振动边缘场景方案设计

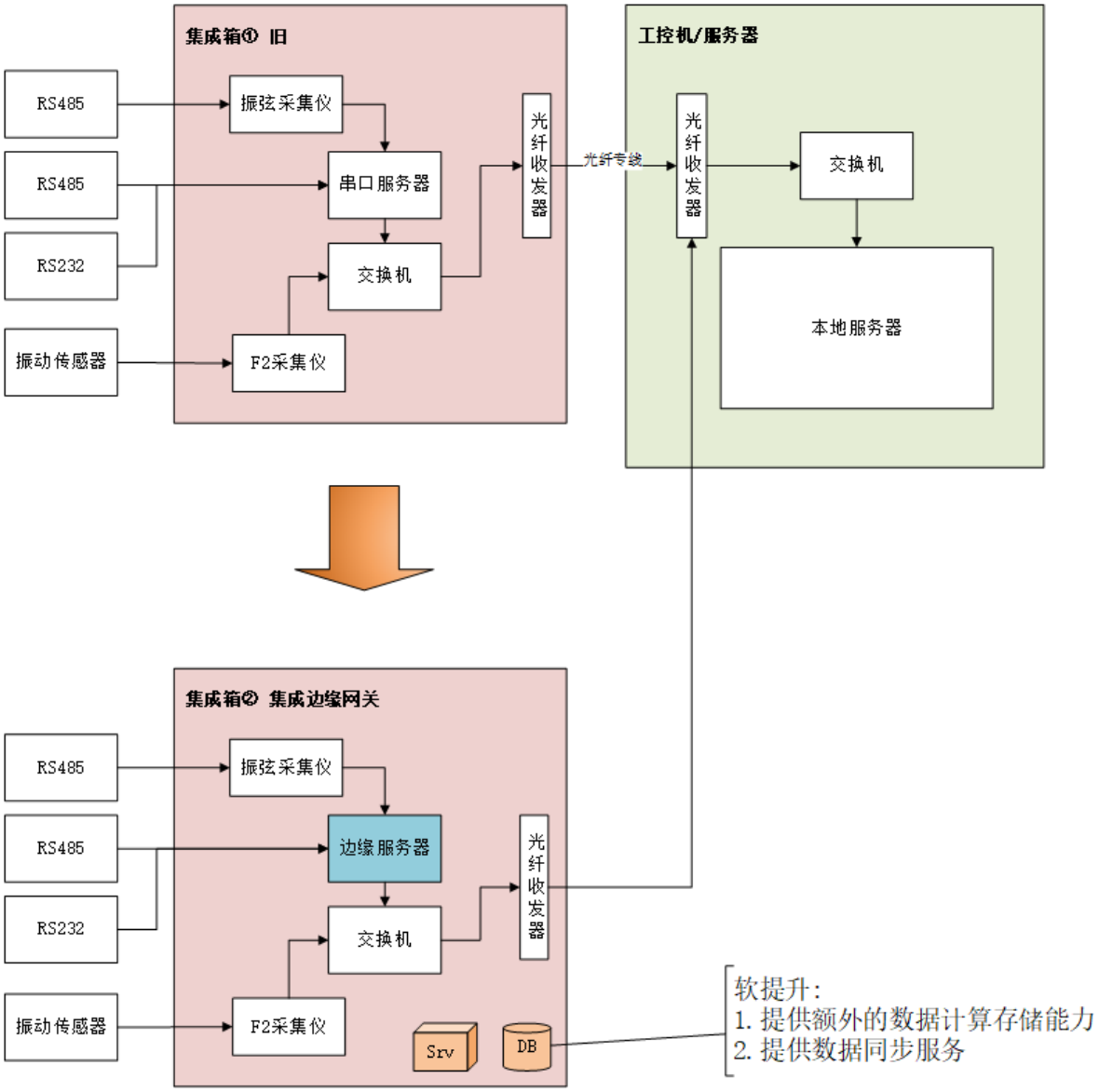
边缘场景

在结构物安全监测，包含物振动监测的场景中，将边缘网关将位于现场组网端（最接近“端”的位置）（一般会集成在采集箱内），通过边缘系统实现数据采集、计算和存储：

边缘场景

在结构物安全监测，包含**振动**监测的场景中，边缘网关将位于现场组网端（最接近“端”的位置）（一般会集成在采集箱内），通过边缘系统实现数据采集、计算和存储：

> 对现场组网的理解可能存在偏差，大概示意如图



可以看出，原来现场的配电箱内除了电源控制外，剩下的是采集控制（采集仪）和传输控制器（包含交换机、光纤收发器等），统一可以看作是对**信号量的调制过程**，即模拟信号->数字信号->光信号。

