



一站式开源持续测试平台

2023 年 9 月

1

持续测试的兴起与现状

2

MeterSphere 加速企业持续测试落地

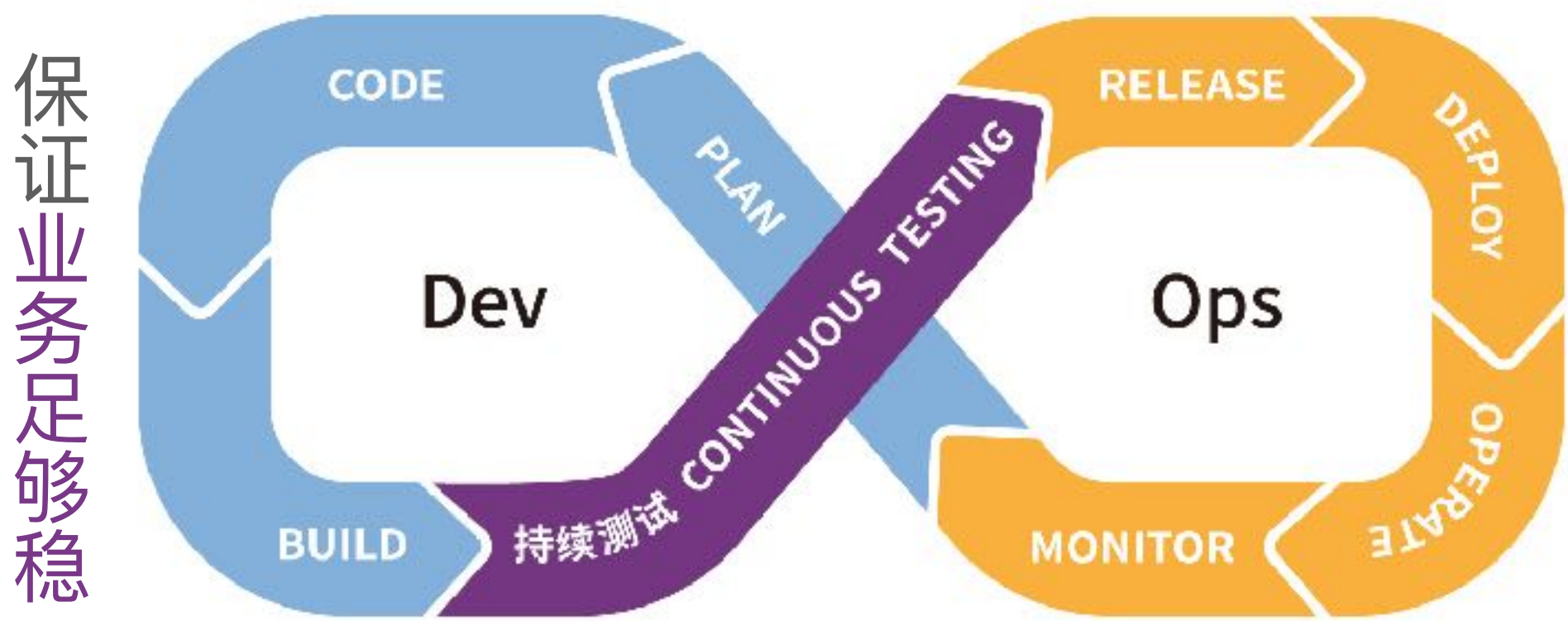
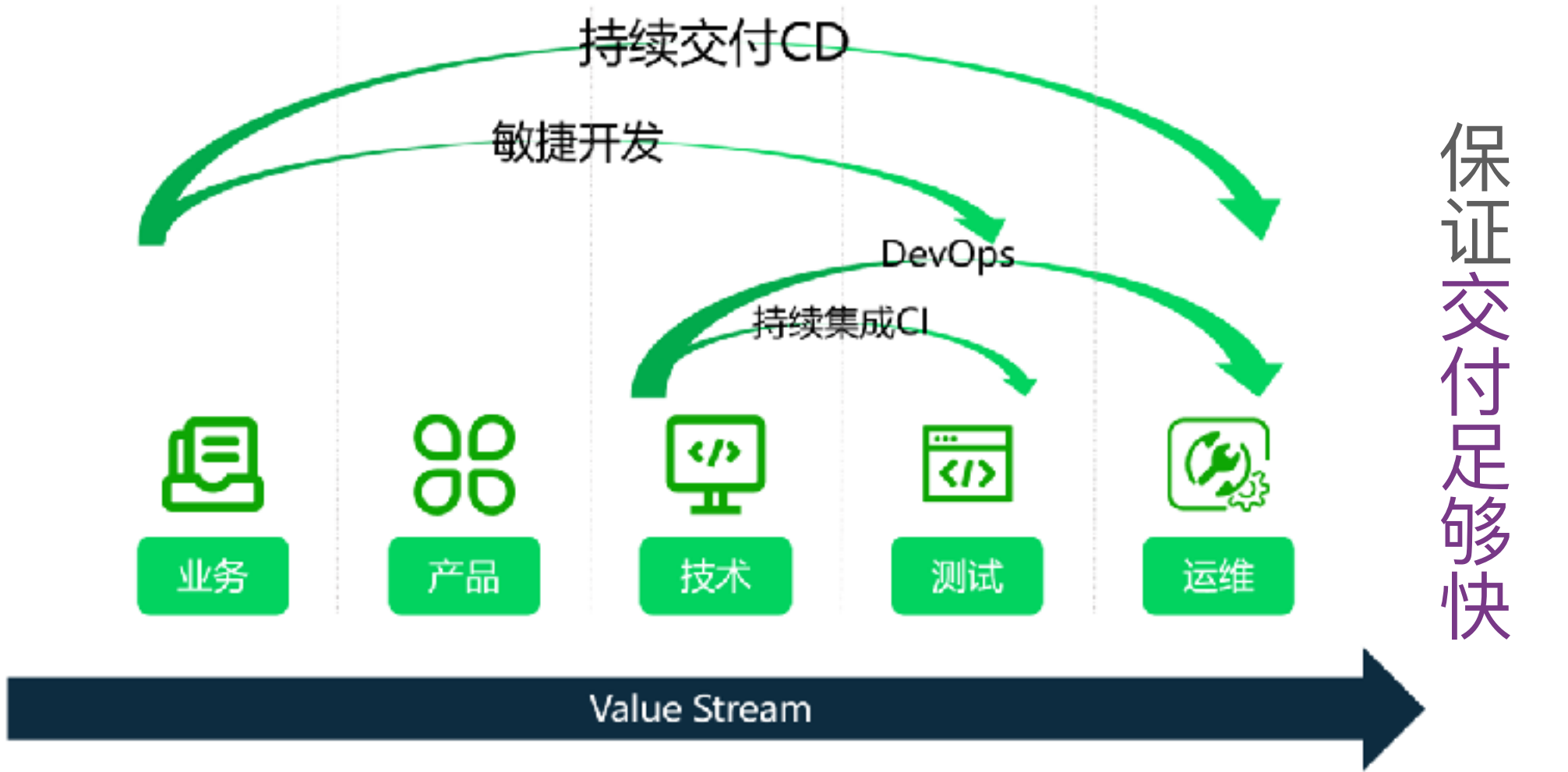
3

MeterSphere 企业版及专业服务

持续测试是持续交付发展的必然需求

持续交付能力是企业核心竞争力

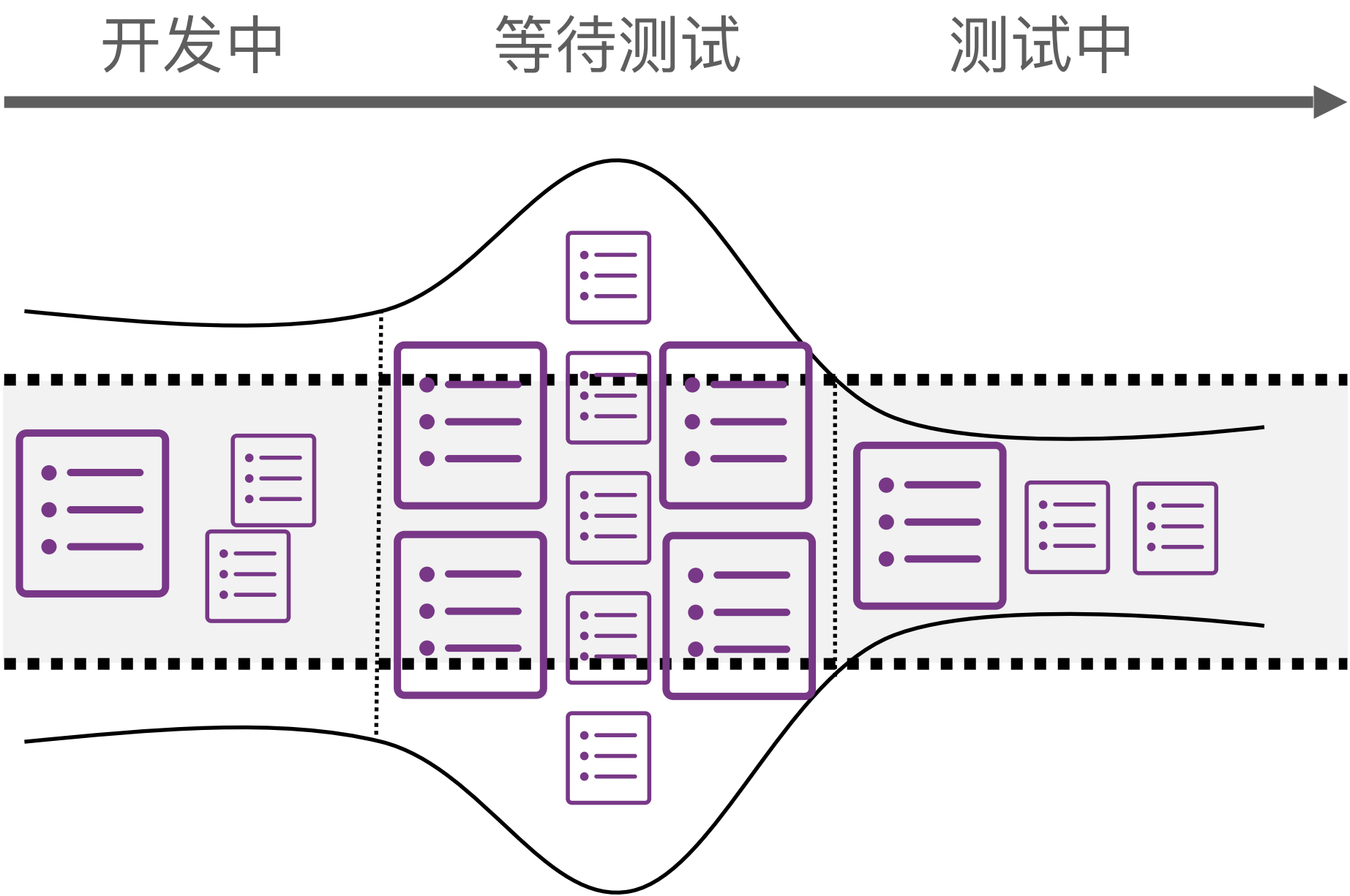
持续测试能力保障业务足够可靠



持续交付可以降低发布风险，提高可靠性，使软件能够根据用户反馈、市场变化和企业战略变更不断进行调整。

持续测试是执行自动化测试的过程，作为软件交付流水线的重要一环，持续测试帮助企业尽快获得软件发布后业务风险的反馈。

为什么测试环节会成为持续交付中的瓶颈？



因为下游处理能力不足导致的阻塞

图片来源：《持续交付 2.0》

造成测试阻塞的三个方面

01. 测试速度

- 非探索性手工执行用例比重过多；
- 用例编写和维护在协同上较为冗余；
- 测试环境准备，测试报告整理等花费了大量的时间。

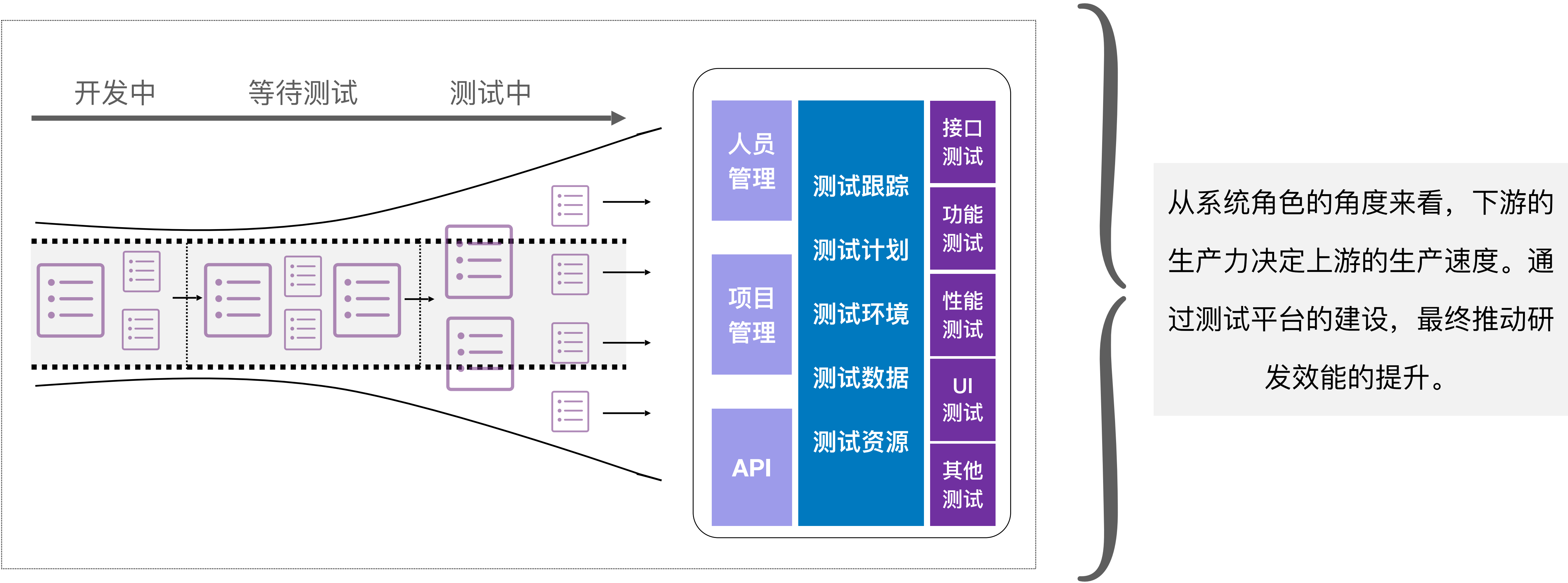
02. 碎片化管理

- 测试工具使用碎片化，不同的测试工具之间不兼容、无联动；
- 测试人员管理碎片化，不同的测试人员协同缺失。

03. 价值衡量

- 需求/用户故事的风险评估缺失；
- 关键性核心业务测试评估不足。

测试平台是提高生产能效最高效方法之一



建设测试平台，提升测试整体的吞吐量

1

持续测试的兴起与现状

2

MeterSphere 加速企业持续测试落地

3

MeterSphere 企业版及专业服务

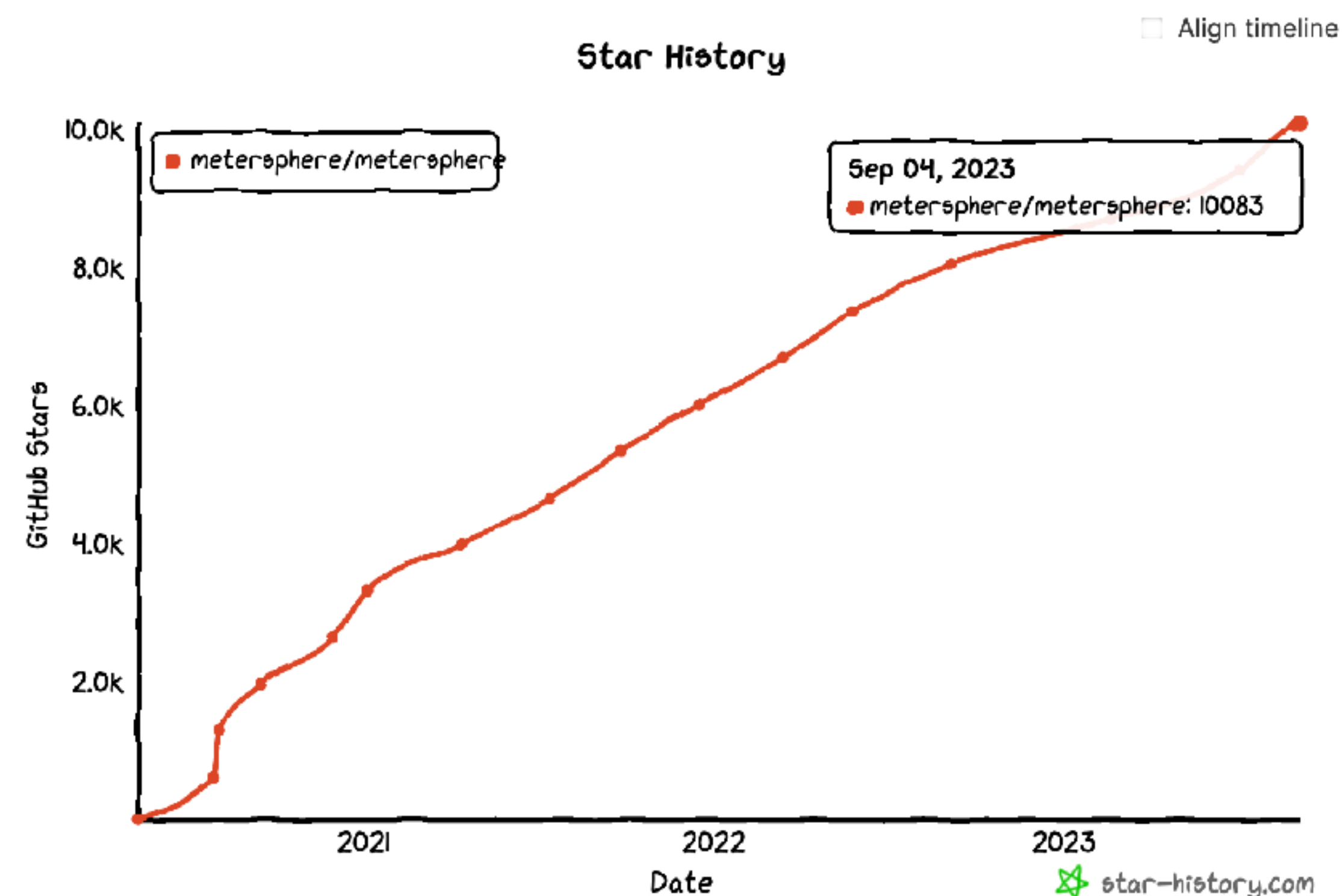
MeterSphere 的使命

MeterSphere 是一站式的开源持续测试平台，遵循 GPL v3 开源许可协议，涵盖测试管理、接口测试、UI 测试和性能测试等功能，全面兼容 JMeter、Selenium 等主流开源标准，有效助力开发和测试团队充分利用云弹性进行高度可扩展的自动化测试，加速高质量的软件交付。

