

# ESB: to Be or Not to Be?

Markus Lohn  
esentri AG  
Ettlingen

## Schlüsselworte

Oracle Fusion Middleware, Middleware, SOA, Integration

## Abstrakt

Wenn in Unternehmen das Thema SOA oder Integration auf der Tagesordnung steht, wird häufig über die Anschaffung einer Middleware und ESB im Speziellen nachgedacht. Die Folge ist dann, alle Aufgaben in diesem Umfeld unbedingt mit der Middleware oder ESB lösen zu wollen. Es wird nicht mehr hinterfragt, ob das im konkreten Fall sinnvoll und richtig ist. Eine Gesamtstrategie fehlt häufig. In diesem Vortrag werden mögliche Alternativen erläutert und eine mögliche Strategie aufgezeigt. Abgerundet wird der Vortrag durch einen Entscheidungsbaum, der bei neuen Integrationsvorgaben die entsprechende Implementierungsvariante selektiert. Der Entscheidungsbaum wurde in einem konkreten Kundenprojekt erstellt.

## Einleitung

Ohne eine systematische Organisation und ohne durchdachte Integrationsarchitektur werden Applikationen fast immer über Punkt-zu-Punkt Verbindungen integriert. Das bedeutet die beteiligten Applikationen tauschen direkt miteinander Daten aus. Mit einer Formel kann exakt ausgerechnet werden wie viele Schnittstellen maximal benötigt werden, wenn jede Applikation miteinander Daten austauschen muss.

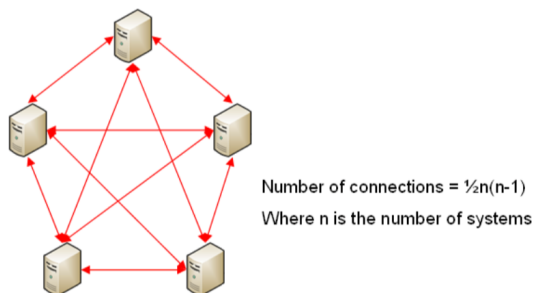


Abbildung 1 Punkt-zu-Punkt Integration

Die Integrationslogik ist direkt in den Applikationen enthalten und führt somit zu Steigerung der Komplexität bzgl. der Überwachung und Steuerung des Datenaustausches. Über die Jahre hinweg entsteht so eine Integrationsarchitektur die schwer weiter zu entwickeln und kontrollierbar ist. Auswirkungen von Änderungen sind nur schwer feststellbar und somit sind Änderungen mit ggf. erheblichen Zeitaufwand verbunden.

Unternehmen die eine Vielzahl von Punkt-zu-Punkt Verbindungen nutzen, denken häufig über den Einsatz einer Middleware mit ESB nach. Insbesondere Hersteller nehmen dieses Argument gerne auf, um auf die Vorteile ihres ESB-Produktes hinzuweisen. Man verfällt schnell dem Irrglauben, dass schon der Einsatz eines ESB alle Integrationsprobleme lösen kann. Ist ein ESB für jegliche Integrationsaufgabe das richtige Werkzeug?

Wenn ein Unternehmen sich für einen ESB entschieden hat, möchte man natürlich schnell von den Vorteilen einer solchen Plattform profitieren. Als eiserne Regel wird nun versucht, jegliche Integrationsaufgabe mit dem neuen Werkzeug umzusetzen. Häufig werden aber sinnvolle technische

Alternativen nicht betrachtet. Somit entstehen Lösungen die nicht sinnvoll mit einem ESB umgesetzt werden sollten. Die daraus entstandenen Probleme, werden folglich auf die schlechte ESB-Software zurückgeführt. Dies führt mitunter zu einer schlechten Akzeptanz einer ESB-Software. Folgen können dann von Vermeidungsstrategie bis Neuauswahl einer „besseren“ ESB-Software gehen. Das Grundproblem bleibt jedoch bestehen, wenn der ESB teilweise zweckentfremdet wurde.

Um eine sinnvolle Integrationsstrategie zu etablieren ist ein ESB nicht zwingend erforderlich. Er kann jedoch ein sinnvoller Bestandteil sein. Hinter dem ESB verbirgt sich kein allgemein gültiger Standard. Vielmehr existieren Architektur-Empfehlungen in Form von Blue Prints, Best Practices und Erfahrungen aus der Vergangenheit mit Integrationsarchitekturen, die festlegen, welche Aufgaben ein ESB erfüllen sollte. Das bietet natürlich Spielraum für Interpretation. Vor allem Hersteller bieten unterschiedliche Produkte und Ansätze für die Umsetzung eines ESB. Leicht fällt die Auswahl nicht, zumal Hersteller teilweise die Bezeichnung ESB für Produkte benutzen, die wesentlich mehr Funktionalität anbieten als eigentlich erforderlich.

