



Fallstudiensammlung

Fallstudie Deutsche Post Brief

Serviceorientierte Architektur

Roger Heutschi, Jan Schemm

Institut für Wirtschaftsinformatik,
Universität St. Gallen

Inhaltsverzeichnis

1	Deutsche Post Brief	1
1.1	Unternehmen	1
1.2	Herausforderungen im Markt	1
2	Ausgangssituation	2
3	Implementierung einer serviceorientierten Architektur.....	2
3.1	Ziele und Überblick über die Teilprojekte	2
3.2	Entwicklung der fachlichen Applikationsarchitektur	3
3.3	Implementierung von Services	5
3.4	Entwicklung der Integrationsarchitektur (Service Backbone)	8
3.4.1	Service Backbone-Architektur	8
3.4.2	Implementierung und Technologieauswahl	10
4	Bisherige Erfahrungen und geplante Weiterentwicklungen.....	11
4.1	Nutzen der serviceorientierten Architektur	11
4.2	Herausforderungen und Erfolgsfaktoren	12
4.3	Geplante Weiterentwicklungen	13
5	Erkenntnisse aus der Fallstudie	13
6	Literatur	15

Zusammenfassung

Da die existierende, historische gewachsene Applikationslandschaft den heutigen Agilitätsanforderungen aus dem Geschäft nicht mehr gerecht wurde, beschloss der Briefdienstleister Deutsche Post Brief 1999 die Neugestaltung der IT-Architektur nach den Prinzipien einer Serviceorientierten Architektur (SOA). In einem an Geschäftsprozessen und Fachobjekten orientierten Vorgehen zerlegte das Unternehmen die Applikationsarchitektur in Domänen, harmonisierte Anwendungssysteme und entwickelte Services für die domänenübergreifende Applikationsintegration. Die technische Basis dazu bildet eine zentrale Integrationsinfrastruktur, die nach dem Best-of-Breed Ansatz Infrastrukturkomponenten verschiedener Anbieter aber auch Eigenentwicklungen zusammenführt. Deutsche Post Brief hat zum Zeitpunkt der Fallstudienenerhebung (Anfang 2005) 20-30 Services implementiert und baut sowohl die fachliche Servicearchitektur als auch die Integrationsinfrastruktur laufend weiter aus.

1 Deutsche Post Brief

1.1 Unternehmen

Mit 83 Briefzentren, 3'500 Auslieferungsstützpunkten und ca. 13'000 Filialen ist der Unternehmensbereich Brief von Deutsche Post World Net der grösster Briefdienstleister Europas. Deutsche Post Brief verarbeitet täglich rund 72 Mio. Sendungen für 3 Mio. Geschäftskunden und 39 Mio. private Haushalte. Neben dem nationalen und internationalen Briefversand bietet der Unternehmensbereich Brief-bezogene Mehrwertdienstleistungen sowie Leistungen in den Bereichen Presse Distribution, Direkt Marketing und Philatelie an. Im Jahr 2003 erwirtschaftete Deutsche Post Brief mit 133'651 Mitarbeitern einen Umsatz von 11'934 Mio. EUR.

DEUTSCHE POST BRIEF	
<i>Gründung</i>	1924 Gründung der Deutschen Reispst als selbständiges Unternehmen, 1989 Umwandlung von öffentlich- in privatrechtliches Unternehmen
<i>Firmensitz</i>	Bonn, Deutschland
<i>Branche</i>	Logistik
<i>Firmenstruktur</i>	Eigenständiger Unternehmensbereich im Konzern Deutsche Post World Net
<i>Geschäftsbereiche</i>	Brief Kommunikation, Presse Distribution, Global Mail, Direkt Marketing, Philatelie
<i>Umsatz</i>	2003: 11,934 Mio. EUR
<i>Mitarbeiter</i>	2003: 133,651
<i>Homepage</i>	http://www.deutschepost.de

Tabelle 1-1: Kurzportrait Deutsche Post Brief

1.2 Herausforderungen im Markt

In den vergangenen Jahren machten sich für Deutsche Post Brief zunehmend der globale Konkurrenzdruck und sinkende Margen bemerkbar. Kürzere Produktlebenszyklen und die Anforderung, bei globaler Präsenz eine konsistente Kundenansprache zu realisieren, setzen eine kurze „Time-to-market“ für neue Leistungen voraus. Ferner war die Fähigkeit gefordert, Unternehmenszusammenschlüsse schnell umzusetzen. So hat Deutsche Post Brief in den vergangenen 10 Jahren mindestens ein halbes Dutzend Unternehmen jährlich integriert [Helbig 2005].

2 Ausgangssituation

Auslöser für die Implementierung einer serviceorientierten Architektur (SOA) im Unternehmensbereich Brief war die Erkenntnis, dass die existierende IT-Architektur den heutigen Agilitätsanforderungen aus dem Geschäft nicht mehr gerecht wurde. Der Unternehmensbereich besass bis dahin keine fachbereichsübergreifende, zentral geplante IT-Architektur. Projekte aus den Fachbereichen wurden in der Vergangenheit zwar in Zusammenarbeit mit der zentralen IT umgesetzt, eine projektübergreifende Koordination und Planung fehlte jedoch. Die Folgen davon waren spezialisierte, isolierte Anwendungssysteme mit grossen funktionalen und datenbezogenen Redundanzen. Die geringe Verfügbarkeit übergreifender Kerninformationen (Kunden, Kosten, Wettbewerberinformationen etc.) und heterogene, für Einzelprojekte implementierte Systemschnittstellen mit vielen undokumentierten Abhängigkeiten führten bei neuen Projekten zu einem hohem Anpassungsaufwand sowie mangelhafter Wiederverwendung bestehender Funktionalität. IT-Projekte wurden zunehmend risikoreicher und konnten neue Geschäftsanforderungen nicht mehr in der geforderten Zeit umsetzen. Die gewachsene IT-Architektur war ausserdem sehr teuer im Unterhalt. Immer grössere Anteile des IT-Budgets flossen in den Betrieb und die Wartung von Systemen und nur ein geringer Teil stand für Neu- bzw. Weiterentwicklungen, z.B. zur Unterstützung neuer Prozesse oder Kundekanäle, zur Verfügung [Herr/Brombach 2004].