如果对应人体，就相当于一套**神经传输系统。**

边缘概念中，我们通常把边缘系统比做**章鱼。**章鱼有**40%**神经元在脑袋里,剩下的**60%**在它的8条腿上,所谓的**用“腿”思考。**

如集成箱②中描述，其中边缘服务器就类似这一区域内传感系统的“**大脑**”。

边缘计算的基本思想则是**功能缓存(function cache)**，是大脑功能的延生。 边缘计算是云计算的延生和补充。

边缘的大脑负责：

- 收集和转发数据 (**信息传导**)
- 数据存储（**记忆功能**）
- 分析和反馈（**思考功能**）

这样的优势显而易见：

👍 分布式和低延时计算

👍 效率更高、更加智能（AI）、更加节能

👍 缓解流量压力、云端存储压力

👍 隐私保护，更加安全

振动边缘设计

拟在`linux`嵌入式板上实现`golang`开发的边缘服务，实现对振动F2采集仪的数据**采集控制、计算、存储和传输**功能。

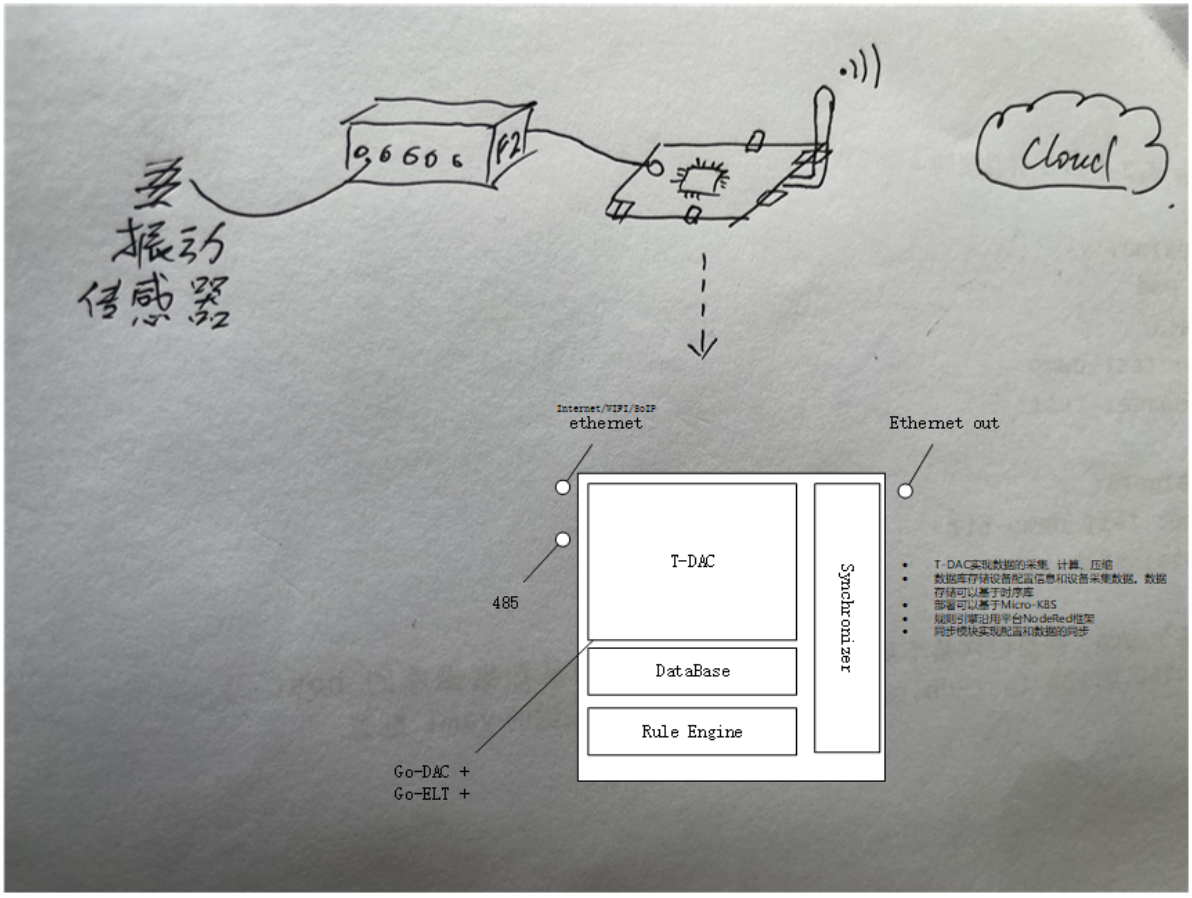
技术选型

硬件：跟硬件同事讨论后，选择了飞凌公司的OK1028A开发版，配置如下

软件：编程语言选择了目前以太DAC一致的Golang，开发Linux环境程序。

部署：考虑使用`MicroK8S`在板子上部署采集程序和其他服务（数据库/消息组件）

整体设计



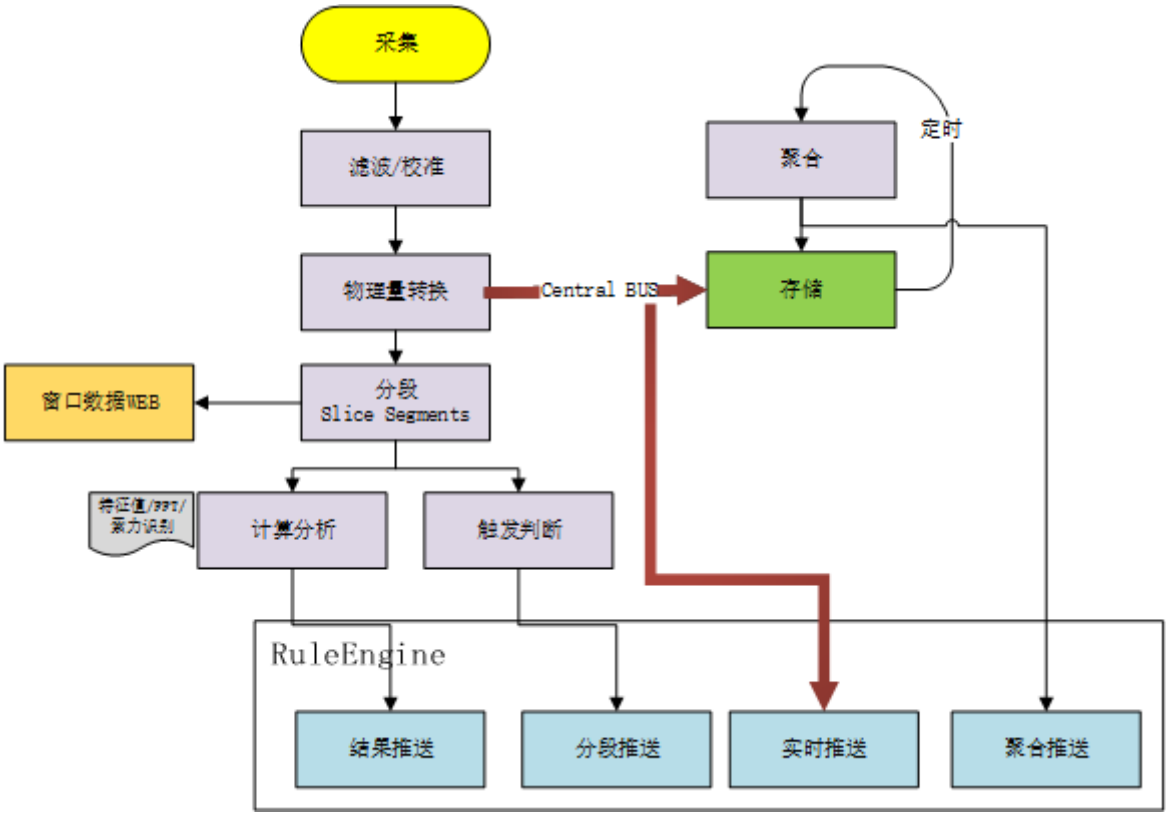
如上，框架在《边缘网关的一些思考》中已经初步介绍，我们在开发板中，至少需要集成：

