Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH)
Fachbereich Informatik, Mathematik und Naturwissenschaften

Diplomarbeit

Rechtssichere Reservierungen über das Internet am Beispiel von WorldCheckInn

zur Erlangung des akademischen Grades eines Diplom-Informatikers (FH)

Christian Thiele

Betreuer: Prof. Dr. Klaus Bastian

Leipzig, 31. Juli 2006

Inhaltsverzeichnis

1. Einleitung

WorldCheckInn stellt einen Dienst für Beherbergungsunternehmen zur Verfügung. Vielreisende hinterlegen in einem vertraulichen Verzeichnis ihre Melde- und Abrechnungsdaten und autorisieren bei der Inanspruchnahme einer Dienstleistung (Reservierung, Check-In, ...) den Hotelier zum einmaligen Zugriff auf diese Daten. Neben Melde-, Stamm- oder Abrechungsdaten enthält das Verzeichnis auch Zusatzinformationen, wie z.B. persönliche Wünsche des Vielreisenden.

WorldCheckInn ist ein Internet-Unternehmen des Hoteliers Alexander Wussler. Im Rahmen einer Zusammenarbeit mit Matthias Schilha und unter Betreuung von Prof. Dr. Klaus Bastian (Fachbereich Informatik, Mathematik und Naturwissenschaften HTWK-Leipzig) wurden technische Grundlagen des Systems WorldCheckInn und deren Umsetzung erarbeitet. Ziel des Systems ist es, auf Basis validierter und beglaubigter Personendaten Geschäftsvorgänge Vielreisender zu vereinfachen und medienbruchfrei abzuwickeln. Dazu zählen das Abschließen rechtsverbindlicher Beherbergungsverträge sowie die Erfüllung der Meldepflicht in Beherbergungsunternehmen.

1.1 Zielsetzung

Die vorliegende Diplomarbeit befasst sich einerseits mit der Frage, inwiefern es möglich ist, Reservierungen über das Internet rechtssicher zu gestalten und wie weit die Automatisierung des Datentransfers unter Einhaltung gesetzlicher Bestimmungen des Datenschutzes möglich ist. Dabei stehen vor allem die Übermittlung von personenbezogenen Daten und die Erfüllung der Meldepflicht im Vordergrund. Für den Abschluss eines rechtssicheren Vertrages werden bereits bestehende Systeme und theoretische Modelle zum Abschließen elektronischer Verträge verglichen und Parallelen zum System WorldCheckInn aufgezeigt.

Anderseits wird betrachtet, wie das System gegen Angriffe über das Internet geschützt werden kann. Daraus ergeben sich wichtige Anforderungen an die technische Umsetzung des Systems. Als Leitfaden wird das IT-Grundschutz-handbuch des Bundesamtes für Sicherheit in der Informationstechnik (BSI) herangezogen und durch weitere Quellen ergänzt. Die Schwerpunkte hierbei bilden die sichere Installation und Konfigurationen des Windows Servers 2003 und des Internet Information Servers, sowie die Umsetzung von Sicherheitsmaßnahmen unter ASP.NET. Hinzu kommt die Verwendung der Spezifikationen der Global Web-Service Architecture (GXA), deren praktische Umsetzung durch Verwendung von Web-Services Enhancements (WSE) unter ASP.NET vorgestellt wird.

1.2 Gliederung

Um einen besseren Einstieg zu ermöglichen, stellt Abschnitt 1.3 zunächst Verträge und Reservierungsformen in Beherbergungsunternehmen vor. Dieser Abschnitt beschreibt den Ist-Zustand und die damit verbundenen Probleme. Im zweiten Kapitel wird das System WorldCheckInn vorgestellt. Es wird eine Übersicht über Ziele, Funktionalitäten, Zielgruppen und das Geschäftsmodell gegeben.

Kapitel 3 umfasst das Thema Rechtssicherheit. In der ersten Hälfte des Kapitels werden rechtliche Rahmenbedingungen vorgestellt, die im Sinne des Datenschutz-gesetzes erfüllt sein müssen, um im Umgang mit personenbezogenen Daten rechtliche Probleme zu vermeiden. Eine besondere Rolle spielt die Erfüllung der Meldepflicht im Bereich der Unterbringung. Es wird geprüft, inwiefern das System WorldCheckInn zur Erfüllung der Meldepflicht beitragen kann. Im zweiten Teil dieses Kapitels werden drei Modelle zur Abwicklung von Rechtsgeschäften im Internet vorgestellt und Parallelen zu dem System WorldCheckInn gezogen. Schwerpunkt ist das Modell nach Wabner, ein Schuldvertragsmodell für den elektronischen Handel. Im Anschluss daran werden die von Schilha definierten, abstrakten Protokollvarianten vorgestellt und auf eine Eignung für die praktische Umsetzung unter WorldCheckInn untersucht. Dem folgt eine Betrachtung von WCI als Softwareagenten.

Ein mit elektronischen Verträgen oftmals in Zusammenhang gebrachter Punkt ist der Einsatz elektronischer Signaturen. Diese werden am Ende des Kapitels näher beschrieben, ihre Beweiswürdigung vorgestellt und ein möglicher Einsatz unter WorldCheckInn betrachtet.

Kapitel 4 beschreibt technische Voraussetzungen und Maßnahmen für sichere Reservierungen über das Internet. Zu Beginn des Kapitels werden die Architektur des Systems, die Systemkomponenten und die Entwicklungsumgebung beschrieben. Anschließend wird das Grundschutzhandbuch des Bundesamtes für Sicherheit in der Informationstechnik als Leitfaden für die weitere Betrachtung des Systems herangezogen. Daraus folgt eine Einteilung des Systems in drei Ebenen. Die erste Ebene beschreibt im Wesentlichen die Bedingungen, die erfüllt sein müssen, um eine solide Grundlage zur Absicherung der Serverkomponenten zu schaffen. Bei der zweiten Ebene, der Anwendungsebene, welche die Schnittstelle zur Außenwelt darstellt, werden Authentifizierungsmethoden unter .NET mit ihren Vor- und Nachteilen aufgeführt. Es werden mögliche Angriffe aufgezeigt und allgemeine Maßnahmen sowie Maßnahmen unter ASP.NET vorgestellt. Im letzten Abschnitt werden zur Sicherung der Übertragungsebene, welche die dritte Ebene darstellt, die Spezifikationen der Global Web-Service Architecture (GXA) vorgestellt. Diese stehen unter dem Namen Web Service Enhancements (WSE) seit Dezember 2005 in der Version 3.0 für ASP.NET zur Verfügung.

Dem schließt sich in Kapitel 5 eine Zusammenfassung der Kapitel 3 und 4 an. Die in diesen beiden Kapiteln besprochenen Punkte werden noch einmal konkretisiert und die daraus erzielten Erkenntnisse vorgestellt. Anschließend werden noch offene Punkte besprochen und es wird ein Ausblick auf weiterführende Arbeiten zu diesem Thema gegeben.

Das Abkürzungsverzeichnis, das Glossar der verwendeten Fachbegriffe, das Literaturverzeichnis und die Beschreibung des beiliegenden Datenträgers befinden sich am Ende der Arbeit.

1.3 Verträge im Bereich der Unterbringung

Prinzipiell ist ein Vertragsabschluss ein zweiseitiges Rechtsgeschäft: Im Warenhandel kommt ein Vertrag immer zwischen einem Käufer und einem Verkäufer zustande. Somit sind einseitige Änderungen oder Ergänzungen durch den Käufer oder den Verkäufer unwirksam. Diese bedürfen jeweils einer Einwilligung des Anderen.

Ein (Kauf-)Vertrag ist zweiseitig verpflichtend, da zwei Schuldverhältnisse bestehen (§433 BGB). Der Käufer ist zur Zahlung des Kaufpreises und Abnahme der Ware verpflichtet. Der Verkäufer hingegen ist zur Übergabe der Ware verpflichtet und muss dem Käufer das Eigentum daran verschaffen (§433 BGB).

Angebot und Annahme kommen zustande, wenn eine der Parteien ein Angebot macht, welches durch die andere Partei angenommen wird. Im eCommerce ist die Präsentation eines Online-Warenangebots jedoch nicht als Angebot zu verstehen [ECIN]. Die Präsentation der Waren ist als eine Aufforderung an den Käufer zu verstehen, seinerseits ein Angebot abzugeben. Dies wird durch den Händler angenommen, indem er das Angebot bestätigt.

1.3.1 Reservierungs- und Beherbergungsvertrag

Bei einem Vertrag im Bereich der Unterbringung sind die Vertragspartner das Hotel und der Hotelgast. Dabei wird unterschieden zwischen einem Beherbergungsvertrag (Hotelaufnahmevertrag) und einem Reservierungsvertrag. Der Gast schließt einen Reservierungsvertrag ab, wenn er direkt oder über Dritte und unter den entsprechenden rechtlichen Voraussetzungen ein oder mehrere Zimmer reserviert. Dieser Vertrag ist erfüllt, sobald sich der Gast im Hotel identifiziert und einen Beherbergungsvertrag abschließt. Der Beherbergungsvertrag schließt den gesamten Hotelaufenthalt und alle zu erbringenden Leistungen sowie die Bezahlung durch den Hotelgast mit ein.

Für Angebot und Annahme gelten die gleichen Grundsätze wie einleitend am Beispiel des Online-Warenhandels beschrieben. Der Reservierungsvertrag und der Beherbergungsvertrag kommen durch die Annahme des Antrags des Hotelgastes durch das Hotel zustande. Wurde der Antrag vom Hotel angenommen, so „stehen die Mittel des zugrundeliegenden Rechtssystems zu seiner Durchsetzung zur Verfügung“ [Weiser, S. 7].

Der Beherbergungsvertrag ist ein Vertrag mit Grundelementen aus dem Mietrecht sowie typischen Bestimmungen eines Kauf- oder Dienstvertrages. Die beiden Willenserklärungen, Angebot und Annahme, können mündlich oder schriftlich zustande kommen. Der Vertrag ist verbindlich, sobald die Zimmerreservierung vom Hotel angenommen wurde [DEHOGA]. Das Mietrecht ist im §535 des BGB verankert. Danach hat das Hotel das vereinbarte Hotelzimmer in einem zum vertragsgemäßen Gebrauch geeigneten Zustand zur Verfügung zu stellen und diesen Zustand während der Mietzeit aufrecht zu erhalten. Der Gast hingegen ist zur Entrichtung des vereinbarten Zimmerpreises verpflichtet, nicht jedoch zur Inanspruchnahme des Zimmers.

Wurde ein Beherbergungsvertrag abgeschlossen, so hat der Hotelier die Sicherheit, sich im Falle einer Nichtinanspruchnahme des Zimmers auf die Paragraphen §§ 535–537 BGB zu beziehen. Zum einem hat er bei Nichterscheinen des Gastes einen Erfüllungsanspruch auf den gesamten Übernachtungspreis. Dabei muss sich der Hotelier den Vorteil anrechnen lassen, den er aus einer anderweitigen Vermietung des Zimmers erlangt oder bei Nichtinanspruchnahme des Zimmers einspart (z.B. Reinigungskosten). Andererseits hat er nicht dafür Sorge zu tragen, dass das Zimmer weiter vermietet wird. Die Beweislast liegt hierbei beim Gast, er hat für die Weitervermietung Sorge zu tragen.

1.3.2 Garantierte und nicht garantierte Reservierungen

Bei Reservierungen wird zwischen garantierten und nicht garantierten Reservierungen unterschieden. Im ersten Fall werden neben der Angabe von Stammdaten auch Abrechnungsdaten, wie zum Beispiel Kreditkartendaten gefordert, um die Kreditkarte zu belasten, wenn eine Stornogebühr anfällt. Nicht garantierte

Reservierungen über Hotelbuchungssysteme sind lediglich einfache Anfragen ohne verbindlichen Charakter, bei denen nur einige Stammdaten verlangt werden.

Ein wichtiger Aspekt von eCommerce-Anwendungen ist die Steigerung der Bedienungsfreundlichkeit und Ergonomie, zum Beispiel schnell zum gewünschten Ziel zu gelangen [Ta03]. Bei Hotelbuchungssystemen ist der Vorteil nicht garantierter Reservierungen, dass der Anwender mit so wenig Aktionen wie nötig von der Auswahl bis zur abschließenden Reservierung gelangt. Bei bestehenden Hotelbuchungssystemen, zum Beispiel Hotel Reservation Service [hrs.de], wurde dieser Aspekt wie folgt umgesetzt. Nach der Auswahl des Hotels erhält man durch Angabe einiger Stammdaten eine Bestätigung und eine Reservierungsnummer per E-Mail. Eine Registrierung ist meist nicht notwendig, wodurch der zeitliche Aufwand gering gehalten wird. Das Zimmer ist reserviert und wird abhängig von den Allgemeinen Geschäftsbedingungen (AGB) des Hotels unter einer gegebenen Anreisefrist frei gehalten. Die Anreisefrist unterliegt keiner gesetzlichen Regelung, ist im Hotelwesen im Allgemeinen jedoch bis 18 Uhr gegeben.

1.3.3 Probleme bei der Umsetzung garantierter und nicht garantierter Reservierungen

Besonderes Merkmal nicht garantierter Reservierungen ist, dass von Seiten des Hotels nur auf eine E-Mail-Adresse zurückgegriffen werden kann. Die Angabe der Stammdaten ist keine Garantie dafür, dass der Reservierende der ist, für den er sich ausgibt. Beabsichtigt ein Angreifer – aus naheliegenden Gründen kann hier als Beispiel ein Konkurrent angenommen werden – das Hotel zu schädigen, so kann er ohne größeren Aufwand in kurzer Zeit viele Reservierungen unter falschem Namen vornehmen. Die Zimmer würden bis zum Ablauf der Anreisefrist freigehalten und müssten innerhalb weniger Stunden neu vergeben werden, um einen finanziellen Verlust zu vermeiden.

Für den Fall, dass das Zimmer in der Zeit zwischen der Reservierung und dem Check-In vor Ablauf der festgelegten Frist vergeben wurde, hat der Hotelgast hingegen die Sicherheit, dass er sich beim zuständigen Hotelbuchungssystem auf eine Reservierung beziehen kann. Im Zweifelsfall könnte er eine E-Mail mit der Bestätigung der erfolgten Reservierung vorweisen, um den Reservierungsvorgang

nachzuweisen. Beide Möglichkeiten bieten jedoch keine Sicherheit, wenn es zum Streitfall kommt. Schadensersatzforderungen können nicht geltend gemacht werden, da die Reservierung keinen verbindlichen Charakter hatte. Durch einen solchen Vorfall ist es aber denkbar, dass der Ruf des Hotels durch den Reservierenden im Nachhinein erheblich geschädigt wird.

Sieht man von der nicht vorhandenen Rechtssicherheit für den Reservierenden ab, so ist dieser klar im Vorteil. Um den Gast zu einer Reservierung im Hotel zu bewegen, wird in einigen Fällen sogar gänzlich auf eine Stornogebühr verzichtet. Anderseits wird auf Seiten des Hotels auch auf drastischere Methoden zurückgegriffen, um einer eventuell absichtlich falschen Reservierung vorzubeugen. So ist zum Beispiel unter http://www.kempinski.com/de/disclaimer zu lesen: „Sofern ein kostenfreies Rücktrittsrecht des Kunden innerhalb einer bestimmten Frist besteht, ist das Hotel seinerseits berechtigt, vom Vertrag zurückzutreten, wenn Anfragen anderer Kunden nach den vertraglich gebuchten Zimmern vorliegen und der Kunde auf Rückfrage des Hotels auf sein Recht zum Rücktritt nicht verzichtet“. Der Hotelgast unterliegt daher dem Zwang, eine garantierte Reservierung vorzunehmen.

Garantierte Reservierungen enthalten neben der Angabe von Stammdaten auch die Angabe abrechnungsrelevanter Informationen. Diese können verwendet werden, wenn eine Stornogebühr anfällt. Der Vorteil dieser Reservierung besteht darin, dass, unabhängig von der Inanspruchnahme des Zimmers, dieses freigehalten wird. Garantierte Reservierungen werden unter WorldCheckInn umgesetzt.

Handelt es sich um einen neuen Hotelgast und kann dieser seine Identität nicht ausreichend nachweisen, so garantiert erst der Beherbergungsvertrag die Richtigkeit der angegebenen Abrechnungsdaten. Der Beherbergungsvertrag wird im Hotel abgeschlossen, nachdem sich der eintreffende Hotelgast ausgewiesen hat. Kann die Identität des Hotelgastes bei einer garantierten Reservierung nicht sichergestellt werden, so können die Probleme nicht garantierter Reservierungen auftreten.

2. WorldCheckInn

WorldCheckInn ist ein Service für Vielreisende. Diese hinterlegen in einem vertraulichen Verzeichnis ihre Melde- und Abrechnungsdaten und autorisieren bei der Inanspruchnahme einer Dienstleistung (Check-In im Hotel, Flugticketbuchung, Autovermietung, ...) den Dienstleistungsanbieter zum einmaligen Zugriff auf diese Daten. Neben Melde- oder Abrechungsdaten enthält das Verzeichnis auch Zusatzinformationen, wie zum Beispiel persönliche Wünsche des Vielreisenden.

2.1 Zielgruppen

Die erste Zielgruppe umfasst Vielreisende, die auf Geschäftsreisen unterwegs sind. Diese legen zumeist Wert auf eine schnelle Abwicklung einer Reservierung über das Internet, sowie dem Check-In und Check-Out im Hotel. Die zweite Zielgruppe, umfasst Dienstleistungsunternehmen, die im Bereich der Unterbringung tätig sind. Angedacht ist es, im Laufe der Weiterentwicklung von WorldCheckInn auch andere Dienstleistungen anzubieten, welche durch die erste Zielgruppe auf Reisen in Anspruch genommen werden können, zum Beispiel Autovermietung oder Reiseticketbuchung.

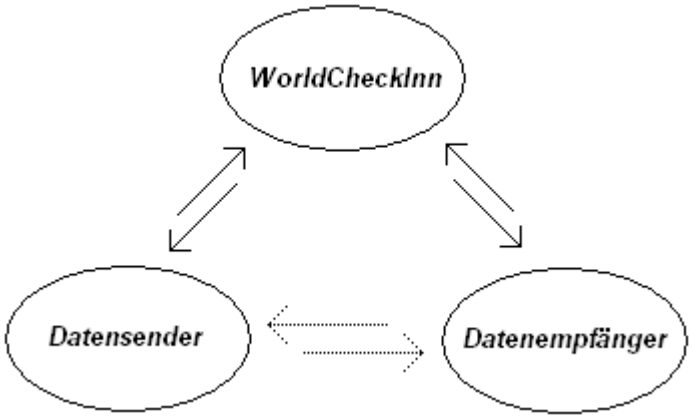

Abbildung 2.1.1: WorldCheckInn bildet die Schnittstelle zwischen den beiden Zielgruppen, Dienstleistungsanbieter (Datenempfänger) und Vielreisender (Datensender)

Als dritte Zielgruppe werden Unternehmen angesehen, die an einer statistischen Auswertung der Daten interessiert sind (Data-Mining Konzepte).

2.2 Übersicht und Funktionalität

WorldCheckInn stellt einen Reservierungsservice (www.worldcheckinn.com) für Hotels und Vielreisende bereit. Hotels nutzen diesen Dienst zur Abbildung ihrer Hotelinformationen und zum Anbieten von Sonderangeboten. Vielreisende nutzen diesen Dienst, um Zimmerreservierungen durchzuführen. Im Folgenden werden die Dienste des Systems, die Registrierung neuer Datensender und -empfänger, Reservierungen und der medienbruchfreie Check-In im Hotel erklärt. Datenempfänger sind Dienstleistungsunternehmen, die aufgrund von Abrechnungszwecken Interesse an den Stamm- und Abrechnungsdaten der Kunden haben, wenn diese die Dienste des Unternehmens in Anspruch nehmen. Datensender hingegen sind Kunden, welche die Dienstleistungen dieser Unternehmen in Anspruch nehmen und dafür diesem ihre Stamm- und Abrechnungsdaten zur Verfügung stellen.

2.2.1 Registrierung neuer Datensender und -empfänger

Hinter der Oberfläche www.worldcheckinn.com verbirgt sich ein Verzeichnisdienst, in dem Sender und Empfänger Daten hinterlegen können. Die Registrierung eines neuen Datensenders umfasst die Speicherung persönlicher Daten (Firmen- und Stammdaten), abrechnungsrelevante Informationen, welche Sonderleistungen in Anspruch genommen werden und Zahlungsmodalitäten.

Abrechnungsrelevante Daten können über WorldCheckInn gepflegt werden. Es kann festgelegt werden, wer in Anspruch genommene Dienstleistungen bezahlt (Firma/Privat), wie bezahlt wird (Kreditkarte, Lastschrift, …) und wie die Rechnung hinterlegt werden soll (Aufbewahrung, Übermittlung an Firma, …). Diese und die oben genannten Informationen werden dem Hotel für die Dauer des Aufenthaltes zur Verfügung gestellt. Abbildung 2.2.1 ist ein Beispiel für die Erhebung abrechnungsrelevanter Daten.

Abbildung 2.2.1: Registrierung eines neuen Datensenders – Angabe abrechnungsrelevanter Daten

Die Registrierung neuer Datenempfänger umfasst Daten zur Ausstattung und Lage sowie Preisinformationen. In Abbildung 2.2.2 wird als Beispiel die Erfassung der Logispreise eines Hotels gezeigt.

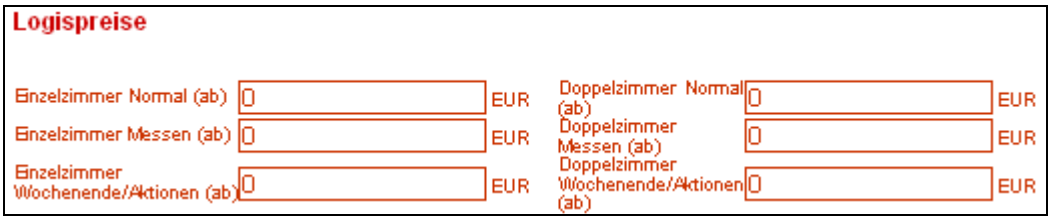

Abbildung 2.2.2: Registrierung eines Datenempfängers – Angabe der Logispreise

Nach Abschluss der Registrierung erhalten Datensender und -empfänger Zugangsdaten und eine Kundennummer.

2.2.2 Reservierungen

Eine Reservierung, beginnend bei der Suche eines Hotels bis zur Auswahl, erfolgt analog zu bereits bestehenden Hotelreservierungssystemen (vgl.: hotel.de, weg.de). Auf Basis gespeicherter Hotelinformationen und umfassender Such- und Vergleichsmöglichkeiten erhält der Datensender eine Auswahl an Reservierungsmöglichkeiten.

Anschließend bekommt der Datensender die Möglichkeit, die Reservierung sofort durchzuführen und abzuschließen. Um dies zu ermöglichen, wird vorab die Zimmerverfügbarkeit des Hotels ermittelt. Anhand der Reservierungsdaten des Hotelgastes, die aus dem Datum der An- und Abreise, Art und Anzahl der Zimmer und die Anzahl der Personen bestehen, wird bestimmt, ob eine Reservierung durchgeführt werden kann. Kann die Reservierung durchgeführt werden, so wird sie in das Hotelmanagementsystem eingetragen und der Datensender erhält eine Bestätigung.

2.2.3 Check-In

Bei einem Check-in über WorldCheckInn erhält das Hotel für die Dauer des Aufenthaltes die freigegebenen Daten des Hotelgastes/Datensenders. Um den Erhalt der Daten und somit den Check-In durchzuführen, wird der Hotelgast aufgefordert, sich per Kunden- oder Kreditkarte an einem Kartenterminal zu authentifizieren.

Am Beispiel WorldCheckInn wird ein Prototyp entwickelt, der zeigt, wie eine solche Authentifikation bei einem Check-In Vorgang im Hotel abläuft: Der Datensender (Hotelgast) authentifiziert sich unter Verwendung eines vorhandenen Kartenlesegerätes per Magnet- oder Chipkarte und der dazugehörigen PIN bei dem Provider des Terminals gegenüber WorldCheckInn. Bei einer erfolgreichen Authentifizierung sendet WorldCheckInn die Daten des authentifizierten Hotelgastes an den Datenempfänger (das Hotel), in welchem die Authentifizierung veranlasst wurde. Der Authentifizierungsvorgang autorisiert also das Hotel, persönliche Daten, insbesondere Abrechnungsdaten vom Datensender (Hotelgast) zu beziehen.

Auf der beiliegenden CD befindet sich im Unterverzeichnis *pflichtenhefte/* das Dokument *schnittstellen_tprovider.pdf*. Dieses Dokument beinhaltet detaillierte Angaben zum Aufbau der Schnittstellen zwischen WorldCheckInn und dem Terminal-Provider, welcher die Authentifizierung des Hotelgastes am Terminal übernimmt.

2.3 Einordnung des Geschäftsmodells

Durch die Zielgruppen und die genannte Funktionalität beschränkt sich WorldCheckInn auf wirklich getätigte Transaktionen. Nimmt ein Hotelgast eine Dienstleistung über WorldCheckInn in Anspruch, so findet eine Transaktion statt. Bei einer Transaktion werden die Daten des Datensenders zum Datenempfänger übermittelt.

Für eCommerce-Anwendungen liegen verschiedene Geschäftsmodelle zugrunde. [Wirtz] teilt diese Modelle von eCommerce-Anwendungen in klassische Modelle ein:

- Marktmodell
- Beschaffungsmodell
- Leistungserstellungsmodell
- Leistungsangebotsmodell
- Distributionsmodell
- Kapitalmodell

Von Bedeutung ist das Kapitalmodell. Dieses Modell beinhaltet das Finanzierungs- und Erlösmodell. Das Erlösmodell beschreibt, wer die angebotenen Dienstleistungen wie finanzieren soll. [Wirtz] unterscheidet zwischen direkter und indirekter sowie transaktionsabhängiger und transaktionsunabhängiger Erlösgenerierung. In Tabelle 2.1 sind diese noch einmal mit Beispielen zusammengefasst.

	Direkte Erlösgenerierung	Indirekte Erlösgenerierung
Transaktionsabhängig	*Transaktionserlöse* *Verbindungsgebühren* *Nutzungsgebühren*	*Provisionen*
Transaktionsunabhängig	*Einrichtungsgebühren* *Grundgebühren*	*Bannerwerbung* *Data-Mining Erlöse* *Sponsorship*

Tabelle 2.3.1: Das Erlösmodell

Durch Transaktionen ergeben sich Transaktionserlöse und Provisionen. Indirekt steigen somit auch die Erlöse aus *Data-Mining*, denn Transaktionen ergeben eine größere Ansammlung von Informationen, die zu diesem Zwecke genutzt werden können. Transaktionen bilden die Grundlage der Umsatzgenerierung unter WorldCheckInn; das Ziel ist die fortlaufende Steigerung der Anzahl dieser Transaktionen.

2.4 Zielsetzung des Systems

Alle Dienste sollen automatisiert, medienbruchfrei und rechtssicher unter Einhaltung der Datenschutzbestimmung durchgeführt werden. Im Detail bedeutet dies:

- Die Vermeidung von Medienbrüchen durch Übertragung der Reservierung und Hotelgastdaten in das Zielsystem, in der Regel ist dies das Hotelmanagementsystem
- Schaffung einer einheitlichen Plattform für sicheres Schließen von Verträgen im Internet
- Beachtung der Datenschutzbestimmungen bei der Erhebung, Verarbeitung und Weitergabe personenbezogener Daten
- Anwendergesteuerte Rechteverwaltung: Der Hotelgast/Datensender bestimmt, wie seine Daten verarbeitet werden. Er legt fest, welche Daten für welchen Zweck verwendet oder weitergegeben werden dürfen

Die damit verbundenen Ziele sind:

- Zeitersparnis und Vereinfachung des Check-In: Der Hotelgast hinterlegt vorab persönliche Daten und Wünsche, die bei einem Check-In an das betreffende Hotel übermittelt werden. Diese Form der Registrierung gibt dem Hotelgast die Möglichkeit, vorab seine Zimmerwünsche zu äußern, so dass diese vom Personal des Hotels vor Anreise des Hotelgastes erfüllt werden können
- Sicherheit für Hotel und Hotelgast durch rechtsverbindliche Vertragsschlüsse und akkreditierte Personen- und Abrechungsdaten
- Die Erhebung, Verarbeitung und Weitergabe akkreditierter personen-bezogener Daten erfolgt ohne Konflikte mit dem Datenschutzrecht

Ziel des Unternehmens ist es jedoch auch, dass das System WorldCheckInn lediglich als Vermittler zwischen Hotel und Hotelgast dient. Der Reservierungsvertrag soll rechtssicher und verbindlich sein und nur zwischen dem Hotel und Hotelgast zustande kommen. Auch alle sich aus dem Vertrag ergebenden Verpflichtungen bestehen unmittelbar und ausschließlich zwischen Hotelgast und Hotel.

3. Rechtliche Voraussetzungen zur Umsetzung des Systems WorldCheckInn

Das Abschließen elektronischer Verträge im Internet unterliegt besonderen Voraussetzungen und Bedingungen. Während sich im Warenhandel die beiden Vertragsparteien gegenüber stehen, kann bei elektronischen Verträgen mangels standardisierter und bewährter Sicherheitsmechanismen nur unzureichend nachgewiesen werden, ob hinter einer vorgegebenen Identität auch eine ladungsfähige Adresse besteht. Dieses Kapitel widmet sich diesem Thema und soll Gemeinsamkeiten, Unterschiede und Probleme im Vergleich zu den ‚klassischen' Vertragsabschlüssen aufzeigen. Eine Abhilfe für die aufgezeigten Probleme könnte das Modell nach Wabner schaffen, ein Schuldvertragsmodell für den elektronischen Handel [Wa03].

3.1 Einhaltung von Datenschutzbestimmungen im Umgang mit personenbezogenen Daten

Bei der Registrierung neuer Datensender werden in einem Verzeichnisdienst Informationen über diese hinterlegt. Dabei handelt es sich um Stammdaten, Daten zu Abrechungszwecken und weitere Informationen, wie zum Beispiel persönliche Wünsche des Datensenders zur Beanspruchung besonderer Serviceleistungen. Darüber hinaus wird jeder Geschäftsvorgang der Datensender gespeichert, so dass mit steigender Nutzung des Systems umfangreiche Kundenprofile entstehen.

Kernpunkte des Systems sind die Erhebung und Speicherung von Daten, die Verarbeitung personenbezogener und nicht-personenbezogener Daten sowie die Weitergabe dieser Daten. Der Begriff „personenbezogene Daten" ist in §3 des BDSG definiert. So sind personenbezogene Daten „Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person

(Betroffener).“ (§3, Absatz 1 BDSG) Diese werden weiter konkretisiert (§3, Absatz 9 BDSG): Als besonders schutzwürdige Daten gelten Daten natürlicher Personen über ihre rassische und ethnische Herkunft, politische Meinung, Gewerkschafts-zugehörigkeit, religiöse oder philosophische Überzeugung, Gesundheit oder ihr Sexualleben.

Als besonderer Schwerpunkt ist das *Data-Mining* akkreditierter Kundendaten anzusehen, mit dessen Hilfe Profile bestimmter Personen oder Gruppen erstellt werden können. Unter *Data-Mining* versteht man die Auswertung einer Ansammlung von Informationen, personenbezogen und nicht-personenbezogen (anonymisiert oder durch Verwendung eines Pseudonyms). Dabei können zum Beispiel Änderungen im Kundenverhalten entdeckt und darauf aufbauend Geschäftsstrategien entwickelt werden.

Dagegen spricht Paragraph 3a im Bundesdatenschutzgesetz (BDSG), der Aussagen zur Datenvermeidung und Datensparsamkeit trifft: „Gestaltung und Auswahl von Datenverarbeitungssystemen haben sich an dem Ziel auszurichten, keine oder so wenig personenbezogene Daten wie möglich zu erheben, zu verarbeiten oder zu nutzen. Insbesondere ist von den Möglichkeiten der Anonymisierung und Pseudonymisierung Gebrauch zu machen, soweit dies möglich ist und der Aufwand in einem angemessenen Verhältnis zu dem angestrebten Schutzzweck steht“. Das Bundesdatenschutzgesetz [BDSG] trifft Regelungen für den Umgang mit personenbezogenen Daten, insbesondere auch für den Verwendungszweck dieser Daten.

Die Erhebung der Daten muss rechtmäßig sein. Dies kann aufgrund gesetzlicher Regelungen, wie zum Beispiel die Erfassung von Meldedaten in Beherbergungs-unternehmen, oder mit einer Einwilligung der betroffenen Person geschehen (§4, Absatz 1 BDSG). Die Einwilligung darf nicht erzwungen werden (§4a, Absatz 1 BDSG). Des Weiteren hat der Datensender das Recht auf Auskunft, für welchen Zweck die Daten erhoben, verwendet und an wen sie übermittelt werden (§4a, Absatz 3 und §34, BDSG). Die Daten müssen nach dem Verwendungszweck wieder gelöscht werden (§35, Absatz 2 BDSG).

§§ 27–38 des BDSG treffen Regelungen zum Umgang mit Personendaten im nichtöffentlichen Bereich. Demnach bedarf nicht nur die Erhebung, sondern auch die Verarbeitung und Weitergabe personenbezogener Daten einer nicht erzwungenen

Einwilligung des Betroffenen. Über diesen Umstand ist der Betroffene bereits bei der Erhebung seiner Daten zu unterrichten. Anderseits bedarf es einer zusätzlichen Einwilligung.

Für das Verarbeiten von Daten werden in §3 BDSG die Begriffe Anonymisieren (Absatz 6) und Pseudonymisieren (Absatz 6a) eingeführt. Demnach ist das Anonymisieren von Personendaten eine Veränderung, die nicht mehr oder nur durch unverhältnismäßig großen Aufwand wieder den Rückschluss auf eine natürliche Person zulässt. Pseudonymisieren ersetzt ein Identifikationsmerkmal des Betroffenen durch ein anderes, so dass dieser nicht mehr bestimmt werden kann. Pseudonymisierte Nutzerprofile dürfen nicht mit Daten des Betroffenen zusammengeführt werden. Besteht aber die Möglichkeit, dass der Betroffene im Nachhinein durch automatisierte Datenverarbeitung anhand bestimmter Merkmale wieder identifiziert werden kann, so muss dieser nach §4 des Teledaten-schutzgesetzes (TDDSG) darüber unterrichtet werden.

