

# Web 三层结构

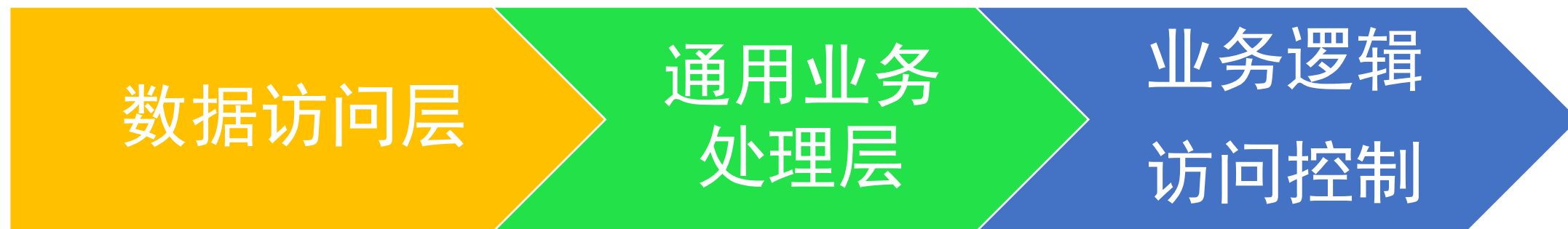
Java Web MVC 黄隆

DAO

Service

Controller

# 垂直层级



# 要点

- 小系统，相对集中式
- 扩展性稍弱，结构简单，实现容易
- 通过分布式Session集群
- 不允许横向调用，只能垂直调用

# DAO层

- DAO层主要是做数据持久层的工作，主要与**数据库**进行交互
- DAO设计的总体规划需要和设计的表，和实现类之间一一对应
- DAO层更多是单一的SQL语句，没有必要进行事务控制，因为事务开销并不便宜
- DAO层的设计首先是设计DAO的接口，和**mybatis mapper.xml** 实现并映射
- DAO层由Service层调用，异常往上抛
- DAO层日志由数据库连接池输出

数据持久化

# Service层

- Service层主要负责业务模块的**逻辑应用**设计
- Service层被Controller层调用，调用DAO层
- Service层进行事务控制是相当必要的，对于多条SQL进行**事务控制**，如果某个SQL执行失败，那么应当对已经执行的SQL语句进行回滚
- Service层没有必要进行try-catch，直接往上抛异常就可以。如果非要做日志处理，进行try-catch最后往上抛异常（**不建议处理异常**）
- Service层封装的业务逻辑有利于业务逻辑的**独立性和重复利用性**，程序显得非常简洁

