Konzeption und Implementation eines WAN basierten Publikationssystems mit Fokus auf DRM & URM Komponenten

Bachelorarbeit

von Tobias Müller

(01/477652)

Universität Konstanz

Fachbereich Informatik & Informationswissenschaften

Erstgutachter: Prof. Dr. Rainer Kuhlen

Zweitgutachter: Prof. Dr. Oliver Deussen

Februar 2004

GEWIDMET MEINEN ELTERN
ROLAND UND VERA MÜLLER.

Kurzreferat

Um von potentiell nutzbringender Information profitieren zu können, muss diese sowohl grundlegend verfügbar und somit auffindbar, als auch effizient recherchierbar sein. Die Einordnung von Informationen im wissenschaftlichen Umfeld in entsprechende Kategorien, fällt aufgrund der Menge der Veröffentlichungen im traditionellen Wege schwer. Immer mehr Veröffentlichungen werden deshalb online in einem mehr oder weniger aufwendigen Publikationssystem veröffentlich. Diese Arbeit bietet einen Überblick über Onlinepublikationssysteme, über das mit dem Schutz der Urheberrechte im digitalen Raum unweigerlich verbundene Digital Rights Management, sowie über das als Gegengewicht zum DRM angesehene User Rights Management.

In Ergänzung der theoretischen Einführungen in jene Gebiete, beschreibt diese Arbeit den innerhalb eines Projektpraktikums entstandenen Prototypen des Onlinepublikationssystems evobo – *the evolution of book*. Im Rahmen dieser Arbeit wird dabei sowohl auf das im Hintergrund liegende Framework im Allgemeinen, als auch auf die Implementation von DRM und URM im Speziellen eingegangen.

Abstract

To be of benefit, potentially valuable information needs to be available and reachable on the one side and efficiently researchable on the other. The classification of information in corresponding classes within the scientific community is hard, based on the overwhelming amount. Based on this problem more and more publications are published online in a more or less elaborated online publishing system. This work presents an overview of online publishing systems, of Digital Rights Management which is inevitably connected to the protection of copyright in a digital environment and of User Rights Management which on the other side is regarded as counterbalance to DRM.

In addition to a theoretical introduction in those areas, this work presents the prototype of the online publishing system evobo – *the evolution of book* – which was implemented during the practical part. In context of this work the underlying framework will be discussed in general and the implementation of DRM and URM in specific.

Inhaltsverzeichnis

3. Digital Rights Management

4. User Rights Management

Abbildungsverzeichnis

Kapitel 1

Einleitung

1.1 Überblick

Die vorliegende Bachelorarbeit besteht aus vier Teilen, wovon die ersten drei die Bestandteile dieser Arbeit – Onlinepublikation, DRM[1] und URM[2] – theoretisch aufarbeiten und der letzte Teil die praktische Implementation des Prototypen EVOBO auszugsweise vorstellt.

Im ersten Teil werden, unabhängig von der Problematik des Geistigen Eigentums an sich, die Möglichkeiten und Problembereiche von Onlinepublikationssystemen sowie deren Einsatzmöglichkeiten diskutiert. Ergänzend werden Anwendungen von Onlinepublikationssystemen vorgestellt, die jene diskutierten Möglichkeiten in unterschiedlichem Maß implementieren.

Im zweiten und dritten Teil führt diese Arbeit in den Schutz des Geistigen Eigentums (Intellectual Property) innerhalb digitaler Systeme zum einen, in den Schutz der Rechte des Nutzers zum anderen ein. Während der Schutz des Geistigen Eigentums über DRM bereits hinlänglich diskutiert wurde, vernachlässigte die Diskussion im Hinblick auf stärkeren Schutz meist die Rechte des Nutzers. Im Rahmen des URM sollen diese Rechte letztlich ebenso wie das Urheberrecht (Copyright) in die digitalen Systeme eingeführt werden. Die dazu notwendigen Ideen und Umsetzungen innerhalb der Rechteverwertung & –verwaltung werden diskutiert.

---

[1] Digital Rights Management

[2] User Rights Management

Der abschließende vierte Teil beschäftigt sich mit der praktischen Realisierung eines DRM & URM Systems, das die Grundsätze dieser Arbeit verwirklicht. Im Hinblick auf die unter 1.3 aufgeführte Aufgabenstellung werden hier neben der grundlegenden Systemarchitektur auch die angewandte Verschlüsselung und Speicherung beschrieben.

1.2 Motivation und Intention

Diese Arbeit ist durch unzählige Veröffentlichungen motiviert, die sich zum einen mit dem Publizieren von wissenschaftlichen Arbeiten in einem Onlinemedium beschäftigen, zum anderen aufzeigen wie Geistiges Eigentum über DRM geschützt werden kann. Insbesondere die Arbeiten von [Odl00] und [Law01] sind hierbei nennenswert. Sie beschäftigen sich mit der zukünftigen Kommunikation innerhalb der wissenschaftlichen Gemeinschaft und zeigen dabei den Trend auf, dass wissenschaftliche Arbeiten, um effektiv wahrgenommen zu werden, online publiziert werden müssen.

Während in den genannten Arbeiten jedoch von der freien Verfügbarkeit wissenschaftlicher Arbeiten ausgegangen wird, kann eine solche Ausgangssituation nicht generalisiert werden, da bereits mit der Veröffentlichung das Problem der Wahrung der Urheberrechte entsteht. Entsprechend können Veröffentlichungen zwar online erfolgen, es muss dabei jedoch auf den Schutz der mit dem Werk verbundenen Rechte geachtet werden. DRM zum Schutz der Rechte des Autors bzw. des Rechteinhabers kann mittlerweile als De-facto-Standard angesehen werden und ist prinzipiell als individualisiertes Kontroll- und Abrechnungsverfahren in elektronischen Umgebungen sinnvoll [Kuh02e].

DRM darf jedoch weder zur Kreativitätsbarriere verkommen, noch sollte DRM die Rechte des Nutzers beschneiden. Aus einem Vortrag Lessigs heraus wurde das Interesse des Autors an der Idee der „Free Society“ [Les02] geweckt, die sich in Teilgebieten mit den Ideen des User Rights Managements, wie es von Prof. Dr. Kuhlen als Mittel, um nicht unverträgliche Gegensätze (Informationskriege) zwischen dem Interesse der Informationswirtschaft an Kontrolle über die Nutzung von Wissensobjekten und dem Interesse der Öffentlichkeit an deren möglichst freien Nutzung entstehen zu lassen [Kuh02e], propagiert wird.

In [Les02] führt Lessig vier aufeinander aufbauende Thesen an, die als Motivation für ein solches Gegengewicht gesehen werden können:

- *Creativity and innovation always builds on the past.*
- *The past always tries to control the creativity that builds upon it.*
- *Free societies enable the future by limiting this power of the past.*
- *Ours is less and less a free society.*

Heutige Kreativität und Innovation bauen auf Kreativität und Innovation der Vergangenheit auf. Bereits morgen ist heute Vergangenheit, entsprechend führt der zu rigide Schutz der Innovationen von Heute über das Mittel DRM zum Verlust eben jener Kreativität und Innovation in der Zukunft. Wobei zu bemerken ist, dass eine freie Gesellschaft diesen Verlust, sofern sie frei bleiben möchte, nicht zulässt.

Dies sind die drei Grundsteine der vorliegenden Bachelorarbeit und Motivation über das Gebiet der Onlinepublikation mit dem besonderen Fokus auf DRM & URM zu schreiben, deren Ziel es ist, dem Interessierten ein Hintergrund zu schaffen, der es ihm erlaubt die genutzten Techniken, die geführten Diskussionen und die angewandten Verfahren selbst einzuordnen.

1.3 Ausgangsvoraussetzungen und Vorgaben

Ausgangspunkt der praktischen Implementation waren während des Semesters abgehaltene Treffen, bei denen die Teammitglieder Ideen zu einem System zusammentrugen, das primär als Zugriffsmedium auf, in begrenztem Umfang geschützte, Wissensobjekte dienen sollte und bei dem auf den konsequenten Einsatz der Aspekte des URMs geachtet werden sollte.

Wie üblich gliedert sich das Gesamtprojekt einer Bachelorarbeit in zwei Teile, d.h. dieser individuellen Abschlussarbeit ging bereits ein teamorientierter, praktischer Teil voraus. Nach mehrmaligen Neuzusammensetzungen des Projektteams ergaben sich folgende praktischen Arbeitsbereiche:

① Grundlegende Systemkonzeption & –implementation,
d.h. Schaffung eines Systemkerns und Integration der DRM Schutzfunktionen

② Bereitstellung einer Präsentationskomponente, sowie Gestaltung eines abstrakten Mehrwert-Systems welches zur späteren Schaffung neuer Mehrwertkomponenten dient

③ Konzeption & Implementation von Mehrwertkomponenten, sowie Implementation der Anrechnung der Mehrwertschaffung

Diese drei Teilbereiche wurden unter den noch verbliebenen vier Teammitgliedern aufgeteilt, wobei festgelegt wurde, dass ein Teammitglied für die spätere Evaluation bzw. theoretische Einordnung sorgen sollte. Der Autor dieser Arbeit war für den unter ① genannten Teil der Implementation zuständig.

Auf technischer Ebene wurden freie[3] Softwarekomponenten eingesetzt, deren genaue Aufschlüsselung in Kapitel 5 folgt.

[3] Im Sinne der Definition der Free Software Foundation (FSF)

Kapitel 2

Onlinepublikationssystem

2.1 Begriffsdefinition

Eng gefasst muss der Begriffsbestandteil Publikationssystem für die nachfolgenden Ausführungen als nicht ausreichend angesehen werden, kann doch ein Publikationssystem als „*System zur Veröffentlichung von Informationen*“ [*WISSa*] angesehen werden. Wenngleich diese Definition sehr weit gefasst ist, wird sie weder den vorgestellten Systemen, noch den Wissensmanagement Ansätzen EVOBOS gerecht. Vielmehr stellen die nachfolgend vorgestellten Systeme wie z.B. CROSSREF [Wal02] zwar die Basis zur Referenzierbarkeit einer Veröffentlichung bereit, nicht jedoch die eigentliche Veröffentlichung. Wird jedoch der Terminus Veröffentlichung für ein Medium wie dem Internet mehr als ein „Zugänglich machen“, denn ein echtes „Veröffentlichen“ definiert, was zum aktuellen Zeitpunkt eher korrekt sein dürfte, wird aus einem System zur Referenzierung letztlich ein System zur Publikation, werden damit doch Veröffentlichungen durch Referenzieren zugänglich gemacht. Ergänzend bietet z.B. CrossRef erweiterte Recherchemöglichkeiten, die wiederum der Zugänglichkeit einer Veröffentlichung beitragen. Letztlich soll der Begriff Publikationssystem der Einfachheit und der Universalität halber in dieser Arbeit Verwendung finden, obwohl Systeme unter diesem Begriff subsumiert werden, denen die ursprüngliche Definition nicht gerecht wird.

Onlinepublikationssystem wird in dieser Arbeit entsprechend als ein System definiert, das innerhalb der Spannbreite eines Systems liegt, das zum reinen Zugang zu bereits anderweitig veröffentlichten Wissensobjekten dient, und dem System, das ein vollständiges Wissensmanagement unterstützt.

2.2 Online oder Unsichtbar

Mit diesem Titel (im original „Online or Invisible“) erschien 2001 ein Paper von S. Lawrence (→ [Law01]), das die Probleme der modernen wissenschaftlichen Veröffentlichungen aufzeigte – die Menge wissenschaftlicher Veröffentlichungen übersteigt die Möglichkeiten des Wissenschaftlers diese zu recherchieren, identifizieren und zu nutzen – und dadurch die These aufstellte, dass Onlineveröffentlichungen stärker von der Scientific Community wahrgenommen, respektive im Umkehrschluss auch öfters zitiert werden.

Nun wird in [Law01] ebenfalls beschrieben, dass eine höhere Zitationsquote nicht durch das Medium selbst entsteht, sondern aus den effektiveren und dadurch zeitsparenden Retrievalmöglichkeiten hervorgeht.

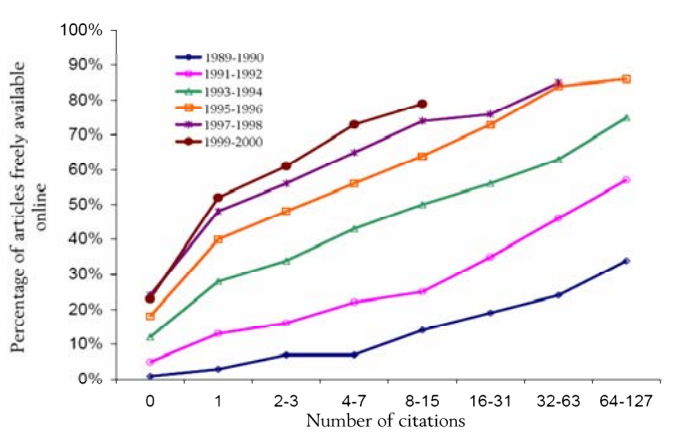


Abbildung **1** – **Abhängigkeit #Zitierungen/Online**

„[…] there's a sense in which the journal articles prior to the inception of that electronic abstracting and indexing database may as well not exist, because they are so difficult to find. Now that we are starting to see, in libraries, full-text showing up online, I think we are very shortly going to cross a sort of critical mass boundary where those publications that are not instantly available in full-text will become kind of second-rate in a sense, not because their quality is low, but just because people will prefer the accessibility of things they can get right away.”

Clifford Lynch, 1997
zitiert in “If it's not on the Web, it doesn't exist at all”
(S. Stevens-Rayburn & E. N. Bouton)

Abbildung 1 zeigt die Analyse von 119 924 Konferenzartikeln aus dem Bereich „Computer Science“ und den damit verbundenen Disziplinen. Entsprechend sind häufiger zitierte Artikel, sowie neuere Artikel, mit entsprechend höherer Wahrscheinlichkeit im Netz verfügbar. Während in [Law01] nur die These aufgestellt wird, dass unter der Voraussetzung gleicher Qualität Onlineartikel einfacher zugänglich sind und somit eine kleinere Beschaffungshürde zugrunde liegt, wodurch sich die häufigeren Zitierungen ableiten lassen, geht Odlyzko soweit, die These aufzustellen „*[…] there will be no doubt that print versions will be eclipsed*“ [Odl02].

In Ergänzung dazu, dass veröffentlichte Artikel über das Internet verfügbar gemacht werden müssen, um die Wahrscheinlichkeit einer Zitatation zu erhöhen, zeigt Odlyzko, dass es einen Anstieg der elektronischen Kommunikation gibt, die sich vom reinen „zugänglich machen“ innerhalb eines neuen Mediums hin zu einem Ausnutzen der neuen und einzigartigen Möglichkeiten des Mediums Internet bewegt, wie beispielsweise Interaktivitäten, und dadurch nicht mehr in die vorgegebene Bahn des traditionellen „journal publishing“ passen [Odl02].

Auch Kuhlen sieht in Ausleihstatistiken (→ [Kuh02d]) den Hinweis, dass es einerseits die Bibliotheken der Buchwissenschaften geben wird, die jedoch eher die Archive des Wissens sein werden und andererseits die modernen Referenzeinrichtungen, die zwar kaum eigene Bestände haben, aber umfassende Nachweis- und Auslieferungsleistungen überwiegend digitaler Wissensobjekte erbringen [Kuh02d]. Entsprechend werden die Bibliotheken wissenschaftliche Zeitschriften nicht mehr in ausgedruckter Form zur Verfügung stellen können, sondern allenfalls noch als „Verwaltungszentrale“ für digitale Wissensobjekte – nicht zuletzt entsprechend online verfügbarer Veröffentlichungen – dienen.

Auch wenn die Wahrscheinlichkeit nicht sehr hoch ist, dass der beschriebene Wechsel rapide von statten gehen wird, so sind erste Trends nicht zu übersehen. Nicht zuletzt kündigen Bibliotheken bereits heute Journal Abonnements auf und nutzen die Möglichkeiten des document delivery's – beispielsweise Subito –, nachdem diese Kosten/Nutzen Rechnungen bezüglich der Abwägung beider Beschaffungsmaßnahmen durchführten [Odl02].

	Anzahl Zugriffe	Jahr
e-math	1 200 000/Monat	Anfang 1999
arXive	2 000 000/Monat	1999
netlib	2 500 000/Monat	1999
SciELO	63 695/Januar	1999

Tabelle **1** – **Anzahl Zugriffe Onlinedienste** [Odl02]

Basierend auf Serverlog Auswertungen von FirstMonday [Odl02] (in Tabelle 1 nicht aufgeführt) ließ sich eine Steigerungsrate der Zugriffe von 50%-100% (bezogen auf FirstMonday) innerhalb eines Jahres feststellen. Im Rahmen bereits vorhandener Erfahrungen mit Internetangeboten kann dieser Wert auf ähnliche Angebote, wie sie in Tabelle 1 aufgeführt sind, verallgemeinert werden. Im Hinblick auf arXive stehen für 1999 weitere informative Werte zu Verfügung: So wurden insgesamt ca. 7 Millionen Volltexte herunter geladen. Bei ca. 100 000 verfügbaren Volltexten, entspricht dies 70 Downloads eines Artikels pro Jahr für den Hauptserver. Entsprechend [Odl02] kommt bei der Betrachtung aller Server und unter der Annahme, dass alle Artikel über die gesamte Zeit gleich oft heruntergeladen werden, ein kumulierter Wert von 140 Downloads/Artikel heraus. Korrigiert man diese Zahl für jeden Artikel zeitabhängig, so kommt man auf eine Lesequote der Artikel, wie sie für etablierte Journals typisch ist [Odl02].

Entsprechend der bereits hohen Verfügbarkeit von Onlineveröffentlichungen steigt mit jeder Weiteren der Druck auf die Autoren, die Zugänglichkeit (Accessibility) ihrer eigenen Werke zu vereinfachen. Wenn, wie in [Kuh02d] beschrieben, die Bibliotheken die Möglichkeiten bieten Wissensobjekte recherchierbar zu machen, so ergibt sich genau jener neue Markt einer Dienstleistung, Onlineveröffentlichungen unter einem Portal mit mächtigen Recherchemöglichkeiten anzubieten. Zeitgleich ergibt sich für Verlage die Möglichkeit dafür zu sorgen, dass zum einen Veröffentlichungen entsprechend dem neuen Medium aufbereitet werden und zum anderen, dass der Verlag mit seinen Lektoren als Verwalter tätigt wird und somit die in 2.3.3 beschriebenen Maßnahmen durchführen kann.

Genau an diesem Punkt versucht das Projekt EVOBO Prototyp für kommende Wissens- & Veröffentlichungsportal zu sein. Wobei EVOBO kein Portal an sich, sondern vielmehr eine Technologiesammlung darstellt, die von entsprechenden Rechteinhabern genutzt werden kann, um Onlinepublishing und die damit verbundenen neuen Möglichkeiten voranzutreiben.

2.3 Erweiterte Möglichkeiten des Onlinepublishings

Wie bereits während der Begriffsdefinition unter 2.1 herausgearbeitet, bieten sich beim Onlinepublishing, über papierbasierte Veröffentlichungen hinaus, weitergehende Möglichkeiten. Diese können wie folgt kategorisiert werden:

- mediale Anpassung der Inhalte
- mediale Anpassung der Diskussion
- mediale Anpassung der Erzeugung

2.3.1 Mediale Anpassung der Inhalte

Die mediale Anpassung der Inhalte stellt keine großen Herausforderungen an den Autor dar, da sie ihm letztlich größeren Freiraum bietet. Während bei papierbasierten Veröffentlichungen beispielsweise Quellcodefragmente einer Visualisierungstechnik abgedruckt werden müssen, können bei Internet – respektive WANs auf Internettechnologie basierend – Veröffentlichungen Referenzimplementationen mitgeführt werden, die es dem Leser erlauben, die Mächtigkeit des Algorithmus im Rahmen des Möglichen bereits innerhalb kürzester Zeit auszuloten und erlangtes Wissen entsprechend schneller zu nutzen.

Insbesondere innerhalb der wissenschaftlichen Weiterbildung an Universitäten dürfte diese Art der Anpassung willkommen geheißen werden, fällt es doch Studenten, zumindest zu Beginn ihres Studiums, meist leichter anhand von Beispielen zu lernen. Selbstverständlich ist die mediale Anpassung nicht nur innerhalb der Lehre einsetzbar, sondern könnte insbesondere auch bei der Forschung – im folgenden Beispiel bezogen auf die Physik – interessant sein. So ist z.B. vorstellbar, dass parametrisierbare Gleichungen interaktiv genutzt werden können. Einfachstes Beispiel, ohne jeglichen Anspruch auf Komplexität, wäre die Berechnung einer Umlaufbahn eines Planeten unter parametrisierten Fremdeinwirkungen, wobei interaktiv die Auswirkungen demonstriert werden könnten.

2.3.2 Mediale Anpassung der Diskussion

Ebenfalls wenig problematisch, aber gleichwohl effektiver, stellt sich die mediale Anpassung der Diskussion über ein Wissensobjekt dar. Neben den bisherigen Arten von Diskussion, beispielsweise über das Bezugnehmen einer fremden Veröffentlichung innerhalb der eigenen oder auch bei Konferenzen, ergeben sich bei Onlineveröffentlichungen wesentlich davon differierende Diskussions- & Kommunikationsarten. Während Diskussionen bislang unter wenigen Partnern stattfanden und die Inhalte einer solchen grundlegend erst einmal nur den Teilnehmern zur Verfügung standen, stehen Ergebnisse einer Diskussion innerhalb des Internets z.B. durch spezielle Foren jederzeit allen zur Verfügung. Neben dem zeitversetzten Diskutieren erlaubt das Medium auch Echtzeitdiskussionen, deren Inhalte durch z.B. Logdateien ebenfalls mit minimalem Aufwand recherchierbar gemacht werden können.

Letztlich nutzt bereits die heutige Kommunikation bzw. Diskussion über gedruckte Werke in vielen Fällen andere Medien. Kaum ein Autor wird nicht schon einmal via E-Mail Informationen eingeholt, oder mit einem Co-Autor diskutiert haben. Entsprechend werden hier Informationen aus einem Medium bereits in ein anderes überführt. Wird nun letztlich das Wissensobjekt selbst in das neue Medium transferiert, so geschieht sowohl Veröffentlichung als auch Diskussion innerhalb eines Mediums und ist entsprechend von der Art als gleichwertig zu erachten, da gleiche Retrieval- & Referenzierverfahren angewandt werden können. Somit ist der Transfer des vollständigen Wissensobjekts, bestehend aus der Einheit der Ausgangspublikation sowie der darauf aufbauenden Diskussion, einem teilweisen Transfer von Komponenten vorzuziehen.

Entsprechende Aufteilungen von Diskussionsdaten und –verfahren sind bereits in der Literatur verfügbar. So unterscheidet beispielsweise [KSF02] zwischen „Primary data“, „Referential data“ & „Metrics data“ und den zugehörigen Erfassungsverfahren. Während „Primary data“ den Ausgangspunkt der Diskussion darstellt, bezieht sich „Referential data“ auf „Primary data“ und/oder „Referential data“. Beide werden explizit erfasst.

Wohingegen „Metrics data“ implizit über das System erfasst werden können. Ergänzend beschreibt [KSF02], wie die erwähnten Daten dazu genutzt werden können, die Gewissenhaftigkeit der Teilnehmer und die Frequenz der Teilnahme zu verbessern. Um die Gewissenhaftigkeit und daraus resultierend die Qualität der Beiträge zu erhöhen, werden folgende Funktionalitäten eingeführt:

- Member Identity
- Identity in Context
- User Control of Resource
- Status Metrics
- Self-Policing and Enforcement
- Profanity Metrics
- Banning

Obwohl sich [KSF02] auf spezialisierte Gruppe der Musikbranche bezieht – Gitarristen bzw. Bassisten – und gleichzeitig anführt, dass die oben genannten Funktionalitäten nur innerhalb solch spezialisierter Diskussionsgruppen zum gewünschten Ergebnis führen, so sind diese dennoch für die Diskussionsmöglichkeiten innerhalb eines Onlinepublikationssystems anwendbar, sind doch die Diskussionsteilnehmer insbesondere bei wissenschaftlichen Veröffentlichungen meist ebenso in ihren Bereichen spezialisiert, wie die in [KSF02] untersuchten.

2.3.3 Mediale Anpassung der Erzeugung

Obwohl prinzipiell möglich, ist es fraglich, ob an der grundlegenden Struktur der Erzeugung eines Wissensobjekts – im Fokus dieser Arbeit einer Publikation – Anpassungen vorgenommen werden sollen. Grundlage eines geänderten Erarbeitungsablaufes kann das Ergebnis der in 2.3.2 beschriebenen, an das Medium angepassten, Diskussion darstellen.

Vorstellbar wäre es, dass in eine neue Veröffentlichung Erkenntnisse aus den beschriebenen Diskussionsmöglichkeiten eingearbeitet werden. Wesentlich komplexer wäre die Idee des „living book“, welches letztlich eine Ausgangsveröffentlichung darstellt, die durch Befugte[4] mit den Ergebnissen der in 2.3.2 beschriebenen Diskussionen in einem fortlaufenden Prozess ergänzt wird.

Wobei sich bei dieser Art der fortlaufenden Publikation mehrere Fragen stellen, unter anderem auch folgende drei:

- Wie kann auf eine solche Publikation referenziert werden?
- Auf welcher Basis werden neue Publikationen gerechtfertigt?
- Wie regelt sich die Reputation für eine solche Publikation?

Referenzen auf ein sich änderndes Objekt zu setzen, scheint zwar prinzipiell nicht erwünscht zu sein, stellt jedoch technisch kein Problem dar. So kann der in 2.4.1 beschriebene DOI[5] soweit ergänzt werden bzw. entsprechend in der jetzigen Fassung genutzt werden, dass Referenzen nicht nur auf ein Wissensobjekt gesetzt werden können, sondern zusätzlich auch auf unterschiedliche Versionen des gleichen Objekts. Problematisch wird ebenfalls der Abdruck einer solchen Veröffentlichung in z.B. Journals. Denkenswert ist nur, dass entweder keine Zurückführung in „ältere“ Verbreitungsmöglichkeiten erfolgt, oder aber, dass mit jedem Abdruck sowohl Version, als auch entsprechender DOI mit abgedruckt werden. Ein solcher DOI würde auf die eventuell veraltete, aber für die Referenz korrekte Onlineversion des Abdrucks führen, gleichzeitig jedoch auch die Möglichkeit bieten, online auf die aktuelle Version zurückzugreifen.

---

[4] Befugte müssen nicht nur Autoren, sondern können auch vom Herausgeber bestimmte Personen sein, denen das Recht eingestanden wird „offizielle“ Ergänzungen zu einer Ausgangsveröffentlichung hinzuzufügen. Ideen hierzu können beispielsweise aus dem Prinzip der Versionsverwaltung bei Softwareprojekten kommen.

[5] Digital Object Identifier

Letztlich scheint die Problematik der Referenz sowohl online, als auch offline mit akzeptablem Aufwand lösbar zu sein.

Größeres Gewicht dürfte die Frage nach der Rechtfertigung einer neuen Publikation eines bereits behandelten Gebiets mit sich bringen. Diese Frage lässt sich darauf zurückführen, ab wann eine Ausgangspublikation nicht mehr länger „erweitert" werden soll und stattdessen entweder die Erweiterungen zusammen mit der Ausgangspublikation eine neue Publikation bilden, oder die bisherigen Erkenntnisse in eine vollständig neue Publikation überführt werden. Im Prinzip der Versionskontrolle könnten Subversionen (beispielsweise 1.1) dazu genutzt werden inkrementelle Erweiterungen durch Befugte zu referenzieren und neue Hauptversionen (beispielsweise 2.0) für eine neue Publikation stehen. Prinzipbedingt muss festgelegt sein, dass in Version A enthaltene Informationen ebenfalls und in gleicher Form in Version B enthalten sein müssen, wenn gilt, dass

$$\text{Version}(A) < \text{Version}(B) \wedge \text{Version}(B) - \text{Version}(A) < 1$$

Durch diese Festlegung ist die Gültigkeit einer Referenz auch in der komplexen Struktur eines „living books" sichergestellt, da innerhalb von fortlaufenden Erweiterungen jegliche Informationen der Ausgangspublikation enthalten sind, in neuen Publikationen, d.h. mit neuer Hauptversion, Informationen aus der „alten" Publikation weggelassen werden dürfen.

Wichtige Frage bleibt, insbesondere auch im Sinne des Geistigen Eigentums, dessen Verwertung und dessen möglichen Schutz, welchem Autor die Ausgangspublikation zugeordnet wird, welchem die Ergänzungen und welchem die jeweiligen, möglichen Folgepublikationen mit neuer Hauptversionsnummer.

Ohne jedoch Versuchsdaten über das Konzept „living book“ selbst zu besitzen, kann diese Frage im jetzigen Stadium des Systems nicht eindeutig beantwortet werden, zumal auch die Literatur bislang keine eindeutige Antwort liefert. Gleichwohl weist Kuhlen bereits in seinem Vortrag „Wenn Autoren Kollaborateure werden – was ändert sich dann?“ darauf hin, dass der (individuelle) Urheberbegriff für Wissen und Information sich in kollaborativen Umgebungen verflüchtigt und damit auch der Eigentumsbegriff und damit schließlich das klassische proprietäre Verwertungskonzept zumindest problematisch wird.

Entsprechend ist die Frage nach der Zurechnung des Wissensobjekts bzw. Teilobjekten davon zu einer oder mehreren Person die einzige, die in diesem Kontext ungeklärt bleibt, insbesondere dann, wenn beliebig viele Autoren theoretisch auf Wortebene Veränderungen durchführen können.

