



函数平台扩展数据能力



构建计算服务 何必从零开始？



OR

DataFlux.*f(x)*

无需从零开始搭建Web服务，任务队列等组件，在「非业务」工作上投入资金、人力。使用DataFlux Func 可以帮助用户直接开展业务系统开发。

DataFlux Func 包含了每个计算服务都需要的Web端应用、API 接口、分布式队列系统，以及市面上常见的数学、AI、机器学习相关开源库，做到开箱即用。



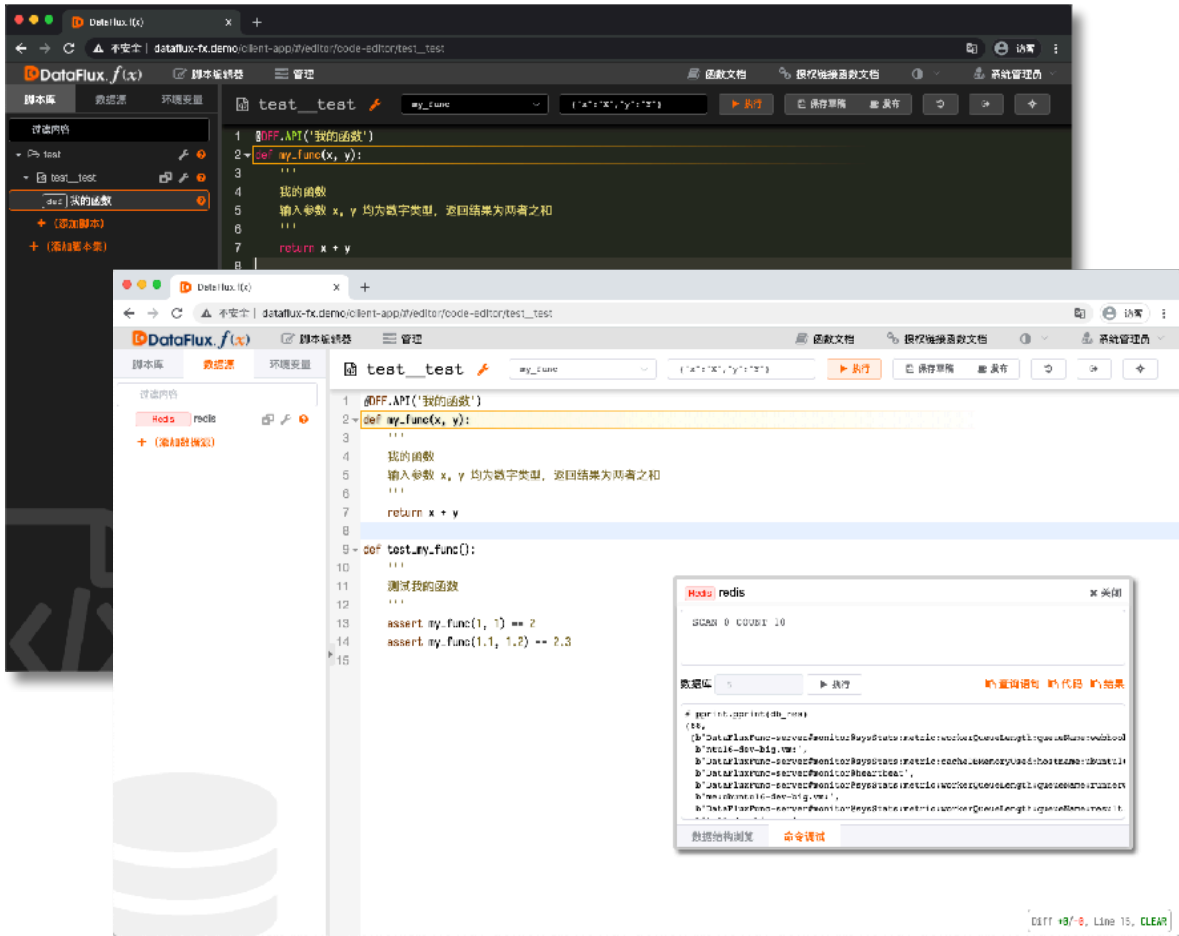
```
1
2 @DFF.API('计算两数相加')
3 def sum(x, y):
4     x = int(x)
5     y = int(y)
6     return x + y
```

起个名字 函数就是API

DataFlux Func 极大简化了构建HTTP API 的方式，在实际编码过程中，只需要一行代码即可将一个函数转换为一个HTTP API接口，并且同时具备分布式任务调度能力。

