Vorlesung am _____

Serviceorientiertes eGovernment

Enterprise Service Bus (ESB), Business Process Management (BPM)

Dr. Frank Sarre
Lehrbeauftragter der LMU München

Enterprise Service Bus (1)

Kommunikationsinfrastruktur

- Bei verteilten Systemen ist eine nachrichtenorientierte Infrastruktur sehr häufig anzutreffen
- Bisher waren die Lösungen in der Regel äußerst komplex und waren (leider) sehr stark herstellerabhängig
- Ein neuer Ansatz setzt auf offene Standards

→ **Enterprise Service Bus (ESB)**

Grundsatz:

Alle Komponenten der Anwendungslandschaft kommunizieren nur über die Infrastruktur und nicht mehr direkt (wie zum Beispiel bei Web-Services üblich).

Enterprise Service Bus (2)

Aufgaben eines ESBs

- Inhaltsbezogenes Weiterleiten von Nachrichten („content-based routing“)
- Transformation von Daten
- Flexible Steuerung von Prozessschritten eines (Teil-) Geschäftsprozesses („service orchestration“)
- Trennung der Transportlogik von der Schnittstellenimplementierung (und der Anwendungslogik)
- Monitoring von Schnittstellen und Geschäftsaktivitäten
- Sicherheitsfunktionen
- Sicherstellung eines bestimmten Qualitätsstandards der Service-Erbringung („quality of service“ [QoS])

Enterprise Service Bus (3)

Ausgangslage

- Heterogene IT-Landschaft mit zahlreichen unterschiedlichen Schnittstellen diverser Anwendungssysteme
- Nicht alle Systeme können und sollen als Web Services umgestaltet werden

Aufgaben

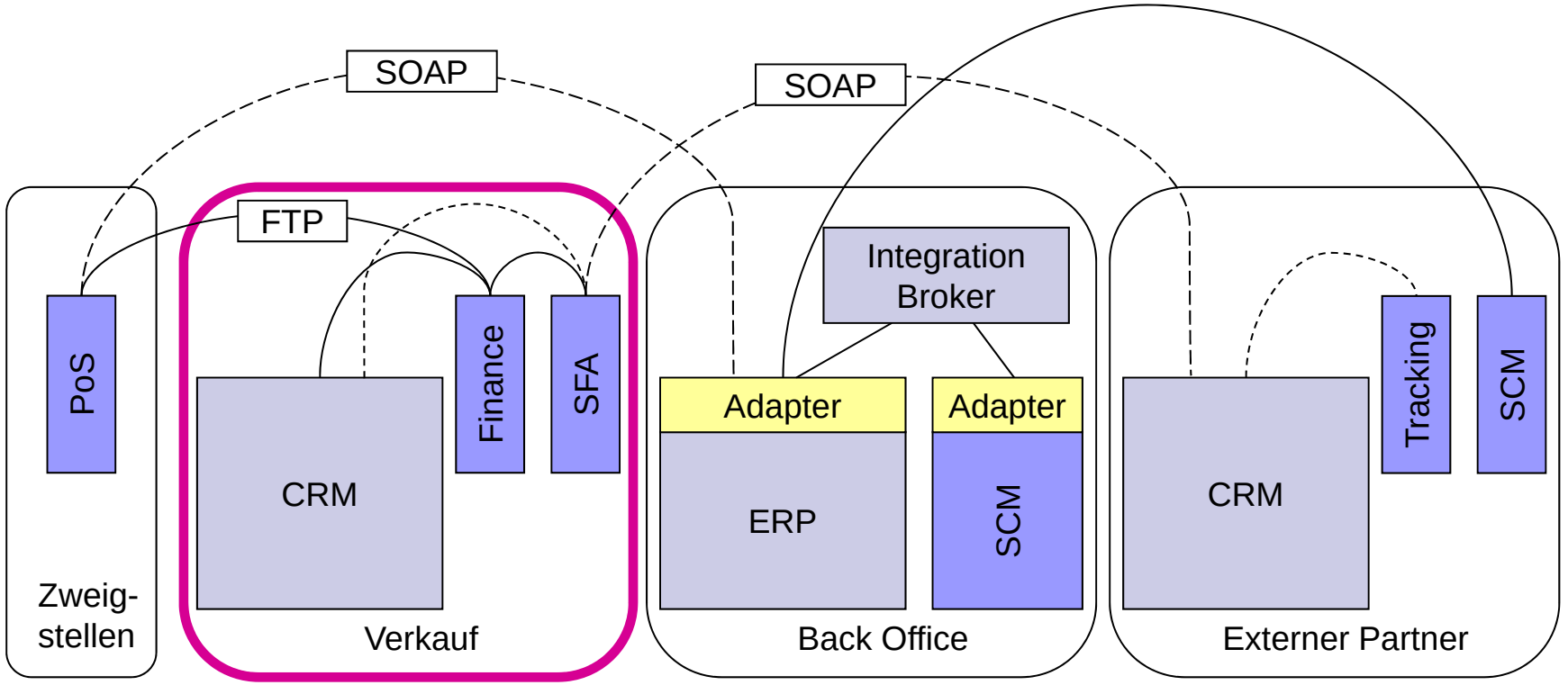
- Schrittweise Umgestaltung der IT-Landschaft
- Punkt-zu-Punkt-Verbindungen der bestehende Anwendungen möglichst transparent gestalten
- Kommunikationsprotokoll vereinheitlichen
- Kommunikationssicherheit verbessern
- Quality of Service erhöhen

Enterprise Service Bus (4)

Beispiel

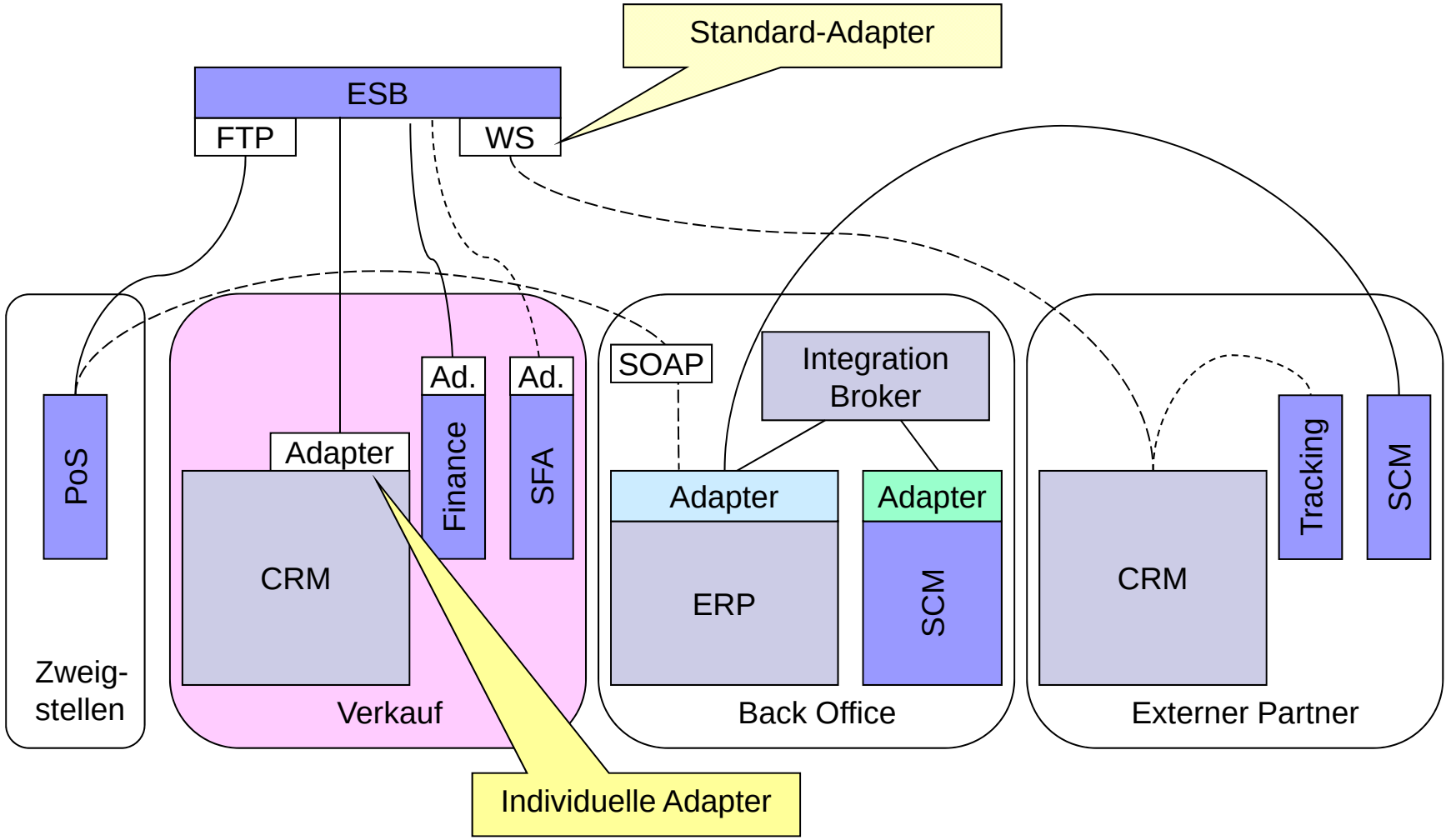
Quelle: D. Chapel, Enterprise Service Bus – Theory in Practice

