

Service-orientierte Software-Architekturen

Prof. Dr. U. Hoffmann
FH Wedel

Enterprise Service Bus

Enterprise Service Bus

ESB

Aufgaben
Heterogenität
Routing
Verantwortlichkeiten
höherwertige Leistungen

Zusammenfassung

Lose Kopplung

- ▶ unterschiedliche Formen
- ▶ Kommunikation
synchron/asynchron
- ▶ Datentypen
gemeinsame fundamentale Datentypen
Mapping von Datentypen
einheitliches Business Object Model
- ▶ Kompensation
Alternative zu verteilten Transaktionen
- ▶ Infrastruktur
zentrale/dezentrale Kontrolle

Enterprise Service Bus

ESB

Aufgaben
Heterogenität
Routing
Verantwortlichkeiten
höherwertige Leistungen

Zusammenfassung

ESB

Aufgaben

Heterogenität

Routing

Verantwortlichkeiten

höherwertige Leistungen

Der *Enterprise Service Bus* ist das technische Rückgrat einer SOA-Landschaft. (engl. *Backbone* oder auch *Backplane*)

Welche Aufgabe hat der ESB?

- ▶ Konnektivität herstellen
 - ▶ Daten transformieren
 - ▶ (Intelligent) routen
- } Kernaufgabe:
Interoperabilität
- ▶ Mit Sicherheitsaspekten umgehen
 - ▶ Mit Aspekten der Zuverlässigkeit umgehen
 - ▶ Services verwalten (managen)
 - ▶ Möglichkeit zum Überwachen, Protokollieren und Debuggen bereitstellen.

Heterogen!

- ▶ unterschiedliche Plattformen und Hardware
- ▶ unterschiedliche Middleware und Protokolle

„Datentransformation ist ein inhärenter Teil des Busses in einem ESB-Deployment. Transformations-Dienstleistungen, die auf die einzelnen am Bus angeschlossenen Anwendungen spezialisiert sind, findet man überall und kann man von überall aufrufen. Da Datentransformation ein derartig integrierter Bestandteil eines ESBs ist, kann man es auch so betrachten, dass ein ESB den Impedanz-Unterschied zwischen Anwendungen ausgleicht.“

David Chappel, *Enterprise Service Bus*

- ▶ Gefahr von Performance-Problemen
- ▶ Übliches Vorgehen:
 - ▶ festes Zwischenformat definieren (etwa SOAP)
 - ▶ in das alle Plattformen und APIs konvertieren

Wie findet ein Service–Aufruf seinen Weg vom Nutzer zum Anbieter und wieder zurück (für die Antwort)?

Große Bandbreite von

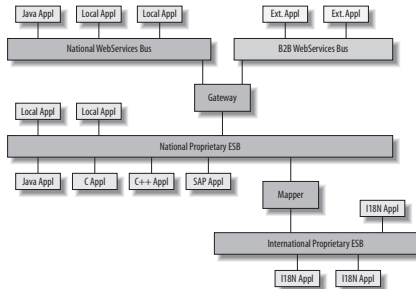
- ▶ trivial bis
- ▶ kompliziert (viele Protokolle, Prioritäten, Geltungsbereiche)

Homogener oder Heterogener ESB?

Homogenität des ESB

- ▶ ist vorteilhaft (einfacher zu handhaben, entwickeln, ...)
- ▶ ist nicht zwingend
- ▶ ist in der Praxis selten dauerhaft der Fall
- ▶ Bedeutet einheitliche Protokoll *und* Konventionen (etwa zur Fehlerbehandlung)

Ein technisch heterogener ESB ist logisch *ein* ESB.



Details der Infrastruktur sollen keinen Einfluss auf die fachlichen Schnittstellen der Services haben.

