

构建物联网类项目和互联网类项目的异同

本文概述

个人这几年在物联网行业摸爬滚打，做过业务系统，也做过中间件和网关。经过这几年的沉淀，也算是对这个领域内稍微有点皮毛认知。加上陆陆续续经历了几家公司，也见了不少物联网相关的项目，但是发现大部分公司对于物联网认识不深刻，在技术架构方向一直在套用互联网领域的技术和设计，虽然说这样短期能实现功能，达到“能用”的程度，但是对于后期发展和业务扩大埋下了隐患，以至于最后因为技术崩盘而导致业务萎缩失败。因此这些年对于这些项目中出现的问题以及设计方面有些自己的想法和建议，希望能让一些纯粹的做软件级以及架构出身的开发者们对于物联网和互联网有个初步的认识和区分。

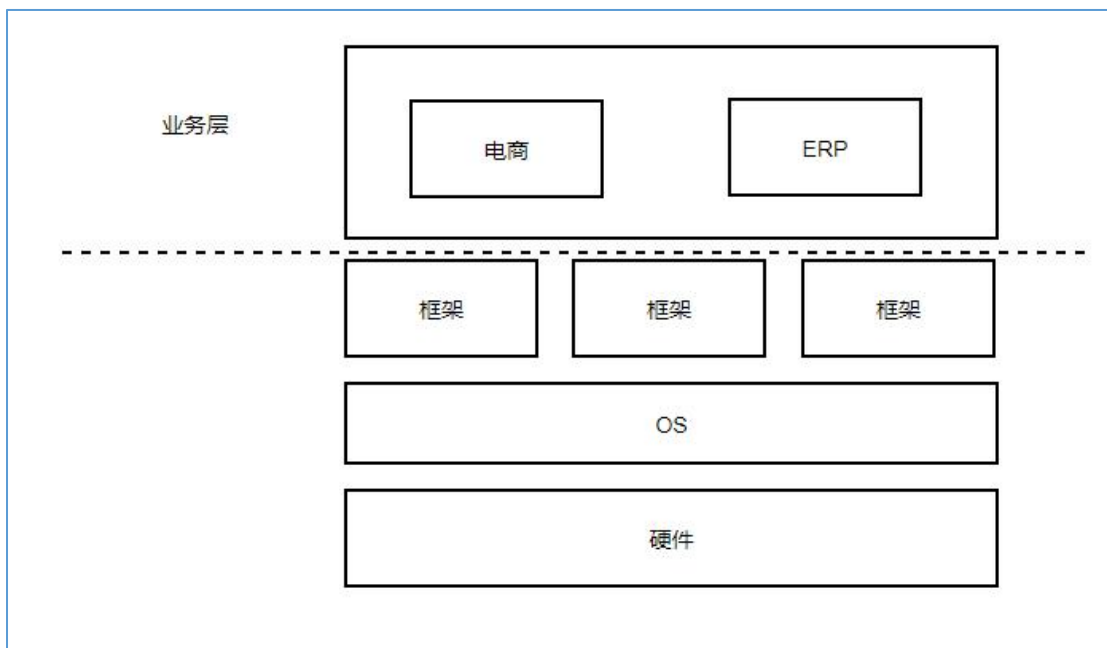
常见互联网技术栈

编程语言一般以 Java，Go，Python，PHP 为主，大部分要求数据库以及缓存等组件，但是互联网技术有个典型特征就是以实现功能为主，也就是绝大部分时刻我们花时间在研究业务系统，比如下面几个场景：

- ❖ 超市购物满 100 送 10 元购物券，然后让用户三个月内用完，不然就作废；
- ❖ 每当用户推荐 APP 给新的用户，然后给这个老用户增加一定的积分，同时把他的推荐权限给取消；
- ❖ 食堂或者酒店的电子点餐单每天特定时间自动更新菜单，比如中午有炒菜，但是早上是豆浆包子。