IT-Landschaft mit unterschiedlichen Schnittstellentechnologien



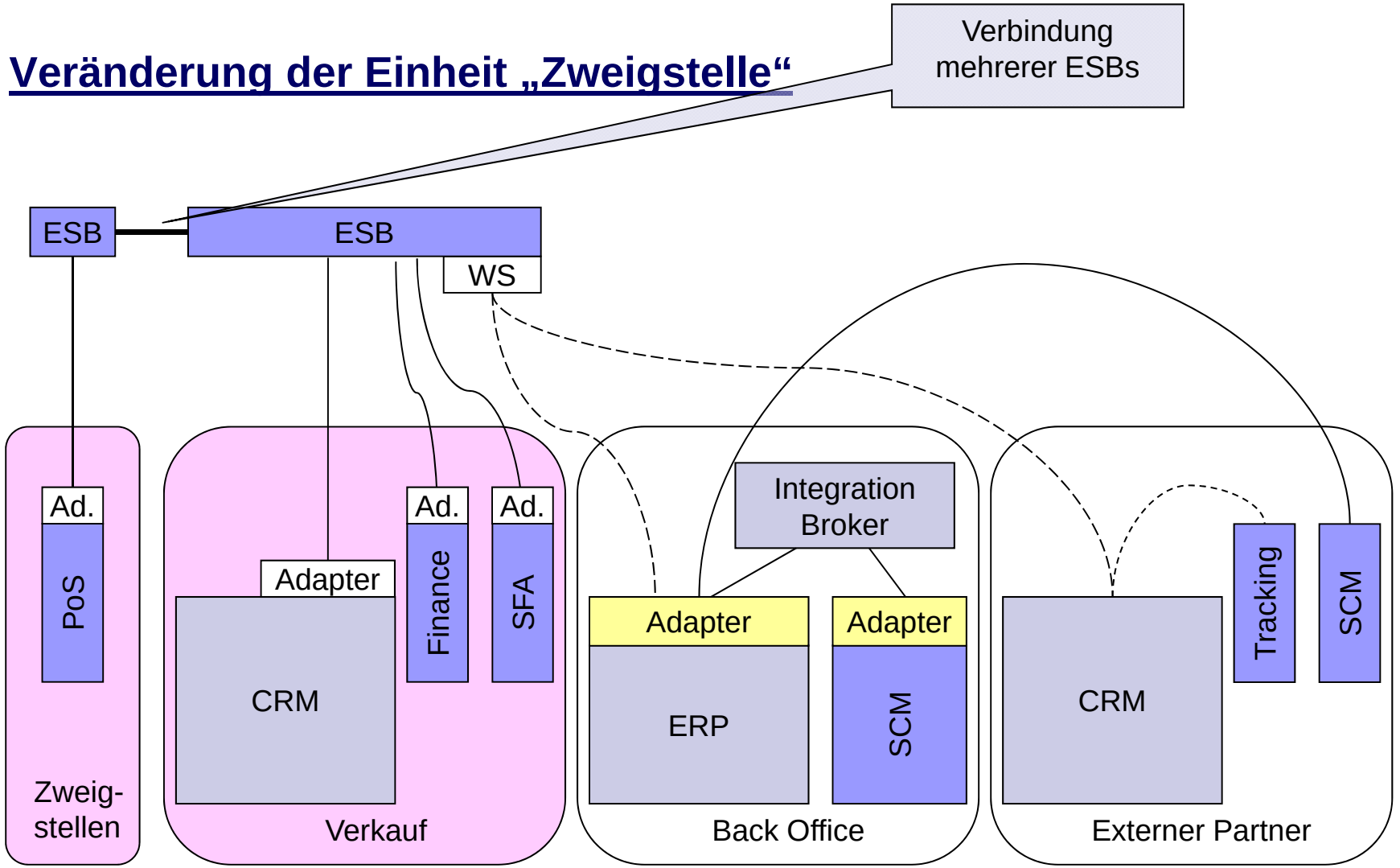
Enterprise Service Bus (5)

Veränderung der Einheit „Verkauf“



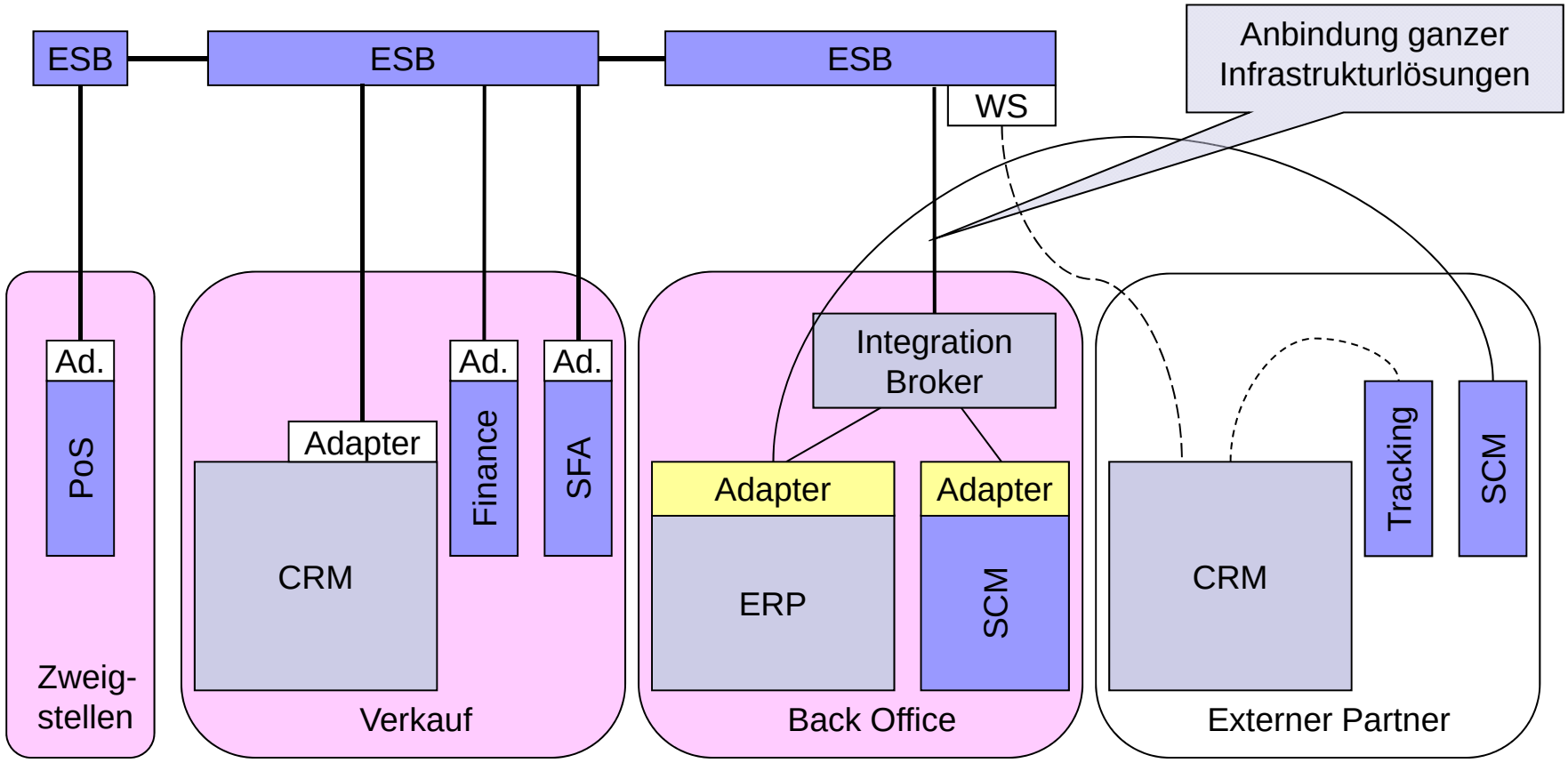
Enterprise Service Bus (6)

