



SIMPL

-

**SimTech: Information Management, Processes and
Languages**

Datenbankanbindungen

Michael Hahn



Agenda

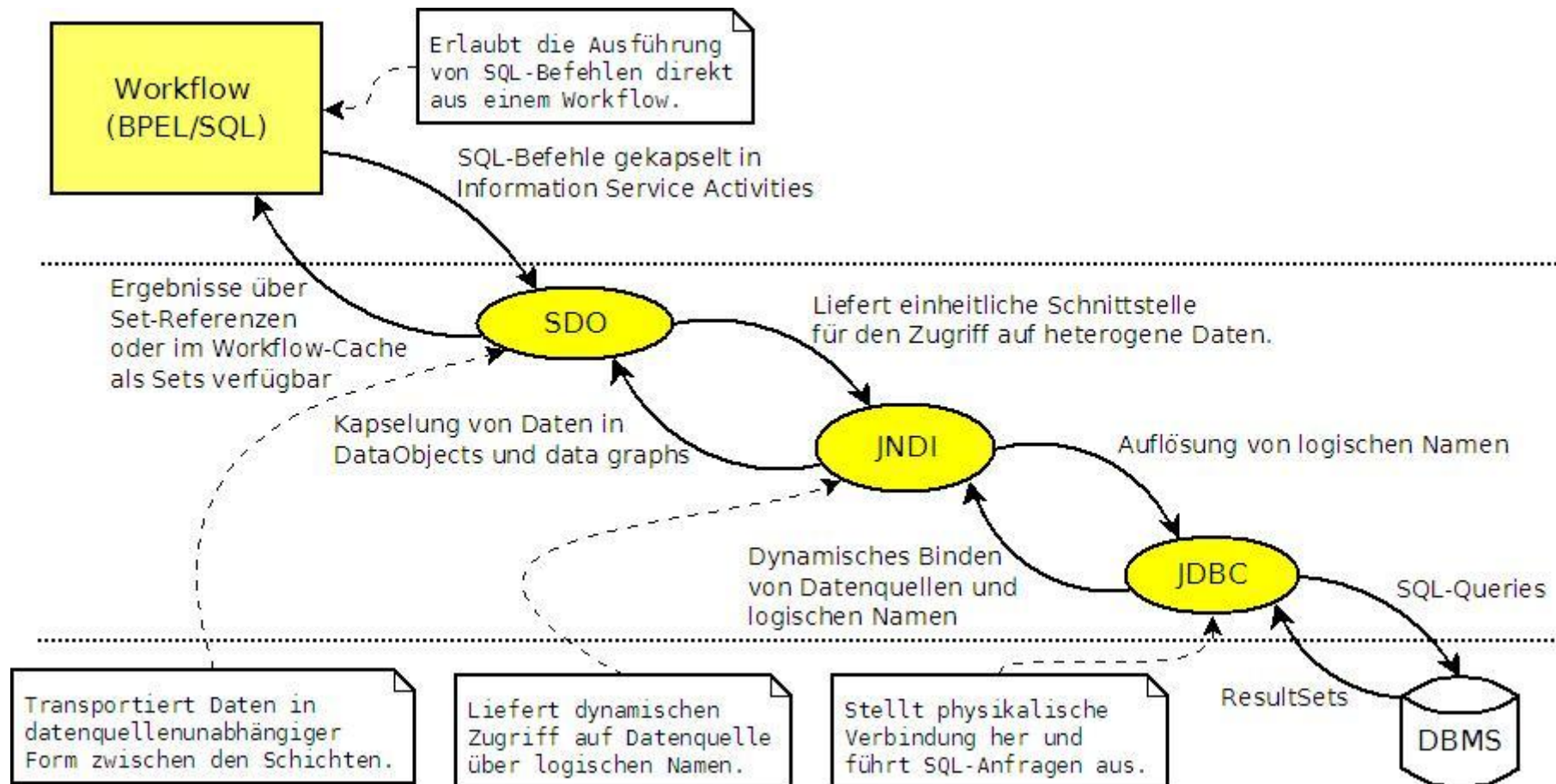
- Motivation
- JDBC
- JNDI
- SDO
- BPEL/SQL
- Zusammenfassung

