

Ralf Steinmetz, Rainer Berbner, Ivan Martinovic

Web Services zur Unterstützung flexibler Geschäftsprozesse in der Finanzwirtschaft

1. Einführung
2. Web Services – Definition und technische Aspekte
 - 2.1 Definitiverischer Hintergrund
 - 2.2 Service-oriented Architecture (SoA)
 - 2.3 Relevante Standards
 - 2.3.1 SOAP
 - 2.3.2 WSDL
 - 2.3.2 UDDI
 - 2.4 Sicherheit und Trust als eine zentrale Herausforderung
3. Geschäftsprozesse auf Grundlage der Web Service Technologie
 - 3.1 Business Process Execution Language (BPEL)
 - 3.2 Prototypische Implementierung eines Kreditprozesses
4. Einfluss der Web Service Technologie auf die weitere Entwicklung des Finanzsektors
5. Zusammenfassung

Literatur

1 Einführung

Seit einigen Jahren befindet sich der deutsche Finanzsektor in einer strukturellen Krise. Die Gründe hierfür sind mannigfaltig und werden unter Experten kontrovers diskutiert [o.V. 2003a]. Hierbei ist unbestritten, dass sich die deutschen Finanzdienstleister neu aufstellen müssen, um im Wettbewerb mit der internationalen Konkurrenz bestehen zu können.

Wird nach Lösungsansätzen zur Überwindung der gegenwärtigen Krise im Finanzsektor gefragt, so geben Analysten häufig die Antwort, dass die Fertigungstiefe von derzeit ca. 80% deutlich verringert werden müsse, um Kosten zu sparen [König 2002, S. 517; Breuer 2003]. So sei die Fertigungstiefe der auch nach internationalen Maßstäben als kompetitiv geltenden deutschen Automobilindustrie bei mittlerweile rund 30% angelangt. Finanzdienstleister müssten sich auf Kernkompetenzen konzentrieren und im Rahmen eines *Business Process Outsourcing (BPO)*, sofern gesetzlich möglich, Geschäftsprozesse (oder Teile davon) auslagern [Katzensteiner et al. 2004].

Als weiterer Lösungsansatz wird eine flexiblere Gestaltung der internen Geschäftsprozesse diskutiert. Ziel soll es sein, den anspruchsvoll gewordenen Kunden Dienstleistungen bedarfsgerecht (*on demand* bzw. *Just-in-Time*) anbieten zu können [Lund et al. 2003]. Hierfür sind flexibel gestaltete Geschäftsprozesse, die dynamisch adaptiert werden können, von entscheidender Bedeutung.

Diese Lösungsvorschläge können jedoch nur umgesetzt werden, wenn es den Finanzdienstleistern gelingt, ihre Geschäftsprozesse optimal durch IT-Systeme zu unterstützen [o.V. 2003b, S. 60-61].

Gerade aber hier liegt die große Herausforderung: Die IT-Landschaft vieler großer Finanzdienstleister ist durch eine hohe Anzahl monolithischer, miteinander kommunizierender Legacy-Anwendungen geprägt, deren Heterogenität und Komplexität kaum noch beherrschbar scheint [o.V. 2003b, S. 60]. Zahlreiche Versuche, dies durch EAI-Projekte in den Griff zu bekommen, haben nicht zu den erwarteten Erfolgen geführt [Hackmann 2001].

Die effektive technische Realisierung des Outsourcing von Geschäftsprozessen, d.h. die Kopplung der internen IT-Systeme mit denen

des Outsourcing-Partners wird durch diese Heterogenität und Komplexität fast unmöglich gemacht.

Mit Fragestellungen im Zusammenhang der optimalen Unterstützung von Geschäftsprozessen durch IT-Systeme beschäftigt sich ein Forscherteam unter der Leitung von Prof. Steinmetz im Rahmen des E-Finance Lab. Das E-Finance Lab [<http://www.efinancelab.de>] ist ein Kooperationsprojekt der Universitäten Frankfurt und Darmstadt sowie einer Vielzahl von Industriepartnern.

Im folgenden wird mit der Web Service Technologie ein noch junger Ansatz untersucht, der mit dem Anspruch antritt, bestehende Legacy-Anwendungen effektiv zu integrieren und Prozesse so flexibel zu gestalten, dass Dienstleistungen „*on demand*“ angeboten werden können.

