



Lösungsentwicklung im Zeitalter des eCommerce

Abstract

Das Business mit seinen schnellen Veränderungen und die IT mit neuen technologischen Entwicklungen sind gemeinsam die Triebfedern für die Veränderung der IT-Landschaft eines Unternehmens. Nun stehen die IT-Abteilungen heute häufig vor dem Problem, dass die bestehenden Systeme (noch) nicht flexibel genug sind, um den schnellen Veränderungen des Business folgen zu können. Eine weitere Rahmenbedingung ist die geforderte kurze "Time-to-Market". Es gilt also eine Zielarchitektur zu entwickeln, die so flexibel gestaltet ist, dass sowohl Standardpakete vom Markt wie auch Eigenentwicklungen zum Einsatz kommen können. Dabei ist gleichzeitig zu berücksichtigen, dass eine solche Neuorientierung nur mit einer sorgfältig geplanten Migration erreicht werden kann. Das Migrationskonzept muss dabei in erster Linie der Reduktion der Komplexität dienen, um die zunehmende Komplexität des Business bewältigen zu können. Auf dieser Basis sind die verschiedenen Konzepte zur Enterprise Applikation Integration (EAI) und zu prozess- und komponentenorientierter Lösungsentwicklung entstanden.

Solution Development in the Era of eCommerce

Business with its ever-changing requirements and IT bringing up new technologies at a glance are both the driving forces for changes in the IT infrastructure of an enterprise. On the other hand IT departments in many cases face the fact that their existing Systems are not (yet) flexible enough to cope with the rapid changes in business. Furthermore the short time to market imposes additional restrictions. Therefore it is necessary to develop a target architecture that gives IT the flexibility to integrate existing applications with standard of-the-shelf packages and new custom development. The key to this strategy is a well defined migration szenario that first of all helps to reduce complexity to cope with the increasing complexity in the business areas. Based on these ideas the different concepts for Enterprise Application Integration (EAI) and process and component based development have been designed.



Lösungsentwicklung im Zeitalter des eCommerce

Vom Business zur IT-Lösung

Es gibt heute grundsätzlich zwei Triebfedern für die Veränderung der IT-Landschaft eines Unternehmens. Die erste Triebfeder ist das Business. Ständig neu entstehende Geschäftsideen und permanente Veränderungen in den bestehenden Geschäftsfeldern erfordern vom Management eine permanente Anpassung der Geschäftsprozesse. Die IT als Enabler des Business ist dadurch gezwungen, die unterstützenden Funktionen so flexibel zu gestalten, dass kurzfristige Anpassungen möglich sind und die Geschäftsprozesse möglichst durchgängig abbildbar sind. Die zweite wesentliche Triebfeder ist die technologische Entwicklung, die oftmals erst neue Geschäftsideen ermöglicht. In diesem Fall ist es Aufgabe der IT zusammen mit dem Business die Potentiale neuer Technologien zu erarbeiten und das Business bei der Formulierung, Konkretisierung und Umsetzung so initiiert neuer Geschäftsideen zu unterstützen. Die Federführung bleibt aber auch in diesem Fall beim Business-Management.

Nun steht die IT heute häufig vor dem Problem, dass die bestehenden Systeme noch nicht flexibel genug sind, um den schnellen Veränderungen des Business folgen zu können, weshalb sie oft als Bremser erfahren wird. Vor diesem Hintergrund wurden in den letzten Jahre viele Konzepte und Technologien entwickelt, die der IT schnell zu mehr Flexibilität verhelfen sollen, was zur Folge hatte, dass sich viele IT-Abteilungen aufgemacht haben, ihre Infrastrukturen mit neuen Technologien zu renovieren und zu modernisieren. Um aber die bestehenden Altsysteme mit ihren vielfältigen Leistungen und komplexen Strukturen abzulösen, bedarf es eines recht hohen Aufwands, was gleichzeitig lange Entwicklungszeiten zur Folge hat mit der Gefahr, dass die neu erstellte Infrastruktur bei Fertigstellung bereits wieder nicht mehr den Anforderungen genügt.

Auf dieser Basis sind die verschiedenen Konzepte zur Enterprise Applikation Integration (EAI) entstanden, die als permanenter Entwicklungsprozess der Infrastruktur hin zu einer den Anforderungen des Business genügenden Flexibilität führen soll. Die Grundidee aller dieser Konzepte ist dabei eine Big Bang Einführung zu vermeiden, da solch ein Konzept verlangt, dass während der Umsetzungsphase keine neue Funktionalität für das Business realisiert werden kann, da man die Altsysteme praktisch einfrieren muss. Dies führt aber zu einer Situation, mit der das Business nicht leben kann, da sonst in der Zwischenzeit ein Konkurrent einen anvisierten Platz einnimmt und das eigene Unternehmen nur noch mit einer Me-too Strategie folgen kann.

Eine weitere Rahmenbedingung dieser Ansätze ist, dass die Entwicklungszeit neuer oder modifizierter Business-Services kurz ist, um die notwendige kurze Time-to-Market zu erreichen. Die Entwicklungsstrategie muss also so gestaltet werden, dass mit jeder neuen Business-Anforderung eine erneute Iteration eines kontinuierlichen Verbesserungsplans nachvollzogen wird. Dabei werden bei der ersten Iteration die Grundsteine der neuen Infrastruktur basierend auf dem aktuellen Kenntnis- und Technologiestand gelegt werden. Eine solche Vorgehensweise ermöglicht eine sukzessive bedarfsorientierte Dekommissionierung der bestehenden Altsysteme bei gleichzeitiger Flexibilität zur Bereitstellung neuer Unterstützungsfunktionen.

Gleichzeitig wird die Zielarchitektur so flexibel gestaltet, dass sowohl Standardpakete vom Markt wie auch Eigenentwicklungen zum Einsatz kommen können. Dabei geben Designregeln wie "Buy before Build" oder umgekehrt "Build before Buy" den Ausschlag, in welchem Masse Standardpakete eingesetzt werden. Aus der geforderten kurzen Time-to-Market folgt, dass Altsysteme, so sie die geforderten Geschäftsfunktionen bereitstellen können, integriert und mit Standardpaketen vom Markt ergänzt werden, und nur dort wo keine Standardpakete verfügbar sind, Eigenentwicklungen eingesetzt werden.