Veränderung der Einheit „Zweigstelle“



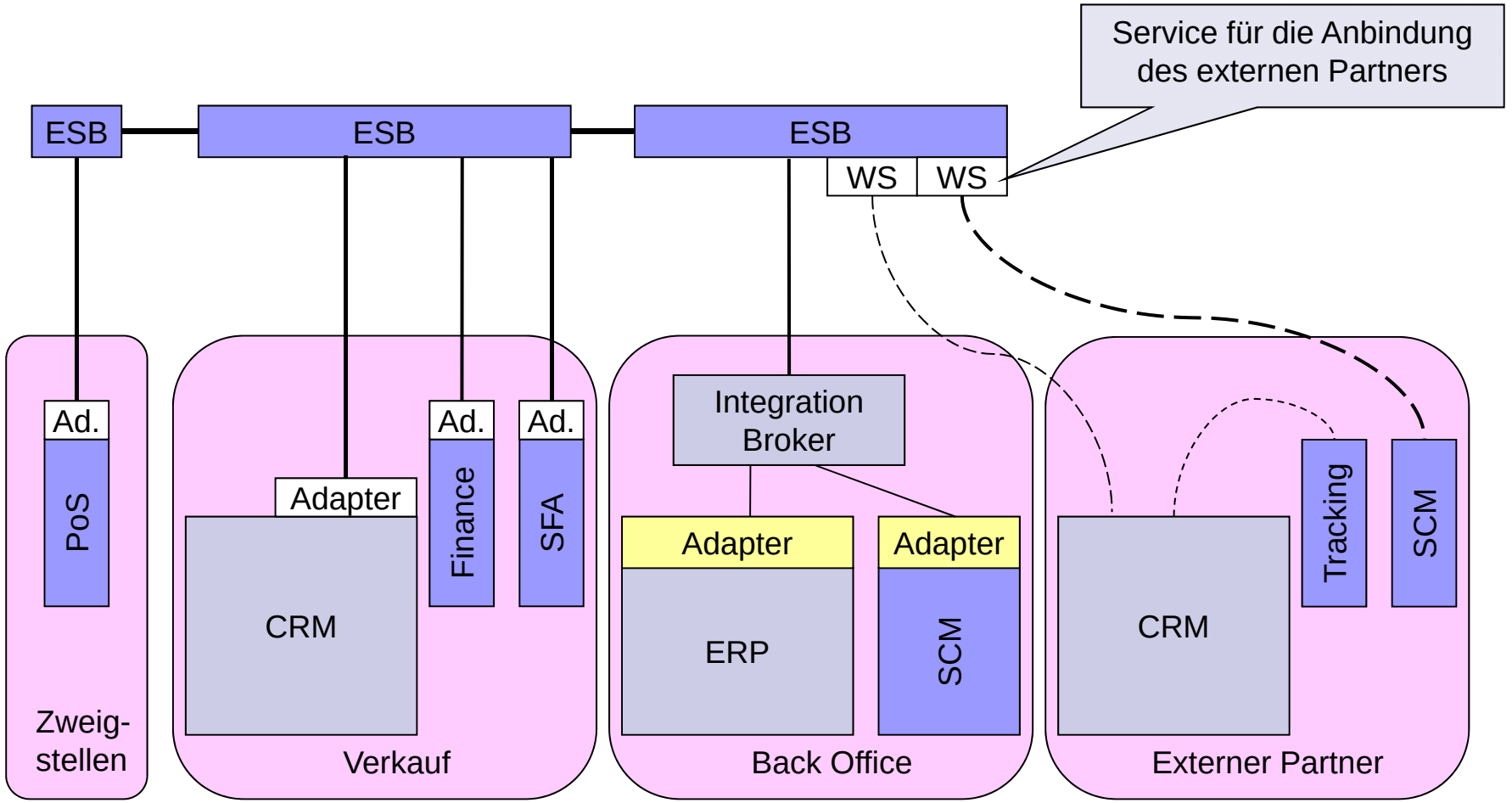
Enterprise Service Bus (7)

Veränderung der Einheit „Back Office“



Enterprise Service Bus (8)

Einrichtung weiterer Web Services „nach außen“



Enterprise Service Bus (9)

Wichtige Aspekte eines ESBs

- **Mehrere ESBs** können zu einem einzigen ESB **verknüpft** werden
- Innerhalb des ESBs wird in der Regel ein **einheitliches Kommunikationsprotokoll** verwendet, üblicherweise ein Protokoll, das auf XML basiert
- Der Datenaustausch muss über ein **nachrichtenorientiertes Kommunikationssystem** erfolgen, das folgende wesentliche Eigenschaften unterstützt:
  - **Sichere Zustellung** von Nachrichten (reliable messaging)
  - **Synchrone und asynchrone** Kommunikation
  - Unterstützung verschiedener **Message Patterns**
  - Zwischenspeicherung von **Prozesszuständen**
- Anwendungen werden entweder über Standardschnittstellen (Standardadapter) oder über individuelle Adapter angebunden.

Enterprise Service Bus (10)

Inhaltsbezogenes Weiterleiten von Nachrichten („content-based routing“)

- Daten sollen aufgrund bestimmter Dateninhalte unterschiedlich behandelt werden

  Beispiel:

  Bestellungen von Auslandskunden werden an ein anderes CRM geschickt als Bestellungen von Inlandskunden

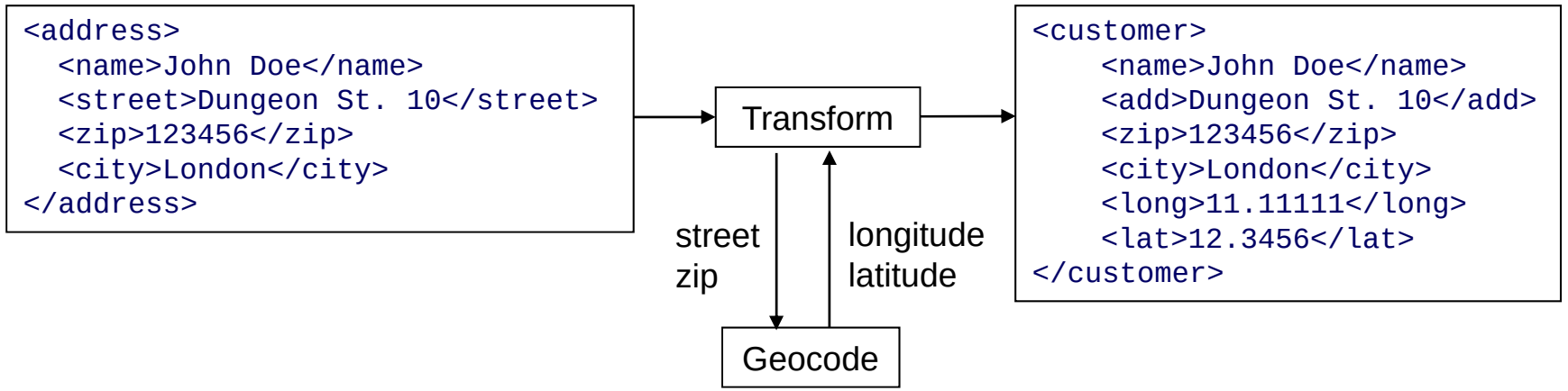
Transformation von Daten

- Passen eingehende Daten z.B. nicht zur Datenstruktur des Zielsystems, muss ein Mapping und ggf. eine Ergänzung der Daten vorgenommen werden
- Die Ergänzung der Daten kann durch Aufruf weiterer Services erfolgen

