交我唱APP

软件架构视图说明

版本 1.5

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 5/5/2021 | 1.5 | 更新内容 | 鲍辰 |
| 12/4/2021 | 1.0 | 更新内容 | 鲍辰 |
|  |  |  |  |
|  |  |  |  |

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 5

3.1 概述 5

4. 部署视图 5

5. 实现视图 6

5.1 粗粒度构件图 6

5.2 细粒度构件图 6

6. 技术视图 6

7. 核心算法设计 7

7.1 音频数据流 7

7.2 效果器 7

7.3 基于数据增强的自监督打分模块 7

8. 质量属性的设计 8

软件架构文档

# 简介

## 目的

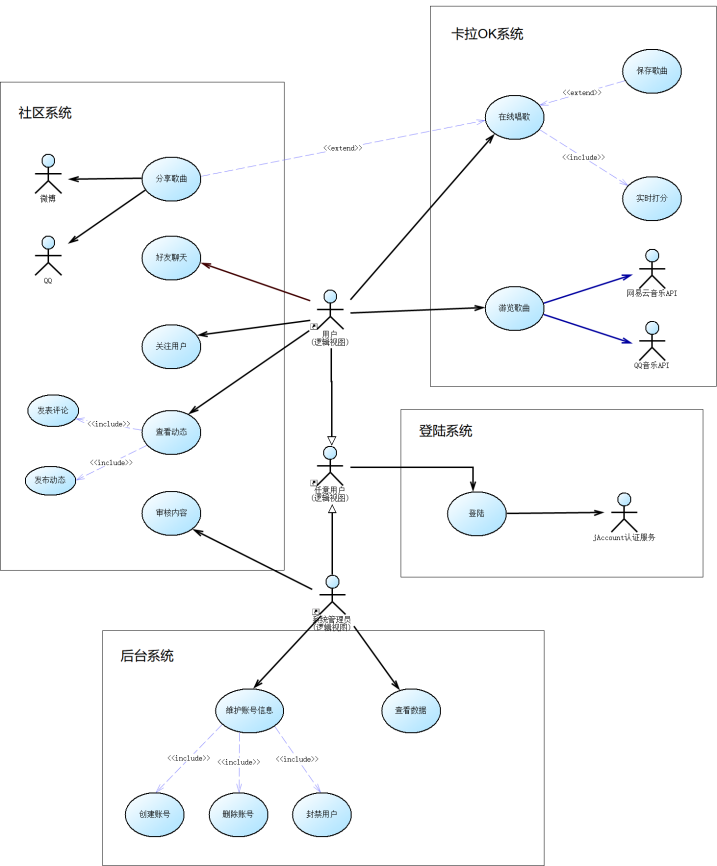
本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

## 参考资料

可用的参考资料如下:

