

EDI

Electronic
Data
Interchange
(Elektronischer Datenaustausch)

Klassisches EDI - der Kern

Einleitung - die Kernkomponenten
File Transfer- und Messaging-Standards
UN/EDIFACT und EANCOM im Detail
Applikationsschnittstellen
Konverter- und Mappingtechniken

EDI: Die Kernkomponenten



- EDI-Standardaustauschformate
- Applikationsschnittstellen
- Mapping
- Routing
- Messaging / File Transfer
- Extras
 - Archivierung
 - Reporting
 - Alarmierung
 - Tracking & Tracing

26.06.2002

EDI - SS 2002

3



Applikationsschnittstellen

Schnittstellenarten

Die SAP IDoc-Schnittstelle - ein
"prominentes" Beispiel

Methoden der Inhouse-Anbindung



- Dateischnittstellen
 - die traditionelle Methode
- Austausch von Speicherstrukturen durch IPC-Verfahren
 - für zeitkritische Anforderungen
- Direkte Kopplung über Datenbankzugriffe auf Applikationstabellen
 - nur in speziellen Umgebungen geeignet
 - risikoreich aus Applikationssicht
 - nicht gerade modular
 - elegant, wo der Einsatz vertretbar ist

26.06.2002

EDI - SS 2002

5

Dateischnittstellen: Datenformat-Arten



- *Fixed record*-Strukturen
 - Positiv:
 - einfach
 - schnell in der Verarbeitung
 - gut zu parsen
 - Negativ:
 - unflexibel bei späteren Anwendungen
 - verschwendet viel Platz
 - Beispiel:
 - SAP IDocs

26.06.2002

EDI - SS 2002

6

- *Variable record*-Strukturen
 - Positiv:
 - Kompakt
 - Flexibel bei Feldlängenänderungen
 - Negativ:
 - Trennzeichen erforderlich (verkomplizieren das Parsen),
 - ESC-Mechanismus erforderlich (ESC = "escape")
 - Nur seriell verarbeitbar
 - Beispiel:
 - CSV (*comma separated variables*)

- Sonstige - *key/value*-Listen
 - Auch "*stanzas*" bzw. Strukturen wie Windows *.ini-Dateien
 - Positiv:
 - sehr flexibel, auch bei Aufnahme neuer Felder
 - selbst-dokumentierend (über sprechende *keys*)
 - Negativ:
 - Overhead
 - kein *a priori*-Wissen über den Ort eines erwarteten Wertes
 - Beispiel:
 - reclD=header,orderNo=12345,orderDate=20020607, ...
 - reclD=item,matNo=1234,quantity=10,...

- Sonstige - *Markup*, insb. XML

- Positiv:

- ideal für hierarchische *record*-Strukturen
- selbst-dokumentierend
 - in einfacher Form über sprechende *tags*
 - ggf. auch detailliert, über DTD bzw. Schema

- Negativ:

- sehr großer overhead
- hoher Speicherverbrauch - massendatentauglich?
- komplexes Interface (DOM)

- Bemerkungen zum Begriff *flat file*

- Vorsicht - keine einheitliche Verwendung!

- Zwei recht konträre, aber gebräuchliche Bedeutungen:

- a) serialisierte Speicherstruktur eines komplexen Typs (etwa: "flachgeklopfte Hierarchie")
- b) 1:1-Abbildung des EDIFACT *interchange* als *fixed record*-Format, 1 *record* pro Segment

Die SAP EDI-Schnittstelle- ein "prominentes" Beispiel

Das IDoc-Konzept
IDoc-Struktur
IDoc-Verwaltung
Besonderheiten, Ausblick

Das IDoc-Konzept

- IDoc = *intermediate document* = Zwischenbeleg
- Eine Struktur
 - zwischen Anwendungsbeleg einerseits
 - und klassischem, sehr dem EDI-Format nachempfundenen "flat file"
- Kernidee:
 - Abstraktionsschicht oberhalb konkreter EDI-Standards
 - z.B. UN/EDIFACT oder ANSI X12
 - Ein IDoc-Typ kann Quelle/Ziel mehrerer EDI-Nachrichtentypen sein, z.B. PEXR2002 <-> REMADV, PAYEXT, CREEXT, ...
- Einheitliche Kommunikationsform zwischen
 - verschiedenen SAP-Systemen bzw.
 - SAP und sog. "Sub-Systemen", insb. EDI-Servern

