

Eine Architektur zur Abrechnung von Web Services

Markus Burghardt, Nick Gehrke, Prof. Dr. Matthias Schumann

Institut für Wirtschaftsinformatik Abteilung II
Universität Göttingen
Platz der Göttinger Sieben 5
37073 Göttingen
mburgha@uni-goettingen.de
ngehrke@uni-goettingen.de
mschuma1@uni-goettingen.de

Abstract: Die Web Service Technologie bietet die Möglichkeit, Anwendungen verteilt und unabhängig von Programmiersprache und Plattform zu implementieren, wobei weit verbreitete Standards wie HTTP oder XML als Basis verwendet werden. Der Erfolg der Web Service Technologie hängt jedoch nicht nur von diesen technischen Determinanten ab, sondern auch ökonomische Aspekte wie beispielsweise die Abrechnung der Inanspruchnahme eines speziellen Web Services können als wichtige Kriterien für die Durchsetzung aufgefasst werden. Im vorliegenden Beitrag wird die Architektur eines Abrechnungsdienstes für Web Services vorgestellt. Hierfür werden zunächst die beiden Möglichkeiten einer Abrechnung analysiert und die notwendigen Anforderungen an einen solchen Abrechnungsdienst erarbeitet. Darauf aufbauend wird eine Architektur für einen Abrechnungsdienst für Web Services vorgestellt.

1. Einleitung

Web Services stellen eine moderne und neue Möglichkeit zur Realisierung verteilter Anwendungen dar. Dabei werden sie nicht als eine neue Technologie aufgefasst, sondern können als eine Kombination bereits etablierter Technologien angesehen werden, die im Zusammenspiel plattform- und programmiersprachenunabhängige verteilte Anwendungen ermöglichen. Web Services stehen nicht in einem direkten Konkurrenzverhältnis zu bereits verbreiteten Technologien für verteilte Anwendungen wie z.B. CORBA oder RMI, denn mit CORBA und RMI ist die Realisierung objektorientierter Anwendungen möglich, wohingegen Web Services vom Grundsatz her nur entfernte Funktionsaufrufe unterstützen [Kn02]. Web Services sollten also als Ergänzung zu den bereits genannten anderen Technologien gesehen werden. Inwieweit sich Web Services in Zukunft durchsetzen werden, lässt sich zur Zeit noch nicht abschließend beurteilen. Man muss jedoch davon ausgehen, dass der Erfolg und die Verbreitung der Web Service Technologie nicht nur von der benötigten Fachlichkeit, der Wirtschaftlichkeit und den technischen Determinanten wie beispielsweise der Verwendung von Internetstandards (z.B. HTTP und XML) abhängt, sondern auch Aspekte wie die Abrechenbarkeit von Web Services bei deren Durchsetzung eine wesentliche Rolle einnehmen.

Dieser Beitrag untersucht deshalb Abrechnungsmöglichkeiten für Web Services, um eine Entlohnung von Dienst Anbietern zu ermöglichen. Hierzu werden zunächst einige Grundlagen der Web Service Technologie dargestellt und die beiden prinzipiellen Möglichkeiten der Abrechnung aufgezeigt. Aus diesen beiden Möglichkeiten wird dann die Notwendigkeit eines zentralen Abrechnungsdienstes hergeleitet und diese Thematik vertieft. Um eine zentrale Abrechnung von Web Services zu ermöglichen, müssen die abzurechnenden Web Services bestimmte Qualitätsmerkmale aufweisen bzw. Anforderungen erfüllen. Es ist also zunächst notwendig, diese Anforderungen herzuleiten, um darauffolgend die technische Umsetzbarkeit zu analysieren. Darauf aufbauend wird eine Architektur für einen Abrechnungsdienst vorgestellt, mit der beliebig viele Web Services abgerechnet werden können. Es werden die notwendigen Informationsflüsse und Funktionalitäten analysiert, die für das Abrechnungsmanagement benötigt werden.

2. Grundlagen

In diesem Abschnitt werden kurz die Web Service Architektur und die Basistechnologien dargelegt. Daran anschließend werden die grundsätzlichen Möglichkeiten der Abrechnung von Web Services aufgegriffen, um daraus die Notwendigkeit eines zentralen Abrechnungsdienstes zu begründen.

