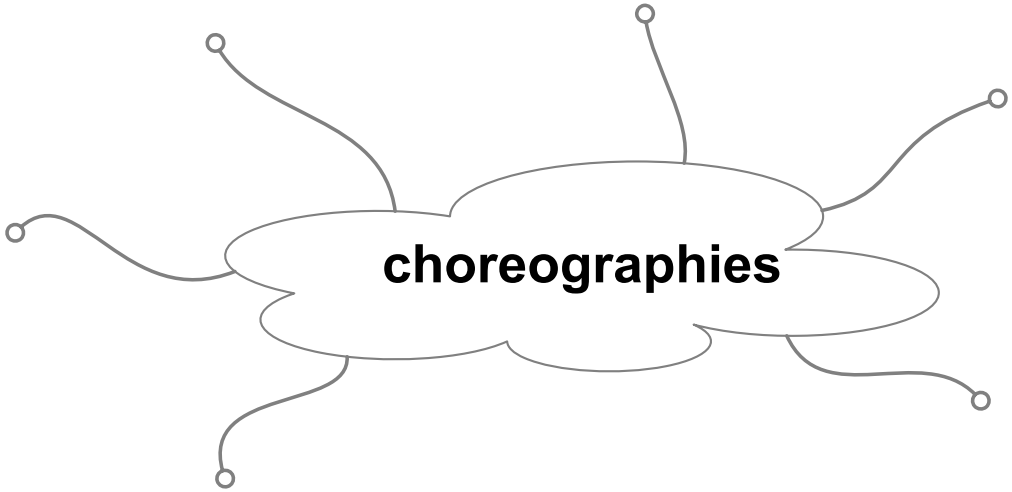
Kapitel 8: Choreographie

Choreographie und Beschreibungssprachen

Agenda

- **Einführung**
- Let‘s dance
- CDL
- Zusammenfassung

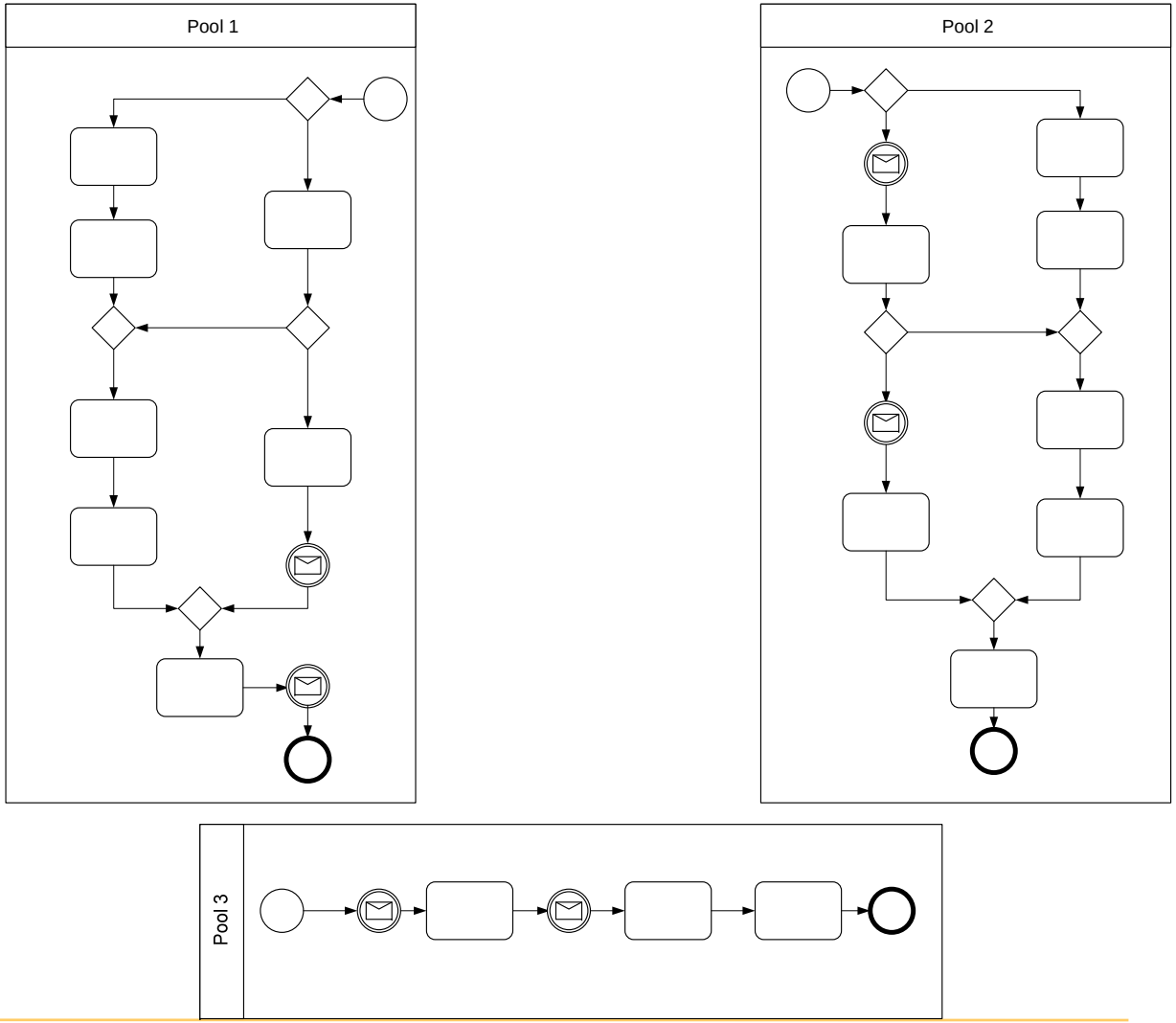
Was sind Choreographien?



- Choreographies = global interaction models

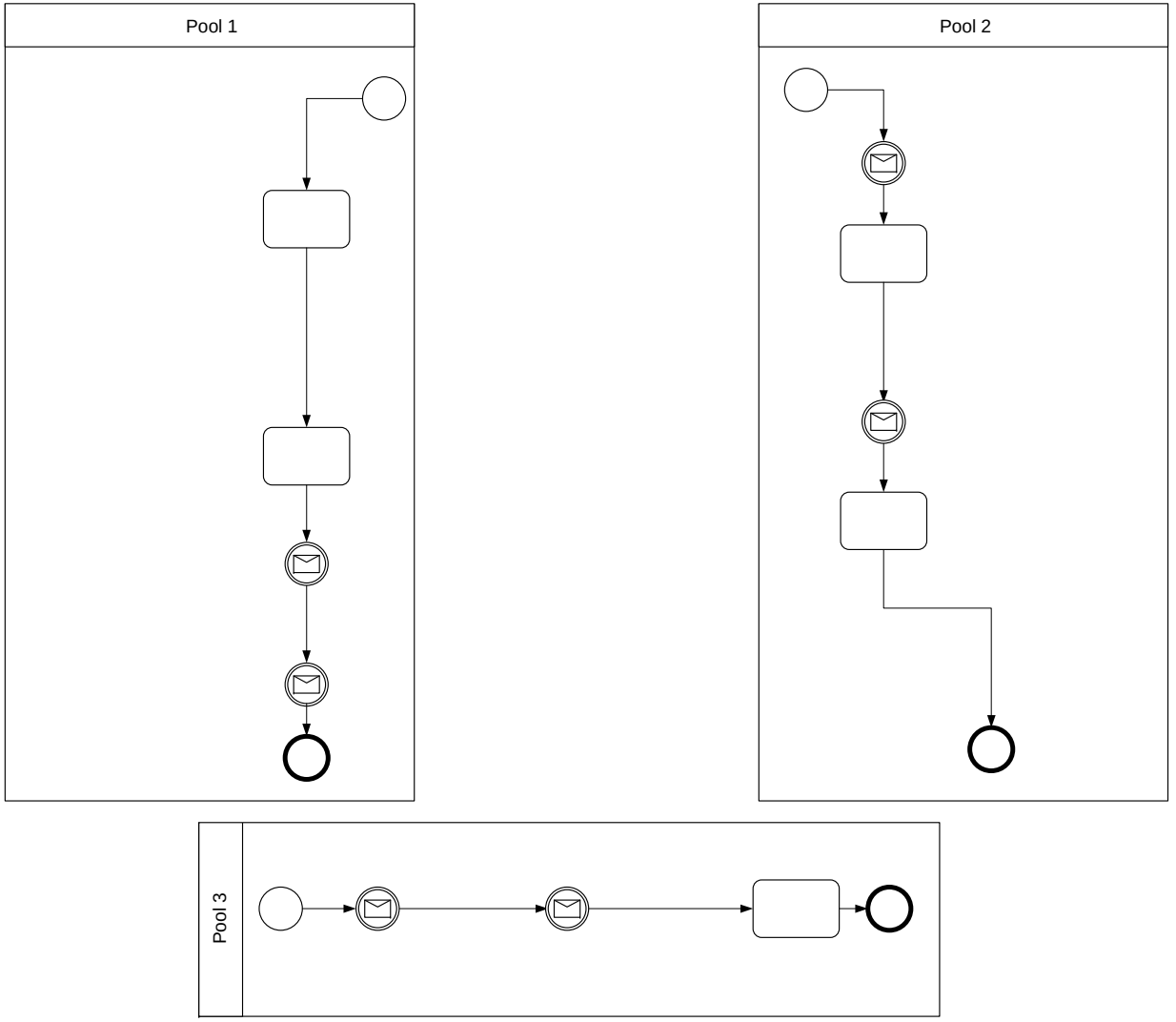
Choreographie vs. Orchestrierung
Public vs. Private View