In Kapitel 2 werden die technischen Aspekte der Web Service Technologie vorgestellt. Mit BPEL (Business Process Execution Language) wird in Kapitel 3 eine Sprache beschrieben, die die Komposition von Web Services zu Geschäftsprozessen ermöglicht. In diesem Zusammenhang wird die prototypische Implementierung eines Bankprozesses mit BPEL beschrieben, die am Lehrstuhl von Prof. Steinmetz realisiert wird. Der Einfluss von Web Services auf innovative Geschäftsmodelle im Finanzsektor wird in Kapitel 4 thematisiert. In Kapitel 5 erfolgt eine Zusammenfassung der Ausführungen.

2 Web Services – Definition und technische Aspekte

In diesem Kapitel wird der Begriff Web Service definiert und die der Web Service Technologie zugrunde liegende Architektur vorgestellt. Zudem werden die Standards beschrieben, die den Kern der Web Service Technologie darstellen.

Das Thema Sicherheit wird als eine zentrale Herausforderung im Rahmen der Web Service Technologie adressiert.

1.1 Definitiverischer Hintergrund

Bei Web Services handelt es sich um XML-basierte Softwarekomponenten, die über ein Netzwerk (meist das Internet)

aufrufbar und ausführbar sind [Booth et al. 2004]. Hierbei wird zwischen der eigentlichen Implementierung des Web Service sowie seiner Servicebeschreibung unterschieden. Die Servicebeschreibung macht Aussagen zu den Schnittstellen des Web Service bzw. zur angebotenen Funktionalität.

Web Services basieren auf offenen Standards und sind unabhängig von Plattformen, Betriebssystemen und Programmiersprachen. Dies hat zudem auch die Unabhängigkeit von Herstellern zur Folge.

1.2 Service-oriented Architecture (SoA)

Der Web Service Technologie liegt das Paradigma der *Service-oriented Architecture* (Abkürzung: SoA, Abbildung 1) zugrunde [Natis 2003].

Dieses Konzept basiert auf Services als Grundbausteinen. Ein Service stellt hierbei eine abgeschlossene, unabhängige Komponente dar, die eine klar definierte Funktionalität anbietet [Yang, Papazoglou, van den Heuvel 2002]. Ein Service abstrahiert von zugrunde liegenden Entitäten, Objekten und Klassen. Die Unabhängigkeit eines Services ermöglicht seine Wiederverwendbarkeit in verschiedenen Zusammenhängen.

Eng mit dem SoA-Ansatz verbunden ist das Prinzip der *Lösen Kopplung*. Dies bedeutet, dass die verwendeten Services so gekoppelt sein sollten, dass sie ohne großen Aufwand durch andere Services ersetzt werden können [Web Services Architecture Working Group 2004].

Das SoA-Konzept ermöglicht somit ein Höchstmaß an Flexibilität, welches für Finanzdienstleister wichtig ist, um zeitnah auf dynamische Märkte und Kundenwünsche reagieren zu können.

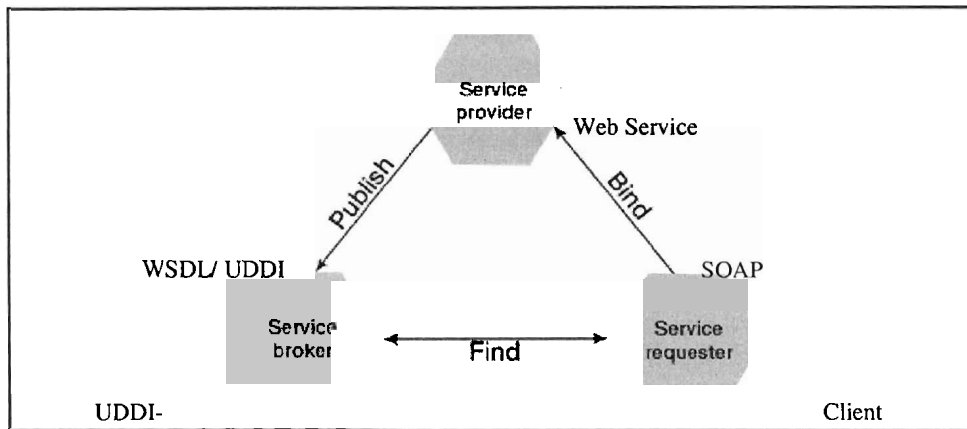


Abbildung 1: Service-oriented Architecture (Kreger 2001, S. 7).

