

# Einbindung eines Buchungs- und Ticketingsystems in eine bestehende Anwendungslandschaft

Harald Lange

sd&m  
Lübecker Str. 1  
22087 Hamburg  
harald.lange@sdm.de

**Abstract:** Mit der Einführung eines Buchungs- und Ticketingsystems bei einem führenden Touristikunternehmen wurden mehrere operative Systeme durch ein neues System ersetzt. Das neue System ist in eine bestehende Anwendungslandschaft eingebunden und übernimmt damit die Verantwortung für alle existierenden Schnittstellen zu Nachbarsystemen. Eine EAI-Plattform schlägt die Brücke zwischen den Anforderungen der Nachbarsysteme und den Services des Buchungssystems.

## 1 Einleitung

Für ein führendes Touristikunternehmen hat sd&m ein neues Buchungs- und Ticketingsystem konzipiert, realisiert und eingeführt.

Ausgangspunkt war eine heterogene Anwendungslandschaft. Die operativen Abläufe waren auf verschiedenen Systemen verteilt, die z. T. von unterschiedlichen Partnern betrieben wurden. Die Systeme waren veraltet, unflexibel und zum Teil mit hohen Betriebs- und Wartungskosten behaftet.

Das neue Buchungssystem sollte kundenorientiert sein und eine hohe Flexibilität bzgl. der Produkt- und Tarifstrukturen aufweisen. Unter anderem sollten folgende Aufgaben von dem System übernommen werden:

- Verwaltung von Produkten und Preisen
- Buchung von Dienstleistungen
- Ausstellen und Einlösen von Tickets

## 2 Anwendungslandschaft

Abbildung 1 zeigt die Anwendungslandschaft vor Einführung des neuen Buchungssystems. Die vom Buchungssystem abgelösten Systeme sind hellblau markiert. Sämtliche Schnittstellen, die zu den verbleibenden Systemen führen, sind von dem neuen Buchungssystem zu bedienen. Dabei sind zwei Typen von Schnittstellen zu unterscheiden. Online Schnittstellen werden über eine direkte TCP/IP Verbindung mit dem Buchungssystem verbunden. Alle anderen Schnittstellen sind dateibasiert.

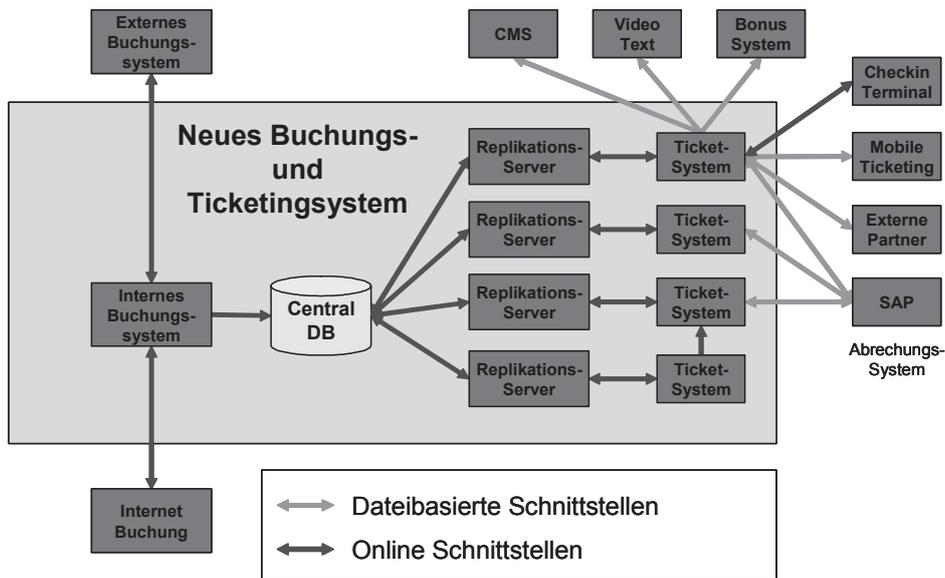


Abbildung 1: Änderung der Anwendungslandschaft mit Einführung des neuen Buchungs- und Ticketingsystems

## 3 Integration über EAI-Plattform

### 3.1 Architektur des Buchungssystems

Das Buchungssystem ist komplett in Java implementiert und besteht im Kern aus einem Zentral-Server und einer Vielzahl von GUI Clients, über die die Nutzer Zugang zum System erlangen.