  Beispiel:

  Die Ergänzung, welche Geolokalisationsdaten eine Adresse hat
- XML-Nachrichten werden häufig mit Hilfe von XSLT transformiert

Beispiel:



Enterprise Service Bus (12)

Verminderung von Schnittstellen durch einheitliches XML-Format

- Solange jede Einzelkommunikation zwischen Anwendungen individuell abgebildet wird, nimmt **die Anzahl der individuell zu konfigurierenden Schnittstellen** nicht ab
- Abhilfe durch Schaffung eines **firmenweiten XML-Dialekts**, der die wesentlichen Dateninhalte definiert („kanonisches XML“)
- Jede Nachricht wird vom ESB zunächst in dieses Format übersetzt und in dieser Form verschickt (**Eingangstransformation**).
- Der Empfänger entnimmt daraus die Informationen, die er braucht (**Ausgangstransformation**).

→ **Verringerung der Anzahl** der individuellen Schnittstellen

→ **Verbesserte Übersichtlichkeit,** da Schnittstellen zentral definiert werden

→ Evtl. **Aufblähung des Datentransfervolumens** und daraus resultierende Performanceprobleme

Enterprise Service Bus (13)

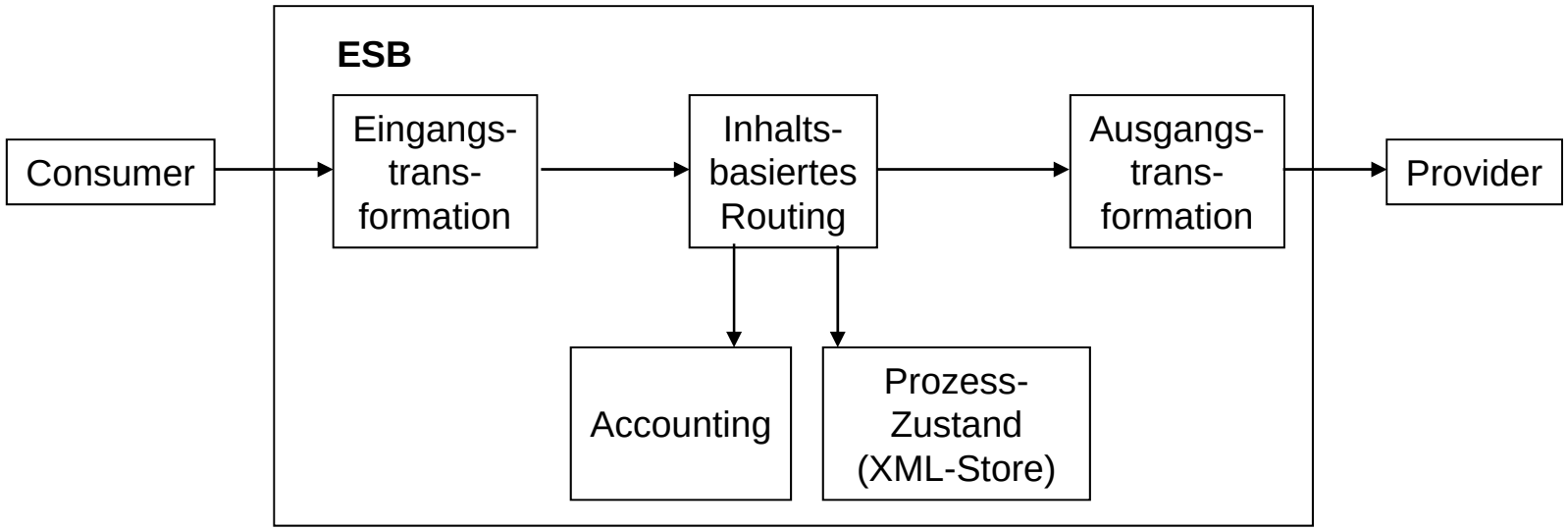
Steuerung von Prozessschritten eines (Teil-) Geschäftsprozesses („service orchestration)“

- Die Schritte eines Geschäftsprozesses können über eine **Abfolge von Services**, die durch den ESB gesteuert werden, abgebildet werden
- Bei komplexer Steuerung ist der Einsatz einer **Business Rules Engine** üblich, die die Schrittfolge inhaltsbezogen steuert
- Alternativ kann auch ein „**Orchestrierungsservice“** verwendet werden
- Häufig ist die Zwischenspeicherung von Prozesszuständen erforderlich (z.B. mit Hilfe von XML-Storage, um Persistenz zu erreichen)

Enterprise Service Bus (14)

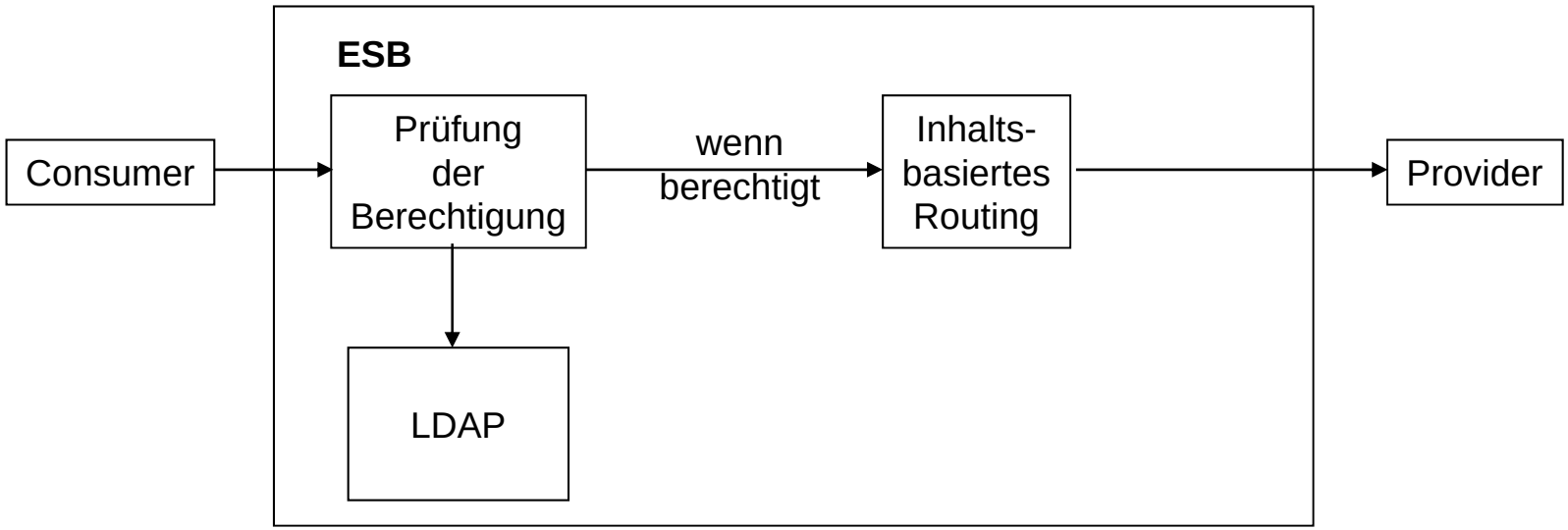
Monitoring von Schnittstellen und Geschäftsaktivitäten