Der SoA liegen drei Rollen zugrunde, die sich wie folgt beschreiben lassen [Kreger 2001, S. 7-8]:

- Der Service Requestor ruft einen Web Service auf, um dessen Funktionalität zu nutzen. Die Rolle des Requestor kann von einer Person oder einem anderen Web Service wahrgenommen werden.
- Zentrale Aufgabe des Service Broker ist das Bereitstellen, Veröffentlichen und Vertreiben von Web Services verschiedener Anbieter. Hierzu stellt der Broker Verzeichnisse (sog. *Service-Registries*) bereit, die die Beschreibungen der veröffentlichten Web Services verwalten.
- Der Service Provider ist der Eigentümer des Web Service. Er stellt den Web Service zur Verfügung. Der Service Provider kann eine Beschreibung des Service an den Service Broker übermitteln, damit der Service für eine Vielzahl von Nutzern verfügbar ist.

1.3 Relevante Standards

Die Standards SOAP, WSDL und UDDI bilden den eigentlichen Kern der Web Service Technologie (Abbildung 2).

Service Flow	BPEL	(Kap. 3.1)
Service Publication	UDDI	(Kap. 2.3.3)
Service Discovery	UDDI	(Kap. 2.3.3)
Service Description	WSDL	(Kap. 2.3.2)
Messaging	SOAP	(Kap. 2.3.1)

Abb. 1.3.1 Web Service Standards

1.3.1 SOAP (Simple Object Access Protocol)

SOAP ist ein XML-basiertes Kommunikationsprotokoll, das den Austausch von strukturierten Daten in verteilten, heterogenen Umgebungen erlaubt [Gudgin et al. 2003]. Es kann in Kombination mit Transportprotokollen, wie z.B. HTTP oder FTP, verwendet werden.

Eine SOAP-Nachricht besteht aus einem SOAP-Header und einem SOAP-Body, die im SOAP-Envelope zusammengefasst werden. Der SOAP-Envelope stellt ein Framework zur Spezifizierung des Inhaltes von Nachrichten dar. Zudem wird beschrieben, wie dieser Inhalt zu verarbeiten ist. Der SOAP-Body enthält den eigentlichen Inhalt (Aufruf der Operation und Aufrufparameter). Der SOAP-Header erlaubt die Anreicherung der Nachricht mit Metadaten, die z.B. zur Realisierung von Billing-/ Accounting-Lösungen verwendet werden können [Kreger, S. 23].

1.3.2 WSDL (Web Service Description Language)

WSDL stellt ein XML-Format zur syntaktischen Beschreibung von Web Services dar [Christensen et al. 2001]. Hierbei wird spezifiziert, wie der Web Service aufgerufen werden kann und welche Methoden er anbietet. Zudem werden die Protokolle beschrieben, an die der Web Service gebunden wird.

Die WSDL-Beschreibung eines Web Services ist in zwei Teile gegliedert [Kreger, S. 15-17]:

- Die *Service Interface Definition* beschreibt die Schnittstellen des Web Service. Von besonderer Bedeutung in diesem Zusammenhang sind die sog. *PortTypes*. Diese stellen eine Menge von abstrakten Operationen dar. Jede dieser Operationen wird mit jeweils einer Input- und Output-Nachricht versehen, die in mehrere Segmente aufgegliedert werden können.
- Die *Service Implementation Definition* beschreibt die konkrete Realisierung des Web Service. Hierzu wird das in der *Service Interface Definition* beschriebene Binding an einen bestimmten Port gebunden.

1.3.3 UDDI (Universal Description, Discovery and Integration)

Bei UDDI handelt es sich um einen XML-basierten zentralen Verzeichnisdienst, der es Unternehmen erlaubt, nach Web Services externer Provider zu suchen und eigene Web Services zu veröffentlichen. UDDI wurde von Ariba, Microsoft und IBM entwickelt [Kreger, S. 18-21].

Die Suche registrierter Web Services erfolgt über Metadaten, die die Funktionalität und Kategorie des Web Services näher beschreiben. Das UDDI-Metamodell setzt sich aus drei Teilen zusammen. Allgemeine Informationen über den Service Provider, der den Web Service bereitstellt, werden in *White Pages* abgelegt. Technische Informationen, wie z.B. Schnittstellenbeschreibungen oder Informationen über die Bindung an bestimmte Netzwerkprotokolle werden in *Green Pages* zusammengestellt. Die *Yellow Pages* kategorisieren die Service Provider anhand bestimmter Standardkriterien [Bellwood et al. 2002].

