

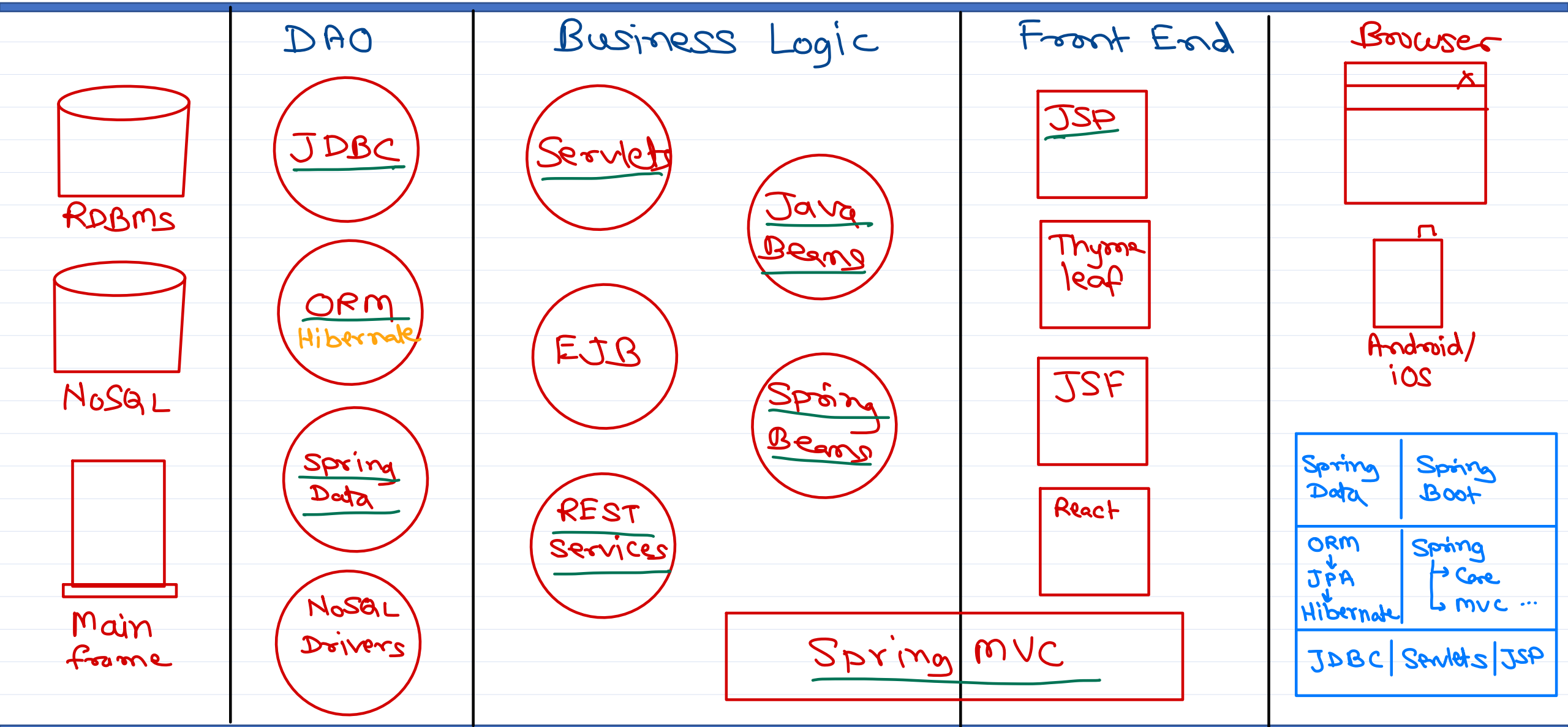


Advanced Java

Trainer: Nilesh Ghule



Java EE - Enterprise applications



Java EE → set of specifications

- servlet
- JSP
- JSF
- JPA
- JTA
- EJB
- JDBC

web servers



follow Java EE
Specs and
develop web
servers.