Wobei, auch wenn die gestellten Fragen zumindest teilweise befriedigend beantwortet werden konnten, festgehalten werden muss, dass sich der Nutzen eines solchen „living book“ Systems, in diesem Umfang, im Versuch zuerst zeigen muss. Denkbar ist sehr wohl, dass die innerhalb der Diskussion erreichten Ergebnisse in eine neue Veröffentlichung einfließen können, aber an bereits veröffentlichte Werke neue Informationen anzuhängen scheint, zumindest im wissenschaftlichen Umfeld, nach diesen Fragestellungen weiterhin fragwürdig. Um jedoch den beschriebenen Versuch durchführen zu können und anhand der dadurch erlangten Daten Schlüsse ziehen zu können, ist in der Implementationsspezifikation für den EVOBO Prototyp dennoch ein solches „living book“ System vorgesehen. Die Realisation einer solchen Komponente soll sich dabei jedoch auf die „Vorablizenz[6], und somit auf unveröffentlichte Werke, begrenzen.

[6] Die Bedeutung einer Vorablizenz und die damit verbunden Rechte werden in Kapitel 5 näher beschrieben.

Interessant in diesem Versuch wird es sein, zu sehen, ob der Autor einer Ausgangspublikation die Erweiterungen/Änderungen die im Rahmen des living books entstanden für seine Publikation in die Endrevision übernehmen wird.

Sollte es zu dem in [Kuh03b] beschriebenen Paradigmen Wechsel kommen, wonach die Gemeinschaft weg von der statischen Sicht auf Informationen, bezeichnet als „knowledge warehouse approach", und hin zu dem kommen, was Kuhlen als „dynamic or communicative view on knowledge management" [Kuh03b] bezeichnet, so entspricht das Ergebnis des Wechsels genau der Idee des „living books". Wobei anzumerken ist, dass auch Kuhlen in [Kuh03b] zu dem Schluss kommt, dass „Communication in electronic environment needs coordination and management".

Dieses Management und die Koordination zu definieren, so dass es der Komplexität der Idee in allen Bestandteilen gerecht wird, stellt die Hauptaufgabe bei der Lösung des genannten Problems dar, könnte aber gleichzeitig ein neues Betätigungsfeld für Verlage werden.

Dass die genannte Idee der Erweiterung einer Ausgangspublikation durch unbegrenzt viele, auch anonyme, Autoren technisch problemlos möglich ist, zeigen so genannte WIKIS. Zusammengefasst handelt es sich bei einem Wiki – hawaiianisch für „schnell'" – um eine Webseite, deren Autoren die Besucher selbst sind. Jeder Besucher kann jede Seite editieren und Links auf neue Seiten innerhalb des Wikis setzen. Durch das Setzen eines neuen Links, wird innerhalb des Wikis eine neue Seite angelegt. Diese neue Seite kann wiederum von allen Besuchern bearbeitet werden.

Neben dem unter `http://www.c2.com/cgi/wiki` erreichbaren, ursprünglichen WikiWiki, entstehen z.Zt. unzählige weiterführende Implementation mit jeweils erweiterten Funktionalitäten bis hin zu einer Versionsverwaltung & IP Tracking in FlexWiki[7]. Letztlich stellt ein Wiki, nicht nur im rechtlichen Sinne, den unkonventionellsten Umgang mit Wissen dar und entspricht der Idee des „living book" in einer unkontrollierten[8] Umgebung. Als erfolgreiches Beispiel eines solchen Systems sei WikiPedia[9] genannt. Dass das kollaborative Arbeiten in seinem Ergebnis effektiver sein kann, als die vielleicht entstandenen Einzelarbeiten beschreibt [KBGSS02].

2.4 Merkmale eines Onlinepublikationssystems

Ergänzend zur Bereitstellung/Veröffentlichung kommen bei einem System, das innerhalb des Internets operiert noch weitere Funktionalitäten hinzu, die zur Verfügung gestellt werden sollten.

① Eindeutige Kennung eines digitalen Wissensobjekts
② Klassifizierung & Beschreibung von Wissensobjekten
③ Ausnutzung medienabhängiger Fähigkeiten (z.B. Hyperlinks)
④ Bereithalten der Referenzierungen

[7] http://www.flexwiki.com

[8] unkontrolliert im Sinne von „nicht moderiert": Jeder Nutzer kann frei in ein Wiki schreiben. Gleichzeitig ist es jedem Nutzer jedoch auch möglich bisherige Inhalte zu löschen. Wodurch die Mitglieder in die Lage versetzt werden, sich selbst zu regulieren.

[9] http://de.wikipedia.org/

2.4.1 Eindeutige Kennung eines digitalen Wissensobjekts

Insbesondere in einem schnellen Medium wie dem Internet, in dem Verweise schnell altern [Cap01] bzw. durch neue ersetzt werden, bedarf es der Vergabe einer eindeutigen Kennung für jedes Wissensobjekt, unter der genau dieses Wissensobjekt und die entsprechende Revision/Version referenziert wird. Die in, auf Internettechnologie[10] basierenden, WANs genutzten URLs reichen für die permanente Erschließungen innerhalb von Publikation trotz ihrer prinzipiellen technischen Eignung aufgrund der fehlenden sozialen Infrastruktur [Pas03] nicht aus. Zu schnell können einmal gesetzte Hyperlinks ins Leere zeigen und somit eine benötigte Referenz oder einen Verweis für den Nutzer, zumindest solange keine weiteren Hilfsmittel wie z.B. Suchmaschinen genutzt werden, unauffindbar machen. Entsprechendes zeigt sich auch in einer kürzlich durchgeführten Studie[11], wonach vier Prozent der Links der Internetquellen-Angaben der Zeitschriften NEJM, JAMA und Science bereits nach drei Monaten, zehn Prozent nach 15 und 13% nach 27 Monaten nicht mehr erreichbar waren.

Eine mögliche Lösung der Problematik nennt sich DOI – Digital Object Identifier – und entstammt der International DOI Foundation, Inc.

„*The DOI is a system which provides a mechanism to interoperably identify and exchange intellectual property in the digital environment.*"

(Auszug aus [PHH03], Seite 15).

Mit dieser Definition geht das DOI Framework direkt auf die in [Cap01] herausgearbeiteten Eigenschaften eines effektiven Linksystems ein: offen, generalisierbar & robust.

---

[10] Als Internettechnologie sind in diesem Kontext neben den zugrunde liegenden Protokollen wie IP oder den darauf aufbauenden Protokollen TCP bzw. UDP auch mit dem Internet aufgekommenen bzw. verbreiteten Technologien wie z.B. Hyperlinks.

[11] Science 302 (2003) 787

Das DOI Framework besteht aus den Komponenten [PHH03]:

- Numbering: identifier
- Description: metadata
- Resolution: handling
- Policies: buiness model

Ablauf: Ein über einen Identifier versehenes und mit Metadaten [BDM03] beschriebenes Wissensobjekt kann unter einem bestimmten Geschäftsmodell über ein Auflösungssystem (handling) referenziert werden. Entsprechend ist das DOI Modell beliebig zur Referenzierung einsetzbar, da weder das grundlegende Auflösungssystem, noch das Geschäftsmodell irgendwie beeinträchtigt werden. Abbildung 2 verdeutlicht den Ablauf der Auflösung eines DO Identifiers.

Präfix | Suffix

Beispiel Identifier: 10.1000 / 123456.

Das Präfix unterteilt sich in zwei Komponenten, wovon die „10." den Identifier als DOI „identifiziert" und „1000" das unter diesem Präfix registrierte Institut bezeichnet, welches für diesen Präfix Suffixe registrieren darf. Die zweite Komponente des Präfixes kann entsprechend Spezifikation in unbegrenzt vielen Schritten unterteilt werden, so dass beliebig viele Organisationen mit eigenem Präfix versorgt werden können.

Das Suffix ist bezogen auf dem Präfix eindeutig und definiert so wiederum eindeutig das über den DOI referenzierte Wissensobjekt innerhalb des über das Präfix definierten Instituts. Aufbau, Länge und Eigenschaften des Suffixes können von dem durch das Präfix definierte Institut frei gewählt werden, wodurch sichergestellt wird, dass bereits durchgeführte Indexierarbeiten der einzelnen Institute erhalten bleiben.

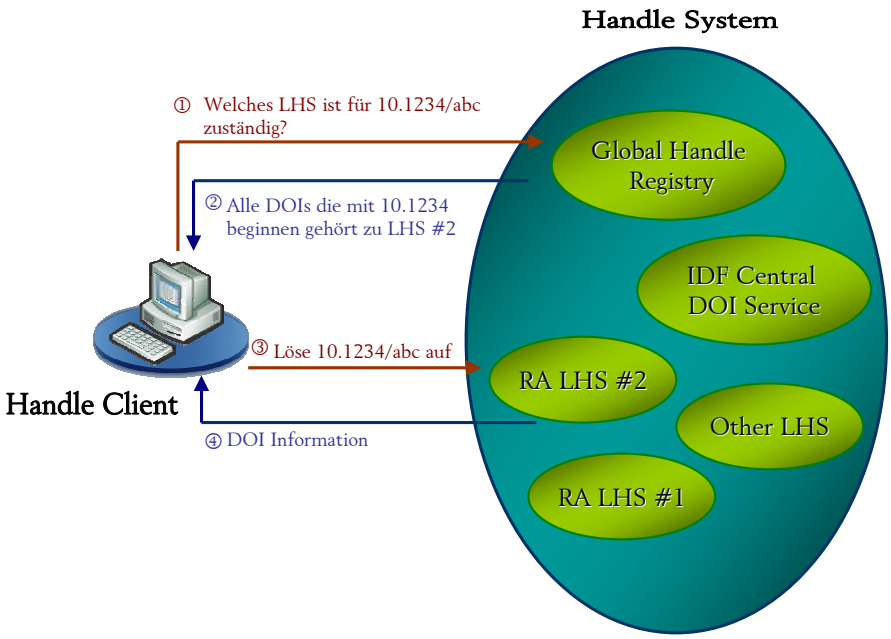


Abbildung 2 – DOI Handling

Basierend auf [Cap01] sollte ein Linksystem die Möglichkeit bieten unter mehreren verschiedenen Quellen für die gleiche Veröffentlichung die Beste auszusuchen – entspricht dem „appropriate copy" Problem. Diese Funktionalität ist insbesondere dann wichtig, wenn z.B. ein Institut einen lokalen Server mit den Kopien der bezogen Artikeln besitzt, da der online Zugriff über andere Systeme zu kostenintensiv wäre. Entsprechend kann der Auflösung (Resolution) eines DOIs ein lokaler Server vorgeschalten werden, der im Falle eines Treffers auf dem lokalen Server das dort gespeicherte Wissensobjekt zurückliefert und andernfalls die Auflösung an den übergeordneten Server weiterleitet, wonach dieser die Dereferenzierung vornimmt.

Abbildung 3 zeigt eine beispielhafte Implementation des DOI Frameworks für textbasierte Arbeiten mit mehreren Bezugsquellen und weiterführenden Metainformationen. Wobei diese Metainformationen innerhalb des DOI nicht fest definiert sein müssen, sondern durch Anwendungsserver innerhalb der Institute bereitgestellt werden können. DOI selbst nutz die ISTC[12] Metadatenstruktur.

[12] International Standard Textual Abstraction Code number

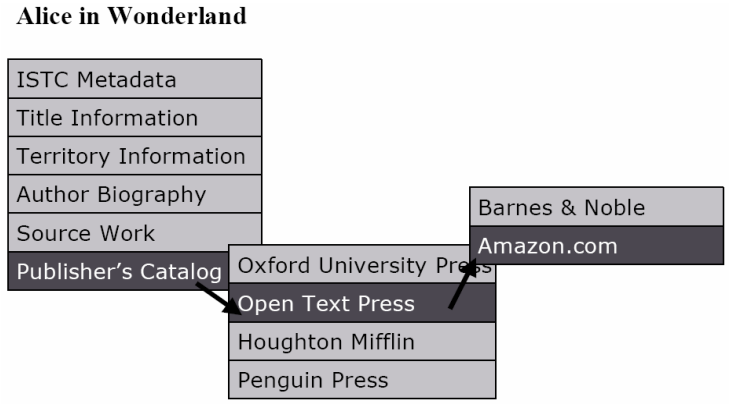


Abbildung **3** – **Beispielhafte DOI Implementation**

Während Zeitschriften wie Nature bereits jeden Aufsatz über einen DOI registrieren, bemängeln Kleinverlage, dass das System für sie zu teuer ist. Diese können das jetzt von der Deutschen Bibliothek (DDB) auf URN Basis angebotene Adressierungsschema nutzen [Kur04].

Auf einer anderen Ebene setzt das OpenURL Framework an (→ Van de Sompel & Beit-Arie, 2001). Hier dient bereits die Namensauflösung der ISPs[13] zur Dereferenzierung, wobei das prinzipielle System ähnlich dem des DOI aufgebaut ist: An einen Linkserver werden innerhalb der URL Metadaten übergeben. Entsprechend wird die URL durch die beiden Bestandteile Linkserver|Metadaten kodiert. Das OpenURL Framework ist bei der NISO[14] zur Standardisierung beantragt [*NISOa*].

[13] Internet Service Provider

[14] National Information Standards Organization

2.4.2 Klassifizierung & Beschreibung von Wissensobjekten

Um Wissensobjekte recherchierbar zu gestalten, müssen diese mit entsprechenden Metadaten versehen werden. Entsprechend der Problematik der Vergabe der Metadaten bietet es sich an, wie in [Kuh02d] beschrieben die Bibliotheken diese Aufgaben übernehmen zu lassen. Durch die Übernahme der schematischen Aufbereitung von Wissensobjekten und ggf. deren Bereitstellung innerhalb eines DOI Frameworks, gelänge es Bibliotheken in die ebenfalls in [Kuh02d] beschriebene neue Domäne der Bibliotheken einzutreten.

2.4.3 Ausnutzung medienabhängiger Möglichkeiten (z.B. Hyperlinks)

Die Ausnutzung der unter 2.3 erarbeiteten, medienabhängigen Fähigkeiten – dort für auf Internettechnologie basierende Systeme – muss in einem Onlinepublikationssystem für dessen Akzeptanz und Einmaligkeit umgesetzt sein.

Die obligatorische Ausnutzung von z.B. Hyperlinks muss nicht extra erwähnt werden, sondern fällt ebenfalls in medienabhängige Fähigkeiten, die jedoch mittlerweile so zum Standard wurden, dass es üblich ist, selbst in gedruckten Werken auf beispielsweise URLs zu referenzieren. Weitere Beschreibungen der Besonderheiten des Cyberspaces, wie ihn das Internet und mit ihm jedes Netzwerk, das auf dessen Technologie basiert, aufspannt, finden sich in [Col02].

2.4.4 Bereithalten der Referenzierungen

Unter dem Bereithalten der Referenzierung ist im Rahmen dieser Arbeit zu verstehen, dass die innerhalb einer Veröffentlichung referenzierten Arbeiten ebenfalls innerhalb des gleichen Mediums, d.h. ohne Medienbruch, mit den gleichen Eigenschaften und dem gleichen „Look & Feel“[15] wie die Ausgangspublikation zur Verfügung stehen.

---

[15] Look & Feel bezeichnet hier nicht die Art, wie eine Arbeit geschrieben sein sollte, soll entsprechend keine „Gleichschaltung“ der Autoren bedeuten, sondern vielmehr die Art und Weise der Recherchierbarkeit, der Erreichbarkeit der Nutzbarkeit des Angebots über z.B. Verlage hinweg beschreiben.

Mit den gleichen Eigenschaften bedeutet insbesondere, dass die referenzierten Wissensobjekte eindeutig über z.B. einen DOI gekennzeichnet sein müssen und mindestens über die standardisierten Metadaten verfügen müssen, so dass z.B. Referenzen nicht unbedingt tatsächlich verfolgt werden müssen, sondern lediglich ein Abstrakt einer Referenz digital angefordert werden kann.

2.5 Systemvorstellung

Diese Systemvorstellung soll drei bereits existente Dienste nennen und kurz vorstellen. Alle drei Dienste unterscheiden sich in ihrem Umfang und in ihrer Leistung.

2.5.1 CrossRef

Bei CROSSREF[16] [Wal02] handelt es sich um eine Initiative der Publishers International Linking Association (PILA) [Cap01], die Mitglied des IDFs ist und für ihre Mitglieder als Registrar DOIs vergeben kann. CROSSREF selbst hält entsprechend der DOI Definition keine Daten bereit, sondern liefert den „Linking Backbone“ und somit die Struktur zum DOI Framework. CROSSREF wurde 2000 gegründet und begann mit 12 der größten Verlage (major publisher), wobei diese Zahl bis 2002 bereits auf 92 anstieg [Wal02].

Da CROSSREF den konsequenten Einsatz des DOI Frameworks darstellt, erbt es auch dessen Probleme, wobei eines bereits in [Cap01] benannt wurde: Das zur Verfügung stellen von Wissensobjekten mehrerer Institute führt unweigerlich zu Problemen, wenn diese vergütet werden müssen. Entsprechend stößt der Rechercheur bei seiner Verfolgung der Verknüpfung unweigerlich auf unterschiedliche Abrechnungssysteme der einzelnen Institute, für die er sich jeweils einzeln anmelden muss. Daraus resultierend ist eine schnelle Überprüfung der referenzierten Artikel, abhängig von der Verwertungspolitik des Rechteinhabers, nur noch eingeschränkt möglich und zerstört dadurch eine der wichtigen Eigenschaften der Onlinepublikation: Erreichbarkeit (Accessibility).

[16] http://www.crossref.org

2.5.2 Vascoda

Während CROSSREF generell beliebige Referenzierungen auflösen kann, handelt es sich bei VASCODA[17] um ein interdisziplinäres Zugangsportal zu 23 virtuellen Fachbibliotheken, den vier großen Informationsverbünden: EconDoc[18], GetInfo[19], InfoConnex[20] & Medizin, sowie zur Elektronischen Zeitschriftenbibliothek[21] [*VASC*].

Vascoda wird unterstützt durch das Bundesministerium für Bildung und Forschung (BMBF) sowie der Deutschen Forschungsgemeinschaft (DFG) und bietet über die beschriebene Anbindung den Zugriff auf alle deutschen Informationsdienste, die durch die Bereitstellung öffentlicher Gelder gegründet wurden. Sowohl das BMBF als auch die DFG betrachten VASCODA als den Grundstein zur „German digital library" [*VASC*]. Während die meisten, der über VASCODA bereitgestellten, Informationen frei[22] verfügbar sind, bietet VASCODA ebenfalls einen Zugang zu kommerziellen Datenbank an, deren Abrechnung auf pay-per-view Basis durchgeführt wird.

2.5.3 SpringerLink

SpringerLink[23] bezeichnet sich selbst als „einer der führenden Online-Informationsdienste für naturwissenschaftliche, technische und medizinsche Bücher und Zeitschriften. Für Wissenschaftler an Hochschulen, in Firmen sowie in den bedeutendsten Wissenszentren stellt Springerlink die bevorzugte Informationsquelle dar." (→ „About" auf SpringerLink).

---

[17] http://www.vascoda.de

[18] Bereich: Wirtschaft

[19] Bereich: Naturwissenschaften & Technik

[20] Bereich: Pädagogik, Sozialwissenschaften & Psychologie

[21] Die EZB ist ein kooperativer Service von 209 Bibliotheken zur Bereitstellung von elektronischen Zeitschriften, die im Internet publiziert werden [*VASC*].

[22] Hier im Sinne von kostenlos

[23] http://www.springerlink.de

SpringerLink umfasst 11 Fachgebietsbibliotheken, mit ebenfalls interdisziplinärem Angebot, das sowohl über 200 Bücher als auch über 500 Volltext-Zeitschriften verfügt. Neben dem Springer Verlag selbst nutzen auch Urban & Vogel, sowie Steinkopff und Birkhäuser die SpringerLink Plattform.

Neben dem Zugang, zu ebenfalls in gedruckter Form erhältlichen Informationen, bietet SpringerLink unter der Bezeichnung Online First™ Publikationen vor ihrer gedruckten Veröffentlichung an. Die so verfügbaren Publikationen sind über eine zugewiesene DOI bereits zitier- & recherchierbar. Ergänzend beinhaltet SpringLink die Möglichkeit Veröffentlichungen dem Medium gerecht über Simulationen, Video, Ton & 3-D-Darstellungen zu ergänzen.

2.6 Resümee

Dieser Überblick über Onlinepublikationssysteme mit der abschließenden Nennung dreier Beispielsysteme sollte der Orientierung und Abgrenzung dienen. Herausgearbeitet sollten insbesondere die Unterschiede zwischen den Möglichkeiten papierbasierter Veröffentlichungen und z.B. im Internet bereitgestellten Arbeiten. Durch neue Möglichkeiten entstanden neue Probleme, die für die Schaffung eines Gesamtbildes wichtig erscheinen. Insbesondere bei Ideen, die nicht zuletzt einen anderen Denkansatz und entsprechend auch einen anderen Verwertungsgrundsatz verfolgen (müssen), wie z.B. das living book, bedarf es noch notwendige Regelungen des Managements dieser Informationen zu finden. Erste erfolgreiche Wiki Anwendungen zeigen zwar, dass die prinzipielle Idee durchführbar zu sein scheint, dennoch muss die bereitgestellte Funktionalität zum einen erst von der wissenschaftlichen Gemeinschaft akzeptiert werden und zum anderen muss der erwähnte Paradigmen Wechsel bezüglich der Betrachtung von Information und Wissen erfolgen.

Klar abgegrenzt sollten nach diesen Ausführungen die notwendigen Funktionalitäten für ein System mit dem Anspruch von EVOBO sein. Die ergänzende Auflistung dreier bereits existenter Systeme dient als Ideenlieferant für ergänzende Funktionalitäten innerhalb des Designs des Prototyps und zeigt weiterhin das Gespür einiger Verlage, die bereits die Prioriät der Erreichbarkeit ihrer Veröffentlichungen bzw. der durch sie vertretenen Autoren korrekt erkannt haben und entsprechend funktionale System (→ SpringerLink) anbieten.

Kapitel 3

Digital Rights Management

3.1 Geschichtliches

Bereits 1986 beschäftigte sich ein Report, herausgegeben vom U.S. Congress, weniger mit dem Begriff des Geistigen Eigentums an sich, sondern vielmehr mit den Problemen der Übernahme der bisherigen Regelungen bezüglich des Urheberrechts (Copyright) in das digitale Zeitalter.

Die bereits in diesem Report herausgearbeiteten Problemfelder sind teilweise heute von noch größerer Bedeutung als zum Herausgabezeitpunkt:

- Eindeutige Identifizierung des Autors

  Mit dem Eintritt in das digitale Zeitalter wurde es möglich, dass mehrere Autoren gleichzeitig an einer Arbeit schreiben konnten. Entsprechend existiert eine so entstehende Arbeit in verschiedenen Formen, mit verschiedenen Inhalten, an verschiedenen Orten, wobei es sich bei jeder um ein Original handelt. Nicht nur das identifizieren der Autoren einer solchen Arbeit entsprechend des Copyrights stellt dann ein Problem dar, sondern auch der zu leistende administrative Aufwand.

- Verstöße identifizieren & Rechte erzwingen

  Mithilfe des Computers, der zur Verfügung stehenden Massenspeicher und Bandbreiten wird in kürzerer Zeit eine größere Anzahl an Verstößen möglich, die gleichzeitig das Durchsetzen von Rechten aufgrund der Vielzahl erschwert. Einzelpersonen können beispielsweise ihr gesamtes Musikarchiv weltweit und zeitgleich Millionen von Nutzern online anbieten.

- **Die private Nutzung**

  Am Ende des zweiten Weltkriegs standen der Bevölkerung drei Informationskanäle zur Verfügung: Gedruckte Publikationen, Radio und Schallplatten. Die Bevölkerung konnte diese Informationskanäle zwar frei nutzen, es war ihnen aufgrund der technischen Gegebenheiten jedoch nicht oder nur schwer möglich Kopien der verfügbaren Informationen zu erzeugen.

  Mittlerweile steht eine Unzahl weiterer Kanäle zur Verfügung und die Konsumenten können aufzeichnen, kopieren und vervielfältigen. Während analoge Kopien über Pauschalvergütungen geregelt wurden, stellt sich die Frage bei digitalen Werken erneut, insbesondere da die Kopie in diesem Fall 1:1 dem Original entsprechen kann. Die Leichtigkeit mit der eine private Kopie erzeugt werden kann wiederum führt zu einem administrativen Problem aufgrund der Menge von möglichen Urheberrechtsverletzungen.

- **Abgeleitete Werke**

  Am Beispiel einer Tageszeitung, die ihr Inhaltsverzeichnis online zur Verfügung stellt, ergibt sich eine neue Problematik. Ein Service der nun die Inhaltesverzeichnisse mehrerer Tageszeitungen online abruft und diese dem Benutzer in aufbereiteter Form präsentiert erzeugt letztlich nur ein abgeleitetes Werk mehrerer geschützter Werke.

  Fraglich ist, wem der schöpferische Akt zugeordnet wird und wem entsprechend das Urheberrecht zuerkannt wird. Im Copyright gilt der Grundsatz, dass abgeleitete Werke ebenfalls dem Autor des Originalwerks vergütet werden müssen. Als Resultat dieser Regelung wiederum kann der Fall entstehen, dass solche Angebote nicht mehr möglich sind und entsprechende nützliche, komprimierte Informationen nicht mehr zur Verfügung gestellt werden.

- **Schützenswerte Güter**

  Eine Frage die ebenfalls in [USC86] gestellt wurde, handelt von der Anpassung des Copyrights auf neue Publikationsformen. Während bislang beispielsweise Reden geschützt waren, war es 1986 ein Computerprogramm nicht. Wobei es reine Definitionssache war, Computerprogramme als eine Art von Rede nur eben in einer anderen Sprache anzusehen.

  Problematisch stellt sich auch folgende Szenerie dar: Ein Künstler erzeugt mittels eines Raytracers ein vollständig künstliches Bild. Wer ist Autor dieses Bildes? Der Künstler, der eine Struktur für den Raytracer vorgegeben hat oder der Programmierer des Raytracers, dessen Algorithmus das Bild erzeugt hat. Das erzeugte Bild ist technisch nur die Lösung einer mathematischen Gleichung bezogen auf die vom Künstler vorgegeben Strukturdefinition, wobei der künstlerische Akt ebenso auf den Künstler selbst, wie den Programmier fällt. Beide arbeiteten letztlich zeitversetzt an einem Werk.

- **Immaterielle Werke**

  Die elektronische Verfügbarkeit von Informationen führt zu dem Problem, dass neue Geschäftsmodelle gefunden werden müssen, die zuvor nicht notwendig waren. Wer bislang ein Journal kaufte, war im Besitz des Journals. Dies ist bei immateriellen Werken nicht mehr zwingen der Fall, was zu Lizenzierung und daraus resultierend zu erhöhten Beschaffungskosten führen kann.

- **Internationaler Zugriff**

  Der weltweite Zugriff ist ebenfalls problematisch. So können Informationen die beispielsweise in Deutschland unter dem Urheberrecht vor ungewollter Vervielfältigung geschützt sind in Ländern ohne Urheberrecht/Copyright unbegrenzt, auch mit kommerziellem Interesse, vervielfältigt werden.

Diese kurze Auflistung zeigt, dass die mit dem digitalen Raum verbundenen Probleme bezüglich des Urheberrechts/Copyrights keinesfalls lapidar ausgedrückt „einfach lösbar“ sind. Die Probleme scheinen uns mittlerweile sehr vertraut. Obwohl bereits 1986 veröffentlicht, wurde dennoch bislang meist keine vollständig zufrieden stellende Lösung gefunden.

3.2 Begriffsdefinition

Eine Herleitung des Begriffs DRM ergibt sich bereits aus den Ausführungen von [USC86], wonach ein entsprechendes System folgende Rechte für ein Werk verwalten können sollte:

① Das Recht es zu besitzen oder digital kontrollieren zu können
② Das Recht die Leistungen zu nutzen
③ Das Recht die Verwendung zu regulieren
④ Das Recht den Ertrag zu erhalten
⑤ Das Recht es aufzubrauchen oder zu zerstören
⑥ Das Recht es zu ändern
⑦ Das Recht es an andere abzutreten
⑧ Das Recht es zu verbreiten
⑨ Das Recht anderen den Zugriff darauf zu verwehren

Entsprechend fasst [Rum03] zwei mögliche Definitionen

„Digital rights management (DRM) is a type of server software developed to enable secure distribution — and perhaps more omportantly, to disable illegal distribution — of paid content over the Web. […]“

——

“DRM covers the description, identification, trading, protecting, monitoring and tracking of all forms of usages over both tangible and intangible assets. […]“

in eine – zugegeben einfache – Definition zusammen.

"*DRM includes everything that someone does with content in order to trade it*"

Die Möglichkeiten mit Geistigem Eigentum zu handeln ergeben sich aus dem Urheberrecht bzw. Copyright. Beide regeln zwar den Schutz und dadurch die Handelbarkeit, setzen jedoch unterschiedliche Akzente.

3.2.1 Urheberrecht und Copyright

Da DRM praktisch synonym für den Schutz der Urheberrechte bzw. des Copyrights geläufig ist, muss geklärt werden, wo die Unterschiede in beiden Systemen liegen.

Das Urheberrecht-System, auch als kontinentales System bezeichnet, gibt vor, dass das Urheberrecht dem Urheber von Natur aus zu steht (Urheberpersönlichkeitsrecht, Schöpferprinzip). Das Urheberrecht schützt die Interessen des Urhebers an seinem Werk.

