

# Neues im WebSphere Message Broker V7

*Carsten Mai*  
*[cmai@de.ibm.com](mailto:cmai@de.ibm.com)*



# Das Herz eines 'interconnected' Unternehmens: der Enterprise Service Bus (ESB)

## Was ist ein ESB?

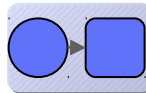
Ein ESB ist eine flexible Verbindungs-Infrastruktur zur Integration von Anwendungen, Systemen und Services.

## Ein ESB leistet zwischen Requestor und Service



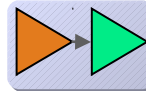
### ROUTET

Kommunikations-Flüsse zwischen Services



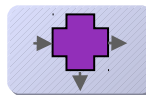
### KONVERTIERT

zwischen verschiedenen Transportprotokollen



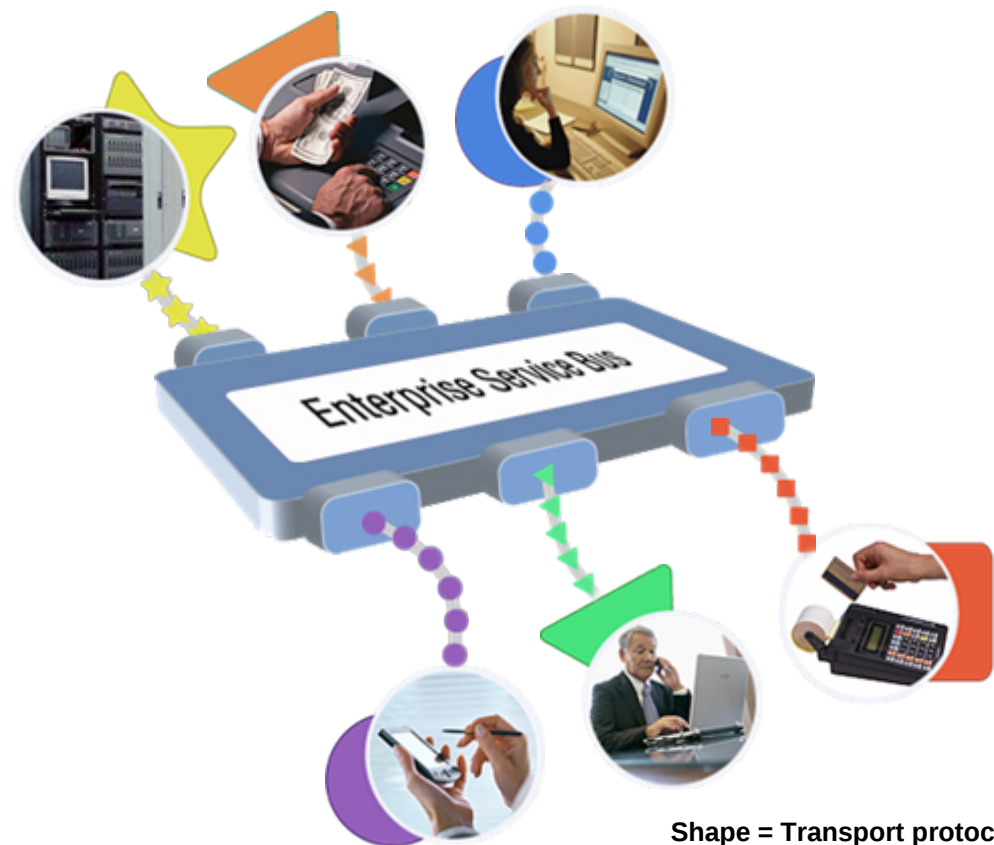