Der ESB kann ein Baustein innerhalb einer Integrationsplattform sein und erfüllt somit definierte Verantwortlichkeiten. Wenn der ESB entsprechend dieser Vorgaben eingesetzt wird, steht einem erfolgreichen Betrieb nichts im Wege.

### **Aufgabe eines ESB**

Ein ESB hat zum Ziel Interoperabilität herzustellen bzw. zu sichern. Um das Ziel zu erreichen muss er drei wesentliche Aufgaben erfüllen:

#### **Konnektivität**

Die Konnektivität umfasst das Verbinden verschiedener Applikationen und Services. Applikationen sind mit unterschiedlichen Technologien implementiert und bieten vielfältige Schnittstellen für den Austausch von Informationen an. Beispiele für Schnittstellen sind APIs, Webservices und Dateitransfer etc. Der ESB muss sicherstellen, dass die Verbindung zwischen den Applikationen über die unterschiedlichen Technologien und Schnittstellen reibungslos funktioniert. Idealerweise bietet der ESB auch Möglichkeiten an, mit ungünstigen Eigenschaften der angebotenen Applikationen umzugehen. Als Beispiel kann die Reduzierung der Systemlast durch Throtteling genannt werden.

#### **Datentransformation**

Ein ESB muss das Datenformat zwischen den integrierten Anwendungssystemen transformieren können. Bei Bedarf müssen auch zusätzliche Informationen zu Nachrichten zugeordnet werden können.

#### **Routing**

In einem Integrationsszenario kann die Notwendigkeit bestehen eine Nachricht an mehrere Ziele weiterzuleiten. Aufgrund der definierten Regeln und Vermittlungslogik muss der ESB den richtigen Endpunkt für die Nachricht ermitteln und die Zustellung durchführen. Bei Bedarf muss der ESB auch die Möglichkeit unterstützen eine Übersetzung zwischen unterschiedlichen Transportprotokollen zwischen Applikationen vorzunehmen.

Neben den genannten Hauptaufgaben gibt es noch eine Reihe von Querschnittsthemen, die ein ESB auf jeden Fall erfüllen muss, um auch einen stabilen Betrieb zu ermöglichen:

#### **Sicherheit**

Applikationen tauschen geschäftskritische Informationen aus. Aus diesem Grund ist es enorm wichtig, dass der ESB die Authentifizierung, Autorisierung und Verschlüsselung eingehender und ausgehender Nachrichten gewährleistet.

## Betrieb

Als zentraler und geschäftskritischer Bestandteil einer IT-Infrastruktur müssen die Abläufe innerhalb des ESB überwacht und verwaltet werden können. Geeignete Werkzeuge müssen zur Verfügung gestellt werden.

## Management

Der ESB muss Funktionen für die Verwaltung von Service zur Verfügung stellen und insbesondere die Abhängigkeiten aufzeigen können.

Mit den genannten Aufgaben stellt ein ESB viele nützliche Funktionen zur Verfügung die in Integrationsszenarien benötigt werden. Viele Patterns aus dem Buch Enterprise Integration Patterns (Gregor Hohpe) werden durch einen ESB automatisch implementiert, z. B. Content-Based Router, Message Bus. Allerdings kann der ESB nicht alle Integrationsaufgaben zufriedenstellend lösen. Im Rahmen einer Integrationsstrategie wird ein Blue Print entwickelt. Der OSB kann hier einen wichtigen Bestandteil darstellen. Welche Verantwortlichkeit wird dem ESB zugedacht?

