

# 第9讲 物联网基础（补充）

# 目录

01

## 物联网架构和相关技术

- 1 物联网架构
- 2 主流物联网技术
- 3 和其它热点关系

02

## 物联网云平台

03

## 物联网设备端开发

# 物联网架构



特征

体系化

分层化

合作化



核心：云平台

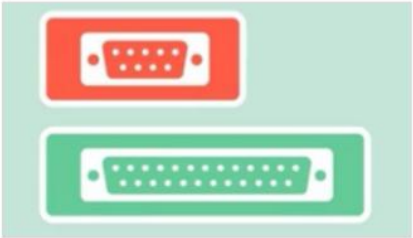
- 在互联网中是核心，软件就是商业模式，核心竞争力，物联网中需要下层的承载
- 云厂家做的少，明确做物联网云：阿里 华为 云的集中性越高越好 对接客户需求
- 网络不跟需求挂钩
- 设备端复杂多样，类型多，存在芯片硬件差异，质量参差不齐，筛选甄别难度大，标准无法制定，例如消防特殊行业有特殊标准，圈子小

# 物联网技术—有线传输

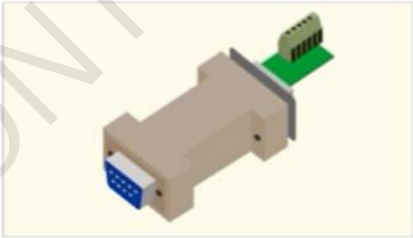
RS232串口： 串行通信接口， 全名是“数据终端设备（DTE）和数据通讯设备（DCE）之间 串行二进制数据交换接口技术标准”， 是电脑与其它设备传送信息的一种标准接口。

RS-485总线： 在要求通信距离为几十米到上千米时或者有多设备联网需求时， RS232无法满足， 因此诞生了RS-485 串行总线标准。

比较项	RS-232	RS-485
通讯距离	不超过20米	1200m的理论值， 实际300-500m
传输方式	不平衡传输方式， 单端通信	平衡传输， 差分传输方式
通信数量	一对一通信	总线上最多允许128个收发器
传输速率	38.4Kbps	10Mbps



RS-232



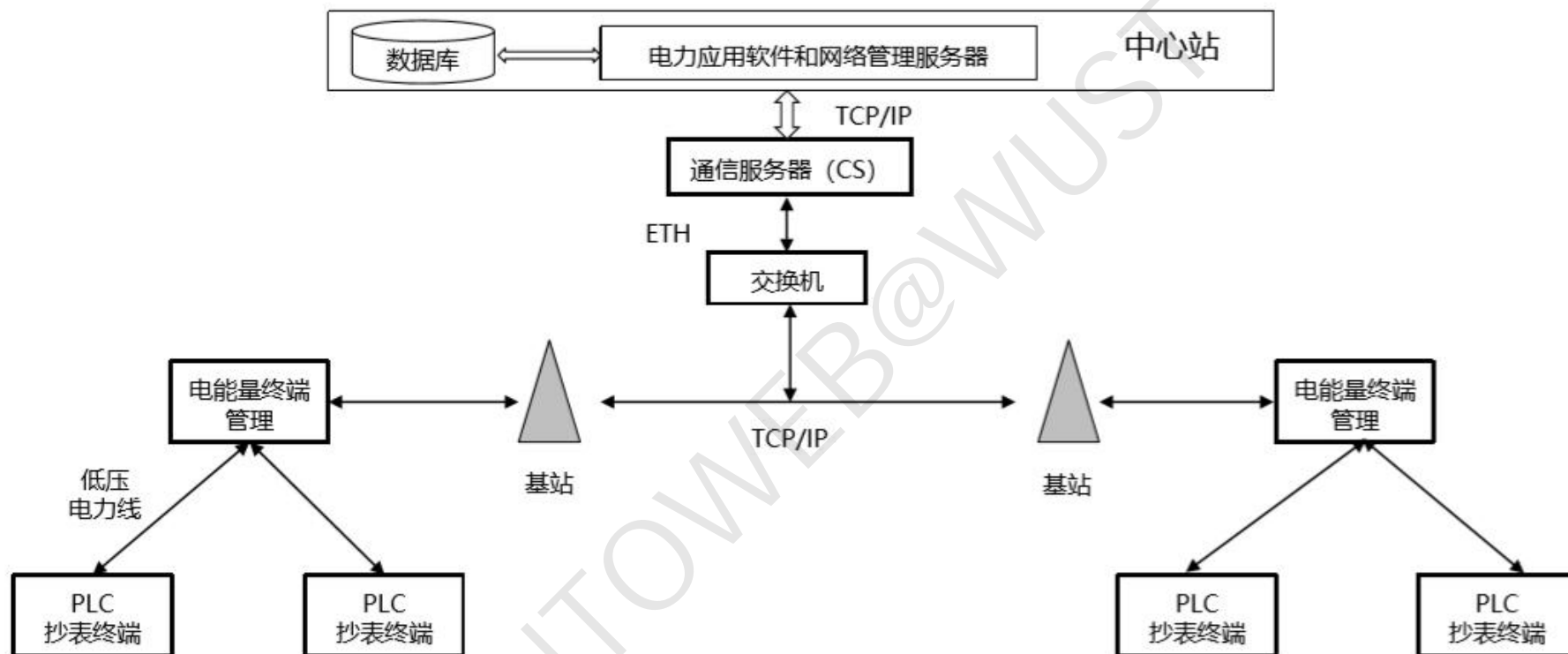
RS-485

USB： 通用串行总线， 是一个外部总线标准， 支持设备的即插即用和热插拔功能， 具有传输速度快、 使用方便、 连接灵活， 独立供电等优点。

Modbus是一种串行通信协议， 已经成为工业领域通信协议的业界标准。 由带智能终端的可编程序控制器和计算机通过公用线路或局部专用线路连接而成。 它可应用于各种数据采集和过程监控。

M-Bus (Meter Bus)—户用仪表总线， 它是一种专门为消耗测量仪器和计数器传送信息的数据总线设计的。 M-Bus在建筑物和工业能源消耗数据采集有多方面的应用。

# 物联网技术—有线传输



- 电力线通信 (Power Line Communication, 英文简称PLC) 技术是指利用电力线传输数据和媒体信号的一种通信方式。该技术是把载有信息的高频加载于电流然后用电线传输接受信息的适配器再把高频从电流中分离出来并传送到计算机或电话以实现信息传递。

# 物联网技术—有线对比

通信方式	特点	适用场景
ETH	协议全面、通用、成本低	智能终端、视频监控
RS-232	一对一通信、成本低、传输距离较近	少量仪表，工业控制等
RS-485	总线方式、成本低、抗干扰性强	工业仪表、抄表等
M-Bus	针对抄表设计、使用普通双绞线、抗干扰性强	工业能源消耗数据采集
PLC	针对电力载波、覆盖范围广、安装简便	电网传输、电表

# 物联网技术—无线传输

				优点	缺点	场景
无线通讯方式	短距离无线	蓝牙	速度1mbps, 距离0.1—10m, 增大功率可达100m	速率快, 低功耗, 安全性高	网络节点少, 不适合多点布控	鼠标, 无线耳机, 手机电脑等
		WIFI	射频频段2.4G, 5G, 50—100m	覆盖广, 速度快	稳定性差, 功耗略高, 组网能力差	智慧家庭
		zigbee	低速 (20—250kbps) 短距离 (10—100m), 短时延	低功耗, 高容量, 高安全, 易组网	难兼容, 维护难	工业, 智慧家庭, 智慧楼宇
		zwave	基于射频短距离通信, 速率9.6kbps或40kbps, 室内30m—室外100m	低速, 低成本, 低功耗	标准不开放, 唯一来源 sigma designs	智能家居, 监控和控制
	蜂窝网络	2/3/4G				
		Nbiot	速率<100kbps, 1—20km			水表, 停车, 宠物跟踪, 垃圾桶, 烟感, 零售
	LPWA	lora	长距离 (1—20km), 节点数万级, 百万级, 寿命3—10年, 速率0.3—50kbps			智慧农业, 智能建筑, 物流追踪
		sigfox	100bps每天每设备140条消息, 每条消息12字节, 1—50km	低功耗, 高容量, 可拓展	免授权频段	智慧家庭, 智能电表, 移动医疗, 远程监控, 零售

