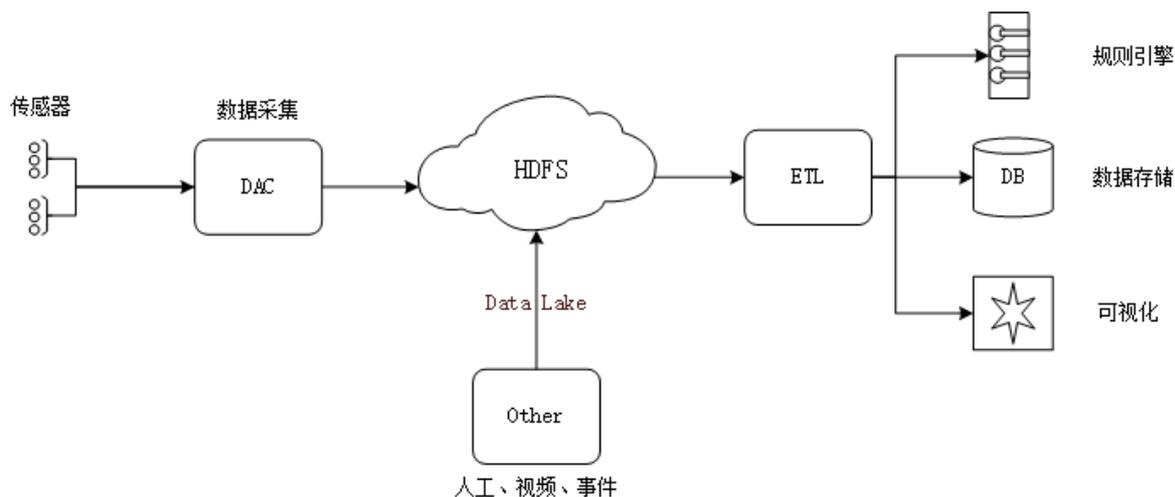


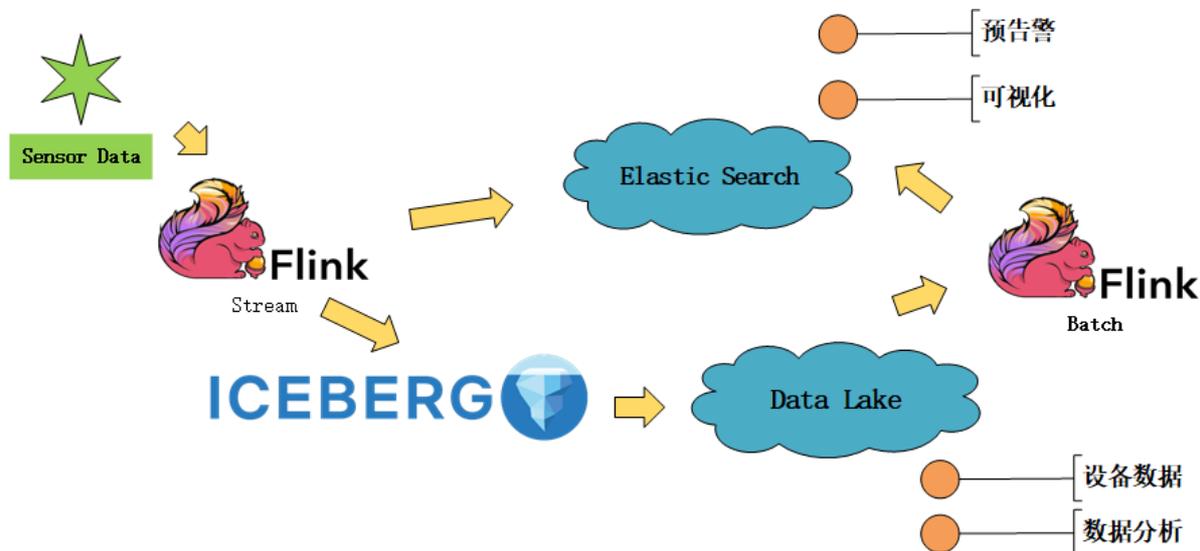
后端整体架构设计

感知平台后端整体架构设计如图所示



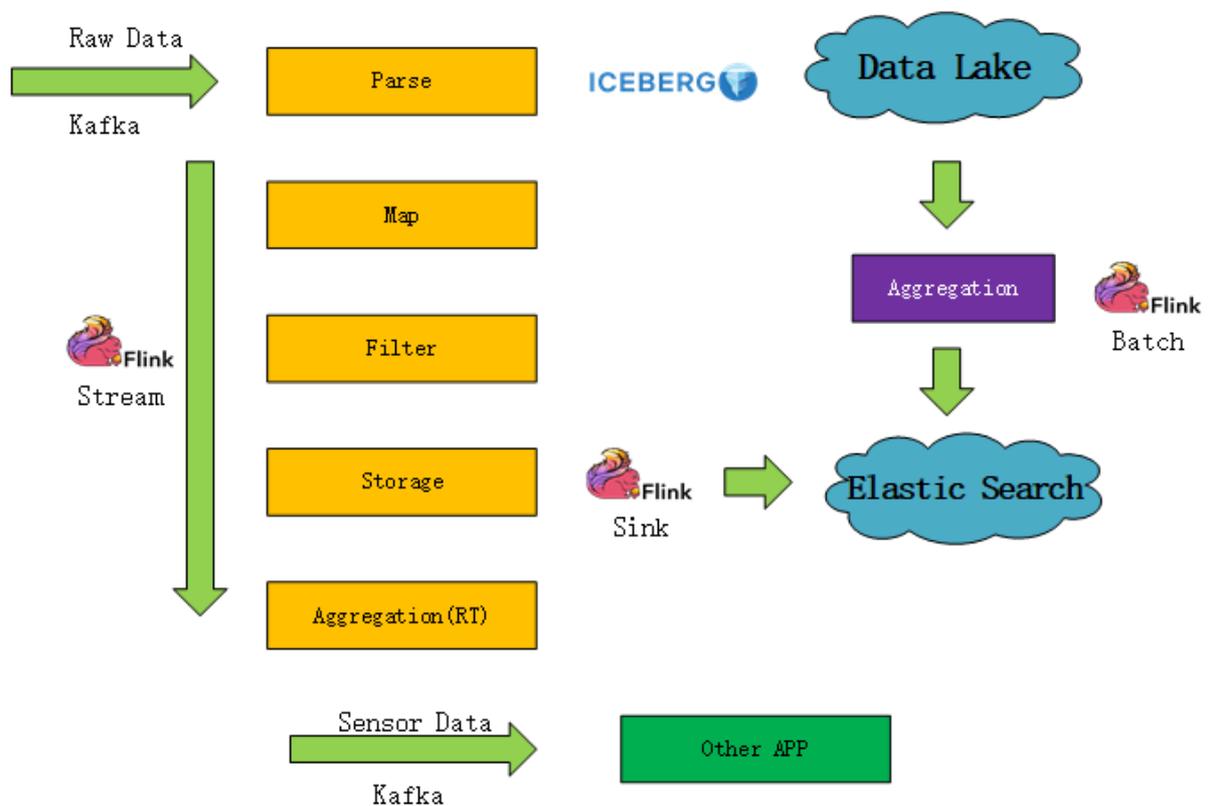
保留现有采集控制模块(DAC), 然后将以DAC采集的数据以及其他相关的感知数据, 存储到数据湖中, 这部分数据一般是原始的采集指纹, 我们对数据的采集频率和数值不做改动。ETL是后续数据处理流程, 将原始数据转换为业务感知数据, 即用户期望的数据格式, 是一种半结构化的数据。感知平台将其持久化在存储媒介上, 同时使用消息规则引擎, 可以将数据推送到任意地方。感知平台还提供一部分的数据可视化分析功能。

目前, 从技术上选型主要是用来 Apache Flink 和 Apache Iceberg 两个框架。Flink主要提供流式计算和批量聚合能力, Iceberg数据湖应用主要用于保存原始数据。技术框架整体的应用如下图:



ETL流程说明

我们着重于拓展ETL进程的功能, 新增包括数据入湖、钩子接口、脚本化计算等功能, 增强ETL的可扩展性。ETL的设计如下:



我们引入IceBerg数据湖方案，它支持Table Schema表格式定义，以及快速的upsert/delete操作，并支持ACID原子性语义保证。基于现有的HDFS分布式文件存储底层，打造存储设备采集结果的数据湖。IceBerg支持Streaming和Batch增量数据流接口，并可以和Flink流式计算框架很好的集成。