Das eingeführte Buchungssystem ersetzt eine Reihe verschiedener Altsysteme und übernimmt gleichzeitig deren Aufgabe hinsichtlich bestehender Schnittstellen. Bei den zu versorgenden Nachbarsystemen handelt es sich unter anderem um Abrechnungssysteme, Auskunftssysteme und mobile Ticketverkaufssysteme. Es wurden insgesamt 19 Schnittstellen zu 13 Systemen umgesetzt. Alle dateibasierten Schnittstellen (siehe Abbildung 1) werden über eine EAI-Plattform bedient. Ausgenommen von der EAI Plattform sind die Online angebotenen Systeme, die über eine direkte TCP/IP Kommunikation mit dem Buchungssystem verfügen.

Die Konzeption der Schnittstellen, die über die EAI-Plattform realisiert wurden, lag folgenden Entwurfsentscheidungen zu Grunde:

- Das Buchungssystem enthält keine spezifischen Eigenschaften der Nachbarsysteme.
- Das Buchungssystem bietet einen einheitlichen Zugang. Dieser erlaubt den Aufruf von so genannten Services. Jeder Service Aufruf wird über eine eigenständige Transaktion abgehandelt. Alle Services sind zustandslos. Technisch verbirgt sich hinter einem Service der Aufruf einer Java Methode.
- Services werden sowohl von den GUI Clients als auch von sämtlichen Nachbarsystemen genutzt.

### **3.2 Umsetzung der EAI Schnittstellen**

Die Schnittstellen zu den externen Systemen sind in beiden Richtungen über den Austausch von Dateien realisiert.

Als Vermittler zwischen den externen Systemen und dem Buchungssystem kommt eine EAI-Plattform zum Einsatz, die die folgenden Aufgaben übernimmt:

- Überwachung eintreffender Dateien der angebundenen Systeme. Dateien aus den externen Systemen werden den Import-Schnittstellen zugeführt. Umgekehrt versorgen die Dateien aus dem Buchungssystem die Export-Schnittstellen.
- Einlesen der Dateien und syntaktische Transformation in die Terminologie des Buchungssystems (Import) bzw. der externen Systeme (Export). Bei diesen Transformationen handelt es sich im wesentlichen um Datenbank Lookups.
  - Im Falle von Import-Schnittstellen liefert das Mapping als Ergebnis den aufzurufenden Service mitsamt den benötigten Eingangsdaten. Beides wird mittels einer XML Datei an das Buchungssystem übergeben. Über den Service Aufruf hinaus enthält diese XML-Datei weitere Konfigurationsparameter.
  - Im Falle von Export-Schnittstellen liefert das Mapping als Ergebnis die Daten des externen Systems in dem gewünschten Format.

Als EAI-Plattform kam Microsoft BizTalk Server 2004 zum Einsatz. Die Verbindung zwischen den externen Systemen und dem Buchungssystem ist in Abbildung 2 dargestellt.

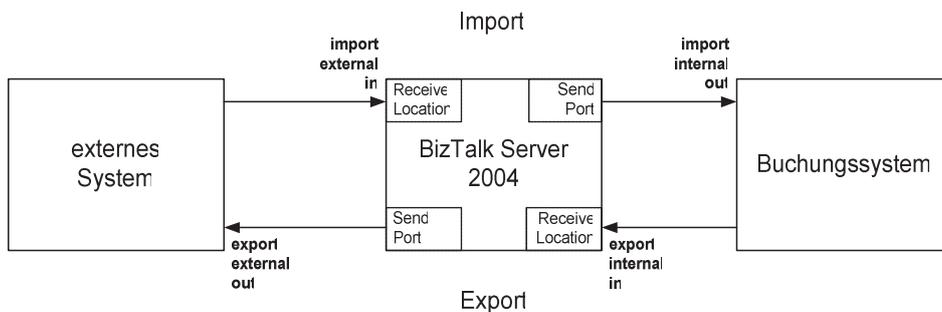


Abbildung 2: Kopplung der externen Systeme an das Buchungssystem via BizTalk

### 3.3 Schnittstellenadapter des Buchungssystems

Auf Seiten des Buchungssystems werden die von BizTalk gelieferten Service Anfragen in Form von XML-Dateien entgegengenommen und verarbeitet. Darüber hinaus werden auf Seiten des Buchungssystems Export-Schnittstellen zu gegebenen Zeiten angestoßen und deren Ergebnis an BizTalk weitergeleitet.