Motivation

- Ziel: Integration von Datenmanagement in Workflows (mit BPEL/SQL)
- Anforderungen dafür:
 - Bindung von Daten in einen Prozess (mit JDBC)
 - Datenquellenauswahl dynamisch zur Laufzeit (mit JNDI)
 - Datenquellen mittels logischer Namen nicht physikalischer Adressen anbinden (mit JNDI)
 - Einheitlicher Umgang mit heterogenen Daten und Datenquellen (mit SDO)
 - Unabhängigkeit zwischen Daten und ihrer Quelle (SDO)



Motivation (2)





JDBC

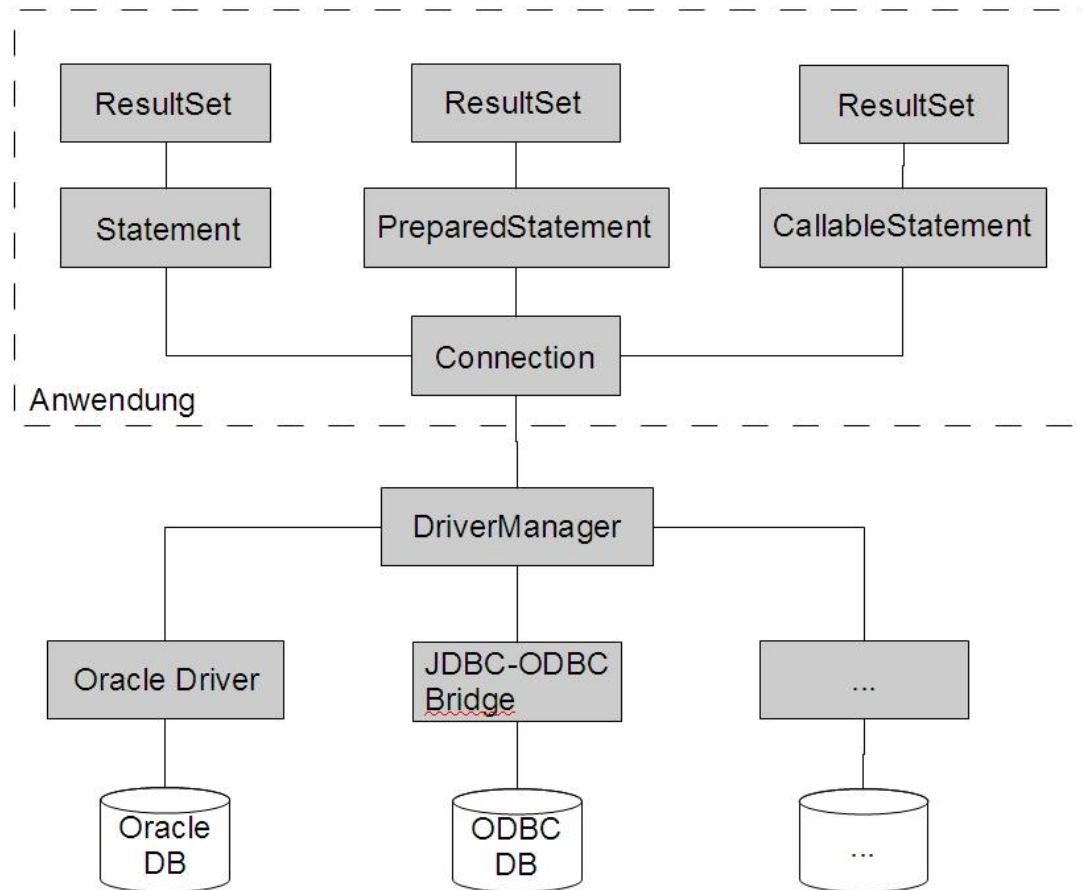
Überblick

- Realisiert physikalische Verbindung zwischen Anwendung und DB
- Hersteller- und Plattformunabhängig
- Ermöglicht dynamisches Binden von Datenquellen zur Laufzeit
- Liefert Methoden für die Ausführung von SQL-Queries, DDL und DML-Befehlen und für die Bearbeitung von Query-Ergebnismengen
- Unterstützt weitere Konzepte, wie z.B. Transaktionen, Savepoints, Database Metadata, ...