- Wichtige Schnittstellen müssen hinsichtlich Auslastung, Verfügbarkeit und möglichen Fehlern **überwacht** werden (→ Kontrolle der Quality of Service)
- Für die Kostenrechnung könnte die **Häufigkeit von Serviceaufrufen** protokolliert werden
- Überwachung des Zustands von Geschäftsprozessen durch sog. **Business Activity Monitoring** (BAM)



Sicherheitsfunktionen

- Es muss sichergestellt sein, dass nur berechtigte Benutzer auf einen Service zugreifen
- Sicherheitsinformationen müssen in jeder Nachricht berücksichtigt werden (→ WS-Security oder andere Methoden wie z.B. LDAP)



Enterprise Service Bus (16)

Sicherstellung eines bestimmten Niveaus an Quality of Service

- Prüfung der Verfügbarkeit und der Performance durch das Monitoring
- Vorrang für bestimmte Dienste aufgrund einer vorgegebenen Priorisierung
- Transparentes Routing zu Ersatzsystemen, wenn Services ausfallen (evtl. auch durch Hardwaremechanismen)
- Transparente Verteilung der Last auf verschiedene Systeme
- Transparentes Hinzuschalten ergänzender Systeme, wenn die bestehenden Systeme die Last nicht mehr verarbeiten können

Hub & Spoke-Architektur vs. verteiltes Routing

- Das Routing von Nachrichten kann entweder **zentral** oder **dezentral** gesteuert werden:

  a) **Zentrales Routing** bedeutet, dass für alle Nachrichten, die über den ESB geschickt werden, eine zentrale Instanz die Routing-Entscheidungen trifft (sog. **Hub & Spoke-Architektur**).

  → Evtl. hohe Belastung der zentralen Systeme

  b) Beim dezentralen Routing (**itinerary-based routing**) enthält die Nachricht selbst die notwendigen (Routing-)Informationen, um zum nächsten Kommunikationspartner weitergeleitet zu werden.

  → Kein Engpass bei einem zentralen System

  → Verfeinerung des Konzepts der losen Kopplung

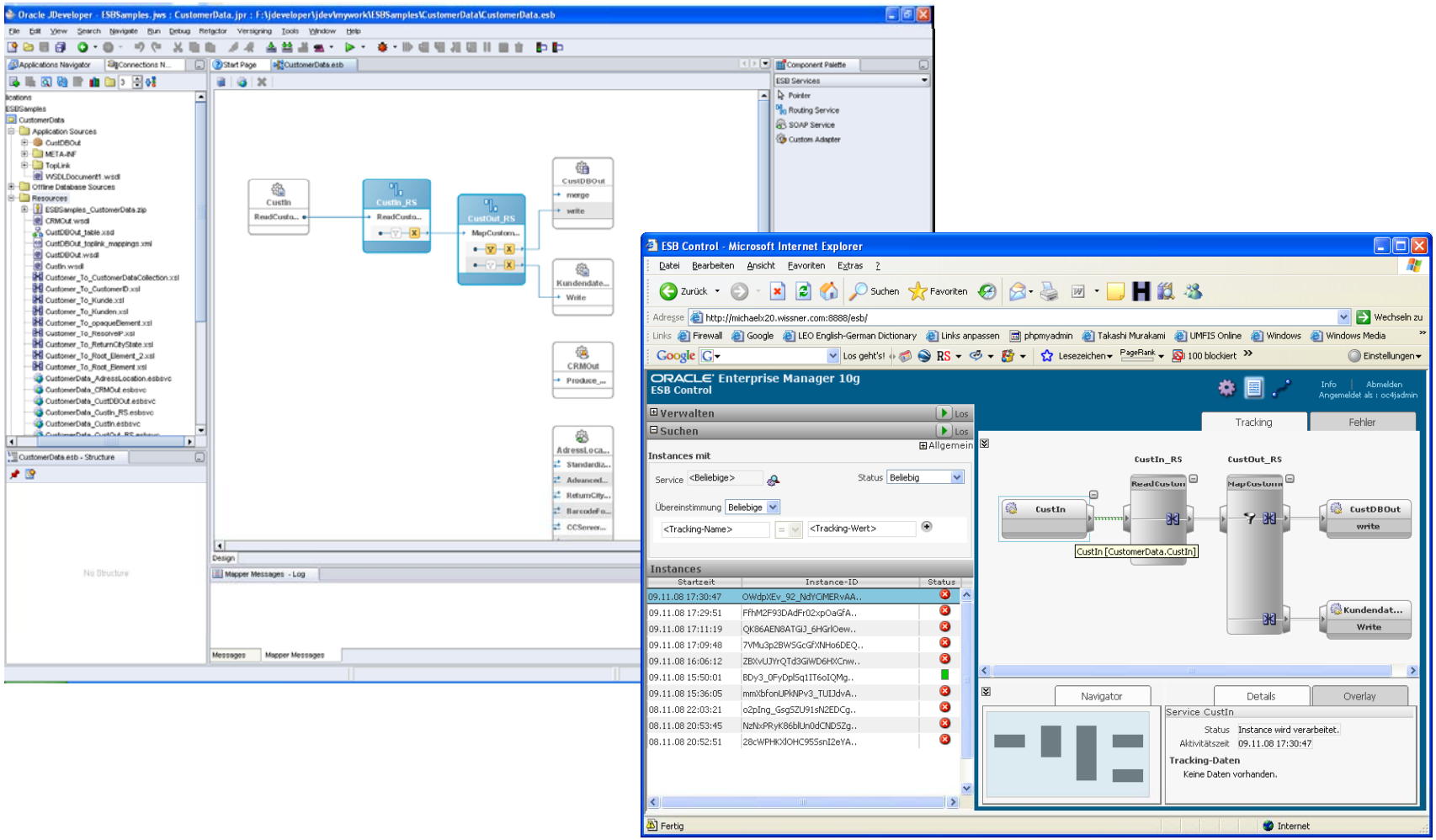
Enterprise Service Bus (18)

Wichtige Aspekte im Hinblick auf Web Services

- Lösung für das Problem der synchronen Aufrufe des HTTP-Protokolls
  - Request-Nachrichten werden vom ESB akzeptiert und zwischengespeichert
    (für den Consumer ist der Aufruf damit erledigt)
  - Der ESB sorgt dann dafür, dass die Nachricht zuverlässig an den Empfänger übermittelt wird (→ reliable messaging)
- Entkopplung der Punkt-zu-Punkt-Verbindungen
  - Zugangspunkte für die Web Services sind nicht mehr die tatsächlichen physischen Zugangspunkte der Services, sondern die Zugangspunkte des ESBs!
- Bereitstellung von fehlenden Funktionalitäten zur Zugriffskontrolle und im Hinblick auf die Sicherheit durch den ESB

Enterprise Service Bus (19)