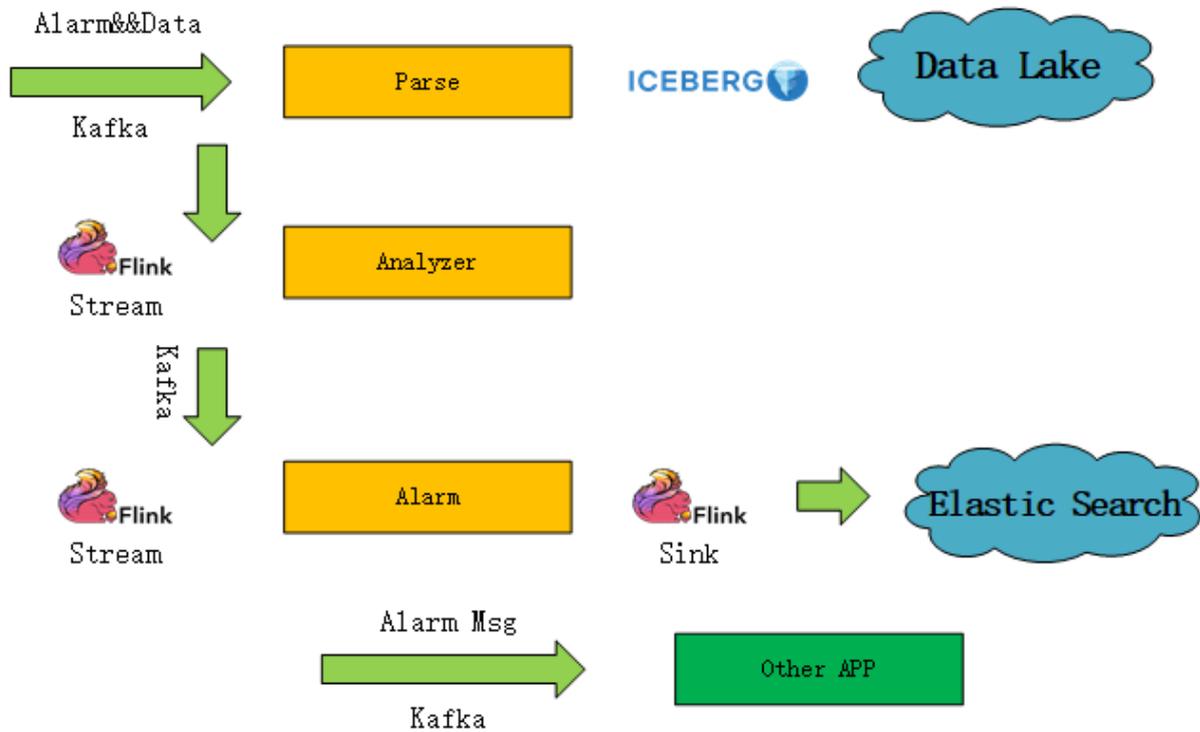
另外，数仓方面，依然采用ElasticSearch这个NoSql数据库，主要目的是保留目前技术栈，依赖其开箱即用分布式能力、高性能检索能力。另外日志这种需要全文检索的数据，也是储存在ElasticSearch中。

流程方面，采集数据通过kafka通道，进入ETL。这里的采集数据应该包括设备数据以及状态数据，经过ETL的Parse模块解析提取后，输出至ICEBERG的数据湖中，后面的流程基本跟原ET处理流程一致，经过计算（Map）和过滤（Filter），然后将感知数据存储至ElasticSearch中。这里保留Aggregation(RT)实时聚集功能，是为了得到近实时的增量聚合数据。最终，经过实时计算处理后的数据，输出到消息管道(Kafka)中。

其中相关进程模块说明如下：

- Parse 数据解析
- Map 数据计算转换
- Filter 合理值过滤、数据降噪
- Storage 存储到数据仓库(ElasticSearch)
- Aggregation(RT) 实时增量聚合
- Aggregation 定时聚合方法