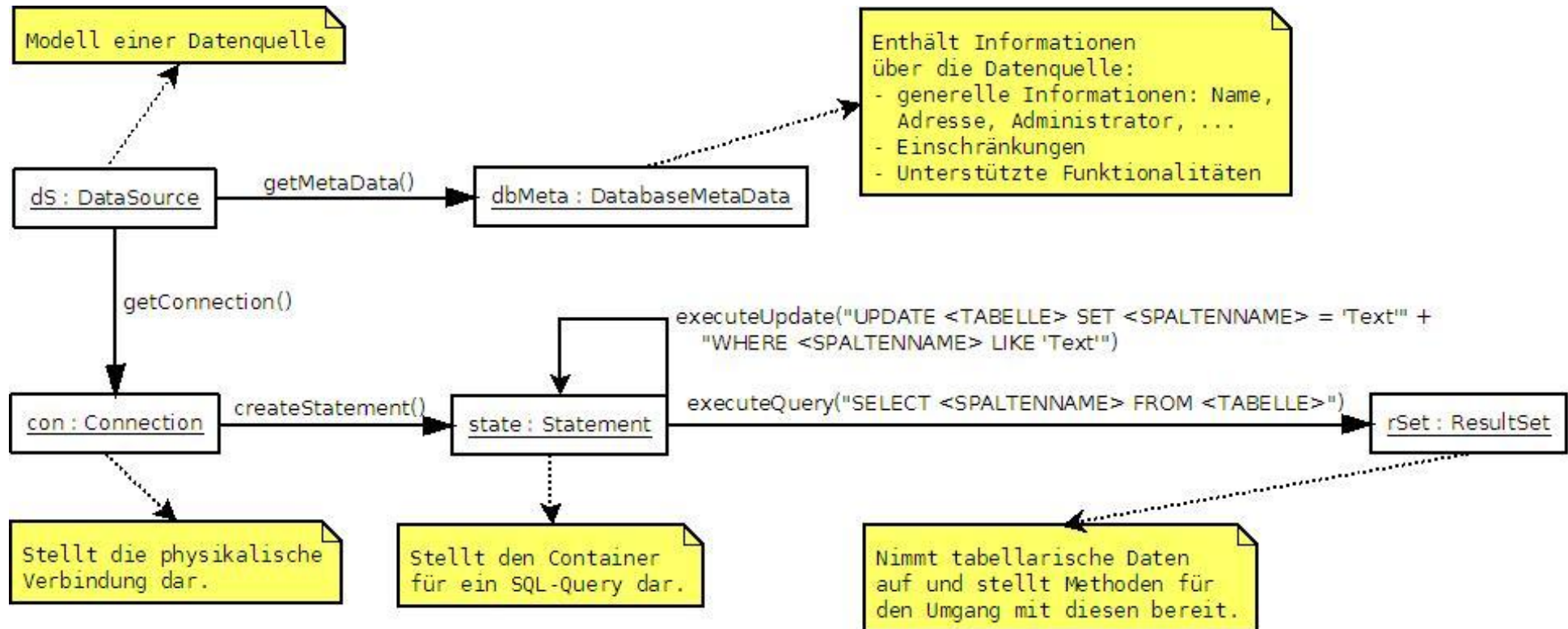


Architektur





API





Beispiel

DB-Tabelle KUNDEN:

Id	Name	Vorname	Umsatz
1	Meyer	Hans	345,20
2	Müller	Lisa	210,39
...

alternativ:

```
(1) stmt.executeUpdate("UPDATE KUNDEN" +
(2)   „SET UMSATZ = 405,20" +
(3)   "WHERE NAME LIKE 'Meyer'");
```

DB-Tabelle KUNDEN nach Bearbeitung:

Id	Name	Vorname	Umsatz
1	Meyer	Hans	405,20
2	Müller	Lisa	210,39
...

```
(1)  MyDataSource dS = new MyDataSource();
(2)  Connection con = dS.getConnection();
(3)  Statement stmt = con.createStatement(
(4)    ResultSet.TYPE_FORWARD_ONLY,
(5)    ResultSet.TYPE_CONCUR_UPDATABLE);
(6)  ResultSet rSet = stmt.executeQuery(
(7)    „SELECT NAME, UMSATZ FROM KUNDEN“);
(8)  while (rSet.next()){
(9)    if (rSet.getName == "Meyer") {
(10)      float umsatz = rSet.getFloat(2);
(11)      umsatz += 60.0;
(12)      rSet.updateFloat(2, umsatz);
(13)      rSet.updateRow();
(14)    }
(15)    rSet.next();
(16)  }
(17) rSet.close();
(18) stmt.close();
```