IDoc-Struktur



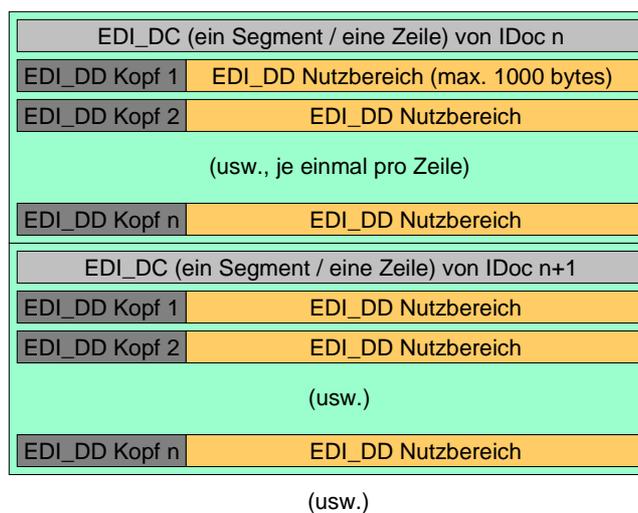
- EDI_DC
 - Allgemeiner Verwaltungskopfsatz
 - In etwa analog zu einer Kombination aus UNH und UNB
 - Geht jedem IDoc genau einmal voraus
- EDI_DD
 - Verwaltungsrahmen für Nutzsegmente
 - Nutzsegmente (je max. 1000 Bytes):
 - mit Versionsnummer
 - fester Satzaufbau
 - Hierarchie
 - Erweiterbarkeit
- EDI_DS
 - Statussatz, für Rückmeldungen ausgehender Belege

26.06.2002

EDI - SS 2002

13

IDoc-Struktur, schematisch



26.06.2002

EDI - SS 2002

14

IDoc-Struktur



- XML-IDocs
 - Gleicher Inhalt, neue Verpackung
 - Eher noch größeres Datenvolumen als bei den bereits recht großen IDocs mit *fixed record*-Struktur
- **IDoc-Dokumentation** - *online*-Beispiele
 - HTML-Dokumentation, wie man sie aus SAP R/3 exportieren kann
 - a) EDI_DC, EDI_DD, EDI_DS
 - b) Nutzdaten-Beispiel: ZINVOIC1

26.06.2002

EDI - SS 2002

15

IDoc-Verwaltung



- Übergabetabellen (DB)
- Eigene Statusverwaltung der IDocs
 - intern wie extern
 - mit eigenem *user interface*
- Dokumentation in IDoc-Verwaltung integriert
 - IDoc-Formatbeschreibungen sind jederzeit exportierbar
 - in strukturierter Form, vergleichbar XML-DTD/Schema
 - als HTML-Dokumentation
 - als „C“-Headerdateien (Makros, *struct's*, ...)
- Technische Schnittstellen
 - EDI (i.w. Dateischnittstelle plus *RFC* (SAP's RPC-Methode))
 - ALE („*application link enabling*“) - *high-level RFC*

26.06.2002

EDI - SS 2002

16

Die technische EDI-Schnittstelle



- EDI Ports - oder: SAP erreicht das Betriebssystem
 - Verzeichnisstruktur:
 - Basisverzeichnis z.B. /usr/sap/edi/<SAPSYS>/<PORTNAME>
 - ./in für eintreffende IDoc-Dateien
 - ./out für ausgehende IDocs
 - ./status für eingehende Dateien mit Statusätzen
 - Ausgehende Daten: **rfcexec**
Nahtstelle zwischen SAP RFC und Betriebssystem
 - Polling vs. Triggern für jedes IDoc vs. Trigger pro Exportdatei (Vorsicht - ggf. splitten)
 - Eingehende Daten und Statusätze: **startRFC**
 - Polling (durch SAP) vs. Triggern
 - Problem Fehlerbehandlung: Welches IDoc ist betroffen, welche sind schon verarbeitet?