- Private Views:
  - Interne Details



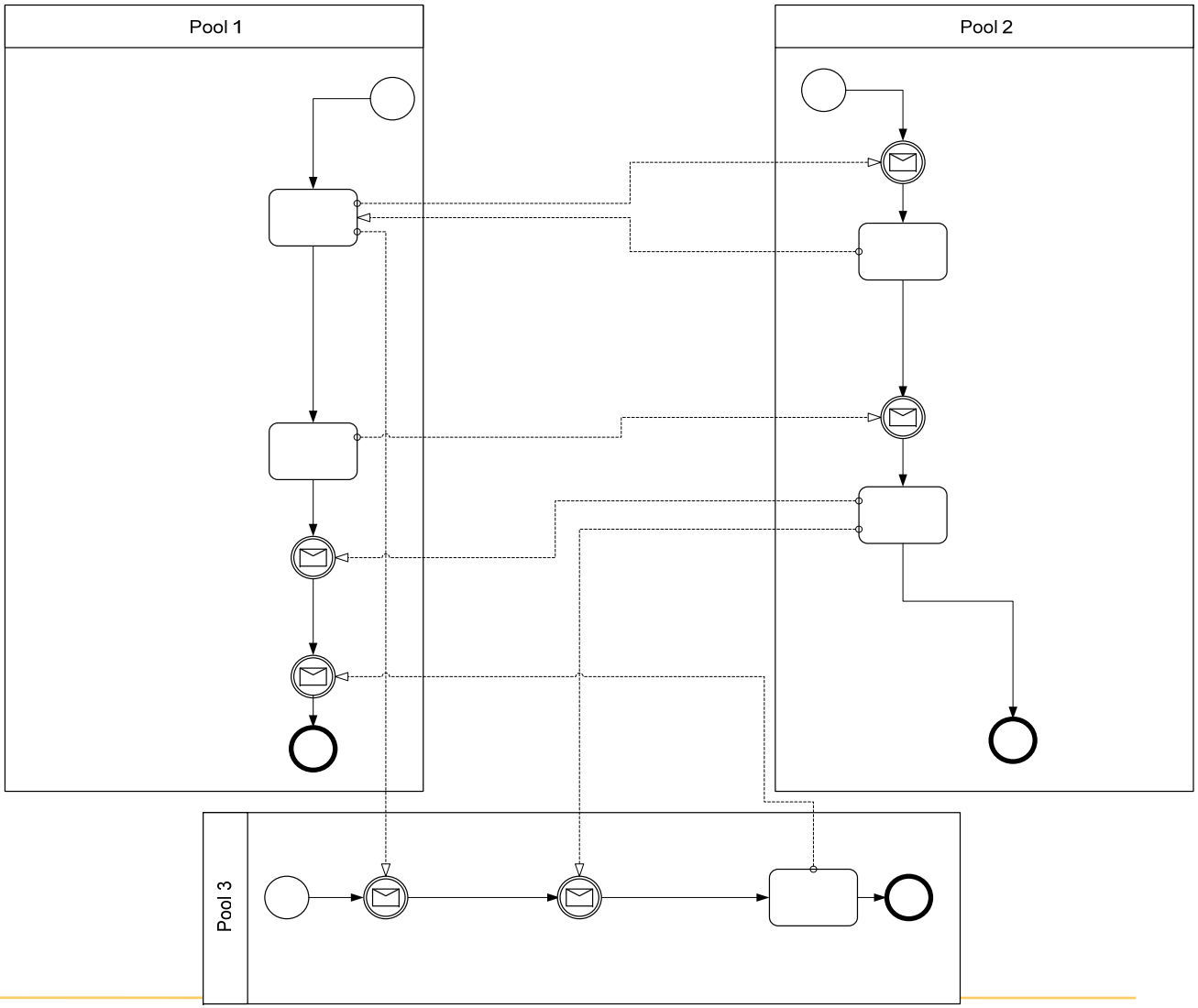
Choreographie vs. Orchestrierung
Public vs. Private View

- Public Views:
  - Keine internen Details
  - Nur Schnittstellen an die Partner
  - Entspricht *Abstract BPEL*



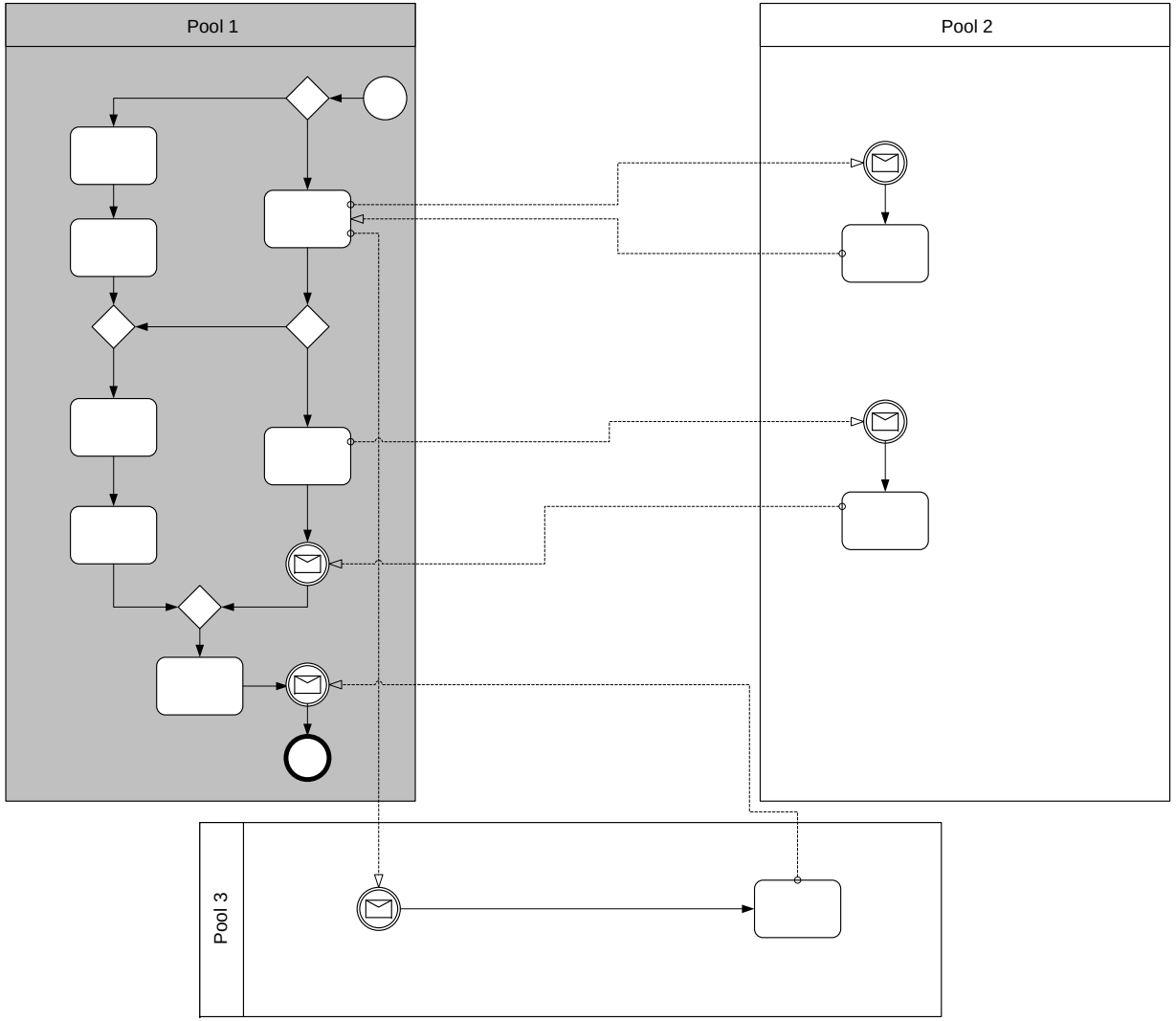
Choreographie vs. Orchestrierung
Public vs. Private View

- Choreographie:
  - Globale Sicht auf die Public Views



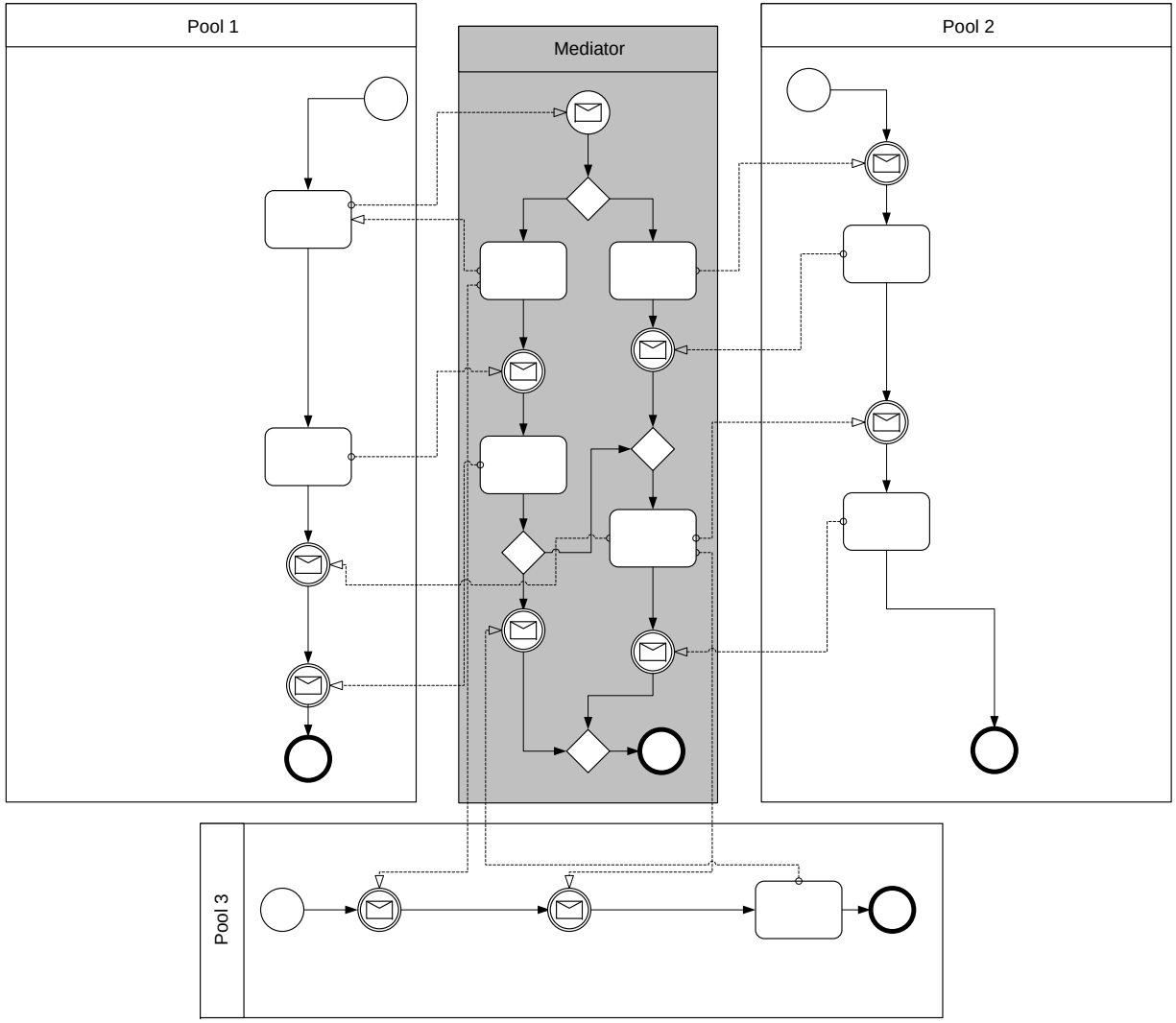
Choreographie vs. Orchestrierung
Public vs. Private View

- Orchestrierung:
  - Fast wie Private Views
  - Aber auf Web Services bezogen
  - Ausserdem Verbindung zu den Public Views der anderen Prozesse
  - Entspricht *Executable BPEL*



Choreographie vs. Orchestrierung
Public vs. Private View (1)