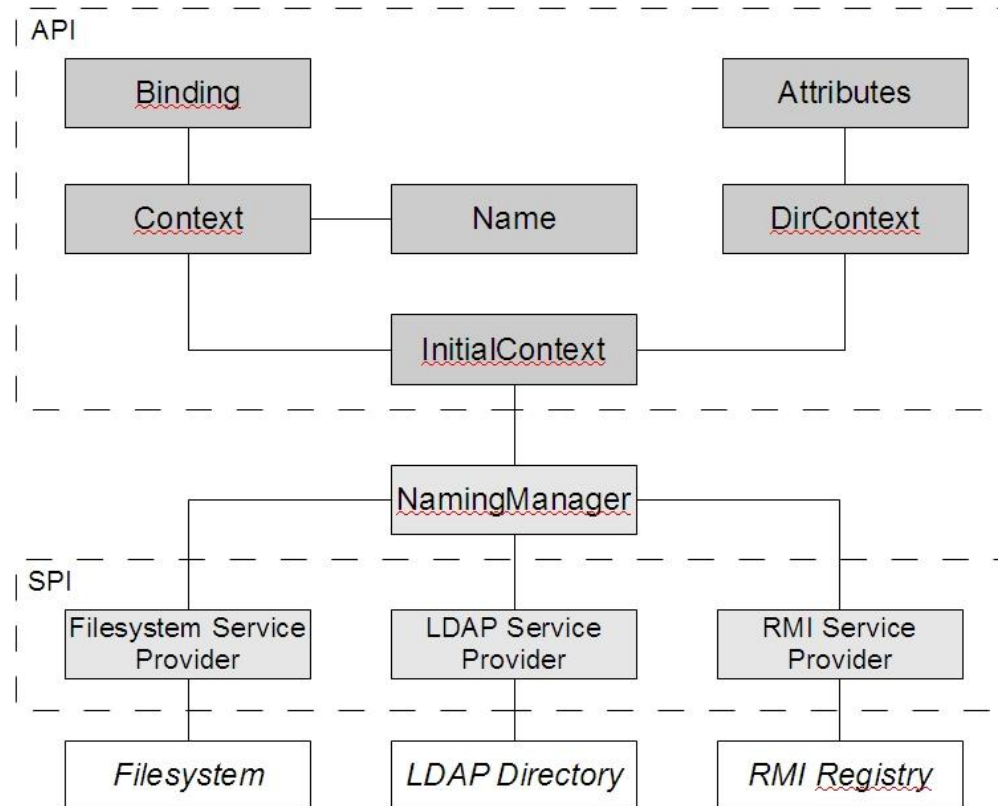
JNDI

Überblick

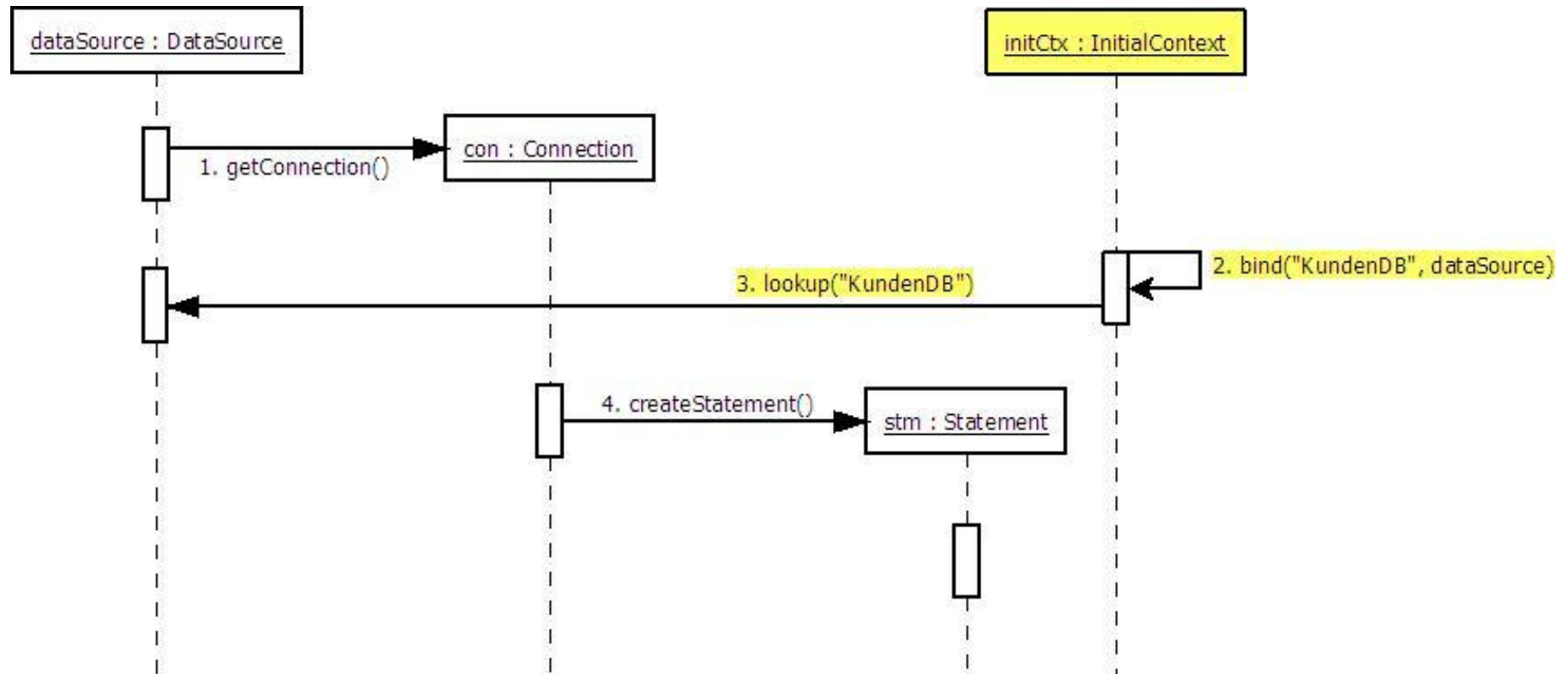
- **Java Naming and Directory Interface**
- Schnittstelle für einheitlichen Zugriff auf *Naming and Directory Services*
- Verknüpft Objekte mit logischen Namen
- Verknüpfungen sind zur Laufzeit änderbar
- Event-Konzept informiert Anwendung über Änderungen von Bindungen



