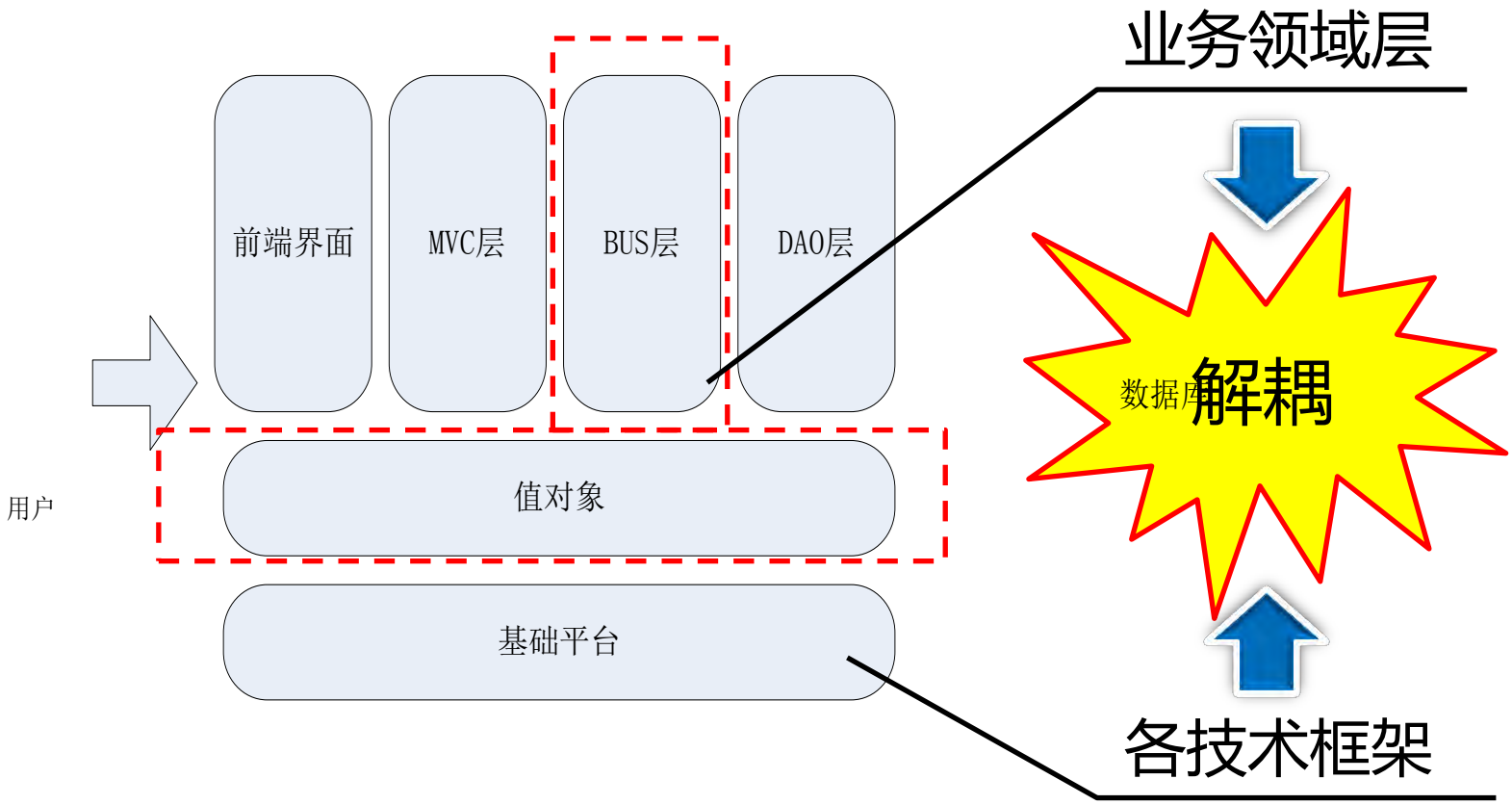
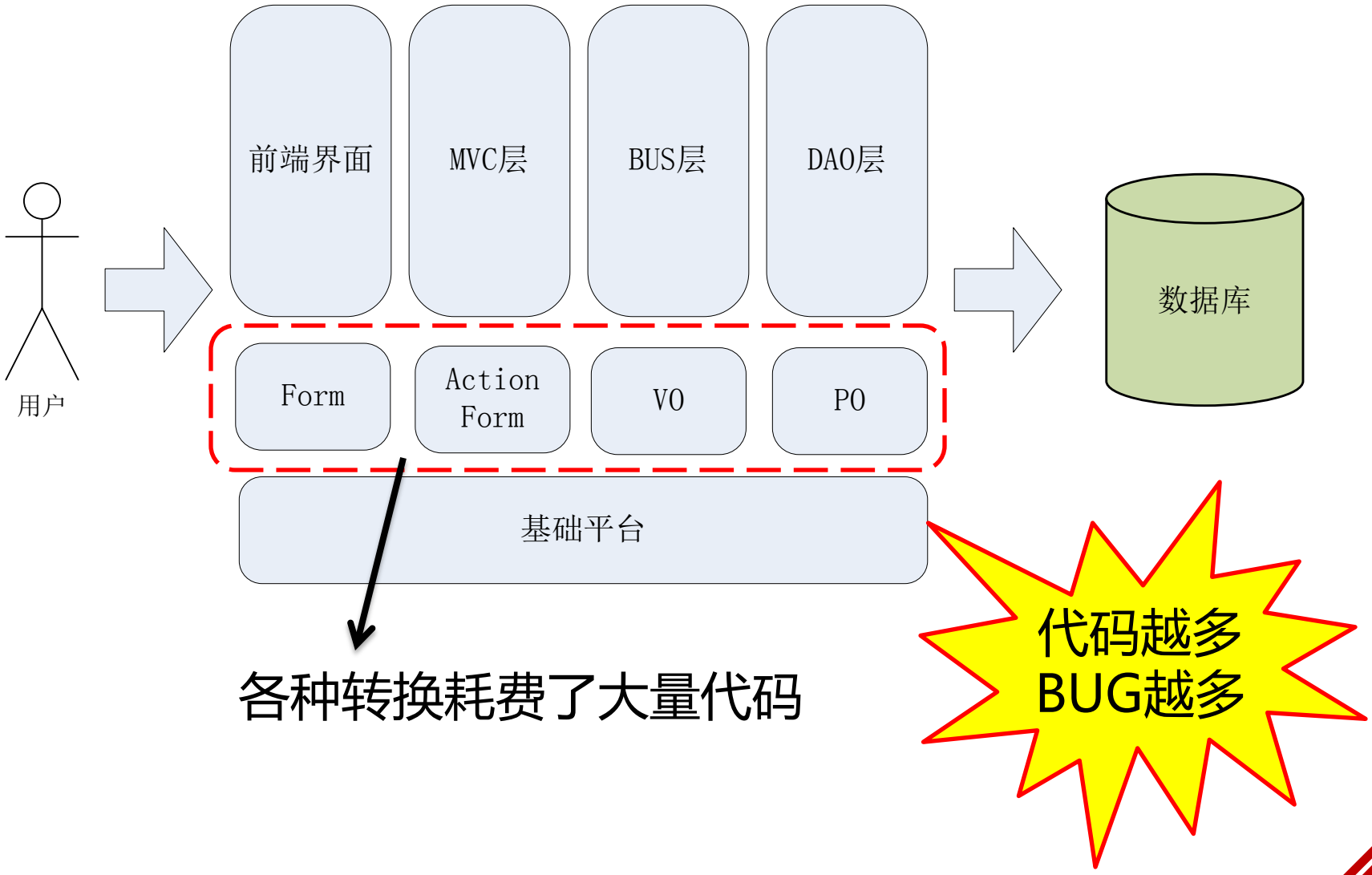
架构演进不是那么简单的事儿