### TRANSFORMIERT

zwischen verschiedenen Datenformaten



### IDENTIFIZIERT und VERTEILT

Business Events



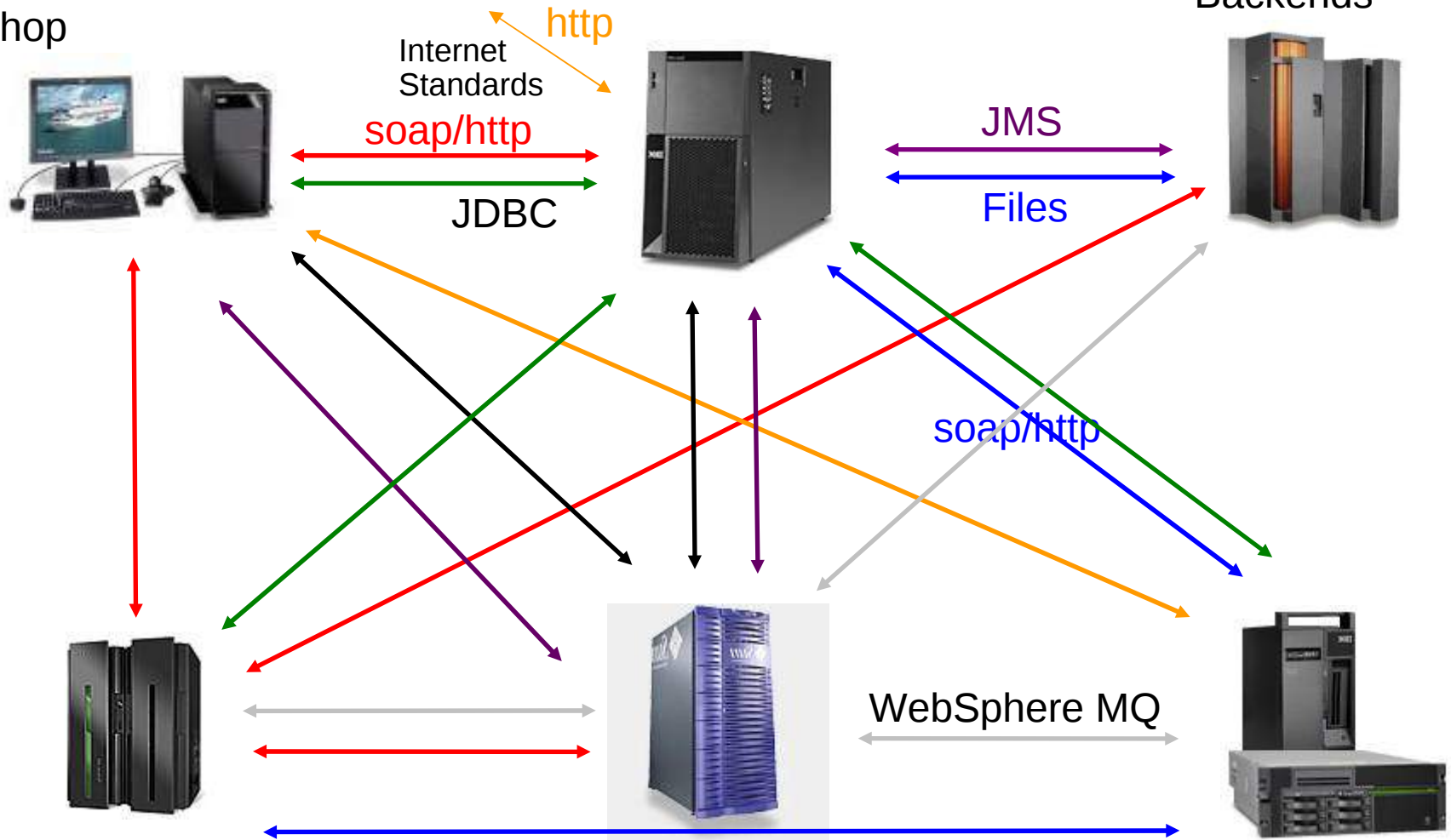
Shape = Transport protocol  
Color = Data format

# Komplexe Anwendungen ohne ESB

Kunden  
Web Shop  
Portal

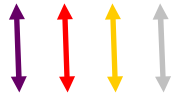
Partners, Lieferanten

Backends

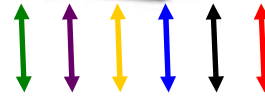


# Ein ESB vereinfacht die Verbindung

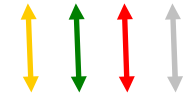
Kunden  
Web Shop  
Portal



Partners, Lieferanten

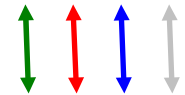
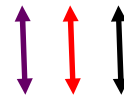
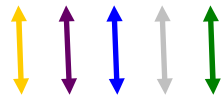