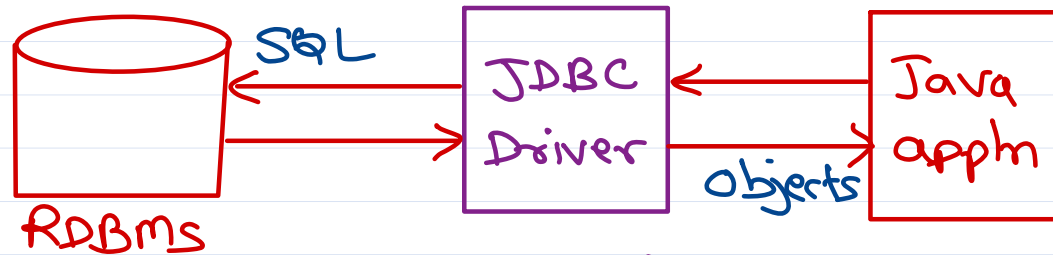
web developers



follow Java EE
Specs to implement
various components
in web appln.



JDBC



Type - IV Driver

- ✓ Fully in Java
- ✓ Platform independent
- ✓ Fast execution
- ✓ Db Specific

JDBC is a specification/standard.

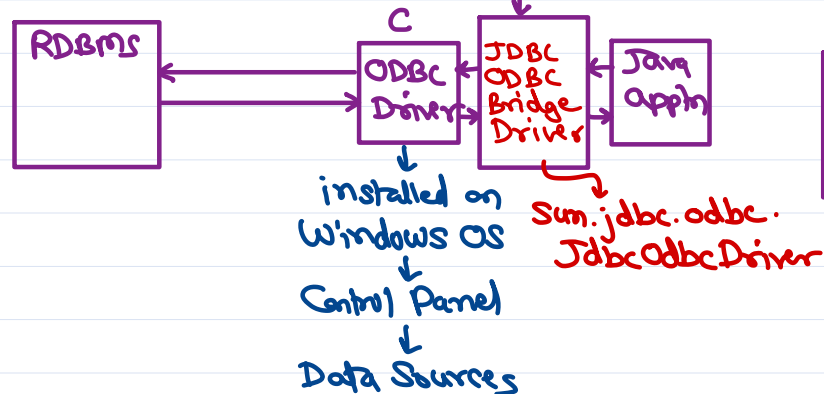
JDBC specs = interfaces + helper classes.

- ① Driver
 - ② Connection
 - ③ Statement
 - ④ ResultSet
 - ⑤ DriverManager*
 - ⑥ PreparedStatement
 - ⑦ CallableStatement
- } java.sql Package

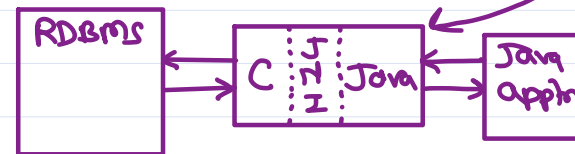
JDBC implementations = JDBC drivers.

- ✓ Diff drivers for diff RDBMS.
- ✓ given in form of jar files = zip *.class files.
- ✓ classes in jdbc driver jar implement JDBC intfs.

Type-I driver.

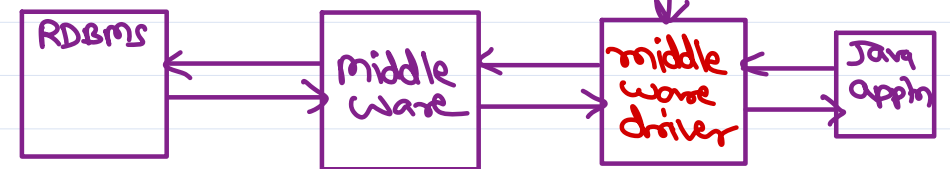


Type-II Driver



- ✓ DB specific
- ✓ not platform independent
- ✓ faster

Type - III Driver



- ✓ specialized requirements only

JDBC Programming Steps - PreparedStatement.

① Add JDBC driver jar into project/classpath.

Eclipse - Java Project → Properties → Java Build Path → Libraries → Add External Jars →
Select mysql driver jar → Apply & Close

② Load JDBC driver & register it.

