

# **B2Bi im Überblick**

Seminar Anwendungsorientierte Middleware  
SS 2003



Vorgetragen von:

**Denis Rathig**

# Gliederung



- ◆ Einleitung
- ◆ Überblick über B2Bi Frameworks
- ◆ Aktuelle Technologien für B2Bi
- ◆ Trends bei der Unterstützung von B2Bi
- ◆ Forschungsprototypen

# Einleitung



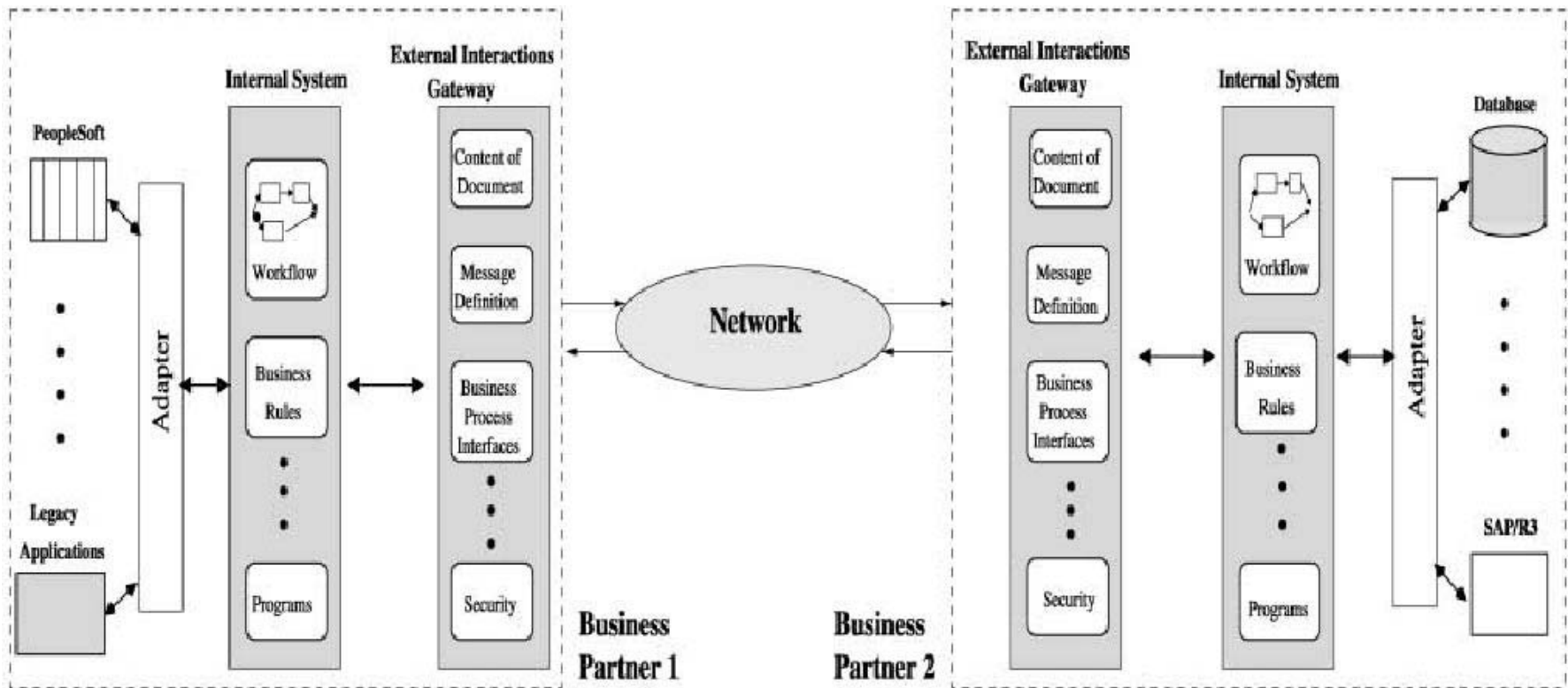
- ◆ B2B interactions
  - Definition:  
**Interaktionen** bestehen aus der Interoperation und der Integration von internen und externen Business-Anwendungen
- ◆ Probleme:
  - B2B-Anwendungen können aus autonomen, heterogenen und verteilten Komponenten zusammengesetzt sein
  - Das Umfeld von B2B-Anwendungen ist sehr dynamisch, wodurch Probleme der Änderbarkeit und Skalierbarkeit von B2B-Anwendungen auftreten können
- ◆ Die zugrunde liegende Arbeit schlägt ein Framework vor, um B2Bi Lösungen vergleichend bewerten zu können, dazu dienen:
  - eine 3-Schichten-Architektur, sowie
  - unterschiedliche Dimensionen