Backends



Enterprise Service Bus

Registry/Repository



# Einfach zu verstehen für existierende und neue Nutzer

## ▪ Minimale Voraussetzungen:

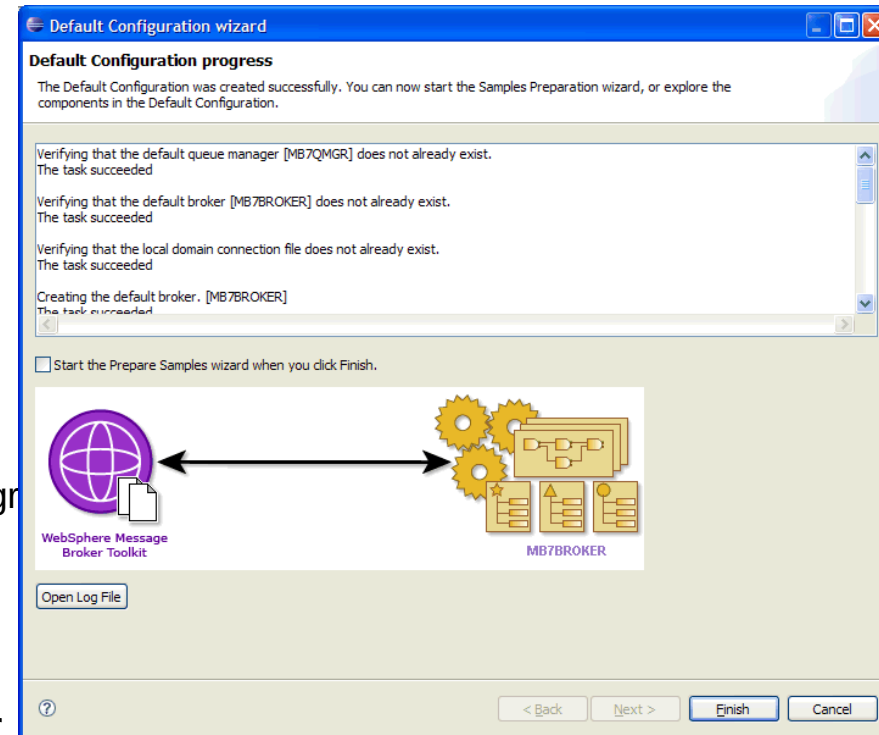
- Anwendungsentwicklung im Message Broker Toolkit
- Message Flows laufen in der Broker Runtime
- MQ 7.0.1 ist einzige Software-Voraussetzung
- Ab Version 7 **NICHT** mehr benötigt:
  - Broker Database, Config Mgr, User Name Server

## ▪ File basierendes Deployment und Konfiguration

- File system für Ressourcen/Konfiguration
  - Backup (online) und Restore der gesamten Konfiguration  
**mqsibackup|restore|broker**
- Transient Status (aggregation, collections...) in MQ

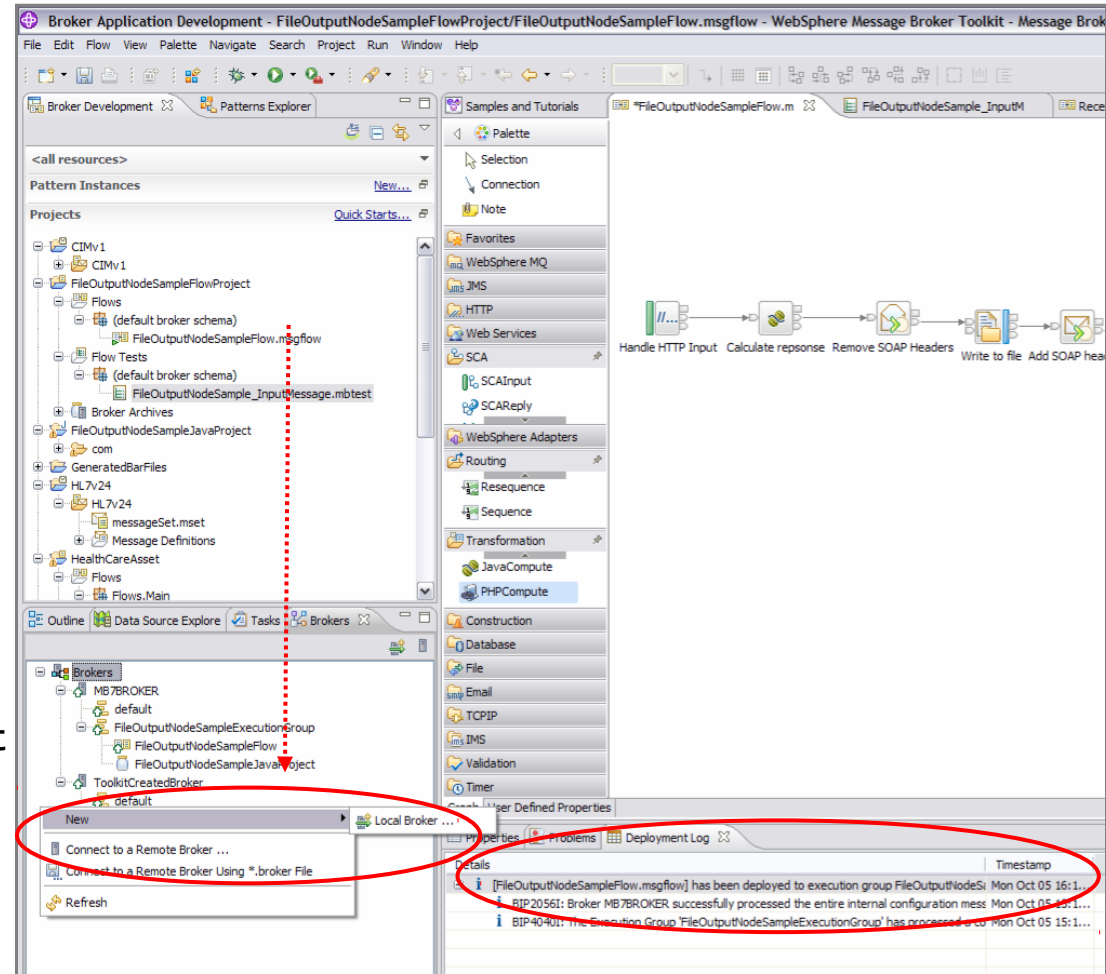
## ▪ Komponenten früherer Versionen von Message Broker

