BPEL und Transaktionen

Referenten:
Guido Neander, Senior-Berater, MT AG, Ratingen
Arne Platzen, Leiter Competence Center „Oracle SOA“, MT AG, Ratingen

managing technology

MT AG Key Facts

ORACLE Platinum Partner

MT AG MANAGING TECHNOLOGY – ENABLING THE ADAPTIVE ENTERPRISE

- Gründung 1994
- Inhabergeführte AG:
  Aktienkapital 1.500.000 €
- Hauptsitz Ratingen;
  Niederlassung Dortmund
- Mitarbeiter:
  > 200 Festangestellte
  > 65 Freie Mitarbeiter

- Full-Service-Dienstleistung für alle Phasen des Software-Lifecycle
- Herstellerunabhängige Expertise in den marktführenden Technologien wie Oracle, IBM, Microsoft, SAP und OpenSource
- Themen- und Lösungs-Know-how in den Kerndisziplinen des Adaptive Enterprise

Agenda

- Allgemeines zu Transaktionen
- Notwendigkeit von Transaktionskonzepten im SOA-Umfeld
- Try-Cancel-Confirm
- BPEL-Compensation
- BPEL-Transaktionssteuerung
- Mediator-Transaktionssteuerung

Agenda

- BPEL und Fehlererkennung
- Transaktionsstandards und Protokolle
- Fazit

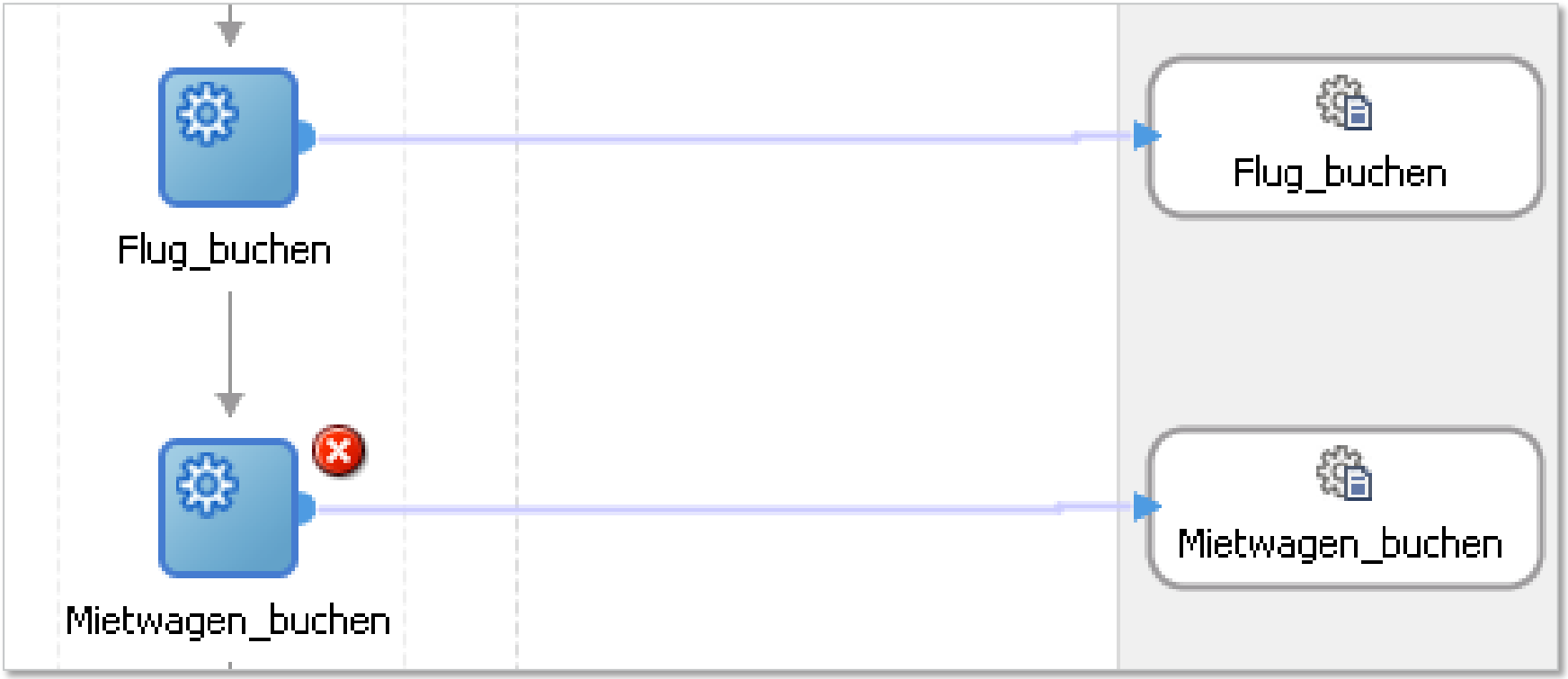
Einleitung

- Notwendigkeit von Transaktionen in der SOA-Welt?
- Integration von Transaktionskonzepten in die SOA-Welt?
- Alternativen bei Realisierung?
- Folgen für den SOA-Architekten?

Allgemeines zu Transaktionen

- Transaktion = mehrere Operationen, die eine logische Einheit bilden
- ACID
  - Atomacy = Transaktion muss vollständig ausgeführt werden
  - Consistency = Daten müssen hinterher konsistent sein
  - Isolation = Keine gegenseitige Beeinflussung
  - Durability = Ergebnis muss festgeschrieben sein
- Während der Transaktionsverarbeitung sind Teile des System gesperrt, um Inkonsistenzen zu vermeiden
- Verteilte Transaktion = Teile der Transaktion finden auf verschiedenen Systemen statt

