DB-Aspekte des E-Commerce, Schwerpunkt: Anwendungen

Thema: Online-Identität

Christian Mathis
mathis@informatik.uni-kl.de
27. Januar 2002

Arbeitsgruppe Datenbanken und Informationssysteme
Universität Kaiserslautern

Zusammenfassung: Systeme zur Bereitstellung und Verwaltung von „Online-Identitäten" bilden die Grundlage für sogenannte „Web-Dienste". Die Authentifizierung stellt die Hauptaufgabe solcher Systeme dar. Jedoch werden Benutzern oder e-Business-Partnern noch eine Reihe anderer Möglichkeiten geboten, welche zum Beispiel e-Business-Transaktionen vereinfachen. Mit Microsofts „Passport" und dem „Extensible Name Service" der Firma OneName werden in diesem Seminar zwei mögliche Lösungen vorgestellt und untersucht.

Inhaltsverzeichnis

1. Begriffliche Abgrenzung

Das Schlagwort „Online-Identität“, oder synonym „Internet-Identität“, wird in verschiedenen Zusammenhängen benutzt:

- In der Soziologie und Sozialpsychologie bezeichnet es ein Forschungsfeld, das sich mit den speziellen Ausprägungen und Veränderungen der Identität eines Menschen im bzw. durch das Kommunikationsmedium Internet, beschäftigt.
- Gelegentlich bezeichnen Web- und Domain-Hoster den Auftritt einer Firma oder Privatperson im Internet als dessen „Online-Identität“.
- Im Bereich des elektronischen Handels ist damit die Identität eines Partners im e-Commerce Geschäftsprozess gemeint. Ein Partner kann hier ein Kunde (**C**onsumer), ein Anbieter (**B**usiness) oder eine staatliche/öffentliche Institution (**A**dministration, Government) sein.

Im Folgenden werden wir den dritten Punkt behandeln. Dabei gehen wir in Kap. 2 auf die Notwendigkeit von Online-Identitäten ein, welche zur Entwicklung von Diensten zur Bereitstellung und Verwaltung von Online-Identitäten geführt hat. Solche Dienste werden im folgenden als OI-Dienste bezeichnet. Es wird ferner beschrieben, welche Anforderungen an OI-Dienste gestellt werden müssen, um Online-Identitäten sinnvoll einsetzen zu können. In Kap. 3 und Kap. 4 werden mit Passport und XNS zwei solche Dienste vorgestellt und miteinander verglichen.

2. Einleitung: Web-Dienste und Online-Identitäten

2.1. **Problemstellung.** Glaubt man den Aussagen der großen Software-Unternehmen, so soll der Mensch stärker ins Zentrum ihrer Bemühungen rücken. Dahinter steht die Erkenntnis, dass trotz jahrzehntelanger Entwicklung, aus der immer leistungsfähigere und komplexere Software hervorging, die Bedienfreundlichkeit und Durchschaubarkeit für den Endanwender hinterher hinkt [1]. In der verteilten Rechnerwelt, dem Internet, in der sich ein Benutzer heute befindet, muss dieser in der Lage sein, mit vielen verschiedenen Geräten und Bedienschnittstellen umgehen zu können. Datenredundanz bei der Speicherung der persönlichen Daten des Benutzers bei personalisierten Web-Seiten, Services und Applikationen führt zu einem erheblichen Verwaltungsaufwand, der vom Benutzer erbracht werden muss. Dabei sind Datenschutzaspekte aufgrund der weiten Verteilung der Daten schwer oder sogar unmöglich zu kontrollieren. Aus Benutzersicht wäre es ebenfalls wünschenswert die Isolation zwischen Softwarekomponenten, Web-Seiten etc. zu vermindern, um eine kontollierte Kooperation herbeizuführen. So könnte zum Beispiel bei der Buchung eines Fluges das Datum automatisch in den Kalender des Buchenden eingetragen werden.
Das Zauberwort zur Lösung dieser Probleme heisst „Web-Dienste“. Hierbei handelt es sich um IT-Dienstleistungen, die speziell auf den Anwender zugeschnitten sind und über ein Netzwerk in Anspruch genommen werden können.

2.2. **Microsofts Projekt „HailStorm“ als Beispiel für einen Web-Dienst.** Das bekannteste Beispiel für einen solchen Dienst ist ein Projekt der Firma Microsoft mit dem Codenamen „HailStorm“, welches als Teil der .NET Strategie im Frühjahr 2002 gestartet werden soll. In der Anfangsphase wird Microsoft unter anderem folgende Dienste („HailStorm Foundation Services“) bereitstellen:

- myAddress - Elektronische und geographische Adresse für eine Identität
- myProfile - Name, Spitzname, Lebenslauf, Foto
- myContacts - Elektronisches Adressbuch
- myNotifications - Austausch von Nachrichten
- myInbox - e-Mail, Voicemail, beinhaltet existierende Mail-Systeme
- myCalendar - Zeit und Aufgabenmanagement
- myDocuments - Speicherung von Originaldokumenten
- myApplicationSettings - Einstellungen von Anwendungen
- myFavoriteWebSites - URLs von Lieblingsseiten
- myWallet - Quittungen, Zahlungsinstrumente, Gutscheine etc.

- myDevices - Geräteeinstellungen, Nutzungsrechte
- myUsage - Verwendungsnachweis für obige Dienste

An sogenannten „HailStorm end-points“ (Anwendungen, Geräte, Dienste etc.) kann ein Benutzer auf die von HailStorm angebotenen Dienste zugreifen. Als Infrastruktur wird hierbei das .NET-Framework verwendet, das via SOAP und XML-Nachrichten mit diesen Diensten kommuniziert und so eine Kooperation der beteiligten Komponenten ermöglicht. Alle Daten werden nun zentral verwaltet und obliegen der Kontrolle des Benutzers. So sollen die Probleme hinsichtlich Datenredundanz und Datenschutz gelöst werden. Die Verwendung offener Standards soll Drittanbietern ermöglichen, plattform-, geräte- und netzanbieterunabhängig an HailStorm teilzunehmen und Vertrauen bei Kunden zu erwecken.

2.3. **Die Notwendigkeit von Online-Identitäten für Web-Dienste.** Die rapide technische Weiterentwicklung des Internet, z. B. in Bezug auf Datendurchsatz und Verfügbarkeit, und die Popularität unter Endanwendern und Firmen ermöglichen Herstellern von Software die Realisierung dieser neuen Ideen. Programme werden netzwerkfähig gemacht, es werden Dienste z. B. zur Kommunikation oder Datenhaltung aufgebaut, die online verfügbar sind usw. Der Trend geht vom „machine-centric application model“ zum „distributed computing model“[2]. Dieser Wechsel birgt ein neues Verständnis von Software: „Software as a Product“ wird zu „Software as a Service“. Die Bezahlung der Software erfolgt pro Benutzung oder pro Benutzungszeit. Wurde bisher die Software über eine Lizenz an die Hardware gebunden, auf der sie lief, so muss sie nun an den Benutzer gebunden werden, der sie benutzt, egal von welcher Hardware aus. Anstelle der Lizensierung tritt nun die Autorisierung.
Um eine Person zur Nutzung einer Software über das Internet autorisieren zu können, muss diese Person zuerst identifiziert werden, sie muss sich authentifizieren. OI-Dienste schalten sich zwischen die Person und den Anbieter der Software und ermöglichen die Authentifizierung über eine von ihnen bereitgestellte Online-Identität.
Eine Online-Identität ist eine Sammlung von benutzerspezifischen Daten, die einen Benutzer in einem Computer-Netzwerk eindeutig identifiziert. Dabei kann die Spanne der zur Online-Identität gehörenden Daten sehr groß sein. Für manche ist eine e-Mail Adresse allein schon eine Online-Identität, weil sie bereits global eindeutig ist, für andere gehört dazu der Name, die Adresse, Beruf, Geschlecht, Zahlungsinformationen usw. Da die Korrektheit der vom Benutzer angegebenen Daten maschinell nicht überprüft werden kann, bis evtl. auf die e-Mail Adresse, stellt sich die Frage, welche Beziehung zwischen einer Online-Identität und einer realen Identität besteht und welche Wichtigkeit dieser Beziehung bei der oben genannten Authentifizierung und Autorisierung zukommt. Diese Frage wird in Kap. 5 noch einmal aufgegriffen.