Architektur



Beispiel





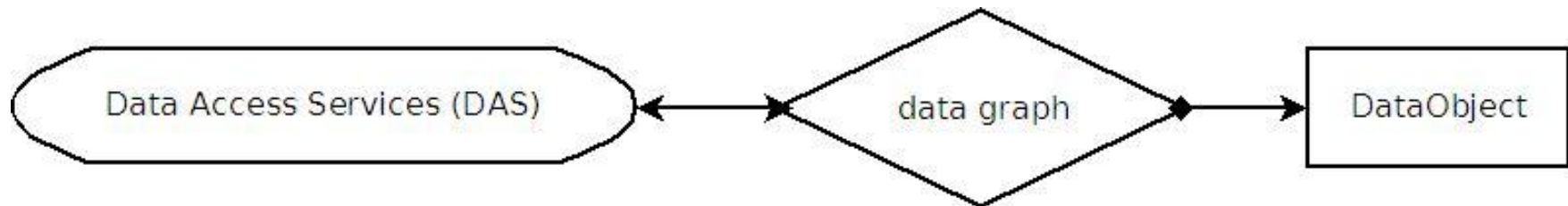
SDO

Überblick

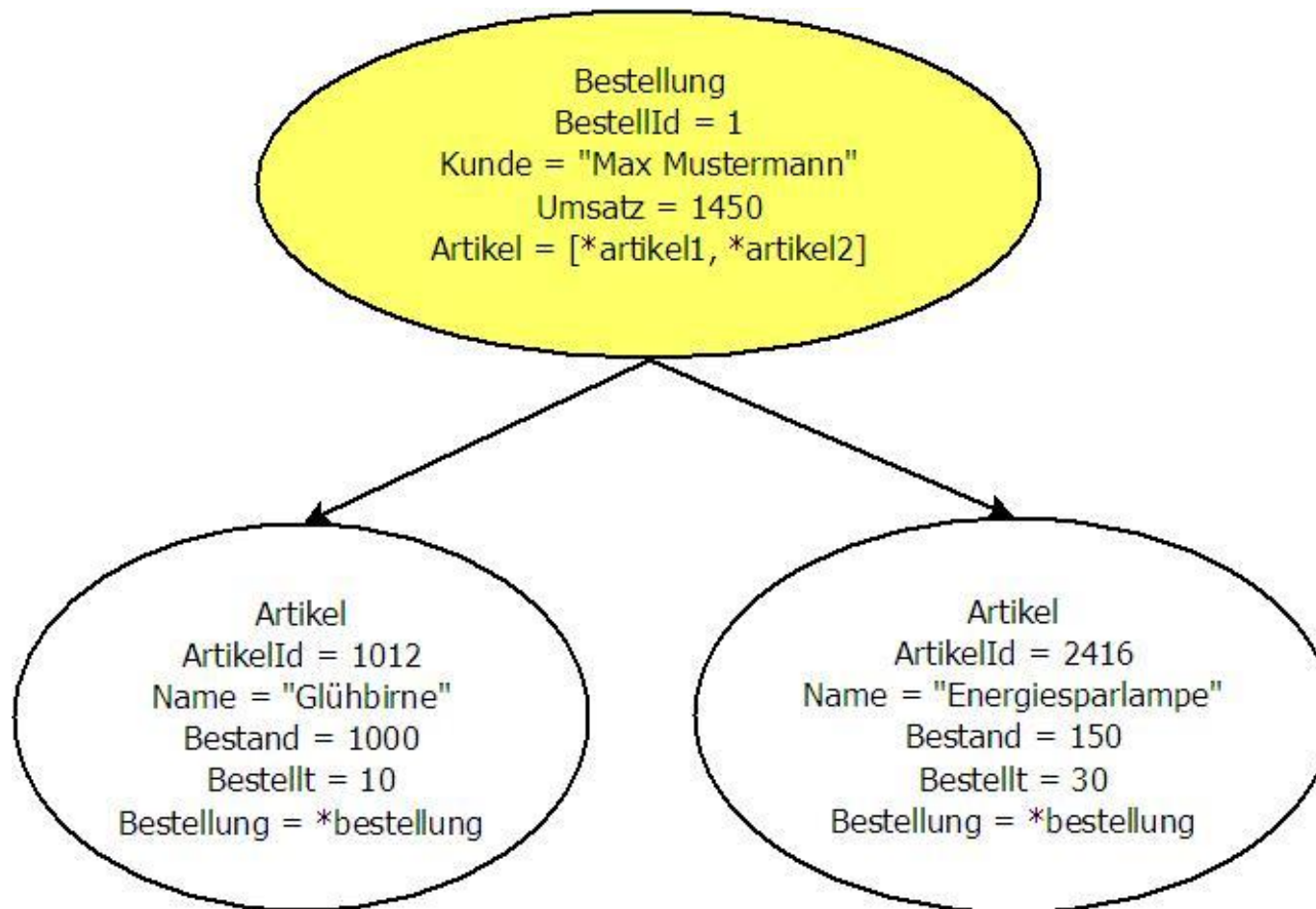
- **Service Data Objects**
- Schnittstelle für einheitlichen Zugriff auf heterogene Daten und Datenquellen
- Data Access Services ermöglichen einheitlichen Zugriff auf Datenquellen
- DataObjects kapseln Daten einer Datenquelle
- Modellierung von DataObjects mittels XML oder Java-Interfaces
- Bietet Änderungsverfolgung um Daten konsistent zu halten