开发人员无需关注传统Web开发中的HTTP请求体、响应体，任务调度等一系列「非业务」功能，最终极大地简化开发过程，提高开发效率。



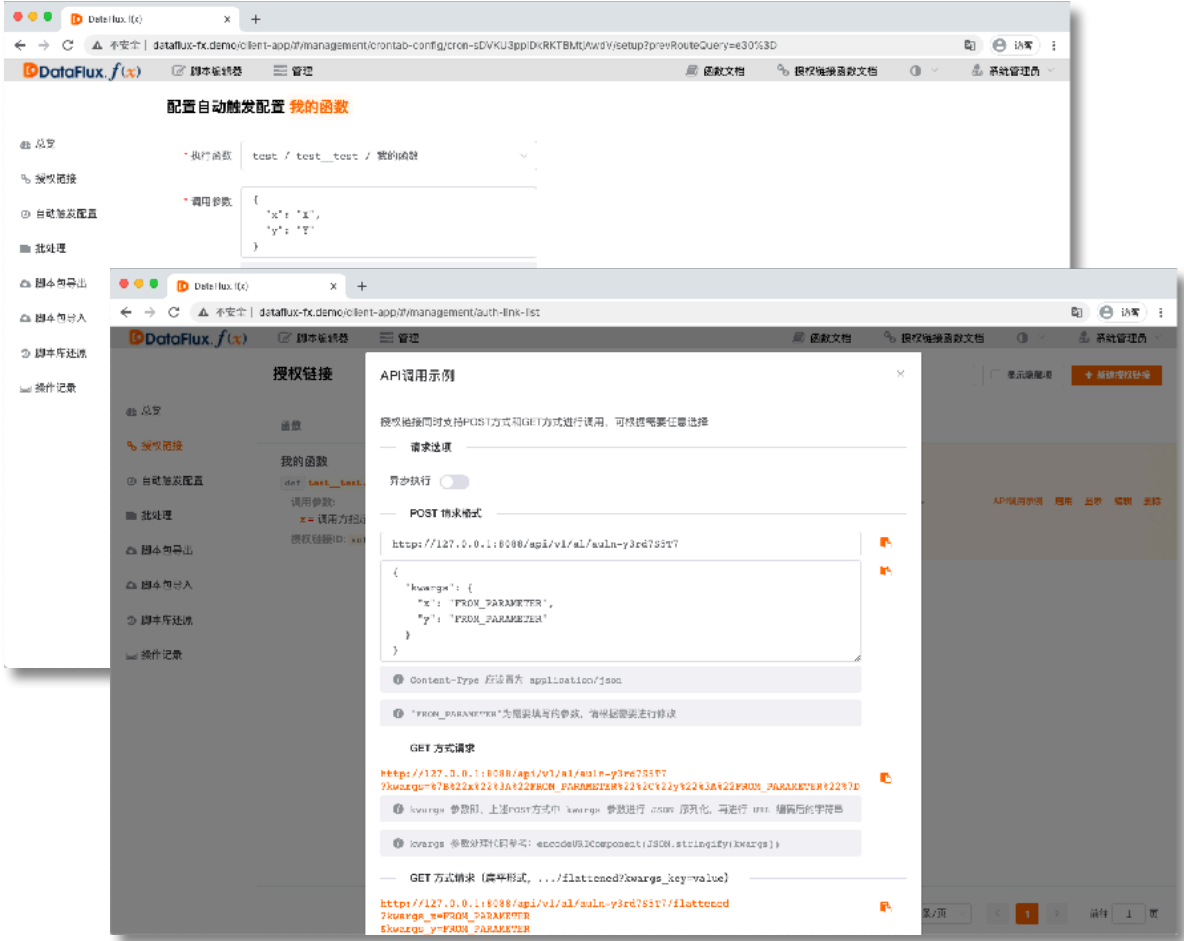


完备的脚本编写、管理功能 支持直接执行函数

DataFlux Func 内置了在线代码编辑器、数据源管理等开发环境。开发人员无需在本地搭建开发环境，打开浏览器即可开始进行开发。