Als Reaktion auf diese Herausforderungen beschloss Deutsche Post Brief, eine IT-Architektur nach den Prinzipien serviceorientierter Architekturen (SOA) zu entwickeln. Die zentrale Zielsetzung bestand darin, unternehmerische Agilität und technische Flexibilität durch eine bessere Ausrichtung der IT auf Geschäftsanforderungen (Business / IT Alignment) und eine stärkere Entkopplung von Geschäft bzw. Prozessen und unterstützender Technologie zu erreichen [Helbig 2005].

3 Implementierung einer serviceorientierten Architektur

3.1 Ziele und Überblick über die Teilprojekte

1999 initiierte Deutsche Post Brief ein strategisches Projekt zur zentralen, geschäftsorientierten Neugestaltung der Applikationsarchitektur. Ziel dieses Projekts war es, eine *logische Applikationsarchitektur* als Generalbebauungsplan für eine redundanzfreie Implementierung von Anwendungsfunktionalität und eine konsistente Datenpflege zu entwickeln. Damit sollte eine bessere Wiederverwendbarkeit von

Funktionalität zur Unterstützung neuer Prozesse sowie ein gezielter und isolierter Ausbau um zusätzliche Funktionalität erreicht werden. Die Ansiedlung des Projekts im Vorstand des Unternehmensbereichs Brief sorgte für die notwendige Managementunterstützung. Resultat des Projekts war eine fachliche serviceorientierte Architektur, bestehend aus voneinander abgegrenzten Anwendungsdomänen und Serviceschnittstellen für den Zugriff auf Funktionen und Daten in diesen Domänen.

Das strategische Projekt führte zu einer Reihe von Implementierungsprojekten: Einerseits wurden verschiedene *Fachprojekte* für die Implementierung einzelner Domänen und Services (z.B. der Aufbau einer zentralen Kundendatenbank) angestoßen. Andererseits entwickelte die IT-Abteilung den *Service Backbone (SBB)*, eine Integrationsplattform für die Kopplung von Service-Anbietern und -Nutzern. Der SBB wurde Ende 2001 in einer ersten Version produktiv eingesetzt.

Im Rahmen der Umstellung auf SOA bildete Deutsche Post Brief den zentralen IT-Bereich „Service Oriented Platform (SOP)“ Group, welcher die Weiterentwicklung des SBB und seine Verbreitung in den Unternehmensbereichen Brief sowie Express / Logistik (DHL) verantwortet.

3.2 Entwicklung der fachlichen Applikationsarchitektur

Ausgangsbasis für die Entwicklung der fachlichen Applikationsarchitektur waren die geschäftlichen Anforderungen der Fachbereiche [Herr/Bath 2004]. Ein Projektteam, bestehend aus Prozessverantwortlichen aus den Fachbereichen sowie Mitarbeitern aus der Fach-IT, analysierte Kernprozesse wie Produktentwicklung, Kampagnenmanagement, Kanalstrategie, Reklamationsmanagement oder Marketing und Vertriebsplanung. Im Rahmen dieser Analyse fand in verschiedenen Fällen auch ein Prozessredesign statt. Nicht in die Untersuchung eingeschlossen wurden Unterstützungsprozesse wie Personalmanagement oder Finanzen, die standardisiert sind und für das Unternehmen kein Differenzierungspotenzial bieten. Ausgeschlossen wurden auch bereits sehr stark IT-durchdrungene Prozesse wie die Routenoptimierung, für die spezialisierte IT-Lösungen existierten oder sehr spezifische Prozesse wie Philatelie (Briefmarkenkunde).

Das Projektteam zerlegte die Geschäftsprozesse in ihre betriebswirtschaftlichen funktionalen Bestandteile und bestimmte die generischen Geschäftsfunktionen und Datenflüsse, aus denen sich die Prozesse zusammensetzen. Im Anschluss wurde untersucht, welche der identifizierten Geschäftsfunktionen in mehreren Prozessen vorkommen. Die Geschäftsfunktionen mit Bedarf an IT-Unterstützung wurden anhand ihrer wechselseitigen Abhängigkeiten geclustert und fachlichen Applikationsdomänen zugeordnet. Als Cluster-Kriterium zur Abgrenzung der Domänen dienten Datenströme (Input-Output-Beziehungen) zwischen den einzelnen Geschäftsfunktionen: Gruppen

von Funktionen mit vielen und komplexen Austauschbeziehungen untereinander wurden jeweils in eigenen Domänen zusammengefasst, wobei die Domänen untereinander möglichst wenige Schnittstellen aufweisen sollten. Für den Domänen-externen Zugriff auf Funktionen und Daten definierte das Projektteam Serviceschnittstellen. Domänenübergreifende Geschäftsprozesse wie etwa das Kampagnenmanagement nutzen die in den Domänen gekapselte Funktionalität über mehrere Services (s. Abbildung 3-1).

