch Kapitel „DRM – Digital Rights Management“).

Cactus Data Shield (CDS100 + CDS200)

Ab Anfang des Jahres 2000 begann BMG Entertainment, die Musik-Abteilung des Bertelsmann-Konzerns, Audio-CDs in den Handel zu bringen, welche mit dem Kopierschutz „Cactus Data Shield“ (CDS) versehen waren. Entwickelt wurde dieser Mechanismus in Zusammenarbeit mit dem israelischen Software-Unternehmen Midbar und Sonopress aus Deutschland.

CDS verhindert nicht nur das Auslesen („Grabben“) und Kopieren der Audio-Tracks mit dem PC, sondern auch das Abspielen auf allen CD-ROM-Laufwerken. Der Kopierschutz beruht auf manipulierten TOC-Einträgen (Table of Contents).

Der Kopierschutz wirkt sich nicht auf die Überspielung per SPDIF-Digitalausgang aus. Es ist also möglich, eine mit dem CDS-Verfahren geschützte CD per DAT oder Audio-CD-Recorder zu überspielen. Eine so erstellte Kopie verhält sich wie eine nicht kopiergeschützte Audio-CD.

Da Audio-CDs mit ungültigen TOC-Einträgen den Red-Book-Standard (siehe [IEC-908]) verletzen, dürfen sie demzufolge auch das Siegel „Compact Disc Digital Audio“ nicht tragen. Nach massiven Problemen zahlreicher Kunden mit CDS-geschützten CDs gestand BMG Schwierigkeiten mit dem Kopierschutz ein. Auch der Handel zeigte sich wegen der erhöhten Rücklaufquote verärgert. [CTCDS-00] & [MP3World]

SafeAudio

„SafeAudio“ (Macrovision) baut gezielt Störungen in die Datenstruktur der CD ein, welche die Fehlerkorrektur eines CD-Players oder CD-ROM-Laufwerks beim Abspielen problemlos herausfiltert und somit nicht vom Hörer wahrgenommen werden. Beim „Grabben“ der Audio-CD erzeugt „SafeAudio“ hingegen Fehlermeldungen und verweigert die Kopie.

Mittels einer abgewandelten Version des Gerätetreibers CDFS.vxd für Windows 9x, lässt sich „SafeAudio“ jedoch umgehen. Dieser mountet die Audio-Tracks als WAV-Dateien und gibt somit direkten Zugriff auf die einzelnen Tracks, ohne dass man diese überhaupt von der CD „grabben“ muss. Allerdings funktioniert dieses Vorgehen nicht mit allen Laufwerken. [SafeAudio-01]

Key2Audio

„Key2Audio“ (Sony) arbeitet ebenfalls mit einer Manipulation der TOC, so dass nach den Audio-Tracks ein (korrupter) Daten-Track erscheint. Während Audio-CD Player die Daten am äußeren Rand ignorieren, versagt bei CD-ROM-Laufwerken die Fehlerkorrektur, so dass diese die CD gar nicht erst einlesen können. Zusätzlich ist das Copy-Bit gesetzt, somit wird die Original-CD bereits als Kopie gekennzeichnet. Dies soll das Anfertigen von Kopien mittels des SPDIF-Ausganges verhindern. Die Musik lässt sich somit aber auch nicht mehr auf Mini-Disc kopieren.

Aber auch dieser Kopierschutz lässt sich durch einen simplen Trick umgehen: Mittels eines dunklen Filzstiftes werden die Trennlinie zwischen den Audio-Tracks und dem Daten-Track und Teile des Daten-Tracks abgedeckt. Das PC-Laufwerk findet nun unter der in der TOC angegebenen Stelle nichts und behandelt das Medium fortan wie eine normale Audio-CD. (Dieses Vorgehen funktioniert auch bei CDS-geschützten CDs.) [ChipK2A-02]

Sony reagierte im Januar 2002 mit einer Änderung des Schutzverfahrens. Demnach macht man nun auf geschützten CDs die ursprünglich sichtbare Trennlinie zwischen den Musiktracks und dem Bereich, der den korrupten Daten-Track enthält, unsichtbar, damit die richtige Stelle für ihren Anti-Schutz-Strich nicht mehr zu lokalisieren ist. [HeiseK2A-02]

Doch auch diese Maßnahme brachte nicht den gewünschten Erfolg. Laut einem Testbericht des Magazins für Computertechnik (CT) ließ sich eine mit „Key2Audio“ geschützte Test-CD (Maxi-CD ‘Pop’ der Gruppe Nsync) mittels eines Plextor-CD-RW-Brenners sowohl mit WinOnCD 3.8 als auch via WinDAC „graben“ beziehungsweise kopieren. [CTK2A-01]

MusicGuard

„MusicGuard“ wurde von der Firma TTR Technologies entwickelt. Die Arbeit an diesem Verfahren wurde jedoch eingestellt. Stattdessen entwickelte TTR Technologies zusammen mit Macrovision das „SafeAudio“ Verfahren weiter (siehe „SafeAudioV3“).

Während der CD-Herstellung wird der Kopierschutz in den Glas-Master eingebettet und besondere Informationen werden in den Datenstrom eingefügt. Diese beeinflussen die Musikqualität nicht. Versucht der Anwender, eine Kopie zu erstellen, bricht entweder der Vorgang ab oder die Soundqualität der Kopie verschlechtert sich gegenüber dem Original drastisch. „MusicGuard“ ist praktisch ohne Bedeutung, da es nicht eingesetzt wurde. [TecMG-00]

MediaCloQ

„MediaCloQ“ wurde von der Firma Sunncomm entwickelt. Wie „MediaCloQ“ genau arbeitet ist schwer herauszufinden, da sich die Entwickler sehr bedeckt mit Informationen über das Verfahren halten. Es gibt zwei Versionen dieses Verfahrens, wobei die zweite Version scheinbar prinzipiell die gleiche Funktionsweise besitzt wie die erste.

„MediaCloQ“-CDs werden im Multisession-Format ausgeliefert und auch dieses Verfahren arbeitet mit einer veränderten TOC: Der Kopierschutz kennzeichnet Audio-Tracks als Datenspuren. Deshalb können die meisten CD-ROM-Laufwerke „MediaCloQ“-CDs nur mit Mühe einlesen. Im Explorer sind die Audio-Tracks nicht sichtbar, angezeigt werden nur einige Dateien. Darunter eine EXE-Datei (executable), die auf die Website des jeweiligen Musik-Labels zugreift.

Die geschützten CDs sorgen für Qualitätseinbußen bei der Kopie, indem das beim Audio-Grabbing auftretende ‚Jitter‘-Problem (nach einer Unterbrechung des Datenstroms gehen ohne Korrekturmaßnahmen Blöcke verloren, da der exakte Punkt des Leseabbruchs nicht bestimmt werden kann) künstlich verstärkt wird. Mit dem Einsatz eines modernen CD-Brenners und geeigneter Software lässt sich jedoch auch dieser Kopierschutz oft umgehen. [CDMedia-02]

SafeAudioV3

Seit Anfang 2002 ist „SafeAudioV3“, welches in Zusammenarbeit von Macrovision und TTR Technologies entwickelt wurde, verfügbar. Im Unterschied zu Sonys „Key2Audio“ ist dieses Verfahren nicht darauf ausgerichtet, das Abspielen von Audio-CDs am PC zu verhindern. Wie bereits seine Vorgänger scheint „SafeAudioV3“ durch gezielt verfälschte Fehlerkorrekturdaten (Error Correction Codes, ECC) für starkes Rauschen bei digital ausgelesenen Compact Discs zu sorgen. Ein neues Verfahren namens „AudioLock“ soll zusätzliche Sicherheit bringen und verschiedene Sicherheitsstufen ermöglichen. Außerdem unterstützt „SafeAudioV3“ Multisession-CDs und ist somit für weitere Kopierschutzverfahren offen, etwa um per digitaler Rechteverwaltung (DRM) von Drittanbietern geschützte MP3-Dateien auf einem Daten-Track zu realisieren.

Da dieser Kopierschutz noch relativ neu ist, sind noch keine effektiven Umgehungsmöglichkeiten bekannt. Es kann aber davon ausgegangen werden, dass mit wachsender Verbreitung von derart geschützten CDs, auch hierfür Angriffsmethoden veröffentlicht werden. Eine Möglichkeit, welche auch bei allen anderen genannten Verfahren anwendbar ist, ist die Digital/Analog-Wandlung und anschließend ein erneutes Digitalisieren. [SafeAudioV3-02]

2.3.2 Video-DVD und DeCSS

Die Filmindustrie versuchte, die Daten ihrer Video-DVDs (Digital Versatile Disc) durch ein „Content Scrambling System“ (CSS) vor unbefugtem Zugriff und Raubkopien zu schützen. Jeder Hersteller eines DVD-Players, Hardware- oder Software-Decoders erhielt einen Schlüssel (40-bit Key), um die DVD zu entschlüsseln. Dazu liegen in einem geschützten Bereich auf jeder Video-DVD alle erlaubten Schlüssel.

Mit der Verbreitung der Cracker-Tools „DOD DVD Speed Ripper“ und „DeCSS“ (im Herbst 1999) dürfte ein Albtraum der Hollywood-Studios wahr geworden sein: Beide Werkzeuge entschlüsseln mittels des CSS-Verfahrens geschützte DVD-Videos und kopieren die Datenströme auf die Festplatte. Im Unterschied zum „Speed Ripper“ enthält „DeCSS“ einen kompletten Schlüssel, der alle derzeit auf dem Markt befindlichen DVD-Videos knackt. „DeCSS“ wurde ursprünglich vom Hacker-Trio „MoRE“ entwickelt, um DVD-Filme unter Linux sehen zu können.

„DeCSS“ macht sich im Wesentlichen einen Implementierungsfehler des Software-DVD-Players von Xing Labs zu Nutze: Die Entwickler von Xing hatten ihren Player Key offenbar ungenügend geschützt. Nach „DeCSS“ tauchten im Internet in rascher Folge Methoden zur Durchführung von Brute-Force-Attacken gegen das Schlüsselsystem auf und schließlich sogar eine Auflistung aller 400 Player-Keys. [DeCSS1-99] & [DeCSS2-99]

Während die Filmfirmen in der Software eine Verletzung des DMCA (Digital Millenium Copyright Act: 1998 vom US-Kongress verabschiedetes Gesetz zum Schutz des Copyrights von elektronischen Medien [DMCA]) sehen, machen Open-Source-Vertreter und Verbraucherschützer geltend, „DeCSS“ garantiere Anwendern die Freiheit, DVDs auf Betriebssystemen ihrer Wahl sehen zu können.

Inzwischen existieren auch genügend andere Hacker-Tools, welche eine kryptografische Attacke auf den Schlüssel der zu decodierenden DVD durchführen und somit den durch Reverse Engineering gewonnenen Xing-Schlüssel gar nicht benutzen. [DeCSS3-02]

2.3.3 Elektronische Bücher

Adobe Acrobat eBook Reader

Im November 1999 stellte Adobe mit „Web Buy“ eine E-Commerce-Lösung für PDF-Dokumente zur Verfügung. Adobe bemühte sich, sein PDF-Format für den Einsatz auf E-Books vorzubereiten und dem kurz zuvor verabschiedeten Open-eBook-Standard (OEB) etwas entgegenzusetzen, hinter dem unter anderem Microsoft steht.

Am 15. Juli 2001 stellte der russische Programmierer Dimitry Sklyarov auf dem Hackertreffen „DefCon 9“ ein Programm vor, mit dem sich der Kopierschutz des „Acrobat eBook Readers“ umgehen lässt. Der Programmierer hatte den „Advanced eBook Processor“ (AEBPR) im Rahmen seiner Doktorarbeit geschrieben, um die Lücken beim Kopierschutz von PDF-Dateien zu demonstrieren. Der AEBPR habe auch legitime Zwecke, so kann der rechtmäßige Besitzer von Dokumenten für den „Acrobat eBook Reader“ erst unter Umgehung des Kopierschutzes eine (Sicherheits)kopie anfertigen, das Dokument auf einem anderen Rechner verwenden oder es ausdrucken. [AcroReader1+2]

Microsoft Reader

Im Februar 2000 präsentierte Microsoft seinen „E-Book-Reader“. Die dazugehörige Software „Microsoft Reader“ gibt es nicht nur für spezielle Lesegeräte, sondern auch für die Windows-PC-Plattform. Im August 2000 erschien die kostenlose PC-Version des „MS Readers 1.5“. Bis dahin gab es die Software nur vorinstalliert in der Version 1.0 für die neueste Generation der Windows-CE-Geräte. Für (englisches) Lesematerial tat sich Microsoft mit der Buchhandelskette Barnes & Noble zusammen, um über deren Online-Buchhandlung für den „MS Reader“ formatierte E-Books anzubieten.

Ein Jahr später, Ende August 2001 wurde der Kopierschutz des „Microsoft Readers“ geknackt. Dies gelang einem Hacker, der sein Tool einem Reporter des Technology Review, einer Zeitschrift des Massachusetts Institute of Technology (MIT) in Boston, vorgestellt hatte. Demnach sei es jetzt möglich, E-Books über die Software auf weitere Computer und Notebooks als diejenigen zu laden, für die das jeweilige E-Book lizenziert wurde. Zudem stelle das Überspielen der elektronischen Bücher auch auf andere E-Book-Reader kein Problem dar. [MSReader1-3]

Glassbook Reader

Am 14. März 2000 stellte Adobe unter großer öffentlicher Aufmerksamkeit Stephen Kings Erzählung „Riding the Bullet“ vor. Dieses war das erste E-Book eines Literatursuperstars, das nur als Datei und nicht als Papierversion in den Handel kam. Obwohl Adobe mit „ePaper“ ein komplettes E-Commerce-Paket für den Vertrieb von PDF-Texten im Programm hatte, entschied der Software-Hersteller sich für den angeblich sicheren E-Book-Handel mit Glassbook. Doch nur wenige Tage später tauchte das Werk unverschlüsselt im Web auf.

Um das E-Book zu lesen, musste man sich zunächst den passenden Reader herunterladen. Einigen Hackern gelang es, die für diesen Reader verschlüsselte Version zu knacken und das E-Book als PDF- und Textdatei zu verbreiten. [Glassbook-00]

Gemstar Rocket-eBook

Gemstar konzentriert sich neben dem Verkauf von Lesegeräten auf das Lizenzgeschäft mit den Verlagen, um auch vom Verkauf der E-Books zu profitieren. Gemstar verschlüsselt ein E-Book gerätespezifisch, so dass kein anderes Lesegerät die Datei anzeigen kann. Damit untersagt die Firma dem Kunden, ein E-Book zu verleihen. Mit weniger strikten Sicherheitsvorkehrungen hätte man die Verlage nicht davon überzeugen können, Lesematerial zur Verfügung zu stellen, hieß es. Die Konsumenten reagierten entsprechend zurückhaltend auf ein derart restriktives System. [Gemstar1+2]

2.3.4 DRM - Digital Rights Management

Prinzipiell beruht DRM auf der „Verpackung“ verschlüsselter digitaler Werke in einer Software-Schicht, welche die Copyright- und Nutzungsinformationen oder Verweise auf selbige enthält. Ein DRM-System kann als ein „Wächter“ über die im inneren enthaltenen Inhalte angesehen werden. Der Benutzer erhält nur über diese „Schutzschicht“ Zugriff auf den Inhalt. DRM-Systeme sollen dafür sorgen, dass Rechteinhaber genau die Vergütung erhalten, die sie für die Nutzung ihrer virtuellen Ware für angemessen halten.

Grundsätzlich können verschiedene Ziele von DRM-Systemen erkannt werden:

- Nicht-Konsumierbarkeit nicht bezahlter Inhalte gewährleisten (Vertraulichkeit)
- Schutz vor unautorisierter Veränderung der Inhalte (Integrität der Daten)
- Identifizierbarkeit urheberrechtlich geschützter Werke und deren Urheber (Authentizität der Daten)
- Verhinderung der Anfertigung illegaler Kopien

Diese Ziele werden durch kryptographische, organisatorische und spezielle Sicherheitsmechanismen realisiert, von denen sich allerdings viele in der Praxis als untauglich zum Schutz gegen ernsthafte Angriffe herausstellten.

„Insbesondere Verfahren, die nur das Kopieren von Inhalten verhindern sollen, sind deutlich unsicherer gegenüber Verfahren, die auf die Einschränkung illegaler Nutzungsmöglichkeiten zielen.“ Zu diesem Schluss kommt Dr. Hannes Federrath (TU Dresden) in seiner Studie „Scientific Evaluation of Digital Rights Management (DRM) Systems“. [FederrathDRM-02]

Sicherheitsziele und korrespondierende Techniken

Vertraulichkeit (Confidentiality)	Informationen sollen nur berechtigten Personen (oder Geräten) zur Verfügung stehen.	Verschlüsselungsverfahren, Öffnung der Inhalte mittels sicherer Hardware, Software-Verfahren
Authentizität der Daten (Data Authenticity)	Der Urheber einer Ware soll jederzeit ermittelt werden können.	Digitale Wasserzeichen
Integrität der Daten (Data Integrity)	Es soll sichergestellt werden, dass Daten nicht unautorisiert geändert werden.	Digitale Fingerabdrücke

Tabelle 2: Sicherheitsziele und korrespondierende Techniken

Weiterhin ist zu beachten, dass sich bei Multimedia-Daten die Datei verändern kann, ohne dass dadurch der Inhalt beeinflusst wird, zum Beispiel durch Kompression. Hier wäre es sinnvoll, dass zusätzlich zu einem Fingerabdruck auf Basis der Datei (also der Syntax), auch eine Art Fingerabdruck des Inhaltes (der Semantik) angefertigt wird. [DittWohl-02]

Die praktische Umsetzung dieses Ansatzes findet man zum Beispiel im „AudioID“-System, welches am Fraunhofer Institut für Integrierte Schaltungen (IIS) entwickelt wird. Mittels „AudioID“ wird dem Musikstück eine Art inhaltsbasierter Fingerabdruck entnommen. Mittels dieses Fingerabdruckes soll dann das einfache Auffinden von Audio- und Videodaten im Internet ermöglicht werden. [AudioID-02]

In der Praxis könnten solche Systeme einen wichtigen Beitrag vor allem für das Auffinden von illegal angebotenen Inhalten leisten.

Spezielle DRM-Techniken: Digitale Wasserzeichen und Fingerabdrücke

a) Digitale Wasserzeichen (Watermarks)

Digitale Wasserzeichen sind transparente Muster, welche mittels eines Einbettungsalgorithmus unter Verwendung eines geheimen Schlüssels in digitales Datenmaterial eingebracht werden. In den meisten Fällen sind digitale Wasserzeichen unsichtbar bzw. in Audio-Verfahren unhörbar. Es existieren allerdings auch Verfahren, die bewusst sichtbare Markierungen an dem zu schützenden Objekt anbringen und ebenfalls als Wasserzeichen bezeichnet werden.

Mittels digitaler Wasserzeichenverfahren sind der Nachweis der Urheberschaft von Datenmaterial und die Rückverfolgung illegaler Kopien zum Kopierer möglich.

In der Theorie müssen Wasserzeichentechniken bestimmten Anforderungen gerecht werden. Die wichtigsten Anforderungen sind Robustheit und Sicherheit:

Robustheit: Die in das Datenmaterial eingebrachte Wasserzeicheninformation muss gegenüber zufälligen Veränderungen des Datenmaterials oder Medienverarbeitungen widerstandsfähig sein.

Sicherheit: Es darf nicht möglich sein, die eingebrachte Information zu zerstören, aufzuspüren oder zu verfälschen, wobei ein möglicher Angreifer volle Kenntnis des Wasserzeichenverfahrens hat, ihm mindestens ein markiertes Datenmaterial vorliegt, ihm jedoch der geheime Schlüssel unbekannt ist. [DittWohl-02]

Der Sicherheitsaspekt kann in der Praxis jedoch nicht immer genügend erfüllt werden, wie verschiedene Angriffe zeigen.

Der „StirMark“-Angriff basiert auf der nicht-linearen Transformation eines Bildes, infolgedessen es nicht mehr möglich ist, die Position des eingebetteten Wasserzeichens zu erkennen.

Beim „Mosaic“-Angriff wird ein geschütztes Bild (z.B. ein Web Image) in mehrere Teile getrennt, welche erst durch den Browser wieder zusammengefügt werden. Dieses Vorgehen verhindert (zumindest bei simplen Mechanismen) die automatische Aufspürung geschützter Inhalte auf Web-Seiten. [FederrathDRM-02]

Im April 2001 wurden auf der Website *Cryptome* die Ergebnisse einer wissenschaftlichen Arbeit von Edward Felten (Professor an der Fakultät für Computer Science an der Universität Princeton) veröffentlicht, in der erfolgreiche Angriffe auf die von der Secure Digital Music Initiative (SDMI) entwickelten Schutzmechanismen für digitale Musik beschrieben werden. In der Arbeit wird unter anderem auf die Überwindung der Sicherheitsmechanismen des „Verance Watermark“ eingegangen, welcher bereits in kommerziellen Produkten zum Einsatz kommt (etwa in „SDMI Phase I“-Projekten und insbesondere als Kopierschutz von DVD-Audio). [SDMI-Hack-01]

Untersuchungen zur Qualität von digitalen Wasserzeichen und zum Einsatz innerhalb von Content-Management-Systemen werden vom H2O4M-Projekt durchgeführt. Konkretes Ziel dieses Projektes ist es, digitale Wasserzeichen zum Nachweis der Authentizität und Integrität von Multimediadokumenten zu klassifizieren und zu bewerten. Zusammenfassend wird festgestellt: „Heute existierende Verfahren sind anwendungsspezifisch und haben sehr uneinheitliche Verfahrensparameter und teilweise sehr geringe Sicherheitsniveaus. Es fehlen einheitliche Definitionen von Qualitätsparametern, um die Verfahren vergleichbar zu machen und den Anwendern eine Verfahrenstranzparenz zu geben.“ [H2O4M-02]

b) Digitale Fingerabdrücke (Fingerprints)

Digitale Fingerabdrücke sind das Ergebnis der Anwendung von Hash-Funktionen. Eine Hash-Funktion stellt eine Abbildung von Nachrichten beliebiger Länge auf Wörter fester Länge dar. Die so generierten Wörter (Fingerabdrücke) können zur Integritätsprüfung benutzt werden, d.h. um die Unverfälschtheit von Daten nachzuweisen, sofern die verwendete Hash-Funktion als sicher gelten kann.

Sichere Hash-Funktionen müssen zwei Anforderungen erfüllen:

Un-Umkehrbarkeit (Einweg-Hash-Funktion): Es darf nicht möglich sein (mit vernünftigem Aufwand) aus einem einmal erzeugten Hashwert die ursprüngliche Bytefolge zu rekonstruieren. werden

Kollisionsfreiheit: Es darf nicht möglich sein (mit vernünftigem Aufwand), zwei Bytefolgen zu konstruieren, die den gleichen Hashwert haben. Die kleinste Veränderung in einer Bytefolge muss zu einem anderen Hashwert führen. [WobstKRYPT-98]

Die bisher häufig eingesetzten Hash-Funktionen MD4 (Message Digest) und MD5 sind für heutige Sicherheitsanforderungen ungeeignet, da für diese (aufgrund ihrer kurzen Ausgabelänge von 128 Bit) bereits Kollisionen gefunden wurden. Aus den Erfahrungen mit den oben genannten Verfahren kann man schließen, dass bei Funktionen mit einer Ausgabelänge von 160 Bit wie RIPEMD-160 und SHA-1 (Secure Hash Algorithm) im Mittel etwa 2^{80} Versuche benötigt werden, um eine Kollision zu finden.

Um einen höheren Grad der Sicherheit zu erreichen, wurden mit dem Advanced Encryption Standard (AES) Hash-Funktionen mit höheren Ausgabelängen (SHA-256, SHA-384, SHA-512) eingeführt. Damit ist zumindest für einen bestimmten Zeitraum eine gewisse Sicherheit gewährleistet. Dennoch wird das Ziel für die Zukunft darin bestehen, Hash-Funktionen zu entwickeln (die basierend auf obigen Anforderungen) beweisbar sicher sind. [Berberich-01]

„Naive“ Sicherheitsmechanismen

So genannte „naive“ Sicherheitsmechanismen wie zum Beispiel Regional-Codes, inkompatible Formate und Medien und DRM-Systeme ohne Schutz vor Entfernung stellen selbst für Gelegenheitstäter keine große Behinderung dar, da sie sich mit geringem Aufwand umgehen lassen.

Beispiele:

- Regional-Code bei Video-DVDs: Umgehung durch Verwendung eines „region free“-Players oder durch Manipulation der Player-Hardware
- Inkompatible Audio-CD-Formate, welche das Abspielen auf PC-Hardware verhindern sollen (siehe „Kopierschutzverfahren für Audio-CDs“): Umgehung mittels einfacher Tricks („Filzstifthack“) oder Hacker-Tools
- Setzen des Copy-Bits zur Verhinderung der Anfertigung von Kopien mittels des SPDIF-Ausganges: Umgehung durch einfaches Zurücksetzen des Copy-Bits [FederrathDRM-02]

Microsoft DRM-2

Im Oktober 2001 veröffentlichte ein Programmierer unter dem Pseudonym "Beale Screamer" Dokumente und Beispielcode, mit denen sich Microsofts Digital Rights Management Version 2 („MS DRM-2“) knacken lässt. In den Dokumenten werden detailliert die Schwächen von Microsofts DRM erläutert und mit dem Programm „FreeMe“ wird eine Beispielimplementierung zur Verfügung gestellt, die geschützte Dateien im WMA-Format (Windows Media Audio) von vorhandenen Lizenzen „befreit“. [MSDRM-2-01]

Sichere DRM-Systeme

Zwar wird es wahrscheinlich nie ein System geben, dass zu 100% sicher ist, jedoch lassen sich (laut [FederrathDRM-02]) folgende Punkte zusammenfassen, die bei der Realisierung eines DRM-Systems berücksichtigt werden sollten um ein Höchstmass an Sicherheit (im Sinne des Content-Providers) zu erreichen:

- Die Öffnung der Inhalte innerhalb „vertrauenswürdiger“ Hardware-Systeme ist generell sicherer als die Verwendung von Software-Systemen.
- Sowohl Software- als auch Hardware-Systeme müssen ausforschungssicher gekapselt werden.
- Das DRM-Signal muss in solcher Weise mit dem Inhalt verbunden sein, dass dieser ohne das DRM-Signal nutzlos ist.
- Eine Öffnung des Inhaltes unter Umgehung des DRM-Signals darf nicht möglich sein.

