Advanced Java

Trainer: Nilesh Ghule

JSTL

* Popular tag lib by Sun Microsystem.
* Five components :
  1. core → programming, redirection, url, ...
  2. fmt → format date-time & numbers
  3. functions → util fns for strings, ...
  4. sql → execute sql queries on db.
  5. xml → xml generation & parsing, ...

* <%@ taglib prefix="c" uri="... /core" %>

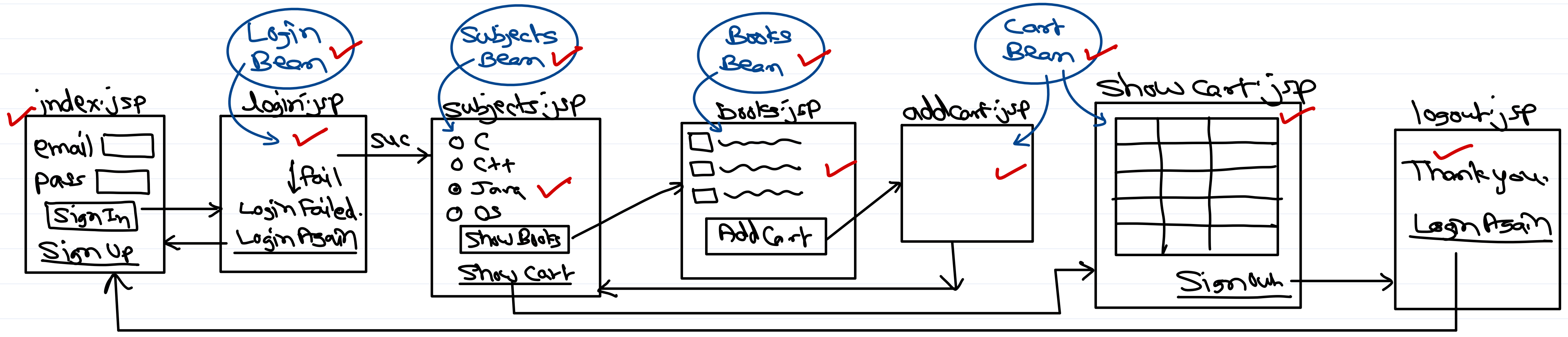
```
* <c:forEach var="b" items="${bb.books}">
          var name         collection
    ${b.id}
    ${b.name}
</c:forEach>
```

* <c:redirect url="~" />

```
<c:choose>
    <c:when test="${cond1}">
        ~
    </c:when>
    <c:when test="${cond2}">
        ~
    </c:when>
    <c:otherwise>
        ~
    </c:otherwise>
</c:choose>
```

```
<c:if test="${cond1}">
    ~
</c:if>
```

