



# Integration Patterns mittels OSB - Beispiele aus der Praxis

DOAG Konferenz Nürnberg 2014  
Ulrich Haug  
19. November 2014

© CGI Group Inc. 2014

November 2014

**CGI**

Experience the commitment®

# Referent: Ulrich Haug



Manager SOA und EAI Architekturen,  
CGI (Germany) GmbH & Co. KG

## Schwerpunkte

- Service Orientierte Architekturen und Enterprise Application Integration
- SOA Stacks, insbesondere
  - Oracle Service Bus
  - TIBCO Business Works, ServiceGrid
- Enterprise Java Architekturen



# EAI/SOA gehört zu CGIs strategischen Betätigungsfeldern

## Zahlen

### 300+ Success Stories

Im Bereich EAI & SOA

### 140+ Kunden

Im Bereich EAI & SOA

### 1.200 EAI/SOA Exp.

Weltweit

### Ein europäischer Marktführer in SOA

Quelle: Forrester

### No.1 AM in Europa

Quelle: PAC

## Thought Leadership

CGI ist ein anerkannter Thought Leader im Bereich EAI & SOA



CGI unterhält strategische Partnerschaften mit allen großen EAI/SOA Anbietern

Partner

# Inhalt

**1 Herausforderungen von Integration**

**2 OSB als Integrationsplattform**

**3 Beispiele an Integrationspatterns**



# Herausforderung von Integration

## Ziel von Integration

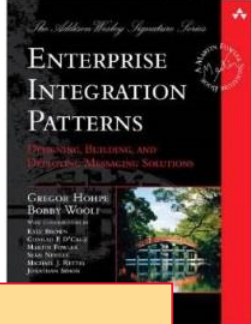
- Datenaustausch
- Aufruf externer Funktionen und Operationen
- Kommunikation von Ereignissen, etc.

## Hürden

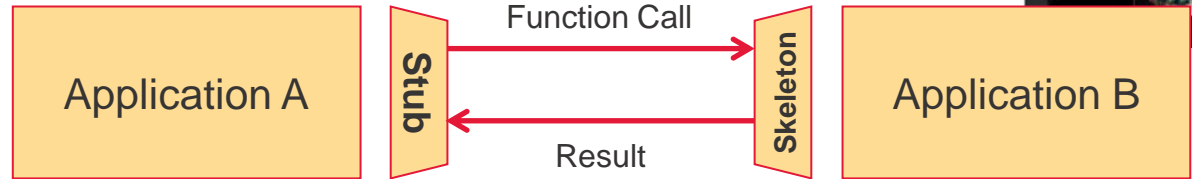
- Stark heterogene Systemlandschaften
  - Systemplattformen
  - Datenprotokolle
  - Transportprotokolle
- Unterschiedliche Anforderungen an Mengengerüste und Laufzeitverhalten
- Legacy Systeme, die keine oder kaum Veränderung zulassen
- Unzuverlässige, langsame Netzwerke
- kaum Standards für Integration – etabliert hat sich Web Services, WSDL, XML, XSD