内置的数据源管理器不仅对代码访问数据库的处理做了统一封装，同时也是一个简易数据库客户端。开发人员可以直接在线访问数据库，不用额外在本地安装各种数据库客户端软件。



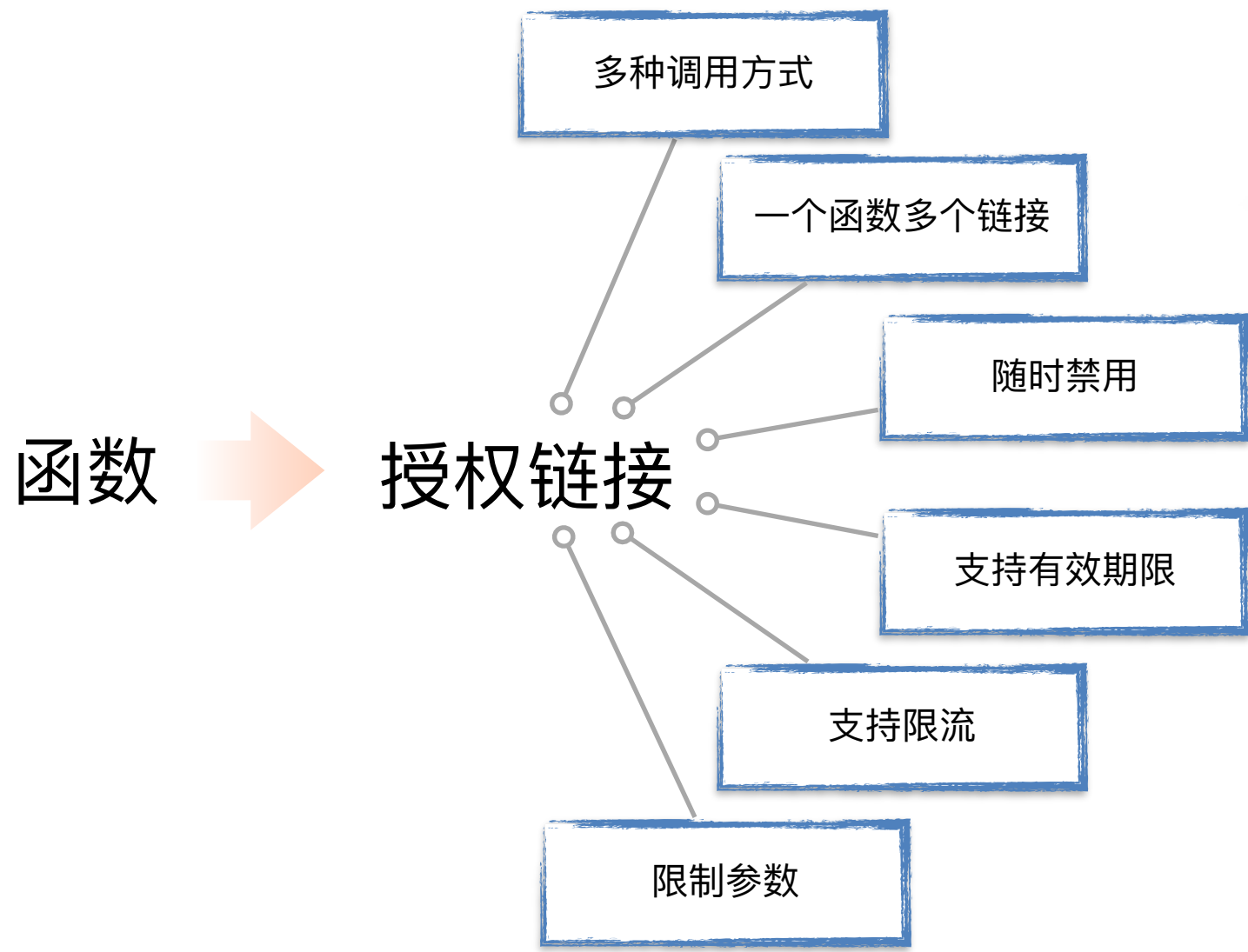


丰富的内置功能

除了基本的代码编辑器和数据源管理器，DataFlux Func 同时内置了其他丰富功能，如：

- 函数文档
- 授权链接公开函数为API
- 自动触发函数执行
- 批处理方式投递任务
- 脚本包导入导出
- 脚本库还原
- ...