## Integrationsplattform

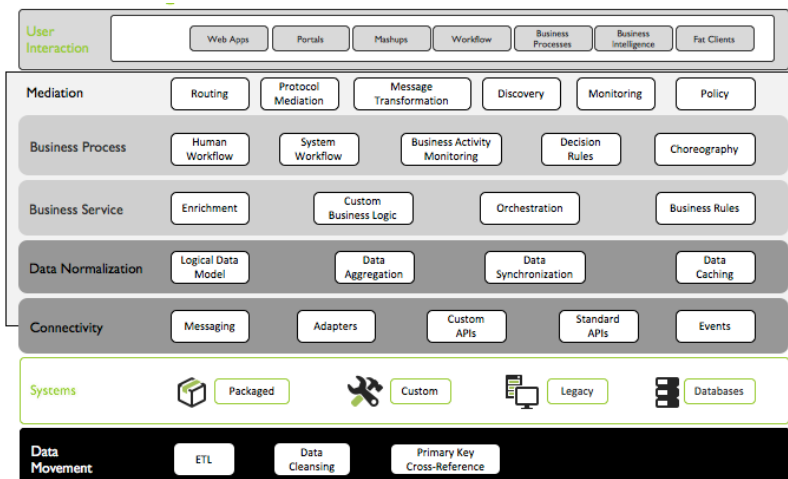


Abbildung 2 Blue Print einer Integrationsplattform von Oracle

Eine Integrationsplattform muss vielfältige Aufgabenstellungen erfüllen. Typischerweise wird eine Plattform in unterschiedliche logische Schichten mit verschiedenen Verantwortlichkeiten eingeteilt. Die einzelnen Aufgabenstellungen können dann mit unterschiedliche Technologien und Werkzeugen umgesetzt werden.

Eine Integrationsplattform muss Antworten auf folgende Themen liefern:

- Konnektivität zu EIS Systemen aufbauen
- Synchronisation von großen Datenmengen (Batch)
- Event Management und Zeitsteuerung (Jobs)
- Caching-Mechanismen
- Transformation
- Enrichment
- komplexe Logik mit Geschäftsprozessen
- Business Activity Monitoring

- Human Interaction
- Referenzdaten Management

Neben dem synchronen Austausch von Informationen zwischen Applikationen muss die Plattform auch geeignete Antworten für den Austausch von großen Datenmengen im Batch bereitstellen. Allzu häufig werden solche Aufgabenstellungen mit einem ESB umgesetzt. Probleme mit Timeouts, Transaktionskontrolle und Fehlerbehandlung sind dann vorprogrammiert. Für die Verarbeitung von Datenmengen im Batch gibt es bessere Werkzeuge als einen ESB. Natürlich kann im Vorfeld analysiert werden, ob eine Batchverarbeitung auch auf eine "real-time" Synchronisation umgestellt werden kann, die dann wiederum sehr einfach mit einem ESB implementiert werden kann.

Datenaustausch zwischen Applikationen können auch Teilschritte eines übergeordneten Geschäftsprozesses sein. Somit wäre die Integration über einen prozess-getriebenen Ansatz sinnvoll. Eine prozess-getriebene Integration kann besser mit einer Prozess- und Business Rule Engine umgesetzt werden, als mit einem ESB. Es gibt auch Hersteller, die solche Ansätze in ihr ESB-Produkt integriert haben. Auch dieser Aspekt ist in der Integrationsplattform zu berücksichtigen.

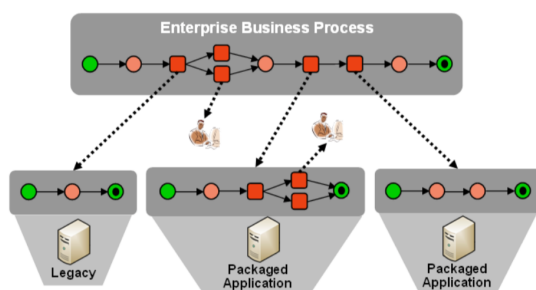


Abbildung 3 prozess-getriebene Integration

In der dargestellten Integrationsplattform ist der ESB dem Mediation Layer zugeordnet. Seine Aufgabe besteht darin die Kommunikation zwischen den einzelnen Komponenten in dieser Integrationsarchitektur über alle Schichten hinweg sowie zwischen den Applikationen, die sich mit der Plattform verbinden, sicherzustellen. Dabei spielt das Prinzip der losen Kopplung eine wichtige Rolle. Die genaue Lokation des Kommunikationspartners ist nicht bekannt und somit können Änderungen viel einfacher durchgeführt werden. In diesem Fall ist der Einsatz eines ESB sinnvoll.

### Implementierungsvarianten

Im Rahmen eines Kundenprojektes wurden verschiedene Implementierungsvarianten für Integrationen identifiziert. Diese Varianten werden nicht mit einem ESB umgesetzt und dienen später als mögliche Optionen im Entscheidungsbaum.