BookShop using JSP — Customer side

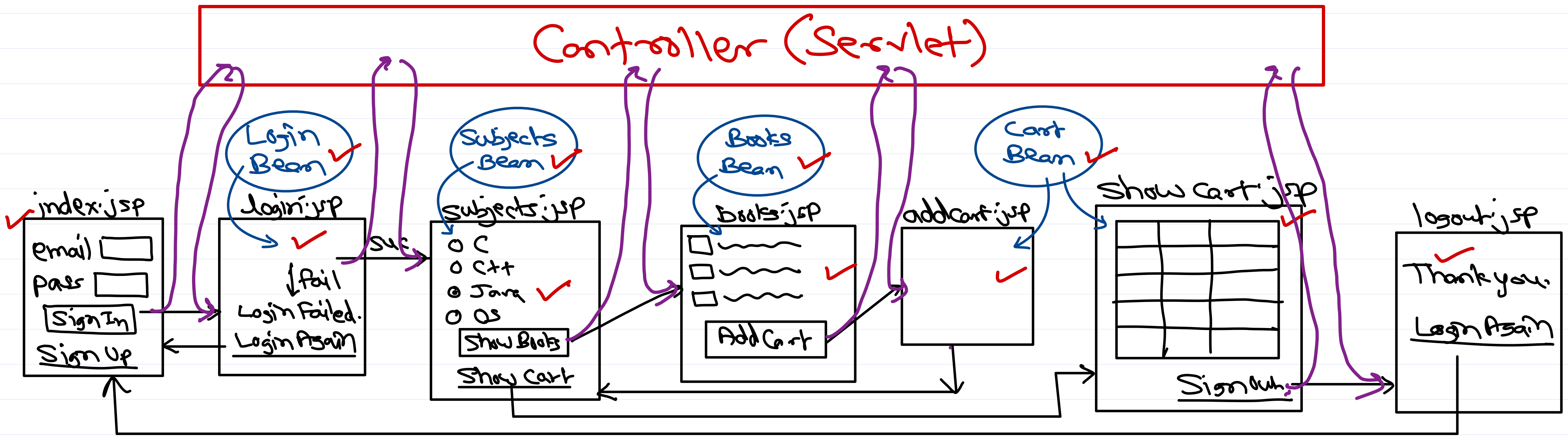


* model-View architecture.
  - ✓ model → data of appln (input/output) ⇒ java beans
  - ✓ view → display data ⇒ JSP pages.
* characteristics
  1. views are tightly coupled with each other.
  2. beans are tightly coupled with views.

✓ good for small applns.

X for big appln any changes into view / model need to reflect at multiple places.

BookShop using JSP — Customer side



* model-View-Controller arch
- ✓ model → Data ⇒ Java Beans
- ✓ View → Display ⇒ JSP
- ✓ Controller → Loose coupling betn
  - View → View
  - & View → model

✓ maintainable.

✗ Complex for small appln.

Spring

* 2003 - Rod Johnson
  ↳ XML based config (Java 1.4)
* 2007 - Spring 2.5
  ↳ added annotation config (Java 5)
* 2010 - Spring 3
  ↳ popular spring version
* Lightweight Comprehensive framework
  to simplify Java development.
  → basic Spring libs : in 1-2 MBs.
  → Covers many features of Java dev.
* Dependency Injection.
* Interfaces & POJOs (spring beans)
  ↳ Loose coupling betn client & impls.
  → Simple classes (fields + ctor + get/set + BL).

* readable exceptions.
* TDD : Test Driven Development.
  ↳ unit testing
  ↳ integration testing

```
class Arith {
    add(x,y) {
        return x+y;
    }
}
```

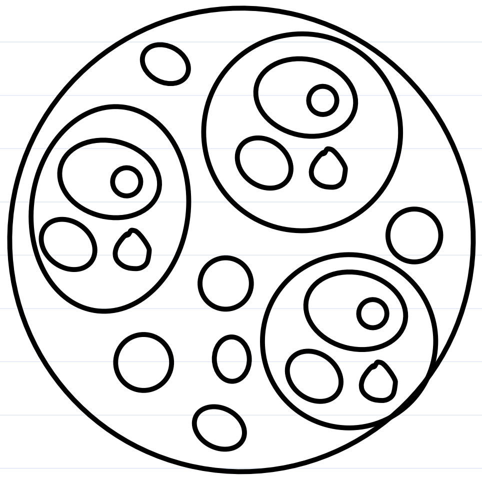
```
testAdd() {
    a = new Arith();
    r = a.add(2,3);
    assert 5 == r;
           ↑      ↑
     expected   actual
}
```

* Modular & Flexible

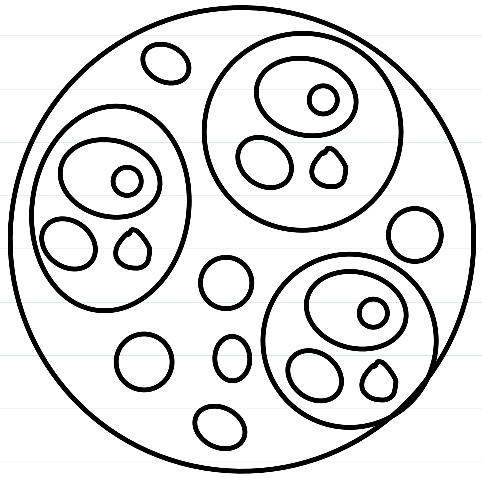
jdbc	core	web	...
mvc	orm	security	...

* works will all Java tech.
* Eliminate boiler plate (repeated) code.