Beispiel eines ESB-Produkts: Oracle



BPM --- Ausgangssituation

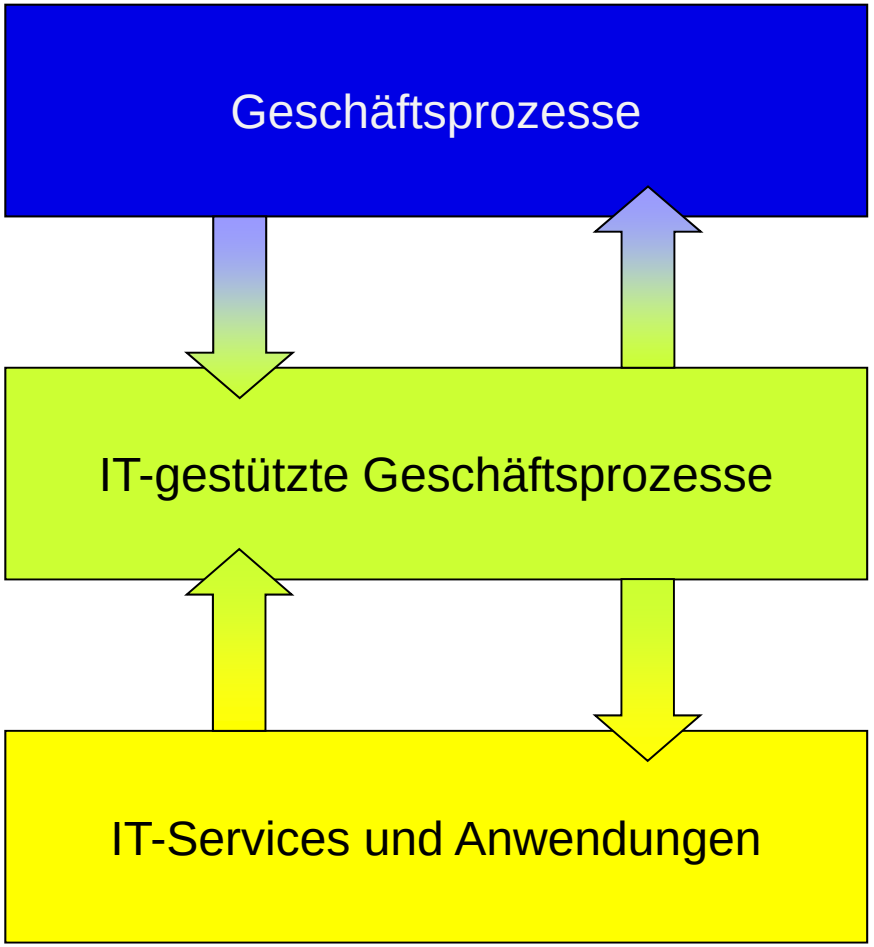
Geschäftsprozess

- Reale Aufgaben und damit verbundene Tätigkeiten
- Beschreibung eines Geschäftsprozesses:
  - nicht dokumentiert
  - informell dokumentiert
  - halbformal dokumentiert, z.B. durch einfache Arbeitsanweisung
  - formal dokumentiert, z.B. durch Ablaufplan für ISO-Zertifizierung
  - formal und in elektronisch verwertbarer Form dokumentiert (z.B. mit Hilfe eines Modellierungswerkzeugs)

IT-Services und Anwendungen

- Bestehende Systeme und deren Services, die „nach außen“ angeboten werden

Ziel


Geschäftsprozesse
IT-gestützte Geschäftsprozesse
IT-Services und Anwendungen

Vorgehensweise (1)

Top-Down

- Definition von Geschäftsprozessen und Entwicklung der dazu passenden IT-Services

  → In komplexen IT-Umfeldern unrealistisch, da bereits viele Anwendungen vorhanden sind, die genutzt werden müssen / sollen.

Bottum-Up

- Anpassung der Geschäftsprozesse an vorhandene Services und Anwendungen

  → Gefahr, mit veralteten Methoden weiterzuarbeiten

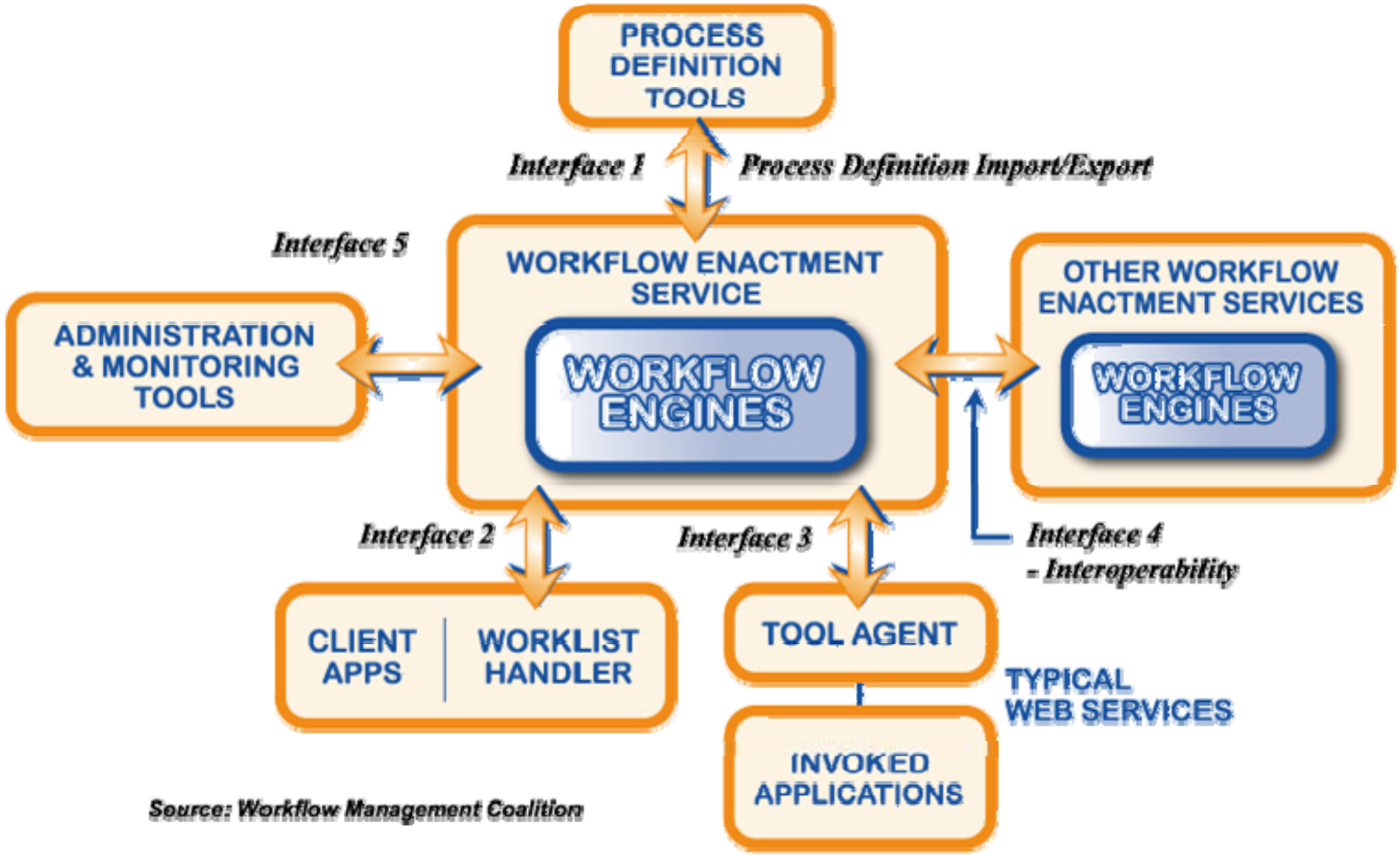
Vorgehensweise (2)

Kombination aus Top-Down und Bottom-Up

- Was soll erreicht werden?
- Welche Services und Anwendungen sind bereits vorhanden?
- Was kann davon verwendet werden?
- Was muss neu gemacht werden?
- Welche technische Randbedingungen gibt es,
  z.B. nicht mehr modifizierbare Systeme?
- Welche Funktionen sollen wieder verwendbar sein?
- …

→ Kreativer, iterativer Prozess

Referenzmodell der Workflow Management Coalition (WfMC)



Workflow- oder Prozess-Engine?

Beide Engines bieten ähnliche Funktionalitäten:

- Verarbeitung von Prozessbeschreibungen
- Steuerung der angebundenen Dienste
- Speicherung von Verwaltungsinformationen
- Überwachung und Protokollierung

Was ist jedoch der Unterschied zwischen „Workflow“ und „Prozess“?

Ein Prozess beschreibt, **was** geschehen soll.

Ein Workflow beschreibt, **wie** es durchgeführt wird.

→ Diese Definition ist aber nicht durchgängig verwendet, so dass beides oft synonym verwendet wird.

Prozessbeschreibung (Process Description)

- An erster Stelle steht die Aufgabe, den Geschäftsprozess so zu beschreiben, so dass er sich auf einer Prozess- oder Workflow-Engine technisch ausführen lässt.

