UNDINE
Smartcards und PKI zur medienbruchfreien elektronischen Softwarebeschaffung

Matthias Honka

ask|net AG
Vincenz-Prießnitz-Straße 3
76131 Karlsruhe
matthias.honka@asknet.de

Zusammenfassung Software bietet sich wie kein zweites Produkt für den Handel im Internet an. Alle Prozeßschritte, von der Produktinformation, -auswahl, ggf. Genehmigungsworkflow, Bestellung, Lieferung und Bezahlung sowie nachgelagerter Support durch Updates etc. können rein digital abgewickelt werden. In realen Shops werden diese Möglichkeiten jedoch noch relativ zögerlich eingesetzt. Hintergrund dafür sind bisher wenig verbreitete Sicherungstechnologien für das kopierbare Gut Software.

Unter diesen Voraussetzungen wurde das Projekt UNDINE entworfen. Sein Ziel ist alle Prozeßstufen der Softwarebeschaffung ohne Medienbruch über das Internet abwickeln zu können. Als zentrale Lösung für die vielfältigen Anforderungen bieten sich die Nutzung von Public Key Infrastrukturen (PKI) und Smartcards an. Zentrale Ziele des Projekts sind die prototypische Integration von PKI in eine Web-Applikation und das Sammeln von Erfahrungen im Einsatz beim Nutzer.

1 Motivation

Der Handel mit Software unterscheidet sich in einem wesentlichen Punkt vom Handel mit anderen Waren: Das Produkt ist selbst digitale Information damit elektronisch austauschbares Gut. Daher ist es möglich die Softwarebeschaffung für Forschungseinrichtungen, Unternehmen oder Einzelkunden rein digital über das Internet abzuwickeln.

Um diese Ziele zu erreichen sind verschiedene Hürden zu überwinden, die sich gerade aus der leichten elektronischen Verteilbarkeit von Software und aus den spezifischen Eigenarten des Onlinehandels ergeben. Hier sind zu nennen:

1. Betrugsmöglichkeit durch illegale Kopie und Nutzung von Softwareprodukten
2. Die direkte digitale Lieferung während des Einkaufs erfordert die direkt verbundene Bezahlung oder Rechnungsstellung.
3. Betrugsmöglichkeit im Bereich der verbreiteten elektronischen Onlinebezahlmethoden ibs. bei Kreditkarten.
4. Verschiedene Lizenzbedingungen der Software-Hersteller, deren Einhaltung soweit als möglich durch die Händler gewährleistet werden muß, bspw. Studentenlizenzen, Campuslizenzen.

5. (Halb-)automatische Updatemöglichkeit von Produkten oder Produktteilen, bspw. Virusscanner.

Der aktuelle Beschaffungsarbeitsablauf [Abb. 1] von Software für Einzelkunden oder Mitarbeiter in Institutionen sieht folgendermaßen aus:

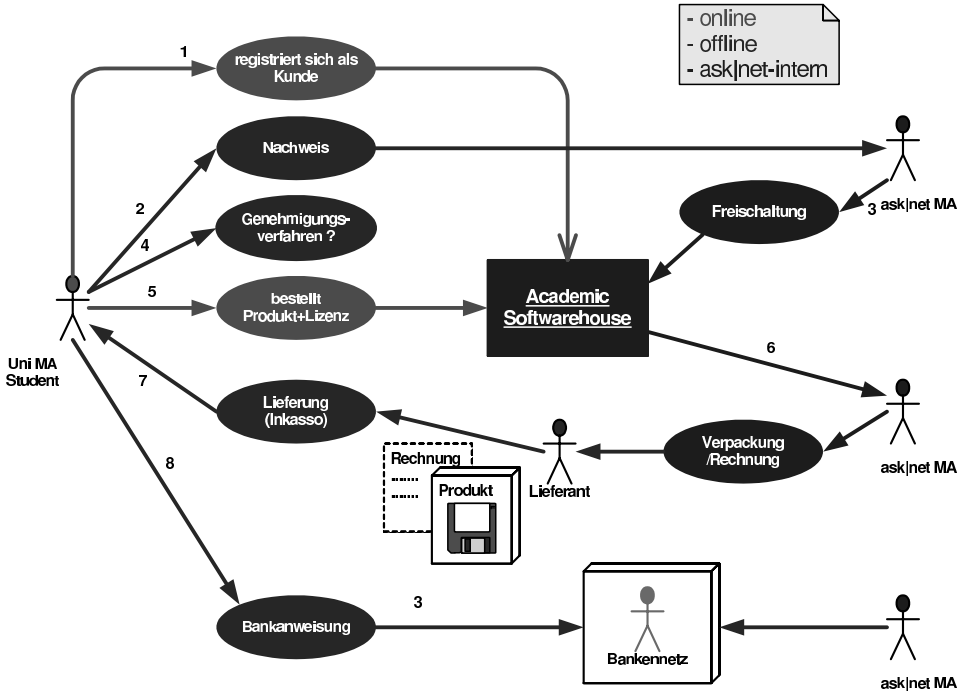

Abbildung 1: Shop-Workflow

Die sich ergebenden Medienbrüche sind zu finden bei:

1. Authentifizierung und Autorisierung: Die in einem Webformular eingegeben Daten bei der Nutzerregistrierung müssen verifiziert werden bevor die Freischaltung zum Kauf von Lizenzen aus einem Campus-Vertrag oder Studentenlizenzen möglich ist.
2. vorgelagerten Beschaffungsprozessen in Institutionen: Hier sind häufig noch Papier-Laufzettel zu finden, da Genehmigungschritte signiert werden müssen.
3. Lieferung von Produkt und Lizenz: Bei Paketlieferung ist ein Dienstleister in Anspruch zu nehmen, Adressierung und Verpackung geschehen z.T. von Hand. Der Lieferant kann ggf. auch das Inkasso übernehmen (Nachnahme).
4. Bezahlung per Rechnung: Die Rechnung liegt der Produktlieferung bei oder wird separat an eine Rechnungsadresse versandt. Die Bezahlung selbst erfolgt wieder durch Überweisung.

2 Digitale Zertifikate und Smartcards

Ohne auf den theoretischen und organisatorischen Hintergrund von Public-Key-Infrastrukturen und Hardwaretoken einzugehen, läßt sich der Nutzen von publicKey-Kryptographie und Zertifikaten kurz zusammenfassen.

Private und public Key bilden ein Schlüsselpaar mit dem komplementäre Aufgaben wahrgenommen werden können.

⇒ Zur Versendung vertraulicher Nachrichten wird der - möglichst zertifizierte - öffentliche Schlüssel des Adressaten zur Kodierung verwendet. Damit wird eine Entschlüsselung nur durch ihn möglich.

⇒ Zum Nachweis der eigenen Identität oder Vertrauenswürdigkeit, bzw. zur Signatur von Dokumenten wird der eigene private Schlüssel verwendet.

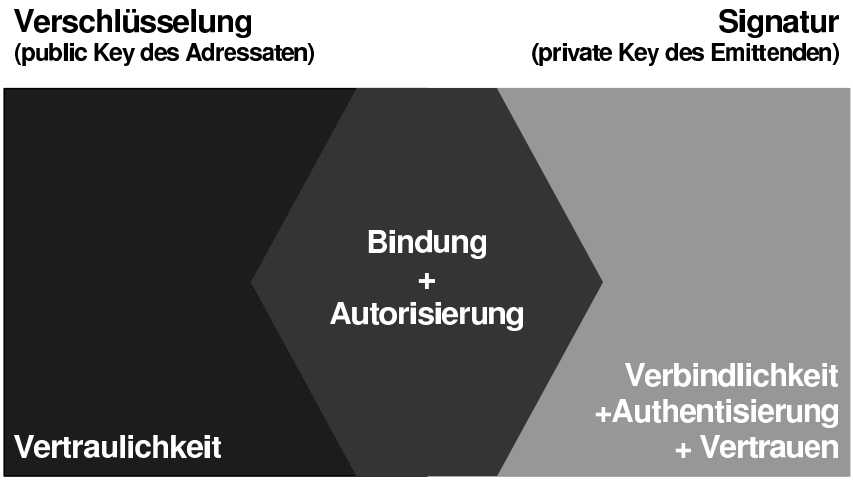

Abbildung 2: Komplementäre Nutzung von public und private Key

Beide Kodierungsverfahren zusammen ermöglichen die Autorisierung von Nutzern für exklusive digitale Dienste. Die automatische Authentifizierung läßt eine differenzierte Zugriffskontrolle zu und Verschlüsselung von Datenströmen verhindert die Ausspähung durch Dritte. Zudem können Datenlieferungen an den Adressaten gebunden werden, so daß Lieferung und Entpackung zertifikatsbasiert kontrolliert werden. Im extremsten Fall wäre sogar die Nutzung der Daten oder einer Software, wie etwa bei Digital-Rights-Managment-Systemen (DRMS), kontrollierbar.