- Orchestrierung
- Kann auch als Mediator-Prozess eingesetzt werden

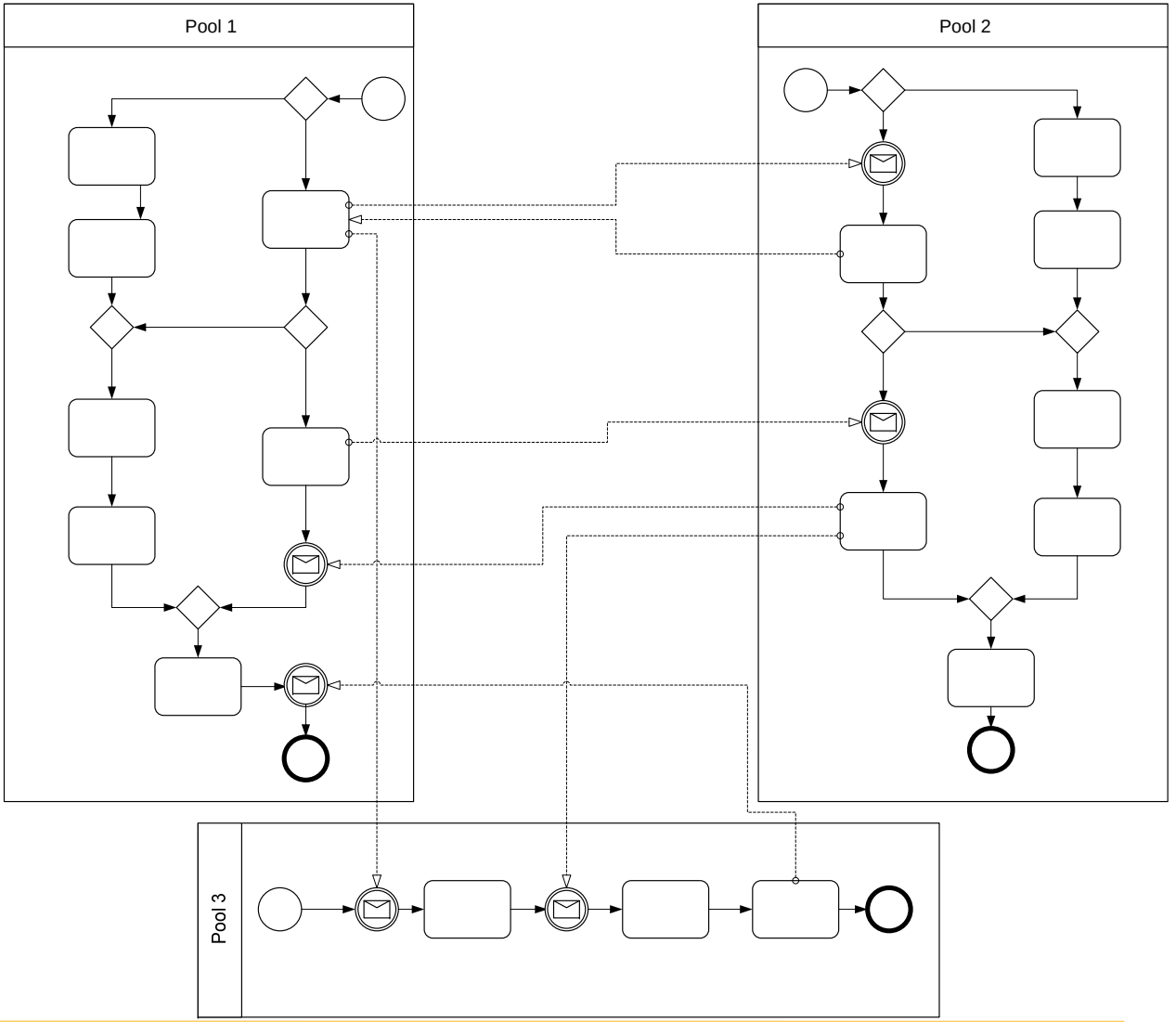


Choreographie vs. Orchestrierung
Public vs. Private View (2)

- Orchestrierung:
  - Fast wie Private Views
  - Aber auf Web Services bezogen
  - Ausserdem Verbindung zu den Public Views der anderen Prozesse
  - Entspricht *Executable BPEL*
- Kann auch als Mediator-Prozess eingesetzt werden
  - Vermittelt zwischen verschiedenen Public Views
  - Alternative zu Choreographien mit Trusted Third Party

Choreographie vs. Orchestrierung
Public vs. Private View

- "Kollaborativer Gesamtprozess":
  - Die Kombination aller Sichtweisen
  - Mit Mediator – sofern vorhanden
  - Entspricht ungefähr der Kombination mehrerer Prozesse in *Executable BPEL*



Sprachen / Notationen für Choreographien

- Grafische Modellierung:
  - BPMN
  - Let's Dance
  - …
- XML-Sprachen:
  - WS-CDL
  - WSCI
  - BPSS
  - (WSCL)
  - (BPEL)

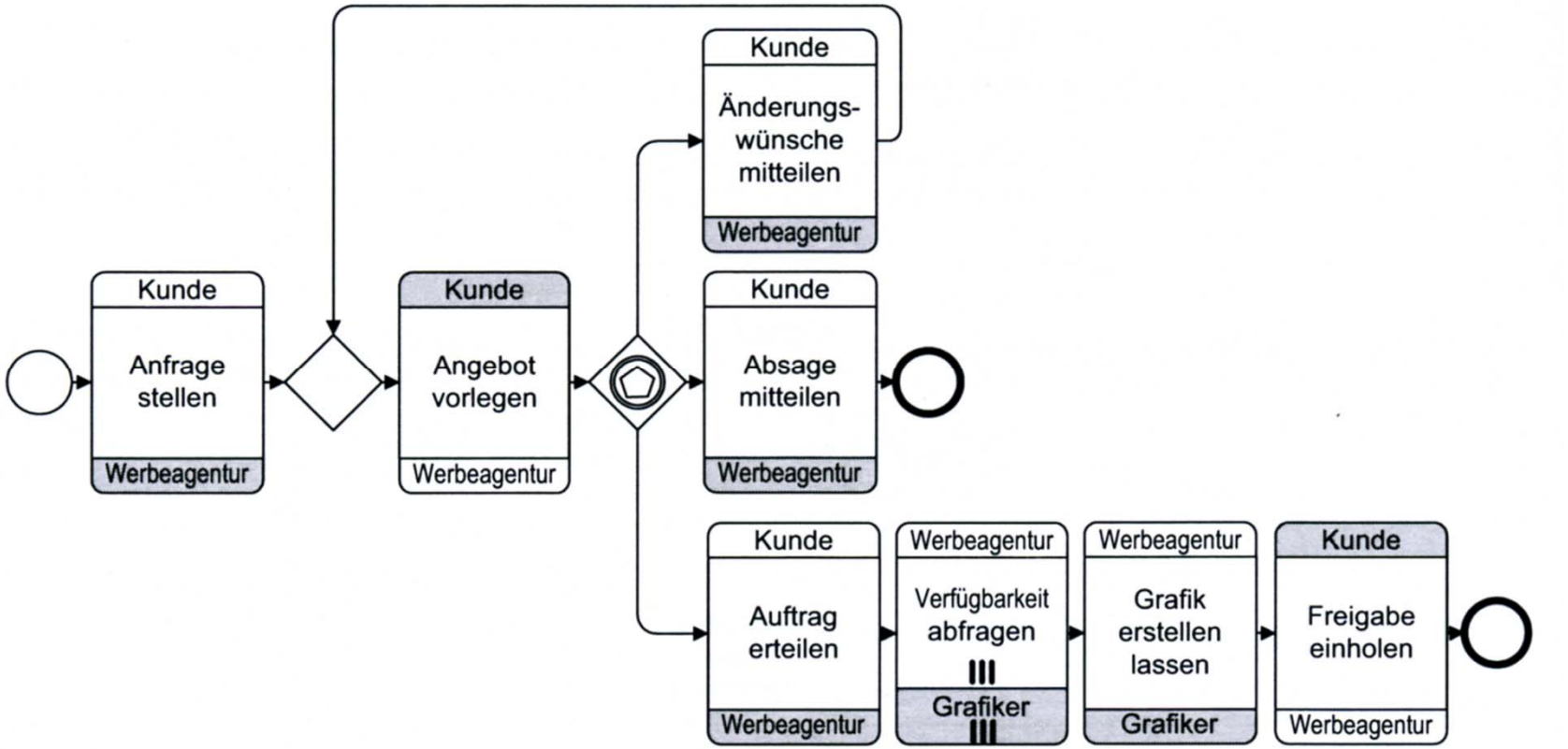
→Queensland Univ. of Tech & SAP Research Brisbane

→W3C Choreography Working Group W3C Working Note

Wdh. aus Kapitel 8:
Choreographie in BPMN

- Motivation: Synchronisation der Kommunikation von Prozessen
- Problematik: aus Black-Box Darstellung von (externen, unternehmensübergreifenden) Prozessen sind zeitliche Abfolge und Prozesslogik nicht notwendigerweise ersichtlich (z.B. Problematik von Schleifen)
- Ziel: Darstellung der Sequenz der auszutauschenden Nachrichten zwischen Prozessbeteiligten
- Unabhängige Prozesse mit Fokus auf Nachrichtenaustausch, Verwendung der Prozessnotation (z.B. Sequenzfluss, Gateways)
- Choreographie-Aktivität: Austausch einer oder mehrerer Nachrichten
  - Darstellung der beteiligten Partner
  - Hell: (initialer) sendender Partner, dunkel: (initial) empfangender Partner
  - Bidirektionale Nachrichten möglich

Wdh. aus Kapitel 8:
Beispiel Choreographiediagramm

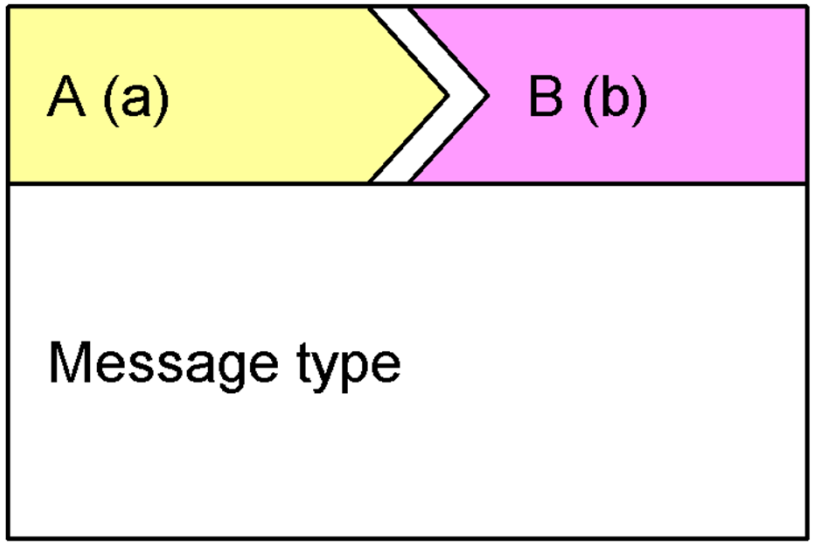


Agenda

- Einführung
- **Let‘s dance**
- CDL
- Zusammenfassung

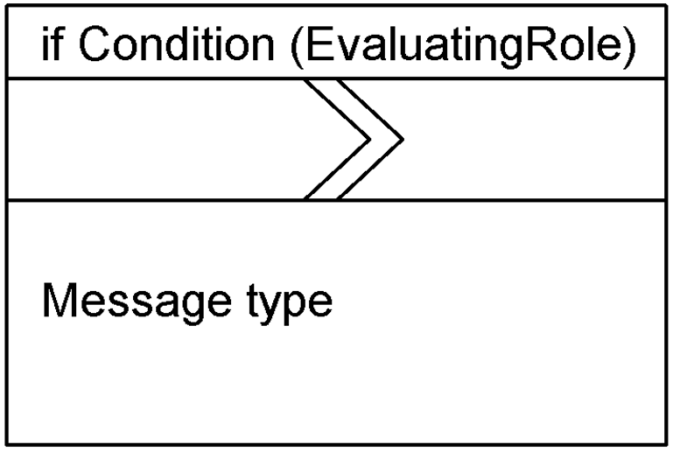
Let's Dance: Konstrukte (1)

- "Let's Dance is a language for modeling service interactions and their flow dependencies."
- Beschreibt den Kontroll- und Datenfluss einer Choreographie in grafischer Notation
  - Mehrere Parteien (Rollen) tauschen Nachrichten aus, um ein gemeinsames Ziel zu erreichen
    - Rollen können von Aktoren gespielt werden
    - Beispiel: Aktor a spielt Rolle A und schickt eine Nachricht an Aktor b, der Rolle B spielt.