Das BDSG trifft zudem Regelungen in §4b für die Weitergabe von (Adress-) Daten in Ländern der Europäischen Union (EU) und Ländern, die nicht der EU angehören. Dies betrifft zum Beispiel die Weitergabe von Adressdaten: Beim Check-in im Hotel wird das Hotel unter anderem autorisiert, Adressdaten des Hotelgastes über WorldCheckInn zu beziehen. So unterliegt die Weitergabe von Daten ins Ausland keiner besonderen Genehmigung, wenn es sich um ein Mitglied der Europäischen Union (EU) handelt und der Betroffene seine Zustimmung zur Weitergabe gegeben hat (Zweck der Datenerhebung). Die Übermittlung an Nicht-EU-Staaten ist unzulässig, wenn ein angemessenes Datenschutzniveau nicht gewährleistet ist. Die übermittelnde Stelle trägt die Verantwortung für die Zulässigkeit der Übermittlung. Bei Nicht-EU-Staaten muss vorab mit dem Empfänger (vertraglich) geklärt werden, ob er alle Regelungen für ein angemessenes Datenschutzniveau nach $4b, Absatz 2 und 3 BDSG einhält. Diese übermittelten Daten dürfen ausschließlich nur für den genehmigten Zweck – in diesem Falle zur Durchführung des Check-In – verwendet werden (§4b, Absatz 6 BDSG).

3.2 Meldedatenerfassung und -weitergabe

Die gesetzlichen Regelungen für die Weitergabe von Personendaten an Dritte wurden bereits im vorangegangenen Abschnitt aufgeführt. Bei der Inanspruchnahme einer Dienstleistung, zum Beispiel dem Check-In-Vorgang, wird der Datenempfänger autorisiert, die vom Datensender hinterlegten Daten zu beziehen. Dies umfasst auch die Meldedaten des Datensenders. Üblicherweise wird der Hotelgast bei Anreise im Hotel dazu aufgefordert, seiner Meldepflicht nachzukommen. Dazu muss er einen Hotelmeldeschein ausfüllen und unterschreiben. Der Hotelmeldeschein beinhaltet Fragen zur Person (Familiennamen, frühere Familiennamen, Rufname, Tag und Ort der Geburt, Anschrift, Staatsangehörigkeit) und Tag der Ankunft (§19, Absatz 2 Sächsisches Meldegesetz). Die Registrierung neuer Datensender unter WorldCheckInn umfasst auch die Aufnahme der Meldedaten, wie sie für den Check-In benötigt werden.

Der Hotelier ist verpflichtet, den Hotelgast auf seine Meldepflicht hinzuweisen. Der Hotelgast ist dazu verpflichtet, den Meldezettel handschriftlich auszufüllen (§18, Absatz 2 SächsMG) und mit seiner Unterschrift bestätigen. Der Hotelier ist verpflichtet, die handschriftlichen Meldezettel über einen Zeitraum von zwölf Monaten aufzubewahren.

3.2.1 Identitätsnachweise

Das Meldegesetz ist Ländersache. So ist zum Beispiel im Hessischen Meldegesetz [HMG] zu lesen, dass sich die Ausweispflicht nur auf beherbergte AusländerInnen bezieht. Im Sächsischen Meldegesetz hingehen wurde im Zuge der Gleichbehandlung von Ausländern und Nicht-Ausländern die Ausweispflicht allen Gästen auferlegt (§18 „Pflichten des Meldepflichtigen“ SächsMG).

Es besteht aber keine Pflicht, dass sich der Gast gegenüber dem Hotelier ausweisen muss, da der Hotelier kein Anrecht darauf hat, die Identität des Gastes zu erfahren [RIT]. Allerdings muss die Verweigerung nach §19 SächsMG des Gastes im Hotelmeldeschein vermerkt werden.

3.2.2 Erfüllung der Meldepflicht durch WorldCheckInn

Die Weitergabe personenbezogener Daten an Dritte, zum Beispiel zur Erfüllung der Meldepflicht, bedarf einer Zustimmung des Betroffenen (vgl. Abschnitt 3.1). Die zu übermittelnden Daten unterliegen in Sachsen einerseits den Bestimmungen des Sächsischen Datenschutzgesetzes (SächsDSG) zum Schutz der persönlichen Daten des Betroffenen bei deren Weiterleitung an Dritte. Anderseits unterliegen diese auch dem Meldegesetz. So ist im Sächsischen Meldegesetz [SächsMG] verankert, dass diese auf Grund der schutzwürdigen Interessen des Betroffenen (§22 SächsMG) vor unbefugter Einsichtnahme zu sichern sind (§19, Absatz 3 SächsMG). Allerdings ist die automatisierte Datenverarbeitung der Meldedaten nur zwischen Beherbergungsstätten und Behörden beziehungsweise nur zwischen Behörde und Behörde geregelt. (Siehe §28–34 SächsMG).

Für die gesetzte Forderung, dass WorldCheckInn lediglich als Vermittler dient und keine Haftung für die Richtigkeit der Meldedaten übernimmt, können an dieser Stelle keine grundlegenden Aussagen getroffen werden. Vielmehr wird dies durch §18 SächsMG verhindert, in dem gefordert wird, dass der Meldezettel handschriftlich ausgefüllt werden muss.

„Never fill up forms again“ ?

Der Einsatz qualifizierter elektronischer Signaturen bei der Übermittlung von Meldedaten verliert bereits dadurch an Bedeutung, dass in §18 Absatz 2 SächsMG die Handschrift des Gastes im Hotelmeldeschein verlangt wird. Erschwerend kommt die Erkenntnis in Abschnitt 3.2.1 hinzu, dass keine Überprüfung der Meldedaten des Hotelgastes vorausgesetzt oder gefordert werden kann. Dadurch kann die gestellte gesetzte Forderung zur Überprüfung der Hotelgastdaten nicht durchgesetzt werden. Dies muss im Rahmen der Abwicklung des Check-In über WorldCheckInn als Voraussetzung erfüllt und umgesetzt werden. Die Umsetzung des Slogans „Never fill up forms again“ scheitert jedoch bereits an der Forderung der Handschrift des Hotelgastes.

„Die Pflicht zur handschriftlichen Ausfüllung entfällt nur, wenn der Gast wegen einer körperlichen Behinderung oder aus anderen Gründen (z.B. Analphabet) dazu nicht in der Lage ist“ (§18 Pflichten des Meldepflichtigen SächsMG).

3.3 Treuhandservices als vertrauenswürdige Vermittler

Abhilfe für die in Abschnitt 1.3 gezeigten Probleme bezüglich der garantierten und nicht-garantierten Reservierungen könnte eine Reiseversicherung, wie zum Beispiel www.elvia.de schaffen. Diese garantiert, dass das Hotel bei einer kurzfristig abgesagten Reservierung eine Stornogebühr erhält, welche jedoch nicht durch den Gast bezahlt werden muss. Dieser zahlt den Beitrag der Reiseversicherung, welche die Stornogebühren übernimmt. Das Hotel hat somit weniger Umsatzeinbußen. Wenn diese Versicherung bei der Reservierung zwingend ist, um den Reservierungsvorgang abzuschließen, ist eine zusätzliche Sicherheit gewährt.

Negativ anzumerken ist, dass diese Versicherung und die damit verbundenen zusätzlichen Reisekosten dem Gast attraktiv gemacht werden müssen. Darüber hinaus übernimmt der Versicherer nur Stornogebühren, welche durch schwerwiegende Probleme begründet sind (Krankheit, Todesfall in der Familie, o.ä.).

Treuhandservices stellen im elektronischen Handel einen Dritten dar, der die Erfüllung des Geschäftsvorganges überwacht und teilweise auch übernimmt. Im Allgemeinen erfolgt dies durch die Angabe von Stammdaten des Käufers und Verkäufers sowie einer Vorrauszahlung des Käufers auf ein durch den Treuhandservice bereitgestelltes Konto.

Das Unternehmen iloxx [iloxx.de] bietet einen Treuhandservice („SAFETRADE“) an, welcher die Abwicklung des gesamten Geschäftsvorganges, von der Zahlung bis zur Lieferung und den notwendigen Empfangsbestätigungen überwacht. Der Käufer überweist das Zahlungsmittel auf ein durch iloxx bereitgestelltes Konto (Sicherung des Zahlungseinganges beim Verkäufer). Anschließend wird dem Verkäufer der Eingang des Zahlungsmittels mitgeteilt, so dass dieser mit der Lieferung beginnen kann. Nach Empfang der Ware beim Käufer wird das Zahlungsmittel an den Verkäufer überwiesen. Um zu vermeiden, dass der Käufer dem ihm zur Verfügung gestellten Zeitraum zur Überprüfung der Ware nicht ausnutzt, trägt dieser die Kosten für die Inanspruchnahme des Treuhandservices.

Die Schuldtilgungsmittel, die Ware des Verkäufers und das Zahlungsmittel des Käufers, werden also auf beiden Seiten solange zurückgehalten, bis die Garantie gegeben ist, dass beide Seiten erfüllen können. Erhält der Verkäufer die Ware nicht, so kommt es auch nicht zur Überweisung des Zahlungsmittels. Auf der Seite des Käufers wird der Empfang der Ware mit einer Unterschrift bestätigt. Dieser hat zusätzlich beim Empfang der Ware die Möglichkeit, diese zu überprüfen, bevor er den Empfang bestätigt (Schutz vor Falschlieferung). Hinzu kommt, das iloxx als Treuhandservice die Zustellung und Abholung der Ware überwacht. Besteht die Gefahr einer Rückbuchung durch den Käufer, würde diese das von iloxx bereitgestellte Konto belasten. Das Konto des Verkäufers hingegen bleibt davon unberührt.

Iloxx erhöht somit die Vertrauenswürdigkeit der Vertragsparteien untereinander, aber nicht gegenüber dem Treuhandservice. Iloxx ist auch ein auf Gewinn ausgerichtetes Unternehmen „dem man [...] nicht vertrauen möchte“ [Weiser, S. 13].

Im Gegensatz zur Vorkasse hat der Verkäufer bei Treuhandservices den Nachteil, dass die Zahlung bei ihm erst nach dem Versand eingeht, wenn der Käufer die Freigabe erklärt hat. Verweigert der Käufer die Zahlung oder kommt es zu einer Rückbuchung, so hat der Verkäufer, zum Beispiel aufgrund entstandener Lieferkosten, das Nachsehen. Der Kreditkarteninhaber kann die Buchung zurückbuchen lassen, wenn er behauptet, die Zahlung nicht veranlasst zu haben. Kann der Zahlungsempfänger nachweisen, dass der Karteninhaber selbst gezahlt hat, etwa durch die Lieferadresse, dann bleibt die Buchung bestehen.

Käufer, beziehungsweise Verkäufer, haben keine Garantie dafür, dass hinter der vorgegebenen Identität des Anderen eine ladungsfähige Adresse besteht. Der Treuhandservice soll zwischen diesen untereinander unbekannten Parteien eine Sicherheit geben. Wiederum kann dieser selbst vom Käufer oder Verkäufer gefälscht sein. So hatte etwa die Firma Escrow (www.escrow.com) bereits mit gefälschten Treuhandservices zu tun, welche ihren Namen missbrauchten und gefälschte Webseiten veröffentlichen, wie zum Beispiel www.escrow-commerce.de und www.escrow-team.de [ebay.de]. Daraufhin wurde das Projekt www.sos4auctions.com ins Leben gerufen, um gefälschte Webseiten schneller bekannt zu machen.

3.4 Das Modell nach Wabner

Das Modell nach Wabner [Wa03] beschreibt ein Schuldvertragsmodell für den elektronischen Handel zum Abschließen und Erfüllen von Verträgen. Dieses Modell setzt einen unabstreitbaren, rechtssicheren Vertragsabschluss um. Käufer und Verkäufer schließen und erfüllen die Verträge nicht untereinander, sondern über zwei Schuldvertragszentren.

Das Modell nach Wabner fordert, neben dem Vertragsabschluss, die Überwachung und Dokumentation der Erfüllung mit anschließender Empfangsbestätigung durch beide Parteien. In diesem Zusammenhang wird erläutert, wo sich Erfüllung und Empfangsbestätigungen bei Dienstleistungen wiederfinden. Diese können, müssen aber nicht Bestandteil für die Inanspruchnahme von Dienstleistungen über das Internet sein. Sie erweisen sich jedoch in einigen Punkten als sinnvoll.

3.4.1 Eigenschaften des Modells

Im Folgenden werden die Eigenschaften des Modells vorgestellt. Eine ausführliche Beschreibung befindet sich in [Wa03].

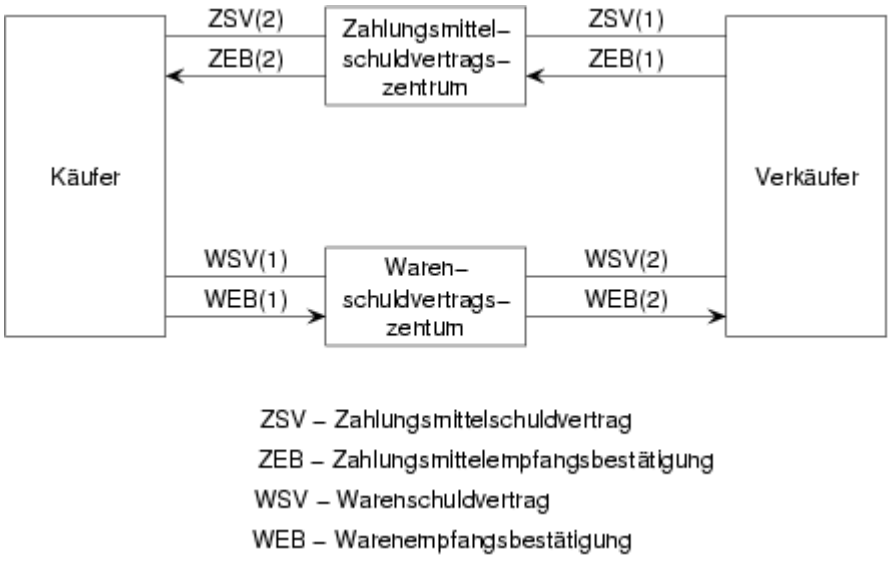

Abbildung 3.4.1: Das Modell nach Wabner

Einführung von Schuldvertragszentren: Bestandteile des in Abbildung 3.4.1 dargestellten Modells sind Käufer und Verkäufer sowie zwei vertrauenswürdige Schuldvertragszentren, das Zahlungsmittelschuldvertragszentrum (ZSVZ) und das Warenschuldvertragszentrum (WSVZ). Diese Schuldvertragszentren stellen dabei staatlich überwachte Instanzen dar, welche die Erfüllung der Schuldverträge überwachen und dokumentieren. Käufer und Verkäufer schließen den Vertrag über diese beiden Instanzen ab.

Teilung der beiden Schuldverträge in jeweils zwei weitere Verträge:
Die beiden Schuldverträge des Modells sind der Zahlungsmittelschuldvertrag und der Warenschuldvertrag. Der Zahlungsmittelschuldvertrag verpflichtet den Käufer zur Zahlung des Kaufpreises und Annahme der Ware. Der Warenschuldvertrag verpflichtet den Verkäufer zur Lieferung der Ware und zur Verschaffung des Eigentums an den Käufer. Diese beiden Schuldverträge werden in jeweils zwei Teilschuldverträge aufgespalten, die jeweils mit einem Schuldvertragszentrum geschlossen werden. Alle Teilschuldverträge werden durch eine Geschäfts-identifikationsnummer (s.u.) einem Gesamtvertrag zugeordnet.

Einführung einer eindeutigen Geschäftsidentifikationsnummer: Die Geschäfts-identifikationsnummer (GID) [Schilha] ist ein gemeinsames Identifikationsmerkmal der Teilschuldverträge. Anhand dieser können nach Abschluss des Gesamtvertrages alle vier Teilschuldverträge zugeordnet werden. [Schilha] beschreibt in seiner Bachelor-Arbeit die Notwendigkeit der Eindeutigkeit der Geschäftsidentifikations-nummer zur Kollisionsvermeidung.

Wahrung der Anonymität: Die Schuldvertragszentren können die Teilschuld-verträge anhand der GID dem Gesamtvertrag zuordnen. Zusatz-informationen, wie zum Beispiel die Lieferadresse sind nur zur Erfüllung des Teilschuldvertrages zwischen Käufer und Schuldvertragszentrum notwendig. Über ein Schuldvertrags-zentrum hinaus werden demnach keine sensiblen Informationen weitergegeben, daher ist eine Anonymität der beiden Vertragsparteien gegeben.

Der Vertragsabschluss ist rechtssicher, wenn zu den vorab ausgehandelten Vertrags-parametern beiderseitige Willenserklärungen in Form einer Unterschrift vorliegen. Im Warenhandel ist der Käufer zur Zahlung des Kaufpreises und Annahme der Ware verpflichtet, der Verkäufer zur Übergabe der Ware und Verschaffung des Eigentums an den Käufer. Des Weiteren wird gefordert, dass das Gesamtsystem nicht

korrumpierbar ist. Wabner schlägt daher vor, das ZSVZ und WSVZ staatlich überwachte Instanzen sind, welche den gesamten Vertragsabschluss sowie seine Erfüllung überwachen und dokumentieren.

Ist das Modell nach Wabner rechtssicher, so ist auch jedes darauf basierende Protokoll rechtssicher [Weiser, S. 20]. In Abschnitt 3.4.3 wird näher darauf eingegangen und geklärt, ob lediglich der Vertragsabschluss bei der Inanspruchnahme von Dienstleistungen über das Internet stattfinden kann oder sogar die Erfüllung des Vertrages und die Ausstellung der Empfangsbestätigungen möglich sind.

Durch [Wabner] und [Weiser] wurde die Vertrauenssymmetrie eingeführt. Die Vertrauenssymmetrie ist im Modell nach Wabner gewährleistet, wenn keiner der Vertragspartner seinem Gegenüber mehr Vertrauen entgegenbringen muss, als dieser ihm. Unter bestimmten Umständen wird es dennoch gefordert und als angebracht angesehen, einen Vertrauensvorschuss zu erbringen, insbesondere wenn es sich um einen unbekannten beziehungsweise nicht vertrauenswürdigen Käufer handelt und die Erfüllung des Vertrages nicht gewährleistet werden kann. Da Dienstleistungen nicht rückgängig gemacht werden können, ist ein Vertrauensvorschuss durch den Verkäufer nicht möglich. Kann die Vertrauenswürdigkeit des Käufers nicht gewährleistet werden, so muss dieser einen Vertrauensvorschuss erbringen, um für eventuell entstehende Kosten bei Nichtabnahme oder Nichtinanspruchnahme der Dienstleistung aufzukommen. Damit entsteht ein Bruch in der Vertrauenssymmetrie. Wünschenswert ist es also, einen abgeschlossenen, rechtssicheren Vertrag vorliegen zu haben, so dass auf beiden Seiten mit der Erfüllung begonnen werden kann.

3.4.2 Protokollvarianten und Rückholschritte

[Schilha] beschreibt in seiner Bachelor-Arbeit drei abstrakte Protokollvarianten für eine Umsetzung des Modells nach Wabner. Diese enthalten unter anderem sogenannte Rückholschritte. Rückholschritte sind das Rückgängigmachen der Teilschuldverträge, wenn der Gesamtvertrag scheitert. Die Kosten für die Rückholschritte werden dem Verursacher in Rechnung gestellt. Ist dies nicht möglich, da der Verursacher nicht eindeutig identifiziert werden kann, so müssen besondere Regelungen getroffen werden.

Die drei folgenden Protokollvarianten sind mit Bezug auf den elektronischen Warenhandel entstanden; das heißt, der Abschluss eines Vertrages erfolgt auf elektronischem Wege. Anschließend findet ein Warenversand in elektronischer Form (zum Beispiel als eBook) oder per Post statt.

- **Das asynchrone Protokoll:** Dieses Protokoll ermöglicht die sofortige Lieferung der Ware, allerdings sind zwei Rückholschritte notwendig, wenn der Gesamtvertrag scheitert.
- **Das synchrone Protokoll:** Es ist zwingend, den Vertragsabschluss fertig zustellen, bevor mit der Erfüllung beider Seiten begonnen werden kann. Daher sind keine Rückholschritte notwendig.
- **Das hybride Protokoll:** Die Schuldtilgungsmittel werden in den Schuldvertragszentren festgehalten. Scheitert der Gesamtvertrag, so ist ein Rückholschritt notwendig.

Im weiteren Verlauf wird deutlich, welche Protokollvarianten sich für eine Umsetzung unter WorldCheckInn eignen. Zusätzlich muss betrachtet werden, welche Bedeutung Erfüllungen und Empfangsbestätigungen bei der Inanspruchnahme von Dienstleistungen, insbesondere bei Reservierungen haben.

Unabhängig davon, wie viel Rückholschritte gemacht werden müssen, wenn der Gesamtvertrag scheitert, kann bereits vorab eine Abgrenzung durchgeführt werden: Das asynchrone Protokoll besagt, dass mit der Erfüllung der Schuldverträge bereits begonnen werden kann, aber nicht begonnen werden muss, bevor der Gesamtvertrag abgeschlossen wurde. Unter WorldCheckInn könnte aber nur der Käufer bereits vorab mit der Erfüllung beginnen, wenn die Voraussetzungen dafür gegeben sind. Dienstleistungen sind nicht rückgängig zu machen, daher ist eine Erfüllung auf Basis von Teilschuldverträgen durch den Verkäufer mit einem Risiko verbunden. Bezieht man sich hierbei auf die Herstellung eines Unikates (s.u.), ist dem sogar abzuraten. Wenn durch eine unbekannte Ursache der Gesamtvertrag nicht zustande kommt, ist erneut zu klären, wer die entstandenen Kosten zu tragen hat.

Das hybride Protokoll setzt voraus, dass vor der Erfüllung der Schuldverträge die Schultilgungsmittel beider Parteien in den Schuldvertragszentren vorliegen müssen. Dies kann bei Dienstleistungen nur einseitig durch den Käufer erfüllt werden, wenn die Möglichkeit dafür gegeben ist (vgl. Treuhand). Da sie nicht lagerbar und rückgängig zu machen sind, trifft dies auf alle Formen von Dienstleistungen zu.

Das synchrone Protokoll sagt aus, dass der Gesamtvertrag zustande gekommen sein muss, bevor mit der Erfüllung begonnen werden kann. Sofern es die gegebene Dienstleistung ermöglicht, kann mit der Erfüllung begonnen werden. Mit dem synchronen Protokoll ist auch die Vertrauenssymmetrie gewahrt.

3.4.3 Abschluss und Erfüllung des Vertrages

Der Gesamtvertrag kommt zustande, wenn alle vier Teilschuldverträge abgeschlossen und dem Gesamtvertrag zuordbar sind. Somit liegt für beide Parteien bei der Verwendung des synchronen Protokolls ein rechtskräftiger Vertrag vor, auf den Bezug genommen werden kann. Anschließend erfolgt die Erfüllung des Zahlungsmittelschuldvertrages durch den Käufer und des Warenschuldvertrages durch den Verkäufer sowie die Ausstellung der Empfangsbestätigungen.

Bei der Inanspruchnahme einer Dienstleistung liegt es aufgrund der Eigenschaften einer Dienstleistung nahe, das Modell nach Wabner auf den Vertragsabschluss zu reduzieren: Eine Dienstleistung ist eine Leistung, die nicht der Produktion eines materiellen Gutes dient [wiki:Dienstleistung]. Dienstleistungen werden von natürlichen oder juristischen Personen zu einem bestimmten Zeitpunkt oder innerhalb eines Zeitrahmens erbracht. Dienstleistungen sind nicht materiell, die erbrachten Leistungen können jedoch in Form eines auf den Bedürfnissen des Verkäufers zugeschnittenen Gutes erbracht werden, wie zum Beispiel die Herstellung eines Unikates, welches dem Käufer anschließend geliefert wird. Die Fertigstellung des Unikates ist also nicht materiell, sondern eine Dienstleistung. Dienstleistungen sind nicht lager- oder übertragbar. Jedoch kann das Resultat wiederum ein Gut sein. Eine Dienstleistung ist also „jede Art von Arbeit, die im Sinne oder nach dem Wunsch eines Kunden ausgeführt wird. Anders beschrieben – das „Bemühen, das Bedürfnis des Kunden zu befriedigen“ [wiki:Dienstleistung]. Eine Ware hingegen ist ein bereits fertig gestelltes materielles Gut, welches dem Käufer angeboten wird, jedoch nicht durch diesen in Auftrag gegeben wurde.

Das Modell nach Wabner schließt jedoch mit ein, dass die Erfüllung mit abschließender Bestätigung überwacht und dokumentiert werden muss. Dies ist vor allem dann wichtig, wenn aufgrund der räumlichen und zeitlichen Trennung Quittungen für Käufer und Verkäufer ausgestellt werden müssen. Als Beispiel wird

die Lieferung eines fertig gestellten Unikats angenommen. Dies gehört nicht zur eigentlichen Dienstleistung (Herstellung des Unikates), ist jedoch notwendig um nachzuweisen, dass die Leistung erbracht wurde (Empfangsbestätigung).

Betrachtet man den Abschluss eines elektronischen Vertrages zur Inanspruchnahme von Dienstleistungen, so bleibt der Zahlungsmittelschuldvertrag in seiner ursprünglichen Form bestehen. Der Käufer ist zur Bezahlung der Dienstleistung verpflichtet, was bei entsprechender Protokollvariante auch sofort umgesetzt werden kann. Auf Grund der genannten Eigenschaften von Dienstleistungen sollte der Verkäufer erst mit der Erfüllung beginnen, wenn der Gesamtvertrag zustande gekommen ist.

Die Erfüllung des Schuldvertrages durch den Käufer kann bereits auf der Basis von abgeschlossenen Teilschuldverträgen vonstatten gehen. Der Hotelgast nimmt die Reservierung eines Hotelzimmers über ein Hotelbuchungssystem vor. Er kann bereits mit der Erfüllung beginnen, wenn ein rechtssicherer Vertrag vorliegt. Als Vergleich kann hier das sogenannte nicht akkreditierte eBay-Konto (www.ebay.de) herangezogen werden. Der Verkäufer erfüllt vorab den Zahlungsmittelschuldvertrag, indem er ein gesetzliches Zahlungsmittel auf ein durch eBay bereitgestelltes Konto überweist. Somit hat der Verkäufer die Gewissheit, dass die Leistung bezahlt werden kann. Die Erfüllung des Schuldvertrages durch den Verkäufer hingegen setzt einen rechtskräftigen Gesamtvertrag voraus. Dies ist vor allem dann von Bedeutung, wenn ein zeitlicher Rahmen oder ein Zeitpunkt festgelegt wird, zu dem die Leistung zu erbringen ist. Hier kann das vorhin schon erläuterte Beispiel, dass auf Wunsch eines Käufers ein Unikat hergestellt wird, herangezogen werden. Der Käufer kann bereits nach Vertragsabschluss den Zahlungsmittelschuldvertrag erfüllen und der Händler kann mit der Herstellung des Unikats beginnen, vorausgesetzt, der abgeschlossene Gesamtvertrag liegt vor.

Die Lieferung des Unikates wiederum kann aber auch Bestandteil des Vertrages sein (Herstellung und Lieferung) und somit zu der Dienstleistung gehören. Dieser Vorgang ist daher mit der Erfüllung des Warenschuldvertrages vergleichbar und eine Umsetzung auf Basis des Modells nach Wabner mit Vertragsabschluss und Erfüllung beziehungsweise Empfangsbestätigungen wäre möglich.

Erfüllung bei Reservierungen

Der Gast erfüllt die im Reservierungsvertrag vereinbarten Pflichten, wenn er das Zimmer in Anspruch nimmt, also einen Beherbergungsvertrag antritt oder die Stornogebühr bezahlt. Der Hotelier erfüllt seine Pflicht, wenn er sicherstellt, dass das reservierte Zimmer innerhalb der Anreisefrist des Gastes und während des gesamten Aufenthaltes zur Verfügung steht. Die Erfüllung der Tatsache, dass das Zimmer für den vereinbarten Zeitraum zur Verfügung stehen muss, ist aber auch bereits abgeschlossen, wenn der Hotelgast den Beherbergungsvertrag antritt. Der Beherbergungsvertrag beinhaltet auch die Dauer des Aufenthalts, so dass mit Abschluss dieses Vertrages der Hotelgast bestätigt, dass die Leistung vollständig erbracht und somit alle vertraglich vereinbarten Pflichten durch das Hotel eingehalten wurden. Für den Hotelgast und das vorab in der Reservierung vereinbarte Zahlungsmittel gilt dasselbe.

Warum sind Empfangsbestätigungen von Bedeutung?

Zur vollständigen Dokumentation des Geschäftsvorganges gehören die Empfangsbestätigungen. Das Modell nach Wabner besagt, dass die Erfüllung des Vertrages durch den Schuldner vom Gläubiger bestätigt werden muss. Jede der beteiligten Parteien, Käufer und Verkäufer, fällt einmal in die Rolle des Gläubigers und muss den Empfang des Schuldtilgungsmittels bestätigen, um so später jeweils die Erfüllung ihrer Schuld nachweisen zu können [Weiser, S. 7]. Analog dazu kann bei Inanspruchnahme einer Dienstleistung der Käufer dazu veranlasst werden, die Dienstleistung zu bestätigen, wenn diese erbracht wurde. Der Verkäufer wiederum kann eine Zahlungsmittelempfangsbestätigung ausstellen.

Der Hotelgast kann bei einer Reservierung aufgefordert werden, die Dienstleistung zu bestätigen, wenn diese erbracht wurde (Check-In). Der Hotelier hingegen bestätigt, dass das Zimmer innerhalb der festgelegten Anreisefrist in Anspruch genommen wurde. Beide Parteien bestätigen somit bei diesem Vorgang die Erfüllung des Vertrages durch den anderen. Die Bestätigung durch den Hotelier gewinnt vor allem dann an Bedeutung, wenn der Hotelgast zur Vorlage eines Ausweispapieres verpflichtet ist. Dadurch ist im Nachhinein die Gewissheit gegeben, dass unter der vorgegebenen Identität eine ladungsfähige Adresse existiert, auf die Bezug genommen werden kann.

Als vertrauenswürdig kann also angesehen werden, wer bereits über ein Hotelbuchungssystem reserviert und sich im Hotel persönlich ausgewiesen hat. Von diesem Hotelgast sind Meldedaten bis hin zu Abrechnungsdaten hinterlegt, die bereits anhand des Personalausweises beglaubigt sind.

3.5 Verträge durch Softwareagenten

Softwareagenten können anstelle einer natürlichen Person als Vertragspartner auftreten, um Verträge für diese abzuschließen. Ein Beispiel für die Nutzung von Softwareagenten ist das Projekt SESAM [SESAM].

SESAM ist ein neuartiges Konzept zur ökonomischen Nutzung virtueller Kraftwerke. Unter virtuellen Kraftwerken versteht man einen Zusammenschluss „von kleinen, dezentralen Kraftwerken zu einem Verbund, die von einer zentralen Warte gesteuert werden.“ [wiki:Virtuelles Kraftwerk] Das Projekt SESAM hingegen beschreibt einen Verbund von dezentralen Kraftwerken, die nicht von einer zentralen Stelle aus gesteuert werden. Dieser Verbund besteht aus Stromverbrauchern, Stromerzeugern und jenen, die beides sind. Diese werden miteinander so koordiniert, dass der Strom effizient erzeugt, verteilt und genutzt werden kann. Damit diese Koordination und Effizienz erreicht wird, werden sogenannte Softwareagenten eingesetzt. Diese können anstelle eines Stromverbrauchers oder -erzeugers Verträge abschließen, die kurzzeitig und jederzeit kündbar sind.

3.5.1 Rahmenverträge

Das Projekt SESAM ist sich der rechtlichen Probleme und Hürden von Softwareagenten bewusst und fügt einen weiteren ‚Schwerpunkt‘ hinzu. Verträge und Allgemeine Geschäftsbedingungen (AGB) werden nicht vorab in starren Rahmenverträgen verankert, sondern sind frei und individuell zwischen den Softwareagenten mit Hilfe von Mediatoren verhandelbar. Um zu verhindern, dass die

AGB keine unzulässigen oder widersprüchlichen Paragraphen enthalten, verfügen die Mediatoren über formalisiertes Recht. Das heißt, Verträge und AGB können je nach Anforderung frei formuliert und anhand des formalisierten Rechtes auf Einhaltung gesetzlicher Rahmenbedingungen geprüft werden.

Die Registrierung neuer Datenempfänger unter WorldCheckInn beinhaltet die Verpflichtung zum Abschluss eines Rahmenvertrages in schriftlicher Form, um zukünftige Dienste in Anspruch zunehmen. Dies sind Rahmenverträge im Sinne der Rahmenvereinbarungen wie sie in § 305 Absatz 3 BGB geregelt sind. Das heißt, es besteht die Möglichkeit, AGB für eine bestimmte Art von Rechtsgeschäften im Voraus zu vereinbaren. Diese müssen nach §305, Absatz 2 BGB für jeden Vertragspartner einsehbar sein.

Der Nachteil dieser Rahmenverträge ist, dass nach §305c BGB Unklarheiten immer zu Lasten des Verwenders der AGB gehen. Alles was nicht explizit erwähnt ist, ist nicht Vertragsgegenstand. Somit können im Nachhinein keine zusätzlichen Leistungen geltend gemacht werden. Die Verträge und dazugehörigen AGB müssen die zugrundeliegenden gesetzlichen Bestimmungen einhalten.

3.5.2 WorldCheckInn in der Rolle eines Softwareagenten?

Bereits bestehende Hotelbuchungsportale geben die Reservierungen über unsichere Kommunikationsmittel, wie zum Beispiel E-Mail weiter oder tragen sie in das eigene Portal ein, von dem die Reservierungen dann durch den Hotelier in das eigene System übernommen werden. Der Hotelier ist gezwungen, den Stand seiner Zimmerverfügbarkeit aktuell zu halten, indem er fortlaufend Informationen vom Hotelbuchungssystem in sein eigenes System überträgt und umgekehrt. Einerseits ist er dazu vertraglich verpflichtet (zum Beispiel hotel.de), andererseits muss er Überbuchungen vermeiden. WorldCheckInn wiederum automatisiert diesen Datentransfer zwischen Hotelbuchungssystemen und Hotelmanagementsystemen.