Grundsätzlich muss zwischen öffentlichen und privaten UDDI-Verzeichnisdiensten unterschieden werden. So ist der Zugriff auf private UDDI-Verzeichnisse auf bestimmte Kunden und Geschäftspartner begrenzt, während öffentliche UDDI-Verzeichnisdienste frei zugänglich sind [Kreger, S. 20].

1.4 Sicherheit und Trust als eine zentrale Herausforderung

Gerade in der Finanzwirtschaft wird Sicherheitsfragen von jeher ein hoher Stellenwert eingeräumt.

Die Kommunikation im Web Service Umfeld zwischen Service Requestor und Service Provider findet oftmals in einer unsicheren, offenen Umgebung (z.B. Internet) statt. Daher sind ausgereifte Sicherheitskonzepte notwendig, wie sie z.B. im Bereich Multimedia und Kommunikationsnetze bereits etabliert sind [Steinmetz, Nahrstedt 2004]:

- Authentifizierung der kommunizierenden Partner
- Vertraulichkeit der zu übertragenden Daten und Nachrichten
 - Integrität der Daten (Nichtveränderbarkeit)
- Nachweisbarkeit (non-repudation)
 - Autorisierung für den Zugriff auf Ressourcen

Spezifikationen für Sicherheitsstandards für Web Services sind seit geraumer Zeit zwar zahlreich vorhanden, es fehlt aber an deren Umsetzung durch die Hersteller.

Von besonderer Bedeutung im Kontext der Web Service Sicherheit ist die Spezifikation *Web Service Security* [Atkinson et al. 2002]. Diese wurde von IBM, Microsoft und Verisign als ein Framework zur Sicherung von SOAP-Nachrichten vorgestellt. Ziel ist hierbei der Austausch von verschlüsselten und signierten Nachrichten im SOAP-Format.

Web Service Security beschreibt grundlegende Mechanismen, um unterschiedliche Sicherheitsmodelle zu unterstützen und dient somit als Grundlage für weitere Sicherheits-Spezifikationen im Web Service Umfeld.

Web Service Security wurde motiviert durch die Notwendigkeit, folgende Anforderungen zu unterstützen [Atkinson et al. 2002]:

- Mehrfache Sicherheitsmodelle zur Authentifizierung und Autorisierung
- Verschiedene Vertrauensdomänen von Web Services
- Verschiedene Verschlüsselungstechnologien
- Ende-zu-Ende-Sicherheit für SOAP-Nachrichten

Die bisher vorhandenen IT-Sicherheitsmechanismen, die hauptsächlich auf kryptographischen Mechanismen basieren, bieten eine breite Palette an Sicherheitskonzepten, die die Sicherheit der Nachrichten und deren

Austausch unterstützen (z.B.: Public Key Infrastruktur, Kerberos-Technologie, Virtuelle Private Netze). Auf diesen Konzepten beruhen auch die Mechanismen für sicheren Nachrichtenaustausch zwischen Endpunkten im Internet, wie z.B. Secure HTTP (HTTPS), Secure Sockets Layer (SSL) und X.509 Digitale Zertifikate.

Fokus eines umfassenden Web Service Sicherheitskonzeptes ist es daher, diese verschiedenen Sicherheitsanforderungen und Sicherheitsmechanismen miteinander zu nutzen. Ziel ist hierbei die Interoperabilität dieser verschiedenen Sicherheitsmechanismen zu realisieren.

Diese Herausforderung ist nicht trivial, da verschiedene Web Service Anbieter auch unterschiedliche Sicherheitsmechanismen verwenden können. Die Heterogenität verschiedener Mechanismen ist vor allem in den Bereichen Identity-Management und Zugriffsmechanismen (d.h. Feststellung der Identität des Nutzers und seine Zugangsberechtigung zu bestimmten Ressourcen) anzutreffen.

Im Gegensatz zum klassischen Client/Server-Konzept liegen der Web Service Technologie die Leitmotive Verteilung und Dezentralisierung zugrunde. Dadurch entstehen neue Fragen und Anforderungen an Sicherheitskonzepte, wie z.B. das Entstehen von Vertrauen zwischen Service Requestor und Provider. Um gerade solche Fragen beantworten zu können, hat das Thema Trust eine besondere Bedeutung gewonnen. So bietet die Spezifikation *WS-Trust* Mechanismen zur Unterstützung der Vertrauensbildung im Web Service Umfeld.

Der kryptographische Vertrauensbildungsprozess (gegeben durch verschiedene Autorisierungs-Tokens oder Digitale Zertifikate) ist alleine aber nicht ausreichend. Zukünftig werden vielmehr auch Faktoren wie Dienstgüte, Zuverlässigkeit und Reputation im Kontext von Web Services eine wichtige Rolle spielen, um Trust zu gewährleisten.