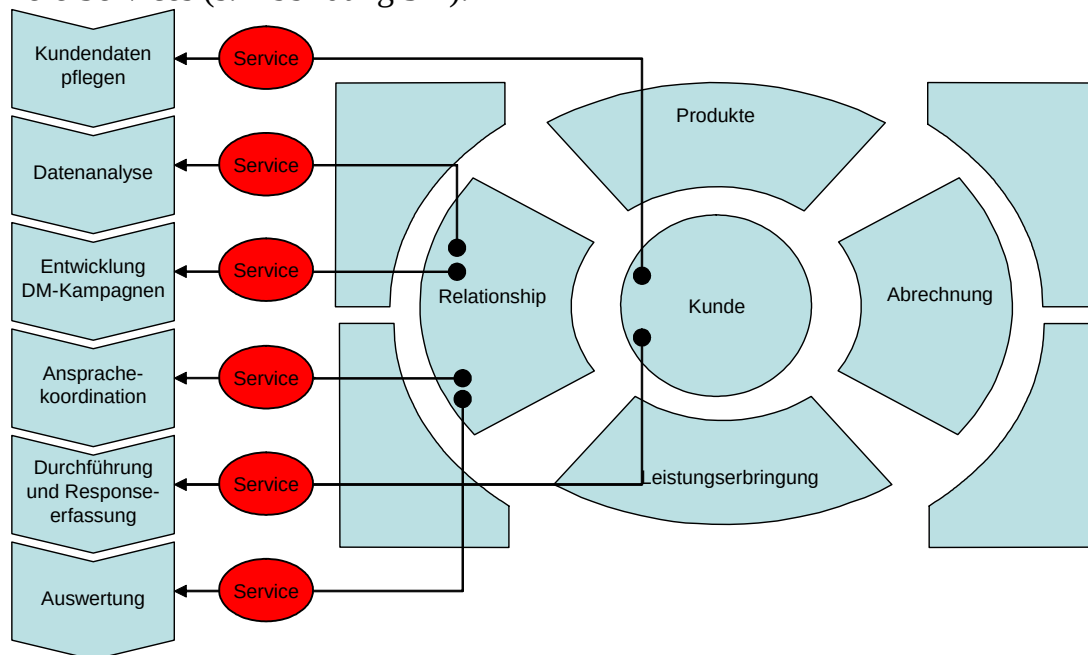


Abbildung 3-1: Applikationsdomänen und Zugriff über Services am Beispiel des Prozesses Kampagnenmanagement [vgl. Helbig 2005]

Die Hauptkriterien, nach denen Services gebildet wurden, waren:

- *Generizität und Wiederverwendungspotenzial.* Services sollten nicht für einen spezifischen Service-Nutzer ausgelegt, sondern in unterschiedlichen Prozessen und Applikationen nutzbar sein.
- *Fachlich sinnvolle und beschreibbare Dienstleistung.* Die Services kapseln die fachliche Geschäftslogik, welche einen Geschäftsnutzen bietet, und sind aus fachlicher Sicht beschreibbar.
- *Langfristige Stabilität.* Die Serviceschnittstellen sollten möglichst stabil sein. Dies hoffte das Projektteam aus fachlicher Sicht zu erreichen, indem es die Services entlang der generischen Geschäftsfunktionen und Informations- bzw. Geschäftsobjekten, die sich im Zeitverlauf wenig ändern, definierte. Aus technischer Sicht sollen Services eine stabile Abstraktionsschicht darstellen, die Serviceanbieter von den Servicenutzern trennt und eine schnelle und einfache Änderung der Serviceimplementierung erlaubt. Eine Massnahme zur Förderung

der technischen Stabilität war die Verwendung breit akzeptierter Industriestandards (UML, XML, WSDL etc.) für die Servicespezifikation und -entwicklung.

Das Vorgehen zur Identifikation von Services orientierte sich an Ansätzen der objekt-orientierten Softwareentwicklung [Bath 2005]. Als Basis für die Serviceidentifikation diente, abgeleitet von den zu Anwendungsdomänen zusammengefassten Geschäftsfunktionen und Geschäftsobjekten, eine Sammlung von Fachklassen. Diese Klassen stellten Kandidaten für mögliche Services wie z.B. „CustomerManagement“ dar. Anhand der Basisoperationen Create, Read, Update, Delete (CRUD) auf den Fachklassen und deren wichtigsten Attributen wurden Serviceoperationen wie bspw. „createCustomer“ oder „readCustomer“ definiert. In einem nächsten Schritt fand eine Eliminierung überflüssiger Services durch eine Analyse von Use Cases und Domänenbeziehungen statt: In den Use Cases wurden nicht benötigte Operationen sowie Services und Operationen herausgefiltert, die kein Wiederverwendungspotenzial haben oder nur innerhalb einer Domäne Verwendung finden. Die Use Case Analyse gab ausserdem Hinweise auf fehlende Services und Operationen.

Das über neun Monate laufende Projekt (ca. 6 Monate wurden alleine für die Prozessanalyse aufgewendet) entwickelte eine fachliche Applikationsarchitektur, welche die fachliche Funktionen und Daten prozessunabhängig und redundanzfrei in den Applikationsdomänen abbildet. Jeder Zugriff auf die in den Domänen gekapselte Geschäftslogik soll zukünftig über Serviceschnittstellen stattfinden. Teilweise kommunizieren auch Sub-Cluster innerhalb der Domänen über Services. Momentan bieten die Domänen 20-30 Serviceschnittstellen, die jeweils mehrere Operationen unterstützen. Wie sich im Nachhinein zeigte, richten sich die identifizierten Services derzeit grösstenteils nach Geschäftsobjekten wie „Kunde“, „Rechnung“ oder „Vertrag“ und bieten datenbearbeitende Operationen wie „anlegen“, „suchen“ oder „löschen“. Wenige Services (z.B. Dublettenprüfung oder Adressvalidierung) kapseln komplexere Geschäftslogik, die sich an Prozessaktivitäten orientiert.