Er besitzt aufgrund des Persönlichkeitsrechts das Veröffentlichungsrecht, das Recht der ersten Inhaltsmitteilung (§12 UrhG), das Recht auf Anerkennung der Urheberschaft (§13 UrhG) und das Recht, Entstellungen und Beeinträchtigungen des Werkes zu verbieten (§14 UrhG). Der Urheber kann bestimmen, ob und wie sein Werk zu veröffentlichen ist und besitzt das Verwertungsrecht an seinem Werk. Das Verwertungsrecht dient der Durchsetzung der materiellen Interessen.

Das Verwertungsrecht wiederum setzt sich aus den drei Hauptverwertungsrechten (körperlich): Vervielfältigungsrecht (§16 UrhG), Verbreitungsrecht (§17 UrhG) & Ausstellungsrecht (§18 UrhG) und den Nebenverwertungsrechten (nichtkörperlich): Vortrags-, Aufführungs- und Vorführrecht (§19 UrhG), Senderecht (§20 UrhG), Recht der Widergabe durch Bild- und Tonträger (§ 21 UrhG) & Recht der Wiedergabe von Funksendungen (§22 UrhG) zusammen. Diese beispielhafte Aufzählung ist aus §15 UrhG entnommen. Ergänzend besitzt der Urheber das Bearbeitungsrecht (§23 UrhG), mit dem er die Bearbeitung seines Werkes erlauben kann und es gleichzeitig vor unerlaubter Veränderung schützen kann.

Das Urheberrecht ist unverkäuflich. Lediglich die Vervielfältigungs- und Verbreitungsrechte an einem Werk für bestimmte Auflagen und Verbreitungsgebiete können erworben werden. Das Übertragen von Verwertungsrechten entspricht dem Einräumen von Nutzungsrechten an jedem einzelnen Verwertungsrecht, wobei die Nutzungsrechte (Nebenverwertungsrechte) abtretbar sind. Der Urheber kann die Verwertungsrechte innerhalb des Urheberrechts der BRD einem Dritten für einen begrenzten zeitlichen Rahmen einräumen [*JEREa*].

Im Kontrast dazu ist das Copyright in den USA Teil des persönlichen Eigentumsrechts und schützt vor unautorisierter Vervielfältigung, Veränderung oder Verbreitung. Geschützt wird vom Copyright die Ausdrucksform einer Idee, nicht jedoch die Idee als solche [*COPYa*]. Am Beispiel bedeutet das, dass das Computerprogramm „Microsoft Word“ nicht kopiert werden darf, wohl aber der Grundgedanke eines Textverarbeitungsprogramms für den PC – den eventuellen Schutz über ein Patent nicht beachtend. Während der Schutz von Computerprogrammen in der EU diskutiert wird, vertritt bereits [USC86] die Auffassung, dass ein Computerprogramm letztlich nichts anderes als ein, in einer vom Computer „verstandenen“ Sprache, geschriebenes Werk ist und entsprechend innerhalb des Copyright-System zu schützen ist. Einer der grundlegendsten Gegensätze beider Systeme dürfte die Möglichkeit des Verkaufs des Urheberrechts innerhalb des Copyrights sein, eine Möglichkeit die das Urheberrecht nicht vorsieht.

Letztlich ergibt sich der grundlegende Unterschied, dass im angelsächsischen Copyright-System Urheber und Verwerter gleichgesetzt/ausgetauscht werden können, wohingegen das kontinentale Urheberrecht-System zwischen beiden differenziert. Im Fortgang dieser Arbeit wird der Einfachheit halber der Begriff Copyright universell für Copyright und Urheberrecht gleichbedeutend genutzt, auch wenn für die BRD die Regelungen des Urheberrechts tragend sind.

3.2.2 Grenzen der Definition

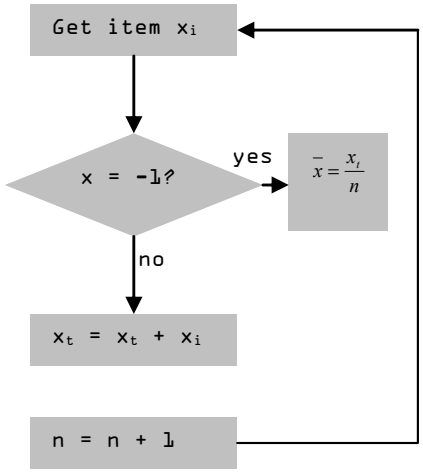
Da es sich bei DRM um ein System zum Schutz vor illegaler Distribution handelt, kann die gelieferte Definition auf den Schutz des Urheberrechts (Copyrights) bzw. Verwertungsrecht im digitalen Raum zurückgeführt werden. Während die technischen Probleme des DRMs erst in den folgenden Abschnitten diskutiert werden, ergibt sich bereits nach dem ersten Definitionsversuchs des Begriffes eine Erblast an Problemen – insbesondere die Frage was unter dem Copyright als schützenswert eingestuft wird, verursachte Definitionsprobleme.

Ein Beispiel, entnommen aus [USC86]: Ein Computerprogramm, das den Mittelwert einer Menge von gegebenen Zahlen berechnen soll. Entsprechend der obigen Ausführungen kann ein dafür erstellter Algorithmus über das Copyright geschützt werden. Wenn man sich jedoch die möglichen Repräsentationsformen (a-f) eines solchen Algorithmus anschaut (Abbildung 4), wird es problematisch eine Einordnung in „schützenswert“ und „Allgemeingut“ zu treffen, wenngleich die Funktionsweise der Programme bei allen Beispielen die gleiche ist.

a) To compute an average, sum the numbers in the set to be averaged, and divide by the number of items in the set

b) $$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x$$

c)



d)
```
Get a data item
  add item to total
  increment index
  if no more item, continue:
    else go to get
divide total by index
```

e)
```
Call average
10, 12, 17, 22,8
```

f)
```
01100100110101O1
01001001101100l1
1001110101001100
100110100110...
```

Abbildung **4** – **Repräsentationsformen eines Algorithmus**

3.3 Komponenten eines DRM Systems

Im Folgenden werden grundlegende Komponenten eines DRM System genannt und deren Einsatzmöglichkeiten diskutiert.

3.3.1 Identifikation und Metainformationen

Hauptbestandteil eines Onlinepublikationssystems stellt die bereits in Abschnitt 2.4.1 beschriebene eindeutige Kennung eines digitalen Wissensobjekts dar. Die Notwendigkeit innerhalb eines DRM System ergibt sich daher implizit. Während die bisherigen Ausführung Probleme bezüglich der Referenzierbarkeit behandelt haben, ergeben sich innerhalb des DRM neue Problemfelder.

So muss beispielsweise definiert werden, welche Werke gleich sind. Zur Veranschaulichung soll folgendes Beispiel dienen: Nehmen wir an, es gäbe von Schillers „Die Räuber“ genau zwei Ausgaben, eine Taschenbuchausgabe und eine in edlem Leder gebundene. Bezogen auf den textuellen Inhalt sind beide identisch. Nehmen wir weiter an *A* referenziert die Taschenbuchausgabe und *B* die in Leder gebundene. Unfraglich wird jeder damit einverstanden sein, dass die gleichen Texte mit zwei unterschiedlichen Identifiern referenziert werden können. Bei einer Bestellung des Kunden X im Buchhandel muss klar zwischen beiden identifziert werden, entsprechend ist die tatsächliche Notwendigkeit für zwei unterschiedliche Identifier gegeben. Ein Student Y der Germanistik, dem es letztlich nur auf den Text ankommt, wird keinen Unterschied zwischen A und B feststellen können.

Dieses kleine Beispiel veranschaulicht bereits, dass „gleich“ nicht grundlegend definiert werden kann. So kann ein Identifier die Anfrage ist A = B nicht beantworten. Diese Anfrage kann nur mit weiteren Informationen wie z.B. „Ist A = B, bezogen auf den Inhalt“ beantwortet werden. Tatsächlich ergibt sich daraus ein Problem der Identifikation innerhalb des DRMs. Digital können theoretisch unendlich viele unterschiedliche Versionen bereitgestellt sein, die von einem DRM verwaltet werden müssen. So ist beispielsweise eine Kopie dieser Arbeit grundlegend nicht gleich dem Original. Selbstverständlich ist der Inhalt identisch, aber das Erstellungsdatum der Datei wird sich beispielsweise unterscheiden.

Auch stellt sich die Frage, ob zwei Absätze, die getrennt vom System bereitgestellt wurden, mit dem Dokument gleichzusetzen sind, das aus genau diesen zwei Absätzen besteht. Neben der Frage nach Gleichheit ergibt sich aus dem letzten Absatz bereits das weitere Problem der beliebigen Granularität.

Ein Identifier für digitale Objekte müsste in der Lage sein, eine Granularität bis auf Bitebene zu erlauben. Beispiele unterschiedlicher Granularitäten in der analogen Welt sind ISBN, ISTC und BICI. Während ISBN das Werk identifiziert, können einzelne Kapitel oder Absätze über BICI referenziert werden. ISTC wiederum abstrahiert vom Transportmedium auf den textuellen Inhalt und ist als Lösung des oben genannten Problems der zwei Versionen eines Werkes zu sehen, dadurch ist ISTC in der Hierarchie oberhalb von ISBN einzuordnen. Praktikabel bleibt dieses Konzept nur, wenn man es darauf einschränkt, dass ein Identifier immer dann differenzieren können muss, wenn es notwendig ist („Functional Granularity“).

Diese Notwendigkeit stellt ein Problem des Digitalen Rechte Managements dar, da bei verteilt entstehenden Werken im Hinblick auf die Rechteverwertung die Differenzierung tatsächlich auf Satzebene stattfinden kann – Im Extremfall des living books auf Zeichenebene. Auch die Vergabe unterschiedlicher Identifier wiederum birgt das Problem der Zusammengehörigkeit. So beschreiben die Identifier A.1 und A.2 zwar etwas Ähnliches, doch in welcher Beziehung diese zwei einzeln identifizierten Werke stehen, wird nicht klar. Es ist nur klar, dass sie nicht gleich sein können, da ansonsten der gleiche Identifier vergeben worden wäre.

Unter all den bislang verfügbaren Identifern – wie URL, PURL, DOI, u.a. – zeigte sich, dass das, auch vor dem Hintergrund des DRM entwickelte, DOI System die bislang umfassendste Implementation eines solchen Identifiers darstellt [Pas03]. Es ist nicht nur eine Implementation der Konzepte von URN und URI, sondern basiert auch auf einer sozialen Infrastruktur, die Persistenz garantieren kann.

3.3.2 Authentifizierung

Während sich 3.3.1 auf die eindeutige Identifizierung eines Wissensobjekts bezogen, sind die Authentifizierungsmöglichkeiten des Nutzers und damit dessen eindeutige Identifizierung gegenüber dem System innerhalb eines Digital Right Managements ebenfalls von tragender Bedeutung. Neben dem direkten Angriff auf die Schutzmechanismen steht dem Angreifer ein weiterer Weg offen, innerhalb des DRM System an das gewünschte Objekt zu gelangen: Er gibt sich als legitimer Nutzer aus und gelangt so über eine falsche Identität an geschützte Inhalt.

Während im Alltag eine Authentifizierung des Gegenübers bei vertraulichen Geschäften meist „einfach“ und praktisch automatisch über biologische Merkmale – Stimme, Gesichtsform u.s.w. – geschieht, stellt sich dies innerhalb des digitalen Raums als problematisch heraus. Erst in jüngster Zeit werden Biometrische Systeme tatsächlich eingesetzt. Deren Einsatz innerhalb von Abrechungs- bzw. Internetidentifikationssystem scheint allerdings auch heute noch wegen des nicht geringen Aufwands in weiter Ferne, nicht zuletzt müsste jeder potentielle Kunde über eine entsprechend Hardwareausstattung verfügen.

In der Regel wird zur Authentifizierung innerhalb des Internets eine Kombination aus frei wählbarem Benutzername und ebenfalls frei wählbarem Passwort genutzt. Eine Person die beide Informationen besitzt legitimiert sich gegenüber dem anbietenden Server als jene Person, als die sie sich ausgibt. Aufgrund der Ungebundenheit zwischen Daten und realer Person muss dies nicht der tatsächliche Inhaber des Benutzernamens sein. Ergebnis ist letztlich eine Authentifizierung über Wissen. Bekannte Beispiele für solche Verfahren sind ebay.com, amazon.com oder auch die eigene Hausbank.

Während dieses System für den Nutzer zweifelsohne das akzeptabelste darstellt, verfügt dieses System über mehrere Schwachpunkte. Zum einen kann es sein, dass sowohl Benutzername wie auch Passwort auf herkömmlichem Wege ausgespäht werden, vergleichbar mit dem Ausspähen des EC Pins. Um innerhalb des digitalen Systems zu bleiben, sollen im Weiteren ausschließlich Angriffsmöglichkeiten digitaler Natur aufgezeigt werden. Beispielsweise verwenden viele passwortbasierte Zugangssysteme HTTP als Protokoll anstellte des sichereren SHTTP oder SSL/TLS.

Bei der Verwendung des ungeschützten HTTP Protokolls ergibt sich an jedem Computer, der sich innerhalb der Verbindungsstrecke des lokalen Computers zum Daten liefernden Server befindet, die Möglichkeit die gesendeten/empfangen Daten mitzuhören. So werden sowohl Passwort als auch Benutzerkennung während der Übertragung mitgeschnitten und können später für die eigene Verwendung genutzt werden. Zu erwähnen bleibt, dass die Hausbank meist noch eine weitere Sicherheitsschwäche aufweist: Der Benutzername ist nicht frei wählbar, sondern in einer klar definierten Weise über die öffentlich zugänglichen Kontoinformationen generiert. Ein möglicher Angriff gestaltet sich in einem solchen Fall noch einmal einfacher, da der Benutzername der zu unterwandernden Person bereits im Voraus bekannt ist.

Sicherer wird das System über die Verwendung von SHTTP oder den populäreren Verbindungen über SSL. Bei dem von Netscape initiierten Standard handelt es sich um einen zusätzlichen Layer zwischen TCP und HTTP, der dafür sorgt, dass über TCP nur noch im Voraus verschlüsselte Informationen ausgetauscht werden. Die Verschlüsselung basiert auf dem während des SSL Handshaking ausgehandelten Schlüssel:

- Der Server sendet dem Client seinen öffentlichen Schlüssel (innerhalb eines SSL Zertifikats)
- Der Client verschlüsselt eine geheime Zufallszahl mit dem öffentlichen Schlüssel des Servers und sendet diesen zurück. Diese Zufallszahl wird von Server und Client dazu genutzt, die symmetrischen Schlüssel zu berechnen.

Prinzipbedingt ist die Kommunikation so, davon ausgehend, dass die Verschlüsselung sicher ist, geschützt vor unerwünschten „Lauschern". Dennoch bietet auch SSL Angriffspunkte. Neben der in Abschnitt 3.3.3 noch beschriebenen prinzipiellen Problematik von Verschlüsselungsalgorithmen bietet sich für SSL Verbindung die man-in-the-middle Attacke[24] an. Durch diesen Angriff können die nun eigentlich geschützten Daten auch weiterhin ungehindert ausgespäht und missbraucht werden. Auch wenn ein man-in-the-middle Angriff vom Benutzer anhand des damit verbundenen Austauschs des Zertifikates erkannt werden kann, bleibt die Gefahr in Bezug auf die Mehrzahl der potentiell unbedarften Nutzer bestehen.

Eine weitere Möglichkeit der Authentifizierung wäre denkbar, indem der Server eine mit dem öffentlichen Schlüssel des Nutzers verschlüsselte Passwortabfrage beim Benutzer startet. Diese wird von einem speziellen Client mittels des privaten Schlüssels des Nutzers dechiffriert. Der Nutzer gibt die entsprechenden Authentifizierungsinformationen ein, verschlüsselt diese mit dem öffentlichen Schlüssel des Servers und versieht die entstandene, chiffrierte Antwortnachricht zusätzlich mit einem Message Authentication Code (MAC) – beispielsweise MD5 – der wiederum mit dem öffentlichen Schlüssel des Servers verschlüsselt wird. Somit wäre es dem Nutzer möglich sich eindeutig gegenüber dem Server zu authentifizieren, wobei der Server wiederum in der Lage wäre eindeutig sicherzustellen, dass die ursprüngliche Passwortabfrage nur vom gewünschten Nutzer dechiffriert wurde, die Antwortnachricht des Nutzers noch der von ihm ursprünglich abgesanten Nachricht entspricht und dass der Nutzer die Passwortabfrage positiv beantworten konnten.

[24] Definition: http://en.wikipedia.org/wiki/Man_in_the_middle_attack

So kompliziert das Verfahren erscheinen mag, so effizient ist es zeitgleich. Ähnliche Verfahren werden beispielsweise bei elektronischen Zahlungssystemen[25] eingesetzt. Um das System weiter zu verbessern, könnte der Nutzer im Voraus beispielsweise fünf unterschiedliche Passwortanfragen speichern, die der Server nach einem Zufallsprinzip auswählt. Damit wäre sichergestellt, dass der Angreifer, sollte er denn die ursprüngliche Passwortanfrage abgefangen bzw. selbst erzeugt haben, nicht durch Zufall eine richtige Antwortnachricht senden kann. Einzige unsichere Stelle bei diesem Verfahren wäre die Festlegung der Passwortabfragen durch den Kunden zu Beginn. Dies würde innerhalb des Internets nur über einen potentiell unsichereren Kanal erfolgen können.

Neben den genannten Verfahren soll abschließend noch die so genannte Single Sign-On Variante Erwähnung finden, die in jüngster Zeit, trotz anhaltender Bedenken die Privatsphäre betreffende, an Popularität gewinnen konnte. Insbesondere das mit Microsofts Marktmacht publik gemachte Microsoft Passport[26] gewinnt hier zunehmend an Popularität. Erwähnenswert bleibt auch die Alternative der Konkurrenz zu Microsoft, die unter Anführung von SUN unter dem Namen Liberty Alliance Project[27] ebenfalls eine Single Sign-On Lösung anbieten.

Das Prinzip beider Lösungen ist gleich: Der Benutzer meldet sich nicht mehr bei dem jeweiligen Inhaltsanbieter bzw. DRM Systembetreiber an, sondern stattdessen bei Passport[28] als Authentication Service Provider (ASP). Dort hinterlegt der Nutzer seine persönlichen Daten, im einfachsten Fall seine E-Mail Adresse und legt ein Konto (Account) an. Hierfür vergibt der Benutzer bei seiner Passportanmeldung ein selbst gewähltes Passwort. Die spätere Authentifizierung des Nutzers gegenüber dem System geschieht direkt über Passport.

---

[25] Ähnlich beispielsweise bei beim Secure Electronic Transaction (SET) Protokoll

[26] http://www.passport.net

[27] http://www.projectliberty.com

[28] Passport als Stellvertreter für Single Sign-On Systeme genannt.

Der Benutzer meldet sich über Passport an, das DRM System erhält von Passport die eindeutige Kennung des Nutzers und erfährt, ob der Benutzer sich gegenüber Passport positiv identifizieren konnte. Bei einer positiven Identifizierung kann das DRM System den Nutzer über die zurückgelieferte, eindeutige Kennung im eigenen System anmelden. Während das System sowohl neue Vorteile, beispielsweise einen universell gültigen Login, als auch neue Nachteile, beispielsweise eine mögliche Unterwanderung der Privatsphäre, für den Nutzer eines DRM Systems bringt, wird die Angriffsmöglichkeit letztlich nur um eine Instanz verschoben, daher sind auch Single Sign-On System nicht als sicher einzustufen.

Für DRM Anwendungen als sicher einzustufenden Authentifizierungsmöglichkeiten sind bei der im Massenmarkt eingesetzten PC Architektur schlicht nicht möglich. In Kombination mit z.B. zukünftigen Identifikationssystemen innerhalb der PC Architektur – bekannt unter dem Namen TCPA Services, Palladium, NGSCB oder auch schlicht Fritz Chip – (siehe auch [BS03]) ist eine real sichere Authentifizierung des Nutzers gegenüber eines Servers möglich. Eine Beispielarbeit auf diesem Gebiet stellt [PM03] dar, welche den Teil des ASP in den sicheren Raum des NGSCB verlagert und der Nutzer somit sein eigener ASP wird. Neben der Verlagerung des ASP in das NGSCB des lokalen Rechners beugt, der Vorschlag in [PM03] über SSL/TLS und Sicherheitserweiterungen des DNS bzw. einem ausgeklügelten Challenge/Response Systems einer man-in-the-middle Attack vor.

3.3.3 Inhaltssicherung

Wichtigstes Kriterium bei der Bewertung und gleichzeitig ebenfalls eines der größeren Problemfelder eines DRM Systems dürfte mitunter die Inhaltssicherung darstellen. Die tatsächliche Inhaltssicherung innerhalb dieser Arbeit zu beschreiben, ist aufgrund der Geheimhaltungspolitik der Anbieter schlicht unmöglich. Dennoch kann das grundlegende Konzept der Inhalts- und Lizenzverwaltung zusammengefasst wiedergegeben werde.

Die grundlegenden Komponenten der Inhaltssicherung basieren auf der Dateiverschlüsselung, der Schlüsselverwaltung sowie der Zugriffsregulierung.

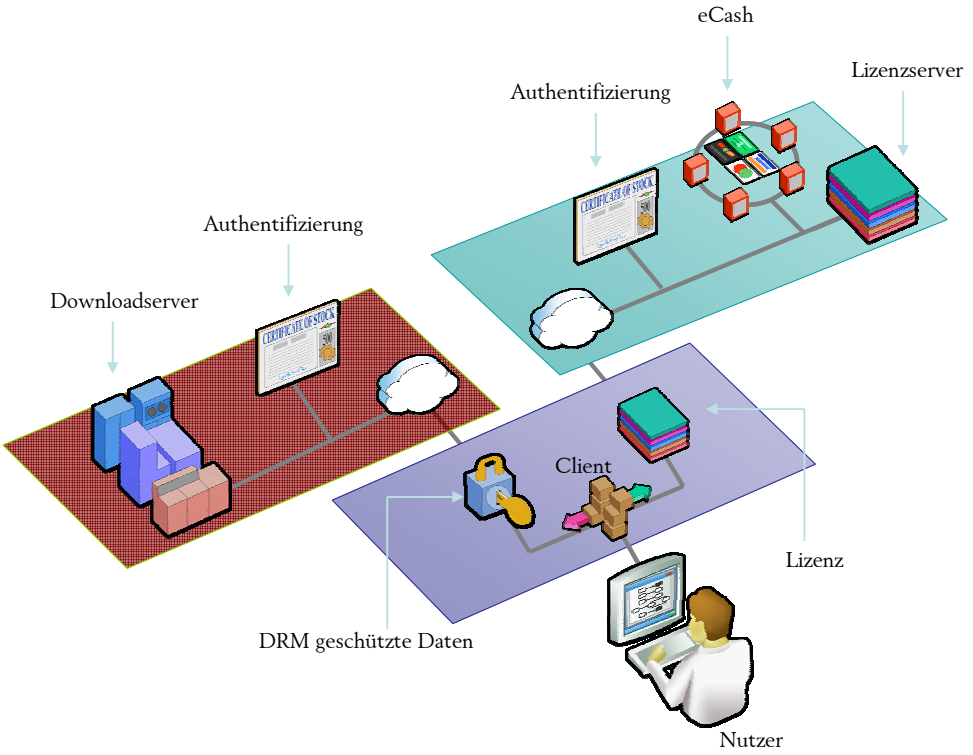


Abbildung **5** – **Zusammenspiel Download- & Lizenzserver**

Bevor die Daten im WAN zur Verfügung gestellt werden, werden diese verschlüsselt und mit Metadaten angereichert. Zusätzlich kann eine Rechte-Beschreibungsdatei zur Verfügung gestellt werden, die entweder zusätzlich auf dem Lizenzierungsserver bereitgehalten werden kann, oder die ebenfalls den verschlüsselten Daten hinzugefügt wird. Die so verfügbar gemachten Daten werden über eine digitale Signatur oder eine MAC (→ Abschnitt 3.3.2) signiert und damit vor Veränderung geschützt. Auf einem getrennten Server werden entsprechende Nutzungslizenzen verfügbar gehalten. Durch diese Trennung von Inhalt und Lizenz können die geschützten Inhalte auf beliebigen Servern verfügbar gemacht werden, was für eine große Verbreitung genutzt werden kann.

Die verschlüsselten Daten liegen nach dem Erwerb auf dem lokalen Rechner vor. Für die Nutzung der Daten wird vom Client eine entsprechende Lizenz vom Lizenzierungsserver nachgeladen. Die Lizenzierungsmöglichkeiten werden entweder in der Rechte-Beschreibungsdatei gespeichert, oder können über den Lizenzierungsserver bei Anfrage verfügbar gemacht werden. Der Client kann die verschlüsselten Daten mit der in der Lizenzdatei enthaltenen Informationen lokal – meist in eine auf den lokalen PC gebunden Version – entschlüsseln und dadurch innerhalb einer sicheren Umgebung verfügbar machen (siehe Abbildung 5).

Das beschriebene Verfahren bietet, wie prinzipbedingt alle Verfahren der Inhaltssicherung lokal verfügbarer Daten, zwei größere Schwachstellen, die es erlauben, die Inhaltssicherung grundlegend zu umgehen. Die erste Schwachstelle der Inhaltssicherung ergibt sich aus der Übermittlung des Inhalts und der damit einhergehenden Verschlüsselung des Inhalts. Die Inhalte gelangen auf den Zielrechner und werden dort lokal verschlüsselt gespeichert. Durch diese lokale Zugriffsmöglichkeit ergibt sich das Problem des analytischen Vorgehens. Letztlich muss der verschlüsselte und dadurch geschützte Inhalt lokal nutzbar sein, sei es auch über einen proprietären Client.

Durch das Vorhandensein des Clients und damit letztlich des Algorithmus und der verschlüsselten Inhalte, kann der Nutzer gezielt den Algorithmus respektive Client „angreifen“ und über diesen Weg an die Inhalte gelangen. Ausgehend von hinreichend sicheren Algorithmen (AES, RSA) mit entsprechender Schlüsselgröße könnte die Aussage getroffen werden, dass ein solcher Angriff entweder nicht zum Ziel führen würde oder aber aufgrund des Aufwands wirtschaftlich unrentabel wäre. Theoretisch wäre dieser These beizupflichten, dennoch zeigt [And94] (hier bezogen auf Bankautomaten) die Schwachstelle eines jeden Algorithmus: dessen Implementation.

Selbst unzählige, von Cryptoexperten überprüfte, Chiffrieralgorithmen werden unwirksam, wenn deren Implementation es entweder erlaubt einen direkten Angriff auf den Algorithmus durchzuführen oder über kontrollierte Nebeneffekte, die durch Implementationsfehler von Extern ausgelöst werden können, dessen Schlüssel erarbeitet werden kann. Es bleibt festzuhalten, dass nicht der Algorithmus an sich „angegriffen" wird – dieser ist im Idealfall hinlänglich bekannt – sondern dessen fehlerhafte Implementation.

Die zweite Sicherheitslücke ergibt sich wiederum aus der lokalen Verfügbarkeit der Inhalte und deren „Wiedergabemöglichkeit". Unter der Annahme, dass den Angriffen auf die Implementation des Algorithmus effektiv entgegengewirkt werden kann, müssen die lokal vorliegenden Inhalte über einen lokalen Client verfügbar gemacht werden. Dieser kann im Extremfall[29] für jede Wiedergabe/jedes Anzeigen eine Lizenz von einem Lizenzserver herunterladen. Bei diesem Vorgang wäre ein Angriff auf diesen Prozess denkbar, und wurde auch bereits praktisch umgesetzt. So kann mit der empfangen Lizenz der Inhalt, wiederum im Extremfall, für eine einmalige Nutzung verfügbar gemacht werden. Diese Lizenz wird von einem speziellen Client dazu genutzt, den geschützten Inhalt einmalig freizulegen und den ungeschützten Datenstrom abzufangen.

[29] Im Extremfall deswegen, da die Akzeptanz eines solchen Falles, also dem tatsächliche Onlinegang für jede Wiedergabe, innerhalb der Allgemeinheit aufgrund von Sicherheitsbedenken gegenüber dem Schutz der eigenen Privatsphäre nicht gegeben sein dürfte. Zusätzlich dürfte es in solch einem Fall als sicherer anzusehen sein, dass die Inhalte nur online verfügbar gehalten werden, da dann ebenfalls nur eine Onlineverbindung pro Konsum notwendig wäre.

Wenngleich der Aufwand Schutzroutinen zu umgehen in jüngster Zeit verhältnismäßig aufwendig wurde, bleibt festzuhalten, dass die Inhaltssicherung annähernd unmöglich sicher zu bewerkstelligen ist. Dies liegt, wie auch schon unter 3.3.2 angemerkt, an der Entwicklungsgeschichte des PC, dessen primäres Ziel bislang nicht das Sperren von Inhalten dem eigenen Nutzer gegenüber war. Entsprechende Veränderungen werden neue Hardwaremodule bringen, die nur innerhalb eines sicheren Kanals kommunizieren.

Wobei in der Essenz festzuhalten bleibt, dass es sich bei den angedachten Lösungen um die Verlagerung der Problematik auf Hardware handelt. So werden zwar generalisierte Angriffe auf Schutzmechanismen gehemmt, gleichzeitig bietet auch diese Verlagerung keinen Schutz gegen die gezielten Angriffe eines Individuums auf die physikalisch vor ihm stehenden, schützenden Hardwarekomponenten. Dies soll allerdings, entsprechend der Pressepolitik von – beispielsweise – Microsoft, innerhalb von NGSCB und zugrunde liegendem TCPA, auch nicht dem primären Ziel entsprechen [*MICRa*].