TCPA (Trusted Computing Platform Alliance)

Die Umsetzung der oben genannten Punkte ist Bestandteil des Hardware-gestützten Konzepts der TCPA. Die von Compaq, HP, IBM, Intel und Microsoft gegründete Initiative hat sich als Ziel gesetzt, zukünftige PC-Generationen mit einem speziellen Sicherheits-Chip auszustatten und so eine „sichere Plattform“ zu schaffen. Der zentrale Baustein des Konzepts, der so genannte „Fritz-Chip“, soll verantwortlich sein für Benutzerauthentifizierung und –identifikation und Verschlüsselung. Das Konzept geht von der Annahme aus, dass der Rechner grundsätzlich eine unsichere Umgebung darstellt. Geschützte Inhalte lassen sich auf dem System erst dann wiedergeben, wenn der Chip die erfolgreiche Überprüfung der Umgebung ausgeführt hat. Diese Überprüfung beinhaltet den Test der Hardwarekomponenten, des BIOS, des Betriebssystems, der Treiber und Anwendungen. Wenn eine kritische Komponente kein gültiges Zertifikat vorweisen kann, verweigert der Chip dieser Komponente den Zugriff auf geschützte Inhalte. Microsofts Schutzsystem „Palladium“, welches Bestandteil der nächsten Windows-Version sein soll, soll als Software-Schnittstelle für die von der Hardware vorbereitete sichere Umgebung eingesetzt werden. Unter „Palladium“ werden sich DRM-geschützte Inhalte ohne ausdrückliche Erlaubnis des Rechteinhabers nicht mehr kopieren lassen, da das Betriebssystem schon den Versuch blockiert. Die Umgehung wird dadurch verhindert, dass nur ausdrücklich zugelassene Anwendungen auf die Inhalte zugreifen dürfen.

Die „LaGrande“-Technologie (LT) von Intel, welche die Realisierung von „vertrauenswürdiger“ Hardware im Sinne der TCPA ermöglicht, könnte bereits im kommenden Prescott-Kern des Pentium 4 enthalten sein. Diese Technik soll später auch in Bauelemente für Mobilgeräte und Server integriert werden.

Der Erfolg eines solchen Systems scheint aus wirtschaftlicher und technischer Sicht fraglich zu sein. Zum einen müsste das System vollkommen sicher gegen Angriffe sein und zum anderen werden sich immer Hersteller finden, die Systeme ohne integrierte DRM-Mechanismen anbieten. Aus wirtschaftlicher Sicht stellt sich die Frage, warum sich jemand ein System kaufen sollte, dass ihn derartig in seiner Freiheit einschränkt. [Himmelein-02] & [LaGrande-02] **Bitte unbedingt für TCPA www.trustedcomputing.org beachten (Nützel)**

2.4 Kompressionsverfahren für Audio- und Videodaten

Ohne das MP3-Kompressionsverfahren für Audiodaten (und das DivX-Kompressionsverfahren für Videodaten) wäre die derzeitige Entwicklung im Zusammenhang mit dem Konsumverhalten von virtuellen Waren nicht denkbar. Erst komprimierte Multimediadaten in hoher Qualität und dabei mit geringer Dateigröße haben zur Verbreitung von P2P-File-Sharing-Systemen geführt und damit die Unterhaltungsindustrie in Zugzwang gesetzt. Aus diesem Grund soll im Folgenden ein kurzer Überblick über die relevanten Codecs (Compressor/Decompressor) gegeben werden.

MP3 (MPEG1 Layer 3)

MPEG1 Layer 3 (Kurzform: MP3) wurde ursprünglich von der Motion Picture Expert Group für die Codierung der Tonspur von Filmen standardisiert. Das Verfahren basiert im Wesentlichen auf dem gezielten Weglassen unwichtiger, weil nicht hörbarer Informationen. Dadurch wird der Platzbedarf für das Abspeichern von Musik auf ca. ein Zehntel (bei einer Datenrate von 128KBit/s und einer Samplingrate von 44,1kHz) gesenkt, bei einer Qualität, die nach den Ergebnissen des Hörtests in der CT 3/2000, tatsächlich sehr nah an der CD liegt.

Die verbreitetsten MP3-Encoder kommen von Fraunhofer, Xing und LAME (ein freien MP3-Encoder mit einer eigenen Engine). Des weitern existiert eine ISO-Referenzimplementierung, auf welcher viele Free- und Shareware-Encoder basieren.

Die ersten MP3-Encoder kamen vom Institut für Integrierte Schaltungen der Fraunhofer Gesellschaft. Ein weiterer Vertreter, der Xing-Codec war anfangs massiv schneller als der Fraunhofer-Encoder und konnte außerdem als erster VBR (Variable Bitraten) anbieten. Für die MP3-ISO-Referenzimplementierung, die auch unter dem Namen "dist10" bekannt ist, sind die Quellen frei erhältlich. Auch LAME stammt ursprünglich von den ISO-Sourcen ab, ist aber praktisch eine totale Re-Implementierung [Leitner-00].

DivX

Bei DivX handelt es sich um ein Videokompressionsverfahren, für welches das AVI-Format als Container funktioniert, der üblicherweise eine Videospur und eine (oder mehrere) Audiospuren beinhaltet. Es existieren verschiedene Varianten des DivX-Codecs.

Beim „Original“ DivX;-) 3.11 Alpha handelt es sich um eine gepatchte Variante von Microsofts MPEG-4 V3. Dem Open-Source-Projekt OpenDivX dagegen liegt die MPEG-4-Referenz-Implementierung des EU-Projekts „MoMuSys“ zugrunde. DivX 4.12 war die letzte Open-Source Version von OpenDivX. Seit DivX5 wird der Codec kommerziell von DivXNetworks vertrieben. [Zota-02].

2.5 Vertriebswege virtueller Waren : Methoden des File-Sharings

Zwar haben File-Sharing-Systeme wie „Napster“ und „Gnutella“ (siehe Kapitel 2.5.2: „P2P-File-Sharing-Systeme“) das Tauschen von Dateien wesentlich vereinfacht, aber grundsätzlich ist dies auch ohne solche Systeme möglich: im Usenet, per FTP, per E-Mail, im IRC-Chat oder direkt mittels physischer Datenträger. Die wohl entscheidende Eigenschaft von P2P-File-Sharing-Systemen ist die bequeme Möglichkeit, die Ressourcen tausender verbundener Rechner auf einmal nach gewünschten Dateien zu durchsuchen. Begünstigt wurde der Erfolg von Tauschbörsen wie „Napster“ im Wesentlichen von vier Faktoren: durch die ständig steigende Zahl der Internet-Nutzer, die wachsende Bandbreite pro Nutzer, den wachsenden lokalen Speicherplatz und die Entwicklung von leistungsfähigen Kompressionsverfahren für Audio- und Videoinhalte (MP3, DivX).

Im Folgenden werden zunächst kurz einige der Technologien vorgestellt, welche schon in der Vergangenheit zum Dateiaustausch eingesetzt wurden (und die sich teilweise nach wie vor großer Beliebtheit erfreuen). Der Schwerpunkt wird aber bei der Erläuterung der Funktionsweise von Peer-to-Peer-File-Sharing-Systemen liegen. Es wird auch anhand von Fallbeispielen gezeigt, wie die Unterhaltungsindustrie versucht, ihre Inhalte vor illegaler Verbreitung zu schützen.

2.5.1 „Klassische“ Methoden des File-Sharings

Usenet

Das Usenet gehört neben dem WWW und E-Mail zu den wichtigsten Diensten des Internet. Es handelt sich hierbei um eine Sammlung von Diskussionsforen zu allen nur denkbaren Themen, welche mit einem eigenen Protokoll und einer verteilten Serverstruktur arbeitet. Entwickelt wurde das „UNIX User Network“ 1979 von Studenten der Duke University in North Carolina. Mittels des UUCP (Unix to Unix Copy Protocol) wurden Nachrichten in verschiedenen Gruppen ausgetauscht. Zum Austausch der Nachrichten stellten die einzelnen Rechner eine Verbindung zu den Servern her, luden eventuell vorhandene Nachrichten hoch und neue eingehende Nachrichten herunter. Im Jahr 1986 fand dann eine grundlegende Neuorganisation des Usenet statt.

Es entstand eine Hierarchie von Oberkategorien zu bestimmten Themenbereichen (z.B. EDV oder Kultur). Das UUCP wurde vom „Network News Transfer Protocol“ (NNTP) als Übertragungsprotokoll abgelöst. 1987 entstand die Alt-Hierarchie, eine offene Newsgroup-Hierarchie, in der praktisch jeder neue Gruppen erzeugen kann.

Mit zunehmender Bandbreite der Server und Clients wurde die alt.binaries-Hierarchie zum Austausch von Binärdateien eingerichtet. Neben MP3-Dateien werden hier auch digitalisierte Bücher, Software und Videos getauscht.

Da das juristische Vorgehen gegen Serverbetreiber wenig sinnvoll zu sein scheint (das Landgericht München hat im November 1999 im Fall CompuServe entschieden, dass Provider nicht für die über News-Server verteilten Inhalte verantwortlich gemacht werden können), werden im Usenet in der Regel individuelle Nutzer belangt. [MöllerTAU]

Internet Relay Chat (IRC)

Der IRC ist ein Textzeilen-Kommunikationsdienst, welcher 1988 vom Finnen Jarkko Oikarinen erfunden wurde. Innerhalb von themenspezifischen Gruppen (Channels), die jeder erzeugen kann, lassen sich Gespräche per Tastatur führen. Nachdem man eine Zeile geschrieben hat, schickt man diese dann in den Channel, man sieht also nicht in Echtzeit, was die anderen tippen, sondern nur das abgeschickte Ergebnis.

IRC-Server lassen sich wie Usenet-Server zu Netzen zusammenschalten, die ihre Kommunikation synchronisieren. Die Netzstruktur ist aber eine andere: IRC-Server sind in einem kettenartigen Netz organisiert. Nachrichten und Befehle werden nur an die jeweils benachbarten Server weitergeleitet (deshalb der Name „Relay“-Chat). Ein wesentlicher Unterschied zum Usenet ist, dass es nicht „das“ IRC-Netz gibt. Stattdessen gibt es verschiedene Netze wie EFNNet, Udernet oder Dalnet. Die Bedeutung dieses textbasierten Mediums beim Tausch von Dateien liegt darin begründet, dass es auf einfache Weise möglich ist, mit vielen verschiedenen Usern gleichzeitig zu kommunizieren, Tauschkontakte zu knüpfen und Links zu neuen „Warez“-Seiten bekannt zu geben. Die größten Channels auf einigen Servern sind folglich die, in denen MP3s, Software-Kopien, Filme oder andere illegale Waren angeboten werden. Da aber jederzeit neue Channels angelegt werden können, wäre die Löschung einzelner Channels wenig wirksam. [MöllerTAU-00]

FTP (File Transfer Protocol)

FTP ist ein Client-Server-Protokoll zum Austausch von Text- und Binärdaten, welches 1985 definiert wurde. Mehrere Benutzer haben dabei unterschiedliche Rechte, auf die Verzeichnisse eines Rechners zuzugreifen und dort Dateien hoch und herunter zu laden. FTP wurde und wird nicht nur zum Tausch privater und wissenschaftlicher Daten eingesetzt, sondern auch zum Tausch von raubkopierten Inhalten. Zunehmend an Relevanz für den Tausch großer Dateien (z.B. Filme) gewinnt der passive Modus (PASV), auch bekannt als FXP, mit welchem Server-zu-Server-Transfers möglich sind. Hat man einmal seine Daten in einem Archiv mit großer Bandbreite, kann man sie so schnell zu einem anderen Server übertragen. FTP-Server, welche die gewünschten Inhalte anbieten, kann man im IRC finden, wenn man die entsprechenden Channels besucht oder indem man entsprechende Suchmaschinen benutzt. Viele FTP-Server mit illegalem Material schalten ihren Zugang erst nach Prüfung des Benutzers frei oder fordern Uploads, bevor sie Downloads erlauben. [MöllerTAU]

WWW (World Wide Web)

Das WWW bietet verschiedene Möglichkeiten, um (illegale) Inhalte zu finden, zu tauschen bzw. herunter zu laden:

- „Warez“-Seiten (raubkopierte Software, Filme, MP3-Files usw. werden auf diesen Seiten angeboten), mit Links zu anderen „Warez“-Seiten oder direkten Downloadmöglichkeiten (FTP, HTTP, Speicherplatz-Anbieter). Auf einigen dieser Seiten ist auch die Bestellung von CDs mit dem gewünschten Inhalt möglich.
- Speicherplatz-Anbieter, die Backup-Platz für beliebige Dateien anbieten. Dabei sind bestimmte Bereiche meist nur dem Besitzer zugänglich (der natürlich sein Passwort anderen mitteilen kann), andere aber offen.
- Spezielle Suchmaschinen, die Ressourcen erfassen und indexieren.

„Warez“-Seiten werden in der Regel bei Freespace-Providern zur Verfügung gestellt. Es ist in allen Fällen üblich, Zugänge sofort zu löschen, sobald eine entsprechende juristische Beschwerde erfolgt. Kaum ein Freespace-Provider ist bereit, einzelne Dateien zu überprüfen. Auf einigen der oben erwähnten „Warez“-Seiten, ist neben dem Download von raubkopierten Dateien auch eine Bestellung von CDs mit dem gewünschten Inhalt möglich. [MöllerTAU]

E-Mail

Das Tauschen von Dateien per E-Mail erfolgt mittels „Attachments“, d.h. es werden (mit dem gleichen Kodierungsstandard wie im Usenet) Binärdateien an die eigentliche E-Mail angehängt. Die maximale Größe der anzuhängenden Dateien hängt von den Begrenzungen des jeweiligen Dienstanbieters ab. Der Provider legt fest, wie groß diese Dateien sein dürfen und wie viel er davon maximal speichert, eventuell gibt es auch Bandbreitenbeschränkungen.

Mailserver sind aber oft nicht für großen Datenverkehr ausgelegt, aber relativ kleine Dateien (Bilder, Texte) sind kein Problem. Der Tausch von MP3s oder gar ganzen Filmen ist aber nicht praktikabel. Eine wichtige Rolle beim Tauschen per Mail spielen Mailing-Listen. Diese Listen sind auf Verteiler-Servern gespeichert und beinhalten die E-Mail Adressen von verschiedenen Usern. Jede Mail, die an den Verteiler geschickt wird, geht an alle User in der Liste. Auf diese Weise kann auf neue „Warez“-Seiten aufmerksam gemacht werden oder können Suchanfragen weitergegeben werden. [MöllerTAU]

2.5.2 P2P-File-Sharing-Systeme

Ein kurzer Überblick über die Entwicklung von P2P

Der Begriff "Peer-to-Peer" bezieht sich auf eine spezielle Art von Netzwerken. Der direkte Datenaustausch zwischen „Gleichen“ (Peers) - mit vollständiger oder zumindest signifikanter Autonomie von zentralen Servern – wird als „Peer-to-Peer“ bezeichnet. Jeder Teilnehmer eines solchen Peer-to-Peer Netzwerkes ist gleichzeitig Client und Server, Anbieter und Konsument.

Das Internet (bzw. sein Vorläufer, das ARPANET: Advanced Research Project Agency Net) war ursprünglich im Wesentlichen ein Peer-to-Peer System. Lange Zeit vor der Wandlung zu einem Netzwerk, in welchem die Client/Server-Struktur dominiert, war es das Ziel des ARPANETs den Austausch von Ressourcen zwischen Universitäten in den USA zu ermöglichen und so die Kapazitäten besser auszunutzen. Bis 1994 benutzte das gesamte Internet ausschließlich das Modell der statischen IP-Adressen.

Alle vernetzten Rechner waren ständig online und somit jederzeit erreichbar. Die Änderung einer IP-Adresse galt als selten und ungewöhnlich. Mittels des DNS-Systems (welches für diese Umgebung entwickelt wurde) konnte so jedem Rechner ein fester Name zugeordnet werden.

Mit der Verbreitung der ersten Webbrowser und der damit verbundenen Zunahme der Anzahl der Internetbenutzer begannen die Internet Service Provider (ISP) mit der dynamischen Vergabe von IP-Adressen. Aufgrund der zeitlich begrenzten Gültigkeit dieser IP-Adressen war (und ist) eine Namenszuordnung mittels des DNS nicht mehr möglich. Den PC-Benutzern mit temporärem Zugang zum Internet war es somit nicht möglich, lokale Ressourcen direkt anderen Nutzern zur Verfügung zu stellen. P2P-Systeme umgehen die Benutzung des DNS, indem sie ihre eigenen „Name Spaces“ schaffen.

Das Finden von anderen aktiven Teilnehmern kann dabei auf verschiedene Weise erfolgen (siehe Abschnitt „Koordiniertes P2P vs. Dezentrales P2P“). Zwar benutzen einige Applikationen (z.B. ICQ oder SETI@Home) das P2P-Modell schon seit längerer Zeit (ICQ seit 1996), doch insgesamt war und ist das Internet von Client/Server-Strukturen beherrscht. [ShirkyP2P-00]

Im Jahr 1999 veröffentlichte der amerikanische Student Shawn Fanning ein MP3-File-Sharing-Programm namens „Napster“, welches das Prinzip „P2P“ innerhalb kürzester Zeit für Millionen von Internetbenutzern zugänglich machte. (Seit 1984 wurden ca. 23 Millionen Domain Namen registriert, bei „Napster“ wurde innerhalb von 16 Monaten die gleiche Anzahl von nicht-DNS Adressen angemeldet.)

Seit dem Erfolg von „Napster“ (und seinen zahlreichen Clonen) ist auch die allgemeine Öffentlichkeit auf P2P aufmerksam geworden. Das amerikanische Forschungsinstitut Gartner Group prognostiziert, dass bis zum Jahr 2007 die P2P-Netzwerke überwiegen werden und schon im Jahr 2003 ca. 30% der Unternehmen P2P-Applikationen einsetzen werden.

Eine vorsichtiger Schätzung von Frost & Sullivan hält es für wahrscheinlich, dass bis 2007 6 Millionen Unternehmen P2P-Technologien einsetzen werden. [FrasP2P-02] & [ShirkyP2P]

P2P scheint das Potential zu besitzen, um in Zukunft im Bereich der elektronischen Kommunikation eine wichtige Rolle zu spielen. Bei dem Austausch von Dateien ist es sogar möglich, dass P2P das dominierende Modell im Internet wird.

Koordiniertes P2P vs. Dezentrales P2P

Ausgehend von der im vorherigen Abschnitt genannten Definition für „Peer-to-Peer“ existieren zwei grundlegende Architektur-Modelle: „koordiniertes Peer-to-Peer“ und „dezentrales Peer-to-Peer“. Im Folgenden werden diese beiden Architekturen am Beispiel jeweils eines konkreten File-Sharing-Systems kurz vorgestellt und ihre Funktionsweise erläutert.

a) Koordinierte P2P-Systeme

Dieses Modell verwendet einen zentralen Verwaltungsserver, welcher ein Verzeichnis aller verbundenen „Peers“ und ihrer freigegebenen Ressourcen verwaltet. Wenn ein „Peer“ eine Suchanfrage startet, wird der Server kontaktiert, welcher eine Liste mit Dateien generiert, die dieser Anfrage entsprechen.

Dies geschieht, indem er die Suchanfrage mit den freigegebenen Ressourcen aller aktiven Benutzer vergleicht. Der „Peer“ kann nun aus dieser Liste die gewünschte Datei auswählen und sich direkt mit dem Rechner, der diese zur Verfügung stellt verbinden. Der eigentliche Datentransfer findet zwischen den „Peers“ statt.

Der bekannteste Vertreter für dieses Modell ist das MP3-File-Sharing-System „Napster“. Der Quellcode dieses Programms stand zwar ursprünglich nicht öffentlich zur Verfügung, aber da das Protokoll zur Datenübertragung gehackt wurde, war es schnell bis ins letzte Detail entschlüsselt und dokumentiert. Auf dieser Grundlage entstanden nicht nur etliche „Napster“-Client-Klone, sondern auch zwei offene Server-Projekte „OpenNap“ und „JNerve“. [MöllerTAU]

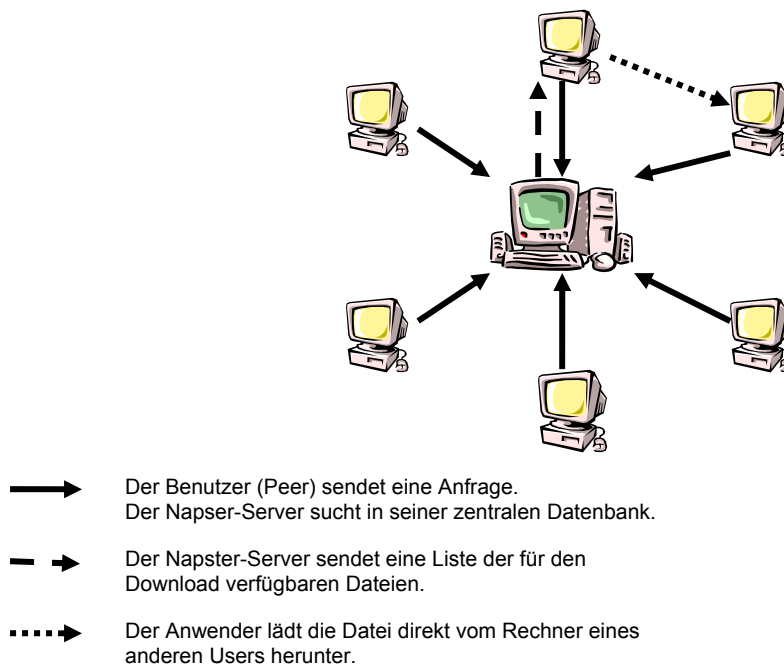


Abbildung 3: Koordiniertes P2P-System

b) Dezentrale P2P-Systeme

Das zweite Modell stützt sich nicht auf den Einsatz eines zentralen Verwaltungsservers, sondern im Wesentlichen auf die einzelnen Knoten („Peers“). Aber auch bei den verschiedenen Implementierungen dieses Modells existieren teilweise Server-artige Elemente. Diese können z.B. als Einstiegspunkte („Initial Connection Point“) funktionieren und so einen (bequemen) Einstieg in das P2P-Netzwerk ermöglichen oder als Verwaltungspunkte für Suchanfragen (siehe „FastTrack“). Die ersten Versionen von „Gnutella“ (neben „Napster“ ein weiterer File-Sharing-Pionier) stellten zum Beispiel beim Start eine Verbindung zu einem Server namens „findshit.gnutella.org“ her. Dieser Server diente als so genannter „Host Cache“, welcher eine Reihe gültiger IP-Adressen von „Gnutella“-Peers zurücklieferte. Prinzipiell ist es aber auch möglich, statt mit zentralen „Host Caches“, mit lokal gespeicherten Listen oder mit fest im Client implementierten (Einstiegs-)Adressen zu arbeiten.

Diese lokalen „Host Caches“ werden gefüllt, indem jeder „Servant“ (Bezeichnung für „Gnutella“-Clients) in regelmäßigen Abständen Ping-Pakete an seine unmittelbaren Nachbarn verschickt.

Wenn der Adressat des Ping-Pakets erreichbar ist, sendet er ein Pong-Paket (Antwort auf ein Ping-Paket) auf dem gleichen Weg zurück. Die Pong-Pakete enthalten neben der IP-Adresse des Absenders, dessen Port, sowie die Anzahl und Grösse seiner freigegebenen Dateien. Beim nächsten Programmstart werden die IP-Adressen im lokalen „Host Cache“ durchprobiert, um so erreichbare „Servants“ zu finden.