Spring - Dependency Injection



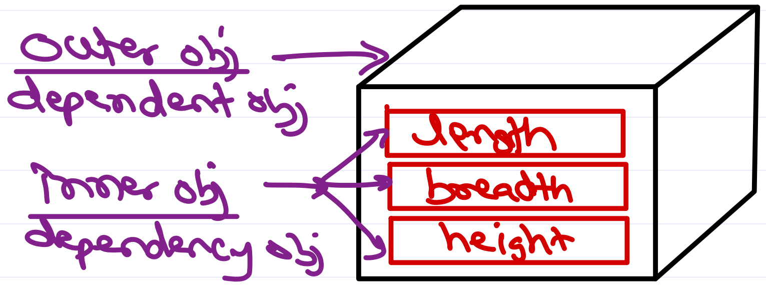
```
Car c = new Car();
 c.setEngine(e);
 c.setChassis(ch);
 c.setWheels(w);
 ...
 c.drive();
```



```
Car c = get car;
 c.drive();
```

Spring Container will create & initialize bean objects.
Spring Container = IoC container

↑ Inversion of Control ↓



```
class BoxImpl {
  int l, b, h;
  //ctor
  //get/set
  void calcVol(){
    return l*b*h;
  }
}
```

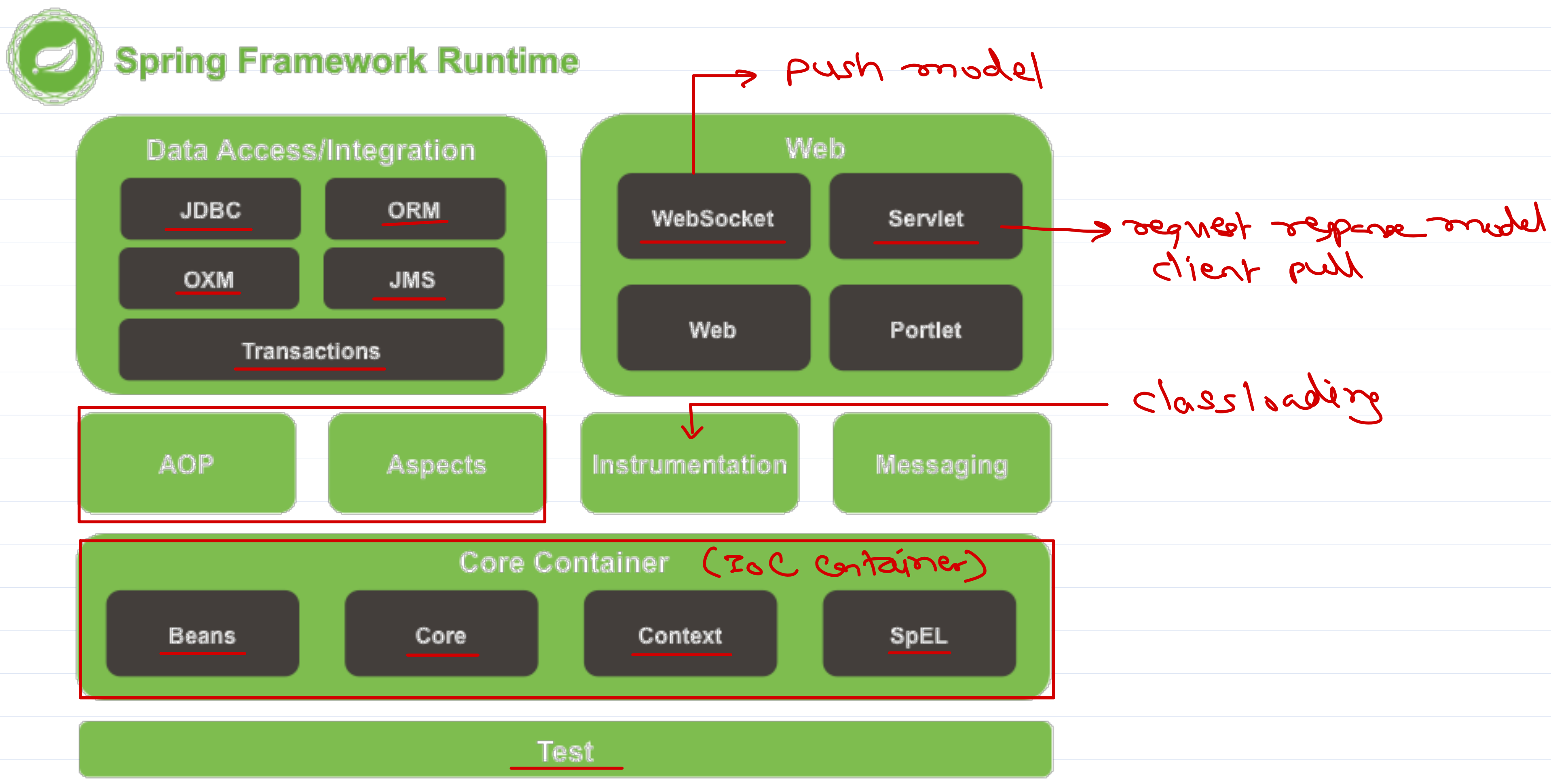
```
b=new BoxImpl();
b.setLen(5);
b.set Br(4);
b.set Ht(3);
print(b.calcVol());
```

```
ctx → Spring Ctx
b= ctx.getBean
     (BoxImpl);
print(b.calcVol());
```