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

③ Create database connection.

```
url = "jdbc:mysql://localhost:3306/dbname";  
con = DriverManager.getConnection(url, "user", "passwd");
```

④ Create parameterized Sql statement.

```
stmt = con.prepareStatement("param sql query");
```

⑤ Set params, execute query & process result.

```
stmt.setInt(1, val1);  
stmt.setString(2, val2);  
cnt = stmt.executeUpdate();
```

OR

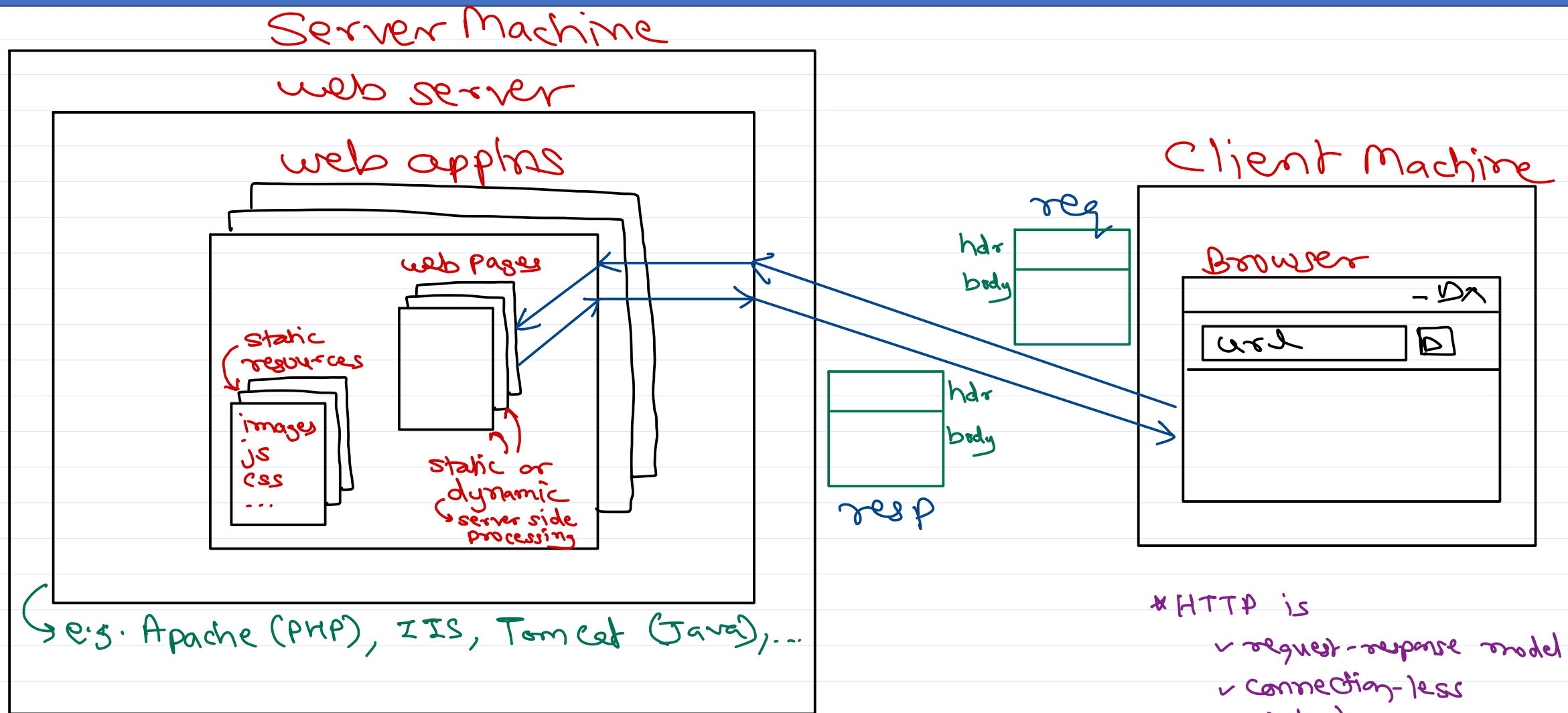
```
rs = stmt.executeQuery();  
while(rs.next()) {  
    v1 = rs.getInt("col1");  
    v2 = rs.getString("col2");  
    ...  
}  
rs.close();
```