上面我列举了三种典型场景，包含了电商场景，推广场景，购物场景，我们在上面的场景下需要关注的就是业务本身的实现过程，即：聚焦于如何实现这个业务。

在需求点的驱动下，我们会选择技术栈，比如可能选择 Java，然后中间件用 XXXMQ，框架用 Spring Boot，服务注册管理用 Nacos，前端用 VUE 等。在这种场景下面，我们是不关注过程和细节的，比如数据怎么处理，怎么到达，以及如何保证数据和网络的可靠性。因为架构设计选择的基础件已经帮我们实现了这些东西。



如上图所示，我们关注点在业务层

常见物联网技术栈

在物联网场景下，技术栈和互联网场景基本一致，比如 Java，Go，Python 等，但是分的更细了：Java 常做后端，Go 常做后端或者边缘计算，Python 常做 AI 相关的。但是他们都是在物联网场景下的技术链上的环节。

典型的场景;

- ❖ 城市水电远程抄表，安防等



- ❖ 某地几万平方公里光伏电站统一管理



- ❖ 汽车数据管理以及车辆维护跟踪

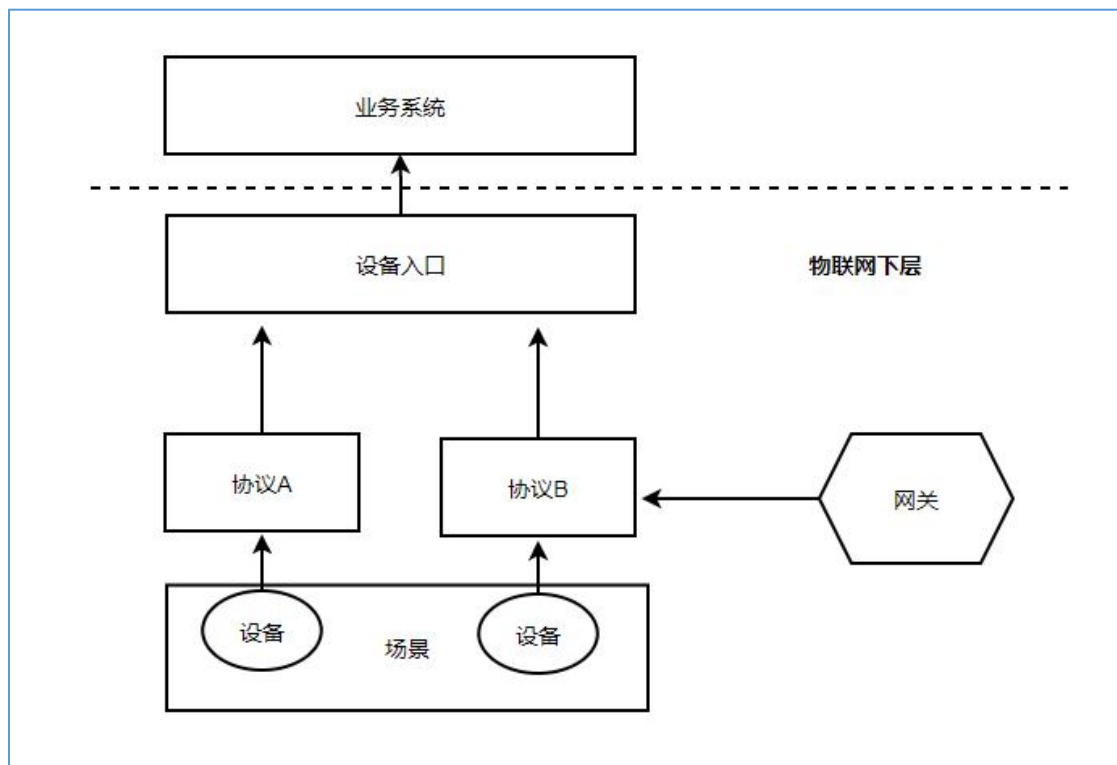


上面我列举了三种场景，都是我们日常能接触到的，我们不难发现物联网场景和互联网场景最大的区别：

1. 应用系统设以备接入为主
2. 设备分布广泛，量大，环境可能恶劣
3. 数据量庞大而数据结构单一简单

从上面的案例以及总结我们不难看出，做物联网相关项目，我们需要将在注意力下移，在应用层之下还需要关注几个点：

- 设备接入
