# Frontend Services, der nächste Service Layer

Carol Gutzeit

msg Applied Technology Research
msg systems AG
Robert-Bürkle-Strasse 1
85737 Ismaning
carol.gutzeit@msg-systems.com

**Abstract:** Die Basis einer Service-orientierten Architektur sind Stand heute die Backendservices, die als Ressource des Unternehmens auf dem Business Layer zur Verfügung stehen. Sie bilden die Basis für Prozess-orientierte Lösungen, Serviceorchestrierungen oder einfach die Verwendung der Services durch Applikationen. Diese Services sind meist ein Teil von Softwareprodukten und Anwendungen, die ebenfalls zugehörige Benutzeroberflächen bereitstellen. Die Wiederverwendung dieser Services auf Oberflächenebene (Presentation Layer) würde einen deutlichen Mehrwert im Gegensatz zur Verwendung der Backendservices auf Geschäftsebene (Business Layer) bieten. Das Paradigma hierzu wären Frontend Services, die dies ermöglichen.

## 1 Backendservices

Im Rahmen diese Dokumentes wird der Begriff Backendservice wie folgt definiert: Es handelt sich um die Realisierung einer standardisierten Service Schnittstelle über die eine fachliche Funktionalität eines Systems oder einer Komponente bereitgestellt wird. Die Ausführung der Servicefunktionalität erfolgt vollständig automatisiert durch Systeme ohne eine Benutzerinteraktion (Abbildung 1 Nr.3).

Die Kopplung von Systemen innerhalb einer Service-orientierten Architektur zur Unterstützung der geschäftlichen Anforderungen, baut meist auf Backendservices auf. Sie stellen somit wiederverwendbare Geschäftsfunktionalität bereit. Auf der Basis dieser Schicht, bestehend aus einer Vielzahl von Services, sollen neue prozessgesteuerte und hochwertigere Anwendungen (Abbildung 1 Nr. 1) entstehen. Was hier jedoch verloren geht sind die bestehenden Oberflächen, die die Fachlichkeit für den Anwender nutzbar zur Verfügung stellen. Die Backendservices sind meist keine autark entwickelten Softwarekomponenten, sondern ein Teil von IT-Systemen oder IT-Produkten. Diese Systeme bieten meist eine eigene Benutzeroberfläche an. Die Services stammen meist aus der Schnittstelle zwischen Presentation und Business Layer des Systems. Nehmen wir zum Beispiel aus dem Versicherungsumfeld ein IT-System für das Policenmanagement. Für die Erstellung eines Angebotes stellt es eine Oberfläche zur Verfügung, mit der die relevanten Daten abgefragt werden und im Anschluss Angebotsvarianten mit unterschiedlichen Leistungen und Preisen angeboten werden. In der Oberfläche wird intelligent gesteuert, dass bei Eingabe bestimmter Werte andere

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Felder ausgeblendet werden, bzw. Produkte oder Produktoptionen nicht zur Auswahl stehen. Bei der Auswahl einer KFZ Versicherung schränkt die PKW Marke zum Beispiel die Liste der Modelle ein. Einem Mann werden bei der Krankenzusatzversicherung keine Fragen zu Sonderleistungen während der Schwangerschaft gestellt, usw. Bei der realen Verwendung dieser Services, vor allem in Business Process Management Lösungen wird man feststellen, dass in dem Business Layer nur ein Bruchteil des Wertes des IT-Systems steckt, die den Service bereitstellt. In unserem Beispiel werden über die Services Operationen bereitgestellt, um Anträge zu speichern und zu laden, um eine Rate für einen Antrag zu berechnen und um die Produktdaten für die Auswahllisten zu laden. Die Oberflächen werden in einem BPM Ansatz meist in der BPM Suite realisiert und sind somit redundant vorhanden. In dem angeführten Beispiel ist für die Prozessrealisierung ein Verständnis der Produktvarianten und Abhängigkeiten der Felder untereinander notwendig. Bis auf die Berechnung der Raten wird kaum Geschäftsfunktionalität aus dem bestehenden IT-System über die Backendservices wiederverwendet.