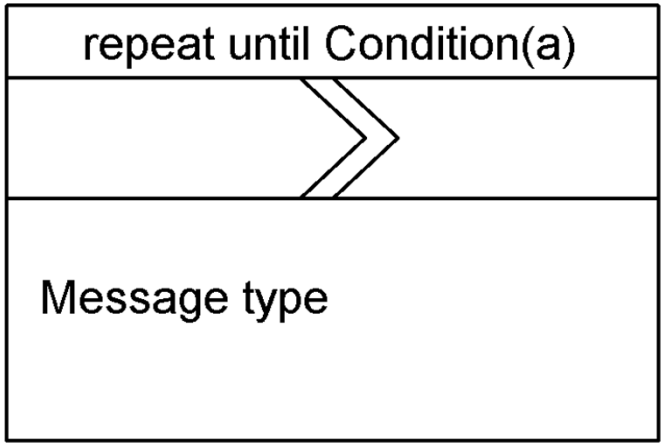


Let's Dance: Konstrukte (2)

- Kontrollfluss: explizit
  - *"Guard" (**Fallunterscheidung**):* Nur wenn eine bestimmte Bedingung erfüllt ist, wird X ausgeführt.



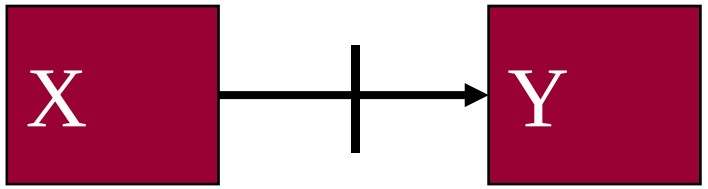
  - *"Repeat" (**Schleife**):* Solange eine Bedingung erfüllt ist wird X wiederholt



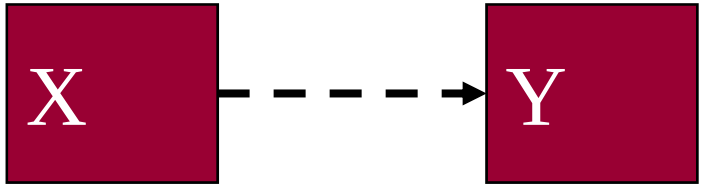
Let's Dance: Konstrukte (3)

- **Kontrollfluss: explizit**
  - *"Precedes" (kommt-vorher):* X muss vor Y ausgeführt werden. Wird X explizit nicht ausgeführt, wird Y ebenfalls nie ausgeführt.

  - *"Inhibits" (verbietet):* X verbietet Y, d.h. wenn X ausgeführt wird, wird Y nicht ausgeführt.

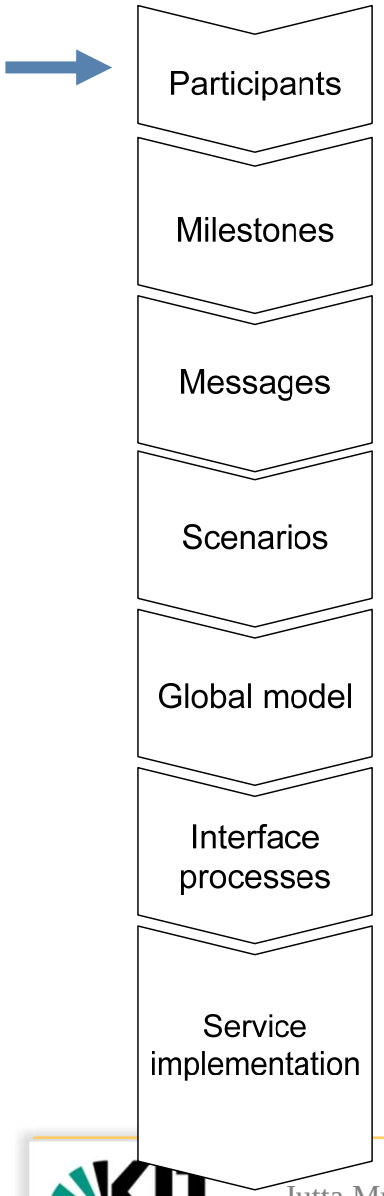


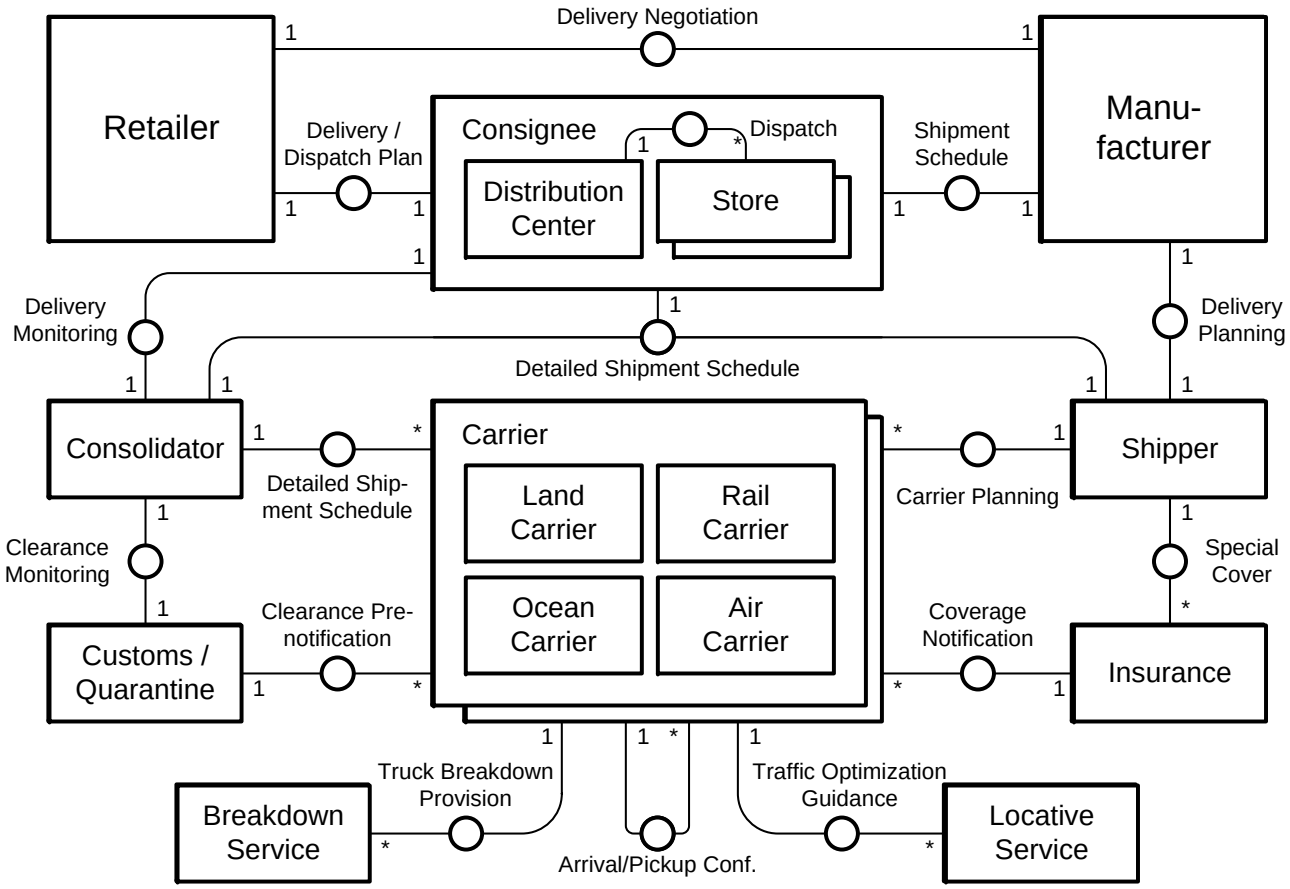
  - *"Weak-precedes" (weiches-kommt-vorher):* X kommt vor Y, aber wenn X nicht ausgeführt wird, wird Y trotzdem ausgeführt. D.h. sobald von X bekannt ist, dass es abgeschlossen ist (erfolgreich oder nicht), wird Y ausgeführt.