Notwendigkeit von Transaktionskonzepten



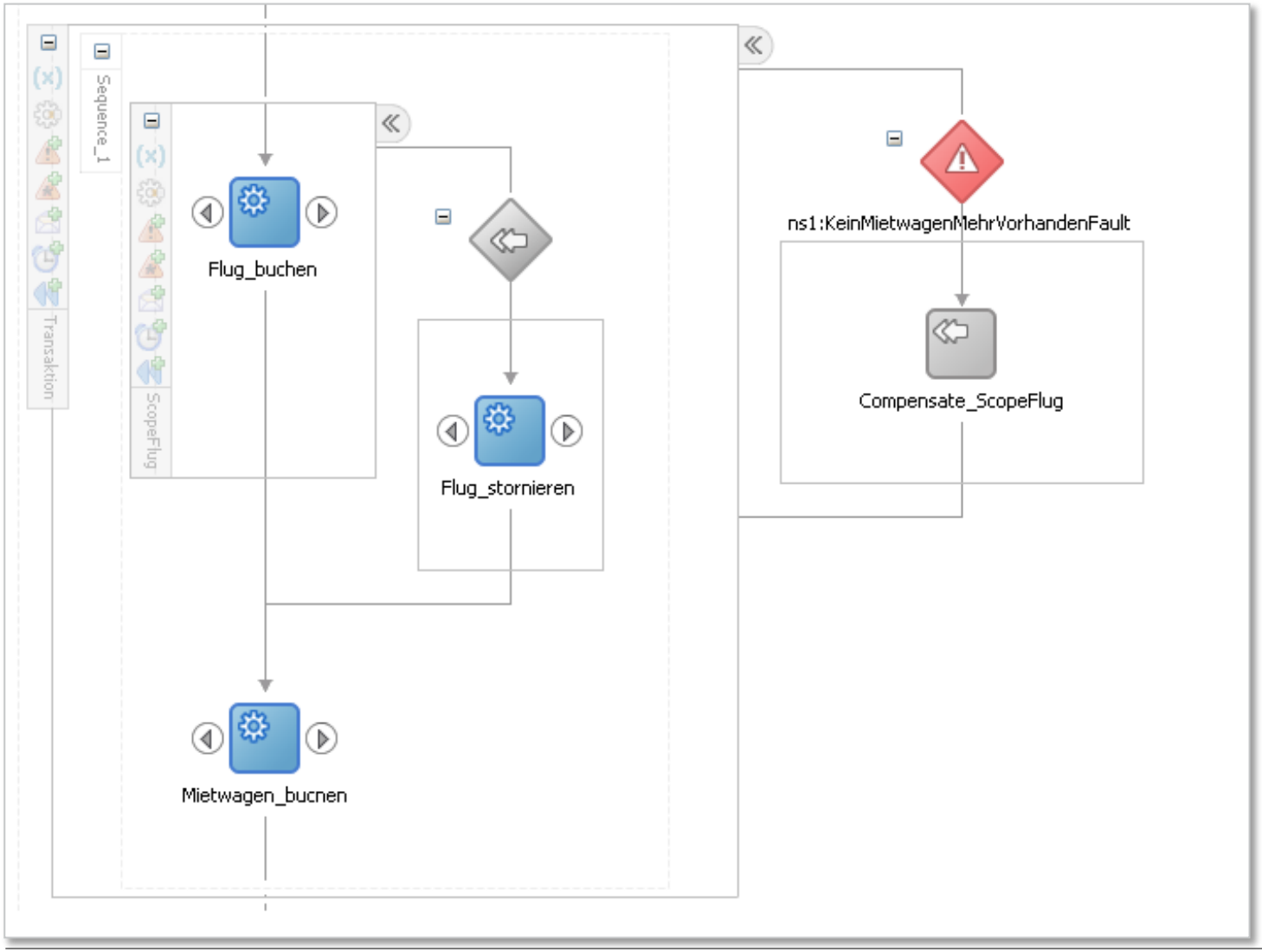
Try-Cancel-Confirm-Mechanismus

- Eine mögliche Lösung: Try-Cancel-Confirm
  - TRY: Aufruf eines Reservierungsservice
  - PENDING: System reserviert für eine gewisse Zeit die entsprechende Ressource
  - CONFIRM: Bei erfolgreichem Ablauf aller Operationen wird die Reservierung bestätigt
  - CANCEL: Wenn während der Verarbeitung ein Fehler aufgetreten ist, wird alles storniert
- Nachteil: Blocken der Ressource (Reservierung) nur für kurze Zeit möglich
- TRY-, CONFIRM- und CANCEL-Operationen müssen vom aufgerufenen Service unterstützt werden

Lösung in BPEL: KOMPENSATION

- Methoden, um bisher erzeugte Änderungen rückgängig zu machen
- Inverse Operation (oder Rollback) reicht häufig nicht aus
- „Logisches UNDO“ = Methode, welche fachliches Storno ausführt
- Z. B. Storno einer Bestellung in einem Fremdsystem, Versand einer Korrektur-E-Mail,…
- Stornomethode muss vom Service zur Verfügung gestellt werden

BPEL: Compensation Pattern

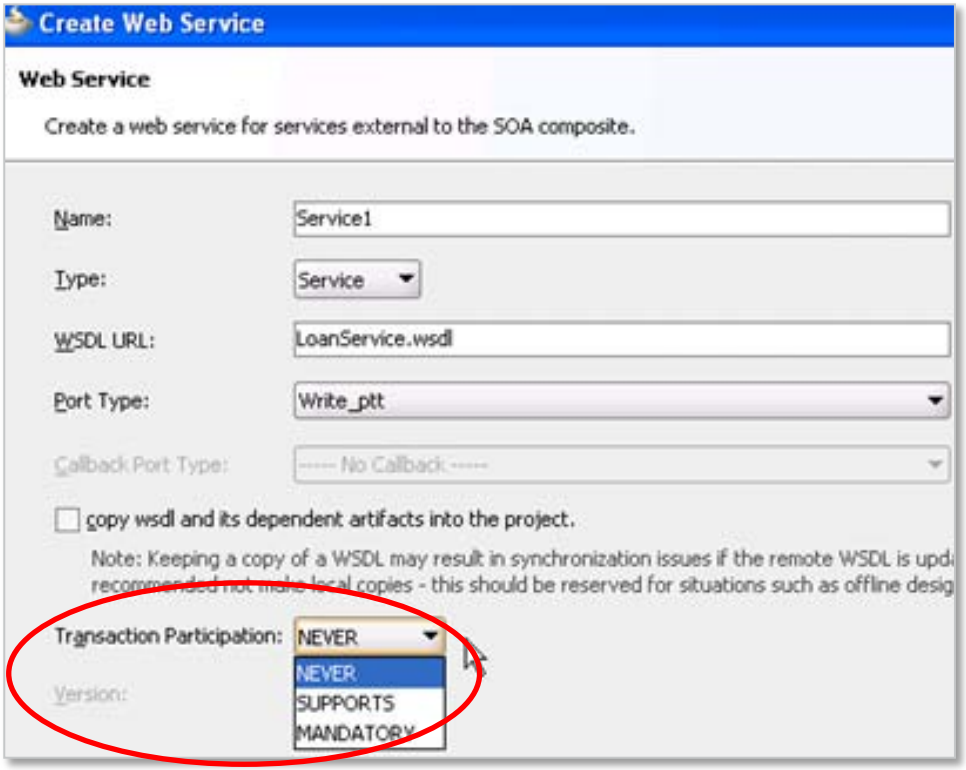


BPEL Transaktionssteuerung

- Transaktionssteuerung in BPEL oder einem Composite
  - Übernahme einer bestehenden übergeordneten Transaktion, wenn eine existiert
  - Ausführung in einer neuen Transaktion
- Transaktionssteuerung kann durch Properties erfolgen
- Teilweise verschiedene Properties in SOA Suite 10g und 11g
- Transaktionssteuerung kann durch Breakpoint-Aktivitäten erfolgen

BPEL Transaktionssteuerung

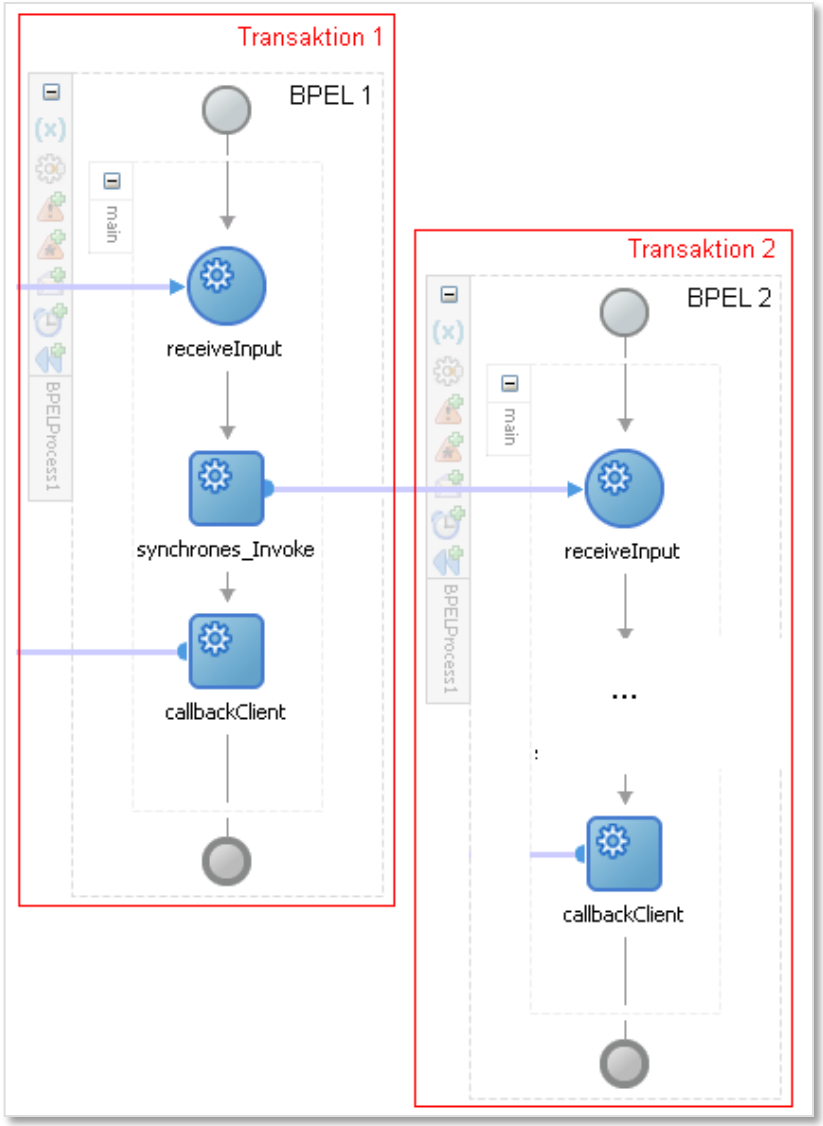
- Transaktionssteuerung in SOA Suite 11g: Properties im Composite
- „Exposed Services“ und „External References“: Property „Transaction Participation = Never/Supports/Mandatory/ WSDLDriven“
- BPEL-Komponente: Property „bpel.config.transaction= requiresNew/required“




```
<component name="InternalWarehouseService">
  <implementation.bpel src="InternalWarehouseService.bpel"/>
  <property name="bpel.config.transaction"
       many="false" type="xs:string">required | requiresNew</property>
```