1. 全民K歌，2021，腾讯。
2. 唱吧，2021，酷智科技。
3. 酷狗唱唱，2021，酷狗。
4. 唱鸭，2021，破壁者科技。

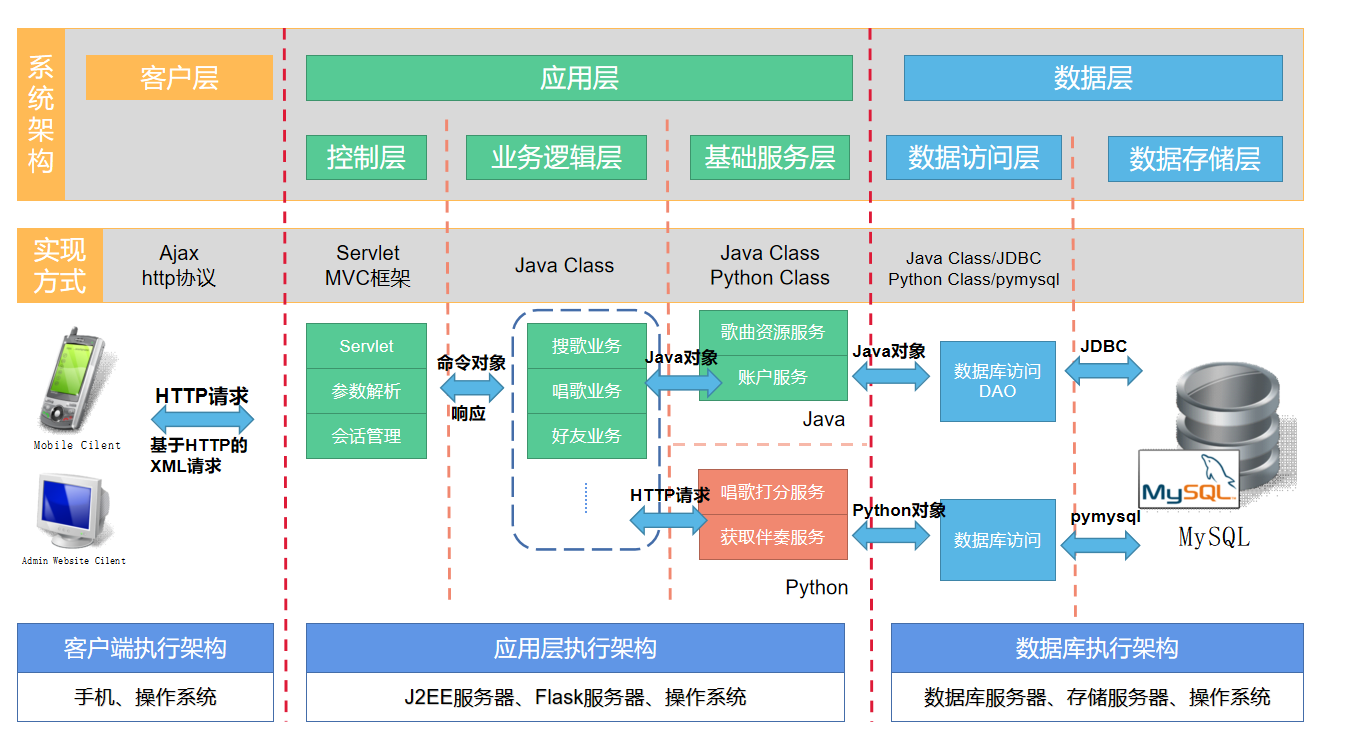
# 用例视图



交我唱APP的用例和用例实现，具体详见作业6.

# 逻辑视图

## 概述



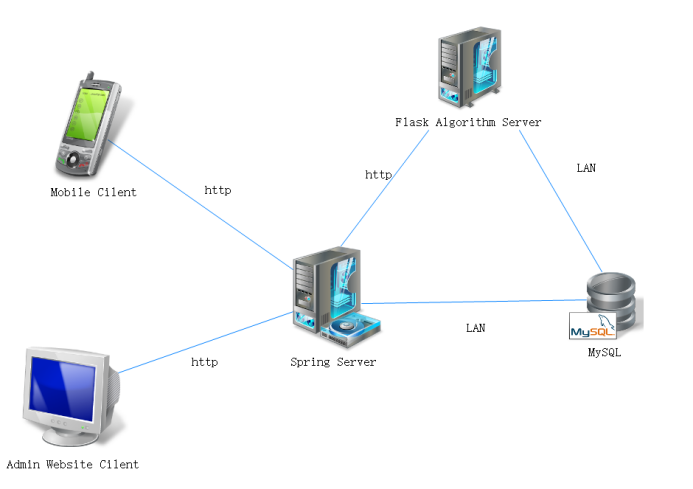
交我唱APP在架构设计时，采用了三层架构（3 Tires Application），即为了“高内聚，低耦合”的思想进行架构设计。

首先是客户层，客户层为了两类客户端，一个是用户所使用的安卓APP客户端，另一个是管理员使用的Web网站管理客户端。客户层通过http请求和基于http的XML请求与服务器（应用层）进行通讯。其间所设计的负载均衡层，由于概念还不清楚，故暂时未画出。

应用控制层接收到http请求后，通过Servlet构件路由到对应的业务组件中。对于Java原生实现的服务逻辑，业务逻辑层调用基础服务层的服务直接进行处理。对于需要Python辅助实现的算法服务，使用http请求与Flask服务器进行请求。