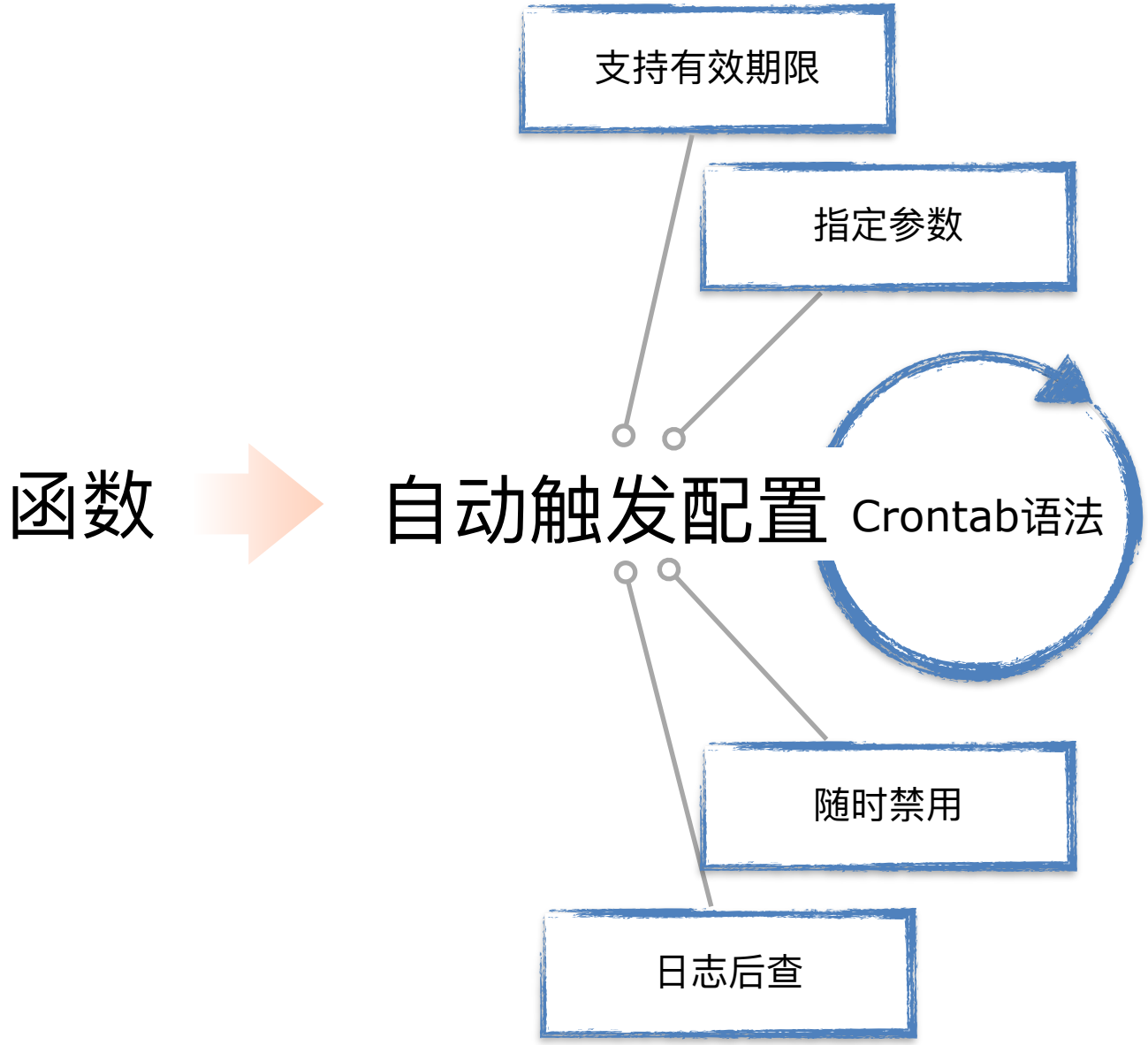
功能强大的授权链接

授权链接允许将一个函数公开为一个标准HTTP API。每个函数都可以配置多个授权链接供不同目的使用。

通过授权链接方式访问函数，支持GET、POST等多种调用方式。且每个授权链接可以单独设置禁用、有效期限、按年/月/日/时/分/秒进行限流。

此外，授权链接还可以对函数的某些参数进行限制，允许或禁止外部传入，避免开发人员反复编写大量大同小异的包装函数。





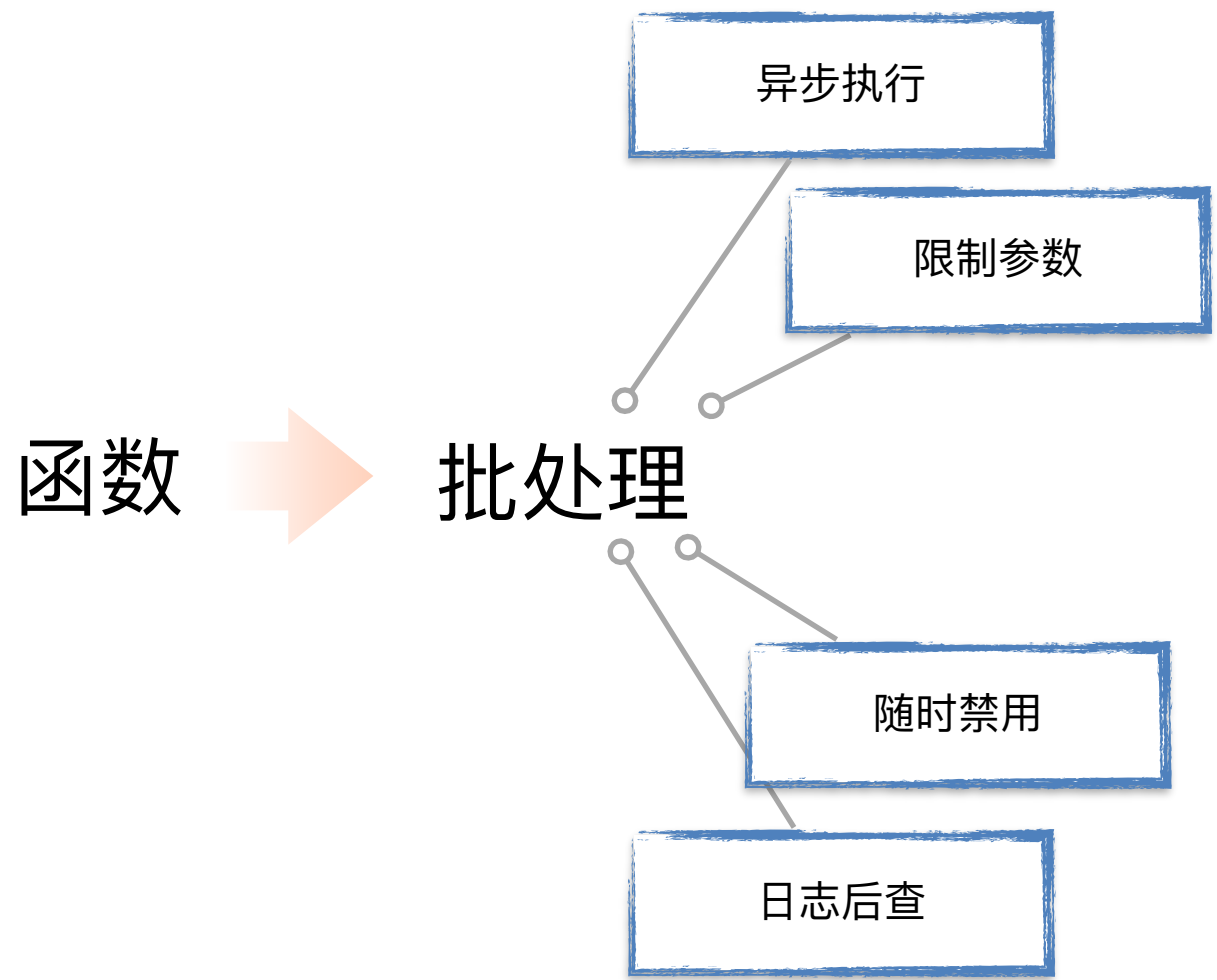
自动化总是不能缺席

函数除了主动调用执行外，也可以通过平台提供的自动触发配置来自动执行。

