**基于物联网的动产质押系统详细设计文档  
 ——第二小组**

目录

[**一、** **文档概述** 1](#_Toc26566921)

[1. 目的 1](#_Toc26566922)

[2. 系统相关术语 1](#_Toc26566923)

[**二、** **系统数据流设计** 1](#_Toc26566924)

[1. 系统数据流 1](#_Toc26566925)

[1) 数据流图 1](#_Toc26566926)

[**三、** **系统物理架构设计** 2](#_Toc26566927)

[1. 系统整体物理架构 2](#_Toc26566928)

[**四、** **系统软件架构设计** 2](#_Toc26566929)

[1. 系统整体软件架构 2](#_Toc26566930)

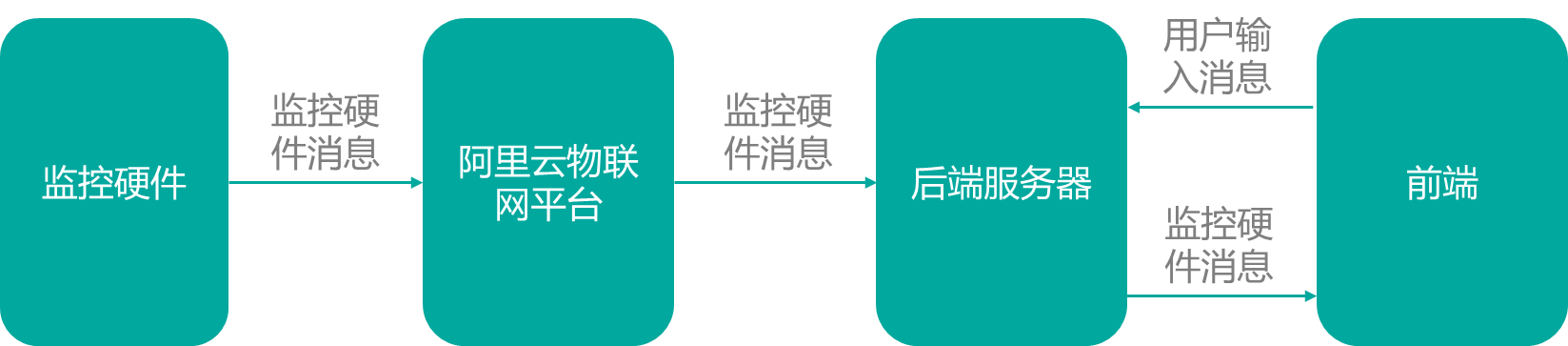
[2. 前端软件架构 3](#_Toc26566931)

[3. 后端软件架构 3](#_Toc26566932)

[4. 云端及云平台接口代码软件架构 3](#_Toc26566933)

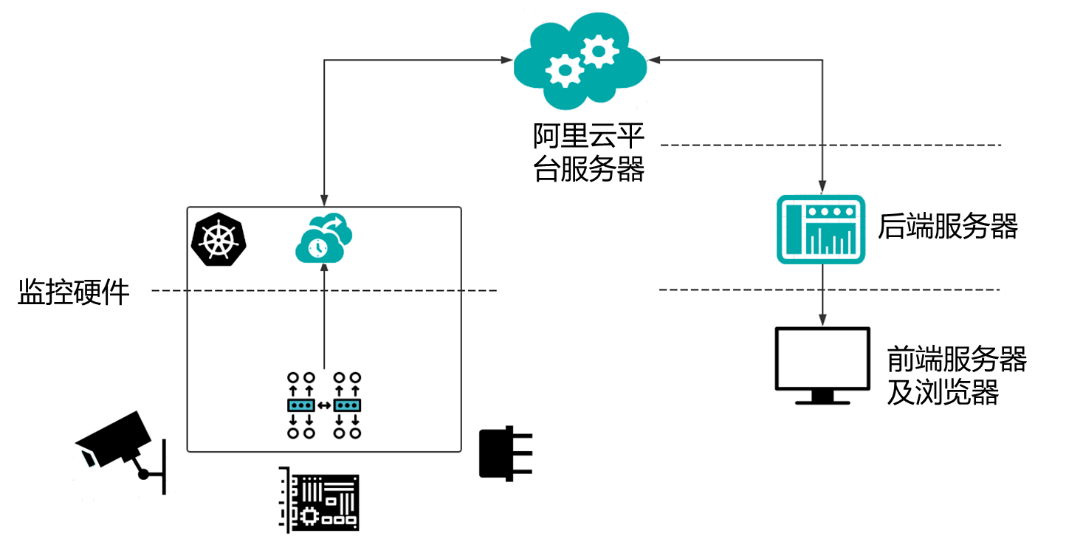
[5. 监控硬件软件架构 4](#_Toc26566934)