广受欢迎的开源项目

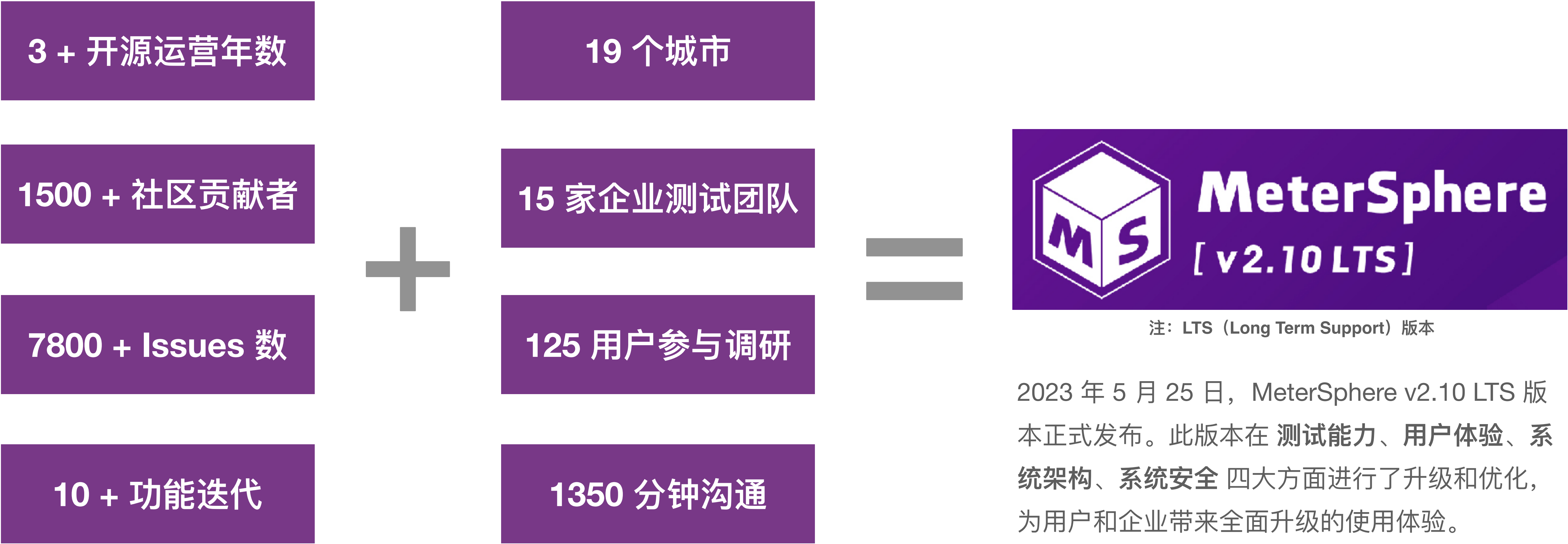


- 累计下载次数 180,000 +
- 日均下载次数 150 +
- 超过 350 人的 Issue 贡献者
- 2020 年度 OSC 中国开源项目评选 “最佳人气项目”
- TesterHome 评选 “MTSC 2021 年度开源项目”、
“MTSC 2022 年度最佳开源项目”
- 2022 年 GVP —— Gitee 最有价值开源项目



注：GitHub Star 数量统计截止至 2023 年 9 月

MeterSphere v2.10 LTS 发布



累计下载超过 180,000 次，使用群体持续扩大

- 用户群体分布广泛，参与社区讨论的用户超过 10,000 人 -



MeterSphere 关键能力设计：平台一站式

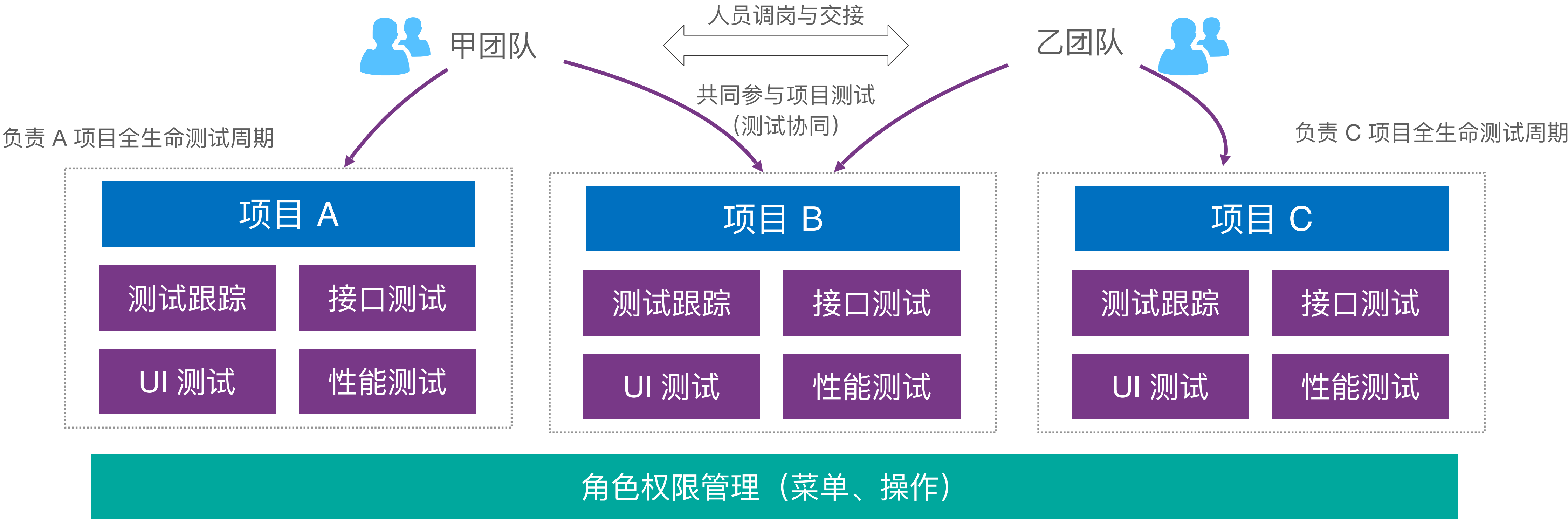
- 一个平台解决测试人员协作、测试能力、测试数据一站式问题 -



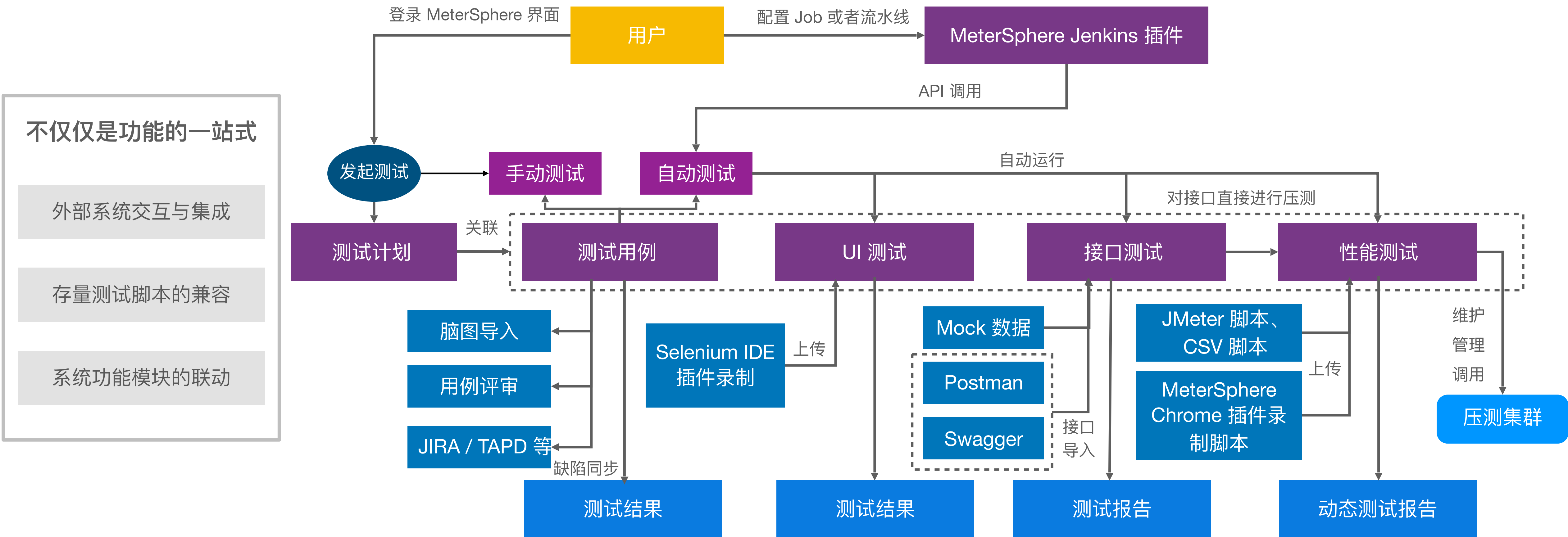
MeterSphere 关键能力设计一：协作一站式

- 以项目为维度的测试管理，让测试不再局限于具体测试人员-

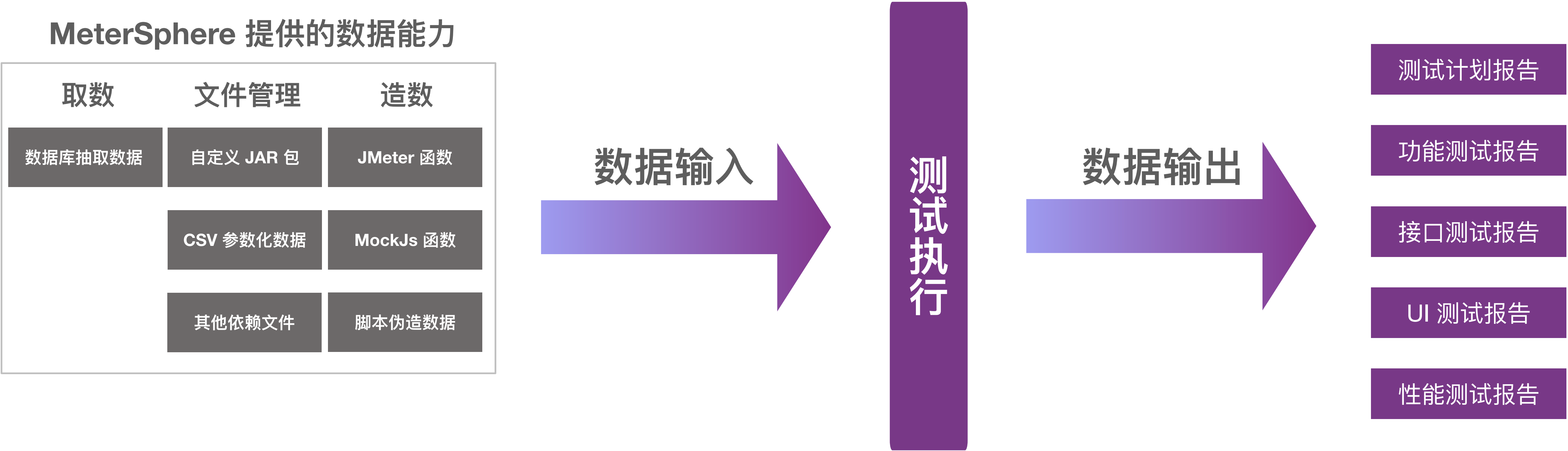
共享用例库	用例脑图	评审协同	接口规范	版本管理	数据度量
项目交接	场景编写	计划制定	脚本规范	消息中心	报告分享