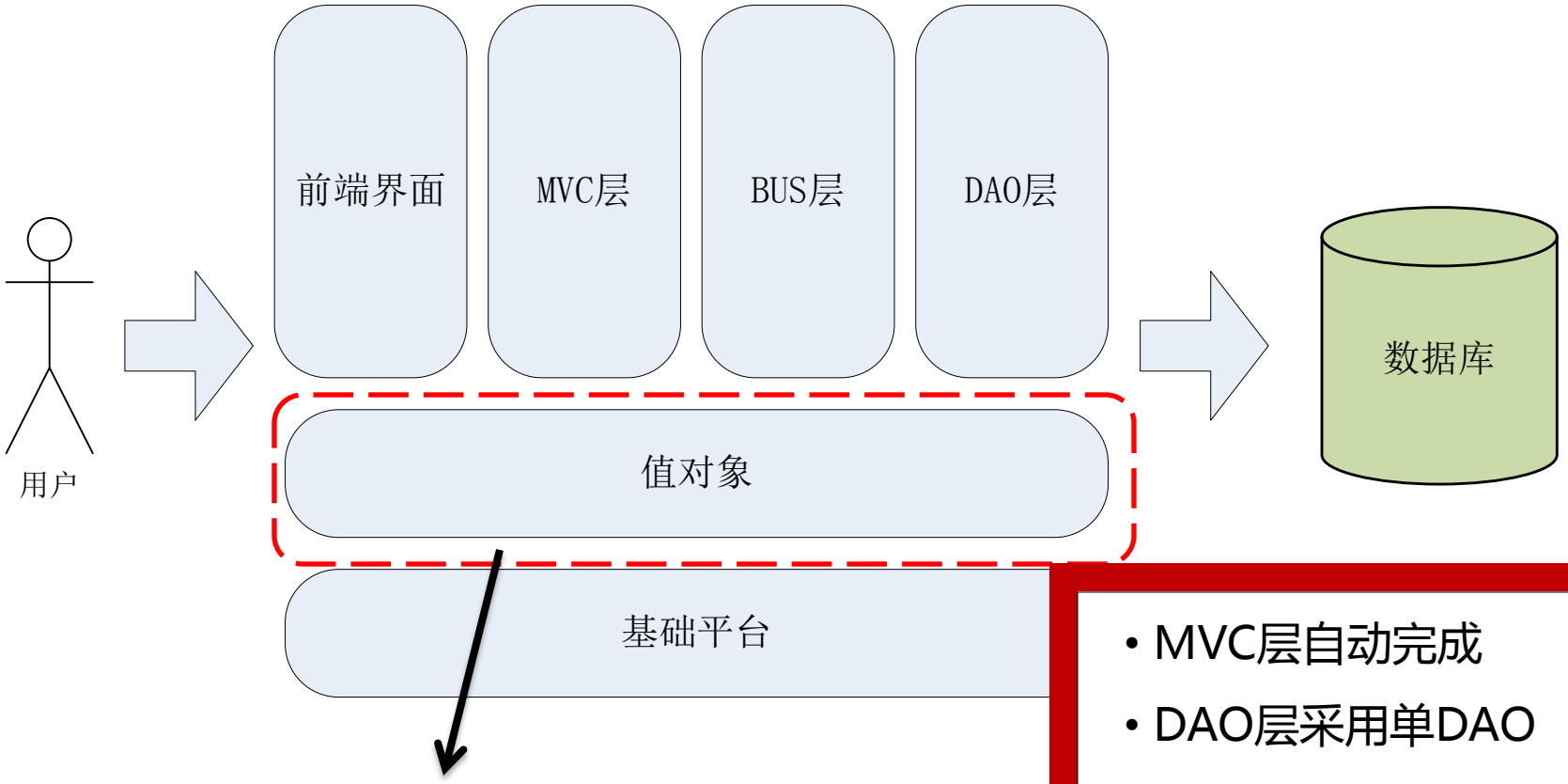
像更换零件一样更新系统



易维护设计：过去的架构设计



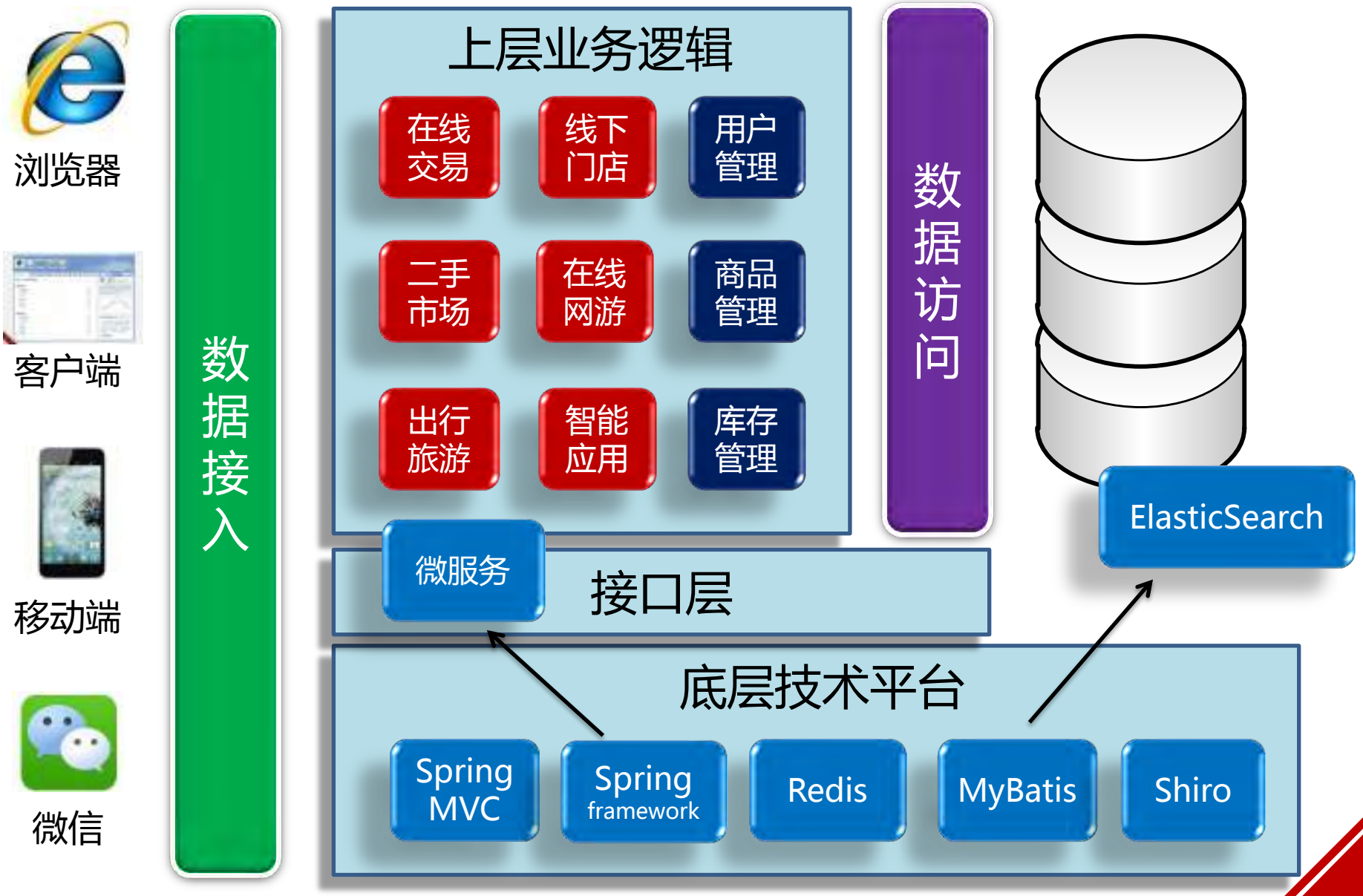
易于维护的架构设计



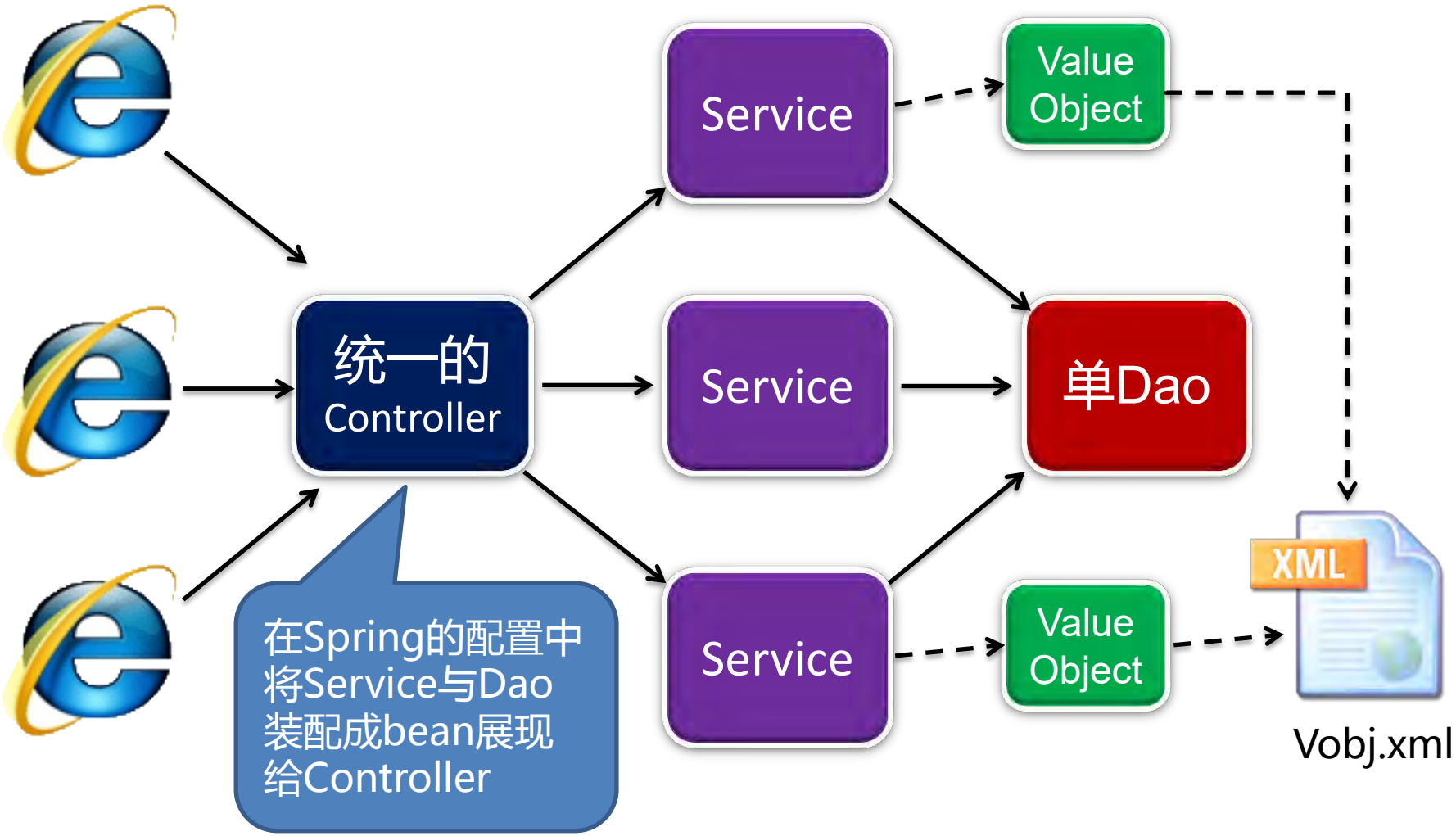