Beide Aufgaben werden von einem Schnittstellenadapter übernommen, der als Java Client-Prozess implementiert wurde.

#### Import-Schnittstellen

Für die Import-Schnittstellen werden Verzeichnisse auf einkommende Service Anforderungen überwacht. Jede gefundene Datei wird über einen XML Serialisierer eingelesen und liefert ein Java Objekt, welches den aufzurufenden Service, das zu übergebende Java-Objekt sowie weitere Konfigurationsparameter enthält. Über die Konfigurationsparameter wird festgelegt, wo mögliche Ergebnisdaten abgelegt werden sollen und was mit den Eingangsdaten im Falle einer erfolgreichen bzw. fehlerhaften Verarbeitung erfolgen soll.

#### Export-Schnittstellen

Sämtliche Export-Schnittstellen werden über eine zeitliche Steuerung angestoßen. Hierzu kommt eine Java API zum Einsatz, die die Funktionalität eines UNIX cron daemon anbietet. Die über diese zeitliche Steuerung gestarteten Java-Threads entnehmen die Parameter für den Service Aufruf aus einer Java Property Datei. Aus den Parametern wird ein Java Objekt wie beim Import erzeugt, so dass der weitere Ablauf für den Aufruf des Service identisch ist zum beschriebenen Import.

## 4 Zusammenfassung

Die Einbindung des Buchungssystems in die gegebene Anwendungslandschaft erfolgte, ohne dass die betreffenden Nachbarsysteme speziell im Buchungssystem berücksichtigt werden mussten. Dies wurde durch folgende Entwurfsentscheidungen ermöglicht:

- Das Buchungssystem stellt eine fachliche API in Form von zustandslosen Services bereit. Die Services stellen den einzigen Zugang zum Buchungssystem dar und werden sowohl von den GUI-Clients als auch für die Schnittstellen zu den Nachbarsystemen genutzt.
- Das Buchungssystem enthält keine spezifischen Eigenschaften der Nachbarsysteme. Diese werden stattdessen auf der EAI-Plattform in die Nomenklatur des Buchungssystems überführt.
- Für die Kopplung zwischen BizTalk und dem Buchungssystem wurde ein Schnittstellenadapter implementiert, der Service Anfragen von BizTalk in Form von XML-Dateien entgegen nimmt und für die Ausführung des entsprechenden Service sorgt. Eine technische Anbindung über Web Services hätte zusätzliche Aufwände auf Seiten des Buchungssystems mit sich gebracht.
- Die Service Aufrufe von BizTalk erfolgen asynchron per XML-Datei. Durch diese Abkopplung zwischen Service Aufrufer (BizTalk) und dem Buchungssystem ist das Gesamtsystem robuster. Damit können Service Aufrufe von BizTalk weiter eintreffen auch wenn z. B. der generische Client nicht läuft.
- Durch die Kopplung per Datei wird eine einfachere Nachvollziehbarkeit ermöglicht. Die verarbeiteten Dateien werden nach erfolgreicher Verarbeitung in ein Verzeichnis `processed` bzw. bei fehlerhafter Verarbeitung in ein Verzeichnis `error` verschoben.
- Die Wiederholung eines möglicherweise fehlgeschlagenen Service Aufrufs ist leicht zu bewerkstelligen, da lediglich die gegebene XML Datei erneut in das betreffende Verzeichnis kopiert werden muss.

## 5 Literaturverzeichnis

[BR02] Buhrmann P.; Richter, J.-P.: Integrationsprojekte auf Basis von EAI-Technologie – Architektur und Entwicklung, Software Management 2002, S. 143-154

[Ch03] Chen, X.: BizTalk Server 2002 – Design and Implementation, apress, 2003

[Sd05] sd&m AG: Enterprise Integration – Das Leistungsangebot von sd&m zu serviceorientierter Architektur EI/SOA, sd&m, 2005

[Si04] Siedersleben, J.: Moderne Softwarearchitektur, dpunkt Verlag, 2004