自动触发配置同样允许每个函数具有多个自动触发配置，单独设置禁用、有效期限。此外，自动触发配置还会保留运行时日志和错误信息，方便后续查询。

利用自动触发配置，可以方便实现各种主动监测、定期报告、数据获取等自动化功能。



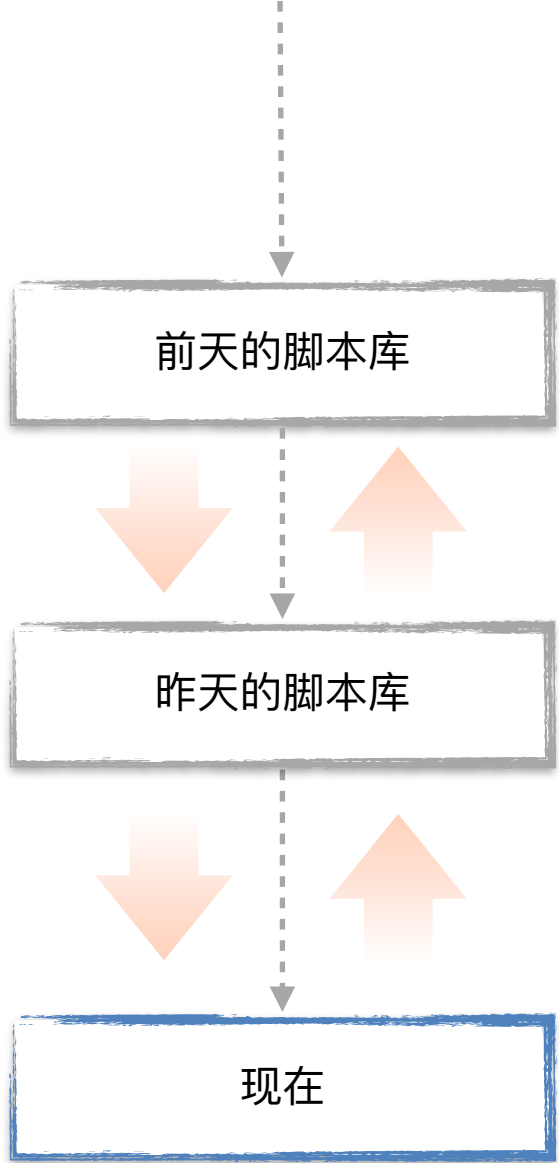


重活 慢慢来 不用等

除了同步方式执行函数，DataFlux Func 还提供了批处理功能允许函数以异步方式执行。调用方只需要向平台投递任务，接口会立刻返回，而无需等待任务实际完成。

批处理非常适合执行需要较长时间的复杂任务。





世界上没有后悔药

但有还原功能