从界面到数据库统一起来
实现自动化的值对象转换

- MVC层自动完成
- DAO层采用单DAO
- 前端界面、BUS层、值对象才是我们真正维护的地方

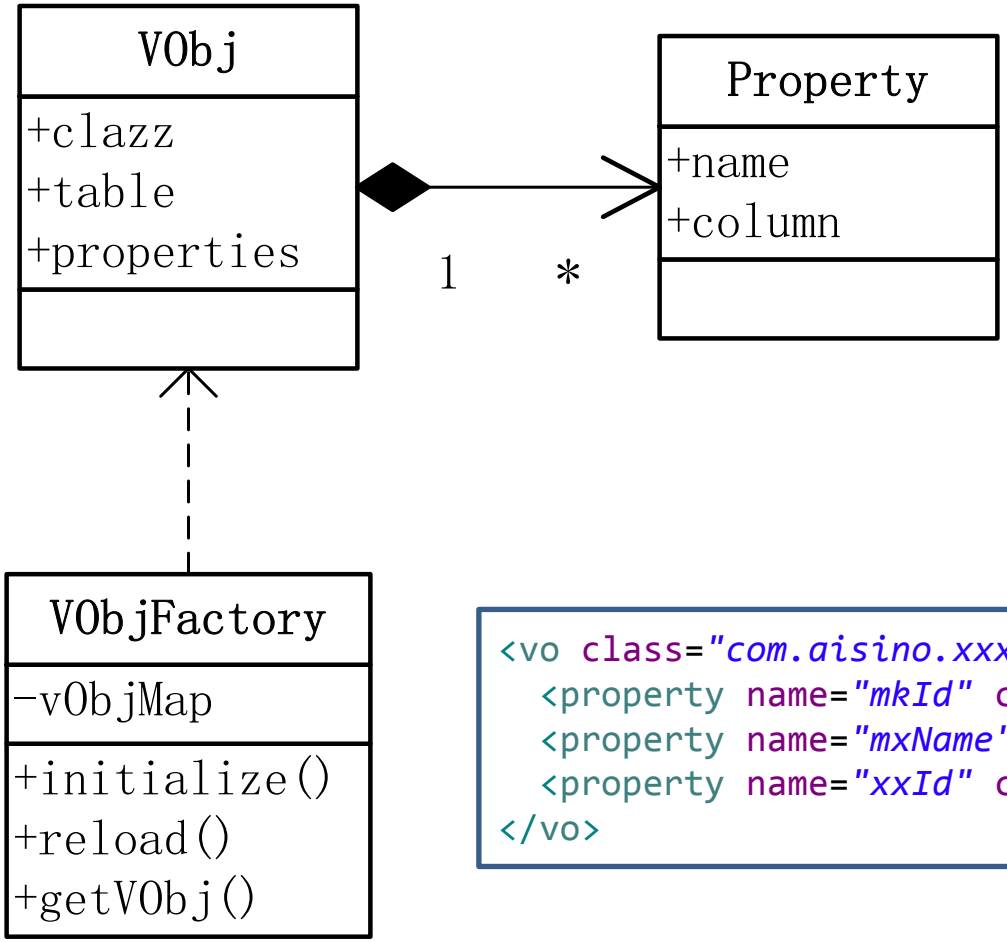
支持领域驱动的架构设计



支持领域的架构：增删改


Service
Value Object
统一的 Controller
Service
单Dao
Service
Value Object
XML
Vobj.xml
在Spring的配置中将Service与Dao装配成bean展现给Controller