在数据层中，无论是Java原生实现的服务，还是Flask实现的Python服务，会通过数据访问层，通过对应的接口，访问到数据库对象，对数据增删改查。

# 部署视图

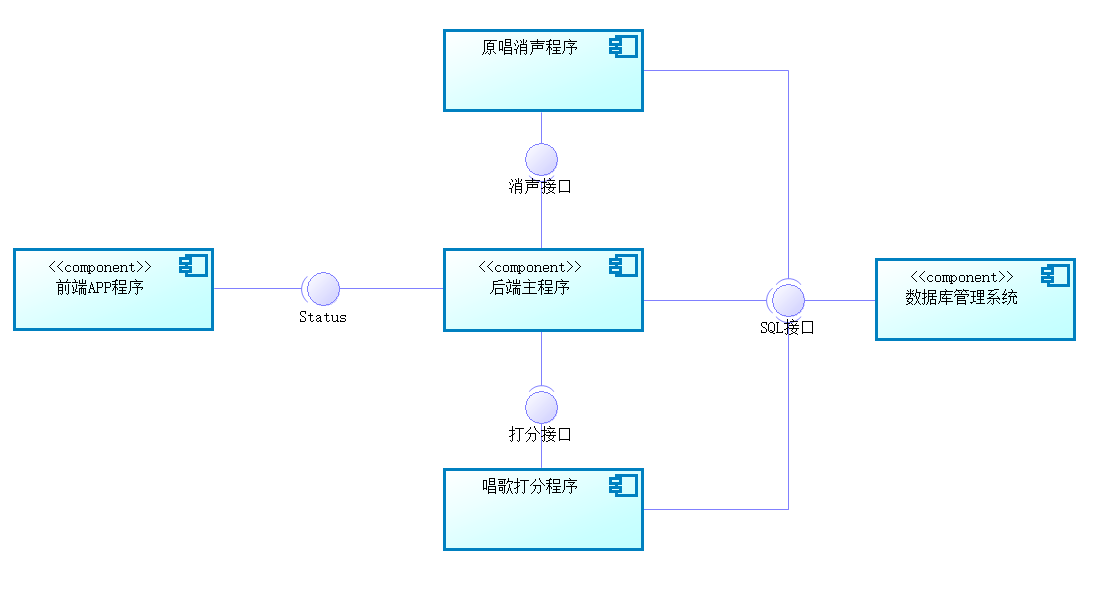


交我唱APP的部署视图主要可以分为五个节点，两个客户端（一个用户客户端和一个管理员客户端），两个服务器（一个基于Spring框架，一个基于Flask框架）和一个MySQL数据库。