Architekturzentrierte Lösungsentwicklung

Die folgende Abbildung 1 zeigt die Zusammenhänge der verschiedenen Architekturbereiche.

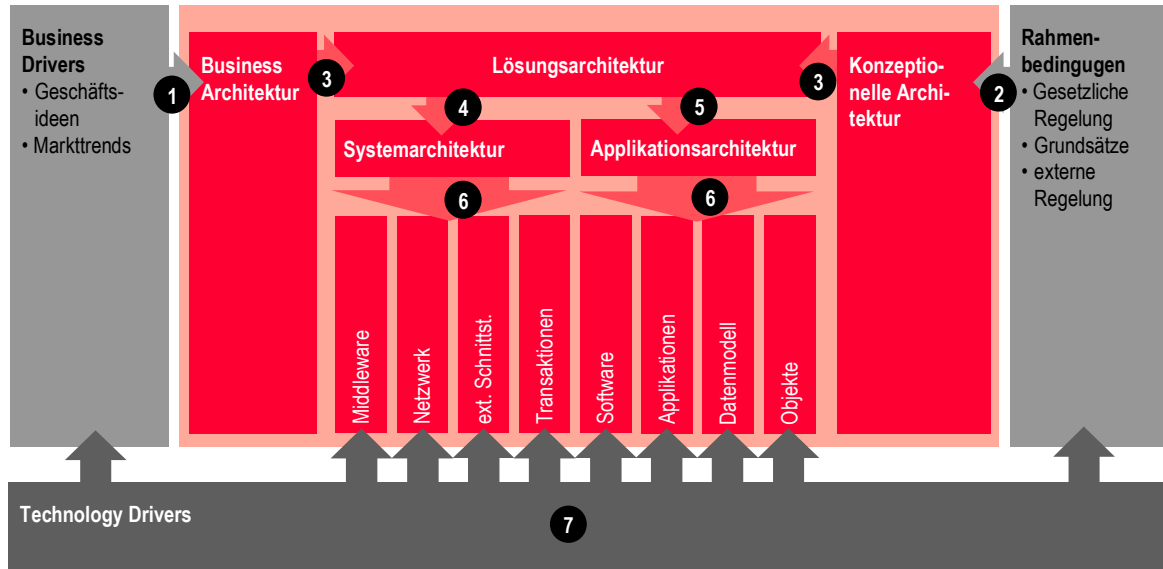


Abbildung 1 – Architekturen und Einflussfaktoren

- Die geschäftlichen Einflussfaktoren einerseits und die Rahmenbedingungen andererseits bilden die äusseren Voraussetzungen für den Entwurf einer Lösungsarchitektur. Geschäftsideen oder Markttrends fliessen in Form der daraus entstehenden Business-Anforderungen in die Business-Architektur ein ①.
- Die Business-Anforderungen stellen die funktionalen Anforderungen dar. Aus den Rahmenbedingungen andererseits leiten sich Grundregeln und Verfahren für die Lösungsentwicklung her, die in der Konzeptionellen Architektur zusammengefasst sind ②.
- Die Konzeptionelle Architektur ist also die Formulierung der nicht-funktionalen Anforderungen. Business-Architektur und Konzeptionelle Architektur beeinflussen beide die Lösungsarchitektur ③.
- Aus der Lösungsarchitektur werden die Systemarchitektur ④ (technisch und betrieblich orientiert) und die Applikationsarchitektur ⑤ (Abbildung der Geschäftsprozesse und -funktionen in Workflows, Prozessdefinitionen und applikatorische Funktionen) entwickelt.
- Durch weitere Detaillierung entstehen daraus die einzelnen Domänenarchitekturen ⑥.
- Die technologischen Einflussfaktoren wirken zum einen direkt auf die Domänenarchitekturen beeinflussen aber auch die Rahmenbedingungen, die den neuen technischen Möglichkeiten angepasst werden müssen, und sollten auf ihre Tauglichkeit zur Ermöglichung neuer Businessfunktionen und -modelle untersucht und dann in die Definition neuer geschäftlicher Einflussfaktoren einfließen ⑦.

