



# Asynchrone EAI mit XML

## Ein Praxisbericht

Christian Peterhans

[christian.peterhans@amrio.ch](mailto:christian.peterhans@amrio.ch)

28. Oktober 2003



# Firmenportrait AMRIO AG

---

- Anwendungsarchitekturen
- Anwendungsdesign
- Anwendungsentwicklung
- Beratung für alle Bereiche der Anwendungsentwicklung
- EAI

# Ausgangslage

---

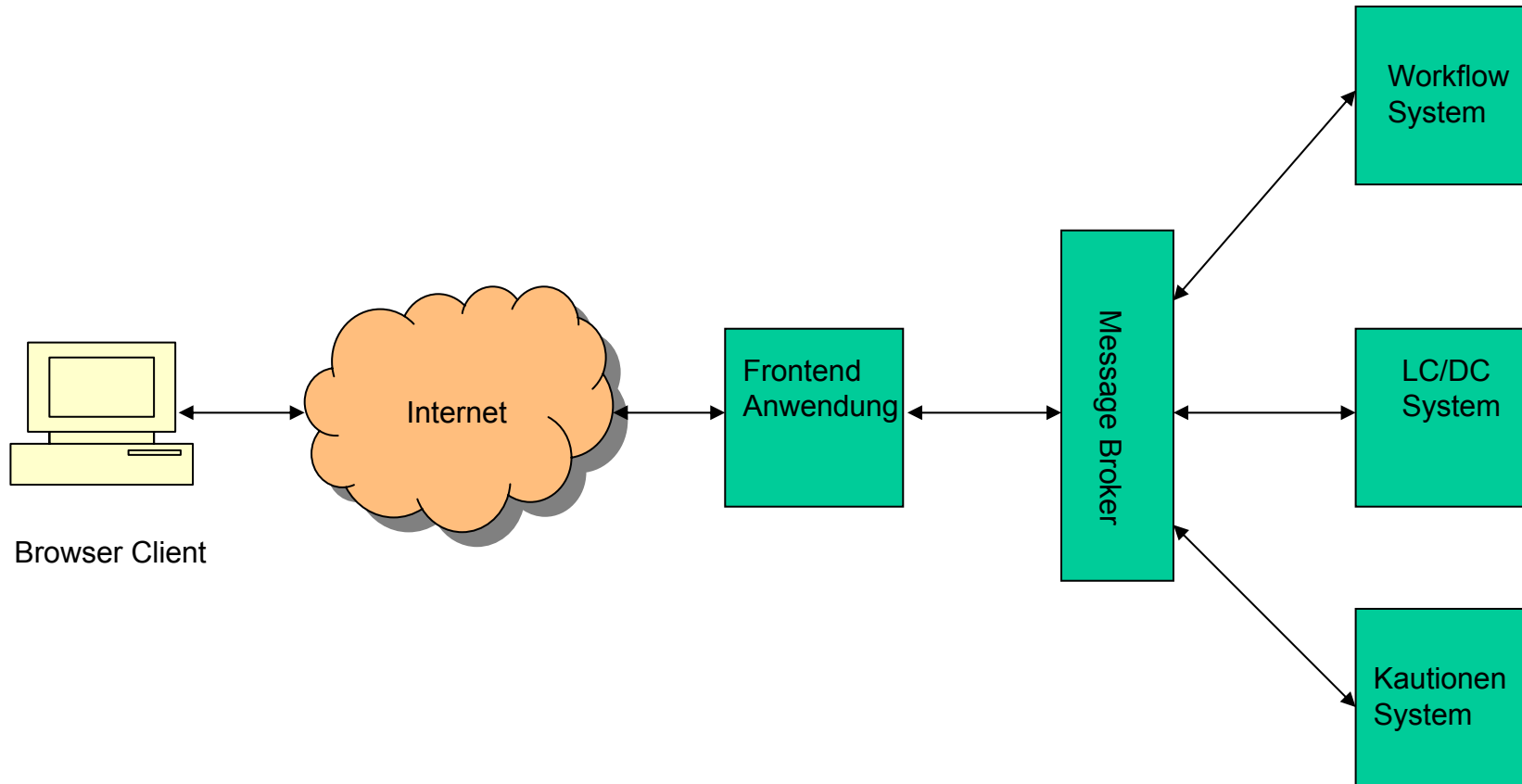
- Eingekauftes Internet Frontend für Handelsfinanzierungen
- Drei Backend Systeme
  - Akkreditiv und Dok Inkasso System
  - Kautionen System
  - Workflow System