WorldCheckInn geht gegenüber bereits bestehenden Hotelbuchungssystemen noch einen Schritt weiter: WorldCheckInn tritt anstelle des Hoteliers gegenüber dem Reservierenden auf und schließt in dessen Vertretung Reservierungsverträge ab, welche fest im hoteleigenen System mit vertragstypischen Eigenschaften verankert

werden. Hier zeigt sich eine ähnliche Beziehung wie bei den virtuellen Stromagenten des Projektes SESAM. Der Mensch als natürliche Person setzt zum Abschließen von Verträgen einen computergestützten Agenten ein. Der Unterschied zum System WorldCheckInn besteht nur darin, dass zwischen dem Softwareagent und der natürlichen Person vorab keine Allgemeinen Geschäftsbedingungen vertraglich vereinbart werden. An diesem Punkt genießt WorldCheckInn gegenüber dem Projekt SESAM einen entscheidenden Vorteil: Zur Nutzung des Systems WorldCheckInn werden „starre Rahmenverträge mit fest vordefinierten Allgemeinen Geschäftsbedingungen“ [Ra06] abgeschlossen. Da vorab ein Vertrag zwischen Datenempfänger und Softwareagenten geschlossen wird, reduzieren sich die Probleme beim Einsatz von WorldCheckInn Grundsätzliche Hürden sind technischer Natur (ausgenommen Datenschutz und -weitergabe), wie zum Beispiel die Zusicherung, dass für einen gewünschten Zeitraum die benötigte Zimmeranzahl frei ist und reserviert werden kann. Die Vertragsvorlagen unterscheiden sich nur in Vertragsparametern, die durch den Gast festgelegt und technisch zugesichert werden müssen. Variable Parameter sind zum Beispiel Anzahl der Gäste, Anzahl der Nächte oder Ermäßigungen.

3.6 Einsatz und Beweiswürdigung digitaler Signaturen

Zertifikate ermöglichen den Einsatz qualifizierter digitaler Signaturen, welche der handschriftlichen gleichgesetzt sind [SigG]. Sie sollen die Authentizität, Integrität und Unabstreitbarkeit des elektronischen Vertrages sowie die Identität des Signierenden sicherstellen. Rechtssicherheit kann hergestellt werden, wenn im Rechtssystem festgelegt wird, unter Anwendung welcher technischer Verfahren die getätigten Geschäfte als Rechtsgeschäfte anerkannt werden [Weiser]. Im Signaturgesetz [SigG] ist dies unter anderem in §7 SigG „Inhalt von qualifizierten Zertifikaten“ und §17 SigG „Produkte für qualifizierte elektronische Zertifikate“ verankert.

Nach dem Signaturgesetz wird zwischen folgende Signaturarten unterschieden:

- Einfache elektronische Signatur
- Fortgeschrittene elektronische Signatur
- Qualifizierte elektronische Signatur
- Qualifizierte elektronische Signatur eines akkreditierten Zertifizierungsdiensteanbieters

Für die einfache und die fortgeschrittene elektronische Signatur bestehen keine rechtlichen Vorgaben zur Qualität der Signaturen. Diese Signaturen sind als Beweismittel zulässig, unterliegen aber einer freien Beweiswürdigung. Anhand der fortgeschrittenen elektronischen Signatur muss der Signaturschlüsselinhaber eindeutig und ausschließlich identifizierbar sein. Grundlage der qualifizierten elektronischen Signatur ist ein qualifiziertes Zertifikat und der Umstand, dass die Signatur mit einer sicheren Signaturerstellungseinheit erstellt wurde. Die Verwendung und Voraussetzungen für sichere Signaturerstellungseinheiten sind in §17 und §18 des Signaturgesetzes [SigG] verankert. Darüber hinaus muss der Herausgeber des Zertifikates die Anforderungen als Zertifizierungsdiensteanbieter erfüllen. Ist der Zertifizierungsdiensteanbieter freiwillig akkreditiert, so besteht die Gewissheit, dass dieser Anbieter durch öffentlich anerkannte Stellen nach §18 SigG überprüft wurde. Gegenüber der einfachen und der fortgeschrittenen elektronischen Signatur bietet die qualifizierte elektronische Signatur eine erhöhte Beweiskraft. Abbildung 3.6.1 fasst dies noch einmal zusammen.

Ein qualifiziertes Zertifikat muss die folgenden Informationen enthalten:

- Namen des Signaturschlüsselinhabers
- Signaturprüfschlüssel
- Algorithmus
- Laufende Nummer des Zertifikats
- Gültigkeit des Zertifikates
- Namen des Zertifizierungsdiensteanbieters
- Beschränkung auf bestimmte Anwendungen
- Angaben, dass es sich um ein qualifiziertes Zertifikat handelt
- Weitere optionale Attribute des Schlüsselinhabers

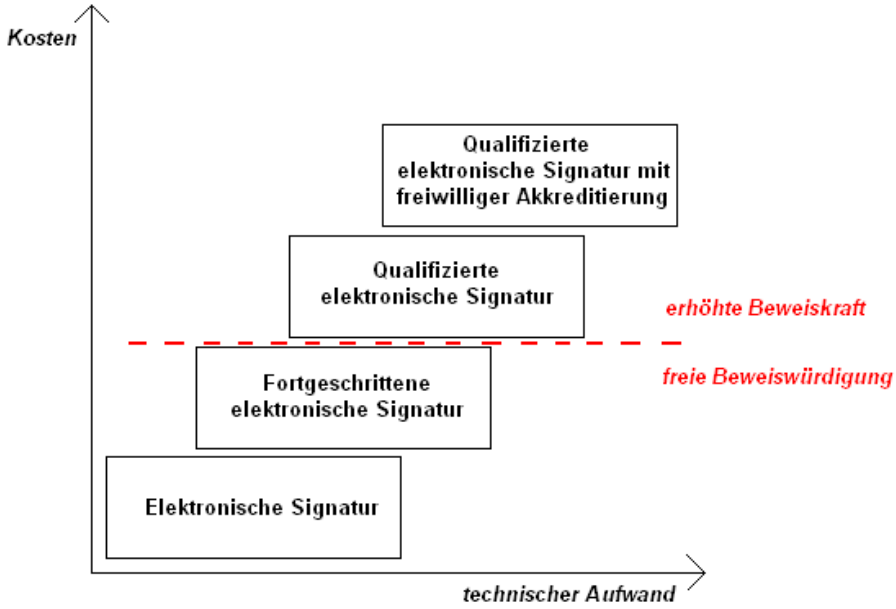

Abbildung 3.6.1: Übersicht der Signaturformen

Gültigkeit von Zertifikaten

Bei der Verwendung von Zertifikaten ist die Unterschrift nur gültig, wenn zum Zeitpunkt ihrer Leistung auch das Zertifikat gültig war. Dieses Problem beschreibt bereits [Weiser] und erklärt in diesem Zusammenhang auch die Problematik des Speicherns von Zertifikaten beziehungsweise ihrer Gültigkeit. Auf Grund dessen ist es zwingend erforderlich, dass bei Leistungserbringung die Gültigkeit überprüft und festgehalten wird. Es besteht die Gefahr, dass im Nachhinein die Überprüfung der Gültigkeit für den Zeitpunkt der Leistungserbringung nicht mehr gewährleistet ist, zum Beispiel bei einem Verlust des Zertifikates.

Um den Zeitpunkt einer Signatur zu garantieren, kommen qualifizierte Zeitstempel zum Einsatz. Qualifizierte Zeitstempel sind elektronische Bescheinigungen eines Zertifizierungsdiensteanbieters, der nach dem Signaturgesetz mindestens die Anforderungen nach §§ 4–14 und §17 beziehungsweise §23 SigG für nicht-EU-Länder einhält. Zeitstempel werden genutzt, um nachzuweisen, dass der Inhalt einer Datei und bereits vorhandene Signaturen zu einem bestimmten Zeitpunkt vorlagen.

Zertifikate für Schlüssel, die nicht mehr sicher sind, können über eine so genannte Certificate Revocation List gesperrt werden. Eine Certificate Revocation List (Zertifikatsperrliste) ist eine Liste mit Informationen über die Gültigkeit von Zertifikaten. Diese dient dem Sperren von unsicheren Schlüsseln. Sie ist Bestandteil der PK-Infrastruktur und ist signiert in der Datenbank abgelegt oder liegt dem Verzeichnisdienst der Zertifizierungsstelle vor [wiki:CRL].

Zertifikate unter WorldCheckInn

Bei der Registrierung neuer Datensender oder der erstmaligen Reservierung wäre der Einsatz von Zertifikaten zu nennen. Die Echtheit des Zertifikates wird durch eine digitale Signatur einer vertrauenswürdigen Organisation oder Instanz, zum Beispiel einer Behörde, garantiert. Allerdings bieten auch Organisationen, wie zum Beispiel https://cert.startcom.org/, kostenlose Zertifikate ohne jede Authentifizierung des Antragsstellers an [heise.de01]. Zudem wird der geheime Schlüssel nicht lokal, sondern auf dem Server generiert. An die Certification Authority (CA) wird nur ein Zertifizierungsantrag übermittelt, den diese unterschreibt und zurücksendet. Daher ist keine Garantie gegeben, dass nicht doch eine Kopie des geheimen Schlüssels des Antragsstellers weiterhin auf dem Server gespeichert wird, um im Nachhinein die verschlüsselten Verbindungen einer Client-Server-Kommunikation zu dechiffrieren.

Ein Zertifikat ist dann als seriös oder vertrauenswürdig einzustufen, wenn es einer seriösen Zertifizierung unterlag. Dies bedeutet, dass der Antragssteller sich persönlich ausgewiesen haben muss. Mit dem PostIdent-Verfahren der Deutschen Post ist dies gegeben [www.signtrust.de]. Eine Außenstelle der Post fungiert dabei als Registrierungsstelle.

Der allgemeine Ablauf einer Registrierung bei einer Registrierungsstelle gestaltet sich durch

- die Identifizierung des Antragsstellers,
- die Festlegung des Geltungsbereiches der Signatur,
- die Weiterleitung des Antrages an eine Zertifizierungsstelle und
- der Ausgabe an den Antragsteller.

Daraus ergibt sich der Ablauf der Registrierung unter WorldCheckInn. WorldCheckInn dient lediglich als automatisierter ‚Vermittler'. Zwischen dem Hotelgast und WorldCheckInn besteht eine räumliche und zeitliche Trennung. Das Hotel stellt hierbei den einzigen Kontakt zum Hotelgast dar, da sich dieser beim erstmaligen Einchecken persönlich ausweisen kann. Dies drängt sich vor allem dann auf, wenn es sich nicht um ein qualifiziertes Zertifikat handelt. Der Vorteil einer Identitätsprüfung besteht darin, dass auch ‚schwächere' Zertifikate beziehungsweise Signaturen Anwendung finden.

Beim Fehlen eines Zertifikates könnte das Hotel als Registrierungsstelle dienen. Der Antragsteller wird identifiziert, indem er sich im Hotel persönlich ausweist. Anschließend laufen die oben genannten Schritte einer Registrierung ab. WorldCheckInn nimmt somit die Rolle einer Zertifizierungsstelle ein. Für die Generierung des qualifizierten Zertifikats müssen die Voraussetzungen nach §5 SigG gegeben sein.

4. Anforderungen an die technische Umsetzung des Systems WorldCheckInn

Zentraler Bestandteil von WorldCheckInn ist der Transfer personenbezogener Daten. Daher sind Sicherheitsbetrachtungen aller Geschäftsvorgänge, von der Registrierung neuer Datensender bis zum Check-In notwendig, da hierbei die Speicherung und Übertragung personenbezogener Daten erfolgt. Diese unterliegen dem Datenschutzgesetz. Personenbezogene Daten sind besonders schutzbedürftige Daten, deren bekannt werden die Betroffenen erheblich beeinträchtigen sowie dem Ansehen von *worldcheckinn.com* schaden kann. Dadurch entsteht ein Vertrauensverlust, der im Nachhinein erhebliche finanzielle Einbußen mit sich bringen kann. Deshalb wird der Schutz dieser Daten sowie die Wahrung der Vertraulichkeit und Integrität wichtigster Bestandteil weiterer Sicherheitsbetrachtungen sein.

Nicht-technische Maßnahmen, zum Beispiel organisatorische Vereinbarungen (Zugang nur für autorisiertes Personal) oder interne Regelungen, zum Beispiel abgeschlossene Serverräume, fallen in den Aufgabenbereich des Hosting-Providers, bei dem WorldCheckInn betrieben wird. Diese Maßnahmen werden außer Betracht gelassen. Es soll hier jedoch noch einmal darauf hingewiesen werden, dass diese vor Inbetriebnahme des Systems vertraglich vereinbart werden müssen.

4.1 Architektur

Abbildung 4.4.1 zeigt das Gesamtsystem der Prototypentwicklung, wie sie in Abschnitt 2.2 beschrieben ist. Die Komponenten des Gesamtsystems sind WorldCheckInn, registrierte Datenempfänger (Hotels) und Terminal-Provider. Zwischen Hotel und Terminal-Provider werden Terminal-Transaktionen durchgeführt, zum Beispiel die Abrechung per Kreditkarte. WorldCheckInn nimmt die Rolle eines Dritten ein, der vom Terminal-Provider informiert wird, wenn ein

registrierter Datensender im Hotel den Check-In angestoßen hat. Daraufhin gibt er anhand dieser Information die zugehörigen Daten an das Hotel weiter.

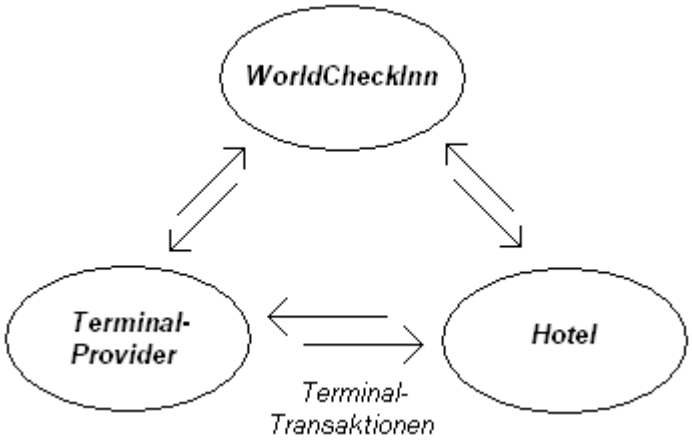

Abbildung 4.4.1: Gesamtsystem

Zur besseren Veranschaulichung der Architektur wird vorab beschrieben, aus welchen Segmenten das System WorldCheckInn besteht. Es wird eine Einteilung des Systems in drei Schichten [TI04] vorgenommen: die Präsentations-, Funktionalitäts- und Datenhaltungsebene. Auf der Präsentationsebene befindet sich die Oberfläche der Internet-Anwendung www.worldcheckinn.com. Die darunter liegende Funktionalitätsebene verbindet die Präsentationsebene mit der Datenhaltung und sorgt für die Abwicklung aller Geschäftsvorgänge (siehe Abbildung 4.1.2). Die Datenhaltungsebene enthält Informationen, die für alle Geschäftsvorgänge auf der Funktionalitätsebene und der Präsentationsebene zur Verfügung stehen. Da alle drei Ebenen jedoch nicht an eine gemeinsame Plattform gebunden sind, können sich diese z. B. in einem Intranet an unterschiedlichen Orten befinden. Abbildung 4.1.2 zeigt eine Übersicht aller Segmente. Anschließend folgen einführende Erläuterungen zu den einzelnen Segmenten des Systems.

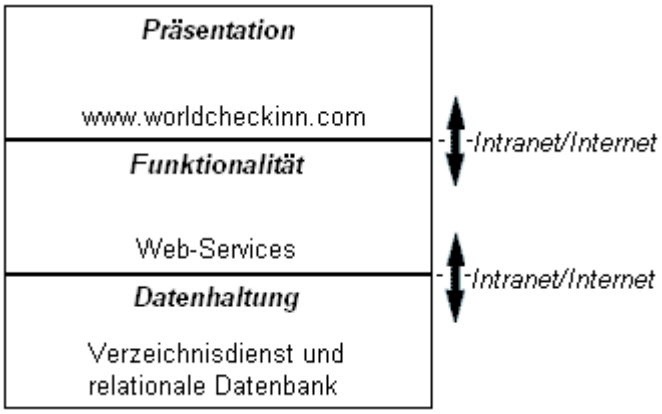

Abbildung 4.1.2: Drei-Schichten-Architektur von WorldCheckInn

Ziel dieser Drei-Ebenen-Architektur ist die Trennung von Präsentation, Anwendung (Funktionalität) und Datenhaltung. Änderungen in den einzelnen Ebenen wirken sich nicht auf darüber oder darunter liegende Ebenen aus [TI04].

4.1.1 Präsentationsebene

Bestandteil der Präsentationsebene ist die Webanwendung, eine grafische Oberfläche für den WWW-Browser des Clients. Die Oberfläche befindet sich auf einem WWW-Server. Sofern nicht nur statische Webseiten aufgerufen werden, dient der WWW-Server dem Zweck, andere Anwendungen aufrufen zu können.

Zur Entwicklung der Webanwendung kommt ASP.NET zum Einsatz. ASP.NET ist Bestandteil des .NET-Frameworks zur Erstellung von dynamischen Webseiten und ist programmiersprachenunabhängig. ASP.NET dient einerseits der Erstellung von dynamischen Webseiten auf der Präsentationsebene, andererseits der Implementierung von XML Web Services auf der Anwendungsebene. ASP.NET basiert auf dem Prinzip der Active Server Pages (ASP). Diese werden serverseitig ausgeführt und bestehen aus HTML-Code und optionalem Programmcode [Sch02]. Für die Präsentationsebene steht der Internet Information Server (IIS) zur Verfügung. Dieser dient zur Darstellung der Active Server Pages und der Ausführung von .NET-Anwendungen.

4.1.2 Funktionalitätsebene

Der Funktionalitätsebene liegen ASP.NET Web-Services zugrunde. Ein Web-Service kann einen oder mehrere Dienste über eine Schnittstelle anbieten. Zum Einen kann er durch einen Client angesprochen werden, zum Anderen können weitere Agenten (Web-Services) mit diesem Informationen austauschen. Auf dieser Ebene befindet sich ein Web-Service zum Zugriff auf den Verzeichnisdienst der Datenebene. Der Web-Service soll einerseits der Webanwendung auf der Präsentationsebene eine Schnittstelle zum Zugriff auf den Verzeichnisdienst anbieten. Andererseits enthält er eine Schnittstelle für weitere Agenten, also Web-Services, die eine einheitliche Schnittstelle zum Zugriff auf die Datenhaltung der Datenempfänger besitzen. Abbildung 4.1.3 soll diese 1:n-Beziehung veranschaulichen.

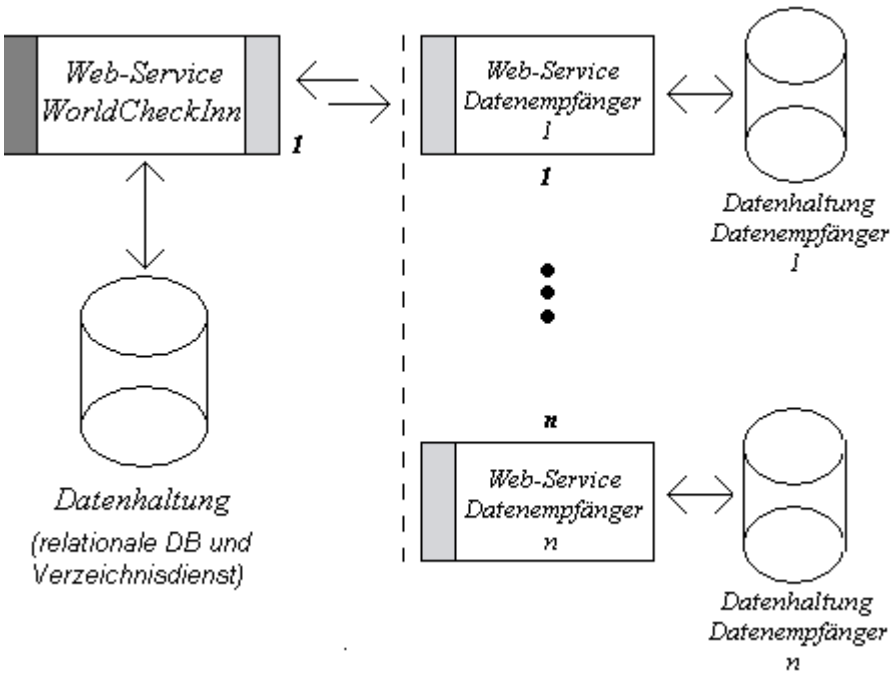

Abbildung 4.1.3: 1:n-Beziehung des WorldCheckInn Web-Services

4.1.3 Datenhaltungsebene

Die Datenebene setzt sich aus einem Verzeichnisdienst zum Speichern von Informationen, die nur selten geändert werden (Datenempfänger und Datensender - Informationen), und einer relationalen Datenbank zum Speichern von Informationen, die häufigen Änderungen unterworfen sind (Reservierungsdaten, Transaktionsdaten), zusammen. Zur Umsetzung des Verzeichnisdienstes kommt der Verzeichnisdienst OpenLDAP zum Einsatz. Dieser basiert auf dem Netzwerkprotokoll LDAP (Lightweight Directory Access Protocol) [RFC2251]. Ein LDAP-Verzeichnisdienst eignet sich als Verzeichnis zur zentralen Benutzerverwaltung, sowie zum Aufbau eines Authentifizierungsdienstes. Neben umfangreichen Verzeichnisoperationen ist eine Zugangskontrolle durch Erweiterungen, wie zum Beispiel SASL (Simple Authentication and Security Layer) [RFC 2222], vorhanden und durch die Unterstützung von Transport Layer Security (TLS) [RFC 2246] kann der Transportweg mittels asymmetrischer Verschlüsselung gesichert werden.

In [Weiser] werden bereits die Einsatzmöglichkeiten von zentralisierten Verzeichnissen in Form von Verzeichnisdiensten beschrieben. Dabei liegt der Schwerpunkt auf der Herstellung und Wahrung von Sicherheit in Verzeichnisdiensten. Verzeichnisse finden unter anderem Anwendung als Sicherheitsinfrastruktur, zur Verwaltung von Rechten und Eigenschaften (Benutzerverwaltung, Zugriffsrechte) oder der Speicherung von Zertifikaten und Zertifikats-Widerrufslisten (CRL). Die von Weiser untersuchten Verzeichnisdienste sind in der folgenden Tabelle zusammengefasst:

	DNS	**(R)WHOIS**	**X.500**	**(Open-) LDAP**
Authentifizierung, Zugriffskontrolle	Auf Anwendungsebene notwendig	In RWHOIS, wenig bewährt	vorhanden	Durch SASL und TLS
Such-möglichkeiten	schlecht	schlecht	Umfangreiche Suche: Lesen, Vergleichen, Auflisten	Umfangreiche Suche: Lesen, Vergleichen, Auflisten
Ändern von Datensätzen	nur mit Dynamic Update (selten genutzt)	nicht möglich	möglich	möglich
Einsatz-möglichkeiten	Namespaces, White Pages	Universell	Universell	Zentrales Ressourcen-verzeichnis, Authentifizierungs-dienst

Tabelle 4.1.1: Bewertung der Verzeichnisdienste nach Weiser

4.1.4 Ablauf des Check-In

Zur besseren Veranschaulichung der Geschäftsvorgänge der Funktionalitätsebene wird in Abbildung 4.1.4 der Ablauf eines Check-In im Hotel vereinfacht dargestellt. Anhand der Schritte 1 bis 6, welche im darauf folgendenden Text erläutert sind, wird deutlich gemacht, welche Rolle dem Terminal-Provider zufällt.

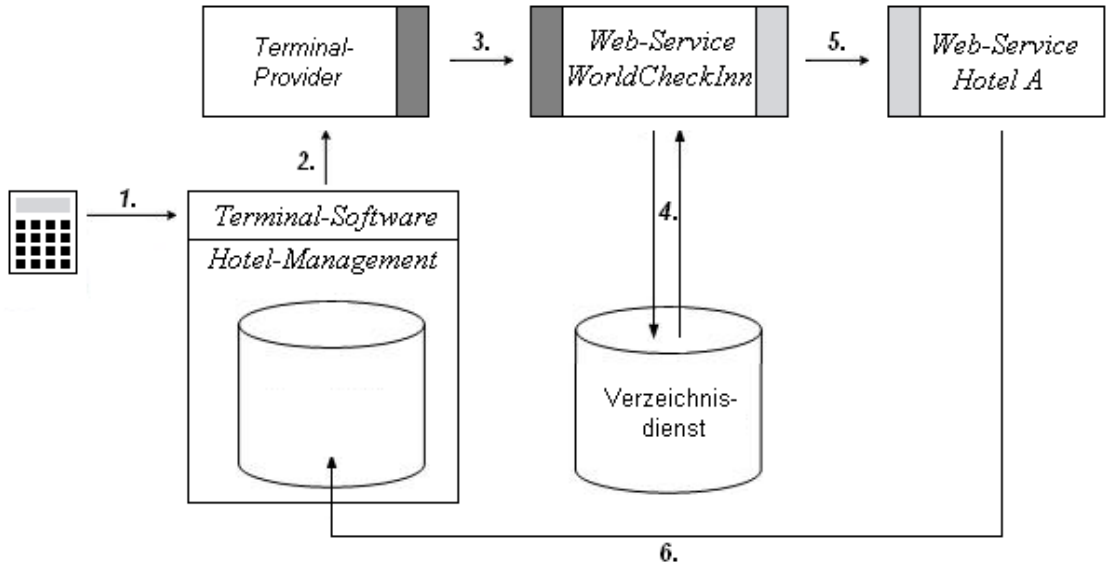

Abbildung 4.1.4: Der Check-In

1. Die Kartendaten der Kunden- oder Kreditkarte werden eingelesen und die dazugehörige Persönliche Identifikationsnummer (PIN) eingegeben
2. Der Datensender wird anhand von Kunden- oder Kreditkarte und der dazugehörigen PIN beim Terminal-Provider authentifiziert
3. Bei erfolgreicher Authentifizierung gibt der Terminal-Provider die Kundennummer des Datensenders und -empfängers an die Schnittstelle des WorldCheckInn-Web-Services weiter
4. Dieser liest alle freigegebenen persönlichen und abrechnungsrelevanten Daten des Datensenders im Verzeichnisdienst aus
5. Anschließend erfolgt der Aufruf einer Schnittstelle des hotelspezifischen Web-Services,
6. welcher alle Informationen in die Datenbank des Hotels einträgt.

4.2 Entwicklungsumgebung .NET

Hinter .NET verbirgt sich ein Framework, welches eine Ansammlung von Klassenbibliotheken und Diensten enthält. Die .NET Klassenbibliothek ist eine zentrale Ansammlung von Systemfunktionen, zum Beispiel Dateisystemzugriffe. Diese Klassen liegen in sogenannten *Global Assemblies*. Vorteil dieser *Assemblies* ist, dass diese von allen Programmiersprachen verwendet werden, welche .NET unterstützt (Visual Basic .NET, C#, JScript .NET, Visual C++). .NET besitzt eine eigene Speicherverwaltung, die unter dem Namen *Garbage Collector* bekannt ist. So wird verhindert, dass Zeiger falsch gesetzt werden oder nicht mehr benötigte Objekte im Speicher verbleiben. Das .NET Framework unterstützt neben dem Aufruf von Programmcode unterschiedlicher Sprachen auch die Vererbung von Klassen, welche in einer anderen objektorientierten Sprache geschrieben sein können. Verantwortlich dafür ist die Microsoft Intermediate Language (MSIL), die CLR und die Common Language Specification (CLS) [Sch02]. .NET-Compiler erzeugen den plattformunabhängigen Zwischencode MSIL. „Die CLS ist ein Regelwerk für Compiler, das festlegt, wie die Umsetzung von sprachspezifischen Konzepten in die MSIL erfolgen muss. Die CLS definiert, welche Konstrukte jede .NET-Programmiersprache bereitstellen sollte. Grundvoraussetzung für die Integration ist das Common Type System (CTS), das ein einheitliches System zur Implementierung von Datentypen definiert“ [Sch02, S. 27].

Die Version 1.0 des .NET Framework wurde im Januar 2002 freigegeben und liegt seit November 2004 in der Version 2.0 (Community Preview) vor [nethistory]. Eine Liste aller Werkzeuge erhält man unter www.dotnetframework.de/dotnet/tools.aspx. ASP.NET liegt inzwischen in der Version 2.0 vor. Voraussetzung für ASP.NET war bisher der Internet Information Server (IIS), inzwischen liegen jedoch auch Portierungen für andere Web-Server vor, zum Beispiel für den Apache.

ASP.NET steht der ganze Funktionsumfang von .NET-Security zur Verfügung. Unter dem Stichwort „.NET Security” verbergen sich rollen- und codebasierte Sicherheitsrichtlinien und Funktionen. .NET Security setzt auf dem Sicherheitsmodell des Betriebssystems auf. Zur sicheren Ausführung des Programmcodes kommt die .NET-Laufzeitumgebung Common Language Runtime (CLR) zum Einsatz. Die zwischengeschaltete CLR sorgt für eine kontrollierte Ausführung von Programmcode. Auf Anwenderseite überwacht die CLR den auszuführenden Code

von fremden Webseiten, zum Beispiel ActiveX-Steuerelemente oder Java-Applets. Dem Namespace `System.Security` liegt die Struktur des Sicherheitssystems der CLR, einschließlich der Basisklassen für Berechtigungen, zugrunde [MSDN]. Hinzu kommen Werkzeuge zum Schutz des Programmcodes. Wie auch bei Fehlermeldungen auf der Datenbank- oder Verzeichnisdienstebene kann der aufgedeckte Code dem Angreifer verhelfen, Sicherheitslücken in der Implementierung zu finden. Mit Hilfe dieser Werkzeuge, welche dem gesamten .NET Framework zur Verfügung stehen, kann der Code unlesbar gemacht werden.

In Abschnitt 4.7 wird zur Darstellung und Anwendung der Web Service Enhancements (WSE) das Visual Studio .NET verwendet. Das Visual Studio wiederum ist selbst ein Werkzeug des .NET Frameworks. Es bietet eine grafische Entwicklungsumgebung für die Erstellung von Anwendungen auf Basis des .NET Frameworks. Für ASP.NET ist es ein hilfreiches Werkzeug zur Erstellung von dynamischen Webseiten oder Web-Services. Verschiedene Editoren ermöglichen das Bearbeiten von HTML-, XML- oder ASP- beziehungsweise ASPX-Dateien.

4.3 Herangehensweise nach IT-Grundschutz des Bundesamtes für Sicherheit in der Informationstechnik

In [heise.de02] wird gefragt: „Wie bringe ich die Anforderungen an meine IT-Infrastruktur mit den Sicherheitsansprüchen unter einen Hut?“

Das IT-Grundschutzhandbuch des Bundesamtes für Sicherheit in der Informationstechnik [BSI] bietet einen Leitfaden zur Erstellung eines IT-Sicherheitskonzepts für kleine, mittelständige und große Unternehmen. Es werden grundlegende Sicherheitsmaßnahmen aus den Bereichen Infrastruktur, Organisation, Personal, Technik und Notfallvorsorge aufgezeigt. Das Grundschutzhandbuch ist in Bausteine untergliedert. Jeder Baustein beschreibt zunächst die zu erwartende Gefährdungslage und anschließend Maßnahmen, die der Erstellung und Umsetzung eines Sicherheitskonzepts dienen. Darüber hinaus wird das Grundschutzhandbuch

halbjährlich aktualisiert, um die Forderung nach ständiger Kontrolle und Überarbeitung des Sicherheitskonzepts im laufenden Betrieb umsetzen zu können:

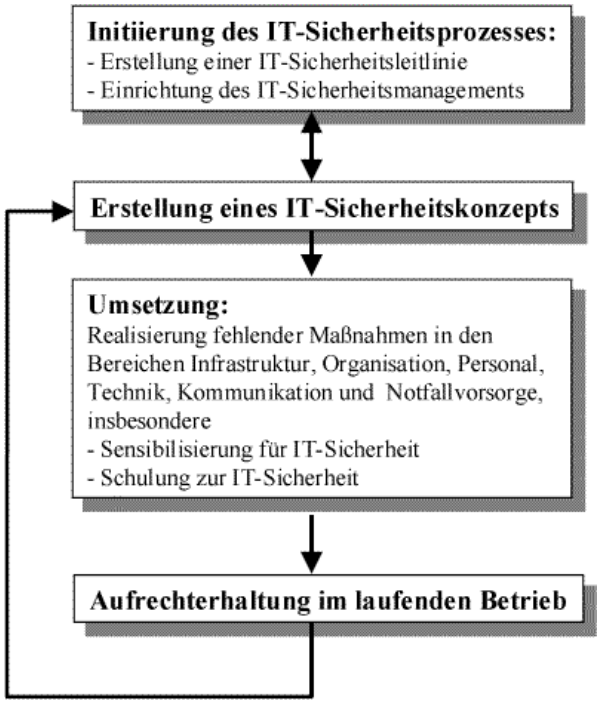

Abbildung 4.3.1: Erstellen des IT-Sicherheitskonzepts [BSI]

4.4 Anforderungen

Standard-Sicherheitsmechanismen bilden einen guten Basisschutz, sind jedoch nicht ausreichend. Erforderliche zusätzliche Sicherheitsmaßnahmen „müssen individuell auf der Grundlage einer ergänzenden Sicherheitsanalyse ermittelt werden“ [BSI]. Auch muss berücksichtigt werden, dass vertragliche Regelungen zur Umsetzung von Sicherheitsmechanismen in Hotels, welche Zugriff auf das System WorldCheckInn haben, nur bedingt durchgesetzt werden können. Daher ist es erforderlich, entsprechende Sicherheitsmaßnahmen zu treffen, wie zum Beispiel die

Zugriffsrechte auf ein Minimum zu begrenzen, um einem Angriff auf eine System-Komponente entgegen zu wirken oder die Auswirkungen zu minimieren.

WorldCheckInn wird auf einen Hosting-Provider ausgelagert (*Outsourcing*, dt.: auslagern). Dies umfasst die Bereitstellung benötigter Hardware sowie die Umsetzung der geforderten Sicherheitsmaßnahmen. Auf Basis der bereitgestellten Hardware werden eigene Anwendungen implementiert und installiert (*Application Hosting*). Dadurch entsteht eine starke Abhängigkeit zum Dienstleistungsanbieter, was im schlimmsten Fall die Existenz des Unternehmens gefährden kann. Gleiches gilt für den Wechsel des Dienstleistungsanbieters, wenn während des Betriebes geforderte Maßnahmen oder Änderungen nicht eingehalten oder umgesetzt werden können. Deshalb fallen Sicherheitsaspekten und vertraglichen Regelungen zentrale Rollen zu [BSI01]. Zur Einhaltung der Sicherheitsaspekte gehören ebenfalls Maßnahmen zur Kontrolle „der vertraglich vereinbarten Ziele und Leistungen sowie der IT-Sicherheitsmaßnahmen“ [BSI01]. Als Wegweiser für das *Outsourcing* ist das Kapitel 3 „Übergeordnete Komponenten“ von Bedeutung. Dieses Kapitel enthält Regelungen zum Outsourcing von IT-Anwendungen und -Systemen.

