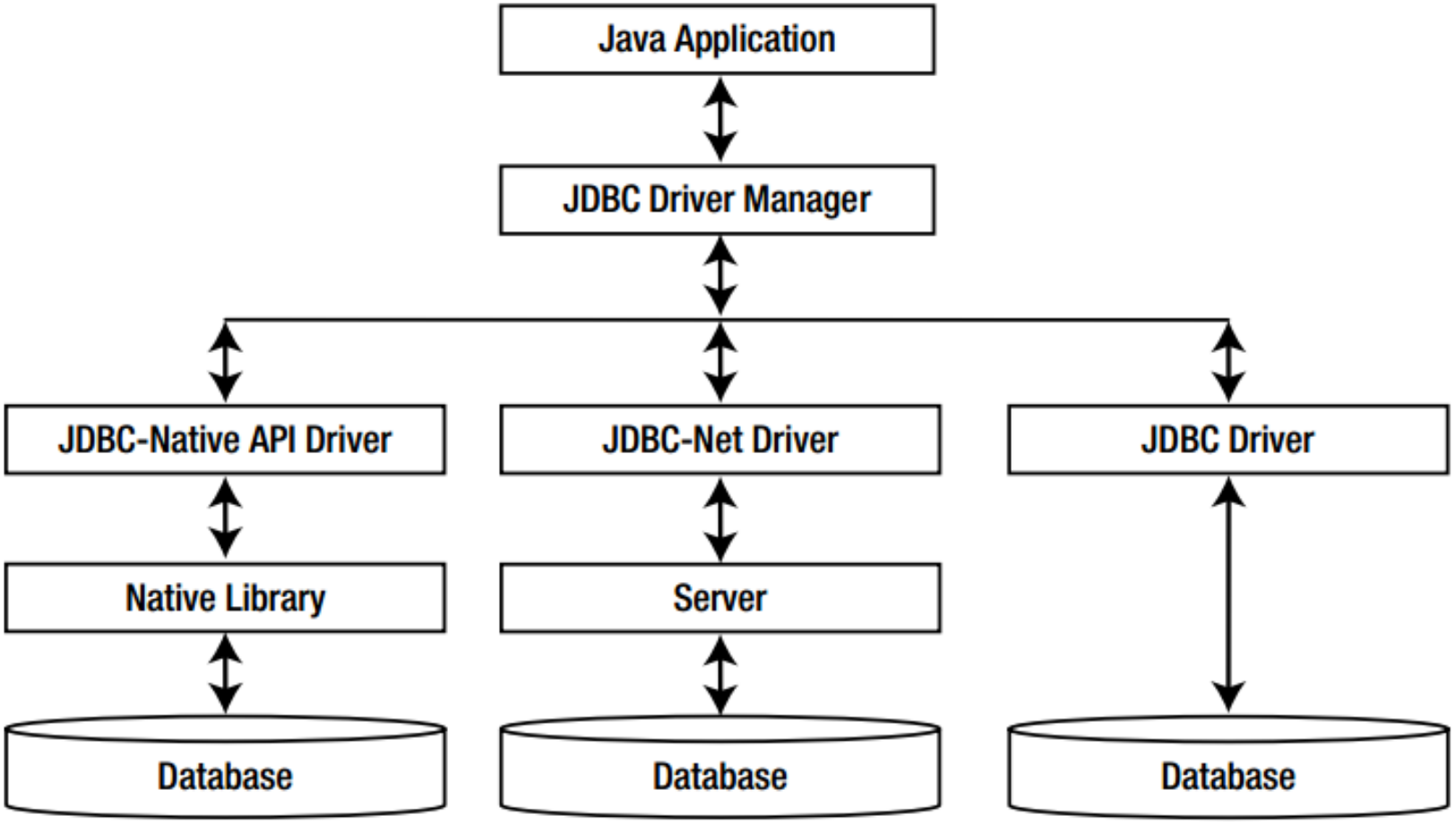
JDBC API

WHAT IS JDBC API?

- The JDBC API provides a standard database-independent interface to interact with any tabular data source. Most of the time, it is used to interact with a relational database management system (RDBMs).

TYPES OF JDBC DRIVERS



DATABASE CONNECTIONS

- Import JDBC Packages.
- Register JDBC Driver.
- Database URL Formulation.
- Create Connection Object.

IMPORT JDBC PACKAGES

- Requires that you include the packages containing the JDBC classes needed for database programming. Most often, using import java.sql.* will suffice.

```
import java.sql.*;
```

REGISTER JDBC DRIVER

- Requires that you initialize a driver so you can open a communication channel with the database.

```
Class.forName("com.mysql.jdbc.Driver");
```

CREATE CONNECTION OBJECT

- Use DriverManager.getConnection() method to create a Connection object, which represents a physical connection with the database.

```
String DB_URL = "jdbc:mysql://localhost/EMP";
String USER = "username";
String PASS = "password";
conn = DriverManager.getConnection(DB_URL, USER,PASS);
```

EXECUTE A QUERY

- Requires using an object of type Statement for building and submitting an SQL statement to the database.

```
Statement stmt = conn.createStatement();
String sql = "SELECT id, first, last, age FROM Employees";
ResultSet rs = stmt.executeQuery(sql);
```

CLEAN UP THE ENVIROMENT

- Requires explicitly closing all database resources versus relying on the JVM's garbage collection.

```
rs.close();
stmt.close();
conn.close();
```

STATEMENTS, PREPAREDSTATEMENT AND CALLABLESTATEMENT

- Statement
- PreparedStatement
- CallableStatement

RESULTSET

A ResultSet object maintains a cursor that points to the current row in the result set.

- Navigational methods.
- Get methods.
- Update methods.

TRANSACTION

- Auto-commit
- Commit
- Rollback