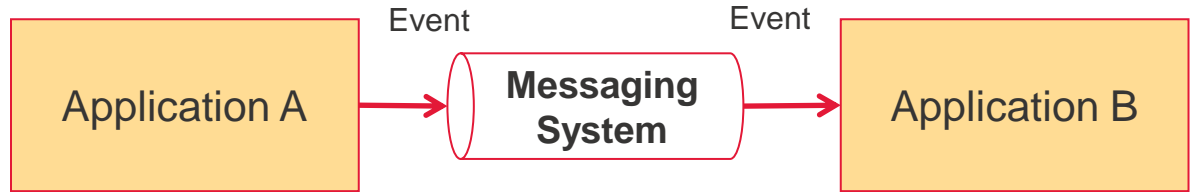
# Integrations-Stile



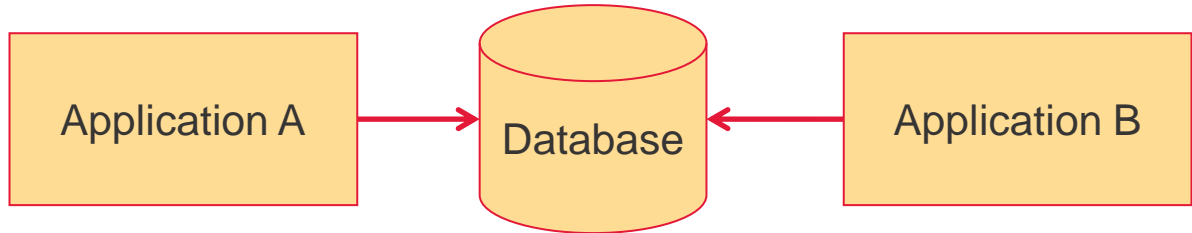
Remote Procedure Calls



Messaging



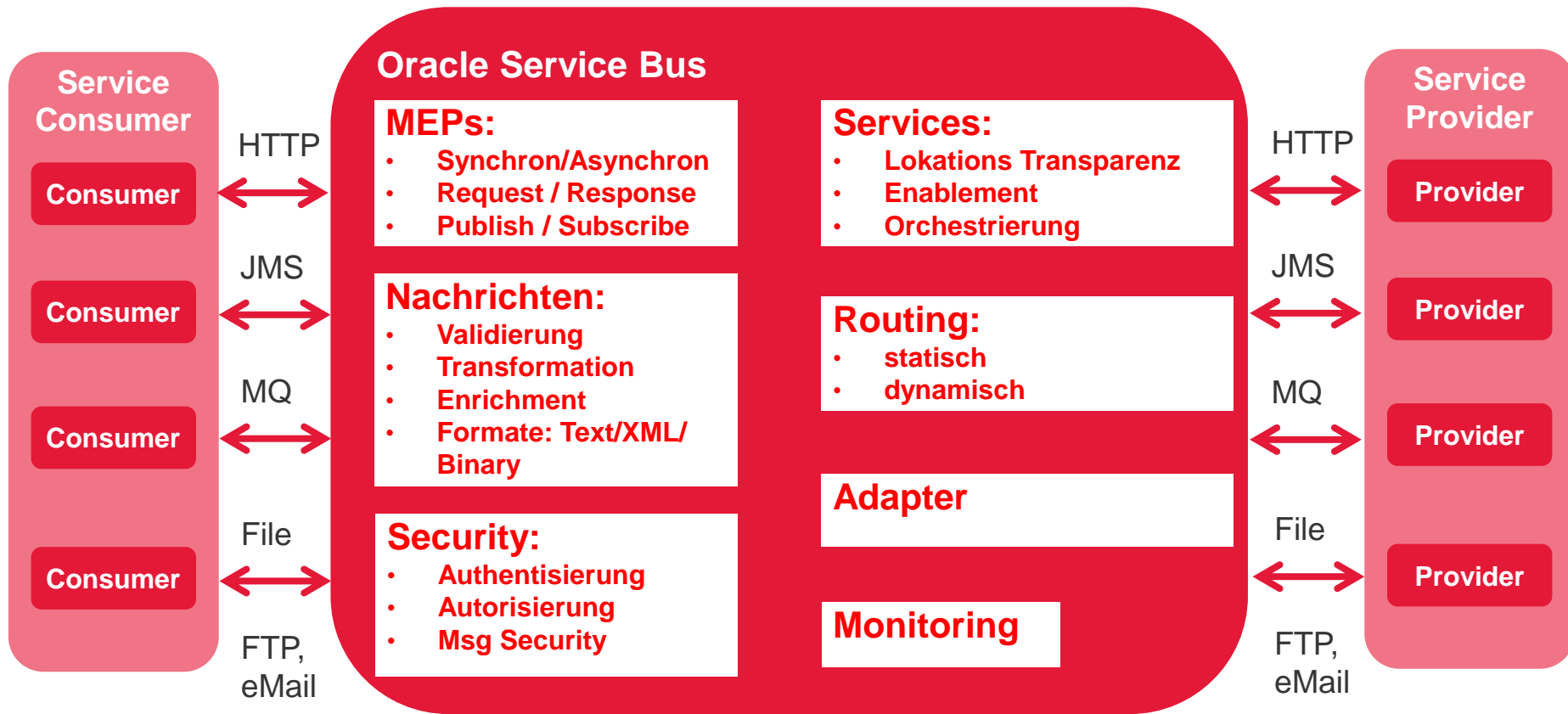
Shared Database



File Transfer



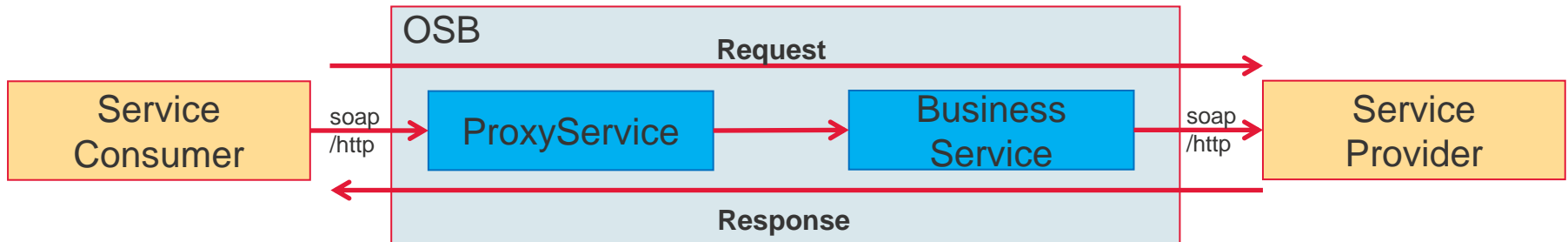
# OSB als Integrationsplattform



Mittels der vom OSB bereitgestellten Funktionalitäten können unterschiedlichste Anforderungen an Integration umgesetzt werden