值对象与表的映射



Vobj.xml

```
<vo class="com.aisino.xxx" tableName="T1">
  <property name="mkId" column="mk_id" isPmKey="true"/>
  <property name="mxName" column="mx_name" isPmKey="false"/>
  <property name="xxId" column="xx_id" isPmKey="true"/>
</vo>
```

增删改操作的设计实现

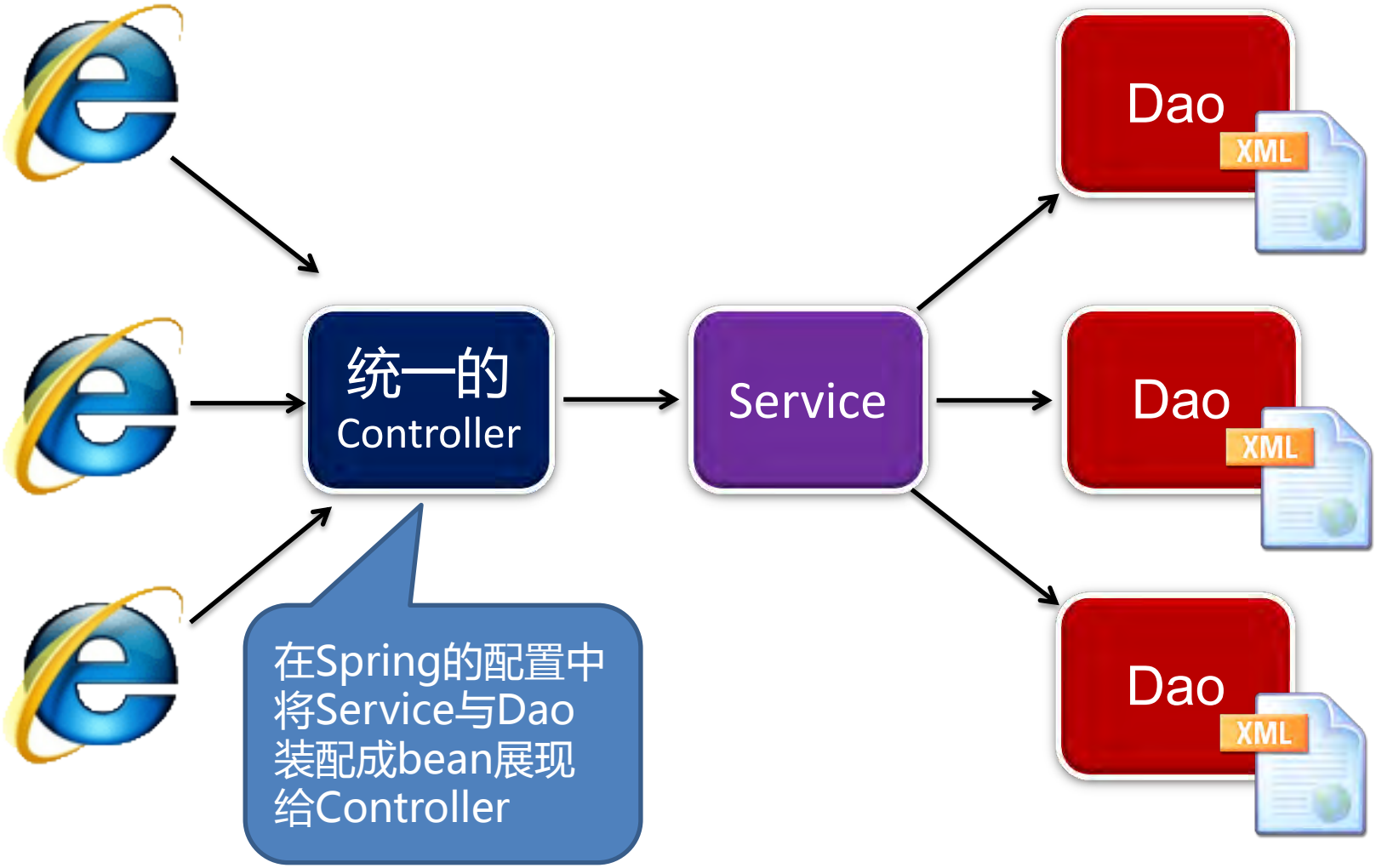
单Controller：

1. 根据前端参数Bean，从Spring中获得Service
2. 根据前端参数method，通过反射获得调用方法
3. 通过反射获得调用方法的第一个参数作为值对象
4. 通过反射创建值对象，根据反射获得值对象的属性，从前端Json中获得对应数据，写入值对象
5. 根据前端参数args数组，获得其它参数
6. 传入参数，使用反射调用该方法

单Dao：

1. 调用VObjFactory.getVObj(class)
2. 根据vObj.getTable()获得表名
3. for(Property prop : vObj.getPreperties()) {
   1) 通过prop.getColumns()获得值对象对应字段
   2) 运用反射从值对象中获得属性对应的值
   3) 形成SQL语句
4. 通过SQL语句执行数据库操作

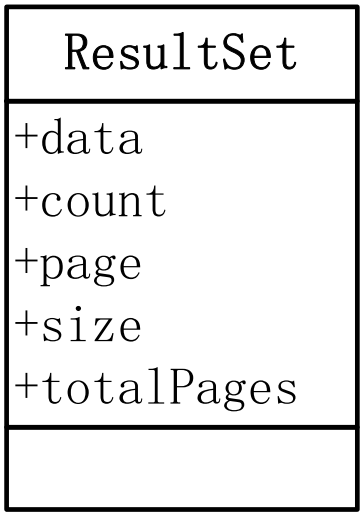
支持领域的架构：查询


统一的
Controller
Service
Dao
XML
Dao
XML
Dao
XML
在Spring的配置中
将Service与Dao
装配成bean展现
给Controller