„Bei Anwendung des IT-Grundschutzhandbuch wird aber nur ein Soll-Ist-Vergleich zwischen empfohlenen und bereits realisierten Maßnahmen durchgeführt. Dabei festgestellte fehlende und noch nicht umgesetzte Maßnahmen zeigen die Sicherheitsdefizite auf, die es durch die empfohlenen Maßnahmen zu beheben gilt.“ [BSI] Das Gesamtsystem wird in den folgenden Abschnitten als bereits umgesetzt angenommen. Dabei dient das Grundschutzhandbuch als Leitfaden zur Erstellung eines Modells und der Infrastruktur. Anschließend wird der Schutzbedarf für alle Komponenten ermittelt. Diese lassen sich in die Ebenen Server, Anwendung und Kommunikation einteilen und werden in Abschnitt 4.5 bis 4.7 erläutert.

Das in Abbildung 4.4.1 dargestellte Modell umfasst die bereits in Abschnitt 4.1 erläuterten Komponenten der Präsentations-, Funktionalitäts- und Datenebene und deren Beziehungen zueinander. Zusätzlich erscheint der Internet Information Server, welcher aufgrund der in Abschnitt 4.2 genannten Eigenschaften grundlegend, jedoch nicht zwingend, für die Umsetzung der ASP.NET Web-Services und der ASP.NET-Webanwendung (www.worldcheckinn.com) ist. Die hotelspezifischen Web-Services und der Browser des Clients sind nicht in der IT-Strukturanalyse enthalten. Damit gemeint sind auch zusätzlich erforderliche technische Maßnahmen auf allen Clients,

wie zum Beispiel Antivirus-Lösungen, Router mit integrierter Firewall oder der lokale Schutz von Datenbanken.

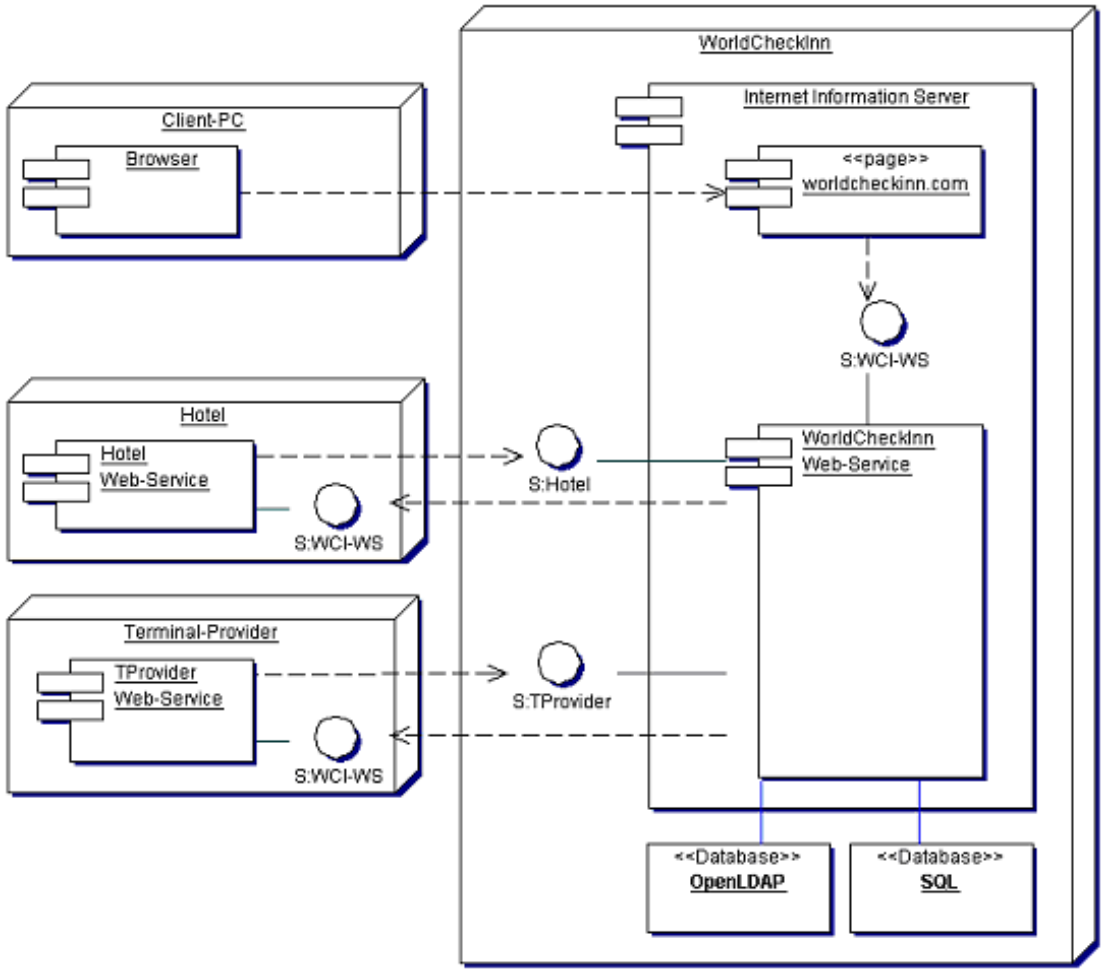

Abbildung 4.4.1: Beziehungen der Komponenten des Gesamtsystems

Zur Festlegung der Outsourcing-Strategie wird zunächst eine IT-Strukturanalyse erstellt und alle Anwendungen (Tabelle 4.4.1) werden erfasst. Anschließend wird der Schutzbedarf ermittelt. Der Schutzbedarf wird in Vertraulichkeit, Integrität und Verfügbarkeit untergliedert und kann in drei Kategorien eingeordnet werden. Bei der ersten Kategorie, „niedrig bis mittel“, sind Schadensauswirkungen begrenzt und überschaubar. Sind die Schadensauswirkungen beträchtlich, so ist der Schutzbedarf „hoch“. Als existentiell bedrohlich werden die Anwendungen der Kategorie, „sehr hoch“, betrachtet.

Anw. -Nr.	IT-Anwendung/ Informationen	Pers.bezogene Daten
A1	SQL-Datenbank	X
A2	OpenLDAP-Verzeichnisdienst	X
A3	Internet Information Server	-
A4	ASP.NET Webanwendung	X
A5	ASP.NET Web-Service	X

Tabelle 4.4.1: Erfassung der IT-Anwendungen und der zugehörigen Informationen

Nr.	Bezeichnung	Pers. Daten	Grundwert	Schutz-bedarf	Begründung
A1	SQL-Datenbank	X	Vertraulichkeit	Sehr hoch	Reservierungsdaten stehen immer in Bezug zu mind. einer Person und sind somit besonders schutzbedürftige Daten, deren bekannt werden die Betroffenen erheblich beeinträchtigen kann
			Integrität	Hoch/ Sehr hoch	Reservierungsinformationen können mit der Datenbank des Hotels abgeglichen werden
			Verfügbarkeit	Hoch	Häufige Änderungen, jedoch ist ein Betrieb des Systems vorübergehend weiter möglich
A2	OpenLDAP - Verzeichnisdienst	X	Vertraulichkeit	Sehr hoch	Personendaten sind besonders schutzbedürftige personenbezogene Daten, deren bekannt werden die Betroffenen erheblich beeinträchtigen kann

Nr.	Bezeichnung	Pers. Daten	Grundwert	Schutz-bedarf	Begründung
			Integrität	Sehr hoch	Die Veränderung von Personendaten wirken sich nur gering aus, sensibel sind aber Preisangaben und abrechnungsrelevante Daten (z.B. Kontodaten)
			Verfügbarkeit	Sehr Hoch	Finanzieller Verlust, da bei Ausfall normaler Check-In notwendig ist
A3	Internet Information Server		Vertraulichkeit	Hoch	Konfigurationsdateien dürfen nicht eingesehen werden
			Integrität	Hoch	Änderungen in den Konfigurationsdateien können den Ausfall wichtiger Sicherheitsvorkehrungen bewirken, ist aber durch Zugriff von außen (Ausnahme:Remotezugriff) nicht möglich
			Verfügbarkeit	Sehr Hoch	Ausfall bewirkt Ausfall des gesamten Systems und somit finanziellen Verlust
A4	ASP.NET Webanwendung	X	Vertraulichkeit	Sehr hoch	Vortäuschung einer falschen Identität kann weitreichende finanzielle Auswirkungen für den Betroffenen haben
			Integrität	Hoch	Vgl. A3
			Verfügbarkeit	Mittel	Unterbrechungen wirken sich nicht auf den Abruf von Daten aus (Check-In), jedoch sind finanzielle Verluste möglich, da keine Reservierungen möglich sind
A5	ASP.NET Web-Service	X	Vertraulichkeit	Hoch	Konfigurationsdateien und Quellcode dürfen nicht eingesehen werden
			Integrität	Hoch	Vgl. A3

Nr.	Bezeichnung	Pers. Daten	Grundwert	Schutz-bedarf	Begründung
			Verfügbarkeit	Sehr hoch	Ein Ausfall des Web-Services verhindert jegliche Transaktionen

Tabelle 4.4.2: Schutzbedarfsermittlung der IT-Anwendungen

Anhand Tabelle 4.4.2 wird deutlich, dass das Gesamtsystem WorldCheckInn stark abhängig vom dem ASP.NET Web-Service und der zugrunde liegenden Datenhaltung ist. Hier müssen besondere Vorkehrungen getroffen werden, etwa die Einrichtung eines zweiten Web-Services auf einem anderen IT-System oder eine alternative Schnittstelle für die ASP.NET Webanwendung zum Zugriff auf den Verzeichnisdienst oder die SQL-Datenbank. In Abbildung 4.4.2 wird noch einmal die Position des WorldCheckInn Web-Services als Schnittstelle zu allen Komponenten des Gesamtsystems verdeutlicht. Er muss dementsprechend mit der zugrunde liegendenden Datenhaltung redundant ausgelegt werden.

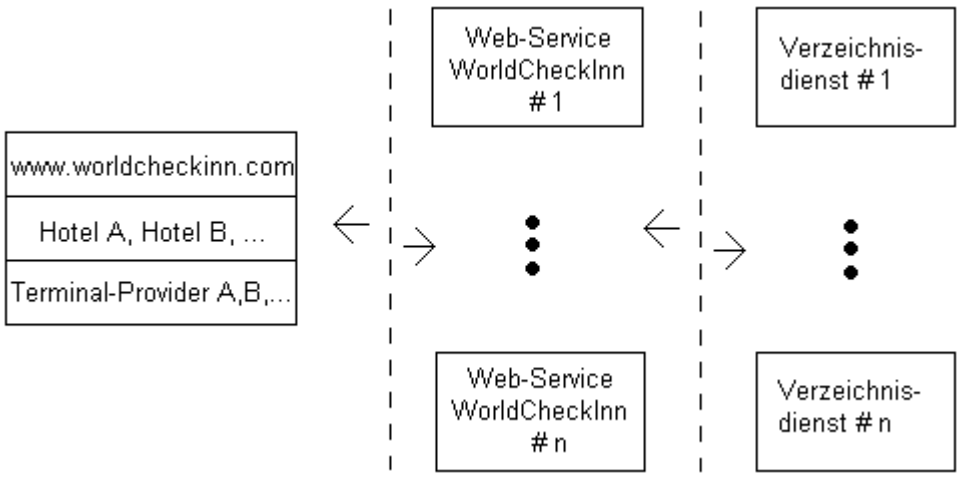

Abbildung 4.4.2: Redundante Auslegung des Web-Services und der Datenhaltung

Der Schutzbedarf des Gesamtsystems WorldCheckInn ist als sehr hoch anzusehen. Der Schutzbedarf der Clients, zum Beispiel in Hotels, kann nicht vorausgesetzt werden, vielmehr muss eine vertragliche Regelung in Kraft treten, damit ein Mindestschutz umgesetzt werden kann. Eine erfolgreiche Absicherung kann allerdings nicht, beziehungsweise nur eingeschränkt kontrolliert werden, zum Beispiel durch Penetrationstests auf diese. Auch muss abgewogen werden, inwiefern sich der Schaden an Clients bzw. Servern in anderen Hotels auf das System WorldCheckInn auswirkt.

4.5 Sicherheit auf Serverebene

Gefährdungen und Maßnahmen des IIS sind im Kapitel 7 „Datenübertragungseinrichtungen“ mit den entsprechenden Verweisen im Gefahren- und Maßnahmenkatalog zusammengefasst. Im Folgenden werden Aufgaben zum Sichern des IIS 6.0 auf Basis des Windows Servers 2003 Enterprise Edition erläutert. Dabei werden zur Orientierung relevante Punkte des Grundschutzhandbuches aufgeführt und durch eine Prüfliste der Microsoft TechNet-Gruppe Sicherheit [MSTN] erweitert. Zur Vervollständigung werden zu jedem Schritt weitere Informationen und in einigen Punkten Verweise auf weiterführende Quellen angegeben.

1. Absicherung der Administrator- und Benutzerkonten beim IIS-Einsatz

Standardkonten des Windows Servers 2003 und des IIS sind die des Administrators und des Gastes. Diese können nicht gelöscht, aber umbenannt oder deaktiviert werden. Es ist sicherzustellen, dass das Gast-Konto deaktiviert ist (Standardeinstellung). Wird das Konto Administrator umbenannt, so ist ein komplexer Name zu wählen, der keinen Rückschluss auf die Bedeutung des Kontos zulässt. Analog dazu sollte die Beschreibung des Kontos geändert werden und ein komplexes Passwort gewählt werden.

Ein weiterer Standard ist das Konto IUSR_MACHINE. Ist der anonyme Zugriff aktiviert, so wird keine Authentifizierung verlangt, sondern ein Zugriffs-Token für jeden Anwender angelegt, um auf dieses Konto zuzugreifen. Sofern es nicht genutzt

wird, sollte es deaktiviert sein [SecGuide]. Wird der anonyme Zugriff dennoch verwendet, so ist sicherzustellen, dass das Konto IUSR_MACHINE die minimal benötigten Rechte hat.

Aufgrund ihrer Zugriffsrechte fällt den Administratorkonten ein besonderes Augenmerk zu. Es sollte nur eine kleine Anzahl (1–2) dieser existieren. Zwischen Administrator und Anwender besteht eine strikte Trennung, das heißt, Anwender und Administratoren dürfen keine gemeinsamen Konten verwenden.

2. Schutz von sicherheitskritischen Dateien bei Einsatz des IIS

Administratorwerkzeuge, wie zum Beispiel *ftp.exe* zum Starten eines FTP-Servers (File Transfer Protocol), sollten sich nicht im Wurzel-Verzeichnis (*Root*) des Web-Servers befinden. Seit Version 6.0 des IIS sind Administratorwerkzeuge standardmäßig in einem anderen Verzeichnis hinterlegt.

3. Überwachen des IIS-Systems

Die Überwachung (Protokollierung) der Zugriffe auf das System dient der nachhaltigen Erkennung von Angriffen. Für die Protokollierung wird das W3C-Format, ein Standard des W3C, vorgeschlagen [MSTN]. Die Protokolldateien sind wiederum durch NTFS-Berechtigungen (New Technology File System) oder eine ACL (Access Control Lists) zu sichern. Zur Überwachung des Systems gehört auch die Archivierung sowie die regelmäßige Überprüfung der Protokolldateien.

4. Sicherstellen der Verfügbarkeit und Performance

Zur Kontrolle der Performance bietet sich die Installation und Konfiguration von MOM-Agents oder einer ähnlichen Überwachungslösung an. Hinter MOM (Microsoft Operations Manager) verbirgt sich ein Manager zur Verwaltung und Überwachung von komplexen IT-Systemen [MSTN]. Dabei fällt das besondere Augenmerk auf die Überwachung der Performance und Verfügbarkeit sowie auf das schnelle Beheben von Fehlern. Die erfolgreiche und schnelle Fehlerbehandlung wird anhand von Events (dt.: Ereignisse) durchgeführt. Diese Events überwachen den Zustand der Windows- Umgebung, führen Analysen durch und tragen somit zur Fehlerbehandlung bei.

Allerdings gestaltet sich die richtige Konfiguration der Events als sehr komplex, da von jeder Anwendung und dem zugrunde liegenden Betriebssystem Events ausgelöst werden. Um nicht in einer Flut von Informationen zu versinken und somit wichtige kritische Ereignisse zu übersehen, liegt es in der Aufgabe des Administrators den vorhandenen Filter vorab richtig zu konfigurieren. Mit Hilfe der Filter kann bereits im Vorfeld das Augenmerk auf besonders wichtige Ereignisse gelegt werden, zum Beispiel „Backup fehlgeschlagen“. Alternativ stehen auch weitere Tools, wie zum Beispiel der IIS Tracer zur einfacheren und komfortableren Überwachung von Webanwendungen zur Verfügung (http://iismonitor.motobit.com/).

Zum Sicherstellen der Verfügbarkeit gehört auch das Abwehren von DOS-Angriffen, zum Beispiel SYN-Flooding. Hier kann die Implementierung von IPSec-Filtern Abhilfe schaffen. IPSec dient zur Verschlüsselung der Netzwerkkommunikation und stellt die Integrität, Authentizität und die Vertraulichkeit sicher. Für den Windows Server 2003 steht eine IPSec Implementierung zur Verfügung, die zusätzlich als Paketfilter verwendet werden kann. Mit Hilfe eines Paketfilters kann der ein- und ausgehende Datenverkehr geprüft werden. Durch Festlegen von Filterregeln wird verhindert, dass unerwünschte Pakete, zum Beispiel unerwünschte IP-Adressen, die Zieladresse erreichen. Dazu steht unter anderem der „IP Security Policy Wizard“ (Windows Server 2003) und das Kommandozeilenwerkzeug IPSecCMD zur Verfügung. Der „IP Security Policy Wizard“ stellt ein grafisches Front-End zur Verfügung. Analog zur Festlegung von Filterregeln sollte der Sicherheits-konfigurations-Assistent [SCW] in folgendem Sinne verwendet werden:

- Blockierung von ungenutzten Ports
- Absicherung von offenen Ports über IPSec
- Import von Windows-Sicherheitsvorlagen für Einstellungen, die nicht vom Assistenten abgedeckt werden

5. Deaktivieren nicht benötigter Dienste beim IIS-Einsatz

Alle Windows-Dienste, die nicht verwendet werden, sollten deaktiviert sein. Dies betrifft die Dienste:

- FTP
- SMTP
- NNTP
- Telnet

Sofern keine FrontPage-Erweiterungen zur Verwaltung von Websites benötigt werden, ist dieser Dienst ebenfalls zu deaktivieren. Dazu steht erneut der Sicherheits-konfigurations-Assistent [SCW] zur Verfügung. Er unterstützt unter anderem das Deaktivieren nicht benötigter Dienste sowie das Deaktivieren nicht benötigter IIS-Weberweiterungen.

6. Absichern von virtuellen Verzeichnissen und Web-Anwendungen beim IIS-Einsatz

Virtuelle Verzeichnisse vermeiden das (un)beabsichtigte Löschen von Dateien. Diese Verzeichnisse dienen im Wesentlichen dazu, Angriffe wie zum Beispiel *Path Traversal* abzuwehren, wenn die Formulare einer Webseite unzureichend geschützt oder Formulareingaben nicht ausreichend geprüft werden. Andererseits verhindern sie das Herunterladen der kompletten Datenbank der Webanwendung, wenn sich diese im Wurzelverzeichnis (C:\Inetpub\wwwroot) befindet. ASP.NET Anwendungen sollten immer in virtuellen Verzeichnissen hinterlegt werden. Damit erspart sich der Entwickler zusätzlichen Aufwand, um wichtige System- oder Konfigurationsdateien gegenüber Angreifern zu schützen. Seine Aufgabe ist es, die virtuellen Verzeichnisse anzulegen. Dazu wird ein Link an einer bestimmten Stelle im Server-Verzeichnis angelegt, der es Anwendern erlaubt, auf die Dateien des verlinkten Ordners zuzugreifen.

Diese Verlinkung erlaubt es, Verzeichnisse unterhalb des Wurzelverzeichnisses zur Verfügung zu stellen. Diese können sich jedoch an gänzlich anderer Stelle befinden, sogar auf einem anderen Server im Netzwerk. Dadurch wird erreicht, dass sich Webseiten oder Teile einer Seite auf einer anderen Partition befinden, die nicht Bestandteil des Systems ist.

Bei dem Anlegen der virtuellen Verzeichnisse müssen Lese- und Schreibzugriffe durch den Entwickler gesetzt werden. Einerseits sollten Verzeichnisse, auf die in *Include*-Anweisungen verwiesen wird, keine Lese- und Schreibberechtigung für (Web-)Anwender haben. Die eingeschränkte Lese- und Schreibberechtigung ist auch bei virtuellen Verzeichnissen erforderlich, die einen anonymen Zugriff ermöglichen. Andererseits sollten Schreibzugriffe generell nur in Ordnern möglich sein, die eine Authentifizierung des Anwenders voraussetzen und zum Beispiel für einen Upload von Bildern oder Texten zur Verfügung stehen [SecGuideB].

Mangelnde Eingabeüberprüfungen, wie bei SQL-Injection oder XSS, haben bei *Path Traversal*-Angriffen unter Umständen gravierendere Auswirkungen, wenn diese zum Beispiel das Löschen wichtiger Systemdateien zur Folge haben. Ein *Path Traversal*-Angriff ist dann erfolgreich, wenn bei einer Eingabe unerwartete, relative Pfadangaben verwendet werden. Als Beispiel kann hier ein Forum herangezogen werden, welches Benutzerprofile verwaltet. Es besteht die Möglichkeit, eigene Bilder für das Profil hoch zu laden und diese auch wieder zu löschen. Das Löschen erfolgt mittels eines einfachen Befehls, wie in Codebeispiel 4.5.1 dargestellt:

```
using System.IO.File;
...
string sDateiname = textBox1.Text;
File.Delete("/userForum/Files/Upload/" + sDateiname)
```

Codebeispiel 4.5.1: Eine einfache Anweisung zum Löschen von Dateien

Durch die Eingabe von `../secret/config.ini` sieht die Zeichenkette wie folgt aus:

```
File.Delete("/userForum/Files/Upload/../secret/config.ini")
```

Codebeispiel 4.5.2

Nach Auflösung der relativen Pfadangabe ergibt sich:

```
File.Delete("/userForum/Files/secret/config.ini")
```

Codebeispiel 4.5.3

Wie in diesem Beispiel veranschaulicht wird, kann es dem Angreifer gelingen, wichtige Systemdateien ohne Administratorrechte zu löschen oder zu überschreiben. *Path Traversal*-Angriffe können aber bereits am IIS durch das Deaktivieren der Einstellung „Übergeordnete Pfade“ (Abbildung 4.5.1) verhindert werden (Eigenschaften des Stammverzeichnisses einer Webanwendung).

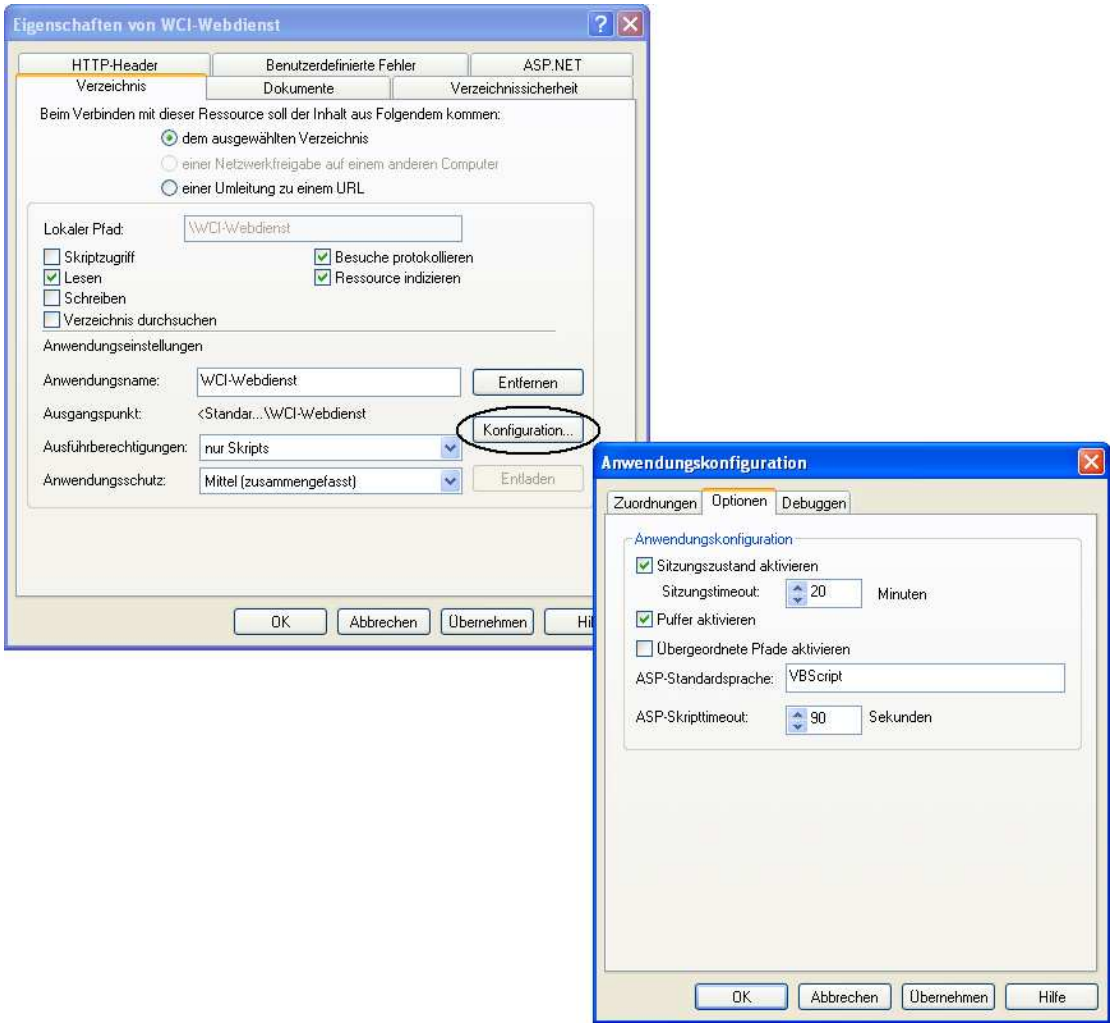

Abbildung 4.5.1: Deaktivieren der Einstellung „Übergeordnete Pfade“

7. Entfernen von Beispieldateien, Administrations-Scripts und der RDS-Unterstützung des IIS

Nach der Installation des ISS existieren die Verzeichnisse IISSamples, IISAdmin, IISHelp und MSADC, welche Beispieldateien und Scripts enthalten. Scriptdateien können dazu ausgenutzt werden, um Informationen über den Web-Server zu erhalten. Diese und andere nicht notwendige virtuelle Verzeichnisse mit Scriptdateien sind zu entfernen. Als Beispiel wäre die Scriptdatei *showcode.asp* zu nennen. Diese Scriptdatei enthält eine Schwachstelle, mit deren Hilfe Angreifer in der Lage sind, sich Dateien vom Web-Server anzeigen zu lassen. Diese Schwachstelle ist in der Version 4.0 des IIS vorhanden und wurde ab Version 6.0 behoben. Weitere Beispielverzeichnisse sind unter [BSI02] aufgelistet. Unter [MSTN01] sind Verzeichnisse aufgelistet, die andere Versionen der *showcode.asp* enthalten.

Die Komponente RDS (Remote Data Services) ermöglicht einen Fernzugriff über das Internet auf Datenbankressourcen des IIS. Zum Entfernen dieser Komponente reicht es nicht aus, die Zuordnung zum virtuellen Verzeichnis MSDAC zu entfernen. Zusätzlich müssen alle Dateien und Unterverzeichnisse dieser Komponente, zu finden in (*...\Programme\Gemeinsame Dateien\System\MSDAC*), und der Registrierungsschlüssel in (*HKLM\System\Current ControlSet\Services\W3SVC\ Parameters\ADCLaunch)* entfernt werden.

Im Allgemeinen sollten Fernzugriffe nur eingeschränkt verfügbar sein. Das Benutzerrecht „Auf diesen Computer vom Netzwerk aus zugreifen“ sollte der Gruppe „Jeder“ entzogen werden. Der Windows Server 2003 kann per Fernzugriff verwaltet und konfiguriert werden. Ist kein sicherer Zugriff möglich, ist immer eine lokale Anmeldung des Administrators vorzuziehen.

8. Anwenden aller erforderlichen Service Packs und/oder Updates

Unter http://www.microsoft.com/technet/security/tools/default.mspx findet sich eine Reihe von Werkzeugen, mir deren Hilfe Windows-basierte Systeme immer auf dem aktuellen Sicherheitsstandard gehalten werden können. Diese Werkzeuge verschleiern jedoch die Tatsache, dass der Aufwand, ein System durch Updates oder Patches sicher zu halten, analog zur Komplexität des Systems steigt. Daher bietet sich der Microsoft Baseline Security Analyzer (MBSA) an, welcher für jede

Einzelkomponente des Gesamtsystems in festgelegten Zeitabständen Überprüfungen auf das Vorhandensein aktueller *Updates*, *Hotfixes* oder *Patches* durchführt.

Dieses Werkzeug und andere Überwachungsprogramme geben allerdings keine vollständige Garantie, immer *Up-to-date* zu sein. Sie bieten jedoch eine komfortable und einfache Möglichkeit zur sicheren Konfiguration und Aktualisierung aller Serverkomponenten.

9. Genehmigung für Kontoübertragungen

Benutzerkonten des Windows Servers 2003 können an andere Anwender übertragen werden. Für solche Kontoübertragungen sollte eine Genehmigung angefordert werden. Daher ist die Markierung der Domänenkonten-Eigenschaft ‚für Delegierungszwecke vertrauenswürdig' aufzuheben.

10. Sichern von Dienstkonten

Konten, die berechtigt sind aktivierte Windows-Dienste zu nutzen, sollten minimale Berechtigungen zugeordnet sein. Dies soll den Schaden minimieren, den ein Angreifer anrichten könnte, wenn er über diesen Dienst Betriebssystembefehle ausführen kann oder sich über unentdeckte Sicherheitslücken Zugriff auf das System verschafft. Daher ist sicherzustellen, dass aktivierte Dienste nur mit Konten ausgeführt werden, welche die geringsten Berechtigungen haben.

11. IIS-Authentifizierung

Verwendet man die Authentifizierungsmechanismen des IIS, so stehen folgende Schemen zur Verfügung [IIS]:

- Standardauthentifizierung
- Digest-Authentifizierung
- Integrierte Windows-Authentifizierung
- Zuordnung von Client-Zertifikaten

Als Standard ist die anonyme Authentifizierung vorgesehen. Bei dieser Form der Authentifizierung werden keine Anmeldeinformationen des Clients übergeben, es findet also keine Clientauthentifizierung statt. Die Standardauthentifizierung setzt

voraus, dass der Anwender den Internet Explorer auf einem Windows-Betriebssystem verwendet. Die Authentifizierung läuft im Hintergrund zwischen Browser und Server ab und verwendet den Login und das Passwort, mit dem sich der Anwender bereits am PC angemeldet hat. Allerdings werden die Informationen nur mit Base64 codiert [MSDN] und bieten daher keinen ausreichenden Schutz. Die Digest-Authentifizierung sowie die integrierte Windows-Authentifizierung verwenden einen Hashwert. Diese Mechanismen bieten jedoch kaum einen besseren Schutz. Ohne den Einsatz von SSL/TLS, kann ein Angreifer die Kommunikation zwischen Server und Client aufzeichnen. Die Verwendung von Zertifikaten ermöglicht die Zuordnung eines oder mehrerer Zertifikate zu einem Benutzerkonto. Diesem strengen Authentifizierungsschema steht allerdings eine aufwendige Konfiguration gegenüber.

Die Standardauthentifizierung, Digest-Authentifizierung, integrierte Windows-Authentifizierung und die Zuordnung von Clientzertifikaten setzen allerdings ein Windows-Benutzerkonto voraus und sind daher eher für Intranet-Lösungen gedacht. Es ist also notwendig, für außenstehende Verbindungen ein eigenes System für die Authentifizierung zu entwickeln (Abschnitt 4.6.2 „Individuelle Authentifizierungs-mechanismen“) oder auf ASP.NET-Authentifizierungsmethoden zurückzugreifen. Für all diese Mechanismen stellt der IIS die Verschlüsselung mittels SSL/TLS zur Verfügung. Diese Verschlüsselung kann auch bei der Entwicklung eigener Authentifizierungsmechanismen integriert werden.

Anwender ohne Benutzerkonto haben keinen Eintrag in der NTFS-Dateizugriffsliste (Access Control List, ACL). Anhand einer ACL wird überprüft, ob das authentifizierte Benutzerkonto auf die Ressourcen zugreifen darf. Wird ein eigenes System für die Authentifizierung entwickelt, so muss auch die Autorisierung selbst definiert werden.

4.6 Sicherheit auf Anwendungsebene

Auf Serverebene können Anforderungen an die lokale Administration gestellt werden. *Application Hosting* bedeutet, dass eigene Anwendungen auf der IT eines Providers ausgelagert werden, dieser jedoch nicht für die Korrektheit und Sicherheit der Anwendungen zuständig ist. Abbildung 4.6.1 zeigt alle Komponenten der Anwendungsebene unter WorldCheckInn.

