

# Dynamische XML-Verarbeitung

Bernhard Eichhorn  
MIC Management Consulting GmbH  
München  
&  
Rainer Krohn  
EDV-Beratung Rainer Krohn GmbH  
Aßling

**MIC Management Consulting GmbH**  
Kronacher Str. 4 81549 München  
Tel. +49 / 89 / 680 711 61  
[www.mic-muenchen.de](http://www.mic-muenchen.de)  
[office@mic-muenchen.de](mailto:office@mic-muenchen.de)

# Allgemeines

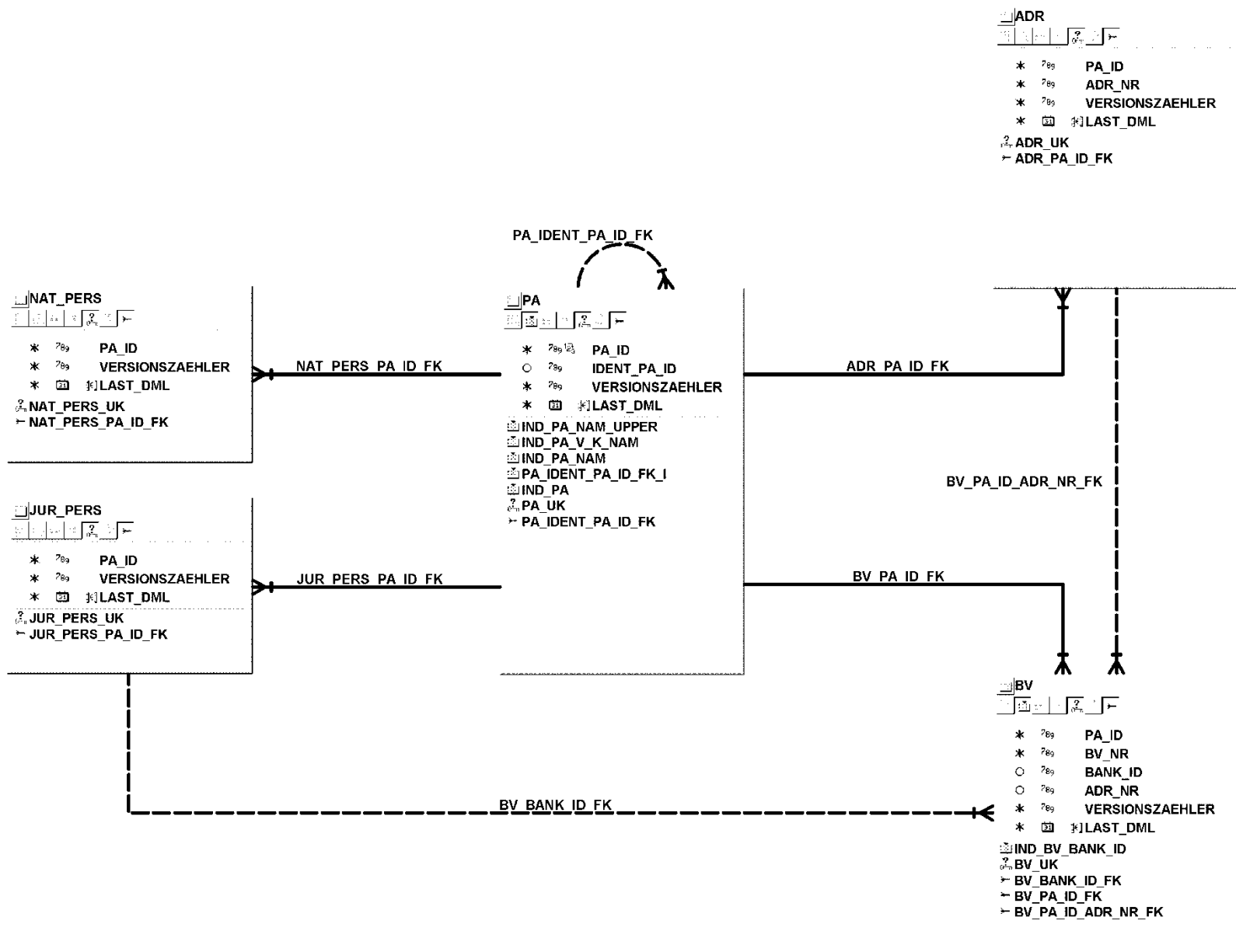
- XML kommt als Datenaustauschformat in fast allen Bereichen der Datenverarbeitung zum Einsatz. Insbesondere in der Versicherungswelt ist XML die Grundlage für den Datenaustausch z.B. bei:
  - Rentenbezugsmitteilung
  - Meldungen an die Zentrale Zulagenstelle für Altersvermögen (ZfA) im Falle der Riesterrenten
- Große Änderungshäufigkeit bei den zugrunde liegenden XML-Strukturen
  - Datenaustausch mit der ZfA nach 5 Jahren Riesterrente aktuell mit der XML-Version 19.b
- Auch in Zukunft müssen immer mehr Zusatzinformationen in die XML-Strukturen integriert werden
  - Steuernummer
  - IBAN, BIC
- Immer wiederkehrende Anpassungsaufwände in den entsprechenden Software-Modulen