## 2 Frontendservices

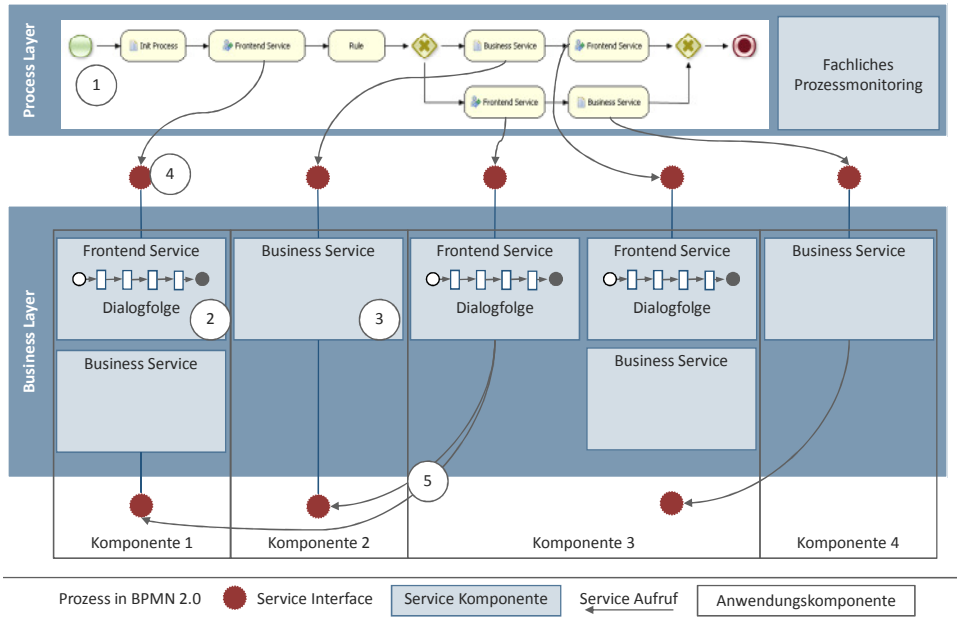


Abbildung 1 Service Kommunikation

Im Rahmen diese Dokumentes wird der Begriff Frontendservice wie folgt definiert: Es handelt sich um die Realisierung einer standardisierten Service Schnittstelle über die eine fachliche Funktionalität eines Systems oder einer Komponente bereitgestellt wird. Die Ausführung der Servicefunktionalität erfolgt in dem ein Benutzer Daten in

Eingabemasken erfasst oder Entscheidungen über die Oberfläche eingibt (z.B. einen Antrag genehmigen). Die Entscheidungen fällt der Benutzer basierend auf Daten, die ihm über die Oberfläche dargestellt werden (Abbildung 1 Nr.2).

Um die in Oberflächen abgebildete Funktionalität Service-orientiert und damit wiederverwendbar bereitzustellen, müssen Frontendservices ähnlich wie Backendservices den Designkriterien Service-orientierter Architektur folgen [PSD07]. Der Einstieg geschieht jedoch über eine Benutzeroberfläche und die Abarbeitung des Service benötigt menschliche Interaktion. In [QE08] wird vergleichbar von Bestands-, Funktions-, Prozess- und Interaktionsservices gesprochen. Ziel ist es die vorhanden Oberflächen so zu gestalten, dass sie die Ziele der Service Orientierung (Benefits of service-oriented computing [PSD07] unterstützen. Im Fokus stehen hier „Intrinsic Interoperability“ und „Increased Federation“.

### 2.1 Was ist bei dem Design von Frontendservices zu beachten

Bei der Standardisierung und Entwicklung von Frontendservices sind abweichend von Backendservices die folgenden Dinge zusätzlich zu klären und zu definieren. Diese Entscheidungen zielen darauf die Desigkriterien aus [PSD07] speziell für Frontendservices zu erfüllen

- (Kriterien: Loose Coupeling, Reusability, Standardized Service Contract, Discoverabilty) Es muss eine Festlegung der Schnittstelle und der Middleware erfolgen, die für die Koordination der Services verantwortlich ist (Frontend ESB / Portal). Übergreifende Standards zum Beispiel analog zur WebService Spezifikation für Backendservices existieren hier praktisch nicht, daher ist vor allem hier Spezifikationsbedarf vorhanden.