MeterSphere 关键能力设计二：能力一站式

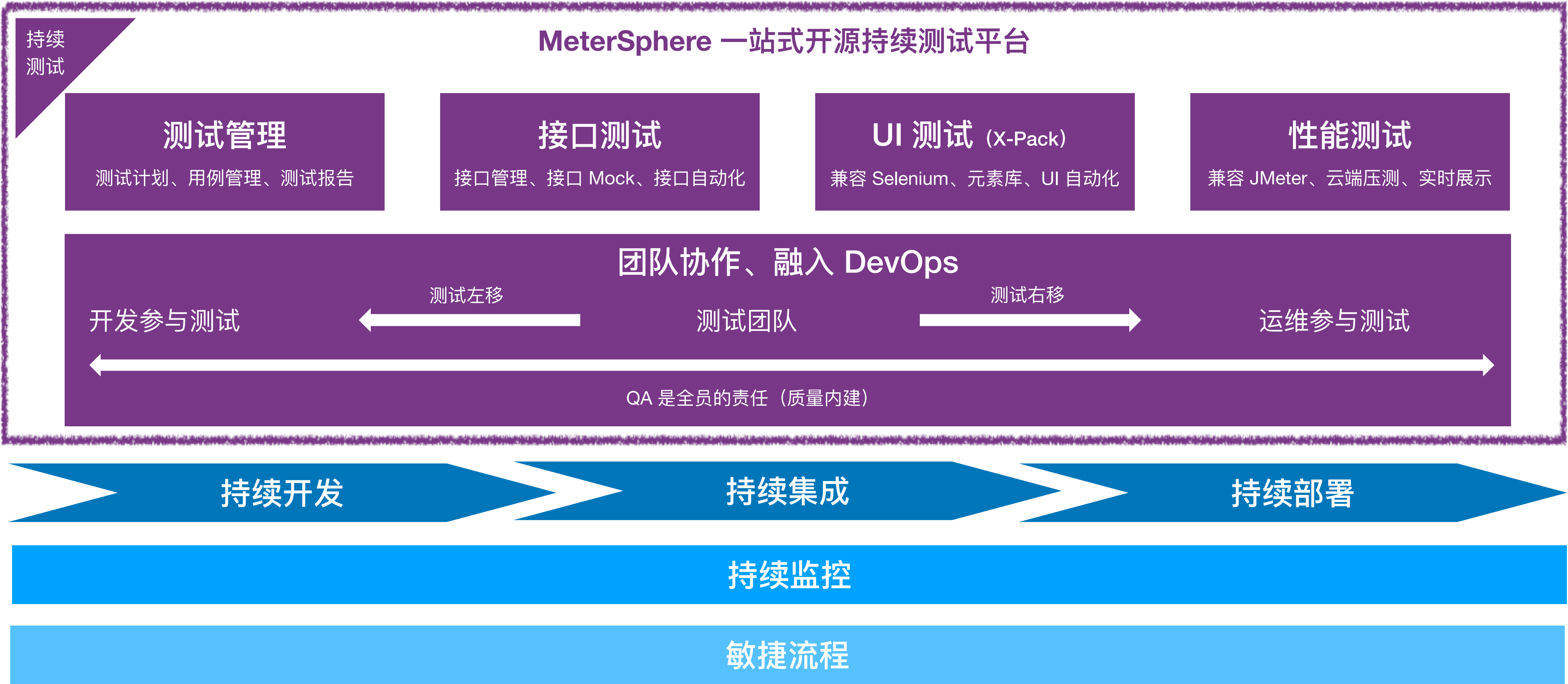


MeterSphere 关键能力设计三：数据一站式



依据三个关键能力打造一站式持续测试平台

- 让测试成为一种能力，为不同的团队进行赋能 -



MeterSphere 的核心功能设计

- 五大核心能力，满足企业持续测试成熟度模型 -

一、测试管理

全场景的测试跟踪能力

二、接口测试

灵活的接口自动化能力

三、UI 测试

自然语言的 UI 自动化测试

四、性能测试

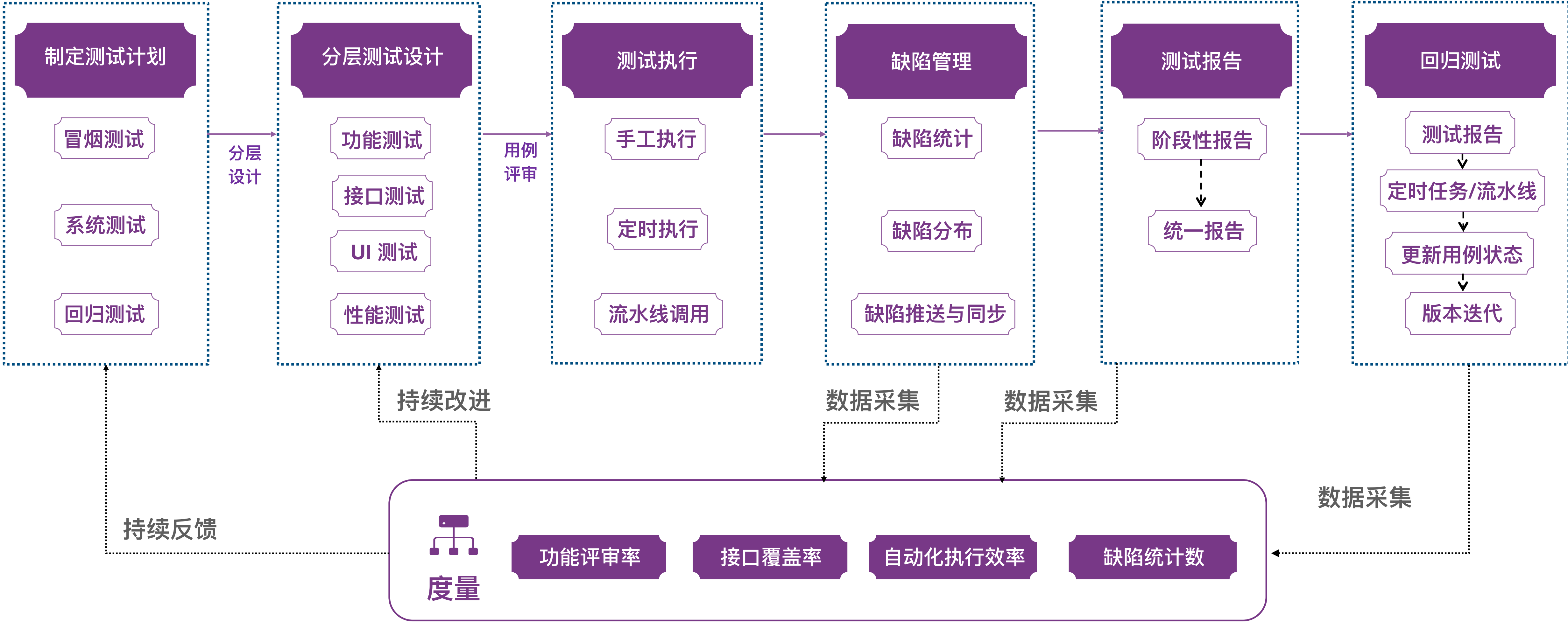
大规模的性能测试能力

五、团队协作、融入 DevOps

融入业务交付线能力

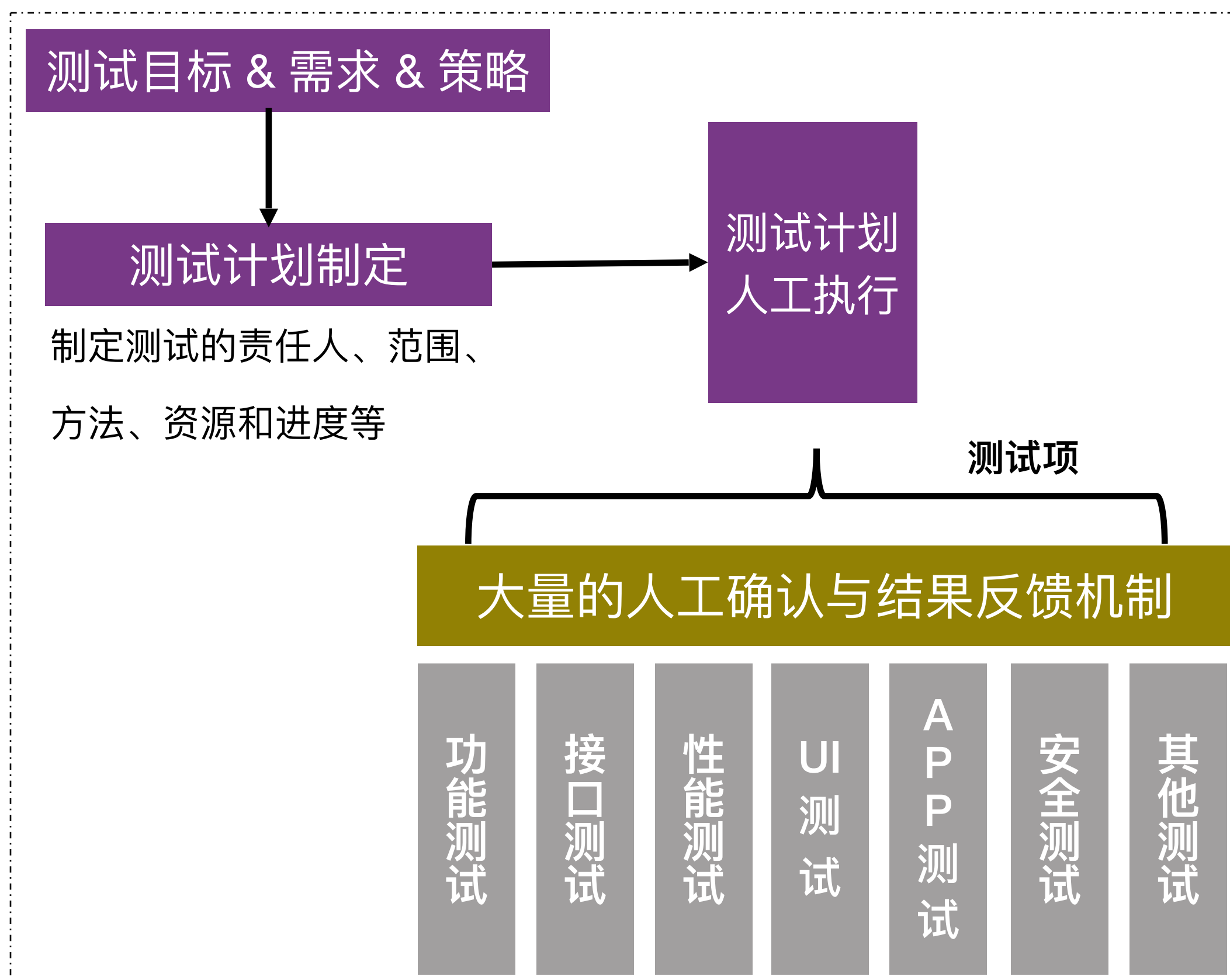
注：测试成熟度模型参考《持续测试白皮书》。白皮书中定义了测试成熟度的四个等级：
Level 1:手动为主；Level 2：初步尝试；Level 3：实践阶段；Level 4：熟练运用。

MeterSphere 测试过程全跟踪与度量

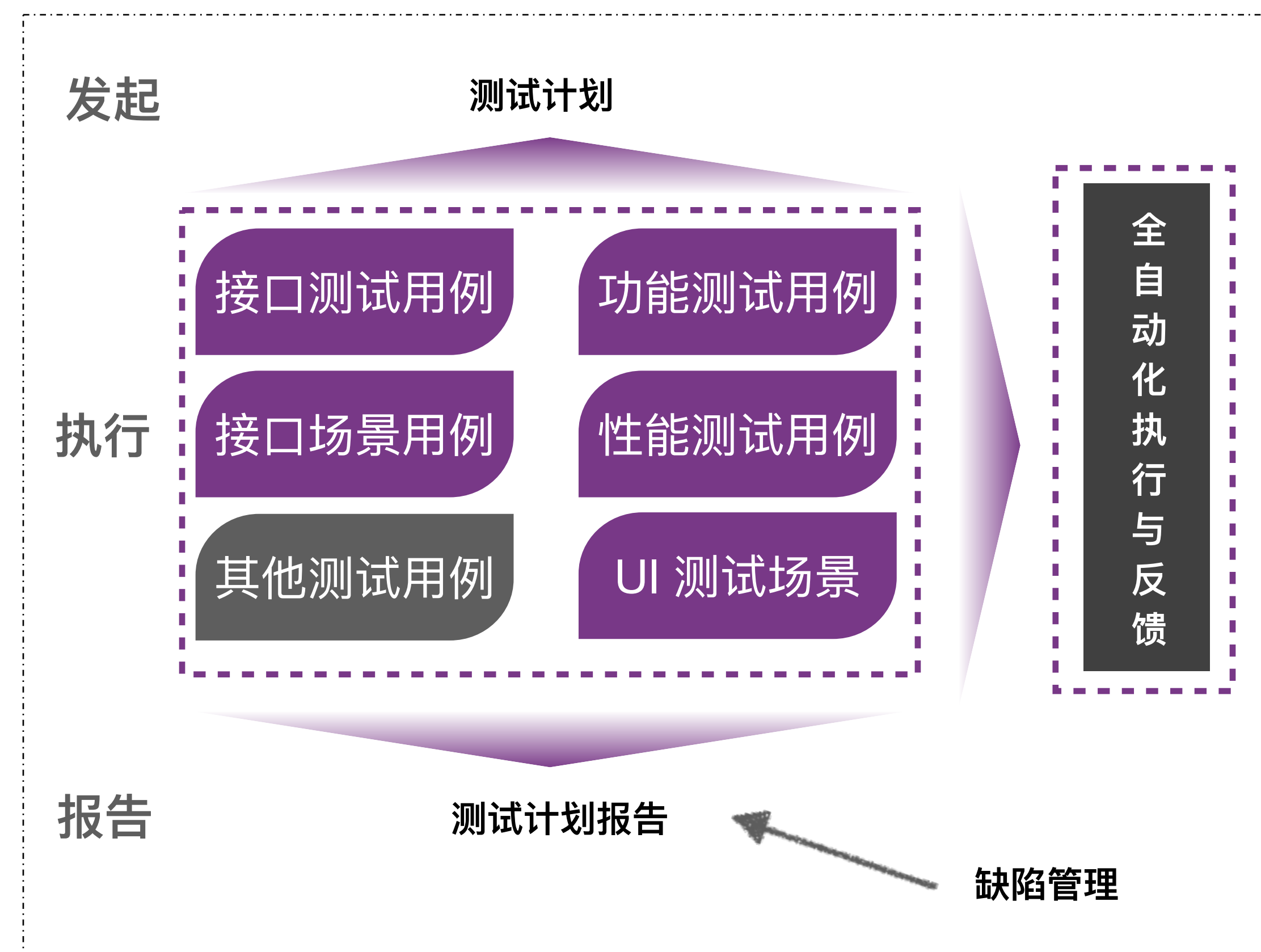


MeterSphere 测试管理核心：一体化全场景的测试计划

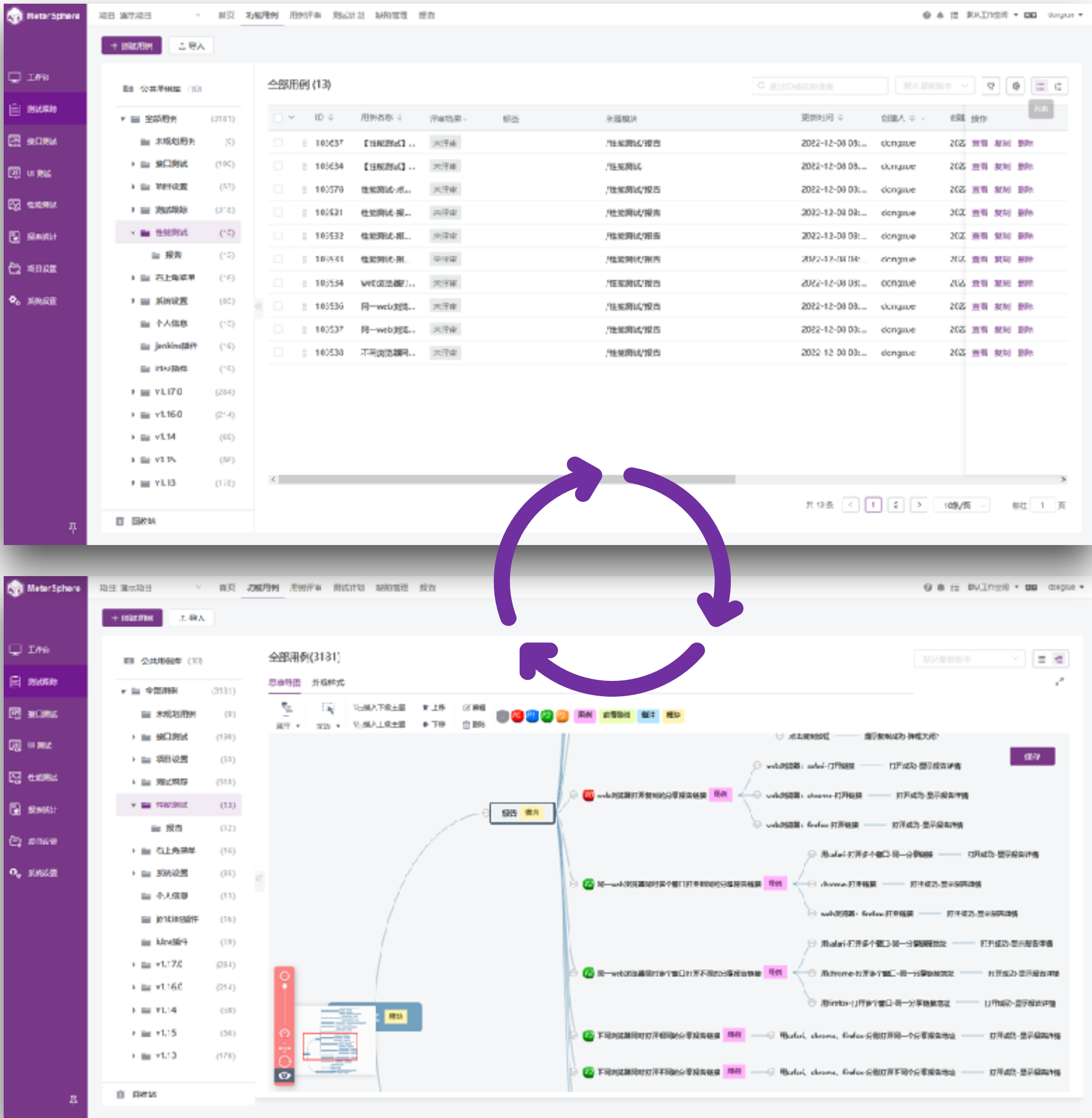
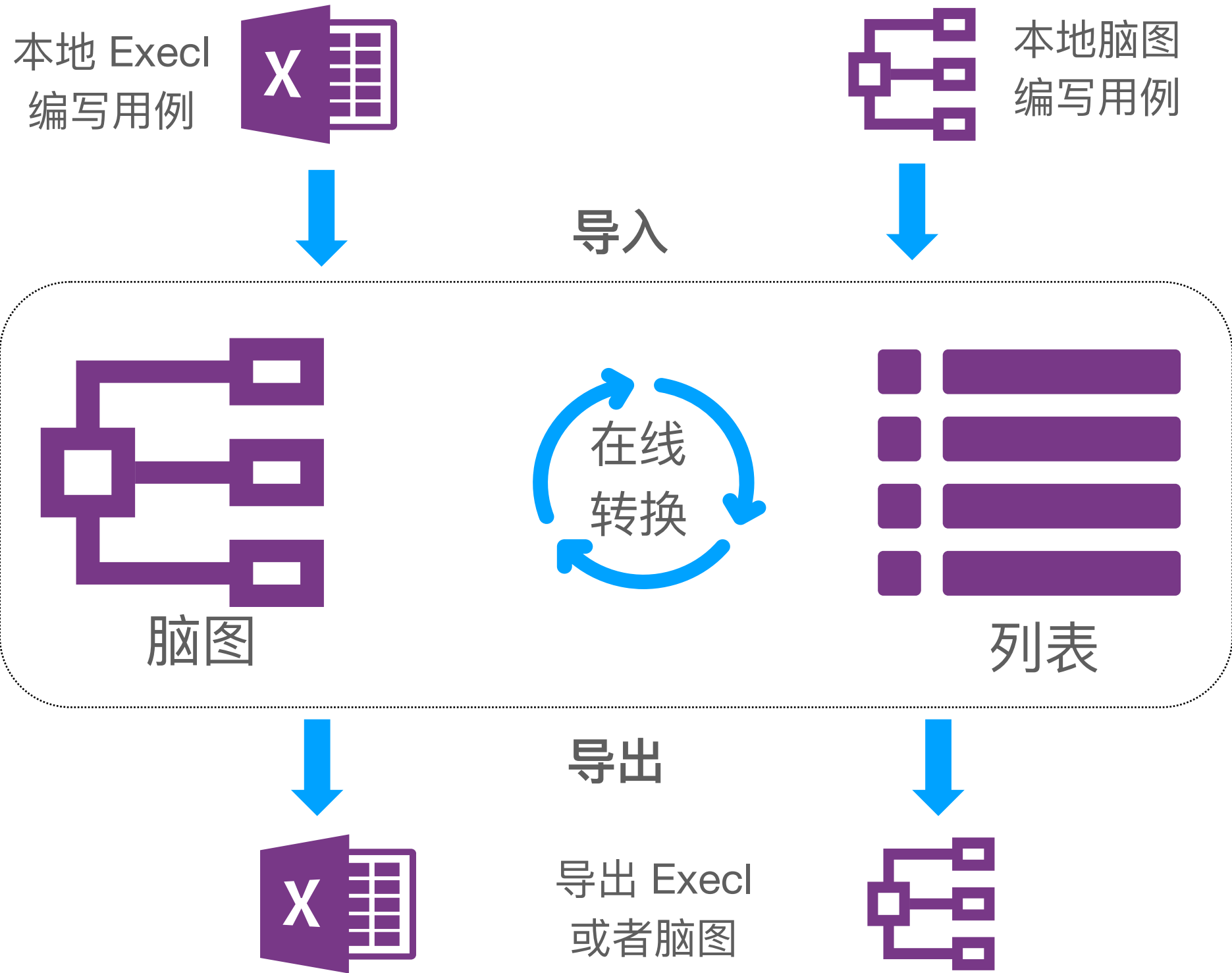
人工模式的测试计划执行



