

# Fallbeispiel: Kundenauftragsmanagement als Ausgangspunkt für eine SOA-Einführung eines Logistikdienstleisters

Mario Thalheim

sd&m AG  
Wanheimer Straße 68  
40468 Düsseldorf  
mario.thalheim@sdm.de

**Abstract:** Mit einer serviceorientierten Architektur kann ein Logistikdienstleister schnell und flexibel auf neue Marktanforderungen reagieren, da er wiederverwendbare Komponenten schafft und redundante Implementierungen vermeidet. Basis für ein SOA-Vorhaben ist der Umbau eines fachlich abgegrenzten Teils der Anwendungslandschaft, der dann als Keimzelle für eine unternehmensweite SOA-Einführung dienen kann.

## 1 Fachliche Zielsetzung

Unternehmen im schnell wachsenden Logistiksektor müssen flexibel und schnell auf neue Kundenanforderungen reagieren. Für große und mittelgroße Logistikdienstleister reicht es keinesfalls mehr, reine Basisdienstleistungen wie Transport, Umschlag und Lagerung anzubieten. Die Kunden wünschen komplette, auf sie zugeschnittene Lösungsangebote. So kann ein Logistikdienstleister Vorteile im Markt erringen, wenn er seinen Kunden den Mehrwert bietet, Verantwortung zu übernehmen für das Management von *Purchase Orders* auf der Beschaffungsseite des Kundenunternehmens und *Sales Orders* auf der Distributionsseite.

Beschaffungsseitig kann das beispielsweise so aussehen: Der Kunde K hat einen Rahmenvertrag mit seinem Lieferanten L über die Herstellung von bestimmten Waren. Die Abwicklung der Beschaffung übernimmt dann der Logistikdienstleister LDL. Dieser stimmt selbst mit L ab, wann welche Waren wo abzuholen sind, und transportiert die Waren bzw. lässt sie transportieren. Sind *Purchase Orders* auf Teilstücken gemeinsam zu transportieren, so fasst er sie zu Sendungen zusammen. Er informiert frühzeitig den Kunden über die Einhaltung des Warenflusses und übermittelt ihm regelmäßig Reports über die Gesamtperformance der Logistikkette einschließlich der Lieferfähigkeit von L.

Auf der Distributionsseite kann LDL seinen Kunden anbieten, *Sales Orders* bis zum Lager dessen Kunden abzuwickeln – und sogar diese Läger selbst zu führen (im Sinne eines „Supplier Managed Inventory“).

Um diese Fachlichkeit umzusetzen, wird eine Lösung namens „Purchase Ordner und Sales Ordner System“ (POSOS) geschaffen.

## 2 Herausforderungen

Derzeit sind die Anwendungslandschaften bei vielen Logistikdienstleistern sehr heterogen und komplex, sind durch ein Nebeneinander nach Unternehmensfusionen geprägt und können mit neuen Anforderungen kaum mehr Schritt halten. Typisch sind folgende Eigenschaften:

- Je nach Transportmodus (Land, Luft, See) existieren unterschiedliche Anwendungen.
- Jede regionale Einheit ist selbst für „ihre“ Anwendungslandschaft verantwortlich. So existieren bei großen Logistikdienstleistern dutzende Transportmanagementsysteme und Lagerverwaltungssysteme. Jedes dieser Systeme mag aufgrund seiner Funktionen für die entsprechende Region am besten geeignet sein, aber ein Großteil der Funktionalität besteht mehrfach. Wer ein neues, weltweit einzusetzendes System schaffen möchte, muss zunächst alle Regionaleinheiten befragen, mit welchen Schnittstellen das neue System denn kommunizieren darf.

Zielzustand für LDL ist nun, dass

- POSOS *in sich* eine flexible Lösung darstellt, mit der für jeden Kunden ein entsprechendes Szenario zusammengestellt werden kann und
- die Anwendungslandschaft des LDL *in ihrer Gesamtheit* flexibel wird, damit POSOS und weitere Lösungen leicht integriert werden können.

Um wirklich flexibel zu werden und nicht nur ein weiteres System neben die anderen zu stellen, bietet sich an, die Anwendungslandschaft klar zu strukturieren und nach den Prinzipien der serviceorientierten Architektur (SOA) zu gestalten. Dadurch wird die IT wieder flexibler und kann schneller und kostengünstiger auf die Anforderungen aus dem Geschäft reagieren (siehe auch [HK04]).

Der Weg ist dabei vorgezeichnet: Zunächst wird POSOS selbst nach SOA-Prinzipien konstruiert und über Adapter in die bestehende Landschaft eingegliedert. In einem zweiten Schritt dient POSOS dann als „Keimzelle“ für eine serviceorientierte Architektur der gesamten Anwendungslandschaft.