2.4. **Anforderungen an Dienste zur Verwaltung von Online-Identitäten.** Eine Online-Identität repräsentiert und identifiziert einen Benutzer im Internet. Deshalb müssen Dienste, die solche Identitäten anbieten, die Daten ihrer Benutzer vor unbefugtem Zugriff schützen. Damit die Vorteile einer Online-Identität sinnvoll genutzt werden können, müssen sie außerdem gewisse funktionale Möglichkeiten bieten:

- Für eine Privatperson:
  - Bereitstellen und Verwalten von Online-Identitäten
    Zur Online-Identität eines Benutzers gehört eine netzwerkweit eindeutige Kennung (Kombination aus Login-Name und Passwort), unter der er seine benutzerspezifischen Daten verwalten kann. Zu diesen Daten gehören demographische Daten wie Name, Geschlecht, Adresse, oder auch geschäftliche Daten wie Lieferadresse, Kreditkartennummer usw. Die Benutzerdaten müssen online verfügbar sein.
  - Single Sign-In (SSI)
    Beim Besuch von personalisierten Web-Seiten wird oft eine Anmeldung (Login) verlangt. Möchte der Benutzer aus Sicherheitsgründen nicht immer dieselbe Name-Passwort-Kombination benutzen, oder ist der gewünschte Name bei der Registrierung schon belegt, sammeln sich mit der Zeit viele verschiedene dieser Kombinationen an,

die verwaltet werden müssen. Single Sign-In ermöglicht dem Benutzer, sich mit seiner Online-Identität auf allen Web-Seiten, die SSI unterstützen, anzumelden.
  - Vereinfachung von e-Business-Transaktionen durch kontrollierte Weitergabe von Benutzerdaten
    OI-Dienste stellen Benutzerdaten zur Abwicklung von e-Business-Transaktionen automatisch bereit, damit diese nicht manuell eingegeben werden müssen.
  - Sicherheit und Privatsphäre
    Dem Benutzer muss die Kontrolle über seine Daten gegeben werden. Das heißt, er entscheidet, zu welchen Zwecken seine Daten weitergegeben werden dürfen. Die Datenhaltung und der Austausch muss mittels sicheren Systemen und Protokollen geschehen.
  - Benutzbarkeit von allen Netzwerkgeräten aus
    Für den Benutzer ist es wünschenswert, nicht nur über seinen Web-Browser auf seine Online-Identität zugreifen zu können, sondern auch von anderen netzwerkfähigen Geräten wie Telefonen, PDAs etc.
- Für ein e-Business
  - Agglomeration von Mitgliedern
    Damit der Einsatz eines OI-Dienstes, etwa zur Authentifizierung, für ein e-Business Sinn macht, ist es wünschenswert, dass der OI-Dinest über eine große Anzahl von Mitgliedern verfügen.
  - Sicherer Umgang mit Daten
    Um bei Kunden Zweifel an der Sicherheit des e-Commerce zu beseitigen, müssen OI-Dienste sicher mit Daten umgehen und diesen Umgang offenlegen.
  - Einfache Implementierung
    Der Umgang mit Online-Identitäten, die von einem OI-Dienst angeboten werden, muss für das e-Business einfach sein und wenig zusätzliche Hard- und Software fordern. Nur so lohnt sich für das e-Business die Umstellung auf ein neues System bzw. dessen Einführung.

Die vier bekanntesten Dienste, die Online-Identitäten anbieten und verwalten, sind:

- Passport von Microsoft
- Liberty Alliance von Sun und anderen
- eXtensible Naming Service (XNS) von OneName
- Magic Carpet von AOL

Anhand Passports und XNS werden wir in den nächsten Kapiteln betrachten, wie obige Forderungen umgesetzt werden.

3. Microsofts Passport

Erklärtes Ziel von Microsoft ist, mit Passport ein globales Authentifizierungsnetzwerk zu erschaffen [3]. Dabei ist Passport, von den oben genannten, schon jetzt das am weitesten entwickelte System. Microsoft verweist auf 165 Millionen registrierte Benutzer und ca. 30 Partner-Seiten[1]. Der Erfolg oder Misserfolg von OI-Diensten hängt maßgeblich vom Verwendungsgrad bei Drittanbietern ab. Je mehr Web-Seiten das Anmelden über einen OI-Dienst ermöglichen, desto reizvoller wird dessen Verwendung für Benutzer. Dementsprechend hoch sind Microsofts Bemühungen um Kunden und Geschäftspartner. Zum Beispiel hat jeder Inhaber eines Kontos bei Microsofts e-Mail-Dienst Hotmail automatisch einen Passport. Ferner wird jeder Benutzer bei der Registrierung von Windows XP dazu angehalten, sich einen Passport zu erstellen.

3.1. **Die Entwicklung von Passport.** Das Passport Projekt startete 1999, um das Microsoft Network (MSN) für Benutzer angenehmer zu gestalten. Mit einem einzigen Login konnte nun auf alle MSN-Dienste zugegriffen werden. Im „Wallet“ werden den Zahlungsverkehr betreffende Daten abgelegt, um sie bei einem Kauf innerhalb des MSN nicht jedesmal neu eingeben zu müssen. Heute wird ein Passport zum Zugriff auf alle Micosoft-Consumer-Services wie MSN Messenger,

[1] ein Drittel davon sind Microsoft-eigene Seiten

MSN Internet-Access usw. benötigt. In Zukunft stellt Passport die zentrale Komponente zur Authentifizierung in .NET dar [4].

3.2. **Passports Fähigkeiten.** In 2.4 wurden bereits die grundlegenden Anforderungen an einen OI-Dienst charakterisiert. Passport stellt diese und andere durch folgende Dienste bereit [5]:

- Single Sign-In (SSI)
- Express Purchase (EP)
- Kids Passport

SSI stellt den zentralen Passport-Dienst dar. In Zukunft werden alle Web-Dienste von Microsoft SSI zur Authentifizierung benutzen. Ein Benutzer erschafft sich eine SSI-Identität indem er sich auf der Passport-Seite (http://www.passport.com), einer Partner-Seite oder bei der Anmeldung von Windows XP registriert. Folgende Daten können dabei im Passport abgelegt werden.

- Vorname
- Nachname
- e-Mail-Adresse
- Land
- Bundesstaat
- Postleitzahl
- Zeitzone
- Geschlecht
- Geburtsdatum
- Beruf

Bei der Anmeldung auf der Passport-Seite ist mindestens die Angabe der e-Mail-Adresse erforderlich. Bei der Nutzung der anderen Möglichkeiten, d. h. beim Anmeldung auf einer Partner-Seite oder bei Windows XP, werden zusätzliche Informationen erfragt. Zur Zugriffskontrolle kann festgelegt werden, welche Informationen an Partner-Seiten weitergegeben werden sollen:

- e-Mail-Adresse
- e-Mail und Vor- und Nachname
- Alle Daten

Bei der Anmeldung auf einer Partner-Seite mittels Passport wird die e-Mail-Adresse als Login-Name und das Passwort, das der Benutzer bei der Registrierung gewählt hat, verlangt. Ferner kann der Benutzer entscheiden, ob er sich nach dem Schließen des Browsers oder dem Herunterfahren des Rechners erneut anmelden muss, oder ob seine Sitzung auch nach einem Neustart (des Browsers oder Rechners) fortgesetzt werden soll. Die erste Möglichkeit wird für Benutzer empfohlen, die an öffentlich zugänglichen Geräten Passport benutzen. Manche Partner-Seiten unterstützen „automatisches Login“. Das heißt, beim Navigieren von einer Partner-Seite zur nächsten bleibt der Benutzer angemeldet.

Um e-Commerce-Transaktionen zu vereinfachen, bietet Microsoft „Express Purchase “an. Der Benutzer legt in seinem „Wallet“ folgende Informationen ab:

- Zahlungsinformationen
  - Kartentyp
  - Kartennummer
  - Name auf der Karte
  - Gültigkeitsdatum
  - Beschreibung zur Identifizierung der Karte
- Rechnungsadresse für obige Karte
  - Vor- und Nachname
  - Adresse
  - Bundesland
  - Postleitzahl
  - Land
  - Telefonnummer
  - e-Mail-Adresse

– Beschreibung zur Identifizierung der Adresse

Bietet eine Partner-Seite die Bezahlung über EP an, so werden die Daten aus dem Wallet dem Verkäufer zu Verfügung gestellt. Der Benutzer muss sie also nicht ständig selbst neu angeben. Weil EP einfacher zu implementieren ist, und mehr direkte Vorteile als SSI bietet, ist es mittlerweile weiter verbreitet.

Kids Passport stellt zum SSI noch folgende Funktionalität bereit: Stellt eine Partner-Seite beim Zugriff mittels Kids Passport fest, dass der Benutzer unter 13 Jahre alt ist, wird er auf eine Seite umgeleitet, auf der sich ein Elternteil mit dessen Passport authentifiziert und für das Kind die angeforderte Seite freigibt oder sperrt. Das Kind hat auch die Möglichkeit, eine e-Mail an seine Eltern zu schicken, die die URL der Seite zum Freischalten der angeforderten Seite enthält.

Wie Passport die Privatspäre seine Benutzer schützt und wie sicher die Verwendung von Passport ist, wird in 3.6 diskutiert.

Um die Verwendung von Passport auf andere Geräte neben dem Computer auszudehnen, plant Microsoft in Passports Version 2.0 andersartige Authentifizierungsmechanismen, die PINs oder eine Telefonnummer benutzen.

In den nächsten drei Abschnitten wenden wir uns einigen technischen Details von Passport zu.

3.3. **Passports „Single Sign-In"-Prozess.** Die Architektur des Single Sign-In Prozesses wurde vom Authentifizierungsmechanismus für Microsofts Hotmail und Firefly Technologies Personalisierungstechnologie abgeleitet. Der Ansatz ähnelt stark der Kerberos-Architektur, welche in vielen UNIX-Systmen und auch Windows 2000 zur Authentifizierung zum Einsatz kommt [4].