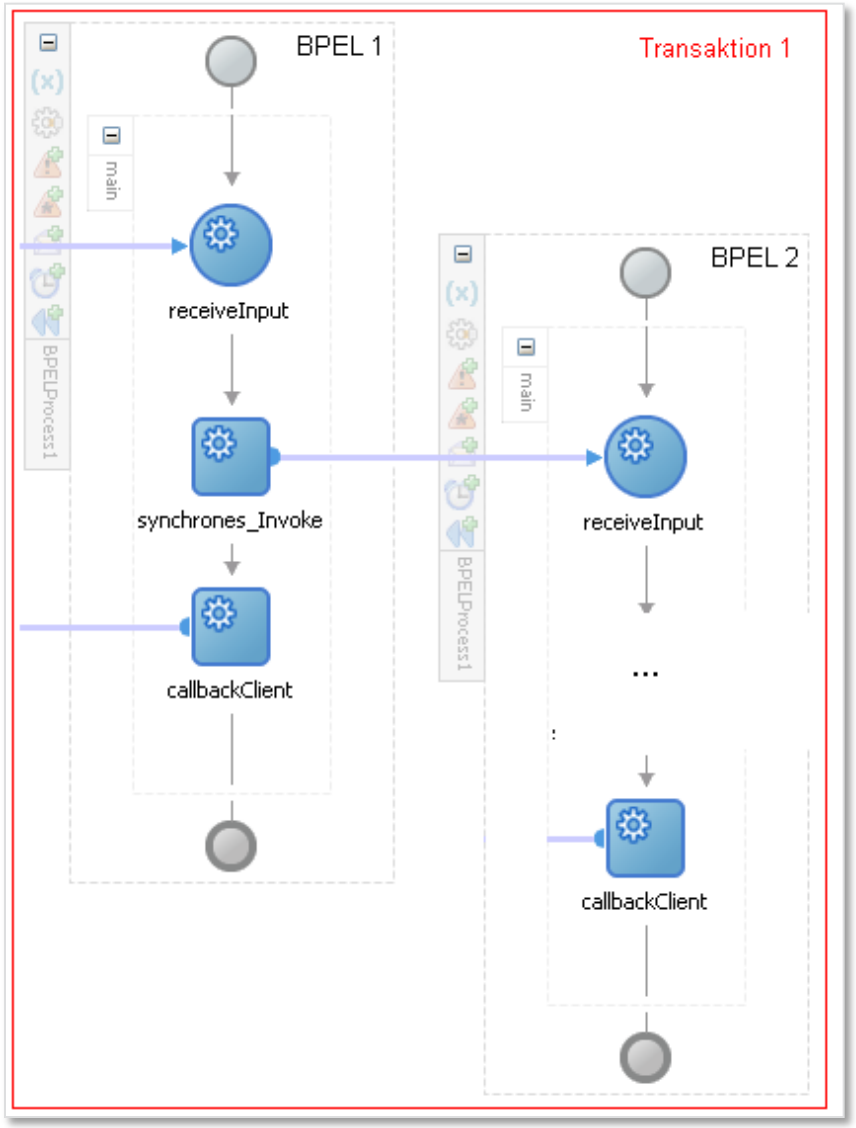
BPEL Transaktionssteuerung

- Synchrones Invoke ohne Transaktions-Properties



BPEL Transaktionssteuerung

- Synchrones Invoke
- Property: „bpel.config.transaction= required“



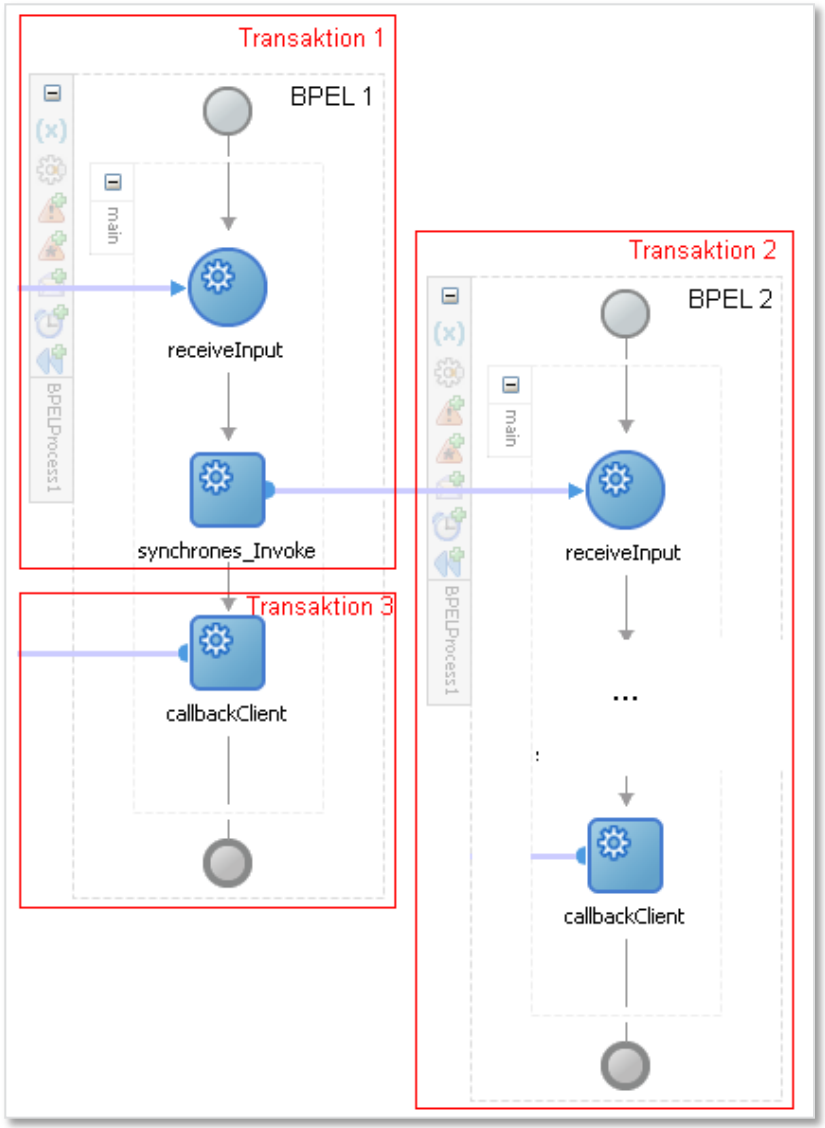
BPEL Transaktionssteuerung

- BPEL komplettiert die lokale Transaktion bei einer Breakpoint-Aktivität oder am Ende des Flusses (Dehydration)