3 PKI in der Softwarebeschaffung

Es wurden prototypisch Teillösungen implementiert und die Einsetzbarkeit und Integrierbarkeit vorhandener PKI-Lösungen in ein vorhandenes System, hier eine Web-Shop-Applikation mit gegebenem Workflow, erprobt und realisiert.

Die verschiedenen Sachverhalte bei der digitalen Softwarebeschaffung und den dortigen Einsatzmöglichkeiten von PKI lassen sich grob in folgende Segmente mit absteigender Bedeutung einteilen:

Segment	**Beschreibung und Anforderungen**
Authentifizierung Autorisierung der Kunden	Kunden müssen für verschiedene Produkte oder Dienstleistungen ausreichend und vertrauenswürdig identifiziert sein. Lieferung von bestimmten Produkten ist nur an bestimmte Nutzerkreise möglich (Studentenlizenz, Updaterechte). ⇒ Nachweis der Identität und Zugehörigkeit (Immatrikulationbescheinigung)
Lieferung von Produkten, Lizenzen	Sofortige Lieferung von digitalen Produkten und Lizenzen nach der Bestellung oder Bezahlung über das Internet. ⇒ Sicherung des Lieferkanals, ggf. Bindung der Entpackung und Installation an den Käufer oder Hardware des Käufers durch Verschlüsselung von Produkten oder Archiven.
Abrechnung Bezahlung	Online-Bezahl-Technologien sollen die Bezahlung vor der Lieferung gewährleisten oder digitale Abrechnungsmethoden sollen anwendbar sein. ⇒ Vertrauenswürdige Sofortbezahlung (digitales Bargeld?) ⇒ Versendung von Rechnungen
Zeichnung von Vertragsbedingungen	Kunden müssen ggf. vor dem Kauf über best. Rahmenbedingungen für die Produktnutzung informiert werden und sie akzeptieren. ⇒ Signierung von Vertrags/Lizenzbedingungen.
Beschaffungswesen	Bisherige Offline-Beschaffungsprozesse im Vorfeld des Kaufs per Laufzettel sollen als Dienstleistung online ermöglicht werden ⇒ Digitale Signierung von Genehmigungen
Andere Nutzungsmodelle für Software	Kontrolle der Software über die Lieferung bis hin zur Nutzung ⇒ Pay-per-use, Try-before-buy mit Sandboxtechnologie (geringe Akzeptanz, Durchdringung) ⇒ Application-Service-Providing

4 Erprobte und umgesetzte Lösungen

Um alle Varianten des Zertfikatseinsatzes austesten zu können hat sich die ask|net beim Bau der Prototypen auf vier Sektoren [Abb. 3] konzentriert:

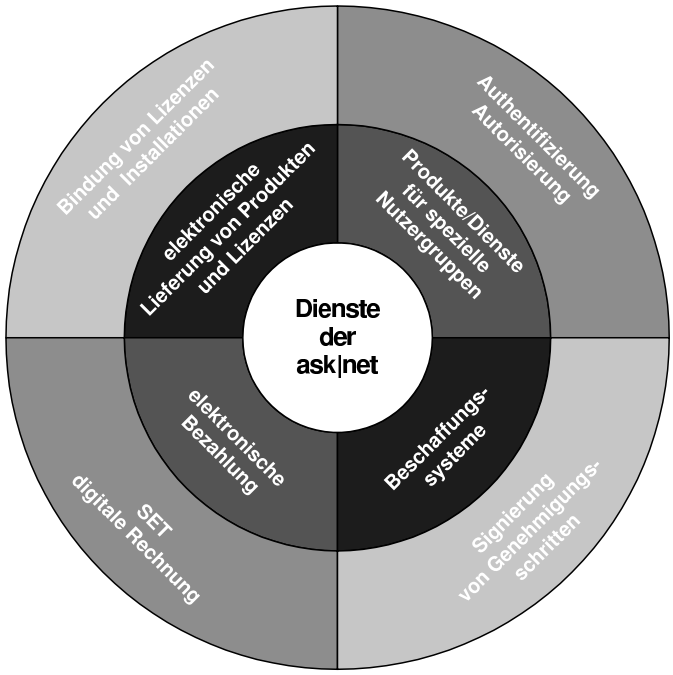

Abbildung 3: Dienste die zertifikatsbasiert implementiert wurden

Somit sind alle Fälle von Diensten, die Verschlüsselung, Zertifikate der Kunden oder vom Dienstleister ask|net erfordern abgebildet.