# Projekt Vorgaben

---

- Architekturvorgaben
- Bolero Dokumente wenn immer möglich
- Keine direkte Datenmanipulation in den Backend Systemen
- Minimale Anpassungen in den Backend Systemen

# Lösungsansatz

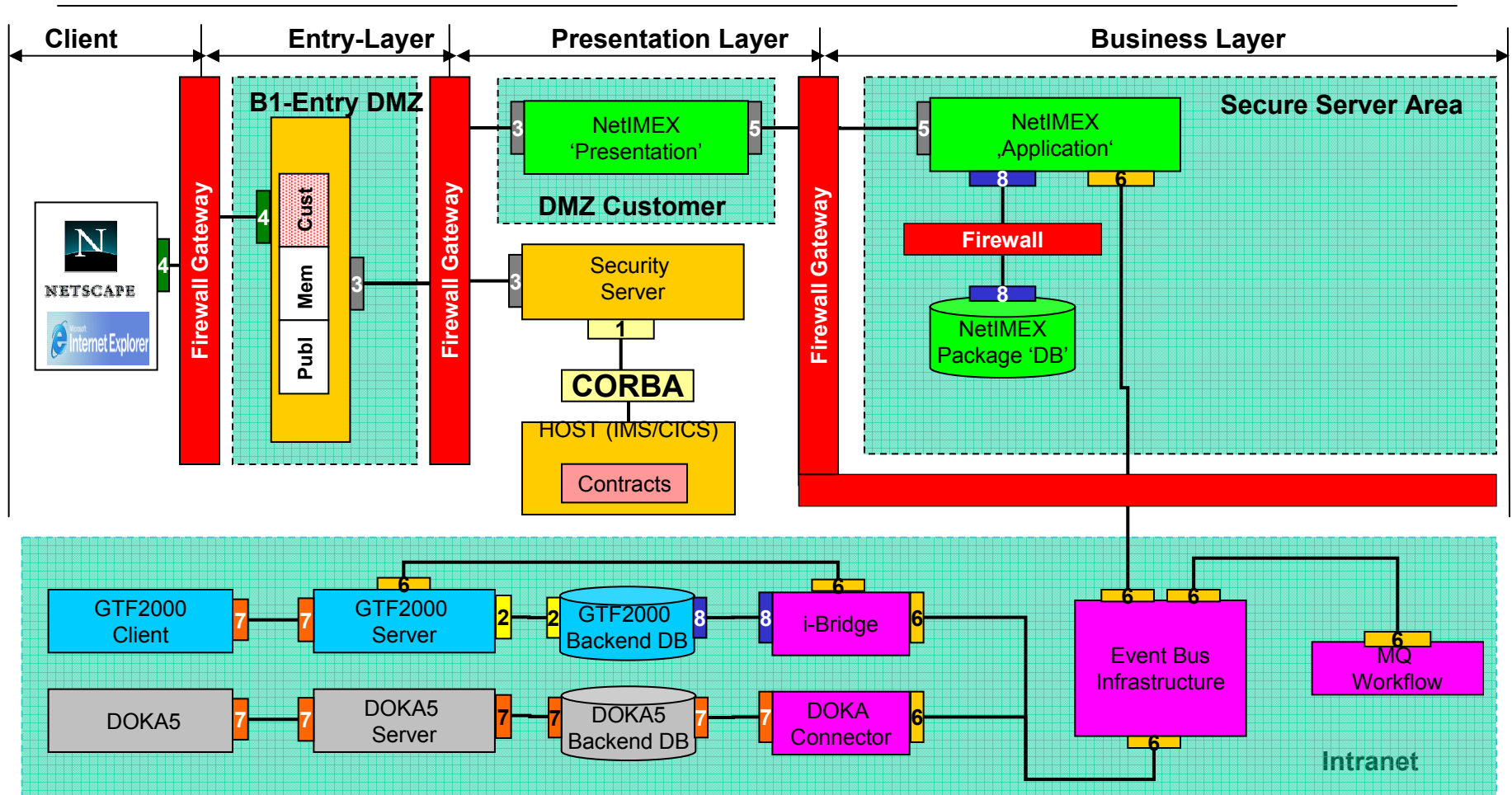