3 Geschäftsprozesse auf Grundlage der Web Service Technologie

Nach der Vorstellung von BPEL, einer Sprache zur Komposition von Web Services zu Geschäftsprozessen, wird die Vorgehensweise bei der

prototypischen Implementierung eines Geschäftsprozesses aus dem Bankenbereich beschrieben.

1.5 Business Process Execution Language Execution Language (BPEL)

Im August 2002 wurde die von IBM, BEA und Microsoft entwickelte Spezifikation *Business Process Execution Language for Web Services* (Abkürzung: BPEL bzw. BPEL4WS) vorgestellt [Andrews et al. 2003].

BPEL stellt gewissermaßen eine Vereinigung von IBM's WSFL (Web Service Flow Language) und Microsoft's XLang dar. Sowohl IBM als auch Microsoft propagieren BPEL als Nachfolger von WSFL bzw. XLang [Leymann, Roller 2002, S. 1]. Daher ist es wahrscheinlich, dass BPEL sich als Standard für die Komposition von Web Services zu Geschäftsprozessen durchsetzen wird. Zudem wurde im April 2003 eine OASIS-Arbeitsgruppe eingerichtet, die die weitere Ausarbeitung und Standardisierung von BPEL vorantreiben soll [o.V. 2003c].

BPEL ist eine XML-basierte Sprache zur Beschreibung von Geschäftsprozessen und deren Interaktion auf Grundlage der Web Service Technologie. Konzeptionell betrachtet, entstehen mit BPEL modellierte Geschäftsprozesse durch die Komposition verschiedener Web Services. Der so modellierte Geschäftsprozess kann seinerseits als Web Service von anderen Web Services aufgerufen werden. Hierbei kann es sich sowohl um unternehmensinterne als auch um Unternehmensgrenzen überschreitende, verteilte Geschäftsprozesse handeln.

Geschäftsprozesse können entweder als ausführbare oder als abstrakte Prozesse definiert werden [Andrews et al. 2003].

Ausführbare Prozesse modellieren das Verhalten der an der Geschäftstransaktion teilnehmenden Partner. Abstrakte Prozesse spezifizieren den wechselseitigen Nachrichtenaustausch zwischen den Teilnehmern der Geschäftsbeziehung sowie das Format und die Attribute der auszutauschenden Nachrichten. Die interne Implementierung wird hierbei jedoch nicht veröffentlicht. So ist der abstrakten Beschreibung eines Kreditprozesses zwar zu entnehmen, dass ein Kreditantrag angenommen bzw. abgelehnt werden kann, die hierfür ausschlaggebenden

Entscheidungskriterien bleiben jedoch verborgen. Abstrakte Prozesse sind nicht ausführbar. Die Unterscheidung zwischen abstrakten und ausführbaren Prozessen berücksichtigt, dass Unternehmen im Allgemeinen kein Interesse daran haben, dass Geschäftspartner Einblick in ihre internen Geschäftsprozesse haben.

1.6 Prototypische Implementierung eines Kreditprozesses

Am Lehrstuhl von Prof. Steinmetz wird im Rahmen des E-Finance Lab eine prototypische Implementierung eines Kreditprozesses mit BPEL erstellt (Abbildung 3). Es soll untersucht werden, inwieweit eine Modellierung von Bankprozessen auf Grundlage der Web Service Technologie möglich ist.

Der zugrunde liegende, generische Kreditprozess wird hierbei auf seine wesentlichen Prozessschritte reduziert.

Initiiert wird der Kreditprozess durch die Kreditanfrage seitens des Kunden. Hierbei ist es unerheblich, ob der Kunde die Anfrage via Internet selbst startet oder ob ein Sacharbeiter die Kundendaten in das System eingibt. Im Anschluss daran wird die Kreditwürdigkeit des Kunden geprüft. Hierzu wird das Einholen einer SCHUFA-Auskunft über den Antragsteller nachgebildet und ein Rating anhand interner Kriterien erstellt. Auf Grundlage dieser Informationen wird die Entscheidung über die Kreditvergabe getroffen. Diese wird dem Kunden dann schriftlich mitgeteilt.