MeterSphere 中的一体化测试计划

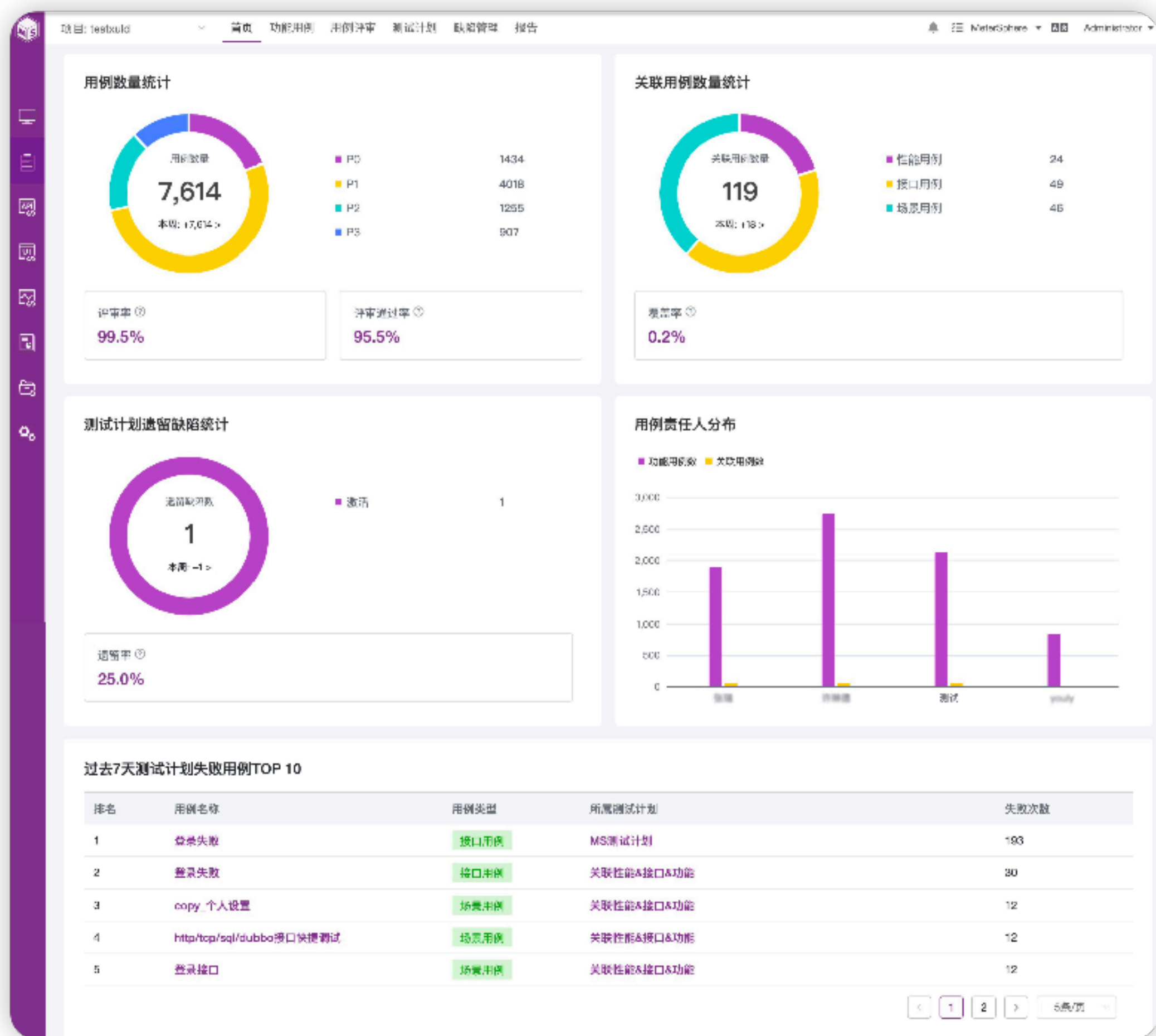


MeterSphere 灵活的功能用例编写维护



MeterSphere 测试管理核心：测试任务量化

- 对测试任务进行跟踪，量化测试任务进度与测试数据 -



The dashboard displays a list of test plans with the following columns: 名称 (Name), 状态 (Status), 测试阶段 (Test Phase), 测试进度 (Test Progress), 所属项目 (Project), 计划开始 (Plan Start), 计划结束 (Plan End), and 操作 (Actions).

名称	状态	测试阶段	测试进度	所属项目	计划开始	计划结束	操作
MeterSphere_v1.20.0_Tes...	进行中	系统测试	99%	MeterSphere	2022-04-21 00:00:00	2022-04-27 00:00:00	[Icons]
MS-接口导入测试_第二轮	进行中	冒烟测试	99.8%	MeterSphere	2022-07-12 00:00:00	2022-07-13 00:00:00	[Icons]
MeterSphere_v1.19.0	已结束	冒烟测试	100%	MeterSphere	2022-03-15 00:00:00	2022-03-24 00:00:00	[Icons]
回归测试-彭斌	已结束	冒烟测试	100%	MeterSphere			[Icons]
MeterSphere_v1.16.0	进行中	系统测试	99.9%	MeterSphere	2021-12-16 10:08:56	2021-12-22 19:00:00	[Icons]
UI测试	未开始	冒烟测试	0%	MeterSphere			[Icons]
MeterSphere_v2.2.0	进行中	系统测试	88.1%	MeterSphere	2022-09-18 00:00:00	2022-09-22 00:00:00	[Icons]
1	进行中	冒烟测试	0%	MeterSphere			[Icons]
MS_v1.18	已结束	冒烟测试	100%	MeterSphere	2022-02-16 00:00:00	2022-02-24 00:00:00	[Icons]
测试状态	已结束	冒烟测试	100%	MeterSphere			[Icons]

接口测试的现状与问题：接口测试多样化

01

工具型接口测试

优势：

简单，易用，普及度广；

缺点：

规范难以制定，依赖个人的测试习惯；

场景：

灵活，适合个人。

02

框架型接口测试

优势：

灵活度高，功能多；

缺点：

维护与编写成本高，更加依赖测试人员个人能力；

场景：

灵活的业务场景与测试方式。

03

引入平台型接口测试

优势：

适合团队，方便度量和集成；

缺点：

闭源商业化平台价格较贵，开源平台的稳定性欠佳和功能较少；

场景：

适合团队和集成 DevOps 等。

04

自研型接口测试

优势：

需求满足度高，自主可控；

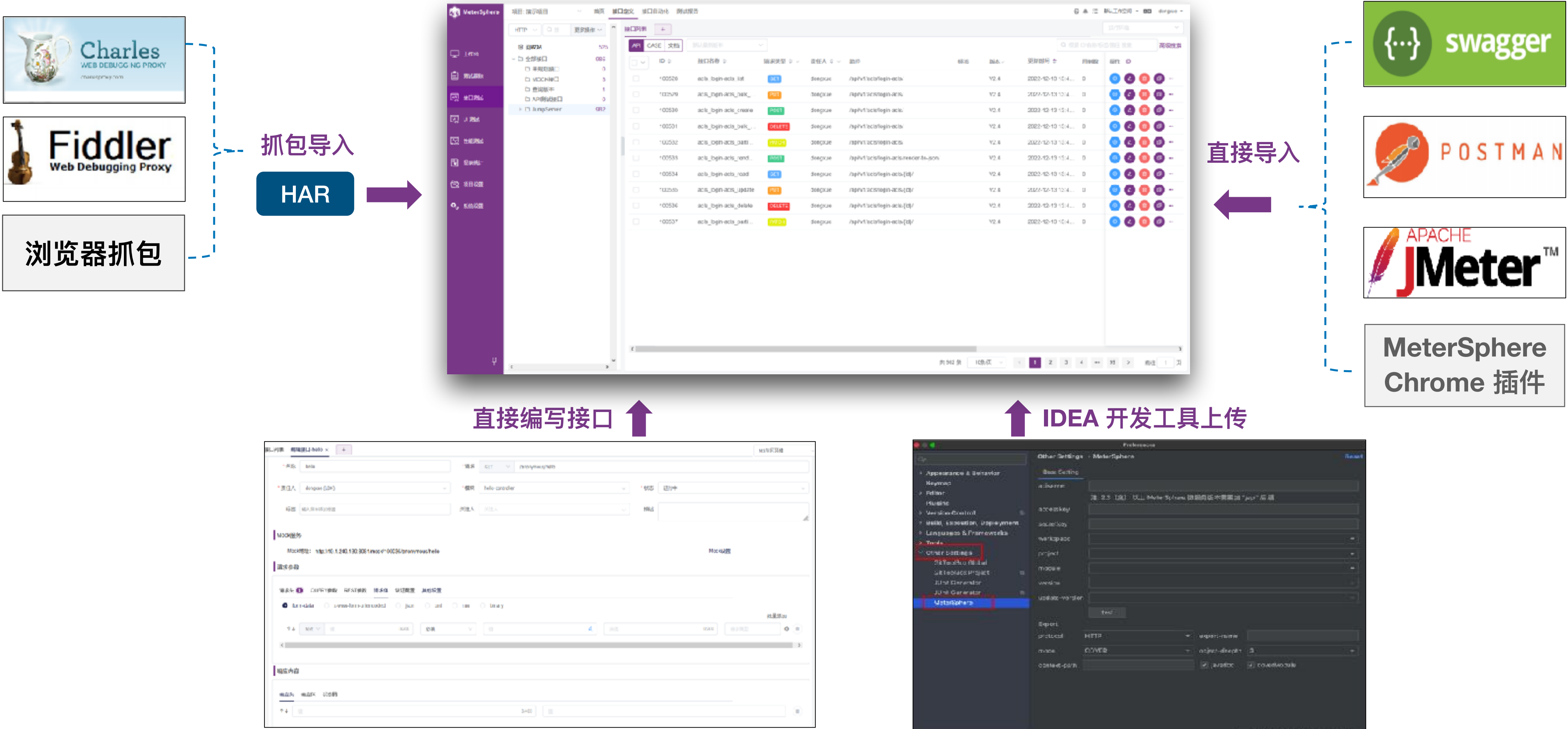
缺点：

投入产出比低，研发人力成本级别高；

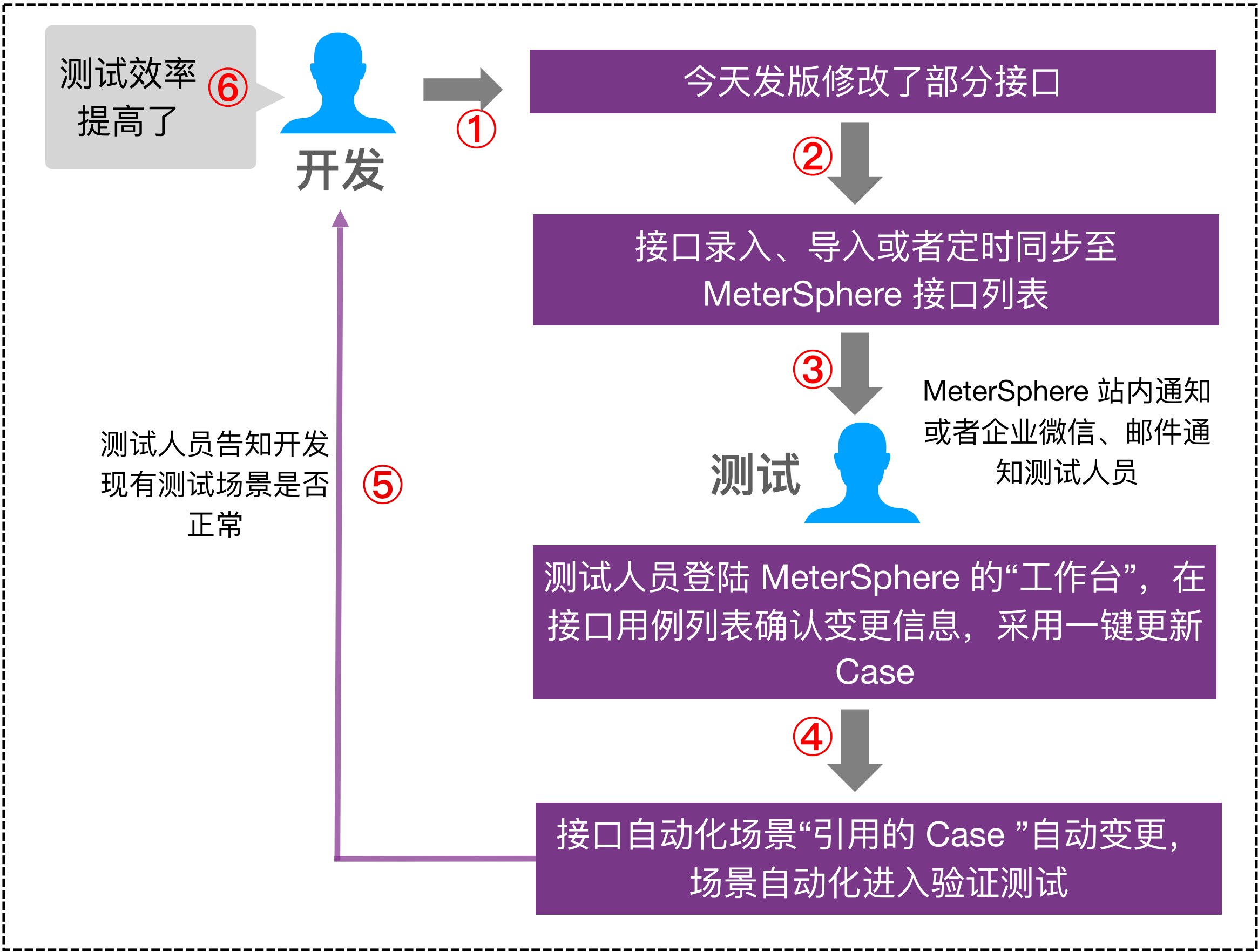
场景：

适合团队和集成 DevOps。

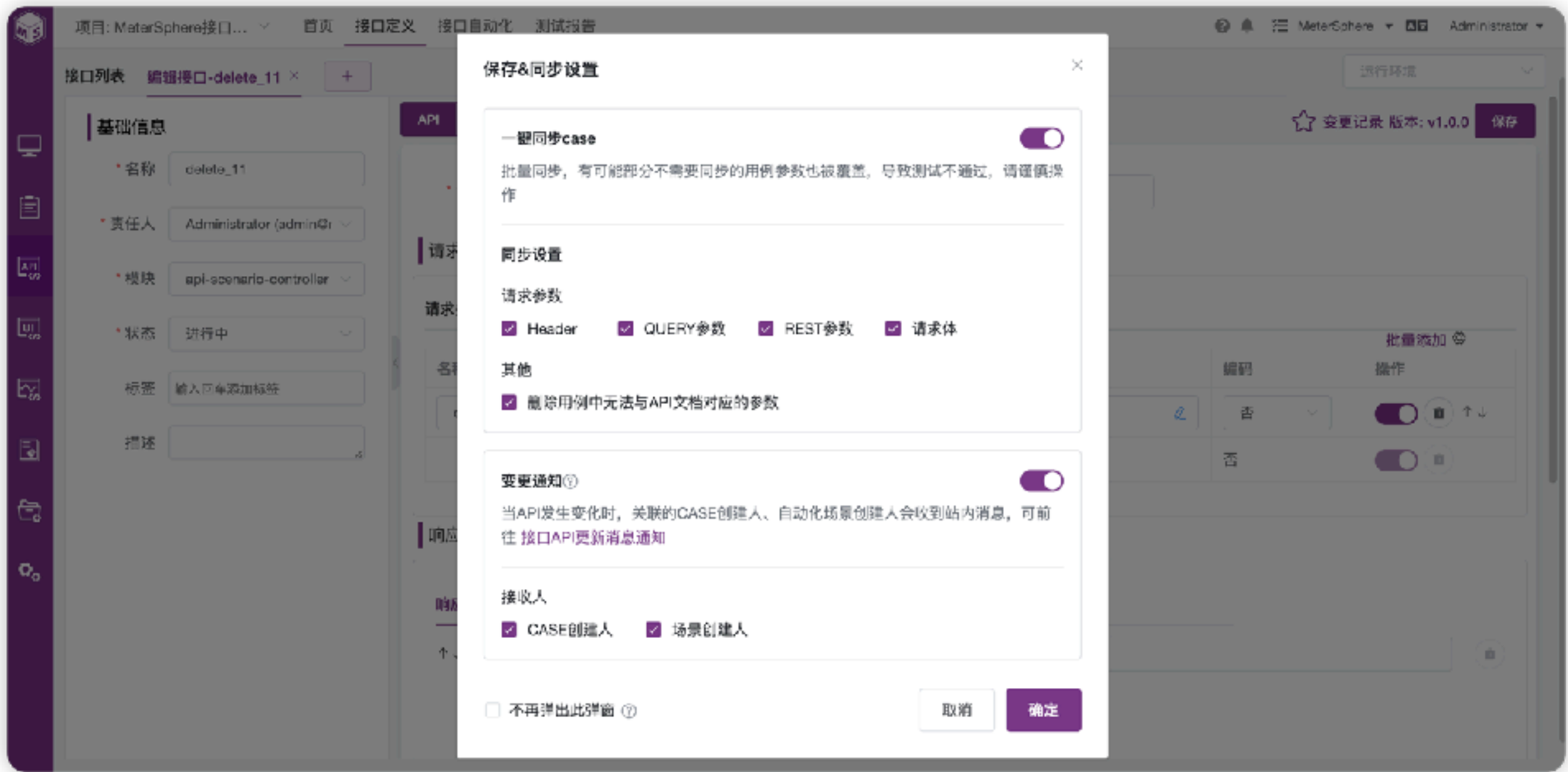
统一的接口定义与维护管理



高效的维护接口测试用例和自动化场景



同步要素将变更内容一键同步给用例和场景



MeterSphere 灵活的接口场景编排设计

- 解决现有接口测试的问题，覆盖多种接口协议，自由组合接口自动化测试场景 -

插件扩展协议

MQTT

WebSocket

Debug

GRPC

FTP

自定义扩展

↑ 按需扩展

开发说明

UI生成器: <http://www.form-create.com/designer/?fr=home>

UI组件说明: <https://form-create.com/v2/element-ui/global.html>

插件入口: 和jar包同名配置, 内容格式: 如果多个入口, 按照每行一个入口

Upload组件特殊用法: 用UI生成器生成后 修改 type=msupload 即可

动态数据初始化: 组件增加事件 change, click 等事件 emit参数: name = 事件 inject为传递参数: paramName 为当前组件的field 名称, 第二个参数是JSON串, 可任意格式, 可在 UiscriptApi.customMethod 的入参接收; 如需获取表单中其他组件值当参数传递 可用 \$field 名称 这种方式提取 omitPrefix参数: prefix 为固定前缀, 平台会监听 prefixvisiblechange, prefixclick 等事件

完整示例: "emit": [{"name": "visible-change", "inject": [{"paramName", "[method":"initInterleave", "去参数" "到field"]"}], "omitPrefix": "prefix",