1. **文档概述**
   1. 目的
      1. 该文档的目的是详细地表述基于物联网的动产质押系统的设计方案，包括系统业务逻辑设计，系统物理架构设计，系统软件架构设计，系统软件运行流程设计
   2. 系统相关术语
      1. 前端：将质押物信息展示给用户的界面及其软件
      2. 后端：监控信息存储处理和与前端交互的软件
      3. 云端：阿里云提供的已经部署的提供服务的服务器
      4. 云平台接口：与云端交互而按照阿里云提供的接口编写的代码
      5. 监控硬件：动产质押系统中需要使用的监控硬件
      6. 物模型：用来描述的物理信息的模型
2. **系统数据流设计**
   1. 系统数据流
      1. 数据流图

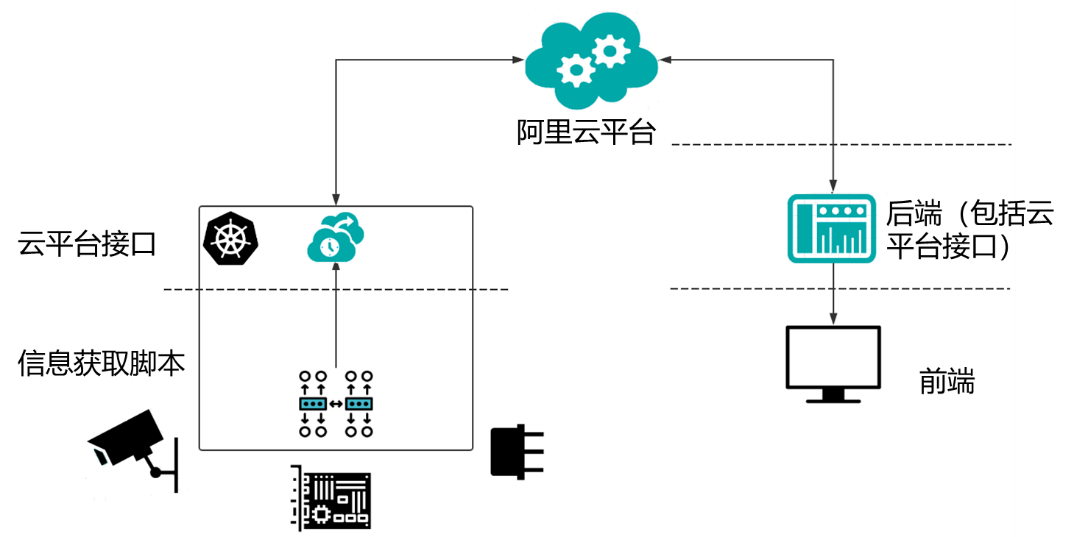


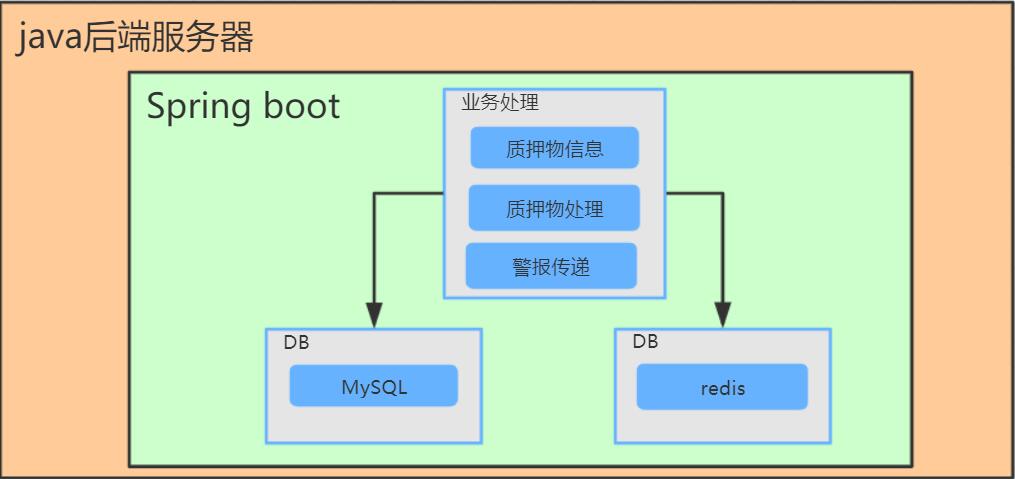
* + 1. 数据流说明
       1. 前端负责接收用户输入和显示格式化的监控硬件信息
       2. 后端负责持久化监控硬件的数据和用户输入的数据，并处理所有的业务逻辑（包括商品的出入库，商品是否异常的判断）
       3. 阿里云平台负责接收来自硬件的消息，并实时转发
       4. 监控硬件负责处理各个传感器的消息，并通过深度学习得到的模型，对摄像头拍摄的照片进行识别，最终处理成格式化信息并上传