# Gliederung

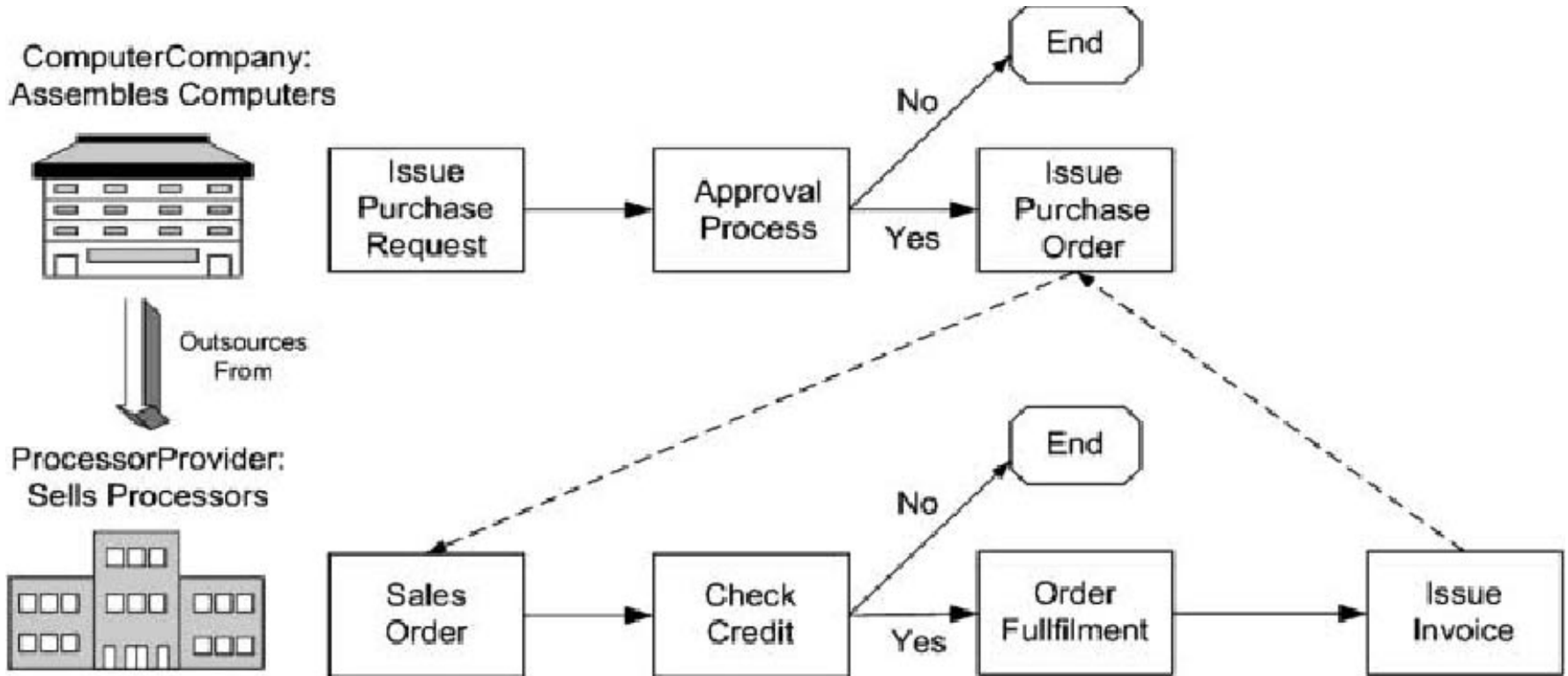


- ◆ Einleitung
- ◆ **Überblick über B2Bi Frameworks**
- ◆ Aktuelle Technologien für B2Bi
- ◆ Trends bei der Unterstützung von B2Bi
- ◆ Forschungsprototypen

# Architektur von B2Bi Frameworks



# Beispiel



# Schichten in B2Bi Frameworks

Interaktionen in B2B Anwendungen treten in drei Schichten auf:

- ◆ Die **Kommunikations-Schicht** stellt Protokolle zur Verfügung, um Nachrichten zwischen Partnern austauschen zu können (z.B. HTTP, SOAP).
  - Ziel ist eine nahtlose Integration der Kommunikationsprotokolle
- ◆ Die **Inhalts-Schicht** stellt Sprachen und Modelle zur Verfügung, um Informationen so beschreiben und organisieren zu können, dass sie von den Partnern verstanden und benutzt werden können. Für Interaktionen in der Inhalts-Schicht ist es notwendig, dass die beteiligten Systeme die Semantik Inhalte und Arten der Business-Dokumente kennen.
  - Ziel ist eine nahtlose Integration von Daten-Formaten, Daten-Modellen und Sprachen.
- ◆ Die **Geschäftsprozess-Schicht** beschäftigt sich mit Konversations-Interaktionen (z.B. vereinigte Geschäftsprozesse) zwischen Services.
  - Ziel ist es zu ermöglichen, dass autonome und heterogene Partner online ihre Möglichkeiten und Bedingungen für eine Zusammenarbeit anzeigen und auf dieser Grundlage Peer-to-Peer-Interaktionen mit beliebigen Partner starten können. Dazu muss die Semantik der Geschäftsprozesse des Partners bekannt sein.

# Bewertungsdimensionen in B2Bi Frameworks I

- ◆ **Kopplung** zwischen Partnern: Maß für die Stärke und Dauer der Kopplung von Businesspartnern
  - **starke Kopplung:** Große gegenseitigen Abhängigkeiten
  - **lose Kopplung:** Austausch von Business-Informationen *On Demand*
  - **Vorübergehende Dauer:** schnelle und kurze Partnerschaft (z.B. nur eine Transaktion)
  - **Lange Dauer:** Geschäfte mit vorausgehender Partnerschaft
- ◆ **Heterogenität:** Maß für die Verschiedenheit von Businesspartnern. Anwendungen benutzen verschiedene Datenstrukturen (z.B. XML, relationale Datenbanken) Standard oder proprietäre Semantiken (z.B. Standardontologien). Es kann auch strukturell Unterschiede in der Geschäftsprozess-Schicht (z.B. Benutzung von APIs, Dokument-Austausch-Protokollen, Inter-Enterprise-Workflows).



# Bewertungsdimensionen in B2Bi Frameworks II

---

- ◆ **Autonomie:** Maß der Einhaltung globaler Kontrollregeln.  
Partner-Systeme können bezüglich ihres Designs, ihrer Kommunikation und Ausführung autonom sein. Bei vollständig autonomer Kollaboration wird jeder Partner als Blackbox angesehen, die Information austauschen kann (also Nachrichten empfangen und versenden kann). Partner interagieren über wohldefinierte Schnittstellen, die ihnen erlauben stärkere Kontrolle über eigene Implementierungen und den Betrieb von Services auszuüben, und ihnen die Flexibilität geben, die eigenen Prozesse zu verändern ohne die Zusammenarbeit zu beeinträchtigen.
- ◆ **externes Management:** Maß der externen Sichtbarkeit und Veränderbarkeit der Anwendungen von Partnern.  
Um effektiv von den Partner benutzt werden zu können, muss eine Anwendung so definiert sein, dass die Überwachung und Kontrolle ihrer Ausführung, die Messung ihrer Performanz und Vorhersagbarkeit von Status und Verfügbarkeit möglich ist.