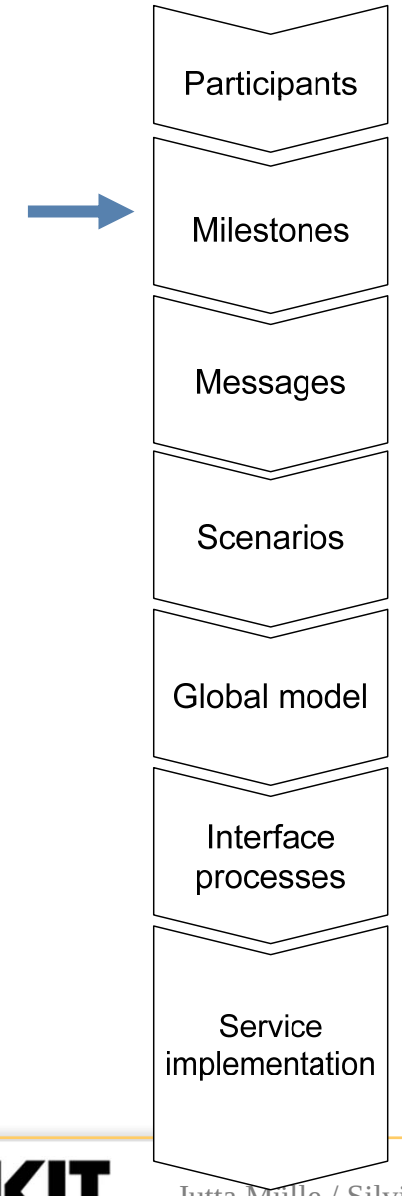
- **Datenfluss: implizit (Nachrichtenaustausch)**

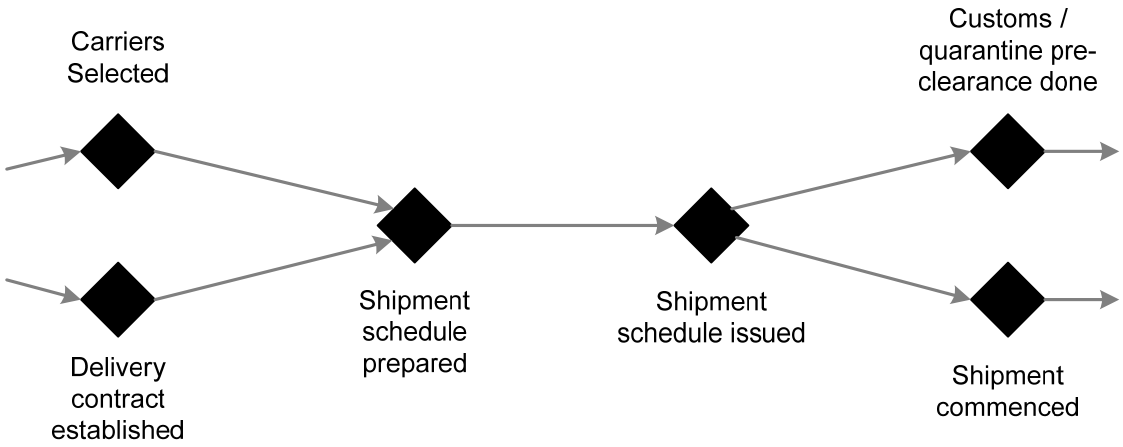
Let‘s Dance für top-down Choreographie-Design



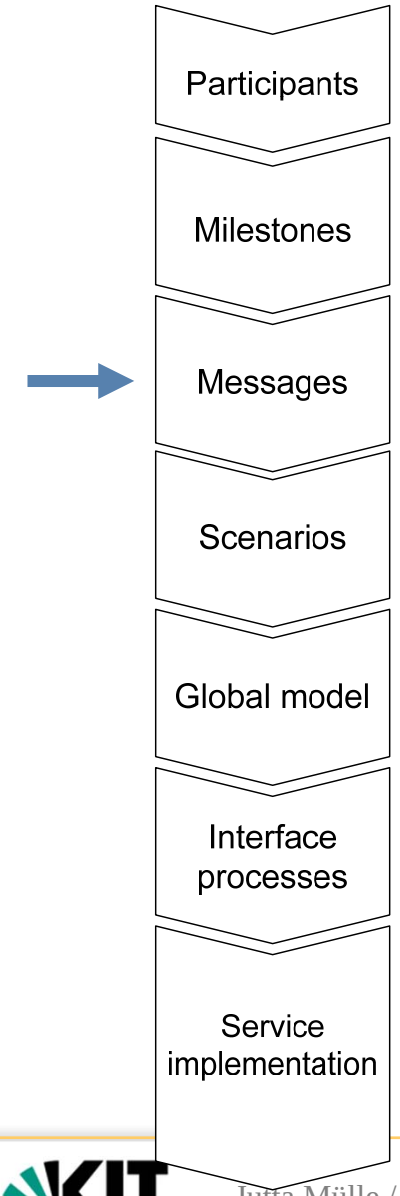


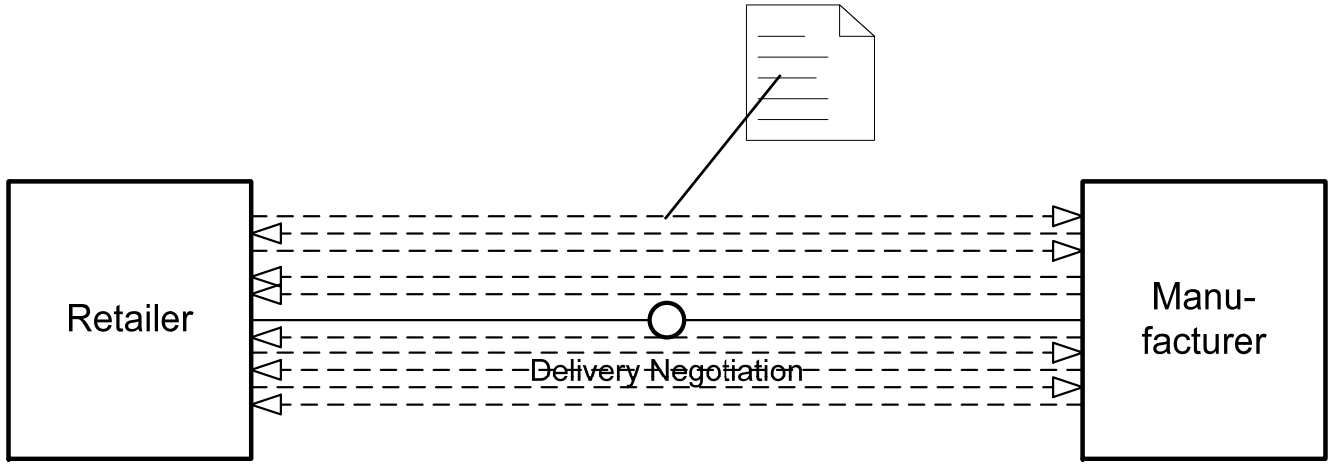
Let‘s Dance für top-down Choreographie-Design



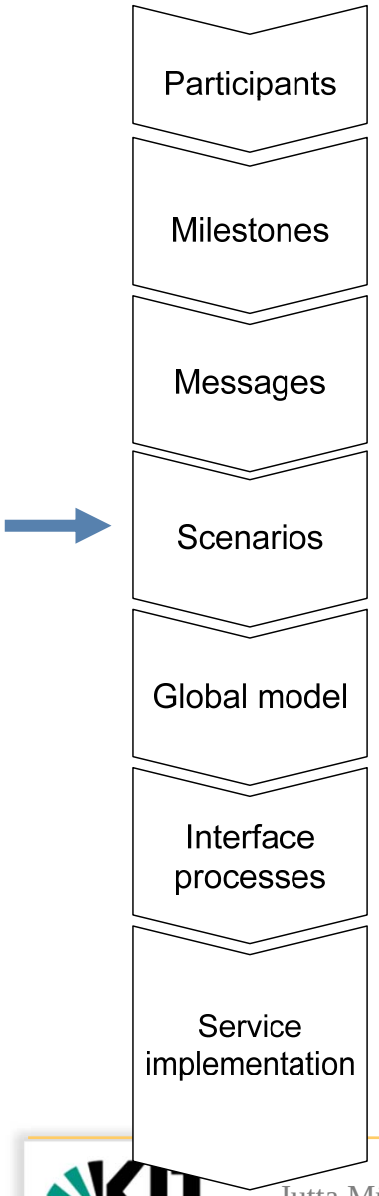


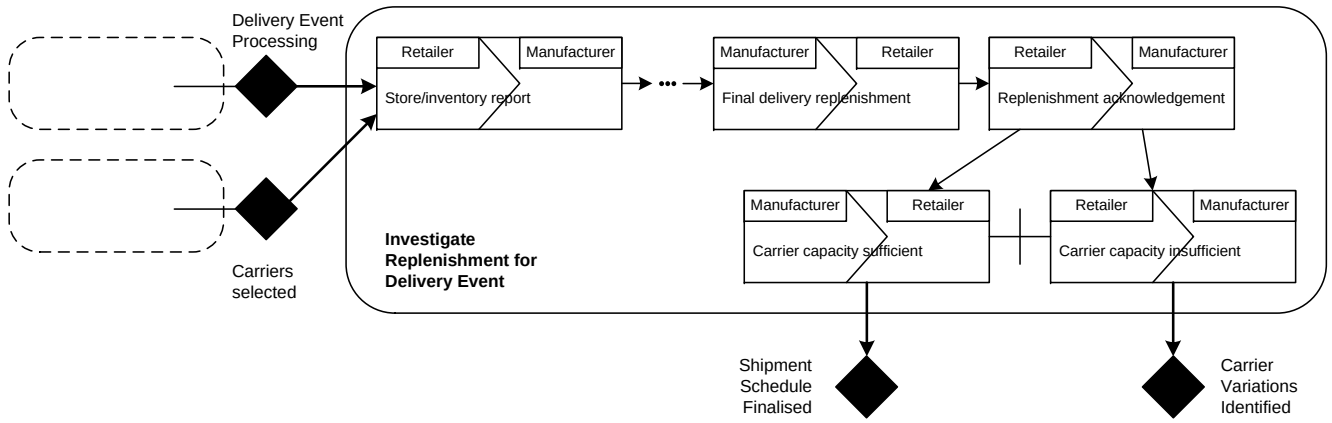
Let‘s Dance für top-down Choreographie-Design





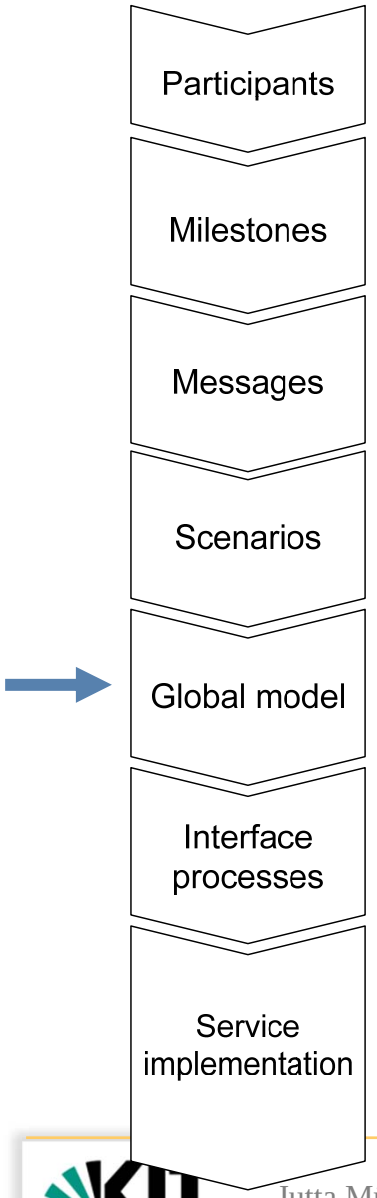
Let‘s Dance für top-down Choreographie-Design