26.06.2002

EDI - SS 2002

17

Die technische EDI-Schnittstelle



- ALE Ports
 - Austausch von Speicherstrukturen direkt über ein Netzwerkprotokoll, ohne Dateisystemkontakt
 - Erfordert ALE-Fähigkeit auf beiden Seiten
 - Gut insbesondere für zeitkritische Anwendungen
 - ALE wird insbesondere zum Belegtausch zwischen SAP-Systemen verwendet, seltener zwischen SAP und Subsystemen wie EDI-Servern.
 - Strukturanpassungen notwendig, falls Kopplung zwischen SAP-Systemen mit unterschiedlichen *releases*,
 - etwa R/3 4.6 zu koppeln mit R/3 3.1

26.06.2002

EDI - SS 2002

18

- Besonderheiten
 - Applikationsserver-Gruppen bilden:
 - startfc wendet sich an einen *messaging server*,
 - dieser bestimmt Servermitglied aus der Gruppe für den Import
 - Vorteile
 - Lastverteilung
 - Risikostreuung
 - Permanente Verfügbarkeit der EDI-Schnittstelle auch wenn ein Applikationsserver gewartet wird
- Ausblick / Verwandte Themen
 - BAPIs (Business APIs)
 - genormte *high-level* Schnittstellen z.B. zur Erzeugung eines kompletten Geschäftsdokuments durch ein externes System
 - Business Integration Server (MOM, XML, *Web services*)

Mapping und *Mapping*-Techniken

Das Grundproblem

Das *Metamap*-Konzept

Andere Ansätze

Spezielle *Mapping*-Aufgaben

Werkzeuge und methodische Ansätze

Zum Begriff „Mapping“



- engl. „map“ = „Abbildung“ (math.)
- Herstellung / Beschreibung des Zusammenhangs zwischen eingehenden und ausgehenden Daten eines EDI-Konverters, i.d.R. auf Belegebene. Beispiel:
 - Eine SAP IDoc-Struktur „ZINVOIC1“ umwandeln in eine EANCOM-Nachricht „INVOIC“
- Der Begriff suggeriert eine einfache 1:1-Zuordnung zwischen Feldinhalten auf Quell- und Zielstrukturseite, beschreibbar durch eine Zuordnungsvorschrift in Tabellenform
- Tatsächlich können derartige I/O-Beziehungen recht komplex werden. Sie sind dann besser algorithmisch als tabellarisch beschreibbar.

26.06.2002

EDI - SS 2002

21

Mapping: Das Grundproblem



- Wunsch und Wirklichkeit
 - Die Anwendung möchte oder kann nur Daten in immer gleicher Weise bereitstellen bzw. akzeptieren.
 - Geschäftspartner benutzen zwar EDI-Standards, aber diese Standards lassen noch eine Fülle von partnerspezifischen Variationen zu.
- Beispiel für Variationen trotz Standardisierung
 - Die Tabelle der Partnerspezifika (ca. 100 x 8 !) aus den Anwendungsrichtlinien des AK Handel zu Rechnungsdaten
- Mapping-Techniken
 - Je nach Vielfalt der Anforderungen ist die passende *Mapping*-Technik (und damit u.U. das dazu passendste Konverterprodukt) zu wählen.

26.06.2002

EDI - SS 2002

22

Das Metamap-Konzept



- Algorithmischer Grenzfall:
 - *Mapping* = Software-Entwicklung
- Gemeinsamer, komplexer Programmcode
 - für viele Partnerspezifika
 - mit vielen Fallunterscheidungen / Logikabschnitten
 - redundanzarm, leicht zu warten bei gemeinsamen Veränderungen
- Softwareschalter und optionale Parameter
 - Steuerung im Einzelfall
 - Zusammenfassung zu **Schalter-Sets**
 - Organisation z.B. als Lookup-Tabelle:
 - Eine Zeile pro Schalternamen, eine Spalte pro Set
- Auswahl des zuständigen Schalter-Sets mittels Absender/Empfängererkennung
 - z.B. aus UNB-Segment oder EDI_DC

26.06.2002

EDI - SS 2002

23

Das Metamap-Konzept



- Redundanzvermeidung der 2. Ordnung:
 - Funktionale Schaltergruppen.
 - Beispiel Netto-vs. Bruttoabrechnungsverfahren
 - Eine Gruppe repräsentiert „Netto“, eine andere „Brutto“
 - Schalter-Sets
 - Komplette Schaltersammlung
 - Gruppierung: Jeweils ein Set für alle Partnerkennungen mit gleichen Anforderungen
 - Vererbungskonzept:
 - Sets erben Werte von Schaltergruppen und anderen Sets
 - Geerbte Werte können bei Bedarf lokal überschrieben werden
 - Beispiele:
 - Set X wie Set A, aber mit Nettoabrechnungs-Gruppe
 - Set B wie Set A, zusätzlich Schalter S := „M“