# Projektbeschreibung

- Auftraggeber:  
Swiss Life AG Deutschland  
Projektlaufzeit: 11.2009 bis 04.2010



- Aufgabenstellung:
  - Anbindung eines neuen Provisionssystems an das zentrale Partnersystem zur Pflege von Partnerdaten, Adressen und Bankverbindungen.
  - Die Kommunikation zwischen der Anwendung und der Datenbank erfolgt über den Enterprise Service Bus (ESB).
  - Für die einzelnen Bereiche der Pflege von Partner-, Adress- und Bankverbindungsdaten sind eigene XML-Strukturen vorgegeben, die sowohl für Insert-, Update- oder Delete-Operationen verwendet werden sollen.
  - Das Partnersystem sollte für jede XML-Struktur einen eigenen Dienst zur Verfügung stellen, der die XML-Daten validiert und in die entsprechenden Tabellen einarbeitet.

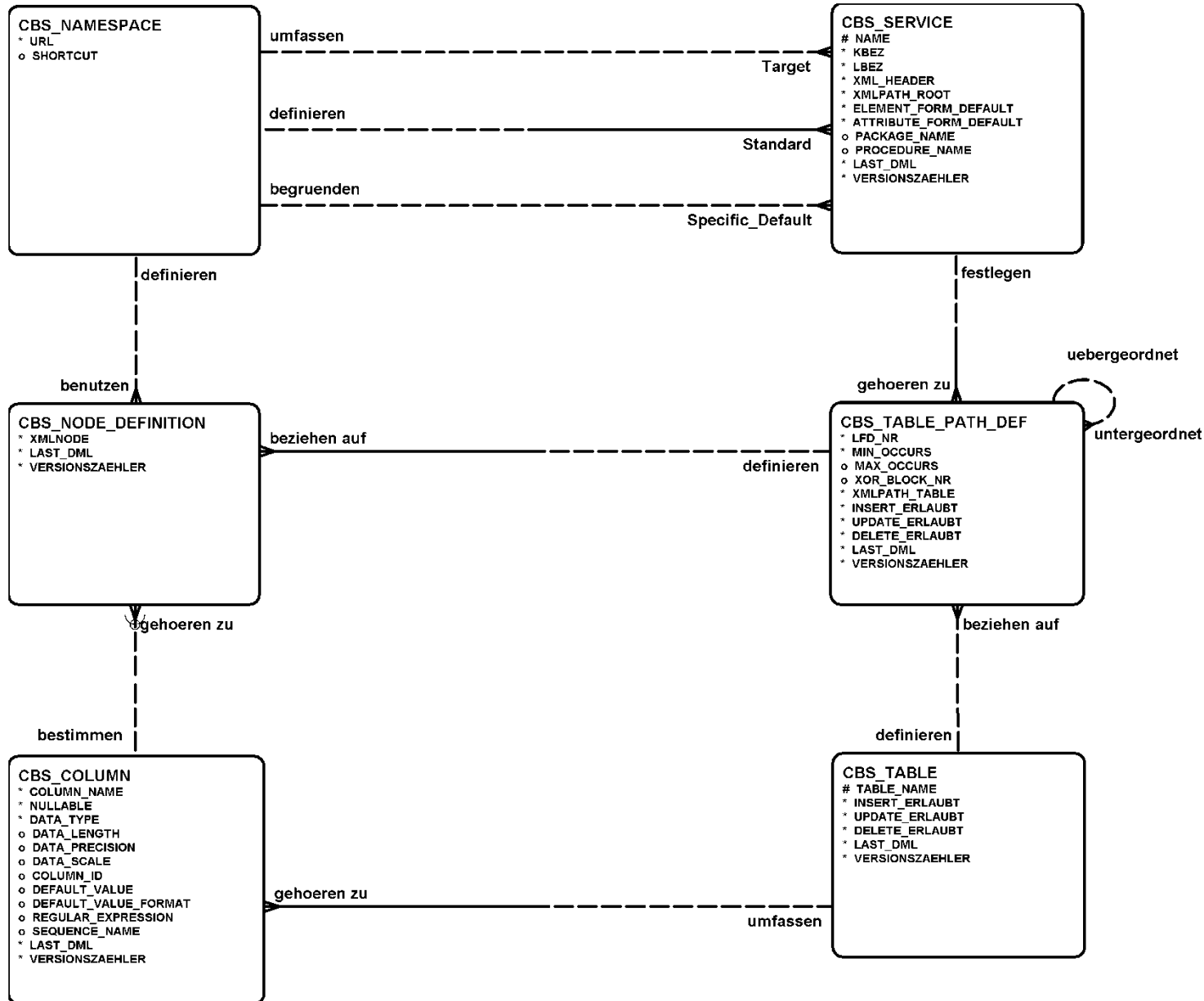


# Projektanforderungen

- Direkte Verarbeitung von XML-Datensätzen in relationale Datenbankstrukturen.