- (Kriterien: Reusability, Composability) Die Sicherstellung des einheitlichen Layouts muss definiert werden. Ein Frontendservice muss sich in unterschiedliche Kontexte einpassen können. Ein einheitliches oder steuerbares Design muss somit berücksichtigt werden. Dies könnte zum Beispiel durch unterschiedliche Cascading Style Sheets (CSS) und Kontextinformationen zur Laufzeit erreicht werden.

- Frontend Services benötigen wie auch die Backendservices Daten in Form von Informationsobjekten, die beim Start oder bei der Rückgabe übergeben werden (Abbildung 1 Nr.4). Auf dieser Ebene versucht man jedoch diese Daten auf identifizierende Merkmale zu reduzieren, um zum Beispiel in lang laufenden Prozessen Datenaktualität zu gewährleisten, um die Datenhoheit nicht zu unterwandern und bei Web basierter Kommunikation den Sicherheitsanforderungen Rechnung zu tragen. Dies resultiert in dem Pattern, dass für die Abarbeitung von Frontend Services der Datenaustausch über Backendservices hinweg erfolgt (Abbildung 1 Nr.5). Nehmen wir das Beispiel vom Anfang, dann steigt man in das Policensystem mit einem Partner ein, den man zum Beispiel zuvor über das CRM System ermittelt hat. Der Einstieg in den Frontend Service kann nun mit der Kundennummer, oder mit einer SessionID erfolgen. Über die SessionID kann an einer zentralen Stelle die Kundennummer ermittelt werden. Im Anschluss können mit

der Kundennummer über einen Backendservice die vollständigen Kundendaten aus dem CRM System ermittelt werden, die für einen Antrag notwendig sind.

### 2.2 Dialogsteuerung vs Prozesssteuerung

Für einen Benutzer resultiert die Kombination verschiedener Anwendungen auf Frontend Ebene in einem Maskenablauf. Im Detail kann man hier allerdings 2 Arten des Designs unterscheiden: Dialogsteuerung und Prozesssteuerung. (vergl. [IFDB11]) Die Dialogsteuerung basiert darauf, dass ein Ausgang aus einem Dialog oder Dialogablauf verbunden ist mit dem darauffolgenden Dialog oder Dialogablauf. Dies muss nicht statisch sein, sondern es kann auch durch eine zentrale und konfigurierbare Zentrale geschehen und kann abhängig vom jeweiligen Status des Dialogs sein. Es handelt sich aber an eine Aneinanderreihung von Dialogabläufen (Abbildung 2 unten). Bei der Verwendung durch eine Prozesssteuerung basiert die Kommunikation auf dem Request Response Pattern. Die Steuerung koordiniert über die Prozessdefinition (Abbildung 2 oben) die Serviceaufrufe. Auf diese Weise könnte man auch Frontendservices in einem komplexeren Frontendservice komponieren, was bei einer Dialogsteuerung nicht möglich ist.