3.3.4 Inhaltsbasierte Identifikation

Als Ergänzung der Inhaltssicherung ist die Inhaltsbasierte Identifikation anzusehen. Als Ausgangsbasis zur Inhaltsbasierten Identifikation dienen die Authentifizierungsinformationen des Kunden, der sich gegenüber dem System legitimiert. Die von ihm angeforderten Inhalte werden bei Bedarf für ihn aufbereitet. Aufbereitet meint in diesem Zusammenhang das Modifizieren des ursprünglichen Objekts mittels eines später nicht oder nur schwer erkennbaren eindeutigen Merkmals, das sich aus den Authentifizierungsdaten des Kunden sowie der Identifikation des Inhaltes ergibt. Entsprechend kann das vom Kunden bezogene Objekt später eindeutig zur Inhalts- und Kundenzuordnung genutzt werden.

Letztlich stellt die Inhaltsbasierte Identifikation über beispielsweise Wasserzeichen (Watermarking) zwar keine weitere Inhaltssicherung dar, kann dennoch als weiteres Schutzsystem klassifiziert werden. In Ergänzung zur eigentlichen Inhaltssicherung bringt die Inhaltsbasierte Identifikation, vom Standpunkt der Qualität des Schutzes, das System auf die nächste Ebene. Gelingt es dem Kunden beispielsweise die Inhaltssicherung zu umgehen, so ist der nun frei verfügbare[30] Inhalt dennoch weiterhin eindeutig dem Kunden zuzuordnen, was es für ihn letztlich unmöglich macht, die Inhalte an Dritte weiterzugeben.

Wichtig bei Einsatz eines Wasserzeichens ist, wie der Namen bereits suggeriert, die Nicht-Sichtbarkeit. Ein Wasserzeichen darf die Verwendung des ursprünglichen Inhalts weder behindern noch beeinträchtigen.

Auswirkungen eines Wasserzeichens auf ein Grafikobjekt sind in Abbildung 6 sichtbar, (a) & (c) sind nicht markierte Original, (b) & (d) die jeweils mit einem Wasserzeichen versehen Versionen.

Die Anwendung auf Inhalte wie z.B. Grafik, Ton oder auch Video gestaltet sich verhältnismäßig problemlos. Für den Einsatz bei reinen Textdokumenten ist ein Wasserzeichen zur eindeutigen Identifizierung ungeeignet. Selbst wenn der Text als Grafik ausgegeben und mit einem Wasserzeichen versehen wird, so sind mittlerweile OCR Algorithmen in der Lage auch bei niedriger Auflösung des Quellmaterials (beispielsweise 92 DPI Bildschirmauflösung) annähernd vollautomatisch Text zu erfassen. Innerhalb einer solchen Lösung wäre ein Wasserzeichen daher entfernbar. Rechtlich waren Wasserzeichen bislang problematisch, da nicht nur der Kunde über die „signierte“ Datei verfügt, sondern auch der Anbieter eine solche Datei erstellen konnte. Entsprechend konnte dem Kunden nicht eindeutig nachgewiesen werden, dass er seine Kopie vervielfältig hatte und nicht der Anbieter eine Kopie mit entsprechendem Wasserzeichen vervielfältigte.

---

[30] Im Sinne von „nicht mehr den elektronisch auferlegten Lizenzierungsschranken unterlegen“

Eine entsprechende Lösung ist das anonyme Wasserzeichen (im Paper werden die Begriffe Wasserzeichen und Fingerprint vermischt), beispielsweise in [WLL03] beschrieben.

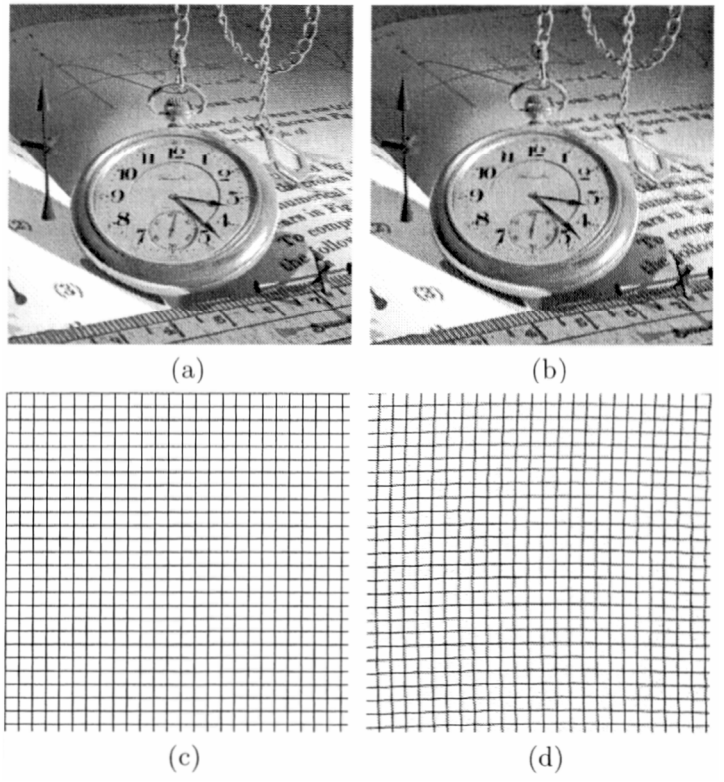


Abbildung 6 – **Auswirkungen eines Wasserzeichens**

Ebenfalls zur Durchsetzung der Verwertungsrechte kann eine weitere Technik der Inhaltsbasierten Identifikation eingesetzt werden: Fingerprints. Fingerprints, auch Signatur genannt, basieren auf dem Prinzip der Mustererkennung. Die vorliegenden Objekte werden über eine Mustererkennung erfasst und die markant beschreibenden Merkmale als Fingerprint gespeichert. Ein Fingerprint wird ergänzend noch mit entsprechenden Metadaten verknüpft, die dem Nutzer auf Basis eines Fingerprints weiterführende Information bereitstellen können.

Eine Anfrage auf den Datenbestand wird nun mittels eines Anfrageobjekts durchgeführt (siehe Abbildung 7). Das Anfrageobjekt sollte die Eigenschaft aufweisen, die das Ergebnis ebenfalls aufzeigen soll. Vom Anfrageobjekt wird über die Mustererkennung ein Fingerprint erzeugt. Dieser Fingerprint wird mit den bereits erzeugten Fingerprints verglichen und die gefundenen Objekte können entweder in einer nach Ähnlichkeit geordneten Liste ausgegeben werden oder es wird Boole'sche festgestellt, ob ein Objekt in der Ergebnismenge liegt oder nicht.

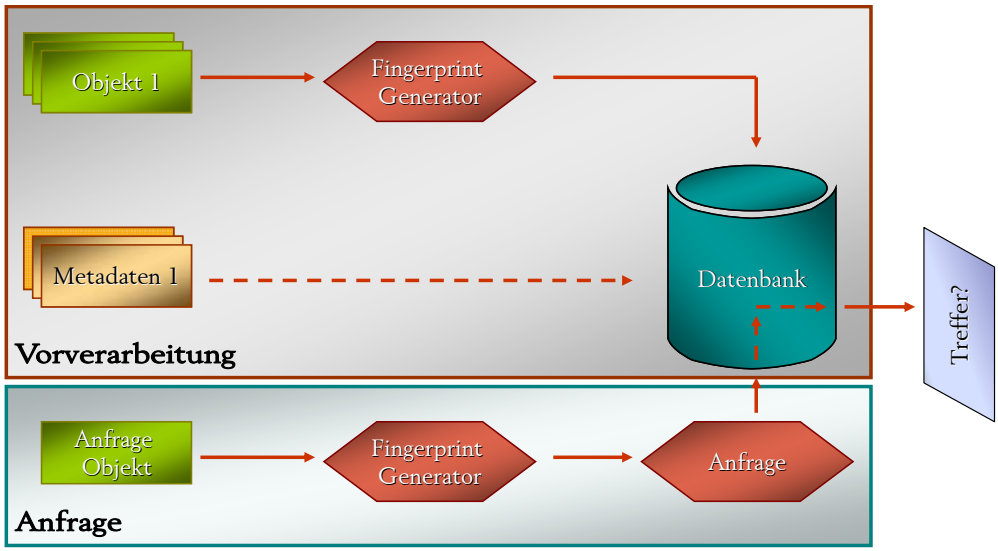


Abbildung **7** – **Schema Fingerprintverwaltung & –nutzung**

Während die eigentliche Bestimmung von Fingerprints, im Gegensatz zu Wasserzeichen, ein effizienteres Retrieval darstellt, kann die vorgestellte Technik auch für DRM eingesetzt werden. Prominentestes Beispiel eines solchen Einsatzes in jüngerer Zeit ist das so genannte Counterfeit Deterrence System (CDS), entwickelt von der Central Bank Counterfeit Deterrence Group. Wenn auch als primitives und rigides System einzustufen, sorgt das CDS dennoch wirkungsvoll für die Durchsetzung der „Verwertungsrechte" an Banknoten im digitalen Raum und ist entsprechend als Digital Rights Management System zu klassifizieren.

Das Einscannen bzw. Bearbeiten von digitalen Banknotenabbildern oberhalb einer Auflösung von 72DPI wird rigoros mit dem Hinweis auf die Verwertungsrechte blockiert [Tri04]. In diesem Fall dient „infiltrierte" Software dazu, dass eine digitale Verwertung/Bearbeitung abhängigkeitsfrei sowohl vom Verwendungszweck als auch vom Nutzer unterbunden wird.

3.3.5 Beschreibungssprache für Rechte

Eine Beschreibungssprache für Rechte – Rights Expression Language – („REL") dient prinzipiell zur Beschreibung der Rechte eines Teilnehmers an einem Objekt. Eine REL muss flexibel genug sein, Geschäftsmodelle und Regelungen der Verwertung digitaler Objekte/Inhalte unabhängig von monetären[31] Überlegungen definieren zu können. Entsprechend leitet sich auch die Definition ab.

> *„A rights expression language has to support the implementation of frameworks which enable the interoperability of DRM systems and agents on the basis of digital contracts (or digital documents)"* [Gut03].

Als zusätzliches Muss-Kriterium muss eine REL ebenfalls maschinell les- & auswertbar sein. Aus diesem Grund und der vereinfachten Austauschbarkeit basieren mittlerweile alle nennenswerten RELs auf dem XML Standard. Nennenswerte Beschreibungssprachen wären in diesem Kontext neben XrML[32] noch ODRL[33], DREL[34].

---

[31] Beispielsweise stellt die „Creative Commons" Initiative Lizenzmöglichkeiten zur Verfügung, die es Rechteinhabern ermöglichen der „Public Domain" Rechte einzuräumen, die spezielle auf ihre „Creative Works" zugeschnitten sind.

[32] Extensible Rights Markup Language. Entwickelt von ContentGuard, ein Joint Venture aus Xerox und Microsoft

[33] Open Digital Rights Language. Entwickelt von der ODRL Initative (http://www.odrl.net)

[34] Digital Rights Expression Language. Entwickelt von der DREL Initiative, die wiederum vom IEEE Learning Technology Standard Committe gegründet wurde.

Kleinere oder veraltete Lösungen wie Digital Property Rights Language (Xerox), Custom Digital Rights Language (Octalis) oder Extensible Media Commerce Language (RealNetworks) werden im Folgenden nicht weiter beachtet. RELs basieren auf dem zugrundliegenden XML Prinzip und besitzen einen klar definierten Syntax sowie eine klar definierte Semantik, wobei der Syntax als Rights Language Concept (RLC) und die Semantik, welche letztlich einen Menge gültiger „Sätze“ meint, als Rights Data Dictonary (RDD) bezeichnet werden. Ein REL dient zur Verwaltung letztlich dreier Bereiche: Rechte, Güter & Teilnehmer [Ian01].

- **Rechte** — werden als Formulierungen verstanden, die den Zugriff auf oder eine bestimmte Nutzung eines Objekt erlauben.
- **Güter** — hierbei handelt es sich um das von den definierten Rechten betroffene Objekt. Beispielsweise eine Beschreibung mittels eines DOI zur Identifikation und zur Bereitstellung externer Metadaten.
- **Teilnehmer** — Regelt die Zugriffsarten basierend auf der Rolle des Teilnehmers. So erhält der Autor eines Wissensobjekts beispielsweise schreibenden Zugriff, während dem Kunde allenfalls lesenden Zugriff gewährt wird.

Um der Komplexität der unterschiedlichsten Geschäftsmodelle, insbesondere auch der noch zu erarbeitenden, tatsächlich dem digitalen Raum angepassten, gerecht werden zu können, erlauben RELs Verschachtelungen der einzelnen Basiskriterien, so dass beispielsweise Rechte in Abhängigkeit von weiteren Rechten erteilt werden können. Exemplarisch soll kurz auf XrML eingegangen werden. XrML wird deshalb näher vorgestellt, weil XrML 2.0 als die zukünftige Rechtbeschreibungssprache für MPEG21[35] gewählt wurde.

[35] MPEG-21 Requirements Version 1.3.
Dokument ISO/IEC JTC 1/SV 29/WG 11/N5232

XrML wird von drei XML Schemata beschrieben: Das XrML core schema, das XrML standard extension schema (sx) und das XrML content extension schema (cx). Nachfolgendes Beispiel ist aus [Gut03] entnommen und zeigt das Wurzelelement „`license`“. Das betroffene Objekte sowie die betroffenen Teilnehmer werden über „`resource`“ und „`principal`“ referenziert, während „`grant`“ die tatsächliche Lizenzierungsmethode beschreibt. Ein Recht wird über „`rights`“ mit den zugehörigen „`conditions`“ ausgedrückt.

Das Beispiel erlaubt dem Besitzer des x509[36] Zertifikats die Verwendung der Ressource „evoboRes“ bis zum Ende des Jahres 2010, wobei der Syntax beispielhaft und nicht XML konform ist und die entsprechenden Schemata `dsig` & `cx` nicht geladen werden.

```
<?xml version="1.0"?>
<license>
  <grant>
    <keyHolder>
      <info>
        <dsig:x509Data>
         … signature information for owner …
        </dsig:x509Data>
      </info>
    </keyHolder>
    <cx: print/>
    <cx:digitalWork>
      <cx:locator>
        <nonSecureIndirect URI="http://www.uni-konstanz.de/evoboRes"/>
      </cx:locator>
    </cx:digitalWork>
    <validityInterval>
      <notAfter>2010-12-24T23:59:59</notAfter>
    </validityInterval>
  </grant>
</license>
```

[36] Entsprechend dem ISO Authentication Framework

3.3.6 Elektronische Abrechungssysteme

Zum legalen Erwerb einer Lizenz im Rahmen von DRM geschützten Inhalten gehört die – ebenfalls im digitalen Raum stattfindende – elektronische Abrechnung. [AJSW97] unterscheidet die verfügbaren Systeme in zwei grundlegende Kategorien: Bargeld ähnlich und Scheck ähnlich, während [SS03] diese granulierter klassifiziert in Online oder Offline, Zahlungszeitpunkt (Pre-Paid, Pay-Now & Pay-Later), Hardware- oder Softwarebasiert, Anonym oder nicht anonym, In-Band oder Out-Band Authorisierung, Verschlüsselt oder unverschlüsselt & Probabilistisch oder Deterministisch.

Abbildung 8 zeigt das grundlegende Schema eines elektronischen Abrechnungssystems: Das Institut des Nutzers stellt dem Nutzer ein Zahlungsmittel zur Verfügung. Dieser erwirbt beim Anbieter mittels dieses Zahlungsmittels das geschützte Objekt. Der Anbieter wiederum leitet das vom Kunden erhaltene Zahlungsmittel an sein Institut weiter. Dieses wiederum erhält vom ausstellenden Institut die entsprechende monetäre Buchung. Wobei die Begriffe Institut und Zahlungsmittel abhängig von der tatsächlichen Implementation sind.

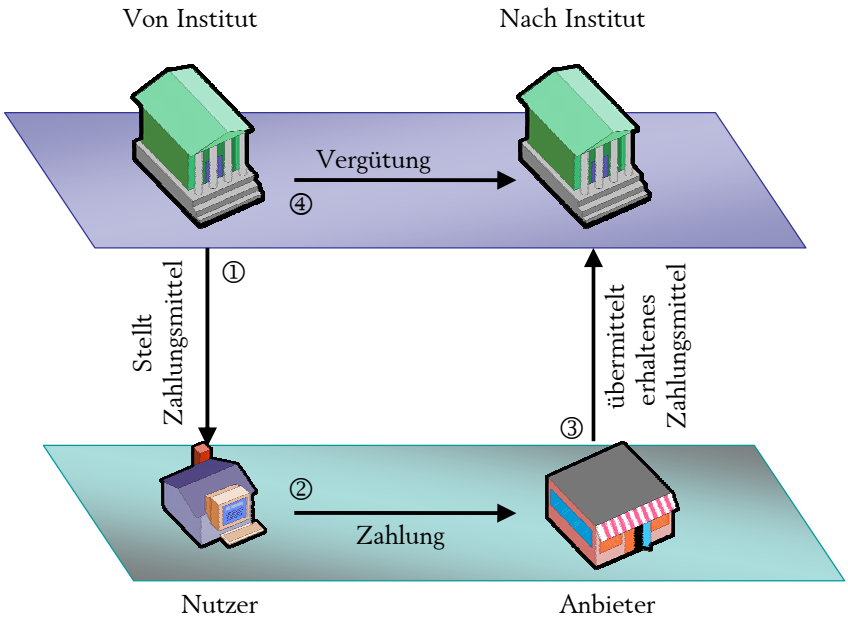


Abbildung **8** – **Schema eines elektronischen Abrechungssystems**

Grundlegende Funktionalitäten die ein elektronisches Abrechungssystem, von der Abrechnung selbst abgesehen, bieten muss, sind [AJSW97]:

- **Integrität & Autorisation** – Das innerhalb des Systems befindliche Geld kann sich nicht durch eine Aktion eines Teilnehmers erhöhen. Einem Teilnehmer darf ohne seine ausdrückliche Autorisierung kein Geld abgebucht werden.
- **Diskretion –** Teilnehmer können die Vertraulichkeit einzelner Informationen festlegen. Zusätzlich kann vollständige Anonymität gefordert werden.
- **Verfügbarkeit & Ausfallsicherheit** – Eine Transaktion muss zu jeder Zeit durchführbar sein, darf nur vollständig abgeschlossen werden und darf sich zu keinem Zeitpunkt in einem nicht definierten Zustand befinden.

Standardtransaktionen im normalen Zahlungsverkehr sind in dieser Arbeit nebensächlich und werden nicht weiter betrachtet. Für DRM interessant sind die so genannten Micropayments, die der beliebigen Granularität der online verfügbaren Inhalte Rechnung tragen. Die Idee der Micropayments basiert darauf, dass die Nutzung von komplexen Verschlüsselungs- und Autorisationssystemen bei einer Vielzahl von Zahlungen im Mikro-/Minibereich (< € 5.-) unpraktikabel wird – beispielsweise der Einsatz des SET37 Protokolls. Entsprechend geht man bei Zahlungen in diesem Bereich Kompromisse bei der Sicherheit zugunsten der Praktikabilität ein. Beispielhaft sei das μ-IKP System genannt, bei dem letztlich eine Einweg Hashfunktion Verwendung findet [AJSW97]. Eine Funktion $f()$ kann informal dann als Einwegfunktion bezeichnet werden, wenn es schwer ist, einen Wert x zu einem als $y = f(x)$ gegebenen Wert zu finden. Die Variable x ist hierbei das Urbild von y.

[37] SET: Secure Electronic Transaction. Entstand aus den Systemen Secure Transaction Technology (STT) [Visa] und Secure Electronic Payment Protocol (SEPP) [MasterCard]. SET dient als Protokoll für Kreditkartenzahlungen.

Bei einer gegebenen Funktion f(), wählte der Nutzer zufällig einen Anfangswert X und berechnet rekursiv

$$A^0(X) = X$$
$$A^{i+1}(X) = f(A^i(X))$$

Die Werte $A_0, \ldots, A_{n-1}$ werden als Coupons bezeichnet, und erlauben es dem Nutzer n Micropayments mit einem festgelegten Wert v zu tätigen. Nun sendet der Nutzer dem Anbieter A_n und v auf einem sicheren Weg. Dieser überprüft, beispielsweise mittels seiner Bank, dass A_n tatsächlich auf einer gültigen Urbildkette beruht, die zur Zahlung verwendet werden kann. Bei einer positiven Antwort werden die weiteren Komponenten $A_{n-1}, A_{n-2}, \ldots, A_0$ der Kette dem Anbieter zugänglich gemacht. Um die Zahlung abzuschließen, übermittelt der Anbieter die partielle Kette $A_i, \ldots, A_j$ $(0 \leq i \leq j \leq n)$ an seine Bank und bekommt $v(j-i)$ gutgeschrieben.

Ein weiteres wichtiges Ziel, auch im Rahmen der User Rights Management, betrifft die Anonymität der Zahlungen. Diese kann im Rahmen der elektronischen Abrechnung wiederum in zwei Arten unterschieden werden: Verfolgbarkeit (traceability) und Verknüpfbarkeit (linkability). Während man mit Verfolgbarkeit die Möglichkeit eine Zahlung bis zum Nutzer zurückzuverfolgen bezeichnet, meint man mit Verknüpfbarkeit die Möglichkeit zwei getrennte Zahlungen eines Nutzers zusammenzuführen. Um Anonymität sicherzustellen kann beispielsweise eine blind signature [SS03] genutzt werden. Ebenfalls anonym nutzbar sind Pre-Paid Lösungen oder auch die beispielsweise unter dem Label T-Pay angebotene Möglichkeit über einen Telefonanruf zu einer speziellen Nummer, die Zahlung durchzuführen.

Interessanterweise zeigte sich in einer Benutzerbefragung, dass Nutzer über die mögliche Verfolgbarkeit und Verknüpfbarkeit innerhalb von elektronischen Zahlungssystemen weniger (13,5%) besorgt sind, als beispielsweise über die Sicherheit der Transaktion (98,4%) [Abr01]. Diese Aussage wird auch durch [Odl03] gestützt. Dort wird beschrieben, dass Privatsphäre zwar von potentiellen Nutzern als wichtig erachtet wird, gleichzeitig allerdings sind die gleichen Nutzer nicht gewillt, Maßnahmen zu ergreifen ihre Privatsphäre zu schützen und geben freiwillig und unbedarft Informationen über sich heraus.

3.4 Zukünftige Entwicklungen des DRM

Die zukünftigen Entwicklungen des DRMs abzuschätzen fällt schwer, halten doch die jeweiligen Anbieter ihre detaillierten Pläne geheim. Klar abzusehen ist jedoch die Ausweitung des Schutzes über DRM auf weitere Gerätesparten wie z.B. Mobilfunkgeräten. Unter dem Schlagwort „Mobile DRM“ werden hier entsprechende Lösungen entwickelt, die beispielsweise in bereits existierende Verwaltungssysteme integriert werden können und zusätzlich Inhaltes- & Lizenzierungsformen für Kleinstgeräte bereitstellen. Neben Mobile DRM scheint sich ein weiterer Trend aufzuzeigen, der mit den Reglementierungen in Adobes PhotoShop und Jascs Paint Shop Pro begonnen hat: die Ausweitung von DRM auf Standardanwendungen. Wie in Abschnitt 3.3.4. ist der digitale Verwertungsschutz – wobei eine mögliche Verwertung von Geldscheinen im herkömmlichen Sinne eher fragwürdig scheint – für Geldscheine bereits fester Bestandteile dieser Programme.

Denkbar sind auch weitaus weitgreifendere Systeme, wie sie z.B. in Microsofts Office 2003 bereits verfügbar sind. Der dort unter dem Menüpunkt „Berechtigungen“ erreichbare und z.Zt. nur in der für Firmenkunden gedachten Professional Version verfügbare Informationsrechte-Dienst (IRM)[38], ermöglicht in Zusammenarbeit mit dem Windows Rights Client eine feinstufige Regulierung der auf den Inhalt anwendbaren Funktionen. Während diese in der Version 2003 als firmen-internes System vermarktet wird, das dem Schutz von geheimen Firmenunterlagen dienen soll, stellt sich der Verwendung als DRM nichts entgegen. So bleibt abzuwarten, in wieweit Microsofts Office dazu dienen kann, DRM in Standardanwendungen zu implementieren. Insbesondere die Entwicklungen innerhalb der Office Suite sollten beobachtet werden, da diese zum einen immer noch den De-facto-Standard darstellt und zum anderen einen leichten Einstieg in DRM bieten kann. Dokumente werden aufgrund des De-facto-Standards mit beispielsweise Word erstellt, diese wären somit bei einer Ausweitung der bereits vorhanden Funktionalitäten problemlos in digital geschützte Dokumente zu überführen.

Proprietäre Algorithmen dürften innerhalb von DRM System weiterhin bereits als erster Schutzmechanismus angesehen werden. So zeigt beispielsweise die Analyse des Programms UNFUCK deutlich, wie sehr Microsoft bei der Implementation Ihres DRM Schutzes für Multimediainhalte auf den Verwirrungsfaktor setzte [HW03]. Unzählige Aufrufe innerhalb der Schutzmechanismen führen in leere Methoden, unzählige weitere führen in Zykeln wieder zum Ursprung [HW03]. Während der Schutz über proprietäre Algorithmen zu null konvergiert, sorgt der Einsatz solcher zu einer Herstellerabhängigkeit innerhalb DRM geschützter Materialien. Was wiederum zur Folge hat, dass der Nutzer an einen vom Hersteller gelieferten Client zur Nutzung von geschützten Inhalten gebunden ist und nicht, wie gewohnt, das für ihn angenehmste Programm nutzen kann.

[38] http://www.microsoft.com/office/editions/prodinfo/technologies/irm.mspx

Ein weiteres Ziel der Anbieter dürfte dennoch die zukünftige Standardisierung des Schutzes darstellen, was in der Folge die Zugangshürden senken würde. So zeigt die Gründung des Content Reference Forum (CRF) [Jur03], dass die Notwendigkeit der Standardisierung erkannt wurde und diese nun auch verstärkt vorangetrieben werden soll.

Neben Standardisierung, neuen Einsatzgebieten und Senkung der Einstiegshürden, scheinen zukünftig auch neue Verbreitungswege für digital geschützte Inhalte interessant zu werden, so zeigt [OZL03] erste Möglichkeiten der Verbreitung über Peer-to-Peer (P2P) Netze – ironischerweise z.Zt. meist von einem Klientel verwendet, dass die Rechteverwerter mit „Pirat“ umschreiben – und baut dabei auf dem P2P Payment Protocol auf.

Interessanteste zukünftige Neuerung dürfte die Einführung der unter dem Namen „Trusted Systems“ propagierten Hardwareerweiterungen des Personal Computers werden. Während das Trusted Platform Module (TPM) zwar auch abseits des PCs eingesetzt werden kann, ist er vorerst das primäre Ziel. Festgehalten werden muss, dass eine Trusted Platform nicht zwangsläufig und ebenso wenig ausschließlich für DRM eingesetzt werden muss, dennoch erlaubt und vereinfacht sie den Einsatz von DRM auf höherer Ebene. Durch die Sicherung der Hardware und der damit einhergehenden Sicherung der Kommunikationswege der Hardware über Zertifikate entstehen neue, wesentlich effektivere Wege des Schutzes digitaler Inhalt über Digital Rights Managmement (Vergleiche [KG03] und [*EFFTa*])

Kapitel 4

User Rights Management

4.1 Definition

Während innerhalb des Copyrights zwar *fair use* [39] definiert ist, besitzt DRM durch die Möglichkeit jede aktuelle Nutzung von Wissensobjekten in beliebig skalierbarer Größe genau zu registrieren und abzurechnen bzw. genau festlegen zu können, in welchem Ausmaß die angebotenen Wissensobjekte bzw. Informationsprodukte überhaupt genutzt werden können, das Potential die Prinzipien des *fair use* zu unterlaufen. Zweifelsfrei ersichtlich scheint zu sein, dass eine solche Unterwanderung weder im privaten noch im öffentlichen Sektor gewünscht sein kann, würde doch damit über Bord geworfen, was sich seit Jahrzehnten als „funktionierend" herausstellte.

Entsprechend propagiert Kuhlen in [Kuh02e] und [Kuh03a] das User Rights Management, kurz URM als Gegenstück zum Digital Rights Management und schließt dabei in [Kuh03a] einen Zusammenhang zwischen DRM für die Sicherstellung der Verwertungsinteressen und URM für die Sicherstellung der Nutzungsinteressen. In der Literatur – auch in den genannten Aufsätzen – wird keine klare Definition des Begriffs URM, sondern lediglich eine Beschreibung der davon betroffenen Teilaspekte, geliefert. In Anlehnung an die in 3.1 angeführte Definition für DRM

"*DRM includes everything that someone does with content in order to trade it*"

---

[39] In der deutschen Gesetzgebung entspricht dies den „Schranken des Urheberrechts" (§45-63 UrhG), die dem öffentlichen, aber auch berechtigten privaten Interesse an einer gewissen Freizügigkeit bei der Nutzung geistiger Produkte Rechnung tragen sollen. [Kuh02b]

könnte URM in einer davon abgeleiteten Definition als

„URM provides everything that someone needs in order to trade content in a manner acceptable for both parties – a fair manner"

beschrieben werden. Wenn auch die gelieferte Definition großen Interpretationsspielraum bietet, so werden die unter URM im Allgemeinen verstanden Grundsätze abgebildet und ein gewisser Freiheitsgrad in der Auslegungsmöglichkeit wird beibehalten. Letztlich ist auch an einem Handel DRM geschützten Gütern nichts auszusetzen, wenn beide Partner den Handel als fair – im Gegensatz zu „aufgezwungen" – empfinden und somit durch den Einsatz des URMs eine informationelle Symmetrie erstellt werden konnte. In dieser Situation ist auch die gegebene Definition zu verstehen. Sie bildet zweifelsfrei die mögliche Eigenständigkeit des URMs nicht ab. Davon ausgehend, dass URM prinzipiell dazu gedacht sein soll, die Symmetrie zwischen Verwertungsinteressen und Nutzungsinteressen wieder herzustellen, scheint es jedoch nur logisch, eine entsprechende Definition für URM zu geben, die im Hinblick auf den bereits erfolgten Einsatz von DRM folgt.