- Entfernen der Broker Datenbank berührt nicht die Zugriffe zu Nutzerdatenbanken; vereinfachter Betrieb des Brokers
- Configuration Manager Funktion nun komplett im Broker
  - Toolkit und Explorer können mehrere Broker managen
  - CMP admin API wird für bestehende Anwendungen weitergeführt
  - Publish subscribe nun im MQ V7.0.1 angesiedelt
  - Neue **SYSTEM.BROKER.AUTH.\*** Queues zur Definition der Zugriffsrechte via MQ Security Definition
- User Name Server wird nicht mehr benötigt
  - Topic- und Inhaltsbasierte Zugriffsrechte in MQ definiert



# Leistungsfähige Anwendungsentwicklung

- **Umfassende AD Perspektive**
  - unterstützt alle AD life cycle Schritte
  - Visuell konsistent mit MBX
  
- **Create, modify, delete Brokes**
  - Create local Broker & remote connect
  - Create local & remote Exec Groups
  - Start & stop Brokers, Execution Groups
  - Autokonfigurieren/Konfigurieren der debug Ports
  - Automatisches Deployment beim Unit test (**.mbtest**)
  
- **Schnelles Compile, Debug, Test (CDT)**
  - Debug local & remote brokers
  - Drag & drop Message Flows direkt
  - Test message flow direkt oder **.mbtest**
  - Drag & drop BARs vom FileSystem
  - Synchrones Deployment Status Log



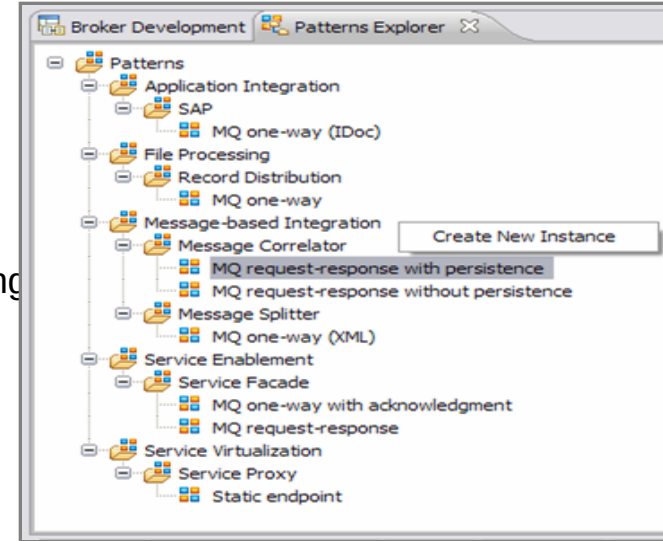
# Patterns für einfachere Entwicklung

## Patterns basierte Entwicklung

- Top-Down, parameterisierte Lösungen
  - e.g. Web Service façades, Message processing, File to queue
- IBM vorgefertigte Patterns
  - Vereinfacht Erstellung häufiger Anwendungsfälle unter Beachtung
- Ergänzt den bestehenden 'Bottom-up' Ansatz

## Patterns Explorer

- Verzeichnis von verfügbaren Broker Patterns
- Jedes Pattern enthält klare Hilfe bezüglich Kontext und Einsatz
- Create Pattern Instance->Customize->Generate->Deploy