- Die Verarbeitung soll sich invariant bei Strukturänderungen an den Datenbanktabellen und/oder XML-Strukturen verhalten, d.h. die Anpassungsaufwände für die Integration neuer Strukturkomponenten sollen mit minimalem, im Idealfall gar keinem Programmieraufwand möglich sein.
- Erkennung von Strukturänderungen
- Dynamische Einbindung beliebiger Validierungsroutinen
  - semantische Prüfung, z.B. gültige Wertebereiche, Pflichtfelder, Abhängigkeiten von XML-Komponenten untereinander.
  - Datenanreicherung, z.B. Versorgung der Primary-Key-Spalten beim INSERT
  - Rückgabe sämtlicher Fehlermeldungen über kompletten XML

# Realisierung

- Entwicklung eines Metadatenmodells
  - Mapping vom XML-Strukturen auf relationale Datenbanktabellen
  - XML-Schema-Generierung
  - Erkennung von Strukturänderungen
  - Zuordnung von dienst- und anwendungsspezifischen Validierungsroutinen
    - Einbindung Validierungsroutinen zur Laufzeit
    - Generierung der DML-Statements zur Laufzeit
  - Hinterlegung von dienst- und anwendungsspezifischen Regeln
- PL/SQL
  - intensive Nutzung von Dynamic SQL
  - XMLType-Komponenten von Oracle
- keine Funktionalitäten der Oracle XMLDB



