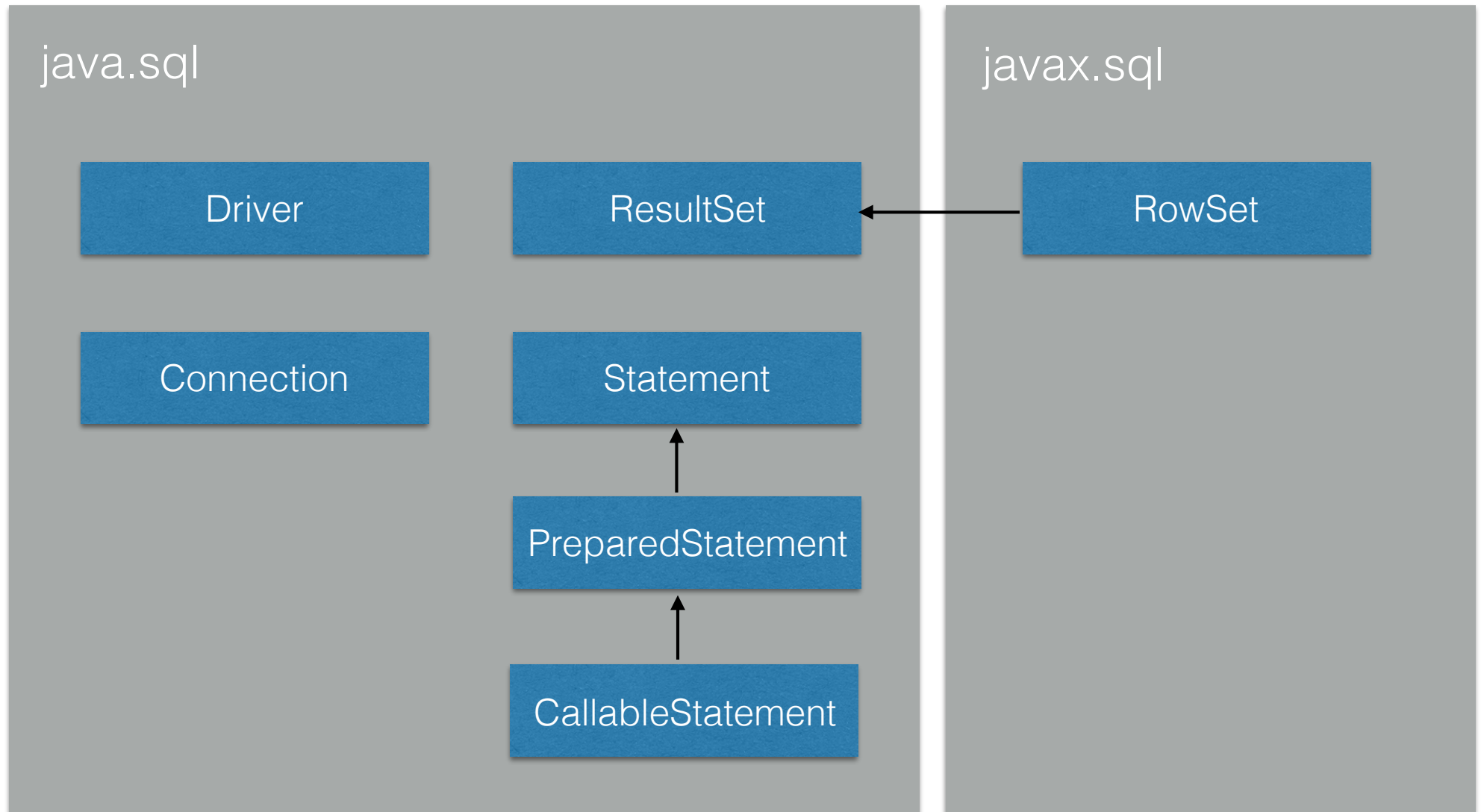


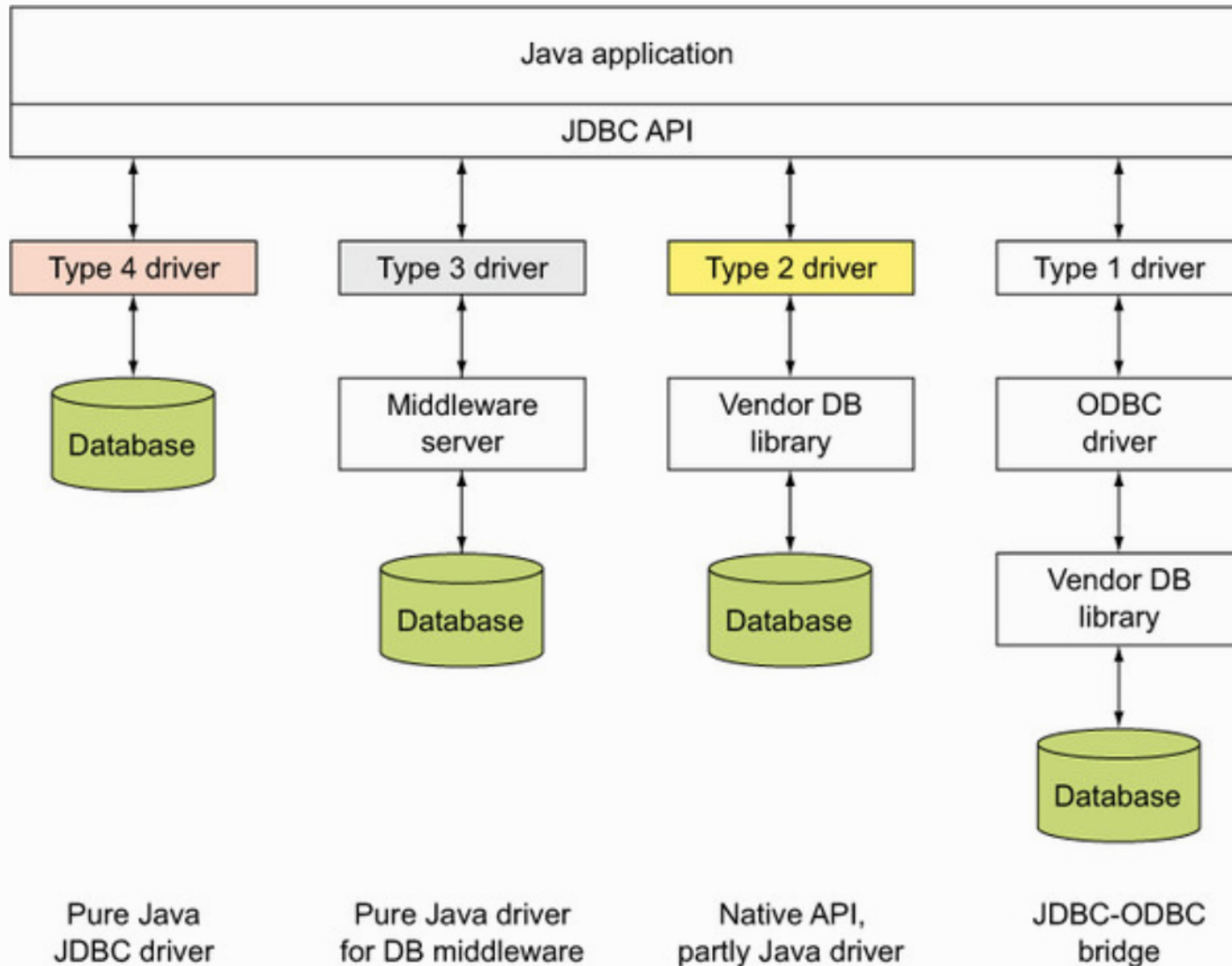
OCP Java SE 8

JDBC

Interfaces



JDBC Typen



Verbindung

- JDBC 3 und davor
 - Treiber laden und registrieren
 - Verbindung aufbauen
- JDBC 4
 - Treiber laden und registrieren (automatisch) und Verbindung aufbauen

Verbindung

- JDBC 3 und davor
 - `Class.forName(...)` // `ClassNotFoundException`
 - `DriverManager.getConnection(...)`
- JDBC 4
 - `DriverManager.getConnection(...)`