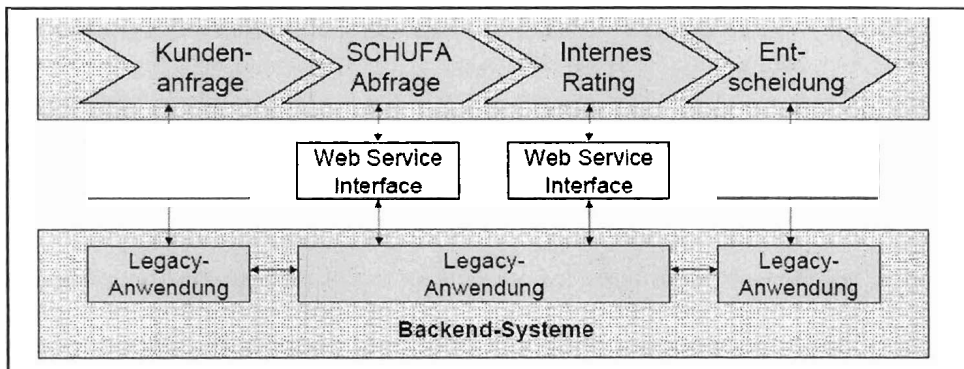


Abbildung 3: Kreditprozess mit BPEL.

Bei der Erstellung des Prototyps wird wie folgt vorgegangen: Im ersten Schritt wird der Kreditprozess mit BPEL modelliert. Jeder Prozessschritt wird durch einen Web Service Aufruf realisiert. Hierzu wird die Funktionalität der existierenden Legacy-Systeme gekapselt und als WSDL-Schnittstelle zur Verfügung gestellt. Alle am Workflow beteiligten Web Services werden ebenso wie ihr Zusammenwirken beschrieben. Ausnahmesituationen mit Kompensationsmechanismen werden modelliert, um bei einer fehlerhaften Ausführung bestimmter Prozessschritte bereits getätigte Änderungen rückgängig machen zu können.

Die praktischen Arbeiten mit BPEL haben gezeigt, dass BPEL zwar noch nicht voll ausgereift ist, aber dennoch das Potential hat, zum führenden Standard für die Orchestrierung von Web Services zu Geschäftsprozessen zu werden. Allerdings dürfen auch einige konzeptionelle Schwachpunkte von BPEL nicht unerwähnt bleiben. So ist BPEL eine hybride Sprache, der kein einheitliches Konzept zugrunde liegt, da die unterschiedlichen Ansätze von Microsoft's XLang und IBM's WSFL bei BPEL Eingang gefunden haben. Dies hat u.a. zur Folge, dass die Steuerung des Kontrollflusses (z.B. die parallele Ausführung von Prozessschritten) durch unterschiedliche Elemente realisiert werden kann [Wohed, van der Aalst 2002].

Als weiteres Problem erweist sich die Tatsache, dass die Anbindung der am Workflow beteiligten Partner nicht dynamisch zur Laufzeit erfolgen kann. Vielmehr müssen die beteiligten Partner schon zur Build-Time

bekannt sein. Die dynamische Anbindung ist Ziel weiterer Forschungsaktivitäten des Lehrstuhls von Prof. Steinmetz im Rahmen des E-Finance Lab Projekts.

4 Einfluss der Web Service Technologie auf die weitere Entwicklung des Finanzsektors

In Kapitel 3 wurde anhand einer prototypischen Implementierung gezeigt, wie Banken ihre internen Geschäftsprozesse auf Grundlage der Web Service Technologie flexibler als bisher gestalten können. Die Funktionalität bestehender Legacy-Anwendungen wird gekapselt und als Web Service angeboten. Die Koordination und Komposition dieser Web Services wird durch BPEL realisiert. Die Kommunikation erfolgt hierbei anhand offener XML-basierter Protokolle.

Die Generierung neuer Bankprodukte und Finanzdienstleistungen wird durch dieses Konzept wesentlich vereinfacht. Eine aufwendige Veränderung der Legacy-Anwendungen bzw. Konfiguration proprietärer EAI-Lösungen entfällt weitgehend. Die Erstellung innovativer Bankprodukte kann im Idealfall durch die Integration zusätzlicher Web Services in den Geschäftsprozess bzw. durch eine Orchestrierung neuer und bereits bestehender Web Services erfolgen. Ziel sollten Bankprodukte sein, die modular aus Bausteinen kombinierbar sind [König 2002, S. 518]. Änderungen an einem mit BPEL modellierten Geschäftsprozess haben keine Auswirkungen auf die darunterliegende IT-Landschaft. Legacy-Systeme können bei Bedarf sukzessive abgeschaltet und durch Neuimplementierungen ersetzt werden.