4.1 Kunden-Authentifizierung

Die Authentifizierung von Kunden bei der Registrierung oder beim Anmelden erfolgt direkt unter Ausnutzung der Möglichkeiten für Clientauthentifizierung von Browsern. Für spezielle Webseiten wird im Webserver eine Clientauthentifizierung vorgeschrieben.

Der Nutzer wird beim Anwählen der Seite auf die notwendige Authentifizierung aufmerksam gemacht und kann frei zwischen software- oder hardwarebasierten digitalen Zertifikaten wählen.

Voraussetzung für Hardwarezertifikate sind die Installation von Smartcardreader mit Treibersoftware und entsprechendem Browserplugin. Bei Netscape-Browsern sind dies pkcs11-Module, bei Microsoft muß der Smartcardhersteller für seine Karten sog. Crypto-Service-Provider für die MS-Crypto-API des Betriebsystems bereitstellen.

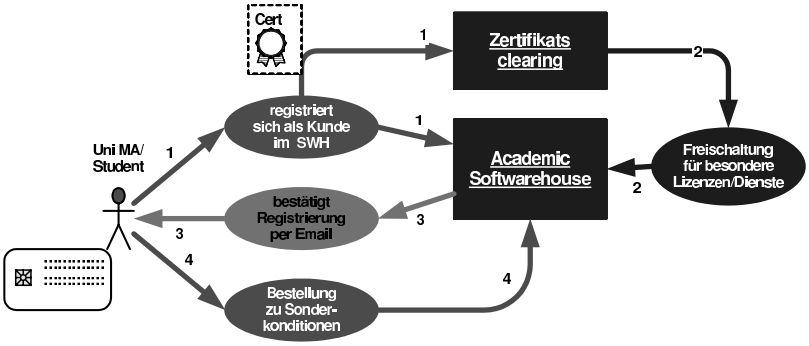

Abbildung 4: Ablauf der automatischen Autorisierung für Dienste

Abbildung 5: Erfolgtes SSL-Login mit digitalem Zertifikat

Ist aus Zertifikatsaussteller oder Zertfikatsattributen die Zugehörigkeit des Nutzers zu bestimmten Gruppen feststellbar, so kann er automatisch für Dienste oder für den Kauf besonderer Produktlizenzen - wie die o.g. Studentenlizenzen freigeschaltet [Abb. 4] werden.

4.2 Digitale Bezahlung und Abrechnung

Digitale Bezahltechnologien werden von ask|net bereits seit langem angeboten. Zur Zeit stehen folgende Methoden zur Bezahlung in den ask|net-Shops zur Verfügung:

1. per Rechnung für vorab autorisierte Kunden
2. Zahlung per Kreditkarte (Visa/Mastercard)
3. SET (mit Kundenzertifikat)
4. Bezahlung per Handy (Paybox)

Die einzige Technologie die auf digitale Zertifikate zurückgreifen könnte ist SET. Die derzeitige Nutzung ist aber sehr gering. Außerdem sind SET-Zertifikate bisher nur für diesen einzigen Anwendungszweck für Endanwender einsetzbar. Die anderen angebotenen Methoden, sind bei den Nutzern ibs. den Einzelkunden durch ihre weitaus einfachere Handhabung bevorzugt.