URL

- `jdbc:subprotocol://<host>:<port>/<database_name>`
- `jdbc:mysql://localhost:3306/meine_db?" +
"user=usr_001&password=geheim"`

DriverManager

- try-with-resources (Connection)
- `DriverManager.getConnection(url_inkl_zugangsdaten)`
- `DriverManager.getConnection(url, username, password)`
- `DriverManager.getConnection(url, prop) //Properties`

SQL

- DML (Data Manipulation Language)
 - Auswahl-/Einfüge-/Änderungs- und Löschoperationen
- DDL (Data Definition Language)
 - Datenstrukturen und verwandte Element beschreiben, ändern oder entfernen
- DCL (Data Control Language)
 - Berechtigungsvergabe von Lese- und Schreibrechten

CRUD

- create (INSERT)
- read (SELECT)
- update (UPADTE)
- delete (DELETE)

Statement

- try-with-resources (Statement)
- executeQuery(select) //Liefert ein ResultSet
- executeUpdate (insert, update, delete und DDL)
- execute // Beliebiges Statement
 - getResultSet
 - getUpdateCount

ResultSet

- Read-Only (default)
- kann mit next in eine Richtung durchlaufen werden (default)
- scrollable und/oder updatable ResultSets können bei der Erzeugung des Statements gewählt werden.

ResultSet

SQL	Java	Methoden
BOOLEAN	boolean	getBoolean(String label), getBoolean(int index)
INTEGER	int	getInt(String label), getInt(int index)
DOUBLE / FLOAT	double	getDouble(String label), getDouble(int index)
REAL	float	getFloat(String label), getFloat(int index)
BIGINT	long	getLong(String label), getLong(int index)
CHAR, VARCHAR, LONGVARCHAR	String	getString(String label), getString(int index)
DATE	java.sql.Date	getDate(String label), getDate(int index)
TIME	java.sql.Time	getTime(String label), getTime(int index)
TIMESTAMP	java.sql.Timestamp	getTimestamp(String label), getTimestamp(int index)