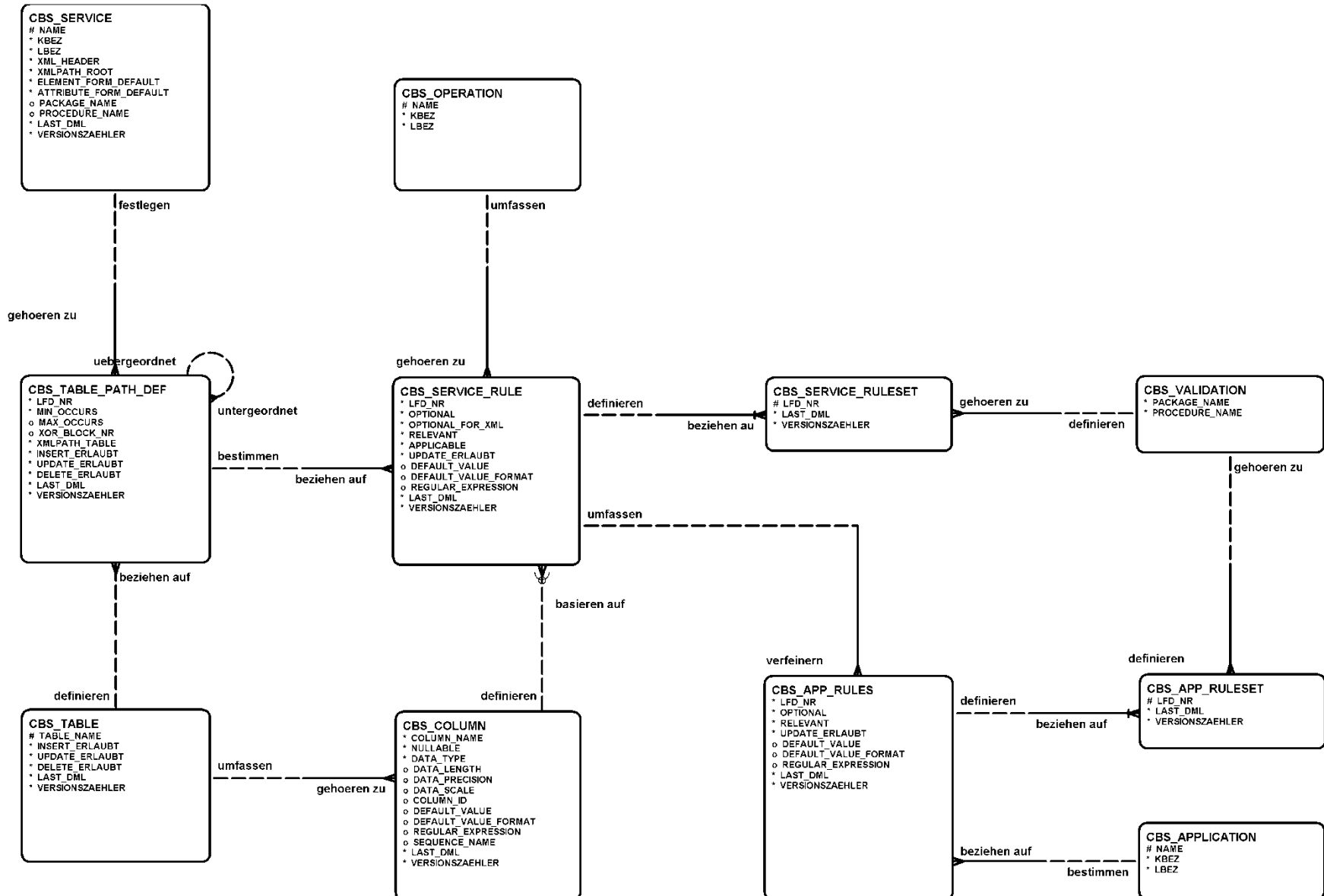
# Mapping XML-Strukturen mit DB-Tabellen

- Definition von "Dienstern"
  - Schnittstelle zwischen Aufrufer und Service-System
  - Hinterlegung der PL/SQL-Prozedur, mit der ein Dienst von außen aufgerufen wird
- Zuordnung von Namespaces
- Verknüpfung von Diensten zu Tabellen des Ziel-DB-Schemas
  - Definition einer Tabellenhierarchie
  - Oder-Verknüpfungen auf gleicher Hierarchie-Ebene, z.B. Partner entweder natürliche oder juristische Person, 1:1-Beziehung
  - Definition von Oder-Verknüpfungen zwischen Tabellen einer Hierarchie-Ebene (=> choice-Gruppe im XML-Schema)
  - mehrfache Verwendbarkeit einer Tabelle auf unterschiedlichen Hierarchiestufen
- Mapping zwischen XML-Komponenten und Datenbanktabellen/-spalten
- Möglichkeit ein XML-Schema zu generieren



# Demo 1

- Einrichten des Dienstes „Partnerdemo“ zur Erzeugung und Pflege von Partner- und Adressdaten
- Zuordnung der relevanten Tabellen unter Berücksichtigung der hierarchischen Struktur
  - Tabelle PA übergeordnet zu Tabelle ADR
- Generierung des dienstspezifischen XML-Schemas
- 1. Ladelauf
- Ergebnisse:
  - Fehler bei Ausführung der generierten Insert-Statements insbesondere wegen Primary-Key-Spalten, die zur Laufzeit vom System vergeben werden und von außen nicht versorgt werden können.
  - Notwendigkeit Spalten für bestimmte Schreiboperationen als irrelevant zu kennzeichnen und/oder Regeln für deren Versorgung zu hinterlegen