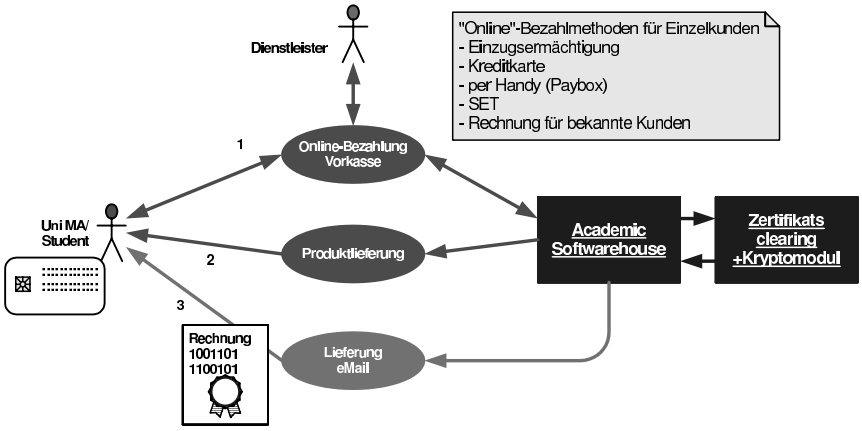

Abbildung 6: Ablauf digitaler Bezahlung

Für Großkunden aus dem Bereich Forschung und Lehre kommt aus organisatorischen Gründen nur die Bezahlung per Rechnung in Frage. Hier ist eine Lösung, durch automatischen Versand von digital signierten Rechnungsemails [Abb. 6], denkbar.

Das Signaturgesetz in der aktuellen Fassung fordert hier allerdings die digitale Signatur für Rechnungen nach den höchsten Sicherheitsstandards, d.h. eine Signatur durch eine natürliche Person. Dies steht dem ursprünglichen Automatisierungsgedanken entgegen. Ggf. kann durch Zusatzvereinbarungen mit Großkunden auch auf Signaturen juristischer Personen umgestellt werden.

4.3 Digitale Produkt Lieferungen

Die sichere digitale Lieferung ist ein sensitiver Aspekt, der bei den Software-Herstellern eine große Rolle für die Vergabe von digitalen Distributionsrechten spielt. Von manchen wird sogar eine technische Lösung für der Kontrolle der Softwarenutzung erwartet. Solche Lösungen wurden im Projekt ebenfalls diskutiert und verworfen:

1. Sandboxverfahren: Vor der Installation der eigentlichen Software muß ein Sandboxsystem auf dem Kundenrechner installiert sein. Dies erfordert tiefe Eingriffe in das Betriebsystem des Kunden, die größtenteils abgelehnt werden.
2. Application-Service-Providing (ASP): Die Software wird als entfernter Service dem Kunden über Internet bereitgestellt und nur die Bedienoberfläche auf Mietvertragsbasis zum Nutzer übertragen. Dies scheitert z.T. an den erforderlichen Bandbreiten

und an sicherheitstechnischen Überlegungen, da alle Kundendaten auf den Servern des ASP-Anbieters liegen.

Durchsetzbar und umgesetzt sind Technologien die soweit wie möglich die Kopie von Installationssoftware verhindert oder nutzlos macht. Bei ask|net ist eine zweischichtige Schutzhülle für Installationspakete entwickelt worden [Abb. 7].

1. Produkte werden per HTTP im SSL-Kanal übertragen.
2. Produkte können zusätzlich als verschlüsseltes selbstentpackendes Archiv (Box of Bits) übertragen werden.

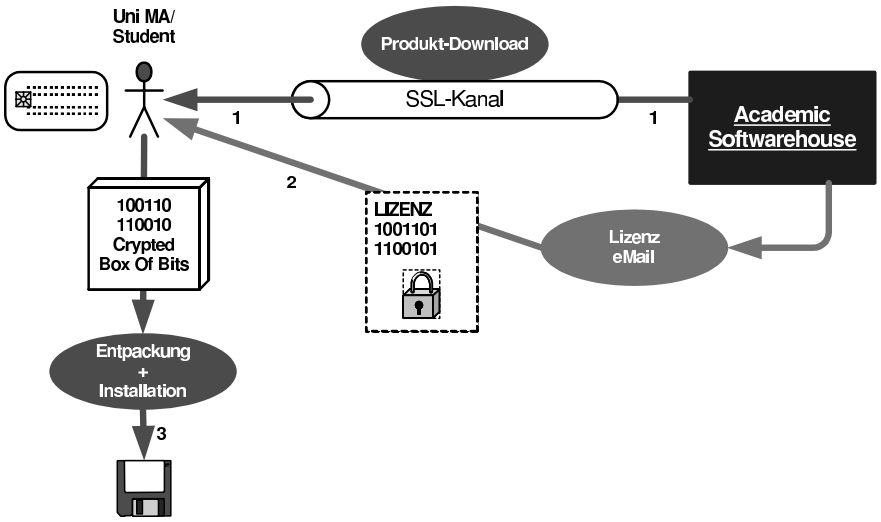

Abbildung 7: Sichere signierte Produktlieferung

Standardmäßig wird in den ask|net-Shops eine Kundenschlüssel bei der Registrierung des Kunden erzeugt und diesem per Email zugesandt.