# Bewertungsdimensionen in B2Bi Frameworks III

---

- ◆ **Anpassungsfähigkeit:** Maß für die Möglichkeit der schnellen Änderung einer Anwendung.

B2B-Anwendungen operieren in einem sehr dynamischen Umfeld wo ständig neue Services verfügbar, existierende Services entfernt und der Inhalt bzw. die Fähigkeiten von Services verändert werden. Firmen müssen schnell auf solche Änderungen reagieren können, sowohl das operationelle Umfeld (z.B. Serverlast) als auch der Markt (z.B. Änderung der Verfügbarkeit, oder der Benutzeranforderungen) sind nicht vorhersagbar.

- ◆ **Sicherheit:**

Sicherheit ist für Inter-Enterprise-Anwendungen von besonderer Wichtigkeit. Bevor B2B E-Commerce sein wahres Potenzial erreichen kann, müssen geeignete Sicherheitsmaßnahmen ergriffen werden, um das Vertrauen der E-Commerce-Partner in sichere Transaktionen herzustellen.

B2B-Anwendungen müssen gegenseitige Authentifikation, ungestörte Kommunikation, Vertraulichkeit, Nicht-Zurückweisbarkeit und Autorisierung gewährleisten, da B2Bi auf geringem gegenseitigen Vertrauen und keinen bzw. geringen Kenntnissen über den Partner basieren können.

# Bewertungsdimensionen in B2Bi Frameworks IV



- ◆ **Skalierbarkeit:**

Fähigkeit eines System in unterschiedliche Dimensionen zu wachsen, z.B. Umfang der zugänglichen Daten, Anzahl der in einem Zeitraum unterstützten Transaktionen und Anzahl der möglichen Beziehungen zu Partnern.

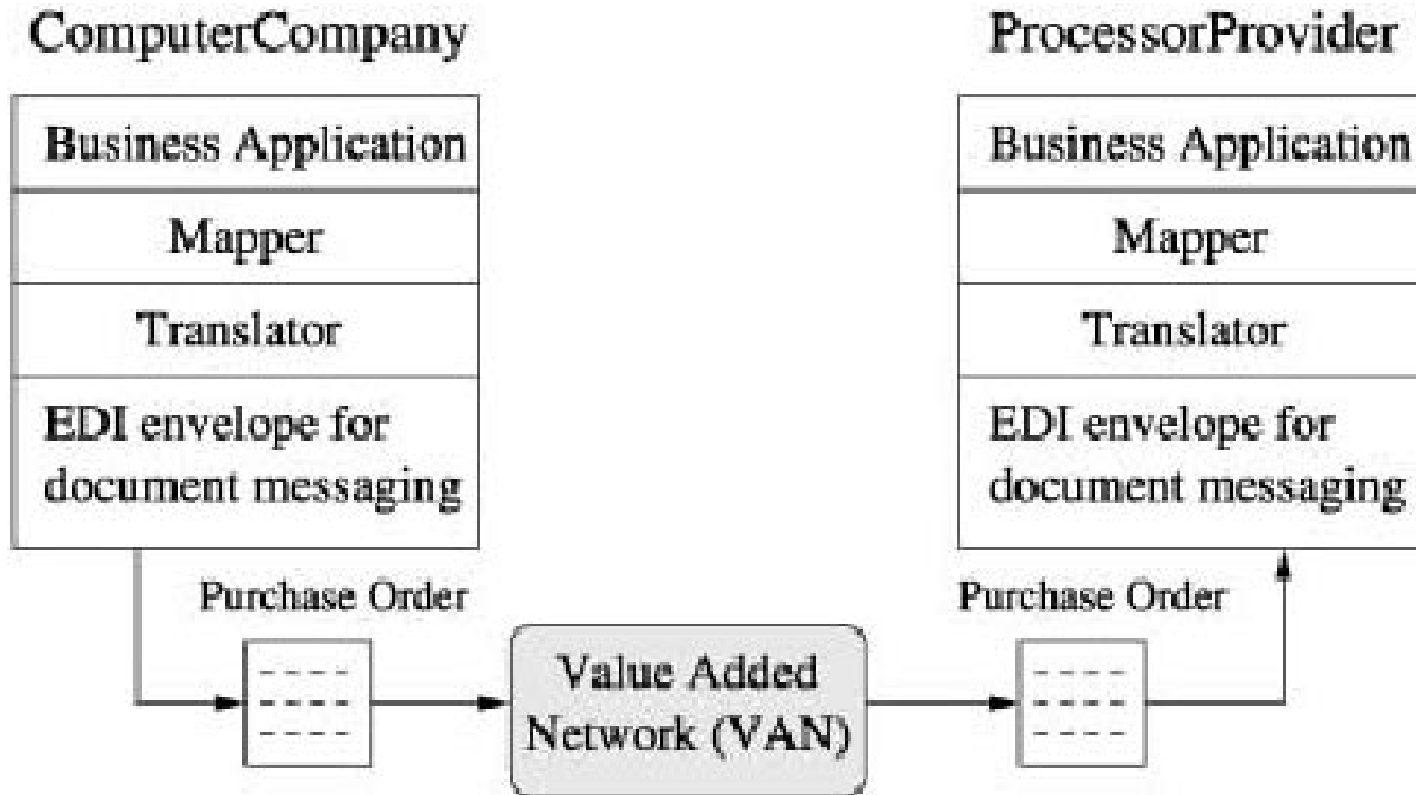
Kosten und Aufwand zur Herstellung neuer Beziehungen zu Partnern sind ein wichtiges Kriterium bei der Bewertung Interaktions-Lösungen in B2B E-Commerce

# Gliederung



- ◆ Einleitung
- ◆ Überblick über B2Bi Frameworks
- ◆ **Aktuelle Technologien für B2Bi**
- ◆ Trends bei der Unterstützung von B2Bi
- ◆ Forschungsprototypen

