

Target: 构建一个模高效安全的机器人控制系统

- 安全（数据安全 操作安全）
- 运维友好(一键部署脚本)
- 团队协作DevOps(GitLab)
- 支持自动化测试(自动化测试框架)
- 模块高内聚低耦合(分层模块化设计)
- 支持接口扩展
- 精准控制

tony

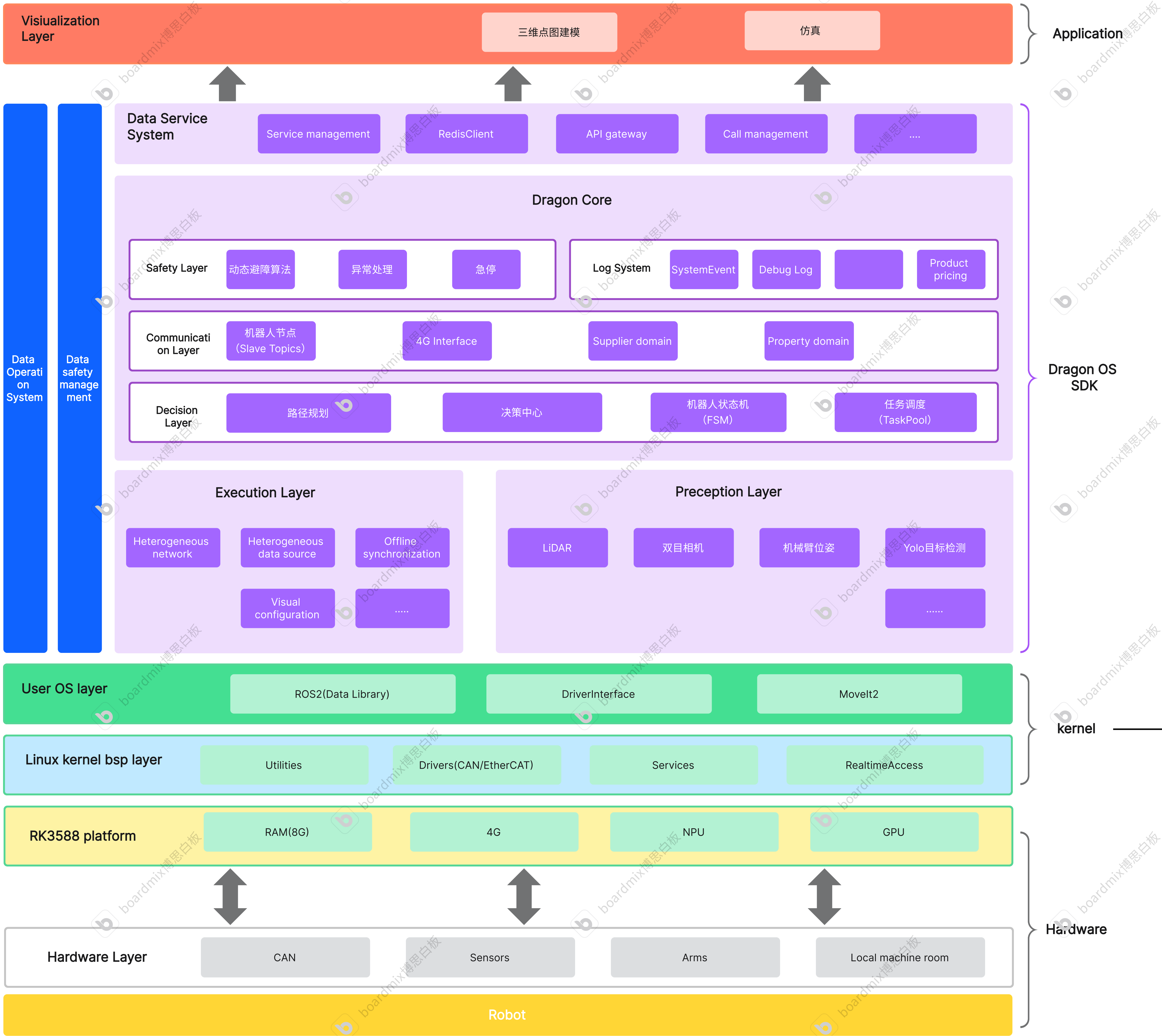
## Architecture Diagram

### 功能模块设计架构

基于ROS 2的搬运机器人控制系统软件架构设计包括以下几个主要功能模块:

1. **\*\*硬件抽象层 (Hardware Abstraction Layer) \*\***
    - **\*\*节点 (Nodes) \*\***: 负责与硬件设备通信, 如电机驱动器、传感器等。
    - **\*\*话题 (Topics) \*\***: 用于发布硬件状态和接收控制指令。
  2. **\*\*感知模块\*\***
    - **\*\*节点\*\***: 使用传感器数据进行环境感知和货物识别。
    - **\*\*话题\*\***: 发布感知结果, 如物体位置、障碍物信息等。
  3. **\*\*决策模块\*\***
    - **\*\*节点\*\***: 负责路径规划和动作决策。
    - **\*\*话题\*\***: 接收感知结果, 发布控制指令。
  4. **\*\*执行模块\*\***
    - **\*\*节点\*\***: 将决策模块的指令转换为电机控制信号。
    - **\*\*话题\*\***: 接收控制指令, 发布电机状态。
  5. **\*\*通信模块\*\***
    - **\*\*节点\*\***: 负责与外部系统通信, 如仓库管理系统。
    - **\*\*服务 (Services) \*\***: 提供RPC调用, 如请求任务、报告状态等。
    - **\*\*动作 (Actions) \*\***: 用于长时间运行的任务, 如导航到指定位置。
  6. **\*\*故障诊断模块\*\***
    - **\*\*节点\*\***: 监控系统状态, 检测故障。
    - **\*\*话题\*\***: 发布故障信息。
  7. **\*\*系统监控模块\*\***
    - **\*\*节点\*\***: 显示系统状态, 记录日志。
    - **\*\*话题\*\***: 发布系统状态信息。
  8. **\*\*用户界面\*\***
    - **\*\*节点\*\***: 提供图形化界面, 供用户操作和监控。
    - **\*\*话题和服务\*\***: 与用户交互, 接收命令, 显示信息。
- \*\*软件架构图概述\*\***:
- **\*\*顶层\*\***: 用户界面, 与用户交互。
  - **\*\*中间层\*\***: 决策模块和执行模块, 负责规划和执行任务。
  - **\*\*底层\*\***: 感知模块和硬件抽象层, 负责与环境交互和硬件控制。
  - **\*\*通信模块\*\***: 贯穿各层, 负责数据交换和外部通信。
  - **\*\*故障诊断和系统监控模块\*\***: 监控整个系统的运行状态。

## DragonOS Based on ROS2 and Moveit2



### Decision Layer

- 运动规划接口
  - /target\_pose 输入目标位姿
  - /collision\_map 输入实时障碍物点云
  - /joint\_trajectory 输出关节轨迹
- 运动控制层接口
  - /joint\_reference 输出目标关节角度
  - /joint\_feedback 输出实际关节角度(编码器反馈)
  - /torque\_command 输出电机力矩指令(mA)
- 机器人状态机
  - 空闲状态 (Idle)
  - 休眠状态 (Sleep)
  - 运动控制状态
  - 错误状态
    - Hard Fault
    - Fault
- 任务调度接口

### Preception Layer

- LiDAR
- Dragon Service
- Install/Deployment Script
- Devie Tree(LiDAR & Cameras)

### BSP Layer

- CAN Driver
- Dragon Service
- Install/Deployment Script
- Devie Tree(LiDAR & Cameras)