Enterprise Service Bus

ESB

Aufgaben

Heterogenität

Routing

Verantwortlichkeiten

höherwertige Leistungen

Zusammenfassung

ESB können sich erheblich voneinander unterscheiden:

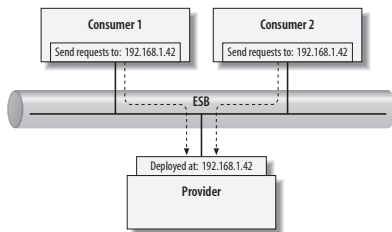
- ▶ **einfacher ESB**: nur bindende Absprachen
 - ▶ Definition eines Protokolls basierend auf Netzwerk–Standardprotokollen
 - ▶ keine Tools oder Software–Komponenten
 - ▶ Aufwand bei Anbieter und Nutzer

- ▶ **elaborierter ESB**: unterstützende Komponenten
 - ▶ viele verschiedene Tools und Programme
 - ▶ zentrale und dezentrale Komponenten
 - ▶ Unterstützung für Service–Designer,
–Implementierer, –Operator
 - ▶ Übernahme vieler Dienstleistungen
 - ▶ wenig Infrastruktur–Aufwand bei Anbieter und Nutzer

Wie stark sind Anbieter und Nutzer gekoppelt?

Alternativen:

- ▶ Nutzer muss die physikalische Adresse des Empfängers angeben oder
- ▶ ESB hat Vermittlungs-Mechanismen



Point-to-Point-Verbindung

- ▶ Nutzer muss Endpunkt genau kennen (IP-Adresse, URL)
- ▶ Request geht an spezifischen Empfänger

Enterprise
Service Bus

ESB

Aufgaben

Heterogenität

Routing

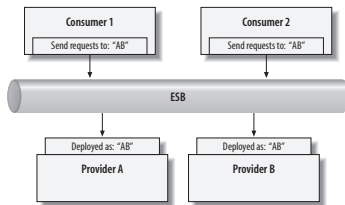
Verantwortlichkeiten

höherwertige Leistungen

Zusammenfassung

Vermittelte Service–Aufrufe

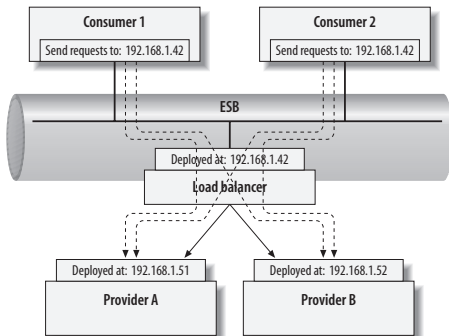
- ▶ Nutzer kennt die genaue Adresse des Anbieters nicht.
- ▶ Einigung auf einen symbolischen Namen, der den Anbieter identifiziert (z.B. URI)
- ▶ statische oder dynamische Konfiguration des ESB
- ▶ Vermittlung durch den ESB ⇒ Lose Kopplung



Vermittlung zur Laufzeit:

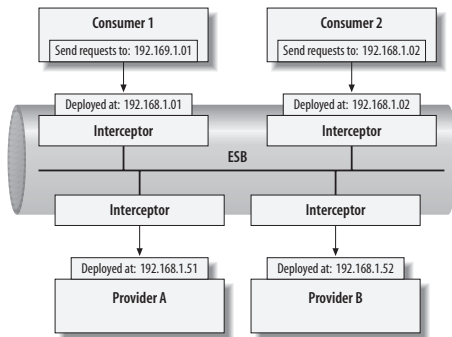
- ▶ Intelligentes Routing
 - ▶ Prioritäten, Geltungsbereiche
 - ▶ verschiedene Konfigurationen
 - ▶ Content–Based–Routing (CBR)
- ▶ Reaktion auf Laufzeitprobleme,
 - ▶ Redundanz, dynamisches Rerouting

- ▶ Interceptoren (Proxies)
- ▶ Indirekte Service-Aufrufe
- ▶ physikalische Endpunkte von Service-Anbietern durch Load-Balancer-Hardware/Software ersetzen.
- ▶ Aus der Sicht der Nutzer wie Punkt-zu-Punkt-Verbindungen