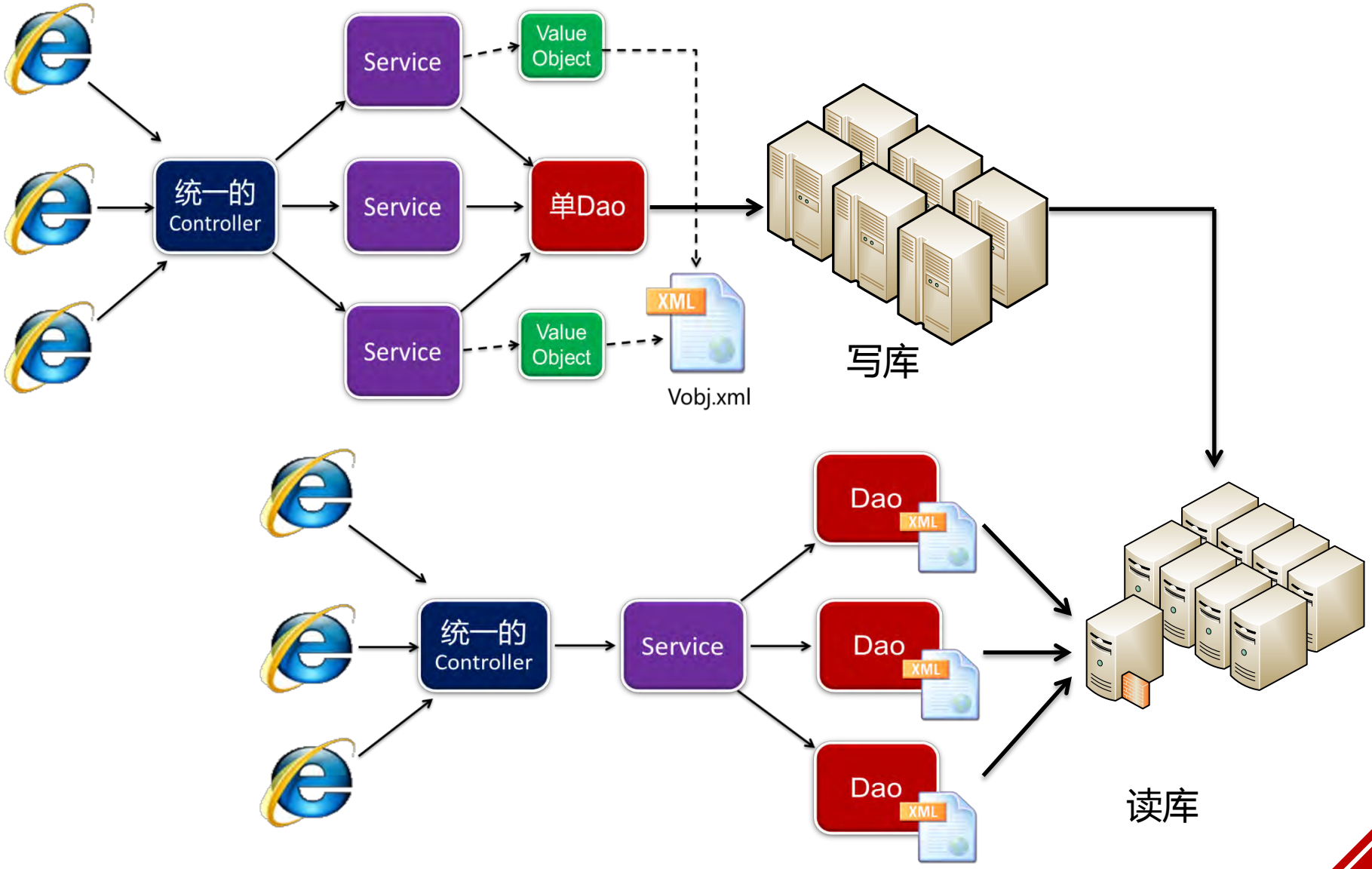
查询功能的设计实现

单Controller：
1. 从前端获得bean、page、size、count等参数
2. 根据bean从Spring中获得serivce
3. 从前端获得查询参数Json，将其转换为Map
4. 执行service.query(map)
5. 以不同形式返回前端（可以在子类中实现）

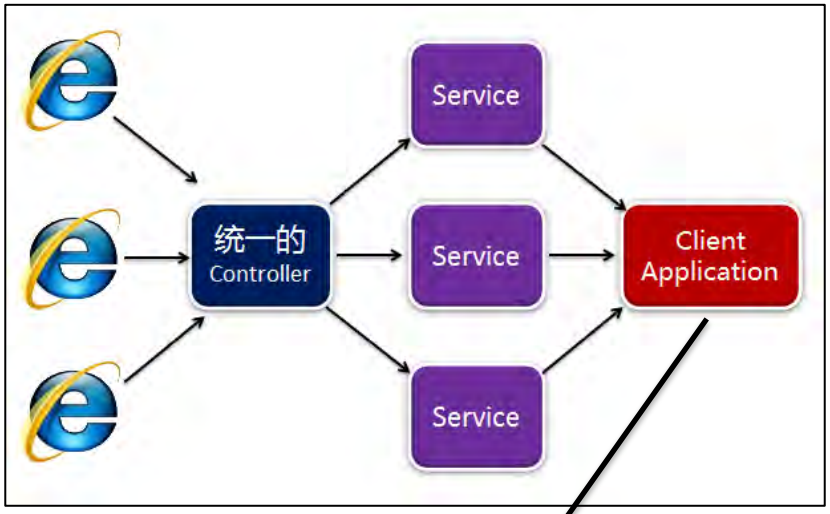
单Service：
1. 将查询参数Map、 page、size传递给Dao，执行查询dao.query()
2. 在查询的前后增加空方法beforeQuery()、afterQuery()作为hook，当某业务需要在查询前后进行处理时，通过重载子类去实现
3. 判断前端是否传递count，如果有则不再求和，否则调用dao.count()求和
4. 计算“第x页，共y页”
5. 将数据打包成ResultSet对象返回