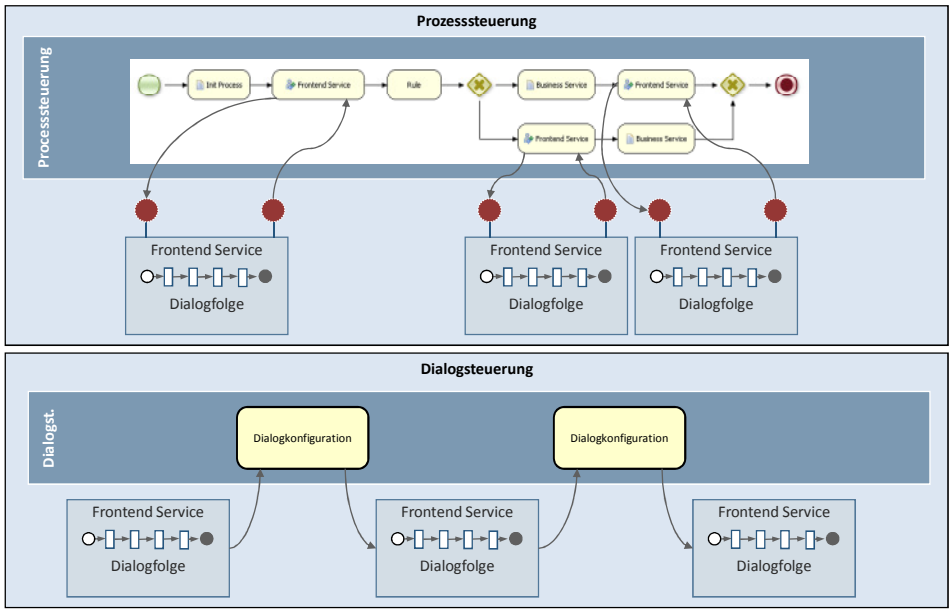


Abbildung 2: Dialogsteuerung vs. Prozesssteuerung

Im Serviceumfeld könnte man die Dialogsteuerung mit einer Event-basierten Architektur vergleichen. Der Erzeuger eines Event erwartet keine Antwort darauf, sondern nur, dass es korrekt verarbeitet wird. Ein Event (zum Beispiel „Versicherungsangebot anlegen“) führt dazu, dass die entsprechenden Masken dem Benutzer angezeigt werden. Beim

Auswählen des Buttons „Vertragsnehmer auswählen“ wird ein solches Event wiederum erzeugt, welches die Masken für die Suche einer Person öffnet. Im Anschluss daran wird das Event „Person in Angebot übernehmen versendet“ und die Angebotsbearbeitung wird fortgeführt.

Beide Architekturen sind mit Fronendservices realisierbar. Das Kommunikationspattern ist aber unterschiedlich und wirkt sich damit auf die technische Realisierung der Service Interfaces aus. Bei der Prozesssteuerung handelt es sich um ein typischer Weise asynchrones Request Reply Pattern, bei der Dialogsteuerung wird nach dem Beenden des Service ein neues „Event“ versendet mit dem aktuellen Kontext.

## 3 Frontend Services nutzen den vorhandenen IT-Bestand effektiver

Wie in Abschnitt 1 bereits erläutern gehen wichtige Funktionen des Presentation Layers mit Dialogabläufen, Suchfunktionen, Regeln für Kombination von möglichen Attributwerten und vieles mehr durch die ausschließliche Verwendung von Backendservices verloren. Sie müssen an anderer Stelle aufwändig erneut realisiert werden muss. Die Konfiguration eines Fahrzeuges zum Beispiel in der Automobilindustrie wird durch komplexe Konfiguratoren abgebildet. Benötige man in einem Bestellprozess eine Fahrzeugkonfiguration und versucht dies über Backend Services und einer eigenen Oberfläche zu realisieren wäre dies sicher der falsche Ansatz. Um den Konfigurator im Bestellprozess als Frontend Service einzubetten sind natürlich auch an ihm Anpassungen notwendig, wie sie Abschnitt 2.1 beschrieben sind. Im Anschluss kann der Konfigurationsservice in Prozesse oder andere IT-Systeme eingebunden werden. Er fügt sich in das Look and Feel des Kontextes ein und liefert als Ergebnis eine Fahrzeugkonfiguration, die über Backendservices abgefragt werden kann. Der Prozess oder das einbindende System kann nun damit weiterarbeiten. Der Aufwand zur Bereitstellung des Frontendservice ist vergleichbar mit der Einbindung eines Systems in eine Backendservice Integration. Die Wiederverwendung und der Mehrwert sind jedoch höher.