# Electronic data interchange (EDI) I



# Electronic data interchange (EDI) II

## ◆ **EDI**

- Interoperabilität in der
  - Kommunikations-Schicht durch VANs
  - Inhaltsschicht durch Definition einer großen (aber festen) Anzahl von Business-Dokumenten durch Komitee
  - Geschäftsprozess-Schicht durch vordefinierte Geschäftsprozesse (nach Kaufauftrag erwartet ComputerCompany Kauf-Bestätigung, Rechnung und Lieferung)
- Stärken von EDI bei Sicherheit und Heterogenität
- starke Kopplung, geringe Autonomie, Anpassungsfähigkeit gering

## ◆ **EDIINT (EDI over the Internet )**

- Internet zur Kommunikation → Kostenreduktion
- Standards zum Dokumentenaustausch über SMTP und HTTP
- Sicherheit: Verschlüsselung und digitale Signierung von Dokumenten mit PGP
- Im Gegensatz zu EDI kein QoS, z.B. automatische Fehlererkennung und -korrektur durch Verzicht auf VAN
- übrige Dimensionen wie bei EDI

# Electronic data interchange (EDI) III

- ◆ **OBI** (Open Buying on the Internet)
  - HTTP zum Austausch von Nachrichten
  - EDI Standard zur Beschreibung des Dokumentinhalts
  - Dokumente in OBI-Objekte gekapselt, die auch nicht-EDI-Nachrichten, wie digitale Signaturen enthalten können.
  - Einfaches Protokoll in der Geschäftsprozess-Schicht, bestehend aus üblichen Aktivitäten wie „wähle Anbieter“ oder „generiere Kaufauftrag“
  - Sicherheit: SSL (*Secure Sockets Layer*) over HTTP, digitale Signaturen und digitale Zertifikate um Authentizität und Integrität von Nachrichten zu gewährleisten.
  - OBI besser als EDI bezüglich Skalierbarkeit und Anpassungsfähigkeit.
  - übrige Dimensionen wie bei EDI und EDIINT

# Komponenten I



## ◆ **Komponenten**

- decken hauptsächlich Interaktionen der Kommunikationsschicht ab
- fokussieren darauf, heterogene Anwendungen syntaktisch zu integrieren
- unterstützen die Zusammensetzung von Anwendungen durch unabhängig entwickelte Komponenten in der Geschäftsprozess-Schicht

## ◆ **CORBA**

- Kopplung stark und langfristig
- Kapselt mögliche Heterogenität
- externes Management durch *event service*
- kaum Unterstützung für Anpassbarkeit
- Sicherheit durch *security service*. Mit Authentifikation, Autorisierung und Verschlüsselung von Nachrichten



# Komponenten II



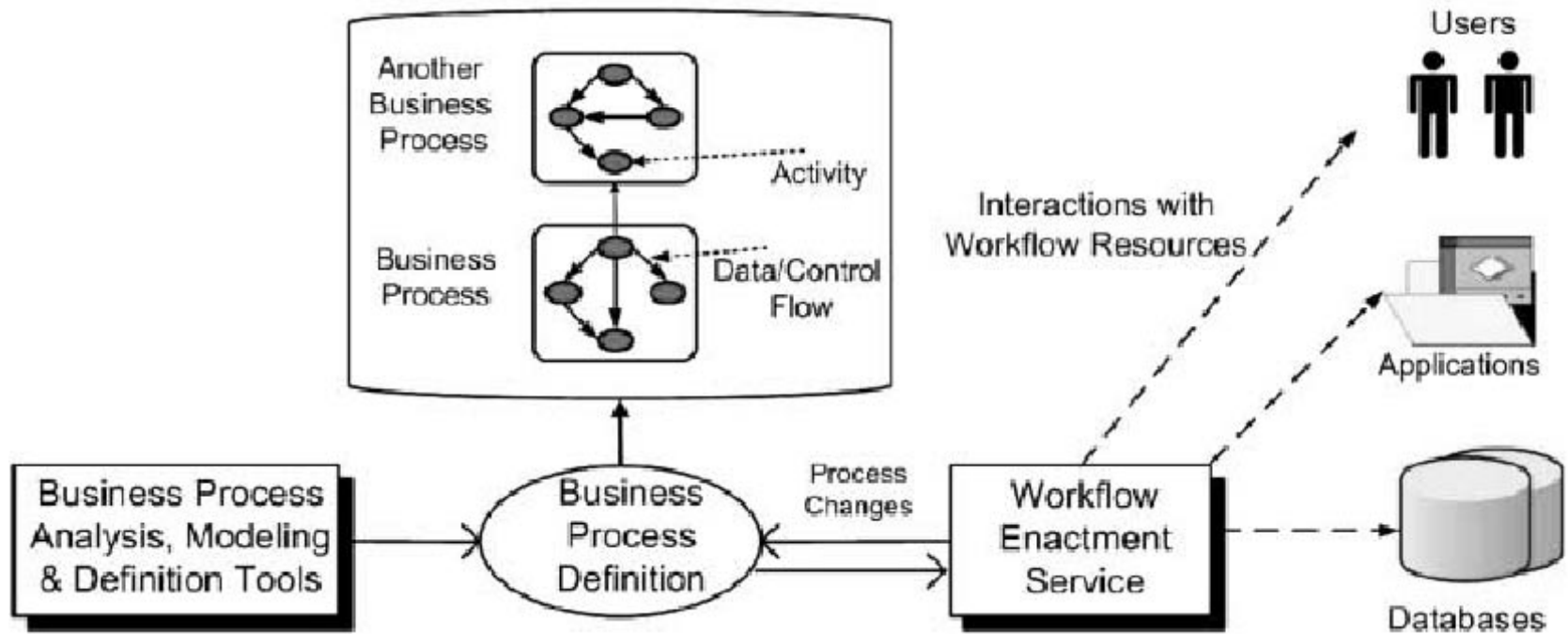
## ◆ **DCOM**

- Heterogenität: meistens auf Windows basierend, Anwendungen meistens in Microsoft's J++, C, C++, oder Visual Basic
- Sicherheit basiert auf dem Windows NT Sicherheitsmodell; damit ist Qualität der Sicherheit nur für Windows Plattformen bekannt
- Dimensionen Autonomie, externe Änderbarkeit, Anpassungsfähigkeit und Skalierbarkeit wie bei CORBA