⑥ close statement & connection.

```
stmt.close();  
con.close();
```



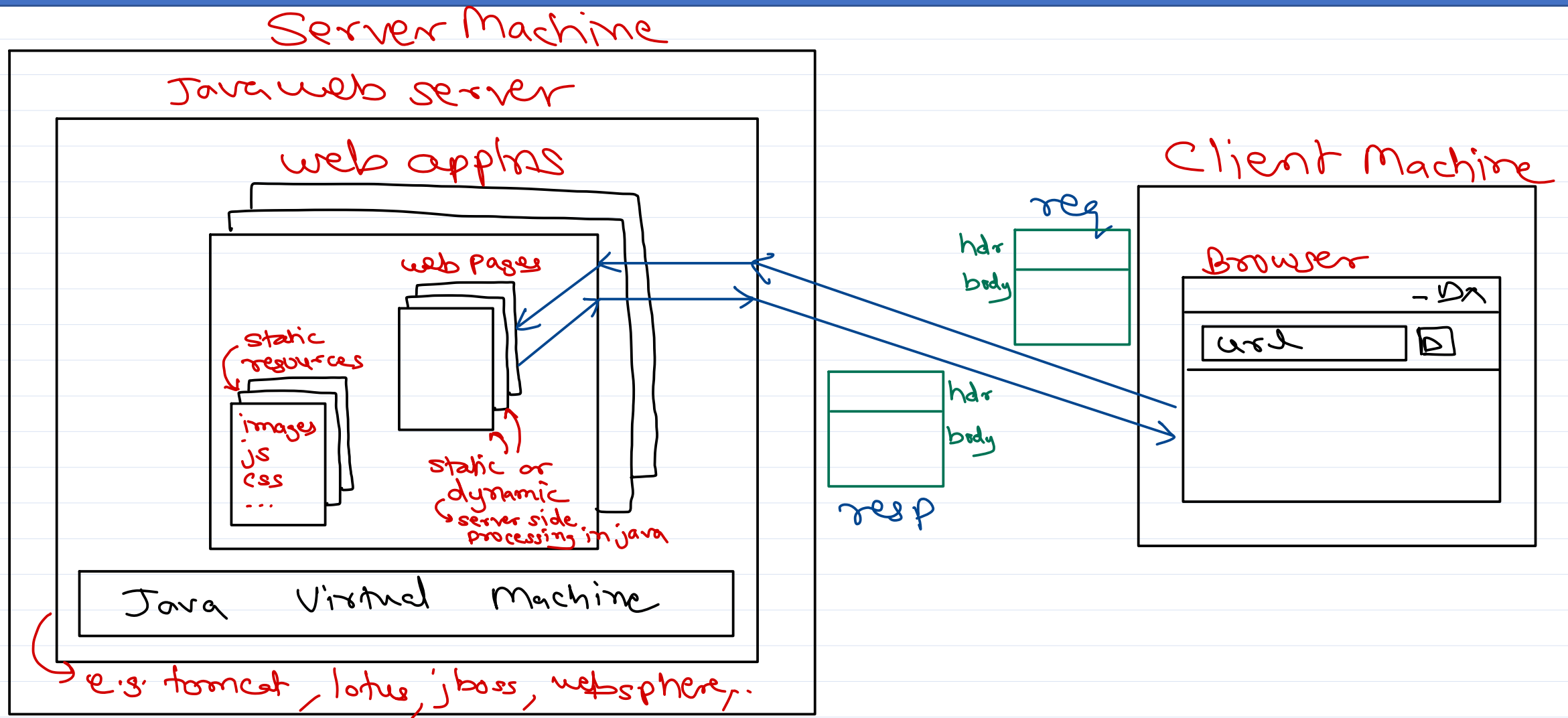
HTTP Protocol



- * HTTP is
 - ✓ request-response model
 - ✓ connection-less
 - ✓ state-less



Java Web Server

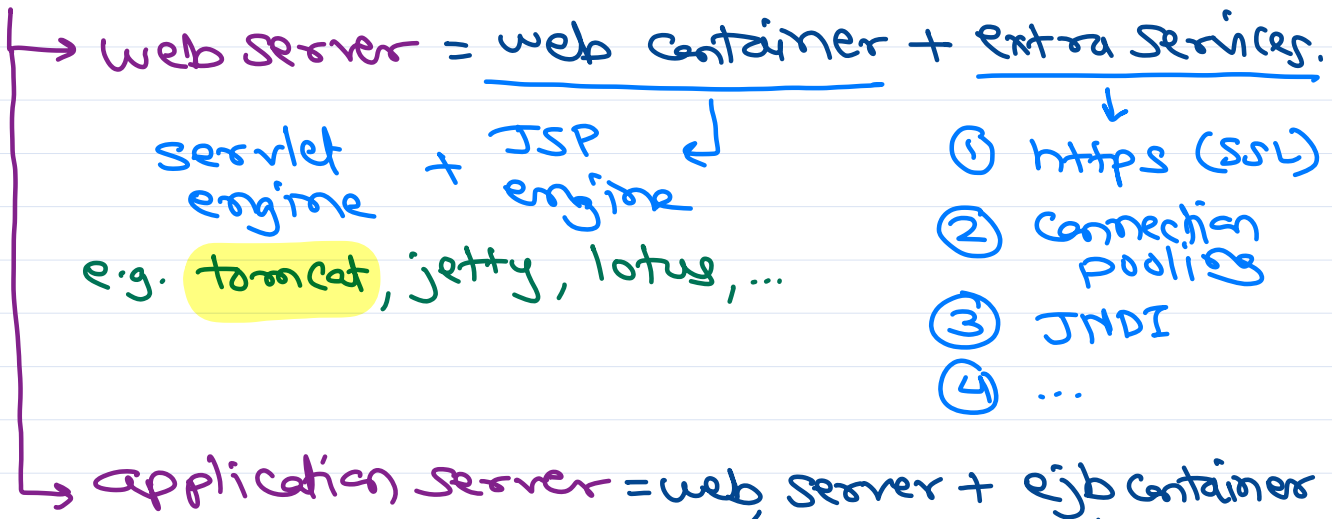