Zudem hat die Web Service Technologie das Potential als technische Infrastruktur für zukünftige „*Financial Value Networks*“ zu fungieren, welche gewissermaßen als Virtuelle Banken bzw. Finanzdienstleister verstanden werden können [Leist, Winter 1998, S. 121-138; Nirschl, Rill, Wimmer 2003, S. 1-2]. Die Idee besteht darin, eine Vielzahl von Bank-/Finanzprozessen an externe Provider auszulagern und deren Dienstleistungen und Produkte unter dem eigenen Namen zu vermarkten (Abbildung 4). Dieser Ansatz verspricht Kostenvorteile, da externe Provider (Spezialbanken, Non-/ Near-Banks) von *Economies of scale-Effekten* profitieren können.

So sind Geschäftsmodelle denkbar, die sich dadurch auszeichnen, dass Finanzdienstleister alle ihre Produkte von externen Providern beziehen und unter dem eigenen Namen anbieten.

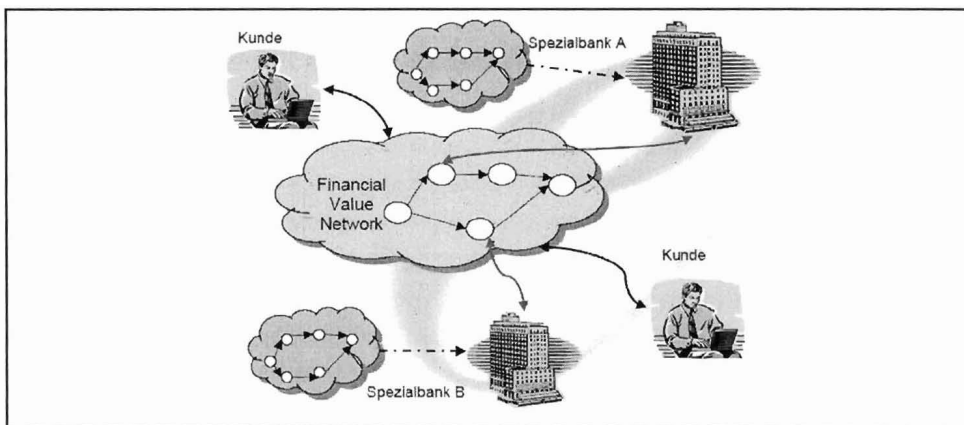


Abbildung 4: Financial Value Network.

Folgt man diesem Ansatz, so sind innovative Geschäftsmodelle wie z.B. Financial Marketplaces denkbar, auf denen Produkte wie z.B. Girokonten, Fonds, Ratenkredite, Immobilienfinanzierungen und Versicherungen von verschiedenen Herstellern angeboten werden, die Real-Time vom Kunden bezogen werden können. Für die Kunden kann dies weitgehende Transparenz und Kostenersparnisse bedeuten; für Finanzdienstleister eine Herausforderung, sich weiter zu spezialisieren, um sich in Preis und Qualität von Konkurrenten zu unterscheiden.

Alle diese Ansätze können jedoch nur realisiert werden, wenn die Geschäftsprozesse der Finanzdienstleister optimal durch IT-Systeme unterstützt werden. Zudem muss eine effektive Integration und Kopplung der IT-Systeme kooperierender Finanzdienstleister möglich sein. Web Services haben hierzu das Potential. Sie basieren auf offenen, XML-basierten Standards und sind unabhängig von zugrunde liegenden Plattformen, Betriebssystem und Programmiersprachen. Allerdings ist es in diesem Zusammenhang unabdingbar, dass die Web Service Technologie weiter ausreift, vor allem im Bereich Sicherheit.

5 Zusammenfassung

Um als Finanzdienstleister zukünftig am Markt erfolgreich bestehen zu können, ist es von zentraler Bedeutung, dass Geschäftsprozesse flexibel gestaltet und (zumindest teilweise) ausgelagert werden können (Outsourcing). Grundvoraussetzung hierfür ist eine optimale Unterstützung der Geschäftsprozesse durch die zugrunde liegende IT-Infrastruktur. Gerade die Kopplung der IT-Systemen des In- und Outsourcers erweist sich technisch als nur schwer realisierbar, da bei vielen Finanzdienstleistern die Heterogenität und Komplexität der IT-Landschaft kaum noch beherrschbar ist.

Die Web Service Technologie ist ein noch junger Ansatz, der mit dem Anspruch antritt, Heterogenität der IT-Systeme zu überwinden und flexible Geschäftsprozesse zu unterstützen. Web Services basieren auf offenen, XML-basierten Standards und sind somit plattform- sowie sprachenunabhängig.