In UNDINE ist es zusätzlich möglich, für jeden Download Schlüssel zu generieren und diese Kunden, die über ein digitales Zertifikat verfügen, mit ihrem öffentlichen Schlüssel kodiert zuzusenden.

Als weitere Sicherheitsstufe ist es möglich die Entpackung des Archives direkt an das Zertifikat zu koppeln, so daß ein Nutzer nur mit seiner Smartcard oder einem installierten Softwarezertifikat das Paket öffnen kann.

So werden Lieferpakete direkt an einen Kunden gebunden, bei dem die Weitergabe eines persönlichen Schlüssels riskant oder mit der gleichzeitigen Weitergabe eigener besonderer Privilegien verbunden sein kann.

Diese Art der Lieferung sicherer Archive verhindert nicht die Kopie vorhandener Installationen. Sie dienen allerdings als Hilfe für System-Administratoren, die die Verteilung von

gekaufter Software in einem Unternehmen kontrollieren und sich so um den Schutz der Original-Installationspakete nicht selbst kümmern müssen.

4.4 Beschaffungssystem

Digitale Signaturen bieten sich hier besonders an. Moderne Beschaffungsysteme (eProcurementsysteme) erlauben schon länger solche Arbeitsabläufe in einer Firma online per Knopfdruck abzuwickeln. Ein Nutzer, der Software anfordert, muß sich nicht mehr selbst um die Genehmigungen kümmern und kann jederzeit Einblick in den Fortgang des Genehmingungsprozesses nehmen. Ask|net bietet ein eProcurementsystem für Großkunden als Web-Applikation an, welches direkt in vorhandene Beschaffungsportale integriert werden kann.

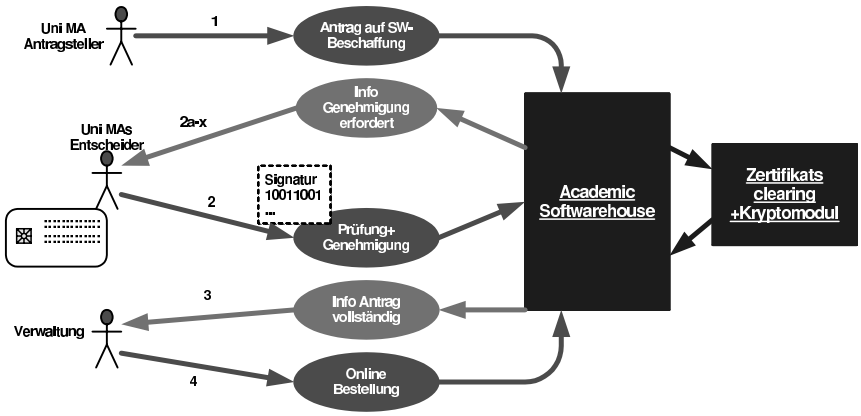

Abbildung 8: Ablauf eines Genehmigungsprozesses

In der Abbildung 8 ist der schematische Ablauf einer Genehmigung aufgezeigt. Ein Nutzer stellt seine gewünschte Bestellung ins System, Genehmigungsinstanzen können automatisch per Email informiert werden und Ihre Entscheidung in einem Formular eintragen.

Erhält eine Genehmiger, eine Aufforderung zur Genehmigung per Email so sieht er bei Aufruf der entsprechenden Seite [Abb. 9] im ask|net-eProcurementsystem den Warenkorb des Anforderers und die bisher durch andere erteilten Genehmigungen. Die Integration einer Signierkomponente erfolgt in Form eines Java-Applets welches zur Signierung mit Smartcards genutzt werden kann.

Die Signatur seiner Entscheidung führt er mit dem ask|net-Signer-Applet [Abb. 10] durch. Dieses greift über eine pkcs11-Schnittstelle auf die installierten Smartcardreader und Smartcards zu. Es benötigt dementsprechend hohe Systemrechte.