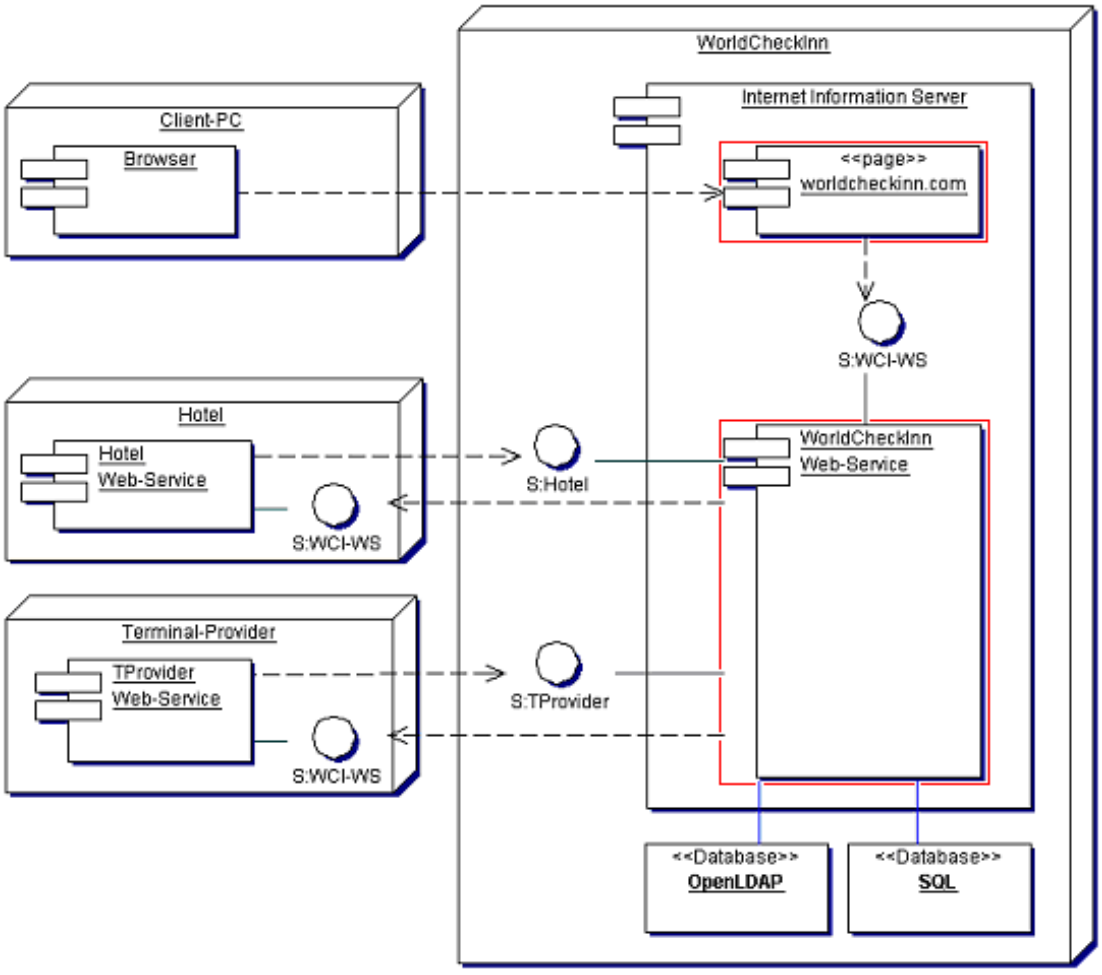

Abbildung 4.6.1: Komponenten der Anwendungsebene

Prinzipiell ist es möglich, ASP.NET Anwendungen auch ohne den Internet Information Server zu betreiben. In Bezug auf die Sicherheit sind ASP.NET Anwendungen jedoch eng mit dem IIS verbunden [Wey02]. Von Bedeutung sind die Unterstützung von Secure Sockets Layer (SSL), die „Beschränkung des Zugriffs auf Ressourcen auf Basis von IP-Adressen oder Domänenamen“ [Wey02], virtuelle Verzeichnisse und Zugriffskontrolllisten, welche die Berechtigung des jeweiligen Benutzers beim Zugriff auf die Ressourcen prüfen.

4.6.1 Authentifizierung und Autorisierung unter .NET

Bei der Authentifizierung wird die Identität eines Benutzers anhand eines oder mehrerer Merkmale geprüft. Eine erfolgreiche Authentifizierung soll eine Garantie dafür geben, dass der Anwender derjenige ist, für den er sich ausgibt.

Ist der Vorgang der Authentifizierung erfolgreich abgeschlossen, so wird unter .NET ein Principal-Objekt erstellt, welches einen Verweis auf ein Identity-Objekt enthält (Abbildung 4.6.2).

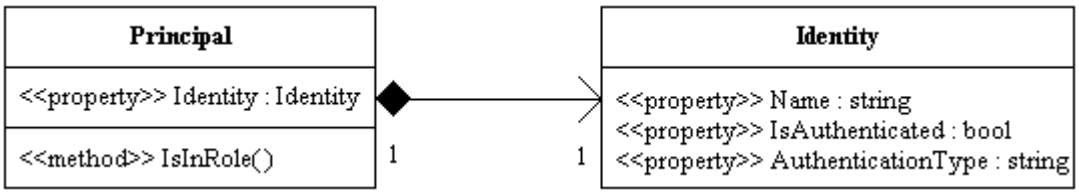

Abbildung 4.6.2: Principal-Objekte unter .NET

Das Principal-Objekt enthält die Identität (Identify-Object) und Rollenmitgliedschaft(en) des authentifizierten Benutzers. Die Rollen eines Objektes können mit der öffentlichen Methode `IsInRole` abgefragt werden. Das Objekt wird an den aktuellen Thread (ein einzelner Ausführungspfad, Abbildung 4.6.3) übergeben. Dadurch steht es für den gesamten Thread zur Verfügung und kann durch interne Methoden angesprochen werden.

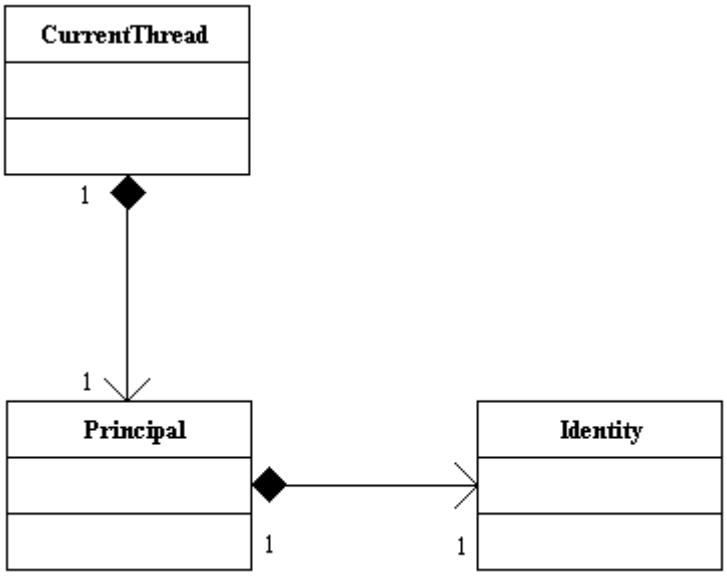

Abbildung 4.6.3: Zugehörigkeit des Principal zum aktuellen Thread

4.6.2 ASP.NET Authentifizierung

Formularbasierte Authentifizierung

Nach einer erfolgreichen Authentifizierung wird ein *Cookie* erzeugt, welches Authentifizierungsmerkmale enthält. Das *Cookie* wird im *Header* (dt.: Kopf) der Webseite hinterlegt. Vorteilhaft wirkt sich dabei aus, dass die formularbasierte Authentifizierung einfach konfigurierbar ist.

Eine aktuelle Statistik von [webhits] in Abbildung 4.6.4, zeigt, dass die Verwendung von *Cookies* bei 98 Prozent aller Anwender aktiviert ist.

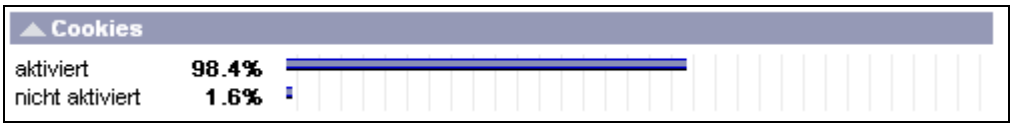

Abbildung 4.7: Verwendung von Cookies

Der Verwendung von *Cookies* steht die Bedrohung durch *Cookie-Manipulation, One-Click*-Angriffen oder anderen Formen von Angriffen auf die Speicherung von *Cookies* gegenüber. *Cookie-Manipulation* gehört in die Kategorie des *Session-Hijacking* (dt.: Sitzungs-Übernahme).

Session-Hijacking ist die Übernahme einer Verbindung durch einen Angreifer. Diese Form des Angriffes kann mit den entsprechenden Tools, zum Beispiel *Juggernaut*, leicht durchgeführt werden. Eine Session (dt.: Sitzung) enthält einen eindeutigen Identifikator, die SID. Angriffspunkt bei *Session-Hijacking* ist entweder die Session-ID oder ein *Cookie*. Ist ein Angreifer in Besitz einer SID gelangt, die jedoch bereits einer authentifizierten Session zugeordnet ist, kann er sich ohne Kenntnis des Passwortes als der betroffene Benutzer ausgeben.

Eine Form des Angriffes durch *Session-Hijacking* ist *Session Fixation*, das Einschleusen eines gefälschten *Cookies* auf einem Computer. Der Angreifer startet eine eigene, nicht authentifizierte Sitzung am Server. Anschließend schleust er das *Cookie*, zum Beispiel mit Hilfe eines Trojaners, in den Browser des Betroffenen ein. Ruft der Betroffene die Website auf, so wird die bereits vorhandene Sitzung weitergeführt. Während der Session des Betroffenen kann die zuvor wertlose, eingeschleuste SID als temporäres Passwort missbraucht werden [Jov04].

Daraus ergeben sich folgende Mindestanforderungen für die formularbasierte Authentifizierung:

- **Verwendung verschlüsselter Verbindungen mittels SSL**: Der Programmierer hat dafür Sorge zu tragen, das die SID nur über einen verschlüsselten Kanal übertragen wird. Es ist nicht ausreichend, einzig das Passwort während des Authentifizierungsvorganges zu schützen. Daher ist es notwendig, auf der gesamten Internetseite nach erfolgreicher Authentifizierung die Verbindung per SSL zu verschlüsseln.
- **Verwendung einer neuen SID:** Es erweist sich als sinnvoll, nach erfolgreicher Authentifizierung eine neue SID als Ersatz für die bisherige zu generieren. Ist eine verschlüsselte Verbindung nicht möglich, stellt das regelmäßige Wechseln der SID eine zusätzliche und sogar notwendige Möglichkeit dar, einen Angriff zu erschweren.
- **Kurzlebigkeit von Cookies:** *Cookies* geben dem Angreifer die Möglichkeit, sich als authentifizierte Person auszugeben. Besteht diese Möglichkeit, so

kann er alle Aktionen ausführen, wofür das Opfer autorisiert ist. Ein Objekt der Klasse `HttpCookie` besitzt die Eigenschaft `Expires`, mit der man die Lebensdauer des *Cookies* beschränken kann:

```
HttpCookie cookie;
// Cookie des Anwenders anfordern und zuweisen:
cookie = FormsAuthentication.GetAuthCookie(user, isPersistent)
cookie.Expires = DateTime.Now.AddDays(1); // Lebensdauer des Cookies
ändern
```

Codebeispiel 4.6.1: Unter ASP.NET die Gültigkeit von Cookies ändern

Seit ASP.NET 2.0 ist es möglich, die SID in der URL einzubetten (siehe auch URL-Autorisierung). Auf diesem Wege kann man auf „Cookies“ verzichten. Diese Methode ist allerdings ohne Verschlüsselung für jedermann sichtbar.

.NET Passport

ASP.NET unterstützt den Microsoft *.NET Passport*-Authentifizierungsdienst. *.NET Passport* ist ein Service, über den Anwender mit einer Benutzerkennung (einer registrierten E-Mail-Adresse) und einem Kennwort problemlos mit nur einer Anmeldung auf alle *Passport*-geschützten Services des Microsoft Network (MSN) und seinen Partnern zugreifen können. Microsoft *.NET Passport* ist ein Dienst, der Entwicklern die Möglichkeit bietet, ein *Outsourcing* der Benutzerauthentifizierung durchzuführen. Die Angabe zusätzlicher persönlicher Daten ist freiwillig [passport]. Da Anwender mit einer Identität alle Dienste nutzen, besteht die Möglichkeit, umfassende Kundenprofile zu erstellen.

Individuelle Authentifizierungsmechanismen

Für individuelle Authentifizierungsmechanismen stellt ASP.NET mehrere Klassen unter dem Namensraum `System.Web.Security` zur Verfügung. Dieser enthält Klassen, mit denen die Sicherheit für ASP.NET in Webserveranwendungen implementiert wird. Für die Erstellung einer individuellen Authentifizierung erlaubt das .NET Framework benutzerdefinierte Rollen, Identitäten und *Principals* zu definieren. Dem Entwickler stehen zwei verschiedene Arten von *Principal*-Objekten zur Verfügung.

Das *Generic-Principal* ermöglicht in einem hohen Maße benutzerdefinierte und plattform-unabhängige Autorisierungsszenarien. Für das *Windows-Principal* nutzt das .NET Framework das traditionelle Windows-Sicherheitssystem und ordnet vorhandene Windows- Benutzerkonten und -Gruppen zu Rollen hinzu.

Der allgemeine Ablauf ist:

- Erfassen der Anmeldeinformationen des Benutzers
- Überprüfen der Anmeldeinformationen anhand eines benutzerdefinierten Datenspeichers, z.B. einer Datenbank
- Abrufen der Rollen, welche dem Benutzer zugeordnet sind
- Erstellung eines *Principal*-Objektes, in dem die Rollen und die Identität des Benutzers abgelegt werden
- Zuordnung des *Principal*-Objektes zum Kontext der Anwendung

4.6.3 Autorisierung unter .NET

Die Voraussetzung für die folgenden Autorisierungsmechanismen ist gegeben, wenn die Benutzeridentität in Form eines *Principal*-Objektes vorliegt. Im .NET Framework stehen vier verschiedene Autorisierungsmechanismen zur Verfügung: rollenbasierte, imperative-, deklarative- und URL-Autorisierung.

Rollenbasierte Sicherheitsprüfung

Das in Abschnitt 4.6.2 vorgestellte *Principal*-Objekt enthält die Methode `IsInRole`. Mittels dieser Methode kann abgefragt werden, ob der Benutzer einer bestimmten Rolle angehört. Mit der statischen Eigenschaft `CurrentPrincipal` wird auf das *Principal*-Objekt des aktuellen Threads zugegriffen.

```
if(System.Threading.Thread.CurrentPrincipal.IsInRole("Manager")) {
// Benutzer ist berechtigt Managerfunktionen auszuführen
} else { // Benutzer ist kein Manager!}
```

Codebeispiel 4.6.2: Verwendung von Principal-Objekten

Die imperative bzw. deklarative Sicherheitsüberprüfung wird mit einem *PrincipalPermission*-Objekt durchgeführt. Das Objekt stellt dabei die Identität und Rolle dar, über die der aktuelle Benutzer verfügen muss, um den Code auszuführen.

Imperativ

Im Codebeispiel 4.6.3 wird eine *PrincipalPermission* angelegt, welche fordert, dass der Benutzer der Rolle „Manager" angehört und dass es sich um die Identität „Hans" handelt. Wahlweise kann der erste oder letzte Parameter beziehungsweise beide Parameter weggelassen werden. Die Methode `Demand` bestimmt zur Laufzeit, ob das aktuelle *Principal* mit dem durch die aktuelle Berechtigung angegebenen *Principal* übereinstimmt [MSDN]. Stimmen die Berechtigungen nicht über ein, wirft die CLR eine `SecurityException` (`System.Security.Security Exception`).

```
PrincipalPermission guestPermission = new PrincipalPermission("Hans",
"Manager");
// ...
guestPermission.Demand(); // Test, ob Berechtigungen übereinstimmen
```

Codebeispiel 4.6.3: Imperative Sicherheitsüberprüfung

Deklarativ

Deklarativ wird bereits zur Übersetzungszeit festgelegt, wer welchen Code ausführen darf. Das Codebeispiel 4.6.4 veranschaulicht, dass nur Benutzer mit dem Namen „Hans" und der Rolle „Manager" berechtigt sind, den darunter liegenden Code auszuführen. Analog dazu besteht die Möglichkeit, zu testen, ob der Benutzer bereits authentifiziert ist.

```
[PrincipalPermissionAttribute(SecurityAction.Demand, Name="Hans",
 Role="Manager")]
public void secretMethod() {
// Code darf nur von einem Manager mit dem Namen Hans ausgeführt werden
}
// oder
[PrincipalPermissionAttribute(SecurityAction.Demand,
Authenticated=true)]
// = True, wenn der aktuelle Principal authentifiziert wurde
```

Codebeispiel 4.6.4: Deklarative Sicherheitsüberprüfung

Weil imperative Befehle bei jedem Methodenaufruf ausgeführt und deklarative Befehle nur einmal zum Ladevorgang ausgewertet werden, ist durch letztere eine Verbesserung der Performance gegeben [MSDN].

.NET stellt Tools zur Verfügung, mit deren Hilfe Administratoren im Nachhinein erkennen können, welche Sicherheitsberechtigungen von bestimmten Klassen oder Methoden gefordert werden. Unter dem Namen *.NET Admin Tool* ist ein solches gegeben, welches die geforderten Berechtigungen aus den Programmbausteinen auslesen kann.

URL-Autorisierung

Anhand der URL-Autorisierung ist eine feinstufigere Autorisierung möglich. Es können bestimmte Identitäten für einzelne Seiten oder Bereiche einer Internetanwendung autorisiert werden [Fr03]. Dafür steht unter ASP.NET das `<authorization>` -Element zur Verfügung, mit dessen Hilfe Zugriffsrechte in der Konfigurationsdatei *Web.config* festgelegt werden können. Die Umsetzung der rollenbasierten Sicherheit innerhalb dieses Elements erfolgt, indem festgelegt wird, welchen Benutzern der Zugriff erlaubt (`<allow>`) und welchen er verboten ist (`<deny>`).

```
<authorization>
<allow users="Hans, Marie" />
<deny users="*" />
</authorization>
```

Codebeispiel4.6.5: Rechtevergabe in der Konfigurationsdatei Web.config

Diesem Autorisierungsverfahren liegt das *URLAuthorzationModule* (`System.Web.Security.UrlAuthorizationModule`) zugrunde. Das Modul stellt Autorisierungsdienste für Dateizugriffsrechte zur Verfügung, wobei das dem aktuellen Benutzer zugeordnete Principal-Objekt zum Vergleich herangezogen wird.

4.6.4 Angriffsmöglichkeiten auf Anwendungsebene

SQL-Injection

SQL Injection ist das Einschleusen bösartiger SQL-Befehle. Diese werden durch nicht validierte Formulareingaben auf dem Server ausgeführt und können nicht erwünschte Änderungen in der Datenbank durchführen, also alle denkbaren Optionen wie das Lesen, Löschen und Ändern von Datensätzen sowie das Ändern von Sicherheitseinstellungen. Ausgangspunkt ist dabei immer ein Formular, welches eine Benutzerauthentifizierung mittels Benutzername und Passwort durchführt. Dabei werden schwache Anfragen ausgenutzt, d.h. die Formulareingaben werden zusammen mit dem SQL-Konstrukt zu einer Zeichenkette gebildet:

```
string sLogin = textBox1.Text;
string sPasswort = textBox2.Text;

string dbConnection = "SELECT userName FROM Users WHERE Login=" +
sLogin + "AND Password=" + sPasswort + ";"; // SQL – Befehl bilden
```

Codebeispiel 4.6.7: Verkettung der Eingabewerte

1. Beispiel

Dem Angreifer ist weder der Benutzername noch das Passwort bekannt. Die Eingabe `' or ''='` im Feld für den Benutzernamen und `' or ''='` im Feld für das Passwort wird übersetzt als:

```
SELECT userName FROM Users WHERE Login='' or ''='' AND Password=
'' or ''=''
```

Codebeispiel 4.6.7: `or ''=''` wird immer wahr

Die Eingabe `' or ''='` kann auch genutzt werden, wenn nur das Passwort oder nur das Login bekannt ist. Die Eingabe `Peter' and ''='' or ''='` als Login und `' or ''='` als Passwort besagt, dass dem Angreifer nur das Login bekannt ist, jedoch nicht das Passwort.

2. Beispiel
Der Angreifer nutzt Kommentare aus. Damit kann zum Beispiel die Passwortabfrage verhindert werden, wenn nur der Benutzername bekannt ist. Der Benutzername bekommt den Anhang '/* und das Passwort den Wert `*/ OR '' = '` . Daraus ergibt sich:

```
SELECT userName FROM Users WHERE Login = 'Peter'/*' AND Password =
'*/ OR '' = ''
```

Codebeispiel 4.6.8: Ausnutzen von Kommentaren

Dies wird übersetzt als:

```
SELECT userName FROM Users WHERE Login = '' OR '' = ''
```

Codebeispiel 4.6.9: Ausnutzen von Kommentaren (2)

Analog dazu können auch einzeilige Kommentare ausgenutzt werden. Der Benutzername `'; drop table Users--` führt zum Löschen der Tabelle User, sofern diese vorhanden ist.

3. Beispiel
Webseiten, die Parameter aus der URL auslesen, ohne diese anschließend zu prüfen, können ebenso für *SQL-Injection* missbraucht werden. In diesem Beispiel erwartet die Seite *search.aspx* zum Suchen den Parameter *topic*. Der erwartete Aufruf sieht wie folgt aus:

```
http://webserver/search.aspx?topic=xml
```

Codebeispiel 4.6.10: Ausnutzen von SQL-Injection (1)

Dieser erzeugt das SQL-Konstrukt:

```
SELECT desc FROM Tabelle1 WHERE topic LIKE '%xml'
```

Codebeispiel 4.6.11: Ausnutzen von SQL-Injection (2)

Ein durch *SQL-Injection* manipulierter Aufruf:

```
http://webserver/search.aspx?topic=xml'; DROP table Users --
```

Codebeispiel 4.6.12: Ausnutzen von SQL-Injection (3)

Das daraus erzeugte SQL-Konstrukt löscht die Tabelle *Users*, sofern diese vorhanden ist:

```
SELECT desc FROM Tabelle1 WHERE topic LIKE '%xml'; DROP table
Users--
```

Codebeispiel 4.6.13: Ausnutzen von SQL-Injection (3)

XSS

XSS (Cross-Site Scripting) ist das Einschleusen von clientseitig interpretiertem Code, zum Beispiel JavaScript, in eine vertrauenswürdige Webseite. Das Prinzip ist wie bei *SQL-Injection* das Ausnutzen von ungeprüften Formulareingaben. Ein gängiges Beispiel dafür bietet das Gästebuch einer Webseite. Das Gästebuch nimmt die Benutzereingaben entgegen und speichert diese. Werden die Eingaben ungeprüft ausgegeben, so kann es zum Beispiel einem Betrachter des Gästebuches passieren, dass ein vertrauliches *Cookie* vom Betrachter zum Angreifer übermittelt wird. Enthält dieses *Cookie* vertrauliche Sitzungsdaten, so kann der Angreifer sich für einen bestimmten Zeitraum Zugang zu einem geschützten Teil der Webseite verschaffen (siehe *Session-Hijacking*).

1. Beispiel
Das erste Beispiel ist das Einbetten eines bösartigen Codes in einem Gästebuch. Dabei wird ein Link manipuliert, der darauf wartet, von einem anderen Benutzer ausgeführt zu werden:

```
<A HREF="http://example.com/page1.aspx?key=<SCRIPT>boeses
Script</SCRIPT>">Klick mich</A>
```

Codebeispiel 4.6.14: Einbetten von bösartigen Links in Gästebüchern

2. Beispiel
HTML-Tags lassen sich nutzen, um unerwünschte Funktionen zu starten:

```
<IMG SCR="javascript:boeseFunktion();">
```

Codebeispiel 4.6.15: Ausnutzen von HTML-Tags

Ausnutzen von Fehlermeldungen

Das folgende Beispiel soll die Auswirkungen von *SQL-Injection* verdeutlichen und die Risiken einer unzureichenden Ausnahme- und Fehlerbehandlung offen legen. Ein typisches Beispiel ist das Auslesen der Datenbankstruktur. Dabei werden die SQL-Befehle *Group By* und *Having* ausgenutzt und es wird vorausgesetzt, dass keine Eingabeüberprüfung implementiert ist. Die Kombination dieser Befehle dient dem

Gruppieren von Datenspalten. Die Eingabe von `' having 1=1 --` führt zu der folgenden Fehlermeldung:

```
Column 'userInformation.userID' is invalid in the select list
because it is not contained in an aggregate function and there is
no GROUP BY clause.
```

Codebeispiel 4.6.16: Ausnutzen von Fehlermeldungen (1)

Anhand dieser Information weiß der Angreifer nun, dass es eine Tabelle mit dem Namen *userInformation* gibt und eine Spalte *userID* heißt. Mit der Eingabe von `' group by userInformation.userID having 1=1 --` erhält er die nächste Spalte:

```
Column 'userInformation.userName' is invalid in the select list
because it is not contained in either an aggregate function or the
GROUP BY clause.
```

Codebeispiel 4.6.17: Ausnutzen von Fehlermeldungen (2)

4.6.5 Sichere Eingabeüberprüfung

Einfache und allgemeine Maßnahmen sind das Verbieten und Löschen von Zeichen (Blacklist Characters), das Zulassen nur bestimmter Zeichen (Whitelist Characters) und das Maskieren von Sonderzeichen (Escape Characters). Allgemein bedeutet hier, dass diese Maßnahmen nicht auf eine bestimmte Progammiersprache oder -plattform beschränkt sind.

Blacklist Charakters

Es werden nicht erlaubte Zeichen oder Zeichenfolgen festgelegt, z.B.:

( ) / , ; . : # <> | \ " ' -- <script>

Wird die Eingabe eines Benutzers erfolgreich auf eines dieser Zeichen getestet, so werden diese gelöscht oder es wird eine entsprechende Fehlermeldung ausgegeben. Allerdings könnte durch die *Blacklist* auch versehentlich ein falscher Wert gesperrt werden. Die Stadt „Neu-Greppin“ würde nicht zugelassen werden, wenn zum Beispiel das Zeichen „-“ wegen des einzeiligen Kommentars gesperrt ist. Eine Abhilfe kann hier nur darin bestehen, die gesamte Zeichenkette „--" (einzeiliger Kommentar) zu sperren. Weitere Schwierigkeiten könnten Namen wie zum Beispiel „O'Reilly“ bereiten. Es wird also deutlich, wie aufwendig das Einführen einer *Blacklist* sein kann.

Außerdem reicht es nicht, Wörter wie „select“, „union“ oder „drop“ zu sperren. Es wird angenommen, dass der einfache Anführungsstrich (*Single Quote*) „ ' “ gesperrt ist. Es erfolgt die Eingabe: `sel'ect`

Bei einer fehlerhaften oder nicht vollständigen Implementierung würde das Single Quote entfernt und das gesperrte Wort „select“ zugelassen werden. Auf diese Weise könnte man eine komplette SQL-Anweisung aufbauen, welche dann als Login oder Passwort als gültig erkannt werden würde. Alternativ könnten auch unerwünschte Zeichen ersetzt werden. „>“ durch „&gt“ , „<“ durch „<“, usw. Dies führt allerdings zu einem vergleichbaren Aufwand wie das Löschen oder Verbieten von Zeichen oder Zeichenketten.

Whitelist Charakters

Es werden nur bestimmte Zeichen erlaubt, die vorher festgelegt werden. Anhand des folgenden Beispiels soll verdeutlicht werden, welcher Aufwand dabei entstehen kann, wenn jedes Zeichen der Eingabe auf sein Vorhandensein getestet wird:

A B C … x y Z

Kann ein bestimmter Zeichensatz im Kopf der Webseite hinterlegt werden, zum Beispiel `charset=ISO-8859-1`, so können nur darin definierte, minimale Zeichen eingegeben werden.

Escape Charakters

Zeichen, die eine Sonderbedeutung in SQL haben, zum Beispiel das Semikolon, können durch Voranstellen des Maskierungszeichens, einem umgekehrten Schrägstrich, als Text gekennzeichnet werden (*Escapen*). Werden bei Namen mit einem Apostroph, wie zum Beispiel „O'Reilly“, fälschlicherweise einfache Anführungsstriche verwendet, besteht die einfachste Lösung aus Sicht der Sicherheit darin, Anführungsstriche zu verbieten. Wenn dies nicht akzeptabel ist, müssen *Single Quotes* durch die Maskierung unwirksam gemacht werden.

Vermeiden von Eingabeänderungen

Es dürfen keine veränderten Eingaben in die Datenbank eingetragen werden, die vom Angreifer eingesehen werden können. Dies wird Anhand des Beispiels `sel'ect` deutlich. Wenn sich ein Angreifer als gewöhnlicher Anwender anmeldet und als Login O’Brian angibt, so kann er beim späteren Einsehen seines Nutzerprofils feststellen, wie mit Single Quotes umgegangen wird. Würden diese („OReilly“) entfernt werden, so könnte er eine komplette SQL-Anweisung aufbauen (vgl. *Blacklist Characters*). Daher soll nochmals darauf hingewiesen werden, wie aufwendig sich das Anlegen einer *Blacklist* beziehungsweise einer *Whitelist* gestalten kann. Dem sind parametrisierte Abfragen oder wenn möglich reguläre Ausdrücke vorzuziehen (Abschnitt 4.6.6).

Tabellen

Wie bereits einleitend beschrieben, sollten Fehlermeldungen keine Informationen über die Tabellenstruktur ausgeben. Steht keine ausreichende Fehlerbehandlung zur Verfügung, kann auf einfache Methoden zurückgegriffen werden. So kann zum Beispiel ein Angriff erschwert werden, indem die Tabellenbezeichnungen in komplexe Namen geändert werden, die keinen Rückschluss auf den Inhalt zulassen.

Passwörter

Ein weiterer Angriffspunkt ist das ‚Knacken‘ von Passwörtern. Dies wird dem Angreifer erleichtert, wenn zum Beispiel Fehlermeldungen den Rückschluss

zulassen, wie komplex die Passwörter maximal aufgebaut sein können. Daher sollte der Komplexität eines Passwortes, zum Beispiel die Anzahl der erlaubten Zeichen, keine Einschränkungen gegeben werden. Ist dies auf Grund von Systemeigenschaften nicht möglich, sollten keine Fehlermeldungen ausgegeben werden, die einen Rückschluss auf die Komplexität des Passwortes geben könnten.

4.6.6 Maßnahmen unter ASP.NET unter Verwendung von CSharp.NET

Parametrisierte Abfragen

Parametrisierte Abfragen verhindern das unerwünschte Ausführen von SQL-Befehlen. Ein schlechtes Code-Beispiel unter C# .NET wäre die Verkettung von Zeichen (Codebeispiel 4.6.18), welche das Einschleusen von SQL-Befehlen zulässt.

```
string spalte2Wert = "Mein Wert";

string strConn = ...
SqlConnection MyConn = new SqlConnection(strConn);
MyConn.Open();
...
SqlCommand cmd = new SqlCommand();
cmd.CommandText = "SELECT sum(anzahl) FROM Bestellungen WHERE
KundenID = '" + spalte2Wert + "'";
cmd.Connection = MyConn;
...
object RetVal = cmd.ExecuteScalar();
Response.Write(RetVal.ToString());
...
cmd.Dispose();
MyConn.Close();
```

Codebeispiel 4.6.18: Verkettung von Zeichen

Codebeispiel 4.6.19 ist ein Beispiel für die Verwendung von parametrisierten Abfragen:

```
string spalte2Wert = "Mein Wert";
...
SqlCommand cmd = new SqlCommand();

cmd.CommandText = "select sum(anzahl) from Bestellungen WHERE
KundenID = @spalte2Wert;" // Platzhalter
```

Codebeispiel 4.6.19: Parametrisierte Abfragen unter C# .NET

Reguläre Ausdrücke unter .NET

Mit Hilfe eines regulären Ausdrucks ist es möglich, zu überprüfen, ob eine Zeichenkette nur erlaubte Zeichen enthält, bestimmte Zeichen oder Gruppen von Zeichen in der richtigen Reihenfolge auftreten, oder ob Zeichen in der richtigen Anzahl auftreten.

Zur Überprüfung von Eingaben stehen unter .NET *Regular Expressions* zur Verfügung. Ein regulärer Ausdruck ist ein String, der das Muster beschreibt, nach dem ein anderer String aufgebaut ist. ASP.NET bietet dafür den *RegularExpressionValidator* an. Mit diesem Steuerelement ist es möglich, zu testen, ob eine eingegebene Zeichenfolge einem bestimmten Schema entspricht. Codebeispiel 4.6.20 zeigt ein Beispiel von regulären Ausdrücken unter ASP.NET bei der Überprüfung einer E-Mail-Adresse.

```
<% @Page debug="true" %>
<html>
  <head>
    <title>RegularExpression-Validator</title>
  </head>
  <body>
    <h3>RegularExpression-Validator</h3>
    <p>Bitte geben Sie eine E-Mail-Adresse ein:</p>
    <form runat="server">
      <input type="text" runat="server" id="txtEMail">
      <input type="submit" value=" OK ">
```

Codebeispiel 4.6.20: Der RegularExpressionValidator unter ASP.NET

Interessant ist der Ausdruck: `ValidationExpression="^.+@.+\..{2,}"`

`^`	Markiert den Anfang der Zeichenkette
`+`	Am Anfang steht mindestens ein Zeichen für den Empfänger
`@`	Es folgt genau ein @ - Zeichen
`+`	Es folgen einige Zeichen für die Domain
`\.`	Es folgt ein Punkt
`{2,}`	Den Abschluss bilden mindestens zwei Zeichen

Gespeicherte Prozeduren

Ist keine vollständige, sichere Eingabeprüfung garantiert, so kann zusätzlich auf gespeicherte Prozeduren zurückgegriffen werden. Gespeicherte Prozeduren sind in der Datenbank abgelegt und können mehrere Datenbankoperationen in einem Befehl zusammenfassen. Für diese können auf Datenbankebene Rechte vergeben werden. Berechtigungen können Benutzerkonten oder Rollen zugewiesen werden:

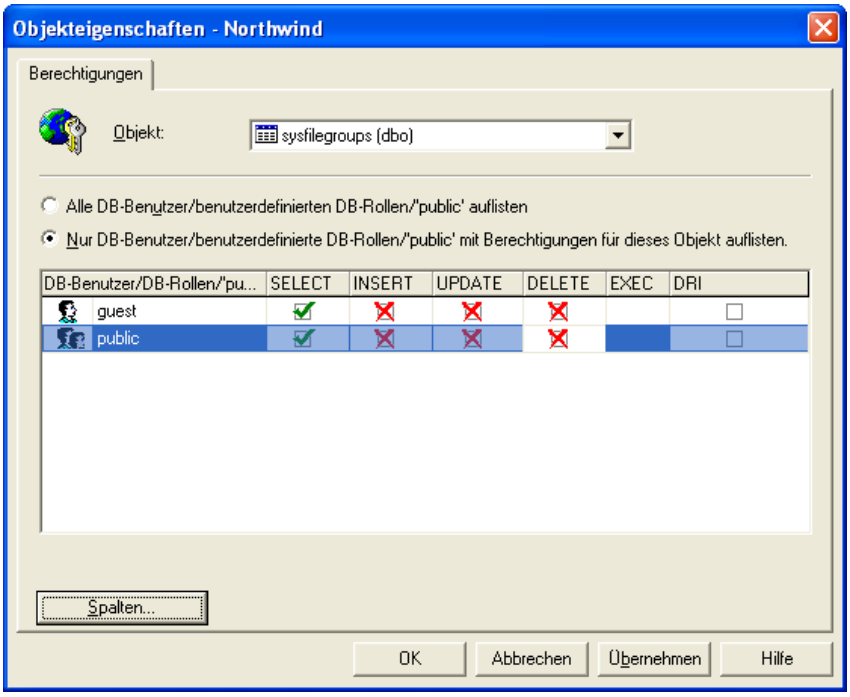

Abbildung 4.6.21: Rechtevergabe bei einer gespeicherten Prozedur am Beispiel des MSSQL - Server

Statt der Eigenschaft *CommandText* (parametrisierte Abfragen) wird auf die Eigenschaft *CommandType* zugegriffen. Der Variablen vom Typ *SqlCommand* wird der *CommandType StoredProcedure* zugewiesen. Anschließend werden die Parameter angelegt und ihnen Werte zugewiesen:

```
...
SqlCommand cmd = new SqlCommand("GetOrganization", sqlConnection);

cmd.SelectCommand.CommandType = CommandType.StoredProcedure;

cmd.SelectCommand.Parameters.Add(new SqlParameter("@orgNumber",
SqlDbType.Integer));
```

Codebeispiel 4.6.22: Verwendung von StoredProcedures unter C# .NET

Gespeicherte Prozeduren steigern darüber hinaus die Effizienz beim Zugriff auf die Daten. Sie sind in der Datenbank abgelegt und können eine Auflistung mehrerer,

vorkompilierter SQL-Anweisungen enthalten, die mit einem einzigen Aufruf ausgeführt werden.