3.3 Implementierung von Services

Im Anschluss an die fachlich-logische Definition der Services im Rahmen der Applikationsarchitektur begann das Unternehmen mit der Implementierung der identifizierten Services. Dabei legte es die Priorität auf Services, die zur Unterstützung neuerer Anwendungsbereiche mit hoher Dynamik (z.B. Customer Care, Call Center und Multi Channel Management) notwendig waren. Mit der Implementierung von Services ging teilweise auch eine Harmonisierung der Anwendungssystemlandschaft einher: Um Komplexität und Wartungs- bzw. Betriebsaufwände zu reduzieren, verfolgt die IT insbesondere bei Neuentwicklungen

das Ziel, die Servicefunktionalität physisch möglichst in einem zentralen System (z.B. einer zentralen Kundendatenbank) zu realisieren. Dies ist jedoch nicht immer möglich, z.B. wenn für den Service benötigte Daten und Funktionen über mehrere Altanwendungen verteilt sind.

Ein Beispiel für einen implementierten Service stellt der Service „Kundenmanagement“ dar [Krafzig et al. 2004]. Dieser bietet Nutzern wie bspw. Call-Center-, Portal- oder CRM-Systemen über die Operationen „seek“, „get“, „create“, „update“, „delete“, „getChildNodes“ und „checkAdress“ die Möglichkeit, Kundendaten zu suchen, erfassen, modifizieren und validieren. Die Servicenutzer bearbeiten die Entität Kunde über diese Operationen nicht auf Basis einzelner Attribute (Name, Vorname, Adresse etc.), wie dies bei einem Zugriff in Form von Objektmethoden typischerweise der Fall wäre, sondern erhalten und liefern jeweils eine strukturierte XML-Nachricht, welche sämtliche Kundeninformationen (ca. 100 Attribute) enthält. Dadurch steigt zwar die bei der Servicenutzung ausgetauschte Datenmenge, die Anzahl Operationen bzw. Schnittstellen und die Häufigkeit der Interaktion zwischen Serviceanbieter und -nutzer wird aber niedrig gehalten. Abbildung 3-2 zeigt schematisch die Anpassungen der bestehenden Anwendungssystemlandschaft bei der Umsetzung des Services: Existierten früher über 40 unterschiedliche Applikationen mit lokaler Datenhaltung, wurde für den Service eine neue zentrale Stammdatenbank implementiert. Diese setzt hauptsächlich Datenbearbeitungslogik um, mit nur geringem Anteil an Geschäftslogik bspw. für die Identifikation von Dubletten. Verschiedene bereits existierende Anwendungen, insbesondere aber neue Anwendungen die Kundendaten verwenden, nutzen das Stammdatensystem über die Serviceschnittstelle. Derzeit existieren rund ein Dutzend Servicenutzer. Nur wenige Applikationen (z.B. mobile Anwendungen ohne dauerhaften Online-Zugang) verwenden noch lokale Kundendaten, die sie aber regelmässig mit der zentralen Datenbank synchronisieren. Für diese Datensynchronisation werden spezielle, domäneninterne Serviceschnittstelle verwendet.

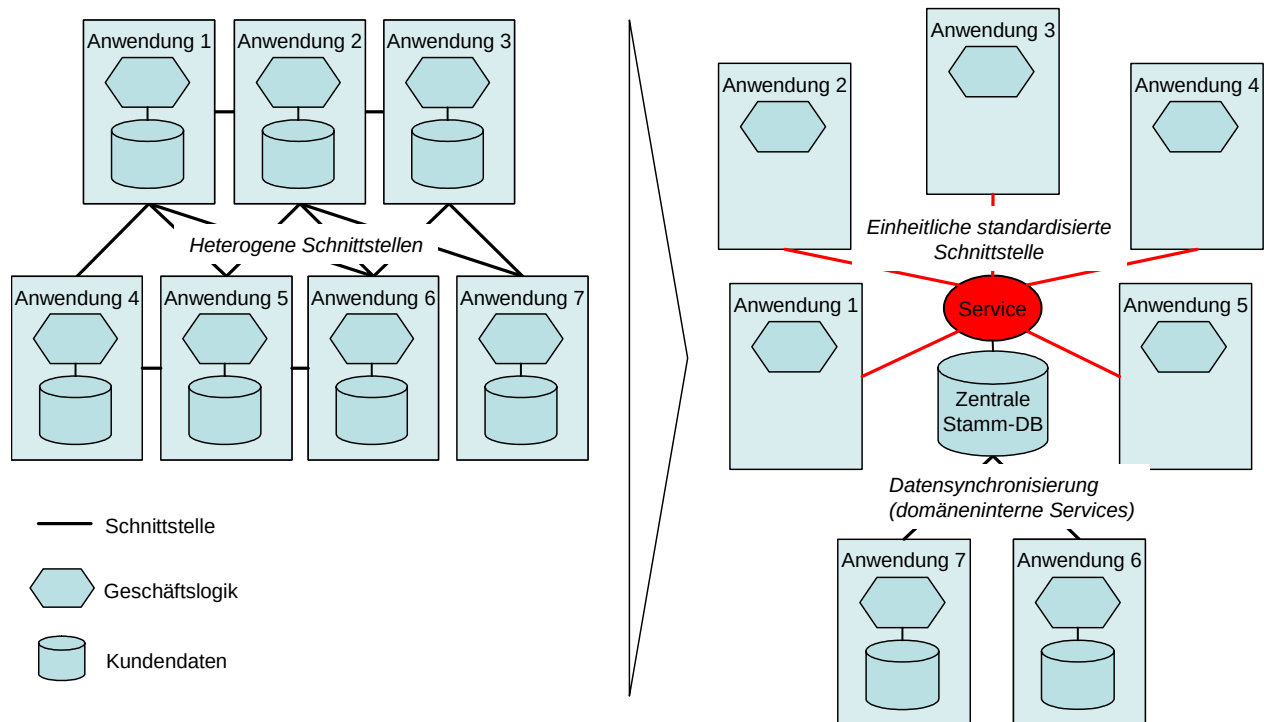


Abbildung 3-2: Implementierung des Services "Kundenmanagement"

