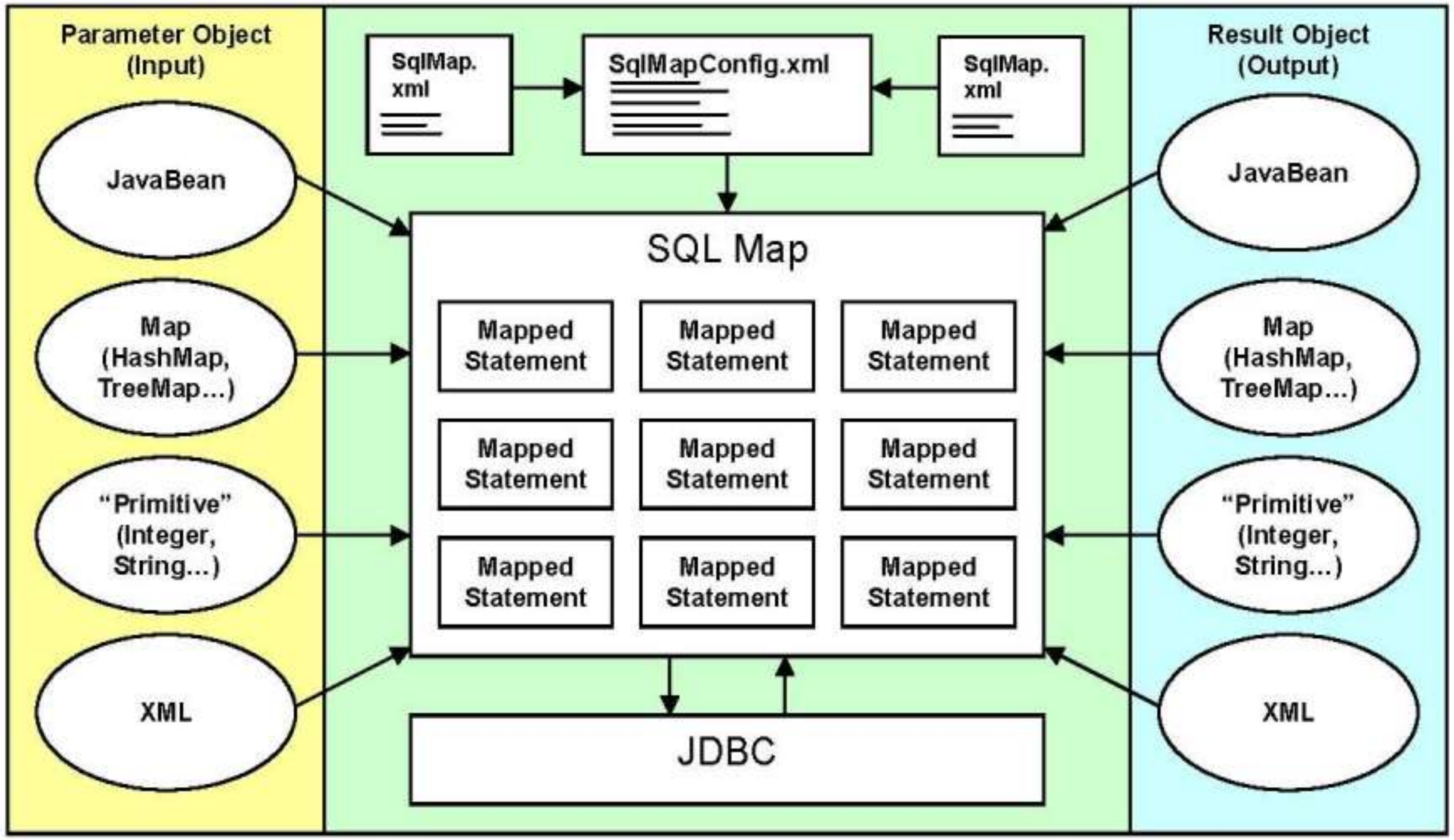
4. Mybatis

1.1 ORM 프레임워크

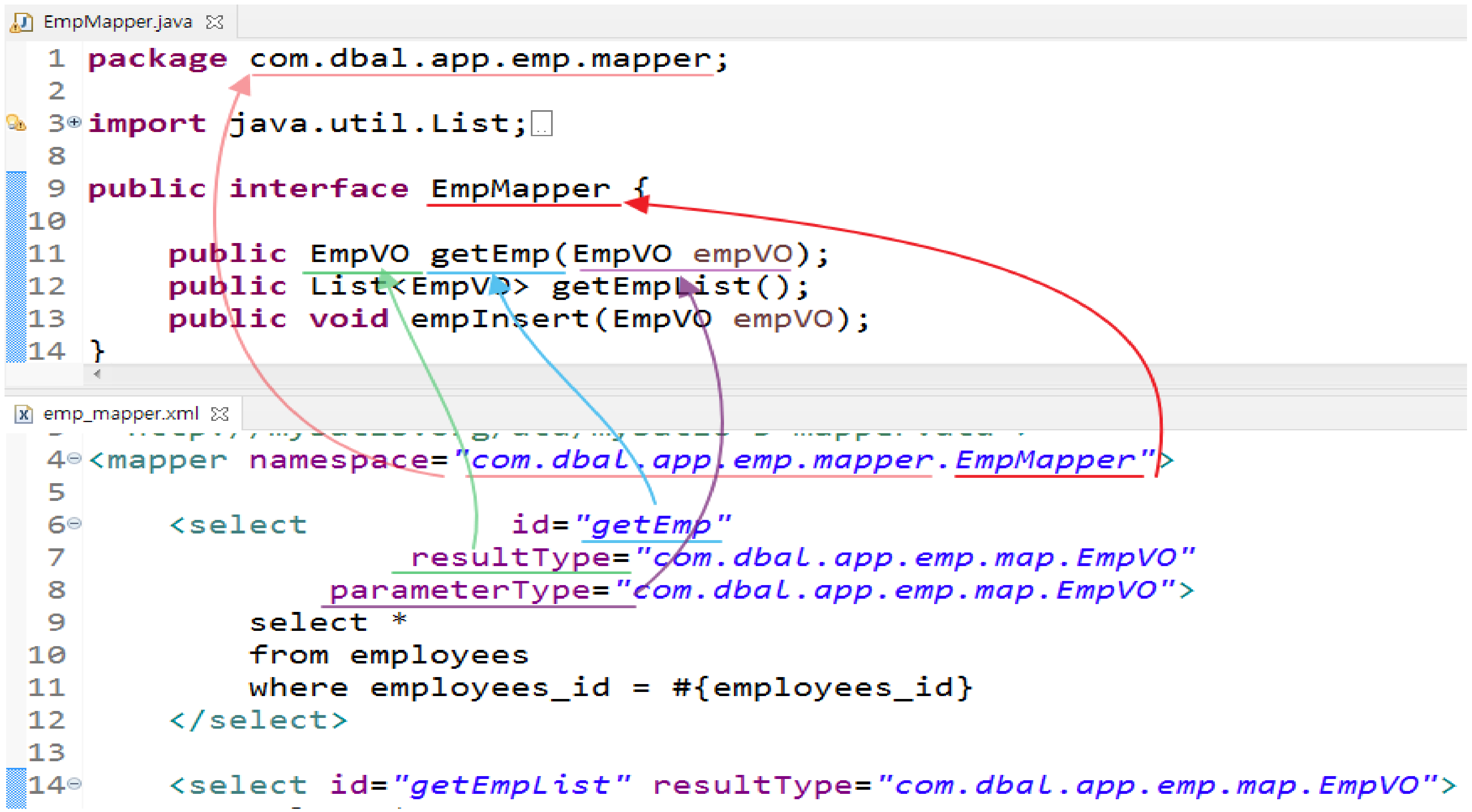
1.2 Mybatis

- JDBC의 모든 기능을 MyBatis가 대부분 제공하므로 한 두 줄의 자바 코드로 DB 연동을 처리.
- SQL 명령어를 자바코드에서 분리하여 XML로 따로 관리
- XML 파일에 저장된 SQL 명령어를 대신 실행하고 실행결과를 VO 같은 자바 객체에 자동으로 매핑까지 해준다.