- 数据采集和处理程序(T-DAC)，这里主要是Go-DAAS负责采集振动数据
- 配置同步控制程序
- 数据库服务
- 规则引擎服务

数据流程

数据从采集开始，经过**ET**过程(图中滤波、校准、物理量计算)之后， 主要分三个主流向：

1. 存储到数据库
2. 计算分析后存储和上报平台
3. 自动聚合(压缩)后的数据上报平台



具体模块说明：

1. 滤波/校准

   校准去直流、滤波算法

2. 物理量计算

   输出电压值到监测物理量值转换

3. 分段

   设置窗口大小、刷新时间，将数据进行分段

4. 窗口数据WEB

   实现http服务提供实时窗口振动数据(类似DAAS实时采集展示)。并通过wobsocket实现实时更新

5. 计算分析

   通过设置的算法，实现特征值(TMRS)、FFT、索力识别对计算

6. 触发判断

   通过设置的触发条件（定时/信号量），对连续信号进行采样保存 （同DAAS采样生成.odb数据文件）

7. 存储

   数据存储到边缘数据库。

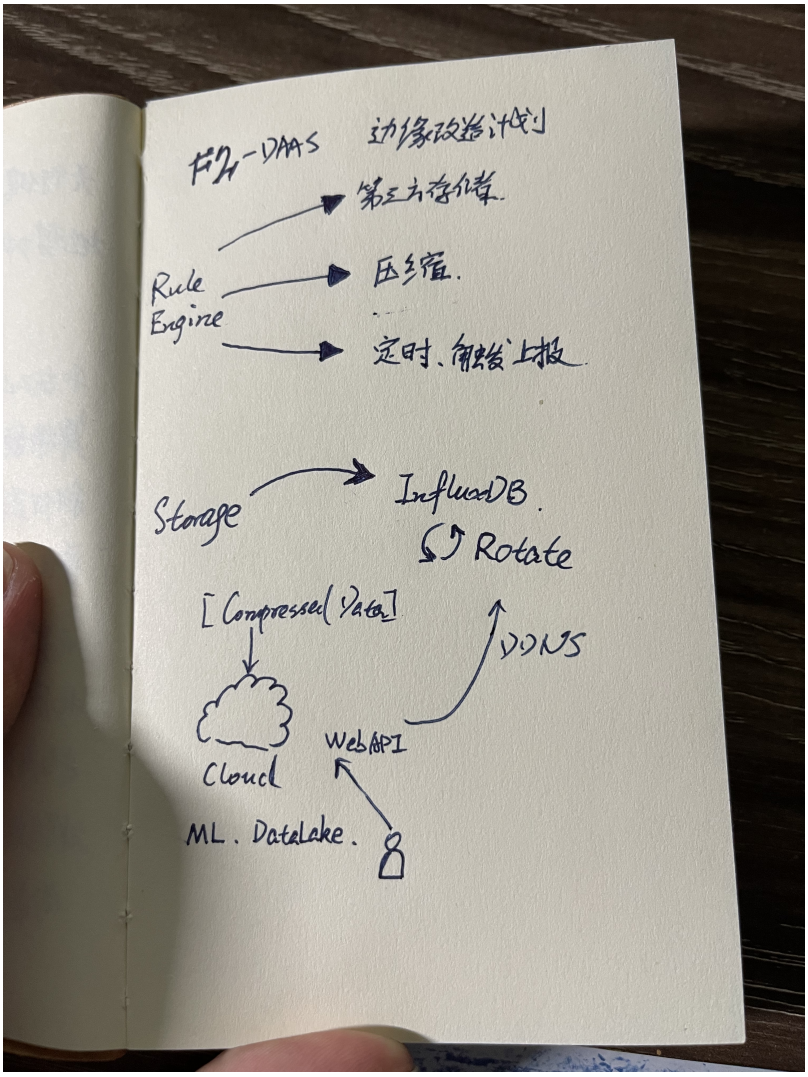
8. 聚合

   数据库中定时生成1s/10s等统计数据

9. 推送

   通过规则引擎实现

关于边缘网关**提供数据**的形式，初步思考：可以通过DDNS或NAT内网穿透，通过端的WEB-API服务向外提供。



采集模块

采集模块的设计稿如下，主要是设备连接控制的过程 。

Session：管理TCP连接会话