自动化场景编排

- 场景管理与环境配置
- 测试步骤拖拽式编排
- 丰富的测试组件
- 场景嵌套与拼接

接口自动化任务
下发执行

接口自动化转性能
测试任务下发执行

虚拟机集群

主机-1
Node-Controller

主机-2
Node-Controller

主机-3
Node-Controller

主机-N
Node-Controller

Kubernetes 集群

JMeter-Pod-1

JMeter-Pod-2

JMeter-Pod-3

JMeter-Pod-N

调度机制可以动态扩容，
适配团队大规模使用。

内置协议

HTTP

TCP

SQL

Dubbo

接口管理

- 接口集管理
- PostMan 导入
- Swagger 导入
- 快捷调试

接口用例

- 用例集管理
- 用例运行管理
- 前后置脚本
- 断言与提取

Mock 服务

- 期望列表
- 请求参数
- 响应状态码
- 请求头
- 请求内容

逻辑组件

- 等待控制器
- 循环控制器
- 条件控制器
- 事务控制器

函数代码库

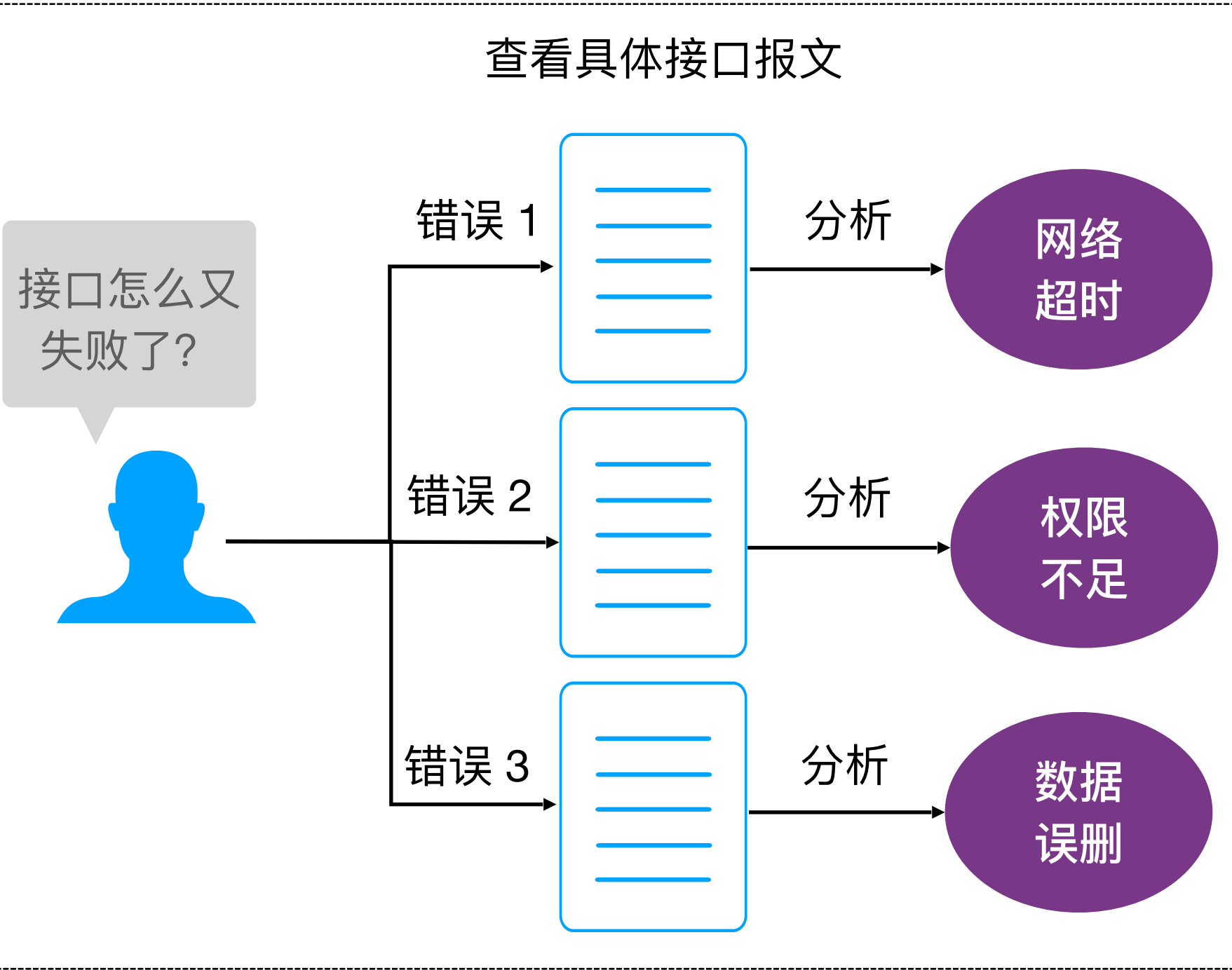
- 兼容 JMeter 函数
- 内置 MockJS 函数
- 自定义脚本
- 内置代码库管理

1:N

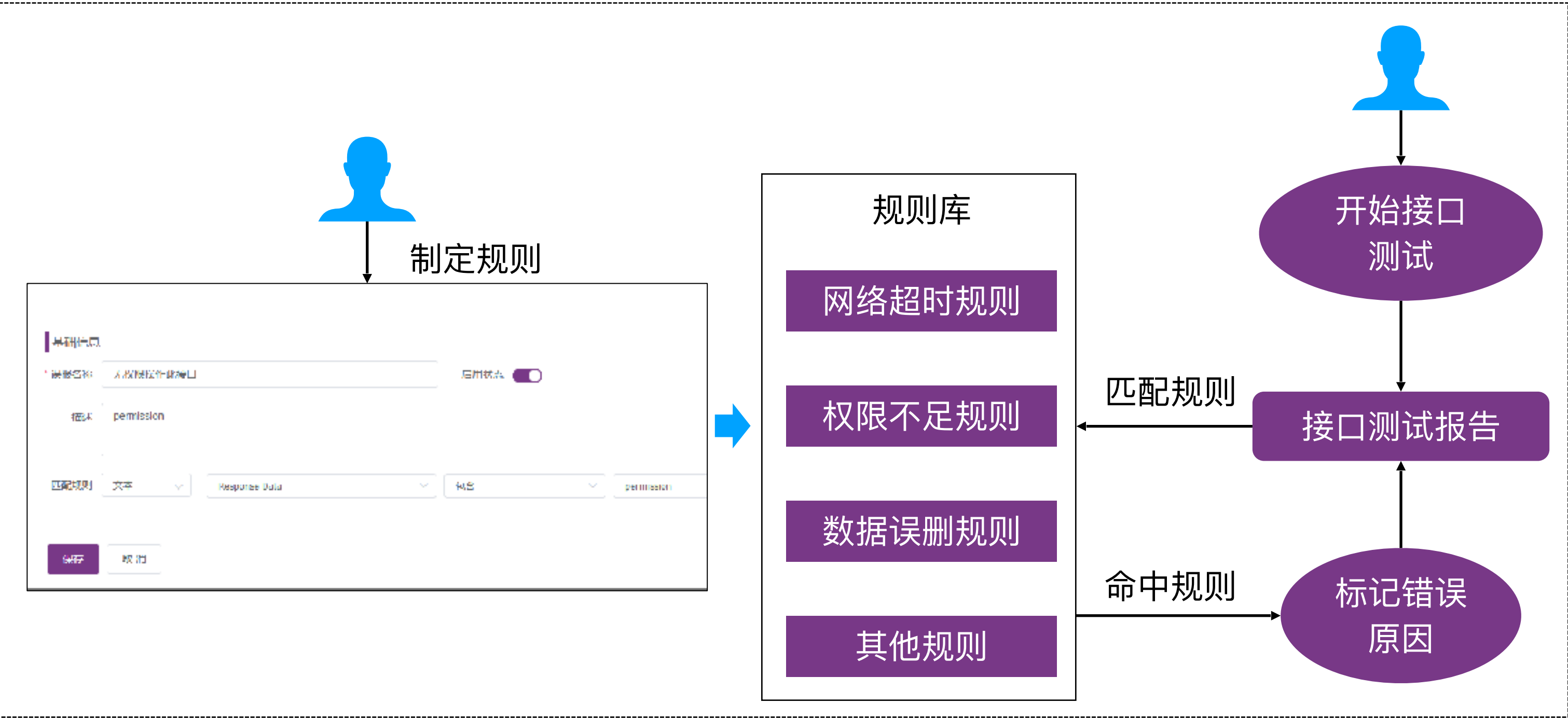
MeterSphere 沉淀问题分析能力的测试误报功能

- 误报规则是问题分析能力的积累，规则越多越丰富，误报识别能力越强 -

以前的接口测试问题定位

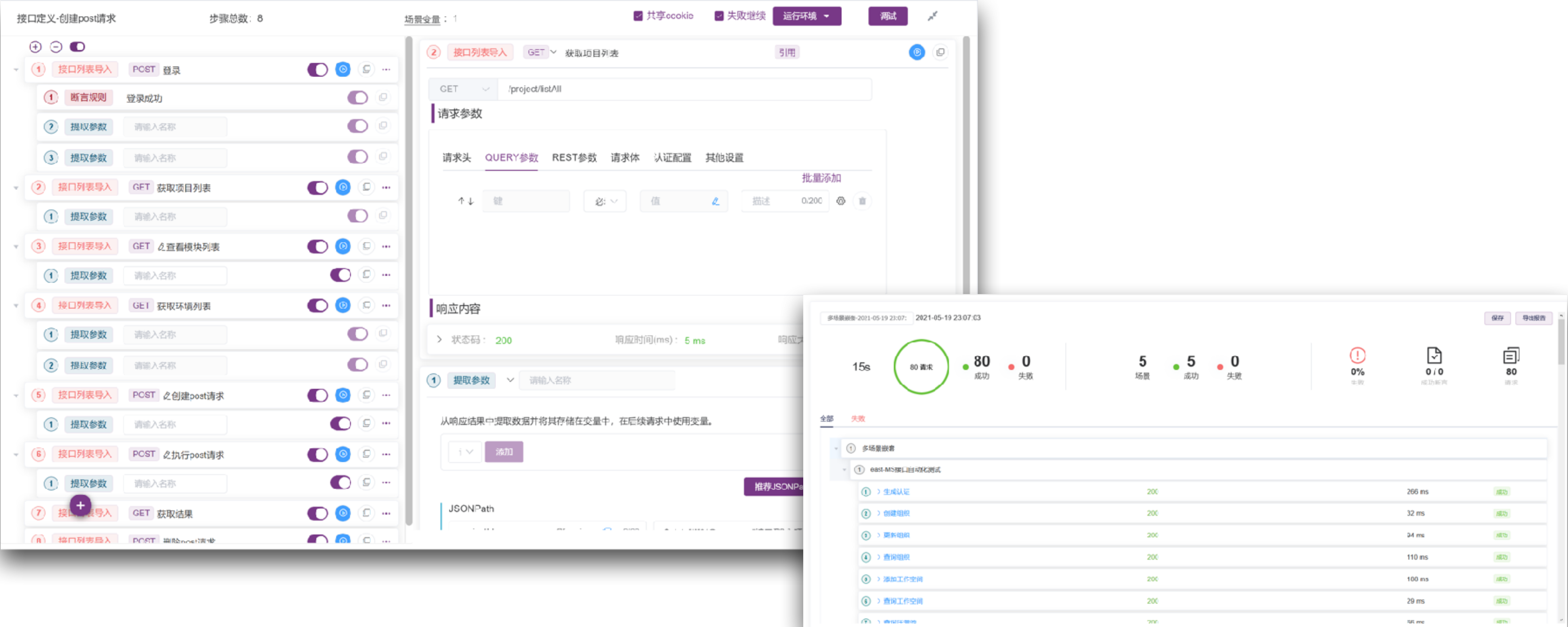


MeterSphere 中接口测试问题误报

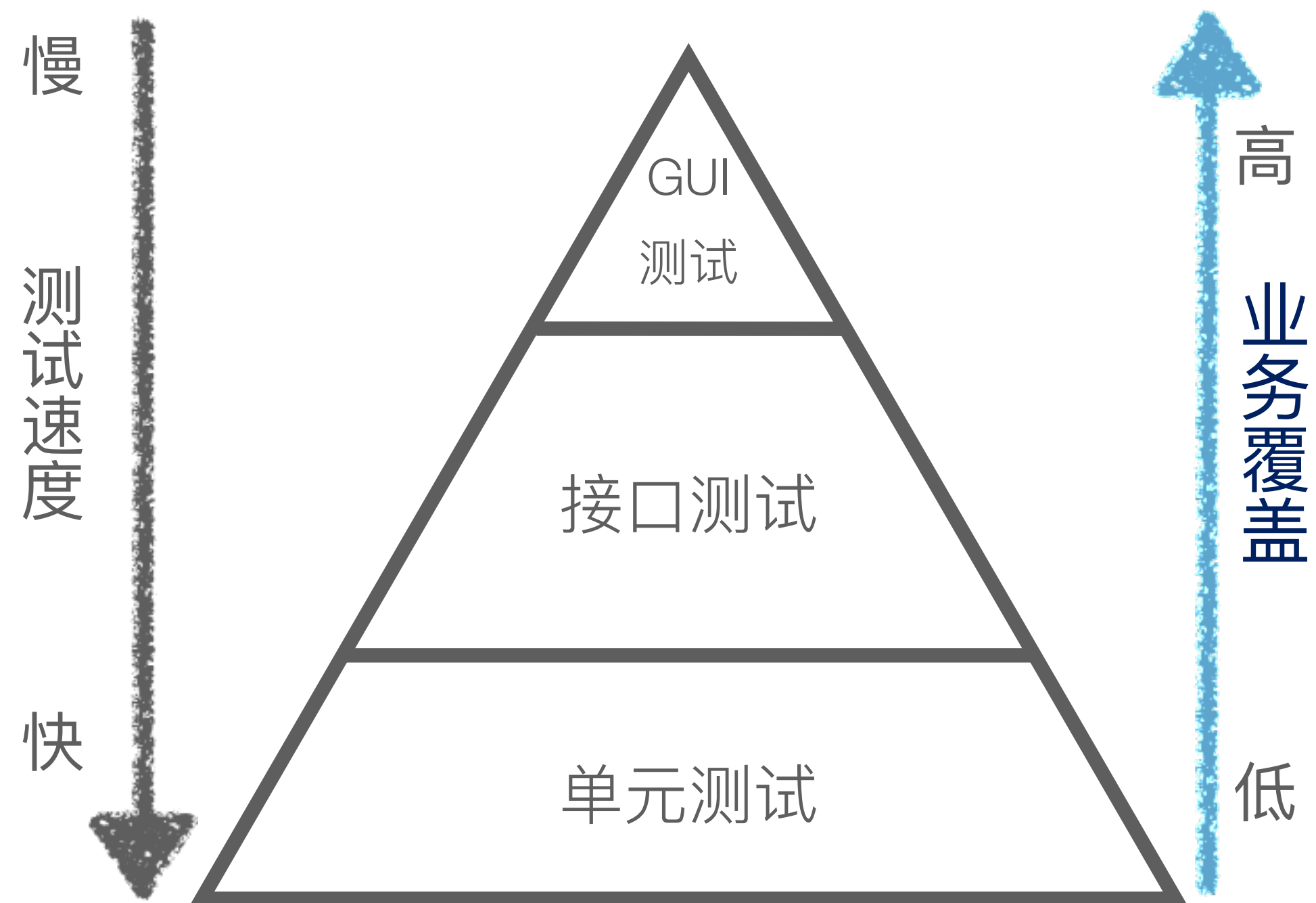


MeterSphere 接口场景编排截图

- 简单易用的图形化接口自动化编排 -

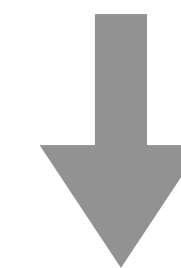


UI 自动化的价值与问题



UI 自动化测试建设价值依旧是遵循测试金字塔原理。MeterSphere 开源社区调研了的 219 份企业表单，其中 82.19% 的报名者所在的团队正在建设 Web 端 UI 自动化测试系统。

