

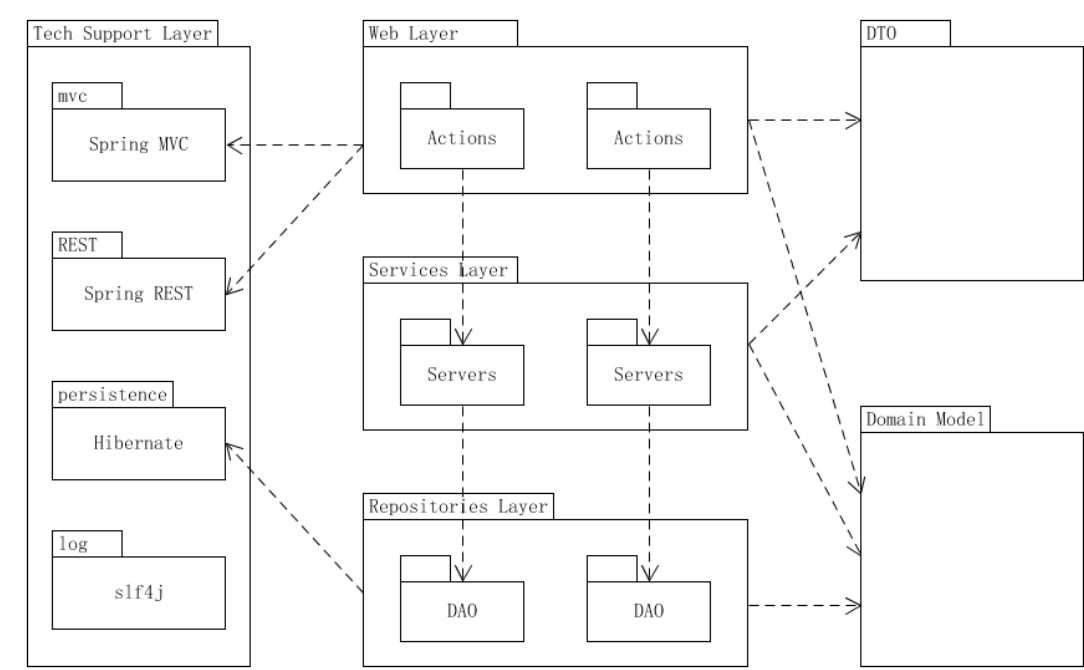
AwesomeTickets

Architecture Documentation

功能与非功能性需求表格：

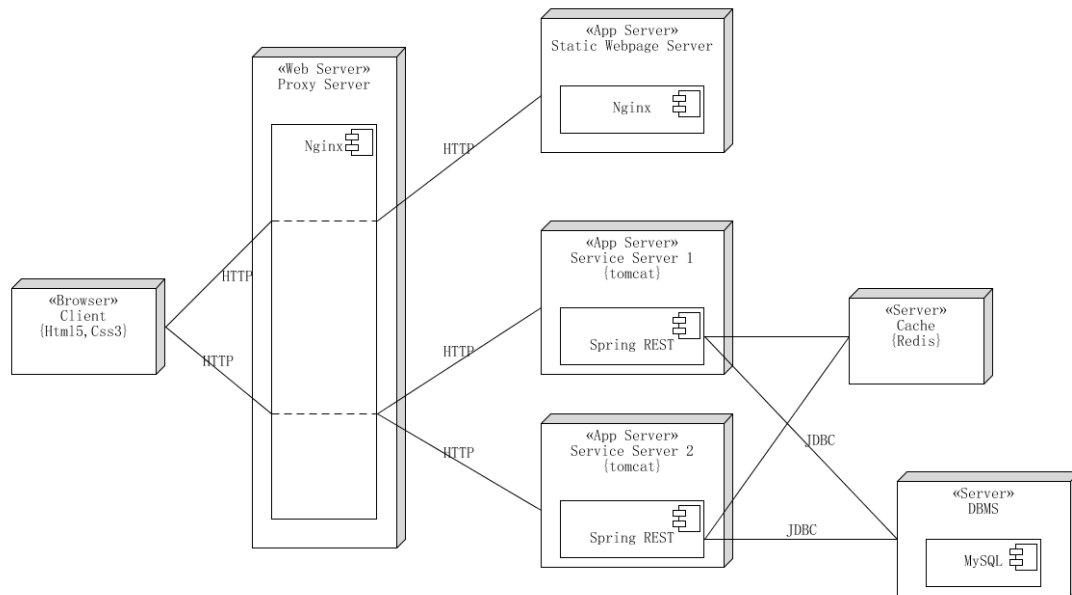
| 非功能需求 | | | 功能需求 |
|---|-------------|-------------------------|--|
| 约束 | 运行期质量属性 | 开发期质量属性 | |
| 1 台云服务器（双核 CPU，内存 2 G，磁盘 20G） 服务端必须能在Linux上运行 购票系统可以在主流浏览器上正确呈现 | 高性能 可伸缩性 | 易理解性 易测试性 模块间松散耦合 | 注册与登录 查询购票历史 选择电影 选择影院 选择日期与时间 选择座位 购买座位 取票 |

RUP 架构的逻辑模型：



考虑到开发期质量属性中有易理解性、易测试性和模块间松散耦合的要求，我们使用已经发展较为成熟的、层次鲜明的Spring MVC框架，持久层采用Hibernate框架。同时，考虑到功能需求中有大量对资源的访问，比如选择电影、影院和座位等，我们引入了Spring REST 框架。

RUP 架构的部署模型：



我们的开发模式是前后端彻底分离，所有的html、css和js等资源放在一个静态网页服务器中，网页展示的数据通过REST接口来获得。分布式后端架构可以满足高性能、可伸缩性的运行期质量需求。考虑到云服务器只有1台，因此上述模型中的节点只是逻辑上的分离，实际将在同一台机器上实现（使用docker）。