Anhand einer prototypischen Implementierung eines Kreditprozesses wurde gezeigt, wie mit BPEL flexible Geschäftsprozesse auf Grundlage der Web Service Technologie realisiert werden können.

Web Services können zukünftig auch als technische Infrastruktur für innovative Geschäftsmodelle, wie z.B. Financial Value Networks verwendet werden.

Voraussetzung für einen umfassenden Einsatz der Web Service Technologie in Unternehmen des Finanzsektors ist jedoch, dass die Web Service Spezifikationen weiter ausreifen.

Literatur

- ANDREWS, T. ET AL. (2003), Specification: Business Process Execution Language for Web Services Version 1.1, 05.05.2003.
- ATKINSON, B. ET AL. (2002), Specification: Web Services Security (WS-Security), Version 1.0, 05.04.2002.
- BELLWOOD, T. ET AL. (2002), UDDI Version 3.0, UDDI Spec Technical Committee Specification, 19.07.2002.
- BOOTH, D. ET AL. (2004), Web Services Architecture, W3C Working Group Note 11 February 2004.
- BREUER, R.-E. (2003), Verkrustungen aufbrechen - Neue Potenziale erschließen, 8. Internationale Handelsblatt-Jahrestagung, „Banken im Umbruch“, Frankfurt am Main, 03.09.2003.
- CHRISTENSEN, E. ET AL. (2001), Web Services Description Language (WSDL) 1.1, W3C Note 15.03.2001.
- GUDGIN, M. ET AL. (2003), SOAP Version 1.2, W3C Recommendation 24.06.2003.
- HACKMANN, J. (2001), Services treiben den Preis für EAI-Projekte, in: Computerwoche Nr. 40, Beitrag vom 05.10.2001, S. 70.
- KATZENSTEINER, T. ET AL. (2004), Business Process Outsourcing: Grenze fließend, in: Handelsblatt, Beitrag vom 12.03.2004.
- KÖNIG, W. (2002), Industrialisierung des Bankgeschäfts, WI-Editorial, in: Wirtschaftsinformatik 44 (2002) 6, S. 517–518.
- KREGER, H. (2001), Web Services Conceptual Architecture (WSCA 1.0), IBM Software Group, Mai 2001.
- LEIST, S., WINTER, R. (1998), Component-Based Banking - Modularisierung der Informationsverarbeitung in Banken als Grundlage virtueller Geschäftskonzepte, in: Weinhardt, C. et al. (Hrsg.), Informationssysteme in der Finanzwirtschaft, Springer, Berlin, 1998, S. 121-138.

- LEYMANN, F., ROLLER, D. (2002), Web Services: Business processes in a Web services world, A quick overview of BPEL4WS, IBM, August 2002.
- LUND, V.T. ET AL. (2003), From banks to banking: The on demand journey, IBM Business Consulting Services, 2003.
- NATIS, Y.V. (2003), Service-Oriented Architecture Scenario, Gartner Group, AV-19-6751, 16.04.2003.
- NIRSCHL, M., RILL, M. WIMMER, A. (2003), Cross enterprise collaboration in the banking industry using Web services and BPEL, in: Jardim-Goncalves, R. et al., Concurrent Engineering - Enhanced Interoperable Systems, Proceedings of CE 2003, S. 419-426.
- O.V. (2003A), Banken - Wege aus der Krise, Internet: <http://www.business-wissen.de>, Beitrag vom 07.05.2003.
- O.V. (2003B), UBS und SAP: Partnerschaft mit Perspektive, Konsequenter Neubeginn, in: SAP Info 104, April 2003, S. 60-63.
- O.V. (2003C), OASIS members form Web Services Business Process Execution Language (WSBPEL) Technical Committee, 29.04.2003.
- STEINMETZ, R., NAHRSTEDT, K. (2004), Multimedia Applications, Springer, 2004.
- WEB SERVICES ARCHITECTURE WORKING GROUP (2004), Web Services Glossary, W3C Working Group Note 11.02.2004.
- WOHED, P., VAN DER AALST, W.M.P. (2002), Pattern Based Analysis of BPEL4WS, Technical Report, TU Eindhoven, 2002.
- YANG, J. PAPAZOGLU, M.P., VAN DEN HEUVEL, W.J. (2002), Tackling the Challenges of Service Composition in E-Marketplaces, in: Proceedings of 12th International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems (RIDE'02).