Gesonderte Fehlerbehandlung (Exceptions) unter CSharp.NET

Unbehandelte Fehlermeldungen können dem Angreifer helfen, die zugrunde liegende Struktur der Datenbank auszulesen oder Implementationsmechanismen zu erkennen. Fehlermeldungen können mutwillig oder unbeabsichtigt entstehen. Der Entwickler steht dem Problem gegenüber, dass Fehlermeldungen dem Anwender weiterhelfen sollen, dem Angreifer jedoch nicht zu viel verraten dürfen.

Die bisher behandelten Punkte zur sicheren Eingabeüberprüfung schließen eine umfangreiche Fehlerbehandlung ein, da die Eingaben des Anwenders einer strengen Überprüfung unterliegen. Allerdings dürfen diese Eingabeüberprüfungen auch keinen Rückschluss darauf geben, wie die Eingabe geprüft oder gegebenenfalls geändert wurde. Zur Fehlerbehandlung unter C#.NET stehen sogenannte *try-catch*-Anweisungen zur Verfügung. Im ersten Block *try* werden Anweisungen ausgeführt, die möglicherweise zu einem Fehler führen könnten. Kommt es zu einem Fehler oder einer Ausnahme, so wird der Block *catch* ausgeführt. Zusätzlich gibt es noch den dritten, optionalen Block *finally*. Wird dieser Block verwendet, so wird dieser in jedem Fall ausgeführt, unabhängig davon, ob es zu einer Ausnahmebehandlung kam oder nicht. Ist ein Fehler aufgetreten, so wird nach der bestmöglichen Ausnahmebehandlung gesucht. Zu einer Anweisung können mehrere Ausnahmen auftreten. Da meistens nicht offensichtlich ist, wie viele verschiedene Ausnahmen auftreten können, greift im Zweifelsfall immer die Klasse *Exception*. Alle anderen Ausnahmen werden von dieser abgeleitet [Lo02].

4.7 Sicherheit auf Übertragungs- und Dokumentebene

Nach [BSI] wird der Schutzbedarf im Bereich der Kommunikation in Verbindungen über Außenbereiche (Internet) und Nicht-Außenbereiche (z.B. ein lokales Netz) eingeteilt. Hinzu kommt die Unterscheidung in Verbindungen, auf denen besonders schutzbedürftige Daten übertragen beziehungsweise nicht übertragen werden dürfen. Die Kommunikationsverbindungen, über die besonders schutzbedürftige Informationen übertragen werden (Hotelgastdaten), entsprechen unter WorldCheckInn der Kommunikation über Außenverbindungen. Diese Verbindung stellt einen hohen Anspruch an die Vertraulichkeit, Integrität und Verfügbarkeit. Auch spielt die Verfügbarkeit des Zuganges über diese Außenverbindung eine wichtige Rolle. Ist der WorldCheckInn-Verzeichnisdienst nicht erreichbar, beziehungsweise ist eine Unterbrechung dieses Kommunikationsweges eingetreten, ist die Funktionalität des Gesamtsystems unterbrochen. Finanzielle Auswirkungen können die Folge sein. Der Kommunikationsweg über unkontrollierte Außenverbindungen, wie zum Beispiel das Internet, stellt einen zentralen Angriffspunkt dar. Die Aspekte Vertraulichkeit, Integrität und Verfügbarkeit sind als sehr hoch anzunehmen. In Abbildung 4.7.1 wird deutlich, wo sich diese Außenverbindungen wiederfinden. Es sind lediglich Kommunikationsverbindungen, über die besonders schutzbedürftige Informationen übertragen werden dürfen, Bestandteil der folgenden Betrachtung.

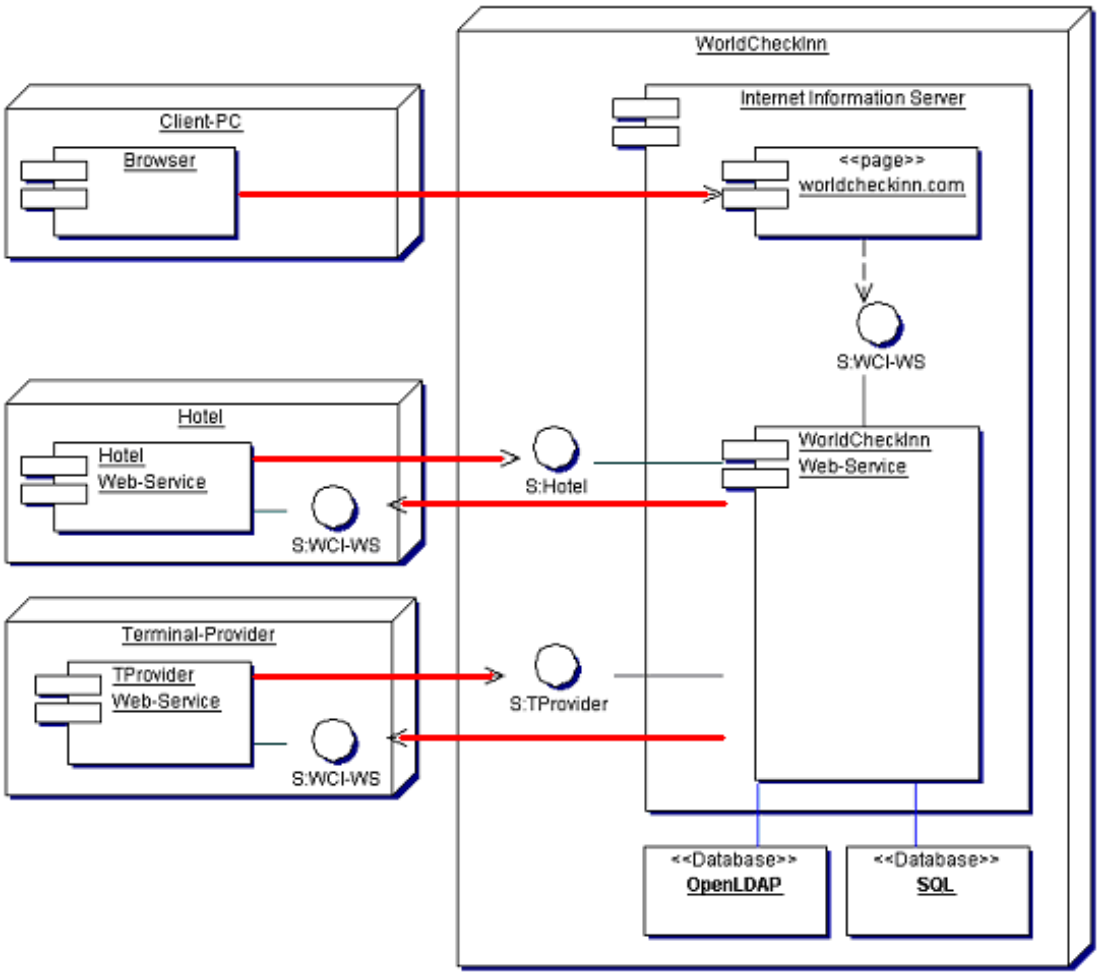

Abbildung 4.7.1: Außenverbindungen – Verbindungen über unsichere Leitungen (rot markiert)

Aufgrund des häufigen Transfers von Personendaten, ist der Schwerpunkt der in Abbildung 4.7.1 dargestellten Kommunikationsverbindungen die Kommunikation der Web-Services untereinander. Das Basisprotokoll der Kommunikation von Web-Services mit Client-Anwendungen oder anderen Web-Services ist das Simple Object Access Protocoll (SOAP).

SOAP-Nachrichten sind XML-basiert. Sie bestehen aus einem oder mehreren *SOAP-Headern* (optional) und einem *SOAP-Body*. Der *Header* kann Metainformationen zur verwendeten Verschlüsselung, Transaktions-IDs oder Routing-Informationen enthalten. Anschließend folgt der *Body* mit den zu übertragenden Informationen.

Diese beiden Elemente sind in dem Element *SOAP-Envelope* zusammengefasst, in dem auch der verwendete Namensraum festgelegt wird.

Aufgrund der Parallelen zum System WorldCheckInn kann hier das durch [wiki:SOAP] bereitgestellte Beispiel einer Gültigkeitsüberprüfung von Kreditkartendaten herangezogen werden. In Codebeispiel 4.7.1 wird anhand der Kreditkartennummer und des Gültigkeitsdatums eine Anfrage gestartet, in Codebeispiel 4.7.2 wird diese mit „true“ (beziehungsweise „false“) beantwortet.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-
                 instance">
  <soapenv:Body>
    <ns1:validate soapenv:encodingStyle="
         http://schemas.xmlsoap.org/soap/encoding/"
         xmlns:ns1="urn:CardValidator">
      <number xsi:type="xsd:string">1234 5678 9876 5432
      </number>
      <valid xsi:type="xsd:string">12/08</valid>
      </ns1:validate>
  </soapenv:Body>
</soapenv:Envelope>
```

Codebeispiel 4.7.1: SOAP-Anfrage (Request) des Clients

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
                  instance">
  <soapenv:Body>
    <ns1:validateResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                   xmlns:ns1="urn:CardValidator">
       <addReturn xsi:type="xsd:string">true</addReturn>
     </ns1:validateResponse>
     </soapenv:Body>
</soapenv:Envelope>
```

Codebeispiel 4.7.2: SOAP-Antwort (Response) des Servers

SSL

Eine der Sicherheitsmechanismen des Internet Information Servers ist die Verschlüsselung des Kommunikationskanals durch Unterstützung von Secure Sockets Layer (SSL). Damit steht diese Form der Verschlüsselung bei der Erstellung der ASP.NET Internetanwendung sowie der ASP.NET Web-Services zur Verfügung.

SSL findet seinen Einsatz auf der Übertragungsebene zur Verschlüsselung des Kommunikationskanals wieder. Das Verfahren wird auf der Transportebene eingesetzt und kann die Nachricht komplett verschlüsseln, deckt jedoch nur die Verschlüsselung zwischen zwei Kommunikationspunkten ab (*Point-to-Point*).

Dem gegenüber steht die Verschlüsselung und Signierung auf der Dokumentebene durch *XML-Encryption* beziehungsweise *XML-Signature*. Wenn z.B. mehr als zwei Partner an der Kommunikation beteiligt sind und eine SOAP-Nachricht beispielsweise über einen Web Service-Vermittler weitergeleitet werden soll, so müsste im Falle von SSL die Nachricht vor der Weiterleitung entschlüsselt werden, um sie danach neu verschlüsselt weiter zu versenden. Mit *XML-Encryption* muss die Verschlüsselung nicht bei jedem Kommunikationspartner neu durchgeführt werden und bleibt somit bis zum eigentlichen Empfänger weiterhin bestehen (*End-to-End*).

XML Signature / Encryption

Auf der Dokumentebene findet der Austausch von Dokumenten statt. Diese sollten ganz oder zumindest teilweise verschlüsselt werden, so dass vertrauliche Daten des Dokumentes der anderen Partei nicht preis gegeben werden. An dieser Stelle ist bereits Abhilfe durch *XML-Encryption* und -*Signature* geschaffen. *XML-Encryption* ermöglicht das Verschlüsseln von Teilen eines Dokumentes, so dass vertrauliche Daten verborgen bleiben. Analog dazu ermöglicht *XML-Signature* das Signieren bestimmter Dokumentteile, die nicht signierten Teile des Dokumentes bleiben änderbar. Ein Anwendungsbeispiel ist das Verschlüsseln von Teilen eines (Vertrags-)Dokumentes, welche von der anderen Partei nicht eingesehen werden dürfen, z.B. die Geschäftsidentifikationsnummer oder andere vertrauliche Vertragsinformationen.

XML-Signature und -*Encryption* sind Bestandteil der Spezifikationen der Global Web-Service Architecture (GXA), deren Inhalt im nächsten Abschnitt näher erläutert wird.

4.7.1 Global Web-Service Architecture (GXA)

Hinter dem Namen Global Web-Service Architecture verbirgt sich eine Sammlung von Spezifikationen, welche durch namhafte Hersteller wie Microsoft, SUN und IBM dem World Wide Web Consortium (W3C), der Internet Engineering Task Force (IETF) und der Organization for the Advancement of Structured Information Standards (OASIS) vorgeschlagen wurde. Diese Spezifikationen befassen sich mit den bisher vernachlässigten Konzepten für Sicherheit, Policies, Routing, zuverlässige Transaktionen und sicheren Nachrichtentransfer bei Web-Services [Cz]. Besonderheit und Grundlage dieser Spezifikationen ist die Kompatibilität untereinander – sie sind je nach Anforderung des Web-Services miteinander kombinierbar. Eine Liste dieser Spezifikationen ist unter [OiO] zu finden.

Von besonderer Bedeutung ist die Spezifikation *WS-Security*, deren Verwendung im nächsten Abschnitt aufgezeigt wird. *WS-Security* wurde von Microsoft, VeriSign und IBM im April 2002 veröffentlicht und ist bereits als Standard herausgegeben. Sie soll die Integrität, Verbindlichkeit und Vertraulichkeit von Nachrichten sichern.

SOAP-Nachrichten werden ohne Erweiterungen als Klartext übertragen. *WS-Security* dient dem Zweck, sicherheitsrelevante Daten in SOAP-Nachrichten einzubetten, SOAP-Nachrichten zu signieren und das Anhängen von *Security Credentials* an SOAP-Nachrichten zu ermöglichen. Dazu setzt *WS-Security* die Standards *XML Signature* und *XML Encryption* des W3Cs ein [OiO]. *Security Credentials* werden für die Authentifizierung verwendet. Dies kann ein Benutzername und Passwort, ein X.509 Zertifikat oder ein Kerberos-*Token* [KT] sein.

Bei der Beschreibung der Sicherheitsanforderungen kommt die Spezifikation *WS-Policy* zum Einsatz. Die Spezifikation ist als ein Filter zu verstehen, der eingehende Nachrichten auf vorab festgelegte Anforderungen/Parameter überprüft, zum Beispiel ob eine Signatur enthalten ist oder die Nachricht eine bestimmte Größe nicht überschreitet.

Werden zwischen Anwender und Web-Service mehr als eine Nachricht ausgetauscht, so kann zur Verbesserung der Performance die Spezifikation *WS-SecureConversation* herangezogen werden. Diese Spezifikation schafft einen gemeinsamen Kontext (Sicherheitskontext-Token) zwischen Anwender und Web-Service, um den Aufwand zu reduzieren, welcher zum Sichern der Nachrichten notwendig ist.

Um unter ASP.NET-Anwendungen *WS-Security*, *WS-SecureConversation* oder andere Spezifikationen der GXA verwenden zu können, bedarf es einer Erweiterung von ASP.NET und des .NET-Frameworks. Unter dem Namen Web Services Enhancements (WSE) wurde solch eine Erweiterung 2002 in der Version 1.0 von Microsoft veröffentlicht [O'Neill03]. Seit Version 2.0 liegen weitere Funktionalitäten zur Erhöhung der Sicherheit von Web-Services sowie Erweiterungen am Nachrichtenmodell vor. Diese Erweiterungen ermöglichen das asynchrone Austauschen von Nachrichten zwischen Web-Services sowie das Verwenden von Funktionen zur Stapelverarbeitung von Nachrichten und das Programmieren ereignisgesteuerter Applikationen. Seit November 2005 liegt die Version 3.0 (beta) vor, welche darüber hinaus gehende Funktionalitäten zur Verfügung stellt. Unter anderem wurde die Programmierung sicherer Web-Services durch die Einbindung von *Security-Profiles* vereinfacht und die Performance erhöht. *Security-Profiles* (dt.: Sicherheitsprofile) sind Sicherungs-Szenarien für Web-Services, die (zur Zeitersparung) per Mausklick anwendbar sind.

Abbildung 4.7.2 veranschaulicht die Kommunikation zwischen einem Client und einem Web-Service mittels *Policies*. Dargestellt wird ein Client, bei dem mit Hilfe einer *Policy* Richtlinien festgelegt sind, die er berücksichtigen muss, um mit dem Web-Service kommunizieren zu können. Diese Richtlinien enthalten Sicherheitsmaßnahmen, die bei ein- und ausgehenden SOAP-Nachrichten berücksichtigt werden müssen. Die ein- und ausgehenden Nachrichten durchlaufen zuerst eine Pipeline, welche ein oder mehrere Filter enthalten kann. Diese Filter prüfen eingehende Nachrichten auf das Vorhandensein bestimmter Merkmale, zum Beispiel ob zum *Soap-Body* im *Soap-Header* der Nachricht eine Signatur enthalten ist. Ausgehende Filter hingegen fügen der Nachricht weitere Bestandteile wie zum Beispiel eine Signatur hinzu oder verschlüsseln den *Soap-Body* der ausgehenden Nachricht.

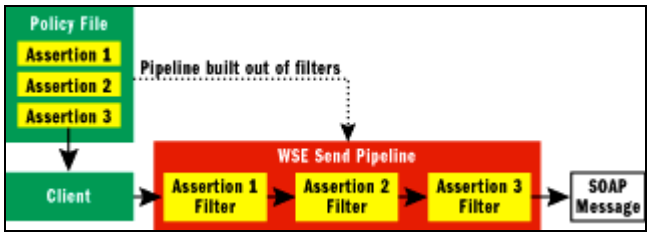

Abbildung 4.7.2: Verwendung der Polices und Aufbau der Pipelines [MSDN]

Abbildung 4.7.3 zeigt den Gesamtaufbau einer Client-Server-Kommunikation unter Verwendung von Ein- und Ausgangsfiltern:

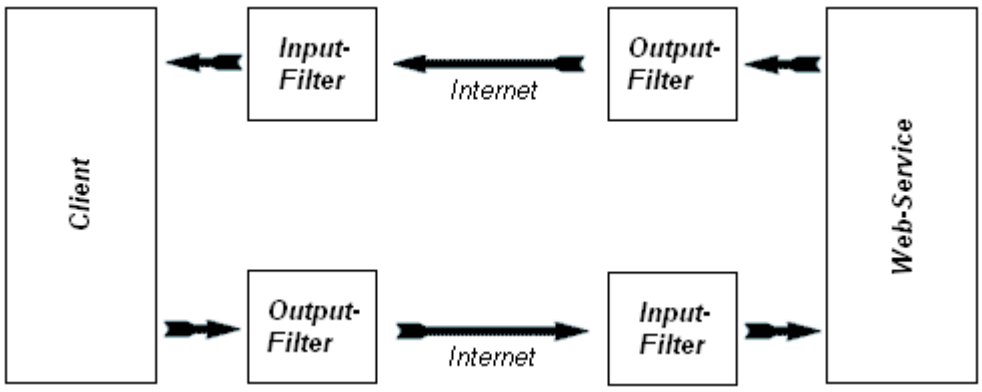

Abbildung 4.7.3: Ein- und Ausgangsfilter bei der Client-Server-Kommunikation

4.7.2 .NET und WSE

Seit dem 07.11.2005 steht eine Beta-Version von WSE 3.0 unter [WSE] sowie ein *Hands On Lab* [WSELab] zum Herunterladen zur Verfügung. Voraussetzung ist das .NET Framework 2.0, ASP.NET 2.0 und das Visual Studio 2005. Da bei dieser Version Probleme während der Installation auftreten können, befindet sich auf dem beliegendem Datenträger im Unterverzeichnis *wse/* eine Installationsanleitung mit anschließender Problembehebung.

Dieser Abschnitt beschreibt die Verwendungsweise von WSE und wertet zum besseren Verständnis die Nachrichtenprotokolle aus. Es werden weiterführende Informationen gegeben, um den Umfang und die Einsatzmöglichkeiten von WSE zur Sicherung von Web-Services besser zu veranschaulichen.

Bei der Installationsanleitung und den folgenden Beispielen liegt das Betriebssystem Windows XP zugrunde.

WSE aktivieren

Einen Rechtsklick auf das aktuelle Projekt im Projektmappen-Explorer des Visual Studio und die Auswahl der Option „WSE 3.0 Settings…“ öffnet das in Abbildung 4.7.4 dargestellte Fenster:

Abbildung 4.7.4: Aktivierung der Web Services Enhancements für das aktuelle Projekt

Die Auswahl der ersten Option fügt dem aktuellen Projekt eine Referenz zu der Microsoft.Web.Services3.dll hinzu. Die Konfigurationsdatei „web.config“ wird erweitert, indem die Sektion `<section name="microsoft.web.services 3".../>` und ein Verweis auf die WSE-Assembly hinzugefügt wird. Diese Änderungen ermöglichen die Verwendung der WSE Proxy-Klassen. Die zweite Option, „Microsoft Web Services Enhancement Soap Protocol Factory“, aktiviert mehrere Erweiterungen für ASP.NET Projekte (Nachrichtenbasierte Sicherheit, MTOM, ...).

```
...
<configSections>
  <section name="microsoft.web.services3"
           type="Microsoft.Web.Services3.Configuration.
                 WebServicesConfiguration,
                 Microsoft.Web.Services3, Version=3.0.0.0,
                 Culture=neutral,
                 PublicKeyToken=31bf3856ad364e35" />
</configSections>
...
<compilation debug="true">
  <assemblies>
    <add assembly="Microsoft.Web.Services3, Version=3.0.0.0,
                   Culture=neutral,
                   PublicKeyToken=31BF3856AD364E35" />
   </assemblies>
   ...
 </system.web>
</configuration>
```

Codebeispiel 4.7.3: Änderungen in der Konfigurationsdatei web.config nach Aktivieren von WSE

Security-Policies

Als nächster Schritt erfolgt die Einbindung einer *Policy*, um Sicherheitsanforderungen eines Web-Services zu definieren. Zur Einrichtung von *Security-Policies* steht der *WSE-Wizard* zur Verfügung. In den folgenden Abbildungen wird

das Szenario „Username for Certificate“ [MSDNTV] umgesetzt. Der Client meldet sich per Benutzername und Passwort am Server an und erhält dafür den öffentlichen Schlüssel eines Zertifikates, mit dem er ausgehende Nachrichten verschlüsseln kann. Abbildung 4.7.5 zeigt das Anlegen einer *Policy*:

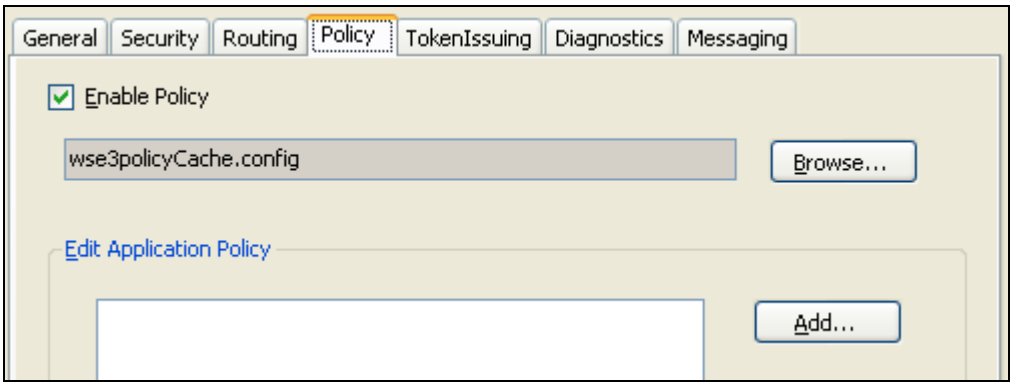

Abbildung 4.7.5: Anlegen einer Policy

Begonnen wird mit der Einrichtung der *Policy* für den WorldCheckInn Web-Service. Die Einrichtung der *Policy* des Clients erfolgt fast analog, lediglich im ersten Menü (Abbildung 4.7.6) wird die Option „Secure a client application“ ausgewählt.

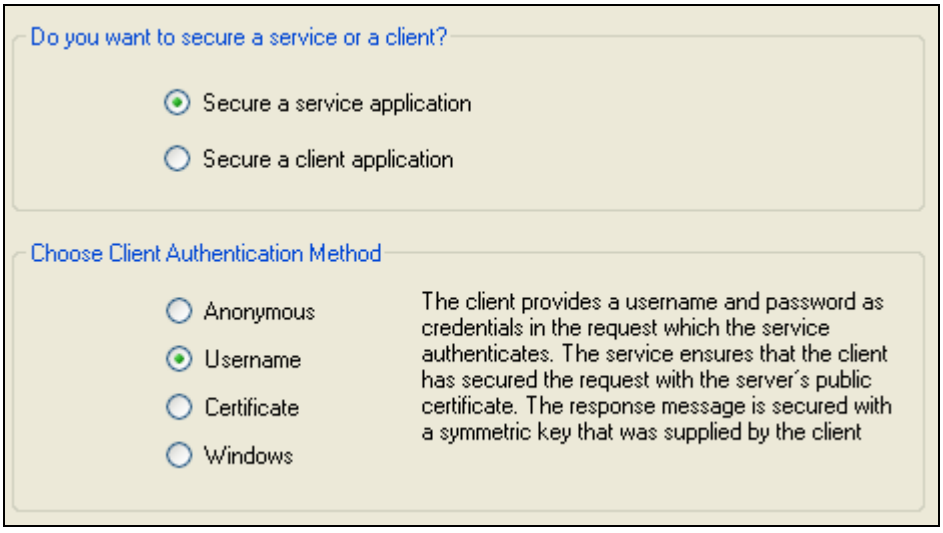

Abbildung 4.7.6: Auswahl der Authentifizierungsmethode

Im zweiten Menüpunkt, der nur bei der Einrichtung des Web-Services erscheint, besteht die Möglichkeit, die Windows-integrierte Authentifizierung zu verwenden. Es können Benutzer oder Rollen festgelegt werden, welche die Funktionen des Web-Services nutzen dürfen:

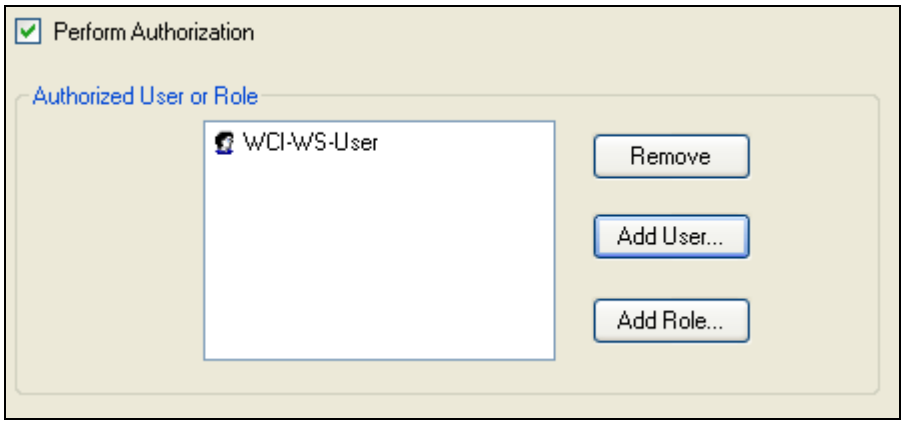

Abbildung 4.7.7: Festlegen von Autorisierungen und Rollen

Im dritten Menüpunkt (Abbildung 4.7.8) werden Signierungs- und Verschlüsselungsmechanismen aktiviert. An dieser Stelle kann auch die Spezifikation *WS-SecureConversation* aktiviert werden („Establish Secure Session“), wo zwischen Client und Server ein symmetrischer Schlüssel ausgehandelt wird, um das bei asymmetrischer Verschlüsselung aufkommende Datenvolumen zu reduzieren.

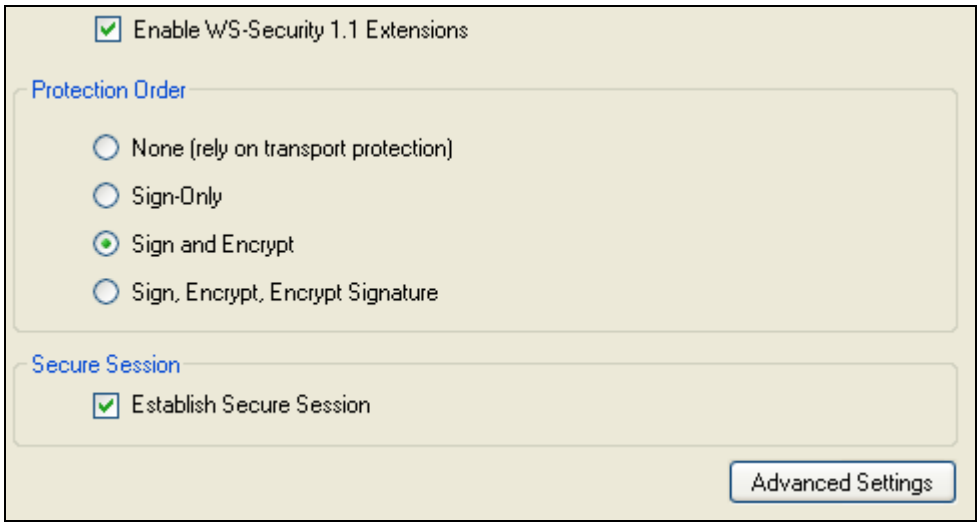

Abbildung 4.7.8: Aktivierung von Signierungs- und Verschlüsselungsmechanismen

Das vierte Menü fordert dazu auf, das Zertifikat des Servers auszuwählen. Zertifikate können auch auf Codeebene definiert werden. Dafür steht der Namensraum `System.Security.Cryptography.X509Certificates` zur Verfügung, welcher Funktionen zum Anlegen, Ändern und Suchen von Zertifikaten bereitstellt.

Abbildung 4.7.9: Auswahl des Serverzertifikates

Der letzte Menüpunkt stellt eine Zusammenfassung der Bezeichnung des Szenarios, der Namen autorisierter Anwender und des Zertifikates des Servers dar:

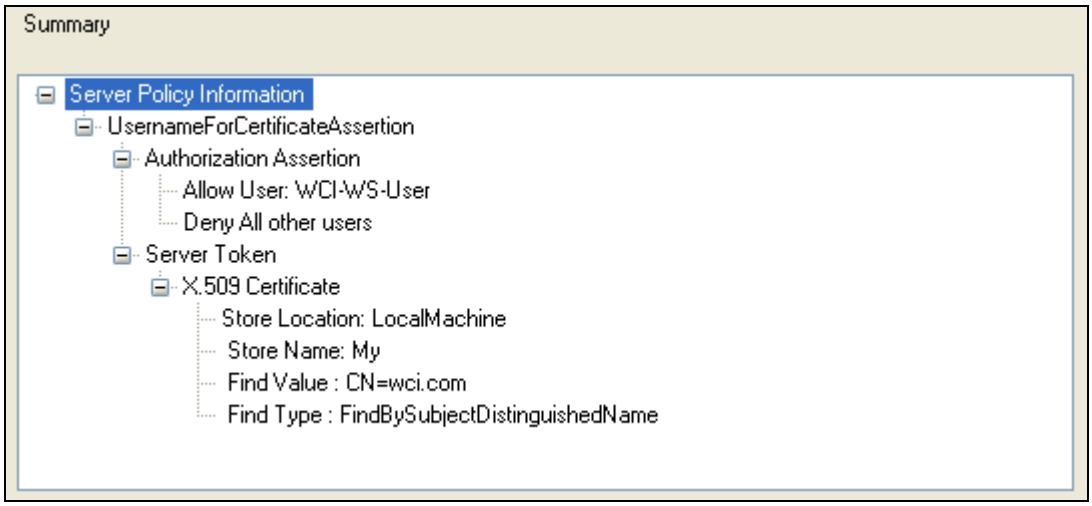

Abbildung 4.7.10: Zusammenfassung

Sowohl auf dem Server als auch auf dem Client erfolgt in der Konfigurationsdatei *web.config* ein Verweis auf die angelegte *Policy*:

```
<configuration>
  ...
  <microsoft.web.services3>
    <policy fileName="wse3policyCache.config" />
  </microsoft.web.services3>
</configuration>
```

Codebeispiel 4.7.4: Policy-Verweis in der web.config

Aufbau einer Policy-Datei

Wurzelelement einer *Policy* ist das Element `policies`. Dieses Element kann mehrere *Policies* enthalten. Vor der Definition der *Policies* befinden sich innerhalb der `extension`-Elemente sogenannte *Policy Assertions*. Diese definieren Anforderungen, die erfüllt sein müssen, um SOAP-Nachrichten zwischen Client und Web-Service auszutauschen. Jedes Element verweist auf eine Klasse, wie zum Beispiel *X509TokenProvider* auf Microsoft.Web.Services3.Design.X509Token Provider. Im Abschnitt 4.7.3 *Custom Assertions* werden dazu nähere Erläuterungen

gegeben. Anschließend folgt das Element `policy`, welches die oben beschriebene und angelegte *Policy* „WCIWebServicePolicy“ darstellt und das Szenario „Username for Certificate„ enthält.

```
<policies xmlns="http://schemas.microsoft.com/wse/2005/06/policy">
  <extensions>
    <extension name="authorization"
     type="Microsoft.Web.Services3.Design.AuthorizationAssertion,
           Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
           PublicKeyToken=31bf3856ad364e35" />
    <extension name="usernameForCertificateSecurity"
               type="UsernameForCertificateAssertion,
                     Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
                     PublicKeyToken=31bf3856ad364e35" />
    <extension name="x509" type="X509TokenProvider,
                                 Microsoft.Web.Services3, Version=3.0.0.0,
                                 Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
    <extension name="requireActionHeader" type="RequireActionHeaderAssertion,
                     Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
                     PublicKeyToken=31bf3856ad364e35" />
  </extensions>
  <policy name="WCIWebServicePolicy">
    <authorization>
      <allow user="WCI-WS-User" />
      <deny user="*" />
    </authorization>
    <usernameForCertificateSecurity establishSecurityContext="true"
      renewExpiredSecurityContext="true" requireSignatureConfirmation="false"
      messageProtectionOrder="SignBeforeEncrypt" requireDerivedKeys="true"
      ttlInSeconds="300">
      <serviceToken>
        <x509 storeLocation="LocalMachine" storeName="My"
      findValue="CN=worldcheckinn.com" findType="FindBySubjectDistinguishedName" />
      </serviceToken>
      <protection>
        <request signatureOptions="IncludeAddressing, IncludeTimestamp,
                                   IncludeSoapBody" encryptBody="true" />
        <response signatureOptions="IncludeAddressing, IncludeTimestamp,
                                    IncludeSoapBody" encryptBody="true" />
        <fault signatureOptions="IncludeAddressing, IncludeTimestamp,
                                 IncludeSoapBody" encryptBody="false" />
      </protection>
    </usernameForCertificateSecurity>
    <requireActionHeader />
  </policy>
</policies>
```

Codebeispiel 4.7.5: Aufbau einer Policy

Zur Einbindung der *Policy* am Web-Service muss vor der Klassendefinition explizit angegeben werden, welche *Policy* verwendet wird:

```
using Microsoft.Web.Services3;
...
namespace WCI_WebDienst
{
  [WebService(Namespace = "http://example.org/invoices")]
  [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
  [Policy("Policy4wci.com")]

  public class wciVerzeichnisdienst :
System.Web.Services.WebService
  {
    //...
```

Codebeispiel 4.7.6: Einbindung einer Policy

Einrichten des Clients

Wird ein Client in einer Intranet-Lösung durch die integrierte Windows-Authentifizierung des IIS authentifiziert, so ist die Angabe von Benutzername und Passwort nicht mehr notwendig. Der IIS greift in diesem Falle auf die Eigenschaft `useDefaultCredentials` zurück. Über Intranet-Lösungen hinaus ist dies jedoch nicht immer möglich, da zum Einen bei dieser Form der Authentifizierung Client und Server auf Windows-basierten Systemen wie XP, 2000 oder NT ausgeführt werden müssen und zum Anderen die Authentifizierung über einen Proxy oder eine Firewall nicht möglich ist. *WS-Security* geht hier einen Schritt weiter und erlaubt die Einbettung von *Security-Token* im Kopf einer SOAP-Nachricht.