告警保留原来的处理流程，如下图所示

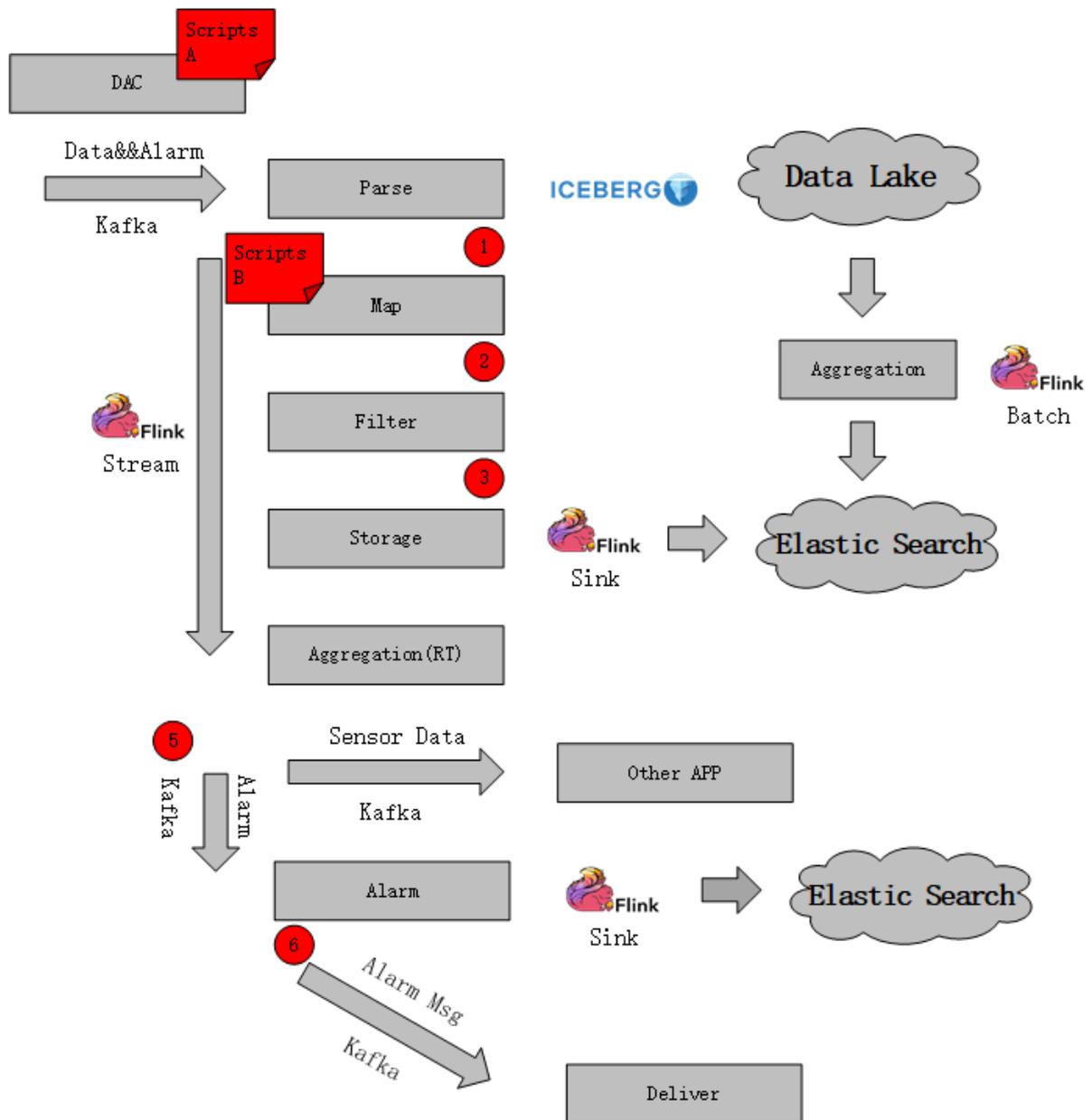


在解析提取(Parse)阶段，将告警原数据保存到数据湖。（这里保存的告警原数据，应该类似告警数据中详情数据，包含设备采集返回的错误码信息和错误内容）

后续步骤，跟原告警处理流程一致。该平台只保留Analyzer模块中阈值判断类数据告警，并提供业务钩子函数或脚本化功能，用户可以根据自己需求做定制。

钩子接口

为方便业务应用扩展，我们在ETL流程中增加部分hook接口(类似RPC的调用，用户可以实现自己的FaaS，提供函数计算服务，通过配置实现到自己的业务流程中)，以及通过脚本化方式扩展部分功能。接口钩子和脚本化调用主要包含在ETL中如下位置：