客户端和服务器、以及服务器和服务器之间，都使用互联网的http协议进行通信。两个服务器和数据库之间，使用局域网进行通信。

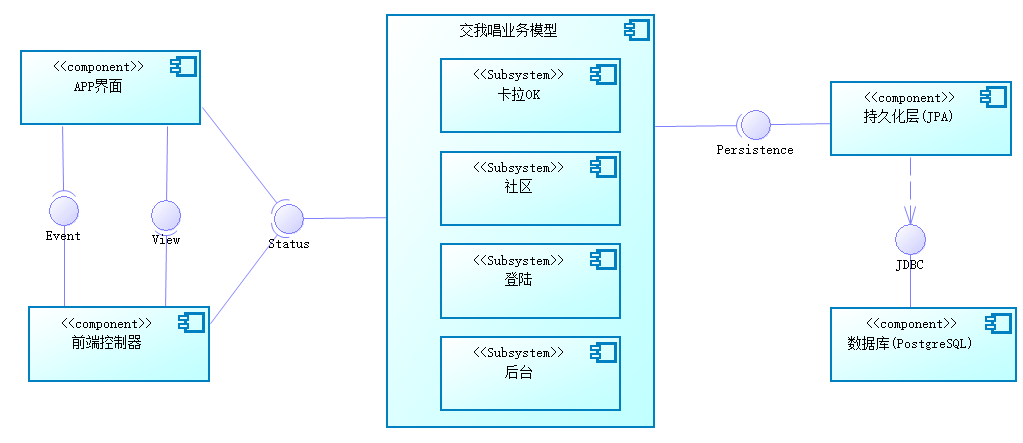
# 实现视图

## 粗粒度构件图



本粗粒度构件图以安装包为单位，刻画了构件内容和数据流。

## 细粒度构件图



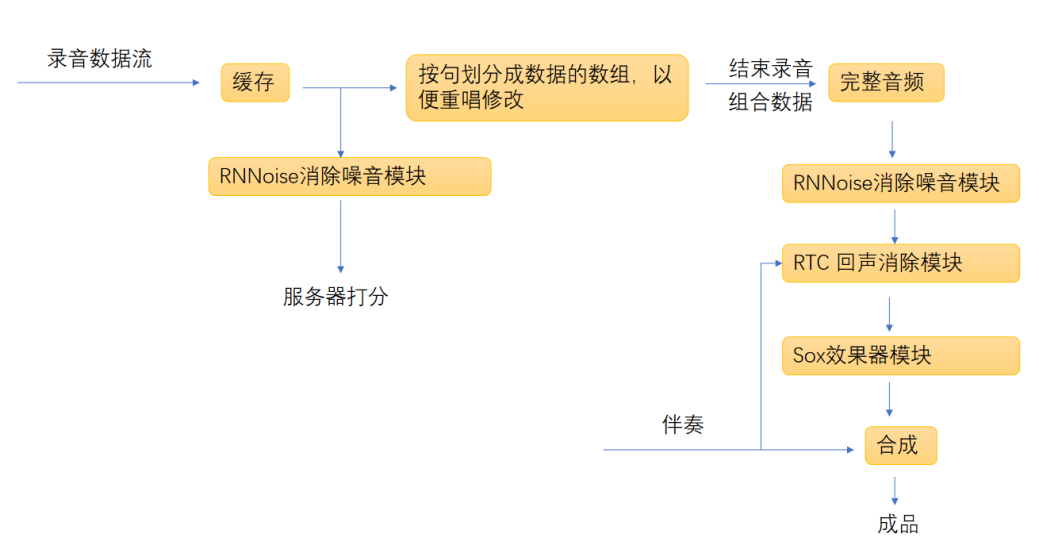
本细粒度构件图较为细致地刻画了各构建和数据流。

# 技术视图

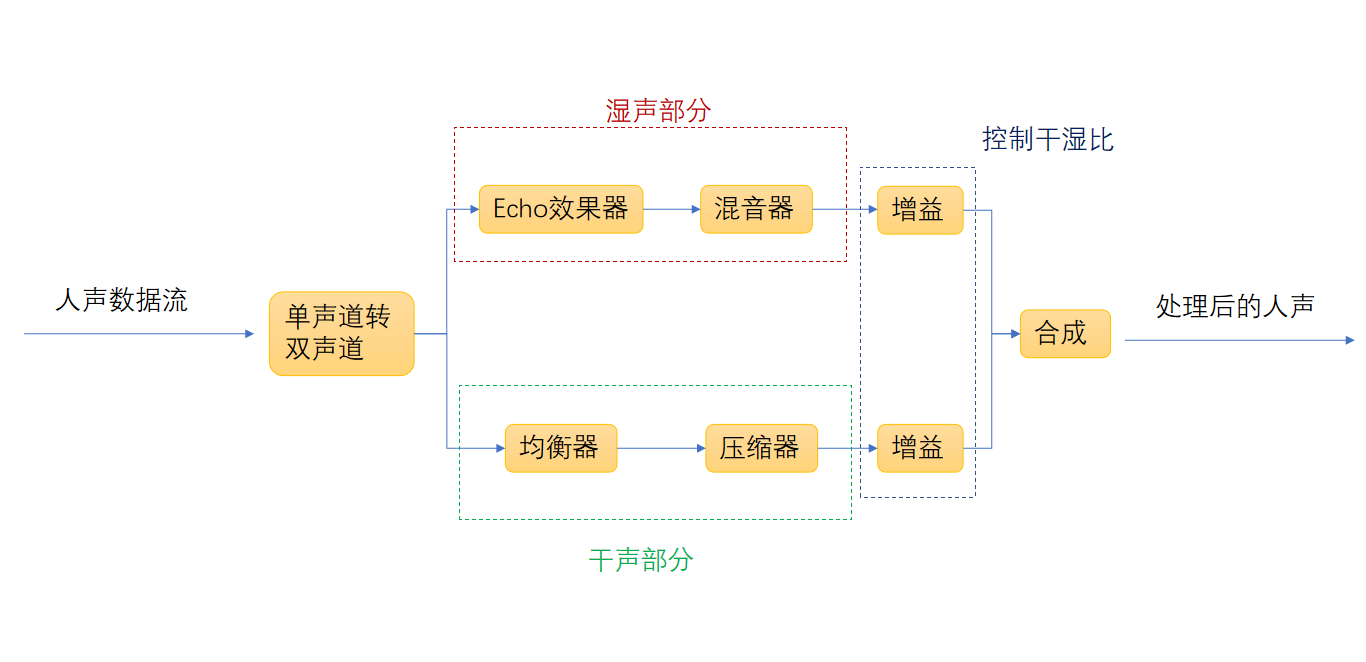
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 前端 | 后端 | 数据库 | 算法 |
| 编程语言 | JS，Java，C | Java，Python | SQL | C,Python |
| 操作系统 | Android OS | CentOS7 | CentOS7 | CentOS7 |
| 框架 | React Native | Spring MVC  Flask | Spring Data JPA  MySQL | Pytorch |