Mit dem Aktivieren von WSE ist das Einbinden des Namespaces `Microsoft.Web.Services3.Security.Tokens` möglich, um ein Token, bestehend aus Benutzername und Passwort, für das Szenario “Username for Certifikate” aufzubauen. Im Codebeispiel 4.7.7 wird das Event „Login1_Authenticate“ durch das Anlegen eines *UsernameToken* erweitert. Es wird aufgerufen, sobald der Anwender sich authentifizieren möchte: Der *Token* wird angelegt und die *Security Credentials* werden einer Instanz der Klasse

`wciVerzeichnisdienstWse` zugewiesen. Gleichzeitig wird der Verweis auf die *Policy* gesetzt.

```
…
using Microsoft.Web.Services3.Security.Tokens;
…
protected void Login1_Authenticate(object sender,
AuthenticateEventArgs e)
{
   wciVerzeichnisdienst.wciVerzeichnisdienstWse vdZugriff = new
   wciVerzeichnisdienst.wciVerzeichnisdienstWse();

   UsernameToken Token = new
   UsernameToken(ConfigurationSettings.AppSettings["UserName"]
   .ToString(),
   ConfigurationSettings.AppSettings["UserPass"].ToString(), P
   asswordOption.SendPlainText);

   vdZugriff.SetClientCredential<UsernameToken>(Token);
   vdZugriff.SetPolicy("Policy4wci.com");

   try
   {
     if (vdZugriff.ueberpruefeZugangsberechtigung(Login1.UserName,
         Login1.Password))
     {
          //...
```

Codebeispiel 4.7.7: Implementieren eines UsernameToken an einem Client

SecureConversation

Die Spezifikation *SecureConversation* bietet ein geeignetes Mittel, um den *Overhead* der Nachrichten beim Austausch mehrerer Nachrichten zu reduzieren. Sie ermöglicht den Austausch eines gemeinsamen Schlüssels zwischen Client und Server, welcher eine symmetrische Verschlüsselung der Nachrichten ermöglicht und somit zu einer Verbesserung der Performance beiträgt.

Um die in Abbildung 4.7.8 aktivierte *SecureConversation* nutzen zu können, muss auf Codeebene der Proxy wiederverwendet werden, welcher der Kommunikation zwischen Client und Server dient:

```
private wciVerzeichnisdienst.wciVerzeichnisdienstWse m_vdZugriff;

protected void Login1_Authenticate(object sender,
                                   AuthenticateEventArgs e)

  UsernameToken Token = new
  UsernameToken(ConfigurationSettings.AppSettings["UserName"]
  .ToString(),
  ConfigurationSettings.AppSettings["UserPass"].ToString(),
                             PasswordOption.SendPlainText);

        if (m_vdZugriff == null)
        {
            m_vdZugriff = new
            wciVerzeichnisdienst.wciVerzeichnisdienstWse();
            m_vdZugriff.SetClientCredential<UsernameToken>(Token);
            m_vdZugriff.SetPolicy("Policy4wci.com");
        }

        try
        {
            if   (m_vdZugriff.ueberpruefeZugangsberechtigung
                 (Login1.UserName, Login1.Password))
            {
                 //...
```

Codebeispiel 4.7.8: Verwendung von SecureConversation

Aufbau ein- und ausgehender Nachrichten

Zur Protokollierung ein- und ausgehender Nachrichten steht eine weitere Funktionalität des *WSE-Kits* zur Verfügung:

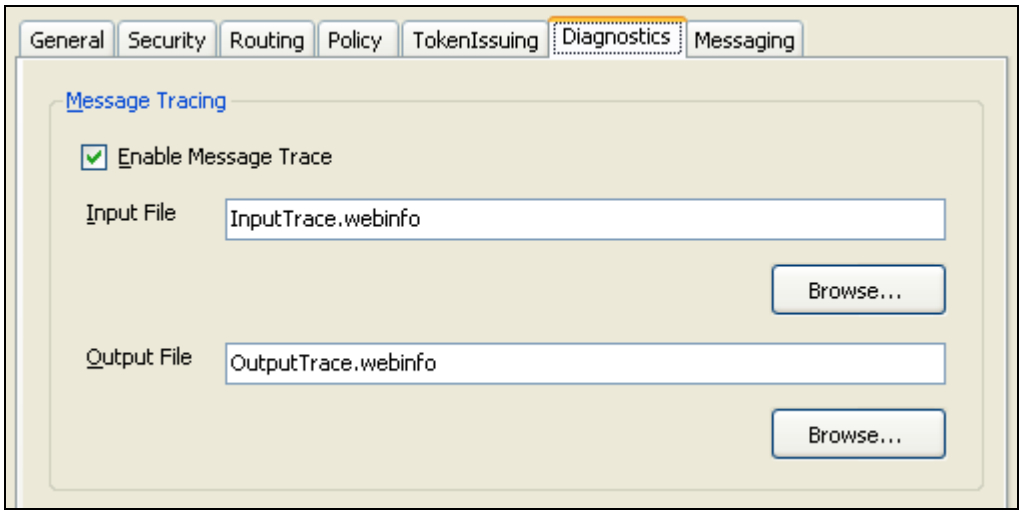

Abbildung 4.7.11: Protokollierung ein- und ausgehender Nachrichten aktivieren

Durch die Einbindung dieser Protokolldateien ist die Aufzeichnung aller ein- und ausgehenden SOAP-Nachrichten möglich. Diese Eigenschaft muss jedoch bei beiden Akteuren des Systems, Server und Client, aktiviert sein. Beim erstmaligen Ausführen des Web-Services werden dann die Protokolldateien angelegt:

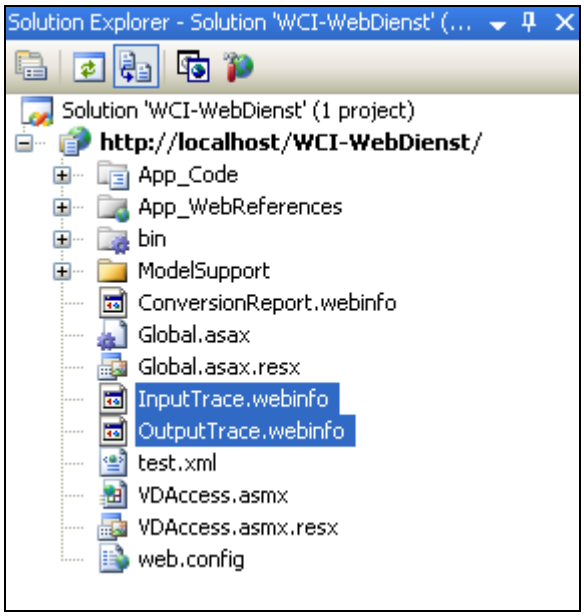

Abbildung 4.7.12: Dateien zur Protokollierung ein- und ausgehender Nachrichten

Inhalt dieser Protokolldateien im XML-Format ist das Wurzelelement <log>, welches mehrere Elemente mit dem Namen <inputMessage> (InputTrace.webinfo) beziehungsweise <outputMessage> (OutputTrace.webinfo) beinhalten kann. Codebeispiel 4.7.9 und 4.7.10 geben eine Übersicht des Aufbaus ein- und ausgehender Nachrichten.

```
<inputMessage>

  <verschlüsselter Header>...</verschlüsselter Header>

  <verschlüsselter Body>...</verschlüsselter Body>

  <processingSteps>...</processingSteps>

  <Header-Klartext>...</Header-Klartext>

  <Body-Klartext>...</Body-Klartext>

</inputMessage>
```

Codebeispiel 4.7.9: vereinfachter Aufbau der Protokolldateien (eingehende Nachrichten)

```
<outputMessage>

  <Body-Klartext>...</Body-Klartext>

  <processingSteps>...</processingSteps>

  <verschlüsselter Header>...</verschlüsselter Header>

  <verschlüsselter Body>...</verschlüsselter Body>

</outputMessage>
```

Codebeispiel 4.7.10: vereinfachter Aufbau der Protokolldateien (ausgehende Nachrichten)

```
  <inputMessage utc="10.03.2006 14:51:24"
messageId="urn:uuid:ad6880ad-cb05-4cd0-8bbe-ff8aa43c9e58">
    <processingStep description="Unprocessed message">
      <soap:Envelope ...>
        <soap:Body>
          <ueberpruefeZugangsberechtigung xmlns="http://...org/">
            <sWCIKdnr>4596-235-5657</sWCIKdnr>
            <sPasswort>qwertz</sPasswort>
          </ueberpruefeZugangsberechtigung>
        </soap:Body>
      </soap:Envelope>
    </processingStep>
    <processingStep description="Entering SOAP filter ..." />
    ...
    <processingStep description="Processed message">
      <soap:Envelope ...>
        <soap:Body>
          <ueberpruefeZugangsberechtigung xmlns="http://....org/">
            <sWCIKdnr>4596-235-5657</sWCIKdnr>
            <sPasswort>qwertz</sPasswort>
          </ueberpruefeZugangsberechtigung>
        </soap:Body>
      </soap:Envelope>
    </processingStep>
  </inputMessage>
```

Codebeispiel 4.7.11: Aufbau einer SOAP-Nachricht beim Login des Clients ohne WS-Security

```
<outputMessage utc="10.03.2006 14:51:24" messageId="urn:...a5b1">
    <processingStep description="Unprocessed message">
      <soap:Envelope ...>
        <soap:Body>
          <ueberpruefeZugangsberechtigungResponse
            xmlns="http://tempuri.org/">
            <ueberpruefeZugangsberechtigungResult>true
            </ueberpruefeZugangsberechtigungResult>
          </ueberpruefeZugangsberechtigungResponse>
        </soap:Body>
      </soap:Envelope>
    </processingStep>
    <processingStep description="Entering SOAP filter ..." />
    ...
    <processingStep description="Processed message">
      <soap:Envelope ...>
        <soap:Header>
<wsa:Action>http://tempuri.org/ueberpruefeZugangsberechtigungRespo
nse
          </wsa:Action>
          <wsa:MessageID>urn:uuid:...eea5b1</wsa:MessageID>
          <wsa:RelatesTo>urn:uuid:...3c9e58</wsa:RelatesTo>
<wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anon
ymous</wsa:To>
          <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-...befc2abd62f">
              <wsu:Created>2006-03-10T14:51:24Z</wsu:Created>
              <wsu:Expires>2006-03-10T14:56:24Z</wsu:Expires>
            </wsu:Timestamp>
          </wsse:Security>
        </soap:Header>
        <soap:Body>
          <ueberpruefeZugangsberechtigungResponse
            xmlns="http://tempuri.org/">
            <ueberpruefeZugangsberechtigungResult>true
            </ueberpruefeZugangsberechtigungResult>
          </ueberpruefeZugangsberechtigungResponse>
        </soap:Body>
      </soap:Envelope>
    </processingStep>
  </outputMessage>
```

Codebeispiel 4.7.12: Antwort des Servers

Die Nachrichten lassen sich in drei Bestandteile gliedern: die Authentifikation, die Signatur der Nachricht und der verschlüsselte Nutztext. Die Authentifizierungsinformationen und die Signatur sind im Element `<wsse:security>` innerhalb des SOAP-Headers untergebracht. Das Element `<wsu:Timestamp>` gibt an, wann die Nachricht generiert wurde (`<wsu:Created>`) und wie lang ihre Lebensdauer ist (`<wsu:Expires>` , vgl. TTL).

Beispiel-Szenario „Zertificate4Zertificate“: Wird ein Zertifikat zur Authentifizierung des Clients verwendet, findet sich im Element `<wsse:security>` das Element `<wsse:BinarySecurityToken>` wieder, welches das Zertifikat im Klartext (nur base64-codiert) enthält.

```
...
<wsse:BinarySecurityToken ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509v3" EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="SecurityToken-7dac43ec-9ead-4184-99d6-
5a73b8973694">MIIBxDCCAXKg...7fdycU0XqiM</wsse:BinarySecurityToken
>
...
```

Codebeispiel 4.7.13: Zertifikate in SOAP-Nachrichten

Das Element `<Signature>` (Codebeispiel 4.7.14) enthält alle Signaturen des Dokumentes. Diese sind innerhalb der Unterelemente `<Reference>` untergebracht. Durch eine eindeutige ID wird Bezug auf signierte Bestandteile des Dokumentes genommen. So wird beispielsweise in Codebeispiel 4.7.14 der *Body* der SOAP-Nachricht referenziert. Der *SOAP-Body* enthält neben dem verschlüsselten Nutztext Informationen darüber, wie verschlüsselt wurde.

```
...
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <ds:CanonicalizationMethod
           Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
           xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
    <SignatureMethod
         Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
    <Reference URI="#Id-fe259b07-1472-45fa-91c8-07d35115c147">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
                              c14n#" />
        </Transforms>
        <DigestMethod
           Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>wddBel1yMeAwg9g1Zcq50rS9k3k=</DigestValue>
    </Reference>
...
  <soap:Body wsu:Id="Id-fe259b07-1472-45fa-91c8-07d35115c147">
    <xenc:EncryptedData Id="Enc-9162dfae-4116-4b2c-b595-
                            f8425878a1f5"
                  Type="http://www.w3.org/2001/04/xmlenc#Content"
                  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
      <xenc:EncryptionMethod
         Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/>
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <wsse:SecurityTokenReference>
          <wsse:Reference URI="#SecurityToken-b4806a29-92fd-42bf-
                               95bd-34e4ce50715d"
       ValueType="http://schemas.xmlsoap.org/ws/2005/02/sc/dk"/>
          </wsse:SecurityTokenReference>
      </KeyInfo>
      <xenc:CipherData>
        <xenc:CipherValue>miEhrB...aqFU9xzQbxNIFw==
        </xenc:CipherValue>
      </xenc:CipherData>
...
```

Codebeispiel 4.7.14: Verschlüsselungs- und Signaturinformationen innerhalb einer SOAP-Nachricht mit WSE

4.7.3 Custom Policy Assertions

Wie bereits in Abbildung 4.7.2 veranschaulicht durchläuft eine SOAP-Nachricht mehrere Filter, wenn diese beim Client oder Server eintrifft oder diesen verlässt. Um Anforderungen einer Server-Client-Kommunikation umzusetzen, die nicht mittels standardisierter Sicherheitsszenarien des *WSE-Kits* (zum Beispiel „Username for Certificate“) abgedeckt werden, bedarf es der Implementation eigener Filter. *Custom Policy Assertions* ermöglichen die Implementation eigener Sicherheitsszenarien auf Basis eigener Filter. Die Implementierung einer *Custom Policy Assertion* bedeutet

- die Implementierung des Eingangsfilters des Servers,
- die Implementierung des Ausgangsfilters des Servers,
- die Implementierung des Eingangsfilters des Clients,
- und die Implementierung des Ausgangsfilters des Clients.

Für die Implemtierung einer eigenen *Policy* muss ein Verweis innerhalb der *Policy* gesetzt werden. Vor Beginn der Policies werden sogenannte *extensions* deklariert. Diese benannten Elemente, wie z.B. *authorization* sind die in Abbildung 4.7.2 dargestellten *Assertions* und finden sich innerhalb der *Policy*-Elemente wieder. Jede dieser *extensions* verweist auf einen bestimmten Typ einer Klasse (*Custom Policy Assertion Class*). Die *extensions* finden sich in den Policies wieder, in denen zusätzlich die Eigenschaften der Typen definiert sind.

```
<extensions>
    <extension name="authorization"
     type="Microsoft.Web.Services3.Design.AuthorizationAssertion,
Microsoft.Web.Services3,
           Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
    <extension name="usernameForCertificateSecurity"
               type="UsernameForCertificateAssertion,
                     Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
                     PublicKeyToken=31bf3856ad364e35" />
    <extension name="x509" type="X509TokenProvider,
                                 Microsoft.Web.Services3, Version=3.0.0.0,
                                 Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
    <extension name="requireActionHeader"
               type="RequireActionHeaderAssertion,
                     Microsoft.Web.Services3, Version=3.0.0.0, Culture=neutral,
                     PublicKeyToken=31bf3856ad364e35" />
  </extensions>
<policy name="Policy4wci.com">
    <mutualCertificate11Security ...
    ...
</policy>
```

Codebeispiel 4.7.15: Benannte Assertions (extensions - Elemente) innerhalb der Policy-Datei

Der Verweis auf eine eigene Klasse wird durch das Hinzufügen einer benannten *Assertion*, eines *extension*-Elementes, gesetzt (Codebeispiel 4.7.16, *soapconventionservice*). Das Attribut *type* verweist auf den Namensraum (*CustomAssertion*), in dem sich die Klasse (*Soapconvention*) befindet. Anschließend wird die *Policy* angelegt, welche die benannte Assertion verwendet. Auf Codeebene wird diese, wie bereits in Codebeispiel 4.7.6 und 4.7.7 veranschaulicht, eingebunden.

```
<extensions>
    <extension name="soapconventionservice"
             type="CustomAssertion.SoapconventionService, CustomAssertion" />
    <extension name="authorization"
     type="Microsoft.Web.Services3.Design.AuthorizationAssertion,
           Microsoft.Web.Services3, Version=... />
    <extension name= ...
  </extensions>
  <policy name="Soapconvention">
    <soapconventionservice />
  </policy>
```

Codbeispiel 4.7.16: Definieren einer Assertion Class

Zum besseren Verständnis zeigt Abbildung 4.7.13, wie *PolicyAssertions*, die dazugehörigen Klassen und deren Verwendung innerhalb einer *Policy* miteinander in Beziehung stehen.

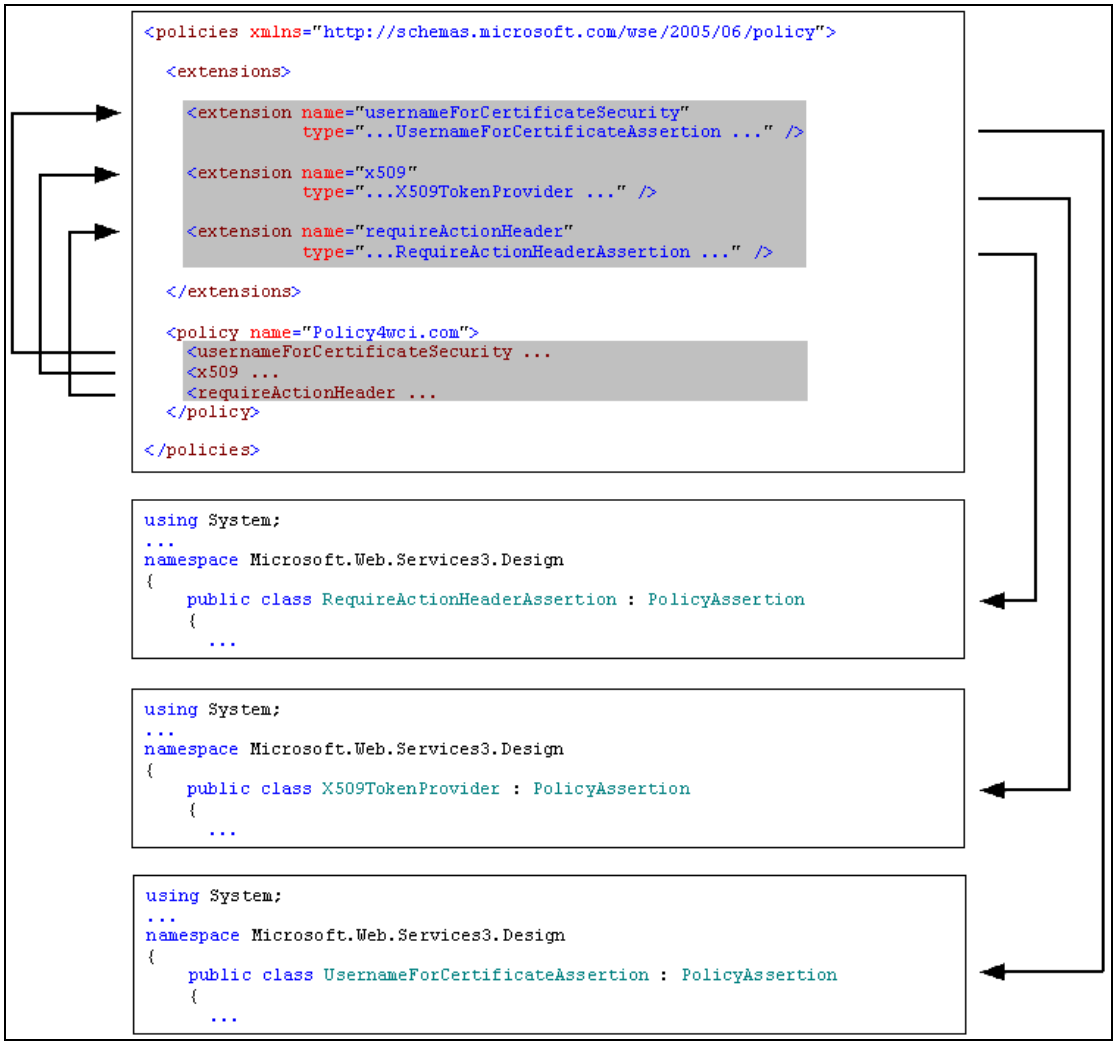

Abbildung 4.7.13: Klassen und deren Verwendung innerhalb einer Policy

Als nächster Schritt wird die Klasse *SoapconventionClient* angelegt. Diese erbt von der Klasse *PolicyAssertion*. Somit können die Methodenaufrufe der Ein- und Ausgangsfilter überschrieben werden. Das folgende Codebeispiel 4.7.17 zeigt dies am Beispiel des Web-Services, bei dem Ein- und Ausgangsfilter überschrieben weden. Die Methoden des Ein- und Ausgangsfilters des Clients sind Bestandteile derselben Klasse und müssen ebenfalls überschrieben werden. Da diese jedoch nicht implementiert werden, geben sie den Wert Null zurück. Eine weitere Methode der

Klasse *PolicyAssertion* ist *ReadXml*. Diese Methode liest vor dem Aufruf der Ein- beziehungsweise Ausgangsfilter die Informationen der *Policy* des Web-Services aus, anhand welcher der jeweilige Filter konstruiert wird.

```
namespace CustomAssertion
{
    public class SoapconventionService : PolicyAssertion
    {
        public override SoapFilter
        CreateClientInputFilter(FilterCreationContext context)
        {
            return null;
        }

        public override SoapFilter
        CreateClientOutputFilter(FilterCreationContext context)
        {
            return null;
        }

        public override SoapFilter
        CreateServiceInputFilter(FilterCreationContext context)
        {
            return new ServiceInputFilter();
        }

        public override SoapFilter
        CreateServiceOutputFilter(FilterCreationContext context)
        {
            return new ServiceOutputFilter();
        }

        public override void ReadXml(XmlReader reader,
        IDictionary<string, Type> extensions)
        {
            reader.Read();
        }
//...
```

Codebeispiel 4.7.17: Implementierung einer PolicyAssertion am Server

Die Implementierung der *Policy-Assertion* am Client erfolgt fast analog, allerdings geben die Methoden des Ein- und Ausgangsfilters des Clients eine Instanz der Klasse `ClientOutputFilter` bzw. `ClientInputFilter` zurück.

Anschließend erfolgt die Implementierung der Klassen *ServiceInputFilter* und *ServiceOutputFilter*. Innerhalb dieser Klassen kann auf den aktuellen Kontext der SOAP-Nachricht zugegriffen werden, wie im Codebeispiel 4.7.18 veranschaulicht wird. So können zum Beispiel bestimmte Eigenschaften des Zertifikates oder Bestandteile des Nachrichtenkopfes überprüft werden, zum Beispiel auf das Vorhandensein der Signatur eines Dritten.

```
public class ServiceInputFilter : SoapFilter
{
   public ServiceInputFilter(){ //...
   }

   public override SoapFilterResult ProcessMessage(
            SoapEnvelope envelope)
   {
     X509SecurityToken certToken =
     RequestSoapContext.Current.Credentials.UltimateReceiver.GetCl
     ientTo ken<X509SecurityToken>();

      if (certToken == null)
        throw new Exception("Zertifikatprobleme", new
            SoapException("Das verwendete Zertifikat stimmt nicht
            mit dem Serverzertifikat überein",
            SoapException.ServerFaultCode));

      X509Certificate cert =
      X509Certificate.CreateFromCertFile("../wciws.cer");

      if (certToken.Certificate.Equals(cert) &&
!certToken.IsExpired)
      {
         //...
      }
      //...
      return SoapFilterResult.Continue;
   }
}

public class ServiceOutputFilter : SoapFilter
{
    //...

    header.AppendChild(businnesID);
```

Codebeispiel 4.7.18: Anlegen und Verwendung von Ein- und Ausgangsfiltern

Die Implementierung eigener Sicherheitsszenarien kann jedoch noch vereinfacht werden, indem Bestandteile (*extensions*) standardisierter Sicherheitsszenarien, z.B. „Zertifikate for Zertifikate“, verwendet werden. Im Codebeispiel 4.7.19 wird der *Policy* „Soapconvention“ die *extension* „mutalCertificate11Security“ hinzugefügt. Diese legt fest, dass Empfängerinformationen, Zeitstempel und der Nutztext signiert und anschließend verschlüsselt werden. Die Antwort des Clients (Response) wird auf Gültigkeit der Signatur des Clients geprüft.

```
<policy name="Soapconvention">

  <soapconventionservice />

  <mutualCertificate11Security establishSecurityContext="true"
    renewExpiredSecurityContext="true"
    requireSignatureConfirmation="true"
    messageProtectionOrder="SignBeforeEncrypt"
    requireDerivedKeys="true" ttlInSeconds="300">
    <serviceToken>
      <x509 storeLocation="LocalMachine" storeName="My"
       findValue="CN=wci.com"
       findType="FindBySubjectDistinguishedName" />
    </serviceToken>
    <protection>
      <request signatureOptions="IncludeAddressing,
       IncludeTimestamp, IncludeSoapBody" encryptBody="true" />
      <response signatureOptions="IncludeAddressing,
       IncludeTimestamp, IncludeSoapBody" encryptBody="true" />
      <fault signatureOptions="IncludeAddressing,
        IncludeTimestamp, IncludeSoapBody" encryptBody="false" />
    </protection>
  </mutualCertificate11Security>
...
```

Codebeispiel 4.7.19: Verwendung vorgegebener extensions bei der Implementierung einer CustomPolicyAssertion

4.7.4 Interoperabilität zwischen .NET und Java

WSE generiert keine WSDL-Definitionen, das heißt, der Entwickler muss auch am Design des Clients beteiligt sein. Daher ist es von Vorteil, bei der Entwicklung eines Web-Services mit WSE auch auf dem Client oder anderen Web-Services WSE einzusetzen, welche auf den mit WSE entwickelten Web-Service zugreifen.

Diese Plattformabhängigkeit widerspricht allerdings sehr dem Gesamtkonzept von Web-Services, die verteilte und objektorientierte Programmierstandards vereinen und mit grundlegenden standardisierten Protokollen auf vielen Plattformen zur Verfügung stehen [Du05]. Für einen plattformübergreifenden Einsatz von WSE sind daher Toolkits notwendig, welche die erweiterten Web-Service-Protokolle verstehen. Für die Interoperabilität bei der Verwendung der Spezifikationen der GXA zwischen Java und .NET stehen zwei Varianten zur Verfügung:

1. Das IBM Web Services Toolkit zur Aktivierung der Funktionalität von *WS-Security* für Java [WSTK1]
2. Java Web-Service Developement Pack (Java WSDP) [WSTK2]

5. Zusammenfassung

Kapitel 3 stellte Anforderungen zur Einhaltung des Datenschutzes vor. Anhand der Meldepflicht wurde die Frage geklärt, in welchem Rahmen es möglich ist, den Datentransfer zu automatisieren. Außerdem wurden grundsätzliche Probleme des elektronischen Handels anhand von Treuhandservices aufgeführt und Parallelen zum System WorldCheckInn aufgezeigt. Betrachtet wurde das erweiterte Modell nach Wabner, welches anhand zweier Schuldvertragszentren die Überwachung und Dokumentation des gesamten Geschäftsvorganges vornimmt. In diesem Zusammenhang wurden die darauf aufbauenden Protokollvarianten nach [Schilha] vorgestellt und die Frage beantwortet, ob eine Umsetzung dieser unter WorldCheckInn möglich ist. Im letzten Abschnitt wurden elektronische Signaturen und deren Beweiswürdigung vorgestellt. Dabei wurden Voraussetzungen und Verpflichtungen einer Zertifizierungsstelle aufgeführt und hinsichtlich einer möglichen Umsetzung unter WorldCheckInn betrachtet.

Kapitel 4 soll als Leitfaden dienen, wie das System WorldCheckInn auf Basis der verwendeten Technologien weitgehend gegen Angriffe über das Internet geschützt werden kann. Zu diesem Zweck wurden theoretische Grundlagen vorgestellt und durch weitere Maßnahmen ergänzt, insbesondere solche, die einen möglichen Angriff abwehren.

5.1 Einhaltung der nicht-funktionalen Anforderungen

Die in Kapitel 3 besprochenen Punkte zeigen Anforderungen rechtlicher Rahmenbedingungen, die einzuhalten sind. Grundlegend gilt, dass die Weitergabe oder Verarbeitung personenbezogener Daten einer Zustimmung des Betroffenen bedarf, es sei denn, diese werden für statistische Zwecke ohne Bezug auf die Person

verwendet. Der Betroffene muss genauestens über die Verarbeitung und Weitergabe seiner Daten informiert werden und ohne Zwang seine Zustimmung dafür geben.

Das Speichern personenbezogener Daten unterliegt nicht nur den Richtlinien der Datenschutzbestimmungen, sondern ist sogar vielmehr als elementarer Bestandteil des Systems anzusehen, da bei Verlust oder Öffentlichwerden vertraulicher Informationen ein nicht zu wiedergutmachender (Image-) Schaden entsteht. Hinzu kommt der besondere Datenschutz bei der Übermittlung personenbezogener Daten, zum Beispiel beim Check-In.

Die Daten des Hotelgastes sollen dem Hotel nur für den Hotelaufenthalt zur Verfügung stehen. Der Datenempfänger (das Hotel) ist nach §4b, Absatz 6 BDSG auf den Verwendungszweck und den §35, Absatz 2 BDSG hinzuweisen, der ihn verpflichtet, die Daten nach Beendigung des Hotelaufenthaltes zu löschen.

Besonders wichtig ist die richtige Einbindung der Allgemeinen Geschäftsbedingungen, um nachweisen zu können, dass diese beim Abschließen einer Reservierung oder Registrierung neuer Datensender und -empfänger zur Kenntnis genommen wurden. Für Registrierungen (mit Vereinbarung des Rahmenvertrages) und Reservierungen gilt, dass der Verwender der AGB – in diesem Fall WorldCheckInn – die Beweislast trägt, dass der Kunde sein Einverständnis gegeben hat. Möchte sich der Verwender der AGB auf die darin festgelegten Rechte berufen, so muss er die wirksame Einbeziehung der AGB nachweisen. So reicht es nicht aus, einen einfachen Hinweis in Form eines Links auf die AGB zu geben. Vielmehr muss die AGB neben der Einwilligung zu den Datenschutzbestimmungen (§§ 4,4a BDSG) in den Prozess einer Reservierung eingebunden werden, so dass es zwingend erforderlich ist, zu bestätigen, dass die AGB zur Kenntnis genommen wurden, um die Reservierung abzuschließen.

Zudem ist es wichtig, das die AGB keine Unklarheiten enthalten, die missverstanden werden können, da dies immer zu Lasten des Verwenders der AGB geht. Dies muss man vor allem dann beachten, wenn die AGB in andere Sprachen übersetzt werden müssen.

5.2 Erfüllung der Meldepflicht

Ziel ist es, die Meldepflicht mit Hilfe beglaubigter Personendaten und qualifizierter elektronischer Signaturen computergestützt und medienbruchfrei zu erfüllen. Zur Erfüllung der Meldepflicht hinterlegt der Hotelgast vorab in einer Datenbank unter WorldCheckInn seine Personendaten. Diese umfassen auch die Meldedaten (Familienname, Geburtsort, Anschrift, Staatsangehörigkeit), die beim Check-In im Hotel benötigt werden.