位置说明:

Hook 1 : Custom Map 扩展计算方法。

Hook 2: Custom Filter 扩展过滤方法。

Hook 3: Custom Before Storage 扩展后续计算。经过平台计算和过滤的数据，在存储前，可扩展业务计算。

Hook 5: Custom Analyze 扩展告警判断。生成自定义告警

Hook 6: Custom Deliver 扩展告警内容，在告警内容分发之前

Scripts A: 脚本实现协议解析

Scripts B: 脚本实现公式计算

视频

视频数据相对独立于传统设备数据。服务端只需提供数据推流服务拉取远端NVR，向前端推送RTMP协议的视频流数据。

将安心云平台的视频配置（NVR设置、摄像头配置）作为设备的一部分（特殊设备），在感知平台进行配置。

监控运维

感知平台扩展监控内容，除了设备接入，将感知过程中的数据、事件、状态，记录到 Prometheus。这边的功能点可以参考：

- 实时监控：数据指标、网络状态。
- 运维大盘：显示设备创建数、激活、在线、活跃设备数（周统计和周同比）
- 在线调试：需要设备在线，包括属性调试、服务调用、远程登录。
- 设备模拟器：模拟数据调试
- 日志服务：云端运行日志、设备本地日志、日志转储
- OTA升级和远程配置

规则引擎、流程和联动*

* 试验性功能，需进一步探索实现可能性

在以太规则基础上，增加数据源，包含ET计算后的感知数据和聚合数据，同时扩展输出方式。

探索流程引擎在数据流程控制中应用，使业务数据可以走ETL流程之外的自定义流程。

场景联动，通过数据/告警消息，触发动作执行：反向控制、告警输出、数据推送等。

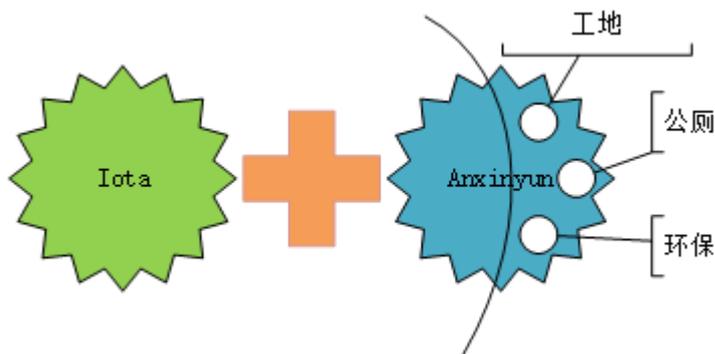
功能设计和实现方案

一) 功能整合

梳理物联网感知平台的功能包含如下（具体参见《物联网接入服务架构思路》）

- 数据接入
 - 产品定义
 - 协议开发
 - 设备管理
 - 部署
- 感知转换
 - 感知模型
 - 提取转换加载ETL
 - 数据聚合、清洗
- 数据服务
 - 数据存储
 - 告警服务
- 其他：视频、可视化

整体平台的架构基础，即实现两个平台感知能力的整合。以太的所有功能加上安心云的部分功能组合：



设计飞尚物联平台架构，主要分以下几个步骤：

第一步：Fork以太

沿用现在以太的代码，在此基础上开发。

代码层关于以太、iota等标签不需要改动，可沿用iota作为基础命名空间或关键词前缀。

UI层可做简单处理，包括登录页定制和logo替换、copyright修改等。

以下是以太目前服务列表（第三方基础服务设施未列出），本平台需要保留。

服务名称	描述	
iota-alert-server	以太告警服务	
iota-api	以太console端WEBAPI	
iota-background	以太admin端WEBAPI	
iota-dac	以太采集服务	
iota-dac-test	以太DAC协议测试服务	
iota-message-center	以太消息中心	
iota-orchestrator	以太DAC编排器	
iota-proxy	以太接入网关代理	
iota-rules-engine	以太规则引擎	
iota-web	以太Console端	
iota-web-background	以太Admin端	