  Für die Beschreibung wurden verschiedene Standards entwickelt

  - WS-BPEL (früher BPEL4WS)
  - BPEL4People
  - BPML
  - EPK / EPC
  - XPDL
  - UML
  - …

Business Process Management Systeme (BPMS) [4]

Schnittstellen für Client-Anwendungsprogramme

- Aufruf von Prozessen / Teilprozessen
- Statusabfrage
- Benutzerinteraktion

Schnittstellen für die Anbindung von Services

- Nachrichtenaustausch
- Mapping verschiedener Protokolle und APIs
- Session Management

Schnittstellen zu anderen Workflow-Engines

- Aufruf externer Prozesse
- Anbindung von externen Systemen

Administrations- und Monitoring-Tools

- Benutzermanagement
- Rollenmanagement
- Protokollierung
- Ressourcenüberwachung
- Überwachung von Prozesszuständen

Modellierungssprachen (1)

BPEL4WS bzw. WS-BPEL (Business Process Execution Language)

- Basiert auf XML
- Standardisierte Programmiersprache (OASIS)
- Modellierung entweder im XML-Code, in der Regel aber mit einer **nicht standardisierten** grafischen Notation
- BPEL ist in einer Workflow-Engine **direkt ausführbar**
- Breite Unterstützung in der Industrie (?)

Modellierungssprachen (2)

XPDL (XML Process Definition Language)

- Standard der WfMC
- Wie BPEL direkt in Workflow-Engines ausführbar
- **Standardisierte grafische Notation**, Verwendung von BPMN

Modellierungssprachen (3)

BPML (Business Process Modeling Language)

- Basiert auf XML
- Standardisierte Modellierungssprache (OMG)
- **BPMN** (Business Process Modeling Notation) ist eine **standardisierte grafische Notation**
- **BPQL** (Business Process Query Language) für die Administration
- **WSCI** (Web Services Choreography Interface) für die Prozesssteuerung
- BPMN lässt sich **auf BPEL abbilden**

Modellierungssprachen (4)

UML (Unified Modeling Language)

- Standard der OMG
- **Standardisierte grafische Notation**, aber nicht stringent auf Prozessmodellierung bezogen

  → Abbildung auf Workflow-Engines ist problematisch

Modellierungssprachen (5)

Weitere Modellierungssprachen

- **BPSS** (Business Process Specification Scheme, Modellierungssprache im Umfeld von ebXML)
- **EPK** (Ereignisgesteuerte Prozesskette, Modellierungssprache im SAP- und ARIS-Umfeld)
- **BPEL4People**
  BPEL mit Berücksichtigung von Benutzerinteraktionen

BPEL (1)

BPEL ist eine XML-basierte Sprache zur Beschreibung von Geschäftsprozessen, deren einzelne Aktivitäten durch Web Services implementiert sind

→ Soll in erster Linie der Orchestrierung von Web Services dienen

BPEL soll das Konzept des „Programmierens im Großen“ ermöglichen (jedoch keine Unterstützung für die Interaktionen mit einem Menschen!)

Erweiterungen:

- BPELJ
- WS-BPEL4People
  (liegt derzeit OASIS als Antrag vor)

BPEL (2)

Wichtige Merkmale:

- blockstrukturiert
- „Handler“ für Fehlerbehandlung
- „Compensation Handler“ für Kompensationen
- „Event Handler“ für Ereignisbehandlung
- BPEL kennt keine „Unterprozesse“!

BPEL (3)

Open Source BPEL Engines:

- Apache ODE („Orchestration Director Engine“)
- Intalio
- BPEL SE (Bestandteil des ESB von Sun Microsystems)

BPEL (4)

Kommerzielle Produkte (BPEL Engines):

- BPEL Process Manager (Oracle)
- XI Solution Manager / NetWeaver (SAP)
- WebSphere Process Server (IBM)
- BPWS4J (IBM)
- BizTalk Server (Microsoft)
- Business Mashup Server 2008 (Serena)

BPEL (5)

Zwei Programmiermodelle

- Executable Business Process Description

  → Orchestrierung

- Business Protocol Description

  → Choreografie

BPEL (6)

Ausdrücke und Werte

- Boolean Expressions → Wahrheitswerte
- Deadline-Valued Expressions → Stichtagswerte
- Duration-Valued Expressions → Zeitraumwerte
- General Expression

→ XPath als Sprache für die Expressions

Umgang mit Variablen

- Streng typisierte Variablen
- Wertzuweisungen

BPEL (7)

Aktivitäten

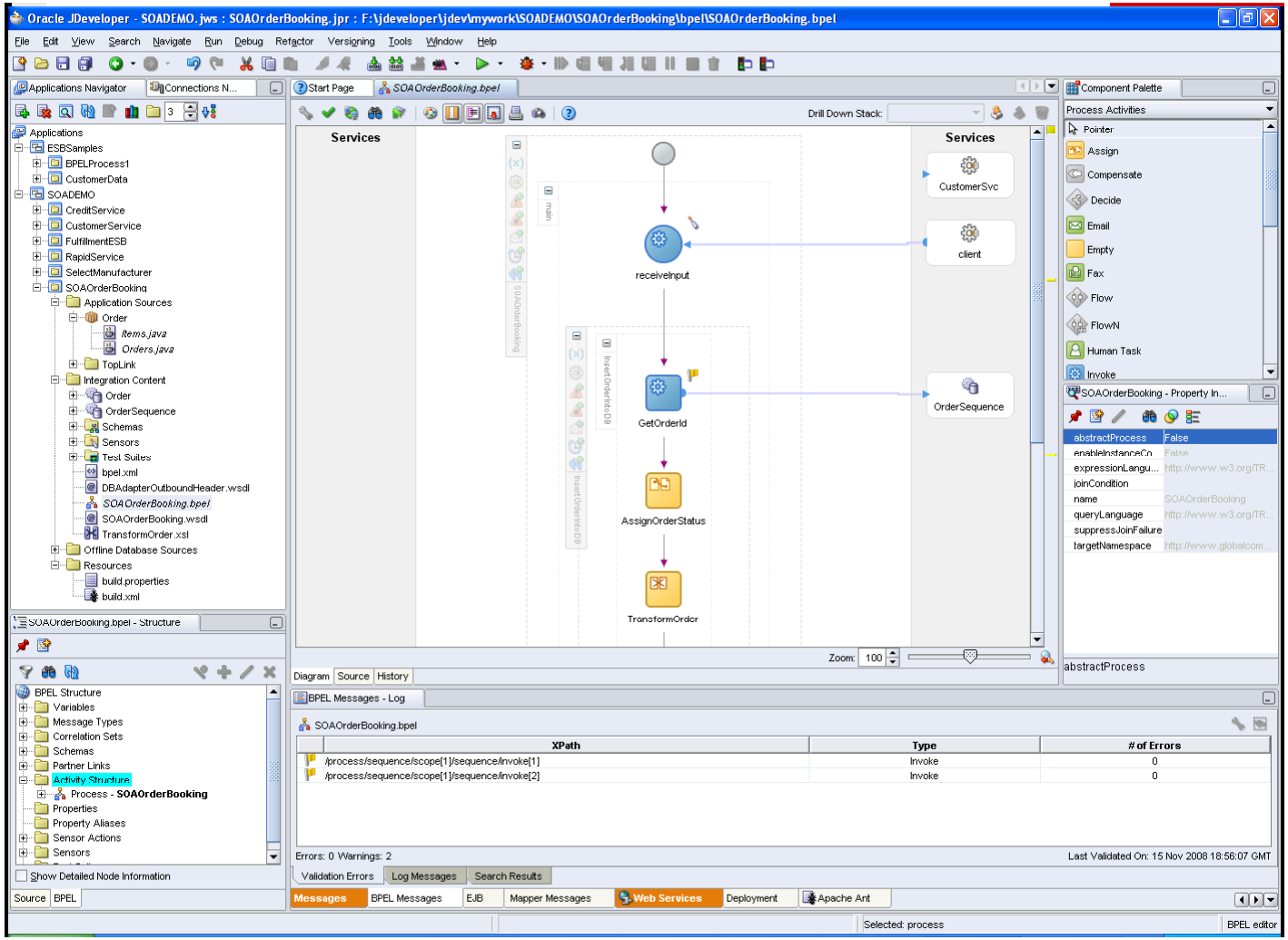
- Invoke → Aufruf von Services (synchron/asynchron)
- Receive → Warten auf Aufruf und Ergebnisrückgabe mit reply
- Reply → Rückmeldung von Ergebnissen
- Assign → Wertzuweisungen
- Throw → Auslösung von Fehlerereignissen
- Terminate → Beendigung der Verarbeitung
- Wait → wait for (warte Zeitspanne), wait until (warten bis)
- Empty → Platzhalter
- Scope → Gruppierung und Begrenzung der Fehlerausbreitung
- Compensate → Fehlerbehandlung innerhalb eines Scopes