支持读写分离的演化

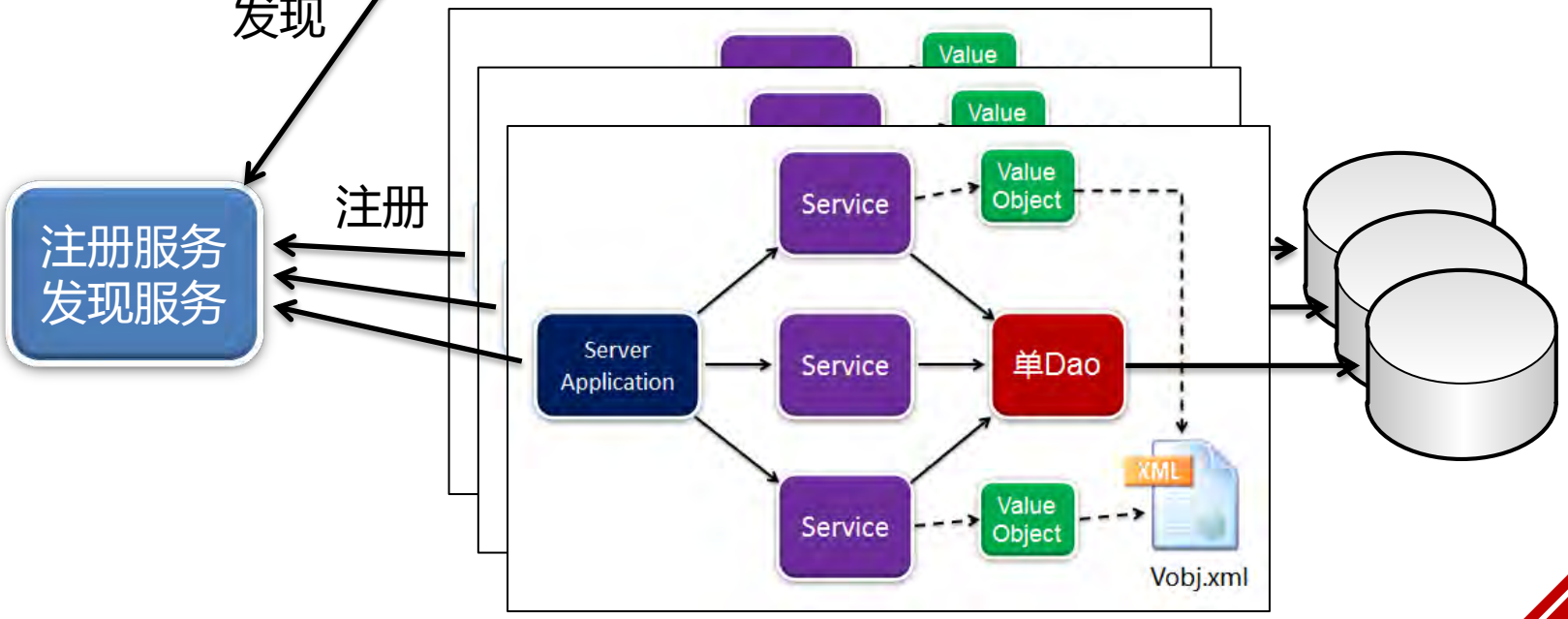

Service
Value Object
统一的 Controller
Service
单Dao
Service
Value Object
XML
Vobj.xml
写库
Dao
XML
统一的 Controller
Service
Dao
XML
Dao
XML
读库