# NB-IoT特性

**10 年**  
电池寿命



**20dB**  
覆盖增益



**5美金**  
模组价格



**5万**  
每小区连接



**NB-IoT**

**NB-IoT技术优势**



**基于现网演进**



**运营商级别的可靠性**



**高安全性**



**全球漫游**

## 低成本

### 5\$ 模组成本

- 简化射频硬件
- 简化协议降低成本
- 减小基带复杂度

## 低功耗

### 10 年电池寿命

- 简化协议，芯片功耗低
- 功放效率高
- 发射/接收时间短

## 广覆盖

### 20dB 增益

- 窄带功率谱密度提升
- 重传次数：16次
- 编码增益

## 大连接

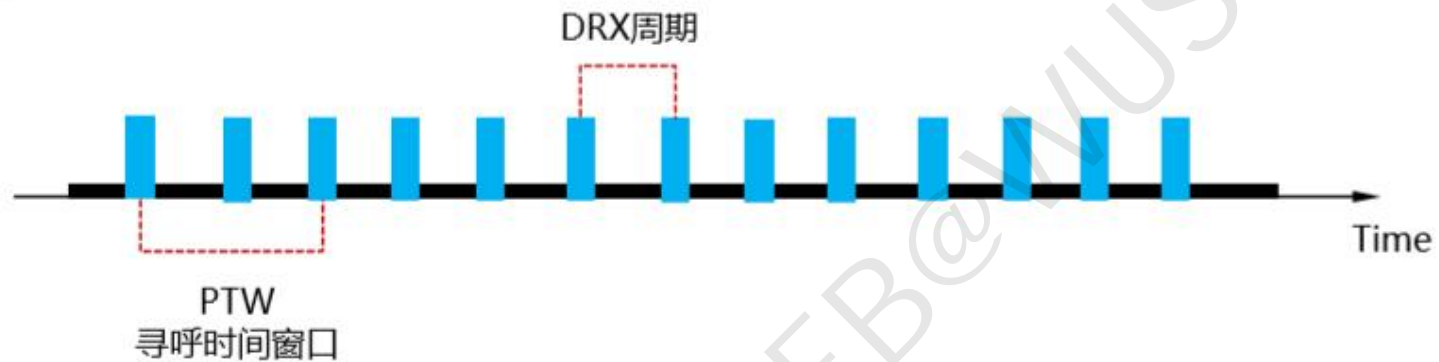
### 50K连接数每小区

- 频谱效率高
- 小包数据发送特征
- 终端极低激活比

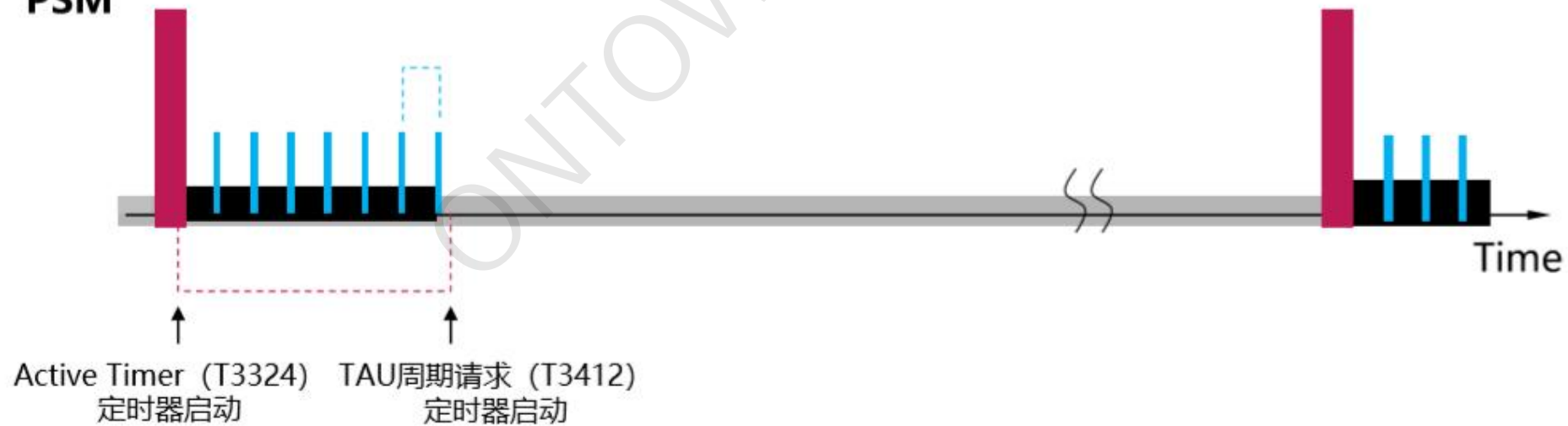


# 省电模式

DRX



PSM



# NB LoRa对比

## 专用网关

- LoRa 需要专用网关才能工作，而 NB-IoT 不需要。NB-IoT 的基础设施是直接连接基站和传感器设立的，LoRa 架构需要通过网关来完成。
- 虽然目前可用的 LoRa 网关很强大，而且通常价格有竞争力，但仍有额外一些硬件需要管理，因此可能是一个潜在的麻烦。而 NB-IoT 根本不需要这些。

## 生态系统

- LoRa 的生态系统比 NB-IoT 或其他蜂窝物联网通信标准的生态系统要强。
- LoRa 已经在许多地区被采纳为物联网网络标准，包括美国、澳大利亚、新西兰、中国台湾地区、荷兰和印度。
- NB-IoT 相比之下较新，2017 年初才在西班牙进行商业首秀（爱尔兰也将使用 NB-IoT 网络）。但在未来几年内，NB-IoT 将有很多机会迎头赶上。

## 所需频谱

- LoRa 物联网技术在非授权频谱上工作。由 LoRaWAN 支持的应用程序成本低，而且电池性反而得到提升（以后会越来越多）。
- 与 LoRa 的异步协议不同，NB-IoT 服务是同步的并使用授权频段（LoRa 和 NB-IoT 都使用低于 1GHz 的频段）。频段授权的成本现在并不低——每 MHz 的成本大于 500 美元。电信运营商可以选择在防护频段、4G LTE 频谱、或者独立网络中部署 NB-IoT。

## 应用匹配

例如，LoRaWAN 适合用于需要较低成本、较高电池寿命，不需要频繁通信的应用程序或设备（比如说一天几次）。相比之下，NB-IoT 最适合需要较短的下行延迟并且更频繁通信的应用程序。两种物联网协议都有其独特的价值体现——不能互相取代。

# NB LoRa对比

## 网络覆盖

- NB-IoT 在这方面更强大，其覆盖范围可延长至 18-21 公里，高于 LoRa 提供的 12-15 公里覆盖范围。
- 然而NB-IoT在城市运作良好，但在郊区或农村地区等没有强大4G 覆盖的地方表现一般。由于 LoRaWAN 不依赖蜂窝数据或 WiFi，其覆盖范围在所有地区都保持相对稳定。

## 电池性能

- 由于 NB-IoT 在蜂窝授权频谱上工作，设备必须定时（相对频繁地）进行网络同步，这相应会消耗电量。
- 基于 ALOHA 的 LoRa 架构中不需要这样的网络同步。在 LoRa 的异步频段中，终端应用程序精确决定设备的“休眠”时间，因此电池电量可以轻松保存。NB-IoT 的线性发射器需要的“峰值电流”比采用非线性调制的 LoRa 多几个数量级，对电池施加额外压力。

## 数据速率

- 在这方面，NB-IoT 或多或少可以将 LoRa 甩出局。窄带设置中的平均数据速率为 200 Kbps，大约是 LoRa 工具运行的数据速率的 20 倍。
- NB-IoT 的较高数据速率使其成为需要更快速数据吞吐量应用程序的理想选择。LoRa 技术做得虽然不错，但 NB-IoT 仍轻松成为“更快速”应用程序的更高效的物联网协议。

## 私有网络

LoRaWAN 技术市场刚刚成熟，在公共网络中得到广泛接受。LoRaWAN 已经在全球各地部署创建“智慧城市”。虽然 NB-IoT 也在公共领域很受欢迎，但不像 LoRa 一样能在私有企业的专有网络中使用。大型企业可以轻松使用 LoRa 创建混合型物联网模型，建设“智能设施”并同时使用公共网络处理设备外信息和活动。而 NB-IoT 只能用于公共网络模式。

# NB相对优势

- 运维成本：LORA需要自组网，自主维护，运营成本高，NB可以通过基站组网，无需自组网。
- 灵活性：LORA需要专用网关，设备数量少时没有优势。
- 网络覆盖和部署时间表：在中国，lorawan相应的商业化和产业链的建立还需要更长的时间和努力去探索。NB-IoT标准在中国有自主知识产权，目前已开始应用，在国内更易推广。
- 安全问题：LoRa工作在1GHz以下的非授权频段，免费频段+公开标准=易受攻击。这虽然有利于LoRa网络的大规模推广，但是却为未来的网络安全埋下隐患。
- 终端：LoRa终端无SIM卡，是弱终端，终端网络认证凭证，没有类似SIM卡的安全存储介质，安全需依赖终端的物理防护，这对于弱终端而言存在极大泄露风险。NB-IoT终端网络认证凭证采用SIM卡存储，提供安全可信存储。
- 厂商：在当前政治环境下，LoRa由美国公司独家私有技术垄断，这无论从国家和IoT产业链，还是公司利益而言，都存在芯片被禁售、支付高昂专利费、数据被窃取、网络被控制等潜在巨大的技术和国家安全风险。NB-IoT属于3GPP开放标准体系，技术专利由多家公司提供，且芯片供应厂家众多（有14家），技术可天然演进到5G，因此技术和国家安全风险很低。

# 边缘计算

## 方式

在靠近终端的网络边缘节点上,提供连接、计算、存储、控制与应用功能,满足用户实时、智能、安全和数据聚合等需求。

- 接入: 各种协议数据格式形成标准统一模型
- 计算: 处理, 计算, 暂存
- 清洗: 原始数据加工, 提炼有效信息上传

## 商业价值:

借助成熟的通信技术,在位于网络边缘的节点上分布式部署计算、存储、安全等能力,把中心节点的计算、存储、通信压力分散到计算能力稍弱的边缘节点,同时实现了服务的低时延、高可靠、低成本,有效地保护了用户的边缘隐私,支持网络从成本中心向商业价值中心的转移。