Beim Anmelden via Passport sind drei Systeme beteiligt:

- Der Passport-SSI-Server. Auf ihm sind die Daten aller registrierten Passport-Benutzer zentral gespeichert.
- Der Browser des Benutzers. Die gesamte Kommunikation im SSI-Prozess findet, bis auf eine Ausnahme, über ihn statt, d. h. es gibt keinen direkten Datenaustausch zwischen der Partner-Seite und dem SSI-Server.
- Die Partner-Seite, auf der ein Passport-Manager installiert ist. Der Passport-Manager ist ein serverseitiges COM-Objekt, das die Authentifizierung auf der Partner-Seite kontrolliert. Seine Aufgaben sind das Sign-In bzw. Sign-Out und die Bereitstellung von Funktionen zum Zugriff auf Benutzerdaten. Außerdem kommuniziert er zeitweise mit dem Passport-Server, um das aktuelle Benutzer-Schema - das Format, in dem Benutzerdaten abgespeichert und verschickt werden - zu erfragen[2].

Die Anmeldung ist in Bild 1 skizziert und erfolgt in bis zu 8 Schritten (vgl. [6]):

(1) Anforderung einer Seite
Der Benutzer fordert entweder eine Seite an, für die die Authentifizierung mittels Passport benötigt wird, oder er klickt auf das „Passport Sign-In"-Symbol auf einer Web-Seite.

(2) Überprüfung auf Site-Cookies
Der Passport-Manager prüft, ob auf dem Rechner des Benutzers Passport-relevante Cookies, sogenannte „Site-Cookies" vorhanden sind. Site-Cookies sind verschlüsselte Cookies, die nur von der Partner-Seite gelesen werden können, die sie abgelegt hat. Sie existieren nur, falls der Benutzer sich schon einmal auf der Partner-Seite authentifiziert hat. Es gibt zwei Sorten von Site-Cookies:
    - Authentication-Credential-Cookie.
      Dieses Cookie enthält die sogenannte „PUID" (Passport User ID, eine global eindeutige Nummer, die als Primärschlüssel verwendet wird) und Zeitmarken, die angeben, wann die aktuelle Passport-Sitzung begann und wann sich der Benutzer das letzte Mal angemeldet hat. Partner-Seiten können in diesem Cookie gewisse Flags setzen: Wenn sich der Benutzer zum Beispiel eine gewisse Zeit nicht authentifiziert hat, kann das Cookie als „stale" (ungültig) markiert werden. Es kann ebenfalls angegeben werden, ob sich der Benutzer beim Besuch der Partner-Seite immer wieder manuell anmelden

[2]Diese ist die einzige Kommunikation zwischen dem Passport-Manager und dem SSI Server

muss, auch wenn er sich schon bei einer anderen Partner-Seite authentifiziert hat. Ist das Cookie vorhanden und nicht „stale", wird der Benutzer zur angeforderten Web-Seite umgeleitet (Schritt 8), ohne sich anmelden zu müssen.

- Profile-Cookie.
  Dieses Cookie kann einen Teil der Daten, die der Benutzer im Passport abgelegt hat, enthalten.

(3) Umleitung zum Passport-SSI-Server.
Sind die erforderlichen Site-Cookies nicht vorhanden oder „stale", so leitet der Passport-Manager den Web-Browser des Benutzers zur Authentifizierung bei Passport um. Diese Umleitung enthält eine verschlüsselte URL, auf die Passport den Browser nach einer erfolgreichen Authentifizierung zurückschickt. Außerdem enthält sie die ID der Partner-Seite und die oben genannten Flags „stale"-ness und „manuelle Anmeldung".

(4) Überprüfen der Passport-Cookies.
Der Passport-SSI-Server durchsucht den Rechner des Benutzers nach sogenannten „Passport-Cookies". Diese Cookies sind vorhanden, wenn der Benutzer aktuell bei Passport oder einer Partner-Seite angemeldet ist. Auch hier gibt es verschiedene Cookies:

- Authentication-Credential-Cookie.
  Wie auch beim Site-Cookie mit demselben Namen enthält dieses Cookie die PUID, die mit einer Zeitmarke versehen ist. Dieses Cookie ist ebenfalls verschlüsselt und nur von Passport zu lesen. Wenn das Cookie vorhanden und nicht „stale" ist, wird der Benutzer zurück zur URL geschickt, die von der Partner-Seite in Schritt 3 angegeben wurde (Schritt 7). Diese Technik erlaubt dem Benutzer das Besuchen von verschiedenen Partner-Seiten, ohne sich bei jeder einzeln anmelden zu müssen. Ist das Cookie nicht vorhanden oder „stale", oder wurde von der Partner-Seite die manuelle Anmeldung verlangt, wird der Benutzer zum Sign-In aufgefordert (Schritt 5).
- Profile-Cookie.
  Dieses Cookie enthält alle Daten, die der Benutzer bei der erstmaligen Registrierung bei Passport angegeben hat. Es ist verschlüsselt und nur von Passport zu lesen.
- Visited-Sites-Cookie.
  Dieses Cookie enthält eine Liste aller besuchten Partner-Seiten während der aktuellen Sitzung. Es wird benutzt, um beim Abmelden sicherzustellen, dass der Benutzer bei allen bisher besuchten Seiten abgemeldet wird. Auch dieses Cookie ist verschlüsselt.

(5) Co-branded Sign-in Formular.
Das Sign-In findet beim Passport-SSI-Server statt. Dennoch möchte Passport dem Benutzer das „Look-and-Feel" der Partner-Seite im Sign-In Formular bereitstellen. Deshalb werden, per clientseitigem JavaScript, Bilder und Texte von der Partner-Seite geladen („co-branding").

(6) Sign-In.
Der Benutzer gibt seine e-Mail-Adresse und sein Passwort im Sign-In Formular an und verschickt die Daten per HTTPS, welche dann mit der Datenbank verglichen werden. Die Partner-Seite hat zu keiner Zeit die Möglichkeit, das Passwort des Benutzers herauszufinden.

(7) Umleitung zur Partner-Seite.
Nach der erfolgreichen Authentifikation des Benutzers, wird dieser zu der URL zurückgeschickt, die von der Partner-Seite in Schritt 3 angezeigt wurde. Die URL enthält die, mit einer Zeitmarke versehene, PUID des Benutzers (welche Passport aus dem eigenen „Authentication-Credential-Cookie" bekannt ist) und die Daten aus dem Passport, welche der Benutzer zur Weitergabe gekennzeichnet hat. Diese Informationen sind Passport aus dem eigenen „Profile-Cookie" bekannt, welches gegen die Informationen in der Datenbank abgeglichen wird. Die URL ist triple-DES-verschlüsselt. Nur die Partner-Seite hat den Schlüssel, um die Nachricht zu dechiffrieren.
Gleichzeitig setzt oder erneuert Passport seine Cookies auf dem Rechner des Benutzers.

(8) Dechiffrieren der Informationen.
Der Passport-Manager entschlüsselt die Informationen aus der URL, setzt das Site-Cookie

auf dem Rechner der Benutzers und leitet diesen zu der in Schritt 1 angeforderten Web-Seite um.

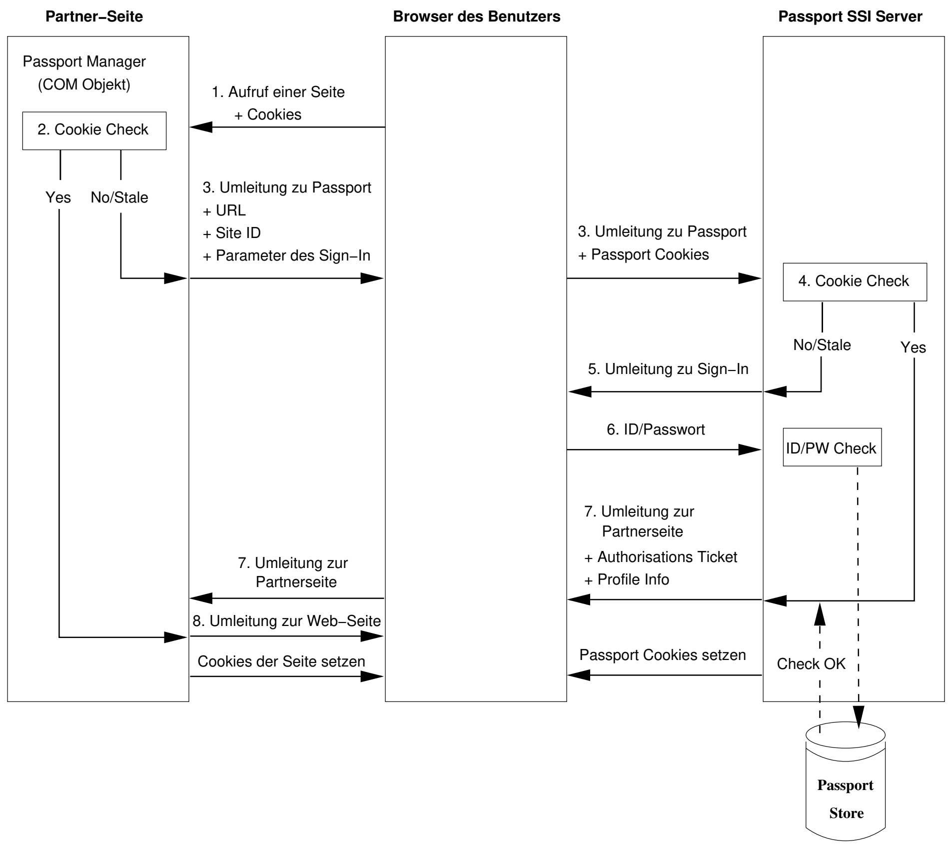

Bild 1: Illustration der Passport-Architektur nach [6]

Hat sich der Benutzer beim Sign-In für das automatische Anmelden entschieden, so bleibt er, wie oben angedeutet, auch nach einem Neustart des Rechners, angemeldet. Dazu verwendet Passport und die jeweilige Partner-Seite persistente Cookies. Meldet sich der Benutzer explizit ab, so müssen diese Cookies gelöscht werden. Klickt der Benutzer auf das Sign-Out Symbol, wird er zum Passport Server umgeleitet. Dieser liest dann aus dem Visited-Sites-Cookie, welche Seiten der Benutzer bisher besucht hat und ruft bei der jeweiligen Seite den Passport-Manager auf, welcher die Site-Cookies im Browser des Benutzers löscht.
Microsoft lässt keine Seite an Passport teilnehmen, die ihre Cookies nicht sauber löscht [7].