Java servers

Sun Microsystems

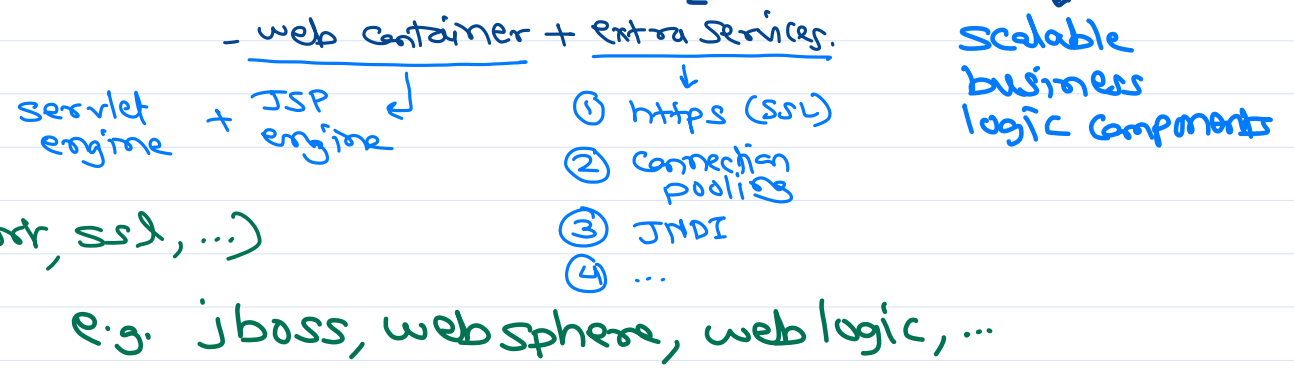
- standards / specifications.
- JVM
- JDBC
- Java EE ← to be followed by java web servers & java web appln.
 - Servlet - JTA
 - JSP - EJB
 - JSF - JMS
 - JPA - ...