26.06.2002

EDI - SS 2002

24

Das Metamap-Konzept



- Redundanzvermeidung der 2. Ordnung (Forts.):
 - Ergebnis: Redundanzen der Schaltermatrix minimiert
 - Voraussetzungen dazu
 - Vererbungstechnik
 - Default-Sets
 - Erst Fachwissen über die Bedeutung der Schalter gestattet sinnvolle Definition von Schaltergruppen
 - Analyse der Schalterbeziehungen und insbesondere die Definition des Vererbungsgraphen erfordert Erfahrung, Branchenkenntnis und Vorausschau
- NAD-Mapping - ein separater Einsatz für Metamaps
 - Partnerfunktionen (z.B. „Besteller“, „Rechnungsempfänger“) aus Sicht des EDI-Partners unterscheiden sich oft von der eigenen Repräsentation in den Stammdaten, also:
 - Partnerspezifische Anpassungen notwendig

26.06.2002

EDI - SS 2002

25

Mapping: Andere Ansätze



- Zweistufiges *Mapping*
 - Quelle - interne Struktur, interne Struktur - Zielformat
- Variante: Semantische Geschäftsprozess-Integration
 - interne Struktur erzwingt semantisch korrekte Bereitstellung
 - zweiter Teil des *Mapping* ist dann frei von Missverständnissen
- Grenzfall: *Mapping* - wörtlich genommen
 - Viele, aber einfache tabellarische *Maps*
 - Vorteile: Keine Programmierlogik, GUIs einsetzbar, *Mapping* nach Spezifikation durch angelernte Kräfte möglich
 - Nachteile: Schlecht wartbar, aufwändig bei syst. Änderungen

26.06.2002

EDI - SS 2002

26

Spezielle *Mapping*-Aufgaben



- Plausi-Checks
 - wenn möglich auch automatische Korrekturverfahren
 - Bsp: Materialstamm; Rechnungslisten
- Nachrüsten (konstanter) Stammdaten
 - Bsp: Metro Lieferantennr.
- Ausnahmetabellen
 - Bsp. spezielle Materialien
- Logs für Warnungen
 - Bsp. EAN-Lücken, ILN-Lücken
- Erzeugen von Meta-Belegen
 - Bsp. Sammel-Rechnungsliste für REWE

26.06.2002

EDI - SS 2002

27

Spezielle *Mapping*-Aufgaben



- Angepasste Fehlerbehandlung
 - *en bloc*
 - alle Belege ablehnen bei einem Fehler
 - notwendig bei Abhängigkeiten zwischen Belegen,
 - z.B. bei einem Bündel Rechnungen mit Rechnungsliste
 - *split*
 - fehlerhafte Belege abtrennen,
 - Rest konvertieren & versenden
 - Verschafft Zeit zur Fehlerkorrektur
 - Hält nur die Problemfälle zurück

26.06.2002

EDI - SS 2002

28

Grundsatzfrage beim *Mapping*



- Welche Logik / Funktion gehört in den Konverter, welche in die Anwendung?
- Empfehlung:
 - Anwendung:
 - Gemeinsame Funktionalitäten
 - generelle Logik
 - Konverter:
 - Partnerspezifika

26.06.2002

EDI - SS 2002

29

Werkzeuge / methodische Ansätze



- Grenzfall 1: Mapping wörtlich genommen
 - Felder möglichst 1:1 abbilden, Programmierlogik vermeiden
 - Graphische Mapping-Werkzeuge evtl. gut einsetzbar
- Grenzfall 2: Große Vielfalt, komplexe Regeln, algorithmisch beschreibbar
 - Klassische Software-Entwicklung mit geeigneter *Mapping-Sprache*
- Technische Umsetzung in Konverterprodukten
 - Interpreter-Ansatz (typisch) vs. kompilierter Code
 - Herstellerabhängig
 - Allgemein: *Tradeoff* zu suchen zwischen
 - einfach zu erlernenden und bedienenden Werkzeugen, sowie
 - flexiblen, günstig zu wartenden, performanten und skalierbaren Umgebungen