Die Web Service Architektur besteht aus den drei Rollen des Dienst Anbieters (Service Provider), des Dienstnutzers (Service Requestor) und des Dienstmaklers (Service Broker), die verschiedene Aufgaben innerhalb dieser Architektur abbilden. Darüber hinaus sind auf der Ebene der Dienstkommunikation SOAP, auf der Ebene der Dienstbeschreibung WSDL und auf der Ebene der Dienstauffindung UDDI als Technologien zentraler Bestandteil dieser Architektur. Auf die genauen Aufgaben der Rollen und die dazu notwendigen Technologien soll aufgrund der zahlreichen Literatur [Be01, BMW02, Bo00, Ce02, Ch01, Kn02, RB02, Oe01] im Rahmen dieses Beitrags nicht weiter eingegangen werden. Auch die Rolle des Dienstmaklers wird im Zusammenhang mit der Abrechnung nicht weiter dargelegt, da der Dienstanbieter bei der Nutzung bereits über den Web Service informiert ist.

Nachfolgend soll kurz die Eignung bisheriger Verfahren der Abrechnung im Internet diskutiert werden, wobei Tabelle 1 exemplarisch einige Abrechnungssysteme nach dem Abrechnungsprinzip kategorisiert.

Prinzip	Abrechnungssystem
Bargeldähnliche Systeme	Cybercoin
	eCash
Scratchcards	Paysafecard
	Micromoney
Billing-Verfahren	NET900 Classic
	X-PressPay
Inkassosysteme	FIRSTGATE click&buy
	NET900 Kontopass
Zahlung per Mobiltelefon	Paybox
	StreetCash
	Payitmobile

Tabelle 1: Überblick über Zahlungsverfahren im Internet

Betrachtet man nun diese Verfahren genauer, so wird bei den Verfahren meistens der Endnutzer als Wirtschaftssubjekt fokussiert, bei der ein manueller Eingriff beim Bezahlvorgang stattfinden muss. Bei der Nutzung von Web Services wird jedoch der Bereich der Integration in eine bestehende Systemumgebung als wesentliches Einsatzgebiet ausgemacht. In diesem Bereich ist jedoch bei jeder Nutzungsanfrage keine Nutzerinteraktion möglich, so dass eine automatisierte Abrechnung über standardisierte Schnittstellen notwendig ist. Aufgrund dieser Tatsache sind existierende Verfahren zur Abrechnung von Web Service Nutzungen in der zur Zeit angebotenen Form nicht geeignet.

Betrachtet man nun die prinzipiellen Möglichkeiten der Abrechnung von Web Services genauer, so sind zwei verschiedene Varianten der Abrechnung denkbar [CI02]:

- **Erstellung eines eigenen Abrechnungsmechanismus**
In diesem Fall muss jeder Dienstanbieter seinen eigenen Abrechnungsmechanismus entwickeln, der die Zahlungen und die Nutzerverwaltung abwickelt.
- **Nutzung eines zentralen Abrechnungsdienstes**
In diesem Fall können die Dienstanbieter einen zentralen Abrechnungsdienst nutzen, der unterschiedliche Tarifmodelle zur Verfügung stellt und die Aufgaben der Zahlungsabwicklung und Nutzerverwaltung übernimmt. Denkbar wären hier beispielweise Tarifmodelle für die Abrechnung pro Nutzung (Charge per call), die Abrechnung für einen Nutzungszeitraum (Charge per period) und die einmalige Abrechnung der Nutzung (One-off charge). Darüber hinaus können auch verschiedene Tarifmodelle, die durch Anwendung von Preisdifferenzierungsinstrumenten entstehen, zum Einsatz kommen [BGS03].

Betrachtet man nun die beiden aufgezeigten Methoden genauer, so sollen nun die Vor- und Nachteile der beiden Methoden skizziert werden. Hierbei sind die Vorteile der einen Methode jeweils Nachteile der anderen und umgekehrt, so dass nur der zentrale Abrechnungsdienst betrachtet wird. Als Vorteil bei der Nutzung eines zentralen Abrechnungsdienstes ist die Konzentration des Dienstanbieters auf seine tatsächlichen