- Breakpoint-Aktivitäten
  - Receive, es sei denn, es ist das 1. Receive und der Fluss ist Teil der übergeordneten Transaktion
  - onMessage
  - Wait, wird erst nach ein paar Sekunden commited
  - onAlarm
  - Invoke, wenn der Partner-Link non-idempotent ist
  - Ende des Flows, wenn der Prozess eine eigene Transaktion besitzt

BPEL Transaktionssteuerung

- Synchrones Invoke
- Partner link Property „idempotent = false“



BPEL Transaktionssteuerung

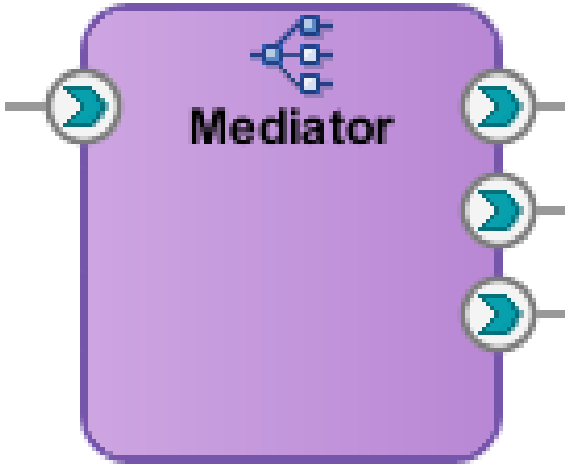
Activity	Transaction Status in BPEL Process	Transaction Status in Target Process or Adapter
Receive	New Transaction	N/A
Receive with Property "transaction=participate"	Use Existing Transaction from Caller	N/A
Invoke Synchronous Process	Use Existing Transaction	New Transaction
Invoke Synchronous Process with Partner Link Property "transaction=participate"	Use Existing Transaction	Use Existing BPEL Transaction
Invoke Synchronous Process with Partner Link Property "idempotent=false"	New Transaction	New Transaction
Invoke Synchronous Process with Partner Link Property "nonBlockingInvoke=true"	New Transaction	New Transaction
Invoke Asynchronous Process	Use Existing Transaction	New Transaction
Invoke Asynchronous Process with Partner Link Property "transaction=participate"	Use Existing Transaction	New Transaction
Invoke Synchronous Process with Partner Link Property "idempotent=false"	New Transaction	New Transaction
Wait < a couple of seconds	Use Existing Transaction	N/A
Wait > a couple of seconds	New Transaction	N/A
Flow	Use Existing Transaction	

Agenda

- Mediator-Transaktionssteuerung
- BPEL und Fehlererkennung
- Transaktionsstandards und Protokolle
- Fazit

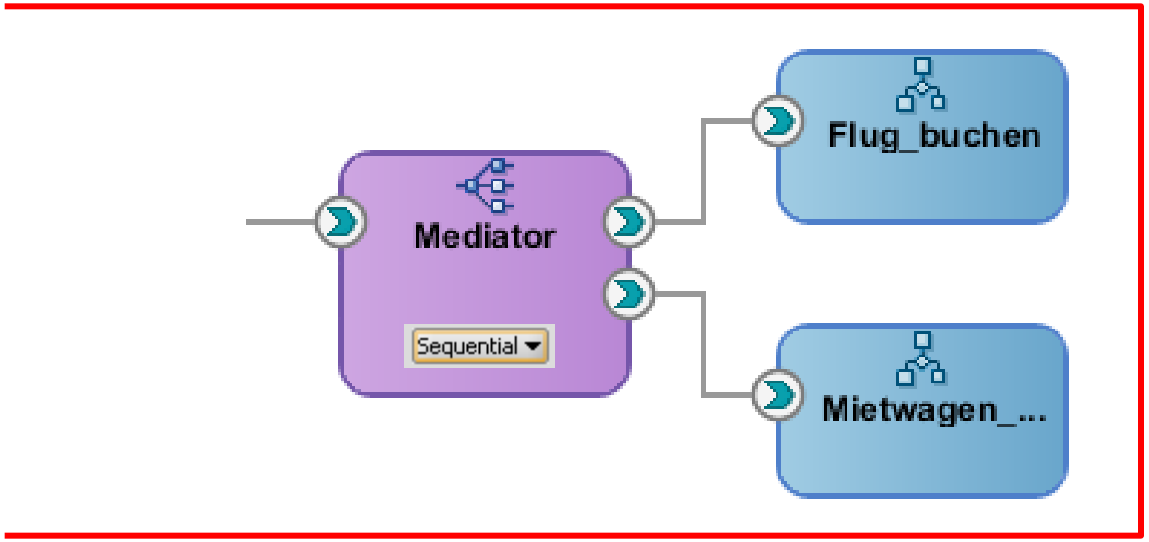
Mediator Transaktionssteuerung

- Bei externem Aufruf des Mediators wird eine neue Transaktion erzeugt
- Steht die Transaktion des aufrufenden Prozesses zur Verfügung wird diese vom Mediator verwendet



Mediator Transaktionssteuerung

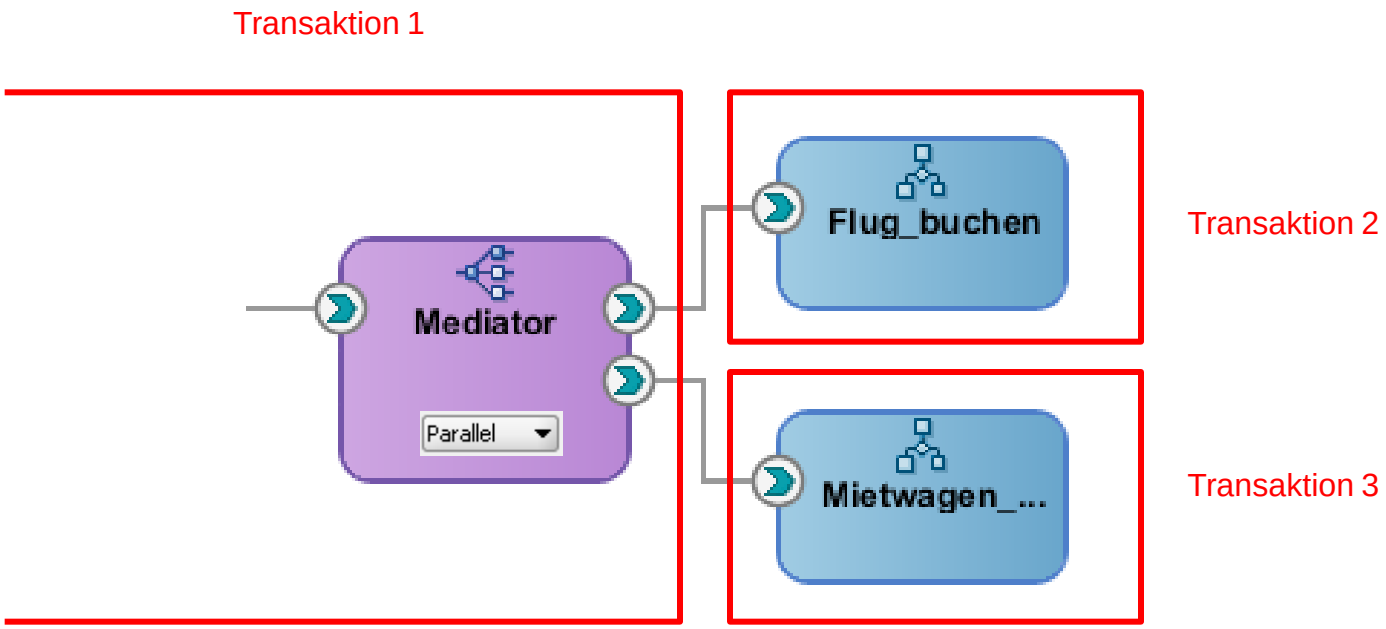
- Bei **sequentiellen** (synchrone) Routing-Regeln werden alle Regeln in einer Transaktion ausgeführt
- Im Fehlerfall erfolgt ein Rollback aller Routings
- Fault-Policies des Mediators werden nicht beachtet