## ◆ **EJB**

- Heterogenität: die meisten EJB Implementationen bieten keine Interoperabilität mit nicht-Java Plattformen
- Verschiedene Implementationen für event services um externes Management zu unterstützen.
- Erleichterung der Anpassbarkeit durch einen *deployment descriptor* für jedes Bean
- andere Dimensionen ähnlich wie bei CORBA und DCOM

# Workflows I



# Workflows II



## ◆ ***The workflow reference model***

- entwickelt von der Work-flow Management Coalition (WfMC)
- beinhaltet standardisierte Interfaces und Datenaustauschformate zwischen Workflow-System-Komponenten
- legt Schwerpunkt auf die syntaktische Integration von Workflow Prozessen, bietet wenig Unterstützung für Inter-Enterprise-Geschäftsprozesse

## ◆ ***Distributed workflow systems (DWSs)***

- teilt eine globale Workflow Spezifikation in mehrere Sub-Workflows, die jeweils von Teilen einer Organisation ausgeführt werden
- alle Teilnehmer an dem globalen Workflow müssen das gleiche Workflowmodell umsetzen
- geht davon aus, dass globale und Sub-Geschäftsprozesse die gleichen Prozess-Definitionen und gleiche Datenaustauschmodelle nutzen
- geeignet um den Geschäftsprozess einer Organisation aus mehreren, verteilten Sub-Workflows zu integrieren

# Workflows III



## ◆ ***Collaborative process management***

- Interorganisationale Geschäftsprozesse durch Trennung in öffentliche und private Prozesse
- öffentlicher Prozess: definiert einen externen Nachrichtenaustausch mit Partner basierend auf einem entsprechenden Protokoll, z.B. EDI oder RosettaNet.
- privater Prozess: beschreibt eine intern ausführbare Aktivität, die die Aktivität eines öffentlichen Prozesses unterstützt
- *process wrapper* zwischen öffentlichen und privaten Prozessen
- *application adapter* zwischen privaten Prozessen und Back-End-Anwendungen
- Trennung sorgt für gute Skalierbarkeit, Autonomie und Überwindung von Heterogenität
- externe Änderbarkeit kann durch das Hinzufügen spezieller Aktivitäten in öffentlichen Prozessen erreicht werden

# Gliederung



- ◆ Einleitung
- ◆ Überblick über B2Bi Frameworks
- ◆ Aktuelle Technologien für B2Bi
- ◆ **Trends bei der Unterstützung von B2Bi**
- ◆ Forschungsprototypen

# eCO I



## ◆ eCO

- soll das Auffinden und Benutzen von Services ermöglichen, unabhängig von den E-commerce Standards und Protokollen die potenzielle Partner nutzen
- Inhalts-Schicht: xCBL (*XML Common Business Library*) um Business Dokumente zu definieren. xCBL besteht aus einem Satz von XML Kern-Dokumenten die übliche Interaktionen bei Business-Transaktionen repräsentieren. Z.B.: Bestellungen, Rechnungen. Business Partner können diese Dokumente erweitern und damit an eigene Bedürfnisse anpassen, oder auch selber Dokumente entwickeln, müssen die anderen Partner aber auf diese aufmerksam machen, um die Interoperabilität nicht zu stören
- Geschäftsprozess-Schicht: Businesses beschreiben ihre Services mit *Business Interface Definitions (BIDs)*. Diese XML-Dokumente beschreiben welche Dokument durch einen Service akzeptiert und produziert werden

# eCO II



- Durch die Benutzung von xCBL und der entsprechenden Dokumente durch alle Partner ist strukturelle **Heterogenität** kein Problem
- **Autonomie** wird durch Trennung der Beschreibung und der Implementierung von Services erreicht
- **Sicherheitsmechanismen** sind optional, durch die dokument-basierten Interaktionen müssen die Partner allerdings nicht direkt auf Partnersysteme zugreifen
- Das eCO-Framework zeichnet sich durch gute **Anpassbarkeit** aus. Lokale Änderungen haben keine Auswirkungen auf Partnersysteme.

# BizTalk I



## ◆ **BizTalk**

- stützt sich auf ein zentrales Schema-Repository und eine Schichten-Architektur
  - das Schema-Repository wird zum Publizieren und Validieren XML-basierender Schemas und deren Versionsverwaltung benutzt
  - Die Architektur besteht aus drei Schichten: *Anwendung*, *BizTalk Framework Compliant (BFC) Server* und *Transport*. Anwendungen kommunizieren miteinander durch die Versendung von Business-Dokumenten über BFC Server
- Kommunikations-Schicht: HTTP, SMTP und Microsoft Message Queue (MSMQ)
- Geschäftsprozess-Schicht: spezielles Biz-Tag enthält Informationen über den Geschäftsprozess, der ein bestimmtes BizTalk Dokument verarbeiten kann.
- *BizTalk Orchestration* soll genutzt werden können, um Inter-Enterprise-Prozesse zu definieren und auszuführen, ist aber noch nicht ausgereift



# BizTalk II



- BizTalk's Repository ist nicht gut **skalierbar**
- BizTalk unterstützt S/MIME (version 3) um BizTalk Nachrichten zu sichern
- BizTalk nutzt spezielle Biz-Tags (z.B. Empfangsbestätigungen) um verlässlichen Transport von BizTalk Dokumenten zu gewährleisten, eine Voraussetzung für später mögliches **externes Management**

# Commerce XML (cXML) I

## ◆ cXML

- besteht aus XML-Schema basierter Sprache und einem Protokoll für Online-Transaktionen
- kann als vereinfachte XML und Internet-basierte Version von EDI angesehen werden.
- setzt die Existenz von *E-commerce-Hubs*, die als vertrauenswürdige Dritte zwischen Beschaffungs-Systemen und Anbieter-Systemen auftreten, voraus. Die Funktion eines E-commerce-Hub ähnelt der des BizTalk-Repository, ohne eine bestimmte Architektur vorzuschreiben.
- zwei Kommunikationsmodelle: *request response (synchron)* über HTTP und *one-way (asynchron)* über HTTP oder andere Protokolle, zur Zeit nur URL Form Encoding.