Kernkompetenzen und somit auf die tatsächliche Funktionalität des angebotenen Web Services zu nennen. Des Weiteren kann durch die Nutzung eines zentralen Abrechnungsdienstes eine Standardisierung des Abrechnungsprozesses erzielt und dadurch eine Senkung der Transaktionskosten erreicht werden. Nachteilig wirkt sich eventuell aus, dass der zentrale Abrechnungsdienst den gewünschten Abrechnungsmechanismus nicht unterstützt. Dieser Nachteil kann jedoch durch eine enge Zusammenarbeit und eine gemeinsame Entwicklung des gewünschten Abrechnungsmechanismus abgemildert werden. Des Weiteren entstehen für die Durchführung der Abrechnung durch den Abrechnungsdienst Kosten, die der Dienstanbieter zu tragen hat und die somit die Einnahmen bzw. die Erlöse des Dienstanbieters schmälern. Allerdings muss diesen Kosten die Entwicklungskosten eines eigenen Abrechnungssystems gegenübergestellt werden.

Da jedoch die Konzentration des Dienstanbieters auf seine Kernkompetenzen und die Möglichkeit der Standardisierung als ausschlaggebend eingestuft werden, wird im Folgenden die zentrale Abrechnung von Web Service Nutzungen vertieft.

3. Architektur eines Abrechnungsdienstes

In diesem Abschnitt soll ein Konzept vorgestellt werden, um die zentrale Abrechnung von Web Services verschiedenster Dienstanbieter realisieren zu können. Hierfür wird zunächst dargestellt, welche Parteien an der Abrechnung beteiligt sind und deren fachlichen Anforderungen formuliert. In einem weiteren Schritt wird die Konzeption der technischen Umsetzung dargestellt. Dabei werden die wesentlichen Teilfunktionalitäten beim Abrechnungsdienst und beim Dienstanbieter aufgezeigt und das Zusammenwirken der einzelnen Teilfunktionalitäten erläutert.

3.1. Fachliche Anforderungen

Zunächst sollen die beteiligten Parteien betrachtet werden, die bei der Abrechnung von Web Services notwendig sind:

- Ein **Abrechnungsdienst** ermöglicht das Abrechnen von beliebigen Web Services unterschiedlicher Dienstanbieter zwischen Dienstnutzern und Dienstanbietern, soweit diese als kommerzielle Web Services registriert sind. Der Abrechnungsdienst ist dabei selbst in Form von Web Services implementiert.
- Die **Dienstanbieter** implementieren die Services und bieten sie kostenpflichtig an, indem die Dienstanbieter nach Nutzung des Web Services durch den Dienstnutzer einen Abrechnungsmechanismus des Abrechnungsdienstes in Anspruch nehmen. Daher müssen die Dienstanbieter ein Abrechnungskonto beim Abrechnungsdienst unterhalten.

- Die **Dienstanbieter** nehmen kommerzielle Web Services von den Dienstnutzern in Anspruch. Damit eine Abrechnung möglich ist, muss auch der Dienstnutzer ein Abrechnungskonto beim Abrechnungsdienst vorhalten.

Abbildung 1 verdeutlicht die Beziehung dieser drei beteiligten Parteien graphisch.

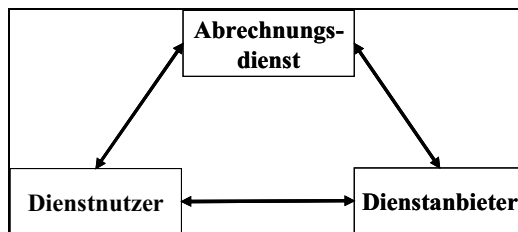


Abbildung 1: Rollen im Abrechnungskonzept

Der Abrechnungsdienst und der Dienstanbieter haben unterschiedliche Funktionalitäten im Abwicklungsprozess einer Web Service Dienstleistung vorzuhalten.

Der **Abrechnungsdienst** muss grundsätzlich folgende Funktionalitäten bieten, um eine Abrechnung von kommerziellen Web Services zu ermöglichen:

1. Dienstnutzer und Dienstanbieter müssen dem Abrechnungsdienst bekannt sein, damit der jeweilige Dienstnutzer eines Web Services belastet und der Dienstanbieter entsprechend entlohnt werden kann. Alle Nutzer des Abrechnungsdienstes müssen insofern eindeutig identifizierbar sein. Hierfür ist es notwendig, dass sich alle Nutzer, also Dienstnutzer ebenso wie Dienstanbieter, zunächst beim Abrechnungsdienst registrieren lassen. Der Abrechnungsdienst muss infolgedessen eine **Nutzerverwaltung** bieten, damit der Abrechnungsvorgang den beteiligten Nutzern zugeordnet werden kann.
2. Die Dienstanbieter der kommerziellen Web Services müssen über ihre Registrierung als Nutzer hinaus ihre angebotenen Web Services registrieren, um so eine genaue Aufstellung über die Nutzungsintensität und die Nutzungsart analog zu einer Einzelverbindungsübersicht bei der Telefonrechnung zu ermöglichen. Dies kommt dem Informationsbedürfnis der Dienstnutzer entgegen. Daher muss der Abrechnungsdienst also neben der Nutzerverwaltung auch eine **Serviceverwaltung** beinhalten.
3. Um eine möglichst komfortable Abrechnung zu ermöglichen, ist es notwendig, unterschiedliche Tarifmodelle auf Basis der im Grundlagenteil dargestellten möglichen Abrechnungsmechanismen anzubieten. Diese Tarifmodelle müssen von Dienstanbieterseite bei Bedarf für einen bestimmten registrierten Web Service aktiviert und mit den entsprechenden Abrechnungsmodalitäten für diesen Web Service initialisiert werden. Auf Seiten des Dienstnutzers ist es nötig, die für einen Web Service aktiven Tarifmodelle und Konditionen erfragen zu können und aus dieser Tarifmodellmenge einen auszuwählen. Daher muss der Abrechnungsdienst auch über eine **Tarifverwaltung** verfügen. Wird vor der Nutzung eines Web

Services durch den Dienstanbieter keine Tariffestlegung durchgeführt, so wird ein Standardtarif für die Abrechnung verwendet, der vom Dienstanbieter bei der Aktivierung der Tarifmodelle festgelegt wird. Auf diese Art und Weise können verschiedenartige Tarifmodelle hinterlegt werden. Dies erlaubt wiederum eine sehr flexible Abrechnung der Web Service Nutzung.

4. Um die Abrechnungen von kommerziellen Web Services zu ermöglichen, muss der Abrechnungsdienst die Abrechnungskonten der Dienstanbieter und –nutzer verwalten und somit eine **Kontenverwaltung** besitzen. Der Abrechnungsdienst muss dazu entsprechende Funktionalitäten bieten, um die Abrechnungskonten flexibel aufladen, Kontostände übermitteln oder Konten belasten zu können. Des Weiteren muss die Möglichkeit zur Erstellung eines Kontoauszugs gegeben werden. Hierbei soll im Folgenden von einer Prepaid-Variante ausgegangen werden.

Der Abrechnungsdienst ist mit seinen Funktionen als Clearingstelle und Trust Center zentraler und unabdingbarer Mitwirkender bei jeder kommerziellen Web Service Anfrage.

Auf Seiten des **Dienstanbieters** eines kostenpflichtigen Web Services ist lediglich, wie bereits erwähnt, die **Dienstfunktionalität** selbst zu implementieren und damit vorzuhalten. Um Anpassungsaufwände so gering wie möglich zu halten, sollte die Entwicklung der Web Services unabhängig davon sein, dass diese später mit Hilfe des Abrechnungsdienstes kommerziell verwertet werden, d.h. für die Entwicklung sollten grundsätzlich keine oder nur sehr wenige Spezifika der Abrechnung beachtet werden müssen.

Der **Dienstanbieter** sollte nach Möglichkeit nur wenige Anforderungen erfüllen, um kostenpflichtige Web Services nutzen zu können. Dadurch können Einstiegsbarrieren vermieden und Nutzungshemmnisse abgebaut werden. Der Dienstanbieter sollte daher in diesem Modell lediglich seine eindeutige **Nutzeridentifikation** verwalten müssen, um einen kostenpflichtigen Web Service zu nutzen.

Tabelle 2 fasst die fachlichen Anforderungen an die Parteien nochmals zusammen.