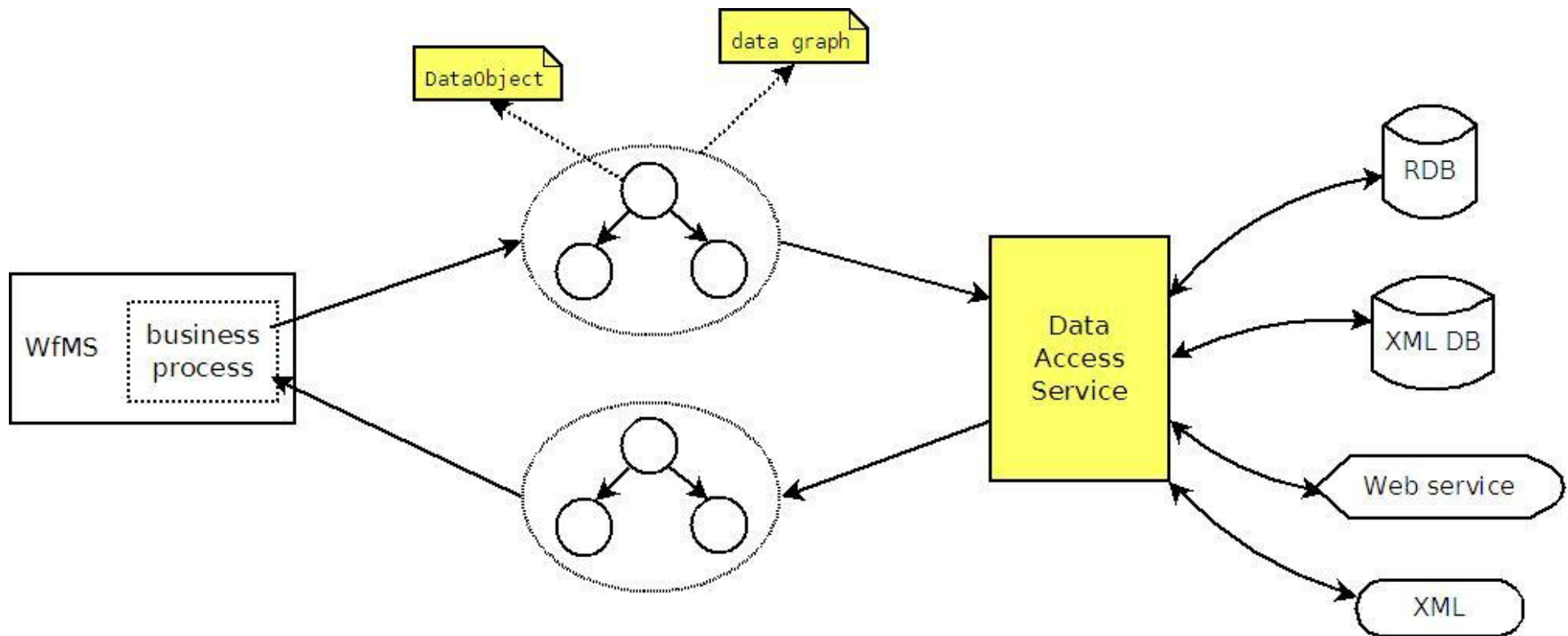
Architektur



Data Graph



Beispiel





BPEL/SQL

Überblick

- inline SQL-Support
- Mehrere herstellerabhängige Ansätze:
 - Microsoft *Workflow Foundation*: Definition und Implementierung neuer Service Activities
 - Oracle *SOA Suite*: XPath Extension Functions
 - IBM *Business Execution Suite*: Information Service Activities