# Commerce XML (cXML) II

- Inhalts-Schicht: cXML definiert einen Satz von XML DTDs zur Beschreibung von Dokumenten, z.B. Bestellungen-Anfragen oder -Bestätigungen. Folgende Elemente zur Beschreibung von Produkten sind vorgesehen: *Anbieter*, *Index*, and *Vertrag*
  - das Anbieter-Element beschreibt Informationen über einen Anbieter (z.B. Adresse, Bestell-Methoden)
  - das Index-Element beschreibt das Inventar des Anbieters (z.B. Produktbeschreibungen und Artikelnummern)
  - das Vertrags-Element beschreibt Verhandlungsergebnisse zwischen Käufer und Verkäufer z.B. bzgl. Preisen oder Mengen
- Geschäftsprozess-Schicht: cXML definiert ein Protokoll, das aus üblichen Aktivitäten, wie z.B. Produktauswahl, Produktbestellung und Bestellbestätigung besteht. E-commerce-Hubs leisten Katalog- und Bestellungen-Management (z.B. Katalog-Veröffentlichung, automatisierte Verfolgung und Überwachung von Kaufträgen)

# Commerce XML (cXML) III



- cXML gut bzgl. Heterogenität, geringe Autonomie und geringe Anpassbarkeit
- Skalierbarkeit besser, da die Kosten von Integration von Partnern bei XML-basierten Frameworks niedriger sind als bei anderen und sowohl Anbieter als auch Käufer nur ihre Beziehungen zu den *E-commerce-Hubs* pflegen müssen und nicht mit allen Business-Partnern
- Sicherheit durch Authentifizierungs-Informationen im Nachrichten-Header.

# RosettaNet I

## ◆ *RosettaNet*

- Ziel: Standardisierung von Produktbeschreibungen und Geschäftsprozessen für Anwendungen des IT-Bestellwesens, das Produkte wie z.B. Mainboards, Peripherie und fertige Systeme und Komponenten wie z.B. Chips und Kabel umfasst
- drei Schwerpunkte bei der Standardisierung für automatische B2Bi:
  - einheitliches Vokabular: Der *RosettaNet Business Dictionary* enthält Vokabeln zur Beschreibung von Geschäfts-Eigenschaften (z.B. Name, Adresse, Steuernummer). Der *RosettaNet Technical Dictionary* enthält Vokabeln zur Beschreibung von Produkteigenschaften (z.B. Bauteile) and Services (z.B. Bestellung)
  - Spezifizierung des Transports von Business-Nachrichten: Das *RosettaNet Implementation Framework* spezifiziert den Inhalt von Nachrichten, Transport-Protokolle (HTTP, eMail, SSL) für Kommunikation und Sicherheitsmechanismen (digitale Zertifikate, digitale Signaturen)
  - Spezifizierung und Harmonisierung des Geschäftsprozesses zum Austausch von Business-Nachrichten. *RosettaNet's PIPs (Partner Interface Processes)* sind vordefinierte XML-basierte *conversations*. Eine *conversation* besteht aus einem Satz von Business-Dokumenten (Bestellung, Bestellungsbestätigung) und einer Nachrichtenaustausch-Logik (z.B. Reihenfolge der Aktionen während der Durchführung einer Bestellung)

# RosettaNet II



- Kommunikations-Schicht: Internet-Transport-Protokolle werden unterstützt
- Inhalts-Schicht: Dokument-Model mit auf XML-basierendem Schema
- Geschäftsprozess-Schicht: RosettaNet liefert eine gemeinsame Basis für B2Bi über PIPs
- **Sicherheit:** wie OBI, also ok
- **Autonomie, Anpassbarkeit, Skalierbarkeit, Kopplung und externes Management** wie eCO, ebenfalls gut

# Electronic Business XML (ebXML) I

- ◆ ebXML (*Electronic Business XML*)
  - Infrastruktur basiert auf einem *Repository*, in dem Information über Firmen, die von ihnen angebotenen Produkte und Services gespeichert werden
  - Kommunikations-Schicht: Firmen tauschen Nachrichten über *Messaging Service* aus, ohne vorgeschriebenes Protokoll, unterstützt werden SMTP, HTTP und FTP
  - Inhalts-Schicht: Firmen interagieren über *Business Dokumente*. Ein *Business Dokument* ist ein Satz von Informations-Komponenten die als Teil eines Geschäftsprozess ausgetauscht werden. Drei Typen von Komponenten:
    - **core components** sind Informations-Komponenten die für alle Industrie-Zweige nützlich sind. Sie werden von ebXML bereitgestellt und in der *core library* gespeichert
    - **domain components** und **business information objects** sind größere Komponenten, die von spezifischen Industriezweigen oder Firmen bereitgestellt und in der *domain library* bzw. *business library* gespeichert werden.