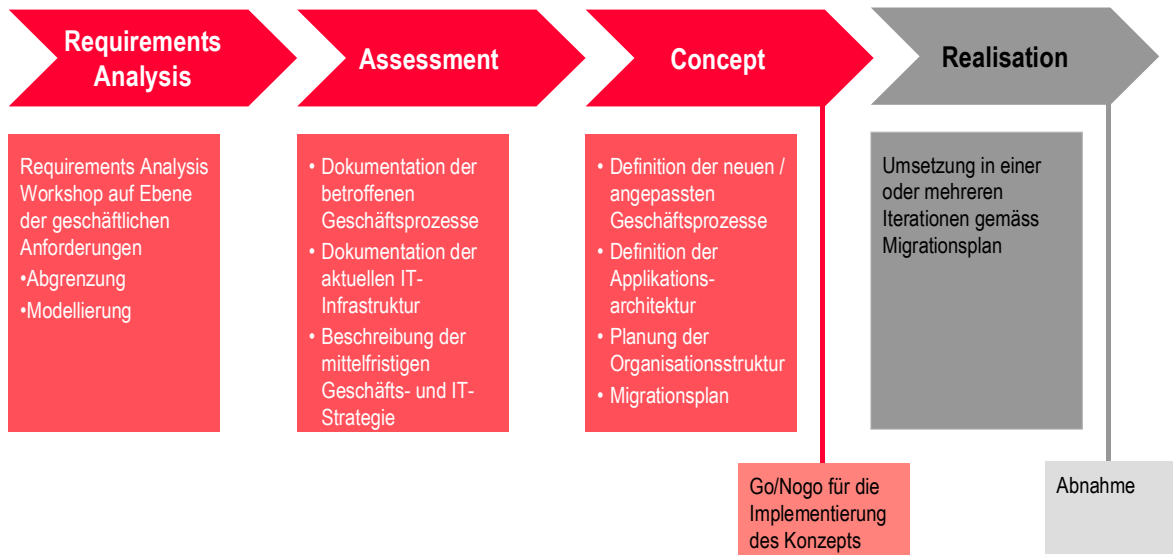


Abbildung 2 – Systor Prozessmodell für die Lösungsentwicklung

Der in Abbildung 2 dargestellte Prozess zeigt den Ablauf der Lösungsentwicklung gemäss der Vorüberlegungen in Abschnitt 0. Dieses Prozessmodell stellt das Grobraster des in Kapitel 0 dargestellten Prozesses dar. Der Vorteil dieses Vorgehens ist, dass die während des Prozesses entwickelten Architekturen oder Architekturteile unmittelbar für die Entwicklung neuer Geschäftslösungen nutzbar sind, während gleichzeitig die Architekturentwicklung iterativ weitergetrieben werden kann.

Prozess- und Komponentendesign

Ausgehend von den Überlegungen in den vorherigen Abschnitten ist eine prozessgesteuerte, komponentenbasierte Architektur, wie sie auch in den Kontakten zur RBG diskutiert wurde, sinnvoll.

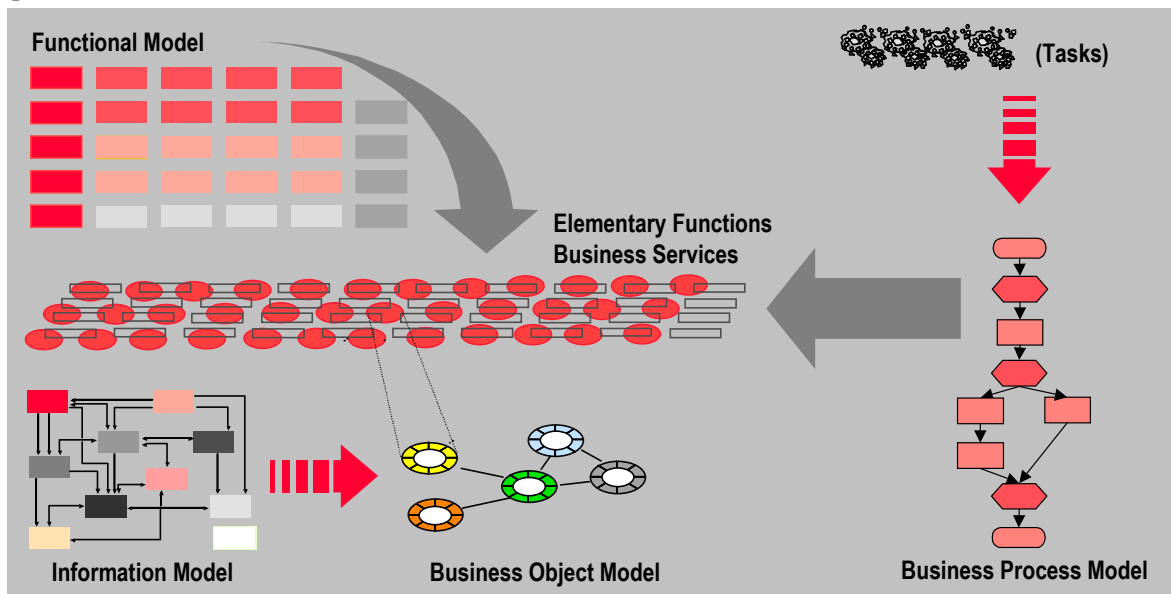


Abbildung 3 – Der Weg zur Anwendungsarchitektur

Abbildung 3 zeigt welche Einflussfaktoren über welche Wege die endgültige Anwendungsarchitektur bestimmen. Einerseits bestimmt das funktionale Modell des Unternehmens die erforderlichen Business Services und Elementarfunktionen. Andererseits liefert das Information Model die Kernentitäten (z.B. Partner, Konto, ...) des Geschäfts, auf denen die Business Services operieren. Die Tasks wiederum bestimmen die Geschäftsprozesse, die ihrerseits die Business Services gemäss der Prozessdefinitionen triggern. Diese drei Einflüsse werden in der Folge auch wichtig für die Bestimmung der Komponentengrenzen.

Zuvor sollen aber einige Begriffe zum besseren Verständnis geklärt werden.

- Ein Geschäftsprozess ist eine Abfolge von Aktivitäten und Entscheidungen, die den Ablauf eines Geschäftsfalls vollständig implementieren. Eine solche Abfolge wird auch als Workflow bezeichnet.
- Ein automatisierter Geschäftsprozess ist eine Prozessdefinition, die eine Folge von Aktivitäten innerhalb eines Geschäftsprozesses ohne Intervention von aussen durch eine Folge von Service-Aufrufen an eine oder mehrere Geschäftskomponenten steuert.
- Eine Business-Komponente ist ein in sich geschlossenes applikatorisches Element, dass über definierte Schnittstellen Services auf einer Kernentität bereitstellt.
- Eine Software-Komponente ist eine (wiederverwendbare) Einheit, die eine Elementarfunktion innerhalb einer oder im Fall der Wiederverwendbarkeit mehrerer Business-Komponente implementiert.

Der erste Schritt im Prozess- und Komponentendesign ist die Definition der Geschäftsprozesse auf der oberen Prozessebene (Workflows) und die Identifikation der in diesen Prozessen benötigten Geschäftsfunktionen. Diese werden entweder von Komponenten bereitgestellt oder können selbst wieder durch automatisierte Geschäftsprozesse abgebildet werden, die mehrere Komponenten involvieren.