#### Manuell

Je nach Anforderung und unter Kostengesichtspunkten kann es sinnvoll sein eine Synchronisation von Daten manuell durch einen Benutzer durchführen zu lassen. Eine Automatisierung würde sich in diesem Fall nicht lohnen.

#### Vorgefertigte Schnittstelle durch Aktivierung und Konfiguration

In der Produktbeschreibung für eine der beteiligten Applikationen wird explizit eine Schnittstelle zur anderen Applikation angegeben. Für den erfolgreichen Betrieb der Schnittstelle ist dann nur noch eine Konfiguration erforderlich. Die Art der technischen Umsetzung dieser Schnittstelle spielt keine Rolle.

Als Beispiel kann die SAP Archivelink Schnittstelle genannt werden. Eine Programmierung ist nicht erforderlich.

### **Batch**

Während der Batchverarbeitung sind keine Eingriffe eines Benutzers möglich. Alle Daten, sowohl Nutzdaten als auch Steuerdaten, werden dem Programm vor Ausführung zugewiesen. Die Entscheidung über den Zeitpunkt der Verarbeitung liegt bei der IT. Üblicherweise werden im Rahmen der Batchverarbeitung große Datenmengen auf einmal verarbeitet, ohne die Verarbeitung zu unterbrechen. Insbesondere das Transaktionsverhalten ist zu berücksichtigen. Beispielsweise ist die Frage zu klären, ob die Transaktion abgeschlossen werden kann, wenn alle Datensätze verarbeitet wurden oder bereits nach jedem Datensatz abgeschlossen werden kann.

### **Punkt-zu-Punkt**

Wie eingangs erwähnt werden viele Punkt-zu-Punkt Verbindungen als Problem identifiziert. In Integrationsszenarien mit hohen Performanceanforderungen, hinsichtlich Antwortzeiten und Durchsatz, ist es sinnvoll die Nachrichten zwischen zwei Applikationen direkt auszutauschen. In diesem Fall kommt keine zusätzliche Middleware zum Einsatz. Der Austausch erfolgt durch direkte API-Aufrufe auf der Gegenseite.

### **Custom Development**

Bezeichnet einen Sonderfall bei der Integration von Applikationen. Zur Anwendung kommt diese Implementierungsart, wenn ein komplexes Mapping der Datenformate erfolgen muss, die mit Standardwerkzeugen nur schwer umsetzbar sind. Ein anderer Anwendungsfall sind besondere Formen der Fehlerbehandlung im vorliegenden Integrationsfall. Ferner ist viel (Business-)Logik in der Schnittstelle zu implementieren.

### **Prozess Engine**

Bei vielen Serviceaufrufen in einem Integrationsszenario oder einer komplexen Integrationslogik mit vielen Schleifen und Bedingungen ist der Einsatz einer Prozessengine sinnvoll. Insbesondere bietet eine Prozessengine folgende wichtige Funktionen:

- Nachvollziehbarkeit und Kontrolle über sog. Audit Trail (= Dokumentation des Prozessablaufs)
- Transaktionskontrolle innerhalb des Prozessablaufs
- Kompensation im Fehlerfall
- Einfache Implementierung von asynchronen Kommunikationspatterns

Aufgrund der Vielzahl der Möglichkeit ist es natürlich auch wichtig Architekten und Entwickler bei der Auswahl zu unterstützen. Insbesondere besteht die Anforderung, dass unterschiedliche Personen bei Anwendung der gleichen Kriterien zum gleichen Ergebnis kommen müssen.

### **Entscheidungsbaum**

Gemeinsam mit dem Kunden wurden Grundsätze und Prinzipien definiert. Beides diente zur Definition von Leitplanken und als Orientierung, wenn eine Entscheidung für eine Implementierungsvariante getroffen werden musste. Nachfolgend ein Beispiel für ein Prinzip:

Prinzip	Minimale Anpassung von Applikationen
Definition	Die massive Anpassung von Applikationen zum Zwecke der Integration muss unterlassen werden.
Folgerung	<ul style="list-style-type: none"> <li>• Applikationen implementieren keine Integrationslogik.</li> <li>• Applikationen müssen über Standardschnittstellen oder entsprechende Adapter integriert werden.</li> <li>• Integrationslogik wird ausschließlich in der Schnittstelle implementiert.</li> </ul>

Darüber hinaus wurden Entscheidungskriterien für verschiedenen Implementierungsarten definiert. Das Ergebnis war ein zweistufiger Entscheidungsbaum. In diesem Entscheidungsbaum wurden die einzelnen Kriterien mit konkreten Werten ausgestattet, um eine zweifelsfreie Entscheidung herbeizuführen. Die einzelnen Werte wurden gemeinsam mit dem Kunden definiert. Es ist durchaus möglich, das in einem anderen Kundenszenario andere Werte sinnvoll sind.