Suchanfragen laufen nach dem gleichen Schema ab. Jeder Rechner, der eine Suchanfrage empfängt, führt eine Überprüfung seiner lokalen Dateiliste durch und

liefert die Treffer über den gleichen Weg, den die Suchanfrage gegangen ist, zurück. Der eigentliche Datenaustausch findet dann direkt zwischen den beteiligten „Servants“ per HTTP statt. Sowohl Ping-Pakete, als auch Suchanfragen sind mit einem Zähler (TTL : „Time to live“) versehen, der bei jedem weiteren Versenden verringert wird. Bei Ablauf des Zählers „stirbt“ die Anfrage und wird nicht mehr weitergeleitet. [MöllerTAU]

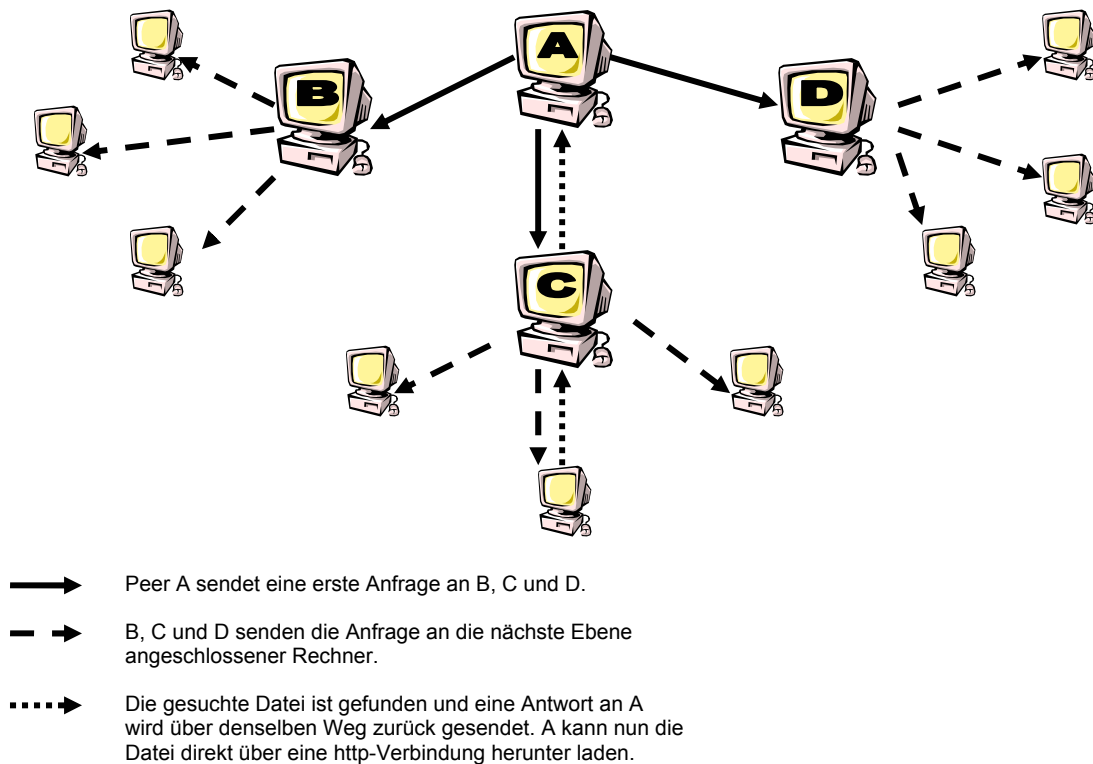


Abbildung 4: Dezentrales P2P-System

Benutzer mit temporärem Internetanschluss stellen für diese Art von Netzwerk ein besonderes Problem dar. Wenn eine Dial-Up-Verbindung unterbrochen wird, kann dabei gleichzeitig eine wichtige „Gnutella“-Verbindung getrennt werden.

Dies kann bei einem hohen Anteil von Dial-Up-Verbindungen zur Fragmentierung des Netzwerkes führen. Als Benutzer kann man außerdem praktisch nicht erkennen, ob man sich in einem abgespaltenen Subnet befindet oder nicht. Um dieses Problem zu lösen, werden so genannte Reflektoren eingesetzt. Die Reflektoren fungieren als temporäre, „Napster“-artige Index-Server im „Gnutella“-Netz, um vor allem Modemnutzer zu entlasten.

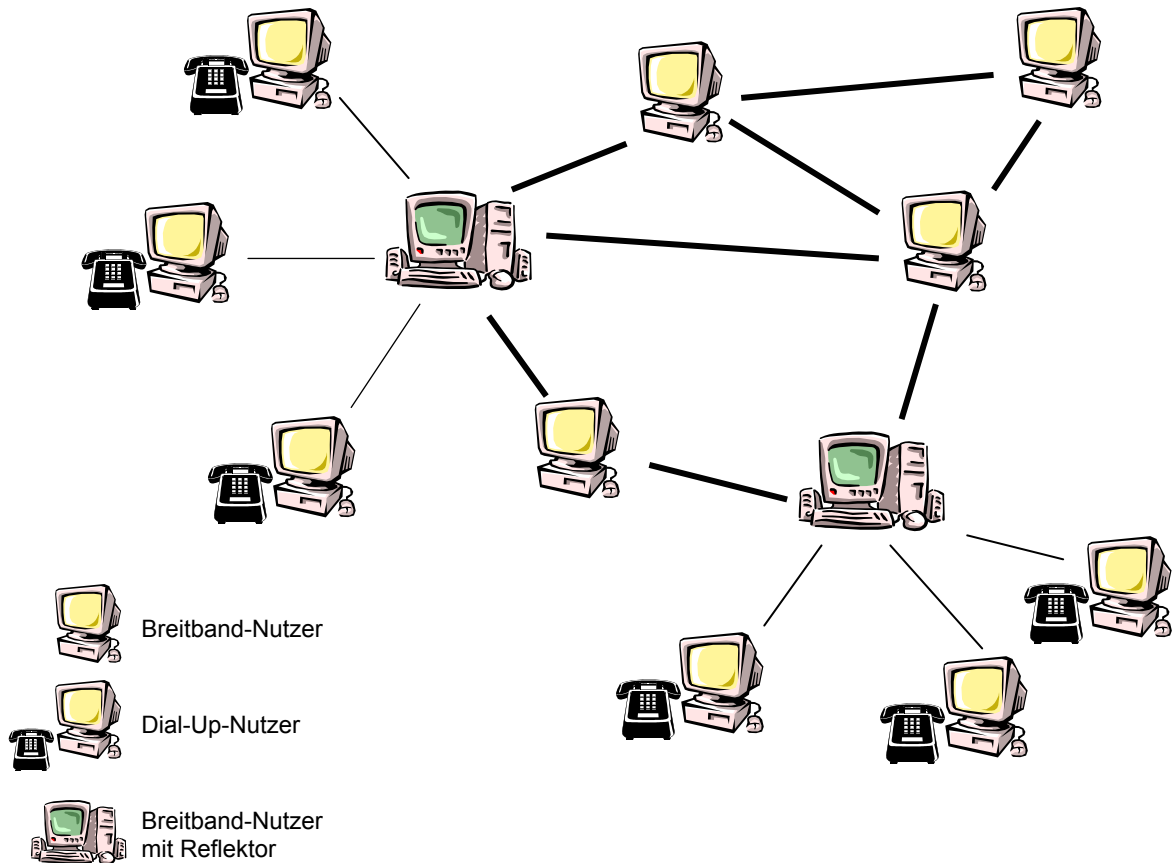


Abbildung 5: "Gnutella"-Netz mit Reflektoren

Mit „FastTrack“ existiert ein weiteres System, das die Aspekte eines dezentralen Netzes mit einer semi-zentralen Struktur verbindet. Die „FastTrack“-Client-Software stellt fest, ob ein Rechner hinsichtlich Performance und Bandbreite der Anbindung als Server geeignet ist und ernennt diesen bei erfüllten Bedingungen zum „Super-Peer“. Die Ernennung der Super-Peers erfolgt automatisch und dynamisch, je nach Last und alle anfallenden Suchanfragen werden von den Super-Peers verwaltet.

Besonders zu erwähnen im Zusammenhang mit dezentralen P2P-Systemen ist auch das Projekt „Freenet“. Dieses System hat sich als Ziel die Etablierung eines Netzwerkes gesetzt, welches zensurresistentes, anonymes und effizientes Abfragen und Tauschen von Informationen ermöglichen sollen. Um dies zu erreichen, werden neben der Dezentralisierung Prinzipien wie Redundanz, Verschlüsselung und dynamisches Routing angewendet. [ZoierP2P-01]

P2P-File-Sharing-Systeme und die Unterhaltungsindustrie : Fallbeispiele

Wie reagiert die Unterhaltungsindustrie auf die neue Qualität der Verbreitung von raubkopierten Inhalten mittels P2P-File-Sharing-Systemen?

Prinzipiell sind verschiedene Ansätze erkennbar: das gerichtliche Vorgehen gegen potentielle zentralisierte Elemente, die Übernahme und anschließende Kommerzialisierung bzw. Stilllegung von P2P-Systemen und natürlich der Einsatz von DRM-Systemen (siehe Abschnitt „DRM - Digital Rights Management“) um digitale Werke sicher zu „verpacken“ und illegal kopierte Inhalte aufzuspüren.

Für den Fall, dass diese Maßnahmen alleine nicht genügen sollten, bieten bereits einige Firmen spezielle Dienste an: Sie fluten gegen Bezahlung Peer-to-Peer-Netzwerke mit „gefakten“ Files, welche die Namen von Filmen oder Liedern tragen.

Anschließend folgen einige exemplarische Fälle, welche die Vorgehensweise der Unterhaltungsindustrie dokumentieren sollen.

a) Napster

Im Dezember 1999 hatte die RIAA, der Verband der amerikanischen Plattenindustrie, „Napster“ verklagt, da das Portal mit seiner illegalen MP3-Tauschbörse gegen Urheberrechte verstoße. Die „Napster“-Seite verteidigte sich mit dem Argument, dass das US-Bundesrecht das Anfertigen von Kopien für den persönlichen Gebrauch erlaube. „Napster“ biete die Titel nur für private Zwecke an und verletze deshalb keine Urheberrechte.

Im Rahmen des Verfahrens legten beide Seiten Studien darüber vor, wie sich die Existenz von „Napster“ auf die Konsumenten von CD's auswirkt. Die Ergebnisse dieser verschiedenen Studien sind jedoch teilweise gegensätzlich. Zum Beispiel ergab eine Corporation - Studie, dass 41 % der „Napster“-Nutzer weniger CDs kaufen, seit sie den Gratis-Tauschdienst nutzen. Im Gegensatz dazu stellte der DMA (Digital Media Association, Interessenverband von Web-Musik- und Web- Video- Unternehmen) fest, dass MP3 Songs aus dem Netz die Nutzer sogar zum Kauf konventioneller Tonträger animieren. Im Februar 2001 fing die Musikindustrie dann auch an, gegen „Napster“ - Alternativen vorzugehen und verschickte Briefe an Internet-Provider, die Verbindungen zu „Napster“ - Alternativen (wie „OpenNap“ - eine Open Source Entwicklung mit verteilten Servern) herstellten. Im Sommer 2001 musste „Napster“ dann sein Angebot einstellen. Nach der Schließung hatte sich „Napster“ vergeblich um die Einrichtung eines kostenpflichtigen Abonnements-Dienstes unter dem Einverständnis der führenden Musiklabels bemüht.

Ende September 2002 wurde eine Absichtserklärung über den Verkauf des Markennamens „Napster“, der Rechte an der Domain und der sonstigen noch existierenden Besitztümer an einen nicht näher bezeichneten Interessenten unterzeichnet.

Dem Medien-Konzern Bertelsmann, der seit dem Jahr 2000 etwa 80 Millionen US-Dollar in „Napster“ investiert haben soll, wurde die Übernahme der insolventen Tauschbörse durch einen Konkursrichter untersagt. [Napster]

b) *Gnutella*

„Gnutella“ wurde zunächst ohne große öffentliche Aufmerksamkeit unter Leitung von Justin Frankel, Chefentwickler des MP3-Players Winamp (Nullsoft), entwickelt. Das Ziel war die Schaffung einer Open-Source-Software zum Tauschen von Dateien. Das GNU im Namen deutet an, dass das Programm unter der GNU General Public License stehen sollte, welche die Quelltextoffenheit von Software sicherstellt.

Nach dem öffentlichen Aufruf zum Betatest des Programms am 13. März 2000, wurde in verschiedenen Medien über „Gnutella“ berichtet und das Unvermeidliche passierte. Seit Juni 2000 gehört Nullsoft zum Medienimperium von AOL. Das Programm verschwand von der Nullsoft-Seite und die offizielle Entwicklung wurde eingestellt. Der Quellcode wurde nie veröffentlicht. Aus unbekanntenen Quellen tauchten jedoch immer wieder neue Versionen auf, welche schwerwiegende Mängel beseitigten. Wichtige Details des „Gnutella“-Protokolls, welche ein anonymer Chatter in einer IRC-Sitzung mitteilte, waren dann schließlich die Grundlage für die Entwicklung zahlreicher „Gnutella“-Clone. Mittlerweile ist das Protokoll vollständig dokumentiert und AOL's Versuch „Gnutella“ totzuschweigen somit gescheitert. [MöllerTAU]

c) *Scour*

Im Juli 2000 verklagte die MPAA (Vereinigung der amerikanischen Filmindustrie) die Firma „Scour“. Betroffen war vor allem das P2P - „Scour-Exchange“-Netzwerk, das ebenfalls den Austausch von Multimedia-Dateien über das Internet ermöglichte. Der Firma wurden Urheberrechtsverletzung und Diebstahl vorgeworfen, da man sich komplette Spielfilme über „Scour Exchange“ herunterladen konnte. Die amerikanische Musikindustrie (vertreten durch die RIAA), in deren „Napster“-Prozess die MPAA als Nebenkläger auftrat, unterstützte die Klage.

Nachdem im September 2000 die Entlassung von zwei Dritteln der Belegschaft angekündigt wurde, stellte „Scour“ im Oktober 2000 dann einen Insolvenzantrag. Mit dieser Maßnahme wollte die Firma sich zunächst der von der Musik- und Filmindustrie angestregten Verfahren erwehren, da diese bis zum Abschluss des Insolvenzverfahrens ruhen mussten. Man erhoffte sich durch diesen Aufschub, geeignete Strategien zur Weiterführung der Firma entwickeln zu können und neue Kapitalgeber zu finden. Die Anstrengungen hatten jedoch nicht den erhofften Erfolg und so wurde dann im November 2000 die endgültige Schließung bekannt gegeben.

Die Firma CenterSpan Communications ersteigerte im Dezember 2000 für neun Millionen US-Dollar die „Scour“ - Technologie, mit dem Ziel eine „ernsthafte legale Konkurrenz“ zu „Napster“ & Co. zu schaffen. Im Vordergrund der neuen Version von „Scour Exchange“ steht vor allem der Schutz von Urheberrechten. [Scour]

Die Strategie des gerichtlichen Vorgehens gegen Betreiber von Tauschbörsen (mit zentralisierten Elementen) scheint auf den ersten Blick erfolgreich zu sein. Die Folgen der (Quasi) -Schließung bzw. versuchten Kommerzialisierung von P2P - Systemen wie „Napster“ können aber aus Sicht der Unterhaltungsindustrie wohl kaum positiv bewertet werden. Die Vielzahl der ehemaligen „Napster“-Nutzer weicht auf andere File-Sharing-Systeme aus, statt darauf zu warten, entsprechende Angebote von der Industrie präsentiert zu bekommen.

Der Kampf gegen die dezentralen P2P-Systeme dürfte weitaus schwieriger werden, als der gegen „Napster“, da es sich hierbei um echte P2P-Netze handelt, welche ohne zentrale Index-Server auskommen. Selbst, wenn ein entsprechender Anbieter schließen müsste, würde das dem Netzwerk selbst nicht schaden, denn die Anwender könnten weiterhin Dateien austauschen. Spätestens wenn diese Systeme dann noch mit dem Merkmal der Anonymität ausgestattet werden, scheint es fraglich, ob eine Kontrolle überhaupt noch möglich ist.

2.5.3 Die Angebote der Musikindustrie

Seit Ende des Jahres 2001 sind zumindest in den USA die lange angekündigten Online-Musicshops der großen Plattenfirmen im Netz. Dort lassen sich inzwischen Inhalte für 10 bis 25 US-Dollar im Monat bzw. 99 Cent pro Song downloaden. Während Bertelsmann, Warner Music und EMI über Musicnet eine Auswahl ihrer Inhalte zur Verfügung stellen, setzen Sony und Universal Music auf ein eigenes Lizenzmodell: Pressplay.

Der Erfolg lässt allerdings bisher auf sich warten. Hauptgrund hierfür sind die strikten Nutzungsregeln der Musikindustrie, da es für die Verbraucher zum Beispiel unverständlich ist, dass sie einen Song nach 30 Tagen nicht mehr abspielen können und erneut dafür bezahlen sollen. Auch die mangelnde Transparenz der derzeitigen DRM-Systeme und die mangelnde Attraktivität des Angebots können als Gründe erkannt werden. [MusicNet-02]

Pressplay versucht sich mit Zugeständnissen an die Verbraucher von MusicNet abzusetzen. So sollen die Kunden monatlich ein begrenztes Kontingent an Downloads auf CD brennen dürfen. Doch damit „wird das Rechtemanagement nicht mehr kontrollierbar“, fürchtet Thomas Kleesch (IBM Deutschland). Hier wird die Zwickmühle, in der die Musikindustrie steckt, erkennbar: Statt sie die Konsumenten mit zuviel Rechten aus, riskieren sie ein Versagen der (ohnehin nur bedingt wirksamen) Schutztechniken. Bei zu starker Reglementierung hingegen, werden die angebotenen Modelle von den Konsumenten nicht akzeptiert. [KrempfIA-01]

Ein kritischer Punkt ist auch die Preispolitik. Zum Beispiel scheint ein Preis von 99 Cent pro Song kaum geeignet zu sein, um Konsumenten zum Kauf zu veranlassen, da sich die Kosten für eine komplette CD (in schlechterer Qualität als die Original-Variante) auf ungefähr 15 Dollar zuzüglich Onlinekosten und Kosten für den Rohling belaufen (falls die Titel überhaupt brennbar sind).

2.5 Zusammenfassende Beurteilung

Nachdem die Bedeutung des (P2P)-File-Sharings anfangs von der Musikindustrie und später auch von der Filmindustrie nicht erkannt bzw. ignoriert wurde, konzentrieren sich die Bestrebungen der Medienunternehmen zurzeit auf folgende Maßnahmen:

- Ausschöpfung bestehender juristischer Möglichkeiten, Forderung nach neuen gesetzlichen Regelungen
- Anwendung von Kopierschutzmechanismen und DRM-Systemen
- Übernahme und Kommerzialisierung bestehender Modelle

Die kommerziellen File-Sharing-Angebote der Medienindustrie sind bisher keine echte Alternative zu den kostenlosen File-Sharing-Systemen. Statt potentiellen Kunden möglichst schnell attraktive Alternativen zum illegalen Erwerb und Tausch von Inhalten zu bieten, werden die Konsumenten durch zu hohe Preise, massive Einschränkungen bei der Benutzung und eine unzureichende Auswahl an Inhalten abgeschreckt (siehe Kapitel 2.5.3).

Parallel dazu versucht die Unterhaltungsindustrie verstärkt, ihre Produkte durch Kopierschutzmechanismen zu sichern und Raubkopierern und Anbietern von entsprechender Software auf rechtlichem Wege beizukommen. Doch genau diesen Versuchen können, nach Meinung verschiedener Experten und den Erfahrungen mit den gescheiterten Bemühungen der Industrie in der Vergangenheit (siehe Kapitel 2.3), keine großen Erfolgsaussichten vorhergesagt werden. Langfristig wird die Unterhaltungsindustrie wohl gezwungen sein, die Gegebenheiten der Online-Welt zu akzeptieren und dementsprechend angepasste Vermarktungs- und Vertriebsstrategien für virtuelle Waren zu entwickeln.

3 Konzeption

Die Konzeption der Client-Komponenten erfolgt nach den Prinzipien der objektorientierten Analyse (OOA) und des objektorientierten Designs (OOD) (siehe [OOA-00]). Das Ziel der Analyse ist die Erstellung eines Modells, welches die fachliche Lösung des zu realisierenden Systems bildet (bei Ausklammerung von Implementierungsdetails). Der Entwurf hat die Aufgabe, die spezifizierte Anwendung unter Beachtung der technischen Randbedingungen zu realisieren (auf einer höheren Abstraktionsebene als die Implementierung).

Die Vorgehensweise hierbei orientiert sich an dem in [OOSWE-98] dargestellten Vorgehensmodell, d.h. die Anwendungsentwicklung erfolgt:

- Anwendungsfallgetrieben: zur Erhebung der Anforderungen an das System werden Anwendungsfälle eingesetzt
- Komponentenzentriert: Trennung von Client- und Serverkomponenten
- Iterativ und inkrementell: Zerlegung der Entwicklung in mehrere gleichartige Schritte (Analyse-Design-Realisierungsabfolge), jede Iteration erzeugt ein neues Teilergebnis, wobei die Gesamtfunktionalität des Systems mit jedem Schritt wächst, die Teilergebnisse der Komponententeams müssen regelmäßig synchronisiert werden

Zur Notation werden unter anderem Elemente der UML (Unified Modelling Language: Sprache zur Spezifikation, Visualisierung, Konstruktion und Dokumentation von Modellen für Softwaresysteme, Geschäftsmodellen und anderen Modellen für Nicht-Softwaresysteme) eingesetzt. [OOSWE-98]

Zunächst erfolgt die allgemeine Beschreibung des Systems (Kapitel 3.1). Ausgehend von dieser Beschreibung werden alle relevanten Anwendungsfälle des Systems dargestellt (Kapitel 3.2). Da sich die Analyse des Systems auf die Client-Seite konzentriert, bietet sich die funktionalitätsorientierte Vorgehensweise an (siehe auch [Balzert-96]). Hierzu werden die Funktionsabläufe der Client-Komponenten skizziert und analysiert (Kapitel 3.3). Im Kapitel 3.4 werden dann die konkreten Klassen und Operationen (das Fachkonzept) abgeleitet, welche die Basis der Implementierung der Prototypen bilden (siehe Kapitel 4).

3.1 Das „Potato System“

Im „Potato System“ (PS) werden Konsumenten zu Vermittlungspartnern virtueller Waren. Dies geschieht, indem sie vom Kaufpreis, den ein Provider für das verkaufte Produkt bekommt, eine Vermittlungsprovision erhalten. Jeder Nutzer kann frei entscheiden, ob er für die virtuelle Ware bezahlen möchte, um Vertriebspartner zu werden und somit selbst von der Verbreitung der Ware zu profitieren. Er hat aber auch die Möglichkeit (PS)-Multimediaprodukte zu konsumieren und zu verteilen, ohne sich um das System zu kümmern.

Dieses Prinzip wird auch als Struktur-Marketing, Network-Marketing bzw. Multi-Level-Marketing (MLM) bezeichnet. Multi-Level-Marketing hat sich als eine erfolgreiche und effektive Art des Absatzes von realen Waren mittels Vergütung von unabhängigen Verkäufern erwiesen. Im Allgemeinen kann ein Verkäufer auf zwei verschiedene Arten profitieren: Zum einen werden Verkäufer an ihren eigenen Umsätzen für Güter und Dienstleistungen an Endverbraucher beteiligt. Zusätzlich können Verkäufer für diejenigen Umsätze vergütet werden, welche von ihnen angeworbenen Personen erzielen (bzw. Personen, die wiederum von diesen Personen angeworben wurden). Ein wesentliches Merkmal jedes legalen Multi-Level-Marketingplans ist, dass die Vergütung einzig durch den Verkauf von Waren und Dienstleistungen an den Endverbraucher erzielt wird und nicht etwa durch das Anwerben von anderen Verkäufern. [MLM]

In der vom „Potato System“ verwendeten Variante des Multi-Level-Marketings für virtuelle Waren, erhält ein Verkäufer nur Provisionen für die von ihm selbst gekauften und verteilten Dateien. Die Möglichkeit der indirekten Beteiligung (wie oben beschrieben), besteht ausschließlich für den Provider der virtuellen Ware. Der Provider wird an sämtlichen erzielten Umsätzen beteiligt.

Um die Funktionsweise des Systems zu veranschaulichen, soll im Folgenden ein Beispielszenario skizziert werden. Zum besseren Verständnis der Abläufe werden zunächst die Personen, welche mit dem System interagieren, vorgestellt.

- **Provider („Fred“):** Ein Provider ist der Schöpfer bzw. der Rechteinhaber einer virtuellen Ware. Fred kann eine reale Person (z.B. der Komponist eines Musikstückes) oder auch eine Rechtsperson (z.B. eine Firma) sein. Jeder Provider muss sich beim System identifizieren und haftet für seine Angaben bezüglich der Urheberschaft der von ihm angebotenen virtuellen Waren.
- **Konsument („Draco“):** Konsumenten zeichnen sich dadurch aus, dass sie die Inhalte konsumieren und/oder weiterverbreiten ohne zu bezahlen. Sie bleiben damit außerhalb des Systems und haben weder Kosten, noch können sie vom System profitieren.
- **Käufer („Ginny“ und „Harry“):** Indem sich ein Konsument für den Kauf der konsumierten virtuellen Ware entscheidet, wird er zum Käufer. Alle Käufer müssen dem System bekannt sein, um die Kauftransaktionen bzw. die Auszahlung der Provisionen korrekt vornehmen zu können. Wenn ein Käufer die gekaufte Ware weiterverbreitet (in der Hoffnung neue Käufer zu finden), wird er zum Vertriebspartner.
- **Betreiber des Accounting Servers („Bill“):** Bill ist der Betreiber des Accounting Servers (AS) und hat somit die zentrale Rolle innerhalb des Systems. Der Accounting Server speichert und verwaltet die Konten der angemeldeten Provider und Käufer und alle anderen Daten die für das „Potato System“ relevant sind.