26.06.2002

EDI - SS 2002

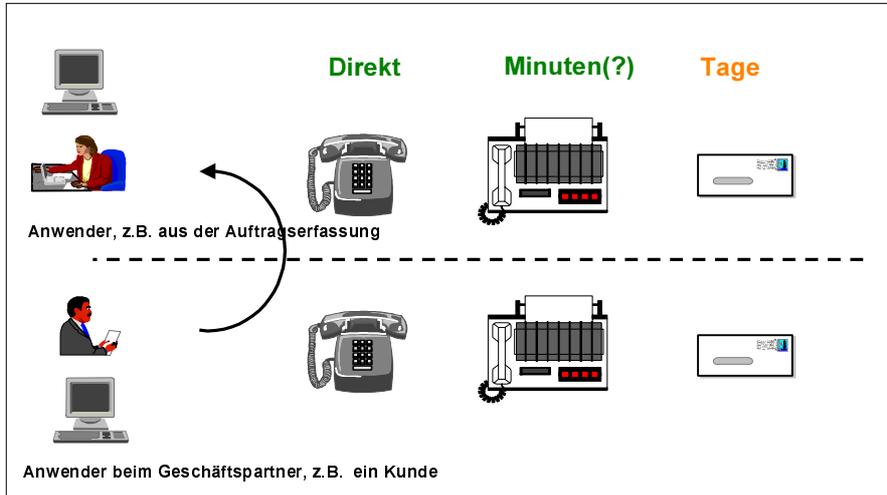
30

Das Umfeld

Tracking, Reporting, Archivierung
Hybride Ansätze
EAI und BPI

- Hinweis:
 - PDF-Dokument mit Abbildungen zu EDI Tracking & Tracing sowie einer integrierten Archivanbindung und Reporting-Funktionen

Tracking, Reporting, Archivierung



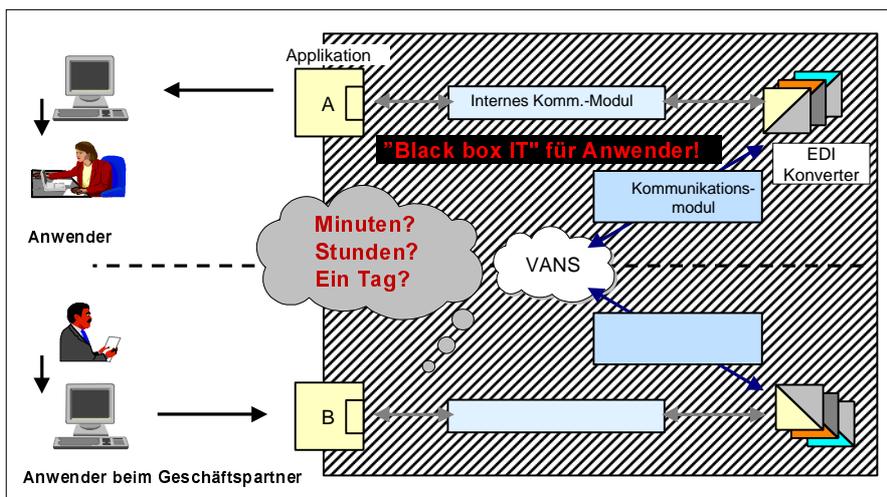
Übermittlungszeiten für Geschäftsdaten zwischen Fachabteilungen, ohne EDI

26.06.2002

EDI - SS 2002

33

Tracking, Reporting, Archivierung



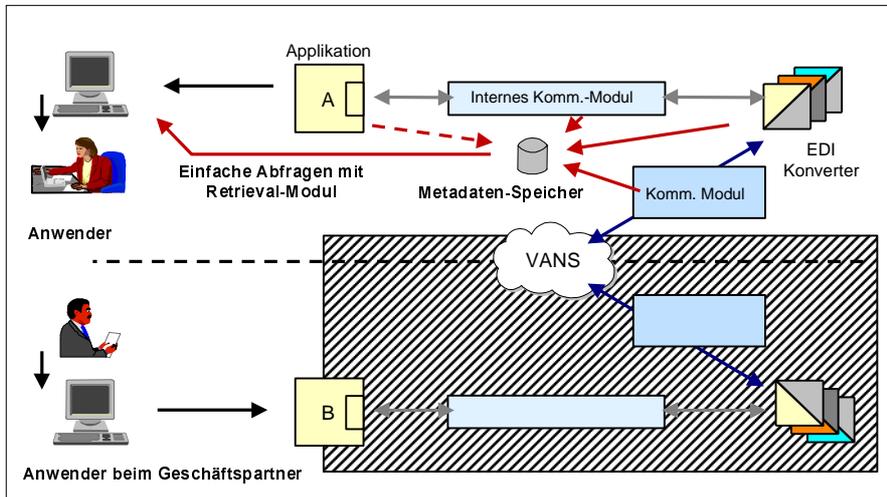
Übermittlungszeiten für Geschäftsdaten zwischen Fachabteilungen, mit EDI: Unbekannt!