VbServer：设备控制主服务，包括管理连接、配置读取、配置设置、启动采集和数据回调

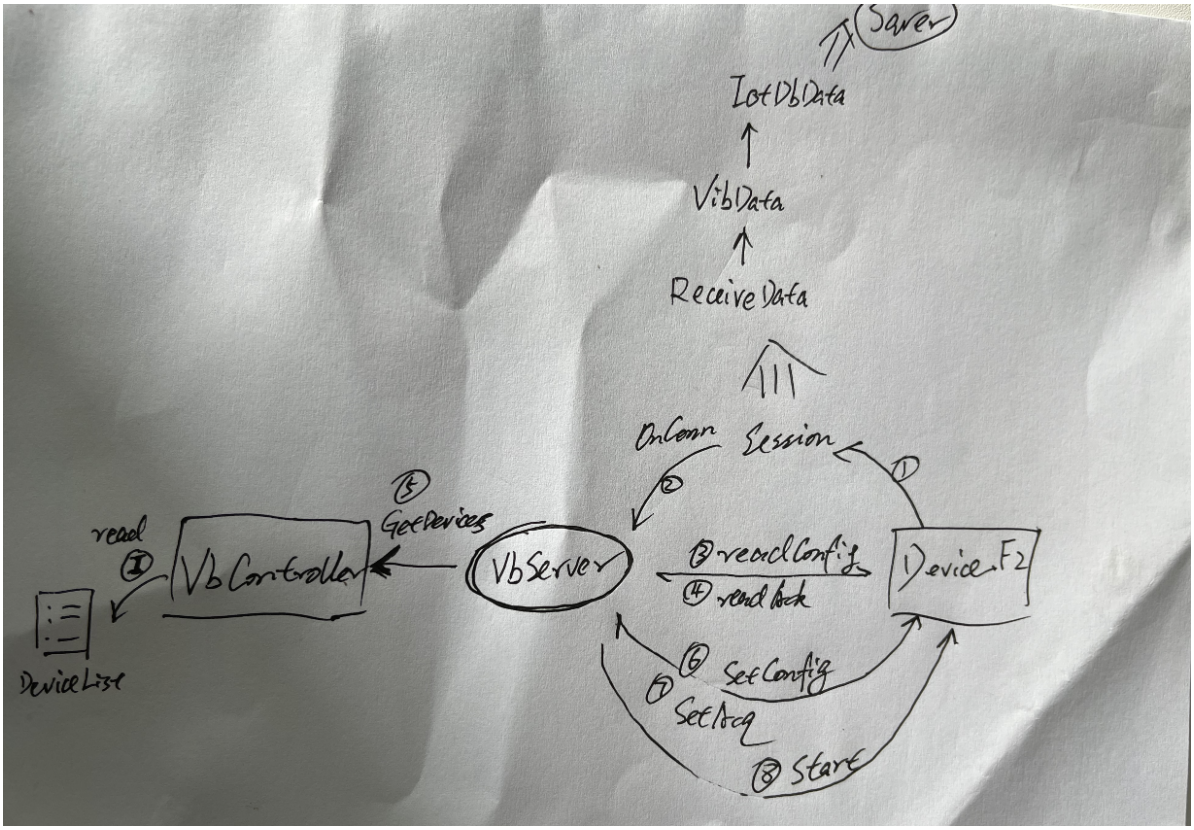
VbController: 负责设备配置和处理流程控制

数据模型：

ReceiveData: 参考C#DAAS，Session上的原始信号数据

VibData：转换后的振动数据

IotDbData: 待入库格式数据



存储设计

在 《边缘数据库选型》中对比了几种数据库，初步拟定使用`InfluxDB`作为边缘端存储引擎。

目前仅考虑振动数据的存储。存储到数据格式如下：

```
bucket : data
measurement: vib
tags: id (设备id)
fields: phy (物理量值)
```

通过配置持续聚集（CQ）实现数据的自动聚合

```
create database data_10sec_agg;

-- 数据库 ${db}

-- 10s持续聚集 每10s执行 (FOR 1min)允许数据晚到1min内
CREATE CONTINUOUS QUERY "cq_ten_sec" ON "vib"
RESAMPLE EVERY 10s FOR 1m
BEGIN
  SELECT mean(*),max(*),min(*),spread(*),stddev(*) INTO
"data_10sec_agg"."autogen".:MEASUREMENT FROM /.*/ GROUP BY time(10s),*
END;
```

同步系统

TODO

推送系统

TODO