# IP: Synchroner Request - Response

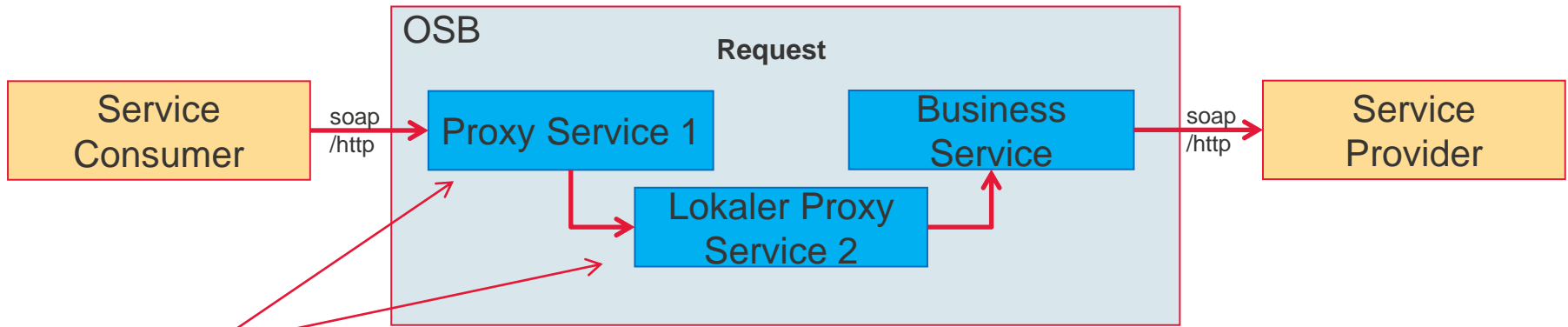


Synchroner  
Request-  
Response

- Umsetzung mittels soap / HTTP
- Im Business Service:
  - Bewusste Entscheidung wie mit Retry und Application Errors umgegangen wird
    - im Allgemeinen deaktiviert
  - Timeouts definieren
- Zu beachten
  - Laufzeitverhalten des Service Providers
  - Zu lange Laufzeiten belegen Threads und Hauptspeicher des WLS/OSBs



# IP: Allgemeine Vorgaben für Integration



Interne  
Message  
Header

- Dient Logging und Monitoring
- Wird in \$header transportiert und beinhaltet
  - Message uuid, Service Name, Operation, ursprünglicher Header

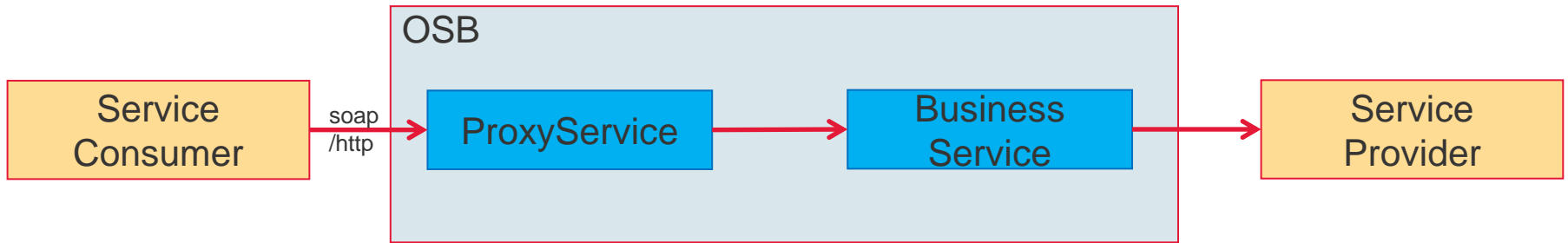
VETRO  
Pattern

- Validierung geschieht in der Regel sowohl bei Request als auch bei Response

Proxy Service

V – alidate  
E – nrich  
T – ransform  
R – oute  
O – perate

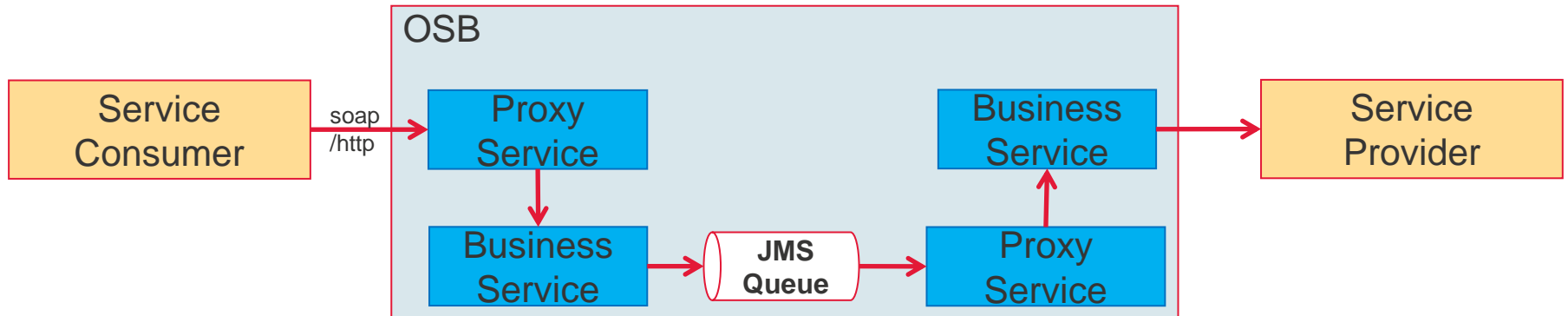
# IP: Oneway für soap/HTTP



## Oneway-Schnittstelle

- Für transaktionale Protokolle wie MQ oder JMS sind solche Schnittstellen i.d.R. nicht problematisch
- soap/http Schnittstellen sind anfällig für unterschiedliche Fehlerbilder:
  - Service Provider steht zum Zeitpunkt des Aufrufs nicht zur Verfügung
  - Die Nachricht entspricht nicht dem geforderten Format, es kommt zu einem Fehler im OSB
  - OSB Knoten bricht bei der Verarbeitung weg