## Pattern Generation

- Einfaches Erzeugen von Message Flows, Message Set und Mappings aus mitgelieferten Pattern
- Pattern Properties ermöglichen die Anpassung an örtliche Gegebenheiten
  - Design time properties beeinflussen die Struktur der Message Flos, z.B. Anzahl von Operationen im WSDL
  - Runtime properties beeinflussen die Flow-Ausführung
  - viele Pattern properties resultieren in Flow properties, die später mittels BAR-Editor änderbar sind
- Die Pattern-Instanz kann neu generiert werden, um Eigenschaften zu verändern

# Enhanced Patterns Support (FP 1)

## ▪ User Defined Patterns

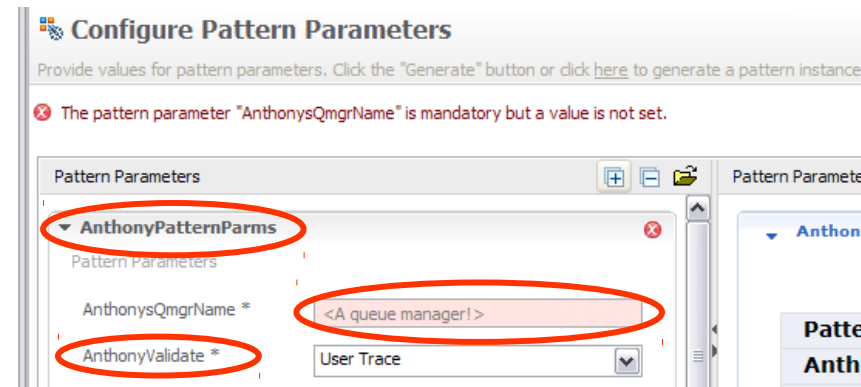
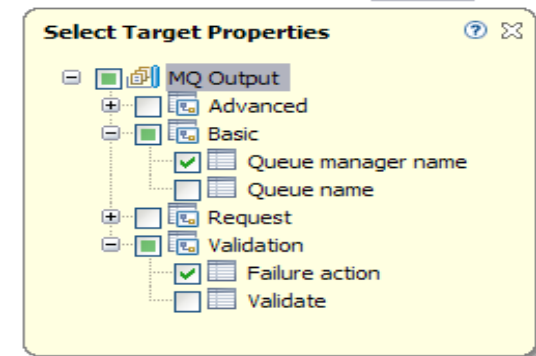
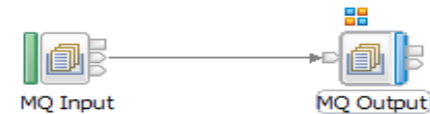
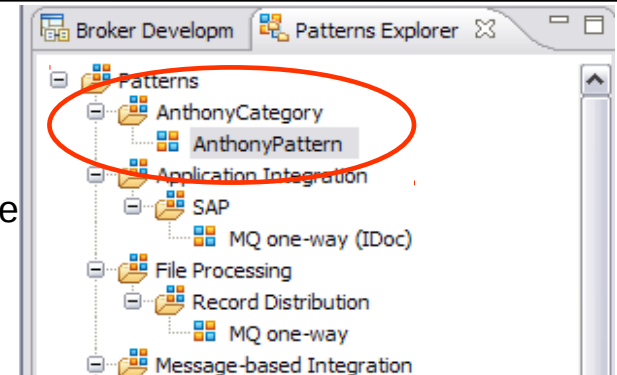
- Eigene Pattern entwerfen und zum Patterns Explorer hinzufüge
  - Ergänzt von IBM gelieferte Patterns
- Erzeugen nachnutzbarer Assets
  - Typischerweise zusammen mit Enterprise Architect
  - Entwickler erzeugt die erforderliche Logik

## ▪ Generieren von Pattern ausgehend von existierenden Message F

- Definieren der Key Message Flow (node) properties
- Erzeugen der Pattern Parameter- & Eigenschafts-Zuordnungen
- Erzeugen der Pattern Documentation und Hilfe; Preview
- Hinzufügen zum Patterns Explorer zur Nachnutzung

## ▪ User-defined Subflows – Nachnutzung als 'Black-Box'

- User-defined Node Packages aus Subflows erzeugt
  - Implementation bleibt verborgen
  - Excellent bei library style node Nachnutzung
- Verteilung & Nutzung via Library Control System



# Patterns Verfeinerung & Communities (FP 2)

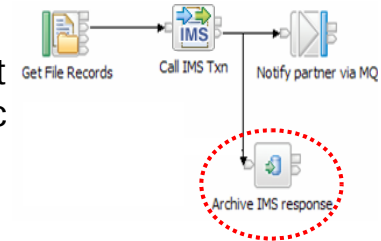
## Patterns Verfeinerung und Editier-Möglichkeiten