Parteien	Aufgaben/Funktionalitäten
Abrechnungsdienst	• Nutzerverwaltung • Serviceverwaltung • Tarifverwaltung • Kontenverwaltung
Dienstanbieter	• Bereitstellung der Dienstfunktionalität
Dienstanutzer	• Verwaltung der eigenen Nutzeridentifikation

Tabelle 2: Fachlichen Anforderungen an die Parteien

3.2. Technisches Konzept

Nachdem im letzten Abschnitt die Anforderungen an die beteiligten Parteien formuliert worden sind, müssen diese nun in ein technisches Konzept umgesetzt werden. Dabei sind die fachlichen Anforderungen dafür in entsprechende Teilfunktionalitäten zu zerlegen. Jede Teilfunktionalität sollte dabei die Granularität besitzen, die es erlaubt, die Teilfunktionalität in einer eigenen Methode unterzubringen. Des Weiteren sollen alle notwendigen Methoden ebenfalls als separate Web Services implementiert werden, damit homogene Zugriffsmechanismen Anwendung finden. Da nur der Abrechnungsdienst und der Dienstanbieter Funktionalitäten vorhalten müssen, wird im Folgenden diese Zweiteilung vorgenommen.

Betrachten man nun die fachlichen Anforderungen an einen zentralen Abrechnungsdienst, so müssen die Bereiche der Nutzer-, der Service-, der Tarif- und Kontenverwaltung abgedeckt werden.

Im Rahmen der Nutzerverwaltung muss der Abrechnungsdienst in diesem einfachen Modell lediglich die Anmeldung eines Dienstnutzers oder Dienstanbieters abdecken. Um diese Zielsetzung zu erreichen, wird eine Methode *UserSubscription* benötigt, der userspezifische Daten (Name, Vorname, Adresse, etc.) übergeben werden. Als Rückgabewert wird die eindeutige *UserID* zurückgeliefert, die der Abrechnungsdienst vergibt.

Im Rahmen der Serviceverwaltung muss die Anmeldung eines kommerziellen Web Services beim Abrechnungsserver ermöglicht werden. Dazu wird eine Methode *ServiceSubscription* benötigt, die als Parameter die *UserID* des Dienstanbieters übergeben bekommt und die eindeutige vom Abrechnungsdienst vergebenen *ServiceID* zurückliefert.

Im Anforderungsbereich der Tarifverwaltung muss der Abrechnungsdienst unterschiedliche Abrechnungstarife (*TariffID*) vorhalten. Dafür müssen für die Dienstanbieterseite Methoden zur Tarifaktivierung mit Konditionsfestlegung, zur Tarifdeaktivierung und zur Festlegung des Standardtarifs vorgehalten werden. Dazu wird eine Methode *EnableTariff* benötigt, die als Parameter die *TariffID*, die *ServiceID* sowie einen Vektor mit den benötigten Konditionen übergeben bekommt. Analog hierzu wird eine Methode *DisableTariff* für die Tarifdeaktivierung und eine Methode *SetStandardTariff* vorgehalten, die beide als Parameter die *TariffID* und die *ServiceID* erwarten. Alle diese Methoden liefern wahr für eine erfolgreiche und falsch für eine fehlerhafte Ausführung zurück. Um die Auswahl eines aktiven Tarifmodells auf Dienstnutzerseite zu unterstützen, müssen Funktionalitäten zur Verfügung stehen, die das Abfragen aktiver Tarifmodelle für einen bestimmten Dienst, das Festlegen auf ein Tarifmodell für die Nutzung und das Abfragen des aktuell aktiven Tarifmodells erlauben. Für das Abfragen der Tarifmodelle ist somit eine Methode *GetTariffInfo* vorzusehen, die als Parameter die *ServiceID* übergeben bekommt und eine Liste aller aktiven Tarifmodelle mit *TariffID* und Konditionen zurückgibt. Für das Festlegen ist eine Methode *SetTariff* notwendig, die die *UserID* des Dienstnutzers, die *ServiceID* und die *TariffID* als Parameter übergeben bekommt und als Ergebnis wahr für eine

erfolgreiche Tariffestsetzung und falsch für eine fehlerhafte Durchführung der Aktion zurückliefert. Für die Abfrage des aktiven Tarifmodells für den Dienstinutzer steht die Methode *GetTariff* zur Verfügung, die als Parameter die UserID und die ServiceID übergeben bekommt und als Rückgabewert die TariffID liefert.