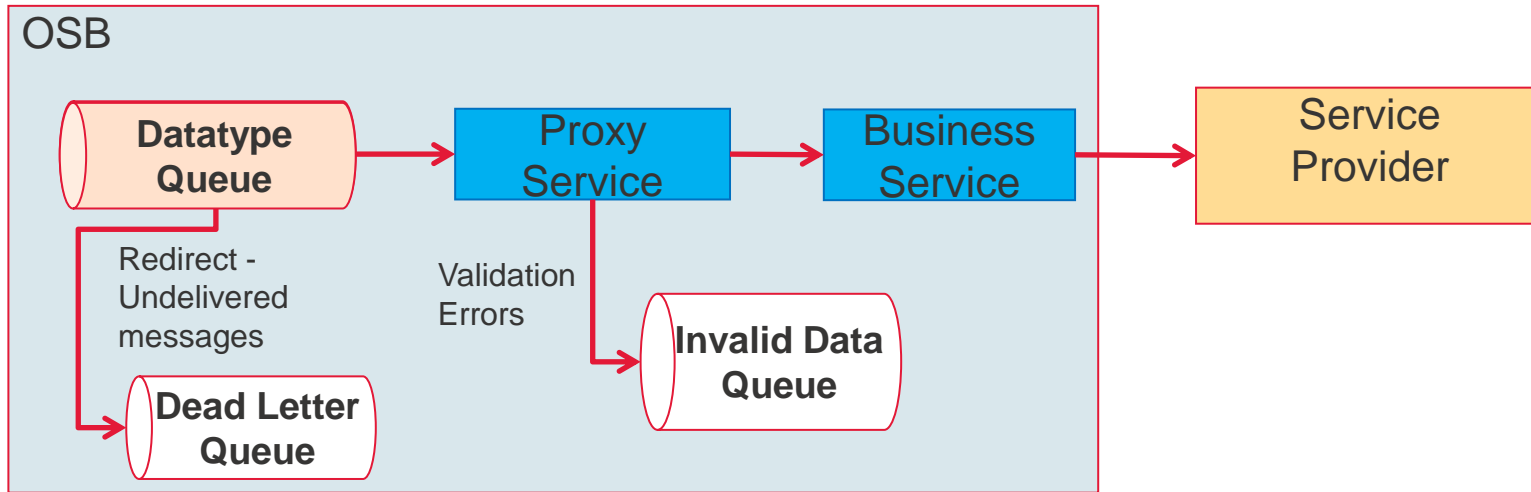
# IP: Oneway für soap/HTTP



JMS Puffered  
Oneway-  
Schnittstelle

- Consumer Schnittstelle wird in Request-Response Schnittstelle geändert
- Nachricht wird in eine **persistente** JMS Queue eingetragen
- Als Response wird eine Quittung zurückgegeben. Diese besteht aus zwei „fachlichen“ Aspekten:
  - Nachricht erfolgreich an den OSB übermittelt. OSB übernimmt im Folgenden die Auslieferung
  - Nachricht **nicht** erfolgreich an den OSB übermittelt (z.B. fehlerhafte Msg, JMS Queue).

# IP: JMS Messaging



Wie sieht das Nachrichtenformat aus?

Verlangen XML als Nachrichtenformat und dedizierte Nachrichten Queues → **Datatype Channels**

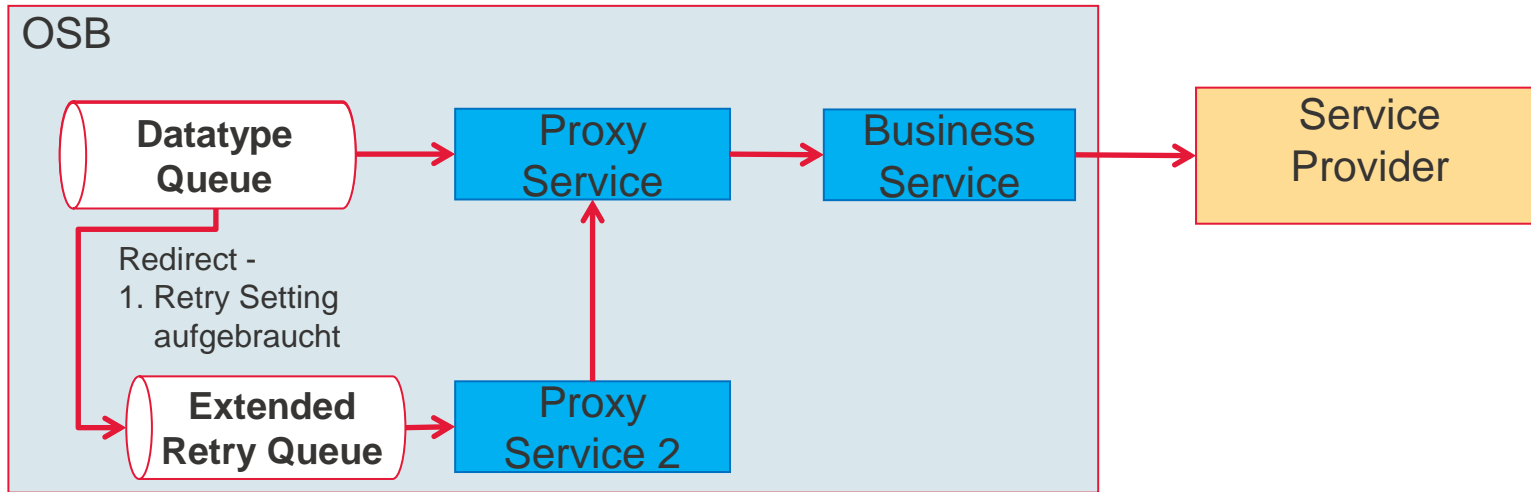
Wie wird mit falschen Nachrichten umgegangen?