1.2 Mybatis


Parameter Object
(Input)
JavaBean
Map
(HashMap,
TreeMap…)
"Primitive"
(Integer,
String…)
XML
SqlMap.
xml
SqlMapConfig.xml
SqlMap.
xml
SQL Map
Mapped Statement
Mapped Statement
Mapped Statement
Mapped Statement
Mapped Statement
Mapped Statement
Mapped Statement
Mapped Statement
Mapped Statement
JDBC
Result Object
(Output)
JavaBean
Map
(HashMap,
TreeMap…)
"Primitive"
(Integer,
String…)
XML

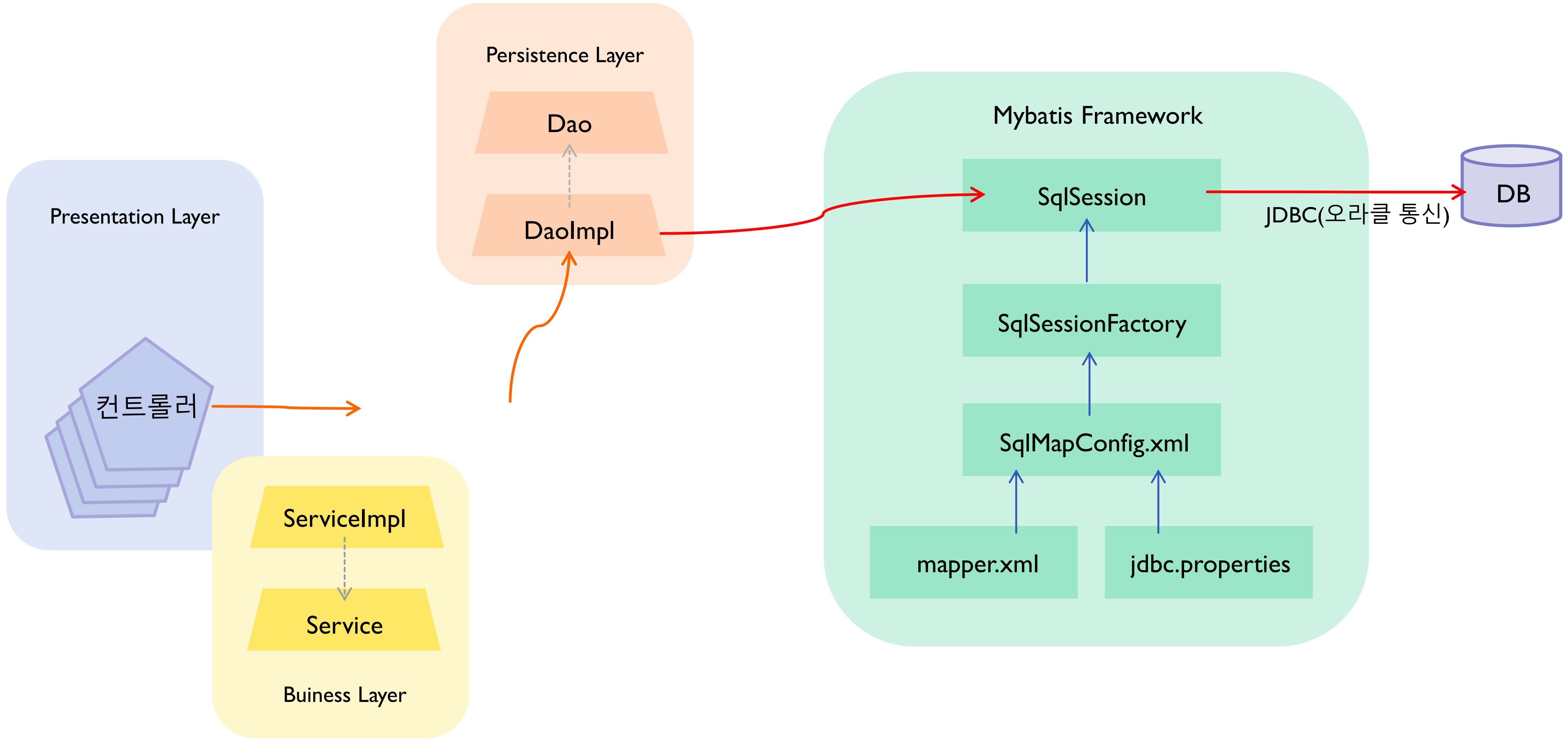
1.2 Mybatis



```java
package com.dbal.app.emp.mapper;

import java.util.List;

public interface EmpMapper {

    public EmpVO getEmp(EmpVO empVO);
    public List<EmpVO> getEmpList();
    public void empInsert(EmpVO empVO);
}
```

```xml
<mapper namespace="com.dbal.app.emp.mapper.EmpMapper">

    <select          id="getEmp"
            resultType="com.dbal.app.emp.map.EmpVO"
        parameterType="com.dbal.app.emp.map.EmpVO">
        select *
        from employees
        where employees_id = #{employees_id}
    </select>

    <select id="getEmpList" resultType="com.dbal.app.emp.map.EmpVO">
```