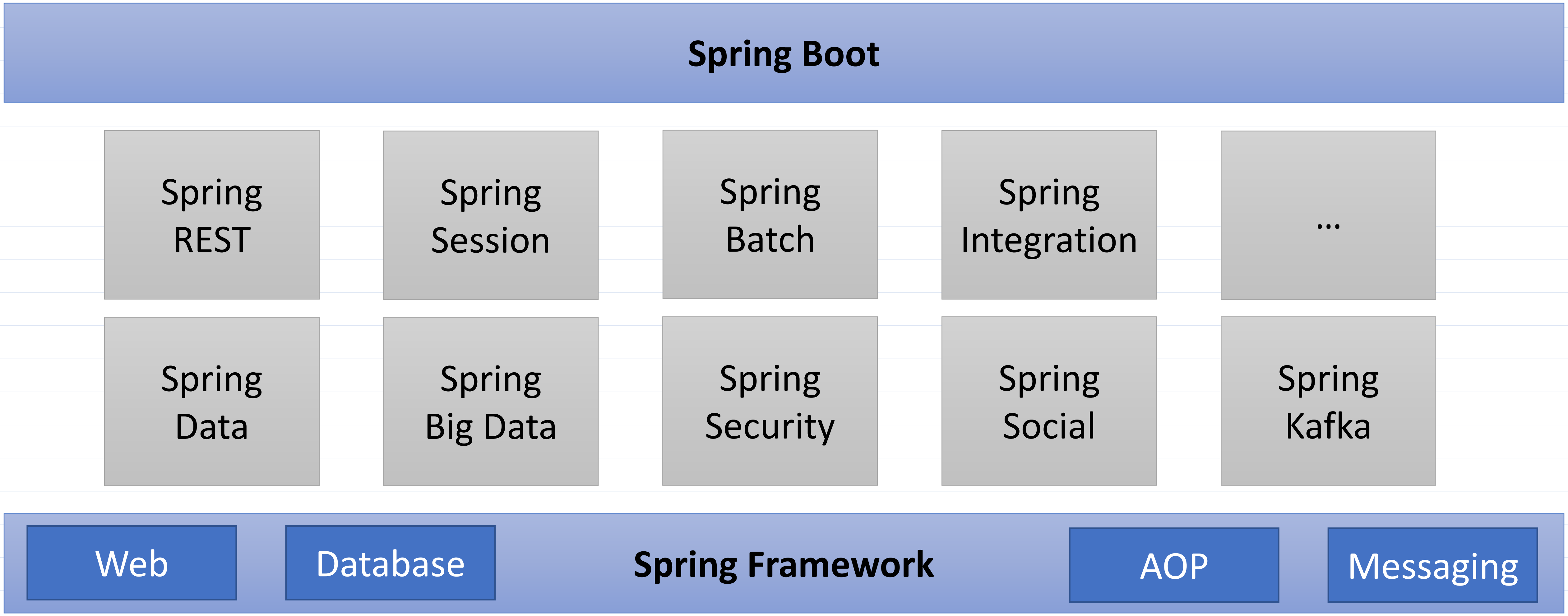
Spring Framework

Spring 5

↳ Project Reactor



Spring Boot



Spring Boot = Spring framework + Embedded web-server + Auto-configuration - XML config - Jar conflicts

SUNBEAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>