## 价值

## 核心价值:

支持实时性业务:可以做到ms级的数据实时分析、事件实时响应,支持动态调整路径。

支持边缘智能分析处理:业务边缘部署灵活调整,网络自动运维。

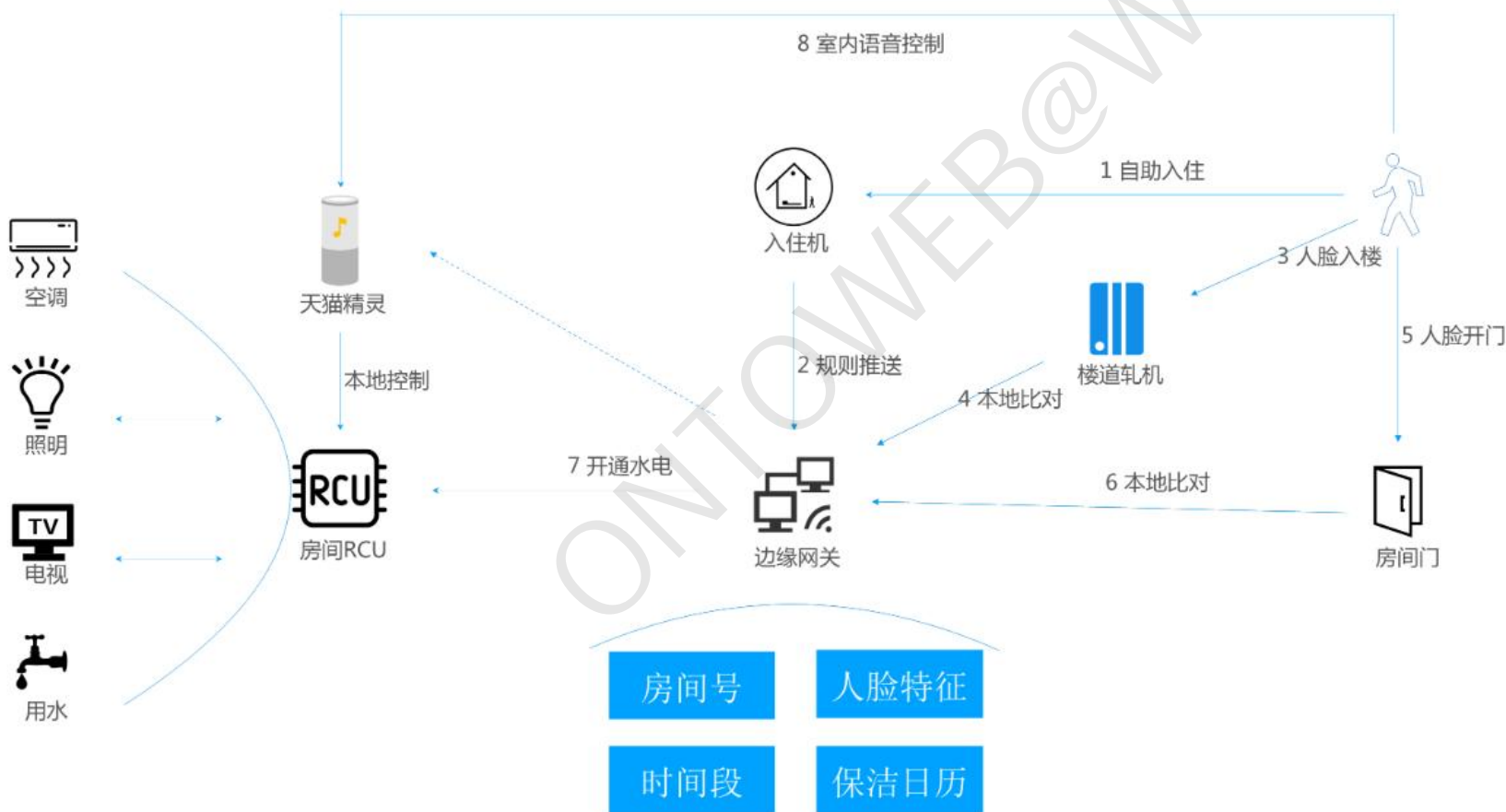
数据聚合:消除数据碎片化、屏蔽无效噪声、数据按需上传。

私有的安全域:包括数据安全、节点安全、网络安全。

# 边缘计算场景

## 未来酒店

通过边缘网关快速集成本地设备后，边缘网关作为本地节点快速响应本地事件，实现本地M2M的智能联动，实现室内室外一体化的语音智能。



### 特点:

- ◆ **设备联动**: 入楼闸机、房间门、空调、照明、水电等智能联动。
- ◆ **边缘计算**: 人脸信息、房间号、保洁日历、时间段等全部由边缘网关计算处理。



# 边缘计算场景

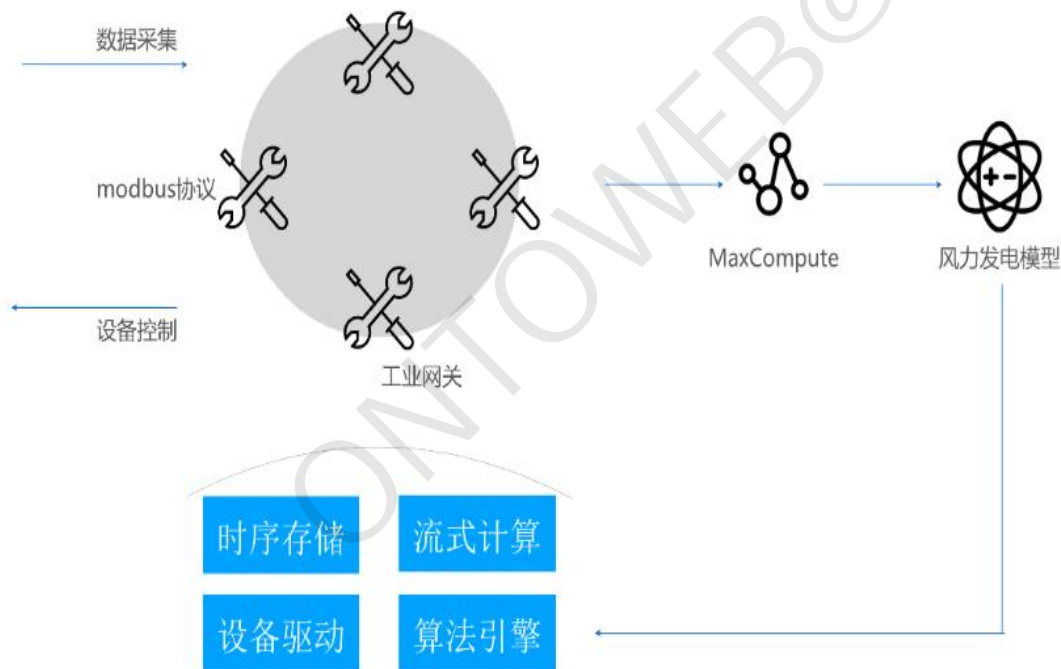
## 风力发电

在风力发电机组本地网络中，部署边缘计算网关，实时采集机组数据。在本地处理采集的数据后，先将数据上传至阿里云MaxCompute，再使用大数据训练模型后，对发电参数，如风向灵敏度、启动延时参数等做优化。将模型转化为算法或者规则导入本地边缘节点，自动调整风电机组参数，提高机组发电性能。



风力发电机组

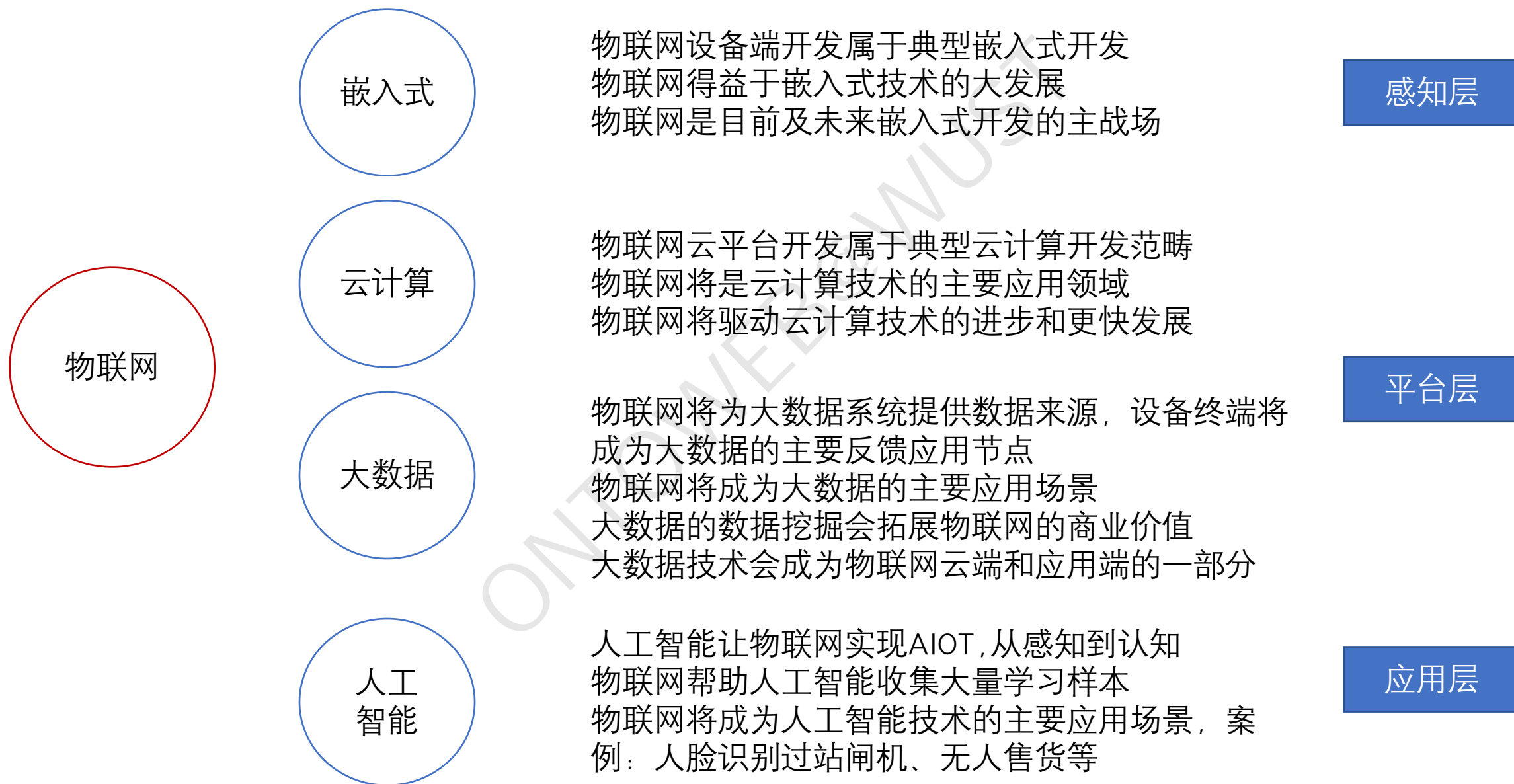
每个机组200个左右数据点位值



### 特点:

- ◆ 数据实时采集：多机组多数据点同时采集。
- ◆ 大数据处理：数据上传至阿里云后，使用大数据训练模型。
- ◆ 即时反馈：算法或规则导入本地边缘节点后，实时自动调整机组参数，实现最优化生产。

