<项目名称>

软件架构文档

版本 <2.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <日/月/年> | <x.x> | <详细信息> | <姓名> |
| 01/07/2019 | 1.0 | 架构初步设计 | 于亚杰，王祖来 |
| 10/07/2019 | 1.1 | 第一次迭代架构文档的完善 | 于亚杰，王祖来 |
| 02/08/2019 | 2.0 | 第三次迭代架构文档的完善 | 于亚杰，王祖来 |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 用例视图 4

3. 逻辑视图 4

3.1 概述 4

3.2 在构架方面具有重要意义的设计包 4

4. 进程视图 4

5. 部署视图 4

6. 实现视图 4

7. 数据视图（可选） 4

8. 核心算法设计（可选） 4

软件架构文档 （简化版）

# 简介

## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。文档面向软件开发人员和评估方，通过各视图了解和改进软件的架构设计。

## 参考资料

《软件工程原理》（作者：沈备军，陈昊鹏，陈雨亭 出版社：高等教育出版社）

# 用例视图

如图所示的用例中，查看条目实现在条目服务中，条目信息用MySQL存储，评论作品实现在评论服务中，非结构化数据用MongoDB存储，服务间分离实现了高可用高性能的架构模式，数据库的分布式存储可维护性更强，体现了微服务架构的优势。此外，发送私信放在私信服务中，添加好友放在好友服务中，查看好友评论与评分的实现为好友服务中调用评论服务、评分服务

# 逻辑视图

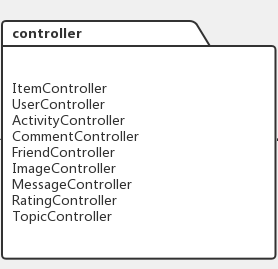
## 概述

软件主要采用B/S架构，MVC架构和微服务架构。应用层分安卓和web，采用React Native进行前端显示。nginx获取并转发RESTFUL请求给控制层，控制层调用接口，将请求转发给网关服务进而调用所需的微服务。网关可起到拦截和分散请求、负载均衡的作用，微服务中比较重要的认证服务可以实现对身份的识别和权限的分配，采用rabbitmq作为消息总线并利用到了websocket协议，用prometheus配上grafana、zipkin、kibana进行监控，logstash和rabbitmq进行日志管理。数据层进行读写分离OR映射等操作，数据存储层由MySQL、MongoDB、Elasticsearch、Redis数据库构成，再测试阶段利用了内存数据库，并在持久层利用Hibernate等工具实现数据持久化。