1. **系统物理架构设计**
   1. 系统整体物理架构
      1. 架构图



* + 1. 架构说明
       1. 系统的物理架构分为四个部分，前端服务器（及浏览器），后端服务器，阿里云平台服务器，监控硬件

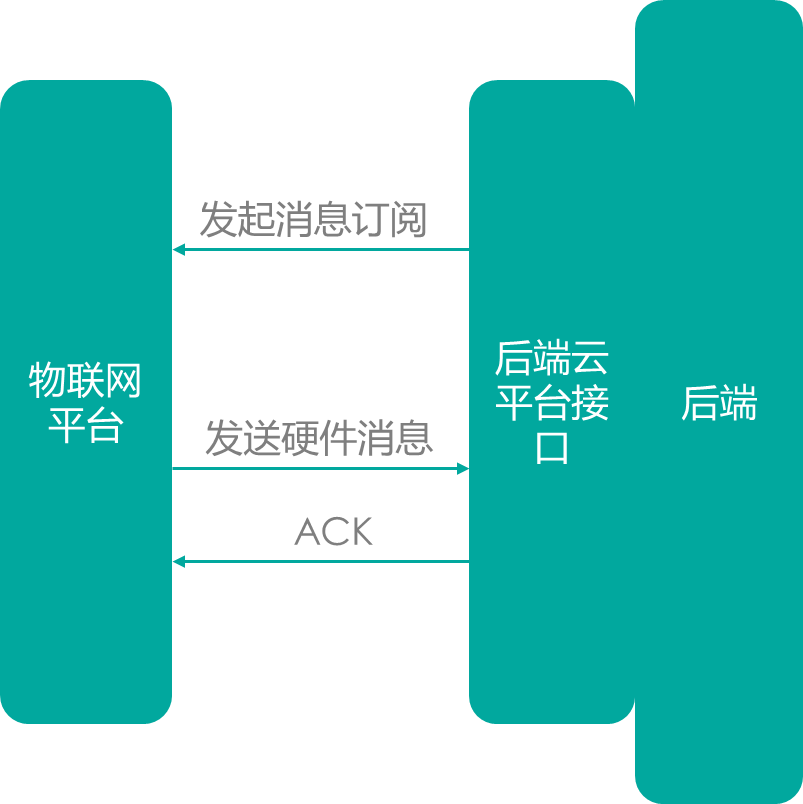
1. **系统软件架构设计**
   1. 系统整体软件架构
      1. 架构图
      2. 架构说明
         1. 整个系统的软件一共为五个部分，前端软件，后端软件（包括云平台接口），阿里云平台软件（由阿里云提供），监控硬件云平台接口，监控硬件信息获取脚本
   2. 前端软件架构
      1. Vue架构 version 2.9.6
         1. Index.vue文件写了左边的控制栏和警报信息
         2. Inwarehouse.vue 文件写了入库页面
         3. ShowAll.vue文件写了货物列表页面和货物详情页面，使用v-if切换
   3. 后端软件架构
      1. 架构图