Im fachlichen Anforderungsbereich der Kontenverwaltung müssen Methoden zum Auffüllen, Belasten und Abfragen von Konten bereitgestellt werden. Die Einzahlung eines bestimmten Betrags auf ein Konto wird dabei durch die Methode *RefillAccount* erfüllt, die als Parameter die UserID und den Betrag übergeben bekommt. Die Methode *DebitAccount* übernimmt dementsprechend die Belastung eines Kontos, in dem ihr als Parameter die UserID des Dienstinutzer und die ServiceID des genutzten Web Services übergeben wird. Als Ergebnis liefern beide Methoden wahr für eine erfolgreiche und falsch für eine fehlgeschlagene Durchführung zurück. Letztendlich kann der Kontostand mit Hilfe der Methode *GetAccountBalance* abgefragt werden, indem der Methode als Parameter die UserID übergeben wird und als Ergebnis der Kontostand zurückgegeben wird. Um die Informationsbedürfnisse der Nutzer des Abrechnungsdienstes zufrieden zustellen, ermöglicht die Methode *GetAccountState* die Abfrage aller Kontenbewegungen. Dafür wird die UserID als Parameter übergeben und als Ergebnis eine Liste aller Kontenbewegungen erzeugt.

Die Tabellen 3 und 4 fassen die eben dargestellten Methoden des Abrechnungsdienstes zusammen. Zur Verdeutlichung wurden die fachlichen Anforderungen ebenfalls den Methoden zugeordnet.

Methode	Parameter	Rückgabe	Anforderung
UserSubscription	UserData	UserID	Nutzerverwaltung
ServiceSubscription	UserID	ServiceID	Serviceverwaltung
RefillAccount	UserID, Betrag	True/false	Kontoverwaltung
DebitAccount	UserID, ServiceID	True/false	Kontoverwaltung
GetAccountBalance	UserID	Kontostand	Kontoverwaltung
GetAccountState	UserID	Liste Konten- bewegungen	Kontoverwaltung

Tabelle 3: Methoden des Abrechnungsdienstes (1)

Methode	Parameter	Rückgabe	Anforderung
UserSubscription	UserData	UserID	Nutzerverwaltung
ServiceSubscription	UserID	ServiceID	Serviceverwaltung
EnableTariff	TariffID, ServiceID, Conditionsvector	True/False	Tarifverwaltung
DisableTariff	TariffID, ServiceID	True/False	Tarifverwaltung
SetStandardTariff	TariffID, ServiceID	True/False	Tarifverwaltung
GetTariffInfo	ServiceID	Liste Tarifmodelle	Tarifverwaltung
SetTariff	UserID, ServiceID, TariffID	True/False	Tarifverwaltung
GetTariff	UserID, ServiceID	TariffID	Tarifverwaltung

Tabelle 4: Methoden des Abrechnungsdienstes (2)

Betrachten man nun den zweiten Block der Anforderungen für den Abrechnungsdienst, so muss dieser, wie bereits im fachlichen Abschnitt dargestellt, nur die eigentliche Funktionalität des Web Services (*OfferedService*) vorhalten.

Nachdem nun aus den fachlichen Anforderungen die notwendigen Teilfunktionalitäten auf der Seite des Dienstanbieters und des Abrechnungsdienstes hergeleitet und dargestellt wurden, soll nun das Zusammenwirken der einzelnen Methoden bei der Nutzung eines kommerziellen Web Services dargestellt werden. Hierbei ist eine Zerlegung in die fünf Phasen der Anbieterregistrierung, der Tariffestlegung, der Nutzerregistrierung, der Tarifauswahl und der Dienstnutzung zur Verdeutlichung sinnvoll.

In der Anbieterregistrierungsphase muss der Dienstanbieter sich selbst und den kommerziellen Web Service beim Abrechnungsdienst anmelden. Dazu nutzt er die Methoden *UserSubscription* und *ServiceSubscription*. Dabei ist diese Reihenfolge obligatorisch, da eine Serviceregistrierung nur mit einer *UserID* möglich ist.

In der Phase der Tariffestlegung aktiviert der Dienstanbieter die Tarifmodelle durch die Methode *EnableTariff* und legt mit Hilfe der Methode *SetStandardTariff* ein Standardtarifmodell fest.

In der Phase der Nutzerregistrierung meldet sich der Dienstnutzer analog zur Anbieterregistrierungsphase mit der Methode *UserSubscription* beim Abrechnungsdienst an.