ESB mit Interceptoren für Anbieter und Nutzer

- ▶ Nutzer kommuniziert nur Punkt-zu-Punkt mit seinem Interceptor
 - ▶ Anbieter sind über Interceptoren erreichbar.
 - ▶ Realisierung im ESB vollkommen von den Services entkoppelt
- ⇒ ESB als Black-Box



- ▶ Insbesondere für Web-Services (nur Punkt-zu-Punkt möglich)
- ▶ Weitere Interceptor-Einsatzgebiete: Monitoring und Sicherheit

Enterprise Service Bus

ESB

Aufgaben

Heterogenität

Routing

Verantwortlichkeiten

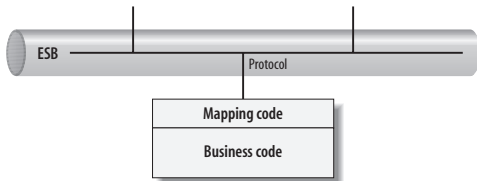
höherwertige Leistungen

Zusammenfassung

Verantwortlichkeit des ESB: Protokollgetriebener ESB

protokollgetriebener Ansatz

- ▶ ESB definiert ein Protokoll
- ▶ Anbieter und Nutzer halten das Protokoll ein, um teilzunehmen
- ▶ von Web-Services mit SOAP verwendet

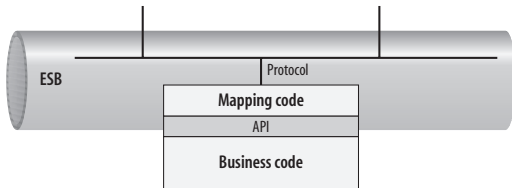


- ▶ ESB unabhängig vom Entwicklungsprozess (und den Teams) der Anbieter und Nutzer
- ▶ Infrastruktur bietet keine Tools, Bibliotheken, APIs
- ▶ Anbieter und Nutzer können selbst entscheiden, welche Tools sie zum Einhalten des Protokolls einsetzen.
- ▶ Neue Systeme: Entwicklerteam ist verantwortlich für den Anschluss an ESB
- ▶ Infrastruktur-Team ist nicht involviert.

Verantwortlichkeit des ESB: API-getriebener ESB

API-getriebener Ansatz

- ▶ ESB definiert plattformspezifische APIs für Services (z.B. Java Schnittstellen)
- ▶ Anbieter und Nutzer verwenden APIs zum Implementieren und Aufrufen von Services.



- ▶ ESB für plattformspezifische APIs verantwortlich
 - ▶ ESB-Team muss sich um Probleme der Anbindung kümmern.
 - ▶ Anbieter und Nutzer konzentrieren sich auf fachliche Aspekte.
 - ▶ Neue Plattform: Infrastruktur-Team ist verantwortlich für den Anschluss an ESB (Bibliotheken, Generatoren)
 - ▶ Anbieter, Nutzer verwenden Code des ESB-Teams
- ⇒ Auswirkungen auf den Entwicklungsprozess

Enterprise Service Bus

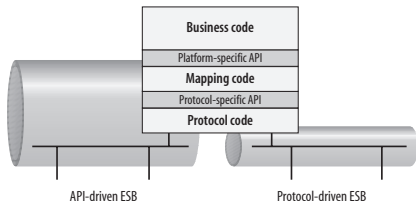
ESB

Aufgaben
Heterogenität
Routing

Verantwortlichkeiten
höherwertige Leistungen

Zusammenfassung

Typische Schichtung von ESB-Bibliotheken und APIs:



- ▶ **Protocol-Code:**
stabile Implementierung des (eines) ESB-Protokolls
- ▶ **Protocol-specific API:**
Schnittstelle für die Kommunikation über das Protokoll
- ▶ **Mapping-Code:**
Plattformspezifische Anbindung des Protokolls
- ▶ **Platform-specific API:**
Schnittstelle zum Aufrufen und Implementieren von Diensten
- ▶ **Business-Code:**
Realisierung der fachlichen Logik

Enterprise Service Bus

ESB

Aufgaben
Heterogenität
Routing

Verantwortlichkeiten
höherwertige Leistungen

Zusammenfassung

- ▶ Lose Kopplung
 - ▶ Wenige fundamentale gemeinsame Datentypen
 - ▶ eigenen Datentypen an den Service–Schnittstellen
 - ▶ Versionierung: mehrere Varianten der gleichen Schnittstelle mit evtl. unterschiedlichen Datentypen
- ⇒ Nutzer sollen eine dünne Mapping–Schicht vorsehen:
Abbildung der Anbieter–Datentypen auf eigene Datentypen

Alternative:

- ▶ ESB bietet Unterstützung für das Mapping
 - ▶ API-getriebener ESB kann mehrere Versionen anbieten
 - ▶ Adapter für das Mapping zwischen Schnittstellen
 - ▶ Definition eines gemeinsamen Objektmodells im ESB
(*Master–Data–Management, MDM*)
- Achtung: enge Kopplung**

Protokolle sichern unterschiedliche *Aufrufsemantiken* zu:

▶ **at most once**

Nachricht wird zugestellt, Empfänger bekommt sie mglw.
Einfach realisierbar

▶ **at least once**

Nachricht wird ggf. mehrfach zugestellt,
bis Empfänger sie bekommt
Durch wiederholtes Senden realisierbar.

▶ **once and only once**

Genau eine erfolgreiche Zustellung der Nachricht wird garantiert
Aufwändig mit Handshakes zu realisieren

Welche Aufrufsemantik soll der ESB garantieren?

- ▶ ESB kann Qualität des Aufrufs verbessern
- ▶ fachliche Anforderungen sind wesentlich

ESB

Aufgaben
Heterogenität
Routing
Verantwortlichkeiten
höherwertige Leistungen

Zusammenfassung

Service–Management

- ▶ Überblick über die Service–Landschaft
- ▶ fachliche und technische Sicht

Überwachen, Protokollieren, Debuggen

- ▶ Nachverfolgen von Service–Aufrufen
- ▶ ESB ist Debugger für verteilte Geschäftsprozesse
- ▶ Zuordnen von Anfragen und Antworten über mehrere Systeme
- ▶ Logging: Wer ruft wann welchen Service auf?
Außerbetriebnahme

Business–Activity–Monitoring (BAM)

- ▶ Überwachung *fachlicher* Aktivitäten
- ▶ Erkennen fachlicher Tendenzen und Ungewöhnlichkeiten
- ▶ Statistik zur Aufrufhäufigkeit von Services
- ▶ Meßwerkzeug etwa für Wirksamkeit von Marketing–Maßnahmen
- ▶ Korrelation von Nachrichten

Enterprise Service Bus

ESB

Aufgaben
Heterogenität
Routing
Verantwortlichkeiten

höherwertige Leistungen

Zusammenfassung

- ▶ Enterprise–Service–Bus ist Infrastruktur von SOA
- ▶ Kernaufgaben: Interoperabilität
(Konnektivität, Daten–Mapping, Routing)
- ▶ zusätzliche Dienstleistungen: Sicherheit, Monitoring, Logging

- ▶ ESBs sind häufig heterogen.
- ▶ Protokoll– und API–basierte ESBs

- ▶ ESB ist Debugger für verteilte Geschäftsprozesse
- ▶ Business–Activity–Monitoring

- ▶ Fragen?
- ▶ Serviceklassifizierung

Enterprise Service Bus

ESB

Aufgaben
Heterogenität
Routing
Verantwortlichkeiten
höherwertige Leistungen

Zusammenfassung