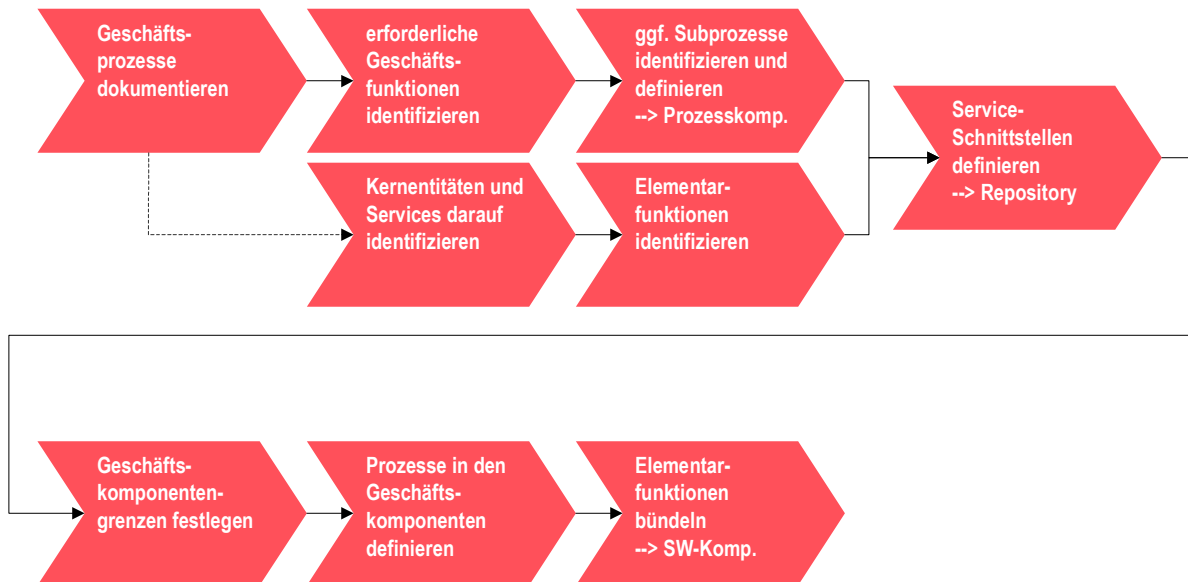


Abbildung 4 – Ablauf beim Prozess- und Komponentendesign

Der unterbrochene Pfeil von der Dokumentation der Geschäftsprozesse zur Identifikation der Kernentitäten bedeutet, dass die Analyse der Geschäftsprozesse es nahelegen kann, weitere nicht unmittelbar offensichtliche Entitäten aufzunehmen. Insbesondere im Bereich der Bewegungsdaten kommt dies gelegentlich vor.

Das Komponentendesign erfolgt typischerweise in mehreren Phasen. In der ersten Phase ist der Schwerpunkt die Definition der Geschäftskomponenten. Die Designgrundsätze hier sind Vollständigkeit der Funktionen bezüglich einer Kernentität, Kapselung dieser Entität und unabhängigkeit von anderen Kernentitäten. In einer zweiten Phase werden die enthaltenen Software-Komponenten betrachtet. Hier geht es um die interne Aufgliederung einer Geschäftskomponente in handhabbare, übersichtliche und wenn möglich wiederverwendbare Objekte. In dieser Phase werden auch generelle Service Komponenten wie zum Beispiel Logging, Audit-Trail, Output-Management und ähnliches betrachtet. Schliesslich gibt es noch die Phase des Packaging. Ziel ist es die Software-Komponenten für das Deployment in sinnvolle Pakete zu schnüren. Die Vorgaben kommen dabei zum Einen aus dem Geschäftskomponentendesign und zum anderen aus Überlegungen wie Performance versus Redundanzfreiheit (bei wiederverwendbaren Komponenten).

Systor's Lösungsdesignprozess [RAAC]

Übersicht

Die bisher dargestellten Überlegungen führten in der Systor zur Definition des Lösungsdesignprozesses [RAAC], der in wesentlichen Teilen die Erfahrungen der Architektur- und Consulting-Gruppe der Systor mit komplexen Projekten in der Finanzindustrie widerspiegelt. Ergänzt wurden und werden diese "Good Practices" mit Überlegungen zu verschiedenen Architektur-Frameworks und Blueprint-Techniken. [RAAC] selbst steht für Requirements Analysis, Assessment and Concept zum Einsatz. Dieser Prozess ist eingebettet in den Projekt Management Prozess [PMP] und stellt damit einen der Lieferprozesse dar.

Zur Vorbereitung dieses Prozesses dienen soweit vorhanden die folgenden Dokumentationen:

- Anforderung, Geschäftsziele und / oder Ideen des Kunden



- Beschreibung der aktuellen Geschäftstätigkeit
- die aktuelle Business Architektur resp. Beschreibung der Geschäftsprozesse
- die mittelfristige Business Strategie
- die aktuelle Systemarchitektur resp. Dokumentation der IT Infrastruktur
- IT Strategie

Dokumentation oder Teile davon, die nicht vorhanden oder unvollständig sind, werden während des Prozesses so detailliert wie erforderlich erarbeitet.

Ziel des Prozesses ist es einerseits die Brücke zwischen Business und IT zu schlagen und vor allem auf beiden Seiten ein gemeinsames Verständnis der Aufgabenstellungen zu fördern. Die Ergebnisse des Prozesses dienen dazu,

- den weiteren Fortgang der Lösungsentwicklung planen zu können
- ausreichende Informationen zur Spezifikation der Prozesse und Komponenten zu haben
- erforderliche organisatorische und infrastrukturelle Massnahmen zu initiieren
- die Gesamtarchitektur iterativ weiterzuentwickeln

Auf diesen Ergebnissen aufbauend können weitere Realisierungsprozesse aufbauen. In der Syster sind derzeit zwei weitere Prozesse definiert, die einerseits zur Prozessmodellierung und andererseits zur Komponentenentwicklung herangezogen werden können. der Prozess **[RAAC]** kann aber auch in eine bestehende Methodik-Landschaft eingefügt werden.