In der Phase der Tarifauswahl, die nur optional ausgeführt werden muss, legt der Dienstnutzer das Tarifmodell für die Abrechnung fest. Hierzu nutzt er die Methode *GetTariffInfo* zur Informationsbeschaffung über mögliche Tarifmodelle und danach die Methode *SetTariff* zur Festlegung des Abrechnungstarifs.

Hierbei sind die beiden Registrierungsphasen nur einmalig vor der ersten Nutzung des kommerziellen Web Services durchzuführen, die Phase der Tariffestlegung kann beliebig oft durchgeführt werden. Die Phase der Tarifauswahl findet lediglich optional statt und kann ebenfalls beliebig oft durchgeführt werden.

In der Phase der eigentlichen Dienstnutzung verwendet der Dienstnutzer den Dienst des Dienstanbieters durch Ausführung der Methode *OfferedService*, wobei der Name der Methode als Platzhalter zu verstehen ist. Vor der Rückgabe des Ergebnisses nutzt der Dienstanbieter jedoch die Methode *DebitAccount* des Abrechnungsdienstes zur Abrechnung der Dienstnutzung. Danach übermittelt er das Ergebnis an den Dienstnutzer.

Abbildung 2 stellt die fünf Phasen der Nutzung eines kommerziellen Web Services dar und verdeutlicht die obigen Ausführungen.

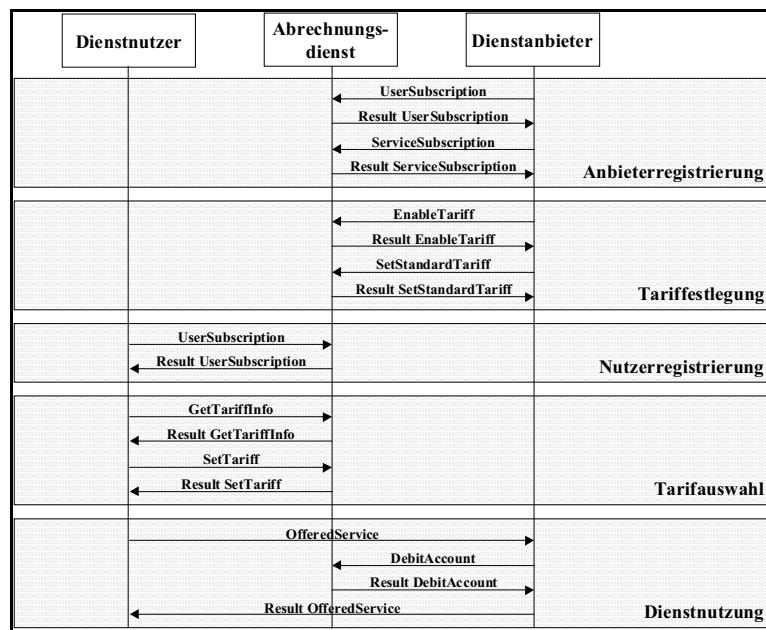


Abbildung 2: Reihenfolge der Methodenaufrufe bei einer kommerziellen Web Service Nutzung

4. Zusammenfassung

Der vorliegende Beitrag stellt zwei Möglichkeiten der Abrechnung von Web Services und denkbare Abrechnungsmechanismen vor. Darauf aufbauend werden fachliche Anforderungen an einen zentralen Abrechnungsdienst für kommerzielle Web Services hergeleitet und eine mögliche Architektur für einen Abrechnungsdienst vorgestellt.

Durch die Möglichkeit der Abrechenbarkeit von Web Services ist daher weiterer Forschungsbedarf im Bereich der Preisdifferenzierung zu erkennen, wodurch eine bessere Ausschöpfung der Zahlungsbereitschaft der Konsumenten erreicht werden kann. In diesem Zusammenhang sind dort speziell die Preisdifferenzierungsinstrumente für digitale Informationsgüter wie Versioning, Windowing und Bundling zu analysieren [Br00, BGS03].

Darüber hinaus gilt es im Rahmen weiterer Betrachtungen das Problemfeld der Sicherheit im Bezug auf Web Services zu untersuchen und zu analysieren, da auch dieser Punkt als wesentlich für die Verbreitung von Web Services angesehen werden kann. Erste Ansätze in diesem Bereich sind u.a. durch XML Digital Signatures, Security Assertion Markup Language und XML Encryption bereits erkennbar und werden zur Zeit in der aktuellen Literatur diskutiert [Ba02, IDS02, Ra02].