# 物联网与其他热点技术的关系





# 目录

01

物联网架构和相关技术

02

物联网云平台

1 协议

2 华为oc平台

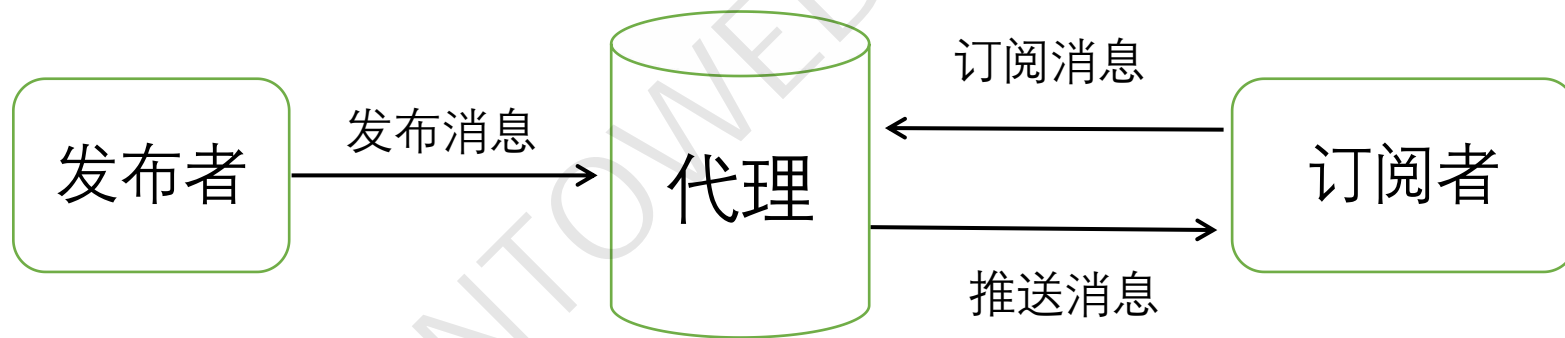
3 阿里物联网平台

03

物联网设备端开发

# MQTT协议

- Message Queuing Telemetry Transport Protocol, 消息队列遥测传输协议
- 一种基于轻量级代理的发布/订阅模式的消息传输协议, 能够提供有序、可靠的双向网络连接
- 以极少的代码和有限的带宽, 为远程设备提供实时可靠的消息服务
- 作为一种低开销、低带宽占用的即时通讯协议, 在物联网中应用广泛



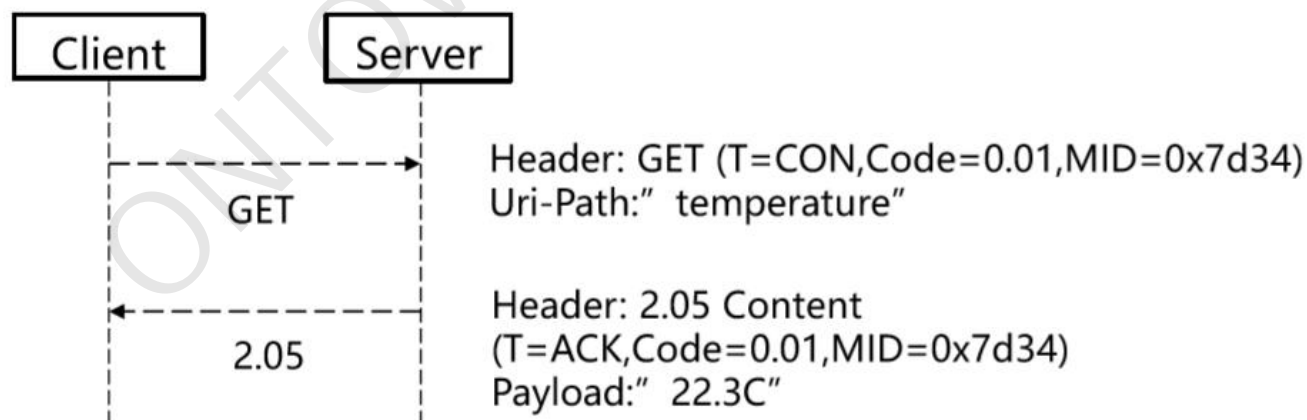
订阅者: 某些设备对特定的信息感兴趣, 并希望订阅它, 这些设备就叫订阅者

发布者: 负责为其他设备生成数据的就叫发布者

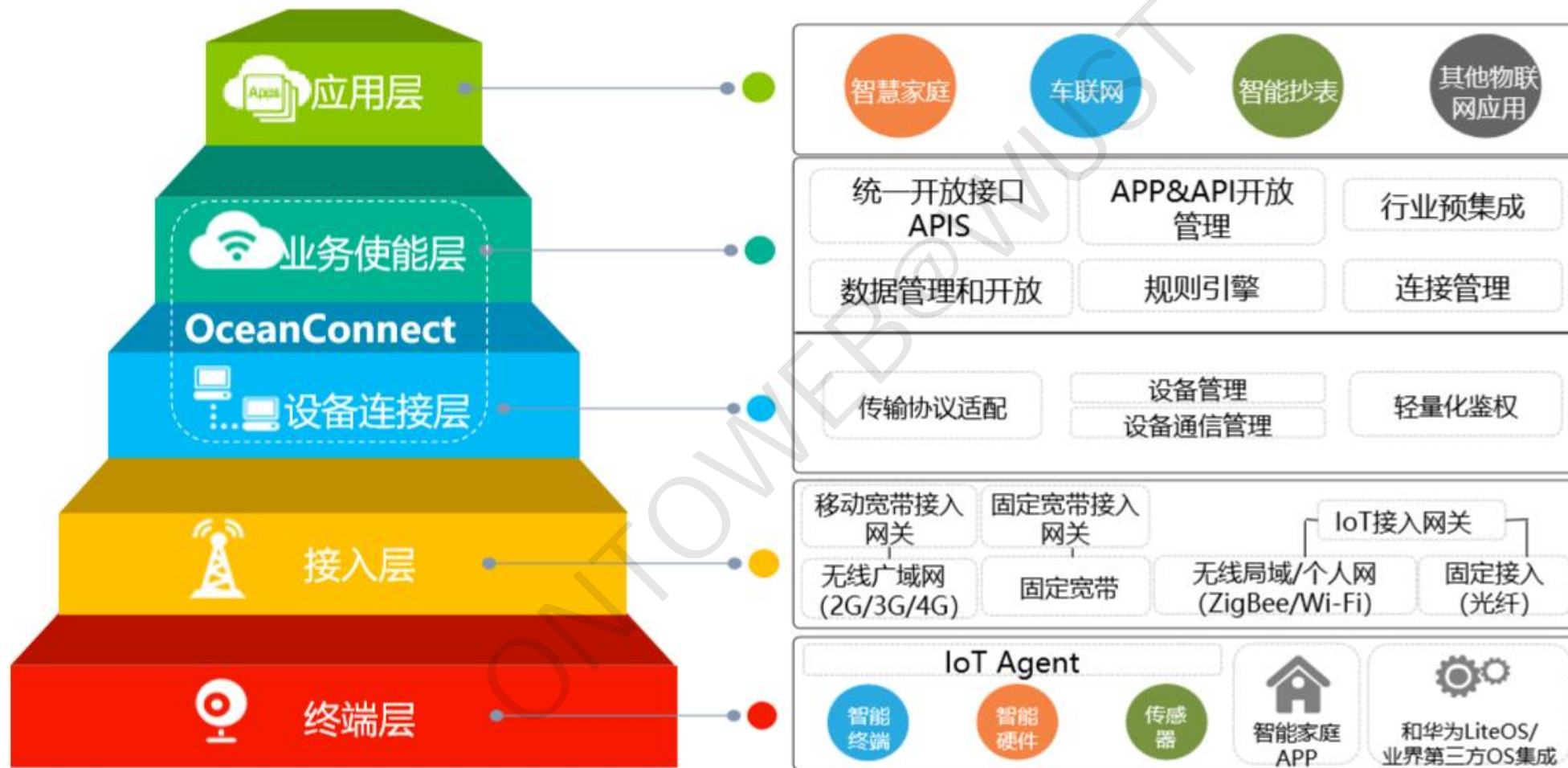
代理: 负责将来自发布者的消息进行存储处理, 并将这些消息发送到正确的订阅者中去

# CoAP协议

- **CoAP** (Constrained Application Protocol, 受限制的应用协议), 专门为资源受限设备 (如传感器节点) 和网络 (如NB-IoT, LoRa) 而设计。CoAP从HTTP协议发展而来, CoAP协议也是采用请求/响应工作模式, 客户端发起请求, 服务器做出响应。为了克服HTTP对于受限环境的劣势, CoAP既考虑到数据报长度的最优化, 又考虑到提供可靠通信。