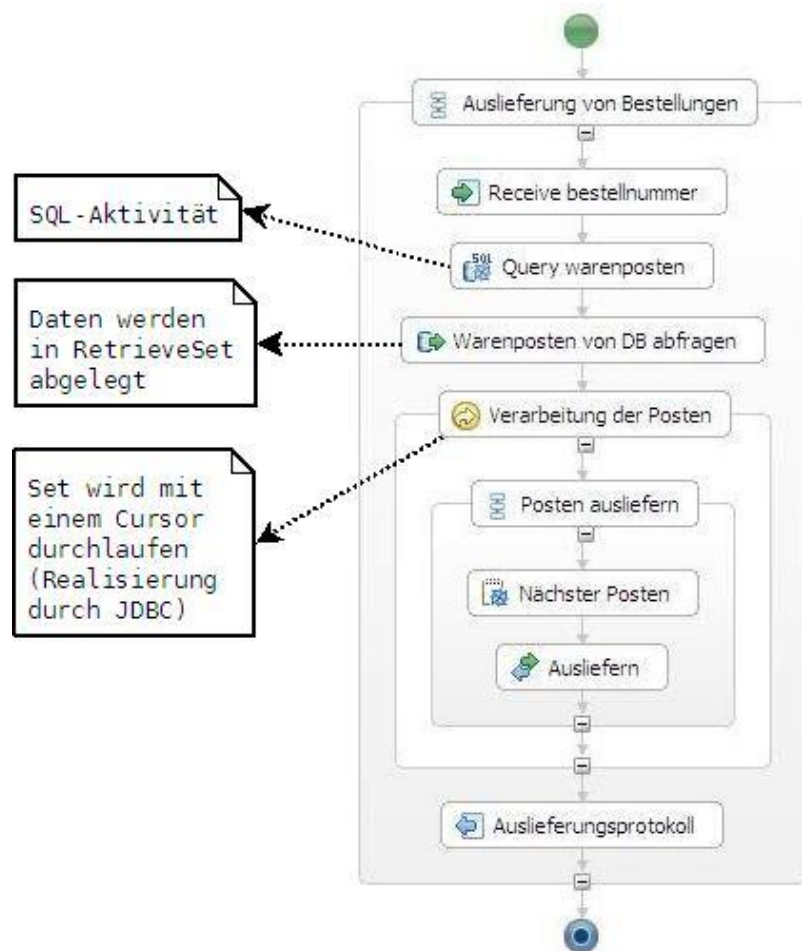


BPEL/SQL - IBM Variante

- *Information Service Activities* erlauben Integration von SQL in BPEL-Prozess
- 3 Typen von *Information Service Activities*:
 - SQL-Snippet: Erlaubt Ausführung von SQL-Befehlen
 - RetrieveSet: Ermöglicht das Laden relationaler Daten in Set-Variablen
 - Atomic SQL Sequence: Kann eine Sequenz von SQL-Snippets und RetrieveSets aufnehmen und ausführen.
- Set-Referenzen sind Pointer auf relationale Tabellen
- Set-Variablen speichern Materialisierung von relationalen Tabellen
- Lifecycle Management für Set-Referenzen
- Datenquelle wird über logischen Namen angesprochen
- BPEL-Variablen als Input-Parameter für SQL-Befehle

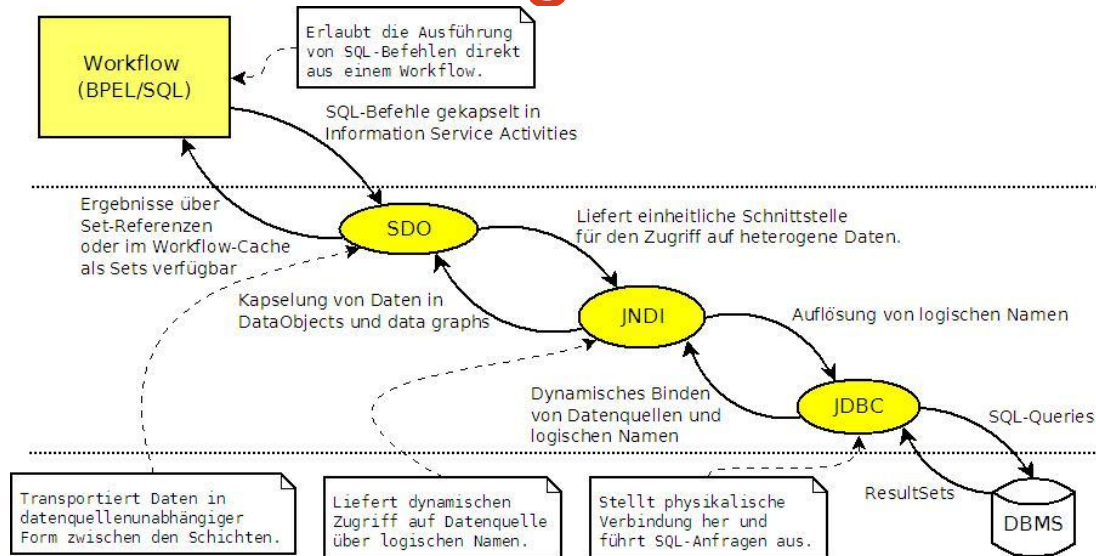


Beispiel



- Sachbearbeiter gibt Bestellnummer ein
- SQL-Query nach Posten der Bestellung
- Ergebnisse der Query werden abgefragt
- Verarbeitung aller Posten der Bestellung
- Posten zur Auslieferung freigeben
- Protokoll der Auslieferungsabwicklung anlegen

Zusammenfassung



- BPEL/SQL sorgt für inline SQL-Support
- SDO transportiert Daten in datenquellenunabhängiger abstrakter Form zwischen den Schichten
- JNDI liefert dynamischen Zugriff auf Datenquellen über logische Namen
- JDBC stellt physikalische Verbindung her und führt SQL-Anfragen aus