Auf der anderen Seite realisiert das vorgestellte Abrechnungskonzept lediglich die Abrechnung einzelner Anfragen. Zur Abwicklung komplexer Transaktionen mit mehreren beteiligten Web Services unterschiedlicher Dienstanbieter muss die dargestellte Architektur erneut diskutiert und eventuell erweitert werden, da eine gewisse Koordinationsleistung zwischen den unterschiedlichen Web Services zu realisieren ist. Des Weiteren ist zu diskutieren, inwieweit das ACID-Prinzip für Transaktionen auf Web Services übertragbar ist und welche Auswirkungen dies auf die dargestellte Architektur hat. Auch in diesem Bereich befindet sich ein erster Ansatz mit dem Business Transaction Protocol (BTP) in Entwicklung [KE02, Oa02].

Daraus folgend muss natürlich auch die Verknüpfung des vorgestellten Abrechnungsschemas mit den Fragestellungen der Sicherheit und der Transaktionalität sowie die rechtliche Sicht auf die Thematik der Abrechnung (Datenschutzproblematik) als weiterer Forschungsbedarf angesehen werden.

5. Literaturverzeichnis

- [Ba02] Bartel, M., Boyer, J., Fox, B., LaMacchia, B., Simon, E.: XML-Signature Syntax and Processing, W3C Recommendation, 2002. Verfügbar unter: <http://www.w3.org/TR/xmldsig-core/>, Abruf am 2002-12-16.
- [Be01] Bettag, U.: Web-Services. In: Informatik Spektrum 5 (2001) 24, S.302-304.
- [BGS03] Burghardt, M., Gehrke, N., Schumann, M.: Implikationen kommerzieller Web Services, In: Tagesband XML-Technologien für Middleware, Middleware für XML-Anwendungen, XMIDX 2003, Berlin, 2003.

- [BMW02] Beimborn, D., Mintert, S., Weitzel, T.: Web Services und ebXML. In: Wirtschaftsinformatik 44 (2002) 3, S. 277-280.
- [Bo00] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N. et al.: Simple Object Access Protocol (SOAP) 1.1. W3C Note (2000). Verfügbar unter: <http://www.w3.org/TR/SOAP/>, Abruf am 2002-12-16.
- [Br00] Brandtweiner, R.: Differenzierung und elektronischer Vertrieb digitaler Informationsgüter, Symposium Publishing, Düsseldorf, 2000.
- [Ce02] Cerami, E.: Web Services Essentials, Beijing u.a., 2002.
- [Cl02] Clark, M.: Selling Web Services. In: Fletcher, P., Waterhouse, M.: Web Services Business Strategies and Architectures, Birmingham, 2002.
- [Ch01] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Services Description Language (WSDL) 1.1. W3C Note, 2001. Verfügbar unter: <http://www.w3.org/TR/wsdl>, Abruf am 2002-12-16.
- [IDS02] Imamura, T., Dillaway, B., Simon, E.: XML Encryption Syntax and Processing, W3C Recommendation, 2002. Verfügbar unter: <http://www.w3.org/TR/xmlenc-core/>, Abruf am 2002-12-16.
- [KE02] Kaleem, M. F.; Eichborn, F.: Der Koordinator – Web Services Transaktion mit dem Business Transaction Protocol. In: XML & Web Services Magazin 2 (2002), S. 84-85.
- [Kn02] Knuth, M.: Web Services – Einführung und Übersicht, Frankfurt, 2002.
- [Oa02] Organization for the Advancement of Structured Information Systems: Business Transaction Protocol Primer 1.0, 2002. Verfügbar unter: http://www.oasis-open.org/committees/business-transactions/documents/primer/BTP_Primer_v1.0.20020605.pdf, Abruf am 2002-12-16.
- [Oe01] Oellermann, W. L.: Architecting Web Services, New York, 2001.
- [Ra02] Raepke, M.: Sicherheit in elektronischen Märkten. In: Heilmann, H. (Hrsg.): Elektronische Marktplätze, HMD 223, 2002, S. 63-75.
- [RB02] Rawolle, J., Burghardt, M.: Web Services – eine Alternative für die zwischenbetriebliche Integration. In: IS Report 6 (2002) 2, S. 40-46.
- [SS00] Scribner, K., Stiver, M.: Understanding SOAP, Indianapolis, 2000.