# 华为云架构



<https://www.huaweicloud.com>

# 华为云业务使能



## 应用管理

NA (Network Application) 接入OceanConnect, 获取鉴权信息, 接入平台。随后携带鉴权信息调用其他应用场景。



## 设备资源管理

NA向平台申请网关, 获取网关的ID和验证码, 待网关完成安全接入OceanConnect流程, 设备将与APP建立从属关系。



## 设备管理

提供网关配置信息更新功能、上传设备的配置信息到OceanConnect等功能。



## 转发代理

NA给网关发送命令消息, 实现对传感器的实时控制。



## 规则

NA能够根据具体的业务需要, 制定规则 (条件) 和动作, 当网关的状态达到指定规则后, 以邮件或者短信的方式对用户进行实时提醒。



## 订阅通知

NA向平台订阅设备变更通知, 当设备发生变更时, 平台向NA推送此消息 (NA启动初始化订阅), NA根据通知类型进行对消息分派对应的服务模块进行处理。



# 华为云业务使能

## 触发条件

### 根据时间定义条件

如定义8:00钟关闭走廊的灯



### 根据设备上报的数据定义条件

如定义当温度计的温度值大于25度时打开空调



### 根据系统事件定义条件

如当系统生成Sensor异常告警时，发送通知到指定手机



## 响应动作

### 响应动作1

系统给传感器下发需要执行的动作，8:00准时关闭走廊的灯



### 响应动作2

系统给传感器下发需要响应的命令，温度值大于25度时打开空调

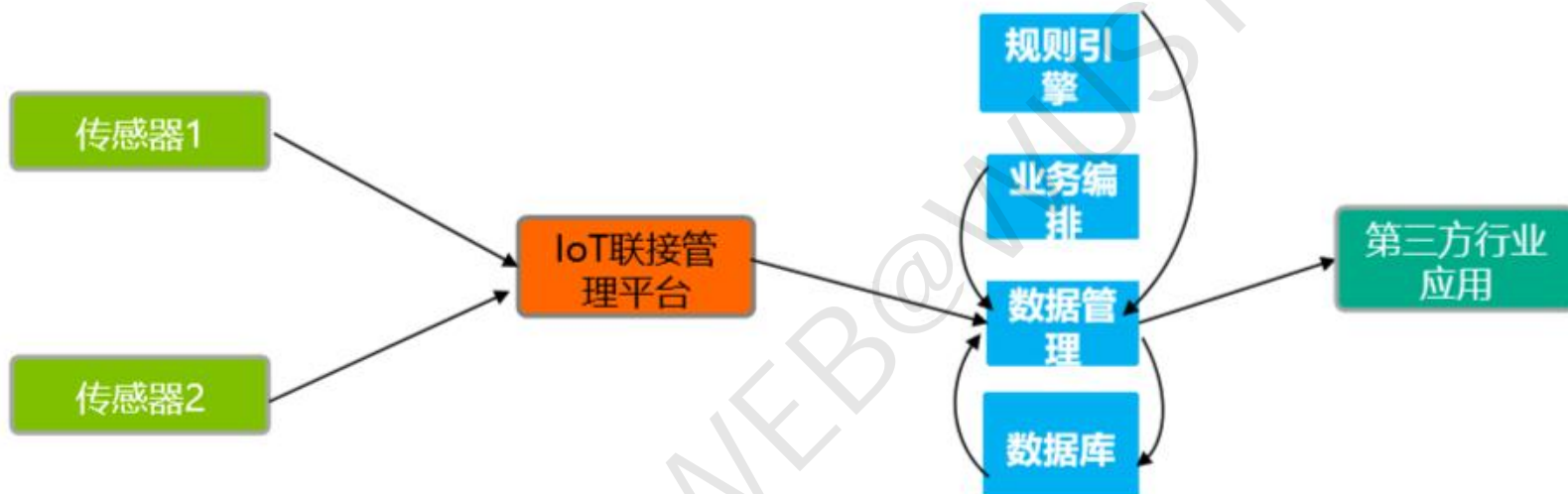


### 响应动作3

发送通知，包含邮件和短信的方式。Sensor异常告警时，发送通知到指定手机



# 华为云业务使能



## 数据格式转换（上报数据）

- 1、传感器1和传感器2上报的原始数据经过IoT联接管理平台后推送给IoT使能平台的数据管理模块。
- 2、数据管理模块根据传感器的类型、生产厂家、型号等信息将原始信息转换成通用数据格式存入数据库。
- 3、数据管理模块将转换后的数据上报给规则引擎、业务编排和第三方行业应用进行相应的业务触发。

注：传感器1和传感器2是不同厂家生产的相同类型的传感器。

## 数据格式转换（控制命令下发）

- 1、规则引擎、业务编排和第三方行业应用将产生的命令发送给IoT使能平台的数据管理模块。
- 2、IoT使能平台的数据管理模块通过向数据库查询目标传感器的生产厂家、型号、传感器类型等信息。
- 3、数据管理模块将数据转换成对应的命令通过IoT连接平台下发给不同的传感器。

# 设备连接演示

The screenshot displays the OceanConnect IoT platform interface. The top navigation bar includes the OceanConnect logo, the current project name 'hallo', and links to the document center, forum, and manufacturer information. The left sidebar contains a menu with options like 'Project Overview', 'Products', 'Product Development', 'Device Management', 'Upgrade/Debug', 'Applications', 'Connection Information', 'Subscription/Debug', 'Application Debug', 'Application Online Development', and 'Web Application Development'.

The main content area is titled 'Product Development > device'. It shows a progress bar with four steps: 01 Profile Definition, 02 Plugin Development (highlighted), 03 End-side Integration Guidance, and 04 Online Debugging. A 'Start Self-help Test' button is visible. Below the progress bar, there are tabs for 'Online Plugin Development', 'Plugin Management', and 'Online Plugin Deployment Success'. A '+ New Message' button is also present.

The central workspace shows a diagram with a box labeled 'light' (Dadi) and a list of data fields: 'messageId' and 'brightness'. A blue arrow points from the 'brightness' field to the 'light' box. The right sidebar contains a 'Device Model' section with a 'Dadi' model and a 'light' attribute, and a 'Details' section with a 'Dadi' entry.

Metadata for the device 'device' is displayed at the top of the main content area:

- 所属行业: 智慧城市
- 厂商名称: hl
- 数据格式: 二进制码流
- 协议类型: CoAP
- 型号: 0001
- 设备类型: Smoke
- 厂商ID: 72c0937c656d4a919aa92abb35d0c0e8

The 'light' device model details are as follows:

属性	命令
light	

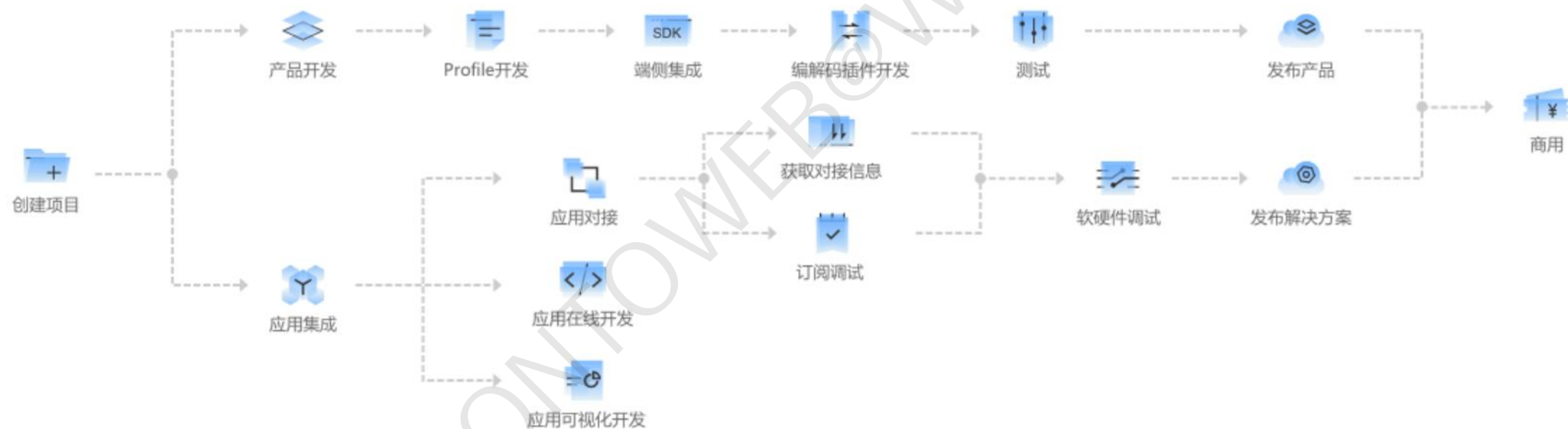
The 'Details' section shows the following information:

详情
Dadi



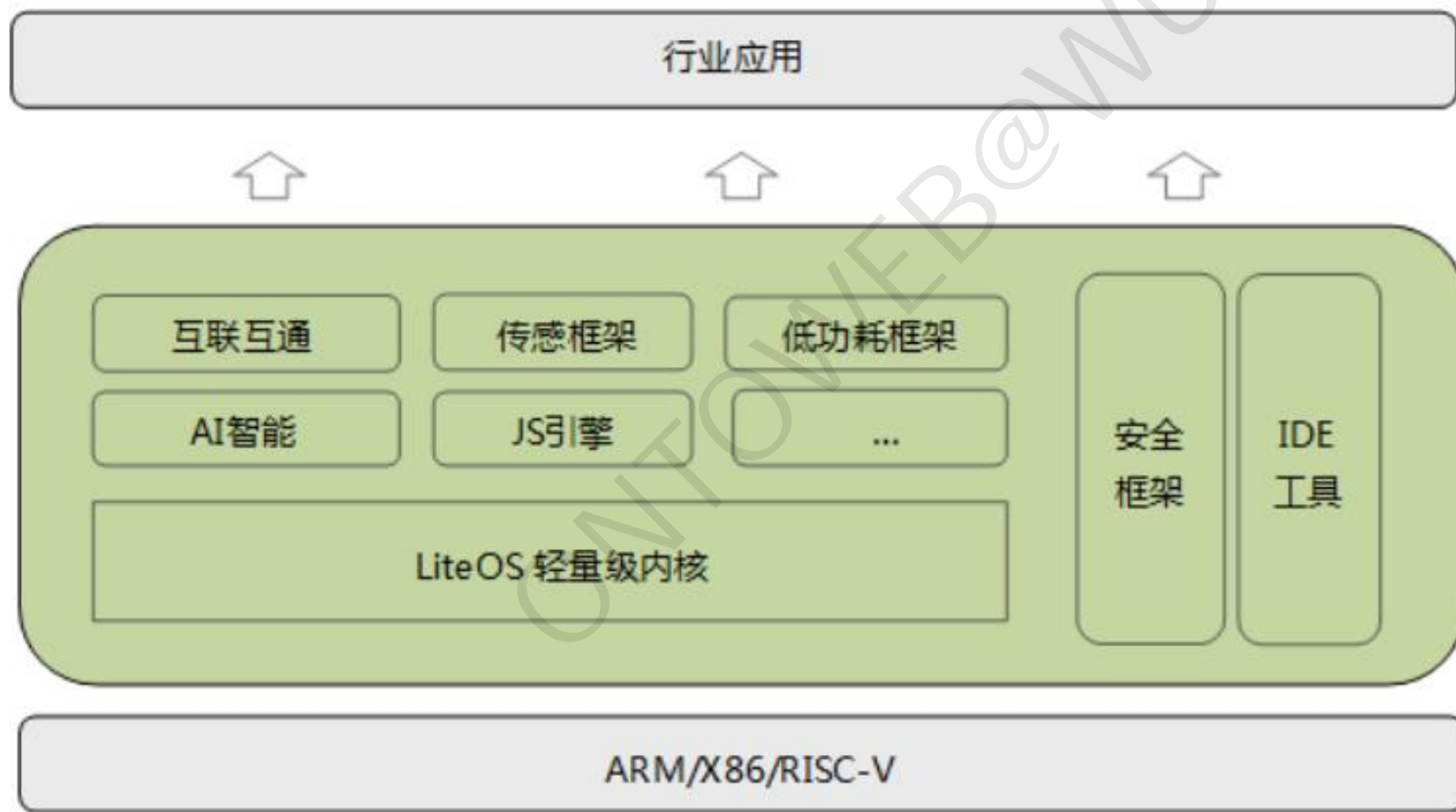
# 华为云

项目开发流程图



# 华为云LiteOs

Huawei LiteoS是华为面向IoT领域,构建的轻量级物联网操作系统,遵循BSD-3开源许可协议,大幅降低设备布置及维护成本,有效降低开发门槛、缩短开发周期。LiteoS开源项目目前支持ARM Cortex-M0, Cortex-M3, Cortex-M4, Cortex-M7等芯片架构。