Szenario:

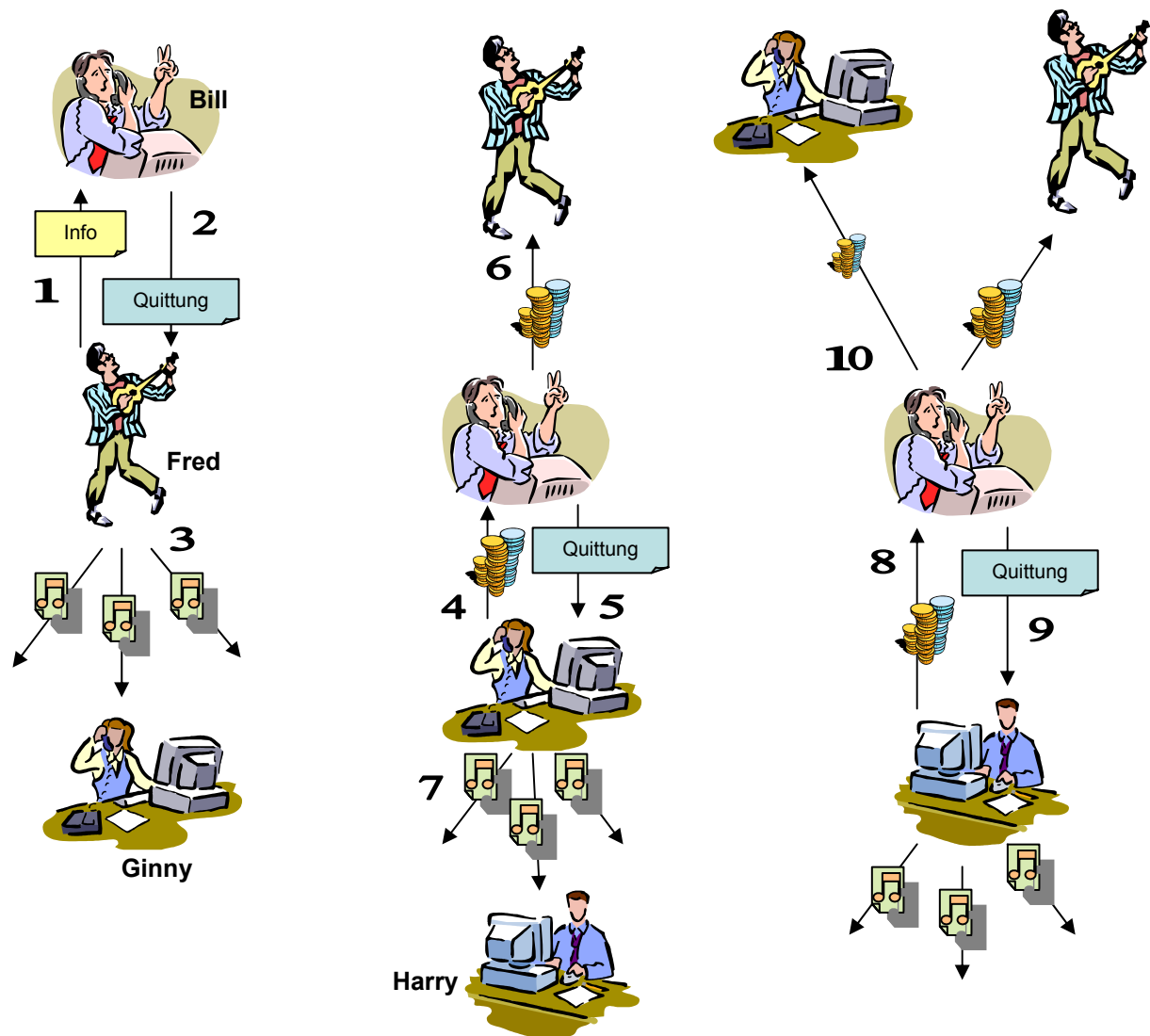


Abbildung 6: „Potato System“ Beispielszenario

Fred ist der Produzent bzw. Anbieter einer virtuellen Ware (z.B. ein Musikstück) und möchte diese kommerziell vertreiben. Es wird angenommen, dass das Musikstück als Datei im MP3-Format vorliegt. Im ersten Schritt muss Fred sich selbst und seine Datei beim Accounting-Server (AS) registrieren (1). Nachdem Fred sich durch persönliche Angaben identifiziert hat, wird für ihn ein Autoren-Account auf dem AS angelegt. Auch über die zu registrierende Datei müssen Angaben gemacht werden, so dass diese auch eindeutig identifiziert werden kann (z.B. charakteristische Merkmale der Datei und des Inhaltes der Datei). Des Weiteren wird auch ein Hash-Wert (Fingerprint) der Datei auf dem AS hinterlegt, um jederzeit eine Integritäts- bzw. Authentizitätsprüfung durchführen zu können. Fred gibt bei der Registrierung auch an, wie viel die jeweilige Datei kostet und wie hoch die Provision sein soll.

Als Bestätigung für die Registrierung seiner Datei erhält Fred einen elektronischen Beleg, welcher mit der Datei verbunden wird (2). Fred kann jetzt seine Datei - zusammen mit dem Registrierungsbeleg - auf allen möglichen Wegen verteilen (zum Download anbieten, in P2P-Systemen zur Verfügung stellen, auf CD etc.) (3). Ginny erhält Freds Datei und kann diese ohne Einschränkung nutzen und auch weiterverteilen. Sie kann aber auch die Information über die Datei bei Bill abrufen (indem sie den Registrierungsbeleg an Bill sendet) und die Datei kaufen. Um eine Datei zu kaufen, muss Ginny einen User-Account für die Auszahlung der Provisionen bei Bill anlegen. Sie muss hierzu minimale Angaben machen. Nachdem Ginny die Informationen zur Datei (Merkmale der Datei, Preis usw.) überprüft hat und der Integritätscheck positiv war (d.h. die Datei befindet sich im Originalzustand), kann der Kaufvorgang durchgeführt werden. Ginny kann nun mittels eines geeigneten Payment-Systems bezahlen (4). Der Kaufvorgang wird in der Datenbank des AS (im Account von Fred) gespeichert und Fred erhält den gesamten Betrag auf seinem Account gutgeschrieben (5). Als Ergebnis der erfolgten Bezahlung erhält Ginny einen neuen Beleg (Quittung), welcher in gleicher Weise an die Datei angefügt wird (6). Ginny kann nun die Datei (mit dem neuen Beleg) weiterverteilen (7). Harry bekommt Ginnys Datei (direkt oder indirekt) und verhält sich analog zu Ginny (8). Die Quittung, welche Harry zusammen mit der Datei von Ginny erhalten hat, wird an Bill gesendet und Harry erhält nach der Bezahlung eine neue Quittung (9). Ginny bekommt auf ihrem Account einen Teil der Bezahlung gutgeschrieben (10). Der Rest des Betrages, abzüglich der Gebühren und Unkosten, geht an Fred (10).

Registrierungsbeleg im Dateinamen: Transaktionsnummern

Das Prinzip besteht darin, den Quittungsbeleg direkt im Namen der registrierten Datei zu speichern. Dies hat den Vorteil, dass kein proprietäres Dateiformat nötig ist, um die Meta-Daten zusammen mit der Datei zu verteilen. Die Transaktionsnummer kann mit jedem beliebigen Dateityp verbunden werden und der Nutzer wird in seinem normalen Konsumverhalten nicht eingeschränkt. Die im Dateinamen abgelegte Information ist eine eindeutige Transaktionsnummer (TAN), mittels derer man die eigentlichen Informationen über die Datei und deren Inhalt beim AS abrufen kann. Lautet die Transaktionsnummer 111, so wird aus MY_SONG.MP3 der neue Dateiname in der Form MY_SONG4fo111.MP3 generiert. Wichtig für die Verteilung der Datei (besonders in P2P-Systemen) ist die Beibehaltung der Information über den Inhalt der Datei im Dateinamen (MY_SONG).

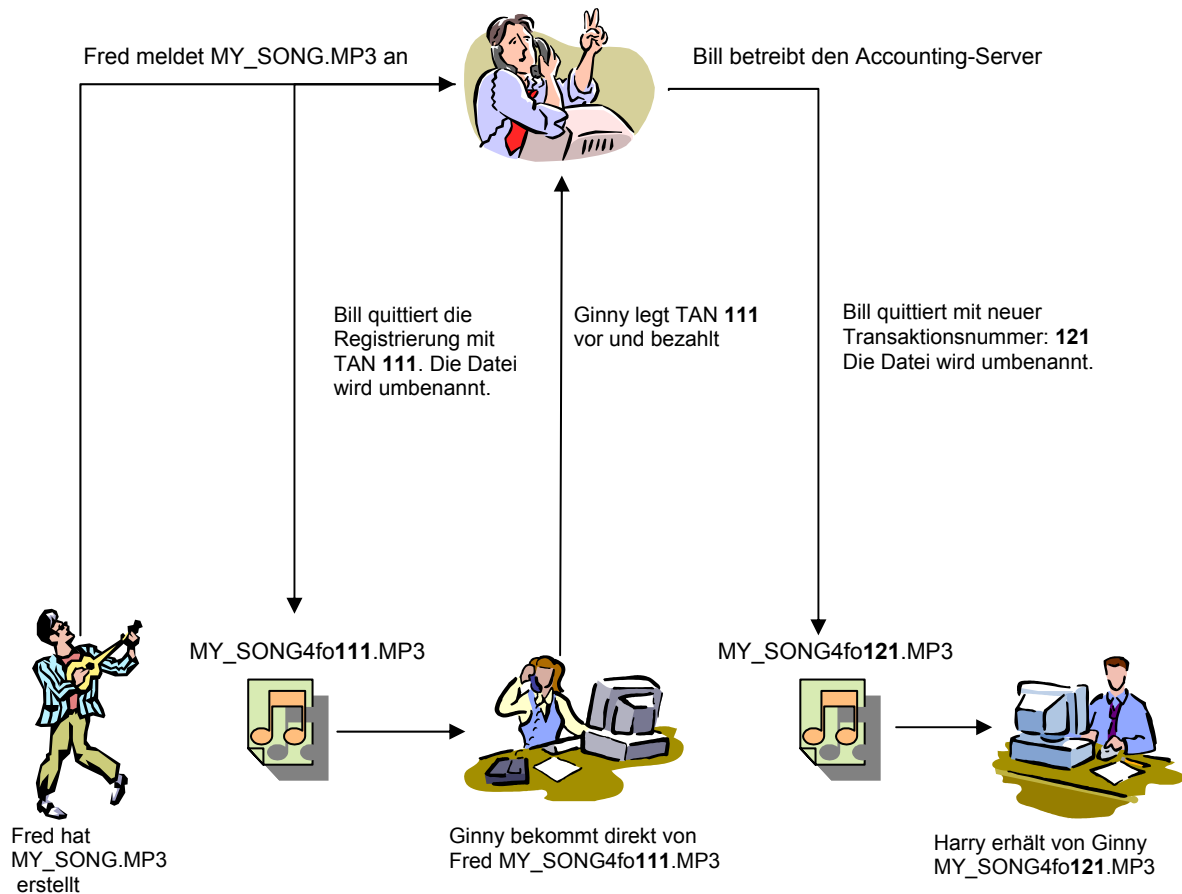


Abbildung 7: Transaktionsnummern und Dateinamen im „Potato System“

Der Aufbau der Transaktionsnummern

Die Transaktionsnummer ist die Quittung für einen erfolgreichen Registrierungs- bzw. Kaufvorgang. Sie wird in den Dateinamen der registrierten bzw. gekauften Datei eingefügt.

Vor der Registrierung hat der Dateiname die folgende Struktur:

original_name = root dot extension

also zum Beispiel: *original_name = MYSONG.MP3*, mit:

root=MYSONG

dot=.

extension = MP3

Nach der Registrierung bzw. dem Abschluss eines Kaufvorgangs erhält der Dateiname die folgende Struktur:

new_name = root delimiter TAN dot extension

also zum Beispiel: *new_name = MYSONG4fo111.MP3*, mit:

delimiter=4fo

TAN = 111 (Beispiel)

Transaktionsnummern können bis zu 16 Ziffern haben und bestehen aus zwei Teilen. Im vorderen Teil verbirgt sich eine Kundennummer, während im hinteren Teil eine kundenspezifische Transaktionsnummer steht.

Beispiel: TAN = 21243

Die erste Ziffer („2“) gibt die Anzahl der Stellen der Kundennummer an. Entsprechend dieser Angabe folgt die Kundennummer („12“). Die folgende Zahl gibt die Anzahl der vom Benutzer durchgeführten Transaktionen an („43“) [4foOrg-02].

Der genaue Algorithmus zur Generierung von eindeutigen Transaktionsnummern wird in [Krauß-02] beschrieben.

Erweiterungen für das „Potato System“: Der Community-Ansatz

Ein wichtiger Punkt, der das PS von reinen MLM-Systemen unterscheidet, ist der Community-Ansatz, d.h. Käufer werden nicht nur finanziell profitieren, sondern auch durch spezielle Angebote. Hierbei ist ein breites Spektrum denkbar: Rabatte bei Konzertkarten oder Fan-Artikeln, Zugang zu limitierten Auflagen von CDs, Fan- und Promotion-Artikeln, Backstage-Pässe oder Treffen mit den Künstlern. Des Weiteren wäre eine namentliche Nennung in Verbindung mit einem entsprechenden Käuferstatus (vergleichbar mit dem Ebay-System) auf der Internet-Seite des „Potato Systems“ denkbar, sehr aktive Käufer könnten so als „offizielle Power-Supporter“ innerhalb der Community erkennbar sein und auch als solche in entsprechenden Foren auftreten.

Weiter wäre es denkbar, ein System mit Käuferempfehlungen einzusetzen, d.h. wenn Ginny eine Datei kauft („FileX“), wird sie darüber informiert, welche Kunden die gleiche Datei (und andere Dateien) gekauft haben. Der AS sendet Ginny dazu eine Liste derjenigen Kunden, deren Liste von gekauften Dateien eine bestimmte Schnittmenge mit Ginnys gekauften Dateien aufweist. Ginny kann auf diese Weise in Kontakt mit Personen ähnlichen Musikgeschmacks treten, um Dateien zu kaufen oder um selbst Dateien zu verkaufen.

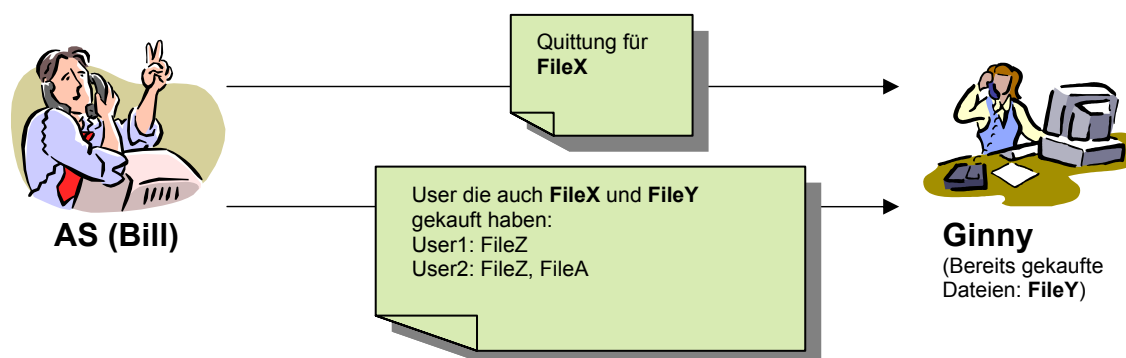


Abbildung 8: Käuferempfehlungssystem

Besonders effektiv wäre diese Art der Käuferempfehlung in Verbindung mit einem P2P-File-Sharing-System im Stil von „Gnutella“. Eine Vermittlung durch Bill wäre hierbei nicht mehr notwendig, da bei einer Suche nach einer bestimmten Datei (oder einer Menge von Dateien) alle erreichbaren Nutzer ermittelt werden könnten, die auch diese Dateien (und andere Dateien mit ähnlichem Inhalt) besitzen. Ginny könnte dann die potentiell ähnlichen Dateien herunterladen oder mit dem entsprechenden Nutzer über ein Chat-System in Kontakt treten.

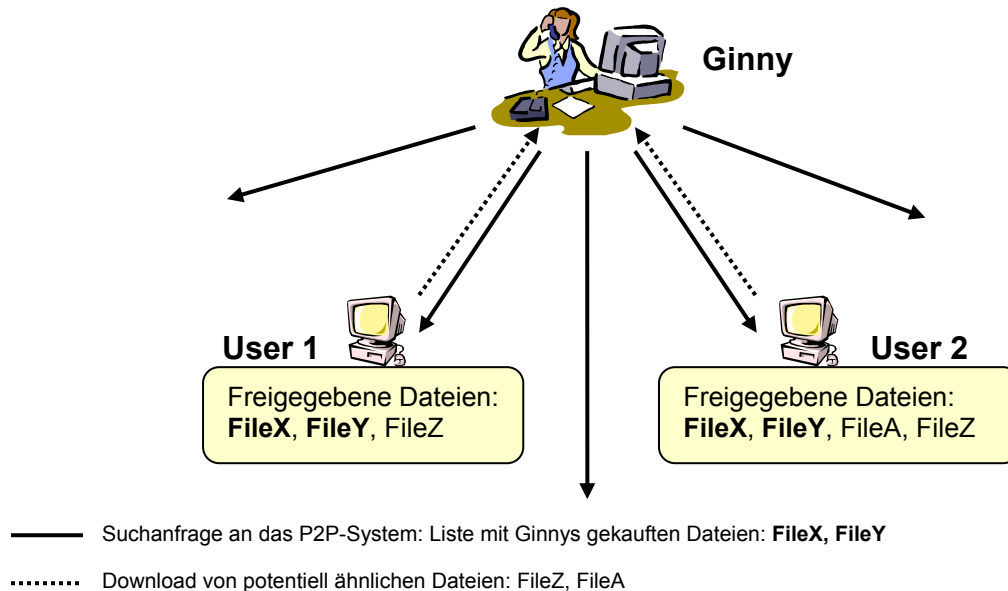


Abbildung 9: Käuferempfehlungssystem mit P2P

Dezentraler Ansatz vs. zentraler Ansatz

Grundsätzlich muss das „Potato System“ folgende Aufgaben erfüllen, um den Interessen der Urheber (Wahrung der Urheberrechte und damit die Möglichkeit der finanziellen Verwertbarkeit der virtuellen Ware) und der Konsumenten (Freiheit beim Konsum plus die Möglichkeit selbst vom Vertrieb der virtuellen Ware zu profitieren) gerecht zu werden:

- Eindeutige Feststellung der **Identität der Provider**: Jeder Provider muss eindeutig identifizierbar sein, um einer eventuelle Verletzung von Urheberrechten Dritter vorzubeugen bzw. um den Verletzer der Urheberrechte haftbar machen zu können.
- Überprüfung der **Authentizität der virtuellen Waren**: Es muss prüfbar sein, ob der Provider tatsächlich der Urheber der von ihm angebotenen Ware ist.
- Überprüfung der **Integrität der virtuellen Waren**: Es muss prüfbar sein, ob eine registrierte virtuelle Ware während der Verbreitung verändert wurde.

Bei der Realisierbarkeit der genannten Aufgaben spielt die Art der verwendeten Netzwerkstruktur eine wesentliche Rolle. Die eine Möglichkeit wäre ein dezentraler Ansatz (P2P-Prinzip), die andere ein zentraler Ansatz (Client-Server-Prinzip).

Da in einem P2P-System jeder Teilnehmer als Provider auftreten könnte, wäre es sehr einfach, auch fremde Inhalte als die eigenen auszugeben. Ein potentieller Käufer könnte einen „echten“ Provider nicht ohne weiteres von einem „unechten“ Provider unterscheiden (seine Zahlungsbereitschaft würde demzufolge äußerst gering sein). Die gleiche Problemstellung würde sich ergeben, wenn mehrere (unkoordinierte) Client-Server-Systeme existieren würden.

Bei einem P2P-System würde außerdem ein Mehraufwand beim Transport und Konsum der virtuellen Ware entstehen, da alle Informationen über die Datei und die durchgeführten Transaktionen zusammen mit der Datei verbreitet werden müssten. Des Weiteren wäre die Abwicklung des Zahlungsverkehrs zwischen den einzelnen P2P-Usern (besonders unter den Aspekten Sicherheit und Effizienz) nicht praktikabel. Zur weiteren Verbreitung der einmal registrierten und auf Urheberschaft geprüften Dateien können natürlich auch P2P-Systeme eingesetzt werden, ebenso wie alle anderen erdenklichen Vertriebswege für virtuelle Waren.

3.2 Anwendungsfälle

Im folgenden Kapitel wird mittels verschiedener Anwendungsfälle das Systemverhalten des „Potato Systems“ aus Anwendersicht dargestellt. Anwendungsfälle (Use Cases) sind ein Bestandteil der UML. Sie beschreiben, wie verschiedene Akteure auf ein System einwirken und mit dem System interagieren, d.h. sie haben die Aufgabe zu verdeutlichen, was das System leisten soll, aber nicht wie das System etwas leisten soll. Die in Kapitel 3.1 bereits zur Beschreibung des „Potato Systems“ verwendeten Personen („Fred“: Content-Provider, „Ginny“ und „Harry“: Käufer, „Bill“: Betreiber des Accounting-Servers, Konsument („Draco“): Person, welche die Ware benutzt ohne sie zu bezahlen) werden auch in diesem Kapitel als Akteure verwendet.

3.2.1 Als Provider registrieren

Akteure	Fred, Bill
Ablaufbeschreibung	<ol style="list-style-type: none"> 1) Fred möchte seine digitalen Inhalte über das „Potato System“ vertreiben. 2) Er gibt seine Benutzerdaten an und erhält nach der erfolgreichen Überprüfung seiner Identität einen Provider-Account mit seinem gewünschten Login auf dem AS. 3) Für Fred wird im System von Bill ein Account angelegt.
Vorbedingung(en)	Fred muss eindeutig identifizierbar sein.
Relevante Eingabedaten	Die Benutzerdaten von Fred
Ausnahmen	Das Login ist bereits vergeben oder die Benutzerdaten von Fred sind nicht ausreichend bzw. Freds Identität lässt sich nicht zweifelsfrei feststellen.
Nachbedingung	Fred ist als gültiger Anbieter von virtueller Ware im System registriert.

Tabelle 3: Anwendungsfall „Als Provider registrieren“

3.2.2 Als Vertriebspartner registrieren

Akteure	Ginny, Bill
Ablaufbeschreibung	<ol style="list-style-type: none"> 1) Ginny will sich beim PS registrieren, um ein Vertriebspartner zu werden. 2) Für Ginny wird im System von Bill ein Account mit ihrem gewünschten Login eingerichtet.
Vorbedingung(en)	Ginny ist noch nicht registriert.
Relevante Eingabedaten	Die Benutzerdaten von Ginny
Ausnahmen	Das Login ist bereits vergeben oder die Benutzerdaten sind nicht ausreichend.
Nachbedingung	Ginny ist als Vertriebspartner im System registriert.

Tabelle 4: Anwendungsfall „Als Vertriebspartner registrieren“

3.2.3 Eine Datei registrieren

Akteure	Fred, Bill
Ablaufbeschreibung	<ol style="list-style-type: none"> 1) Fred will seine virtuelle Ware über das PS vertreiben. 2) Fred übermittelt die Informationen über die zu registrierende Datei an Bill. 3) Bill überprüft, ob bereits eine Datei mit ähnlichem oder gleichem Inhalt registriert wurde (d.h. ob Fred tatsächlich die Rechte an dem angebotenen Inhalt besitzt) und legt bei positivem Ergebnis einen entsprechenden Datensatz zur Datei an. 4) Bill schickt einen elektronischen Registrierungsbeleg zurück an Fred. 5) Die elektronische Quittung wird an die registrierte Datei angefügt.
Vorbedingung(en)	Fred muss beim PS als Provider registriert sein. Er muss im Besitz der Rechte an der angebotenen Ware sein.
Relevante Eingabedaten	Dateispezifische Informationen und inhaltspezifische Informationen, Preis und Provision, optional einen Link zur Datei, optional die Datei selbst um den AS als Plattform für Dateidownload „on-demand“ zu nutzen. (siehe Kapitel 3.2.6)
Ausnahmen	Fred ist nicht der Rechteinhaber des angebotenen Inhaltes.
Nachbedingung	Die registrierte Datei wird eindeutig dem Account von Fred zugeordnet. Die Datei selbst wird mit der Quittung versehen.

Tabelle 5: Anwendungsfall „Eine Datei registrieren“

3.2.4 Eine Datei überprüfen

Akteure	Ginny, Bill
Ablaufbeschreibung	<ol style="list-style-type: none"> 1) Ginny möchte sich vor einem eventuellen Kauf die Daten einer registrierten Datei anzeigen lassen bzw. die Datei hinsichtlich ihrer Unversehrtheit und Herkunft prüfen lassen. 2) Ginny schickt den Namen (bzw. die Quittung) der zu überprüfenden Datei und andere zur Überprüfung verwendbare Merkmale der Datei an Bill. 3) Bill überprüft die Datei auf Authentizität und Integrität und zeigt alle relevanten Daten über die Datei bzw. das Ergebnis der Überprüfung an. 4) Ginny kann nun entscheiden, ob sie die Datei kaufen möchte.
Vorbedingung(en)	Ginny muss bei Bill registriert sein.
Relevante Eingabedaten	Der Name der Datei bzw. die elektronische Quittung, Vergleichsmerkmale der Datei
Ausnahmen	Ungültige Quittung, ungültige (manipulierte) Datei
Nachbedingung	Die Datei ist gültig (im Sinne einer positiven Integritäts- und Authentizitätsprüfung). Ginny erhält von Bill alle relevanten Daten über die gewünschte Datei.

Tabelle 6: Anwendungsfall „Eine Datei überprüfen“

3.2.5 Eine Datei kaufen

Akteure	Ginny, Bill
Ablaufbeschreibung	<ol style="list-style-type: none"> 1) Ginny möchte eine geprüfte virtuelle Ware, die bei Bill registriert ist, kaufen. 2) Ginny sendet eine Kaufanfrage an den Bill und bezahlt. 3) Bill legt für die Transaktion einen eindeutig zuweisbaren Datensatz auf dem AS an und schickt eine entsprechende Quittung an Ginny zurück. 4) Die elektronische Quittung wird an die gekaufte Datei angefügt. 5) Fred erhält seine vereinbarte Provision.
Vorbedingung(en)	Ginny muss bei Bill registriert sein, die Quittung (TAN) muss gültig sein, die Authentizitäts- und Integritätsprüfung muss erfolgreich sein
Relevante Eingabedaten	-
Ausnahmen	-
Nachbedingung	Ginny ist im Besitz einer elektronischen Quittung für den Kauf des Produktes, Fred erhält seine vereinbarte Provision.