支持微服务的转型

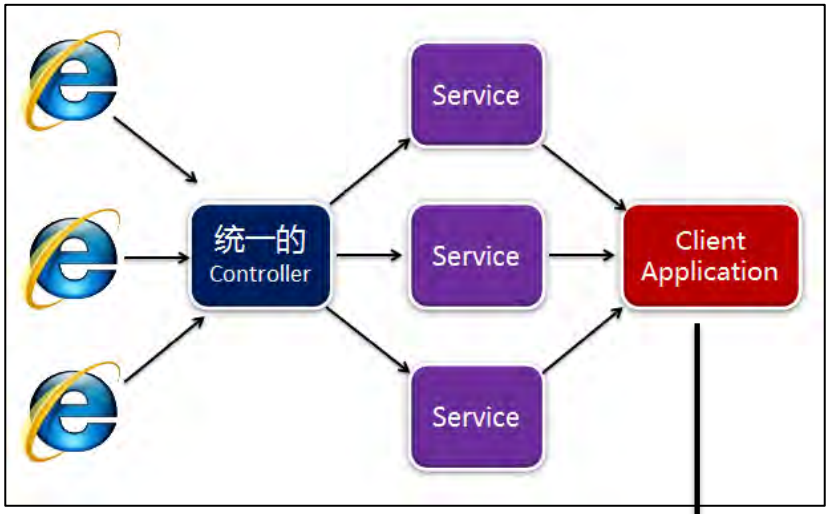


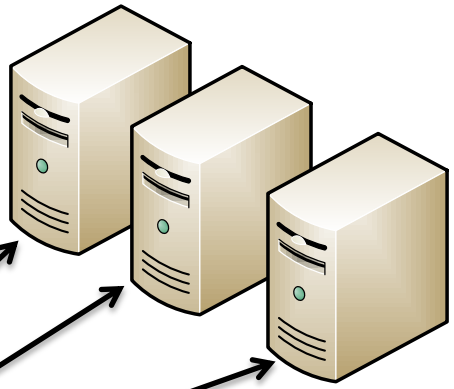
发现

注册

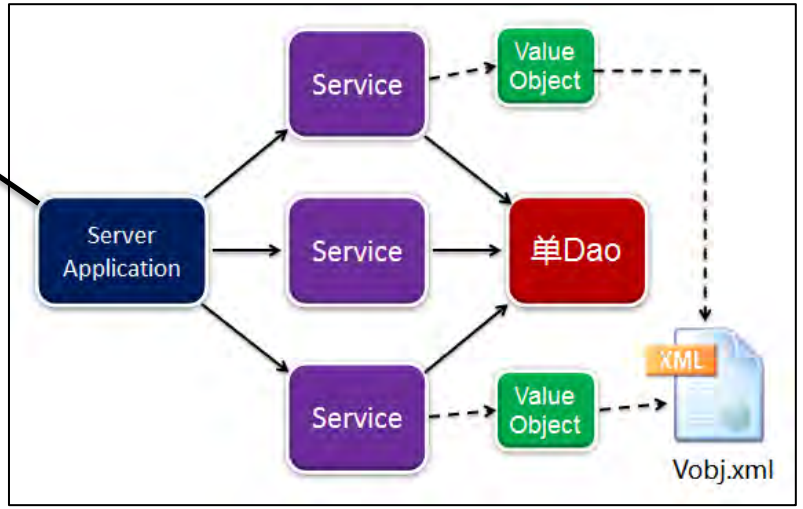


微服务与负载均衡

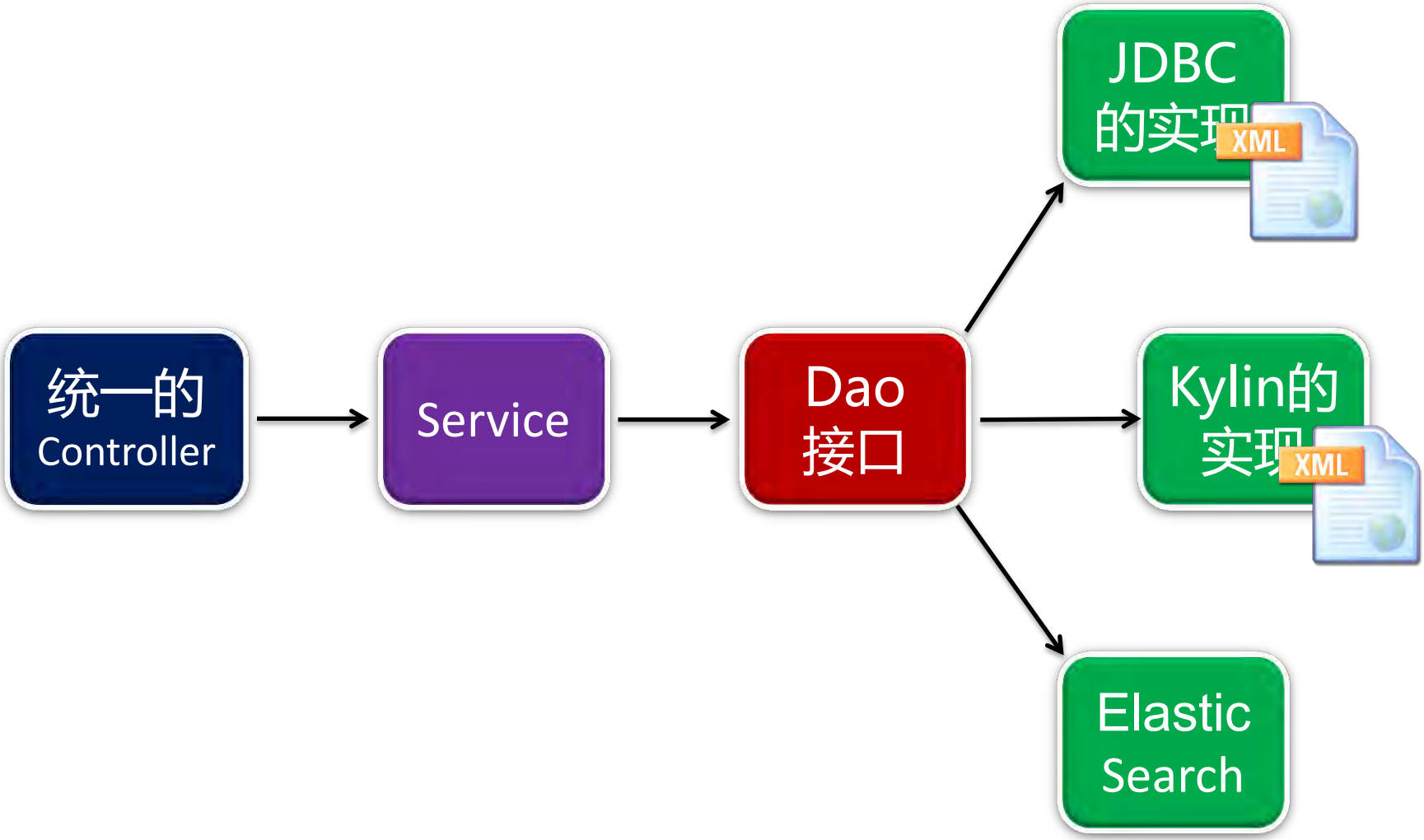




注册服务
发现服务



支持大数据架构转型


统一的
Controller
Service
Dao
接口
JDBC
的实现
XML
Kylin的
实现
XML
Elastic
Search

交流时间