Der Hauptaufwand für die Implementierung des Services entstand in der Entwicklung des Datenmodells der Schnittstelle. Um den Service im Zeitablauf möglichst stabil und generisch genug zu bilden, muss das Datenmodell alle von aktuellen und zukünftigen Servicenutzern geforderten Sichten auf die Entität „Kunde“ abbilden können. Das bedeutet, dass der Service bspw. sowohl für einen „Kampagnen-Kunden“ im Rahmen einer Marketing-Kampagne als auch für einen „Rechnungskunden“ im Rahmen der Auftragsabwicklung die benötigten Informationen (z.B. Kundentyp, Anschrift) in der notwendigen Granularität (z.B. Unterscheidung von Liefer- und Rechnungsanschrift) liefern können muss.

Für jeden implementierten Service existiert neben der technischen Beschreibung der Servicenutzung im zentralen Service Registry (s. Kap. 3.4) eine fachliche Servicebeschreibung. Diese beschreibt die für den Serviceaufruf notwendigen bzw. die vom Service gelieferten Datenelemente (Nachrichten, fachliche Datentypen, Fachklassen, Fehlermeldungen, Geschäftsregeln) sowie die Beziehungen zwischen diesen Elementen (Assoziationen, Generalisierung, Komposition, Abhängigkeiten), notwendige Datentransformationen, das dynamische Verhalten des Services (Sequenzdiagramme), nichtfunktionale Eigenschaften (Servicequalität bzw. Verfügbarkeiten, Antwortzeiten etc.) sowie Beispiele für Daten und Serviceaufrufe.

Um eine abteilungs- bzw. fachbereichsübergreifend konsistente Serviceentwicklung zu erreichen, wurde der Prozess der Serviceentwicklung formalisiert. Die Implementierung eines neuen Services umfasst folgende Schritte [Herr/Bath 2004]:

- Identifikation aller potenzieller Servicenutzer
- Spezifikation der Geschäftsfunktionalität unter Berücksichtigung der Anforderungen aller potenzieller Servicenutzer
- Abgleich der Geschäftsfunktionalität mit den Applikationsdomänen und den bereits implementierten Services
- Erstellung der Servicebeschreibung, der Service Level Agreements (SLAs) und des XML-Schemas zur Publikation des Services im Service Registry
- Serviceimplementierung
- Anbindung des Services an den Service Backbone
- Registrierung des Services im SBB User Directory

Um einen neuen Servicenutzer anzubinden, sind zwei Schritte notwendig:

- Erfassung der Nutzerinformationen im SBB User Directory für die Authentifizierung und Autorisierung
- Lokale Implementierung der Serviceschnittstelle bei der Nutzerapplikation

3.4 Entwicklung der Integrationsarchitektur (Service Backbone)

Der sog. *Service Backbone (SBB)* stellt den Kernbestandteil der technischen Umsetzung der SOA bei Deutsche Post Brief dar und unterstützt als Kernfunktion die Kommunikation zwischen Serviceanbietern und -nutzern.

3.4.1 Service Backbone-Architektur

Die Hauptkomponenten des SBB lassen sich in die in Abbildung 3-3 dargestellten Bereiche der zentralen Infrastruktur (*Core*), der technischen Serviceteilnehmer (*Technical Service Participants*), der Schnittstellen (*Interface*) sowie unterstützender Werkzeuge (*Tools*) für Anwendungsentwickler unterteilen (vgl. [Herr et al. 2004], [Herr/Sannemann 2004]).