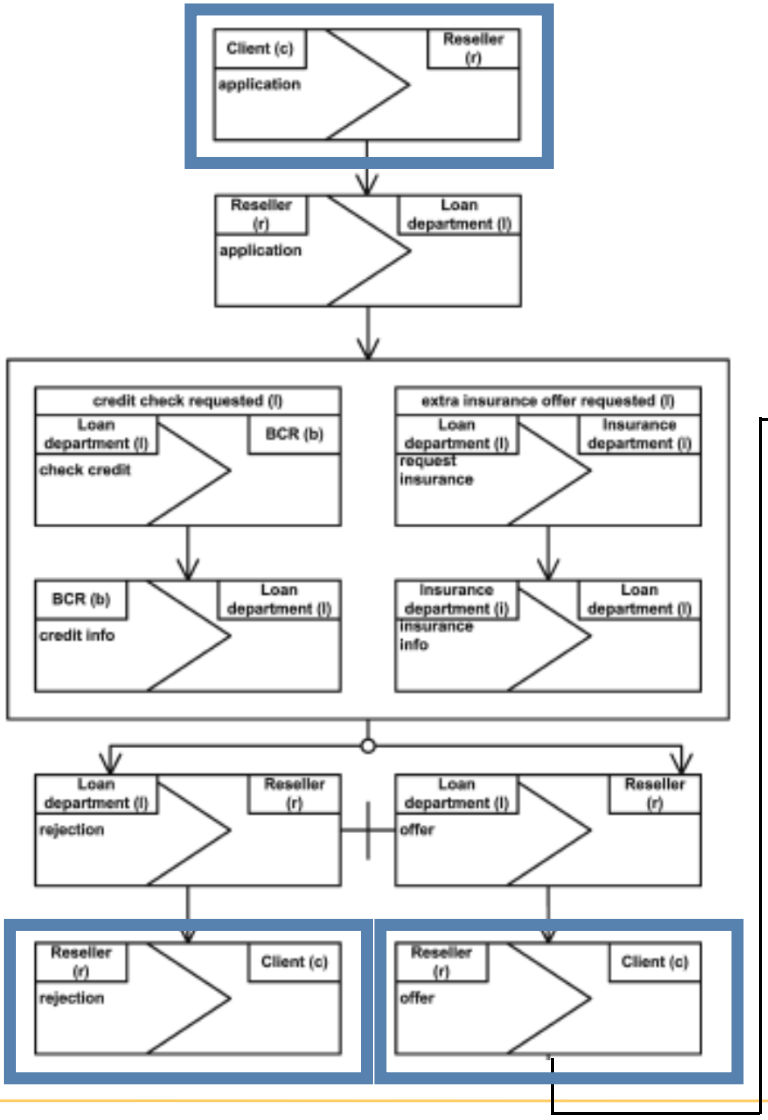




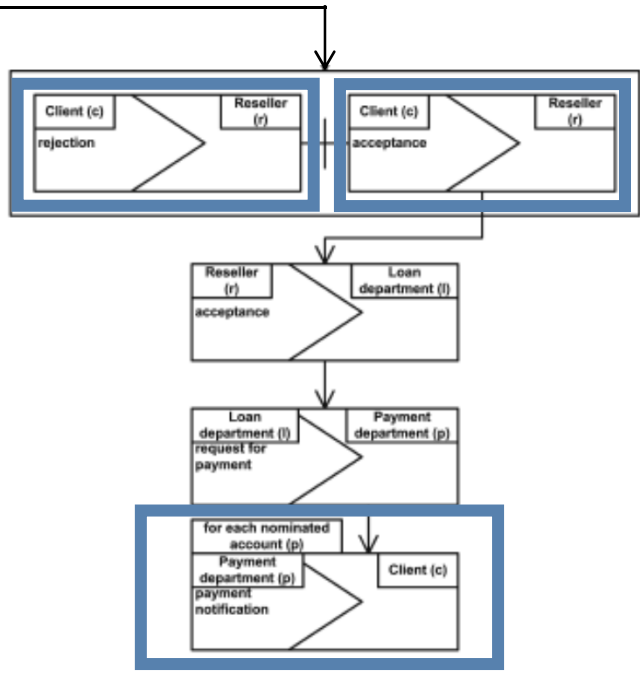
- Formale Ausführungssemantik durch Übersetzung ins Pi-Kalkül (Prozessalgebra)
- Modellverifikation durch Analyse-Algorithmen (Wohlgeformtheit, Konsistenz, Erreichbarkeit, lokale Überprüfbarkeit)

Let‘s Dance für top-down Choreographie-Design

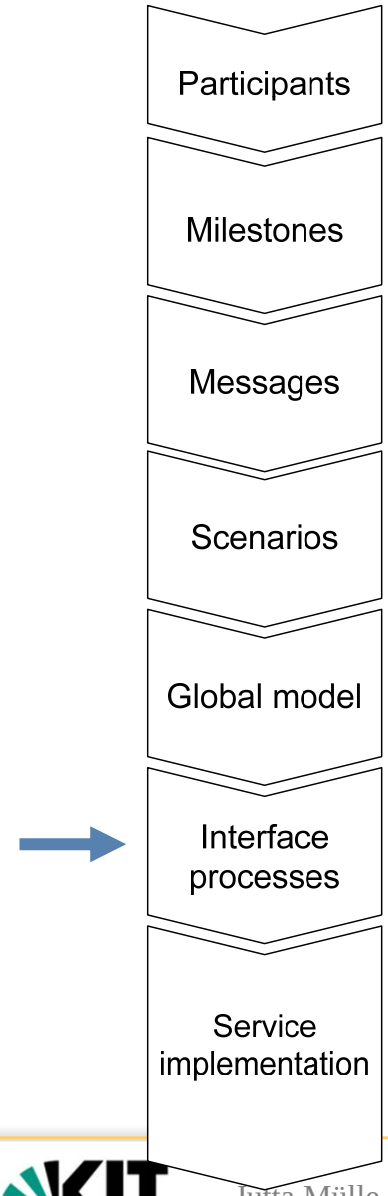


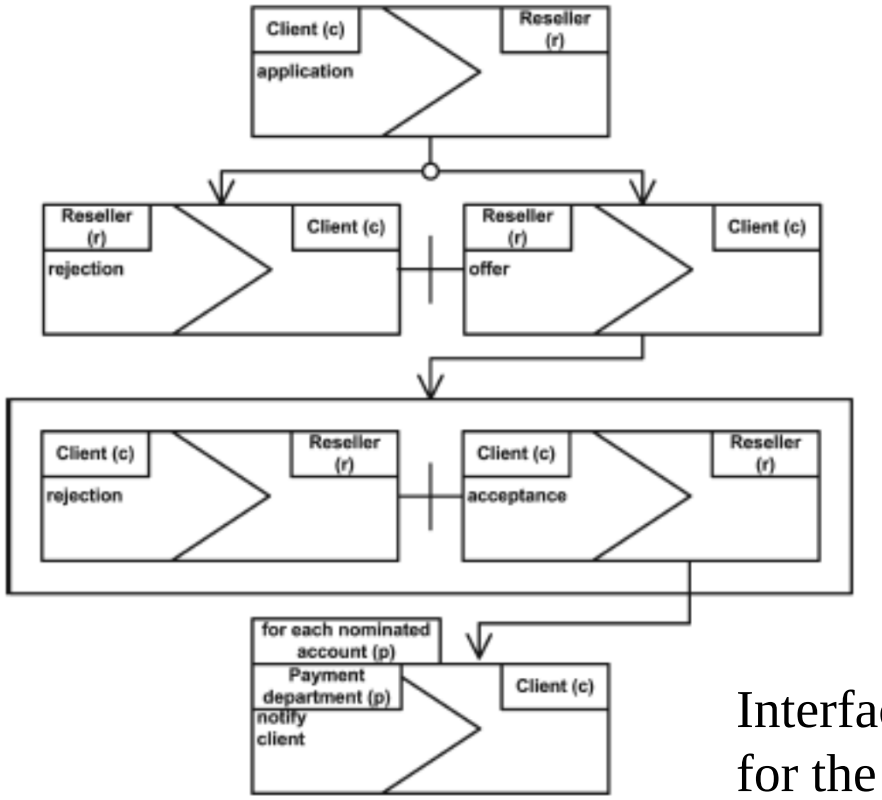


An welchen Interaktionen ist die Client-Rolle beteiligt?



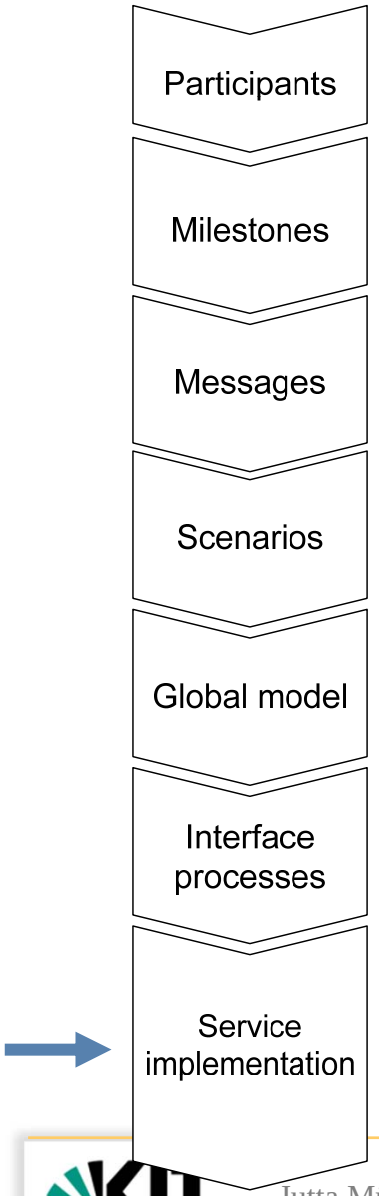
Let‘s Dance für top-down Choreographie-Design