# Einbindung von Regeln

- Ausblendung irrelevanter XML-Komponenten pro Operation
- Versorgung von NOT NULL-Spalten, die von außen nicht geliefert werden können, z.B. Primary-Key-Spalten beim Insert
- Abbildung von (virtuellen) Foreign-Key-Beziehungen
- Hinterlegung von Unique Keys
- Abbildung der Selektionskriterien für Update- und Delete-Statements
- semantische Prüfung, z.B. gültiger Wertebereich für XML-Komponenten
- Definition von Defaultwerten
- Definition optionaler Spalten der Datenbanktabelle als Pflichtfelder
- Abhängigkeiten von XML-Komponenten untereinander, z.B. muss bei natürlichen Personen ein Geburtsdatum angegeben werden
- Datenanreicherung, z.B. Sequences beim INSERT
- Regeln und Validierungsvorschriften können sowohl auf dienst- als auch anwendungsspezifischer Ebene definiert werden

## Demo 2

- Einfügen der Regeln
  - NOT-NULL-Spalten => optional
  - optionale Spalten => Pflichtfelder
  - Abhängigkeiten von XML-Komponenten untereinander
  - Definition UPDATE-Regeln
- Generierung des dienstspezifischen XML-Schemas
- 2. Ladelauf

# Unabhängigkeit XML zu DB-Struktur

- Beispiel: Geburtsdatum einer natürlichen Person
  - XML: Ebene Tabelle NAT\_PERS
  - Datenbank: Ebene Tabelle PA
- Einführung Virtual-Column-Konzept



## Demo 3

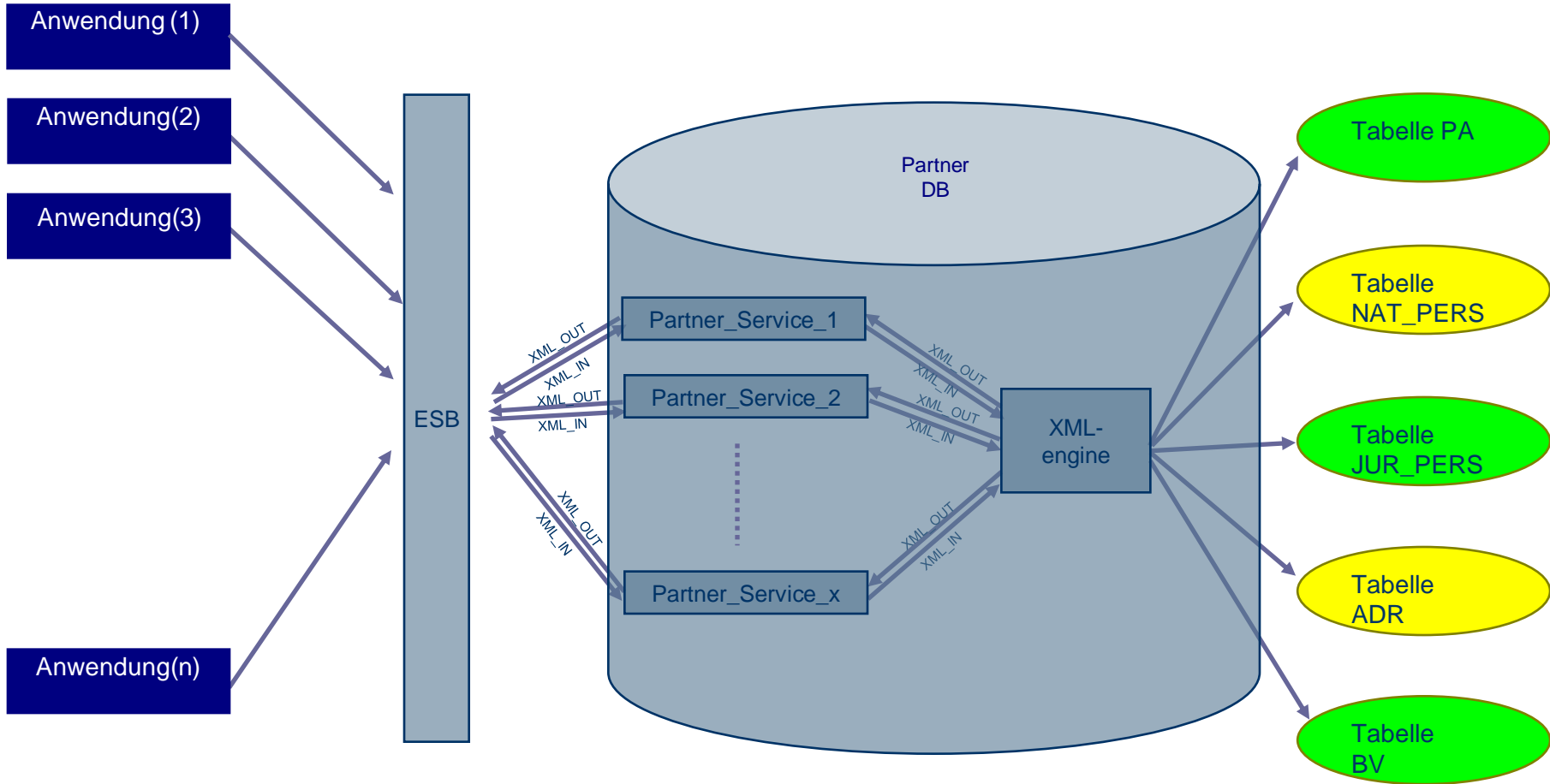
- Erweiterung Demo-Dienst
  - Tabelle NAT\_PERS
  - Definition virtueller Spalten
  - Regeldefinition
- Generierung des dienstspezifischen XML-Schemas
- 3. Ladelauf