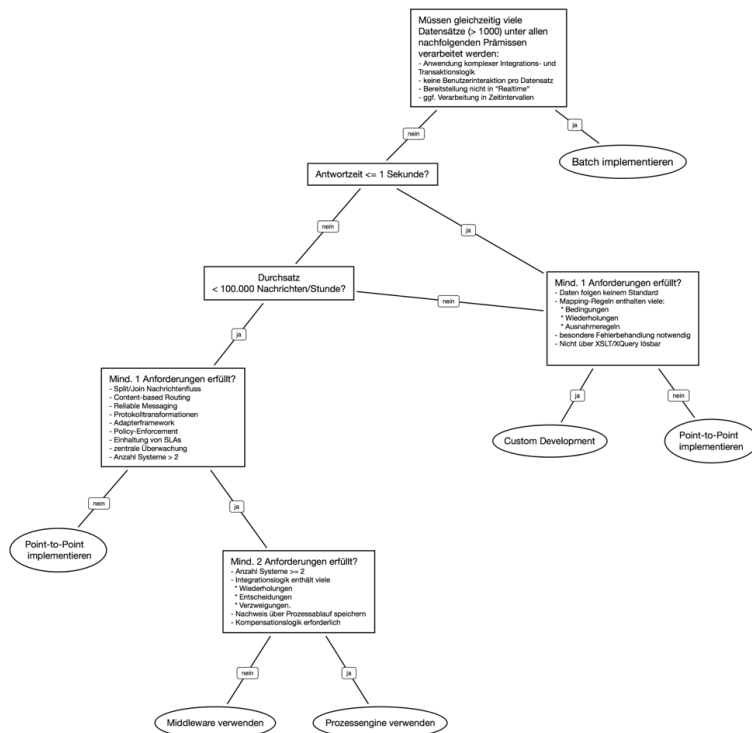


Abbildung 4 Entscheidungsbaum für die Auswahl einer Implementierungsvariante für eine Schnittstellenanforderung

Durch diesen Entscheidungsbaum wird zukünftig festgelegt, welche Implementierungsvariante für Schnittstelle herangezogen werden müssen. Somit werden nicht mehr automatisch alle Schnittstellen über einen ESB realisiert. Der ESB kommt vielmehr zum Einsatz, wenn ganz bestimmte Anforderungen erfüllt sein müssen, z. B. Einhaltung von SLAs, Policy-Enforcement etc.

### Fazit

Im Rahmen einer Integrationsstrategie ist es wenig hilfreich alleine nur auf die Nutzung eines ESB zu setzen. Ein ESB ist sinnvoll kann aber auch nicht alle Integrationsaufgaben sinnvoll umsetzen. Im Vorfeld sollten die Anforderungen und Bestandteile einer Integrationsplattform genau spezifiziert

werden. Der ESB kann dann seine Stärken in dieser Plattform bereitstellen. Vor Auswahl eines neuen Produktes sollten diese Themen analysiert und definiert werden. Eine Auswahl einer geeigneten Plattform oder Produktes gelingt dann leicht. Insbesondere ist eine simple schwarz-weiß Betrachtung im komplexen Umfeld der Integration nicht sinnvoll. In der Zukunft wird das Thema Integration weiter an Bedeutung gewinnen. Aus diesem Grund ist eine durchdachte Integrationsstrategie essentiell. Die Nutzung eines ESB im Rahmen dieser Strategie kann sinn- und nutzvoll sein. Der ESB sollte dann aber auch nur für die Fälle genutzt werden, für die er vorgesehen ist.

**Kontaktadresse:**

Markus Lohn  
esentri AG  
Pforzheimer Str. 132  
D-76275 Ettlingen

Telefon: +49 (0) 7243 354900  
Fax: +49 (0) 7243 3549099  
E-Mail [markus.lohn@esentri.com](mailto:markus.lohn@esentri.com)  
Internet: [www.esentri.com](http://www.esentri.com)