BPEL (8)

Kontrollstrukturen

- Sequence → Folge von Aktivitäten
- While → Wiederholung solange Bedingung gültig
- Switch → Verzweigung
- Flow → Parallele Aktivitäten
- Links → Synchronisation von Aktivitäten

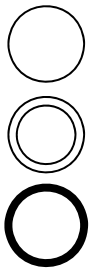
Beispiel: ORACLE



BPMN (1)

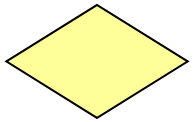
Modellierungsobjekte

- Pools und Lanes (Verantwortungsbereiche)
- Events (Ereignisse)
  - Als Startpunkt
  - zu beliebigen Zeitpunkten
  - als Endpunkt



- Prozess, Sub-Prozess, Task

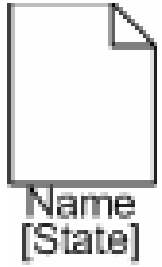
- Gateways



  - Gateways erzeugen **parallele Abläufe** und fassen diese wieder zusammen
  - **Komplexe Entscheidungen** beim Erzeugen und Zusammenfassen **möglich**

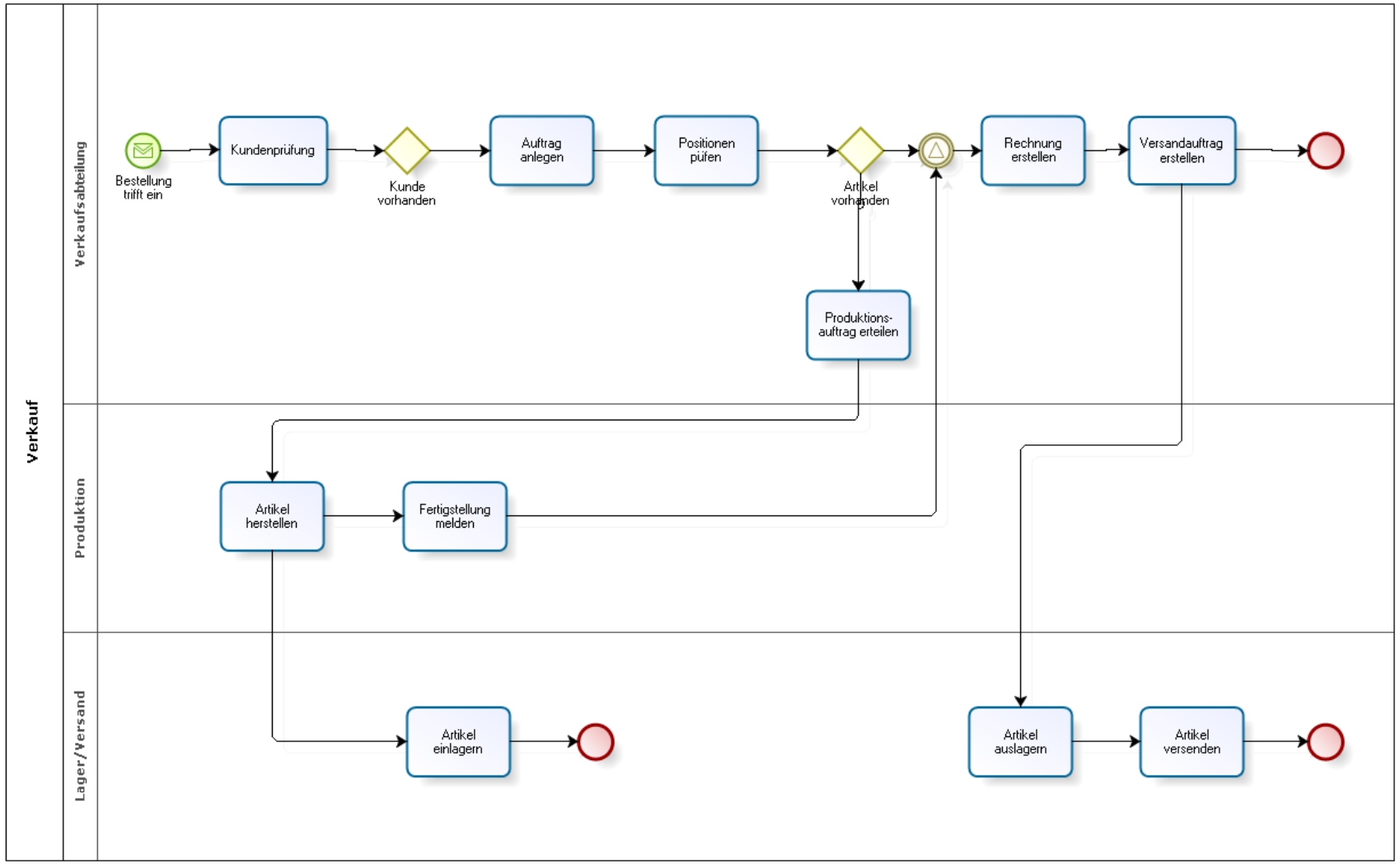
BPMN (2)

- Sequence flow
  - Innerhalb von Pools und Lanes
- Message flow
  - Nur zwischen verschiedenen Pools bzw. Lanes verschiedener Pools
- Datenobjekte
  - Datenobjekte werden als Dokumente dargestellt und sind **nur Annotationen**



  - Der Zustand kann in eckigen Klammern angegeben werden

Beispiel


Verkauf
Verkaufsabteilung
Produktion
Lager/Versand
Bestellung trifft ein
Kundenprüfung
Kunde vorhanden
Auftrag anlegen
Positionen püfen
Artikel vorhanden
Rechnung erstellen
Versandauftrag erstellen
Produktions-auftrag erteilen
Artikel herstellen
Fertigstellung melden
Artikel einlagern
Artikel auslagern
Artikel versenden

Grenzen der Modellierung

- Die Modellierung berücksichtigt in der Regel keine nicht-funktionalen Anforderungen, so z.B. keine SLAs
- Die Komplexität steigt bei Berücksichtigung von Fehlerzuständen und deren Behandlung stark an
- Bei sehr komplexen Problemen lassen sich nicht alle Aspekte adäquat abbilden
- Technische Details überfordern den Anwender in der Fachabteilung, der den Prozess aus Anwendungssicht definieren soll (die Erfahrung ist, dass sich Mitarbeiter der Fachabteilung nicht damit auseinandersetzen wollen)
- Zu grobe Beschreibungen aus Sicht der Anwender sind nicht ausreichend, um die effektive Abbildung auf die technischen Services zu automatisieren.

→ Schlußfolgerung ???

Faustregeln für die Verteilung von Anwendungslogik

- Die fachliche Logik soll nur durch Bestands- oder Funktions-Services abgebildet werden.
- Prozessservices sollen möglichst wenig Fachlogik beinhalten und nur die Koordination der erforderlichen Schritte nach definierten Regeln durchführen.
- Prozessservices müssen einen Prozesszustand verwalten.

Beispiel: Bestellabwicklung