1.3 Mybatis 주요 컴포넌트



2.1 매퍼 설정

- sql-map-config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
<settings>
   <setting name="jdbcTypeForNull" value="VARCHAR"/>
   <setting name="mapUnderscoreToCamelCase" value="true"/>
</settings>
</configuration>
<!-- Alias 설정 -->
<typeAliases>
    <typeAlias alias="board" type="com.springbook.biz.board.BoardVO"/>
</typeAliases>
<typeHandlers>
  <!-- java.sql.Timestamp 를 java.util.Date 형으로 반환 -->
  <typeHandler javaType="java.sql.Date" handler="org.apache.ibatis.type.DateTypeHandler"/>
</typeHandlers>
```

```
@Repository
publicclassEmpDAO {

    @Autowired
    private SqlSessionTemplate mybatis;

    publicList<EmpVO> getEmpList(EmpVO vo) {
        returnmybatis.selectList("com.dbal.app.emp.map.EmpMapper.getEmpList");
    }

}
```

2.1 매퍼 설정

자바 API

```
package com.dbal.app.emp.mapper;
import java.util.List;
import org.mybatis.spring.SqlSessionTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;
@Repository
public class EmpDAO {
	@Autowired private SqlSessionTemplate mybatis;

	public EmpVO getEmp(EmpVO empVO) {
		return mybatis.selectOne("com.dbal.app.emp.map.EmpMapper.getEmp",empVO );
	}

	public List<EmpVO> getEmpList() {
		return mybatis.selectList("com.dbal.app.emp.map.EmpMapper.getEmpList");
	}
	public void empInsert(EmpVO empVO) {
		mybatis.insert("com.dbal.app.emp.map.EmpMapper.empInsert", empVO);
	}
}
```

트랜잭션

```
<!-- TransactionManager bean 등록 -->
<bean id="txManager"
  class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="dataSource" />
</bean>
<!-- @Transactional 어노테이션 처리 -->
<tx:annotation-driven transaction-manager="transactionManager" />
```

6.1 트랜잭션

```
public void insert(EmpVO vo) {
   try {
      //1. connect
      conn = ds.getConnection();
      //트랜잭션 범위 시작
      conn.setAutoCommit(false);
      //2. statement
      String sql = "INSERT INTO EMPLOYEES~~~ ";
      PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.executeUpdate();
      sql = "INSERT INTO MEMBER ~~~ ";
      pstmt = conn.prepareStatement(sql);
      int r = pstmt.executeUpdate();
      //커밋 : 트랜잭션 범위 종료
      conn.commit();
   } catch(Exception e){
      if(conn != null)
         //롤백 : 트랜잭션 범위 종료
         try { conn.rollback(); } catch (SQLException e1) { }
   } finally {
      if(conn != null)
         try { conn.close(); } catch (SQLException e1) { }
   }
}
```

7.1 프로시져

```
<parameterMap type="board" id="boardParam">
        <parameter property="title" mode="IN" jdbcType="VARCHAR" javaType="string"/>
        <parameter property="writer" mode="IN" jdbcType="VARCHAR" javaType="string"/>
        <parameter property="content" mode="IN" jdbcType="VARCHAR" javaType="string"/>
        <parameter property="seq" mode="OUT" jdbcType="NUMERIC" javaType="int"/>
        <parameter property="out_msg" mode="OUT" jdbcType="VARCHAR" javaType="string"/>
</parameterMap>

<insert id="insertBoardProc1" statementType="CALLABLE" parameterMap="boardParam">
        { call BOARD_INS_PROC(?,?,?,?,?) }
</insert>
<insert id="insertBoardProc2" statementType="CALLABLE" parameterType="board">
        { call BOARD_INS_PROC(
         #{title},
         #{writer},
         #{content, mode=IN, jdbcType=VARCHAR, javaType=string},
         #{seq, mode=OUT, jdbcType=NUMERIC, javaType=java.math.BigDecimal},
         #{out_msg, mode=OUT, jdbcType=VARCHAR, javaType=string}
         )
        }
</insert>
```