Eine detaillierte Beschreibung dieses Prozesses ist als Whitepaper bei den Autoren erhältlich.

Prozess- und Komponentenentwicklung

Das Prozess Re-Design hat zum Ziel, die wichtigsten im Grobkonzept erarbeiteten Top-Level-Prozesse im Detail derart zu gestalten, dass ein optimales Zusammenwirken der einzelnen Applikationen und Komponenten im Sinne eines Straight-through-processings ermöglicht wird. Dabei steht im Vordergrund, dass die Prozesse nicht einer radikalen und grundsätzlichen Veränderung unterzogen werden (d.h. kein Business Process Re-Engineering), sondern dass durch stellenweise Anpassungen der existierenden Prozesse und Erweiterung in Bereichen einer noch fehlenden aber möglichen Automatisierung ein maximaler Nutzen aus den Möglichkeiten des neuen Systems garantiert wird („Process follows IT“). Umgekehrt werden durch das Process Re-Design Anforderungen an die neu zu erstellenden Systemkomponenten erkannt und formuliert („IT follows Process“). Desweiteren fließen auch die Auswirkungen der neuen Verkaufskanäle wie z.B. das Internet auf die Prozesse mit in den Prozessentwurf ein.

Das Process Re-Design wird unterstützt durch die Syster Methodik Workflow Engineering^{SY}, welches ein pragmatisches und mehrfach erprobtes Vorgehensmodell, Notationen und Techniken für die Erstellung von Workflow Management Lösungen beinhaltet. Wesentliche Resultate sind:

- die Soll-Prozesse in ihrer graphischen Repräsentation („Aufgabenketten-Diagramme“)
- das Soll-Aufgabenverzeichnis
- das Rollen-Modell
- die in den Prozessen angesprochenen Geschäftsobjekte
- die Führungsgrößen pro Prozess

Die Frage, ob ein Prozess aus der Business Architektur modelliert werden soll wie auch die Frage nach der Reihenfolge, in welcher die einzelnen Prozesse modelliert werden sollen, kann erst nach Erstellung der Business Architektur im Grobkonzept geklärt werden.

Im Komponentendesign werden die identifizierten Business-komponenten in ihren Elementarfunktionen definiert. Aus den so gewonnenen Modulen werden Software-Komponenten zusammengestellt. Dabei richtet sich die Entscheidung, wie die Komponentenbildung erfolgt, nach den allgemeinen Richtlinien für das Lösungsdesign. Eine Grundsatz ist in der Regel die Reuse-Fähigkeit, aber auch andere Regeln können relevant sein, z.B. Redundanz versus Komponentenkommunikation zugunsten der Performance. Es kann auch vorkommen, dass für unterschiedliche Teilsysteme unterschiedliche Regeln zugrundegelegt werden müssen.

Service-Bus und Event-Bus Architektur

Lösungsansatz

Der von SYSTOR empfohlene Vorgehen geht davon aus, dass aktuell bei der AGR beziehungsweise den regionalen Rechenzentren unterschiedliche Lösungen implementiert sind. Diese müssen parallel zur Migrationsphase (auf die neue Architektur) den sich dauernd verändernden Kundenbedürfnissen angepasst werden. Somit muss eine graduelle, schrittweise Migration zur neuen Architektur ermöglicht werden, die die Investitionen in die bestehenden Lösungen wo sinnvoll übernehmen kann, die laufend an die sich verändernden Geschäftsziele adaptiert werden kann und die insbesondere die notwendige kontinuierliche Pflege bestehender Applikationen nicht gefährdet..

Der Vorgehensvorschlag der SYSTOR basiert auf einem Service- und einem Event-Bus, der das Kernstück der neuen Architektur darstellt. Die bestehenden geschäftstragenden Applikationen werden in diese Bus-Architektur "eingeklinkt" und können wo sinnvoll in der Übergangsphase weiterhin auch traditionell betrieben werden.

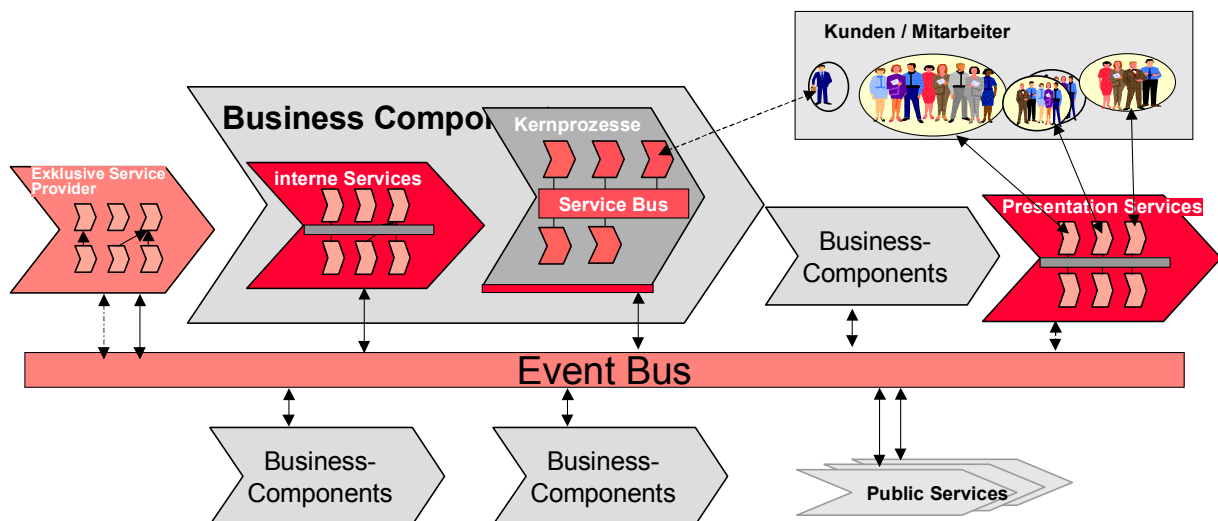


Abbildung 5 – Bus-Architektur einer entkoppelten Applikationslandschaft

Charakteristiken von Bus-Architekturen

Begriffe

Über den Event-Bus werden die langlaufenden, zeitlich entkoppelte Prozessketten sowie die Kommunikation mit externen und/oder nur asynchrone Anbindungen ermöglichen Services ermöglicht. Der Service-Bus verbindet Services die in einem zeitlich eng begrenzten Rahmen ablaufen und/oder eine enge Koppelung erfordern. Der Presentation-Layer steht allen Services zur Verfügung und übernimmt für alle Services das Rendering (die graphische Aufbereitung).