Java Servers

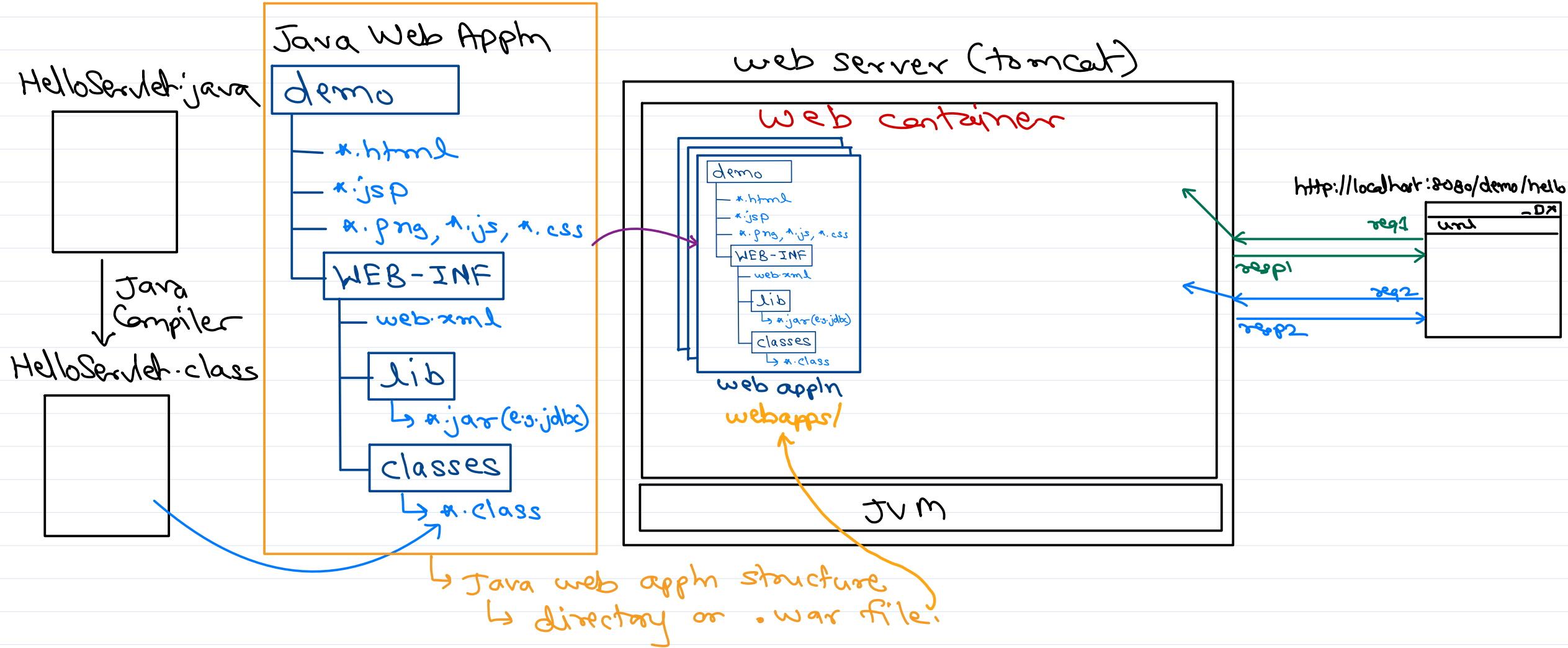


Apache Tomcat → default port = 8080

- bin → tomcat bins, startup/shutdown.
- lib → tomcat libs (jars)
- Conf → tomcat, users, xml, server, xml (change port, ssl, ...)
- webapps → hot deployment folder
- logs → log files (monitoring)
- work → JSP processing
- temp → temp files.



HelloServlet execution





Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