Es stellte sich unter anderem heraus, dass sich die Expansion des Systems WorldCheckInn bereits auf nationaler Ebene als besonders schwierig gestaltet, da die Erfüllung der Meldepflicht länderspezifischen Regelungen unterworfen ist. Zur computergestützten Erfüllung der Meldepflicht im Hotel stellt sich folgendes Problem: „Die Pflicht zur handschriftlichen Ausfüllung trägt dem Charakter des Meldescheins als polizeiliches Fahndungsmittel Rechnung. Die Pflicht zur handschriftlichen Ausfüllung entfällt nur, wenn der Gast wegen einer körperlichen Behinderung oder aus anderen Gründen (z.B. Analphabet) dazu nicht in der Lage ist.“ (§18, Absatz 2 Satz 1–3, SächsMG). Die qualifizierte elektronische Signatur ist der handschriftlichen gleichgestellt, findet aber aufgrund der handschriftlichen Form als Fahndungsmittel keine Anwendung bei der Erfüllung der Meldepflicht. Auch kann die Unterschrift nicht durch andere biometrische Merkmale, wie zum Beispiel ein beglaubigtes Passbild oder ein bereits ausgefüllter, unterschriebener und eingescannter Hotelmeldeschein, ersetzt werden. „Warum das Gesetz ‚handschriftlich‘ fordert (Fahndungszwecke), ist für den Rechtsanwender nebensächlich. Er ist dem Gesetz unterworfen. Zuwiderhandlungen sind ggf. ordnungswidrig bzw. sogar strafbar.“ [Bö06] Allerdings trifft dies nicht auf alle Bereiche im Meldewesen zu. So ist zum Beispiel unter [heise.de03] zu lesen, dass die automatische Datenübermittlung und der Einsatz qualifizierter elektronischer Signaturen im Meldewesen bereits vorangeschritten sind. Die Forderung einer Handschrift, wie sie im §18, Absatz SächsMG oder §26, Absatz 2 des HMG verlangt wird, ist ein „vergessenes Relikt“ [Bö06] aus der Zeit, in der die elektronische Signatur der handschriftlichen noch nicht gleichgestellt war.

Denkbar und wünschenswert ist der Einsatz digitaler Signaturen, da diese zur Vereinfachung des Check-In beitragen könnten und einen Medienbruch vermeiden. Die Verantwortung zur Überprüfung der Gültigkeit und Zugehörigkeit der Signatur

bei einer Certificate Authority (CA) während des Reservierungsvorganges würde dann aber an WorldCheckInn übertragen werden.

5.3 Rechtssichere Reservierungen

Für garantierte Reservierungen muss die Identität des Reservierenden sichergestellt sein. Dies drängt sich vor allem dann auf, wenn es sich um einen neuen Datensender handelt. Da der Hotelier aber den einzigen Kontakt zum Reservierenden darstellt, ist es erforderlich, den Hotelier zur Überprüfung der Daten und den Hotelgast zur Vorlage eines Ausweispapieres aufzufordern. Gleiches gilt für die Überprüfung der Daten des Hotelgastes, wenn diese geändert oder erweitert werden. Der Hotelier kann aber nicht zur Kontrolle der Datensenderdaten verpflichtet und der Datensender zur Vorlage des Ausweises gezwungen werden.

Wünschenswert ist es daher, auf beglaubigte Daten zurückgreifen zu können (vor allem, wenn es sich um die Übermittlung von Daten für die Meldepflicht im Hotelwesen handelt). Zentraler Bestandteil des Modells nach Wabner sind die Vertrauenszentren, welche für die Zusammenführung der Teilschuldverträge sorgen und somit für den Abschluss des Gesamtvertrages. Werden die Schuldvertrags-zentren als staatlich überwachte Instanzen angesehen, so könnten diese die Identität der Vertragsparteien durch den Bezug von Meldedaten sicherstellen.

Mit Hilfe von elektronischen Signaturen kann die Identität des Signierenden sichergestellt und die Unabstreitbarkeit gewährleistet werden. Unter anderem werden diese eingesetzt, um in Systemen, wie zum Beispiel Governikus (http://www.governikus.de/), Verträge online zu signieren. Solche Systeme unterstützen bereits die Verwendung von Signaturkarten und Lesegeräten. Das Signieren von Verträgen ist auch schon von führenden Anbietern wie Adobe (www.adobe.com) umgesetzt und verhindert das Ausdrucken, Unterschreiben und Versenden des Vertrages per Post oder Fax und so den damit verbundenen Medienbruch.

Dem breiten Einsatzfeld von Zertifikaten steht jedoch die fehlende Akzeptanz im elektronischen Handel gegenüber. Hinzu kommt, dass nicht immer die Garantie besteht, dass die Zertifikate aus einer seriösen Quelle stammen. Eine Alternative bietet das PostIdent-Verfahren der Deutschen Post (www.signtrust.de). Dieses Verfahren fordert die Identifikation bei einer Außenstelle der Deutschen Post. Daher kann nachgewiesen werden, ob die angegebene Adresse und persönlichen Daten des Empfängers stimmen. Allerdings ist die Nutzung des Verfahrens für den Empfänger des Zertifikates kostenpflichtig und die Verbreitung auf das Innland beschränkt.

Ausgehend davon, dass die Identität des neuen Käufers nicht mit einem Zertifikat sichergestellt werden kann, so müsste eine andere Maßnahme getroffen werden: Der Käufer leistet einen Vertrauensvorschuss, indem er vorab einen Teil seiner Schuld erfüllt, sofern dafür die Voraussetzung gegeben ist. Damit ist allerdings ein Bruch in der nach [Weiser] und [Wabner] eingeführten Vertrauenssymmetrie getan. Dieser Vertrauensvorschuss geschieht allerdings nur gegenüber dem Zahlungsmittelschuld-vertragszentrum. Ist das ZSVZ staatlich überwacht oder durch diesen akkreditiert und würden alle Vorgänge sicher protokolliert, so könnte er diesen Vertrauens-vorschuss unter Vorbehalt eingehen. Der Vertrauensvorschuss stellt allerdings auch nicht die Rechtssicherheit wieder her.

Anhand der gezeigten Probleme wird deutlich, dass unter WorldCheckInn Verträge, wie im Modell nach Wabner als erforderlich angesehen, nicht abgeschlossen werden können. Nach Weiser entspricht WorldCheckInn somit eher der Funktion eines Markplatzes im Modell nach Wabner. Der Marktplatz stellt (neben den beiden Schuldvertragszentren) ein drittes Vertrauenszentrum dar, welches die Anonymi-sierung und Dokumentation bei der Erstverhandlung der Vertragsparteien ermöglicht. Zu den Aufgaben eines Marktplatzes gehören nach Weiser die Beraterfunktion, die Auswahl geeigneter Vertragsvorlagen und Vertragsparameter sowie die Auswahl eines geeigneten Schuldvertragszentrums. Hinzu kommen die sichere Aushandlung der Geschäftsidentifikationsnummer, die Authentifizierung beider Parteien und die Dokumentation der Erstverhandlung.

5.4 Gewährleistung der Sicherheit

Vertrauenszentren suggerieren anhand von Auszeichnungen (www.trustedshops.de, www.safer-shopping.de) und Zertifikaten einen rechtssicheren Warenhandel und Vertragsabschluss. Dabei verbergen sich hinter diesen Zentren Software-Anwendungen auf Webservern, die denselben technischen Hürden wie andere eCommerce-Anwendungen, zum Beispiel ein Onlineshop, gegenüberstehen. Für die technische Sicherheit kann außer Betracht gelassen werden, ob ein Vertragsabschluss über staatliche Instanzen hinweg oder über einen privaten Drittanbieter geschlossen wird. Grundlegende technische Hürden, welche es zu überwinden gilt, sind gleich:

- Wie wird die Verfügbarkeit sichergestellt?
- Wie wird die Vertraulichkeit gewahrt?
- Wer oder was sichert die Integrität?

Im Laufe der Arbeit hat sich bestätigt, dass Abwehrmechanismen für Computersysteme nie einen hundertprozentigen Schutz, zum Beispiel vor Datenverlust, bieten. Die wesentlichen Dinge zur Sicherung der Verfügbarkeit reduzieren sich auf die redundante Auslegung, wie bereits in Abbildung 4.4.2 vereinfacht dargestellt, und die regelmäßige Sicherung der Datenhaltung des Systems. Um den Grundsatz *„Nie mehr ein Formular zur Datenerfassung ausfüllen zu müssen*“ zu wahren, werden Backups nötig. Da das System im Laufe der Weiterentwicklung durch neue sensible Verfahren, wie zum Beispiel die Einführung eines Bonussystems, erweitert wird, wäre ein Datenverlust existenzbedrohend. Leider wird aber auch der Sicherheit dieser Backups zu wenig Beachtung geschenkt. Dafür befindet sich auf der beiliegenden CD im Unterverzeichnis *backups/* das Dokument *backup_methoden.doc*. Dieses Dokument gibt einige Hinweise für die Wahl des richtigen Backups. Es werden Backupmethoden verglichen und Erläuterungen für die sichere Aufbewahrung und Wiederherstellung gegeben.

5.5 Wahrung der Vermittlerfunktion

WorldCheckInn wurde mit den Eigenschaften und Aufgaben eines Softwareagenten verglichen, wobei einige Ähnlichkeiten festgestellt wurden. Ziel solcher Softwareagenten ist die Entlastung der Menschen, um zum Beispiel automatische Willenserklärungen zu generieren. Das Vertragsrecht im BGB bezieht sich jedoch nur auf natürliche Personen. Vertragsschlüsse über Softwareagenten gelten aber als Willenserklärungen, „[...] denn in der Einrichtung des Agenten selbst liegt eine willentliche Vorbereitungshandlung, aufgrund derer Erklärungen des Agenten dem Anwender zugerechnet werden können.“ [Co02]. Für Softwareagenten, wie sie im Projekt SESAM gedacht sind, haftet also die natürliche Person für die Willenserklärungen des Agenten. Zwischen WorldCheckInn und dem Hotel besteht ein Vertrag, der es erlaubt, garantierte Reservierungen anstelle des Hoteliers vorzunehmen. Hier erweist es sich als wichtig, besondere Regelungen zu treffen, da der Vertrag gegenüber dem Reservierenden verbindlich ist. Wird zum Beispiel beim Check-In eine Überbuchung festgestellt, muss WorldCheckInn gegenüber dem Hotel und dem anreisenden Hotelgast haften.

Deshalb muss einerseits also noch geklärt werden, ob es möglich und durchzusetzen ist, vertragliche Regelungen zwischen Datenempfänger und WorldCheckInn zu treffen, so das sich WorldCheckInn gegenüber Probleme technischer Natur absichern kann. Anderseits ergibt sich aus Abschnitt 3.2.2 die Frage, inwiefern vertraglich festgelegt werden kann, dass der Datensender zum Vorlegen eines Ausweispapieres bei einem Datenempfänger verpflichtet werden kann.

5.6 Ausblick

5.6.1 Zukünftige Dienste

Bisher wurden noch keine Hinweise gegeben, wie sichergestellt werden kann, dass alle Aktionen eines Geschäftsprozesses zwischen Sender und Empfängers ausgeführt werden. Ein Reservierungsvorgang unter WorldCheckInn beinhaltet das Erfassen der Reservierung in der Datenbank des Hotels sowie in der Datenbank des Systems WorldCheckInn. Es muss sichergestellt werden, dass das Zimmer während des Reservierungsvorganges nicht durch einen anderen Prozess reserviert wird. Passiert dies dennoch, so müssen alle Vorgänge rückgängig gemacht werden. Ähnlich verhält es sich bei dem Check-In Vorgang oder dem Registrieren neuer Kunden.

Neben den in Abschnitt 4.7.1 vorgestellten Spezifikationen der GXA gibt es noch die Spezifikation W*S-Coordination* zur Verbindung mehrerer Web-Services verschiedener Unternehmen. So wird *WS-Coordination* dann eingesetzt, wenn ein übergeordneter, globaler Web-Service benötigt wird, um einen Prozess (den Aufruf eines Dienstes) zu überwachen, bei dem mehrere Web-Services verschiedener Unternehmen in Anspruch genommen werden. Dies ist vor allem für die Ausweitung des Systems auf weitere Dienste interessant, wie zum Beispiel die Flugticket-buchung. Ein Datensender möchte ein Hotel reservieren, unter der Bedingung, dass zum Anreisetag noch ein Flugticket zur Verfügung steht. Bekommt er das Flugticket nicht, möchte er auch nicht reservieren und umgekehrt.

Mit Hilfe dieser Spezifikationen kann die Sicherheit gegeben werden, dass alle mit einer Reservierung verbundenen Aktionen ausgeführt oder im Falle eines unerwarteten Fehlers alle Vorgänge rückgängig gemacht werden können.

5.6.2 Vertragsschlüsse in einem abgeschlossenen System

Die technische Herausforderung des Systems WorldCheckInn besteht darin, alle Geschäftsvorgänge sicher zu protokollieren, so dass alle Aktionen, zum Beispiel eine Reservierung oder bestimmte Zimmerpreise (zu einem bestimmten Zeitpunkt) unabstreitbar und zuordbar sind.

Das System Askemos [Askemos] der Firma „softeyes“ (http://www.softeyes.net) bietet ein solches System. Askemos ist ein festes, abgeschlossenes System ohne Superuser, dass durch byzantinische Abstimmung (Zwei-Drittel Mehrheit) entscheidet, ob ein Dokument oder ein Vorgang echt oder gefälscht ist. Ein System ohne Superuser bedeutet, dass es keine natürliche Person gibt, die auf das ganze System, bestehend aus mindestens drei Agenten (Servern), einen alleinigen und allumfassenden Zugriff hat. Dadurch ist das System nicht durch den Menschen korrumpierbar. Aus technischer Sicht kommunizieren die Agenten (Server) untereinander und treffen Entscheidungen durch den Vergleich von Signaturen, Protokollen und Prüfsummen. Alle Prozesse sind atomar, eindeutig und nicht lokalisierbar. Der Anwender kommuniziert über einen Proxy mit diesem System. Die Agenten entsprechen einer Notarfunktion. Alle Dokumente (sowie alle Vorgänge) werden durch jeden Agenten protokolliert und mittels individueller Signaturen gesichert. Dadurch wird die Unabstreitbarkeit aller Vorgänge gewährleistet. Vorgänge in Askemos sind zum Beispiel Rechtevergaben. Jeder Anwender verfügt über eine bestimmte Menge von Rechten, von denen er eine Teilmenge an andere Anwender des Systems abgeben kann, zum Beispiel Lesezugriff auf vertrauliche Dokumente.

Ein solches System, beziehungsweise die zugrundliegende Idee, erweist sich für eine Anwendung unter WorldCheckInn in mehreren Punkten als sehr nützlich. So kann der gesamte Reservierungsvorgang im System festgehalten werden, angefangen bei der Reservierung bis zur Abrechung nach dem Aufenthalt im Hotel. So können, wie bereits besprochen, die Daten des Reservierenden nur die Dauer des Hotelaufenthalts dem Hotel zur Verfügung gestellt werden. Zur Kopie des Datensatzes bedarf es in diesem Falle einem erhöhten Aufwand. Für den Reservierungsvertrag können im System Reservierungsdaten festgehalten werden, wie zum Beispiel An- und Abreisedatum, der vereinbarte Zimmerpreis oder zusätzliche Vergünstigungen. Das Hotel, als weiterer Anwender des Systems, würde im Gegenzug das Zimmer zu den vereinbarten Konditionen als reserviert verbuchen.

5.7 Schlussworte

WorldCheckInn erweist sich dahingehend als innovativ, da es die Chancen des Internets zu nutzen weiß. In der Weiterentwicklung sollen neue Dienste hinzukommen, die alle im selben System vereint sind. Ziel ist es, diese automatisiert, einfach und medienbruchfrei auf Basis rechtssicherer Verträge in Anspruch nehmen zu können. Kritikpunkt ist jedoch die Erstellung von Kundenprofilen für andere Unternehmen oder das *Scoring* zur Ermittlung der Bonität der Kunden, wie es unter WorldCheckInn geplant ist. Verbraucherorganisationen und Datenschutzbehörden konnten bisher nur ungenügend die Persönlichkeitsrechte der Menschen, zum Beispiel die Wahrung ihrer Anonymität, durchsetzen. Der Verbraucher ist sich meistens nicht im Klaren darüber, was mit seinen Daten geschieht. Jedoch schreitet die Forderung nach mehr Transparenz und Persönlichkeitsrechten voran. So ist ein Trend abzusehen, der eventuell einen größeren Kundenzuwachs von WorldCheckInn verhindert, wenn das bisherige Geschäftsmodell in dieser Form hinsichtlich des Umgangs mit personenbezogenen Daten beibehalten wird.

Abkürzungsverzeichnis

AGB	Allgemeine Geschäftsbedingungen
ACL	Access Control Lost
ASP	Active Server Pages
BDSG	Bundesdatenschutzgesetz
BGB	Bürgerliches Gesetzbuch
BSI	Bundesamt für Sicherheit in der Informationstechnik
CA	Certificate Authority
CLR	Common Language Runtime
CLS	Common Language Specification
CRL	Certificate Revocation Lists
CTS	Common Type System
DNS	Domain Name System
EU	Europäische Union
FA	Fachhochschule
FTP	File Transfer Protocol
GID	Geschäftsidentifikationsnummer
GXA	Global Web-Service Architecture
HMG	Hessisches Meldegesetz
IBM	International Business Machines

IETF	Internet Engineering Task Force
IIS	Internet Information Server
LDAP	Lightweight Directory Access Protocol
MBSA	Microsoft Baseline Security Analyzer
MOM	Microsoft Operations Manager
MSIL	Microsoft Intermediate Language
MSN	Microsoft Network
OASIS Standards	Organization for the Advancement of Structured Information
PIN	Persönliche Identifikationsnummer
RDS	Remote Data Services
SächsMG	Sächsisches Meldegesetz
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SUN	SUN Microsystems
TDDSG	Teledienstedatenschutzgesetz
TLS	Transport Layer Security
W3C	World Wide Web Consortium
WSE	Web Service Enhancements
XSS	Cross Site Scripting

Glossar

Authentifizierung
Authentifizierung stellt die Identität von Benutzern sicher.

Autorisierung
Gewährleistet, dass nur berechtigte Personen Programme nutzen oder auf bestimmte Daten zugreifen können.

Bestandsdaten
Bestandsdaten sind personenbezogene Daten, die für die Begründung, Änderung und inhaltliche Gestaltung eines Vertrages erforderlich sind. Das Speichern von Bestandsdaten ist notwendig, um bei einer späteren Einsichtnahme Reklamationen, zum Beispiel nicht erbrachte Leistungen, geltend zu machen.

Datenempfänger
Im Allgemeinen ein Dienstleistungsunternehmen, das Aufgrund von Abrechnungszwecken Interesse an den Stamm- und Abrechnungsdaten des Kunden hat, wenn dieser die Dienste des Unternehmens in Anspruch nimmt.

Datensender
Der in (1) aufgeführte Kunde, der Dienstleistungen des Unternehmens in Anspruch nimmt.

Juggernaut
ist ein *Hijacking-Tool*, mit dessen Hilfe man TCP-Sitzungen durch Ausnutzen von Sitzungs-Daten, wie zum Beispiel *Cookies* kompromittieren kann.

Ladungsfähige Adresse
Ist der tatsächliche Wohnort einer natürlichen Person. Zu einer ladungsfähigen Adresse gehört das Land, der Ort, die Postleitzahl und der Straßenname.

Personenbezogene Daten
Nach §3, Absatz 1 BDSG sind personenbezogene Daten Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren

natürlichen Person (Betroffener). Dies kann zum Beispiel der Name, Vorname, Familienstand, Beruf, das Geburtsdatum oder die Adresse sein.

Penetrationstest
Unter einem Penetrationstest versteht man im Auftrag eines Unternehmens durchgeführte Angriffsversuche auf ein System oder Netzwerk, mit dem Ziel, Sicherheitslücken zu finden.

Signatur
Bei einer elektronischen Signatur handelt es sich um elektronische Daten, welche die Authentizität und Integrität von elektronischen Informationen, meist elektronische Dokumente, sicherstellen soll. Darüber hinaus soll eine elektronische Signatur die Identität des Signierenden gewährleisten. Diese Merkmale sollen wiederum mit Hilfe der elektronischen Signatur verifizierbar (d.h. überprüfbar) sein.

Simple Object Access Protocol (SOAP)
Ist ein Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können. SOAP stützt sich auf die Dienste anderer Standards, XML zur Repräsentation der Daten und Internet-Protokolle der Transport- und Anwendungsschicht zur Übertragung der Nachrichten. Der gängigste Kombination ist SOAP über http und TCP.

Stammdaten
Stammdaten sind wichtige Basisdaten einer natürlichen Person, wie z.B. Name, Vorname, Geburtsdatum.

SYN-Flooding
Ist das Ausnutzen des Handshake-Protokolls einer TCP/IP-Verbindung. Beim Aufbau einer Client-Server-Kommunikation sendet zuerst der Client ein Paket an den Server, welches eine gefälschte nichtexistente Absenderadresse enthält. Anschließend sendet der Server ein acknowledge (ACK) zurück und wartet auf ein ACK des Clients. Da der Server kein ACK vom Client erhält, sendet er das Paket erneut, bis er eine Antwort erhält.

Time-to-live (TTL)
Ist die maximale Lebensdauer einer Nachricht.

Trojaner
Als nützlich getarnte Programme die unerwünschte Funktionen auf dem Computer des Betroffenen ausführen.

Uniform Resource Locator (URL)
Ein Identifikator für eine Ressource in einem Netzwerk, zum Beispiel http://www.worldcheckinn.com/menu1.html

Web-Service
Software-Anwendung, die mit einem Uniform Ressource Identifier (URI) eindeutig identifizierbar ist. Die Schnittstellen sind durch XML definiert. Unterstützt wird die direkte Interaktion mit anderen Software-Agenten unter Verwendung XML-basierter Nachrichten durch den Austausch über internetbasierte Protokolle.

Web Service Description Language (WSDL)
Mit Hilfe von WSDL-Dokumenten können XML-Webdienste beschrieben werden. Ein WSDL-Dokument enthält Informationen darüber, welche Schnittstellen der Web-Service anbietet, mit welchen Parametern diese aufgerufen werden können und welche Werte diese zurückgeben [Fr03].

Zertifikat
In einer Public-Key-Infrastruktur (PKI) dient ein Zertifikat dem Nachweis, das ein öffentlicher Schlüssel eines asymmetrischen Verschlüsselungsverfahrens zu einer angegebenen Person, Institution oder Maschine gehört. Mit Hilfe des Zertifikates können weitere Daten verschlüsselt und signiert werden und somit zum einen die Echtheit (Authentizität) und die Vertraulichkeit (Integrität) der Daten Dritten gegenüber garantiert werden.

Literaturverzeichnis

[Askemos] Wittenberger, Jörg. softeyes GmbH (http://www.softeyes.net/). Askemos – eine verteilte Umgebung. http://wwwm.htwk-leipzig.de/%7Em6bast/RIVL06/Wittenberger060607.pdf (27.06.2006)

[ASPHeute] ASPHeute – Der tägliche Artikel zu Active Server Pages. Web-Services und WS-Security – Authentifizierung anhand Benutzername und Passwort. http://www.aspheute.com/artikel/20030502.htm (13.01.2006)

[BDSG] Bundesdatenschutzgesetz (BDSG) 2001. http://www.datenschutzzentrum.de/material/recht/bdsg2001/bdsg2001.htm (19.10.2005)

[Bö06] Felix Böllmann. Wissenschaftlicher Mitarbeiter Juristenfakultät Universität Leipzig. Persönliche Mitteilung per eMail zur Anfrage ob es möglich sei, die handschriftliche Erfüllung der Meldepflicht durch andere biometrische Daten zu ersetzen. (17.07.2006)

[BSI] Bundesamt für Sicherheit in der Informationstechnik. Das IT-Grundschutzhandbuch. http://www.bsi.bund.de/gshb/ (20.02.2006)

[BSI01] Bundesamt für Sicherheit in der Informationstechnik. Das IT-Grundschutzhandbuch. Outsourcing. http://www.bsi.de/gshb/deutsch/baust/b01011.htm (28.07.2006)

[BSI02] Bundesamt für Sicherheit in der Informationstechnik. Das IT-Grundschutzhandbuch. Entfernen von Beispieldateien und Administrations-Skripte des Internet Information Servers (IIS). http://www.bsi.de/gshb/deutsch/m/m04186.html (30.07.2006)

[Co02] Cornelius, Kai. MMR – MultiMedia und Recht. Zeitschrift für Informations-, Telekommunikations- und Medienrecht. 2002. S. 353 . Zitiert nach Hoeren, Thomas Prof. Dr. . Institut für Informations-, Telekommunikations- und Medienrecht. Skript Internetrecht S. 251.

http://www.uni-muenster.de/Jura.itm/hoeren/material/Skript/skript_Juni2006.pdf (27.07.2006)

[Cz] Codezone – News und Foren für .NET-Entwickler. GXA macht Web-Services Salonfähig. http://www.00001001.ch/NewsSummary/Kolumnen/724.aspx (10.01.2006)

[DEHOGA] Deutscher Hotel- und Gaststättenverband e.V. . Beherbergungsvertrag. http://www.dehoga berlin.de/home/beherbergungsvertrag_1020_924.html (28.10.2005)

[Du05] Duske, Pierre. Diplomarbeit. .NET Security. Fachbereich Informatik, Mathematik und Naturwissenschaften. HTWK Leipzig. April 2005

[ebay.de] eBay – Der weltweite Online-Marktplatz. Diskussions-Foren. Sammelliste für falsche Escrow (Treuhand)-Seiten. http://forums.ebay.de/thread.jspa?threadID=2291731&start=0 (26.07.2006)

[ECIN] ECIN – Electronic Commerce Info Net. Vertragsabschluss im Internet. Zustandekommen eines Vertrages. http://www.ecin.de/recht/vertrag/index-2.html (31.07.2006)

[Fr03] Freeman, Adam. .NET XML Webdienste Schritt für Schritt. Microsoft Press. 2003

[heise.de01] heise online. News vom 24.02.2005. Vorsicht bei kostenlosen SSL-Zertifikaten. http://www.heise.de/newsticker/meldung/56750 (29.07.2005)

[heise.de02] heise online. News vom 20.02.2006. heisec-Konferenz: Das sichere Firmennetz. http://www.heise.de/newsticker/meldung/69829 (20.02.2006)

[heise.de03] heise online. News vom 28.02.2006. Berlin novelliert Meldegesetz: Einfachere Anmeldung – auch im Netz. http://www.heise.de/newsticker/meldung/70179 (27.07.2006)

[HMG] Hessisches Meldegesetz. http://www.datenschutz.hessen.de/Gesetzestexte/_virdat/HMG.htm (11.07.2006)

[Hot] Hotelverband Deutschland (IHA). (Beherbergungsvertrag). http://www.hotellerie.de/home/page_sta_120.html (28.10.2005)

[hrs.de] Hotel Reservation Service (HRS). Hotel-Reservierungssystem für Geschäfts- und Privatreisende. www.hrs.de (31.07.2006)

[IIS] Der Internet Information Server (Microsoft). Authentifizierungs-methoden. http://msdn.microsoft.com/library/deu/default.asp?url=/library/DEU/vsent7/html/vxconIISAuthentication.asp (18.07.2005)

[iloxx.de] iloxx AG. Logistische Transaktionen und sicherer Online-Handel. http://www.iloxx.de/ (29.07.2006)

[Jov04] Jovanovic, Nenad. Diplomarbeit. Entwicklung eines Webbasierten Institutsinformationssystems. Oktober 2004. http://www.infosys.tuwien.ac.at/Staff/enji/msc_thesis_jovanovic.pdf (09.08.2005)

[KT] MSDN-Library. Web Service Enhancements (2.0) KerberosToken Class. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wseref/html/T_Microsoft_Web_Services2_Security_Tokens_KerberosToken.asp (12.01.2006)

[Lo02] Patrick A. Lorenz. ASP.NET Grundlagen und Profiwissen. Hanser Verlag. 2002

[MBSA] Der Microsoft Baseline Security Analyzer (MBSA). http://www.microsoft.com/technet/security/tools/mbsahome.mspx (21.11.2005)

[MSDN] Microsoft Developer Network (MSDN) Bibliothek. http://msdn2.microsoft.com/de-de/default.aspx (31.07.2006)

[MSDNTV] MSDN TV. Web Service Enhancements (WSE) 3.0 and Secure Web Services. http://msdn.microsoft.com/msdntv/episode.aspx?xml =episodes/en/20051027WSE3MF/manifest.xml (31.07.2006)

[MSTN] Microsoft TechNet. Portal für System-, Datenbank- und Netzwerkadministratoren zur sicheren Installation, Konfiguration und Wartung von IT-Systemen bei der Verwendung von Microsoft-Produkten. http://www.microsoft.com/germany/technet/sicherheit/default.mspx (17.11.2005)

[MSTN01] Microsoft TechNet. Portal für System-, Datenbank- und Netzwerkadministratoren zur sicheren Installation, Konfiguration und Wartung von IT-Systemen bei der Verwendung von Microsoft-Produkten. Administrations-Skripte des Internet Information Servers (IIS). http://support.microsoft.com/default.aspx?scid=kb;en-us;232449&sd=tech

[MSTNWS2003] Windows Server TechCenter. Windows Server 2003-Bilbiothek. http://www.microsoft.com/technet/prodtechnol/windowsserver2003/default.mspx (17.11.2005)

[nethistory] dotnetframework.de – Die unabhängige deutsche Info-Site zum Microsoft .NET Framework. Geschichte und Zukunft. http://www.dotnetframework.de/dotnet/DOTNET_Geschichte_Zukunft.aspx (09.08.2005)

[OiO] OiO – Orientation in Objects. Unternehmen für Software-Entwicklung, Schulung und Beratung. Web-Service Spezifikationen. http://www.oio.de/public/xml/web-service-specifications.htm (10.01.2006)

[O'Neill03] O'Neill, Mark. Web Services Security. McGraw-Hill. 2003

[passport] Microsoft Passport-Netzwerk. http://www.passport.net

[Ra06] Dr. Raabe, Oliver. Universität Karlsruhe. In einem persönlichen Gespräch nach dem Studium generale „Rechtssicherheit im Netz“. 21. Juni 2006

[RIT] Recht-im-Tourismus.de . Der Beherbergungsvertrag. http://www.recht-im-tourismus.de/Ausbild/Lektion4IV Beherbergung.html (26.07.2006)

[SächsMG] Sächsisches Meldegesetz. Belz/Rimmele/Wunsch. Deutscher Gemeindeverlag. 1995

[Sch02] Schwichtenberg, Holger. ASP.NET Entwicklerhandbuch. Microsoft Press. August 2002

[SCW] Security Configuration Wizard (SCW). Assistent zur Konfiguration der Sicherheit des Windows Server 2003. http://www.microsoft.com/windowsserver2003/technologies/security/configwiz/default.mspx (31.07.2006)

[SecGuideA] http://download.microsoft.com/download/9/2/3/923d72fb-0076-49b6-96c4-aac1c255a60e/SecurityGuide_Chapter08_ASP.NET_ Security.pdf (18.11.2005)

[SecGuideB] Prüfliste zum Schützen eines Webservers. http://www.microsoft.com/germany/msdn/library/security/ErhoehenDerSicherheitVonWebanwendungen/secmod104.mspx (18.11.2005)

[SESAM] Universität Karlsruhe. Das Projekt SESAM – Selbstorganisation und Spontanität in liberalisierten und harmonisierten Märkten. http://www.sesam.uni-karlsruhe.de. (30.07.2006)

[Ta03] Tamm, Gerrit. Konzepte in eCommerce-Anwendungen. SPC Teia Lehrbuch Verlag. 2003

[TI04] Schneider, Uwe / Werner, Dieter. Taschenbuch der Informatik. Hanser Fachbuchverlag. April 2004

[Wa03] Wabner, Thomas. Ein Schuldvertragsmodell für den elektronischen Handel. Tagungsband LIT‘03. Akademische Verlagsgesellschaft Aka GmbH Berlin. 2003

[webhits] WebHits. Erstellung von Webstatistiken. Statistik über die Verwendung von Cookies. http://www.webhits.de/webhits/browser.htm (15.12.2005)

[Weiser] Weiser, Michael. Diplomarbeit. Rechtsverbindlicher Handel im Internet auf Basis sicherer Verzeichnisdienste. Fachbereich Informatik, Mathematik und Naturwissenschaften. HTWK Leipzig. 2004

[Wey02] Weyer, Christian. XML Web Service Anwendungen mit Microsoft .NET. Addison Wesley-Verlag. 2002

[Wirtz] Wirtz, Bernd. Electronic Business. 2.Auflage. Wiesbaden 2001

[wiki:CRL] Wikipedia – die freie Enzyklopädie. Certificate Revocation List (CRL, dt.: Zertifikatsperrliste). http://de.wikipedia.org/wiki/CRL (28.07.2006)

[wiki:Dienstleistung] Wikipedia – die freie Enzyklopädie. Dienstleistung. http://de.wikipedia.org/wiki/Dienstleistung (28.07.2006)

[wiki:SOAP] Wikipedia – die freie Enzyklopädie. Simple Object Access Protocol (SOAP). http://de.wikipedia.org/wiki/SOAP (28.07.2006)

[wiki:Virtuelles Kraftwerk] Wikipedia – die freie Enzyklopädie. Virtuelles Kraftwerk. http://de.wikipedia.org/wiki/Virtuelles_Kraftwerk (21.06.2006)

[WSE] MSDN-Library. Web Service Enhancements. http://msdn.microsoft.com/webservices/webservices/building/wse/ (10.01.2006)

[WSELab] Web Services Enhancements 3.0 Hands On Lab. Exploring Security. http://www.microsoft.com/downloads/details.aspx?FamilyID=9acd1f8e-97e2-43e2-b484-a74a014a8206&DisplayLang=en (31.07.2006)

[WSTK1] IBM alphaWorks. Web-Service Toolkit. http://www.alphaworks.ibm.com/aw.nsf/FAQs/webservicestoolkit (31.07.2006)

[WSTK2] Sun Developer Network. Web-Services. Securing Web Services and the Java WSDP 1.5 XWS-Security Framework. http://java.sun.com/developer/technicalArticles/WebServices/security (31.07.2006)

Inhalt der beiliegenden CD

Die CD enthält Dokumente und Bilddateien, die im Rahmen der Diplomarbeit und der Weiterführung des Projektes entstanden sind.

Verzeichnis	**Inhalt**	**Beschreibung**
/	readme.txt	Beschreibt den Inhalt der beiliegenden CD
diplomarbeit/	diplomarbeit.pdf	Beinhaltet die vorliegende Diplomarbeit
pflichtenhefte/	schnittstellen_tprovider.pdf (Anlagen: activity4tprovider.gif, datenvolumen_checkin.xls)	Pflichtenheft für Terminal-Provider, in dem auf Schnittstellen zwischen Terminal-Provider und WorldCheckInn eingegangen wird
backups/	backups.pdf	Dokument mit Hinweisen zur richtigen Wahl von Backupmethoden und -systemen
wse/	wse_installation.pdf (Anlagen: clientcert.bat, servercer.bat)	Enthält eine Installationsanleitung für WSE unter .NET mit Client- und Server-Zertifikat
openldap/	wci_vzdienst.pdf (Anlagen: wci_vzdienst.shema, wci_vzdienst_class.gif	Verzeichnisdienstdokumentation mit Shema-Datei für OpenLDAP und Klassendiagramm

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Diese Diplomarbeit hat in der vorliegenden Form und auch auszugsweise bisher keiner Prüfbehörde vorgelegen.

_____	_____
Ort, Datum	Unterschrift