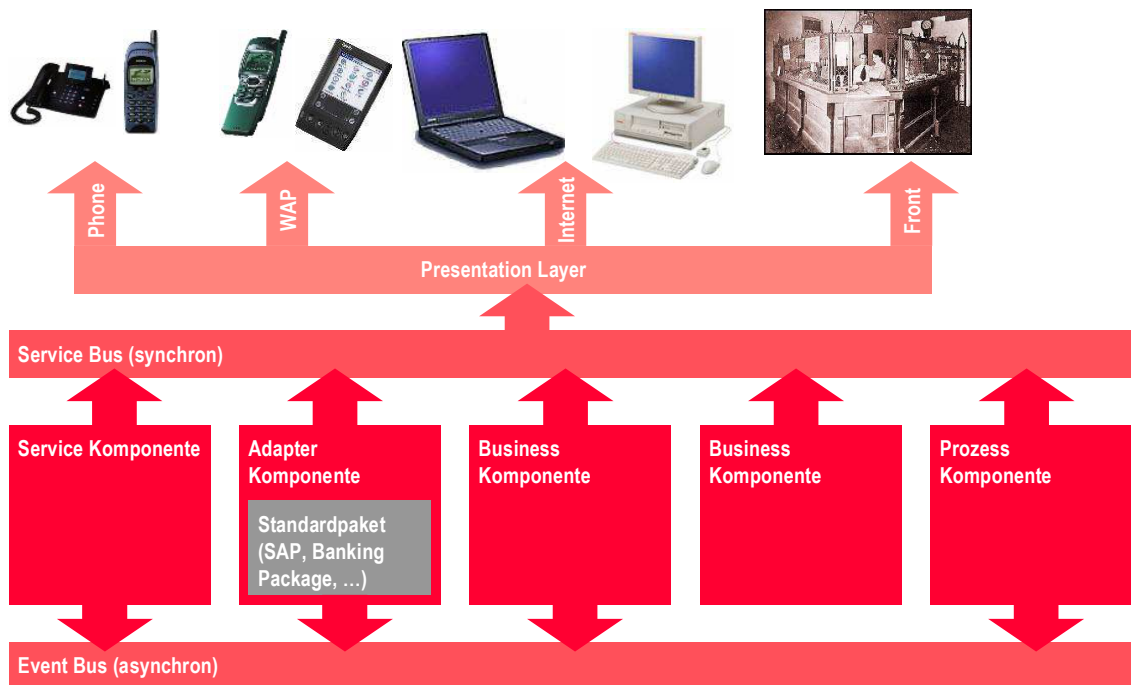
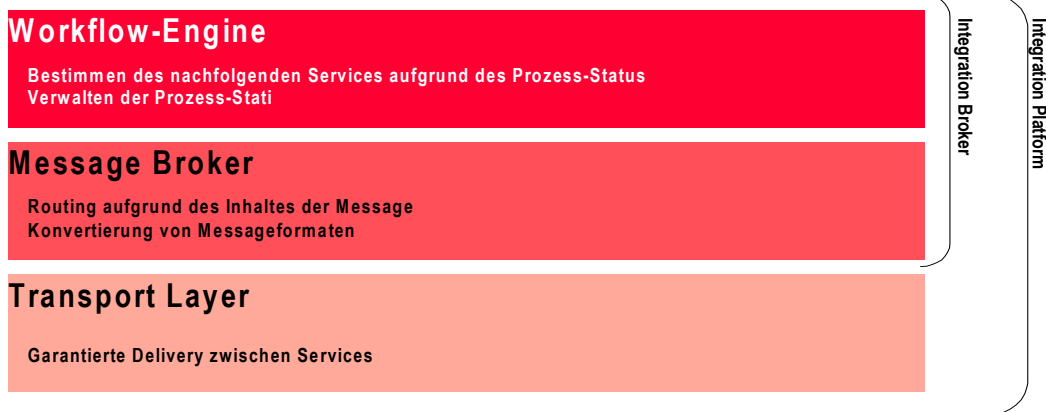


Abbildung 6 – Layer-Architektur einer entkoppelten Applikationslandschaft



Die Bus-Architekturen basieren auf vergleichbaren Konzept, jedoch mit unterschiedlicher Ausprägung und mit unterschiedlichen technischen Komponenten für die einzelnen Schichten/Layer.



Der Transport-Layer als Basisinfrastruktur ermöglicht die Kommunikation zwischen den einzelnen Applikationen und gegebenenfalls den weiteren Schichten. Er wird je nach Anforderungen asynchron oder synchron ausgelegt, mit entsprechenden Konsequenzen auf Funktionsumfang, aber auch auf die Komplexität der Systeme. Adapter für alle gängigen ERP-Systeme und Plattformen ermöglichen die Anbindung unterschiedlicher bestehender Applikationen und Technologien an den Transport-Layer. Sie sind ein wesentliches Kriterium für die Auswahl der optimalen Produkte und bei den grösseren Anbietern für ein breites Spektrum von Applikationen und Betriebssystemen verfügbar. Mögliche Transport-Layer sind u.a. MQSeries von IBM, Microsoft Message Queue (MSMQ), eLink von BEA.