eine Transaktion

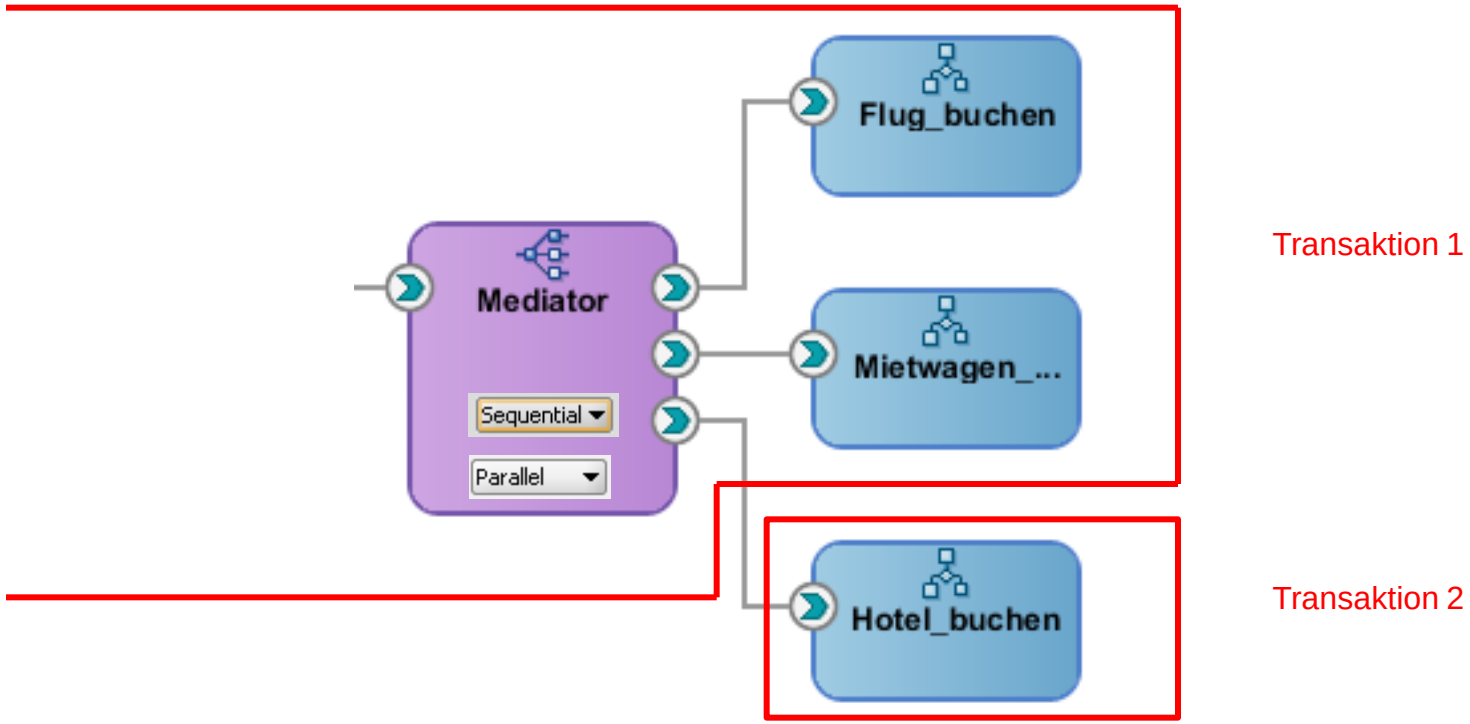
Mediator Transaktionssteuerung

- Bei **parallelen** Routing-Regeln wird für jede Regel eine neue Transaktion verwendet
- Im Fehlerfall erfolgt ein Rollback
- Fault-Policies können verwendet werden, um auf Fehler zu reagieren



Mediator Transaktionssteuerung

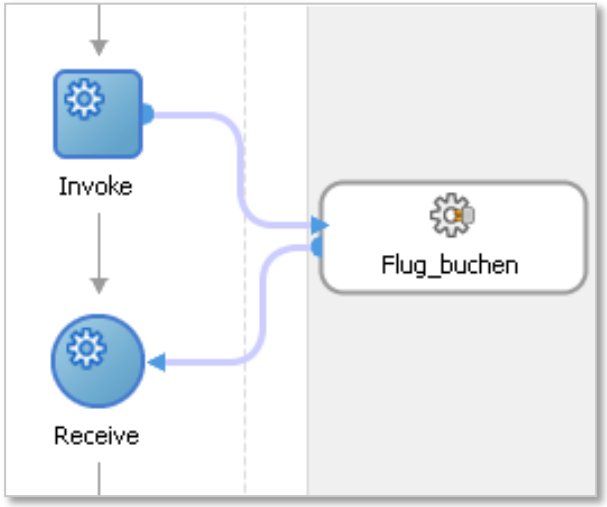
- **sequentielle** Routing-Regeln werden **zuerst** in einer Transaktion ausgeführt
- **parallele** Regeln werden **danach** in jeweils einer eigenen Transaktion durchgeführt.



BPEL und Fehlererkennung

Um eine Kompensation auszulösen, muss jeder Service im Fehlerfall einen Fault zurückliefern.

- Bei synchronen Services werden Fehler standardmäßig zurückgeliefert
- Bei asynchronen Services wird kein Fault zurückgeliefert
  - Ein „Receive Elements“ wartet in einer Endlosschleife auf das Ergebnis eines Service



BPEL und Fehlererkennung

Pick Element

Transaktion 1

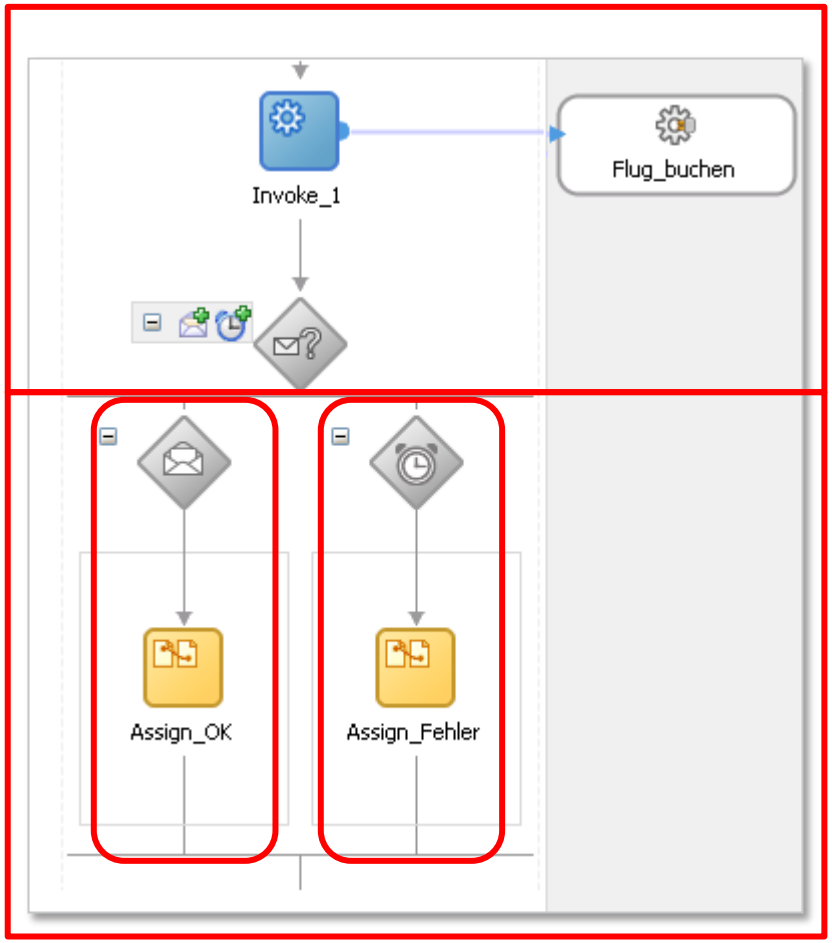
Das Pick-Element beinhaltet die beiden Fälle „on Message“ und „on Alarm“.