3.4. **Passports Express Purchase Prozess.** Die Übertragung von Zahlungsinformationen mittels EP wird über die selben Mechanismen wie beim Anmelden per Passport vollzogen (vgl. [8]). Bild 2 illustriert den Vorgang:

(1) Der Benutzer klickt bei einer Partner-Seite auf das „Express Purchase“-Symbol.
(2) Umleitung zu Passport.
Der Browser des Benutzers wird zum Passport-Express-Purchase-Server umgeleitet. Der URL wird dabei die URL der Partner-Seite, zur Rückkehr nach dem Authentifizierungsprozess, die ID der Partner-Seite, damit Passport erkennt, von welcher Seite die Anfrage stammt, und eine Liste aller Kreditkarten und Länder, für die die Seite eine Bezahlung akzeptiert, angehängt. (Diese Daten werden in Schritt 6 auf der Wallet-Seite angezeigt).
(3) Überprüfen der Cookies.
Der Passport-EP-Server durchsucht den Browser des Benutzers nach Passport-EP-Cookies.

Findet er diese, und zeigen sie an, dass sich der Benutzer innerhalb der letzten 100 Sekunden bei Passport angemeldet hat, wird der Benutzer zu seinen Wallet-Einstellungen umgeleitet (Schritt 5).

(4) Authentifizierung.
Sind keine EP-Cookies vorhanden, oder sind die vorhanden und „stale", vollzieht der EP Server mit dem Benutzer das Single Sign-In, wie in 3.3 dargestellt. Die Partner-Seite spielt in diesem Fall keine Rolle.

(5) Anzeigen der Wallet Informationen.
Ist die Authentifizierung erfolgreich, wird der Benutzer via SSL auf eine Seite umgeleitet, auf der er seine Wallet-Einstellungen prüfen bzw. ändern kann. Wie die Sign-In Seite in 3.3 ist auch diese Seite „co-branded", d. h. per JavaScript sind Texte und Bilder in den Browser geladen worden. Bei EP hat die Partner-Seite zusätzlich die Möglichkeit, zu den Passport Formularen, deren Informationen bei Passport gespeichert werden, eigene Formulare hinzuzufügen, die ihnen die Erfragung zusätzlicher Informationen ermöglichen.

(6) Eingeben und Ändern von Informationen.
Der Benutzer editiert die Informationen auf der Wallet Seite.

(7) Transferieren der Informationen.
Klickt der Benutzer auf das „Continue"-Symbol, werden seine Passport User ID (welche durch den Authentifizierungsprozess bekannt ist) und alle Informationen aus den Formularen (aus Schritt 6) über eine SSL Verbindung und die HTTPS-Post Methode an die Partner-Seite gesendet. Diese erhält somit alle Daten als ob sie vom Benutzer direkt eingegeben worden wären und beendet die Transaktion.

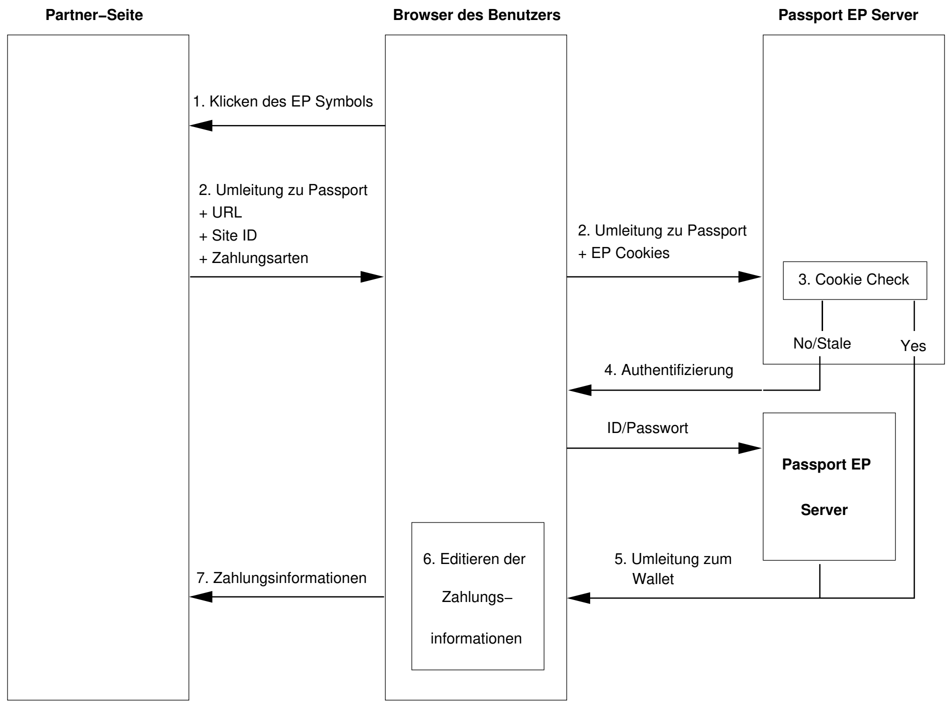

Bild 2: Illustration der Express Purchase-Architektur nach [8]

3.5. **Implementierung von Passport.** Microsoft bietet e-Business-Partnern, die Passport zur Authentifizierung einsetzen wollen, ein „Software Development Kit (SDK)" an, welches im Wesentlichen das Passport-Manager-Objekt, Administrationswerkzeuge, Dokumentationen, eine Testumgebung und eine Beispielimplementation enthält. Als Hard- und Softwarevoraussetzungen erwartet das SDK einen netzwerkfähigen Rechner auf dem Windows 2000 oder XP, die Microsoft

Internet Information Services 5.0, ein Web-Server mit Unterstützung für HTTPS und Zertifikate, installiert sind.
Die Erstinstallation von Passport findet in einer sogenannten „Preproduction“-Umgebung statt, in der die Passport Installation von Microsoft getestet und, bei einem erfolgreichen Test, genehmigt wird. Zur Erstinstallation wird daher eine vorläufige Seiten-ID benötigt, um die sich der Partner bei Microsoft bewerben muss.
Ist eine vorläufige Seiten-ID vorhanden, kann der Passport-Manager installiert werden, der den SSI Prozess überwacht. Zur Implementation von EP ist dieser jedoch nicht nötig. Partner-Seiten, die lediglich EP bereitstellen wollen, müssen nur das EP-Symbol auf ihrer Seite einbinden, welches den Benutzer zum EP-Server umleitet und die in 3.4 genannte HTTP-Zeichenfolge generiert. Außerdem müssen sie in der Lage sein, die Wallet-Informationen des Benutzers über HTTPS-Post empfangen zu können.
Nach der Installation des Passport-Managers erstellt der Partner alle Elemente, die zum „cobranding“ nötig sind.
Im Folgenden wird die neue Seite geprüft. Dabei muss sie eine Reihe von Bedingungen erfüllen. Zum Beispiel müssen die nötigen Skripte zum Entfernen von Cookies beim Abmelden funktionsfähig sein, und es muss nachgewiesen werden, dass Kundendaten „sicher“ verwahrt werden.
Nach erfolgreich bestandenem Test werden dem Partner eine „Production“-ID (und bei SSI-Seiten ein „Production“-Krypt-Schlüssel) übergeben, und er kann seinen neuen Dienst im Internet anbieten [9].
e-Business-Seiten, die eigene Systeme zur Authentifizierung einsetzen, jedoch nun zu Passports SSI migrieren wollen, können ihren Kunden zwei Möglichkeiten bieten [10]:

- Konversion. Alle Benutzer müssen sich mittels Passport anmelden. Das alte Autentifizierungssystem kann so abgelöst werden. Jedoch muss jeder existierende Kunde sein bisheriges Konto in einen Passport umwandeln.
- Koexistenz. Das alte System existiert neben Passport weiter und muss deshalb auch weiterhin gewartet werden. Der Kunde hat die Möglichkeit eins der beiden Systeme zu wählen. Es existiert hier außerdem die Möglichkeit, die bestehenden Konten mit Passports zu identifizieren, wie zum Beispiel bei www.ebay.de.