4.1.1 Erläuterungen

Tatsächlich scheint die gegebene Definition trotz ihrer Unscheinbarkeit, auch im Hinblick auf die noch beschriebenen Einsatzbereiche des URM, ein umfassendes Umdenken der Verwerter zu erzwingen. So ist beispielsweise im Rahmen von URM immer wieder von der Privatkopie die Rede, die durch die technischen Maßnahmen des DRM verhindert werde, obwohl diese durch den Kauf des entsprechenden Rechts grundsätzlich erworben wurde [Kuh03a (Argumentation Ulf Müller)]. Ein DRM System kennt aber weder den Begriff der Privatkopie noch die damit verbundenen technischen Notwendigkeiten eine solche im „gewohnten" Rahmen, nämlich ohne technische Einschränkung, zu ermöglichen. Der gewohnte Rahmen jedoch „sprengt" letztlich genau aus diesem Grund die Möglichkeiten des herkömmlichen DRM.

Gerne wird URM auch als Möglichkeit gesehen, die althergebrachten Geschäftsmodelle der Verwerter zu novellieren und im Rahmen der Einführung von URM entsprechende, auf die Gegebenheiten des digitalen Raums zugeschnittene Geschäftsmodelle parallel einzuführen.

So stellt M. S. Manasse in seinem Aufsatz „Why Rights Management is Wrong" fest: „ *The question before us is not about how to protect the bits, but how to protect the investments in creation of the bits, and how best to preserve the relationships between people and content.*" und nimmt damit Bezug auf "*History has shown every content-protection scheme invented for consumer-grade goods to have almost no impact on piracy, and little impact on casual copying, except when it has doomed the technology carrying it. This is inevitable.*" [*MANAa*]. Zum gleichen Schluss kommt auch Kuhlen, er stellt fest, dass die Informationswirtschaft „eine unabsehbare Fortsetzung der Informationskriege zwischen Kontrolleuren und ‚Hackern'" nicht länger mit ansehen, sondern dieser mit neuen Organisations- & Geschäftsmodelle entgegentreten sollte.

4.2 Einsatzbereiche des URM

Neben intuitiv ersichtlichen Einsatzbereichen des User Rights Management wie z.B. dem Schutz grundlegender Bürgerrechte wie der Privatsphäre, oder auch der Sicherung berechtigter Privatkopien [Kuh02e], müssen in der URM Diskussion ebenfalls Argumente größerer Tragweite diskutiert werden.

So beispielsweise das wissenschaftliche, wissenschaftspolitische Argument, das den garantierten, freien[40] Austausch von Wissen und Information, gedeckt von jenem Argument, das bereits in der Einleitung Verwendung fand: Innovation der Gegenwart baut auf die der Vergangenheit auf, fordert [Les02, Kuh02c].

[40] freizügig und zu fairen, konsensfähigen Nutzungsbedingungen

Weitere für URM geforderte Vorgaben [Kuh03a]:

- Vom Nutzer oder über seine Vertretungen auszuhandelnde Vervielfältigungsfreiheiten
- Ermöglichung von Sicherungen und geräte- und orts-/raumunabhängigen Wiedergabemöglichkeiten
- Alternative Angebote zu einer individuell gestaffelten Pauschalierung
- Transparenz der Abrechnung, ohne dass das wirtschaftliche Anliegen der Rechteinhaber unverhältnismäßig beeinträchtig wird
- Bereitstellung von einfach einzurichtenden DRM/URM Verfahren auch für individuelle Urheber, die damit die Verwertung oder die lizenzierte Freigabe ihrer Produkte selbst steuern können.

4.2.1 Implementationsmöglichkeiten

Während die Implementation von beispielsweise der Bereitstellung einfach einzurichtender Verfahren, oder auch die Schaffung alternativer Abrechnungsmodelle innerhalb der bereits vorhanden DRM System mit absehbarem Aufwand umsetzbar sind, weisen die unter URM zusammengefassten Forderungen teilweise einen philosophischen Charakter auf. Wie in Abschnitt 5.4 dieser Arbeit nachzulesen, handelt es sich bei einer vollständigen Implementation von URM mehr um die Umsetzung der zugrunde liegenden Philosophie innerhalb des gesamten Projekts, als um einen stur einsetzbaren Algorithmus. Beispielsweise stellt die echte[41] Anonymität eines Nutzers innerhalb des Systems die bislang in einem DRM System angewandten Grundsätze ad absurdum. Wie können die Rechte eines Nutzers verwaltet werden, wenn noch nicht einmal dessen Identität bekannt ist? Dennoch zeigen erste Aufsätze, dass die vollständige Anonymität bei bleibender Identifikationsfähigkeit möglich ist [WLL03].

[41] „echte“ bezeichnet hier, dass der Nutzer tatsächlich anonym verwaltet wird und ihm dieser Eindruck nicht nur nach außen hin vermittelt wird, während das System im Hintergrund ein Datenfundus über ihn anlegt.

Bleibt festzuhalten, dass URM selbst kein eigenes System sein kann, sondern es sich bei einem System mit URM im Kern um eine DRM System handelt, das auf einer unterschiedlichen Grundsatzannahme fußt und um entsprechende Anrechungswege erweitert werden kann (siehe Abschnitt 4.2.3)

4.2.2 Rights Locker

Um der Forderung nach Geräte-/Orts- & Raumunabhängigkeit entgegen zu kommen, setzt Digital World Services innerhalb ihrer Plattform ADo²RA auf das Konzept des Rights Locker[42]. Die vom Nutzer erworbenen Rechte/Lizenzen werden zentral gespeichert und sind geräte-/orts- & raumunabhängig abrufbar. Ergänzend zur Vorhaltung der Rechte, wird ebenfalls ein „User Rights Backup" durchgeführt, das neben den Rechten des Benutzers auch dessen erworbene Objekte serverseitig sichert.

4.2.3 Anrechnungs- & Mehrwertmodelle

Ebenfalls in das Gebiet des User Rights Management fallen individuelle, leistungsbezogene Anrechnungs-/Creditingsysteme. Entsprechende Systeme stellen sicher, dass aktiv, selbst erbrachte Mehrwertleistungen zum ursprünglichen Objekt den Erhalt so genannter Credits nach sich zieht. Diese Credits können wiederum zum Erwerb weiterer Objekte innerhalb des Systems genutzt werden. Neben dem auf Credits beruhenden, zwar nicht zwangsläufig, aber dennoch wahrscheinlich monetären System, könnten sich gerade im wissenschaftlichen Umfeld auch reputative Anrechnungsverfahren als durchaus sinnvoll erweisen. So könnte, ein entsprechendes Umfeld vorausgesetzt, über die aktive Teilnahme an der Gemeinschaft der Status und somit die Reputation eines Teilnehmers definiert werden, ohne dass weitere monetäre Flüsse eingeführt werden müssten. Auf Basis dieser Anrechnung könnten mit der Schaffung von Mehrwerten und der Nutzung quasi genossenschaftliche Provisionsmodell entstehen [Kuh02a, Kuh03a].

---

[42] http://www.dwsco.com/rightslocker.html

Diese Anrechnungsmodelle könnten parallel genutzt werden, um Mehrwertmodelle einzuführen. Innerhalb eines Mehrwertmodells werden Basisinformationen kostenfrei zur Verfügung gestellt und Anreize geschaffen, entsprechende Mehrwerte hinzuzukaufen. Somit würde ein Markt zwischen Mehrwertschaffenden und Mehrwerterwerbenden Gemeinschaftsmitgliedern entstehen, der sich entsprechend der üblichen Marktgesetze selbst regulieren könnte.

Ausführlichere Informationen zu möglichen Anrechungs- & Mehrwertmodellen finden sich in den Einzelarbeiten der weiteren Projektteilnehmer.

4.3 Light Weight Digital Rights Management

Neben der Idee des User Rights Management etablieren sich in jüngster Zeit auch die Konzepte des Light Weight Digital Rights Management[43] (LWDRM), das versucht „DRM den Stachel zu ziehen und die Gängelung der Nutzer aufzuheben“ [KA04].

So sieht das LWDRM eine Zweiteilung des Schutzes vor, bei der dem Nutzer, im vorgegebenen Rahmen, die Möglichkeit zur Entscheidung des Kaufverhaltens zurückgegeben wird. Wenngleich angemerkt werden muss, dass LWDRM maximal einen Bruchteil der in URM geforderten Vorgaben darstellt und sich bei genauerer Analyse als Trojanisches Pferd entpuppen könnte, muss es dennoch im Rahmen von URM erwähnt werden, tritt es doch an, den digitalen Handel offener zu gestalten.

LWDRM möchte sowohl den Tausch von Objekten zwischen Freunden, als auch die Verwendung eines Objekts auf beliebigen Geräten des gleichen Besitzers ermöglichen. Für sich alleine gesehen ist dieses Ansinnen im Rahmen von URM als Schritt in die richtige Richtung zu deklarieren. Dabei sieht das Konzept jedoch die bereits angesprochene Zweiteilung des Schutzes vor, zwischen denen der Benutzer auswählen kann:

[43] http://www.lwdrm.com/

① Nutzer nicht registriert → Wiedergabe der Inhalte nur auf dem lizenzierten Gerät möglich (Local Media Format).

② Nutzer registriert → Inhalt wird signiert (X.509 Zertifikat) und ist somit einem Nutzer eindeutig zuzuordnen (Signed Media Format).

Bei Alternative ① ändert sich für den Nutzer im Vergleich zu Standard DRM Systemen nichts. Das erworbene Objekt ist an ein Gerät gebunden und kann nur auf diesem wiedergegeben werden.

Alternative ② ermöglicht die Nutzung der Inhalte gerätungebunden und ermöglicht somit beispielsweise den Tausch mit Freunden, was der vielfach geforderten Privatkopie gleich kommt. Der zur Entschlüsselung notwendige Code wird mit der Signierung in die SMF Datei eingebettet, wodurch auch offline Geräte eine SMF Datei problemlos wiedergeben können. Durch die Einbettung eines X.509 Zertifikat jedoch kann der Inhalt eindeutig einem Nutzer zugeordnet werden, was dem unter URM geforderten Schutz der Privatsphäre widerspricht. Der Benutzer kann das Zertifikat zwar entfernen, die Löschung ist jedoch gleichbedeutend mit der Vernichtung des Objekts, da ohne jenes Zertifikat nicht mehr auf dessen Inhalt zugegriffen werden kann.

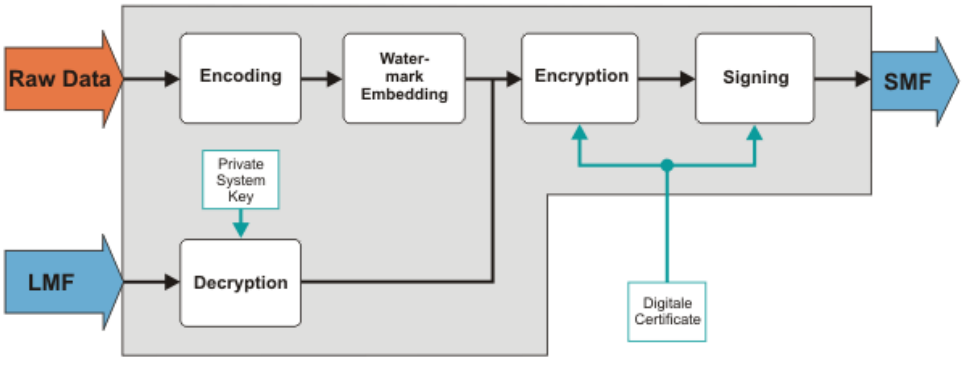


Abbildung 9 – **Überführung einer LMF in eine SMF Datei**

Bleibt festzuhalten, dass LWDRM für die URM Bewegung sehr wohl einen Schritt nach vorne bedeutet, wird bei dieser Implementation doch zum ersten Mal auf eine vollständige Kontrolle des Inhalts nach dem Erwerb verzichtet. Problematisch bleibt die zwingende Aufgabe der Anonymität, um diese neue Möglichkeit zu nutzen.

4.4 Diskussion um Geistiges Eigentum & Urheberrecht

Im Kontext des URM soll im Rahmen dieser Arbeit ebenfalls auf die Problematik der privatwirtschaftlichen Verwertung Geistigen Eigentums hingewiesen werden. So stellt Kuhlen fest, dass sich DRM keinesfalls zu einem Instrument der vollständigen Kommodifizierung, mit einhergehender Kontrolle von Wissen selbst entwickeln darf. Näher betrachtet scheint eine Kommodifizierung von Wissen im Rahmen des Konstrukts „Geistiges Eigentum“ und dessen Verwertung unabdingbar, zielt letztlich die Verwertung doch genau auf eine solche ab. Insofern könnte argumentiert werden, dass nicht DRM alleine Problem innerhalb der Informationsgesellschaft werden kann, sondern auch das zugrunde liegende Verwertungsmodell des schöpferischen Akts. So stellt sich G. Saint–Paul in [Sap02] die Frage „Are intellectual property rights unfair?“ und leitet seinen Aufsatz mit der Feststellung ein, dass sich unsere Gesellschaft in eine Situation bewegt in der Wissen ein wichtigeres Gut als physisches Kapital wird und jenes Wissen über das Konstrukt des Geistiges Eigentums zur handelbaren Ware wird.

In [SVY99] führen S. Shavell und T. v. Ypersele an, dass das am Ende des 18. Jahrunderts in Europa verbreitet Patensystem im darauf folgenden Jahrhundert, im Zeitraum von 1850-1875, aus den bereits mehrfach genannten Gründen attackiert wurde: Zum einen stärkte es Monopolstellungen, zum anderen erschwerte es die Entwicklung neuer, auf patentierten System aufbauenden, Ideen. Als mögliche Lösungen wurde bereits damals ein Entlohnungssystem (reward system) vorgeschlagen, das vorsieht, dass der Staat Entwicklungen aufkauft und diese direkt in das Allgemeineigentum (public domain) übergehen.

Nach ihren Ausführungen kommen die Autoren zu dem Schluss, dass ein System, entweder nur auf staatlicher Entlohnung (rewards) basierend, oder gemischt System aus rewards und intellectual property rights, durchaus große Möglichkeiten als Alternative zu dem Konstrukt des Geistigen Eigentums aufweist und empfehlen die eingehende Untersuchung der Auswirkungen solcher Systeme auf die gesamtwirtschaftliche Situation.

Abschließend sei angemerkt, dass [Gro00] den Wert von Ideen, sowie die Sicherheit des Geistigen Eigentums, in Abhängigkeit der Entscheidung potentiell kreativer Menschen sich kreativ oder als „Pirat" zu betätigen und in Abhängigkeit der zum Schutz des Geistigen Eigentums notwendige Zeitaufwandes der sich kreativ Betätigenden, analysiert. Dabei stellte sich als Ergebnis heraus, dass

① wenn der Anteil potentiell kreativer Menschen die als genial gelten können ausreichend groß ist und jene Genies in Bezug auf „normal" kreative Menschen ausreichend talentiert sind, sich die Genies dafür entscheiden, kreativ tätig zu sein und die „normalen" Menschen sich für ein „Piratendasein" entscheiden.

② desto besser die Umgebung für „Piraterie" geeignet ist, desto geringer ist der Wert einer Idee und daraus folgende, desto unsicher ist der Schutz des Geistigen Eigentums

③ das Vorhandensein von Genies dazu führt, dass der Schutz des Geistigen Eigentums unsicherer wird, aber der Wert einer Idee dennoch höher ist, wenn ein Genie im Vergleich zum „normalen" Menschen ausreichend talentiert ist und damit der durchschnittliche Talentlevel gleichgehalten wird.

④ Wenn der Anteil genialer, kreativer Menschen ausreichend gering ist, so verbrauchen die Erfinder zuviel Zeit zum Schutz ihres Geistigen Eigentums.

4.4.1 Diskussion um Urheberrecht

Abgesehen von möglichen anderen Modellen wie z.B. dem angeführten Rewards Modell, findet z. Zt. ein Ringen um die Novellierung bzw. um die Novellierung der Novellierung unter dem Vorsatz eines „fairen Urheberrechts" statt.

So stellt beispielsweise die Junge Union Hessen, ganz im Gegensatz zur Lobbyarbeit der CDU, innerhalb einer speziell dafür gestarteten Kampagne fest, dass „[…] *das neue Recht stellenweise zu ungerechtfertigten Benachteiligungen der Nutzer urheberrechtlich geschützter Werke führt und massive juristische Probleme aufwirft.*"[44]. Und stellt die populistische Forderung einse „durchsetzbaren Rechtes auf eine Privatkopie", sofern sich das Original im eigenen Besitzt befindet, kontakariert jedoch zeitgleich mit ihren weiteren Forderungen[45] – beispielsweise damit, dass ein klares Verbot gegenüber des privaten Kopierens von Werken, die kommerziell „on demand" oder kommerziell in Leihe zur Verfügung gestellt werden, verhängt werden soll – das grundlegende Verlangen nach einem fairen Urheberrecht und disqualifiziert sich somit selbst in der politischen Auseinadersetzung. Ein Beispiel dafür, mit welcher Unschlüssigkeit teilweise, und damit letztlich zu Gunsten der Verwerter und deren Lobbyarbeit argumentiert wird. Wobei sich die vorgestellten Forderungen auf den §95a Abs. 1 UrhG beziehen, wonach das Umgehen einer „wirksamen technischen Maßnahme zum Schutz" verboten ist. Die genaue Definition einer solche Maßnahme fehlt bislang, ist doch davon auszugehen, dass eine wirksame technisches Maßnahme nicht umgangen werden kann, wäre sie doch in der Schlussfolgerung nicht wirksam, könnte sie umgangen werden. Letztlich kann die Idee des „fairen Urheberrechts" sich durchaus als richtig herausstellen, nur die in dieser Argumentation damit verbundenen Forderungen, scheinen der Bewegung kontraproduktiv entgegenzuwirken.

---

[44] http://www.faires-urheberrecht.de/

[45] http://www.faires-urheberrecht.de/forderungen.php

4.5 Alternative Lizenzierungsmodelle

Wesentlich elaborierter als die Forderung der Jungen Union Hessen stellt sich die Open-Access[46,47] Initiative dar, die sich bemüht, den freien Online-Zugang zur wissenschaftlichen Fachzeitschriftenliteratur voranzubringen und dabei erste Ergebnisse vorweisen kann. So ist mittlerweile mit PLoS Biology[48] die erste Open-Access Zeitschrift im Internet verfügbar und die zweite mit PLoS Medicine bereits in Planung. Entsprechend [*SUEDa*] reagiert die Wissenschaftsgemeinde damit auf die Mittelknappheit der Forschungsinstitute und gleichzeitig auf die hohen Bezugspreise für wissenschaftliche Zeitschriften. Betrachtet man die Argumentation einiger Open-Access Befürworter, so sind Vergleiche zum Rewards Modell nicht ganz von der Hand zu weisen: Beispielsweise wird angeführt, dass die Forschung alleine in den USA mit 57 Milliarden Dollar/jährlich durch Steuermittel unterstütz wird und daher die Öffentlichkeit freien Zugang zu den Ergebnissen erhalten sollte.

Um Veröffentlichungen im Rahmen von Open-Access durchzuführen, muss sowohl der Autor als auch eventuelle andere Rechteinhaber allen Interessierten das freie, unwiderrufliche, weltweite Zugriffsrecht gewähren und ebenfalls eine Lizenz ausstellen, die es allen Interessierten erlaubt, das Werk zu kopieren, zu verwenden, zu verteilen, zu übertragen, es öffentlich zu zeigen und davon abgeleitete Werke zu erstellen und diese in jedem beliebigen, digitalen Medium zu verbreiten. Einzige zwingende Voraussetzung dafür stellt die korrekte Nennung der Autorenschaft dar. Ergänzend ist es Interessierten zu gestatten, eine geringe Anzahl gedruckter Kopien, für den eigenen Gebrauch zu erstellen.

---

[46] http://www.soros.org/openaccess/g/index.shtml

[47] http://www.zim.mpg.de/openaccess-berlin

[48] http://www.plosbiology.org/

Neben den Zugriffsrechten und der weit reichenden Lizenz, muss der Autor das Werk und die damit verbundenen Materialen, sowie eine entsprechende Erklärung zu den Rechten, in einem angemessenen, elektronischen Standartformat in mindestens einem geeigneten Archiv zur Verfügung stellen[49]. Per Definition sind innerhalb der Open-Access Initiative nur wissenschaftliche Arbeiten möglich, zu denen ein freier Zugang möglich ist. Entsprechend könnten Beiträge, die beispielsweise in Lehrbüchern veröffentlicht werden meist nicht im Rahmen von Open-Access zugänglich gemacht werden, da Autoren der jeweiligen Beiträge zumeist erwarten, dafür vergütet zu werden.

Ein ebenfalls von dem bisher bekannten Verwertungs-/Vergütungsmodell abweichendes Lizenzmodell sieht die Creative Commons[50] Bewegung vor, bietet dem Autor bzw. Rechteinhaber jedoch die Möglichkeit die für sein Werk vergebene Lizenz, im vorgegebenen Rahmen, selbst zu spezifizieren. Im Rahmen einer Creative Commons Lizenz darf jeder Interessierte ein Werk kopieren und verteilen, muss dabei jedoch die vom Autor bzw. Rechteinhaber aufgestellten Regelungen beachten. So kann der Autor beispielsweise festlegen, dass eine kommerzielle Verwertung seiner Arbeit nicht gestattet ist[51].

Neben dem, eher als negativem Beispiel, vorgestellten Ansatz der Jungen Union Hessen, zeigen sowohl Open-Access als auch die Creative Commons Bewegung, bezogen auf den wissenschaftlichen Sektor, einen möglichen Weg aus der Urheberrechts-/Verwertungs-/DRM Problematik und sind daher im Namen von URM zu nennen, auch wenn die zu Beginn gelieferte Definition URM nur als Ergänzung zu DRM deklarierte.

---

[49] http://www.zim.mpg.de/openaccess-berlin/berlindeclaration.html

[50] http://creativecommons.org

[51] http://creativecommons.org/license/

Letztlich bezeichnet DRM das digitale Rechte Management und genau jenes ist innerhalb der vorgestellten Lizenzierungsmodelle als „frei“ anzusehen. Daher ist die gelieferte Definition im Rahmen dieser Arbeit durchaus universell gültig, kann die Idee des Open-Access theoretisch als perfekte Synthese von DRM und URM gesehen werden. Wenngleich bei Open-Access selbstverständlich kein echtes DRM zum Einsatz kommt bzw. aufgrund der Lizenzierungsform nicht notwendig wird.

Kapitel 5

Prototyp: evobo.com

Während die vorgehenden Kapitel einen Einblick in Onlinepublikationssysteme, DRM und URM im Allgemeinen geben und damit für den notwendigen Hintergrund sorgen sollten, war es ebenfalls Ziel des dieser Arbeit zugrunde liegende Projektpraktikums einen Prototypen zu schaffen, der in der Lage sein sollte, die Prinzipien des DRM und URM auf die online publizierte Version eines Handbuchs der Informationswissenschaften anzuwenden. Während die eigentliche Anzeige der Inhalte, der Katalog sowie die Mehrwertkomponenten durch weitere Projektteilnehmer implementiert wurden, beschreibt dieses Kapitel neben den grundlegenden Überlegungen, das Basissystem, die enthaltenen DRM Komponenten sowie die Bemühungen zur Schaffung eines User Rights Managements.

5.1 Grundlegende Ideen

Das System evobo soll durch die Integration von User Rights Management zum einen und einem sinnvoll eingesetzten Digital Rights Management zum anderen, dem Nutzer einen Komfort während seiner „Arbeit“ im System gewährten und gleichzeitig ein Lizenzierungsmodell bietet, das er akzeptieren kann. Weiterer prinzipieller Unterschied im Vergleich zu kommerziellen DRM System dürfte die Sicht auf den Kunden als rechtmäßigen Nutzer und nicht als potenzieller Dieb sein. Grundlage soll ein einfach differenziertes Lizenzierungsmodell sein, welches es erlaubt, sinnvoll mit den bereitgestellten Inhalten zu agieren. Das System ist durch eine Zweiteilung in Framework und darauf aufbauender Anwendung auf eine spätere kommerzielle Nutzung insoweit ausgelegt, als dass es das Organisationsmodell erlaubt, dass Inhaltsanbieter ein solches System selbständig einsetzen und durch die Verwendung eines Templatesystem an ihre Bedürfnisse anpassen können.

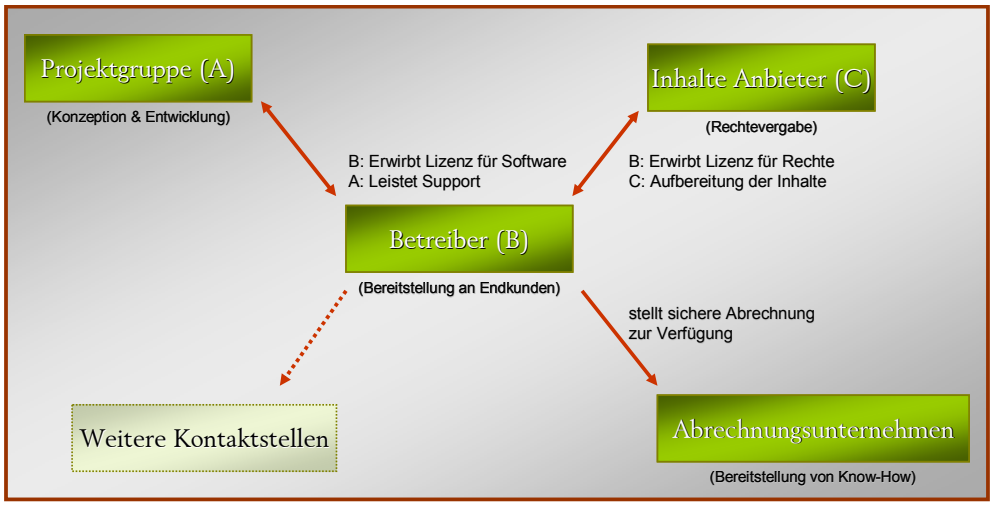


Abbildung **10** – **Organisationsmodell**

Das in Abbildung 10 schematisch gezeigt Organisationsmodell erlaubt unterschiedliche Granulierung in der Zahl der teilnehmenden Partner. Für die Kernentwicklung bleibt weiterhin die Projektgruppe zuständig. Content Provider, Betreiber sowie Abrechnungsinstitution können im Extremfall zusammenfallen, können jedoch auch von einander getrennt, auf die jeweiligen Teilbereiche spezialisiert, agieren.

5.1.1 Klientel und verfügbare Lizenzen

Die innerhalb des Prototyps bereitgestellten Lizenzierungsformen sind, entsprechend der Vorgaben, auf die wissenschaftliche Verwendung der Inhalte zugeschnitten. Das System soll dem bereits online veröffentlichenden Wissenschaftler neue Anreize beispielsweise über das Mehrwertmodell geben und den eher konventionell eingestimmten vom Potential der Onlinenutzung & -veröffentlichung überzeugen. Innerhalb des Prototyps sind fünf unterschiedliche Lizenztypen vorgesehen, wobei davon vier eigenständig sind – Vorab, Einmal, Unbegrenzt, Offline – und ergänzend eine Kombinationslizenz – Drucken – angeboten wird. Die Preisgestaltung entspricht der Nutzdauer und der Qualität der Lizenz (siehe Abbildung 11).

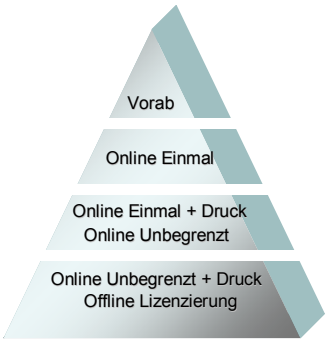


Abbildung **11** – **Preisstruktur**

- **Vorab** – Eine Vorablizenz steht nur für noch nicht veröffentlichte Werke und nur bis zu deren Veröffentlichung zur Verfügung. Diese Lizenz dient dazu bereits im frühen Stadium die Diskussion zu ermöglichen und die Idee des living books einzuführen.
- **Einmal** – Eine Einmallizenz kann beispielsweise dem kurzen Überfliegen eines Papers dienen. Die Einmallizenz entspricht nicht ganz ihrem Namen, statt eines einmaligen Zugriffs gewährt sie für die Dauer eines Tages (= 24 Stunden) Zugriff auf den jeweiligen Inhalt sowie die damit verknüpften Mehrwerte.
- **Unbegrenzt** – Die unbegrenzte Lizenz gewährt einen zeitlich und mengenmäßig unbegrenzten Zugriff auf Inhalte und Mehrwerte.
- **Offline** – Die Offlinelizenz dient dem Download in beispielsweise dem PDF Format. Eine Offlinelizenz bietet keinen Zugriff auf die Onlineinhalte und damit ebenfalls nicht, auf die nur online verfügbaren Mehrwerte.
- **Druck** – Die Drucklizenz kann zur Einmal- & zur unbegrenzten Lizenz zugekauft werden

Zu erwähnen bleibt die Besonderheit der Vorablizenz. Diese ist nur, wie bereits erwähnt, bis zur Veröffentlichung gültig. Der entrichtete Kaufpreis wird beim späteren Erwerb einer anderen Lizenz auf diesen angerechnet.