26.06.2002

EDI - SS 2002

34

Tracking, Reporting, Archivierung



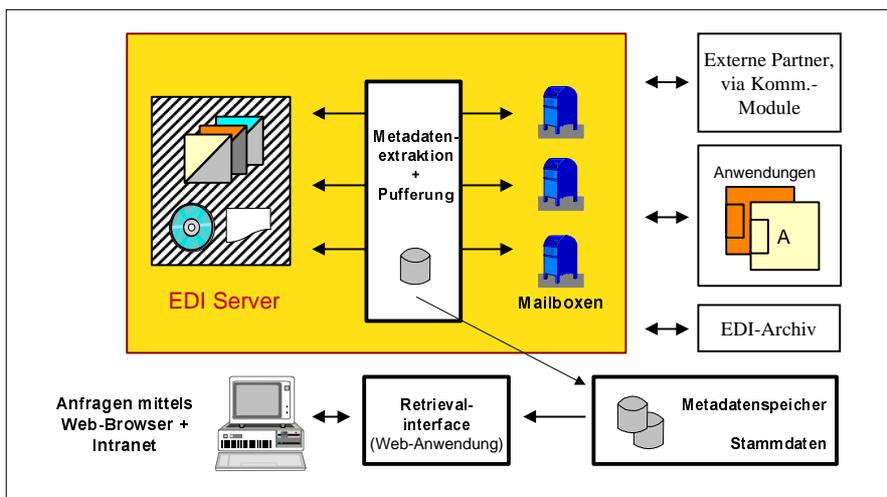
Übermittlungszeiten für Geschäftsdaten mit EDI und T&T: Jederzeit einsehbar!

26.06.2002

EDI - SS 2002

35

Tracking, Reporting, Archivierung



Komponenten von EDI Tracking & Tracing

26.06.2002

EDI - SS 2002

36

Hybride Ansätze



- Zum Lieferanten: Web-EDI
- Zum Kunden: Web-Shops
- EDI-to-paper: ePost
- FAX-to-EDI:
 - OCR, KI-Methoden, plus manuelle Nachkontrolle
 - (hier nur Stichworte)

26.06.2002

EDI - SS 2002

37

EAI und BPI



- **EAI: *Enterprise Application Integration***
 - Einheitliche Infrastruktur / APIs zur Erleichterung von Kopplungen zwischen diversen Anwendungen („A2A“)
 - Konverter können Bestandteile einer Gesamtlösung sein
 - Zentrale Administrierbarkeit spielt eine große Rolle
 - Sinnvoll für große Unternehmen mit einer Vielzahl von Anwendungen an unterschiedlichen Standorten
 - Vgl. 4 Abbildungen aus edi-change 02/2001
- **BPI: *Business Process Integration***
 - Baut auf EAI auf
 - Geschäftsprozesse statt Einzelschritte im Vordergrund
 - Neu: Auch Integration manueller Schritte; workflows

26.06.2002

EDI - SS 2002

38

EAI: Die Klassiker



- *Hub-and-spoke* Ansatz
 - z.B. GE Enterprise
 - siehe auch JMS
- *Message queues*
 - insbesondere IBM MQ Series
 - *point-to-point queues*, damit flexible Topologien realisierbar
- TIBCO, oder: *The Information Bus company*
 - Lokale „Konnektoren“, gemeinsamer „info bus“
 - Zentral administrierbar, sehr gut für *publish/subscribe*-Verfahren

26.06.2002

EDI - SS 2002

39

EAI: Neuere Entwicklungen



- Herstellerneutrales API:
 - *Java Message Service (JMS)*, aus J2EE)
- Infrastruktur hinter dem API im Prinzip austauschbar
 - Beispiel: SonicMQ, lieferte Lösung für CommerceOne-Anbindung
- Sowohl für P2P als auch für P/S geeignet

