

# 前言

---

我申请的项目是wasm函数扩展。这个项目的目标是使 ekuiper 可以调用 WASM 函数，方便用户使用 WASM 函数

扩展 ekuiper 的处理能力。我需要做的部分是实现 wasm 函数扩展框架，使 ekuiper 可以调用 wasm 函数的功

能，最后将其集成到 ekuiper 中

## 对wasm的理解

---

WebAssembly——通常缩写为“WASM”——是一种扩展 Web 平台以支持 C、C++、Rust、Go 等编译语言的技术。这些语言可以编译为特殊的 WASM 二进制格式，然后加载到浏览器中。

wasm 是安全的沙盒，所以可以用于插件的动态扩展。

ekuiper，可以通过实现的 wasm 扩展框架，以隔离沙箱的形式运行自定义代码或函数。

为了运行沙箱，需要 wasm 执行引擎，这里老师给我推荐的是 WasmEdge。WasmEdge 的一个常见用例是将其嵌入到主机应用程序中。它允许我们为我们的应用程序支持第三方插件和扩展。这些插件可以用 wasmedge 支持的任意语言编写，并在 wasmedge 沙箱中安全可靠的执行。

我们需要的是 wasmedge-go。

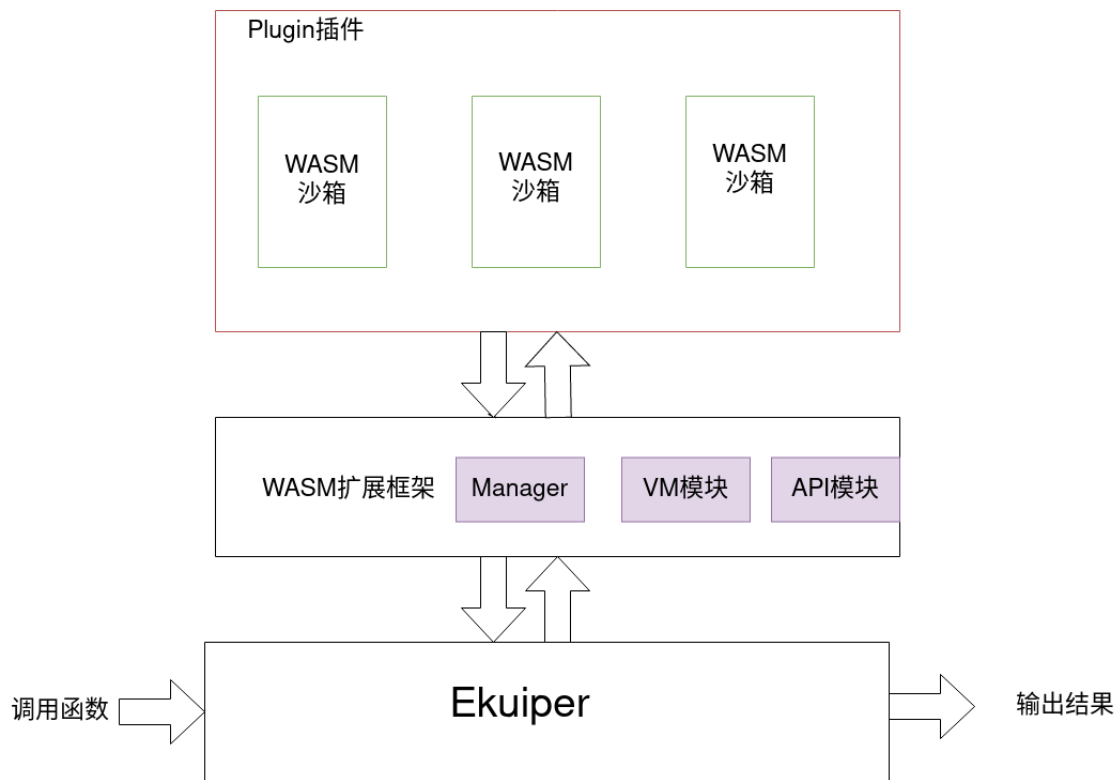
## wasm 扩展框架

---

### 整体框架

---

用户使用 ekuiper SQL 语句处理数据，而在 SQL 语句中使用的函数由 wasm 沙箱中提供。我们现在需要实现一个 wasm 扩展框架，根据用户配置加载沙箱，实现 ekuiper 可以调用 wasm 函数的功能，整体框架预构如下图所示。