Es wartet bis der aufgerufene asynchrone Prozess

Transaktion 2

- ein Ergebnis liefert (on Message) oder
- die maximale Wartezeit (on Alarm)

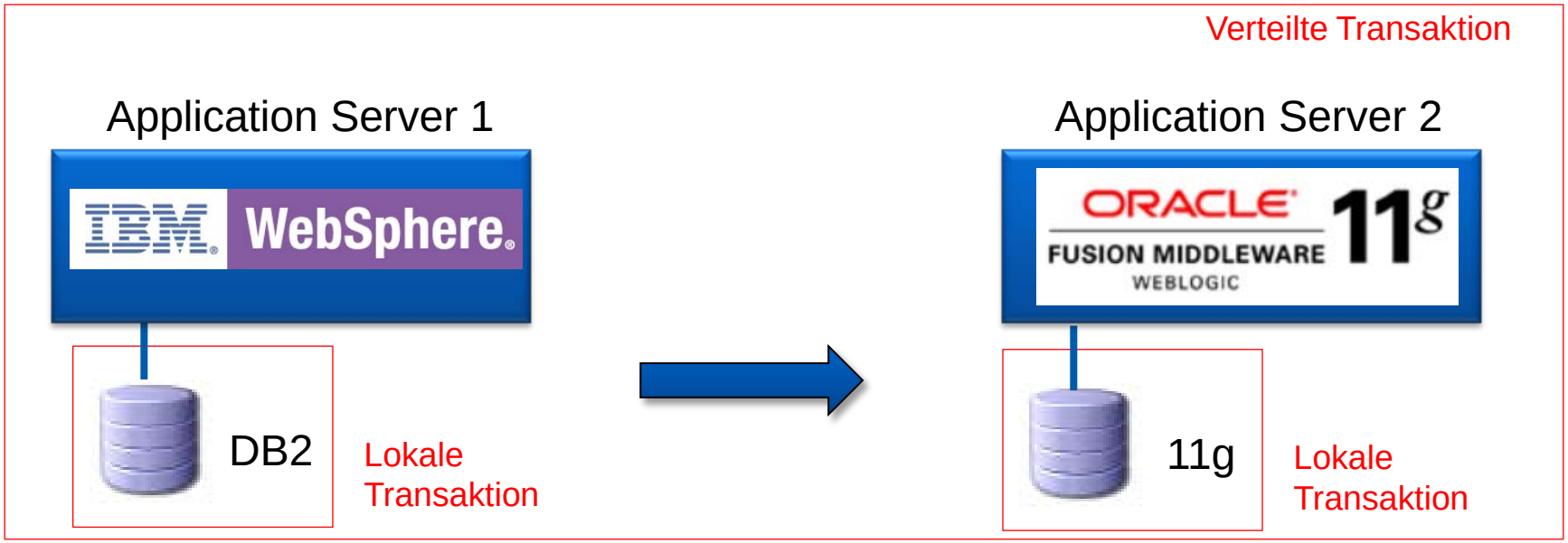
erreicht wurde und führt dann den entsprechenden Block aus.



Transaktionsstandards und -protokolle

- Die Ausführung von verteilten Transaktionen stellt andere Anforderungen an die Transaktionsumgebung

➡ Services müssen in heterogenen Systemen ausgeführt und verwaltet werden



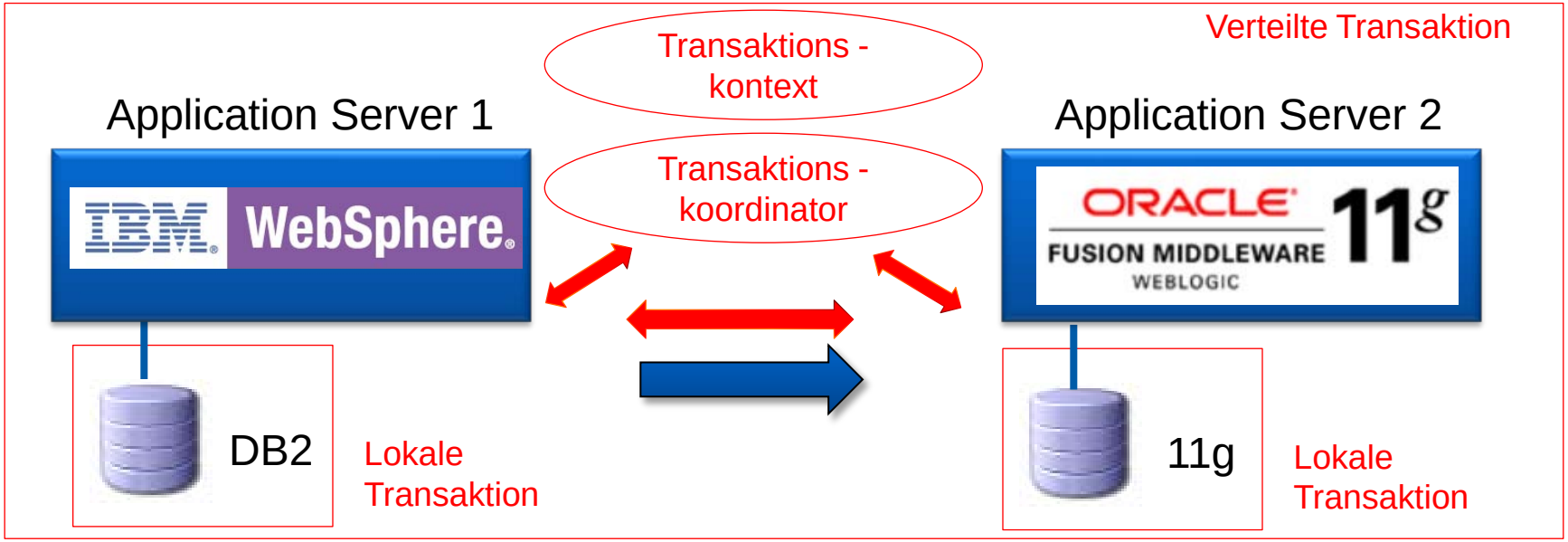
Transaktionsstandards und -protokolle

- Die Transaktionsstandards basieren meist auf dem Koordinator-Teilnehmer-Modell

- Transaktionsstandards lassen sich bezüglich der Länge der von ihnen verwalteten Transaktionen differenzieren
  - Einige Standards eignen sich für kurzlebige Transaktionen, weil Ressourcen während der Verarbeitung geblockt werden
  - Andere finden bei langlebigen Transaktionen Anwendung

Transaktionsstandards und -protokolle

- Identifizierung einer Transaktion durch den Transaktionskontext
- Transaktionskontext muss für die Dauer der Transaktion verwaltet und allen beteiligten Systemen kommuniziert werden
  - Transaktionsstandards und -protokolle
  - Fehlersituationen dürfen nicht zu systemübergreifenden Inkonsistenzen führen



Transaktionsstandards und -protokolle

Transaktionsstandard: Web Service Atomic Transaktion (WS-AT)

- Mit WS-AT können die ACID-Eigenschaften über mehrere verteilte Services auf verschiedenen Applikationsservern erfüllt werden.
- Da die Ressourcen während der Ausführung geblockt werden, ist WS-AT nur für kurz laufende Prozesse geeignet.
- WS-AT wird von verschiedenen Middleware-Herstellern unterstützt