- Weitgehende Steuerungsmöglichkeiten für den Patternauthor
  - Generierte Artefakte können während der Generierung angepasst werden
  - Message Flows, Nodes, Connections, Ressourcen und Eigenschaften

Message flow editing benutzt das Management API

- Alle Toolkit Flow Designer Actions per CMP API verfügbar für Patterns Edit
- Message Flow Construction API: add/remove Node, Change Properties etc

```
transformFlow.connect(outputTerminal, inputTerminal);
```



Pattern Properties für andere Ressourcen nutzbar; z.B. ESQL, Maps, MQSC etc

- PHP basiertes Scripting für flexible Anpassung von Assets, incl. debug

```
DEFINE QLOCAL(LOCAL) DEFPSIST(YES) DESCR('<?php echo $_MB['PP']['queueName']; ?>')
```

## Private und Public Patterns Communities

- Package für Sharing, Upload, Kommentierung, Bewertung, Download... eine Patterns community!
  - Ermöglicht mit Pattern in existierenden Community Technologien zu arbeiten
- Patterns package URL <http://myBiz.com/pattern/class/pattern.patternzip>
  - Double-click install von Pattern in den Patterns Explorer!
- Interne/private Patterns Communities innerhalb von Unternehmen
  - Information sharing, best practices, governance
- Anleitung für das Erstellen einer Community
  - Populäre Community Tools wie Joomla, Drupal & RAM
- Neue Geschäftsfelder
  - ISVs, BPs könnten Pattern verkaufen
  - IBM BlueWorks Patterns community

★★★★★ Great SAP to CICS Pattern!

I found this pattern easy to understand and deploy...

[Learn more](#)

Comment by Anthony on Feb 16 2010

# Web Services Enhancements

## ▪ Web Services Gateway

- Vereinfachung in großen Web Services Proxy Szenarien
  - Flexibly process inbound & outbound WS-\* requests
- Gateway Funktion für generisches SOAP processing
- Ergänzt den bisherigen 1 WSDL per SOAP node Ansatz
- Intelligent Default Processing
  - Auto-detect 1-way Requests
  - Dynamische Validation gegen deployed Schema

## • JSON Parser

- Built-in JSON parser via **JSON** domain, including JSONP
- Einfache Verarbeitung von JSON Daten
  - Volle Unterstützung der JSON Daten Typen, incl. arrays, objects , nulls

## ▪ SOAP/JMS transport & HTTP Enhancements

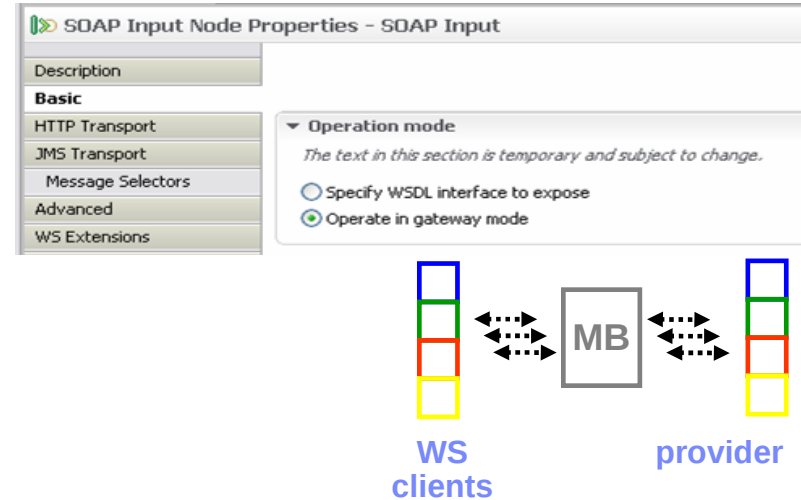
- SOAP/JMS Async Request & Response Nodes
  - Message Flow Thread nicht durch externen SOAP/JMS-Aufruf blockiert
- SOAP/JMS Operationen (optional) in Message Flow Transaction enthalten
- gzip, zlib and deflate compression now handled for HTTP & SOAP nodes

## Web Services funktionale Verbesserungen

- ?wsdl /?xsd Unterstützung; erlaubt Client die Abfrage der Web Service Definition zu

## Enhanced HTTP Input Nodes

- HTTP Input an Execution Groups gebunden; erhöhte Skalierbarkeit und Leistung
- zusätzlich zum 'per Broker single HTTP listener'



```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25,
  "address": {...
}
```