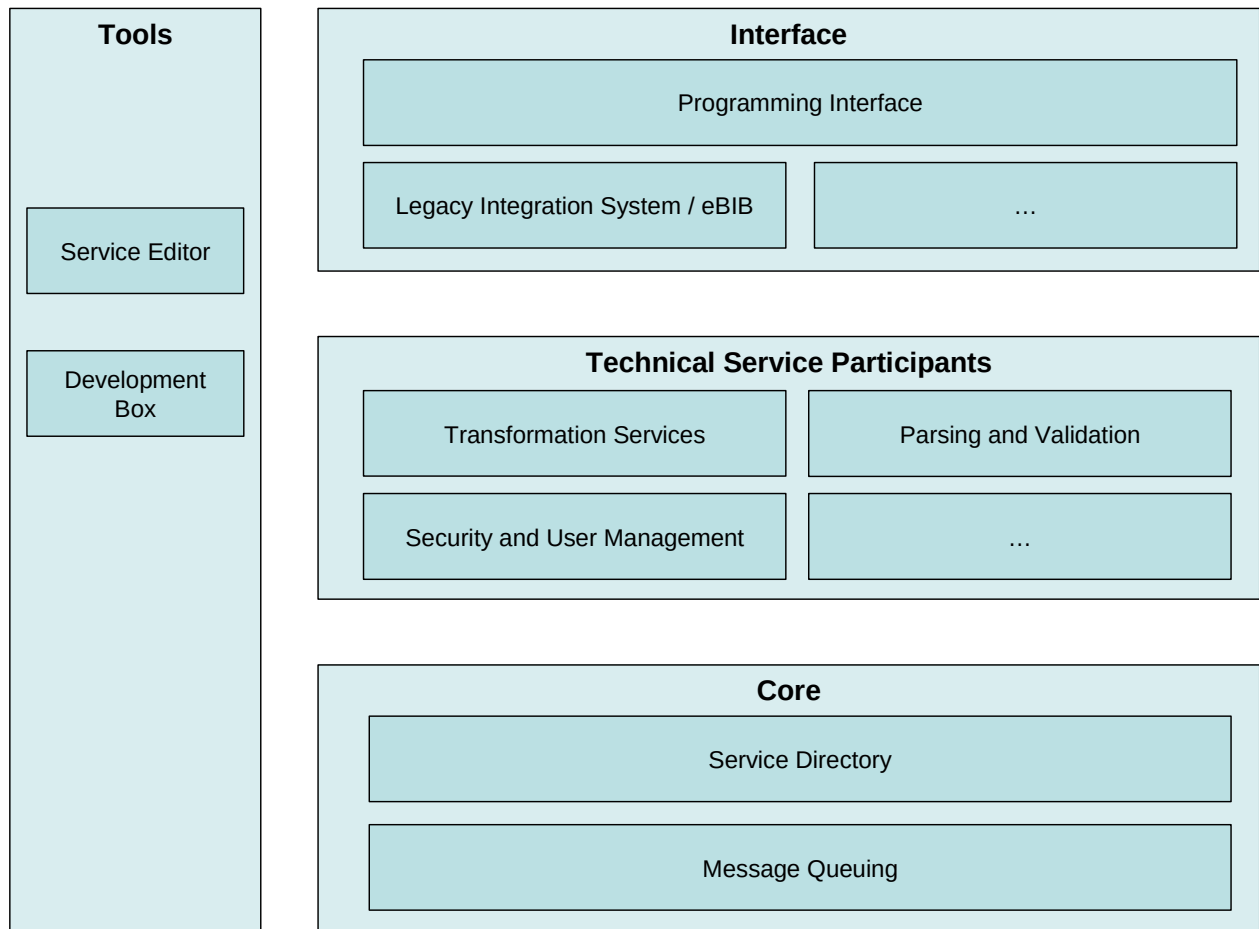


Abbildung 3-3: Aktuelle Komponenten des SBB (in Weiterentwicklung) [vgl. Bath 2005]

Die zentrale Infrastruktur des SBB umfasst die Komponenten *Service Directory* sowie *Message Queuing*. Das *Service Directory* dient als zentrales Verzeichnis bzw. zentrale Registrierung für Services und ermöglicht in erster Linie ein dynamisches Binden von Services zur Laufzeit. Servicenutzer verwenden lediglich eine logische Adressierung von Diensten in Form eines Uniform Resource Locators (URL). Die Übersetzung in eine physische Adresse zur Laufzeit übernimmt der SBB. Neben der Funktion der logischen Adressierung übernimmt das *Service Directory* die Authentisierung und Autorisierung beim Zugriff auf Services unter Rückgriff auf Benutzer- und Berechtigungsinformationen. Darüber hinaus ermöglicht die Komponente eine Versionierung von Services, d.h. Serviceanbieter können unterschiedliche Versionen von Diensten gleichzeitig zur Verfügung stellen. Die Implementierung basiert auf den Protokollen LDAP und HTTP, wobei Standard-Implementierungen von entsprechenden Servern eingesetzt werden. Die Kernfunktion der Komponente *Message Queuing* besteht in der Bereitstellung von Mechanismen zur Sicherstellung der reibungslosen Kommuni-

kation zwischen Serviceanbieter und -nutzer während der Nutzung eines Dienstes. Insbesondere ermöglicht die Komponente asynchrone und persistente Serviceaufrufe. Das Message Queuing basiert auf dem Standard *Java Messaging Service (JMS)* und unterstützt verschiedene Interaktionsstile wie bspw. synchron, asynchron und publish/subscribe. Die Implementierung der Komponente basiert auf einem JMS-konformen Message Broker.

Die *technischen Serviceteilnehmer* stellen Komponenten dar, die klar definierte Integrationsfunktionen über SBB-Schnittstellen anbieten. Beispiele sind die Komponenten *Security and User Management* (stellt Funktionalität zur zentralen Administration von Benutzerdaten und Zugriffsberechtigungen zur Verfügung), *Transformation Service* (stellt Routinen für die Transformation von Nachrichten und Daten bereit) oder *Parsing and Validation* (prüft die Gültigkeit der über den SBB ausgetauschten Nachrichten).

Der Schnittstellen-Bereich des SBB stellt u.a. ein Java-basiertes *Programming Interface (API)* sowie ein *Legacy Integration System / eBIB* für die Anbindung von Altsystemen bereit.

3.4.2 Implementierung und Technologieauswahl

Bei der Implementierung des SBB spielte für Deutsche Post Brief die Unabhängigkeit von einzelnen Anbietern eine grosse Rolle. Das Unternehmen wählte deshalb einen Best-of-Breed Ansatz, der Infrastrukturkomponenten verschiedenster Anbieter aber auch Eigenentwicklungen zusammenführt. So basiert die Plattform z.B. auf IBM WebSphere MQ, dem Sun One Directory Server, den Tomcat Apache Server oder den Produkten der Bea Weblogic Platform [Herr/Bath 2004]. Die Implementierung setzt so weit wie möglich auf Industriestandards und versucht, herstellerepezifische Erweiterungen zu vermeiden. Die Plattform verwendet z.B. WebSphere MQ für das Message Handling, verzichtet aber auf die Nutzung proprietärer Erweiterungen und bindet das Produkt über den Java Messaging Service (JMS) ein [Krafzig et al. 2004].