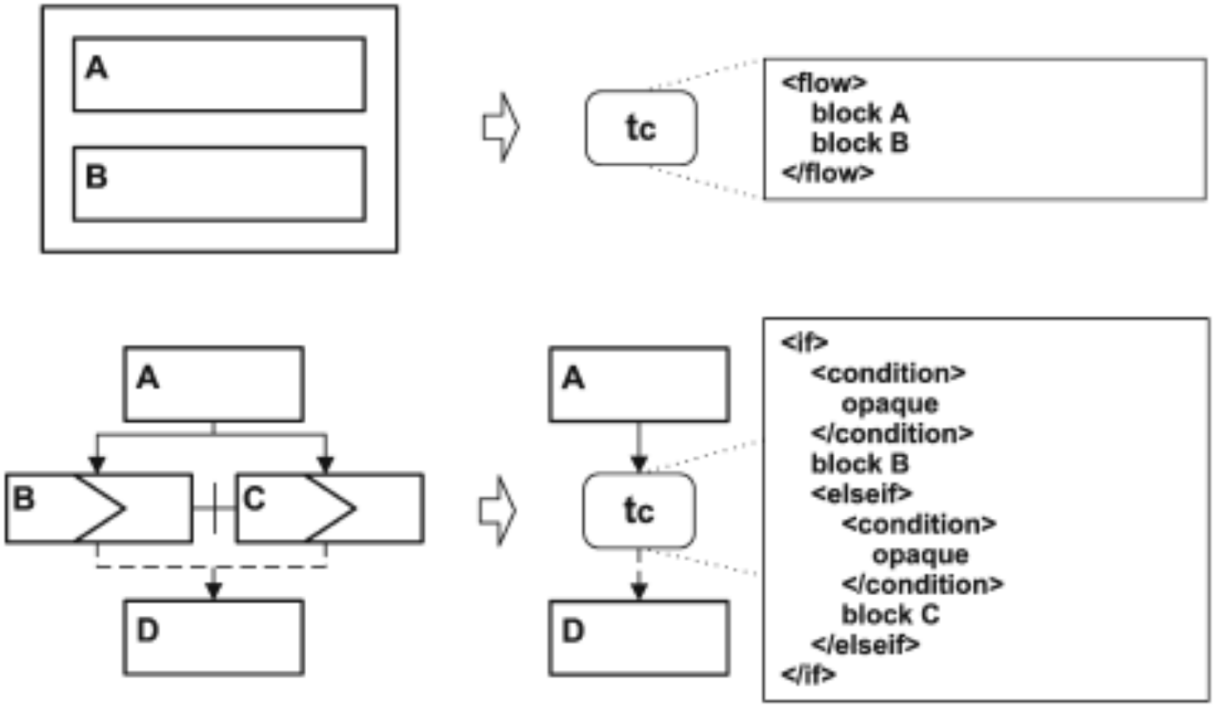


Interface process for the Client

Let‘s Dance für top-down Choreographie-Design

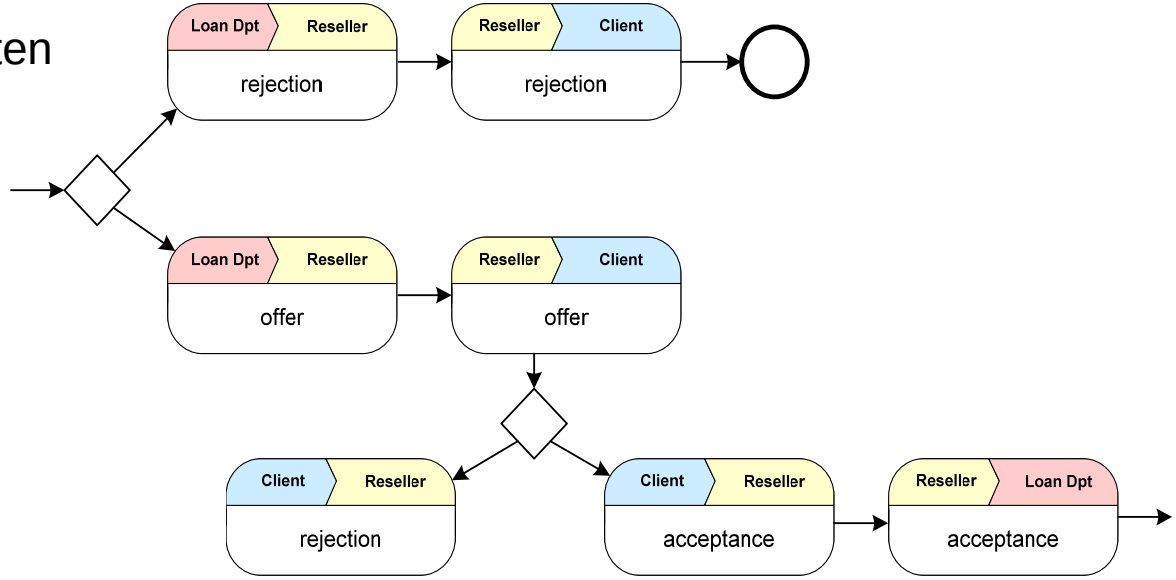


- Übersetzung von Prozess-Schnittstellen nach BPEL
- BPEL-Skelette als Startpunkt für die Implementierung



Grafische Notationserweiterung: BPMN4chor

- Motivation
  - BPMN ist eine weitverbreitete BP-Modellierungssprache
  - Bei Prozess-Modellierern ist die Notation bekannt
- Ausführungssemantik von BPMN teilweise vergleichbar zu Let's Dance
  - Übernehmen des wichtigsten Bausteins von Let's Dance in BPMN → BPMN4Chor
  - Mapping von BPMN4chor zu Let's Dance
    → Verifikation etc. bleibt möglich



Zusammenfassung: Let‘s Dance

- Let‘s Dance : Grafisch-formaler Ansatz zur Choreographie-Modellierung
  - Einfaches Meta-Modell, formale Semantik, Anwendbarkeit von Analyse-Algorithmen
  - Wissenschaftlicher Beitrag in der Geschäftsprozessmodellierung
- Tool-Unterstützung
  - Maestro for Let‘s Dance = Modellierung, Simulation, Analyse
- Andere grafische Notation: BPMN4chor
- Website
  - http://servicechoreographies.com

Agenda

- Einführung
- Let‘s dance
- **CDL**
- Zusammenfassung

WS-CDL allgemein (1)

WS Choreography Description Language (WS-CDL)

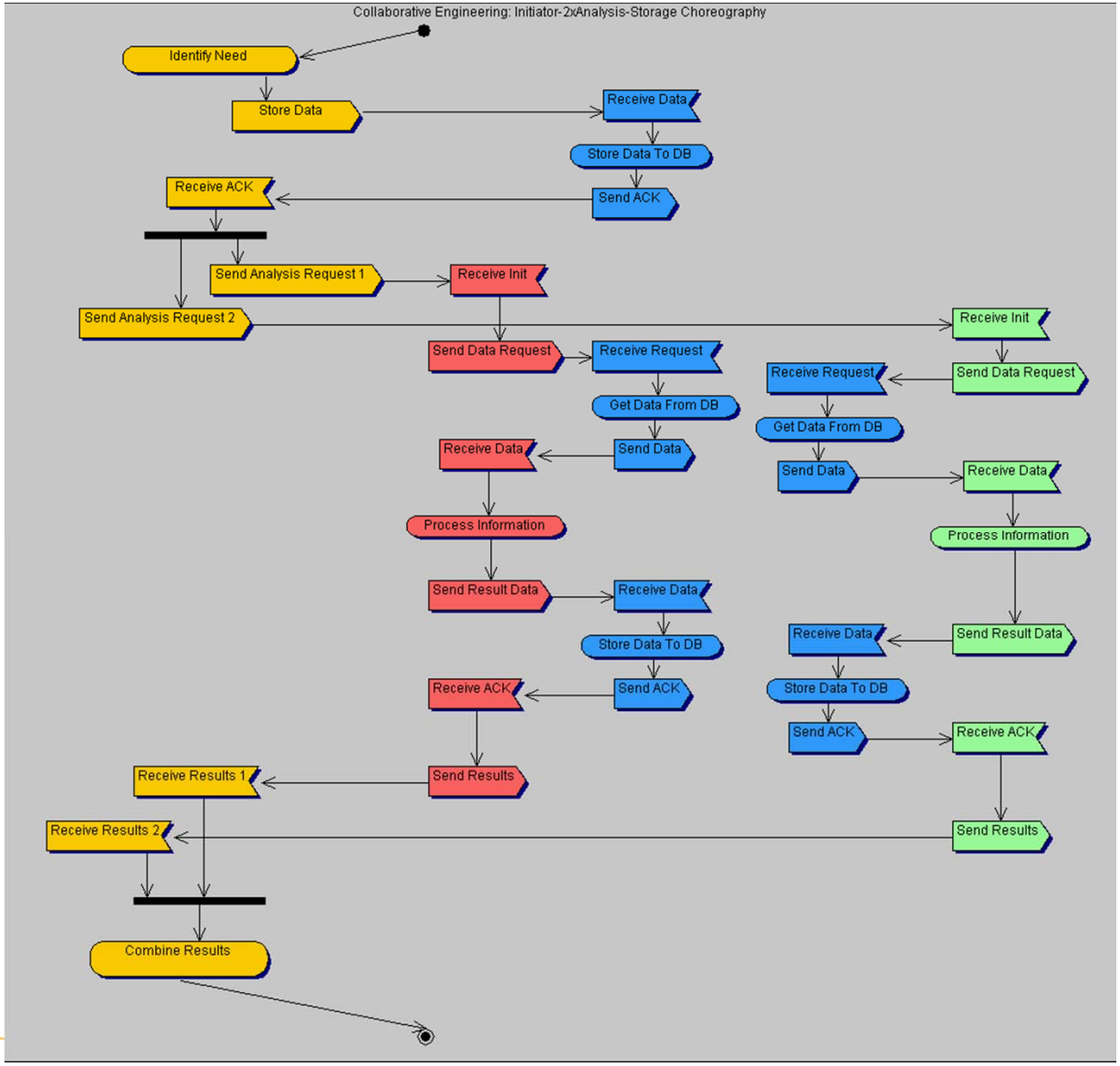
- W3C-Initiative
- Beschreibt kollaborative Geschäftsprozessmodelle:

  Betriebliche Abläufe, die mehrere Partner / Parteien (cdl:participants) umfassen
- Ganzheitliche Sichtweise, d.h. alle Partner sind gleichberechtigt
- XML-basiert
- Auf BPEL-Ebene (oder knapp darüber)

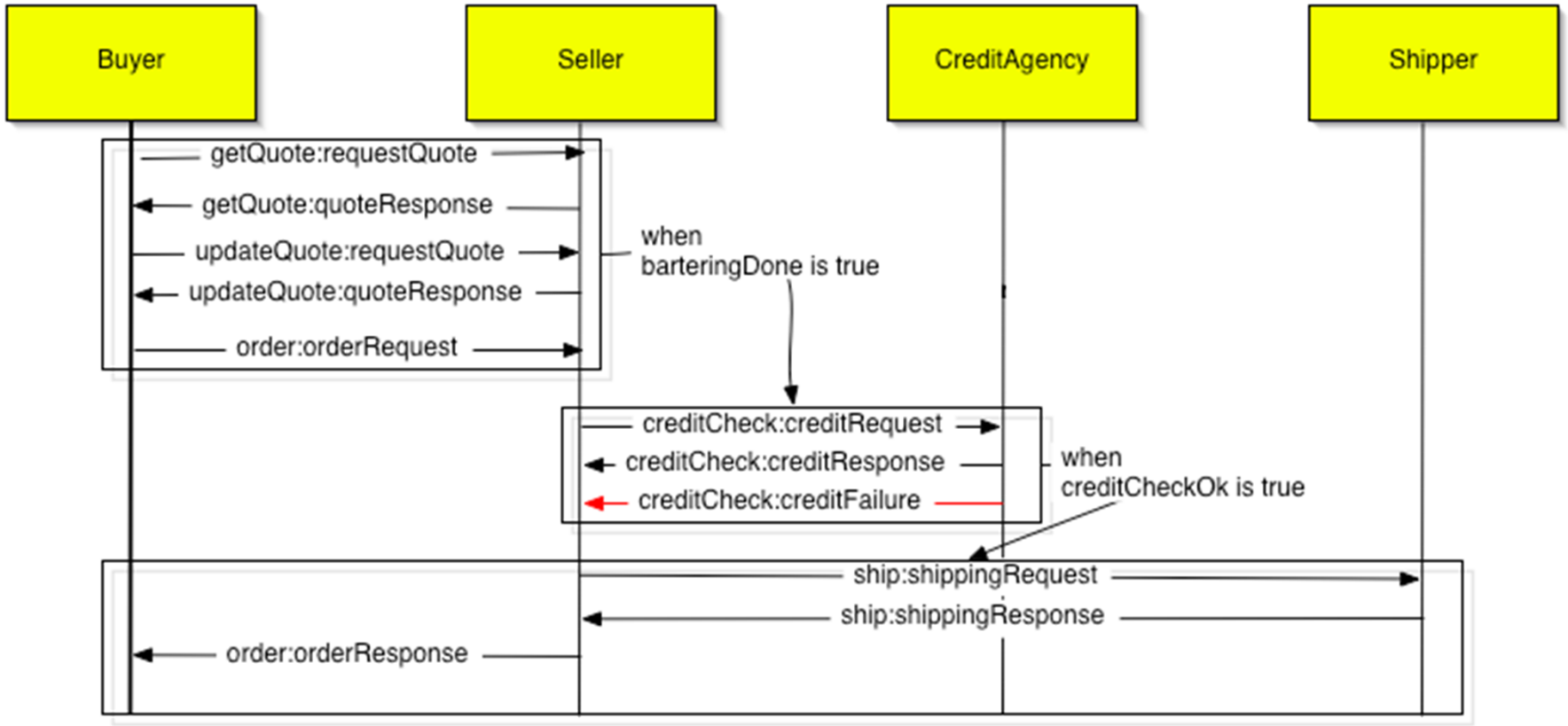
WS-CDL allgemein (2)