5.2 Systemgrundlage

Der Prototyp basiert auf folgender Systemumgebung:

System	Aufgabe
Apache 2.0	Webserver
PHP 5.0	Programmiersprache
mySQL 4.1	Datenbankmanagementsystem
Smarty 2.6	Template Engine

Die Entscheidung fiel durchweg auf freie Software, da diese im Rahmen des Projekts am leichtesten zu beschaffen war. Aufgrund des Einsatzes von PHP 5 war es uns möglich ein objekt-orientiertes Framework zu modellieren, was unter PHP 4 nur rudimentär möglich gewesen wäre. Der Einsatz einer Template Engine wurde aufgrund der notwendigen Anpassungsmöglichkeit erforderlich. Gleichzeitig erleichterte sie die Trennung von Benutzerinterface und Programmcode.

Basis des Systems bildet das entwickelte Framework. Auf Basis dieses Frameworks wurde mit evobo eine Beispielwebanwendung entwickelt.

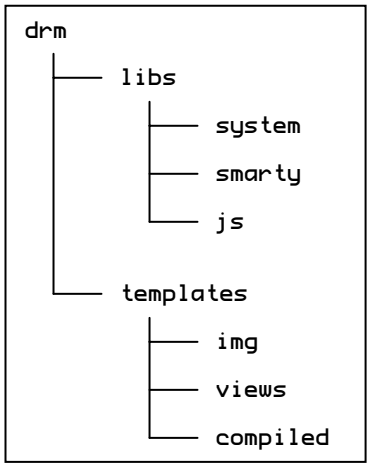


Abbildung 12 – Struktur

Anhand des Verzeichnisaufbaus ist bereits die Struktur des Systems erkennbar. Im Unterverzeichnis libs liegt neben dem evobo Framework, die smarty Klassenbibliothek sowie generelle JavaScript Dateien. Im Unterverzeichnis templates liegen smarty Templates für die Beispielwebanwendung. Die darunter liegenden Verzeichnisse dienen zur Speicherung der Bilder (img), zur Speicherung der Templates der Views des Frameworks (views) sowie zur Speicherung der kompilierten Smarty Templates.

Innerhalb des Wurzelverzeichnisses drm werden die Dateien der Beispielwebanwendung gespeichert. Die in XML Dateien abgelegten, geschützten Inhalte, werde nicht in einem dem Webserver zugänglichen Pfad gespeichert, sondern liegen außerhalb des httpd roots. Dadurch ist sichergestellt, dass nur die Beispielwebanwendung über einen lokalen Zugriff die Inhalte nach außen darstellen kann.

5.2.1 Framework

Die vom evobo Framework bereitgestellte Funktionalität lässt sich den Bereichen Datenzugriff, Anzeige, Katalog, Mehrwerte, Abrechnung, Anrechnung, Inhaltsverwaltung, Benutzerverwaltung sowie Hilfskomponenten zuteilen. Einzige nicht vom Framework bereitgestellt Funktion ist der eigentliche Lizenzierungsprozess. Während dieser zwar wiederum auf die Komponenten des Frameworks zurückgreift, so ist er innerhalb der eigentlichen Anwendung anzusiedeln. Diese Entscheidung wurde vor dem Hintergrund getroffen, eine Möglichkeit der effizienten Abbildung des Geschäftsmodells auf genau diesen Lizenzierungsprozess ohne einen Eingriff in das Framework zu ermöglichen (siehe Abbildung 13).

Objekte, die innerhalb der Prozesse als Bestandteil einer Menge gleicher Objekte auftreten können, verfügen über eine entsprechende Hilfsklasse mit der Ergänzung *Collection*. Collections übernehmen die Verwaltung sowie die Überwachung von Constrains und falls notwendig eine entsprechend sortierte Ausgabe bzw. erlauben es auch eine aggregierten Ansicht über alle enthaltenen Objekte zu implementieren. Beispielhaft für eine solche Zusammenfassung sei die Klasse *Payment* genannt. Über die im Folgenden noch beschriebene Plug-In Schnittstelle stehen innerhalb des Systems mehrere unterschiedliche Abrechnungsmethoden zur Verfügung. Diese sind jeweils einzeln innerhalb von *Payment* Objekten gekapselt. Die *Payment* Objekte sind in *PaymentsCollection* zusammengefasst. Die Anwendung erzeugt über das von *PaymentsCollection* bereitgestellte View Objekt *PaymentsCollectionView* eine aggregierte Ansicht der möglichen Zahlungsarten. Innerhalb der *PaymentsCollection* Klasse wird überprüft ob der Nutzer registriert ist und entsprechend aufbereitete Informationen an den View weitergeben.

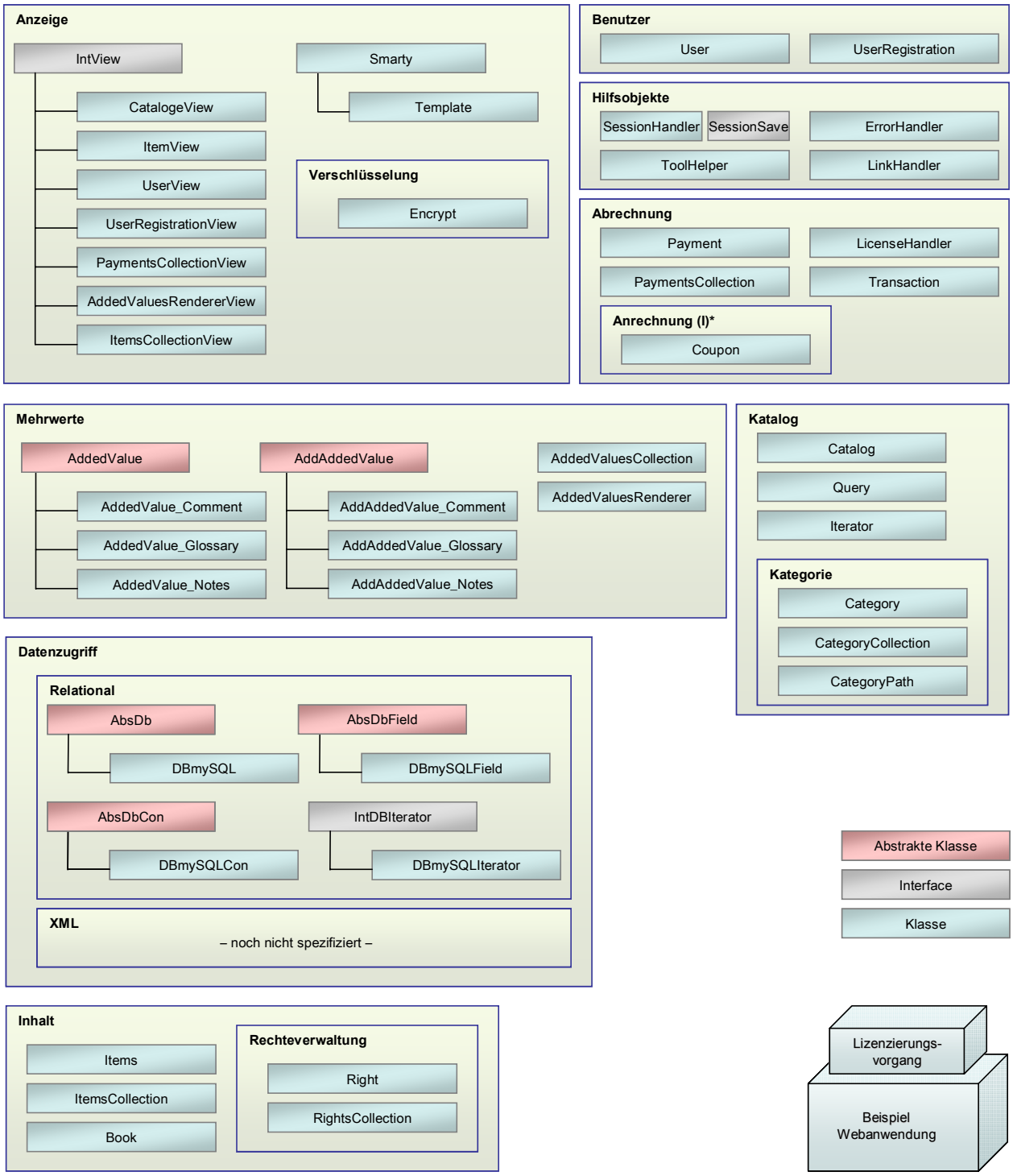


Abbildung **13** – **Überblick evobo Framework & Einordnung Lizenzierungsvorgang**

Falls eine Klasse einen oder mehrere Views zur Verfügung stellt, muss diese das Interface *IntView* implementieren, das die zwei Methoden `fetch()` und `display()` definiert. Die Implementation beider Methoden sorgt neben der Konsistenz zur verwendeten Template Engine dafür, dass innerhalb der Webanwendung die Anzeige entweder direkt über das Framework (`display`) erfolgt oder die Webanwendung die aufbereiteten Daten zur Weiterverarbeitung zwischenspeichern kann (`fetch`).

Klassen, die grundlegende Funktionalität gemein haben, werden entsprechend dem OO Modell von den jeweiligen abstrakten Klassen abgeleitet. Dieser Fall tritt innerhalb des evobo Frameworks nur für *AddedValues* und den Datenbankzugriffskomponenten auf. Durch die Abstraktion im jetzigen Stadium wird erreicht, dass das System während zukünftig möglichen Benutzertests um weitere Funktionalität erweitert werden kann, die sich dann ebenfalls innerhalb des Frameworks einbetten lässt und direkt von den bereits vorhandenen Klassen genutzt wird.

5.2.2 Plug-In System

Das evobo System bietet an zwei Stellen die Möglichkeit über Plug-Ins die Funktionalität zu erweitern, ohne bereits vorhandenen Code anzupassen.

- Innerhalb der Webanwendung kann über den `plugin` Ordner neuer Code zugeführt werden, der entweder automatisch ausgeführt wird, beispielhaft genutzt durch die implementierte Funktion `follow`[52], oder der innerhalb der Webanwendung zur Verfügung steht.
- Innerhalb des Frameworks können weitere Zahlungsmethoden als PlugIns eingeführt werden.

Während es sich bei der Ergänzung der Webanwendung letztlich um einen `require_once()` Aufruf handelt, der den notwendigen Code einbindet und darüber beispielsweise ohne weitere Eingriffe in den Code den direkten Aufruf über die URL ermöglicht, handelt es sich bei den Zahlungsmethoden um ein echtes Plug-In System.

[52] libs/system/plugins/core.function.follow.php

5.2.3 Datenhaltung

Um gegebenenfalls auf Standardaustauschformat innerhalb der Inhalte zurückgreifen zu können, erfolgt die Datenhaltung innerhalb des evobo Prototyps für Inhalte auf XML Basis. Über entsprechende XSL Transformationen gelingt so eine Überführung in die unterschiedlichsten Anzeigeformate (Online: HTML; Offline: PDF). Ein vollständiges Datenbank-Backend auf XML Basis schien ungeeignet, da die Effizienzvorteile eines relationalen System unter anderem für Retrievalmöglichkeiten genutzt werden soll. Entsprechend teilt sich die Datenhaltung wie in Abbildung 14 zu sehen auf.

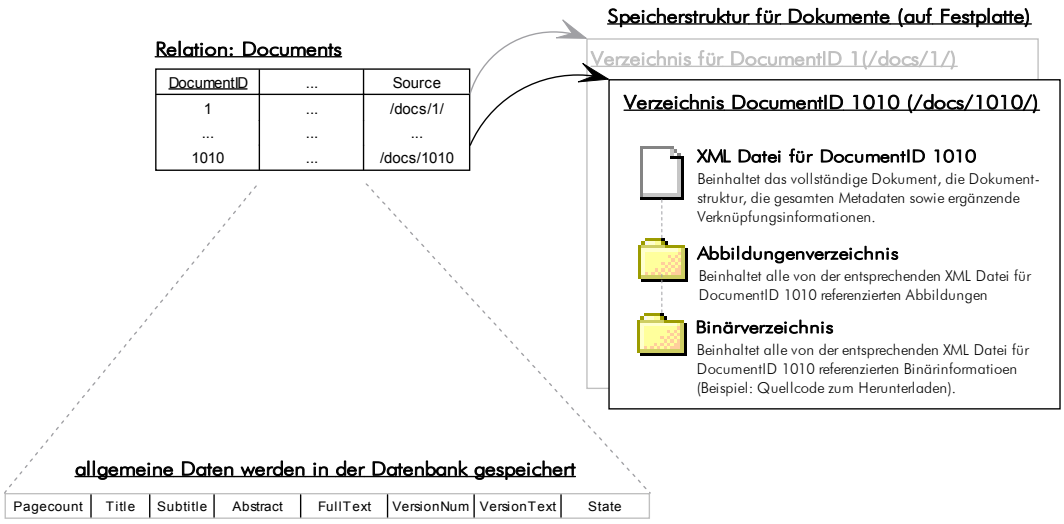


Abbildung **14** – **Speicherstruktur Dokumente**

Ein Dokument – letztlich eine Einheit, wie sie vom Autor definiert wurde – wird innerhalb seines eigenen Verzeichnisses im Sekundärspeicher abgelegt. Dort werden neben dem gesamten Inhalt innerhalb einer XML Datei ebenfalls Verzeichnisse für Abbildungen und weitere Binärdateien angelegt, auf die wiederum die XML Datei verweist. Innerhalb der XML Datei sind neben dem Inhalt, die vollständigen Metadaten gespeichert. Die Metadaten werden nach Upload des Dokuments ausgelesen und in die relationale Datenbank eingespielt. Während des Retrievals wird ausschließlich auf den in der Datenbank vorgehaltenen Datenbestand zurückgegriffen.

Sobald ein Dokument (Klasse: *Item*) im Viewer angezeigt wird, werden dessen Daten aus der XML Datei nachgeladen und die Verknüpfungen zu den Binärdaten verfolgt[53]. Bis zu diesem Zeitpunkt stehen über die Item Klasse der Zugriff auf die in der Datenbank hinterlegten Metadaten, über den Autor / die Autoren, die zugeordneten Mehrwerte, im Rahmen eines Buches (*Book* Klasse) die verwandten Dokumente, sowie die nutzbaren Lizenzen zur Verfügung[54].

5.2.4 Datenzugriff

Der Datenzugriff auf Implementationsseite kann entweder über eine von der zugrunde liegenden Datenbank abstrahierten Anfrage oder über eine direkte Anfrage in der jeweiligen Anfragesprache erfolgen. Das Framework stellt dazu zum einen die abstrakten Klasse *AbsDB*, *AbsDBField*, *AbsDBCon* & *IntDBIterator*, zum anderen mit den Klassen *DBmySQL*, *DBmySQLField*, *DBmySQLCon* & *DBmySQLIterator* eine davon abgeleitete Referenzimplementation für mySQL zur Verfügung.

Die Klasse *AbsDB* selbst stellt bereits die Implementation der Abfragelogik zur Verfügung, greift dabei jedoch auf abstrakt definierte Klassenmethoden zurück, die von entsprechend vererbten Klassen implementiert werden müssen. Um eine Anfrage an die DB zu richten, kann entweder mit der Methode `setSQLQuery()` eine SQL Anfrage übergeben werden, oder über die beiden ebenfalls abstrakt definierten Methoden `simpleQueryBuilder()` und `advancedQueryBuilder()` ein vom Datenbanksystem unabhängiges Verfahren genutzt werden. Bei der Verwendung von `setSQLQuery()` ergibt sich folgender Aufruf:

```
$db = new DBmySQL();
$db->setSQLQuery('SELECT * FROM users WHERE userid = "1"');
$db->query();
$myResult = $db->getResultAsIterator();
```

---

[53] Das Nachladen sowie die dafür notwendigen Erweiterungen der *Item* Klasse wird von anderen Projektteilnehmern implementiert und steht in der mir vorliegenden Fassung noch nicht zur Verfügung.

[54] Bereits vom Autor als Basisfunktionalität innerhalb des Systems implementiert.

Für kürzere Anfragen ist eine vorherige Zuweisung nicht notwendig, daher ist der Aufruf `query()` mit Standardparametern versehen, die es erlauben die Methode entweder ohne Parameter aufzurufen und damit eine bereits zugewiesen SQL Query auszuführen, oder mit einem Parameter aufzurufen, wodurch die SQL Query direkt beim Aufruf von `query()` zugewiesen wird. Der ebenfalls optionalen zweiten Parameter legt fest, ob es sich um eine zwischengespeichert Anfrage handelt.

Da die Anfrage an die Datenbank über den direkten SQL Syntax eine Abhängigkeit zwischen Framework und DB System erzeugt, sieht das Framework ebenfalls eine generalisierten Zugriff vor, der von der jeweiligen Implementation der DB Zugriffsklassen in einen entsprechenden Anfragesyntax übersetzt wird. Über die beiden Methoden `simpleQueryBuilder()` und `advancedQueryBuilder()` wird dieser Zugriff Zusammen mit den Klassen *AbsDBField* & *AbsDBCon* ermöglicht. Während die abstrakten Klassen tatsächlich beide Methoden definieren, wird im Rahmen der Referenzimplementation für mySQL nur `simpleQueryBuilder()` tatsächlich ausprogrammiert. Während bei `simpleQueryBuilder()` die Methodensignatur die genauen Übergabeparameter vorgibt, gilt dies für die `advanced` Fassung nicht.

```
$db = new DBmySQL();
$db->simpleQueryBuilder('update','transaction',
                        array($dbAccess->getDBField('','id','eq','1')),
                        array('amount_price','amount_credits'),
                        array($newAmountPrice,$newAmountCredits));
$db->query();
```

Da sowohl *DBField* als auch *DBCon* implementations- und somit auch von der verwendeten Datenbankzugriffsklasse abhängig sind, besitzt *AbsDB* zwei ebenfalls abstrakt definierte Methode `getDBField()` und `getDBCon()`, die implementiert werden müssen. Diese übergeben die als Parameter übergebenen Werte an den Konstruktur der jeweiligen Implementation der Klassen und geben das dadurch erzeugte Objekt zurück (siehe Beispielcode). Die Referenzimplementation für mySQL unterstützt innerhalb des `simpleQueryBuilder()` Aufrufs die `INSERT`, `UPDATE`, `SELECT` & `DELETE` Statements, jeweils mit entsprechender `WHERE` Klausel.

Bei den Implementationen der Klassen DBField und DBCon handelt es sich letztlich nur um einen Übersetzer der gewählten Standardausdrücke für gleich (eq), ungleich (neq), größer (ge), kleiner (le) in die DB abhängigen, tatsächlichen Ausdrücke. So besteht die Referenzimplementation für DBmySQLField aus folgendem Dreizeiler.

```
class DBmySQLField extends AbsDBField
{
  private $mysqlOperators = array('eq' => '=', 'neq' => '<>',
                                  'ge' => '>', 'le' => '<');

  public function translateOperator()
  {
    return $this->mysqlOperators[$this->getOperator()];
  }
}
```

Beide Klassen werden dazu verwendet, die notwendigen Strukturen intuitiv zu speichern – DBField für die Beschreibung eines Vergleichs, DBCon für die Konkatenations- & Gruppenbildungsmöglichkeiten über Klammerung.

5.2.5 Template Engine

Zur Ausgabe in HTML wird innerhalb des evobo Frameworks – über Views – und innerhalb der Webanwendung – über die Klasse *Template* – die Template Engine Smarty[55] verwendet. Die entscheidenden Gründe für ein Template System wurden bereits genannte. Die Entscheidung für Smarty erfolgte aufgrund der durch das System angebotenen Funktionalität. Im Unterschied zu anderen Template Systemen „kompiliert“ Smarty die von ihm verwalteten Templates beim ersten Aufruf in reinen PHP Code, wodurch bei jedem weiteren Aufruf die Abarbeitung beschleunigt wird. Während des Kompilierens führt Smarty PHP Code, HTML und Smarty eigene Codefragmente in eine direkt interpretierbare PHP Datei zusammen.

[55] http://smarty.php.net

Innerhalb des evobo Systems findet Smarty nur in Form der von der Klasse *Smarty* abgeleiteten Klasse *Template* verwendung. Neben einigen für evobo genutzten Grundeinstellungen implementiert die *Template* Klasse Schutzfunktionen, die innerhalb des evobo Systems genutzt werden können. Für diesen Schutz bietet die *Template* Klasse die Methode `setEncryption()`. Innerhalb dieser Methode wird über den Parameter bestimmt, welche Implementation des Schutzes bei der Inhaltsverschlüsselung eingesetzt werden soll. Über `0` kann die Verschlüsselung deaktiviert werden. Innerhalb des Prototyps wurde eine Verschlüsselungsroutine implementiert, diese kann mit `1` aktiviert werden. Intern setzt die Klasse *Template* das Flag für die Verschlüsselungsmethode, in diesem Fall `dofullencrypt` und registriert einen Outputfilter[56] für Smarty. Innerhalb des Outputfilters wird eine Instanz der Klasse *Encrypt* erzeugt, die mittels der `encrypt()` Methode die Verschlüsselung durchführt.

Aus Effizienzgründen wird ein globales *Template* Objekte pro Session instanziert, das mittels *SessionHandler* allen Frameworkkomponenten zur Verfügung gestellt wird und in den Vorlagen bzw. der Webanwendung über `currentTemplate` als Smarty Variable zur Verfügung steht.

5.2.6 Persistenzfunktionen

Für die Persistenz der Daten über Skriptgrenzen hinweg wird die Klasse *SessionHandler* genutzt. Durch die Einführung einer eigenen Sessionsverwaltung ist das evobo System unabhängig von der zugrunde liegenden Implementation. Entsprechend kann ggf. mit wenig Aufwand eine Anpassung bzw. ein Umstieg von z.B. PHP Session auf DB basiert durchgeführt werden. Über `writeSessionVar()` und `readSessionVar()` können Variablen aus der Session ausgelesen und gespeichert werden.

---

[56] An einen Outputfilter wird fertig erstellter HTML Code zur Nachbearbeitung weitergegeben. Der bearbeitet Code wird dann in die Smarty Ausgabepipeline zurückgeführt.

Bereits innerhalb des SessionHandlers sind erste Sicherheitsfunktionen implementiert. So kann über die statische Funktion `isSessionSet()` abgefragt werden, ob bereits eine Session angelegt wurde. Bei negativem Ergebnis kann eine Session angelegt werden. Deren Konstruktor ruft die private Methode `initSession()` auf. Dieser wiederum legt innerhalb der Session standardmäßig drei Variable an: `__sess_clientip`, `__sess_id` & `__sess_timestamp`. Versucht nun ein Benutzer über Session Hopping[57] mittels der Session eines eingeloggten Nutzers auf dessen Session zuzugreifen und damit seine Lizenzen zu übernehmen, so erkennt der *SessionHandler* ein solches Hopping und zerstört das entsprechende *SessionHandler* Objekt, wodurch sowohl Hopper als auch legitimer Nutzer sofort ausgesperrt werden. Der legitime Nutzer kann sich dann mit seinen Zugangsdaten wieder anmelden.

Neben der eigenständigen Verwaltung der persistenten Daten sowie der beschriebenen Schutzfunktionen bietet der *SessionHandler* ein eigenes Interface. Dieses Interface *IntSaveSession* kann von Klassen, die ein spezielles Session Handling benötigten implementiert werden. Die abstrakte Klasse *AbsDB* implementiert ein solches Interface und erlaubt damit die persistente Speicherung von Datenbankergebnissen, obwohl dies im Session Konzept von PHP nicht möglich ist. Letztlich verwendet die Klasse einen Trick um den Eindruck der Persistenz zu erwecken. Die in der Klassenvariable `$dbQueryString` gespeicherte Anfrage wird über den *SessionHandler* gespeichert. Über die Implementation der Methode `sessionAutoRestore()` des Interface *IntSaveSession* stellt die Klasse sicher, dass bei einem erneuten Aufruf außerhalb des aktuellen Skripts das Anfrageergebnis über die Ausführung der gespeicherten Anfrage erzeugt wird. Über `sessionIsRestored()` kann aus der Webanwendung heraus überprüft werden, ob diese automatische Wiederherstellung des Anfragezustands erfolgreich war. Die Implementation des Interfaces bietet sich entsprechend für Klassen an, die nicht direkt innerhalb einer Session serialisiert werden können.

---

[57] Beispielsweise mittels Ausspähen der SessionID eines anderen Nutzers im Labor und einer damit selbst erzeugten URL.

5.2.7 Retrieval

Für das Retrieval innerhalb des Systems ist aus Framework Sicht der Katalog zuständig. Dem Benutzer bieten sich drei unterschiedliche Möglichkeiten:

- Schnellsuche
- erweiterte Suche
- Katalog

Während die Suche den Zugriff auf die verfügbaren Inhalte über Matching ermöglicht, bietet der Katalog einen Zugriff über Browsing. Die Nutzung einer Schnellsuche dürfte selbsterklärend sein. Für die Implementation der erweiterten Suche ist ein adaptiver Index angedacht. Neben der Suche über die allgemeinen Metadaten werden innerhalb einer Relation ExtendedSearchIndex untypisierte Felder für ein Dokument angelegt (siehe Abbildung 15).

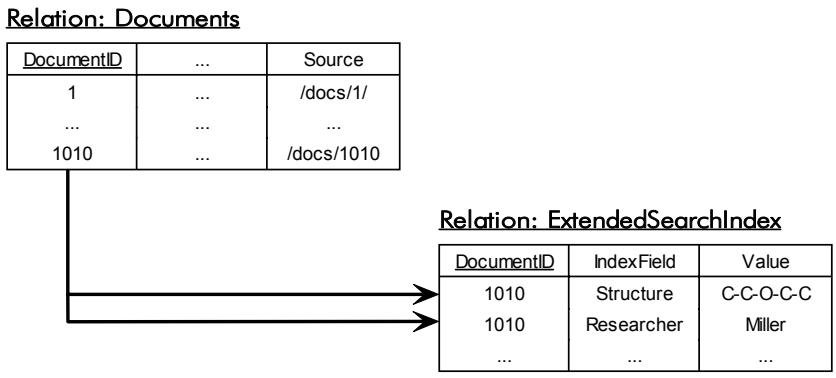


Abbildung **15** – **Schematische Darstellung ExtendedSearchIndex**

Ein Dokument ist einer oder mehreren Kategorie zugeordnet. Für jede Kategorie sind die Felder des ExtendedSearchIndex spezifiziert. So kann beispielsweise bei Publikationen im Bereich Chemie nach der Strukturformel gesucht werden. Der Benutzer erhält innerhalb der erweiterten Suche die Möglichkeit kategorieabhängig nach weiteren Indexfeldern zu suchen. Die dabei durch den Join entstehenden Performanceeinbußen beeinflussen nur jene Anfragen, die gezielt nach erweiterten Feldern suchen. Entsprechend ist die Einbuße im Allgemeinen nicht spürbar.

5.2.8 Statistische Werte

Der bislang fertig gestellte Teil des evobo Prototyps besteht aus 156 Quellcode- & 61 Vorlagendateien. Diese Zahlen verteilen sich wie folgt:

Webanwendung		
Programmcode		28
Templates	Std. Vorlagen	27
	Ansichten (Views)	30
	Zusätze (JS & CSS)	4
Framework		
Smarty	Klassen	4
	Core Funktionen	22
	PlugIns	45
	Verschlüsselung	6
System	Klassen & Interfaces	39
	Views	7
	Core Funktionen	5

In der Gesamtheit ergibt das, ohne Binärdaten wie z.B. Abbildungen, Dateien im Umfang von 1.26 MB oder auch ca. 10 000 Zeilen Code.

5.3 Prozessübersicht

5.3.1 Nutzerverwaltung & -registrierung

Die Nutzerverwaltung & -registrierung erfolgt über die Klassen *User* und *UserRegistration* sowie die hiervon verwendeten Views *UserView* und *UserRegistrationView*. Da evobo nur aktiv ist, wenn ein Nutzer die Seite aufruft, wird beim Aufruf ein globales *User* Objekt erzeugt und als nicht authentifiziert initialisiert.

Über `doLogin()` kann die Webanwendung einen Nutzer am System anmelden. Intern ruft `doLogin()` seinerseits die Methode `validateUser()` auf und versucht in der Datenbank genau einen Benutzer zu finden, auf den die übergebenen Daten passen. Nur wenn die Anzahl gefundener Datensätze genau `1` entspricht, werden die Benutzerdaten eingeladen und in `doLogin()` wiederum über einen Aufruf von `loadUserData()` in die interne Struktur überführt. Wird dabei in `iUserId` ein Wert ungleich `-1` und ungleich `0` eingeladen, gilt der Benutzer als authentifiziert.

Ein Spezialfall tritt ein, wenn der Benutzer Anonymus sich anmeldet. Er hält die `UserID 1` und wird innerhalb des Frameworks speziell registriert. Über die Methode `isAnonymous()` des *User* Objekts, kann festgestellt werden, ob das Framework den aktuellen Nutzer als Anonymus identifiziert hat. Der normale Nutzer kann sich nicht gezielt als Anonymus anmelden, dies ermöglicht nur der Weg über den Erwerb einer Lizenz. Möchte ein nicht angemeldeter Nutzer eine Lizenz nutzen, wird ihm der in Abbildung 16 gezeigt Bildschirm präsentiert.