# Komponenten der Lösung

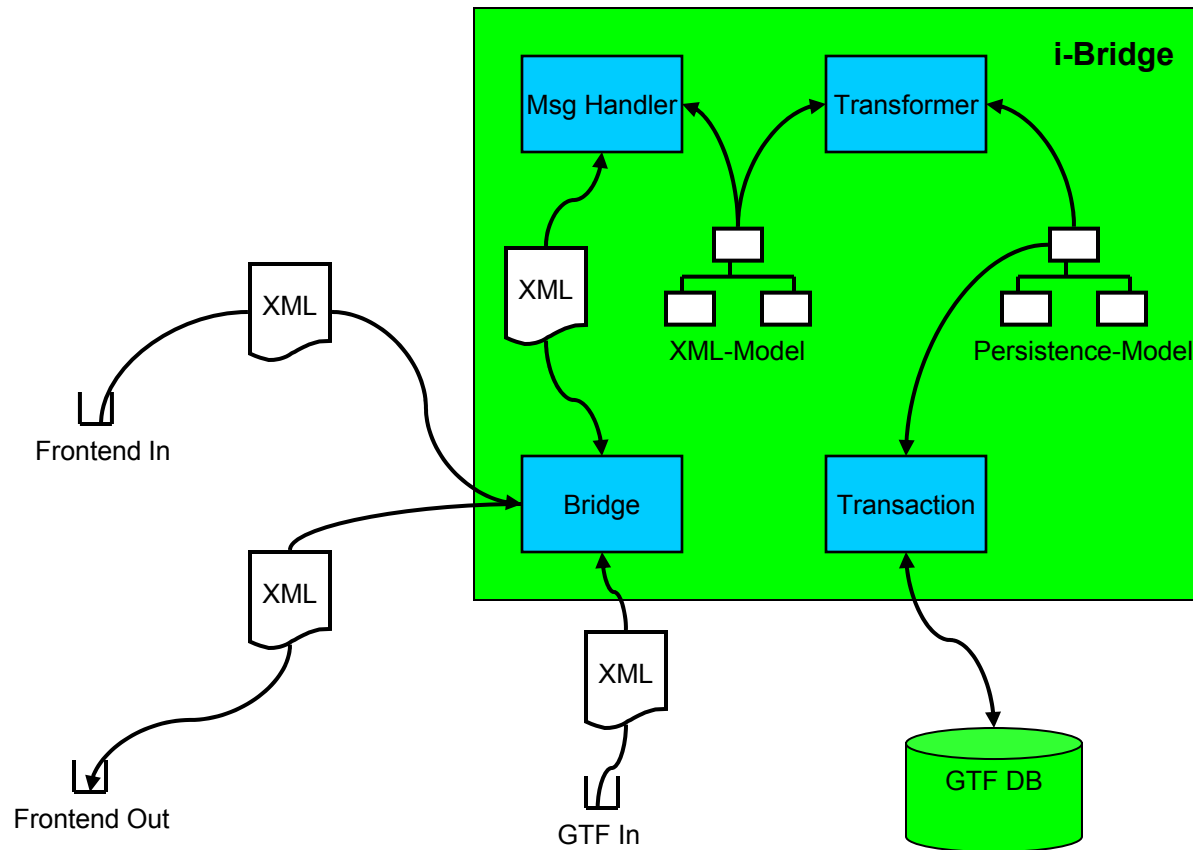
---

- NetIMEX
- EBI (Msg Routing, Transformation)
- MQ Workflow
- DOKA (Kautionen System)
- GTF2000 (Akkreditiv/Dok.Inkasso)
- XML, bolero XML
- Acrobat Reader