Tabelle 7: Anwendungsfall „Eine Datei kaufen“

3.2.6 Dateien verteilen

Bei der Verteilung der Dateien muss zwischen verschiedenen Anwendungsfällen unterschieden werden. Bei der ersten Variante („File Sharing“) wird davon ausgegangen, dass Ginny bereits im Besitz einer registrierten Datei ist und diese nun weiter verteilen möchte. Sie hat also zuerst die Datei erhalten (z.B. durch ein P2P-File-Sharing-System) und sich dann für den Kauf entschieden. Das Verteilen der virtuellen Ware erfolgt unabhängig vom System. Im Gegensatz dazu wird bei den anderen beiden Varianten (Dateidownload „on-demand“) davon ausgegangen, dass Ginny die gewünschte Datei anfordert („on-demand“) nachdem sie auf den Inhalt aufmerksam geworden ist (z.B. durch Radio- oder TV-Sendungen) oder die Datei direkt von Fred (Web-Seite) bezieht. Bei den beiden zuletzt genannten Anwendungsfällen bezahlt Ginny für die Datei, bevor sie ihren Besitz gelangt.

File Sharing

Akteure	Ginny, Harry, Draco, Bill
Ablaufbeschreibung	<ol style="list-style-type: none"> 1) Ginny verteilt die Datei, welche sie gekauft hat (inklusive ihrer zugehörigen Quittung). Harry und Draco bekommen die Datei von Ginny. 2) Harry kauft die Datei (siehe 3.2.5) und verteilt sie weiter. Draco entscheidet sich, die Ware zu konsumieren, ohne dafür zu bezahlen. Er verteilt die Datei auch weiter. 3) Fred und Ginny erhalten Provisionen für die von Harry gekaufte Datei.
Vorbedingung(en)	Ginny hat die Datei bei Bill gekauft
Relevante Eingabedaten	-
Ausnahmen	-
Nachbedingung	Harry und Draco erhalten eine registrierte Datei von Ginny. Harry erhält eine eigene Quittung für die gekaufte Datei. Fred und Ginny erhalten Provisionen für Harrys Kauf.

Tabelle 8: Anwendungsfall „File Sharing“

Dateidownload „on-demand“

Im vorangegangenen Anwendungsfall wurde davon ausgegangen, dass Ginny schon im Besitz der Datei ist, bevor sie sich zu einem möglichen Kauf entscheidet. In den folgenden Anwendungsfällen wird davon ausgegangen, dass Ginny noch nicht im Besitz der Datei ist, sondern auf eine andere Weise auf die Datei aufmerksam wurde (z.B. ein Lied im Radio hört oder durch die Web-Seite des Künstlers) und sich dann entscheidet die Datei zu kaufen. Ein mögliches Szenario für diesen Anwendungsfall könnte wie folgt aussehen:

Ginny hört einen Song im Radio (Fred ist in diesem Fall der Radiosender und hat die Songs der aktuellen Playlist bei Bill registriert und auf dem AS gespeichert) und möchte den Song als MP3-Datei erwerben. Sie schickt eine SMS mit ihrer Email-Adresse an eine spezielle Nummer. Anhand der SMS und der aktuellen Playlist wird eine Email generiert, welche einen Link zur gewünschten Musik-Datei enthält. (Alternativ könnte auch eine Antwort-SMS generiert werden, Ginny bräuchte in diesem Fall keine Email-Adresse anzugeben.) Wenn Ginny den Link öffnet, erhält sie alle nötigen Informationen über die Datei und den Bezahlvorgang. Wenn Ginny noch keinen Account auf dem AS besitzt, wird zunächst ein neues Konto für sie angelegt. Nachdem der Bezahlvorgang erfolgreich abgeschlossen wurde, steht die Datei zum Download für Ginny bereit.

Statt des Downloads der Datei(en) könnte (beim Kauf einer bestimmten Anzahl von Songs) auch die Möglichkeit angeboten werden, die Songs auf einer CD an den Käufer zu verschicken.

Akteure	Bill, Ginny
Ablaufbeschreibung	<ol style="list-style-type: none"> 1) Ginny konsumiert einen Inhalt von Fred (z.B. im Radio). 2) Sie möchte die Datei kaufen und kontaktiert Bill. 3) Bill schickt einen Link zur Datei an Ginny. 4) Ginny bezahlt für die Datei und kann sie sich vom AS downloaden.
Vorbedingung(en)	-
Relevante Eingabedaten	Email-Adresse von Ginny
Ausnahmen	-
Nachbedingung	Ginny erhält den gewünschten Inhalt „on-demand“.

Tabelle 9: Anwendungsfall „Abruf eines Inhaltes – on-demand“

Ein weiterer Fall bei dem Ginny vor dem Erhalt der Ware bezahlen muss, tritt ein, wenn sie eine Datei direkt von Freds Web-Seite herunter laden möchte. (Ob die Datei auf Freds eigenem Server oder auf dem AS liegt spielt hierbei keine Rolle.) Diese Regelung ist sinnvoll für Dateien die sehr neu und noch nicht im System verbreitet sind. Ginny zahlt somit für die Möglichkeit ihrerseits eine Datei verteilen zu können die noch nicht jeder hat (und somit für die Chance einer hohen Provision) und einen Inhalt konsumieren zu können, den sie anderweitig (noch nicht) nicht erwerben kann.

Der Ablauf wäre wie folgt: Ginny möchte von Freds Web-Seite eine Datei herunter laden. Der Datei-Link führt direkt zur Pay-Seite des AS (oder zur Käufer-Registrierungsseite, falls Ginny Erstkäufer ist), auf der die Eigenschaften der Datei und der Preis angezeigt werden. Nachdem Ginny bezahlt hat (und der Kaufvorgang registriert wurde), wird der Download (mit der passenden Quittung für Ginny) gestartet.

Akteure	Bill, Ginny, Fred
Ablaufbeschreibung	<ol style="list-style-type: none"> 1) Ginny möchte eine Datei direkt bei Fred erwerben. 2) Sie wählt den entsprechenden Link und wird zur Kauf- bzw. Registrierungsseite umgeleitet. 3) Nachdem Ginny bezahlt hat, kann der Download der Datei durch Ginny erfolgen.
Vorbedingung(en)	-
Relevante Eingabedaten	-
Ausnahmen	-
Nachbedingung	Ginny erhält den gewünschten Inhalt mit ihrer Quittung.

Tabelle 10: Anwendungsfall „Direkt-Download“

3.3 Die Komponenten des „Potato Systems“

Zur Realisierung der im vorherigen Kapitel dargestellten Anwendungsfälle des „Potato Systems“, werden mehreren Komponenten benötigt. Die für die Client-Komponenten relevanten Anwendungsfälle sind: „Eine Datei registrieren“ und „Eine Datei überprüfen“. Ausgehend von diesen Anwendungsfällen, werden in diesem Kapitel die nötigen Client-Komponenten eingeführt und die damit verbundenen Funktionsabläufe modelliert. Die Registrierung von Providern und Vertriebspartnern, sowie die Realisierung des Kaufvorgangs ist nicht die Aufgabe der in der vorliegenden Arbeit konzipierten Client-Komponenten, sondern Bestandteil der Konzeption von Holger [Krauß-02].

3.3.1 Accounting Server

Zunächst ist der bereits im Kapitel 3.1 erwähnte Accounting-Server (AS) zu nennen. Der AS führt sowohl für Anbieter von Inhalten (z.B. Fred) als auch für Käufer (z.B. Ginny oder Harry) ein Verrechnungskonto. Bei Fred werden für die registrierten Dateien und die Einnahmen verbucht. Bei den Konsumenten werden die Bezahlungen gegen die Provisionen verrechnet. Die Konzeption und Realisierung des AS ist Bestandteil der Diplomarbeit von Holger [Krauß-02].

3.3.2 Creator

Der „Creator“ ist eine Client-Komponente des „Potato Systems“. Sie wird von Fred dazu benutzt, um Dateien beim AS zu registrieren. Die Benutzung der „Creator“-Komponente ist zwingend notwendig, da mittels dieser Komponente alle nötigen Informationen zur Überprüfung der Integrität und Authentizität aus der zu registrierenden Datei extrahiert und an den Server gesendet werden. (Die Alternative wäre das Versenden der kompletten Datei an den AS. Dies ist jedoch bei größeren Datenmengen (ganze CDs, Filme) nicht praktikabel.) Die „Creator“-Komponente übernimmt auch die automatische Umbenennung der Datei(en) nach der erfolgreichen Registrierung. Um die Integrität einer Datei bei einem späteren Kaufvorgang zu prüfen, berechnet die „Creator“-Komponente den Hash-Wert der Datei.

Die Überprüfung der Authentizität einer zu registrierenden Datei erfolgt zusätzlich mittels eines semantischen Fingerabdrucks (z.B. mittels des „AudioID“-Verfahrens [AudioID-02]). Die Überprüfung wird vom AS durchgeführt, wenn Fred eine Datei registrieren möchte, indem der semantische Fingerabdruck der zu registrierenden Datei (welcher vom „Creator“ ermittelt wird) mit allen Fingerabdrücken von bereits auf dem AS registrierten Dateien (und anderen urheberrechtlich geschützten Dateien) verglichen wird. Somit ist es möglich, automatisch festzustellen, ob Fred tatsächlich der Urheber der von ihm angebotenen virtuellen Ware ist. Wenn es sich bei der virtuellen Ware um ein Bild handelt, wird ein Sample in Form eines Thumbnails generiert. Bei einer Musikdatei wird ein Teil des Originals als Hörprobe entnommen.

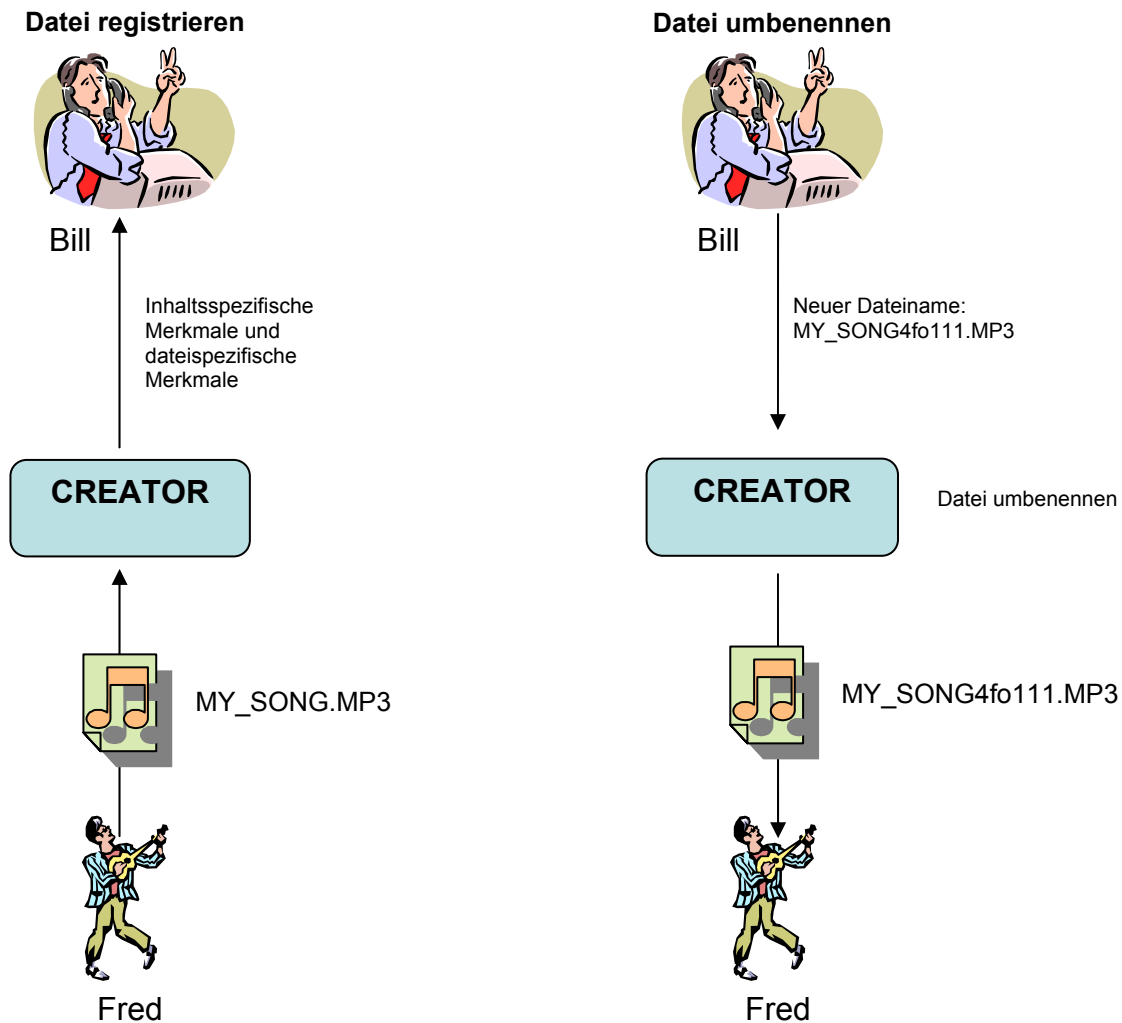


Abbildung 10: Die Aufgaben der „Creator“-Komponente

3.3.3 Redister

Die Client-Komponente „Redister“ (abgeleitet von „Re-Distribution“) ermöglicht Konsumenten und potentiellen Käufern wie Ginny und Harry, eine PS-Datei auf Integrität (und Authentizität) zu überprüfen (Fingerprint-Vergleich). Weiterhin übernimmt der „Redister“ nach einem erfolgreichen Kaufvorgang die automatische Verknüpfung des neuen Quittungsbelegs mit der Datei.

Grundsätzlich ist zu sagen, dass die Benutzung der Clientkomponente „Redister“ optional ist. Ginny soll immer auch die Möglichkeit haben, die TAN der Datei, deren Eigenschaften sie (vor einem eventuellen Kauf) überprüfen möchte, direkt auf der „Potato System“-Webseite einzugeben. Es werden ihr dann (sofern es sich um eine gültige TAN handelt) die Dateieigenschaften und Freds Beschreibung der Datei bzw. ihres Inhaltes angezeigt. Außerdem stehen auch (in Abhängigkeit vom jeweiligen Dateityp) Ausschnitte der Datei zum Download zur Verfügung, um die Authentizität des Inhaltes der Datei zu überprüfen.

Alle oben genannten Informationen werden auch bei der Verwendung der „Redister“-Komponente angeboten. Darüber hinaus ist jedoch eine schnelle Überprüfung der Integrität (d.h. die Prüfung, ob die Datei absichtlich oder unabsichtlich verändert wurde) möglich, indem der auf dem AS gespeicherte Hash-Wert (Fingerprint) der Originaldatei mit dem Hash-Wert der zu überprüfenden Datei verglichen wird. Dieser Vergleich gibt auch Auskunft über die Authentizität der zu überprüfenden Datei, da jede Veränderung des Inhaltes zwangsläufig einen veränderten Hash-Wert zur Folge hat. Außerdem besteht die Möglichkeit, nach dem Kauf die Datei(en) automatisch umbenennen zu lassen. Besonders unter dem Aspekt eines möglichen Missbrauchs des Systems durch einen Angreifer ist die Benutzung der „Redister“-Komponente dringend zu empfehlen (siehe Kapitel 3.3.5: Überlegungen zur Sicherheit).

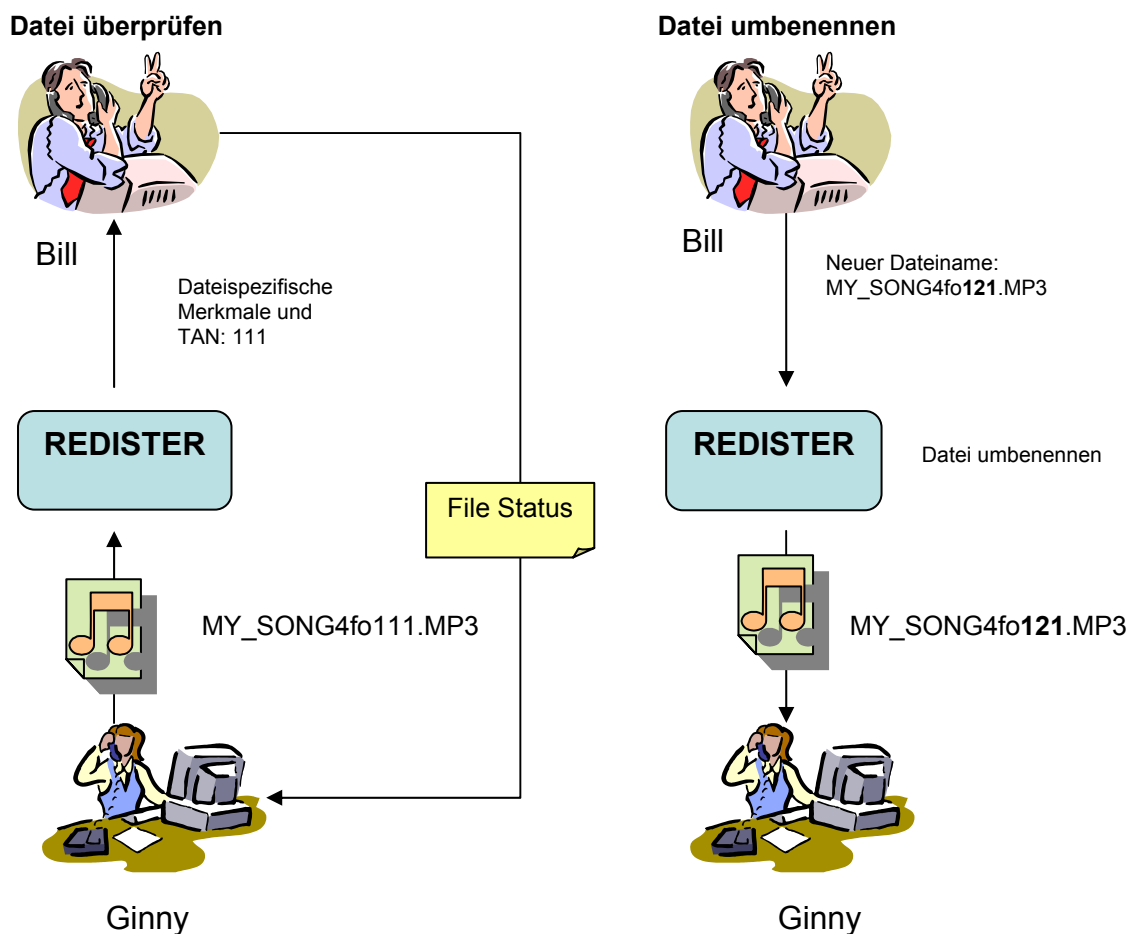


Abbildung 11: Die Aufgaben der „Redister“-Komponente

Eine spezielle Erweiterung des „Redisters“ soll auch Dateinamen auf Multisession-CD-Rs umbenennen können. Ein Szenario könnte wie folgt aussehen: Ginny erhält eine bespielte CD-R von Fred. Nachdem Ginny einen oder mehrere Songs der CD-R gekauft hat, übernimmt der „Redister“-Komponente die automatische Aktualisierung des Inhaltsverzeichnisses der CD-R. Dieses System könnte auch in Stand-Alone-Geräte integriert werden und in Form von „Kopierstationen“ im Handel eingesetzt werden. [4foOrg-02]

3.3.4 Zusammenfassung der Funktionalitäten der Client-Komponenten

In der folgenden Tabelle werden die einzelnen von den Client-Komponenten durchzuführenden Operationen zusammengefasst:

Operation	Verwendungszweck	Komponente
Hash-Wert über die Datei berechnen	Creator: Zur späteren Überprüfung der Integrität und Authentizität mittels Redister Redister: Überprüfung der Integrität und Authentizität	Creator, Redister
Dateispezifische Merkmale ermitteln (Größe, Datum der letzten Änderung)	Creator: Zur späteren Überprüfung der Integrität durch Ginny Redister: Überprüfung der Integrität	Creator, Redister
Semantischen Fingerabdruck ermitteln	Automatische Überprüfung der Authentizität einer zu registrierenden Datei durch den AS	Creator
Inhaltsspezifische Merkmale ermitteln (Samples, Thumbnails)	Zur späteren Überprüfung der Authentizität durch Ginny	Creator
Fred's Beschreibung der Datei und ihres Inhaltes speichern und an den AS übertragen	Zur späteren Überprüfung der Integrität und Authentizität durch Ginny	Creator
Lokale Dateipfade ermitteln und für die spätere Umbenennung temporär speichern	Creator + Redister: Zur späteren Umbenennung der Dateien	Creator, Redister
Die Dateien umbenennen	Creator: nach der erfolgreichen Registrierung wird die Quittung (der neue Dateiname) mit der registrierten Datei verknüpft Redister: nach dem erfolgreichen Kaufvorgang wird die Quittung (der neue Dateiname) mit der registrierten Datei verknüpft	Creator, Redister
Die Informationen in geeigneter Form zwischenspeichern	Creator + Redister: zur Auswertung und Weiterverwendung durch den AS	Creator, Redister
Daten an den AS senden	Creator: alle ermittelten Informationen und optional die Datei selbst an den AS senden zur Speicherung Redister: alle ermittelten Informationen an den AS senden zum Vergleich	Creator, Redister
Daten vom AS empfangen	Creator + Redister: Rückmeldungen vom Server oder Parameter empfangen	Creator, Redister

Tabelle 11: Die Funktionalitäten der Client-Komponenten

3.3.5 Überlegungen zur Sicherheit

Da das „Potato Systems“ wesentlich auf den Mechanismen zur Kontrolle der Integrität und Authentizität basiert, soll in diesem Kapitel ein Überblick über mögliche Angriffsszenarien auf das System gegeben werden.

Um überhaupt einen „sinnvollen“ (d.h. zum Beispiel mit finanziellem Nutzen) Angriff durchzuführen, muss der Angreifer beim System registriert sein. Da aber die Identität jedes Providers, der sich beim System anmeldet, geprüft wird, muss der Angreifer davon ausgehen, dass er nach der Aufdeckung eines Angriffs haftbar gemacht wird. Die Identitätsprüfung stellt somit die Grundlage für ein sicheres System dar. Im Folgenden werden verschiedene Angriffsszenarien hinsichtlich ihrer Durchführbarkeit und Auswirkungen auf das „Potato System“ untersucht.

1. Der Angreifer verschafft sich Zugang zum Server, um Daten zu manipulieren.

Mögliche Intentionen des Angreifers: Manipulation bestehender Accounts, z.B. um direkt finanziellen Nutzen aus falschen Auszahlungen zu erlangen, Anlegen von neuen Accounts mit gefälschten Identitäten, z.B. um weitere Angriffe vorzubereiten. Die Wahrscheinlichkeit eines „erfolgreichen“ (Manipulation bleibt unentdeckt, Auszahlung erfolgt) Angriffs kann durch entsprechende Konfiguration des Servers minimiert werden.

2. Der Angreifer versucht, durch die Verteilung fremder virtueller Ware zu profitieren (Voraussetzung: Der Angreifer besitzt eine Provider-Account).

- a. Der Angreifer versucht, eine „fremde“ virtuelle Ware am System anmelden. Bei der Anmeldung entnimmt die „Creator“-Komponente der anzumeldenden Datei einen semantischen (inhaltsbasierten) Fingerabdruck, eine Probe der Datei und berechnet den Hash-Wert der Datei und sendet diese Informationen an den Server. Der Server führt mittels dieser Informationen eine Authentizitätskontrolle durch, d.h. der semantische Fingerabdruck wird mit den in einer Datenbank gespeicherten Fingerabdrücken verglichen und es findet eine Überprüfung der Probedatei statt. Die Verlässlichkeit der Authentizitätskontrolle hängt dabei wesentlich vom Umfang der Datenbank ab. Da alle bereits im System registrierten Dateien mittels ihres Hash-Wertes eindeutig identifiziert werden können, kann mittels eines Vergleichs der Hash-Werte überprüft werden, ob bereits eine identische Datei im System existiert. Wenn eine Datei ähnlichen oder gleichen Inhaltes (bzw. eine identische Datei) gefunden wird, werden die Urheberrechte des Providers überprüft und der Angriff wäre entdeckt und somit gescheitert.

- b. Der Angreifer könnte eine „fremde“ Datei mit einer eigenen TAN versehen, um so von deren Vertrieb zu profitieren. Wenn Ginny nun die Datei kaufen möchte, berechnet der „Redister“ den Hash-Wert der Datei und sendet ihn an den Server. Da der vom „Redister“ berechnete Hash-Wert nicht mit dem für die angegebene TAN gespeicherten Hash-Wert übereinstimmt, wird der Kaufvorgang nicht durchgeführt und der Angriff ist gescheitert.
- c. Eine effektivere Angriffsmöglichkeit würde sich ergeben, wenn der Angreifer die Authentizitätskontrollen bei der Datei-Registrierung umgehen könnte. Dies wäre möglich, wenn es dem Angreifer gelingen würde, eine laufende Session zu übernehmen („Session Hijacking“: siehe [HSHCB-01]) und dann zum Anmelden der Dateien statt dem Original-„Creator“ eine eigene manipulierte Komponente zu verwenden. Die manipulierte Komponente müsste statt des semantischen Fingerabdrucks und der Probe der „fremden“ Datei, die entsprechenden Daten aus irgendeiner vom Angreifer generierten Datei entnehmen. Der Vergleich dieser Daten würde negativ ausfallen. Den Hash-Wert hingegen müsste der Angreifer aus einer modifizierten Version der „fremden“ Datei berechnen. Eine Neukomprimierung würde zum Beispiel genügen, um eine semantisch unveränderte Datei zu erhalten (damit Ginny bereit ist, dafür zu zahlen) und einen anderen Hash-Wert zu berechnen. Dies ist nötig, um den Vergleich der Hash-Werte zu bestehen, falls genau diese Datei schon im System registriert ist. Der Angreifer würde eine gültige TAN erhalten, mit der er nun die „fremde“ Datei verbreiten könnte.

