# 模块划分

# 驾驶端:连接调度服务,实现叉车列表、视频的显示，故障车辆实时报警，以及对叉车的远程操控

# 车端主程序:连接调度服务,完成车端视频推流,响应驾驶端对车端的控制

# 车端桥接程序:用于车端主程序和ros系统之间数据的中转

# 调度服务:主要的服务端程序,中转车端与驾驶端数据

# RTSP服务:RTSP流媒体服务器

# 后勤服务:负责客户端自动更新功能、更新日志(网页)

# 架构图



# 逻辑视图

以下的时序图中, 但凡客户端(车端和驾驶端的统称)对服务器做出的请求，需要服务器必须有结果返回的，采用http请求，若无需有结果返回，则以websocket形式发送请求到服务，以下未标明通信方式的，默认为websocket。

* 驾驶端拉流



* 远程控制



* 车端数据(含故障)上报



# 技术框架

* 驾驶端:
* 编程语言:C++11
* UI框架:Qt 6.2.4(LTS)
* 调度服务
* 编程语言:go
* web服务框架(包含websocket服务):gin
* 后勤服务
* 编程语言:go
* web服务框架:gin
* 车端主程序
* 编程语言:C++11
* 通信库:Qt 6.2.4(LTS)
* 车端桥接程序
* 编程语言:go
* Ros库:goroslib