除了代码开发以及函数调用方面的各种功能，DataFlux Func 也提供了脚本库还原功能来保护用户的代码。

用户可以像创建快照一样随时创建脚本库的还原点。以供在需要的时候，及时还原代码。



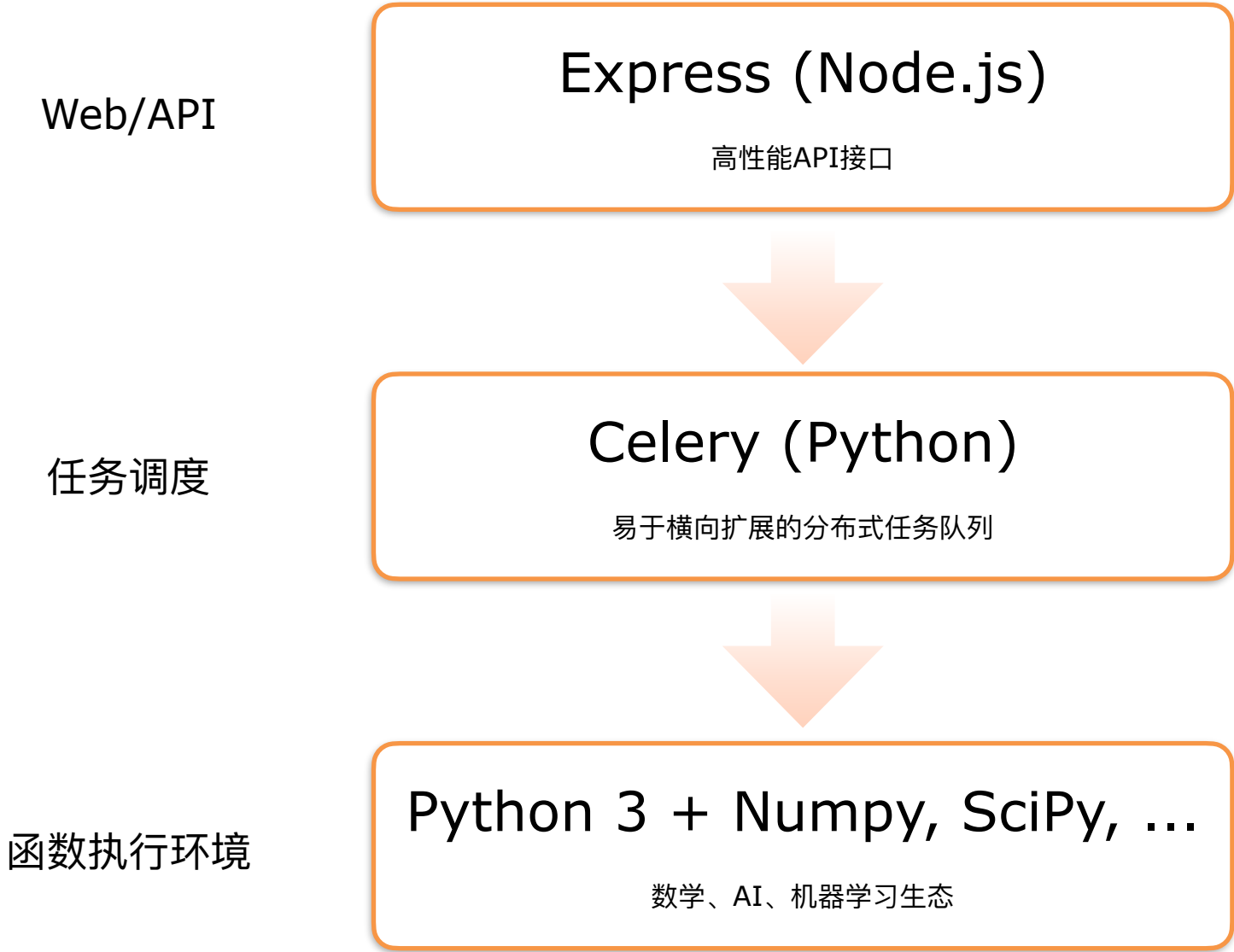


内置多种数据库支持

DataFlux Func 内置了大多数市面上常见数据库，以及其兼容数据库的支持，如：

- InfluxDB
- MySQL
- Redis
- Memcached
- Clickhouse
- Oracle数据库
- Microsoft SQL Server
- PostgreSQL
- mongoDB
- elasticsearch