Der Vergleich der Hash-Werte würde nun nicht mehr zur Aufdeckung des Angriffes führen. Allerdings unterscheiden sich jetzt die semantischen Merkmale der Datei von den auf dem Server gespeicherten Merkmalen. Wenn Ginny nun die Probedatei mit der zu kaufenden Datei vergleicht, wird der Angriff aufgedeckt.

3.4 Entwurf der Client - Komponenten

Der Entwurf der Client-Komponenten erfolgt unter Berücksichtigung der 3-Schichten-Architektur („3-tier“). Das Ziel dieser Architektur ist die Entkopplung der Bereiche Benutzungsoberfläche, Fachkonzept und Datenhaltung, um spätere Änderungen bzw. Erweiterungen der Anwendung zu erleichtern. [OOA-00]

Da bei diesem Entwurf nur die Client-Seite betrachtet wird, entfällt der Bereich der (persistenten) Datenhaltung. Der Entwurf der Benutzungsoberfläche ist implementierungsabhängig und erfolgt mittels der Erstellung von Prototypen (siehe Kapitel 4). Das Ziel dieses Kapitels ist damit die Erstellung des Fachkonzepts für die Client-Komponenten.

Bei der Betrachtung der verschiedenen, von den Client-Komponenten zu erfüllenden Aufgaben, ist eine Einteilung in drei Vorgänge zu erkennen. Diese Einteilung wird durch die Verwendung von drei Klassen umgesetzt:

- Registrierung von neuen Dateien bei Bill („**Creator**“)
- Kontrolle von registrierten Dateien auf Integrität und Authentizität („**Checker**“)
- Anfügen der Quittung (Datei umbenennen) an die Datei nach einem Kaufvorgang bzw. der Registrierung einer neuen Datei („**Renamer**“)

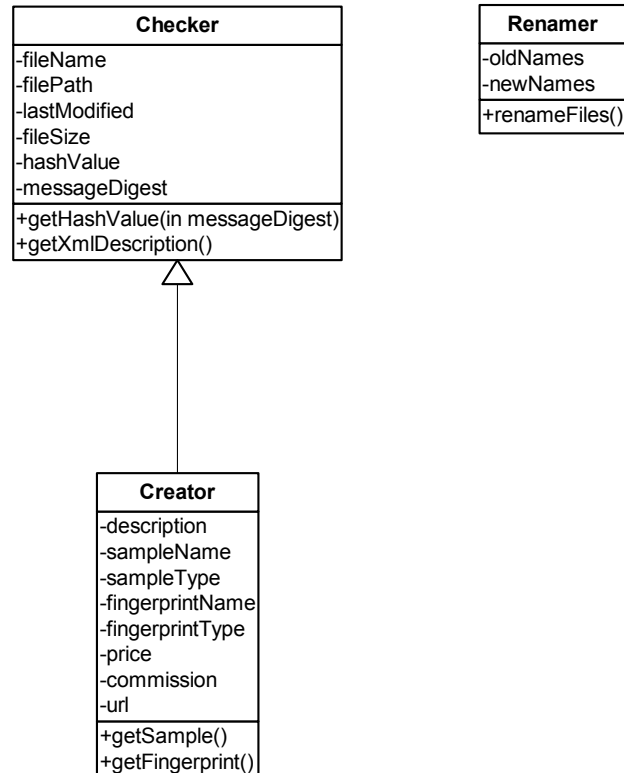


Abbildung 12: Klassendiagramme der Client-Komponenten (Fachkonzept)

4 Implementierung der Client-Komponenten

Die Implementierung der Client-Komponenten soll unter Berücksichtigung der folgenden Punkte durchgeführt werden:

- Portabilität
- Benutzerfreundlichkeit
- Sicherheit
- Erfüllung der geforderten Funktionalitäten

Den besten Kompromiss zwischen allen Anforderungen bietet der Einsatz von Java-Applets (SUN). Java-Applets sind Java-Applikationen, welche nur innerhalb eines Containers (z.B. in einem Java-fähigen Browser) ausgeführt werden können und bestimmten Einschränkungen unterliegen (siehe Kapitel 4.1.3).

Es folgt eine Beurteilung von Java-Applets bezüglich der genannten Kriterien:

- **Portabilität:** Die Client-Komponenten können auf allen Systemen benutzt werden, für die Browser mit Java-Unterstützung existieren (siehe Kapitel 4.1.1). Im Gegensatz dazu müssten bei der Verwendung von Stand-Alone-Programmen unterschiedliche Versionen für die verschiedenen Ziel-Systeme realisiert werden.
- **Benutzerfreundlichkeit:** Es müssen keine separaten Programme installiert werden, da die Komponenten zusammen mit der Web-Seite vom AS geladen werden. Dem Benutzer wird so auch eine konsistente Benutzeroberfläche angeboten.
- **Sicherheit:** Applets werden innerhalb einer eingeschränkten Systemumgebung („Sandbox“) ausgeführt. Zugriffe auf Ressourcen außerhalb dieser Systemumgebung werden nur erlaubt, wenn sie der Sicherheits-Policy der jeweiligen Umgebung entsprechen bzw. die Herkunft der Applets durch eine digitale Signatur eindeutig bestimmbar ist und der Benutzer mit der Ausführung einverstanden ist. Durch den Einsatz von **signierten** Applets wird der Benutzer über den Hersteller der Applets informiert und kann entscheiden, ob er ihm vertraut oder nicht. Der Benutzer kann somit sicher sein, dass es sich bei der benutzten Software wirklich um die PS-Komponenten handelt und nicht etwa um von Dritten manipulierte Programme.
- **Erfüllung der geforderten Funktionalitäten:** Durch den Einsatz von Java als Implementierungssprache (und konkret durch die Verwendung von signierten Applets auf der Client-Seite und Servlets bzw. JSP auf der Server-Seite) sind alle erforderlichen Voraussetzungen für die Realisierung der gewünschten Funktionalitäten gegeben (z.B. durch die Packages *java.security* für sicherheitsrelevante Operationen und *java.net* für Netzwerkoperationen).

Alternativen

ActiveX-Controls

ActiveX ist eine von Microsoft entwickelte Technologie, die aus mehreren Komponenten besteht. Programme, welche auf der ActiveX-Technologie beruhen, können in Form von ActiveX-Controls über das Internet geladen werden und benötigen zur Ausführung einen Container (z.B. einen ActiveX-fähigen Browser). Prinzipiell können ActiveX-Controls in jeder Programmiersprache implementiert werden. Für den Einsatz im Internet wurde von Microsoft nachträglich die Möglichkeit eingeführt, ActiveX Controls mittels der proprietären „Authenticode“-Technik zu signieren. Wenn ein signiertes ActiveX-Control geladen wird, hat der Benutzer die Möglichkeit, die Ausführung zuzulassen oder abzulehnen. Wenn der Benutzer das ActiveX-Control akzeptiert, wird es wie eine normale Applikation auf dem Zielsystem ausgeführt und hat damit vollen Zugriff auf die Systemressourcen. ActiveX-Controls sind plattformabhängig, sie können nur auf Win32-Plattformen ausgeführt werden. [Mack-99]

Wegen der Plattformabhängigkeit und der Sicherheitslücken im Zusammenhang mit ActiveX-Controls kommt der Einsatz dieser Technologie für die Implementierung der Client-Komponenten des „Potato Systems“ nicht in Frage.

Stand-Alone-Programme

Der Hauptnachteil von Stand-Alone-Programmen besteht in der Plattformabhängigkeit. Zusätzlich sind auch bei normalen Applikationen gewisse Sicherheitsrisiken gegeben (Viren, „Trojaner“), deren Bekämpfung in der Hand des jeweiligen Benutzers liegt.

Dennoch stellt die Verwendung von Stand-Alone-Programmen eine Alternative (oder zusätzliche Möglichkeit) zur Verwendung von Java-Applets dar, besonders unter dem Gesichtspunkt der Integration der Funktionalitäten der Clientkomponenten in P2P-Software (wobei dieser Ansatz auch unter Verwendung von Java-Applets umsetzbar ist).

4.1 Verwendung von Java-Applets: Vorüberlegungen

Aus der Entscheidung, die Implementierung der Client-Komponenten mittels Java-Applets durchzuführen, ergeben sich folgende Fragen, welche vor dem Beginn der Realisierung zu klären sind:

- Welche Browser (in welchen Versionen) sind am meisten verbreitet, bzw. werden von den meisten Benutzern verwendet?
- Welche Eigenschaften hinsichtlich ihrer Java-Fähigkeit haben die relevanten Browser?
- Ausgehend vom Ergebnis dieser Fragestellungen: Für welche Java Version sind die Applets zu implementieren, damit die Mehrheit der Internet-User die Applets verwenden kann?

Eine aktuelle Statistik von „theCounter.com“ [theCounter-02] zeigt, welche Browser im Zeitraum vom 01. September 2002 bis zum 30. September 2002 am häufigsten verwendet wurden (beim Zugriff auf die von der Statistik berücksichtigten Seiten). Demnach erfolgten 93,4% der insgesamt 355.476.251 Zugriffe mit dem Internet Explorer (Versionen 3-6 berücksichtigt) und 5,3% mit dem Netscape Communicator (Versionen 4-6 und kompatible Browser berücksichtigt). 0,9% der Aufrufe erfolgten mittels Opera (alle Versionen berücksichtigt). Die restlichen Zugriffe erfolgten mit älteren Versionen der genannten Browser oder mit anderen Browsern. Weiterhin wurde festgestellt, dass 87% (312211931) der benutzten Browser ihre Java-Unterstützung aktiviert hatten.

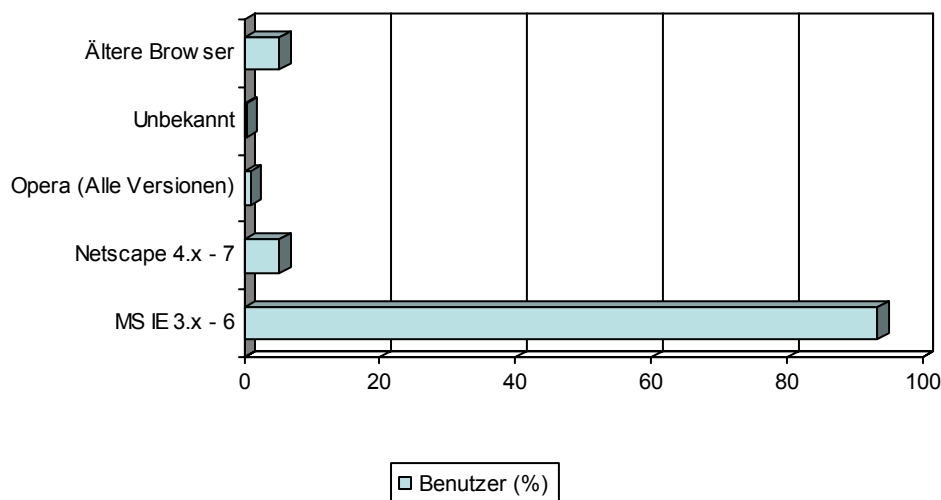


Abbildung 13: Browser-Statistik

Auch wenn eine derartige Statistik mit Vorsicht betrachtet werden sollte, da nicht nur die Vorlieben je nach Zielgruppe variieren, sondern auch die Erkennung der Browser nicht immer sicher gewährleistet werden kann, liefert sie doch zumindest einen Anhaltspunkt für das derzeitige Browser-Nutzungsverhalten.

4.1.1 Browser mit Java-Unterstützung

Da die Namenspolitik der Firma SUN im Bezug auf Java zu Verwirrungen führen könnte, folgt zunächst eine kurze Erläuterung der verwendeten Begriffe. Allgemein erfolgt die Implementierung der „Potato System“ Client-Komponenten mittels Java2. Java2 umfasst alle Versionen des JDKs (Java Developer Kit) ab Version JDK1.2(Final). Alle früheren Versionen des JDKs sind Bestandteil der Java1-Plattform. Wenn im Folgenden von Java1 oder Java2 gesprochen wird, so sind damit die Plattformen gemeint. [Java2-99]

Damit ein Browser Java-Applets ausführen kann, muss dieser Java unterstützen und die Ausführung von Java-Programmen muss in den Browsereinstellungen aktiviert sein.

Im Browser muss ein Java-Laufzeitsystem integriert sein (bzw. ein entsprechendes Java Runtime Environment (JRE) auf dem System installiert sein), in dem dann diese Applets ausgeführt werden. Die Versionsnummern der einzelnen Laufzeitumgebungen korrespondieren mit den Versionsnummern der zugehörigen JDKs.

Beim Aufruf einer mit einem Applet verknüpften Seite oder beim expliziten Starten des Java-Programms aus einer solchen Seite heraus wird nur der Byte-Code des übersetzten Java-Quelltextes zum Client übertragen. Der Java-Byte-Code wird dann in der JVM (Java Virtual Machine) auf dem Client ausgeführt. Der übertragene Code durchläuft zunächst mehrere Kontrollen, um sicherzustellen, dass die Sicherheitsbestimmungen der JVM eingehalten werden (zur Sicherheit von Java Programmen siehe Kapitel 4.1.3). Die Java Virtual Machine wird als virtuell bezeichnet, da sie als eigenständiges Programm oberhalb der Hardware und des Betriebssystems abläuft. Auf diese Weise wird die Plattformunabhängigkeit erreicht.

Die folgende Tabelle zeigt die Java-Unterstützung der wichtigsten (im Sinne der Verbreitung) Browser-Typen und Versionen:

Browser	Java-Unterstützung
Netscape 3.x	Eigene JVM: Java 1
Netscape 4.x	Eigene JVM: Java 1, Unterstützung des SUN Java-Plugins
Netscape 6+	Keine eigene JVM, Unterstützung des SUN Java-Plugins, teilweise Auslieferung mit JRE1.3.1
Mozilla 1.0+	Keine eigene JVM, Unterstützung des SUN Java-Plugins
MS Internet Explorer 3.x bis 5.x	Eigene JVM: Java 1, Unterstützung des SUN Java-Plugins
MS Internet Explorer 6	Keine eigene JVM, Unterstützung des SUN Java-Plugins
Opera 3.x	Keine eigene JVM, Unterstützung des SUN Java-Plugins
Opera 4+	Keine eigene JVM, direkte Unterstützung einer installierten JRE, teilweise Auslieferung mit JRE1.3.1

Tabelle 12: Browserübersicht (Unterstützung für Java)

Um Web-Browser mit einer alten Java-Version zu befähigen, Applets welche eine neuere Java-Version benötigen auszuführen, gibt es die folgenden Möglichkeiten:

- Installation einer aktuellen Java-Laufzeitumgebung (JRE) auf dem Client-Rechner. Danach können alle auf dem System installierten Java-fähigen Browser die neue JVM nutzen.
- Installation des Sun Java Plug-ins im Browser über eine Netzwerkverbindung „on demand“ beim Aufruf einer entsprechend präparierten Web-Seite (siehe Kapitel 4.1.2).

Verschiedene Gründe sprechen für den Einsatz des Java Plug-ins (gegenüber der Verwendung der Browser-internen JVMs):

- ✓ Volle Unterstützung aller Funktionalitäten von Java2
- ✓ Browserunabhängiges, konsistentes Verhalten der Applets
- ✓ Bei der Applet-Programmierung muss keine Rücksicht auf die Einschränkungen oder Besonderheiten der verschiedenen Browser-internen JVMs genommen werden (siehe Kapitel 4.1.4: Signierte Applets).
- ✓ Sicherheitsaspekt: Zum Beispiel warnt Microsoft vor Sicherheitslücken in der eigenen (veralteten) JVM (welche zusammen mit dem Service-Pack 1 für Windows XP erhältlich ist) und empfiehlt die Benutzung des Java Plug-ins von SUN. [MSJVM-02]
- ✓ Java-Unterstützung für die Browser, die keine eigene JVM beinhalten (siehe obige Tabelle)
- ✓ Unterstützung von signierten Applets, wodurch ein einheitlicher Signierungsprozess ermöglicht wird, d.h. es besteht keine Notwendigkeit für unterschiedliche Applet-Versionen für Netscape-Browser und Microsoft-Browser (siehe Kapitel 4.1.3)
- ✓ Das Plug-in ist für alle wichtigen Plattformen erhältlich: Windows 95, Windows 98, Windows ME, Windows NT 4.0, Windows 2000, Windows XP; Solaris 7 – 9 und Linux (Bemerkung: Apple's Mac OS X wird bereits mit einer Java2 Umgebung ausgeliefert und auch für Mac OS Classic bietet Apple eine spezielle JRE (MRJ – Mac OS Runtime for Java) an.)

Als Nachteil könnte der notwendige Download des Plug-ins gewertet werden, wenn der Benutzer die geforderte oder eine neuere Version des Plug-ins (oder alternativ die passende JRE) noch nicht installiert hat. Dieser Punkt relativiert sich jedoch, wenn man berücksichtigt, dass praktisch alle aktuellen Browser (IE 6, NC 6+, Opera, Mozilla) ohne eigene JVM ausgeliefert werden. Außerdem kann davon ausgegangen werden, dass ein Download in der Größenordnung von einigen MByte kein Hindernis darstellt, wie man am Verhalten der Benutzer bzgl. File-Sharing erkennen kann. Auch die Benutzerfreundlichkeit leidet nicht unter Installation des Plug-ins (bzw. der JRE), da diese durch spezielle Tags in der aufrufenden Web-Seite automatisch eingeleitet wird. [DevGuide-02]

4.1.2 Verwendung des Sun Java Plug-ins

Das Java Plug-in ist ein Produkt von SUN Microsystems, welches die Ausführung von Applets mittels der Sun JVM ermöglicht. Um das Plug-in aus einer Web-Seite heraus zu starten, werden spezielle HTML-Tags (*OBJECT* für IE und NS6+, *EMBED* für NS4.x) benutzt. Eine genaue Anleitung zur Benutzung der Tags findet man im „Java Plug-in Developer Guide“. [DevGuide-02]

HTML Tag	Parameter	Browser-Verhalten
OBJECT	<p>Attribut „classid“ ist fest, z.B. „clsid:8AD9C840-044E-11D1-B3E9-00805F499D93“ (<i>dynamic versioning</i>)</p> <p>Beispiel „codebase“ Attribut: codebase="http://java.sun.com/products/plugin/autodl/jinstall-1_3-windows-i586.cab#Version=1,3,0,0"</p>	<p>IE und Netscape 6+: Wenn der Benutzer eine niedrigere Haupt-Version des Plug-ins besitzt, als im „codebase“ Attribut angegeben ist, wird der Benutzer aufgefordert die notwendige Version herunter zu laden. Wenn der Benutzer die gleiche oder eine höhere Version installiert hat, als im „codebase“ Attribut angegeben ist, dann wird diese benutzt.</p>
OBJECT	<p>Attribut „classid“ ist variabel, es kann genau festgelegt werden, welche Plug-in Version benutzt werden soll, z.B. „clsid:CAFEEFAC-0014-0000-0000-ABCDEFEDCBA“ für Version 1.3.1 (<i>static versioning</i>)</p> <p>Beispiel „codebase“ Attribut: codebase="http://java.sun.com/products/plugin/autodl/jinstall-1_3-windows-i586.cab#Version=1,3,0,4"</p>	<p>IE und Netscape 6+: Wenn der Benutzer nicht genau die geforderte Version des Plug-ins installiert hat, wie sie im „codebase“ Attribut angegeben ist, wird er aufgefordert, die notwendige Version herunter zu laden.</p>
EMBED	<p>Das „type“ Attribut hat folgende Form: type="application/x-java-applet;version=1.3" (<i>dynamic versioning</i>)</p> <p>pluginspage="http://java.sun.com/j2se/1.3/jre"</p>	<p>Netscape 4.x: Wenn der Benutzer eine niedrigere Haupt-Version installiert hat, als die im „type“ Attribut angegebene, wird die URL, welche im Attribut „pluginspage“ spezifiziert ist, aufgerufen. Wenn der Benutzer die gleiche oder eine höhere Version installiert hat, wird diese benutzt.</p>
EMBED	<p>Das „type“ Attribut hat folgende Form: type="application/x-java-applet;jpi-version=1.3.1_04" (<i>static versioning</i>)</p> <p>pluginspage="http://java.sun.com/j2se/1.3/jre"</p>	<p>Netscape 4.x: Wenn der Benutzer nicht genau die geforderte Version des Plug-ins installiert hat, wie sie im „type“ Attribut angegeben ist, wird die URL, welche im Attribut „pluginspage“ spezifiziert ist, aufgerufen.</p>

Tabelle 13: Aufruf des Java Plug-ins

Alle Applets des „Potato Systems“ verwenden „dynamic versioning“.

Da der Aufruf der Applets mittels JSP (Java Server Pages) erfolgt, bietet sich alternativ die Verwendung des „<jsp:plugin>“-Action-Elements an. In Abhängigkeit vom verwendeten Browser wird dabei der passende HTML-Code automatisch generiert (korrespondierend zu obigen Ausführungen). [DevGuide-02]

4.1.3 Sicherheitsrestriktionen für Applets

Java-Applets werden im Browser standardmäßig in einer so genannten „Sandbox“ ausgeführt. Der Begriff „Sandbox“ bezeichnet dabei eine Laufzeitumgebung, in dem das ausgeführte Programm bestimmten Restriktionen beim Zugriff auf lokale Ressourcen unterworfen ist.

Im Folgenden werden einige Einschränkungen genannt (die im Zusammenhang mit der Implementierung der „Potato System“-Applets besonders interessant sind):

- Dateien können nicht gelesen oder geschrieben werden
- Dateien können nicht umbenannt oder gelöscht werden
- Neue Dateien oder Verzeichnisse können nicht angelegt werden
- Dateieigenschaften können nicht gelesen werden
- Netzwerkverbindungen können nur zum Herkunftsserver aufgebaut werden
- Auf dem Host können keine weiteren Applikationen aufgerufen und ausgeführt werden
- Die Systemeigenschaften sind nur beschränkt lesbar

Folgende Systemeigenschaften dürfen in der Sandbox-Umgebung nicht gelesen werden:

Eigenschaft	Bedeutung
java.home	Das Java Installationsverzeichnis
java.classpath	Der Java Klassenpfad
user.name	Der Name des Anwenders
user.home	Das Home-Directory des Anwenders
user.dir	Das aktuelle Arbeitsverzeichnis des Anwenders

Tabelle 14: Systemeigenschaften

Applikationen haben im Normalfall uneingeschränkten Zugriff auf alle Ressourcen, während Applets unter den Beschränkungen der Sandbox ausgeführt werden. Fordert ein Programm den Zugriff auf unberechtigte Ressourcen, wird eine spezielle Ausnahmebehandlung (*SecurityException*) ausgelöst. [SecuringJava-99]

4.1.4 Signierte Applets

Mit dem JDK1.1 (Java1) wurden signierte Applets (Signed Applets) eingeführt. Wenn der Benutzer der Quelle des Applets vertraut, erhält das Applet vollen Zugriff auf alle Ressourcen und wird somit wie eine Applikation lokalen Ursprungs behandelt („All/Nothing“-Prinzip). Seit dem JDK1.2 (Java2) ist es mittels einer „Security Policy“ möglich, fein-granulierte Privilegien für verschiedene Applets festzulegen, indem diese in einer lokalen Datei (java.policy) definiert werden. Dieses Schema der Rechtevergabe scheint aber nur in administrierten Umgebungen praktikabel zu sein, da dem normalen End-User die manuelle Konfiguration der Rechte nicht abverlangt werden kann. [Mindprod-02]

Im Folgenden wird ein Überblick über den Vorgang des Signierens von Applets für das Sun Java Plug-in gegeben:

- Der Programmierer des Applets erstellt ein JAR-Archiv (ZIP-Archiv plus Meta-Daten) welches die Applet-Klassen beinhaltet.
- Der Programmierer erstellt ein RSA-Schlüsselpaar, bestehend aus einem privaten Teil (Private Key) und einem öffentlichen Teil (Public-Key-Certificate), z.B. mittels des im Java2 SDK enthaltenen *Keytools*. RSA (benannt nach den Erfindern Rivest, Shamir und Adleman) ist ein verbreitetes Public-Key-Kryptographie-Verfahren (siehe [WobstKRYPT-98]).
- Der Private Key wird benutzt, um das Applet (genauer: den Inhalt des JAR-Archivs) zu signieren. Beim Signieren wird der Hash-Wert von der Originaldatei (z.B. von einem Class-File) berechnet und dann mit dem Private Key verschlüsselt. Das Public-Key-Certificate beinhaltet, neben dem Public Key zur Entschlüsselung der Hash-Werte, Informationen über den Anbieter des Applets, welche dieser während der Erstellung des Schlüsselpaares angegeben hat.
- Ein derart signiertes Applet („self-signed“) gibt aber noch keine Auskunft über die Vertrauenswürdigkeit des Applet-Anbieters, da dieser beliebige Informationen bezüglich seiner Identität machen kann.
- Um seine Vertrauenswürdigkeit zu beweisen, muss der Programmierer sein Public-Key-Certificate von einer offiziellen Zertifizierungsfirma (CA– Certificate Authority) signieren lassen, welche vorher seine Identität überprüft hat. Hierzu muss der Programmierer ein *Certification Signing Request*, basierend auf dem Public-Key-Certificate des zuvor erstellten Schlüsselpaares, an die CA senden (Erstellung auch mittels des *Keytools*).
- Die CA sendet als Antwort ein *Certification Reply*, welche das mit dem Private Key der CA signierte und mit Informationen über die CA versehene Public-Key-Certificate des Programmierers enthält. Der Programmierer muss nun noch das erhaltene Zertifikat in den *Keystore* des erstellten Schlüsselpaares importieren und kann dann mittels des im Java2 SDK enthaltenen Tools *Jarsigner* sein Applet signieren.