Der Message Broker ermöglicht flexibles Routing der Messages im Gegensatz zu "hard" in den Applikationen oder Adaptern festgeschriebenes Routing. Dies erhöht die Flexibilität und Austauschbarkeit der Systemes und ermöglicht eine Trennung zwischen den an den Event- oder Service-Bus angeschlossenen Services und den damit abgebildeten Geschäftsprozessen. Dabei wird der nachfolgende Prozess-Schritt aufgrund vom im Message Broker abgelegten Regeln definiert. Typische Vertreter von Message Brokern sind MQS Integrator von IBM, Level 8, Mercator, NEON, Angebote von BEA und mehrere weitere Anbieter.

Für komplexe Geschäftsprozesse, die eine Prozess-Verfolgung (Überwachung) erfordern oder bei denen die Ergebnisse mehrerer Services gemeinsam vorhanden sein müssen um den nächsten Schritt auszulösen, werden Workflow-Systeme eingesetzt. Die Workflow-Engine übernimmt in diesen Fällen die Steuerung des gesamten Prozesses und kann auch komplexere oder zeitlich abhängige Prozessschritte steuern. Auch Prozess-Überwachung und -Audit- Funktionen können von einer Workflow-Engine wahrgenommen werden. Beispiele von Workflow-Systemen sind Staffware, MQ Workflow.

Event-Bus

Grundsätzlich sind viele Geschäftsabläufe in der Bank asynchron und entkoppelt, also idealerweise über einen Event-Bus abgebildet. Dies bedeutet aber gleichzeitig, dass near-realtime-processing und straight-through-processing erforderlich sind. Wenn ein Kunde einen Zahlungsauftrag online erteilt und sich anschliessend seinen Kontostand, in dem der Auftrag bereits verbucht ist, online ansieht, so erwartet er, dass die Verbuchung auch im Back-Office bereits erfolgt ist. Das heisst die Verarbeitungsdauer des Auftrags



muss kürzer sein als die durchschnittliche Reaktionszeit des Benutzers, der sich die Bestätigung der Auftragsannahme noch ansieht und dann die Anforderung zur Anzeige des Kontostands abschickt.

Solche Anforderungen sind mit der klassischen Batchverarbeitung nicht realisierbar. Andererseits ist eine streng synchrone Verarbeitung auch nicht sinnvoll, da eine Vielzahl von Geschäftsfunktionen im Hintergrund erforderlich sein können, um den vom Benutzer geforderten Request zu bearbeiten, von denen einige möglicherweise gar keine synchrone Schnittstelle bereitstellen. Ausserdem haben Untersuchungen des Benutzerverhaltens gezeigt, dass der Benutzer in der Regel nur wissen will, dass sein Auftrag angenommen wurde und garantiert und falls erforderlich innerhalb einer vorgegebenen Zeitspanne verarbeitet wird, so dass er zu einem späteren Zeitpunkt das Ergebnis sehen kann.

Der grosse Vorteil des Event-Buses liegt in der Tatsache, dass die einzelne Services lose gekoppelt sind und somit mit nur geringen Wechselwirkungen mit anderen laufenden Prozessen bestehen. Dies reduziert die Komplexität des Systems, macht es einfacher modularisierbar und erhöht gleichzeitig die Stabilität der einzelnen den Kunden/Mitarbeitern angebotenen Anwendungen und Services.

Service-Bus

Der Service-Bus als synchrone Integrationskomponente kommt dort zum Einsatz, wo Transaktionskonsistenz über mehrere Services unabdingbar ist und wo die erhöhten internen Abhängigkeiten vertretbar sind. Je nach eingesetzten Komponenten kann Transaktions-Konsistenz über mehrere Prozessschritte des Workflows ermöglicht werden. Primär entscheidend dafür sind die eingesetzten Business-Komponenten, welche insbesondere bei kommerzieller "Off the Shelf" Software (Packages) oft nur beschränkte Möglichkeiten besitzen.

Auch bei synchronen Prozessen muss auf Stufe Architektur entschieden werden, bis zu welchem Grad die übertragenen Messages in einem Meta-Format standardisiert und somit austauschbar definiert werden und inwieweit service-spezifische Messages übertragen werden. Ersteres erfordert mehr Aufwand in der Konzeption der Komponenten, letzteres erhöht die gegenseitigen Abhängigkeiten und somit die Wartungskosten.

SW-Bus innerhalb von Business-Komponenten

Auf standardisierten Messages basierende standardisierte Services sind Voraussetzung, damit Komponenten wiederverwendet werden und somit die entsprechenden Einsparungen in den Entwicklungskosten realisiert werden können. Die Systor setzt je nach Anforderungen auch innerhalb einer Business-Komponente das "Bus-Konzept" mit standardisierten Message-Formaten ein.

Das diesem Konzept zugrundeliegende Framework JWelder beinhaltet u.a. entsprechende Basis-Klassen, Konvertierungs-Services und Adapter-Objekte sowie die entsprechende Entwicklungsmethodik. Dies erlaubt die Integration standardisierter Services wie generisch definierte Logging- und Auditing Services, Authentizierungs-Services, Nummerkreis-Generatoren und anderer Services in neu zu entwickelnde Business-Komponenten und erfüllt so die Voraussetzung für die Wiederverwendbarkeit von Software-Komponenten.

Dasselbe Konzept kann auch auf der Ebene des vorgängig erwähnten Service-Buses realisiert werden, was ermöglicht, auch ganze Service-Komponenten in einer standardisierten Weise wiederverwendbar zu konzipieren.

Kriterien

Der Einsatz einer Bus-Architektur ermöglicht eine flexible Adaption an neue Geschäftsprozesse wie schon erwähnt. Durch das mehrfache Nutzen der in den Bus eingebundenen Services und durch die Möglichkeit, bestehenden Applikation in einer Übergangsphase auch traditionell zu betreiben, werden auch Synergiepotentiale ausgeschöpft und bestehende Investitionen geschützt. Die Auftrennung der Prozesse ermöglicht ein Mandantenkonzept, unabhängig von der Organisationsform, ebenso wird eine dezentrale/föderalistische Systementwicklung ermöglicht sollte sich dies als angebracht erweisen. Die Trennung der Services von der Präsentation erlaubt sowohl das Nutzen aller Vertriebskanäle, sei dies nun ein Filialnetz, via Internet oder an mobile Kunden als auch eine Internationalisierung der Darstellung an unterschiedliche kulturelle Anforderungen.

