实时AI 媒体服务器

1 系统架构

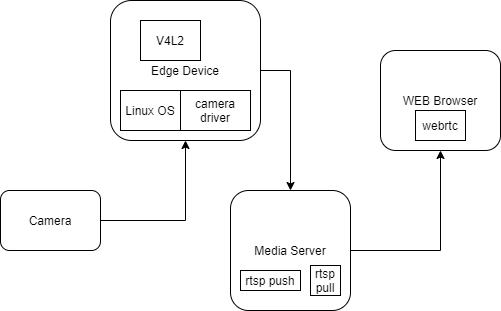
系统主要由四部分组成：

Camera 带有缓存，具备4K实时能力；

Edge 设备，有linux, camera driver, v2l2 编码功能

Media server 开源推流媒体服务器。具备组播和广播功能。

Webrtc WEB系统，基于webrtc的拉流功能。



**2 ．边缘系统构建**

2.1 硬件选型

2.2 Linux 5.4 移植

2.3 Linux Camera 等设备驱动

2.4 V2L2 库移植

2.5 推流任务

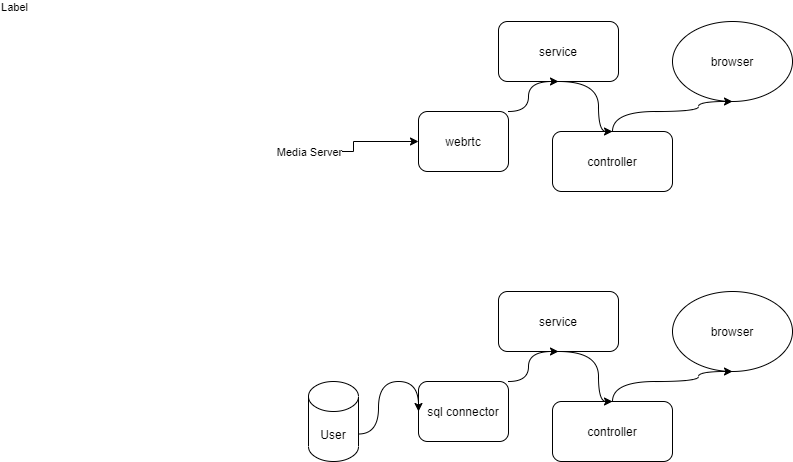
2.6 实时性能测试

**3 媒体系统构建**

1. 云服务器安装
2. 搭建媒体服务器
3. 推流TEST
4. 拉流TEST
5. 实时性测试

**4 应用系统构建**

4.1 应用系统设计



4.2 数据库设计

User

Id, name, email, phone, address, role

Device

Id, edge, type, Startat, status

Edge

Id, mac, ip, user, Start, Status

MediaInfo

Id, device, user,start, end, format, status

4.3 rest接口设计

/api/v1/device

/api/v1/edge

/api/v1/media

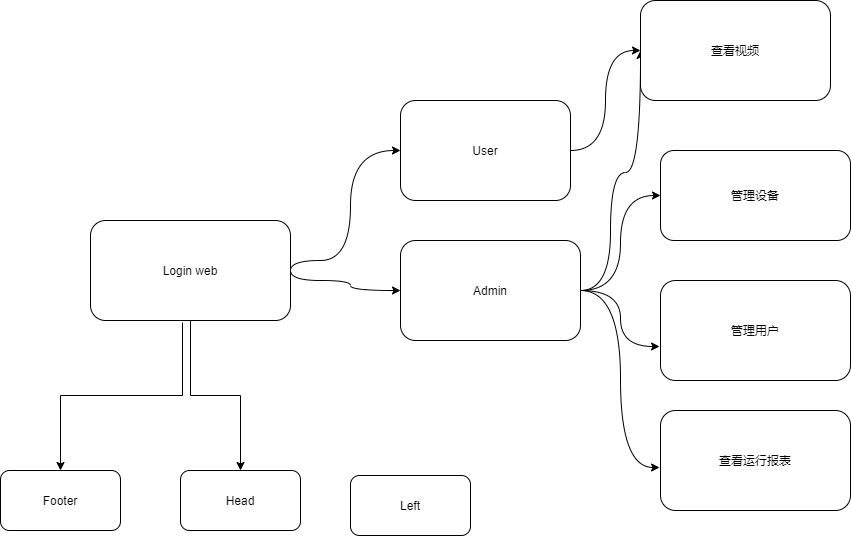
/api/v1/user

4.4 spring boot + webrtc 系统

管理子系统 : 用户，设备，树状管理  
 RTC子系统：webRTC服务：

拉流子系统：

* 1. react + antd + redux frontend构建



**5系统测试**

数据：

0号用户，

EDGE 0

设备0，

设备1

EDGE1

设备2，

设备3

2．媒体服务器搭建

2.1 AWS 购买

2.2数据库

2.3 media server

1. 系统实施

第1 阶段：

1. 购买云服务器
2. 建mongodb数据库
3. 搭建媒体服务器，添加媒体数据4个用于测试
4. 后台服务实现用户拉取
5. 后台服务实现设备拉取
6. 后台服务实现媒体信息拉取
7. 通过功能测试

第2阶段：

1. 添加前端框架，上，下，左
2. 添加前端用户登录功能
3. 进入用户和管理界面

完成上述功能测试

第3阶段

1. 用户界面功能实现显示设备列表
2. 管理界面功能实现显示用户，设备列表
3. 显示用户设备关联树

第4阶段

1. 后端WEBRTC服务实现
2. 前端查看视频界面

第5阶段

1. CRUD用户

第 6阶段

CRUD设备和关联

第7创段：

错误处理