Abbildung **16** – **Anmeldevorgang für Anonymus**

Hier bietet sich den Nutzer entweder die Möglichkeit sich über einen bereits vorhandene Account anzumelden, oder einen neuen anzulegen. Im Rahmen des URM ist es dem Benutzer auch möglich anonym im System zu agieren. Da das System diesen besonderen Status eines Benutzers identifizieren können muss, so dass beispielsweise nicht bei jeder Lizenz nachgefragt wird, und der Zugang zu geschützten Inhalten prinzipiell nur authentifizierten Benutzern möglich ist, nutzt die Webanwendung den speziellen Nutzer Anonymus und meldet mit dessen Daten den Benutzer im System an.

Weiter Funktionalität die von der *User* Klasse bereitgestellt wird umfasst beispielsweise die Identifizierung aller bereits von diesem Nutzer erworbenen Lizenzen. Innerhalb der Methode `getRightsForUser()` wird dazu eine *RightsCollection* erzeugt und diese über deren Methode `populateRightsForUser()` gefüllt.

Die Nutzerregistrierung erfolgt über die Klasse *UserRegistration*. Die Beispielwebanwendung bietet zwei Einstiegspunkte, eine solche Klasse zu instanzieren: Einmal kann der Nutzer explizit auswählen, dass er sich registrieren möchte, das andere Mal entscheidet sich der noch nicht angemeldete Nutzer beim Erwerb einer Lizenz für die Registrierung. Letztlich verwaltet *UserRegistration* während des Vorgangs die Daten, während sich *UserRegistrationView* verantwortlich für die Darstellung des aktuellen Registrationsschritts zeigt. Die Besonderheit an *UserRegistration* liegt darin, dass die Webanwendung dem Objekt über einen Aufruf von `setValidateFunctions()` ein Array übergeben kann, in dem für jeden Schritt beliebige Validierungsfunktionen übergeben werden können. Das Objekt *UserRegistration* nimmt die Daten vom Benutzer entgegen, leitet diese, falls gewünscht, an die entsprechende Validierungsfunktion innerhalb der Webanwendung weiter und bestätigt den aktuellen Schritt nur, wenn die Validierungsfunktion `true` zurückliefert. Liefert die Validierungsfunktion einen Wert ungleich `true` zurück – und zwar in Wert oder Wertetyp – so kann dieser Wert innerhalb des *UserRegistrationView* als Fehlermeldung ausgegeben werden. Im Template steht diese Fehlermeldung über die Variable `error` zur Verfügung.

5.3.2 Abrechnung

Die Abrechnung der einzelnen Werte für eine Lizenz wird nicht innerhalb des Frameworks festgelegt sondern erfolgt über registrierte Plug-Ins. Sobald ein Nutzer sich gegenüber dem System authentifiziert hat – auch der Nutzer Anonymus gilt als dem System gegenüber authentifiziert – erzeugt die Webanwendung innerhalb der Datei `getlicense.pdrm` eine *RightsCollection* und füllt diese über den Aufruf `populateRightsForUserOnItem()`. Diese Liste wird nach der gewünschten Lizenz durchsucht. Wird eine bereits erworbene Lizenz gefunden, erzeugt g`etlicense.pdrm` ein *Transaction* Objekt. Diesem *Transaction* Objekt werden die `BenutzerID` des aktuellen Nutzers sowie die `RightID` der angeforderten Lizenz übergeben. Nun wird überprüft, ob zu dem bereits gefunden Recht auch eine gültige Transaktion innerhalb der DB existiert. Ist dies der Fall, werden dem Benutzer die notwendigen temporären Lizenzen ausgestellt und er wird zu den geschützten Inhalten weitergeleitet. Hat der Nutzer die benötigte Lizenz bereits in seinem Portfolio, wird jedoch keine gültige Transaktion dazu gefunden, bricht das System mit einem `Fatal Error` ab und löscht das Recht aus der Benutzerdatenbank.

Hält der Nutzer die angeforderte Lizenz noch nicht in seinem Portfolio, so erzeugt `getlicense.pdrm` einen entsprechenden *LicenseHandler*, der wiederum eine *PaymentsCollection* erzeugt und die Methode `display()` deren Views aufruft. Hier tritt nun das Plug-In System in Kraft. Das Objekt *PaymentsCollection* sammelt in der DB die Daten aller registrierter Payment Module und bietet dem Benutzer eine Liste zur Auswahl an. Je nach Status des Nutzer – Anonymus / registriert – können die Payment Module ausgewählt werden. Die *PaymentsCollection* leitet den Nutzer letztlich wieder zurück zu `getlicense.pdrm` und liefert gleichzeitig Informationen zur ausgewählten Abrechnungsart mit. Von dort aus wird das Payment Modul geladen und die Kontrolle abgeben. Sobald das Payment Modul sich nach erfolgreicher Abrechnung mit einer positiven Antwort an `getlicense.pdrm` zurückmeldet, werden dem Benutzer die notwendigen temporären Lizenzen ausgestellt und er wird zu den geschützten Inhalten weitergeleitet.

Innerhalb dieser Beschreibung wird die zentrale Bedeutung des Moduls `getlicense.pdrm` sichtbar. Auf dieses Modul kann der Geschäftsprozess abgebildet werden. Prinzipiell kann die hier beschriebene Vorgehensweise grundlegend umgestellt oder auch übergangen werden, je nach Programmierung des Moduls.

5.4 Realisierung Digital Rights Management

Während bereits in den Abschnitten 5.2.6, 5.3.1 & 5.3.2 sicherheitsrelevante Implementationsfragmente und deren integrierter Ablauf vorgestellt wurde, folgen innerhalb dieses Abschnitts die Aufschlüsselung der eigentlichen Verschlüsselung sowie der Ablauf ab dem Zeitpunkt der erfolgreichen Abrechnung.

Vorab sei erwähnt, dass der Schutz aus mehreren kleinen Teilen besteht, die einzeln gesehen keinerlei Schutzwirkung hätten, durch die Kombination jedoch ein Schutz hergestellt wird, der ausreichen sollte, die gemeine kriminelle Energie zu blocken. Ebenfalls sei drauf hingewiesen, dass zur Ansicht der geschützten Inhalte spezielle Clients ebenso wie Java Applets zur Anzeige inakzeptabel sind und somit Algorithmen wie RSA nicht genutzt werden. Ziel war es, die geschützten Inhalte innerhalb eines normalen Browserfenster zugänglich zu machen. Grundvoraussetzung für die eingesetzte Verschlüsselung ist ein aktiviertes JavaScript im Browser. Als weitere Ausgangsvoraussetzung wird festgelegt, dass geschützte Inhalte nur in einem extra dafür geöffneten Fenster ohne Standardbedienelemente geladen werden. Um diese Grundvoraussetzungen zu erfüllen, wird die Möglichkeit zum Erwerb einer Lizenz nur bei aktiviertem JavaScript angeboten und ansonsten ein kurzer Hinweistext ausgegeben, der den Nutzer auf diesen Umstand aufmerksam macht. Mit Zeitpunkt des Mausklicks auf einen Lizenztyp wird ein neues Fenster geöffnet und das globale *Template* Objekt über `setEncryption(1)` dazu angewiesen, die Ausgabe zu verschlüsseln.

5.4.1 Temporäre Lizenzen

Nach erfolgreicher Abrechnung werden dem Nutzer über einen Aufruf von `requestLicense()`, eine Methode der Klasse LicenseHandler, so genannte temporäre Lizenzen („TEMPLIC“) ausgestellt. Diese werden dazu genutzt, abzugleichen, ob einem Nutzer während eines Lizenzierungsvorganges die dabei anzuzeigenden Inhalte bereits erfolgreich angezeigt wurden. Innerhalb der Webanwendung werden in der Datei `config.inc.php` die möglichen TEMPLICs definiert. Bei Erwerb einer Lizenz werden jeweils alle möglichen TEMPLICs erzeugt – für die Beispielanwendung sind dies `index`, `options`, `content` & `addedvalues`. Diese vier TEMPLICs sind 1:1 auf die im Viewer Fenster angezeigten Teilbereiche abbildbar (siehe Abbildung 17).

Wird der Inhalte eines Teilbereichs aufgerufen, ohne dass die entsprechende TEMPLIC ausgestellt wurde, oder nachdem diese bereits wieder gelöscht wurde, wird die Anzeige mit einem `Fatal Error` abgebrochen. Eine ausgestellte TEMPLIC wird von den einzelnen Anzeigemodulen der Webanwendung (`showitem_[module_name].pdrm`) über die Methode `needLicense()` des in der Session gespeicherten *LicenseHandlers* abgefragt.

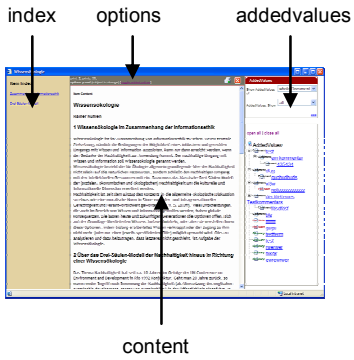


Abbildung **17** – **temp. Lizenzen**

Bei erfolgreicher Abfrage löscht der LicenseHandler die ausgestellte TEMPLIC, so dass eine Wiederverwendung ausgeschlossen ist. Auf diesen Prozess kann die Webanwendung keinen Einfluss ausüben, da er der Sicherheit dient. Die Beispielwebanwendung nutzt die TEMPLICs um sicherzustellen, dass ein anonymer Nutzer – wir erinnern uns, als Anonymus authentifiziert – keinen Zugriff auf die Mehrwerte erhält. Theoretisch könnte der Nutzer die URL der Mehrwerte über „kombinieren“ herausfinden und sie gezielt in einem Browserfenster öffnen.

Wäre beispielsweise die URL für den Teilbereich `index` `http://.../showitem_index.pdrm?itemid=1&p=1234567`, so könnte der Nutzer kombinieren, dass ein Ersetzten von `showitem_index` durch `showitem_addedvalue` zum Ziel führen könnte. Obwohl das Modul `showitem_addedvalue` tatsächlich existiert und es auch für die Anzeige der Mehrwerte zuständig ist, erhält der Nutzer dennoch keine Zugriff auf die Inhalte, da ihm die entsprechende TEMPLIC fehlt, da diese beim Erwerb einer Lizenz nur registrierten Nutzern ausgestellt wird, siehe folgendes Codefragement .

```
if (isSessionValue('newLicense')) {
  $license = getSessionValue('newLicense');
  if ($license->needLicense('addedvalues'))
    $smarty->display('showitem_addedvalues.htm');
  else error(__FILE__,__LINE__,'no license for addedvalues.',1);
 }
else error(__FILE__,__LINE__,'License error [session]',2);
```

5.4.2 Schutzsystem

Für die Darstellung der Inhalte sind die Module der Webanwendung verantwortlich. Diese interpretieren, wiederum mit Rückgriff auf das evobo Framework, die zugrunde liegenden Daten und geben die Ergebnisse innerhalb des Viewer Fenster aus. Jedes Modul kann dabei festlegen, ob die Inhalte geschützt angezeigt werden sollen.

Fordert ein Modul den Schutz an, wird – wie unter 5.2.5 beschrieben – ein Objekt der Klasse *Encrypt* instanziert, das als Outputfilter für Smarty definiert wird. Der eigentliche Schutz erfolgt über eine Kombination aus JavaScript und Zeichenverschiebung. Für die Analyse sollen folgende HTML Datei genutzt werden, deren Quellcode zuerst ungeschützt und danach verschlüsselt angegeben wird.

```
<!-- ungeschützt -->
<html>
  <head>
    <title>test</title>
  </head>
  <body>
    testbody
  </body>
</html>
```

```
<SCRIPT>function NCNoErr(){return true}//onerror=NCNoErr</SCRIPT><META
HTTP-EQUIV="Expires" CONTENT="-1"><SCRIPT LANGUAGE="JavaScript"><!--
e-
val(unescape("%66%75%6E%63%74%69%6F%6E%20%61%28%73%29%7B%0D%0A%20%20%74
%3D%22%31%55%34%77%28%30%44%57%6E%2E%54%25%6F%5F%24%4F%67%5E%7C%2A%58%4
7%78%73%2F%2C%39%7B%5D%4D%29%6C%38%70%50%03%72%56%66%48%43%4C%62%33%3B%
23%3C%3D%40%42%6A%63%6D%46%59%45%75%51%21%7D%53%7F%7E%2B%4A%79%04%4B%35
%4E%5A%49%61%26%52%69%20%41%02%64%32%37%6B%3E%01%3A%74%5B%60%76%2D%65%2
7%71%36%7A%68%3F%22%3B%0D%0A%20%20%6F%3D%6E%65%77%20%53%74%72%69%6E%67%
3B%0D%0A%20%20%6C%3D%73%2E%6C%65%6E%67%74%68%3B%0D%0A%0D%0A%20%20%66%6F
%72%20%28%69%3D%30%3B%69%3C%6C%3B%69%2B%2B%29%20%7B%0D%0A%20%20%20%20%6
E%3D%74%2E%69%6E%64%65%78%4F%66%28%73%2E%63%68%61%72%41%74%28%69%29%29%
3B%0D%0A%0D%0A%20%20%20%20%69%66%28%6E%3D%3D%2D%31%29%20%7B%0D%0A%20%20
%20%20%20%20%20%6F%2B%3D%73%2E%63%68%61%72%41%74%28%69%29%3B%0D%0A%20%20%2
0%20%20%20%63%6F%6E%74%69%6E%75%65%0D%0A%20%20%20%20%7D%0D%0A%0D%0A%20%
20%20%20%69%66%28%6E%3D%3D%30%29%20%7B%0D%0A%20%20%20%20%20%20%6F%2B%3D
%53%74%72%69%6E%67%2E%66%72%6F%6D%43%68%61%72%43%6F%64%65%28%31%30%29%3
B%0D%0A%20%20%20%20%20%20%63%6F%6E%74%69%6E%75%65%0D%0A%20%20%20%20%7D%
0D%0A%0D%0A%20%20%20%20%69%66%28%6E%3D%3D%31%29%20%7B%0D%0A%20%20%20%20
%20%20%6F%2B%3D%53%74%72%69%6E%67%2E%66%72%6F%6D%43%68%61%72%43%6F%64%65
%28%31%33%29%3B%0D%0A%20%20%20%20%20%20%63%6F%6E%74%69%6E%75%65%0D%0A%
20%20%20%20%7D%0D%0A%0D%0A%20%20%20%20%69%66%28%6E%3E%31%29%20%7B%0D%0A
%20%20%20%20%20%20%6F%2B%3D%53%74%72%69%6E%67%2E%66%72%6F%6D%43%68%61%72
%43%6F%64%65%28%6E%2B%33%30%29%0D%0A%20%20%20%20%7D%0D%0A%7D%0D%0A%0D%
0A%64%6F%63%75%6D%65%6E%74%2E%77%72%69%74%65%28%6F%29%0D%0A%7D"));a(")R
td□8U144)RIKZ8)=HY□4bYYjOHcE3ul(j□K&dK(4VB@YH@Yl(27ONKNRI(8U1)=HY□4bYYj
OHcE3ul(idK&It77□5K□(4VB@YH@Yl(27(8U1)=HY□4@□=Hl(=FFdK□tYK&:j□I`I2tjK□:
i2&(4VB@YH@Yl(t□[I(8U1)FVm3jY4<□@LE□LHl(;K`KFN□ikt(8U1)wOO4U144`K□4K&tl
2K`i&Kt7□g[:I□□&I2tgt7<7vI□VK:IT%MU144`K□4i:y2K`4414TTK&tgi2ZI-
BaT.d7'i□□K.%wlO*%4nn4TTK&tgi2ZI-

          [... code der Übersichtlichkeit wegen entfernt ...]

^^OO8U1)^FVm3jY814444)tit□I8tI:t)^tit□I8U144)^RIKZ8U144)57Ze8U14444tI:t
57ZeU144)^57Ze8U1)^Rtd□8");eval(unescape("%64%6F%63%75%6D%65%6E%74%2E%7
7%72%69%74%65%3D%6E%75%6C%6C%3B"));//--></SCRIPT>
```

Aufgrund der wesentlichen Zunahme des Quellcodes kann bereits jetzt geschlossen werden, dass der Originalcode nicht nur verschlüsselt, sondern auch erweitert wurde. So ist beispielsweise zu Beginn des verschlüsselten Codes ein JavaScript Aufruf zu sehen. An dieser Stelle sei erwähnt, dass der Nutzer sich diesen verschlüsselten Code zu keinem Zeitpunkt anzeigen lassen kann. Warum er daran gehindert wird, zeigt die weitere Analyse des verschlüsselten Codes.

Augenscheinlich klar ist, dass innerhalb des verschlüsselten Codes der entsprechende Algorithmus zur Entschlüsselung enthalten sein muss, da keine speziellen Clients eingesetzt werden.

Über `eval()` wird innerhalb von JavaScript Skript Code ausgeführt, der als Zeichenkette vorliegt. Der verwendete `unescape()` Aufruf dürfte selbsterklärend sein. Mit einem selbst geschriebenen Programm, innerhalb der Browser Internet Explorer und Mozilla [Firebird] ist dies nicht möglich, entschlüsseln wir diesen Teil und erhalten folgende JavaScript Funktion.

```
function a(s){
  t="1U4w(0DWn.T%o_$Og^|*XGxs/,9{]M)18pP□rVfHCLb3;#<=@BjcmFYEuQ!}S□~+Jy
     □K5NZIa&Ri A□d27k>□:t[`v-e'q6zh?";
  o=new String; l=s.length;

  for (i=0;i<l;i++) {
    n=t.indexOf(s.charAt(i));
    if(n==-1) { o+=s.charAt(i); continue; }
    if(n==0)  { o+=String.fromCharCode(10); continue; }
    if(n==1)  { o+=String.fromCharCode(13); continue; }
    if(n>1)   { o+=String.fromCharCode(n+30); }
  }
  document.write(o)
}
```

Diese ist für die Dechiffrierung des restlichen noch verschlüsselten Teils zuständig und hält ebenfalls den dazu notwendigen Schlüssel statisch bereit. Der Schlüssel wird bei jedem Einsatz von *Encrypt* zuvor permutiert. Nun ist auch ersichtlich, dass `a("...")` ebenfalls wiederum nur den Aufruf einer JavaScript Funktion darstellt, und zwar der Dechiffrierfunktion. Ebenfalls wieder mit einem selbst geschriebenen Programm, dass die Funktion a() nachahmt, kann jetzt der noch verschlüsselte Teil entschlüsselt werden, wodurch wiederum weitere JavaScript Funktionen sichtbar werden. Diese werden im Folgenden einzeln erläutert.

```
<script language="JavaScript">
<!--
  var agt=navigator.userAgent.toLowerCase();
  var is_nav  = ((agt.indexOf('mozilla')!=-1) &&
((agt.indexOf('spoofer')==-1)
                && (agt.indexOf('compatible') == -1) &&
(agt.indexOf('hotjava')==-1)))
                || (agt.indexOf('gecko') != -1);

  function nstie() { return false; }
  function nrcie() { return false; }
  function nrcns(e) { if(e.which==2||e.which==3) return false; }
  function nskey(val) { return !(val.which > 110 && val.which < 120); }
```

Zu Beginn des Skripts wird der UserAgent abgefragt, so dass entsprechend reagiert werden kann. Nachfolgend werden mehrere Funktionen definiert, von denen `nstie()` und `nrcie()` letztlich nur `false` zurückgeben. Diese beiden Funktionen werden später den Handlern für Drag & Drop bzw. dem Kontextmenüaufruf innerhalb des Internet Explorers zugewiesen, wodurch sichergestellt ist, dass beides deaktiviert wird. `nrcns()` sorgt für Netscape/Mozilla dafür, dass die zweite – im Normalfall rechte – sowie die mittlere Maustaste deaktiviert wird. `nskey()` letztlich trägt Sorge, dass die Tastenkombination Strg+P deaktiviert wird und somit ein manueller Aufruf des „Drucken“ Dialogs unterbunden wird.

```
function nstns()
{
  if(document.layers || is_nav) {
    st=document.getSelection();
    if(st!=""&&st!=" ") { window.find(" ") }
    setTimeout("nstns()",20)
  }
}

function nsb() { window.status=""; setTimeout("nsb()",5) }
```

Da, unterschiedlich zum Internet Explorer, die Browser der Netscap/Mozilla Reihe keine Events für Drag & Drop bzw. die Selektion bieten, sorgt die Funktion `nstns()` dafür, dass im vorgegebenen Zeitintervall von 20 Millisekunden über die Suchfunktion eine eventuelle Selektion wieder gelöscht wird. Sollte der Inhalt der Selektion nicht leer sein, wird die interne Suchfunktion des Browser aufgerufen, die dann nach dem nächsten Leerzeichen sucht. Bei erfolgreicher Suche, wird das Ergebnis selektiert, was dafür sorgt, dass automatisch das nächste Leerzeichen – das zweifelsfrei gefunden werden wird – selektiert und damit die Selektion des Nutzers aufgehoben wird. Die Funktion `nsb()` ist dafür zuständig, dass die Statuszeile keine Verweise anzeigt.

```
function np1()
{
  for(wi=0;wi<document.all.length;wi++) {
    if(document.all[wi].style.visibility!="hidden") {
      document.all[wi].style.visibility="hidden";
      document.all[wi].id="prot";
    }
  }
}

function np2()
{ for (wi=0;wi<document.all.length;wi++) {
    if(document.all[wi].id=="prot") document.all[wi].style.visibility="";
   }
}
```

Die Funktionen `np1()` und `np2()` wiederum finden Verwendung im Internet Explorer. Hier werden diese den Events `onBeforePrint` und `onAfterPrint` zugewiesen, so dass vor dem Druck alle Element des Textes durchlaufen werden und deren Sichtbarkeit auf `hidden` gesetzt wird und nach dem Druck wieder auf „“ und entsprechend auf sichtbar gesetzt werden. Dieser Zwischenschritt wird notwendig, da im Internet Explorer die Tastenkombination Strg+P für Drucken nicht abgefangen werden kann. Durch diesen Schritt wird der Druck zwar dennoch nicht grundlegend verhindert, aber die geschützten Teile werden nicht ausgedruckt, da diese während des Druckvorgangs versteckt sind.

```
if(window.location.protocol.indexOf("file")!=-1)
  window.location="error.php"
```

Dieser Codebestandteil dürfte mit einer der wichtigsten sein. Er trägt Sorge dafür, dass eine eventuell lokal gespeicherte, verschlüsselte Datei nicht lokal dechiffriert im Browser angezeigt werden kann. Das Zugriffsprotokoll auf eine lokale Datei lautet `file`. Entsprechend wird überprüft, ob das verwendete Protokoll jenes `file` Protokoll ist und wenn hierbei ein positives Ergebnis zustande kommt, wird der Browser sofort auf die dafür spezifizierte Seite umgeleitet.

Ein Abschalten von JavaScript bewahrt den Nutzer vor der Weiterleitung, bringt den Angriff jedoch nicht zum Erfolg, da ohne JavaScript die Dechiffrierung nicht gestartet wird und somit die Datei verschlüsselt bleibt. Wir können also davon ausgehen, dass wenn die Datei entschlüsselt wird, dieses Codefragment auch ausgeführt wird.

```
if(document.layers || is_nav) {
  document.captureEvents(Event.MOUSEDOWN);
  window.captureEvents(Event.KEYPRESS);
  document.onmousedown=nrcns
  window.onkeypress = nskey;
 }
else {
  document.onselectstart=nstie;
  document.ondragstart=nstie;
 }

  nstns(); nsb();

  document.oncontextmenu=nrcie;
  window.onbeforeprint=np1;
  window.onafterprint=np2;
</script>
```

Bleibt abschließend noch die Zuweisung der einzelnen, vorgestellten Funktionen an die entsprechende Eventhandler, was wiederum durch das letzte Codefragment bewerkstelligt wird.

Die tatsächliche Verschlüsselung des vorgestellten Skripts sowie des Originalinhalts erfolgt über eine Hash-Funktion. Vor Beginn der Verschlüsselung werden alle im HTML Standard erlaubten Zeichen beliebig permutiert in ein Array geschrieben. Dem Zeichen mit dem ASCII Code 13 sowie dem Zeichen mit dem ASCII Code 10 – beide zum Zeilenumbruch verwendet – werden die ersten zwei Arrayfelder zugeordnet. Alle weiteren Zeichen mit ASCII Code kleiner 31 werden unverändert übernommen. Die Zeichen mit ASCII Code größer 31 bzw. gleich 31 werden um 30 nach links geshiftet und danach im Sinne einer Hash-Funktion verwendet um das Zeichen an der Position des neuen ASCII Werts im Array zurückzugeben. Der bereits abgedruckte Dechiffrieralgorithmus sollte das Verfahren veranschaulichen.

Nach dieser Analyse des verschlüsselten Inhalts zeigt sich, dass neben dem ursprünglichen Inhalt noch zwei JavaScripts beigefügt wurden. Eines für die Dechiffrierung, das andere zur Begrenzung der Interaktions- und damit Manipulationsmöglichkeiten. Selbst wenn der Schutz nach dieser Analyse durchaus als schwach eingestuft werden kann, so ist es fraglich, ob der Aufwand sich gegenüber dem Ertrag rechnet. Wir halten fest, dass erstens der genutzte Schlüssel ständig permutiert wird und ein automatisches Herunterladen der geschützten Inhalte über das Lizenzierungssystem annähernd blockiert wird. Entsprechend ist die Wirksamkeit dieses, zugegeben recht simplen Schutzes, doch als relativ hoch einzustufen, insbesondere unter anbetracht der Tatsache, dass bei diesem Ansatz tatsächlich der beim Nutzer bereits installierte Internet Browser als Clientapplikation ausreicht. Ergänzend muss erwähnt werden, dass die Klasse *Encrypt* beliebig viele verschiedene Verschlüsselungsverfahren verwalten kann.

Während innerhalb des Frameworks für den Prototyp nur die gezeigte Variante zur Verfügung stehen, können bei kommerzieller Nutzung problemlos weitere eingefügt und unter diesen zufällig ausgewählt werden, wodurch die Sicherheit bzw. der zur Umgehung des Schutzes notwendige Aufwand noch einmal um ein Vielfaches erhöht wird, da dann vom Nutzer noch einmal manuell entschieden werden müsste, welcher Schutz gerade eingesetzt wurde und wie er diesen umgehen kann. Diese „Umgehung“ selbstverständlich immer unter der Annahme, dass der Nutzer den geschützten Inhalte bereits lokal vorliegen hat, was im Normfall nicht sein dürfte.

5.4.3 Erweiterungsmöglichkeiten

Das vorgestellt System sieht vor, dass der zur Dechiffrierung notwendige Schlüssel innerhalb des übertragenen, geschützten Inhaltes statisch kodiert vorhanden sein muss. Da dieser Prototyp auch zur Evaluation unterschiedlicher Organisations- & Geschäftsmodelle dienen soll, wurde dieses Konzept beibehalten, um feststellen zu können, wie groß der Anteil der potentiellen Nutzer tatsächlich ist, die das hier vorgestellte Verfahren trotz angepasstem Geschäftsmodell umgehen würden.

Die Schwachstelle innerhalb der vorgestellten Dechiffrierfunktion `a()` ist der lokal verfügbare Code. Dieser Umstand kann durch marginale Änderungen am Quellcode geändert werden. So könnte nachfolgende Funktion zu den bereits eingebetteten hinzugefügt werden und damit ein dynamischer Zugriff auf den zufällig generierten Schlüssel erfolgen.

```
function getKey()
{
  var url = "http://..../showitem_key.pdrm?p=123456789";

  if (document.all)       // Internet Explorer 4.0+
  {
    var xml = new ActiveXObject("Microsoft.XMLHTTP");
    xml.Open("GET",url,false); xml.Send()
    return xml.responseText;
  }
  else                    // Mozilla/Netscape 6.0+
  {
    var xml=new XMLHttpRequest();
    xml.open("GET",url,false); xml.send(null);
    return xml.responseText;
  }
}
```

Die Funktion a() könnten dann entsprechend angepaßt werden.

```
function a(s){
  t = getKey()
  o=new String; l=s.length;
  // ...
}
```

Um sicherzustellen, dass der Key nur einmal angefordert werden kann, wird `showitem_key.pdrm` in das Konzept der temporären Lizenzen eingebettet, wodurch ein manuelles „Nachladen“ des Keys vereitelt und damit einer Dechiffrierung vorgebeugt würde. Die Klassen *Template* und *Encrypt* sehen einen solchen Einsatz bereits vor. So verfügt die Klasse *Encrypt* über die Methode `getKey()`, welche die aktuell genutzte Schlüsselpermutation zurückliefert, während die Klasse *Template* diese Methode nutzt, um innerhalb einer eigenen Klassenvariable den Schlüssel zwischenzuspeichern. Die `showitem_[module_name].pdrm` Module der Webanwendung können somit dafür sorgen, dass der zugeordnete Schlüssel in der eingesetzten Datenbank hinterlegt wird und für den einmaligen Aufruf über `showitem_key.pdrm` zur Verfügung steht.

Das Ergebnis dieser Erweiterung wäre ein Abblocken des Angriffs auf der zweite Ebene der Verschlüsselung: Der Angreifer könnte zwar den Dechiffrieralgorithmus entschlüsseln, käme jedoch nicht an den für die weitere Entschlüsselung benötigten Key, da dieser, sobald er den geschützten Inhalt erhalten hat, bereits nicht mehr zur Verfügung steht. Diese erweiterte Sicherheitsstufe wird vom evobo Framework bereits zur Verfügung gestellt, muss jedoch von der Webanwendung implementiert werden. Der Schlüssel würde über eine SSL Verbindung übertragen, wodurch das „einfache" abhören über beispielsweise einen lokalen Proxy blockiert würde. Weitere Informationen zu möglichen Angriffen folgen im nächsten Abschnitt.