Transaktionsstandards und -protokolle

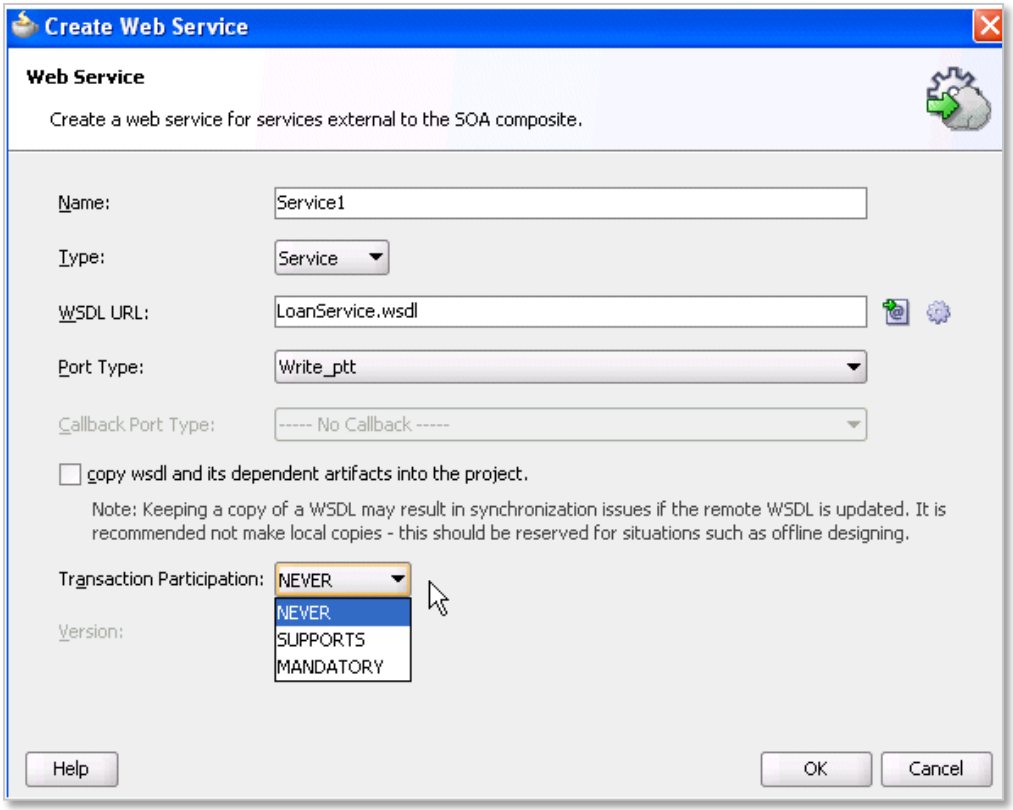
Web Service Atomic Transaktion (WS-AT) mit der Oracle SOA Suite

- Der Weblogic Server 11gR1 (10.3.3) und die Oracle SOA Suite 11g R1 PS2 (11.1.1.3) unterstützt den Standard WS-AT in der Version 1.2.
- Um WS-AT auf einem Weblogic Server nutzen zu können, muss die Domain für WS-AT konfiguriert sein.
- In den BPEL Prozessen ist über Properties ein Transaktionsstart zu setzen und bei den Referenzen ist WS-AT zu aktivieren.
- Beim Aufruf von externen Referenzen ist die Transaktionsteilnahme zu übergeben.

Transaktionsstandards und -protokolle

Transaction Participation

- Never
  Der Service läuft in einer eigenen Transaktion.
- Supports
  Der Service nimmt an der Transaktion teil.
- Mandatory
  Der Service nimmt an der Transaktion teil.
  Wenn beim Aufruf keine Transaktion vorliegt, wird eine Exception Message/ Fault geworfen.



Transaktionsstandards und -protokolle

Umsetzung einer verteilten Transaktion mit Web Services Atomic Transaction (WS-AT)

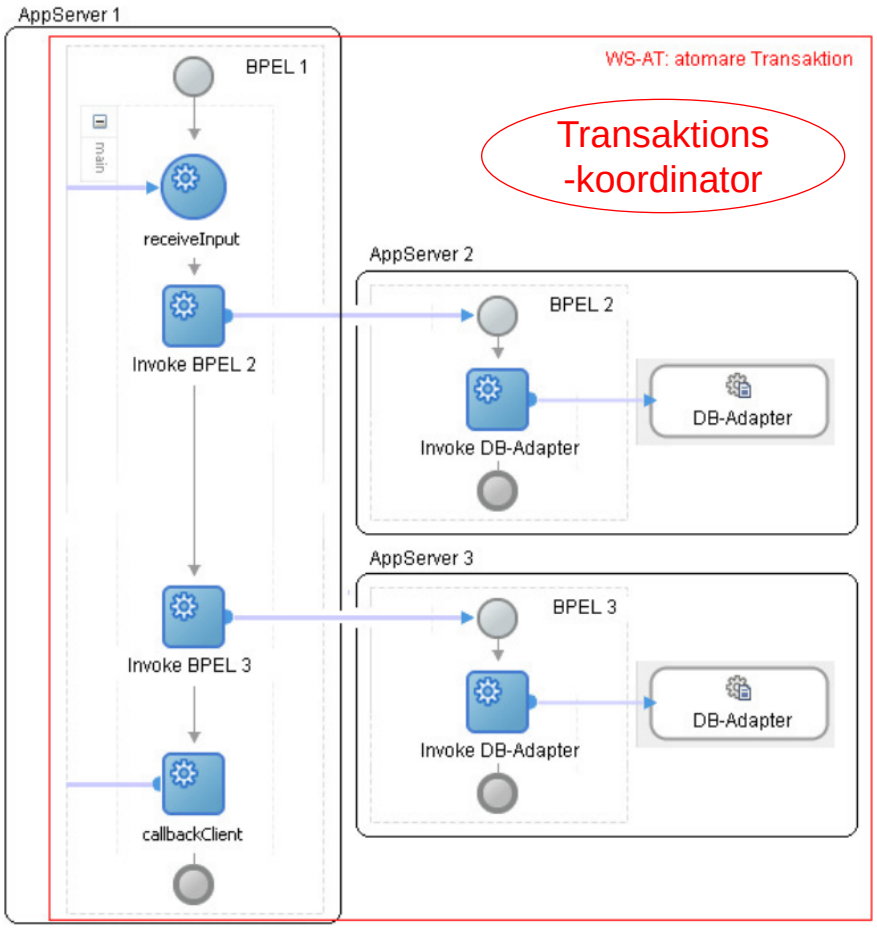
Teil 1: Aktivierung und Registrierung

Transaktionskontext: 4711
Beteiligte Services:

BPEL1 (Initiator)

BPEL2

BPEL3



Transaktionsstandards und -protokolle

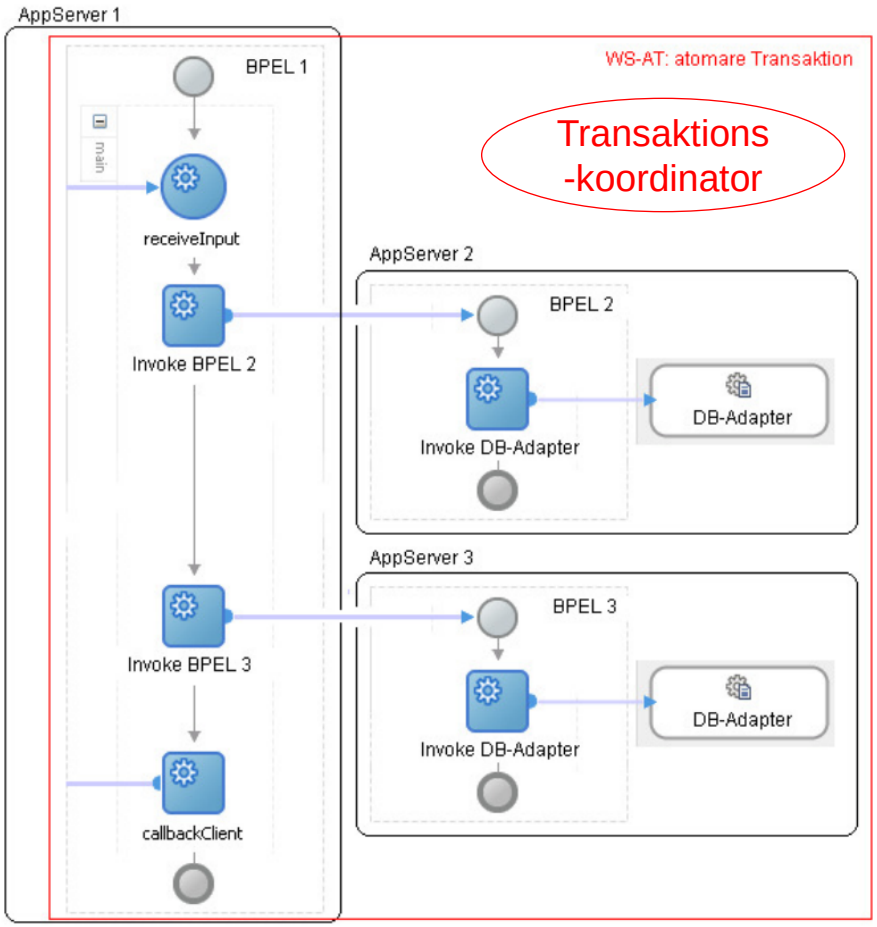
Umsetzung einer verteilten Transaktion mit Web Services Atomic Transaction (WS-AT)