Beide Varianten erfordern zur reinen Installation von SSI einen Zusatzaufwand, der entweder bei den Kunden oder beim e-Business liegt. Express Purchase hingegen bietet für beide Parteien Vorteile: Kunden mit Passport können bequemer einkaufen, für Kunden ohne Passport ändert sich nichts. Auf der e-Business-Seite stellt lediglich die Installation des Systems einen Mehraufwand dar, das Empfangen und Verwalten von Kundendaten ändert sich nicht wesentlich. Dieses Ungleichgewicht zwischen SSI und EP sorgt vermutlich dafür, daß EP heute eine viel größere Anwendung findet als SSI .

3.6. **Datenschutz, Sicherheit und Passports Risiken.** Die zentrale Verwaltung der Benutzerdaten stellt schon beim ersten Anblick der Passport Architektur den kritischsten Datenschutzaspekt dar. Bei einer weiten Verbreitung von Passport, wie es von Microsoft geplant ist, erhält der Konzern durch das Sammeln von sogenannten „operationalen Daten“ die Möglichkeit, detaillierte Profile, die das Verhalten eines Benutzers im Internet charakterisieren, zu erstellen, d. h. Microsoft kann feststellen, welche Passport-Seiten der Benutzer zu welcher Zeit besucht hat. Ob Microsoft tatsächlich solche Informationen sammelt und auch weiterverwendet, sei dahin gestellt.
Die Datenschutzgarantie [11] von Passport sagt zur Verwendung der Benutzerdaten unter anderem folgendes:
Benutzerdaten werden von Passport zu folgenden Zwecken verwendet:

- zur Funktionsweise und dem Verhalten des Passport Kontos und der Passport Dienste,
- zum Senden diverser e-Mails, die unter anderem Umfragen und nähere Informationen zu Passport enthalten,
- zur Weitergabe an Drittfirmen, die zwecks Beantwortung von Kundenfragen oder der Erhebung von Statistiken im Auftrag von Microsoft arbeiten (eine Weiterverwendung ist untersagt),

- zur Weitergabe von statistischen Informationen an Passport Partner. Hierbei werden keine Daten, die eine Identifizierung von Benutzern erlauben, weitergegeben,
- zur Weitergabe bei Bestehen einer gesetzlichen Notwendigkeit.

Ohne persönliches Einverständnis verpflichtet sich Microsoft Passport dazu, keine Daten als in oben beschriebener Weise, weiterzugeben. Das Dokument gibt jedoch keinen Aufschluss darüber, inwieweit die Benutzerdaten Microsoft-intern weiterverwendet werden, und was mit operationalen Daten geschieht. Kritiker schlagen bei der massiven Anhäufung und zentralen Verwaltung von Benutzerdaten Alarm, denn Microsoft habe sich „keinen Namen als Fort Knox des Internet gemacht“ [12]. Darüber hinaus muss sich ein Benutzer bei der Verwendung von SSI und EP mit noch weitaus ernsteren Problemen auseinandersetzen. Sucht man im Internet nach Berichten über Sicherheitslöcher in der Passport Architektur, so wird man schnell fündig. Angreifer nutzen die Unbedarftheit des Benutzers oder den unsicheren Umgang mit Cookies aus. Unter anderem werden folgende Angriffsmöglichkeiten aufgezeigt (vgl. [13] und [14]):

- „Bogus Merchant Attack“
  Der Angreifer gibt sich als Passport-Partner aus und kopiert das Passport Anmeldeformular. Diese Attacke setzt auf die Unachtsamkeit und Unerfahrenheit von Benutzern.
- „Man in the Middle Attack“
  Der Angreifer hat die Kontrolle über einen Rechner zwischen dem Benutzer und einem Passport Partner. Bei der Umleitung (Schritt 3 in 3.3) wird der Benutzer nicht zu www.passport.com geschickt, sondern zu einem weiteren kompromittierten Rechner, der als „Passport-Proxy“ dient. Der Angreifer muss sicherstellen, dass die Kommunikation zwischen dem Benutzer, dem Passport Partner und Passport ungestört ablaufen kann. Beim Anmelden des Benutzers kann der Angreifer dessen Passwort abfangen.
- Stehlen von Cookies
  Durch Sicherheitslücken im Internet Explorer oder durch sogenanntes „Cross-Site Scripting“ [14], welches unsichere Passport-Seiten ausnutzt, werden einem Benutzer Passport-relevante Cookies gestohlen, mit deren Hilfe sich ein Angreifer als der Benutzer ausgeben kann, um so zum Beispiel auf Informationen aus dessen Wallet zuzugreifen.

Kommen wir vom datenzentralen Ansatz von Passport zu dem datenverteilten „eXtensible Name Service“ der Firma OneName. Hierzu wird, analog zur Struktur dieses Kapitels, auf die zeitliche Entwicklung von XNS, seine Fähigkeiten und seine Architektur eingegangen.

4. OneNames XNS

„...XNS is a complete web service for global identity management and negotiated data exchange.“ [15] Bis XNS soweit ist, wird noch ein wenig Zeit vergehen, denn derzeit ist XNS nur eine Web-Seite, auf der einige Whitepaper zu finden sind, die darstellen, wie XNS in Zukunft funktionieren soll. Zwar ist es schon möglich, sich bei XNS zu registrieren, da das System noch von keiner e-Busines Seite implementiert wurde, lässt sich mit der neuen Online-Identität allerdings nicht viel anfangen (bis auf die Versendung von elektronischen Visitenkarten an andere Benutzer, die bei XNS registriert sind). Alles, was in den folgenden Abschnitten vorgestellt und diskutiert wird, bezieht sich deshalb auf die zukünftige Realisierung von XNS.
XNS stellt sich folgendermaßen vor (vgl. [16]):
XNS (eXtensible Name Service) ist ein „Open Source“-Projekt der Firma OneName, welches basierend auf einem XML-Protokoll und sogenannten „Web-Agenten“ (siehe 4.3) universelle Adressierung, automatisierten Datenaustausch und gleichzeitige Kontrolle der Privatspäre seiner Benutzer anbietet. Dabei sind Web-Agenten für den Austausch, das Verbinden und das Synchronisieren von XML-Dokumenten über ein Netzwerk zuständig, während XML zur Definition einer gemeinsamen, erweiterbaren Sprache unter den Web-Agenten benutzt wird.
Aufbauend auf DNS, welches zur Adressauflösung im Internet verwendet wird, stellt XNS einen global verteilten Service dar, der von einer beliebigen Firma, Universität, ISP o. ä. implementiert werden kann. All diese sogenannten „Agenturen“ werden dabei bei der „XNS Public Trust Organization“ (XNSORG) registriert. XNSORG ist eine unabhängige, gemeinnützige Organisation, welche die „XNS Global Trust Community“ überwacht.

4.1. **Die Entwicklung von XNS.** Anfang der 90er Jahre entwickelte die Firma Intermind eine neue Technologie zum automatischen Datentransfer, die sogenannte „Kommunizierende Objekte“, benutzt. Diese Objekte wurden im Januar 1999 als Web-Agenten patentiert und ein auf ihnen aufbauender Adressierungsdienst als Open Source Projekt konzipiert. Mit der Einführung von XML als gemeinsame Sprache für Web-Agenten nannte sich Intermind in OneName um. Da XML für dieses System die Grundlage des Datenaustausches und der Repräsentation von Objekten ist, wurde es, analog zu DNS, XNS (eXtensible Name Service) genannt. Um Datenschutz-, Sicherheits- und Aspekte der Privatsphäre kontrollieren zu können, gründeten die Entwickler XNSORG.
Bisher hat es das Projekt nicht geschafft, im e-Business-Bereich oder bei anderen Institutionen, wie bei Universitäten, ISPs usw., Partner für sich zu gewinnen, was für den Erfolg des Projektes jedoch entscheidend ist. Im Moment steht XNS unter einem sogenannten NDA[3], d. h. es dürfen keine Informationen über den Status des Projektes an die Öffentlichkeit gelangen. Open Source Entwickler, die an dem Projekt Interesse zeigten, sind deshalb verärgert und befürchten, dass XNS von einem großen Unternehmen übernommen wird.

4.2. **Die Fähigkeiten von XNS.** In 2.4 wurden bereits die grundlegenden Leistungen eines OI-Dienstes charakterisiert. XNS stellt diese und andere durch folgende Dienste bereit (vgl [17]):

- XNS Online-Identity
  Die Online-Identität einer Person, Organisation oder eines e-Business wird in einen Web-Agenten abgebildet, welcher die Daten des jeweiligen Benutzers verwaltet. Aufgrund der Trennung zwischen Kunden (Consumers) und Anbietern (Business) wird zwischen privaten Agenten (personal Agents) und geschäftlichen Agenten (business Agents) unterschieden. Ein privater Agent ermöglicht in seiner Grundkonfiguration die Speicherung folgender Daten:
  - Name
  - Beruf
  - Firma
  - Telefon
  - e-Mail
  - Webadresse
  - Adresse
  - Angaben über die Kreditkarte (siehe Passport Wallet)
  - Lieferadresse (siehe Passport Wallet)

  Dabei ist jedes einzelne Feld erweiterbar, d. h. der Benutzer kann z. B. eine beliebige Anzahl von verschiedenen Telefonnummern angeben, etwa vom Mobiltelefon, Autotelefon, vom Büro etc.
- XNS Universal Address
  Jeder Agent hat eine universelle Adresse, über die er angesprochen werden kann (siehe 4.3). Da der Web-Agent alle weiteren Adressen des Benutzers verwaltet, ist der Benutzer bei Kenntnis der universellen Adresse unter all diesen Adressen erreichbar, falls er das möchte. Falls nicht, sperrt er den Zugriff auf seine Adressdaten, siehe XNS Privacy Contracts weiter unten. Ändert sich z. B. die Telefonnummer des Benutzers, bleibt er über die universelle Adresse dennoch erreichbar.
- XNS e-Cards (elektronische Visitenkarten)
  e-Cards bilden im Moment die einzige, praktisch einsetzbare Anwendung von XNS. Man kann eine e-Card an einen anderen XNS Agenten schicken oder an eine beliebige e-Mail Adresse. Der Agent des Benutzers sorgt bei einer Änderung der Daten auf der e-Card für eine Synchronisation mit bereits verschickten e-Cards.
- XNS Single Sign-In
  Das Single Sign-In unter XNS bedeutet das selbe, wie unter Passport, es wird technisch lediglich anders realisiert (siehe 4.4).