Abbildung 9: Ausschnitt aus einem Genehmigungsprozess

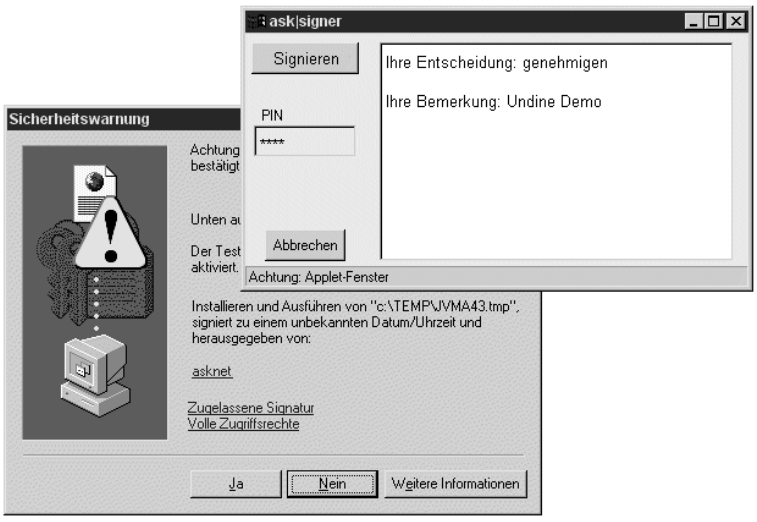

Abbildung 10: Signierung eines Genehmigungsschrittes per Signaturapplet und Smartcard

5 Server Architektur

Der Umbau der serverseitigen Applikationen erforderte den Einbau einer Zwischenschicht, den sog. „Request Broker“, der die mitgelieferten Informationen zu Nutzerzertifikaten auswertet und je nach Anforderung des Nutzers die Autorisierung bzw. die Signatur prüft bzw. bei den ausgelieferten Daten zusätzliche Ver-/Entschlüsselung oder Signierung durch ask|net vornimmt [Abb. 11].

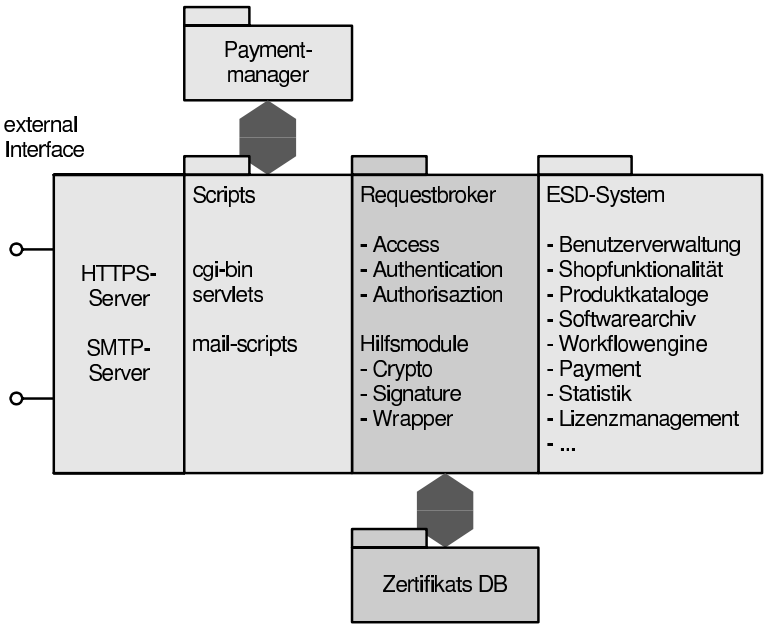

Abbildung 11: Erweiterte Architektur der Digital Logistik Engine

Diese so entworfene Architektur sollte die notwendigen Änderungen am System so gering wie möglich halten und vor- oder nachgelagerte Prozesse der digitalen Logistik weitgehend unbeeinflußt lassen. Die Realisierung prototypischer Prozesse erfolgte in perl, C und Java. Da die Serverarchitektur inclusive Quellcode vollkommen unter der Kontrolle der ask|net steht waren die notwendingen Anpassungen relativ leicht im System zu implementieren.

6 Client Architektur

Die Clients erforderten wesentlich mehr Aufmerksamkeit als zu Projektbeginn angenommen. Die ursprüngliche Annahme, daß gängige Browser die Online-Signierung von Dokumenten wie für Beschaffungswesen oder Zeichnung von Lizenzverträgen direkt unterstützen, mußte verworfen werden.

Andererseits sind die sicheren Betrachtungskomponenten, die von Smartcard-Herausgebern mitgeliefert werden, oft nur mit deren Gesamt-Lösungen für Großkunden kompatibel oder bestenfalls in Standardmailprogramme, wie Outlook oder Lotus Notes, als Pluginmodul einzubinden [Abb. 12].