# Electronic Business XML (ebXML) II

- Geschäftsprozess-Schicht: ebXML definiert Schemata für Geschäftsprozess-Spezifikationen in UML oder XML
  - UML-Version besteht aus UML-Klassendiagramm, das nicht für die direkte Erstellung einer Geschäftsprozess-Spezifikationen gedacht ist, sondern alle für diese Erstellung notwendigen Elemente und Beziehungen darstellt
  - XML-Version ermöglicht die Erstellung von ebXML-konformen Geschäftsprozess-Spezifikationen als XML-Dokument
  - ebXML stellt einen Satz von Geschäftsprozess-Spezifikationen zur Verfügung der von unterschiedlichen Industriezweigen genutzt werden kann. Die Spezifikationen werden in der *business library* gespeichert und können von Firmen an eigene Geschäftsprozesse angepasst werden.
  - Interaktionen zwischen Geschäftsprozessen werden durch *Choreographien* dargestellt. Eine *Choreographie* spezifiziert die mögliche Reihenfolge von Geschäfts-Transaktionen.
  - ebXML definiert *collaboration protocol agreements* (CPAs) mit denen zwei Handelspartner bereits vor einem Handel die Bedingungen zu denen der Handel ablaufen soll festlegen (z.B. Liefer- oder Zahlungsmöglichkeiten)



# Electronic Business XML (ebXML) III



- **Sicherheit:** sichere und verlässliche Kommunikation durch SSL und S-HTTP. Authentizität durch digitale Signatur für einzelne oder eine Gruppe zusammenhängender Nachrichten
- **Autonomie** und **Anpassbarkeit** gut
- **Externes Management** kann durch spezielle Aktivitäten in gemeinsamen Geschäftsprozessen hinzugefügt werden
- **Skalierbarkeit:** ebXML will verteilte Repositories unterstützen, bis jetzt ist aber nur ein Repository spezifiziert

# Web Services I

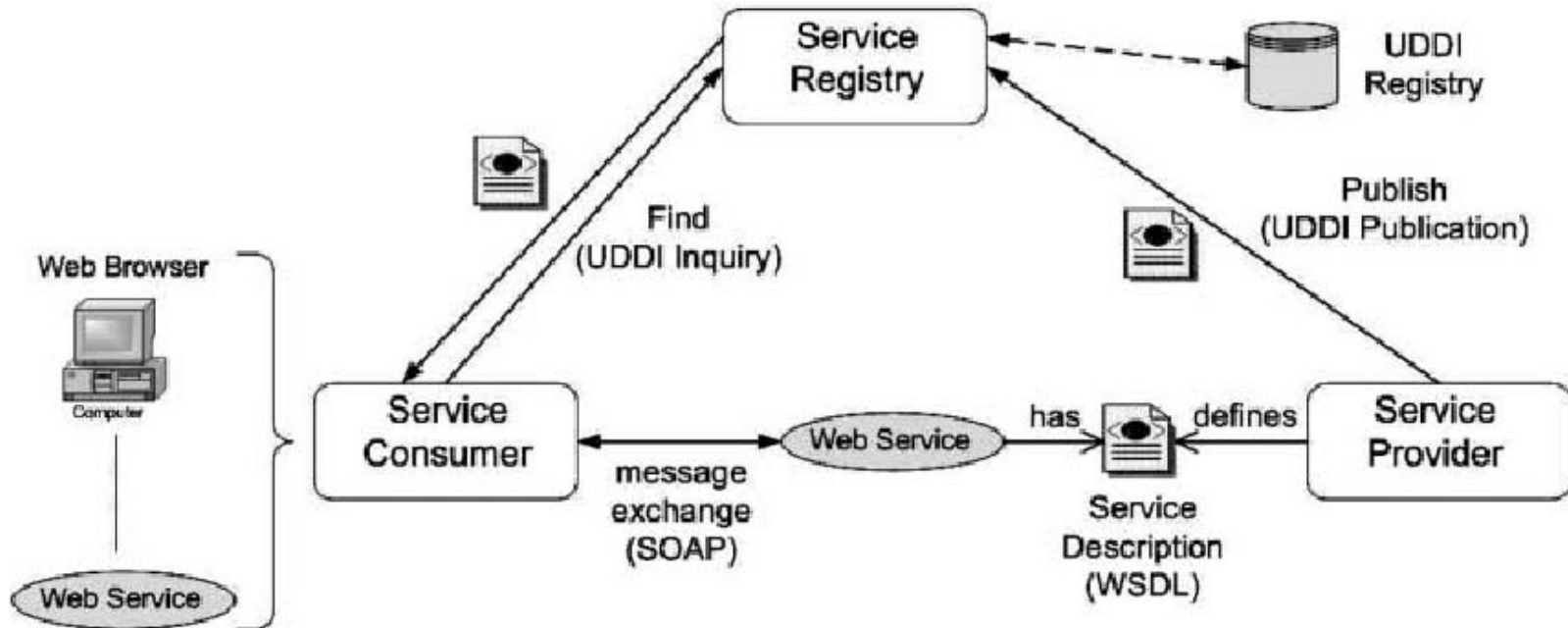


## ◆ **Web Services**

- hier: Business-Funktionen die folgende Eigenschaften haben:
  - *programmatisch zugreifbar*: Web Services werden hauptsächlich designed um von anderen Web Services oder Programmen aufgerufen zu werden. Sie sind über des Web verteilt und über verbreitete Protokolle wie HTTP und SMTP zugreifbar. Web Services müssen ihre Fähigkeiten, inkl. ihrer Operationen, Input und Output Nachrichten, und wie sie aufgerufen werden können, beschreiben
  - *lose gekoppelt*: Kommunikation zwischen Web Services ist *dokumentenbasiert*, in der Regel XML-basiert. Diese Art der Kommunikation sorgt für die lose Kopplung zwischen den Web Services

# Web Services II

## ◆ *The Web service reference model*



# Web Services III



## ◆ *B2Bi in Web Services*

- **Kopplung:** Web Services sind nicht statisch aneinander gebunden
- **Heterogenität:** heterogene Anwendungen (z.B. Java, CORBA-Objekte) können als Basis von Web Services dienen.
- **Autonomie:** Web Services werden durch veröffentlichte Interfaces zugreifbar.
- **externes Management:** WSDL enthält zu Zeit keine Methoden um den Status eines Web Services oder einer Operation abzufragen
- **Anpassungsfähigkeit:** zur Zeit nur manuell möglich.