[3]Non Disclosure Agreement

- XNS Auto Form-Fill
  Auto Form-Fill stellt das Pendant zu Passports Express Purchase dar. Ein e-Business-Partner benutzt zur Definition eines XNS Formulars einen speziellen Editor zum Erstellen eines XNS Datenschemas. Dieses Schema beschreibt die im Formular geforderten Daten, welche zwischen dem Agenten, der das XNS Formular verwaltet, und dem Agenten des Benutzers ausgetauscht werden. Der Benutzer-Agent kann sogar neue Daten bei der ersten Eingabe in XNS Formulare lernen.
- XNS Privacy Contracts
  Vor dem Datenaustausch zwischen zwei Agenten wird (semi-)automatisch ein XNS Vertrag ausgehandelt, der von jedem Agenten gespeichert wird. Dieser Vertrag enthält allgemeine Datenschutz- und Sicherheitsgarantien und die Zugriffserlaubnis oder -verweigerung auf Benutzerdaten, die vom Eigentümer des Agenten definiert werden.
- XNS Linking and Synchronization
  Da bei XNS Daten beim Austausch repliziert werden, müssen sie, um aktuell zu bleiben, synchronisiert werden. Der oben genannte Vertrag kontrolliert diese automatische Synchronisation, bei der der Benutzer festlegt, welche Daten dazu freigegeben werden.
- XNS e-Mail Filters
  Um ungewollte Werbenachrichten per e-Mail zu vermeiden, bietet XNS einen e-Mail Filter an.
- Extensible Synchronization to any application or device
  Über die offengelegte XNS API lässt sich eine Verbindung von einem beliebigen Server oder Client zum Agenten aufbauen. So kann z. B. die Adressbuch-Funktionalität von XNS auf e-Mail Programme, PDAs usw. ausgeweitet werden.
- XNS public directory service
  Zum Auffinden von Privatpersonen, Organisationen etc. stellt XNS ein öffentliches Verzeichnis bereit.

In den folgenden beiden Abschnitten werden wir uns nicht alle technischen Details zur Realisierung dieser Funktionen betrachten, sondern nur solche, die für Online-Identitäten essentiell sind. Im Speziellen gehören dazu die Punkte Universal Address, Single Sign-In, Auto Form-Fill, Privacy Contracts und Linking and Synchronization.

4.3. **Web-Agenten und XML als Schüsselkonzepte der XNS Architektur.** In der frühen Phase des Internet war der Datenfluss zwischen Servern und Clients unidirektional: Server speicherten Daten (Dateien in HTML und Links), die von Clients per HTTP angefordert werden konnten. Um Daten auch von Clients zu Servern senden zu können, etwa um Internetseiten mit Passwörtern zu versehen, musste dieses Modell erweitert werden. Hinzu kamen HTML-Formulare. Die charakteristischen Eigenschaften hinsichtlich Datenhaltung und Funktionalität von Clients und Servern verdeutlicht grob folgende Tabelle (vgl. [18]):

Client (Web Browser)	**Server** (Web Server)
Beschränkte Speicherung von Daten (Cookies)	Großteil der Datenspeicherung
Beschränkte Speicherung von Links (Bookmarks, Browser-History)	Großteil der Speicherung von Links
Keine Speicherung von Formularen	Gesamte Speicherung der Formulare
Fordert Daten an	Liefert Daten zurück
Geringe Automatisierung (Web Scripting)	Vollständige Automatisierung

Tabelle 1: Eigenschaften der Client/Server Architektur

Die Kommunikation und der damit zusammenhängende Datenaustausch zwischen zwei Individuen findet außerhalb des Internet (in der Regel) nicht nur in eine Richtung statt. Personen, Firmen und Organisationen beziehen Informationen und verteilen sie. Um dieses bidirektionale Kommunikationsverhalten ins Internet abbilden zu können, wurden Web-Agenten entwickelt.
Web-Agenten sind Softwarekomponenten, die Client- und Serverfunktionalität in sich vereinen

und dadurch das in obiger Tabelle dargestellte Ungleichgewicht nivellieren. Alle Web-Agenten sind gleichwertig, sie speichern Daten zur Weitergabe, Formulare zur Anforderung von Daten und Links zur Navigation. Die Kommunikation kann somit in beide Richtungen erfolgen, es entsteht ein Peer-to-Peer Netzwerk.
Da Web-Agenten als Kommunikationsmittel für eine große Anzahl verschiedenartiger Individuen mit unterschiedlichen Bedürfnissen und Informationsstrukturen dienen sollen, stellen Erweiterbarkeit und Skalierbarkeit zwei wichtige Anforderungen an Web-Agenten dar. Die guten Erweiterbarkeitseigenschaften von XML sollen sich bei deren konsequentem Einsatz zur Kontrolle, Datenhaltung und Kommunikation von Web-Agenten auf diese übertragen.
Folgende Grafik veranschaulicht die Kommunikation unter Web-Agenten:

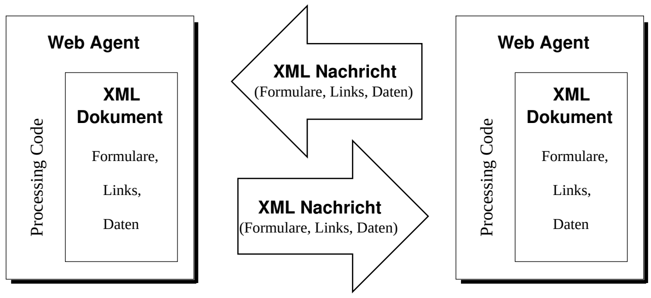

Bild 3: Kommunikation zwischen Web-Agenten nach [18]

Damit Web-Agenten miteinander kooperieren können, müssen sie in eine Umgebung eingebettet werden, die ihnen einen gewissen funktionalen Rahmen bietet. Dieses Framework wird von XNS bereitgestellt:

- Ein Wörterbuch für Web-Agenten
  XML-Dokumente und -Nachrichten sind nur syntaktische Gebilde. Damit Agenten Daten und Nachrichten interpretieren können, muss den Sprachelementen von XML eine Semantik gegeben werden, dies geschieht im XNS Wörterbuch.
- Eine Agentur zur Verwaltung von Agenten
  Ein Web-Agent ist ein Programm, dass auf einem XNS-Server läuft. Damit nicht jeder Benutzer seinen eigenen XNS Server zur Verwaltung seines Agenten betreiben muss, werden Agenten in Agenturen gehostet. Agenturen werden von ISPn, Universitäten, Firmen o. ä. eingerichtet und verwaltet.
- Ein Adressierungssystem für Web-Agenten (vgl. [19])
  Agenturen und ihre Agenten sind über das gesamte Internet verteilt. Die Verbindung zwischen ihnen stellt die XNS Wurzel Agentur (XNS Root Agency) dar, welche ein globales Adressbuch führt, in dem alle Agenturen und Agenten aufgeführt sind. Zur Lokalisierung der einzelnen Agenten dienen spezielle Adressen, die aus drei Teilen bestehen:
  – Ein Name, der aus einer 64-stelligen Zeichenfolge in Unicode besteht und vom Benutzer vergeben wird. Der Benutzer spricht seinen Agenten über diesen Namen an. Der Namen eines Agenten kann auch geändert werden kann, wobei der Agent jedoch derselbe bleibt. Der Name liegt außerdem in einem der drei Namensbereiche „Personal“, „Business“, „General“, die der Zuordnung der Agenten zu einem spezifischen Kontext dienen.
  – Ein XNS Pfad (Zeichenfolge aus Ziffern und Punkten) beschreibt, bei welcher Agentur der Agent zu finden ist. Wechselt der Agent zu einer anderen Agentur, so ändert sich dieser Pfad.
  – Eine XNS Agenten ID, welche sich für den Benutzer niemals ändert. Sie stellt den Primärschlüssel zum Auffinden eines Web-Agenten dar.