维护成本
+
学习成本



不是不想做 UI 自动化，而是如何解决两大核心问题？

MeterSphere 的自然语言 UI 自动化

- 自然语言编写，低学习成本 -

编写代码实现

```
#!/usr/bin/python
#coding=utf-8

from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import Select
import time
# 创建Chrome驱动实例
options=webdriver.ChromeOptions() # 创建Chrome驱动实例
browser = webdriver.Firefox()

#浏览器最大化
browser.maximize_window()

#get方法 打开指定网址
browser.get("https://jms1.fit2cloud.com/core/auth/login/")

#断言网页标题是否含有JumpServer
assert "欢迎使用东区JumpServer" in browser.title

#输入用户名
browser.find_element_by_name("username").send_keys('XXXXXX')

#输入密码
browser.find_element_by_id("password").send_keys('XXXXX')

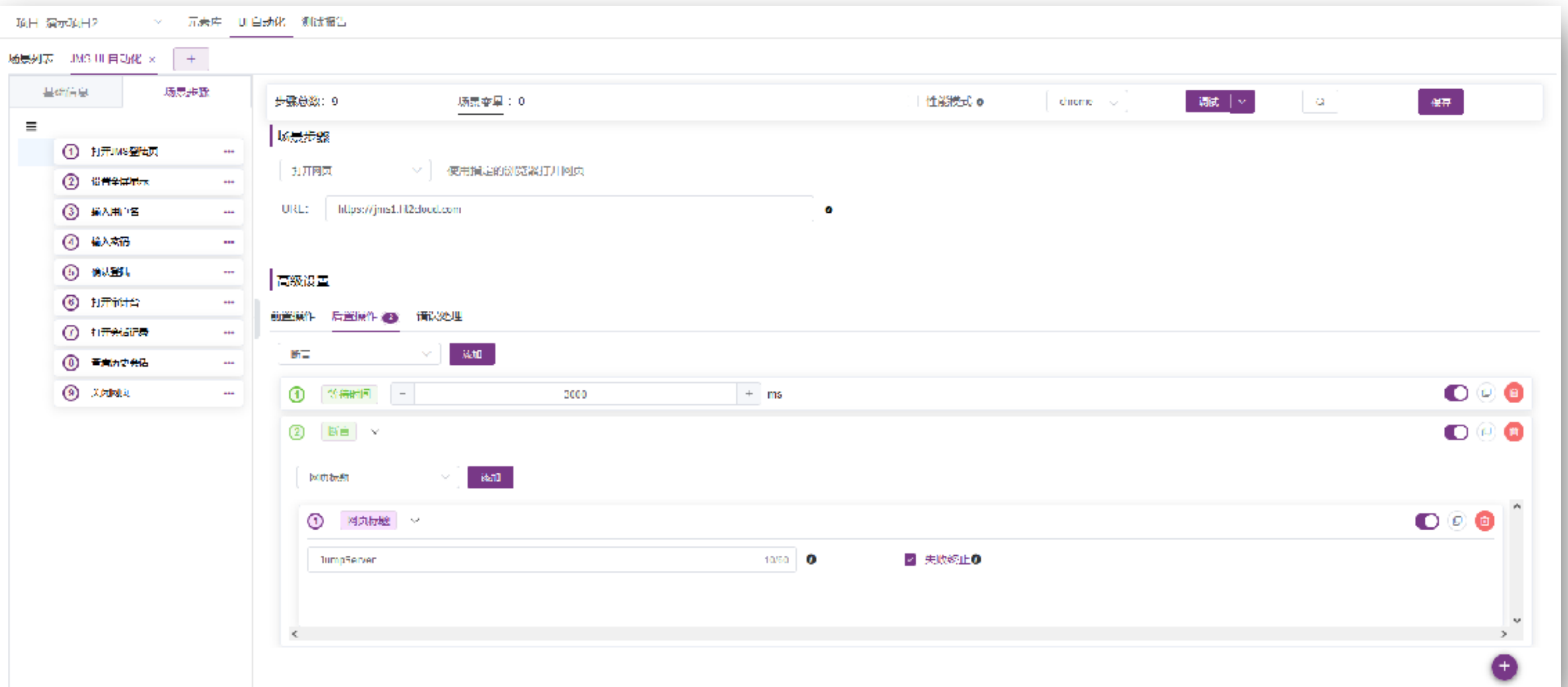
#登陆
browser.find_element_by_xpath("/html/body/div/div[2]/div/div[2]/form/div[5]/button").click()

#打开审计台
browser.get("https://jms1.fit2cloud.com/ui/#/audit/dashboard")

#设置等待时间
time.sleep(2)

#关闭浏览器
browser.quit()
```

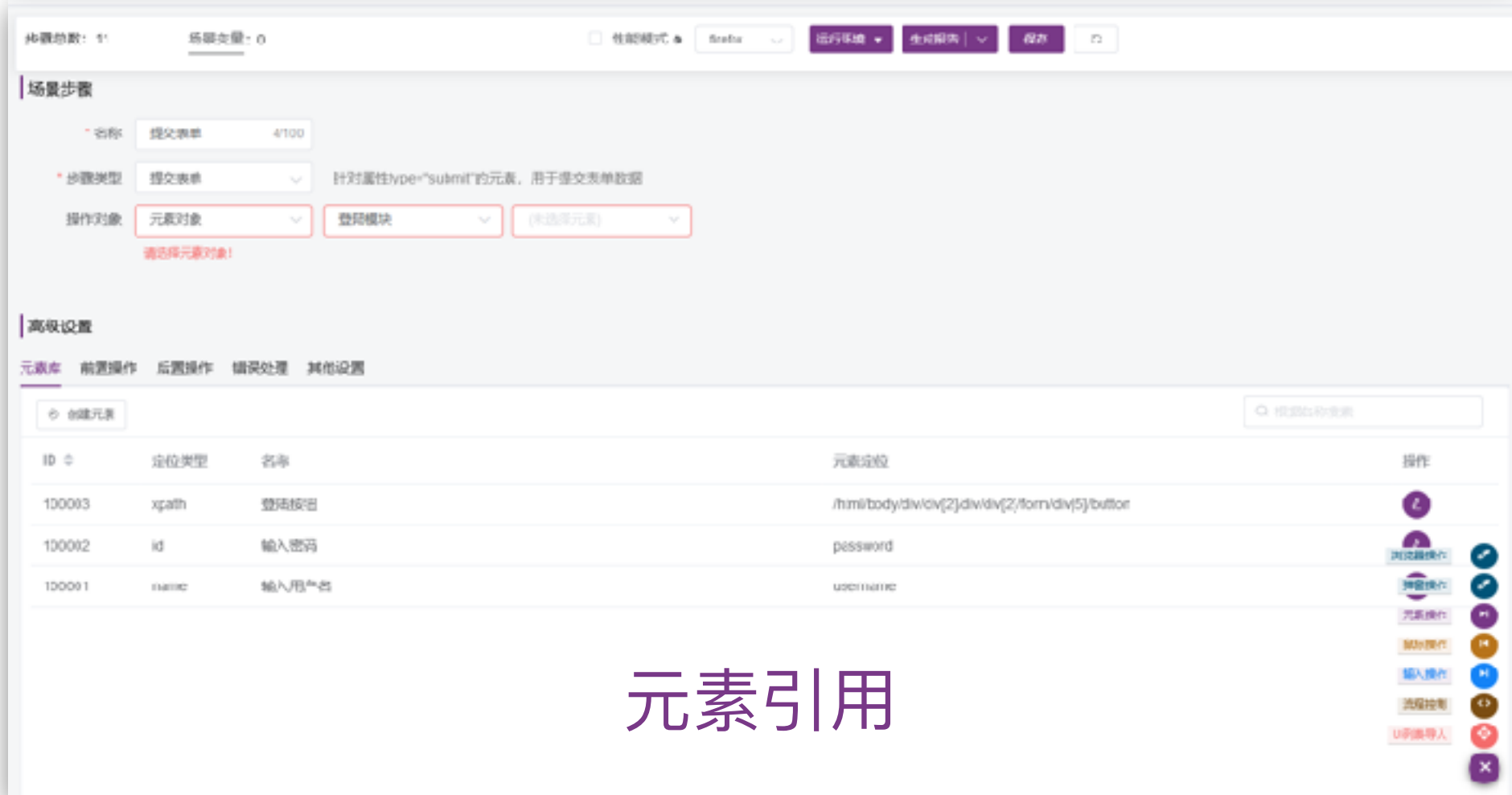
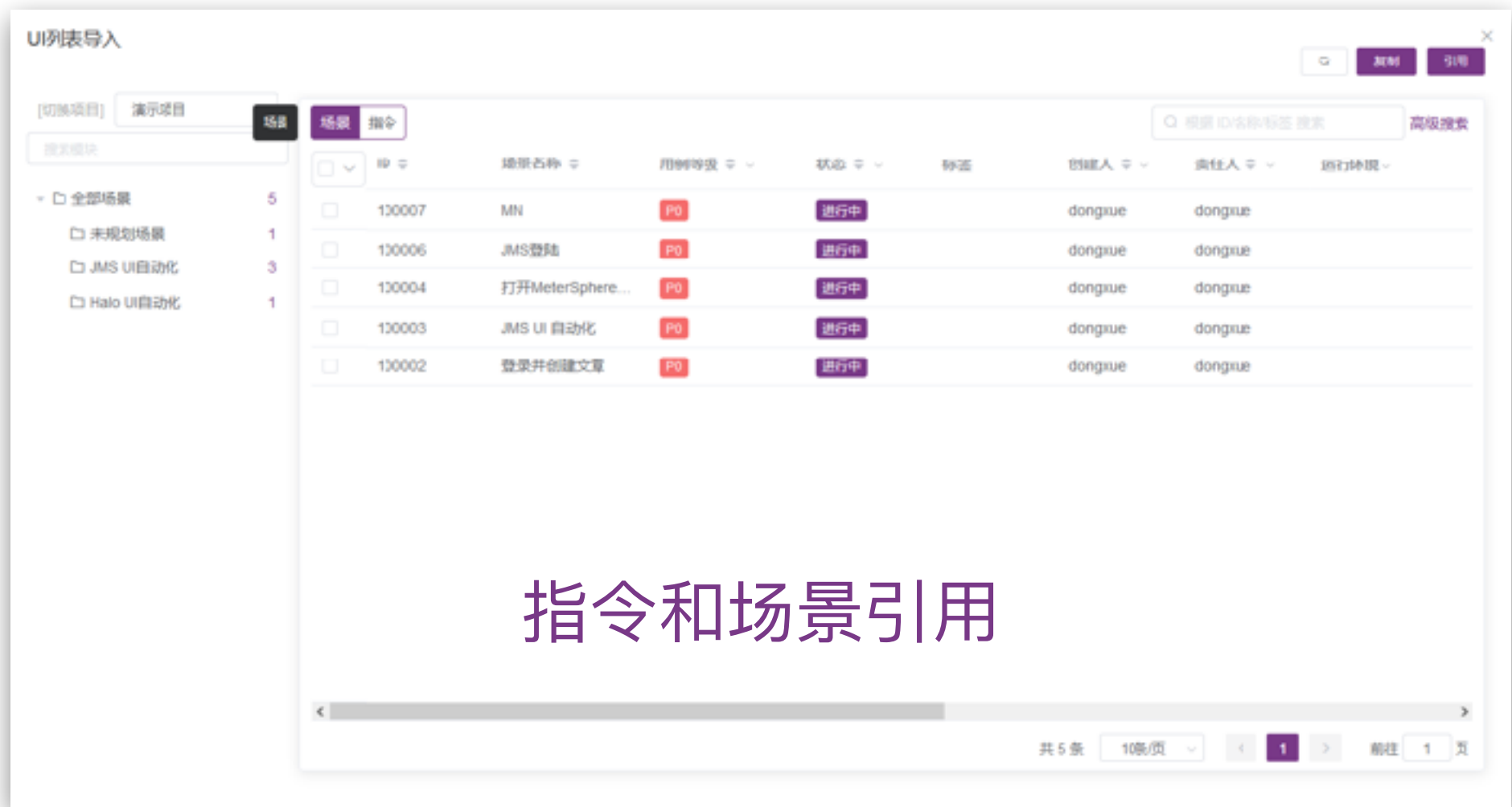
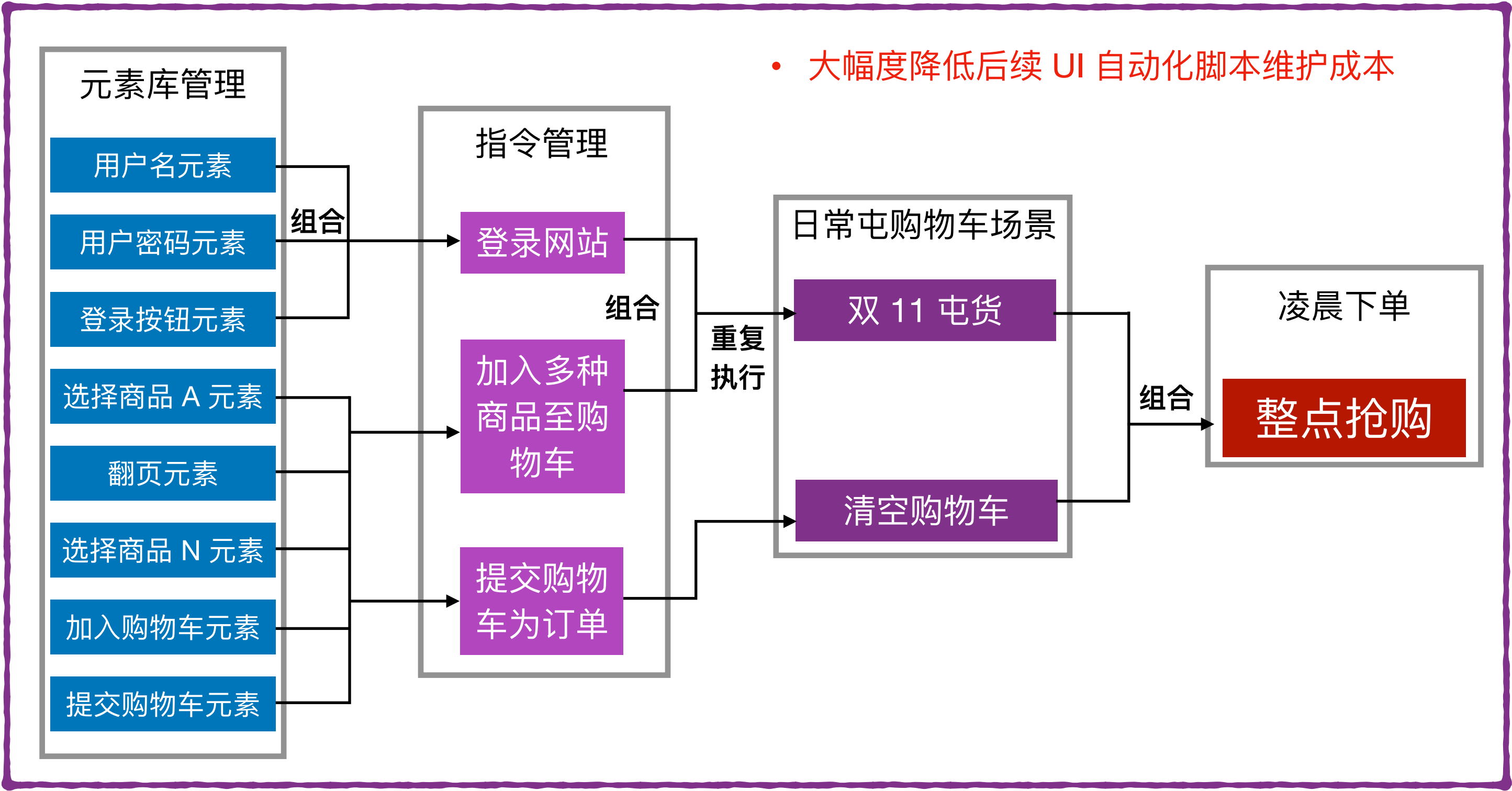
MeterSphere 界面自然语言编排实现



MeterSphere 的自然语言 UI 自动化

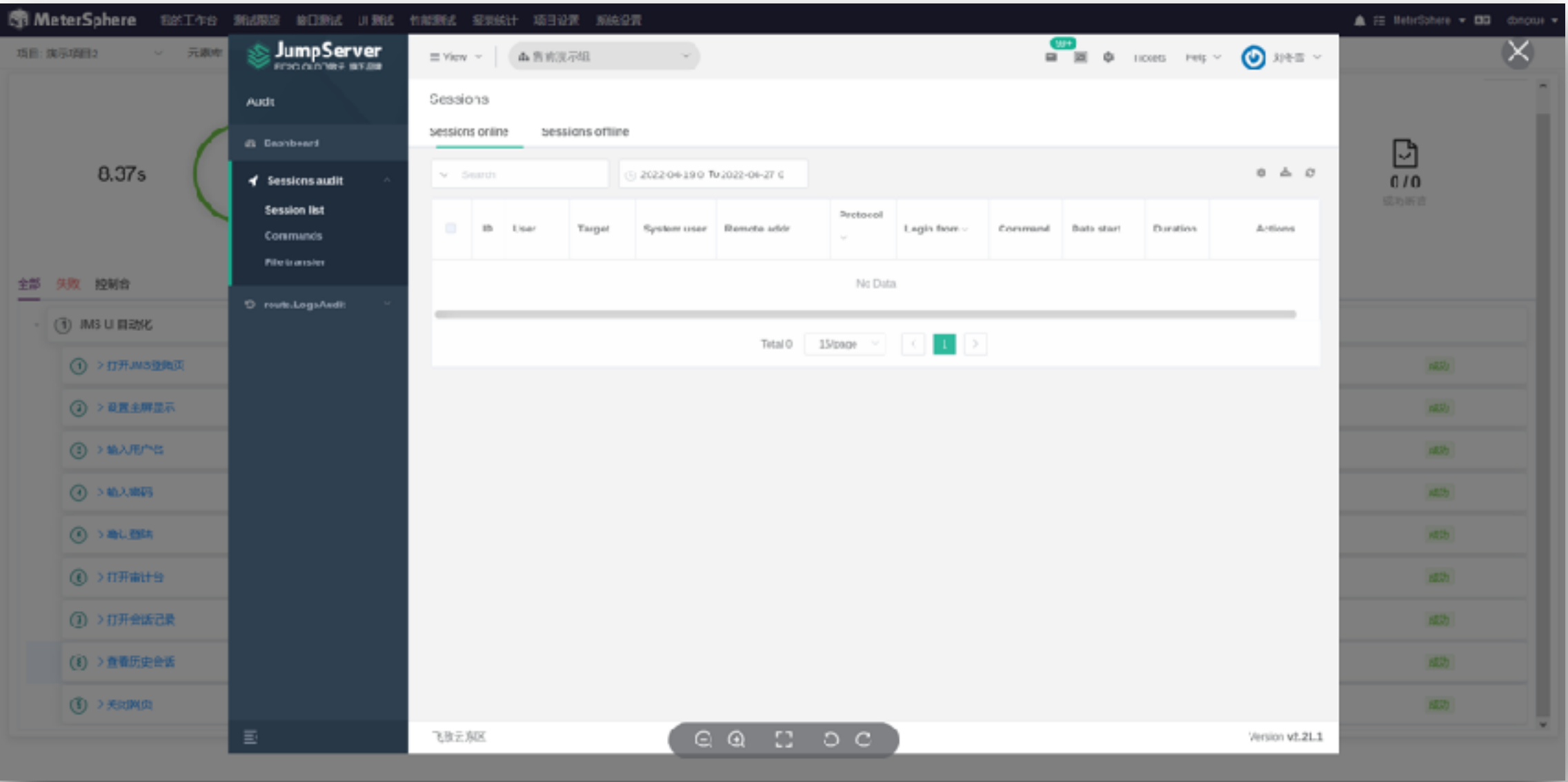
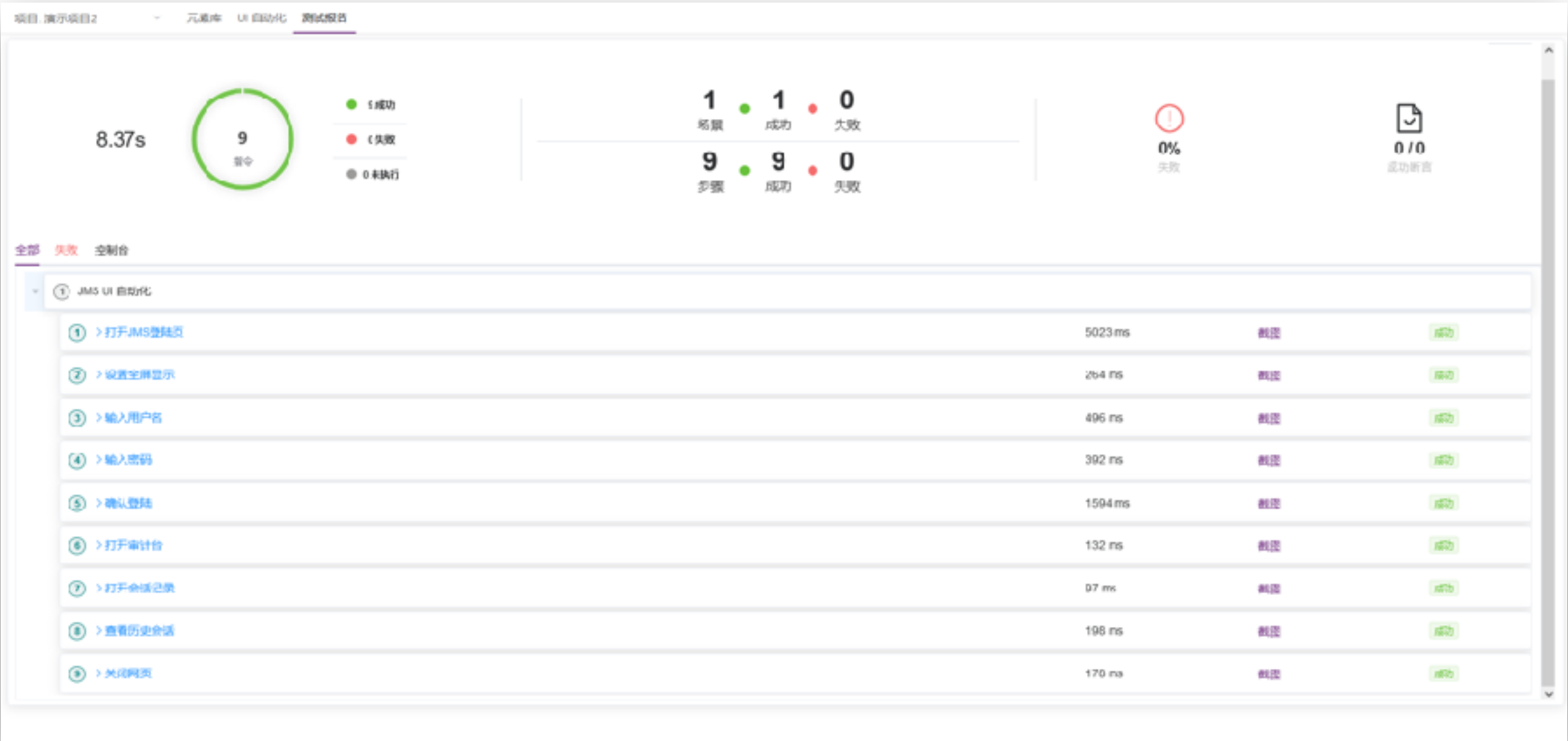
- 元素、指令、场景自由组合引用，低维护成本 -