26.06.2002

EDI - SS 2002

40

EAI: Zielkonflikte



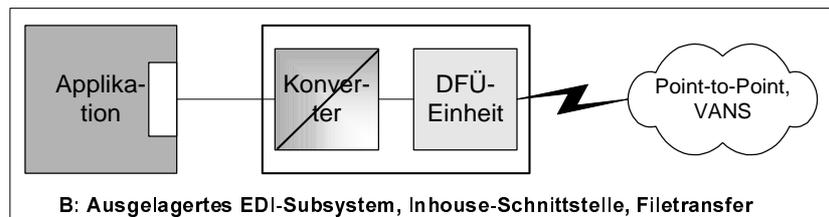
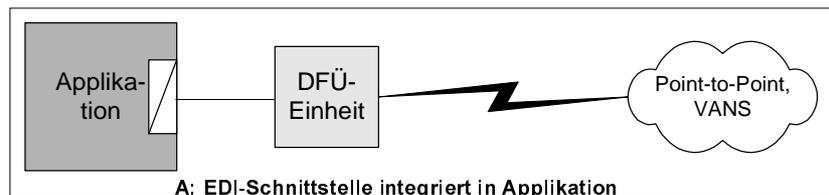
- Einerseits
 - Hoher Durchsatz
 - Eignung für viele kleine Nachrichten (kB-Bereich, auch SMS)
- Andererseits
 - *Heavy duty file transfer*: GB-Bereich
- *Multicasting*-artige Verteilung
 - z.B. Börsenticker-Angaben, per *publish/subscribe*-Verfahren
 - für flüchtige, kleine, sich ständig aktualisierende Nachrichten
- aber auch:
 - Sichere P2P-Übertragung von Dateien mit Zustell-Bestätigung und *Tracing*.
- Fazit: Auswahl einer EAI-Lösung erfordert Bedarfsanalyse.

26.06.2002

EDI - SS 2002

41

EDI/EAI: Historische Entwicklung

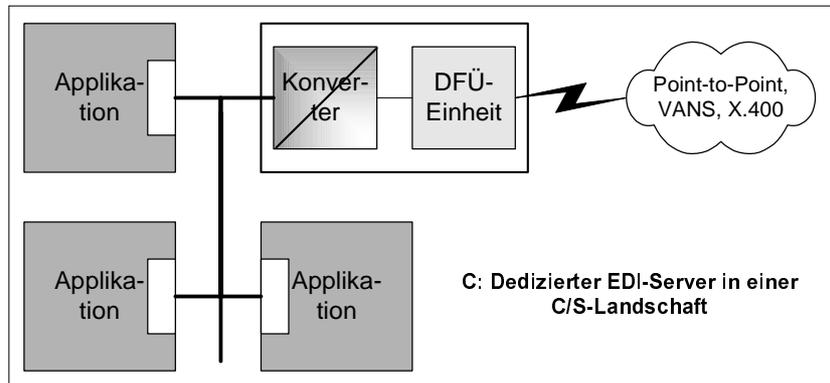


26.06.2002

EDI - SS 2002

42

EDI/EAI: Historische Entwicklung

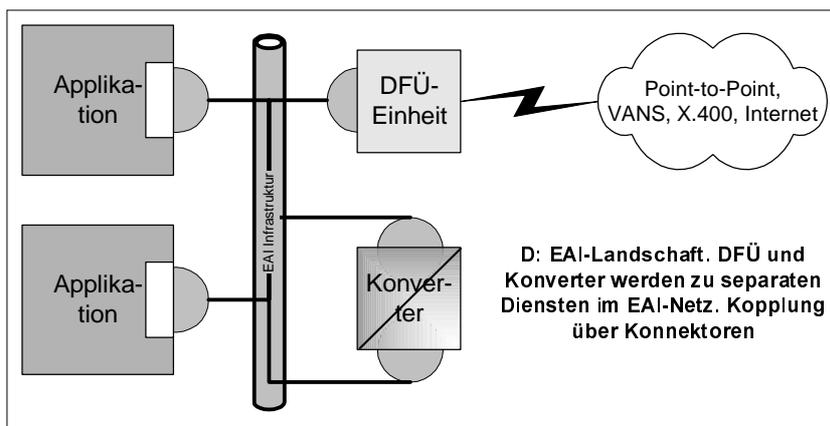


26.06.2002

EDI - SS 2002

43

EDI/EAI: Historische Entwicklung



26.06.2002

EDI - SS 2002

44

Optional: XML-basierte Ansätze

XML-Grundlagen
Probleme und Chancen
Aktuelle Entwicklungen