## Demo 4

- Umgang mit Strukturänderungen
  - neue Spalte an Tabelle PA
- 4. Ladelauf
- Synchronisation Metadatenmodell/Data Dictionary
  - Vorschläge
  - Durchführung
- Generierung des dienstspezifischen XML-Schemas
- 5. Ladelauf



# Systemarchitektur



## Features (1)

- Erkennung und Behandlung von Strukturänderungen an für einen Dienst relevanten Datenbanktabellen
- XML-Schema-Generierung
  - beliebige Schachtelungstiefe gemäß Tabellenhierarchie
    - Einzelsätze
    - Listen
  - beliebige Path- und Node-Definitionen, dadurch invariantes Verhalten z.B. gegenüber Umbenennung von Tabellenspalten
- XML-Daten
  - unveränderter Eingabe-XML
  - angereicherter Output-XML
    - Fehlermeldungen und Hinweise
    - Daten, die z.B. beim Insert erzeugt werden, z.B. Primärschlüssel

## Features (2)

- Dienst- und anwendungsspezifisches Regel- und Validierungswerk
  - virtuelle Fremdschlüsselbeziehungen
  - operationsspezifische Pflichtfelder
  - anwendungsspezifische Pflichtfelder
    - kein Anpassungszwang für Anwendungen, z.B. nach Änderung des XML-Schemas infolge von Strukturänderungen der Datenbanktabelle (Abwärtskompatibilität)
  - Definition von Default-Werten
- Umschlüsselung sprechende Schlüssel <=> künstliche Schlüssel
- Historisiertes Metadatenmodell
- Universelle Einsetzbarkeit

# Projekterfahrungen

- Umstellung der bestehenden Anwendung e-Antrag
  - Erzeugung der Validierungsroutinen
  - Befüllung des Metadatenmodells
  - Umstellungsaufwand inkl. Tests 2 MT
- Einbindung der Steuernummer als neue XML-Komponente
  - Strukturänderung an Datenbanktabelle
  - Erzeugung der Validierungsroutinen
  - Aktualisierung des Metadatenmodells
  - Umstellungsaufwand inkl. Tests 1 MT
- Namesänderung einer Tabellenspalte
  - Strukturänderung an Datenbanktabelle
  - Aktualisierung des Metadatenmodells
  - Umstellungsaufwand inkl. Tests 0,5 MT

# Erweiterungen

- im Test
  - lesende Komponente, d.h. generische Erzeugung von XML-Daten aus Datenbanktabellen
  - Validierung mehrspaltiger Fremdschlüssel-Beziehungen
- geplant
  - aussagekräftiges Meldungssystem (mehrsprachig)
  - Definition eines dienstspezifischen Commit-Zeitpunkts
  - Bildung von Dienste-Hierarchien
  - graphische Oberfläche zur Administration des Metadatenmodells

# Fragen

## Kontakt

Bernhard Eichhorn  
MIC Management Consulting GmbH  
Kronacher Straße 4  
D-81549 München

Telefon: +49 (89) 680 711 61

Fax: +49 (89) 680 711 62

E-Mail [bernhard.eichhorn@mic-muenchen.de](mailto:bernhard.eichhorn@mic-muenchen.de)

Internet: [www.mic-muenchen.de](http://www.mic-muenchen.de)