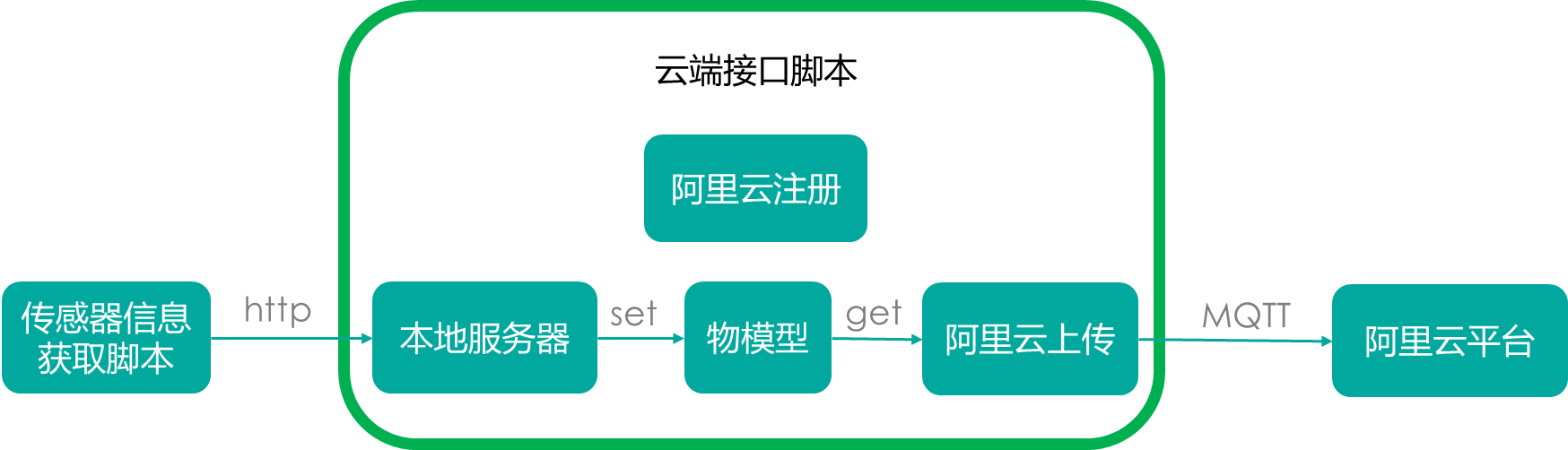
* + 1. 架构介绍

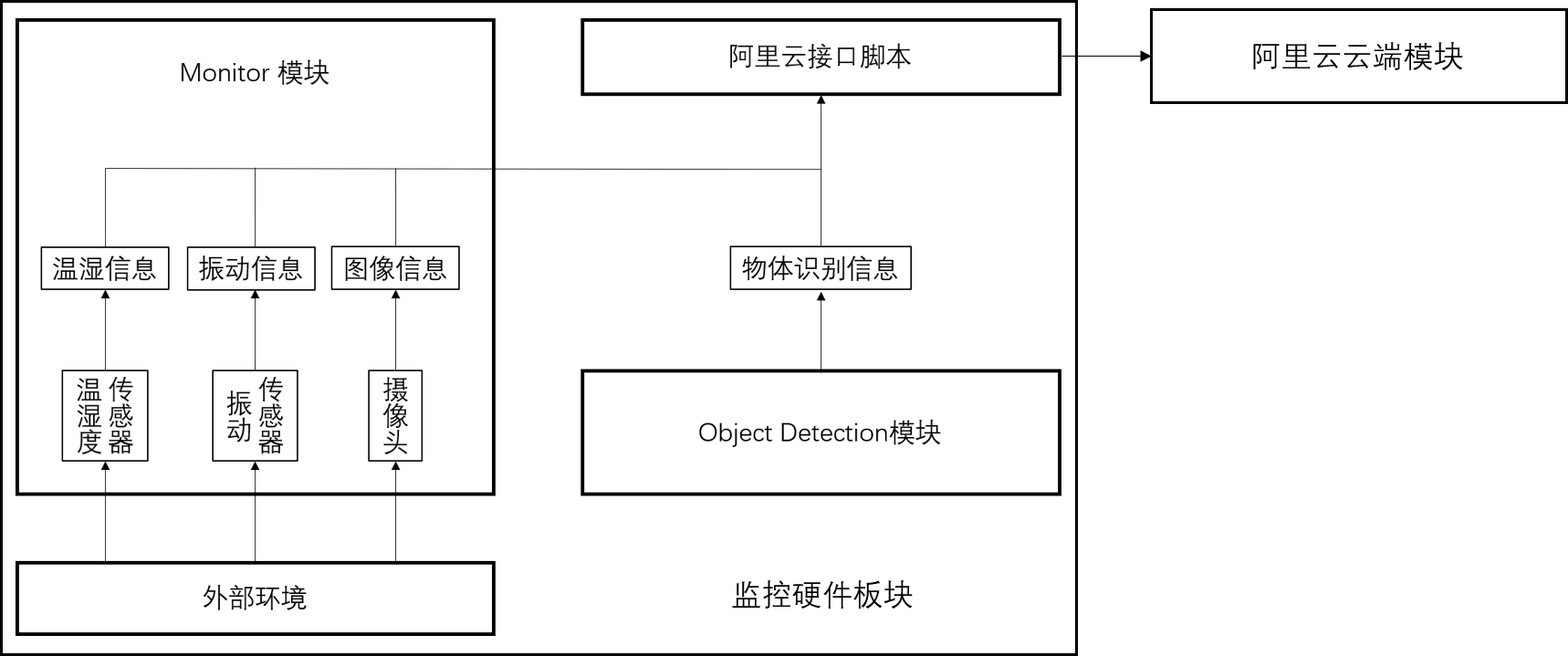
后端采用java服务器，以Spring boot为架构，maven管理资源项目，采用redis和mysql共同存储信息。接收自阿里云数据后，对数据和图像进行存储收集，同时接收前端请求，实现质押物的业务要求，再将数据信息展示给前端。