应用场景:

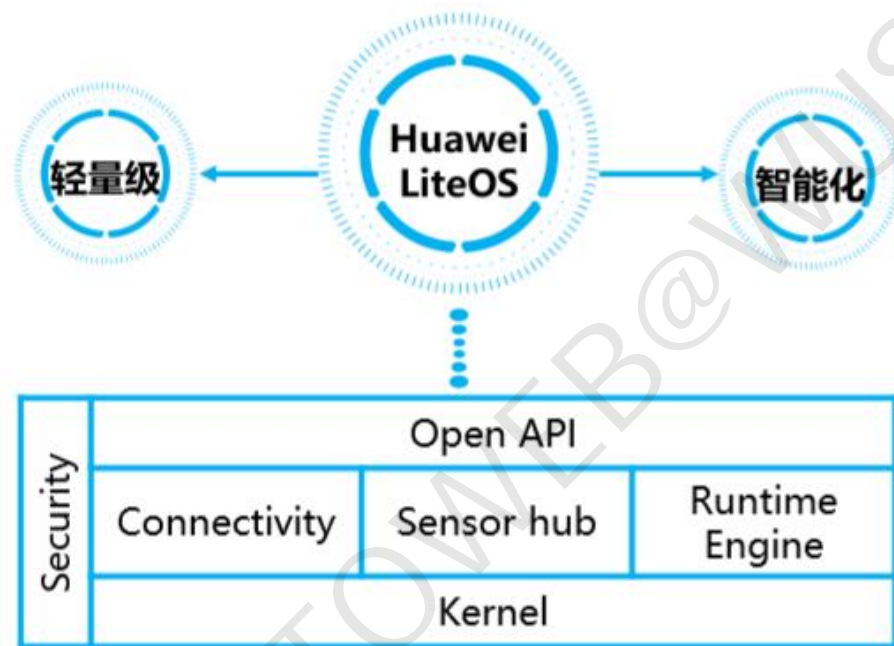
智能家居  
个人穿戴  
车联网  
城市公共服务  
制造业等领域

# 华为云LiteOs

[ KB级内核 ]

[ 毫瓦级功耗 ]

[ 微秒级响应 ]



**缩短终端产品上市时间**

- 行业协议栈支持连接和设备管理
- 支持华为及第三方平台
- 支持海思及第三方的芯片

**[ 传感智能化 ]**

- 降低时延
- 提升精确度

**[ 联接智能化 ]**

- 支持短距/广距多协议
- 自动组网

**[ 应用智能化 ]**

- 芯片级应用虚拟机

# 阿里云

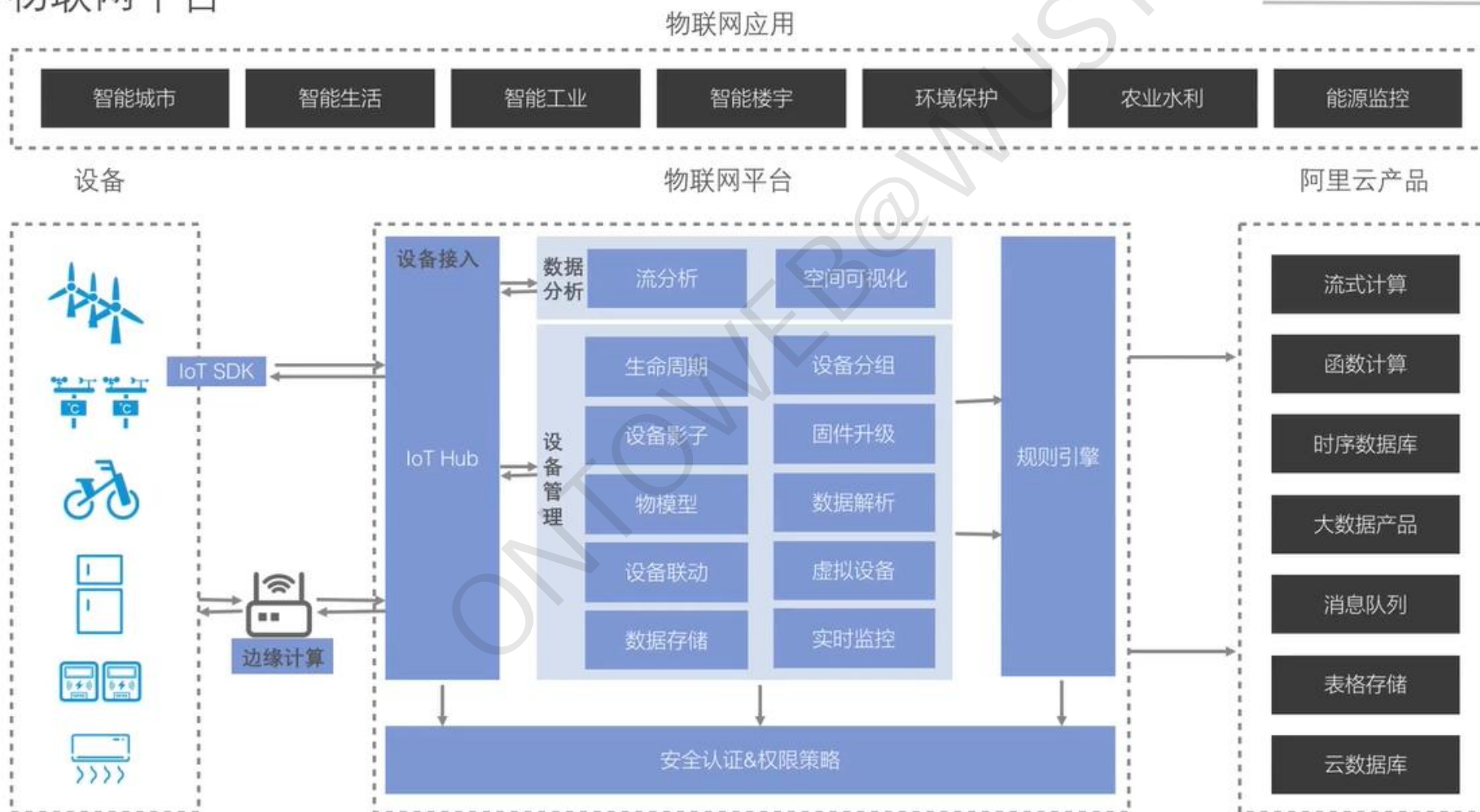


<https://iot.aliyun.co>

# 阿里云架构

## 物联网平台

Alibaba Cloud Link



# 阿里云

## 设备接入

物联网平台提供设备端SDK  
让设备轻松接入阿里云。

- 支持异构设备接入方案，提供2/3/4G、NB-IoT、LoRa等设备接入
- 提供MQTT、CoAP等多种协议的设备SDK，既满足长连接的实时性需求，也满足短连接的低功耗需求。
- 开源多种平台设备端代码，提供跨平台移植指导，赋能企业基于多种平台做设备接入。



# 规则引擎

## 物联网平台功能

### 规则引擎解析转发数据

将物联网平台与阿里云的产品无缝打通。

- 配置规则实现设备与设备之间的通信，快速实现M2M场景。
- 将数据转发到消息队列（MQ）中，保障应用消费设备上行数据的稳定可靠性。
- 将数据转发到表格存储（Table Store），提供设备数据采集 + 结构化存储的联合方案。
- 将数据转发到流计算（StreamCompute）中，提供设备数据采集 + 流式计算的联合方案。
- 将数据转发到TSDB，提供设备数据采集 + 时序数据存储的联合方案。
- 将数据转发到函数计算中，提供设备数据采集 + 事件计算的联合方案。

# 阿里云接入

## 设备接入方式介绍

使用阿里云官方提供的设备端SDK  
运行AliOS-Things物联网操作系统  
使用阿里云认证的通信模组  
使用开源MQTT, CoAP和HTTP协议自行开发

vscode安装 在扩展中搜索alios-studio 安装 C C++ 插件

python2.7.14 安装完成后可以cmd查询

python -V pip -V 版本

pip install --user setuptools 下载安装工具

pip install --user wheel wheel是编译后的文件 可以直接安装 相对于source

pip install --user aos-cube 安装alios 包含gnu

## 虚拟接入

vscode: 查看 控制台 终端

建立进入文件夹 npm install --save aliyun-iot-device-sdk



# Alios Things

AliOS Things是运行于智能硬件设备内的一种平台型软件。针对所适配的不同硬件资源和功能需求，Alios Things分为RTOS版本和Linux版本两个不同的分支，用于服务不同的应用场景

## 针对对象

### 传统硬件/设备开发商

- 实现企业自用物联网硬件产品的开发。如物联网水质监控设备，物联网农业灌溉设备。
- 实现可二次开发的智能硬件的开发或定制。

### 互联网IOT应用开发者

- 向传统厂商定制智能硬件，并通过ALIOS提供的标准接口进行二次开发，形成支撑物联网应用的硬件产品。

## 极简开发

基于Linux之上的轻量虚拟化环境，开发者直接在 Linux 平台上开发硬件无关的 IoT 应用和软件库，使用GDB/Valgrind/SystemTap等 PC 平台工具诊断开发问题

提供 IDE，支持系统/内核行为 Trace，Mesh 组网图形化显示  
通过 Alink 与阿里云计算 IoT 服务无缝连接

## 彻底全面的安全保护

提供系统和芯片级别安全保护  
支持可信运行环境(支持 ARMV8-M Trust Zone)  
支持预置 ID2 根身份证和非对称密钥以及基于 ID2 的可信连接和服务

# 在比邻内部搭建物联网云开发平台

## 物联网信息采集网关

建议使用EMQX (<https://www.emqx.io/>)  
如何与比邻集成平台对接?