Von Web Service Standards erwartet Deutsche Post Brief einen wesentlichen Beitrag zur Anbieter- und Plattformunabhängigkeit der SOA. Sie erfüllen aber momentan nicht alle Anforderungen der SOA Infrastruktur und weisen z.B. Mängel in den Bereichen Versionierung, zuverlässige asynchrone Kommunikation oder Sicherheit (Authentisierung und Autorisierung) auf [Herr/Brombach 2004]. Der Einsatz von WSDL und SOAP ist insbesondere für die zwischenbetriebliche Integration über einen Web Service Gateway geplant, der externen Partnern einen Zugriff mit reduziertem Leistungsumfang auf Services bietet. Ausserdem ist die Verwendung von UDDI für das momentan JNDI / LDAP-basierte Serviceverzeichnis geplant.

Die initialen Implementierungskosten für den SBB sind mit den Kosten für gängige herstellerepezifische Komplettlösungen wie der IBM WebSphere Plattform vergleichbar.

Für Deutsche Post Brief ergeben sich aber aus dem Best-of-Breed Ansatz Vorteile: Neben der grösseren Herstellerunabhängigkeit und geringeren Betriebskosten (z.B. durch den Einsatz von Open-Source Produkten und die Einsparung von Lizenzkosten) profitiert das Unternehmen durch mehr internes Know-how und Kompetenz und bessere Abdeckung der eigenen Bedürfnisse.

4 Bisherige Erfahrungen und geplante Weiterentwicklungen

4.1 Nutzen der serviceorientierten Architektur

Der Unternehmensbereich Deutsche Post Brief konnte dank der Einführung einer serviceorientierten Architektur insbesondere folgende Nutzenpotenziale realisieren:

- *Wiederverwendung von Services und Vermeidung redundanter Softwareentwicklung.* Die entwickelten Services werden durchschnittlich zwei bis drei mal wieder verwendet, wobei einzelne wie der Service „Kundenmanagement“ von vielen, andere nur von einer Anwendung genutzt werden. Die mit der Wiederverwendung und gleichzeitigen Systemharmonisierung einhergehende Redundanzreduktion führte zu einer verbesserten Datenkonsistenz.
- *Kürzere Projektlaufzeiten und schnellere Time-to-market.* Insbesondere bei der Neuentwicklung liessen sich durch Wiederverwendung Projektlaufzeiten von einigen Monaten (inklusive Tests) auf Tage oder Wochen reduzieren. Allerdings ist bei der erstmaligen Implementierung eines wieder verwendbaren Services im Vergleich zur herkömmlichen Entwicklung mit einem Mehraufwand von einigen Tagen oder Wochen zu rechnen.
- *Niedrigere Wartungskosten.* Mit der Domänenbildung und den besser kontrollierten Schnittstellen reduzieren sich Komplexität sowie Redundanzen. Dies führt zu niedrigeren Kosten für die Wartung und den Betrieb der IT-Architektur: Bei gleichbleibendem IT-Budget kann der Unternehmensbereich ca. 7% mehr für Neuentwicklungen ausgeben.
- *Dezentralisierung der Anwendungsentwicklung und bessere Reaktionsfähigkeit auf neue Geschäftsanforderungen.* Die Domänen- und Servicearchitektur nimmt für Deutsche Post Brief eine Vermittlerrolle zwischen der Prozess- und Applikationsarchitektur ein: Sie beschreibt und strukturiert Daten und Funktionalität der Applikationsarchitektur in der Begriffswelt des Business und erleichtert so die Kommunikation zwischen Fachbereichen und IT. Sie stellt gleichzeitig ein Gerüst für die Definition von Verantwortlichkeiten dar: Fachbereiche sind für Daten und Services einzelner Domänen verantwortlich

und besitzen grössere Entwicklungskompetenzen. Der modulare Aufbau der Domänen, klare fachliche Verantwortlichkeiten, standardisierte Architekturrichtlinien und definierte Architekturprozesse erlauben kleinere, isoliertere IT-Projekte mit geringeren Risiken.

Deutsche Post Brief sieht weitere geschäftliche und technische Nutzenhebel wie z.B. den Schutz bestehender IT-Investitionen [Herr/Brombach 2004]. Insgesamt rechnet das Unternehmen für seine Investitionen in eine serviceorientierte Architektur über fünf Jahre mit einem Nettobarwert in zweistelliger Millionenhöhe.

4.2 Herausforderungen und Erfolgsfaktoren

Bisherige Erfahrungen zeigen, dass die zentralen Herausforderungen einer SOA nicht technischer Natur sind, sondern in der *Abgrenzung und semantischen Integration der Services* liegen. Stabile und wieder verwendbare Services setzen voraus, dass die Anforderungen von (zur Entwicklungszeit) unbekannten Servicenutzern berücksichtigt und das Datenmodell bzw. die Servicesemantik so definiert wird, dass der Dienst in unterschiedlichen Geschäftsprozessen genutzt werden kann. Die Anforderungen zukünftiger Servicenutzer sind zum Entwicklungszeitpunkt i.d.R. aber nur beschränkt bekannt. Aufgrund dieser Unsicherheiten ist damit zu rechnen, dass Services für neue Projekte angepasst werden müssen, was Strukturen für ein kontrolliertes Versionsmanagement erforderlich macht.

Der Aufbau einer geeigneten *Governance-Struktur* war eine weitere Herausforderung. Die Entwicklung fachbereichsübergreifend nutzbarer Services setzt eine intensive Kommunikation zwischen dem Bereich, der den Service entwickelt und besitzt, und den nutzenden Bereiche voraus. Einerseits muss das Interesse potenzieller Servicenutzer geweckt werden, andererseits ist darauf zu achten, dass die Anforderungen der nutzenden Fachbereiche die Entwicklungskosten für den implementierenden Bereich nicht zu stark erhöhen. Um dies zu erreichen, wurde ein eigener Bereich „Service Oriented Platform (SOP)“ Group gegründet, der den Entwicklungsprozess unterstützt und die Architekturkonformität des Projekts überwacht.