Um die Signatur eines Applets zu überprüfen, wird das Zertifikat (der Public Key) der jeweiligen Zertifizierungsfirma benötigt. Diese „Root Certificates“ sind zum Beispiel in den verschiedenen Browsern vorinstalliert. Um RSA-Signaturen unabhängig vom Browser überprüfen zu können, ist ein so genannter *Cryptographic Service Provider (CSP)* im Java Plug-in integriert.

Wenn ein signiertes Applet geladen werden soll, extrahiert der *PluginClassLoader* alle Instanzen (und deren „Certificate Chain“), welche das Applet signiert haben, aus dem signierten Applet (bzw. JAR-Archiv) und versucht die Signaturen zu prüfen. Hierzu werden die signierten Public-Key-Certificates der einzelnen Instanzen vom CA angefordert. Die Public Keys aller Instanzen bilden eine hierarchisch geordnete Liste, eine so genannte „Certificate Chain“, beginnend mit dem Public Key des „Signierers“ des Applets und endend bei dem Public Key der CA. Der Public Key eines Zertifikats in der Liste wird dazu benutzt, um die Signatur des jeweiligen Vorgängers zu überprüfen. Das „Root Zertifikat“ der CA ist „self-signed“. Die Annahme hierbei ist, dass die CA vertrauenswürdig ist, da es sich um eine bekannte öffentliche Einrichtung bzw. Firma handelt. Wenn das Plug-in die „Certificate Chain“ erfolgreich geprüft hat, bekommt der Anwender ein Dialogfenster mit einer Sicherheitsmeldung präsentiert. Nun kann er Details über das Zertifikat lesen und festlegen, ob die Sicherheitsbestimmungen aufgehoben werden oder nicht.

In Abhängigkeit von der Ziel-JVM (Browser-Typ) des Applets, unterscheiden sich die benötigten Zertifikate und Vorgehensweisen bei der Signierung. Beispielsweise unterstützt die Microsoft Windows JVM nur die proprietäre Authenticode-Signierungstechnologie um Applets zu signieren, während bei der Verwendung der Netscape JVM Änderungen im Quelltext nötig sind, d.h. es müssen Netscape-spezifische Java-Klassen („Netscape Capabilities API“) verwendet werden. Nähere Informationen und eine genaue Anleitung dazu findet man unter [Signing-99] und [Mindprod-02].

An dieser Stelle wird einer der bereits weiter oben (siehe 4.1.1) erwähnten Vorteile der Verwendung des Java Plug-ins von Sun erkennbar: ein einheitlicher und Browser-unabhängiger Signierungsprozess. Die Vorgehensweise bei der Signierung von Applets für die Verwendung mit dem Java Plug-in werden im Detail im „Java Plug-in Developer Guide“ [DevGuide-02] beschrieben.

4.1.5 Fazit

Im Folgenden werden die Schlussfolgerungen aus den Überlegungen dieses Kapitels zusammengefasst:

- Um die Applets zu benutzen, muss der Benutzer das Java Plug-in (oder alternativ die entsprechende JRE) von Sun installieren. Alle Java-fähigen Browser sind so in der Lage die Applets auszuführen.
- Die Installation des Plug-ins (bzw. die Weiterleitung zur Download-Seite für die JRE) erfolgt automatisch mittels der Einbindung entsprechender Tags in die aufrufende Web-Seite.
- Zur Implementierung der Applets wird Java2 (JDK1.3.1) benutzt. Diese Hauptversionsnummer (1.3) ist auch die Mindestanforderung für die Benutzung der Applets. Wenn auf dem Client-Rechner bereits eine höhere Version der JRE installiert ist, wird diese benutzt, wenn eine ältere Version installiert ist, erfolgt der automatische Download des entsprechenden Plug-ins bzw. der automatische Aufruf der entsprechenden Download-Seite („dynamic versioning“).
- Um die erforderlichen Funktionalitäten anbieten zu können, werden die Applets signiert (gemäß RSA X.509).

4.2 Implementierung der Prototypen

Ein Bestandteil dieser Arbeit ist die Realisierung von Prototypen der Client-Komponenten, mit deren Hilfe ein Feldtest des „Potato Systems“ durchgeführt werden soll. Prinzipiell haben Prototypen die Aufgabe, die Praxistauglichkeit der modellierten Lösungen nachzuweisen. Die Entwicklung der Prototypen erfolgte „evolutionär“, d.h. „alte“ Prototypen wurden nicht verworfen sondern dienten als Basis für neue Prototypen. [OOSWE-98]

Die Funktionalitäten der Client-Komponenten werden auf drei Applets verteilt:

- Creator-Applet: Registrierung von Dateien
- Checker-Applet: Überprüfen von Dateien
- Renamer-Applet: Umbenennen von Dateien

Weitere Möglichkeiten wären die Verwendung eines Applets (welches alle Funktionen erfüllt) oder die Verwendung von zwei Applets („Creator“ + „Renamer“ bzw. „Checker“ + „Renamer“). Der Einsatz dieser Varianten würde zwar einen geringeren Kommunikationsaufwand mit dem Server bedeuten, hätte aber auch eine deutliche Verschlechterung der Übersichtlichkeit und somit der Benutzerfreundlichkeit zur Folge. Die Einteilung der Client-Komponenten in drei Applets macht auch unter dem Gesichtspunkt der Benutzungshäufigkeit Sinn, davon auszugehen ist, dass wesentlich mehr Kaufvorgänge durch Ginny erfolgen, als Anmeldungsvorgänge durch Fred. Es wäre also wünschenswert, wenn das Applet zur Dateiüberprüfung möglichst „schlank“ und übersichtlich realisiert wird. Dies wäre speziell bei der Variante mit einem Applet nicht der Fall.

In diesem Kapitel wird die Implementierung der Funktionalitäten (siehe Kapitel 3.3 und 3.4) der Applets erläutert: die Kommunikation mit dem Server, die Berechnung des Hash-Wertes, die Speicherung der ermittelten Daten und die Konfiguration (z.B. Sprachspezifische Einstellungen) der Applets mittels Init-Parametern (siehe *Anhang A*).

4.2.1 Dateien registrieren (Creator-Applet)

The screenshot shows the Creator-Applet interface. At the top, there are three buttons: "Add File", "Remove File", and "Remove All". Below these is a large empty rectangular area. Underneath that is a checkbox labeled "Delete temporary file". Below the checkbox is a text area containing the instruction: "Insert your description for the selected file here, fill out the fields below and press the 'Apply Settings' button." Below the text area are three input fields: "Link", "Price (Cent)", and "Commission (%)". To the right of the "Price (Cent)" and "Commission (%)" fields is an "Apply Settings" button. At the bottom right of the interface is a "Send Data" button.

Abbildung 14: Screenshot - Creator-Applet

Nachdem die entsprechende Seite mit dem Creator-Applet geöffnet wurde, hat Fred die Möglichkeit, die Dateien die er registrieren möchte, auszuwählen. Das Applet berechnet nun die Hash-Werte der Dateien und extrahiert alle anderen Merkmale der Dateien bzw. ihres Inhaltes. Die Berechnung des Hash-Wertes erfolgt mittels Funktionen der JCA (Java Cryptography Architecture). Die JCA bietet neben Methoden zur Verschlüsselung, Schlüsselverwaltung, Erstellung von digitalen Unterschriften auch Methoden zur Generierung sicherer Prüfsummen (Message Digest). Als Algorithmus wird SHA-1 (Secure Hash Algorithm) mit einer Ausgabelänge von 160 Bit verwendet (siehe Kapitel 2.3.4).

Die ermittelten Informationen werden in einer XML-Datei (temporär) gespeichert (siehe *Anhang B*). Die XML-Datei wird zusammen mit typabhängigen Sample-Dateien (Hörproben, Thumbnails etc.) und dem semantischen Fingerabdruck (Binärdatei) in einem temporären Verzeichnis gespeichert, welches zu einem ZIP-File (Struktur: nur Dateien, keine Unterverzeichnisse) zusammengefasst und per HTTP-POST an den Server (genauer: an das entsprechende Servlet) gesendet wird. Dazu muss das Applet eine Verbindung zur zugehörigen Servlet-URL aufbauen. Danach öffnet das Applet einen Input- und einen Outputstream für die Kommunikation und kann dann via POST oder GET Daten an das Servlet senden, bzw. Daten vom Servlet empfangen. Während bei der GET-Methode die Parameter an die eigentliche Servlet-URL angehängt werden, werden bei der POST-Methode die Daten über einen Outputstream gesendet. Dies hat den Vorteil, dass außer reinen Textdaten auch binäre Werte (der *FileOutputStream* des ZIP-Archivs) an das Servlet gesendet werden können.

Die Ergebnisse der Auswertung (Überprüfung der Authentizität etc.) werden auf einer neuen Seite dargestellt, deren URL per HTTP-POST an das Applet gesendet wird. Das Applet öffnet dann die neue Seite. Die Kommunikation ist damit abgeschlossen (siehe Abb.15).

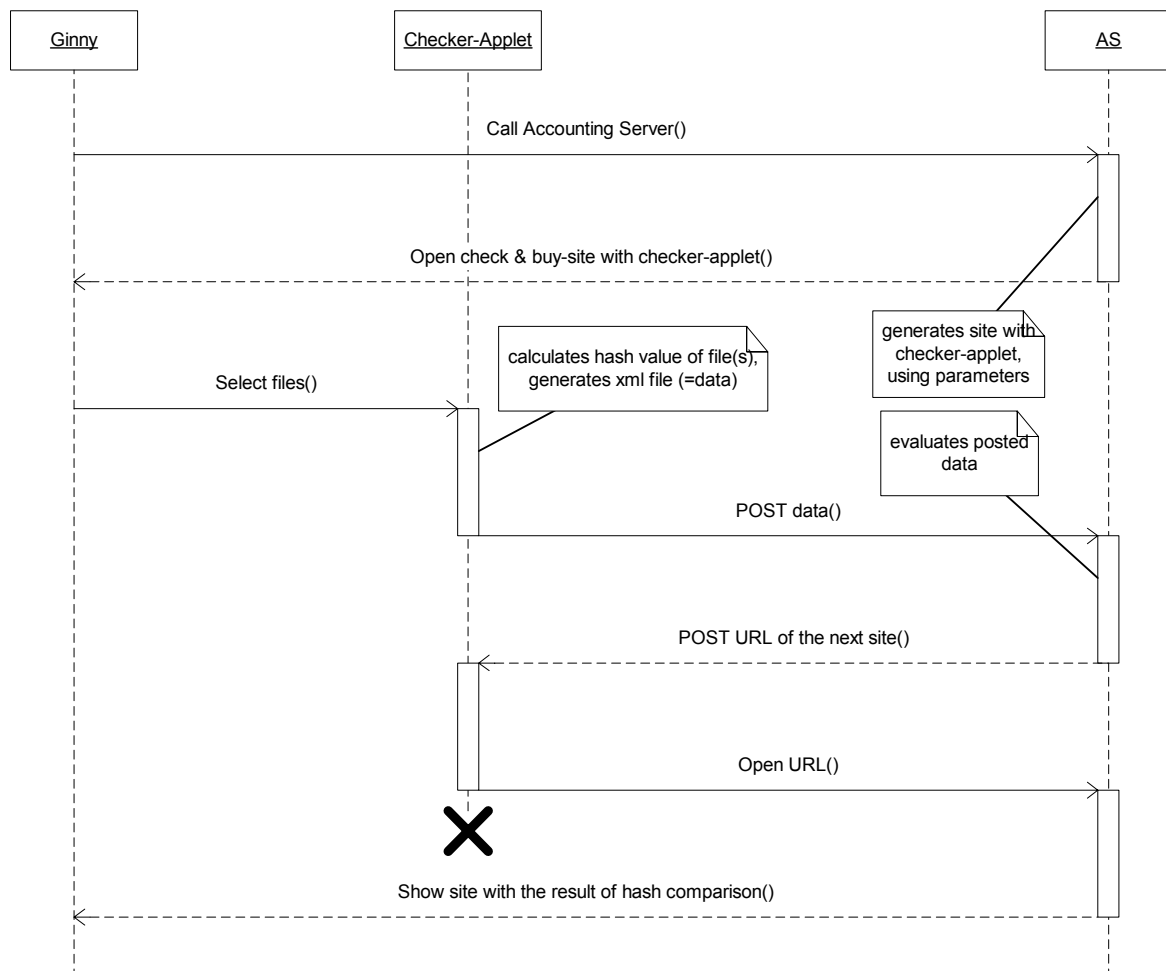


Abbildung 15: Sequenzdiagramm - Datei registrieren

4.2.2 Dateien überprüfen (Checker-Applet)

Nachdem Ginny die entsprechende Seite (mit dem Checker-Applet) des AS in ihrem Browser geöffnet hat, kann sie die Dateien, welche sie überprüfen möchte, auswählen. Alle gewählten Dateien werden in einer Liste dargestellt. Wenn Ginny den Vorgang bestätigt hat, berechnet das Applet die Hash-Werte (SHA-1) der ausgewählten Dateien und trägt diese zusammen mit anderen Merkmalen, wie Dateigröße, Datum der letzten Änderung und lokale Pfade in eine XML-Datei ein. Diese Datei wird per POST an den Server gesendet (siehe 4.2.1). Die Struktur der XML-Datei wird in *Angang B* dargestellt.

Der Server vergleicht anhand des Dateinamens die in der Datenbank gespeicherten Daten mit den vom Applet übermittelten Daten und generiert für die Ergebnisse des Vergleichs eine neue Seite deren URL als Antwort an das Applet gesendet wird. Das Applet öffnet dann diese URL. Der Überprüfungsvorgang ist damit abgeschlossen. Ginny kann nun entscheiden, ob sie die Dateien kaufen möchte (siehe Abb.16).

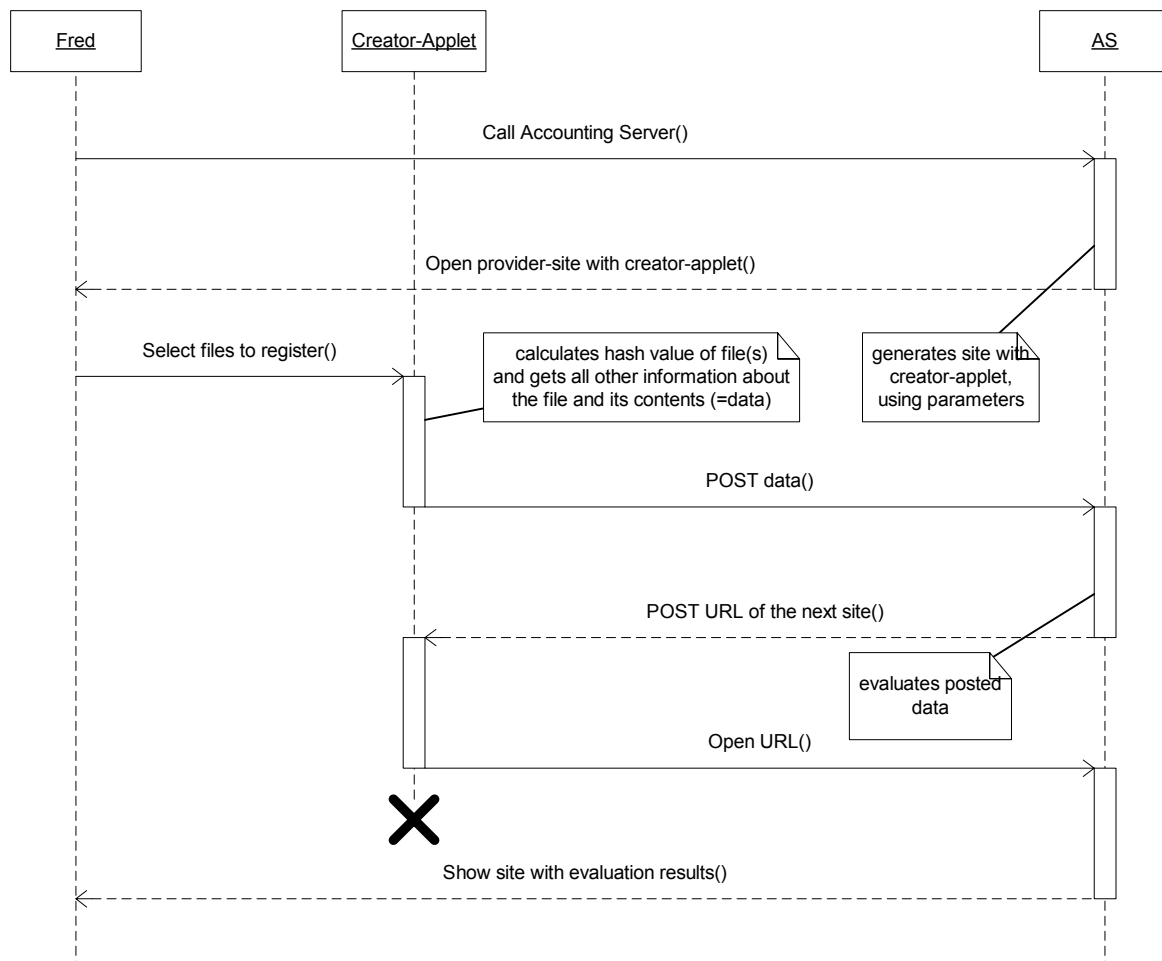


Abbildung 16: Sequenzdiagramm - Datei überprüfen

Der oben dargestellte Vorgang kann alternativ auch ohne Applet durchgeführt werden. Ginny gibt dazu die zu prüfende Transaktionsnummer in ein HTML-Formular ein und erhält als Antwort eine Web-Seite mit Informationen über die Datei.

4.2.3 Dateien umbenennen (Renamer-Applet)

Nachdem Ginny für die Datei(en) bezahlt hat (bzw. Fred neue Dateien registriert hat), generiert der AS eine neue Seite, welche als Parameter die lokalen Pfade zu den Dateien (welche zuvor vom Creator-Applet bzw. vom Checker-Applet zum Server gesendet wurden) und die neuen Dateinamen (Quittungen) enthält (siehe Abb.16). Hierbei muss der Server sicherstellen, dass der Aufruf aus einer gültigen Session, z.B. nach einem Bezahlvorgang, kommt (siehe Abb.17).

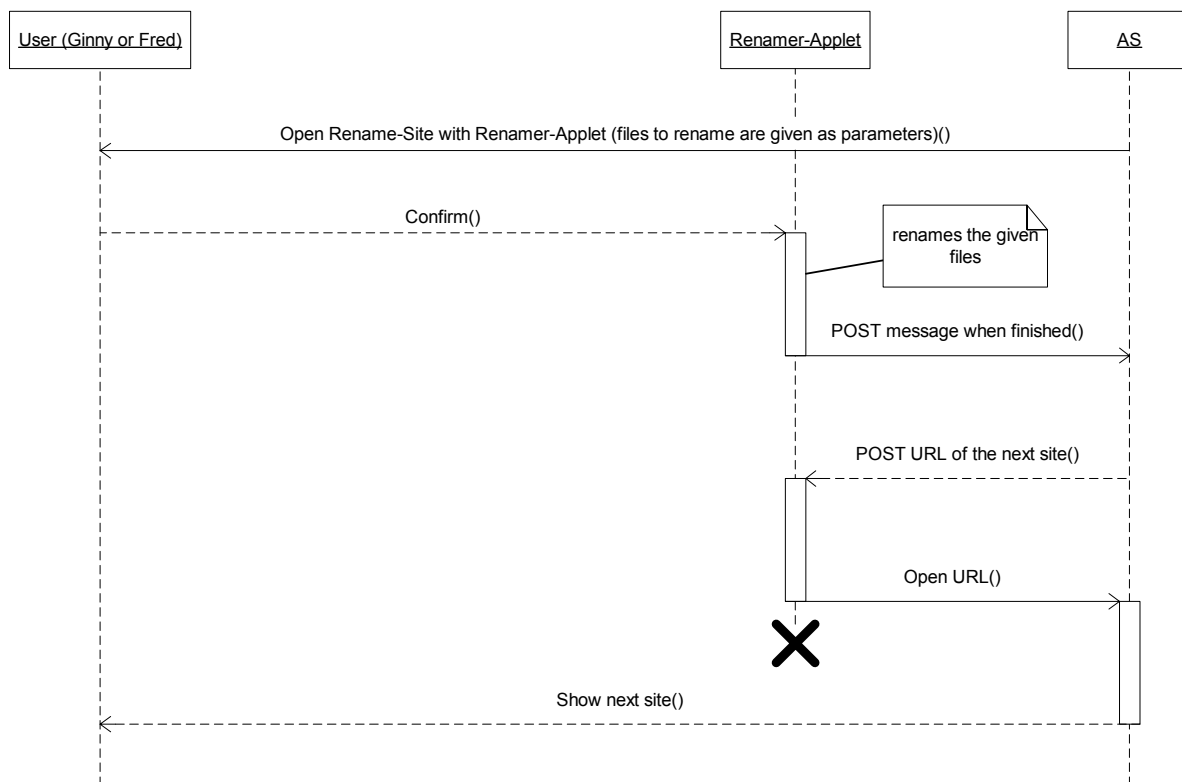


Abbildung 17: Sequenzdiagramm - Dateien umbenennen

Am einfachsten kann das Servlet dem Applet Parameter übergeben, indem es in der HTML-Seite (bzw. JSP-Seite), die das Applet aufruft, das „Param“-Tag dynamisch aufbaut. Diese Parameter können dann vom Applet beim Start ausgelesen und weiter verarbeitet werden. Auf diese Weise werden nicht nur die Informationen für die Umbenennung der Dateien, sondern auch Parameter zur Beeinflussung der visuellen Gestaltung der Applets und zur sprachlichen Anpassung übergeben (siehe *Anhang A*).

Beispiel:

```
<!-- file settings IE -->
  <PARAM NAME = "files" VALUE ="2">
  <PARAM NAME = "old0" VALUE =" D:\DePhazz [Death By Chocolate]\Track03.mp3">
  <PARAM NAME = "new0"
    VALUE =" D:\DePhazz [Death By Chocolate] Dephazz – Online4fo0815.mp3">
  <PARAM NAME = "old1" VALUE =" D:\Glashaus [Glashaus]\Track05.mp3">
  <PARAM NAME = "new1" VALUE =" D:\Glashaus [Glashaus]\ Glashaus – Weckruf4fo0816.mp3">
```

Der Wert des Parameters "files" gibt die Anzahl der umzubenennenden Dateien an. Um die Zuordnung zwischen den alten Dateien und den neuen Dateinamen vorzunehmen, werden Integerwerte an die Zeichenketten „old“ (lokaler Pfad zur Datei) und „new“ (lokaler Pfad zur Datei mit dem neuen Dateinamen) angehängt. Die Parameter werden vom Renamer-Applet eingelesen und verarbeitet. Die umzubenennenden Dateien werden vom Renamer-Applet in einer Liste dargestellt. Nachdem der Benutzer (Ginny oder Fred) den Vorgang bestätigt haben, werden die Dateien umbenannt.

Wenn alle Dateien ordnungsgemäß umbenannt werden konnten, wird eine Message per POST an den Server gesendet. Der Server sendet daraufhin eine neue URL für die nächste Seite, welche vom Renamer-Applet geöffnet wird. Der Vorgang des Umbenennens ist damit abgeschlossen.

Der Vorgang des Umbenennens kann alternativ auch ohne das entsprechende Applet durchgeführt werden, indem eine gekaufte (bzw. registrierte) Datei manuell umbenannt wird. Die nötige Information hierzu (Quittung) wird nach dem Kaufvorgang auf der Web-Seite angezeigt.

5 Zusammenfassung und Ausblick

Zusammenfassend sind folgende Punkte festzustellen: Die Kombination von leistungsfähigen Kompressionsverfahren für Musik- und Videodaten und die Verbreitung von File-Sharing-Systemen hat der Content-Industrie vor Augen geführt, dass eine große Anzahl von Konsumenten nicht gewillt ist, für beliebige Massenware übertriebene Preise zu bezahlen. Die Content-Industrie weigert sich zu erkennen und zu akzeptieren, dass ihr bisheriges Geschäftsmodell nicht mehr funktioniert und modifiziert werden muss, um überhaupt weiter bestehen zu können. Dem entsprechend werden auch keine neuen Geschäftsmodelle entwickelt, sondern es wird versucht mit restriktiven Maßnahmen (z.B. Kopierschutzmechanismen) und Prozessen gegen Tauschbörsen (und demnächst auch gegen einzelne P2P-User?) die Kontrolle über die Inhalte zurück zu erlangen. Die große Hoffnung der Industrie liegt in der flächendeckenden Einführung von DRM-Systemen, obwohl diese wohl eher Konsumenten verschrecken werden und die Erfolgchancen solcher Systeme aus technischer und ökonomischer Sicht gering sind. Auch die bisherigen Online-Angebote der Major-Labels können aufgrund ihrer Preisgestaltung nur als Versuch angesehen werden, das alte Offline-Geschäftsmodell in die Vertriebswelt der virtuellen Waren hinüber zu retten. Aus dem Bewusstsein dieser Situation heraus, wurde das Konzept des „Potato Systems“ entwickelt, dessen Grundprinzip es ist, dem Konsumenten Anreize zu bieten für virtuelle Waren zu bezahlen, jedoch ohne den Konsumenten zu kontrollieren oder unter Druck zu setzen. Die Zahlungsbereitschaft soll viel mehr aus dem Bedürfnis heraus entstehen, zum einen selbst von der Verbreitung der virtuellen Waren finanziell zu profitieren und zum anderen in den Genuss von Mehrwerten zu gelangen, die nur die „Potato System“ - Community bieten kann.