## 扩展框架

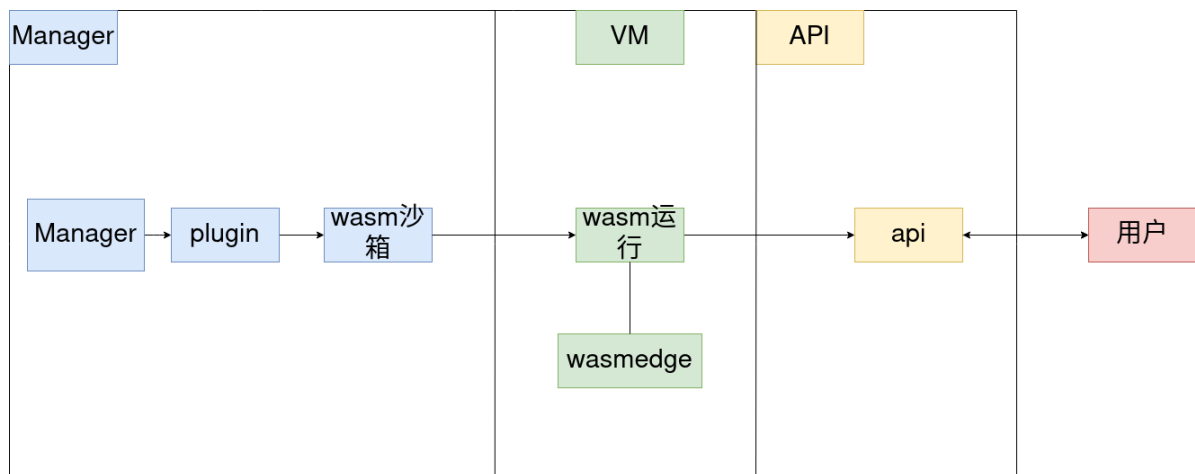
我们可以以插件的形式对 wasm 扩展进行统一管理，插件是一组 wasm 沙箱及其相应配置的集合。用户通过配置来加载、更新以及卸载 wasm 插件等。我设计插件配置如下，参考了 ekuiper 的 portable 插件：

```
{
  "name": "car",                //插件名称
  "language": "go",
  "vm_config": [
    {
      "engine": "wasmedge",      //执行引擎
      "path": "/plugin/XX.wasm", //wasm 文件路径
    }
  ],
  "functions": [
    "link", "rank"              //所需函数
  ]
}
```

当 ekuiper 家在上述配置的时候，用户通过 wasm 扩展框架获取指定插件的沙箱实例，然后后续与 wasm 沙箱暴露的 api 进行交互，调用函数。

wasn 扩展框架大致分为三个部分，分别是Manager、VM 和 API 模块三个部分，他们分别负责不同的分工：

1. Manaager模块：负责对 插件及其配置，沙箱统一管理，进行增删改查等，负责按照用户配置加载 wasm 沙箱。
2. VM 模块：负责 wasm 沙箱的运行，.wasm文件的编译执行等。
3. API 模块：提供对外接口。



具体功能划分：

#### 1. Manager中：

Manager：负责对所有插件进行管理，增删改查等

plugin：插件配置的封装，并进行 wasm 沙箱的加载

wasm沙箱：负责沙箱的生命周期管理，如启动，更新，销毁等。

#### 2. VM 模块：

wasm运行：负责 对 .wasm 文件的编译执行，沙箱的运行。

wasmedge：是沙箱执行引擎

#### 3. API 模块：

api：沙箱对外的 api，当用户使用 wasm 插件编写 ekuiper 扩展时，通过api来与 wasm 插件进行交互。

## 时间安排

我根据项目和自身的实际情况，时间安排如下：

时 间	项目进度
6 月 16 日—— 6 月 30 日	了解wasmedge-go相关实例，学习官网上的函数使用
7 月 1 日 —— 7 月 10 日	编写 wasm 扩展框架架构，Manager，VM 和 API 模块内的函数命名，参数设置等
7 月 11 日 —— 8 月 5 日	完成 Manager ， VM ， API 模块函数的实现，并撰写项目文档，准备项目中期报告
8 月 6 日 —— 8 月 10 日	编写相应的单元测试，修改 Bug ， 完善功能
8 月 11 日 —— 9 月 25 日	将 扩展框架 集成到 ekuiper 中
9 月 25 日 —— 9 月 30 日	继续改进，撰写说明文档，总结项目最终报告