在Web领域，Node.js 具备高性能易部署的特点。而在数据处理领域，Python 具有其他语言无法比拟的生态优势。

DataFlux Func 汲取两者优势结合在一起，依靠分布式队列，结合使用了这两种技术。



 DataFlux. $f(x)$



树莓派4B 2GB版 ¥300

树莓派 足矣

在硬件成本方面，DataFlux Func 非常轻量化，本身对硬件要求非常低。

根据实际测试，在售价仅300人民币的树莓派4B上也能够正常运行。



```
docker pull; docker run
```

为了进一步方便用户部署，DataFlux Func还提供了Docker 方式运行。几行命令就可以完成部署和运行。

 DataFlux. $f(x)$



场景：阿里云DataV 数据对接/转换

非结构化业务数据

```
{
  "name": "张三",
  "labels": ["90后", "时尚", "手游", "程序员"]
}
```

结构化业务数据

ID	name	gender	amount
1	张三	男	100
2	李四	女	200
3	王五	男	150
4	赵六	女	80

原始业务数据



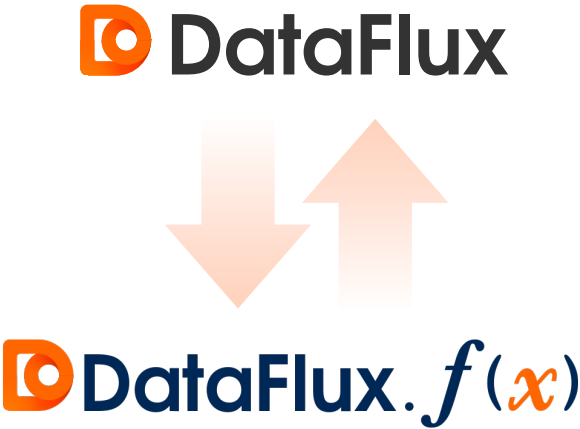
阿里云DataV 大屏数据

利用DataFlux Func 作为各种业务数据的转换层，对接阿里云DataV 大屏，是典型的使用场景。

企业无需改变自身的业务系统即可实现额外的数据可视化需求。由于不直接解除核心业务系统，大屏的开发甚至可以直接交给外部团队实现。



扩展DataFlux Studio 图表分析模式

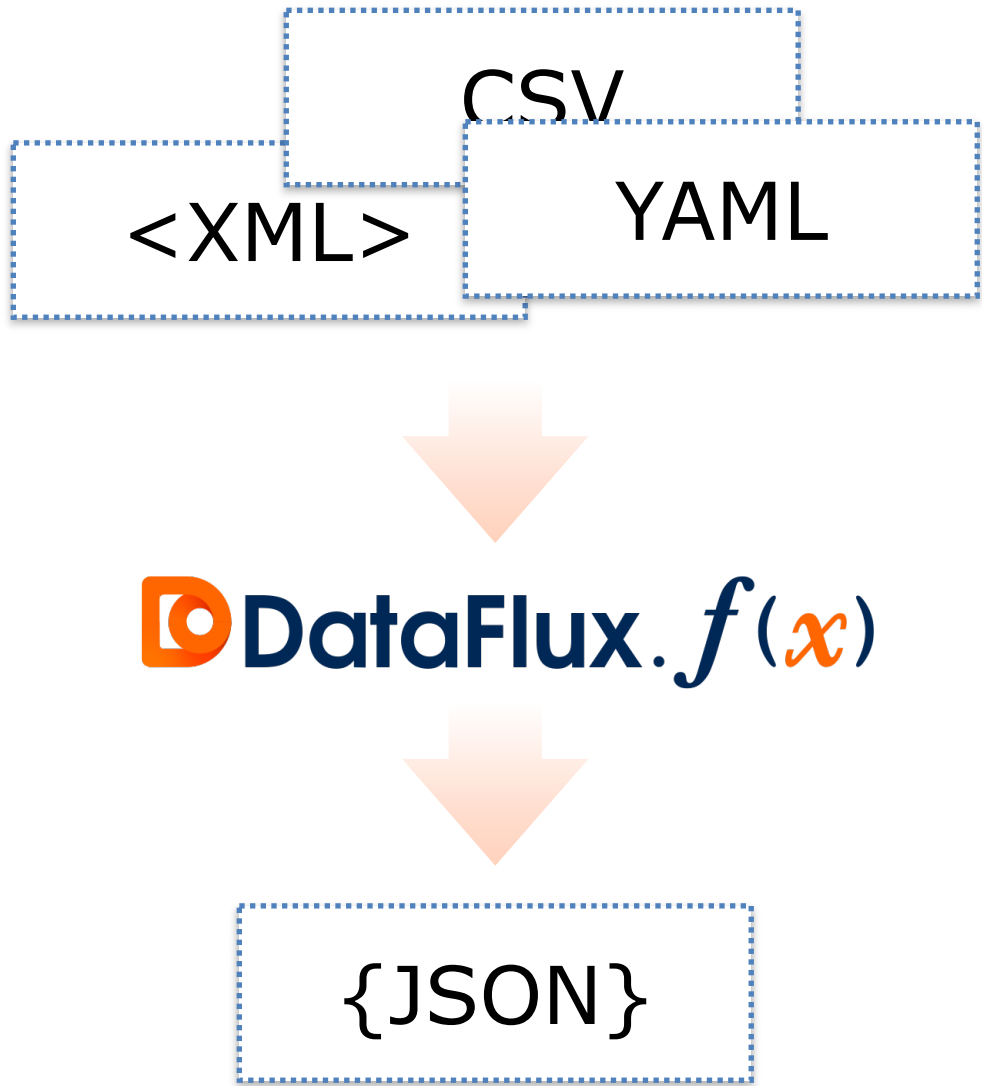


DataFlux Func 的另一个典型场景就是扩展原有系统功能。如DataFlux Studio 中的图表分析模式中的预测函数，实际上就是由DataFlux Func 提供的。

这么做的好处在于，相当于将DataFlux Func 看作一个插件系统集成到本系统。插件系统本身的更新发布、甚至崩溃都不会影响主系统的运行。而某些专业领域的「插件开发」也可以交付给外部团队实现，无需影响本系统。



场景：已有系统API二次封装



外部世界太复杂

统一转换一下

当用户的业务系统需要访问外部数据，且数据格式多变又不统一时，可以利用DataFlux Func 作为中间层，将不同来源的数据转换为统一格式。方便用户核心业务系统开发。