Das „Potato System“ besteht aus mehreren Komponenten. Die Konzeption und prototypenhafte Implementierung der Client-Komponenten ist die Aufgabe dieser Diplomarbeit. Ausgehend von den verschiedenen Anwendungsfällen, wurden zwei Clientkomponenten konzipiert: der Creator, um Provider bei Registrierung von Dateien zu unterstützen und der Redister, um potentiellen Käufern eine einfache und sichere Möglichkeit zu bieten, die Integrität und Authentizität der zu kaufenden Datei zu prüfen. Bei der Realisierung der Komponenten galt es, folgende Anforderungen zu berücksichtigen: Benutzerfreundlichkeit, Portabilität und Sicherheit. Die Verwendung von (signierten) Java-Applets bietet hierbei die beste Lösung. Die Funktionen der konzipierten Komponenten werden durch drei unabhängige Applets umgesetzt: Creator-Applet, Checker-Applet und Renamer-Applet. Eine Aussage darüber, ob die Clientkomponenten in dieser Form und das System insgesamt von den Konsumenten akzeptiert werden, kann erst nach der Auswertung des geplanten Feldversuchs getroffen werden.

Als Erweiterung für das „Potato-System“ wäre die Integration von P2P-Funktionalitäten in die Client-Komponenten denkbar. Dies könnte sowohl in koordinierter Form (der AS könnte als Index-Server fungieren) oder in dezentraler Form realisiert werden.

Anhang A: Init-Parameter für die Appletkonfiguration

Parameter Name	Bedeutung	Beispiel (IE)
fileNotFoundError	„Datei nicht gefunden“ - Meldung	<PARAM NAME = "fileNotFoundError" VALUE = "Datei nicht gefunden: ">
filesLeftMessage	„Noch nicht alle Dateien umbenannt“ - Meldung	<PARAM NAME = "filesLeftMessage" VALUE = "Es sind noch nicht alle Dateien umbenannt.">
successMessage	„Vorgang abgeschlossen“ - Meldung	<PARAM NAME = "successMessage" VALUE = "Datei umbenannt:">
okButtonText	„OK Button“ - Text	<PARAM NAME = "okButtonText" VALUE = "Umbenennen">
okButtonAlt	„OK Button“ - Alternativtext	<PARAM NAME = "okButtonAlt" VALUE = "Schließen">
background	Hintergrundfarbe des Applets	<PARAM NAME = "background" VALUE = "998765">
foreground	Vordergrundfarbe des Applets	<PARAM NAME = "foreground" VALUE = "000078">
files	Anzahl der umzubenennenden Dateien	<PARAM NAME = "files" VALUE = "3">
old	Der alte Dateiname (durch den Integerwert erfolgt die Zuordnung)	<PARAM NAME = "old0" VALUE = "c:\d1.txt">
new	Der neue Dateiname (durch den Integerwert erfolgt die Zuordnung)	<PARAM NAME = "new0" VALUE = "c:\d14fo0001.txt">

Tabelle 15: Init-Parameter (Renamer-Applet)

Parameter Name	Bedeutung	Beispiel (IE)
okButtonText	„OK Button“ - Text	<PARAM NAME = "okButtonText" VALUE = "Daten senden">
removeButtonText	„Remove Button“ - Text	<PARAM NAME = "removeButtonText" VALUE = "Datei entfernen">
addButtonText	„Add File Button“ - Text	<PARAM NAME = "addButtonText" VALUE = "Datei öffnen">
clearButtonText	„Clear List Button“ - Text	<PARAM NAME = "clearButtonText" VALUE = "Liste löschen">
emptyListMessage	„Liste ist leer“ - Meldung	<PARAM NAME = "emptyListMessage" VALUE = "Die Liste ist leer.">
sendDataMessage	„Daten werden gesendet“ - Meldung	<PARAM NAME = "sendDataMessage" VALUE = "Die Daten werden gesendet. Bitte warten...">
dlgYes	Senden-Dialog: „Ja“	<PARAM NAME = "dlgYes" VALUE = "Ja!">
dlgNo	Senden-Dialog: „Nein“	<PARAM NAME = "dlgNo" VALUE = "Noch nicht!">
dlgQuestion	Senden-Dialog: Frage	<PARAM NAME = "dlgQuestion" VALUE = "Die Daten jetzt senden?">
background	Hintergrundfarbe des Applets	<PARAM NAME = "background" VALUE = "998765">
foreground	Vordergrundfarbe des Applets	<PARAM NAME = "foreground" VALUE = "000078">

Tabelle 16: Init-Parameter (Checker-Applet)

Parameter Name	Bedeutung	Beispiel (IE)
okButtonText	„OK Button“ - Text	<PARAM NAME = "okButtonText" VALUE = "Daten senden">
removeButtonText	“Remove Button” - Text	<PARAM NAME = "removeButtonText" VALUE = "Datei entfernen">
addButtonText	“Add File Button” - Text	<PARAM NAME = "addButtonText" VALUE = "Datei öffnen">
clearButtonText	“Clear List Button” - Text	<PARAM NAME = "clearButtonText" VALUE = "Liste löschen">
applyButtonText	„Einstellungen übernehmen Button“ - Text	<PARAM NAME = "applyButtonText" VALUE = "Einstellungen übernehmen">
deleteBox	Temporäre Datei löschen	<PARAM NAME = "deleteBox" VALUE = "Temporäre Datei löschen">
settingsMessage	„Einstellungen wurden gespeichert“ – Meldung	<PARAM NAME = "settingsMessage" VALUE = "Einstellungen gespeichert">
emptyListMessage	“Liste ist leer” - Meldung	<PARAM NAME = "emptyListMessage" VALUE = "Die Liste ist leer.">
sendDataMessage	“Daten werden gesendet” - Meldung	<PARAM NAME = "sendDataMessage" VALUE = "Die Daten werden gesendet. Bitte warten...">
dlgYes	Senden-Dialog: “Ja”	<PARAM NAME = "dlgYes" VALUE = "Ja!">
dlgNo	Senden-Dialog: “Nein”	<PARAM NAME = "dlgNo" VALUE = "Noch nicht!">
dlgQuestion	Senden-Dialog: Frage	<PARAM NAME = "dlgQuestion" VALUE = "Die Daten jetzt senden?">
Background	Hintergrundfarbe des Applets	<PARAM NAME = "background" VALUE = "998765">
Foreground	Vordergrundfarbe des Applets	<PARAM NAME = "foreground" VALUE = "000078">
priceLabel	„Preis Label“ - Text	<PARAM NAME = "priceLabel" VALUE = "Preis (Cent)">
cLabel	„Provision Label“ - Text	<PARAM NAME = "cLabel" VALUE = "Provision (%)">
msgInput	„Falsche Eingabe“ - Meldung	<PARAM NAME = "msgInput" VALUE = "Falsche Eingabe!">
priceDefault	Preis (Default: 100 Cent)	<PARAM NAME = "priceDefault" VALUE = "100">
cDefault	Provision (Default: 25%)	<PARAM NAME = "cDefault" VALUE = "25">

Tabelle 17: Init-Parameter (Creator-Applet)

Anhang B: Struktur der XML-Dateien

a) Checker-Applet

Parameter	Bedeutung
<SenderID>	Name des Applets
<Name>	Original-Dateiname (ohne Pfad)
<LocalPath>	Original-Dateiname (mit Pfad)
<SizeByte>	Dateigröße (Byte)
<LastModified>	Datum der letzten Dateiänderung
<HashValue>	Hash-Wert
<MessageDigest>	Typ des verwendeten MD-Algorithmus

Tabelle 18: XML-Parameter (Checker-Applet)

Struktur:

```
<?xml version="1.0" ?>
<!-- Sender -->
<Sender>
  <SenderID></SenderID>
</Sender>
<!-- File Info -->
<File>
  <Name></Name>
  <LocalPath></LocalPath>
  <SizeByte></SizeByte>
  <LastModified></LastModified>
  <HashValue></HashValue>
  <MessageDigest></MessageDigest>
</File>
```

b) Creator-Applet

Parameter	Bedeutung
<SenderID>	Name des Applets
<Name>	Original-Dateiname (ohne Pfad)
<LocalPath>	Original-Dateiname (mit Pfad)
<SizeByte>	Dateigröße (Byte)
<LastModified>	Datum der letzten Dateiänderung
<HashValue>	Hash-Wert
<MessageDigest>	Typ des verwendeten MD-Algorithmus
<Price>	Preis der virtuellen Ware in Cent
<Commission>	Provision für den Wiederverkäufer in Prozent
<Url>	Link zur Datei
<Description>	Beschreibung
<SampleType>	Art des Samples (z.B. Sound)
<FileName> (Sample)	Dateiname des Samples (ohne Pfad)
<FpType>	Art des semantischen Fingerabdrucks (z.B. Audio-ID)
<FileName> (Semantic Fingerprint)	Dateiname des semantischen Fingerabdrucks (ohne Pfad)

Tabelle 19: XML-Parameter (Creator-Applet)

Struktur:

```
<?xml version="1.0" ?>
<!-- Sender -->
<Sender>
  <SenderID></SenderID>
</Sender>
<!-- File Info -->
<File>
  <Name></Name>
  <LocalPath></LocalPath>
  <SizeByte></SizeByte>
  <LastModified></LastModified>
  <HashValue></HashValue>
  <MessageDigest></MessageDigest>
  <Price></Price>
  <Commission></Commission>
  <Url></Url>

  <!-- Contents Info -->
  <Contents>
    <Description></Description>
    <Sample>
      <SampleType></SampleType>
      <FileName> </FileName>
    </Sample>
    <FingerPrint>
      <FpType></FpType>
      <FileName></FileName>
    </FingerPrint>
  </Contents>
</File>
```

Quellenverzeichnis

- [4foOrg-02] URL: <http://www.4friendsonly.org>
- [AcroReader1-99] Heise Online News (16.11.1999): *PDF-Dateien werden käuflich*
URL: <http://www.heise.de/newsticker/data/jr-16.11.99-000/>
- [AcroReader2-01] Rötzer, Florian: *Russischer Programmierer frei*
URL: <http://www.heise.de/tp/deutsch/inhalt/te/11340/1.html>
- [AudioID-02] Fraunhofer IIS - *AudioID*
URL: <http://www.emt.iis.fhg.de/produkte/audioid/>
- [Balzert-96] Balzert, Helmut: *Lehrbuch der Softwaretechnik*, Spektrum Akademischer Verlag
1996, ISBN 3-8274-0042-2
- [Berberich-01] Berberich, Eric: *Hashfunktionen in der Kryptographie*
URL: <http://www-krypt.cs.uni-sb.de/lehre/veranstaltungen/ss2001/proseminar/vortraege/berberich/>
- [CDMedia-02] URL: http://www.cdmediaworld.com/hardware/cdrom/cd_protections.shtml
- [ChipK2A-02] *Audio-Kopierschutz*
URL: http://www.chip.de/praxis_wissen/praxis_wissen_8725919.html
- [CTCDS-00] *Audio-CD-Kopierschutz verärgert Kunden und Händler*
CT 4/2000
- [CTK2A-01] Peeck, Klaus: *Musik hinter Gittern : Kopierschutz für Audio-CDs*
CT 15/2001
- [DeCSS1-99] Heise Online News (04.11.1999): *15-Jähriger knackte DVD-Video*
URL: <http://www.heise.de/newsticker/data/ghi-04.11.99-000/>
- [DeCSS2-99] Heise Online News (01.11.1999): *Verschlüsselung von DVD-Video geknackt*
URL: <http://www.heise.de/newsticker/data/ghi-01.11.99-000/>
- [DeCSS3-02] Heise Online News (11.01.2002): *Anklage gegen DVD-Hacker Johansen*
URL: <http://www.heise.de/newsticker/data/ghi-11.01.02-000/>
- [DevGuide-02] Java Plug-in Developer Guide
URL:
http://java.sun.com/j2se/1.4/docs/guide/plugin/developer_guide/contents.html
- [DittWohl-02] Dittmann, Wohlmacher: *Aspekte der Sicherheit multimedialer Daten und Anwendungen mittels Kryptographie und digitaler Wasserzeichentechniken*
URL: <http://www.darmstadt.gmd.de/~dittmann/mgdbg/gi.doc>
- [DMCA] URL: <http://www.loc.gov/copyright/title17/92appv.html>
- [FederrathDRM-02] Federrath, Hannes: *Scientific evaluation of DRM systems (2002)*
URL: <http://www.inf.tu-dresden.de/~hf2/drm/>
- [FFinance-01] Frankfurter Finance (Dezember 2001): *Branchen und Märkte: Die Musikindustrie – Mangelnde Innovationen einer kreativen Branche*
URL: <http://www.frankfurterfinance.de/archiv/2001/dez2001.htm>
- [FrasP2P-02] Frascaria, Kareen: *Peer-to-Peer: Die Erneuerung des verteilten Rechnens*
URL: <http://techupdate.zdnet.de>

- [GebhJPK-02] Gebhardt, Gerd : Jahrespressekonferenz der IFPI Deutschland 2002, URL:
<http://www.ifpi.de>
- [Gemstar1-00] Heise Online News (25.11.2000): *Flurbereinigung bei E-Books*
URL: <http://www.heise.de/newsticker/data/jk-25.11.00-001/>
- [Gemstar2-00] Heise Online News (19.10.2000): *Buchmesse: Die zweite E-Book-Generation*
URL: <http://www.heise.de/newsticker/data/jr-19.10.00-000/>
- [Glassbook-00] Heise Online News (31.03.2000): *Stephen Kings E-Book sorgt für Wirbel*
URL: <http://www.heise.de/newsticker/data/jr-31.03.00-000/>
- [GriNütz-02] Grimm, Nützel:
A friendly P2P file sharing system with profit but without copy protection
URL: <http://www.4friendsonly.org>
- [H2O4M-02] *Digital Watermarking*
URL: <http://www.ipi.fhg.de/merit/projects/index.html>
- [Hansen-02] Hansen, Sven: *Kommerzielle Musikangebote im Netz*
CT (Magazin für Computertechnik), 16/2002
- [Heinrich-94] Heinrich, Jürgen: *Medienökonomie*,
Westdeutscher Verlag, 1994, ISBN 3-531-12636-9
- [HeiseDRM-02] Heise Online News (30.01.2002):
Schlüsseltechniken beim Kopierschutz sind noch nicht ausgereift
URL: <http://www.heise.de/newsticker/data/jk-30.01.02-001/>
- [HeiseK2A-02] Heise Online News (26.05.2002):
Musikindustrie reagiert auf Filzstift-Hack gegen Audio-Kopierschutz
URL: <http://www.heise.de/newsticker/data/lab-26.05.02-001/>
- [Himmelein-02] Himmelein, Gerald: *Der digitale Knebel*
CT (Magazin für Computertechnik), 15/2002
- [HSJH-01] Hasselbach, Jens: *Verbreitete Netzwerkprotokolle als beliebte Angriffsziele I*
Hauptseminar im WS 2000-2001, TU Ilmenau
URL: <http://www.tu-ilmenau.de/~hasselb/download/hshcb.zip>
- [IEC-908] URL: <http://www.ee.washington.edu/conselec/CE/kuhn/cdmulti/95x7/iec908.htm>
- [IFPI-02] Bundesverband der Phonographischen Wirtschaft e.V., Jahrbuch 2002 der
Phonographischen Wirtschaft
Josef-Keller-Verlag Starnberg, ISBN 3 7808 0186 8
URL : <http://www.ifpi.de>
- [Java2-99] Steyer Ralph: *Java2*
Markt&Technik 1999, ISBN 3-453-16822-4
- [JavaMag-00] Ebbart, Dieter: *Kommunikation zwischen Servlet und Applet*
URL: <http://www.sdm.de/dt/tec/pub/art/art/kommservapp/kommservapp.htm>
- [JavaStyle-00] *The Elements of Java Style*, Cambridge University Press 2000, ISBN 0-521-777682
- [Krauß-02] Krauss, Holger: *Konzeption und Realisierung der Server-Komponente für ein P2P-
File-Sharing-System, bei dem die User am Umsatz beteiligt sind*
Diplomarbeit, Technische Universität Ilmenau

- [Krempf-02] Krempf, Stefan: *Napsterisierung vs. Venterisierung*
URL: <http://www.heise.de/tp/deutsch/inhalt/te/11475/1.html>
- [KrempfDRM-00] Krempf, Stefan: *Zurück in die Zukunft: Das neue Internet ist ganz das alte*
URL: <http://www.ct.heise.de/tp/deutsch/inhalt/te/8398/1.html>
- [KrempfFIA-01] Krempf, Stefan: Der Musikindustrie droht ein neues Fiasko im Internet
URL: <http://www.heise.de/tp/deutsch/inhalt/musik/11338/1.html>
- [KrempfURH-02] Krempf, Stefan: *Geschützte Kopiersperren*
CT 8/2002
- [KrögerMM-99] Kröger, Sabine: *Musik und Markt – Eine volkswirtschaftliche Betrachtung*
URL: http://www.wiwi.hu-berlin.de/~skroeger/musikseminar/Musik_1999/musikmarkt.html
- [LaGrande-02] Heise Online News (13.09.2002):
IDF: Mehr Details zu Intels Hardware-Verschlüsselung
URL: <http://www.heise.de/newsticker/data/ciw-13.09.02-001/>
- [Leitner-00] Felix von Leitner: *MP3 selbst gestrickt*
CT (Magazin für Computertechnik), 3/2000
- [Mack-99] Mack, Holger: *Sicherheit in Java und ActiveX*
Erschienen in: Fox, D.; Horster, P.:
Datenschutz und Datensicherheit – DuD, Verlag Vieweg, Braunschweig 1999
- [Mindprod-02] URL: <http://mindprod.com/>
- [MLM] World Federation of Direct Selling Associations
URL: http://www.wfdsa.org/legal_reg/ge_ppaper3.asp
- [MöllerTAU-00] Möller, Erik: *Schöner tauschen* (1 bis 4)
URL: <http://www.heise.de/tp/deutsch/inhalt/te/8304/1.html>
- [MP3World] *Kopierschutztechniken*
URL: <http://www.mp3-world.net/d/workshop/copyprotected/index.shtml>
- [MSDRM-2-01] Heise Online News (19.10.2001): *Microsofts Digital Rights Management geknackt*
URL: <http://www.heise.de/newsticker/data/vza-19.10.01-000/>
- [MSJVM-02] Heise Online News (20.09.2002):
Microsoft warnt vor kritischen Sicherheitslücken der eigenen Java-VM
URL: <http://www.heise.de/newsticker/data/kav-20.09.02-000/>
- [MSReader1-00] Heise Online News (24.02.2000): *Microsoft zeigt E-Book-Reader*
URL: <http://www.heise.de/newsticker/data/jr-24.02.00-000/>
- [MSReader2-00] Heise Online News (08.08.2000): *Barnes & Noble und Microsoft bringen E-Books*
URL: <http://www.heise.de/newsticker/data/jr-08.08.00-000/>
- [MSReader3-01] Heise Online News (30.08.2001): *Kopierschutz von Microsoft Reader geknackt*
URL: <http://www.heise.de/newsticker/data/daa-30.08.01-000/>
- [MusicNet-02] Heise Online News (05.06.2002):
RealNetworks schimpft über Online-Strategien der Musikindustrie
URL: <http://www.heise.de/newsticker/data/jk-05.06.02-004/>
- [Musik-02] Heise Online News (16.04.2002): *Musikindustrie sieht Piraten am Umsatz nagen*
URL: <http://www.heise.de/newsticker/data/anw-16.04.02-004/>

- [Napster] Heise Online News: *Verschiedene Artikel zum Thema „Napster“ (2000-2002)*
URL: <http://www.heise.de/>
- [OOA-00] Stollenmaier, Michael: *Objektorientierte Analyse*, URL:
http://public.rz.fh-wolfenbuettel.de/~geiling/swt-abgabe/OOA_Abhandlung_SWT00.pdf
- [OOSWE-98] Oestereich, Bernd: *Objektorientierte SW-Entwicklung – Analyse und Design mit der UML*, Oldenbourg Verlag 1998, ISBN 3-486-24787-5
- [PCWeltFilm-02] *Raubkopien verursachen Milliardenverluste für Filmindustrie*
URL: <http://www.pcwelt.de/news/internet/23142/>
- [Picot-97] Picot, Bortenlänger, Röhl:
Organization of electronic markets: contributions from the New Institutional Economics, The Information Society, Vol.13, Nr.1
- [RötzerCOC-00] Rötzer, Florian:
Schwere Verluste für die Buchverlage und die Musikindustrie prophezeit
URL: <http://www.heise.de/tp/deutsch/inhalt/te/8759/1.html>
- [SafeAudio-01] Heise Online News (01.08.2001): *CD-Kopierschutz von Macrovision ausgetrickst*
URL: <http://www.heise.de/newsticker/data/vza-01.08.01-000/>
- [SafeAudioV3-02] Heise Online News (23.01.2002): *Macrovision bringt neue Audiokopiersperre*
URL: <http://www.heise.de/newsticker/data/vza-23.01.02-000/>
- [Scour] Heise Online News: *Verschiedene Artikel zum Thema „Scour“*
URL: <http://www.heise.de/>
- [SDMI-Hack-01] Heise Online News (22.04.2001): *Brisante Details zum SDMI-Hack*
URL: <http://www.heise.de/newsticker/data/vza-22.04.01-000/>
- [SecuringJava-99] URL: <http://www.securingjava.com/>
- [ShirkyP2P-00] Shirky, Clay: *What is P2P and what isn't?*
URL: <http://www.oreillynet.com>
- [Signing-99] URL: <http://www.suitable.com/>
- [TecMG-00] *MusicGuard*
URL: <http://www.tecchannel.de/multimedia/241/10.html>
- [theCounter-02] URL: <http://www.thecounter.com/>
- [WobstKRYPT-98] Wobst, Reinhard: *Abenteuer Kryptologie (Methoden, Risiken und Nutzen der Datenverschlüsselung)*
Addison-Wesley-Longman, 1998, ISBN 3-8273-1413-5
- [ZoierP2P-01] Zoier, Markus: *Peer-to-Peer Tauschbörsen*.
URL: http://www.iicm.edu/research/seminars/ws_01/zoier-peer2peer.pdf
- [Zota-01] Zota, Volker: *Raubkopierte Filme im Internet – Die Filmindustrie hat ein Problem*
CT (Magazin für Computertechnik), 3/2001
- [Zota-02] Zota, Volker: *DivX im Griff*
CT (Magazin für Computertechnik), 18/2002

Abbildungsverzeichnis

Abbildung 1: Die Überführung in den virtuellen Zustand	9
Abbildung 2: Umsatzverluste der deutschen Musikindustrie durch Raubkopien (in Mio. Euro) [IFPI2002].....	12
Abbildung 3: Koordiniertes P2P-System	34
Abbildung 4: Dezentrales P2P-System	35
Abbildung 5: "Gnutella"-Netz mit Reflektoren.....	36
Abbildung 6: „Potato System“ Beispielszenario	43
Abbildung 7: Transaktionsnummern und Dateinamen im „Potato System“	45
Abbildung 8: Käuferempfehlungssystem.....	46
Abbildung 9: Käuferempfehlungssystem mit P2P	47
Abbildung 10: Die Aufgaben der „Creator“-Komponente.....	55
Abbildung 11: Die Aufgaben der „Redister“-Komponente	56
Abbildung 12: Klassendiagramme der Client-Komponenten (Fachkonzept)	60
Abbildung 13: Browser-Statistik.....	63
Abbildung 14: Screenshot - Creator-Applet.....	71
Abbildung 15: Sequenzdiagramm - Datei registrieren	72
Abbildung 16: Sequenzdiagramm - Datei überprüfen.....	73
Abbildung 17: Sequenzdiagramm - Dateien umbenennen	74

Tabellenverzeichnis

Tabelle 1: Weltweite CD-Verkäufe (1983 – 1998) [KrögerMM-99].....	16
Tabelle 2: Sicherheitsziele und korrespondierende Techniken	23
Tabelle 3: Anwendungsfall „Als Provider registrieren“	48
Tabelle 4: Anwendungsfall „Als Vertriebspartner registrieren“	49
Tabelle 5: Anwendungsfall „Eine Datei registrieren“	49
Tabelle 6: Anwendungsfall „Eine Datei überprüfen“.....	50
Tabelle 7: Anwendungsfall „Eine Datei kaufen“	51
Tabelle 8: Anwendungsfall „File Sharing“	52
Tabelle 9: Anwendungsfall „Abruf eines Inhaltes – on-demand“.....	53
Tabelle 10: Anwendungsfall „Direkt-Download“.....	53
Tabelle 11: Die Funktionalitäten der Client-Komponenten	57
Tabelle 12: Browserübersicht (Unterstützung für Java).....	64
Tabelle 13: Aufruf des Java Plug-ins	66
Tabelle 14: Systemeigenschaften	67
Tabelle 15: Init-Parameter (Renamer-Applet)	77
Tabelle 16: Init-Parameter (Checker-Applet).....	77
Tabelle 17: Init-Parameter (Creator-Applet).....	78
Tabelle 18: XML-Parameter (Checker-Applet)	79
Tabelle 19: XML-Parameter (Creator-Applet).....	79