### 3 Quasar Enterprise – ein konkreter Ansatz für SOA

Für die Umsetzung einer SOA übernehmen wir den Ansatz „Quasar Enterprise“, siehe [VHH06] und [HHV06]. Kurz zusammengefasst:

- Aus der Geschäftsarchitektur und den Geschäftsprozessen leiten wir *Domänen* für die *Anwendungslandschaft* ab. Wir trennen dabei bewusst zwischen der Geschäftsarchitektur, der damit verbundenen Geschäftsstrategie und der Unternehmensorganisation auf der einen Seite und der Struktur der IT-Anwendungslandschaft auf der anderen Seite. Letztere ist dann wiederum die Basis für die technische Systemlandschaft.
- Die Domänen enthalten (Anwendungslandschafts-) *Komponenten* der folgenden *Kategorien*: *Bestandskomponenten* dienen vorwiegend zur Datenhaltung, können aber sehr wohl eine eigene Benutzerschnittstelle für die Datenpflege haben. Nach außen bieten sie elementare Services für die Benutzung ihrer Daten. *Funktionskomponenten* sind Lieferanten für Services, die die Geschäftsfunktionen des Unternehmens unterstützen. *Prozesskomponenten* steuern Abläufe gemäß den Geschäftsregeln. In ihnen werden die Services der unterliegenden Komponenten zu Prozess-Services orchestriert. *Interaktionskomponenten* ermöglichen nun den Anwendern den Zugriff auf die Services einer Anwendungslandschaft.
- Die Komponenten kommunizieren miteinander vorzugsweise über *Service-Schnittstellen* auf Basis einer technischen Service-Infrastruktur. In realen Anwendungslandschaften kann es aber durchaus sinnvoll sein, an bestimmten Stellen andere *Integrationspunkte* zu benutzen: Zur Stammdatenreplikation wird man eher Werkzeuge nutzen, die an *Datenschnittstellen* ansetzen; Portale werden *Präsentationsschnittstellen* der Anwendungen nutzen.

Die Quasar-Enterprise-Prinzipien helfen somit dabei, in einer konkreten Anwendungslandschaft Services zu Komponenten zu gruppieren: Die Komponenten gehören immer zu je einer fachlichen Domäne und zu einer Kategorie. SOA laut Quasar Enterprise bedeutet darüber hinaus, dass fachliche Services nicht notwendigerweise durch technische Services über die SOA-Infrastruktur (also zum Beispiel Web-Services) implementiert sein müssen – auf technischer Ebene dürfen sehr wohl andere Integrationsmechanismen und Schnittstellen genutzt werden.

### 4 Eine SOA-Insel in einem Nicht-SOA-Umfeld

Wie bereits erwähnt, wird zunächst POSOS nach SOA-Prinzipien aufgebaut und in ein Nicht-SOA-Umfeld integriert. Siehe dazu Abbildung 1:

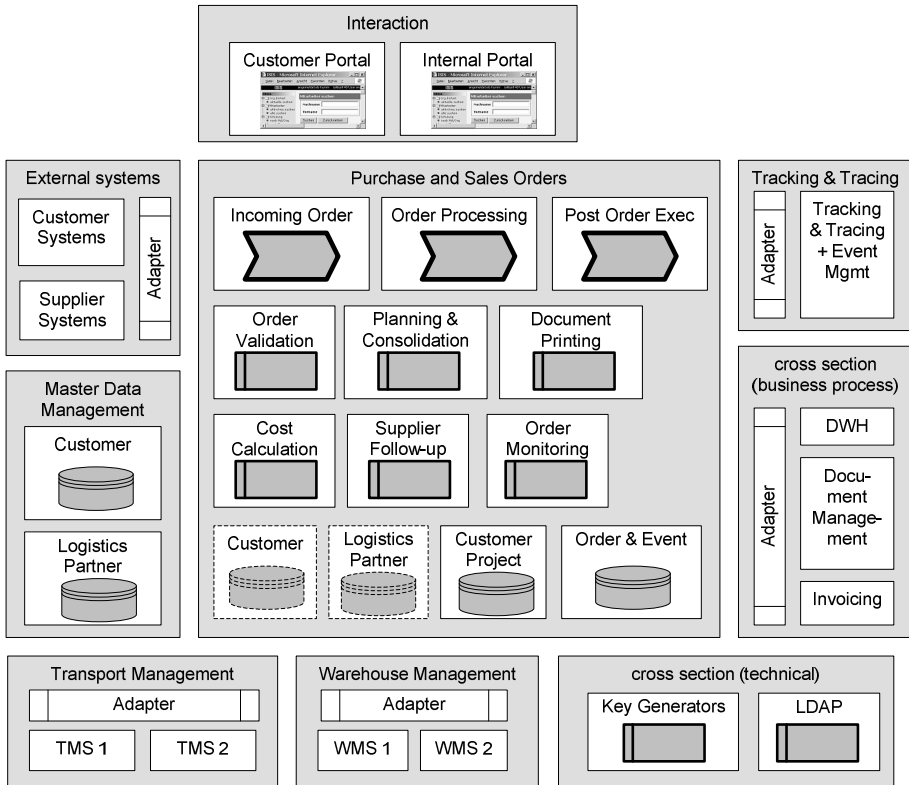


Abbildung 1: POSOS-Komponenten nach SOA-Prinzipien

POSOS besteht aus den Domänen „Interaction“ und „Purchase and Sales Orders“. Die Interaktionsdomäne enthält reine Interaktionskomponenten, die sich auf die unterliegenden Anwendungen abstützen. Die Domäne „Purchase and Sales Orders“ enthält Prozesskomponenten, die die Teilprozesse bei Auftragseingang sowie während und nach der Auftragsabwicklung steuern. Komplexe Funktionskomponenten kapseln Funktionalitäten wie Validierung, Planung und Kostenermittlung. Die Domäne verwaltet „eigene“ Daten wie das Kundenprojekt und alle auftragsbezogenen Daten in entsprechenden Bestandskomponenten, hält aber auch Kopien der allgemeinen Kunden- und Partnerdatenbank<sup>1</sup>. Diese Datenreplikation geschieht auf technischer Ebene über Datenschnittstellen. Die anderen Komponenten kommunizieren miteinander über Serviceschnittstellen. Dies gilt auch für bestehende technische Querschnittsdienste wie Schlüsselgeneratoren und die Benutzerverwaltung im LDAP. Andere bestehende Komponenten müssen hingegen erst über Adapter an die SOA-Infrastruktur angeschlossen werden. Diese Adapter sorgen im Falle der vielen Transport- und Lagersysteme auch für eine fachliche Vereinheitlichung.

<sup>1</sup> Hauptgrund für die Replikation der Stammdaten ist, dass so Performanceengpässe bei häufigem Lesen der Stammdaten über die SOA-Infrastruktur vermieden werden.

## 5 Nächste Schritte auf dem Weg zu einer serviceorientierten Anwendungslandschaft

POSOS selbst wird Keimzelle sein, um unternehmensweit die SOA-Prinzipien zu etablieren. In der POSOS-Umgebung wird es somit folgende Veränderungen geben:

- Die Interaktionskomponenten werden zu Portalen und erlauben den Zugriff auf verschiedenste Anwendungen, etwa POSOS, Transportmanagement, Dokumentenmanagement usw.
- Die internen Systeme werden nach SOA-Prinzipien umgebaut, somit entfallen die Adapter.
- Das Transportmanagement wird neu strukturiert: Statt der Gesamtlösungen TMS1, TMS2 usw. entstehen Einzelkomponenten, die Prozesse bedienen, Funktionen abbilden oder Daten halten. Für das Transport-Scheduling wird es sehr wohl unterschiedlich komplexe Komponenten geben, die zu den Anforderungen des jeweiligen Einsatzlandes passen – zum Beispiel ist in Ländern wie Deutschland ein komplexes Transportnetz zu betrachten, für Norwegen mit seiner länglichen Ausdehnung hingegen sind einfachere Algorithmen sinnvoller. Es wird jedoch zum Beispiel nur *eine* Prozesskomponente zur Verarbeitung von eingehenden Transportaufträgen existieren; für die Speicherung von Sendungsdaten wird *eine* Bestandskomponente ausreichen.
- Tracking und Tracing wird vom Event-Management getrennt. Zurzeit hat diese Komponente (siehe Abbildung 1) eine Doppelfunktion: Einerseits dient sie als Reporting-System für alle Fragestellungen des Tracking und Tracing. Andererseits ist diese Komponente aber Teil des operativen Ablaufs: POSOS beauftragt die Transportmanagementsysteme mit der Durchführung von Sendungen, erhält aber Informationen über die Ausführung nicht direkt, sondern als Ereignisse aus der Tracking- und Tracing-Datenbank.

Mit POSOS als SOA-konformer Lösung kann LDL bereits flexibel Kundenaufträge verwalten – die kundenspezifischen Unterschiede beschränken sich weitgehend auf die Bestandskomponente „Customer Project“ und Teile der Prozesskomponenten. Die Umstellung der gesamten Anwendungslandschaft nach SOA-Prinzipien im zweiten Schritt erlaubt es LDL dann, vergleichbare Zusatzdienstleistungen schneller in der IT zu verankern und somit schneller am Markt anbieten zu können. Außerdem senkt LDL Kosten für Wartung und Betrieb, weil – gerade in den Anwendungen für das Transportmanagement – Funktionalitäten nicht mehr redundant implementiert sind.

## Literaturverzeichnis

- [EI06] Eldracher, M.: Mit Service-orientierter Architektur zu mehr Flexibilität. In: Berater Guide 2006, Jahrbuch für Beratung und Management, H&T Verlag, Verlagsgruppe Handelsblatt, S. 50-55
- [HHV06] Hess, A.; Humm, B.; Voss, M.: Regeln für serviceorientierte Architekturen hoher Qualität. In: Informatik Spektrum , Ausgabe 6/2006
- [HK04] Heimann, T.; Kappes, R.: Mit SOA aus der Kostenfalle. In: IT-Management, Ausgabe 11/2004, S. 44-49
- [VHH06] Voss, M.; Hess, A.; Humm, B.: Towards a Framework for Large Scale Quality Architecture. In: Perspectives in Software Quality - Short Papers of the 2nd International Conference on the Quality of Software Architectures (QoSA)