Nachrichten werden validiert und bei Fehler aussortiert → **Invalid Data Channel**

Was passiert mit Nicht-Auslieferbaren Nachrichten?

Nachrichten werden vom Service Provider nicht entgegengenommen → **Dead Letter Channel**

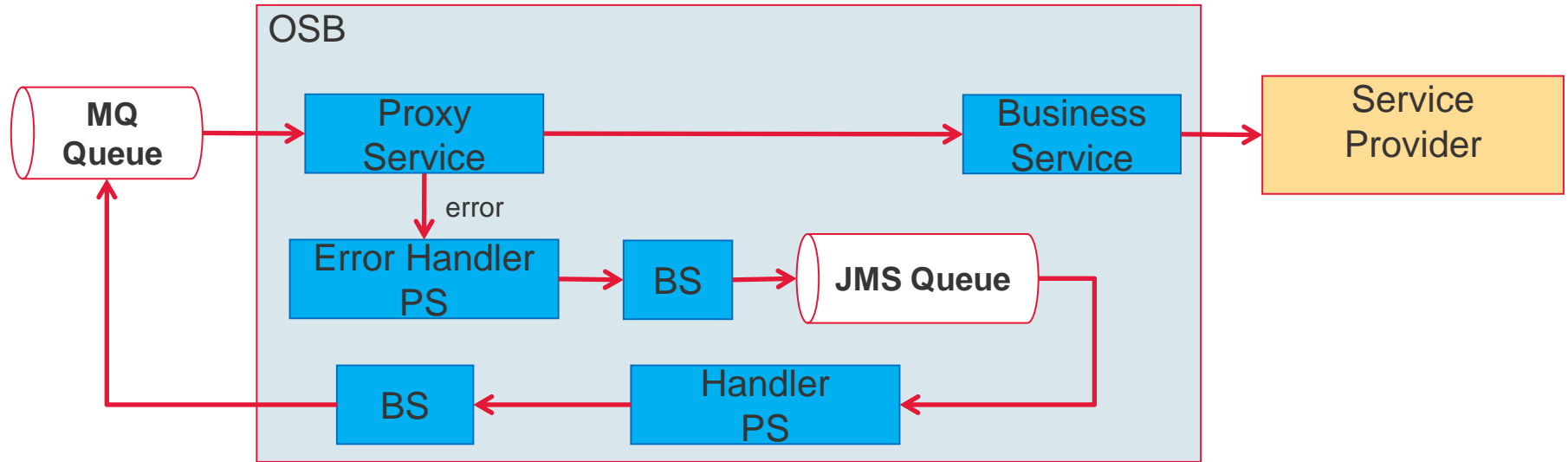
# IP: JMS Messaging Retry Mechanismen



Retry

- Benötigen **XA** JMS Connection Factory und Proxy Service muss Transactions verwenden
- Unterschiedliche Retry-Intervalle über weitere Queues möglich
- Zu beachten:
  - Wie lange ist die Nachricht gültig?
  - Ist der aufgerufene Service Idem potent?
  - Hat der aufgerufene Service eine längere Laufzeit als die Transaktion im Weblogic?

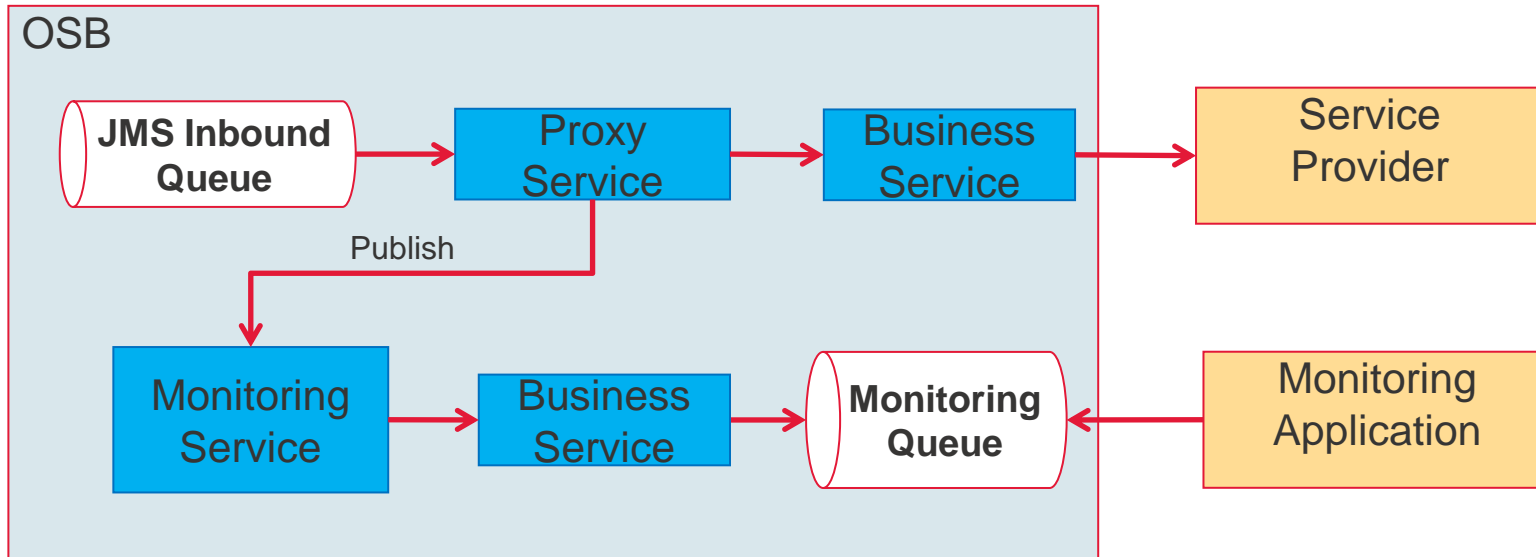
# IP: MQ Integration mit MQ Transport (ohne XA)