Cross Border Integration (B2B) und Collaborative Commerce

Das vorgestellte Konzept lässt sich unter Verwendung entsprechender Transportmechanismen und Datenaustausch-Standards direkt auch auf den B2B Bereich ausdehnen. Um dies zu erreichen wird ein B2B Transportmedium, z.B. ein XML-basierter Message-Router, eingesetzt, der die Event-Übermittlung über die Firmengrenze hinaus mit verschlüsselten XML-Nachrichten ermöglicht. Darüber hinaus lassen sich auch alle bekannten Kommunikationssysteme und Protokolle wie SWIFT oder EDIFACT anbinden.

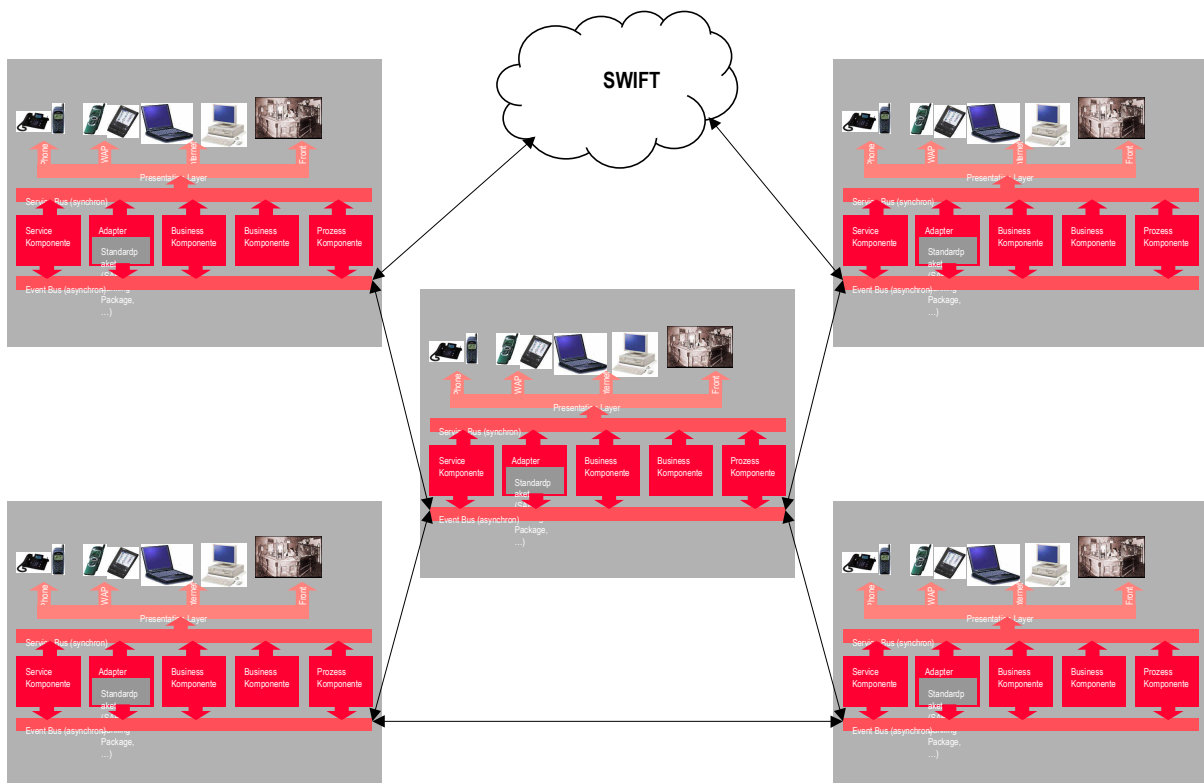


Abbildung 7 – Erweiterung zur B2B-Integration



Wie in Abbildung 7 angedeutet kann die B2B-Integration Point-to-point erfolgen, es ist aber auch möglich dass einer der Partner als Hub fungiert.

Die Bus-Architektur erlaubt neben dem Austausch von Informationen zwischen den Partnern auch das Collaborative Commerce. Denkbar wäre zum Beispiel das Outsourcing des Output Management oder der Abwicklung der Wertschriften-Lieferungen an einen Service Provider, der diese Aufgaben zentral für alle Partner übernimmt.

Europäischer IT Generalunternehmer

Systor ist der leistungsfähige europäische IT-Generalunternehmer. An zehn Standorten – in der Schweiz, in Deutschland, in Großbritannien und in den USA – bietet Systor branchenübergreifende Lösungen und Services aus einer Hand. Besonderen Schwerpunkt bilden das e-Business und die Finanzdienstleistungsindustrie. Vom Business Consulting über Integration und Implementierung bis hin zum funktionalen Outsourcing deckt Systor die Wertschöpfungskette von IT Services vollständig ab.

Contact

Deutschland:
SYSTOR GmbH & Co. KG
Hermann-Heinrich-Gossen-Straße 3
D-50858 Köln
E-Mail: info@systor.com
Tel.: +49 2234 108-0
Fax: +49 2234 108-130

Schweiz:
SYSTOR AG
Baslerstrasse 60
CH-8048 Zürich
E-Mail: info@systor.com
Tel.: +41 1 405 3111
Fax: +41 1 405 3113

Autoreninformation

Ulrich Moser

Solution Architect
Corporate Consulting
Systor AG

Baslerstrasse 60
CH-8048 Zürich
Telefon +41 1 405 3556
Fax +41 1 405 3113
Mobile +41 79 311 2051
ulrich.moser@systor.com
www.systor.com

Bruno Freitag

Senior Solution Architect
Corporate Consulting
Systor AG

Peter Merian-Strasse 84
CH-4002 Basel
Telefon +41 61 228 5713
Fax +41 61 228 5113
Mobile +41 79 207 8682
bruno.freitag@systor.com
www.systor.com