## 时序数据库

建议使用TDEngine (<https://www.taosdata.com/cn/>)

二者有一定重合，如何将上述两个部分与比邻集成平台对接是需要考虑的。

## 时序数据库

建议开发人员熟悉Erlang语言。

# 目录

01

物联网架构和相关技术

02

物联网云平台

03

物联网设备端开发

1 *ARM架构*

2 *Linux 系统介绍*

3 *开发流程*

# 操作系统



10亿连接

互联网时代  
x86架构电脑

**Windows**



100亿连接

移动互联网时代  
ARM架构手机

**Android/iOS**



1000亿连接

物联网时代  
x86/ARM/DSP/MIPS/FPGA...架构终端

**物联网操作系统**

ARM即英国ARM (Advanced RISC Machines), ARM公司只对外提供ARM内核, 各大厂商 (Atmel、TI、飞思卡尔、NXP、ST、和三星等) 在授权付费使用ARM内核的基础上研发生产各自的芯片, 促进了嵌入式和移动端ARM, CPU的发展。

# ARM架构

## 对比介绍:

ARM 微处理器目前包括下面几个系列，除了具有ARM 体系结构的共同特点以外，每一个系列的ARM 微处理器都有各自的特点和应用领域。其目前有Classic系列、Cortex-M系列、Cortex-R系列、Cortex-A系列和Cortex-A50系列5个大类。

**Classic系列：**该系列处理器由三个子系列组成：ARM7系列：基于ARMv3或ARMv4架构。ARM9系列：基于ARMv5架构。ARM11系列：基于ARMv6架构

**Cortex-M系列：**该系列处理器包括Cortex-M0、Cortex-M0+、Cortex-M1、Cortex-M3、Cortex-M4共5个子系列。该系列主要针对成本和功耗敏感的应用，如智能测量、人机接口设备、汽车和工业控制系统、家用电器、消费性产品和医疗器械等。

**Cortex-R系列：**该系列处理器包括Cortex-R4、Cortex-R5、Cortex-R7共3个子系列。面向如汽车制动系统、动力传动解决方案、大容量存储控制器等深层嵌入式实时应用。

**Cortex-A系列：**该系列处理器包括Cortex-A5、Cortex-A7、Cortex-A8、Cortex-A9、Cortex-A12和Cortex-A15共6个子系列，用于具有高计算要求、运行丰富操作系统及提供交互媒体和图形体验的应用领域，如智能手机、平板电脑、汽车娱乐系统、数字电视等。

**Cortex-A50系列：**基于ARMv8架构，允许在32位和64位之间进行完全的交互操作。

# ARM架构

ARM处理器最主要的特点是：耗电少功能强、16位/32位双指令集和合作伙伴众多。具体来说包括以下几个方面：

- 体积小、低功耗、低成本、高性能；
- 支持Thumb(16位)/ARM(32位)双指令集，能很好的兼容8位/16位器件；
- 大量使用寄存器，并且大多数数据操作都在寄存器中完成，指令执行速度更快；
- 寻址方式灵活简单，执行效率高；
- 指令长度固定，可以通过多流水线方式提高处理效率

缺点：ARM要在性能上接近X86，频率就要比X86处理器高，这就会造成高能耗。

优点：

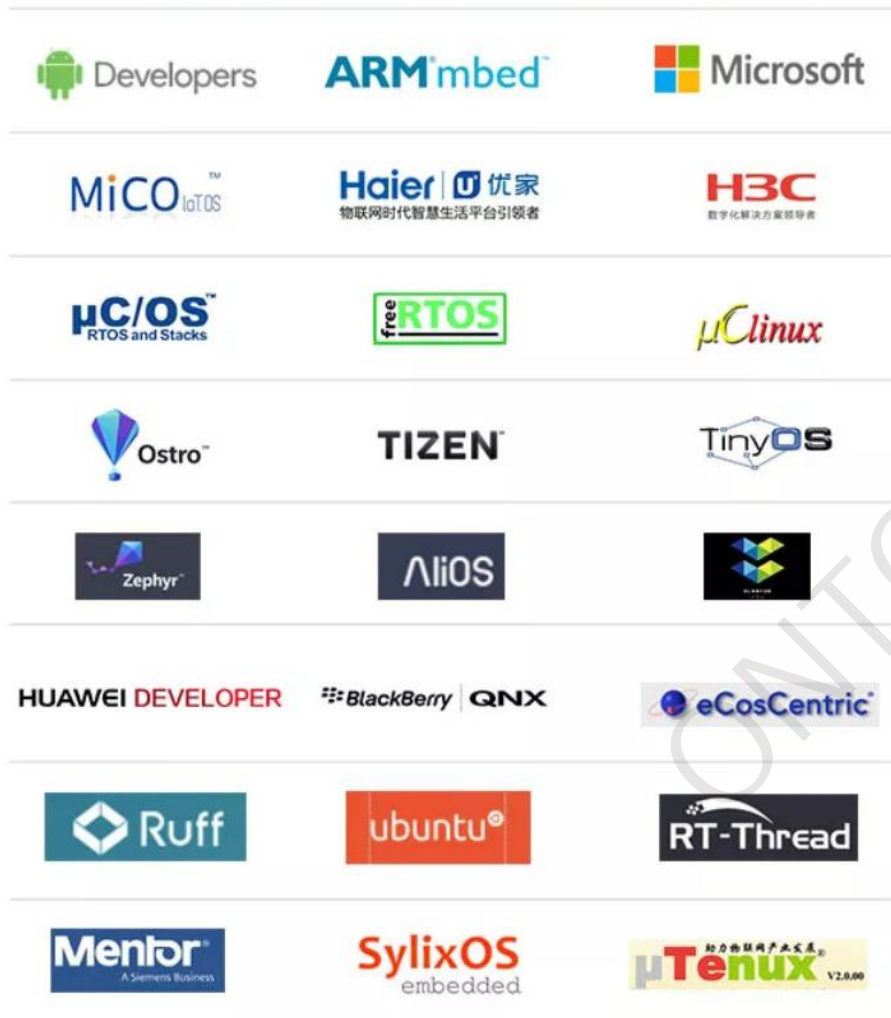
物联网最大价值之一就是通过大量的传感器收集所需数据，通过对这些数据进行计算后再提取有价值的数据，因此物联网的首要前提就是广泛的智能终端设备与传感器的连接。

芯片是实现物物相连的关键，物联网的芯片偏重于解决低功耗、高整合度，低功耗连接使开发人员能够为功耗受限的设备添加更多功能，同时保持尺寸小巧，从而扩大了其应用可能性。添加集成度越来越高的元件，通过即插即用方案简化新应用的开发，快速将新设备推向市场。而这些恰恰正是ARM的优势。



# 物联网操作系统

## 物联网操作系统一览图



出于成本或者定制化的需求，开发者往往更青睐开源的物联网操作系统。开源系统可以大致分为两大类：

- 一类是基于Linux系统改造的物联网OS，如uCLinux、谷歌的Brillo、华为的Lite OS等；
- 一类是非Linux类物联网OS，由开源社区或商业公司按照某种开源许可独立设计实现，如uCOS、FreeRTOS、mbed等。

从技术层面看，物联网操作系统仍处在研发阶段，至今尚无一个比较完善的可商业应用的物联网标准操作系统，物联网时代最大的问题之一就在于沟通，不同设备之间不同标准的沟通、软件与硬件的沟通。

# 物联网操作系统

ARM®mbed™ 操作系统是一种专为物联网(IoT) 中的“物体”设计的开源嵌入式操作系统。该操作系统包含基于ARM Cortex-M 微控制器开发连接产品所必需的全部功能，非常适合涉及智能城市、智能家庭和穿戴式设备等领域的应用程序。

简单来说，Mbed 是一个开发平台，一个基于ARM cortex M 系列的单片机开发平台。主要特点：开发速度快，功能强大，安全性高，为了量产化而设计，可离线开发，也可以在网页上编辑。

Android Things 是谷歌为Google Brillo 更改名称后的新版系统，后者是谷歌在2015年宣布的一款物联网操作系统。尽管Brillo 的核心是Android 系统，但是它的开发和部署明显不同于常规Android 开发。Brillo 把C++ 作为主要开发环境，而Android Things 则面向所有Java开发者，不管开发者有没有移动开发经验。从硬件资源的角度来看，Android Things 属于土豪级的系统，动辄上百MB 的内存显然不适合单片机，这也正常，因为它的主要竞争对手其实是Windows 10 IoT。