* 1. 云端及云平台接口代码软件架构
     1. 与后端对接的云平台接口架构
        1. 架构图



* + - 1. 架构说明
         1. 后端部署后，云平台接口发起订阅请求，并维持与物联网平台的连接，每当物联网平台收到了来自监控硬件的消息，都会向后端转发相应的消息
    1. 与监控硬件对接的云平台接口架构
       1. 架构图



* + - 1. 架构说明
         1. 与硬件对接的云平台接口文件（以下简称接口文件）采用Python语言编写，通过Docker打包后在NANO PC T4上运行，通过阿里云提供的Python SDK与阿里云物联网平台对接，通过在硬件搭建的本地服务器与监控硬件对接，其中，该接口由四个部分构成：物模型文件，阿里云设备注册文件，阿里云数据上传文件，本地服务器文件
         2. 物模型文件通过device\_property.json文件进行初始化，该文件中记录了物模型信息，包括名称和数据类型，初始化完成后文件在内存中可以构建出监控硬件的物模型
         3. 阿里云设备注册文件通过aliyun\_property.json文件进行初始化，该文件中记录了阿里云的账户信息，在云平台接口文件运行后，会先运行阿里云设备注册文件，判断该设备是否已在阿里云物联网平台注册，如果未注册则进行注册，并将阿里云物联网平台响应中的注册信息写入新建的send\_property.json文件
         4. 阿里云数据上传文件通过send\_property.json文件进行初始化，该文件中记录了该硬件在阿里云物联网平台的注册信息，以及上传物模型数据的时间间隔，在阿里云设备注册文件运行后，会按照一定的时间间隔读取物模型信息，并上传至云平台
         5. 本地服务器文件通过localserver\_property.json文件进行初始化，该文件中记录了云平台接口部署的本地服务器所在的端口，在本地服务器文件运行后，会在相应的端口部署本地服务器，在监控硬件的传感器信息获取脚本得到监控信息后，会通过http请求访问该端口，发送硬件信息，本地服务器将根据收到的硬件信息，更新物模型
  1. 监控硬件软件架构
     1. 架构图
     2. 架构说明
        1. 硬件部分有Monitor模块与Object Detection模块，Monitor模块负责对接各类传感器设备并进行初始数据的收集，而Object Detection模块则对摄像头采集的图像信息进行物体识别、新旧比对，最终返回一个3-bit编码的识别信息，三个bit分别代表是否振动、是否移动、数量是否变动，比如返回值6=(110)2代表当前质押物发生振动、被移动、但是数目上没有变化，至于是否报警warning则是后端的业务逻辑具体决定
        2. Monitor模块的采用wring-pi等python开源库采集传感器数据，以HTTP请求的形式向Object Detection模块发送识别请求，以HTTP请求的形式向阿里云接口脚本发送传输请求，向云端推送最新数据
        3. Object Detection模块采用自主修改的yolov3模型，使用Pytorch，数据集为MS-COCO2014，在4×K80的服务器上完成训练，并利用flask完成识别模型的硬件本地部署工作，方便Monitor模块以HTTP请求的形式调用物体识别服务。此处需要强调的是Monitor模块调用Object Detection模块不会将图片作为参数传输，而是在Object Detection服务部署的时候约定一个共享文件夹，里面实时维护两张图片file-new.jpg与file-old.jpg，分别代表本次拍摄的照片与上次拍摄的照片，而Object Detection模块在接收请求后对该约定文件夹中的两张图片进行识别与新旧比对
        4. 另外助教提供的车牌识别(hyperlpr)模块以及表器识别模块因为arm板卡架构本身的支持问题，无法在板卡上部署相应的服务，只能在个人x86架构的服务器上跑通，具体跑通的结果和我们遇到的问题、反思都已经在答辩ppt中予以展示。