MQ Integration  
ohne XA

- Nachricht wird aus MQ Queue geholt und in jedem Fall gegenüber MQ „acknowledged“
- Bei einem Fehler im Messageflow wird die Nachricht an einen Error Handler Proxy Service gegeben und in eine persistente JMS Queue eingestellt
- Ein weiterer Service stellt die Nachricht wieder in die MQ Queue ein

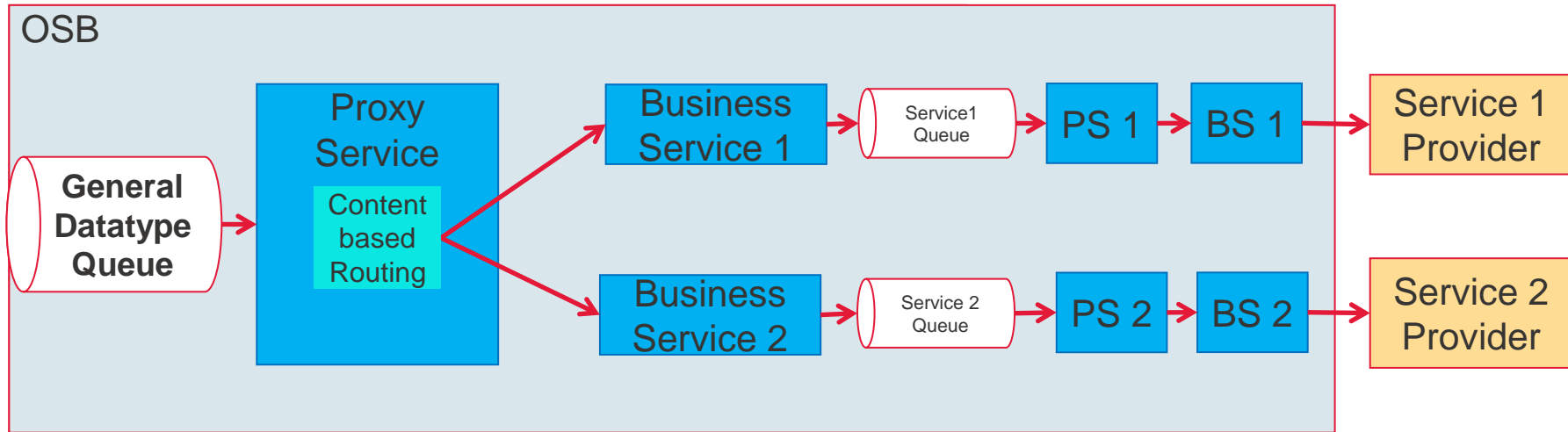
# IP: JMS Messaging Alternative Monitoring



## Monitoring Service

- Alle Nachrichten werden mitgeschnitten → **Wire Tap**
- Monitoring Schnittstelle erlaubt neben der ursprünglichen Nachricht auch Metadaten zu übertragen
- Fehlerhafte Nachrichten werden über das Monitoring ausrangiert