SOAP Asynchronous Request



SOAP Asynchronous Response

# FTE Nodes: Teilnehmer im Managed File Trans



- **FTE Nodes für End-to-End transactionale File-Verarbeitung**
  - FTE Input & FTE output Nodes erlauben MB Files per FTE Transfer zu empfangen und **FTE Input**
- **FTE Input node**
  - Message Flow startet Verarbeitung sobald der FTE Agent einen abgeschlossenen Filetransfer anzeigt
  - FTE Metadata im **LocalEnvironment** verfügbar --> intelligente Verarbeitung möglich
  - File kann als ganzes oder als einzelne Sätze verarbeitet werden (wie File Node)
- **FTE Output node**
  - Erzeugt File und initiiert FTE Transfer mit den Metadaten
  - **LocalEnvironment** ermöglicht dynamisches Überschreiben der Transferparameter und Metadaten
- **FTE Agents werden automatisch Installiert, Konfiguriert und Gemanaged**
  - Wird automatisch mit installiert, automatisch konfiguriert (Agent Name, Transfer Directories etc.)
  - FTE Agents werden mit der Execution Group gestartet

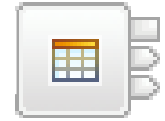


FTE Output

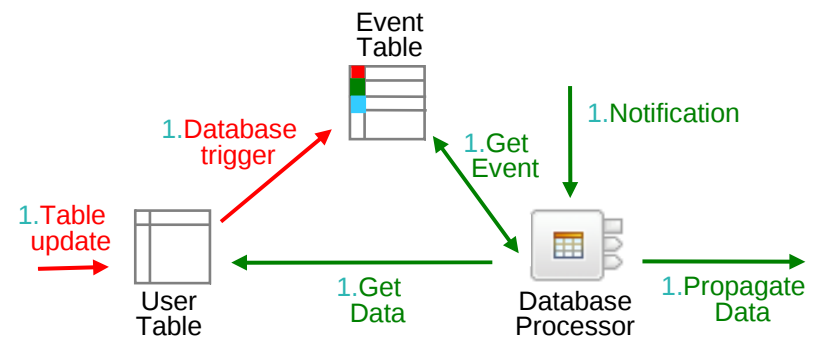
# Database Input Node

## Database Input node

- Triggered & polled Queries um Message Flow zu starten
- Supports Database Update Technologien:
  - Changes (Insert Update, Delete) in Database Tables --> Database Trigger Records in Event Table



DatabaseInput



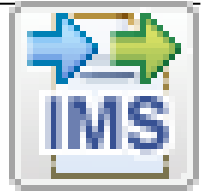
## Nutzt Data Tools für 'Code free' oder SQL basierte Queries

- RAD Tools erlauben das graphische Erstellen der Database Input Node Properties
- Power user können Notification, GetEvent, RetrieveData, PresentData & EndEvent SQL routines spezifizieren

## Operational Control

- Unterstützt alle Message Broker databases
- MB Explorer Resource Statistics pro Database

# IMS Node



## ▪ IMS Request node

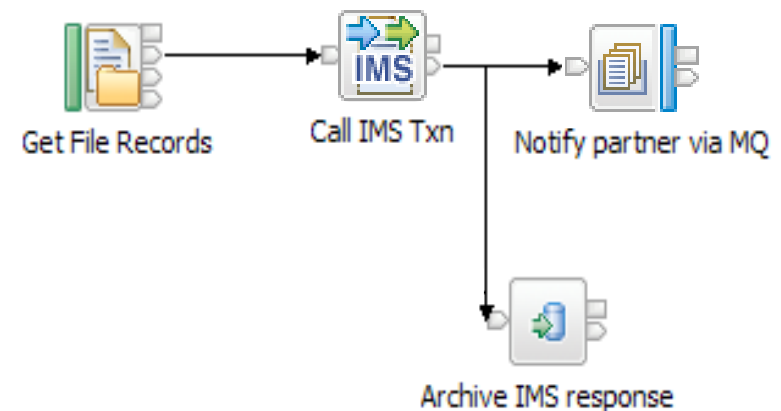
- Message Flows können IMS transactions aufrufen und Antworten verarbeiten
- Typische Szenarien: Web Service->IMS, File->IMS, SAP->IMS...
- Hochleistungs-, synchroner, Multi-platform Zugriff auf IMS
- ergänzt MQ IMS Bridge & IMS Web Services

## ▪ Synchroner Aufruf von IMS Transactions und Command

- Unterstützt viele IMS Facilities
  - MPP, BMP und Fast Path Transaction Regions
  - Commit mode 0, 1
  - SyncLevel NONE, CONFIRM
  - Single & multi segment IMS messages