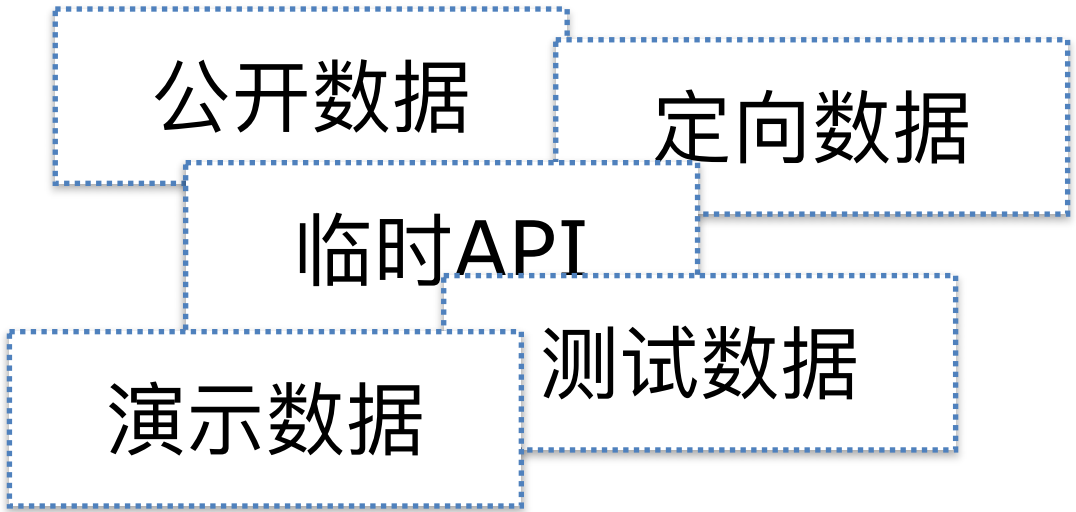
当外部数据发生改变时，只需修改对应的转换函数，无需更新核心业务系统。



核心业务系统



DataFlux. $f(x)$

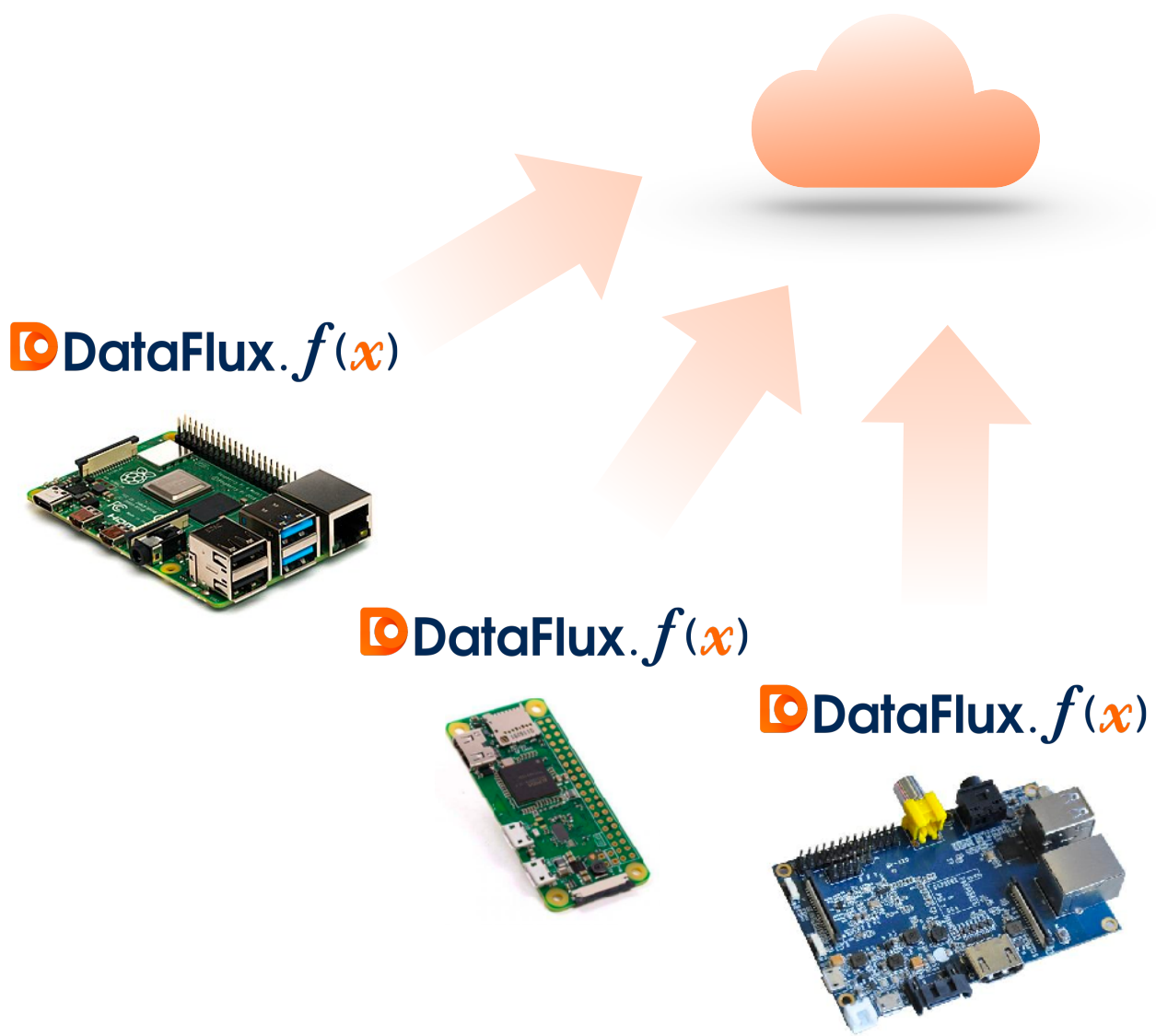


不断拥抱变化 也要稳住后方

当用户的业务系统需要对外公开数据时，也可以利用DataFlux Func 作为媒介。根据业务需要，随时开启或关闭临时API，开放测试数据、演示数据等。

满足业务不断变化的需求同时，让核心业务系统保持稳定。提高安全性，降低故障风险。





不同IoT设备

同一套方案

对于目前发展越来越快的IoT领域，得益于DataFlux Func 轻量级的硬件要求，利用DataFlux Func 作为开发框架，可以减少不同IoT 设备之间的差异，使用同一套开发方案，降低开发周期和成本。





感谢观看

Thanks