## 4 Einsatz bei msg Produkten und Kunden

Die msg Gruppe bietet Lösungen und Produkte an, die eigene und fremde Komponenten miteinander kombinieren. In einer großen Produktsuite handelt sich um einen Best of Breed Ansatz. Die Integration wird durch eine Prozesssteuerung koordiniert. Eine Variante solche Produkte anzubieten basiert auf einer Service-orientierten Architektur mit Backendservices. Diese Möglichkeit hat gezeigt, dass die Funktionalität, die von den Produkten und Anwendungen auf dieser Ebene bereitgestellt werden zu einem Großteil aus CRUD (Create, Read, Update, Delete) Operationen besteht. Wirkliche Businessfunktionalität, findet man nur selten. Dies liegt daran, dass die Businessschicht der Systeme nur die Funktionen bereitstellt, die für den Maskenwechsel benötigt werden. Dies bedeutet meist Entitäten speichern, laden oder einzelne Berechnungen. Das fachlich sinnvolle Ergebnis wird aber meistens nur durch eine Dialogfolge erreicht und dies

bildet die Businessschicht nicht vollständig ab. Ein Teil des Wissens bleibt im Controller und dem Maskendesign der Präsentationsschicht verborgen wie bereits vorher erläutert. Es handelt sich also nicht nur um die Frage der Granularität der Services. In der Realisierung auf Prozessebene stellt man fest, dass sehr viel Aufwand in die Nachimplementierung benötigt wird, um das gleiche Ergebnis zu erreichen, welches das System durch die Oberfläche bereits zur Verfügung stellt. Die Einführung der Frontendservices verschiebt und verringert nun diesen Aufwand. Die Anwendungen werden so erweitert, dass sie Frontendservice fähig werden, um dann zum Beispiel in Prozessen verwendet werden zu können. Hier spielen die Herausforderungen aus Abschnitt 2.1 eine wesentliche Rolle, die jedoch rein technischen Charakter haben und daher wächst der Aufwand nicht mit der fachlichen Komplexität einer Anwendung. Die Summe der Aufwende ist somit kleiner und kalkulierbarer als bei der Einführung einer Lösung auf Backendservices, wie das zuvor erwähnte Fahrzeugkonfigurator Beispiel verdeutlicht. Das wesentliche Kriterium für die Verwendung von Frontendservices bei der Einbindung eines Systems in einer prozessgesteuerte oder hoch integrativen Umgebung ist demnach die Komplexität, die über die Oberfläche bereitgestellt wird.

Es hat sich darüber hinaus in einer Reihe von Kundenbeispielen gezeigt, dass die Wiederverwendung auf Ebene von Frontend Services einen erheblichen Mehrwert bietet, da die Kompetenzen über Fachbereiche hinweg nicht vermischt werden. Ein Fachbereich der seine Anwendung mit der Funktionalität der Anwendung eines anderen Fachbereichs anreichern will, muss sich nur minimal damit auseinandersetzten. Darüber hinaus profitiert er von der Weiterentwicklung der Anwendung, die nicht in seinem Bereich liegt. Ebenfalls ist hier die erhöhte Wiederverwendung ein maßgeblicher Erfolg der erreicht werden kann.

## 5 Ausblick

Zentrale Dialogsteuerungen bewegen sich auf der Ebene von Portaltechnologien. Prozess Management Lösungen nehmen sich auch immer mehr diesem Thema an. Die fehlenden etablierten Standards auf dieser Ebene haben es jedoch bisher unmöglich gemacht auf eine standardisierte Lösung für Frontendservices aufzubauen. Es bleibt abzuwarten, welche dieser beiden Technologiezweige sich dem Thema annimmt und eine allgemeine Lösung vorschlägt. Die ersten Ergebnisse bezüglich WS-HT (WebService HumanTask) könnten eventuell einen Basis darstellen, aber sie müssen sich erst beweisen.

## Literaturverzeichnis

[PSD07] Thomas Erl: SOA Principles of Service Design (Prentice Hall Service-Oriented Computing Series from Thomas ERL); Prentice Hall International; Auflage: 1; 18. Juli 2007

[QE08] G. Engels, A. Hess, B. Humm, O. Juwig, M. Lohmann, J.-P. Richter, M. Voß, J. Willkomm: Quasar Enterprise; dpunkt.verlag; 2008

[IFDB11] O. F. Nandico: Integration für den Benutzer; OBJEKTSpektrum Ausgabe 02/2011