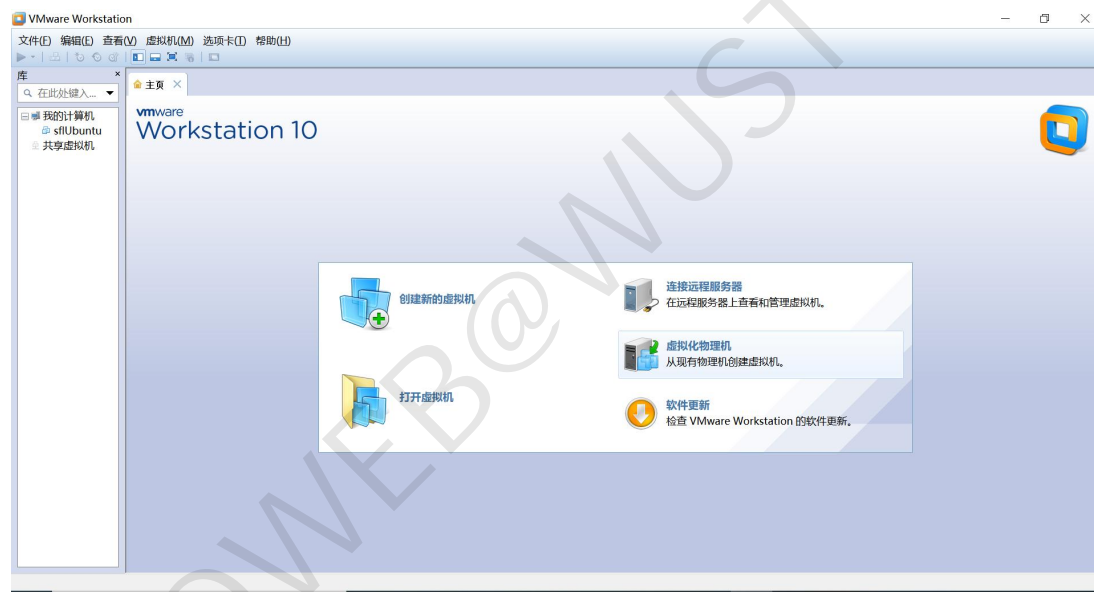
Windows 10 IoT 是面向各种智能设备的Windows 10 版本系列，涵盖了从小的行业网关到大的更复杂的设备（如销售点终端和ATM），种类繁多。结合最新的Microsoft 开发工具和Azure IoT 服务，合作伙伴可以收集、存储和处理数据，从而打造可行的商业智能和有效的业务结果。在构建基于Windows 10 IoT 的解决方案后，合作伙伴将在利用一系列Microsoft技术提供端到端的解决方案时发现更多机会。



# Linux

## 软件安装

虚拟机 vmware



Linux环境:



Ubuntu（又称乌班图）是一个以桌面应用为主的开源GNU/Linux操作系统，Ubuntu 是基于Debian GNU/Linux，支持x86、amd64（即x64）、ARM和ppc架构。

# 常用命令

ls list 列出当前目录下文件 ls -a / ls  
-l (详细信息)  
cd 打开文件夹 cd .. 回到上一层目录  
touch 创建文件 . 隐藏文件 加. 可以隐藏该文件  
pwd 显示工作目录的绝对路径  
mkdir 创建目录  
mv 移动文件 文件重命名  
cp 复制文件 cp -r 复制目录  
rm 删除文件 rmdir 删除目录  
cat 打印

# 传感器

物联网系统中的海量数据信息来源于终端设备，而终端设备数据来源可归根于传感器，传感器赋予了万物“感官”功能，如人类依靠视觉、听觉、嗅觉、触觉感知周围环境，同样物体通过各种传感器也能感知周围环境。且比人类感知更准确、感知范围更广。例如人类无法通过触觉准确感知某物体具体温度值，也无法感知上千高温，也不能辨别细微的温度变化。



传感器定义为将物理、化学、生物等信息变化按照某些规律转换成电参量(电压、电流、频率、相位、电阻、电容、电感等)变化的一种器件或装置。

- ◆ 传感器一般由**敏感元件**、**转换元件**和基本电路组成。敏感元件是直接感知被测量的元件，将被测量变化转换成该敏感材料特性参数的变化。某些敏感元件为无源器件，无法直接输出电压或电流，所以需要通过转换元件特性参数的变化转换成电压或电流。基本电路将转换元件输出的信号进行放大、整形及编码输出。
- ◆ 传感器种类繁多，按照被测量类型可分为温度传感器、湿度传感器、位移传感器、加速度传感器、压力传感器、流量传感器等。按照传感器工作原理可分为物理性传感器(基于力、热、声、光、电、磁等效应)、化学性传感器(基于化学反应原理)和生物性传感器(基于霉、抗体、激素等分子识别)。以下将简述基于物理性的常见传感器工作原理。

# 开发准备

## 硬件板子

stm32l476

## Keil uVersion 5.0

Keil uVersion 5.0（后文简称为Keil 4.0）是单片机程序开发的集成开发环境（IDE），集成了C编译器、宏汇、连接器、库管理和一个功能强大的仿真调试器。

STM32CubeMX是ST意法半导体的主动原创工具。STM32CubeMX集成了一个全面的软件平台，支持STM32每一个系列的MCU开发。这个平台包括 STM32Cube HAL（一个 STM32的抽象层集成软件，确保STM32 系列最大的移植性），加上兼容的一套中间件（RTOS、USB、TCP/IP 和图形），所有内嵌软件组件附带了全套例程。

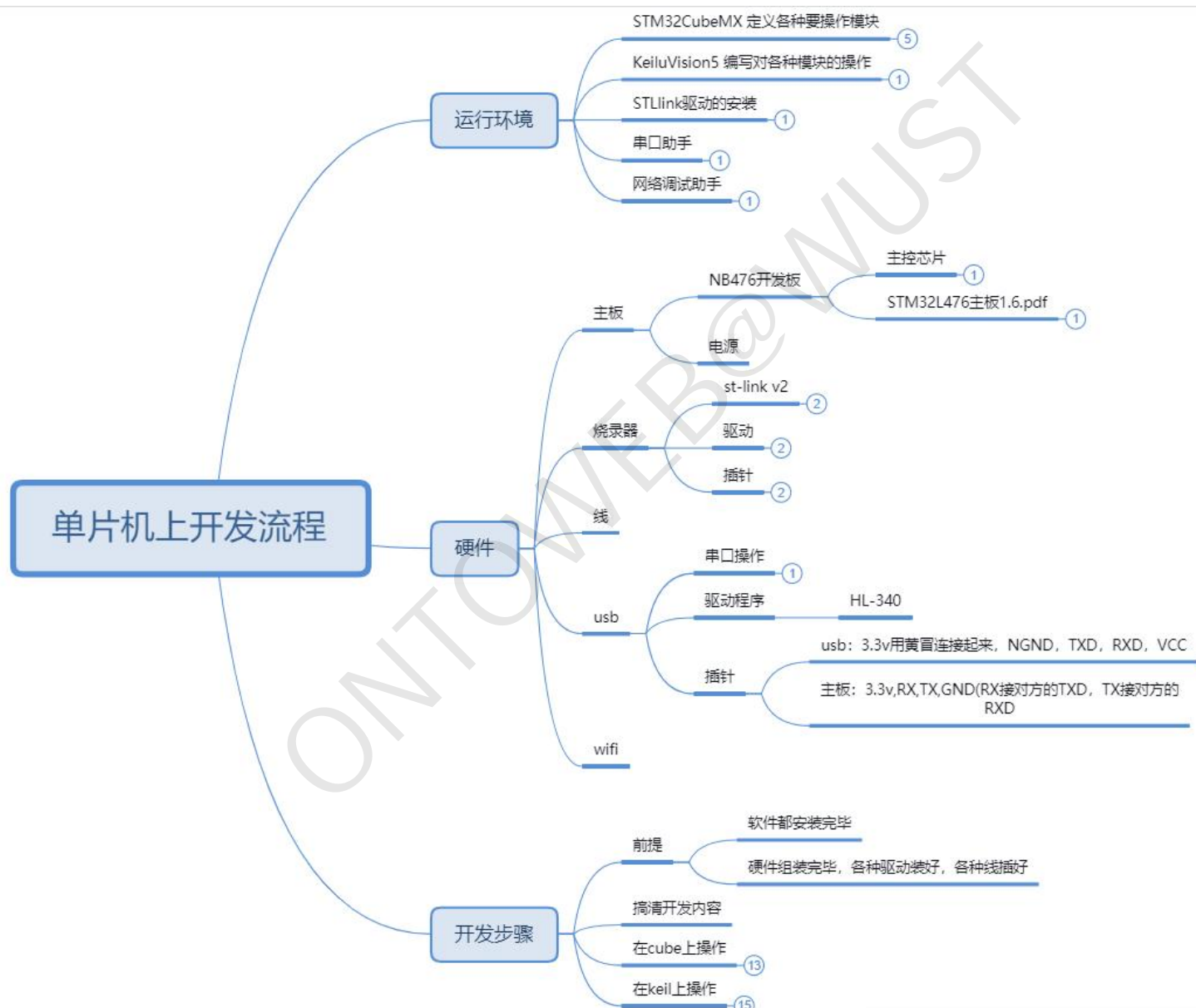
利用可视化界面来进行STM32p配置，所以时钟、滴答定时器、DMA、串口、GPIO等就不用根据数据手册去操作标准库甚至是寄存器了。

## stm32cubemx

STM32CubeMX的特性如下：

- 1.直观的选择 STM32 微控制器。
- 2.微控制器图形化配置：
  - ┆ 自动处理引脚冲突
  - ┆ 动态设置确定的时钟树
  - ┆ 可以动态确定参数设置的外围和中间件模式和初始化
  - ┆ 功耗预测
- 3.C代码工程生成器覆盖了STM32 微控制器初始化编译软件，如IAR、KEIL、GCC。
- 4.可独立使用或作为 Eclipse 插件使用。

# 开发流程



# 认证相关

## HCIA-IoT

最后更新时间: 2019-07-23 16:48:06 点击数: 43619次

### 认证概述

**HCIA-IoT认证**定位于物联网技术普及、物联网基础开发与运维能力的构建、考核和认证。

**HCIA-IoT认证**包括：物联网概述、常见物联网通信方式、常见物联网解决方案介绍（公共事业物联网、车联网、能源工业物联网、智慧家庭物联网）、物联网安全、IoT联接管理平台介绍、物联网操作系统介绍、工业物联网网关介绍、家庭物联网网关介绍、NB-IoT标准及解决方案介绍、华为eLTE-IoT解决方案介绍等。

**通过HCIA-IoT认证**，将证明您掌握了物联网基础知识体系，理解了华为物联网典型产品与解决方案，具备了物联网的基础开发与运维能力。

**拥有通过HCIA-IoT认证的工程师**，意味着企业掌握了运用物联网技术，进行物联网基础开发与运维能力。

### 认证前提条件

无



谢谢观看