  Diese Vorgehensweise hat einige Vorteile:

  - Der Name ist für einen Menschen leicht lesbar.
  - Ein Web-Agent kann mehr als einen Namen haben.
  - Die maschinelle Verarbeitung wird durch den XNS Pfad unterstützt.
  - Mit einem Link (Datenverbindung) zwischen zwei Agenten werden immer die zugehörigen IDs der Agenten verwaltet. Ändert ein Agent seinen Namen oder seinen Ort, so ist er über diese ID immer noch zu erreichen. Deshalb „brechen" Links zwischen Agenten nicht, wenn es nicht explizit erwünscht ist.

  Ein URN[4] zur Lokalisierung eines Agenten sieht z. B. folgendermaßen aus:
  http://xnsresolver.uni-kl.de/4.53921(6B29FC40-00DD010662DA)7.235!=ChristianMathis
  Dabei sind
  - **http://xnsresolver.uni-kl.de** die DNS Adresse des XNS Adressauflösungsnetzwerks
  - **4.53921** der erste Teil des Pfades, welcher die Agentur des Agenten angibt
  - **(6B29FC40-00DD010662DA)** die ID des Agenten
  - **7.235** der zweite Teil des Pfades, welcher den Ort des Agenten bei der Agentur angibt
  - **!=ChristianMathis** der Name eines Agenten für Privatpersonen
- Spezielle Agenten
  Ein großer Vorteil von Client-Server-Architekturen ist, dass ein Client nicht ständig online sein muss, und von verschiedenen Orten aus eingesetzt werden kann. Um diese Funktionalität auf die XNS Architektur übertragen zu können, wurden sogenannte „spezielle Agenten" entwickelt. Dies sind Applikationen, die auf Laptops, PDAs, Mobil-Telefonen usw. laufen und einen Agenten als Server benutzen. So kann zum Beispiel ein Agent ein Adressbuch verwalten, auf das vom Laptop oder Mobiltelefon zugegriffen wird. Typische Aufgaben, die der Agent für den speziellen Agenten erledigt, sind:
  - Authentifizierung des Benutzers eines speziellen Agenten
  - Suchen eines anderen Agenten
  - Datentransfer zu anderen Agenten
  - lokale Datenhaltung
  - Synchronisation von lokalen Daten
- Datenaustausch per XNS-Formular (vgl. [20])
  XNS-Formulare sind Objekte zum Datenaustausch, die mittels XML definieren, welche Daten benötigt werden. Braucht zum Beispiel ein e-Business beim Verkauf die Lieferadresse und die Daten der Kreditkarte eines Kunden, so definiert das e-Business ein XNS-Formular „Purchase" und stellt es über ihren Business-Agenten bereit, etwa unter der URN:
  http://xnsresolver.net/4.53921(6B29FC40-00DD010662DA)7.235!@HM/Forms/Purchase
  Der Datentransfer zwischen Kunde und e-Business erfolgt dann in folgenden acht Schritten, die in Bild 4 illustriert sind:
  (1) Der Benutzer schickt für ein Formular eine HTTP-Get Anforderung an einen XNS-Resolver. Die Anforderung enthält dabei ein Session-Cookie, falls sich der Benutzer während der aktuellen Browser-Sitzung schon einmal bei seinem Agenten angemeldet hat.
  (2) Falls das Cookie im HTTP-Get vorhanden ist, leitet der XNS-Resolver den Benutzer zu seiner Agentur um. In dieser Umleitung ist die Adresse des angeforderten Formulars enthalten. Falls das Cookie nicht vorhanden ist, stellt der XNS Resolver ein Anmeldeformular für den Benutzer bereit, dass ebenfalls mit der Umleitung mitgeschickt wird.
  (3) Die Agentur des Benutzers authentifiziert den Benutzer anhand seines Cookies oder Passwortes.
  (4) Die Agentur des Benutzers sucht den Business-Agenten und dessen Agentur und verlangt eine Kopie des angeforderten Formulars.
  (5) Die Agentur des Benutzers füllt ein dem erhaltenen XNS-Formular entsprechendes, HTML-Formular mit den Daten des Benutzers und schickt es an dessen Browser.

[4] „Uniform Resource Name„, siehe http://www.dlib.org/dlib/february96/02arms.html, oder RFC 1737

(6) Der Benutzer wählt darauf hin die Daten aus, die er freigeben möchte und entscheidet sich für gewisse Verträge zur Wahrung des Datenschutzes (dazu mehr im nächsten Punkt).
(7) Das Formular wird an die Agentur des Benutzers zurückgesendet und von dessen Agenten bearbeitet. Der Agent stellt einen Link zu dem Business-Agenten her und vermerkt darunter die Datenschutzverträge aus dem vorherigem Schritt. Dann führt er abhängig vom Formular einen der beiden folgenden Schritte aus:
   – Durch eine XML Nachricht werden die Daten zum Business-Agenten übertragen
   – Die Daten werden per HTML-Formular an eine vom Business-Agenten festgelegte URL geschickt (zur Rückwärtskompatibilität mit existierenden CGI-Skripten)
(8) Der Agent des Benutzers schickt dessen Browser zu einer Bestätigungsseite auf der Webseite des Anbieters.

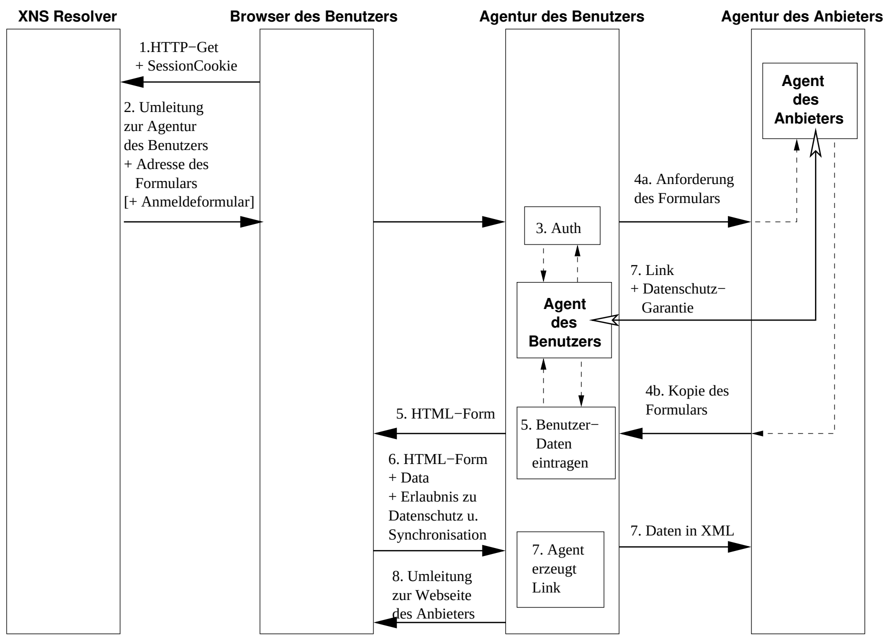

Bild 4: Illustration des Datenaustausches per XNS-Formular

- Verträge zur Wahrung des Datenschutzes
  Durch solche Verträge kann beim Datenaustausch der Verwendungszweck, die Dauer der Verwendung und die Weitergabe an Dritte festgelegt werden.
- Erweiterbarkeit von Agenten: Knowledgepacks
  Ein Kowledgepack ist ein Programm, das zu einer Agentur hinzugefügt wird, um die Funktionalität ihrer Agenten zu erweitern. Dabei ist ein Knowledgepack selbst ein Agent, mit einer eigenen Adresse und seinen eigenen XML Daten, welche bestimmen, welcher Dienst anderen Agenten angeboten wird.

4.4. **Online-Identitäten unter XNS.** Mit der Vorstellung von XNS in den beiden vorherigen Kapiteln wurden bereits die grundlegenden Konzepte einer Online-Identität in diesem System behandelt. Da das Gesamtsystem aus vielen kleinen einzelnen Aspekten besteht, sei hier noch einmal eine kurze Zusammenfassung gegeben:
Die Online-Identität von Personen, Organisationen, Firmen usw. wird durch einen Web-Agenten repräsentiert, der im XNS Framework eingebettet ist, das die Kommunikation bzw. den Datenaustausch zwischen Web-Agenten ermöglicht. Ein Benutzer legt Daten in seinem Web-Agenten ab und definiert, wer, wie lange auf welche Daten zugreifen darf. Über Links werden Daten, die

zwischen Agenten ausgetauscht wurden, synchron gehalten. Da der Zugriff auf einen Agenten und der Datenaustausch durch das Benutzerpasswort geschützt sind, kann ein Web-Agent zur Authentifizierung eines Benutzers herangezogen werden. Die Erweiterbarkeit von Agenten durch Knowledgepacks ermöglicht es, XNS als eine Entwicklungsplattform für Web-Dienste zu benutzen.
Das Single Sign-In entsteht in der XNS Architektur sozusagen als Nebenprodukt des allgemeinen Datenaustausches: Klickt ein Benutzer auf das Single Sign-In Symbol auf einer XNS Partner-Seite, so prüft der Agent des Benutzers, ob bereits ein XNS Link, und damit ein XNS Vertrag, zu dieser Seite, die von einem Business-Agenten überwacht wird, besteht. Falls nicht, wird dem Benutzer ein mit seinen Daten vorgefertigtes Formular zur Registrierung auf dieser Seite gesendet, bei dessen Versendung ein Link zwischen dem Benutzer-Agenten und der XNS Partner-Seite entsteht. Beim nächsten Besuch der Seite wird der Benutzer automatisch von seinem Agenten auf der Seite angemeldet.