## ▪ nutzt IMS TM Resource Adapter

- Integraler Bestandteil von Message Broker
  - IMS Connect ist erforderlich



# CICS Request Node



CICS Request

- **New CICS Request Node**
  - Message Flows können CICS Transaktionen aufrufen und Antworten verarbeiten
  - Typische Szenarien: WebService->CICS, File->CICS, SAP->CICS, IMS->CICS
  - Hochperformanter, synchroner, Multi-platform Zugriff auf CICS
  - ergänzt MQ bridge und Web Services
  
- **Synchroner Aufruf von CICS Programmen**
  - Node unterstützt Specification von COMMAREA Request & Response
  - Timeouts, APPLIDs, SYNCONRETURN ebenso verfügbar
  - Design erlaubt zukünftige Unterstützung von Channels & Containern & 2PC Transactions
  
- **Security**
  - Link level via SSL, und Message Level Identity via security profiles
  
- **Flexible Topology Support**
  - gegenwärtig 2 Tier Topologies: MB->CICS
  - Zukünftig auch 3 Tier (MB->CTG->CICS) Topologies
    - CTG ermöglicht zusätzliche HA und WLM Optionen

# Sequencing und Re-sequencing Nodes

## ▪ New Sequence und Resequence nodes

- Ungeordnet ankommende Messages werden anhand einer Sequenznummer geordnet weitergeleitet
- Sequence Gruppen haben definierte Starts und Ende; innerhalb der Gruppe in Reihenfolge weitergeleitet
- Synergie mit Aggregation & Collector Nodes
- Multi-threaded; verarbeitet mehrere Sequenz Gruppen parallel



Resequence



Sequence

**G1: {1,2,5,4,3} → {1,2,3,4,5}**

## ▪ Resequence node sammelt Messages nach:

- Sequence Nummer Location innerhalb der Message
- Optionaler Group Location bei Sequences von unterschiedlichen Quellen (e.g. account identifier, date)
- Flexible Sequence Start & Ende Definition
  - Literal value, Prädikat or Timer basiert
  - e.g. (0,10)(\$Root/FirstMsg, \$Root/LastMsg)

## ▪ Timeout handling

- Missing message timer überwacht Lücken in der Sequenz
- Duplicate Messages werfen Exception

## ▪ Ergänzt durch Sequence node

- Nutzer kann neue Sequences und Gruppen erzeugen
- Unter Nutzung bestehender oder neuer Felder
- Synchronized Groups stellen richtige Verarbeitungsreihenfolge sicher

**Resequence Node Properties - Resequence**

**Description** ⚠ Path to sequence number: The Body schema element was not found

**Basic**

Path to sequence number \*

Path to sequence group identifier

Missing message timeout

**Start of sequence definition**

Literal

Predicate

Automatic

**End of sequence definition**

Literal

Predicate

Automatic



# PHP Compute Node

## ▪ Was ist PHP?

- PHP ist eine dynamische Script-Sprache, die sich gut für Message Transformation eignet
- Weit verbreitete PHP Kenntnisse und Assets können für Message Broker genutzt werden
- Ausgezeichnete Ergänzung zu ESQL, Java, Mapping, XSLT und WTX Options

## ▪ PHPCompute node

- Erlaubt schnelle Formulierung von Message Transformationen

```
$message->a->b->c = $input_body->Message;
```

```
for ($index = 0; $index < $output_root->Menu->Food->count; $index++) {  
    $item = $output_root->Menu->Food[$index];  
}
```

## ▪ Integriertes Deployment

- Automatisches Deployment des PHP code
- PHP Development Toolkit 2.0 or 2.1 erforderlich (2.0 kann in MB Toolkit eingebettet werden)

# Hypervisor Edition V7.0

## ▪ Vereinfacht Bereitstellung von MB (and MQ)

Schnellere Bereitstellung von Broker-Infrastruktur



## Hyper Visor Edition Packages

- Vorkonfektioniertes VM Image für OS+HW Kombinationen:
  - RHEL 5.5 for VMWare ESX x86-64 initial release
  - Package enthält alle MQ & MB Komponenten
  - Enthält regular VMWare Image & WCA Image
- Image wird ktualisiert, wenn neue Fix Packs erscheinen

## Configuration Patterns & Scripts

- HVE Scenario Configuration Information
  - WCA Base Pattern mit Konfiguration Scripts
- Base WCA Pattern für die häufigsten MQ und MB Topologien
  - *Basic Broker, Advanced Broker, <User Pattern>*
- Script Packages konfigurieren Base Pattern
  - *Create Exec Group, Deploy BAR, Run MQSC, <User Script>*

## ▪ Full Function Deployed Configuration

- Interaktion mit MQ, MB Komponenten genau wie bei 'normalem' MQ , MB