## 在构架方面具有重要意义的设计包

* + 1. Controller

Controller是MVC架构的核心部分，用于接收转发来的请求，并调用服务类来处理相应的业务逻辑。



* + 1. Repository

采用Hibernate来持久化数据，与MySQL等数据库进行连接，实现CrudRepository接口对数据进行操作。



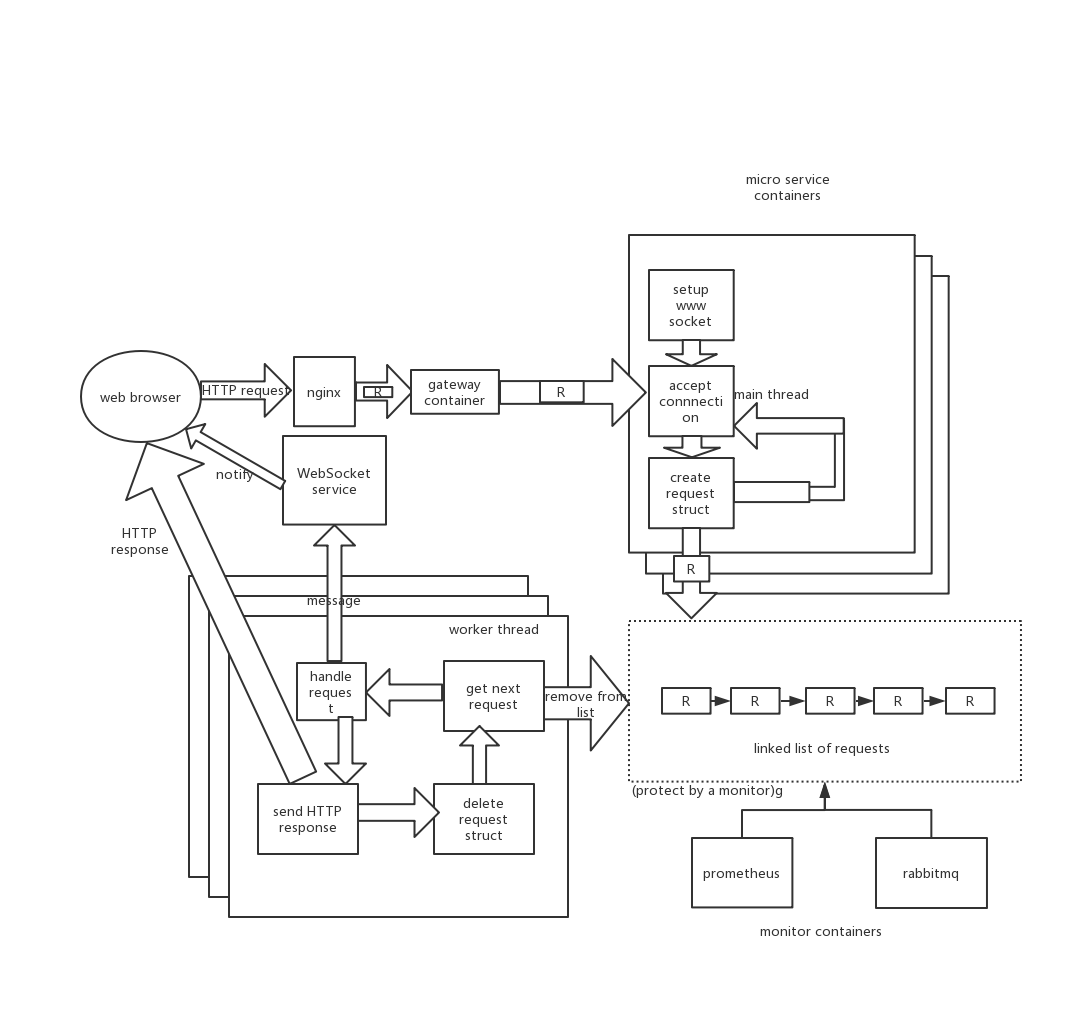
3.2.3 Dao

Dao层实现对MySQL、MongoDB的读写分离。



# 进程视图

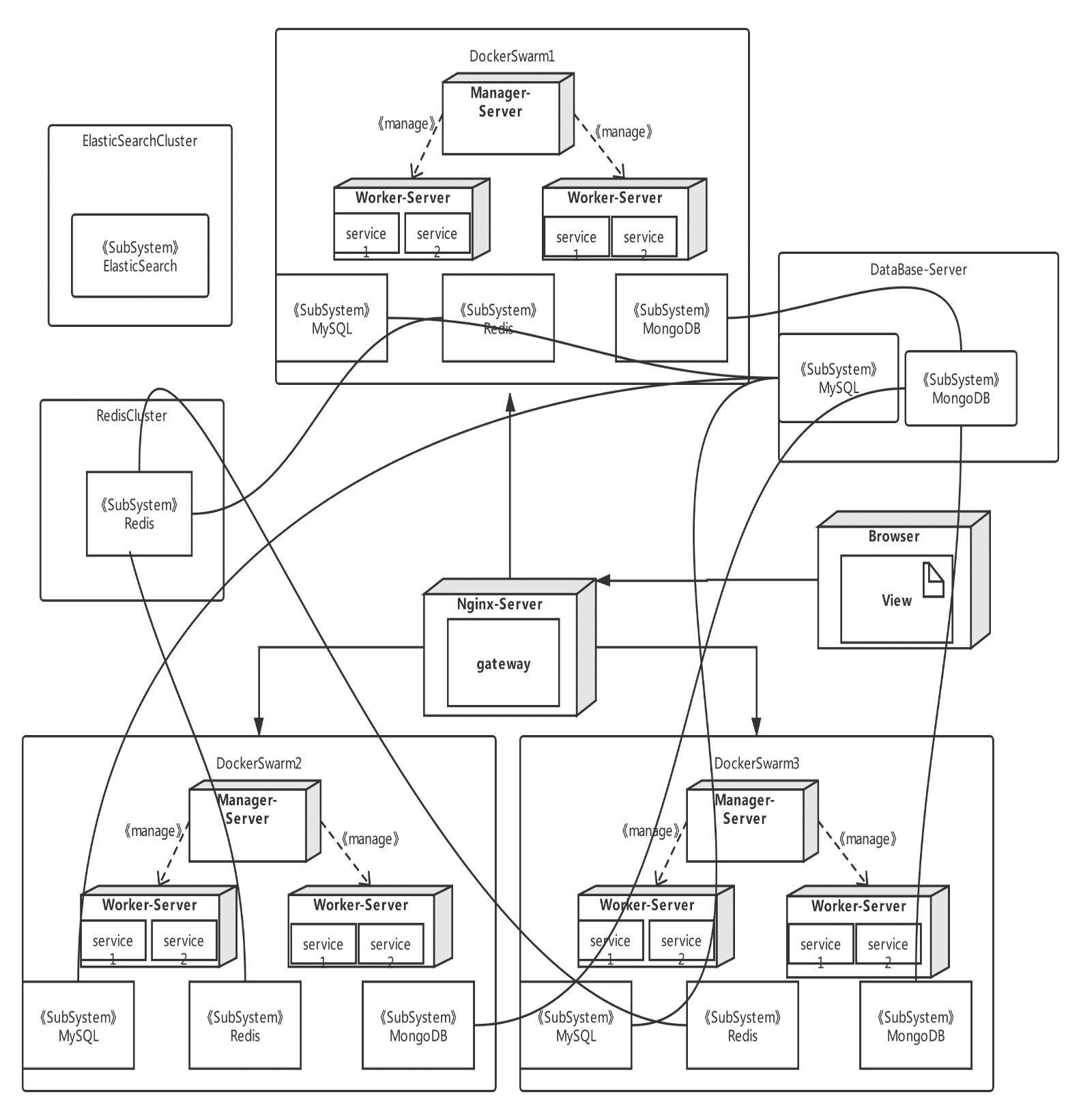
前端请求发送到主服务器网关，主服务器通过Nginx使用负载均衡策略将请求转发到三台服务器的gateway容器中，gateway将请求转发到Docker Swarm中对应的微服务容器。容器中的Java web进程进行多线程请求的处理，将Response发送给客户端，或者将消息发送到由WebSocket实现的消息中间件中，对客户端进行通知。运行Prometheus与rabbitmq容器监控请求，并记录日志。



# 部署视图

服务部署在多个服务器上，共有三个Docker Swarm集群，由主服务器网关来调用，每个集群由manager和worker构成，集群内服务分摊到各个服务器上实现高可用的服务部署。

分布式存储数据库，并进行数据库集群的管理，对MongoDB，MySQL实现主从数据库读写分离。



# 实现视图

各微服务中包的调用关系如下。