4.5. **XNS versus Passport.** Microsofts Passport und XNS der Firma OneName sind Dienste, die es sich zum Ziel gesetzt haben, Internetbenutzer, seien es Firmen, Organisationen oder Privatpersonen, mit einer Online-Identität zu versehen. Obwohl sie im Kern die selben Leistungen anbieten, wie automatisches Vervollständigen von Formularen, Single Sign-In, Schutz der Privatsphäre, etc., ist ihre Herangehensweise sehr verschieden. Während Passport sich der Lücke zwischen Konsumenten und Anbietern anzupassen versucht, indem es beiden Parteien einen einfach anzuwendenden Dienst an die Hand gibt, der nicht mehr Funktionalität als nötig besitzt und der auf aktuellen Technologien aufbaut, sieht XNS seine Aufgabe in der Entwicklung einer kompletten, extrem erweiterbaren Netzplattform, auf der Online-Identitäten die aktiven Elemente sind.
Im Gegensatz zu Passport, das durch seine datenzentralisierte Architektur den Vorteil hat, den Aufwand zur Datenverwaltung an einer Stelle zu haben, sind bei XNS die Daten mehr oder weniger verteilt, was eine aufwändige Infrastruktur für die Datenhaltung verlangt. Obwohl beide Systeme stark verschieden sind, z. B. „XNS Web-Agenten“ versus „Eintrag des Benutzers in einer Datenbank“, lässt sich eine gewisse Ähnlichkeit bei der Realisierung des Datentransfers erkennen (vgl. 3.3, 3.4 und 4.3). Hinsichtlich Vielfältigkeit der speicherbaren Daten und Zugriffskontrolle bietet XNS mit seinem auf Erweiterbarkeit bedachten Ansatz dem Benutzer mehr Möglichkeiten als Passport: Den drei Abstufungen „Zugriff auf die e-Mail-Adresse“, „Zugriff auf e-Mail und Namen“ und „Zugriff auf alle Daten“ bei Passport steht das mächtige Konzept der Verträge zur Kontrolle des Zugriffs und der Verwendung der Benutzerdaten gegenüber. Diese Verträge erlauben z. B. für jedes Datenfeld anzugeben, ob ein fremder Agent darauf zugreifen darf oder nicht. Passport kümmert sich im Gegensatz zu XNS nicht um das Aktualisieren von bereits im Internet verteilten Daten. Zwar kann man im Passport eine neue Adresse angeben, jedoch wird diese nicht automatisch bei allen Seiten aktualisiert, die bisher auf die Adresse zugegriffen haben. An diesen kleinen Beispielen wird schnell deutlich, dass XNS das mächtigere Konzept der beiden darstellt. Dennoch wird vermutlich, neben dem Faktum, dass mit Microsoft das größte und mächtigste Softwareunternehmen hinter Passport steht, die einfachere Handhabung für eine größere Verbreitung von Passport sorgen.

5. Zusammenfassung und Ausblick

„Web-Dienste sind der Millardenmarkt der Zukunft“ [22], und Online-Identitäten spielen in diesem Bereich die wichtigste Rolle zur Authentifizierung. Dienste, die Online-Identiäten bereitstellen und verwalten, bieten Konsumenten und Anbietern eine Reihe von Vorteilen, darunter Single Sign-In, die Vereinfachung von e-Business-Transaktionen usw.
Mit Passport versucht sich Microsoft als Authentifizierungsdienst zwischen Konsumenten und Anbietern zu etablieren. Durch die leichte Implementierung von Passport auf der Anbieterseite und die einfache Nutzung ohne zusätzliche Software auf der Anwenderseite, erhofft sich Microsoft eine weite Verbreitung seines Systems.
Auf Web-Agenten aufbauend will XNS einen globalen Web-Dienst anbieten, der Identity-Management und kontrollierten Datenaustausch ermöglicht. Dabei repräsentieren Web-Agenten einen

Benutzer im Internet und sind gleichzeitig dessen Schnittstelle, um mit anderen Agenten zu kommunizieren. Agenten werden in das XNS-Framework eingebettet, das diese Kommunikation durch Adressierungsmechanismen, XNS-Formulare, Datenschutzverträge usw. unterstützt.
Passport hat für Version 2.0 einige Neuerungen angekündigt, darunter z. B. P3P-Unterstützung, Inline Sign-In, Mobile-Passport usw. Wie die Zukunft von XNS aussieht lässt sich schwer sagen. Es bleibt abzuwarten, was nach der Aufhebung ihres NDAs geschieht. Auf jeden Fall werden mit der verstärkten Einführung von Web-Diensten Online-Identitäten eine wichtigere Rolle spielen als heute. Da Benutzerdaten Kundendaten sind und Kundendaten den direkten Zugang zum Käufer darstellen, werden Firmen Online-Identitäten mit ihren Applikationen zu verbinden wissen, wie es am Beispiel Microsoft schon deutlich zu erkennen ist. Bisher steht Microsoft mit Passport noch allein da, aber mit der Liberty Alliance, einem Projekt, das vom Microsoft-Rivalen SUN initiiert wurde und mittlerweile ca. 30 Mitglieder zählt, darunter AOL, Vodafone, Cisco, Hewlett-Packard und andere, kündigt sich Konkurrenz an. Interessant bleibt außerdem noch abzuwarten, wie stark Online-Identitäten an einen Menschen gekoppelt werden, denn bei den bisherigen Systemen ist es möglich, sich für jemand anderen auszugeben. Vorstellbar sind Lösungen, die die Benutzererkennung am Endgerät ermöglichen, etwa durch Überprüfung biometrischer Daten oder durch das Verlangen einer Smartcard zum Zugriff auf seine Online-Identität.

Literatur

[1] Microsoft HailStorm Whitepaper, Microsoft Corporation, 2001,
http://wallyswisdomwarehouse.weblogs.com/stories/storyReader$135
[2] „HailStorm, Open Services controlled by Microsoft“, Clay Shirky 30.05.2001,
http://www.openp2p.com/pub/a/p2p/2001/05/30/hailstorm.html?page=1
[3] „Q&A: Open Passport Enables a ‚Network of Trust‘“, Microsoft Corporation, 2002,
http://www.microsoft.com/presspass/features/2001/sep01/09-20passport.asp
[4] „A Closer Look at Passport“, Matt Rosoff,
http://www.directionsonmicrosoft.com/sample/DOMIS/update/2001/10oct/1001aclap.htm
[5] „.NET Passport Fact Sheet“, Microsoft Corporation, 2001,
http://www.passport.com/Business/PassportFactSheet.asp?PPlcid=1033
[6] „Passport Single Sign-In Process“, Matt Rosoff,
http://www.directionsonmicrosoft.com/sample/DOMIS/update/2001/10oct/1001aclap_illo1.htm
[7] „Passport Sign-Out“, Matt Rosoff,
http://www.directionsonmicrosoft.com/sample/DOMIS/update/2001/10oct/1001aclap_sb1.htm
[8] „Passport Express Purchase Process“, Matt Rosoff,
http://www.directionsonmicrosoft.com/sample/DOMIS/update/2001/10oct/1001aclap_illo2.htm
[9] „Passport Documentation and Requirements“, Microsoft Corporation, 20.11.2001
http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=/MSDN-FILES/027/001/644/msdncompositedoc.xml
[10] „Microsoft Passport, How it works“, Neal Joffee, 26.06.2001,
http://www.internet-technology.org/idsc6441_redwings/howitworks.html
[11] Microsoft Passport Datenschutzgarantie, Microsoft Corporation, 01.08.2001,
http://www.passport.com/consumer/privacypolicy.asp?PPlcid=1031
[12] „US-Datenschützer gegen Windows-XP“, Heise Newsticker, 26.07.2001,
http://www.heise.de/newsticker/data/kav-26.07.01-000/
[13] „Risks of the Passport Single Sign-On Protocol“, David P. Kormann and Aviel D. Rubin, Computer Networks, Elsevier Science Press, volume 33, pages 51-58, 2000.
[14] „Microsoft Passport to Trouble“, Marc Slemko, 05.11.2001,
http://alive.znep.com/∼ marcs/passport/
[15] „Defining and Comparing XNS“, XNS Public Trust Organization, Juni 2001,
http://www.xns.org/xns/whitepapers/compare/
[16] „Top Ten Questions about XNS“, XNS Public Trust Organization,
http://www.xns.org/xns/faqs/
[17] „XNS in a Nutshell“, XNS Public Trust Organization,
http://www.xns.org/xns/nutshell.html
[18] „How Web Agents work: A quick Primer“, XNS Public Trust Organization,
http://www.xns.org/xns/whitepapers/webagents/
[19] „XNS Naming and Addressing“, XNS Public Trust Organization, 08/2001
http://www.xns.org/xns/whitepapers/naming.html
[20] „NS Data Exchange FAQ“, XNS Public Trust Organization, 08/2001
http://www.xns.org/xns/whitepapers/dataexchange.html

[21] „Kundendaten sind die Schätze der Firma“, Computer Zeitung Nr.41, 11.10.2001
[22] „Liberty Alliance schafft dezentrale Norm für eine Online-Identität “, Computer Zeitung Nr.41, 11.10.2001