- 数据上行
- 指令下行
- 数据沉淀

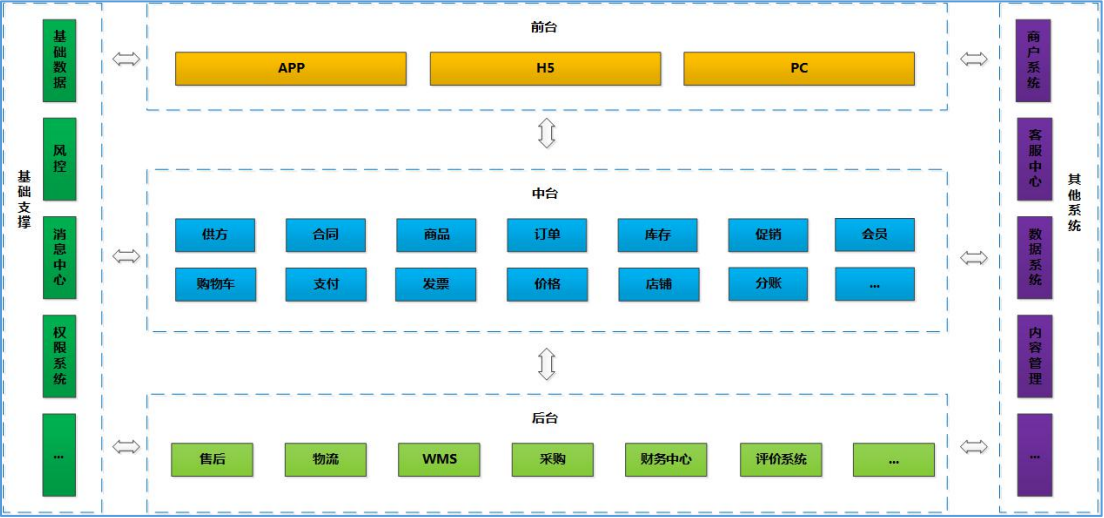


我们的注意力在：接入点，设备分类，协议选择，合适的场景这几个点上

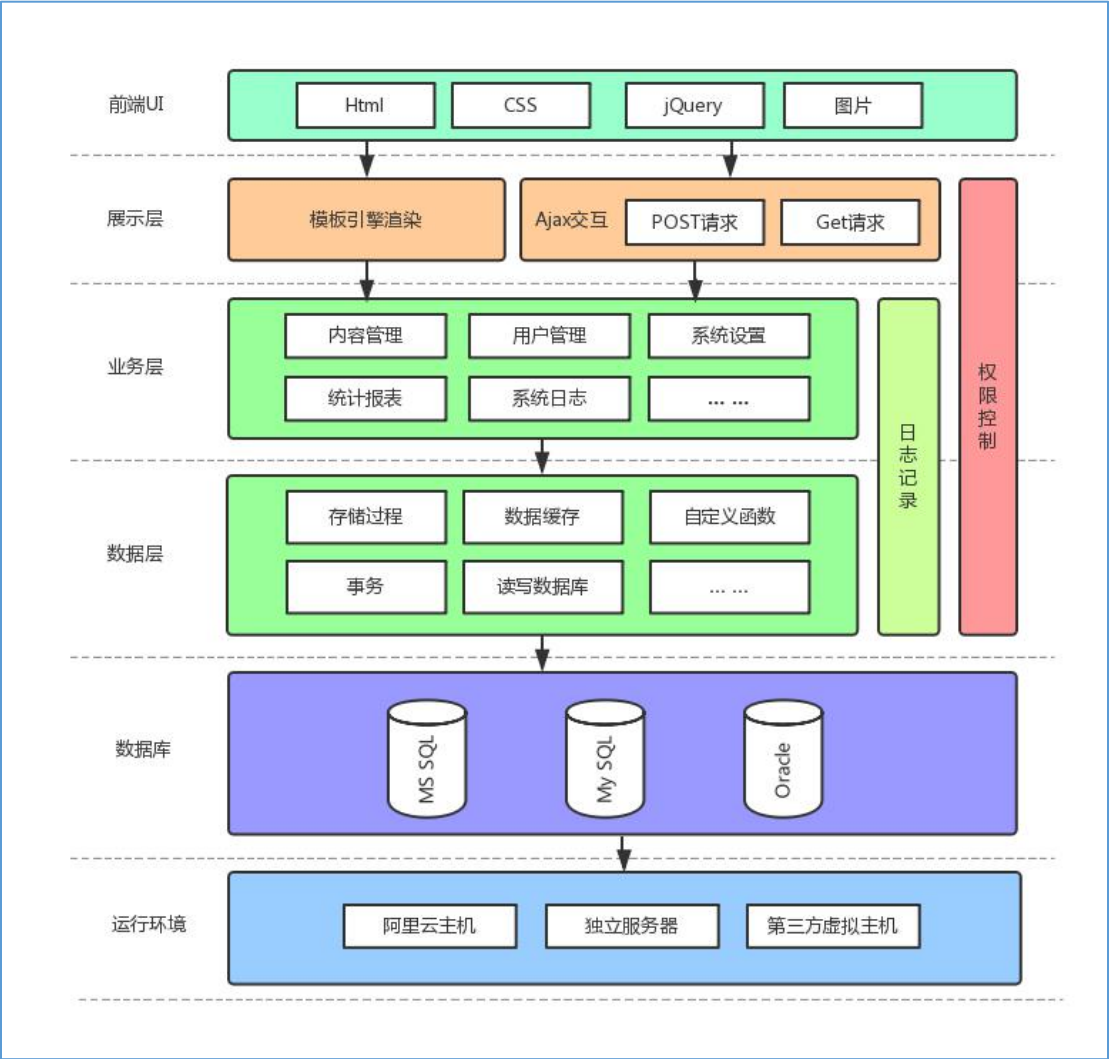
常见互联网架构

这里来展示常见的 2 种互联网场景下的系统结构：

电商



内容类型的系统

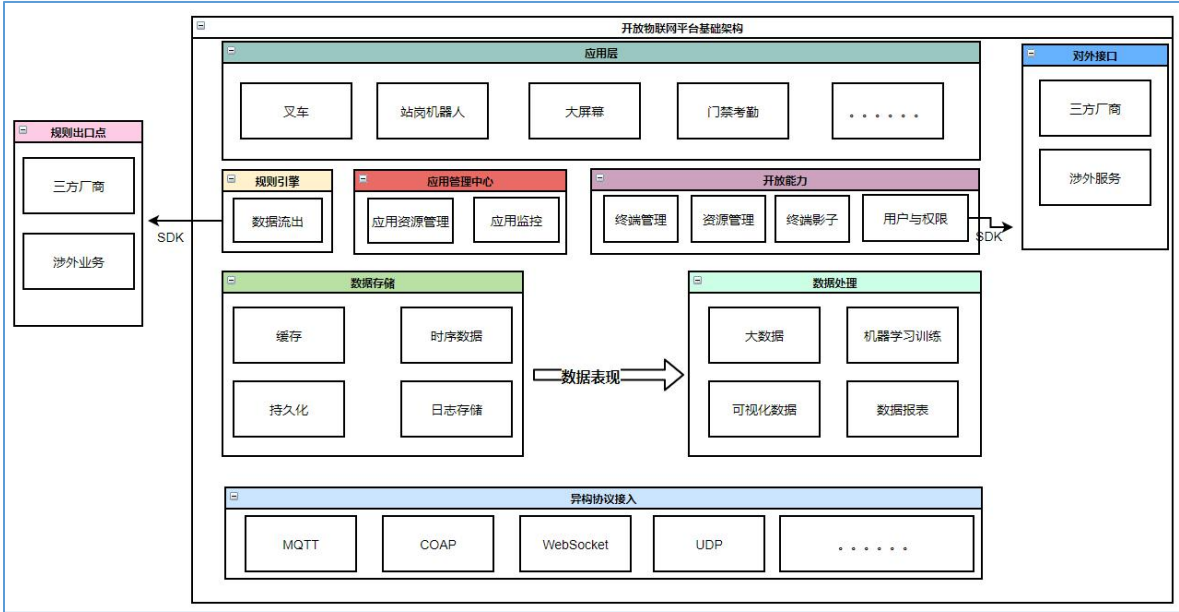


上面展示了 2 种常见的互联网软件系统的架构：基础电商和基础内容系统，我们其实不难发现这两类系统都是关注于“电商”和“内容”的业务，而没有涉及到像考虑环境，部署点情况等外部因素。