# IP: JMS Messaging General Datatype

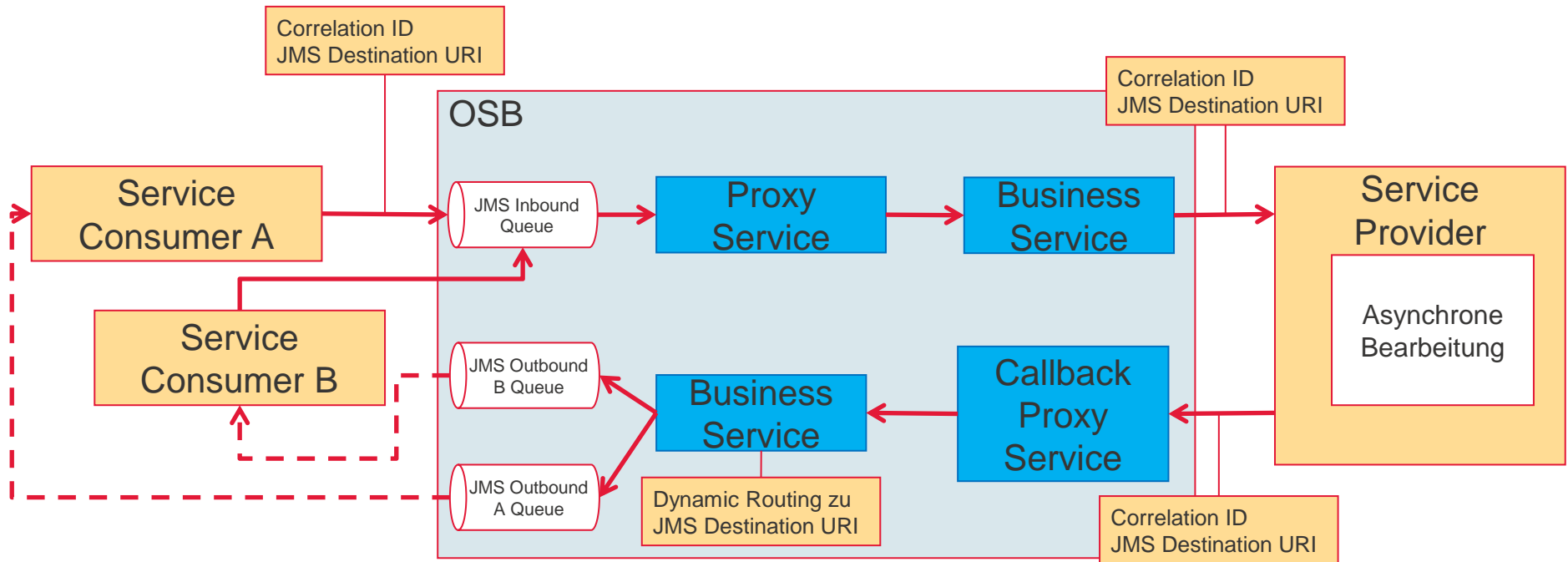


General Datatype

- Datatype Channels führen ggf. zu hoher Anzahl an Queues → **General Datatype Channel**
- Zustellung basierend auf Nachrichteninhalt
- Aber
  - komplexe Nachrichtenverarbeitung
  - unterschiedliche Anforderungen der angebundenen Service Provider (Retry, etc.) verlangen erneute Aufspaltung