- Fokus auf Interaktionen zwischen Partnern
  - Interaktionen und Kontrollfluss-Konstrukte werden detailliert modelliert, partner-interne Details werden nur textuell beschrieben
- Die Choreographie beschreibt, *wer was wann* tut, aber nicht *wie*
- Interaktionen sind die Berührungspunkte zwischen den Parteien, daher der Name *Choreographie*
- Zwischen den Berührungspunkten legen die einzelnen Partner fest, wie genau sie sich verhalten
- Globales Prozessmodell für Kontroll- und Datenfluss
- Datenfluss ist wieder implizit, d.h. durch Variablen und deren Änderung
- Primer: http://www.w3.org/TR/2006/WD-ws-cdl-10-primer-20060619/

WS-CDL allgemein



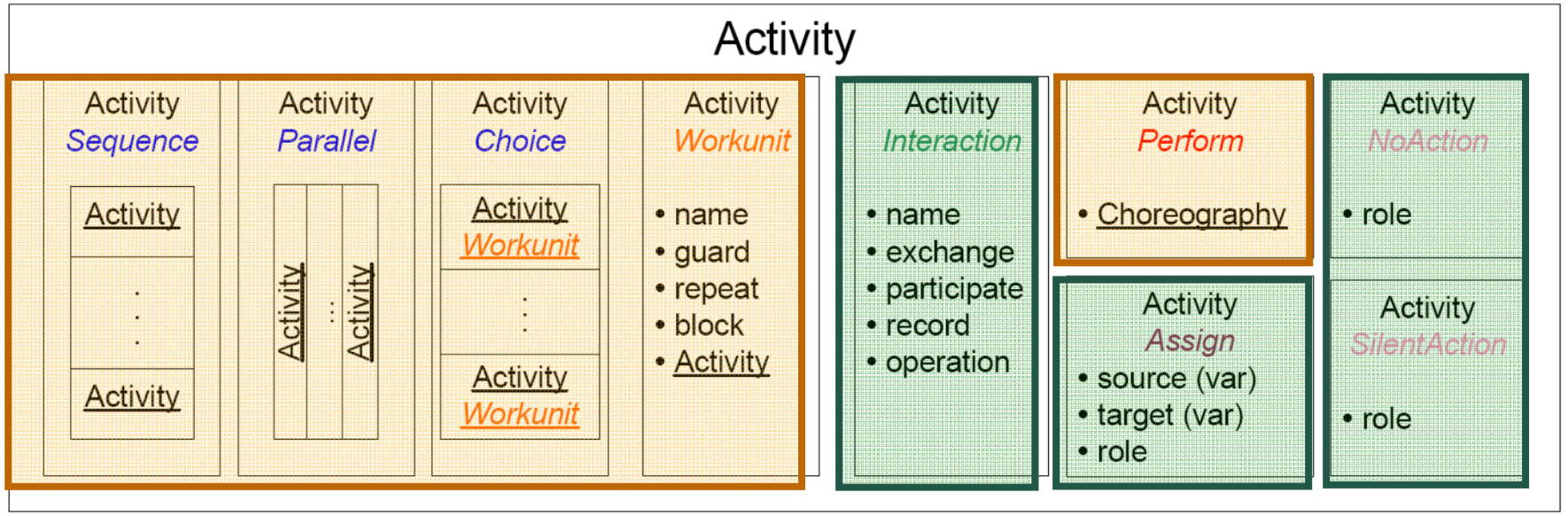
WS-CDL allgemein



WS-CDL: Übersicht

Choreography

- name
- root
- complete
- Exception - Activity - *Workunit*
- Finalizer - Activity - *Workunit*
- Relationship
-



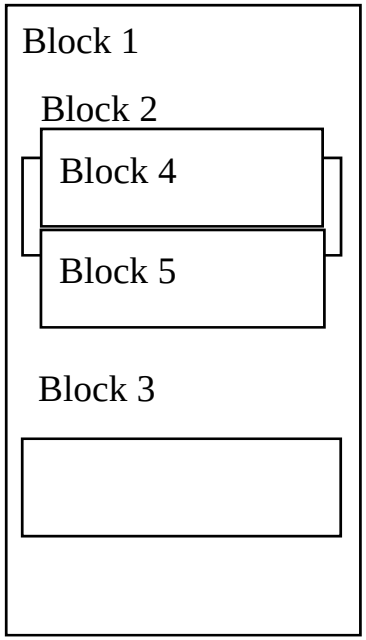
- variables
  - state
  - channel
  - application

Kontrollfluss

Aktivitäten

WS-CDL: Schachtelung und Kontrollfluss

- Prinzip: Schachtelung von Aktivitäten
- Sequenzielle Blöcke
  - Enthaltene Aktionen werden nacheinander abgearbeitet
- Parallele Blöcke
  - Enthaltene Aktionen werden parallel ausgeführt
- Choice-Blöcke
  - Nur einer von mehreren möglichen Prozess-Pfaden wird ausgeführt
- WorkUnits
  - Wie Let's Dance *Guard* und *Repeat*
  - Guard-Condition sagt, ob eine enthaltene Aktion ausgeführt wird
  - Repeat-Condition, ob die enthaltene Aktion wiederholt wird
- Perform
  - Eine Unter-Choreographie wird ausgeführt
- Bei Kontrollfluss-Elementen einige Entsprechungen in BPEL.



WS-CDL: Interaktionen und andere Aktivitäten (1)

- Interaction
  - Eine Nachricht wird von Rolle A zu Rolle B gesendet
  - Nachrichtenaustausch über Channels
    - Empfänger kann zur Laufzeit festgelegt werden
  - Eine Interaction kann mehrere Exchanges enthalten
    - Jeweils ein Nachrichtenaustausch
    - Eine gesendete Nachricht kann eine ‘normale’ Antwort und n mögliche Fehler-Antworten haben
  - Time-Out für Antworten kann festgelegt werden
  - Kann als atomare Transaktion behalten werden

WS-CDL: Interaktionen und andere Aktivitäten (2)

- Assign
  - Datenmanipulation / Variablenzuweisung
- NoAction
  - Eine Rolle tut an einer bestimmten Stelle explizit nichts
- SilentAction
  - Eine Rolle tut an einer bestimmten Stelle etwas (Freitext-Beschreibung), aber die Details davon sind für die Choreographie nicht relevant

WS-CDL: Beispiel

```
<package name="…" version="0.5" targetNamespace=http://...
        xmlns="http://www.w3.org/2004/12/ws-chor/cdl">
    <informationType name="stringType" type="xsd:string"/>…
    <roleType name="AnalysisPartner"><behavior name="Analyzer"/></roleType>…
    <participantType name="AnalysisParticipant">
        <roleType typeRef="AnalysisPartner"/></participantType>…
    <choreography name="SAC_Choreography" root="true">
        <relationship type="AnalysisStorageRel"/>
        <variableDefinitions><variable name="varRawDataAddr_Ana"
         informationType="uriType"/> …</variableDefinitions>
        <sequence>
           <interaction name="getRawDataReq" operation="getRawDataOp"
            channelVariable="chVarGet">
         <participate relationshipType="AnalysisStorageRel"
            fromRoleTypeRef="AnalysisPartner" toRoleTypeRef="StoragePartner"/>
         <exchange name="exRawDataAddr" informationType="uriType"
            action="request">
                    <send variable="cdl:getVariable('varRawDataAddr_Ana','','')"/>
                    <receive variable="cdl:getVariable('varRawDataAddr_Sto','','')"/>
        </exchange>
                </interaction>
                <silentAction roleType="StoragePartner">
        <description type="description">Deliver goods</description>
                </silentAction>
                <interaction name="getRawDataResp" operation="getRawDataOp"
                    channelVariable="chVarGet">…</interaction>
    …</sequence>        </choreography> </package>
```

Agenda

- Einführung
- Let‘s dance
- CDL
- **Zusammenfassung**

Zusammenfassung

- Choreographie
  - spezifiziert ein globales Interaktionsmodell
  - arbeitet mit öffentlichen Sichten von Workflowmodellen
  - wird zur Ausführung in Orchestrierungsmodell umgesetzt
- Let's Dance vs. WS-CDL
  - Beide beschreiben Service-Interaktionen aus einer globalen Perspektive.
    - WS-CDL ist eher auf Software-Entwickler ausgerichtet
    - Let's Dance ist für High-level-Design und -Analyse gedacht
    - WS-CDL ist eine mögliche Implementierungssprache für Let's Dance-Choreographien
- BPMN hat bereits viele Elemente einer Choreografiesprache (und wird in Richtung Ausführbarkeit weiterentwickelt, siehe Kapitel über BPMN).

Literaturhinweise

- Let's Dance
  - http://www.servicechoreographies.com
- WS-CDL
  - http://www.w3.org/TR/ws-cdl-10/
  - Editor (eclipse-plugin)
    - http://www.pi4tech.com
- Nutzung des Foliensatzes zu Let's Dance und WS-CDL von Ingo Weber (aus der Wfms-Vorlesung im Wintersemester 2008/09)

Exemplarische Fragen Kapitel 8

- Was ist eine Choreografie bei der Geschäftsprozessmodellierung?
- Wie ist der Zusammenhang mit der Orchestrierung?
- Welche Modellierungselemente sind in einer Choreografiesprache notwendig?
- Welche Aspekte werden unterstützt?
- Welche Rolle spielen öffentliche oder private Sichten auf Geschäftsprozesse?