# 核心算法设计

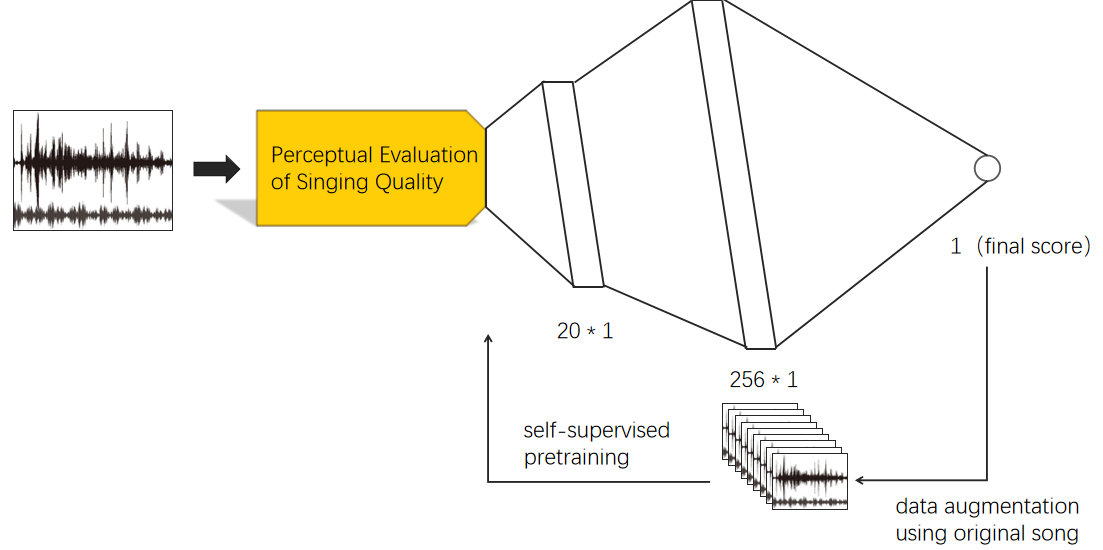
## 音频数据流



## 效果器



## 基于数据增强的自监督打分模块



# 质量属性的设计

后端的原唱消声和唱歌打分算法使用了Flask进行封装，这样面对服务的封装允许我们进一步迁移到分布式服务器和微服务架构上，提升了系统的可移植性和可扩展性。

逻辑视图中选用了三层架构，这样允许开发人员可以只关注整个结构中的其中某一层；可以很容易的用新的实现来替换原有层次的实现；可以降低层与层之间的依赖；有利于标准化；利于各层逻辑的复用，这些优点都提升了软件的可维护性。