# Architektur



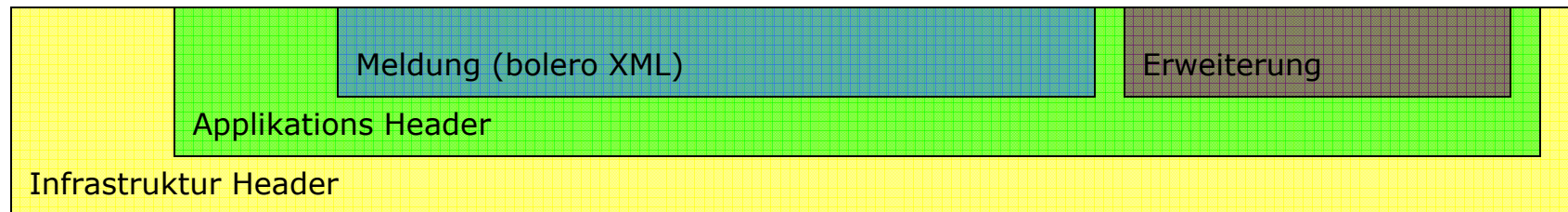
# Aufbau i-Bridge



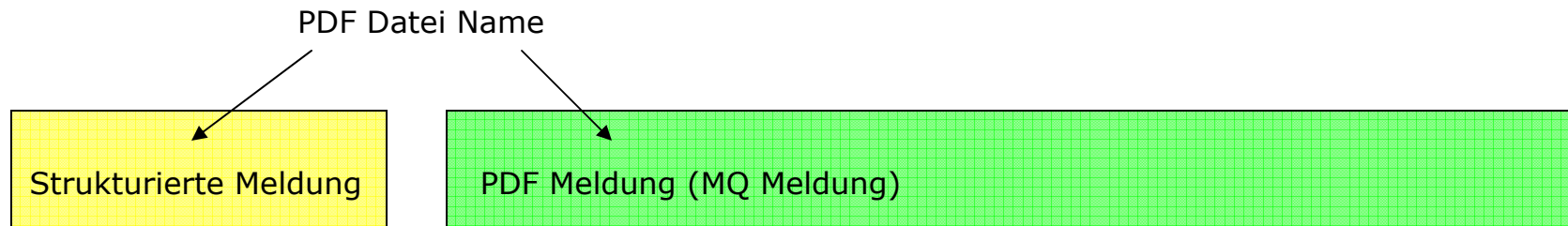


# Aufbau der Meldungen

- Strukturierte (XML) Meldungen



- PDF Meldungen



# Erfahrungen Generell 1/4

---

- Unterschiede der beteiligten Systeme
  - Philosophie
  - Syntax
  - Semantik
  - Datenstrukturen
  - Datentypen
  - Strukturierte und unstrukturierte Daten

# Erfahrungen Generell 2/4

---

- Definition der Schnittstellen
  - Am Anfang des Projektes
    - Genügend Zeit einplanen
    - Aufwändig trotz Verwendung der bolero Strukturen
  - Spätere Änderungen müssen koordiniert werden
  - Aufwand für Änderungen

# Erfahrungen Generell 3/4

---

- Integritätsverantwortung der Komponenten
  - Service vs. Modul
  - Validierung der Daten
  - Fehler Behandlung

# Erfahrungen Generell 4/4

---

- Tool für XML Klassen Generierung und Marshalling/Unmarshalling
  - Änderungen an den Datenstrukturen
  - Navigation mit Konstrukten der Programmiersprache
  - XML Validierung
  - Keine `System.out.println()` Anweisungen im Code

# Erfahrungen mit XML 1/2

---

- Generierung von Klassen
- DTD oder Schema
  - Semantische Unterstützung
  - Reduktion von Validierungscode

# Erfahrungen mit XML 2/2

---

- Grundregeln für die Syntax
  - Namenskonventionen
    - Eindeutige Namen
  - Regeln für Elemente
    - Gleiche Struktur → gleicher Name
  - Element vs. Attribut

**→** Besonders wichtig wenn Klassen generiert werden!

# XML und PDF

---

- PDF Datei kann grundsätzlich alle Zeichen enthalten, auch XML Markup Charakter -> Problem mit XML Parser
- CDATA Sektion wird vom Parser nicht analysiert
- CDATA Sektion darf keinen `]]>` String enthalten