# Web Services IV



- **Sicherheit:** Probleme auf drei Ebenen
  - **communication level:**
    - sicheres XML durch einen Standard für digitale Signaturen in XML
    - S-HTTP, ein Protokoll, mit dem Daten sicher über das Web transferiert werden können
    - Erweiterung von SOAP um Autorisierung des Erhalts und der Verarbeitung von Nachrichten
  - **service description level:**
    - WSDL enthält keine rollen-basierten Zugriffskontroll-Informationen oder Informationen für die Autorisation
  - **firewalls**
    - da SOAP Nachrichten über HTTP durch Firewalls nicht zurückgehalten werden, können böartige requests (z.B. mit eingebettetem böartigem Code) durch Firewalls gelangen
- **Skalierbarkeit:**
  - viel Traffic bei starker Verbreitung von Web Services kann zu Engpässen führen
  - zentrale Registry kann bei Ausfall eine Schwachstelle und ein Flaschenhals für den Zugriff auf und die Publizierung von Web Services sein. Eine verteilte Registry wäre verlässlicher und besser skalierbar aber auch mit entsprechendem Synchronisierungs-Aufwand verbunden

# Gliederung



- ◆ Einleitung
- ◆ Überblick über B2Bi Frameworks
- ◆ Aktuelle Technologien für B2Bi
- ◆ Trends bei der Unterstützung von B2Bi
- ◆ **Forschungsprototypen**

# Collaboration Management Infrastructure I

## ◆ *CMI (Collaboration Management Infrastructure)*

- Architektur für inter-enterprise workflows
- Hauptkomponenten der CMI-Engine: *CORE*, *coordination* und *awareness engines*:
  - *CORE engine* stellt Konstrukte zum Definieren von Ressourcen, Rollen und generic state machines bereit, auf denen die *coordination* und *awareness engines* aufbauen können
  - CMI's *coordination model* erweitert traditionelle Workflow-Koordinations-Elemente mit fortgeschrittenen Elementen, wie *Platzhaltern*. Ein *Platzhalter* ermöglicht dynamische Handelsbeziehungen. Eine *Platzhalter-Aktivität* wird zur Laufzeit durch eine konkrete Aktivität ersetzt, die den gleichen Input und Output wie die Aktivität des Platzhalters hat. Eine Auswahl-Policy muss spezifiziert werden, um die auszuführende Aktivität zu bestimmen. Wenn es mehrere Anbieter für Implementierungen eines Aktivitäts-Interface gibt, kann die Auswahl-Policy sein, die Implementation mit der besten Quality-of-Service auszusuchen
  - CMI's *awareness model* liefert Informationen bzgl. der Rolle oder der Situation eines Prozess-Teilnehmers.

# Collaboration Management Infrastructure II



- **Kopplung:** Handelspartner sind stark gekoppelt. Z.B. Nachrichtenformat und Kommunikationsprotokoll müssen vor der Definition der Aktivität eines Services festgelegt werden
- **Heterogenität** wird durch Proxies ermöglicht, die den Zugriff auf unterschiedliche Informationsquellen, wie relationale Datenbanken, EJBs, und CORBA-Objects erlauben
- **externes Management:** CMI stellt anwendungsspezifische Zustands-Maschinen and Operationen für die Veränderung von Services zur Verfügung. Damit ist das selektive Beobachten von Zustandsänderungen in externen Services möglich
- **Sicherheit:** nur durch rollenbasierte Prozess- und Aktivitätsausführung



# eFlow



## ◆ *eFlow*

- Plattform die die Spezifikation, Komposition und Management von zusammengesetzten Services unterstützt
- Heterogenität: eFlow stellt Adapter für Services zur Verfügung, um die Nutzung verschiedener B2Bi-Protokolle wie OBI und RosettaNet zu ermöglichen

# WebBase of Internet-accessible Services I

## ◆ **WebBIS** (*WebBase of Internet-accessible Services*)

- führt deklarative Sprache für Komposition von Web Services ein. Drei Typen von Services werden eingeführt:
  - **wrapper service**: Objekt das den Inhalt und die Fähigkeiten der zugrunde liegenden Anwendung kapselt. *translators* werden benutzt um auf Operationen proprietärer Anwendungen zuzugreifen. Ein wrapper service beinhaltet einen Satz von *notifications*, die Ereignisse beschreiben, über die Benutzer eines Service informiert werden können, z.B. über den Status des Services oder den Stand der Bearbeitung einer Anfrage
  - **pull-community**: ermöglicht die Komposition von Services aus bereits bestehenden. Methoden der pull-community werden durch das Aufrufen der internen Operationen der Komponenten-Services ausgeführt. Eine pull-community kann sich zur Benachrichtigung über Ereignisse bei den Komponenten-Services anmelden
  - **push-communities**: ermöglichen die Herstellung dynamischer Beziehungen zwischen Services. Sie beschreiben die Fähigkeiten eines erwünschten Services, ohne auf einen bestimmten Anbieter zu verweisen. Services können sich bei einer push community anmelden, um von diesen berücksichtigt zu werden.

# WebBase of Internet-accessible Services II



- Anpassbarkeit: Alle Änderungen an einem Service werden anderen Services, die den geänderten Service benutzen gemeldet, um globale Konsistenz zu gewährleisten. Wenn ein Komponenten-Service gelöscht wird, werden Operationen oder Ereignisse von pull-communities, die sich auf diesen Service beziehen als nicht verfügbar gekennzeichnet. Dafür wird ein Meta-Service definiert, der *change notifiers* genannt wird. Jeder Service ist an einen *notifier* gebunden. Der *notifier* hält Informationen über die Verfügbarkeit des Service. Eine Veränderung der Verfügbarkeit erzeugt einen Event, der an die entsprechenden Services geleitet wird, um auf die Veränderung reagieren zu können.