常见物联网架构

这里来展示 2 个物联网场景下的系统架构：

开放式设备接入平台



私有化园区管理方案



分析上面的两个架构图不难看出，物联网系统还需要在应用层下面分离出来一个接入层，而且需要考虑的点比较多，对项目的垂直化细分的较为明显，比如园区和室外方案就不可互相套用，或者抄表方案和机床控制方案也不可套用。一个物联网系统绝对是从业务到接入到场景一起作用的，耽误脱离任何一个环节都会留下缺陷。

二者为什么不能直接互相套用

到这里已经接近尾声，最后我们聊聊为何看似技术都是一样的，我的电商方案拿过来做机器人控制或者是楼宇方案，怎么就不行了呢？

我总结了这几个点：

- **技术可行但是不完善，只解决了部分问题**

我用 Java、PHP 等可以做物联网系统吗？答案是可以，但是只能解决业务问题，解决不了下层的，比如 PHP 可以实现设备管理以及对接小程序，或者支付等，但是对于下层的设计就不合适了。

- **不能脱离场景只设计业务，导致关键环节缺失**

业务系统往往和部署场景没有太大关系，比如今天我们做“电商管理后台”要不要考虑用户是在信号很弱，而且带宽极低的沙漠里面的 2G 网络时候，怎么设计用户的终端来保持高效通信？我们是不可能干涉用户的，只能假设用户处于完全理想的访问环境下，然后我们的系统保证数据正确性 就可以了，至于用户信号差，那还是不要用了，换一个手机不就行了。但是实际上物联网场景下会在沙漠里面部署光伏电站的，我们必须得设计没信号，没足够带宽的时候还能处理数据，这就是关键环节“终端在哪里，该怎么部署”环节缺失。

- **相同技术在不同领域的应用方式不同**

就拿 redis 来说，我们做互联网软件时绝大部分时间用来做缓存，然后用的比较多的时候也有消息队列，或者分布式锁，分布式配置中心等。但是在物联网场景下我们可能拿来做状态机保存，或者是做影子设备（有时候也叫孪生数字），相反纯缓存和消息队列基本上不怎么用。Mongo Db 也是一样，看到很多项目有用 Mongo Db 保存业务数据的，但是在物联网场景下，Mongo Db 大部分时候用来保存设备数据或者日志。

有很多案例，其实大家多分析对比一下就能了解。

- **纯应用思维容易陷入自我感知错误**

做纯应用的时候，其实大部分时候我们的软件运行在一个温室里面，假设各种恶劣环境都被解决了，典型的就是上面提到的用户终端是什么类型的案例。往往我们要做的就是保证数据正确性。当我们保证用算法以及业务约束实现了数据绝对正确的时候，就会怀疑其他人的不正确，实则相反，当终端设备的环境不确定的时候，我们不能单靠者应用层觉得是对的，就否定来自客户端的请求。

典型案例：某一批设备我们连接服务器的时候，假设 30 秒没有发送数据包，那就是掉线了，但是我们能绝对认为就是 30 秒吗？有时候网络原因以及设备自己的本身原因，可能变成 35 秒的时候，我们怎么去处理这个约束范围。这里就撤出来了时间窗口的问题，还有消息质量（QoS）机制我们怎么去考虑？

上面说的就是“纯应用思维容易陷入自我感知错误”，我们往往很确定自己已有的经验而否定自己未知的一些知识，这点对我们开发者是极其致命的。

总结

本文主要谈了自己的一些对于物联网行业的认识和观点，希望大家能给出建议以便于更加完善。