基于 MeterSphere 页面对象模式编写的“抢购” UI 自动化场景测试



MeterSphere UI 自动化运行报告

- 自动 UI 截图留存 + 可视化测试报告 -



关于性能测试企业需要什么？

- 互联网 + 的发展，性能测试逐渐成为了企业持续高效的测试类型之一 -

性能测试核心目标

- 找出系统性能瓶颈并进行优化；
- 预估系统性能瓶颈。

性能测试实施三要素

1. 结合业务实际需要明确压力测试的关键指标；
2. 以自动化的方式开展局部到全链路的压测；
3. 根据实际业务情况，适当引入外部专家服务。

服务

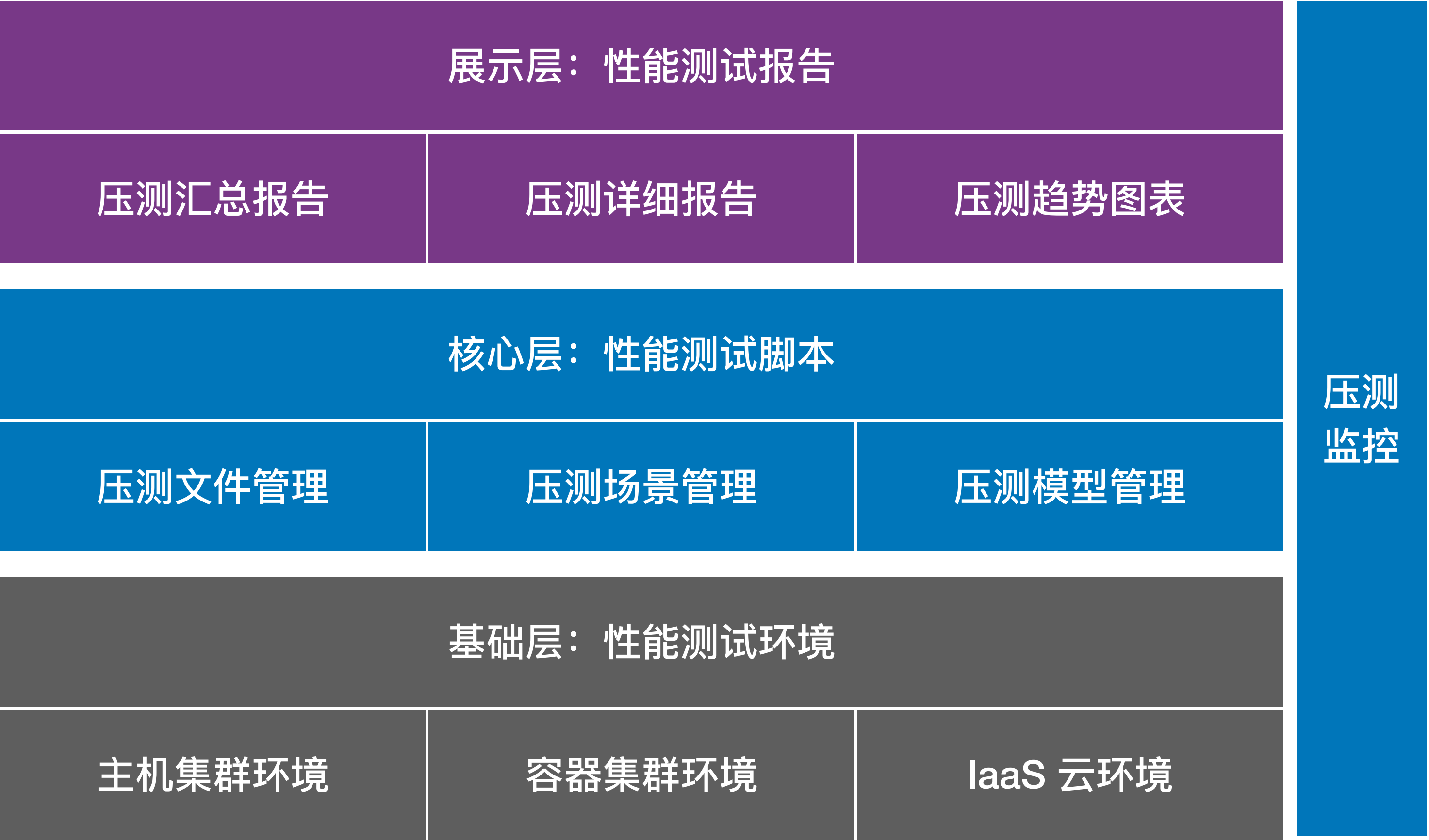
性能测试服务



产品

MeterSphere

MeterSphere 性能测试核心设计



性能测试报告

- 自动生成测试报告
- 多次测试结果对比
- 丰富的报告展示详情
- 测试报告内容导出

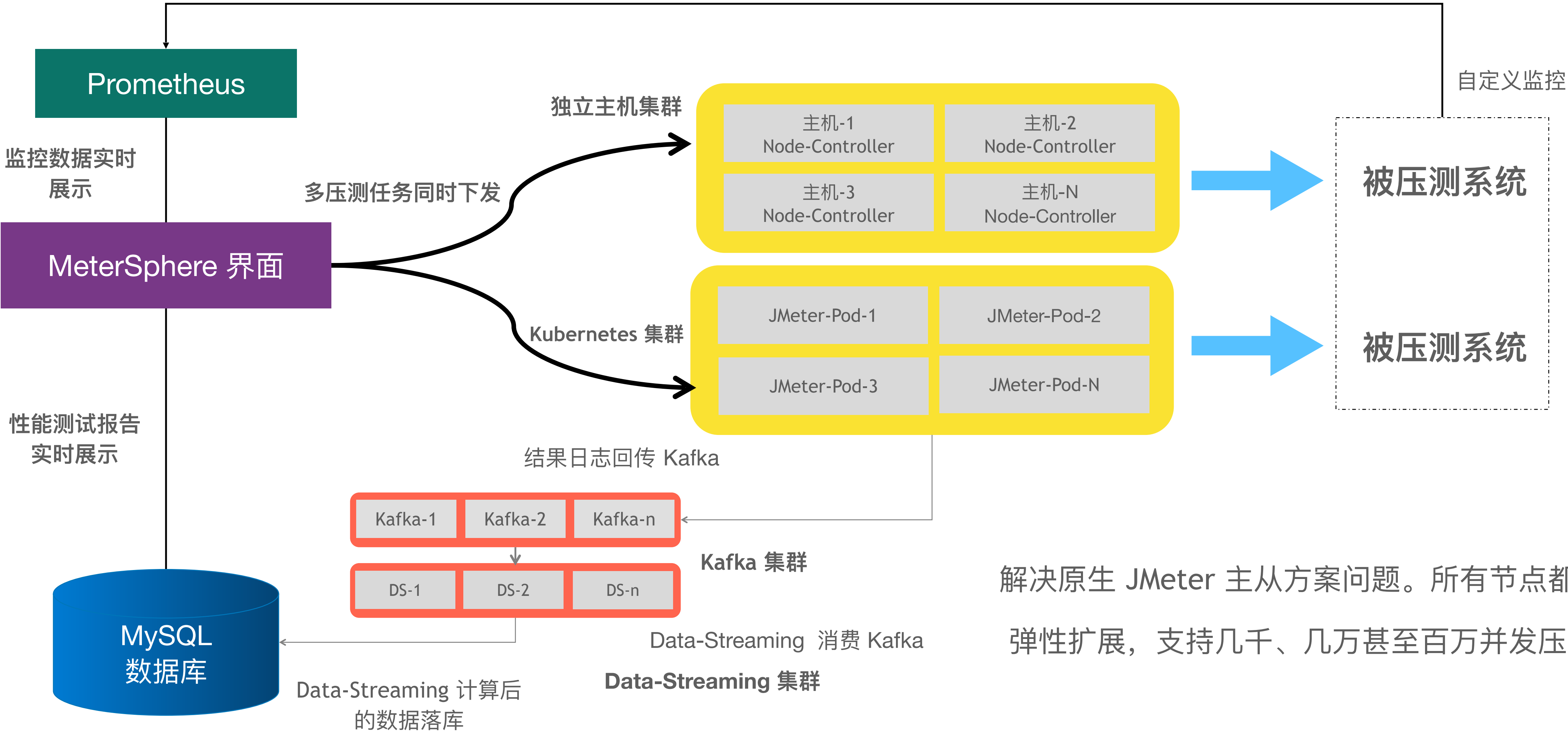
性能测试脚本

- 完全兼容 JMeter 脚本
- 自由上传测试依赖文件
- 通过浏览器插件快速录制

性能测试环境

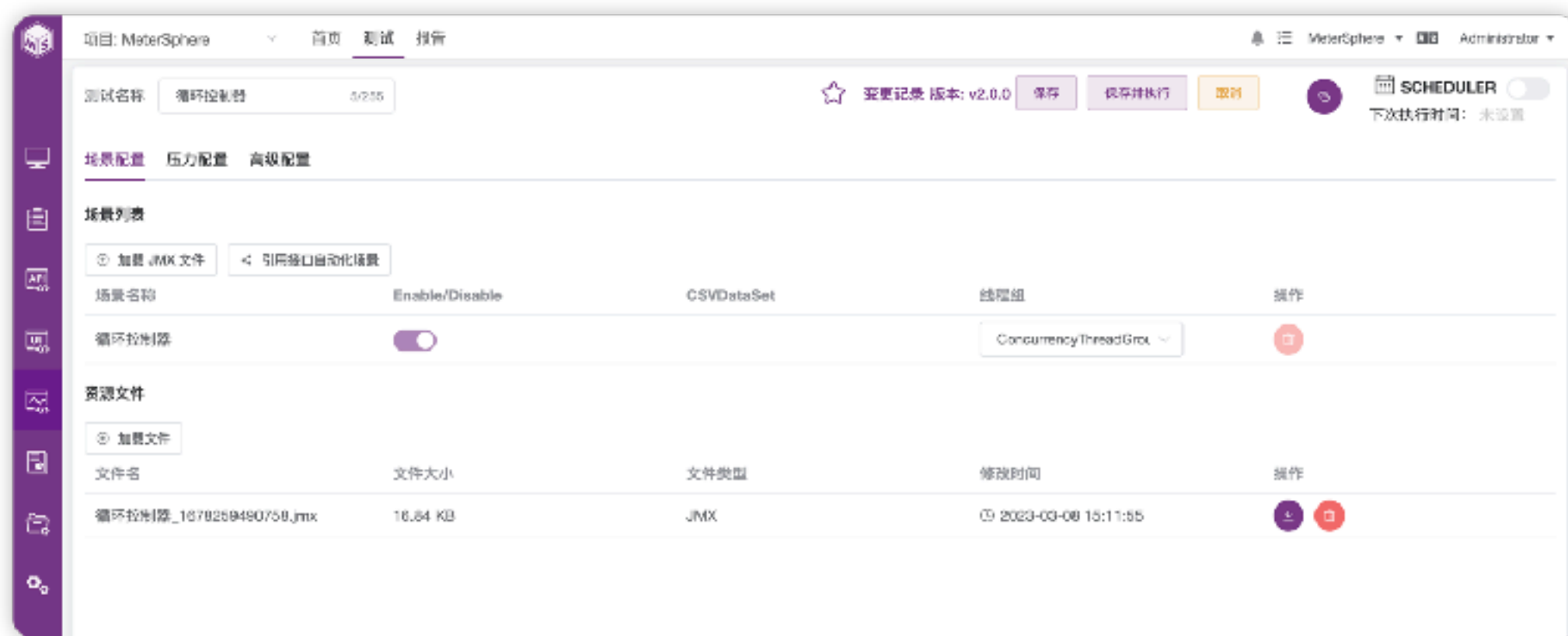
- 压测机统一管控
- 分布式、大规模压测资源池

MeterSphere 大规模压测架构设计



解决原生 JMeter 主从方案问题。所有节点都可以弹性扩展，支持几千、几万甚至百万并发压测。

MeterSphere 性能测试截图



项目: MeterSphere

测试

报告

MeterSphere > test3 > test3

立即停止

再次执行

导出报告

分享报告

报告对比

下载JTL

下载执行文件

执行时长: 0时1分10秒

开始时间: 2022-11-23 21:35:32

结束时间: 2022-11-23 21:36:43

10s

测试概览

测试详情

请求统计

错误记录

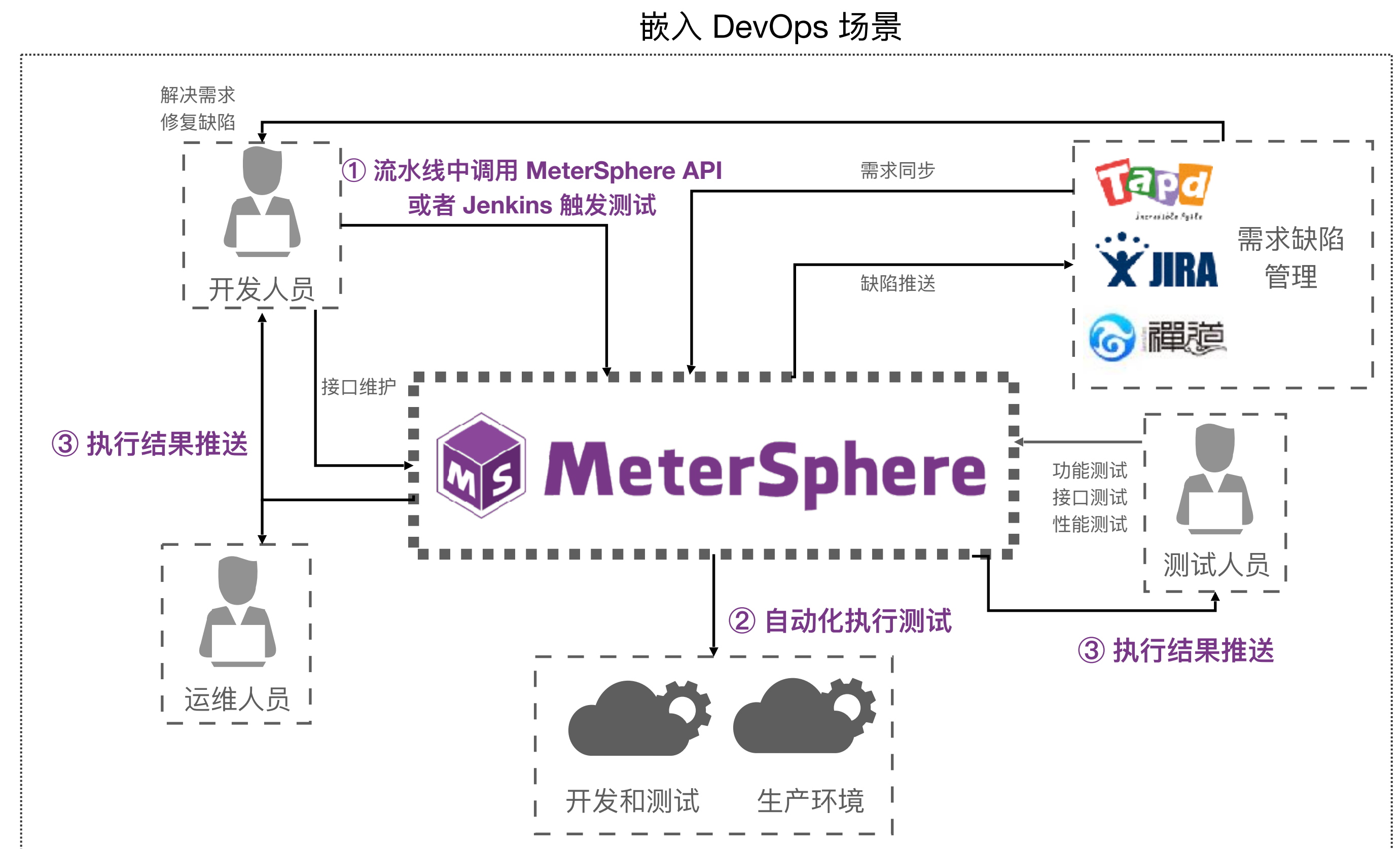
日志详情

监控详情

测试配置

Requests		Executions			Response Times[ms]							Throughput		NetWork(KB/sec)	
Label		Samples	FAIL	Error%	Avg	Min	Max	Med	90%	95%	99%	Trans/s	Recv	Sent	
cfcb6572-bd51-2814-6404-a	c365a789d5e	118713	0	0	11.21	1	6763	7	11	13	21	1675.625	6168.297	0	
Total		118713	0	0	11.21	1	6763	7	11	13	21	1675.625	6168.297	0	