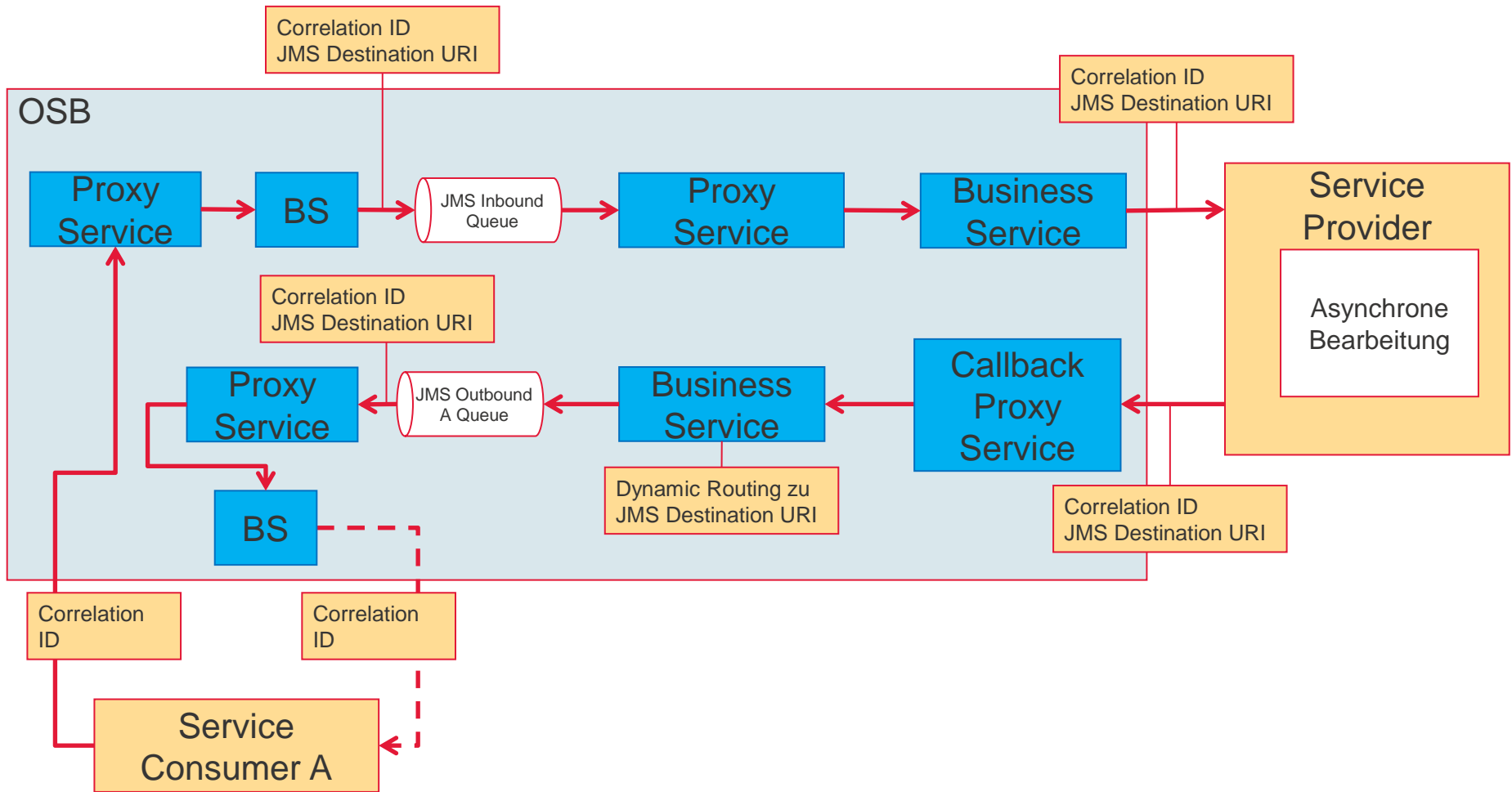
# IP: Request-Callback



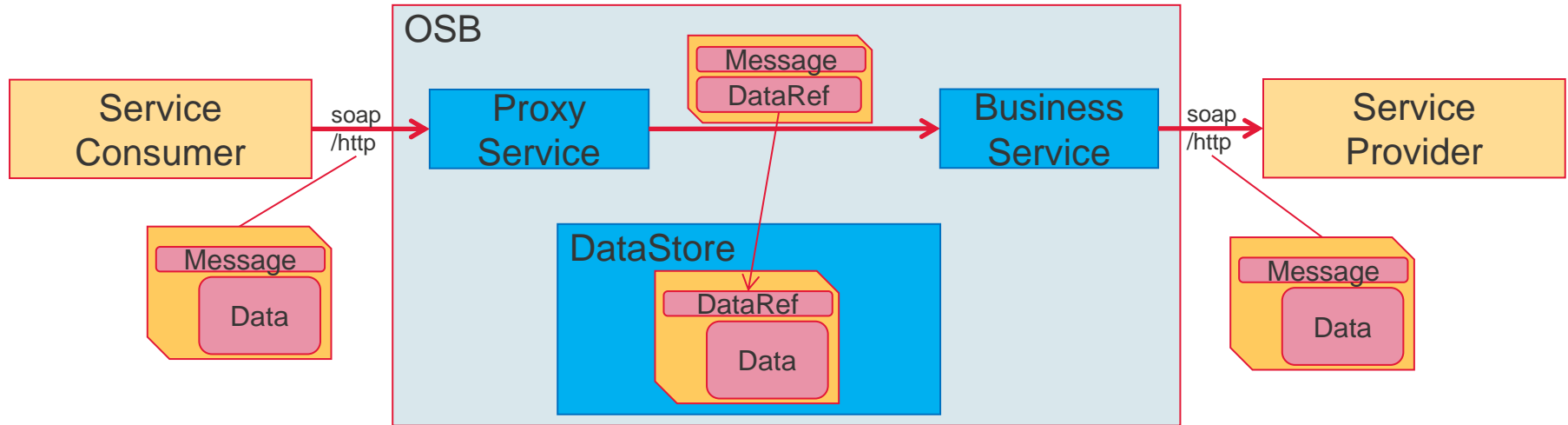
Request-  
Callback

- Request und Callback Schnittstelle wird jeweils als Oneway umgesetzt
- Request und Callback besitzen im Header Correlation ID und Response URI
- Sofern Consumer nicht direkt auf die Queues zugreifen können, werden entsprechende Wrapper (Proxy und Business Services) im OSB realisiert

# IP: soap/http Request-Callback



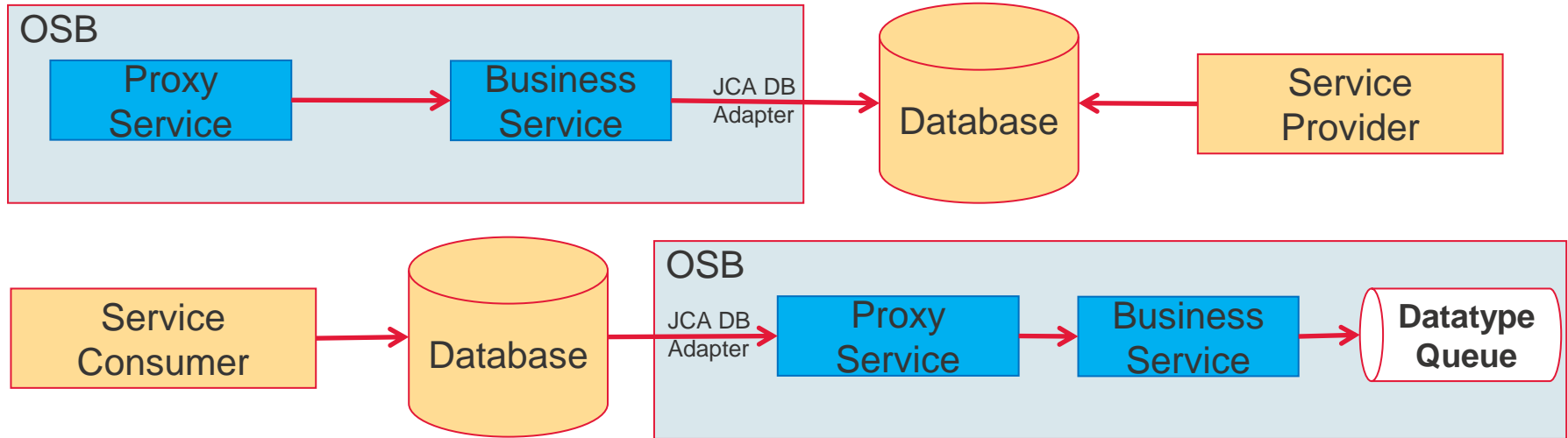
# IP: Umgang mit großen Datenmengen



Große  
Datenmengen

- Bei soap/HTTP wird MTOM verwendet
- Page Attachment to Disk
- Eigenes File Streaming über Java Callout

# IP: Shared Database



Shared Database

- Anwendung sofern Service Provider und/oder Service Consumer kein gängiges Transportprotokoll unterstützt
- Datenbank Tabellen sind reine Schnittellentabellen
- Proxy Service, der auf die DB Tabellen „lauscht“ schreibt Daten in JMS Queue (DB Adapter löscht/markiert die übertragenen Daten)



Ich freue  
mich auf Ihre  
Kommentare &  
Fragen!

**Ulrich Haug**

Manager  
SOA und EAI Architekturen

Mobile: +49 170 5791 256

E-Mail: [ulrich.haug@cgi.com](mailto:ulrich.haug@cgi.com)