Wichtige Erfolgsfaktoren aus Sicht der Architekturverantwortlichen von Deutsche Post Brief sind (vgl. [Herr/Bath 2004], [Bath 2004]):

- Eine geschäftsorientierte, aus fachlich-logischer Sicht startende Architekturentwicklung,
- die Unterstützung und organisatorische Verankerung auf Top-Managementebene und
- die Herstellerunabhängigkeit und Standardkonformität der Serviceinfrastruktur, um technologische Zukunftssicherheit und Flexibilität zu wahren.

4.3 Geplante Weiterentwicklungen

Aufgrund des Erfolgs der SOA plant Deutsche Post Brief, den Ansatz im Konzern und zu externen Geschäftspartnern hin weiter auszurollen. Im Rahmen der Bildung der SOP Group wurde bereits im Jahr 2004 eine Ausweitung auf den Unternehmensbereich Express / Logistik (DHL) vorgenommen. An der Schnittstelle zu externen Geschäftspartnern sollen in Zukunft vermehrt Web Services zum Einsatz kommen.

Weiter ist eine bessere Unterstützung der prozessorientierten Serviceintegration geplant. Der SBB besitzt momentan noch keine Business Process Management (BPM) Komponente, welche das Herauslösen von serviceübergreifender Prozesslogik aus den nutzenden Applikationen und ein Performancemanagement über End-to-End-Prozesse erlauben würde. Die Ablaufsteuerung und Koordination mehrerer Services in domänen- bzw. serviceübergreifenden Prozessen wird in den aufrufenden Applikationen (bzw. in Composite Services, die mehrere Services koordinieren und Nutzern über eine einzelne Schnittstelle zur Verfügung stellen), fest programmiert. Eine Erweiterung der Infrastruktur um eine BPM-Komponente ist für eine zukünftige Version von SBB geplant, birgt aber noch verschiedene fachliche und technische Herausforderungen.

5 Erkenntnisse aus der Fallstudie

Das Fallbeispiel zeigt einen erfolgreichen, geschäftsorientierten und von fachlichen Prozessanforderungen ausgehenden Ansatz der Identifikation und Abgrenzung von Services. Obwohl sich viele der implementierten Services stark an Geschäftsobjekten bzw. Fachklassen orientieren, attestiert das Projektteam der funktionalen Sicht eine hohe Bedeutung, um geeignete Anwendungsdomänen sowie sinnvolle Serviceoperationen und Datenmodelle zu definieren.

Die Erfahrungen des Unternehmensbereichs Brief von Deutsche Post World Net zeigen, dass SOA verschiedene Nutzenpotenziale z.B. in Form von Wiederverwendung oder schnellerer Time-to-market bieten, in der initialen Implementierungsphase jedoch kaum kurzfristige Quick-Wins zu erwarten sind: Um den in der Fallstudie geschilderten Stand zu erreichen, arbeitet das Unternehmen seit über fünf Jahren an der Einführung und Weiterentwicklung der serviceorientierten Architektur.

Die von Deutsche Post Brief realisierte SOA-Infrastruktur ist intern stark auf die Integration von Java-Applikationen ausgerichtet. Zwar existieren C++-Adapter, die Abstraktion bzw. Integration von heterogenen technischen Plattformen lag aber nicht im Hauptfokus des Unternehmens. Technische Interoperabilität spielt im

zwischenbetrieblichen Bereich eine grössere Rolle und wird dort mit der Unterstützung von Web Service Standards verfolgt.

Die Fallstudie illustriert weiter, dass es möglich und unter dem Aspekt der Herstellerunabhängigkeit empfehlenswert ist, eine komplexe Serviceinfrastruktur in einem Best-of-Breed Ansatz zu implementieren. Dies setzt allerdings einen guten Überblick über am Markt erhältliche Produkte und grosses technisches Know-how voraus.

6 Literatur

[Bath 2004]

Bath, U., Serviceorientierte Architektur - Komplexitätsmanagement durch Reduktion, Beitrag in: GI Regionaltreff Bonn 2004

[Bath 2005]

Bath, U., Serviceorientierte Architektur - Komplexitätsmanagement durch Integration, Beitrag in: 16. St. Galler Anwenderforum 2005

[Helbig 2005]

Helbig, J., Welcome and Introduction, Beitrag in: SOA Days Bonn 2005

[Herr/Bath 2004]

Herr, M., Bath, U., SBB Motivation Paper: The business-oriented background of Service Backbone,
http://www.sbb.org/uploads/images/23/SBB_Motivation_Paper_V1_0.pdf,
09.11.2004

[Herr et al. 2004]

Herr, M., Bath, U., Koschel, A., Implementation of a Service Oriented Architecture at Deutsche Post MAIL, Proceedings of the European Conference on Web Services, Erfurt, 2004, S.227-238

[Herr/Brombach 2004]

Herr, M., Brombach, S., SBB Management Paper: Benefits and application range,
http://www.sbb.org/uploads/images/26/SOP_Management_Paper_V1_0.pdf,
09.11.2004

[Herr/Sannemann 2004]

Herr, M., Sannemann, U., SBB Technical Paper: The architecture of Service Backbone,
http://www.sbb.org/uploads/images/24/SBB_Technical_Paper_V1_0.pdf,
09.11.2004

[Krafzig et al. 2004]

Krafzig, D., Banke, K., Slama, D., Enterprise SOA, Prentice Hall, 2004