5.4.4 Angriffsmöglichkeiten

Die erste Angriffsmöglichkeit auf das System wäre ein Angriff zum Zeitpunkt des Ausstellens der temporären Lizenzen. Zu diesem Zeitpunkt könnte der Angreifer über Programme wie beispielsweise TamperIE[58] oder auch den HTTP Debugger Fiddler[59] bzw. einem einfachen zwischengeschalteten Proxy die zur Lizenzausstellung genutzt URL ausspähen. Hierbei würde der Angreifer auf eine ähnliche Zeichenkette wie die folgende stoßen

```
/getlicense.pdrm?itemid=1&license=u&p=d94943825aec652f
```

Das Modul `getlicense.pdrm` wurde bereits beschrieben. Die übergebenen Parameter scheinen implizit klar, `itemid` wird die Identifikationsnummer des Dokuments sein, `license` die Kennung der angeforderten Lizenz und `p` schließlich die SessionID. Und genau hierin liegt die Tücke in diesem Angriff, es wird keine eindeutige Lizenzkennung übermittelt. Im Beispiel bestimmt „`u`" zwar den Typ der Lizenz (= unbegrenzt), weitere Informationen zur Lizenz sind nicht enthalten. Diese werden ausschließlich serverseitig gespeichert und sind damit vor Veränderungen von außen geschützt.

[58] http://www.bayden.com/dl/tamperiesetup.exe

[59] http://www.bayden.com/fiddler/

Eine Lizenzkennung wird zu Beginn des Lizenzierungsvorgangs gespeichert und beim Aufruf von `getlicense.pdrm` bereits auch wieder gelöscht. Der Angreifer müsste entsprechend zwischen Ausstellung und der direkt nachfolgenden Nutzung der Lizenz unterbrechen. Das Stoppen des Datenstroms stellt mit z.B. Fiddler kein Problem dar, bringt den Angreifer letztlich jedoch nicht weiter. Selbst wenn er nun einmal manuell die temporären Lizenzen erzeugen kann, hätte er diese ebenso automatisch vom System erhalten. Denn auch wenn der Nutzer manuell die temporären Lizenzen erzeugt, wird die Lizenzkennung auf dem Server dabei gelöscht. Weitergeben kann der Angreifer diese Zugriffsmöglichkeit ebenfalls nicht, da bei einem Zugriff auf seine Session von einer unterschiedlichen IP, diese Session sofort zerstört wird.

Eine weitere Angriffsmöglichkeit wäre wiederum über einen zwischengeschalteten Proxy oder ebenfalls Fiddler möglich. Es könnte der Datenstrom solange überwacht werden, bis der verschlüsselte Inhalt übertragen wird und dieser könnte dann lokal gespeichert werden. Dieser Angriff wäre zuerst von Erfolg gekrönt. Der Angreifer würde eine ähnliche Zeichenkette wie die folgende vorfinden und könnte dann den übermittelten, verschlüsselten Inhalt lokal abspeichern.

```
/showitem_content.pdrm?itemid=1&p=d94943825aec652faff73d67a190af14
```

Während dieser Angriff zwar für den Prototyp durchaus relevant ist, bleibt er bei einem kommerziellen Einsatz des Systems wirkungslos. Hier könnte eine Verschlüsselung über SSL diesem Angriff wirkungsvoll vorbeugen. Nehmen wir aber an, dass der Angriff wirkungsvoll wäre. Ergebnis wäre die lokal vorhandene aber dennoch chiffrierte Datei. In diesem Fall treffen die Ausführungen aus Abschnitt 5.4.2 zu. Mit einem Browser ohne bzw. deaktiviertem JavaScript könnte die Datei nicht dechiffriert werden, bei einem Browser mit JavaScript wird aufgrund des erkannten lokalen Zugriffs – wir erinnern uns, das `file` Protokoll wird verwendet – eine sofortige Weiterleitung veranlasst, durch die der Angreifer wiederum nicht an die Inhalte käme. Bleibt also die ebenfalls bereits unter 5.4.2 erwähnte, manuelle Methode.

Mit den notwendigen Kenntnissen und einer Programmierumgebung ausgestattet, könnte sich der Angreifer einen kleinen, selbst geschriebenen Dechiffrierer basteln und damit die Verschlüsselung umgehen. Bleibt festzuhalten, dass er dies für jeden Abschnitt eines Dokuments einzeln durchführen muss, da diese aufgesplittet und über den Index getrennt erreichbar sind. Zusätzlich wäre ihm dann der Zugriff auf die Mehrwerte weiterhin versagt. Selbst dieser theoretische Angriff, der praktisch in dieser Form tatsächlich keinerlei Relevanz hat, scheint also den Aufwand für den Ertrag nicht zu rechtfertigen. Insbesondere bei Verwendung des dynamischen nachladbaren Schlüssels, gestaltet sich die lokale Entschlüsselung als anspruchsvoller.

Letzte Methode, um die eben erwähnte SSL Verschlüsselung zu umgehen, wäre ein lokaler man-in-the-middle Angriff mit beispielsweise Achilles[60] durchzuführen und somit tatsächlich auch im kommerziellen Einsatz an die verschlüsselten Inhalte zu gelangen. Dennoch, auch hier stellt sich die bereits erwähnte Problematik des Aufwandes und auch der Mehrwerte, welche letztlich einen Hauptanreiz für das System evobo an sich darstellen sollen. Wobei davon auszugehen ist, dass bei jenem Angreifer, der eine man-in-the-middle Attacke durchführt, tatsächlich das Know-How vorhanden ist, die Verschlüsselung mittels eines eigenen Clients automatisiert zu umgehen. Dennoch bringt eine abschnittsweise Permutation[61] der Verschlüsselungsmethode, auch diesen eventuellen Automatismus an seine Grenzen, wobei wiederum festzuhalten bleibt, dass der Angreifer auch hier auf die manuelle Methode zurückgreifen muss.

---

[60] http://packetstormsecurity.nl/web/achilles-0-27.zip

[61] Diese ist bereits im Prototyp verfügbar, sofern die Klasse *Encrypt* mit weiteren Verschlüsselungsverfahren ergänzt wird.

5.5 Realisierung User Rights Management

Die Umsetzung des URM innerhalb von evobo kann nicht in dem Maße beschrieben werden, wie es für DRM möglich war. Dies liegt einfach darin begründet, dass die Konzepte des User Rights Management in jedem betroffenen Bereich evobos implementiert wurden. Angefangen von der Registrierung eines neuen Users bis hin zur Anrechnung[62] für die erbrachte Leistung bei der Schaffung der Mehrwerte durch einen Nutzer.

Innerhalb der in dieser Arbeit vorgestellten Konzepte findet sich URM explizit zum einen bei der Registrierung eines neuen Nutzers und zum anderen beim Erwerb einer Lizenz. Für die Registrierung als Nutzer sind zwingend nur eine E-Mailadresse und ein selbst vergebenes Passwort notwendig. Jegliche andere Informationen sind optional und können durch das Aktivieren einer Auswahlbox übersprungen werden. Der Benutzer kann sich innerhalb des Systems einen eigenen Alias vergeben, der später von ihm als Pseudonym verwendet werden kann. Entsprechend bleibt der Nutzer selbst nach Registrierung dem System gegenüber in soweit unbekannt, als dass nur eine E-Mail Adresse von ihm bekannt ist, wobei er diese auch speziell für die Verwendung von evobo angelegt haben kann und beispielsweise über einen Webmail Anbieter betreiben könnte, um somit seine Anonymität im Rahmen des Möglichen aufrecht zu erhalten. Festzuhalten bleibt nicht der mögliche Weg, sondern die Tatsache, dass das System dem Nutzer diesen Weg ermöglicht.

Beim Erwerb einer Lizenz steht es dem Benutzer offen, ob er sich Anmelden und damit letztlich seine Identität preisgeben möchte, oder ob er die Inhalte anonym nutzen möchte. Wählt er die anonyme Nutzung aus, werden keinerlei persönlicher Daten abgefragt. Die Möglichkeit der Umsetzung eines solchen Systems, dem anonymen digitalen Einkauf – nicht begrenzt auf den Zahlungsvorgang – in weiteren Umgebungen führt [HT96] vor.

---

[62] Dieses Gebiet wird innerhalb der Projektgruppe von einem Kommilitonen erschlossen.

Im Zusammenhang mit entsprechenden Abrechnungssystemen, wie beispielsweise Pre-Paid Karten, kann sich der Nutzer, bis auf die temporäre Speicherung seiner IP Adresse in den automatisch angelegten Session Informationen, vollkommen anonym im System bewegen und prinzipiell die gleichen Vorteil nutzen, wie ein registrierter Kunde. In der Beispielwebanwendung trifft diese Aussage nicht vollständig zu, da hier dem anonymen Benutzer der Zugriff auf die Mehrwerte verwehrt bleibt. Dies hat allerdings wissenschaftliche Gründe, da die dadurch gewonnenen Daten zur Evaluation genutzt werden sollen, in wieweit die gebotenen Mehrwerte tatsächlich Anreiz sind.

Wie bereits erwähnt ist eine klare Quantifizierung der URM Techniken innerhalb des Prototyps nicht möglich, da URM mehr einer Philosophie gleichzusetzen ist, die in die gesamten Prozessabläufe mit einfließen muss um effektiv zu funktionieren und um die Vorteile tatsächlich zum Nutzer transportieren zu können.

So hat der Nutzer innerhalb des evobo Frameworks und auch der damit bereitgestellten Beispielwebanwendung ebenso das Recht seine Standardanwendung zu nutzen. Obwohl prinzipiell nicht als Recht wahrgenommen, ermöglicht der Verzicht auf einen proprietären Client bzw. den Einsatz von embedded Software wie z.B. JavaApplets oder auch COM-Objekte (ActiveX), dem Nutzer in seiner gewohnten Umgebung zu arbeiten. Während der „normale“ Mensch diesen Umstand wahrscheinlich „nur“ als positiv wahrnimmt, kann er für Kranke bzw. Behinderte mit dazu beitragen, das Recht auf Selbstbestimmung zu erhalten. Die Barrierefreiheit für Blinde beispielsweise kann durch den Einsatz der Standardsoftwareumgebung – d.h. in diesem Falle Standart Internet Browser – gesenkt werden, verfügt er für diese Umgebung beispielsweise schon über einen für ihn komfortablen Screen Reader.

Auch das Vorhalten der Inhalte auf dem Server und die damit verbundne, ausschließlich serverseitige Bereithaltung und Verifikation der bereits erworbenen Lizenzen, entspricht wiederum in Teilen der Idee des URMs. Inhalte sind nicht mehr an einen Rechner des Nutzers gebunden, sondern sind an den Nutzer gebunden. Seine Lizenzen werden online vorgehalten und stehen dem Nutzer von jedem Aufenthaltsort unbegrenzt zur Verfügung. Ähnliche Möglichkeiten stellt beispielsweise auch das Rights Locker Konzept des ADo²RA System der Digital World Services zur Verfügung.

Die hier beschriebenen Teile des URM Gesamtkonzept in evobo basieren auf den vom Framework bereitgestellten, und der damit diese Abschlussarbeit betreffenden, Funktionalitäten. Wesentliche Ausführungen zum Themengebiet der Anrechnung, werden sich in den Einzelarbeiten der übrigen Projektteilnehmer finden.

5.6 Zukünftige Entwicklungen

Zukünftige Entwicklungen bezogen auf den evobo Prototypen könnten auf drei Bereiche Auswirkungen haben: die Verwaltung der Inhalte, die Präsentation der Inhalte oder der Interaktion mit dem System.

Für die Verwaltung der Inhalte wäre die Einführung eines Identcodes für Bücher denkbar, mittels dem man offline gekaufte Bücher beispielsweise Konferenzzusammenfassungen für die online Nutzung freischalten könnte. So biete der Service schuelerlexikon.de[63] einen solchen Zugang zu den innerhalb des eigenen Angebots verfügbaren Werken bereits an. Denkbar wäre es, dass über diese erweiterte Funktionalität Nutzer gewonnen werden können, die das System auf anderem Wege nicht akzeptieren würden. So könnten skeptische Wissenschaftler einen Einblick in die Möglichkeiten der Onlinepublikation erhalten, ohne dabei ein Risiko einzugehen, da sie völlig anonym und ohne Kosten das System testen könnten und bei Nichtgefallen ohne Umstände auf das gedruckte Werk zurückgreifen könnten.

[63] http://www.schuelerlexikon.de/static/lexikon/titel.htm

Bei der Präsentation der Inhalte und der Interaktion mit dem System scheinen weitreichende zukünftige Entwicklungen möglich. So wäre es z.B. denkbar, dass die geschützten Inhalte alternativ auch kontrolliert über ein Wiki System zugänglich gemacht werden könnten, wodurch tatsächliche neue Interaktionsmethoden zur Verfügung gestellt würden. Problematisch wäre in diesem Zusammenhang letztlich zwar der Schutz über DRM und die Zuordnung des Geistigen Eigentums – wie bereits in Kapitel 2 ausgeführt –, dennoch wäre es innerhalb einer kontrollierten Umgebung eine Möglichkeit die Eigenschaften eines living books am Beispiel genauer zu studieren. Die kontrollierte Umgebung könnte beispielsweise über eine spezielle Lizenz hergestellt werden, die nur vom Systembetreuer an Nutzer vergeben werden kann und ohne die ein Zugriff auf die Inhalte über das Wiki nicht möglich ist.

Eine Möglichkeit wäre es auch innerhalb des Systems ein Blog[64] für die Nutzer einzurichten, wobei zuvor der Nutzen eines solchen Einsatzes erschlossen werden sollten. Ebenfalls vorstellbar wäre die Bereitstellung der Mehrwertdaten an Inhaber einer „unbegrenzten" Lizenz über einen RSS Feed. Hierdurch würde es diesen Nutzern ermöglicht, für sie interessante Feeds zu abonnieren und beispielsweise Software wie NewsGator[65] zusammen mit evobo einzusetzen, wodurch wiederum ein weiterer Mehrwert durch die aufbereitete Anlieferung der von anderen geschaffenen Mehrwerte, erzeugt würde.

Von weiteren, spezifischeren Lizenzierungsverfahren ist bei zukünftigen Entwicklungen erst einmal abzusehen, da diese den Nutzer sehr wahrscheinlich eher verwirren würden, als ihm tatsächlich Nutzen zu bringen. Weitere Mehrwertkonzepte sind jedoch empfehlenswert, um den Anreiz der Onlinenutzung noch zu verstärken.

---

[64] http://www.netlingo.com/lookup.cfm?term=blog

[65] http://www.newsgator.com/

Die möglichen Erweiterungen für ein System wie evobo sind unbegrenzt, wenn auch bei manchen Erweiterungen deren Nützlichkeit ausgiebig analysiert werden sollte. So scheint das evobo Framework in weiten Teilen dazu geeignet zu sein, für Kommunikationswege und nicht zuletzt auch für den in Kapitel 2 erwähnten „dynamic or communicative view on knowledge management“ als Implementationsplatform Verwendung zu finden.

Kapitel 6

Schlussfolgerungen

Die in den Kapiteln zwei bis vier beschriebenen Grundlagen konnten einen Einblick in den momentanen Stand der Technik geben und haben gleichzeitig deren Begrenztheit aufgezeigt – sowohl aus technischer als auch auf gesellschaftlicher Ebene. Während die Probleme der Onlinepublikationssysteme eher technischer Natur sind und daher eine zukünftige Verbesserung als wahrscheinlich anzusehen ist, gestaltet sich eine Lösung beispielsweise der DRM – URM Problematik wesentlich komplexer. Während die Inhaltsanbieter bislang eher auf rigide Rechtevergabe und konservative Geschäftsmodelle setzen, und somit letztlich nicht nur die technischen Möglichkeiten beschneiden, sondern in Grenzen auch die Rechte der Nutzer, kann URM einen Ausweg aus diesem Dilemma zeigen. Anstelle immer komplexerer Verschlüsselungs- und Abrechnungsmethoden liegt die Überlegung nahe, alternative Geschäftsmodelle zu entwickeln innerhalb derer der Anreiz einen Schutz zu umgehen minimiert wird. Wenn diese Idee auch als ein Entgegenkommen dem modernen „Piratentum“ gegenüber aufgefasst werden kann, so zeigt sich doch im nicht digitalen Raum, dass sich entsprechende Geschäftsmodelle umsetzen lassen. Letztlich ist der „Schwund“ innerhalb des nicht digitalen Raums einkalkuliert und die dort angewandten Geschäftsmodelle von der Mehrheit als mehr oder weniger gerecht akzeptiert. Denn Anonymität, die immer wieder gerne angeführt wird, für den leichten Diebstahl innerhalb des Internets, gibt es nirgends in größerem Umfang als in der realen Welt. Niemand kennt *A* in einem fremden Ort. Was also hält *A* davon ab, mit einer Sonnenbrille „bewaffnet“ im nächsten Geschäft zum „Pirat“ zu werden? Dieses Beispiel ist selbstverständlich stark vereinfacht, beachtet es die Ortsunabhängigkeit und die damit verbunden mögliche rechtliche Problem bei der Durchsetzung bzw. Verfolgung von Rechtsverletzungen nicht. Diese sind im Geschäft „um die Ecke“, im Gegensatz zum digitalen Raum, im Allgemeinen gegeben.

Dennoch soll das Beispiel Aspekte der Moral aufzeigen, die bislang als akzeptabel funktionierend empfunden wurden. Diese Situation kann und muss auf den digitalen Raum übertragen werden, bevor digitale Inhalte tatsächlich im kommerziellen Massenmarkt akzeptiert werden. Bei diesem Transfer dürfen selbstverständlich die in der Vergangenheit ausgearbeiteten Abkommen mit beispielsweise den Lehranstalten nicht in Vergessenheit geraten. Ein Verlust dieser würde zu Einschnitten innerhalb des öffentlichen Lebens führen, wurden doch bisherige Abkommen eben genau mit Hinblick auf die Verbesserung der Situation verhandelt. Wie also kann man ernsthaft auf die Idee kommen, dass jene Abkommen im digitalen Raum nicht mehr anwendbar sein sollen? Erste funktionierende Ansätze zeigen sowohl Open-Access als auch die Creative Commens Bewegung. Anzumerken bleibt auch, dass Arbeiten wie der vorliegenden bei zu rigider, ausnahmsloser Rechteverwaltung die Grundlage, d.h. der Zugang zu notwendigen Informationen innerhalb der Lehre, entzogen würde.

Der Prototyp evobo soll erste Ansätze zeigen, wie eine mögliche, anfängliche Umsetzung von URM aussehen könnte. Während es bei der Umsetzung von evobo in großen Teilen nur um die technische Seite ging, können die dabei entwickelten Ansätze eingesetzt werden, um attraktivere, dem digitalen Raum zugeschnittene Geschäfts- & Verwertungsmodelle zu gewähren. Größtes Problem innerhalb der digitalen Verwertung dürften die erweiterten Möglichkeiten sein, die insbesondere bei evobo aufgezeigt werden. So ist der Nutzer nicht mehr ausschließlich Konsument, sondern kann auch zum Inhaltsanbieter werden. Informationen sind nicht mehr an starre Veröffentlichungen gebunden, sondern können in beliebig granulierten Mengen erworben werden. Denkbar sind auch bei evobo weiterführende Technologien wie z.B. der Einsatz von RSS Feeds für Mehrwerte. So dass neu hinzugefügte Mehrwerte automatisch zu den Nutzern gepusht werden.

Das evobo Framework bietet vielfältige Möglichkeiten der Erweiterbarkeit und kann dazu dienen innerhalb der Forschung sowohl neue Verwertungs- als auch neue Publikationsmodelle zu evaluieren. Erste kommerzielle Umsetzungen der in evobo angedachten Funktionalitäten sind bereits verfügbar. So ist eine Minimalversion der als „Vorablizenz“ bezeichneten Idee bereits heute im Angebot von SpringerLink zu finden.

Anhang A

Literatur

[Abr01] D. ABRAZHEVICH: *A Survey of User Attitudes towards Electronic Payment Systems.* (2001).

[AJSW97] N. ASOKAN, P. A. JANSON, M. STEINER & M. WAIDNER: *The State Of The Art In Electronic Payment Systems.* IEEE Computer Volume 30 #9 pp. 28-35. (1997).

[And94] R. J. ANDERSON: *Why Cryptosystems Fail.* Communications of the ACM, 37(11) pp. 32-40. (1994).

[BDM03] A. BRAND, F. DALY & B. MEYERS: *Metadata Demystified.* White Paper by The Seridan Press and NISO Press, July 2003. (2003)

[BS03] A. BALDWIN & S. SHIU: *Hardware Encapsulation of Security Services. Proceedings of the* 8th European Symposium on Research in Computer Security, pp. 201-217. (2003).

[Cap01] P. CAPLAN: *A Lesson in Linking.* Library Journal NetConnect: Supplement to Library Journal and School Library Journal, vol.126, no. 17, Fall 2001, pp. 16-18. (2001)

[Col02] R. M. COLOMB: *Information Spaces, the Architecture of Cyberspace.* Springer Verlag. (2002)

[Gro00] H. I. GROSSMAN: *Inventors And Pirates: Creative Activity And Intellectual Property Rights.* NBER Working Paper No. 7898. (2000).

[Gut03] S. GUTH: *Rights Expression Languages,* Digital Rights Management, LNCS 2770, pp. 101-112. (2003)

[HT96] R. HAUSER & G. TSUDIK: *On Shopping Incognito.* Proceedings of the 2nd USENIX Workshop on Electronic Commerce, pp. 251-257. (1996).

[HW03] T. HAUSER, C. WENZ: *DRM Under Attack: Weaknesses in Existing Systems.* Digital Rights Management, LNCS 2770, pp. 206-223. (2003)

[Ian01] R. IANNELLA: *Digital Rights Management (DRM) Architectures*, D-Lib Magazine, Vol. 7. (2001)

[Jur03] N. JURRAN: *Industrie treibt Rechteverwaltung voran,* Magazin c't Ausgabe 1 vom 29. Dezember 2003, p. 23. (2003)

[KA04] S. KREMPL, U. HILGEFORT: *Digitales Rechtemanagement: Hollywood oder Freiheit?,* Magazin c't Ausgabe 4 vom 9. Februar 2004, p. 27. (2004)

[KBGSS02] R. KUHLEN, B. BEKAVAC, J. GRIESBAUM, T. SCHÜTZ, W. SEMAR: *Kollaborativ erarbeitetes Wissen ist mehr als die Summe des Wissens vieler Einzelautoren – ENFORUM, ein Instrument des Wissensmanagements in Forschung und Ausbildung im Informationsgebiet.* Zeitschrift für Bibliothekswwesen und Bibliographie (ZfBB) 2002. (2002).

[KG03] D. KUHLMANN, R. A. GEHRING: *Trusted Platform, DRM, and Beyond,* Digital Rights Management, LNCS 2770, pp. 178-205. (2003)

[KSF02] S. KELLY, C. SUNG & S. FARNHAM: *Designing for Improced Social Responsibility and Content in On-Line Communities.* Proceedings of CHI 2002, April 2002. (2002)

[Kuh02a] R. KUHLEN: *Medienprodukte im Netz – Zwischen Kommerzialisierung und freiem Zugang.* Tagung des Münchner Kreis „Digital Rights Management" 20. November 2002. (2002).

[Kuh02b] R. KUHLEN: *Napsterisierung und Venterisierung – Bausteine zu einer politischen Ökonomie des Wissens.* PROKLA – Zeitschrift für Sozialwissenschaft 32, 4. (2002).

[Kuh02c] R. KUHLEN: *Über die Möglichkeit eines informationsethischen Diskurses über geistiges Eigentum in der Informationsgesellschaft und der Chance der Umsetzung seiner Argument in politisch-rechtliche Kodifizierungen?* Im Rahmen der Konferenz der Heinrich-Böll-Stiftung. (2002).

[Kuh02d] R. KUHLEN: *Ein Schisma der Bibliotheken?* Frankfurter Allgemeine Zeitung 8. April 2002. (2002).

[Kuh02e] R. KUHLEN: *Rahmenbedingungen des Einsatzes von Digital Rights Management.* (2002)

[Kuh03a] R. KUHLEN: *Kauf oder Leasing – Ambivalenz pauschalierter und individualisierter Abrechnung der Nutzung intellektueller Produkte?* Digital Rights Management, Technological, Economic, Legal and Political Aspects in the European Union. Springer Verlag (2003).

[Kuh03b] R. KUHLEN: *Change of Paradigm in Knowledge Management – Framwork for the Collaborative Production and Exchange of Knowledge.* 69th IFLA General Conference and Council. (2003).

[Kur04] J. KURI: *Strichcode fürs Web,* Magazin c't Ausgabe 4 vom 9. Februar 2004, p. 27. (2004)

[Law01] S. LAWRENCE: *Online or Invisible?* Nature Volume 411 #6837 pp. 521ff. (2001).

[Les02] L. LESSIG: *Free Culture.* Keynote, Open Source Convention (2002).

[Odl00] A. M. ODLYZKO: *The Future of Scientific Communication.* The Global Research Village II, Amsterdam 2000, NIWI, 2000, pp. 273-278. (2000)

[Odl02] A. M. ODLYZKO: *The Rapid Evolution of Scholarly Communication.* Bits and Bucks: Economics and Usage of Digital Collections, MIT Press, 2002. (2002)

[Odl03] A. M. ODLYZKO: *The Unsolvable Privacy Problem and Its Implications for Security Technologies.* Information Security and Privacy: 8th Australasian Conference, ACISP 2003, Lecture Notes in Computer Science #2727, Springer, 2003, pp. 51-54. (2003)

[OZL03] J. A. ONIEVA, J. ZHOU, J. LOPEZ: *Practical Service Charge for P2P Content Distribution. Proceedings of the* 5th International Conference, ICICS 2003, pp. 112-123. (2003).

[Pas03] N. PASKIN: *Components of DRM Systems,* Digital Rights Management, LNCS 2770, pp. 26-61. (2003)

[PHH03] DR. N. PASKIN, L. HOUSE & J. HILL: *The DOI® Handbook.* The DOI Handbook Edition 3.3.0, November 2003, International DOI Foundation, Inc. (2003)

[PM03] A. PASHALIDIS & C. J. MITCHELL: *Single Sign-On Using Trusted Platforms. Proceedings of the* 6th International Conference, ISC 2003, pp. 54-68. (2003).

[Rum03] N. RUMP: *Digital Rights Management: Technological Aspects,* Digital Rights Management, LNCS 2770, pp. 3-15. (2003)

[Sap02] G. SAINT-PAUL: *Are Intellectual Property Rights Unfair?* European Association of Labour Economics. (2002).

[SS03] A.-R. SADEGHI, M. SCHNEIDER: *Electronic Payment Systems,* Digital Rights Management, LNCS 2770, pp. 113-137. (2003)

[SVY99] S. SHAVELL & T. VAN YPERSELE: *Rewards versus Intellectual Property Rights.* NBER Working Paper No. 6956. (1999).

[Tri04] A. TRINKWALDER: *Währungspolitik,* Magazin c't Ausgabe 3 vom 26. Januar 2004, p. 44. (2004)

[USC86] U.S. CONGRESS, OFFICE OF TECHNOLOGY ASSESSMENT: *Intellectual Property Rights In An Age Of Electronics And Information.* U.S. Government Printing Office. (1986)

[Wal02] J. WALKER: *CrossRef and SFX: Complementary Linking Services for Libraries.* New Library World, Volume 103 # 1174, pp. 83-89. (2002)

[WLL03] Y. WANG, S. LÜ & Z. LIU: *A Simple Anonymous Fingerprinting Scheme Based on Building Signature. Proceedings of the* 5th International Conference, ICICS 2003, pp. 260-268. (2003).

Anhang B

Internetquellen

[*VASCa*] VASCODA – GERMANY SCIENTIFIC INFORMATION PORTAL.
http://www.laboratorytalk.com/news/vas/vas100.html

[*VASCb*] VASCODA – FACHERÜBERGREIFENDES INTERNETPORTAL.
http://www.wiwi-treff.de/home/index.php?mainkatid=1&ukatid=1&sid=9&artikelid=1068&pagenr=1

[*VASCc*] VASCODA – DIE QUELLEN IM EINZELNEN.
http://www.wiwi-treff.de/home/index.php?mainkatid=1&ukatid=1&sid=9&artikelid=1068&pagenr=2

[*INFOa*] INFOCONNEX.
http://www.iwi-iuk.org/iuk2003/program/stemp/ppt/sld001.htm

[*WISSa*] WISSEN.DE-LEXIKON.
http://www.wissen.de

[*NISOa*] NISO COMMITTEE AX
http://library.caltech.edu/openurl

[*JEREa*] JERÉ MIAS
http://www.jere-mias.de/biwi/urheb1.html

[*COPYa*] DER SCHUTZ GEISTIGEN EIGENTUMS IN DEN USA
http://www.jura.uni-sb.de/urheberrecht/web-dok/1999028.html

[*EFFTa*] TRUSTED COMPUTING: PROMISE AND RISK
http://www.eff.org/Infra/trusted_computing/20031001_tc.php

[*MICRa*] MICROSOFT NGSCB – TECHNICAL FAQ
http://www.microsoft.com/technet/security/news/NGSCB.asp

[*MANAa*] WHY RIGHTS MANAGEMENT IS WRONG
http://www.w3.org/2000/12/drm-ws/pp/compaq.html

[*SUEDa*] FREIER ZUGANG
http://www.sueddeutsche.de/jobkarriere/berufstudium/artikel/304/21283/