ResultSet

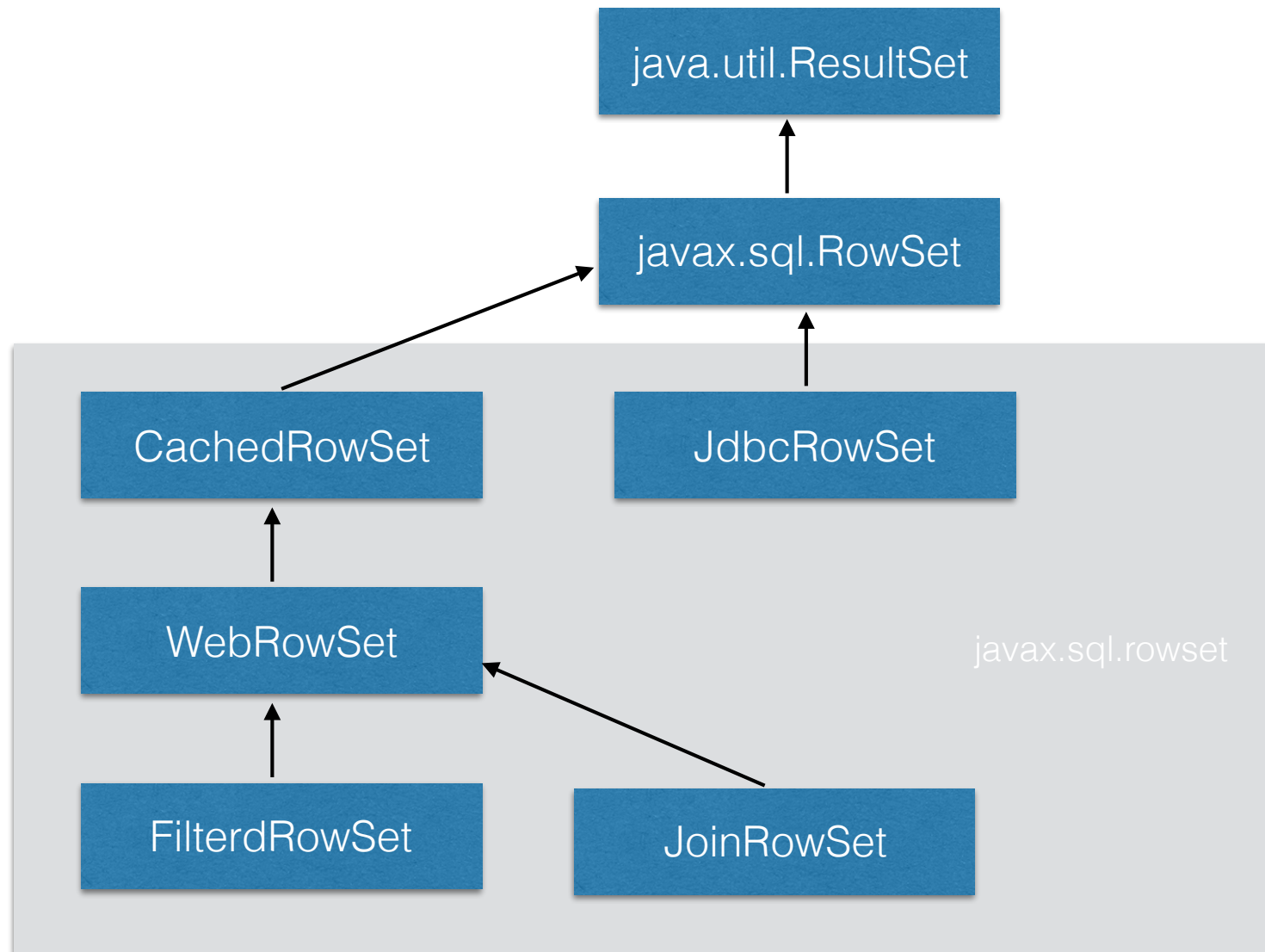
- TYPE_FORWARD_ONLY (default)
- TYPE_SCROLL_INSENSITIVE
 - vor und rückwärts, insensitiv für Änderungen durch andere Anweisungen in gleicher Transaction
- TYPE_SCROLL_SENSITIVE
 - vor und rückwärts, sensitiv für Änderungen durch andere Anweisungen in gleicher Transaction
- CONCUR_READ_ONLY (default)
- CONCUR_UPDATABLE

ResultSetMetaData

RowSet (Interface)

- Ist von ResultSet abgeleitet
- Ist ein Container für Daten
- Verhält sich wie eine JavaBean
- Listener können drangehängt werden
- Ist serialisierbar

RowSet



RowSet Implementierungen

- JDBCRowSet ist ein kleiner Wrapper um das ResultSet, um es als JavaBean zugänglich zu machen. Eine Verbindung zur Datenbank muss bestehen.
- Ein CachedRowSet benötigt initial eine Verbindung zur Datenbank, um mit einer SQL-Anweisung automatisch alle Daten zu lesen. Anschließend ist keine Verbindung zur Datenbank nötig, wobei geänderte Daten später zurückgespielt werden können

RowSet Implementierungen

- Ein WebRowSet erweitert CachedRowSet und bildet Daten und Operationen in XML ab, um sie zum Beispiel für Web-Services übertragen zu können
- Das FilteredRowSet ist ein WebRowSet und ermöglicht eine zusätzliche Selektion mit Prädikaten

RowSet Implementierungen

- Ein JoinRowSet ist ebenfalls ein spezielles WebRowSet, das Daten unterschiedlicher RowSet-Schnittstellen wie über ein SQL-JOIN verbinden kann

RowSetFactory / RowSetProvider

- `RowSetFactory rowsetFactory =
RowSetProvider.newFactory();`
- `JdbcRowSet crs =
rowsetFactory.createJdbcRowSet();`

Transactions

- auto-commit (default) muss ausgeschaltet werden

Savepoints

PreparedStatement

- Precompiled SQL Statements (in der DB)
- beugt SQL-Injection vor
- Können Platzhalter enthalten

CallableStatement

- nutzt Stored Procedures
- schont die Ressourcen