Teil 2: Koordination (2PC)

Transaktionskontext: 4711
Beteiligte Services:

BPEL1 (Initiator) Befehl: Commit

	Phase 1 (prepare)	Phase 2 (commit)
BPEL2	prepared	commited
BPEL3	prepared	commited
	OK	commited



Transaktionsstandards und -protokolle

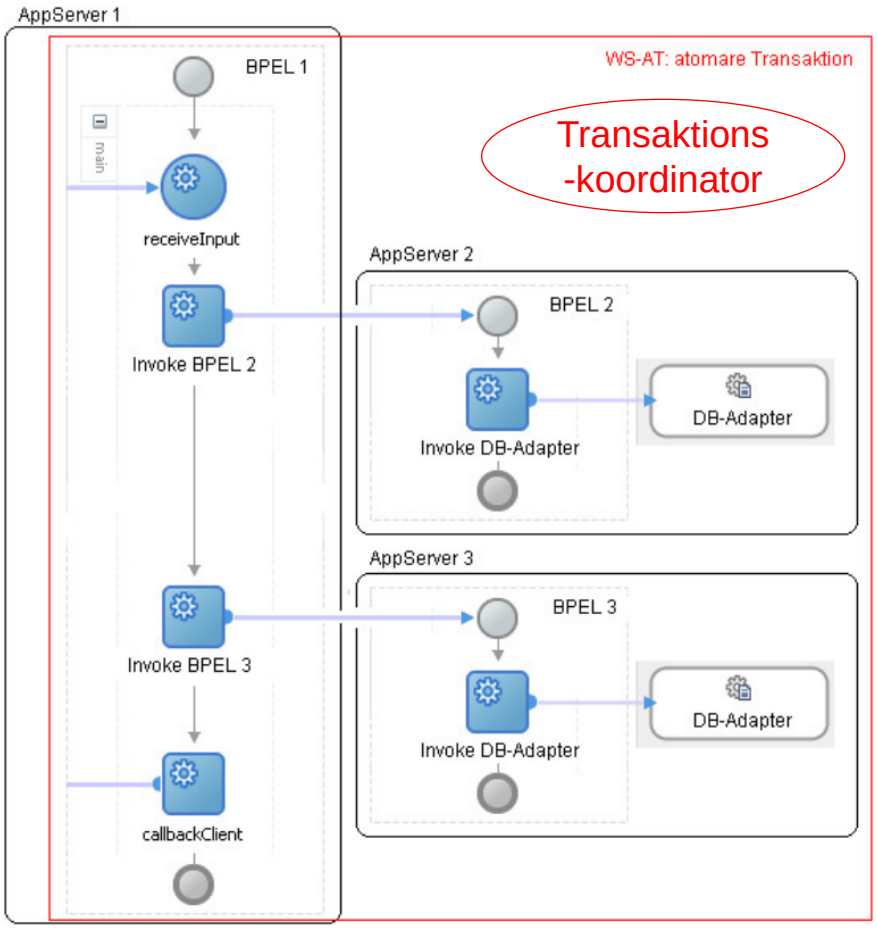
Umsetzung einer verteilten Transaktion mit Web Services Atomic Transaction (WS-AT)

Teil 2: Two-Phase Commit

Transaktionskontext: 4711
Beteiligte Services:

BPEL1 (Initiator) Befehl: Commit

	Phase 1 (prepare)	Phase 2 (Rollback)
BPEL2	prepared	aborted
BPEL3	fault	aborted
	fault	aborted



Weitere Fehlermöglichkeiten

- Systemfehler
- Netzwerkfehler
- Deadlocks
- Hardwarefehler
- Protokollfehler
- Timeout

Transaktionsstandards und -protokolle

Fehlerbehandlung bei Ausfall des Koordinators

- Vor dem Senden der Prepare-Nachricht sind alle Teilnehmer zu protokollieren.
- Loggen aller Antworten der Teilnehmer.
- Anhand der Logs kann der Koordinator wieder an dem Punkt aufsetzen, wo er aufgehört hat.

Fehlerbehandlung bei Ausfall eines Teilnehmers

- Loggen der Anforderungen.
- Loggen der Antworten.
- Anhand der Logs kann der Teilnehmer wieder an dem Punkt aufsetzen, wo er aufgehört hat.

Transaktionsstandards und -protokolle

Zusammenfassung

- Mit WS-AT können die ACID-Eigenschaften über mehrere verteilte Services auf verschiedenen Applikationsservern erfüllt werden.
- Alle Teilnehmer müssen WS-AT unterstützen
- Verteilte Transaktion sollten so einfach wie möglich gestaltet werden.
- Eine über mehrere, heterogene Systeme und Produkte verteilte Transaktion ist häufig komplex und fehleranfällig.
- Vorrangig sollten immer lokale Transaktionen verwendet werden, sofern dies möglich ist.

BPEL und Transaktionen – Fazit

 # 1

Es ist frühzeitig in einem Projekt zu entscheiden, ob eine Transaktionsverarbeitung notwendig ist, um die Transaktionsumsetzung bereits in einem frühen Stadium mit in die Planung einbeziehen zu können.

 # 2

Eine einfache Transaktionsrealisierung in BPEL mit Kompensation und fachlichem Storno lässt sich häufig sehr leicht integrieren.

 # 3

Kompliziertere Szenarien, die Reservierungsmechanismen oder Transaktionsstandards beinhalten, haben meist einen höheren Realisierungsaufwand zur Folge.

Vielen Dank!

?!

MT AG managing technology I Balcke-Dürr-Allee 9 I 40882 Ratingen
Tel. +49 (0) 2102 309 61-0 I info@mt-ag.com I www.mt-ag.com

managing technology

MT AG – Vorträge auf der DOAG

ORACLE Platinum Partner

MT AG MANAGING TECHNOLOGY – ENABLING THE ADAPTIVE ENTERPRISE

Di. 16. Nov.	10:00 – 10:45	Clevere Web-Formulare mit APEX und jQuery	Andreas Wismann
Di. 16. Nov.	13:00 – 13:45	Oracle RMAN – beim Recovery das Disaster erleben?	Volker Mach
Di. 16. Nov.	13:00 – 13:45	Rollout Prozess für APEX Anwendungen	Oliver Lemm
Mi. 17. Nov.	13:00 – 13:45	Audit Vault - Erfahrungen aus der ersten deutschen Produktivumgebung	Volker Mach
Do. 18. Nov.	13:00 – 13:45	Das APEX Migrationsprojekt bei der Union Investment	Niels de Bruijn
Do. 18. Nov.	16:00 – 16:45	BPEL und Transaktionen	Arne Platzen Guido Neander
Stand-by		Rich-Internet-Applications mit jQuery und dem APEX Listener	Klaus Friemelt