- 将测试融入业务交付线，形成开发、测试、运维闭环 -



MeterSphere Jenkins 插件



支持自由风格与 Pipeline 类型构建任务



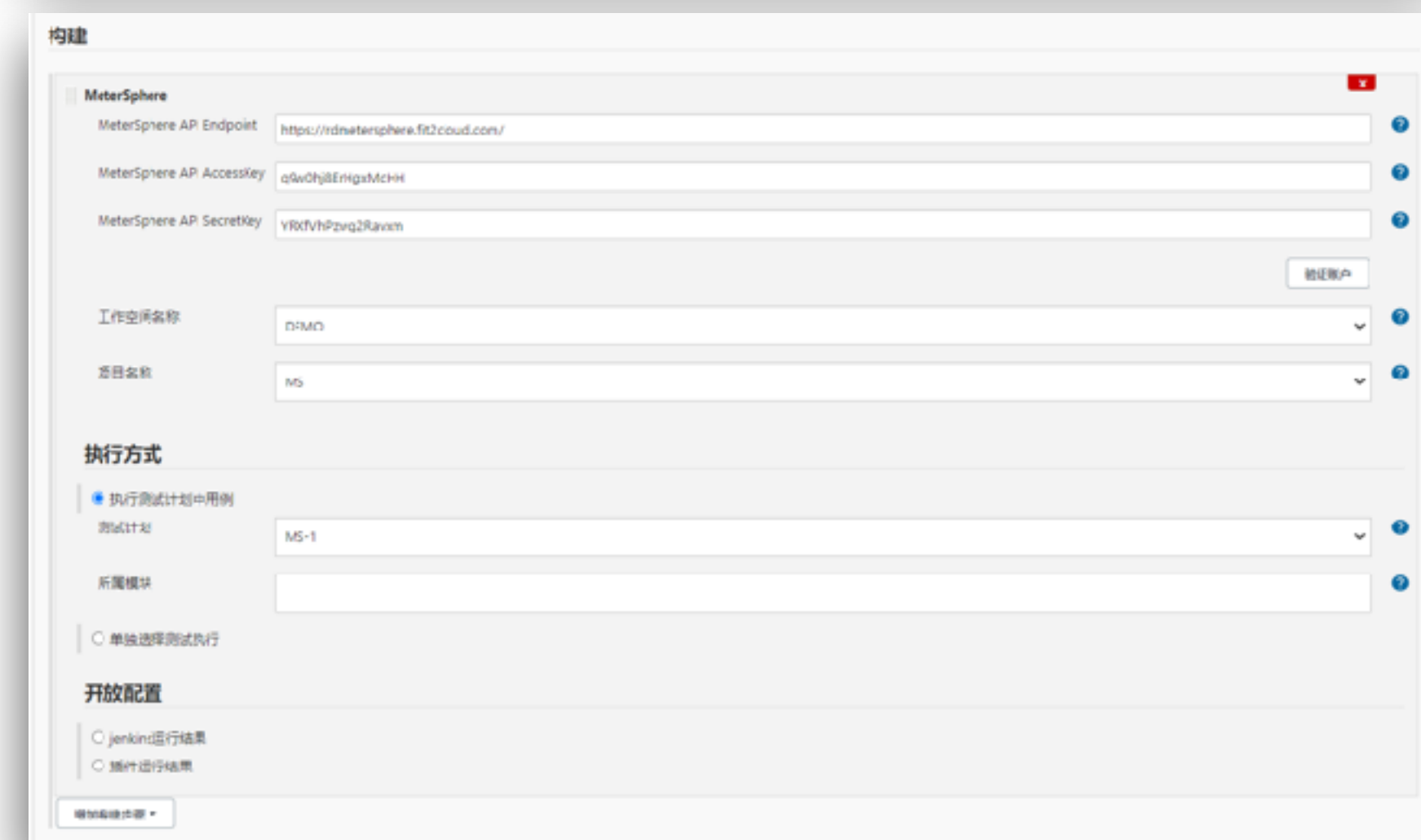
按测试计划触发平台中的测试执行



指定具体测试触发执行



同步测试执行结果返回至 Jenkins



MeterSphere 任务消息与服务集成机制

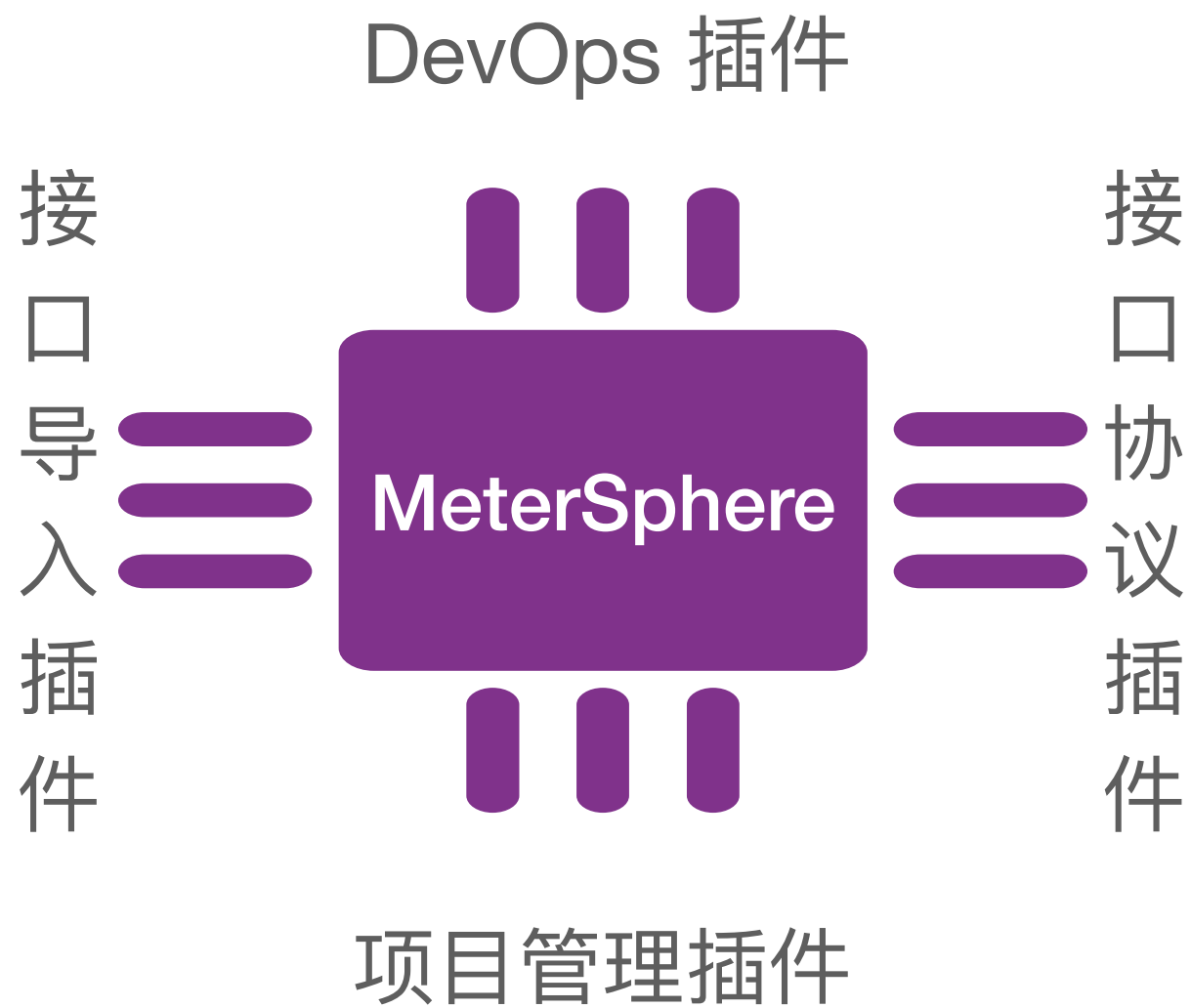


支持测试任务邮件、钉钉、企业微信通知



支持缺陷自动创建至 JIRA \ TAPD \ 禅道

丰富的插件集成机制



序号	插件名	插件类型	应用范围	描述
1	metersphere-jenkins-plugin	DevOps	系统全局	Jenkins 插件，在 Jenkins 中安装该插件后可将 Jenkins 任务中添加 MeterSphere 构建环节可选择指定项目下的测试计划/接口/ UI /性能测试进行触发执行。
2	metersphere-chrome-extensions	API 导入	接口测试	浏览器插件，该插件可将用户在浏览器操作时的 HTTP 请求记录下来并生成 JMX 文件（JMeter 脚本文件），用于在 MeterSphere 中进行接口测试或性能测试。
3	metersphere-idea-plugin	API 导入	接口测试	IDEA 插件，在 IDEA 中安装该插件后，可以将 HTTP 接口直接同步到 MeterSphere。
4	metersphere-plugin-DummySampler	请求	接口测试	DummySampler 插件，可以实现按照用户指定的请求内容及响应结果进行返回。
5	metersphere-jira-plugin	项目管理	测试跟踪	JIRA 项目管理平台对接插件，可以实现测试用例与 JIRA 需求关联，同时可以实现 JIRA 缺陷的双向同步。
6	metersphere-zentao-plugin	项目管理	测试跟踪	禅道项目管理平台对接插件，可以实现测试用例与禅道需求关联，同时可以实现禅道缺陷的双向同步。
7	metersphere-lark-plugin	项目管理	测试跟踪	飞书项目管理平台对接插件，可以实现测试用例与飞书需求关联，同时可以实现飞书缺陷的双向同步。
8	metersphere-pingcode-plugin	项目管理	测试跟踪	PingCode 项目管理平台对接插件，可以实现测试用例与 PingCode 需求关联，同时可以实现 PingCode 缺陷的双向同步。
9	metersphere-plugin-websocket	协议	接口测试	WebSocket 协议插件，可以实现 WebSocket 协议的接口测试。
10	metersphere-plugin-mqtt	协议	接口测试	MQTT 协议插件，可以实现 MQTT 协议的接口测试。
11	metersphere-plugin-thrift	协议	接口测试	Thrift 协议插件，可以实现 Thrift 协议的接口测试。
12	metersphere-plugin-ftp	协议	接口测试	FTP 协议插件，可以实现 FTP 协议的接口测试。
13	metersphere-plugin-grpc	协议	接口测试	GRPC 协议插件，可以实现 GRPC 协议的接口测试。

1

持续测试的兴起与现状

2

MeterSphere 加速企业持续测试落地

3

MeterSphere 企业版及专业服务

MeterSphere 企业版及专业服务

企业版	收费标准	按人年收费；
	服务内容	X-Pack 增强包（按月持续增加） + 企业级支持服务 + 5 人天的专业服务；
测试专家服务	收费标准	按人天收费；
	服务内容	性能测试的压测需求分析、压测流量模型设计、分布式压测环境搭建、压测执行、压测结果分析和优化等服务；
测试专业服务	收费标准	按人天收费；
	服务内容	安装部署、使用培训、数据迁移、故障排除、二次开发（仅接受与产品保持同步）等专业服务。

MeterSphere X-Pack 增强包

系统管理



自定义 Logo
与主题配色

功能用例



版本管理和对比



公共用例库

接口测试



协议插件（MQTT、
Thrift、WebSocket）



误报库



接口变化自动
更新 CASE

测试执行

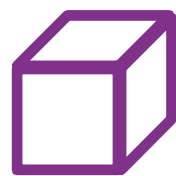


自定义消息
通知模版

性能测试



大规模
性能测试支持



Kubernetes
资源池支持



抽样分析

系统集成



单点登录
系统对接



缺陷双向同步
（JIRA 等）

UI 测试



UI 测试

MeterSphere 助力企业快速构建并运营自己的持续测试平台



了解更多

访问官网



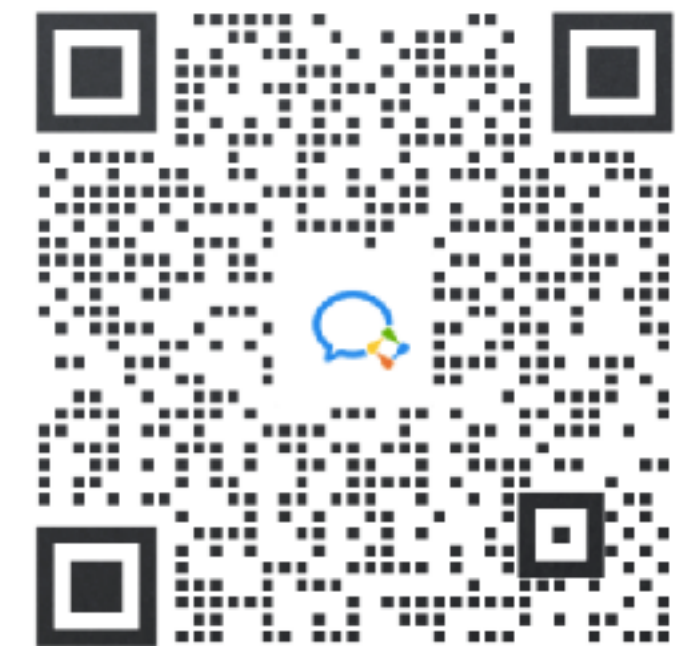
<https://metersphere.io>

一键安装



```
curl -sSL https://github.com/metersphere/metersphere/  
releases/latest/download/quick_start.sh | sh
```

进群交流



微信 / 企业微信扫码进群

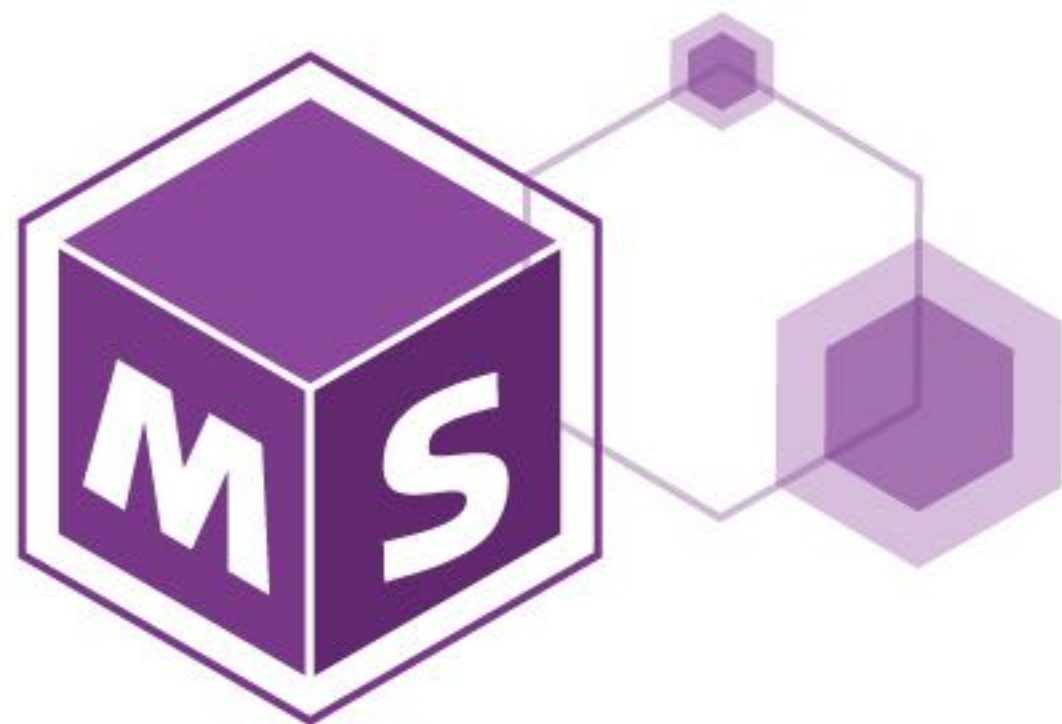
THANK YOU

www.fit2cloud.com

 400-052-0755

北京 · 上海 · 深圳 · 广州 · 南京 · 杭州
苏州 · 武汉 · 成都 · 西安 · 长沙 · 济南
郑州 · 厦门 · 合肥 · 青岛 · 重庆

www.metersphere.io



因为热爱 快意成长

www.metersphere.io