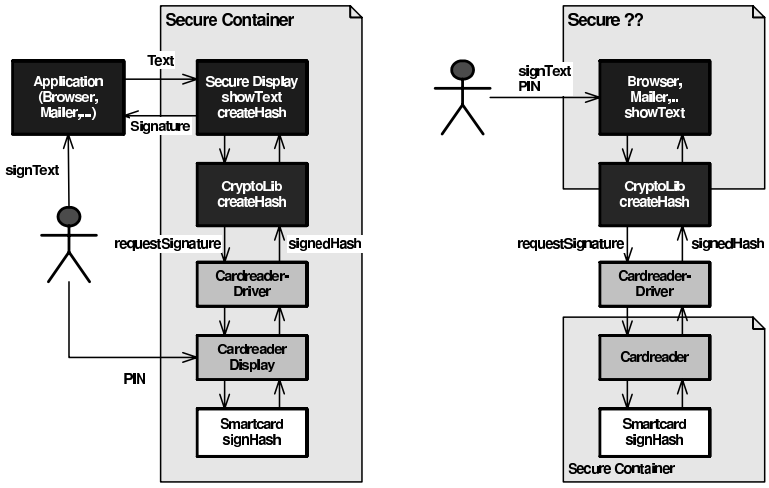

Abbildung 12: Ideale und reale Signierprozesse auf dem Client

Damit stellte sich das Problem der Entwicklung einer eigenen Darstellungskomponente, die zum einen in den gängigen Browsern funktioniert und zum anderen leicht in verschiedene Abläufe integrierbar sein sollte.

7 Fazit

Die Integration von PKI in eigene Webapplikationen gestaltet sich serverseitig durch Anwendung von Open Source-Software, wie Apache, OpenSSL, ModSSL relativ einfach. Wesentliche Module der Applikations-Logik können unberührt bleiben.

Dagegen sind auf den Clients viele Voraussetzungen für eine systemübergreifende und leichte Integration von auf Hardwaretoken basierten Krypto- und Signiertechnologie noch nicht gegeben. Die Anbieter von Lösungen sind gezwungen eigene Komponenten zu entwickeln, die ggf. verschiedene proprietäre APIs der Smartcardhersteller unterstützen.

Standards von Microsoft werden zwar häufig von Herstellern unterstützt, allerdings auf der anderen Seite fehlt die Unterstützung ibs. des Internet-Explorer für die Nutzung der Signierfunktionalität der MS-CryptoAPI. Der IE kann zudem nicht als „sichere Darstellungskomponente“ bezeichnet werden.

Die Verankerung der CryptoAPI im Betriebsystem macht eine Integration von Krypto- und Signierfunktinalität in eigene Applikationen auf der einen Seite einfach, gleichzeitig untergräbt man damit die Unabhängigkeit der Sicherheit von digital signierten Transaktionen

von der Sicherheit des Betriebsystems. Der Aufwand für Applikationsanbieter wird dadurch höher je höher die Sicherheitsanforderungen sind. Ein wünschenswerte Delegation von Verschlüsselung und Signierung durch beliebige Applikationen an sichere Komponenten ist bisher nicht realisiert. [Abb. 12]

Referenzen

Undine
Projektseite: http://www.undine.de
Ask|net: http://wwwasknet.de

Digital Payment
Untersuchung IWW - Uni Karlsruhe: http://www.iww.uni-karlsruhe.de/IZV5/
SET: http://www.setco.org
Paybox: http://www.paybox.de

PKI allgemein
Linksammlung: http://www.pki-page.org
BSI: http://www.bsi.de/literat/doc/index.htm

APIs und Standards
Opencard: http://www.opencard.org
PKCS11: http://developer.netscape.com
CryptoAPI: http://msdn.microsoft.com

Smartcardhersteller:
Utimaco: http://www.utimaco.com
Towitoko: http://www.towitoko.de
IBM: http://www.ibm.com
Basiccard: http://www.basiccard.com

Sonstige Technik
Apache Webserver: http://www.apache.org
Tomcat Servlet Engine: http://java.apache.org
OpenSLL: http://www.openssl.org
ModSSL: http://www.modssl.org
JSSE: http://java.sun.com/products/jsse/

UNDINE wird gefördert durch den Verein zur Förderung eines Deutschen Forschungsnetzes(DFN).