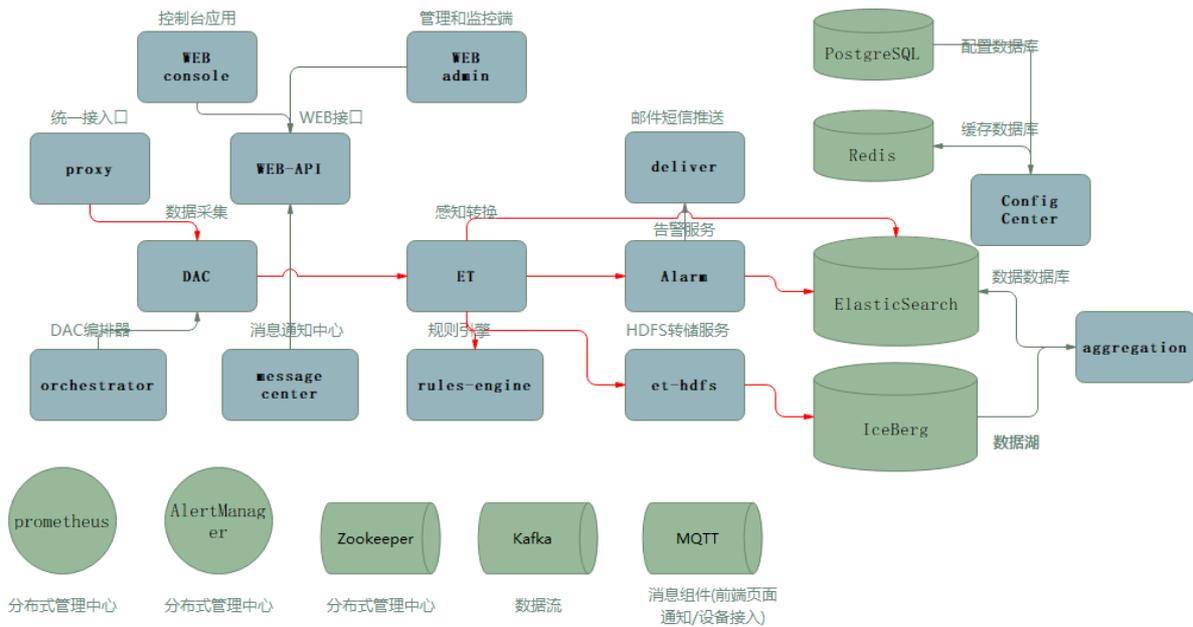
第二步：安心云服务选取/改造

安心云平台部分服务（主要是后端服务）和前端功能需要整合到新的感知平台，梳理其中应该包含的后端服务：

服务名称	描述	
et	数据ETL进程	
alarm	告警进程	
aggregation	聚集计算	
config-center	配置同步redis	
et-hdfs	HDFS数据文件转储进程	
et-recalc	重计算进程	
deliver	邮件短信推送服务	
weather	天气服务	
pyrpc	计算服务(python rpc服务)	
*report-master	报表调度服务 (暂定)	
*report-client	报表生成服务 (暂定)	

改造：安心云的服务端进程中，遗留部分业务逻辑代码，需在整合过程剔除。详见《[ET中业务专有代码](#)》

整体的进程服务组织大致如下图：



第三步：模型定义+数据库整合 + 数据湖定义

感知模型定义，数据库表格设计。

数据湖Schema定义；

Flink结合IceBerg作入湖操做；

二) 扩展接口

主要包括两种方式的功能扩展方法：

1. 定义数据接口：通过rpc方式调用
2. 通过脚本语言交互

三) 功能优化

设备协议解析优化：

1. 定义平台标准通用mqtt-json格式。实现标准java-sdk库作设备上的开发套件
2. 可选标准协议（modbus）
3. 可视化协议组件（协议组成选取指定类型解析、json格式字段映射配置）
4. 扩展目前支持的脚本类型

规则引擎扩展：

1. 数据源扩展
2. 输出方式扩展
3. 扩展HTTP/MQTT输出规则的内容格式定义

场景联动：

1. 定义数据联动规则
2. 触发能力接口
3. 触发告警调用

简化设备接入：

1. 接入demo示范
2. 接入流程向导指引

系统设计原则

简+

易扩展性+