**事务完整性**

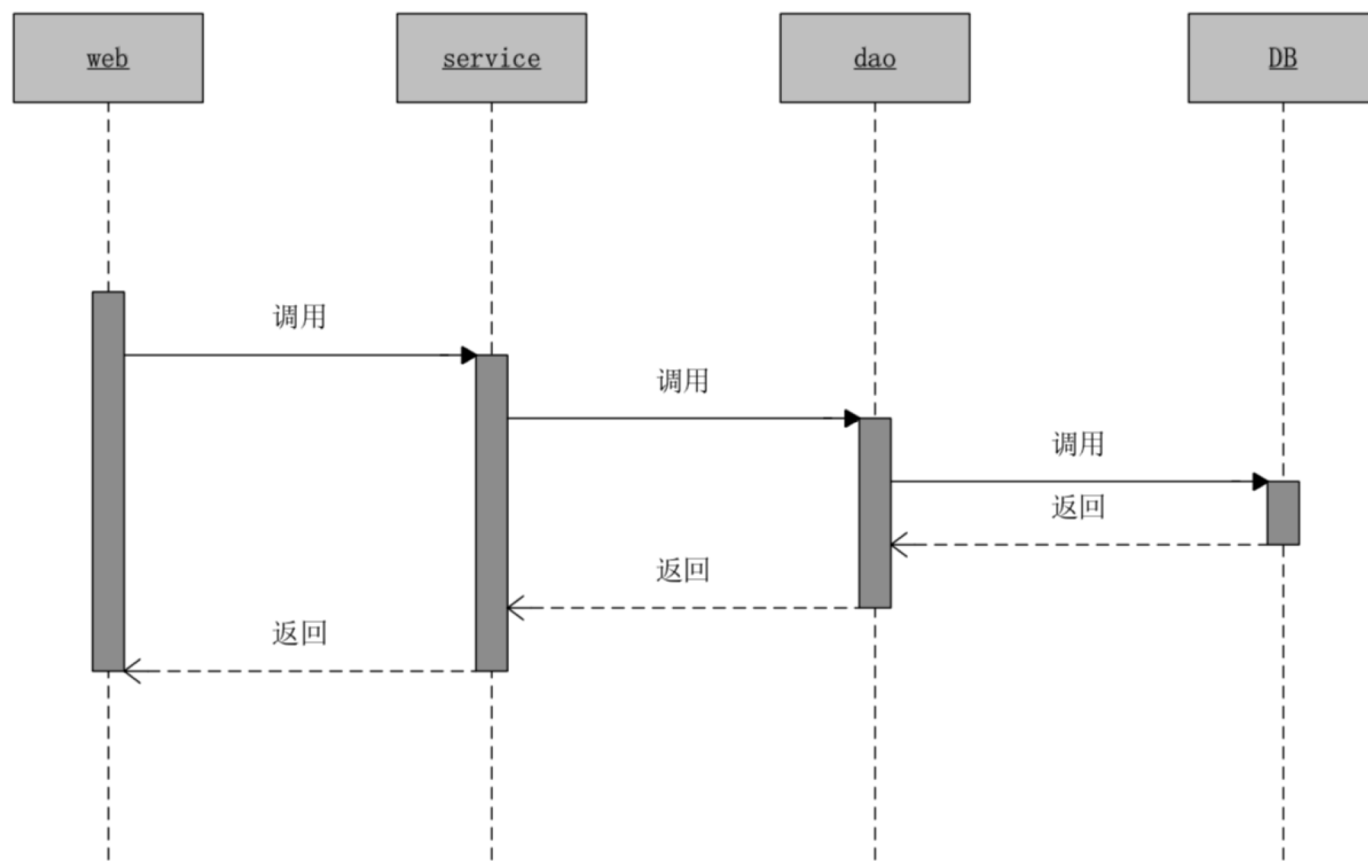
# Controller层

- Controller层的方法对应的是某个URL，控制转发，面向的是应用人员，应该返回应用人员能读懂的信息(VO)
- Controller层必须做异常处理，一般来说会有统一的异常处理方法
- Controller层负责具体的**业务模块流程的控制**
- Controller层主要调用Service层里面的接口控制具体的业务流程
- Controller层**处理日志，参数校验**，调用**外部请求**等业务流程处理，不建议在Service层处理异常
- Controller层提供开放的能力

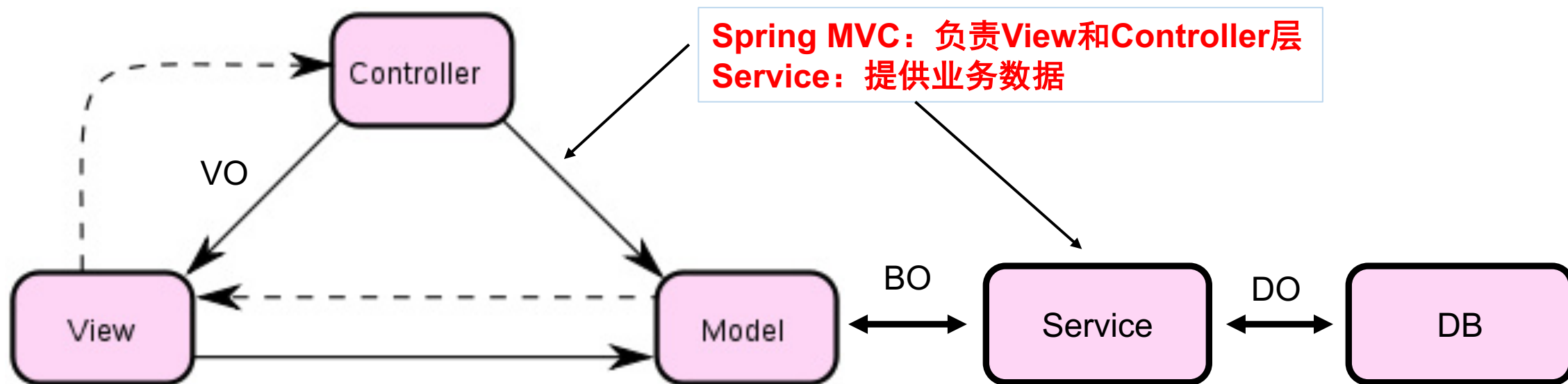
柔和具体业务流程和视图管理  
**最复杂的一层**

演变成四层结构  
简化Controller

# 调用链



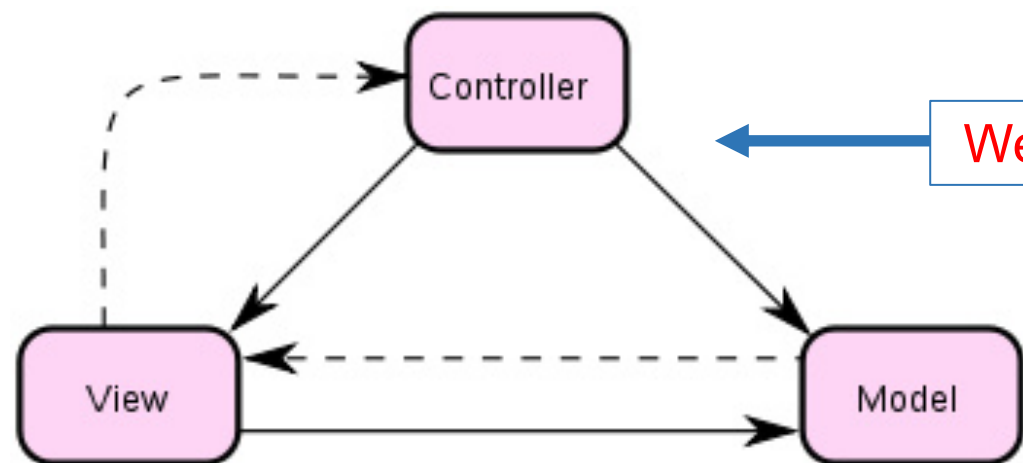
# 经典MVC





# 经典 Java Web框架

View层和Controller层结合比较紧密，需要二者结合起来协同工发。  
View层主要负责前台jsp/html页面的表示



Web MVC

Model是由Service提供的BO

