

基于 HttpRunner 4.0 构建专业级压测工具

抖音集团 - 李隆 (debugtalk)



主办方: TesterHome

关于我

李隆，网名 debugtalk

就职经历：工信部电子五所 => 阿里 UC => 大疆创新

=>字节（飞书 -> 抖音）

个人博客：<https://debugtalk.com>

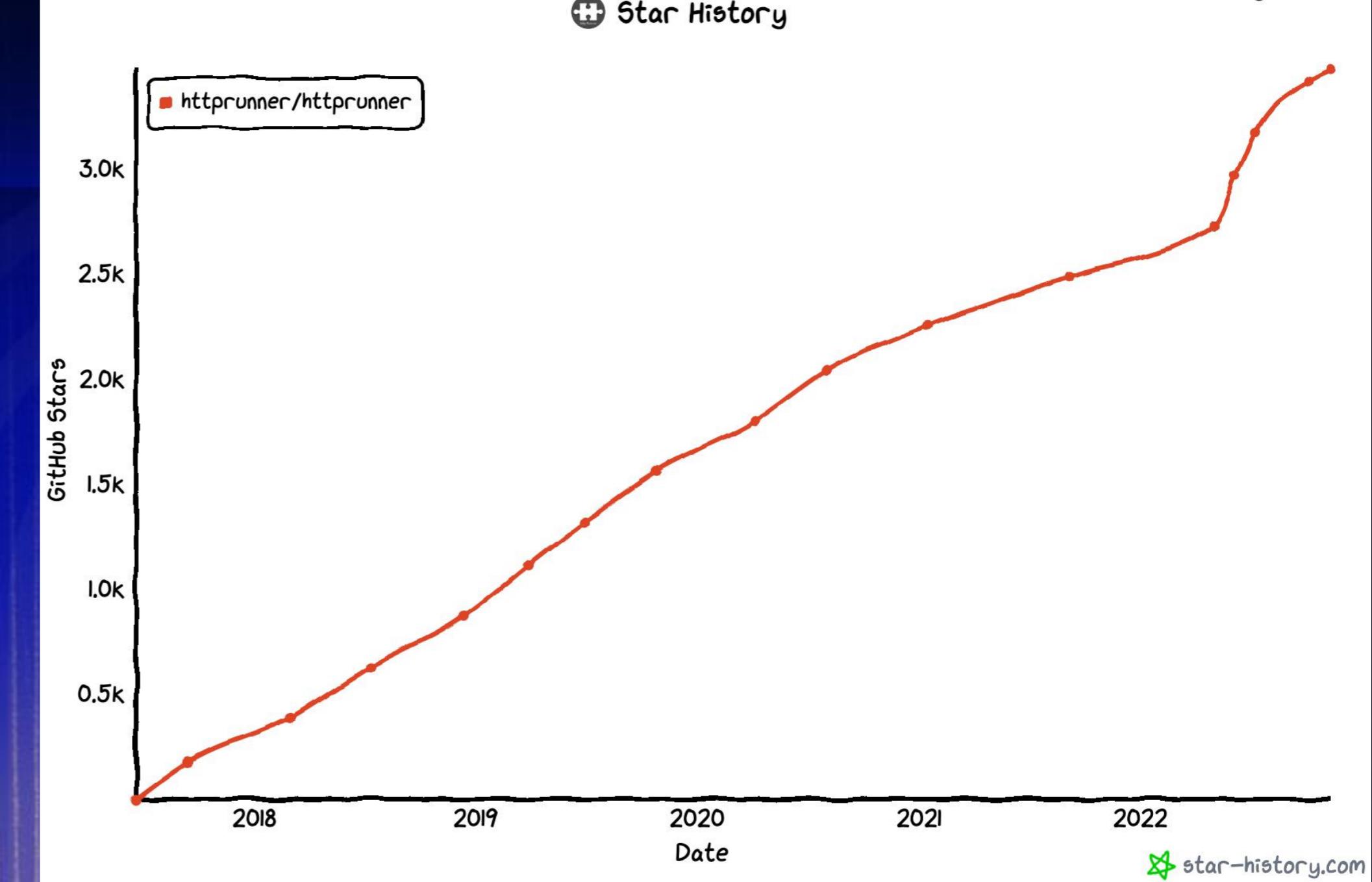
个人公众号：DebugTalk



关于 HttpRunner

2017.06 创建至今：

- 发布了 4 个大版本（v1~v4）和 100+ 个小版本
- 共有 27 位开发者参与了代码贡献，提交了 3000+ 次 commit，代码变更量 30w+
- GitHub star 数：3500+
- 官网：<https://httprunner.com>
- 微信公众号：HttpRunner
- MTSC 2018: 《大疆互联网的一站式自动化测试解决方案（基于HttpRunner）》
- MTSC 2019: 《HttpRunner 2.0 技术架构与接口测试应用》



议题大纲

- 1、HttpRunner 演进史
- 2、HttpRunner 4.0 核心特性解析
- 3、构建专业级压测工具 QuickRunner
- 4、HttpRunner 的开源展望
- 5、Q & A



1、HttpRunner 演进史



v1/v2/v3 的进化历程

v1：雏形阶段，python unittest

v2：步入正轨，版本号规范 + 用例格式标准化 + 模块化拆分

v3：能力增强，python pytest + allure + 链式语法提示

性能部分：Locust

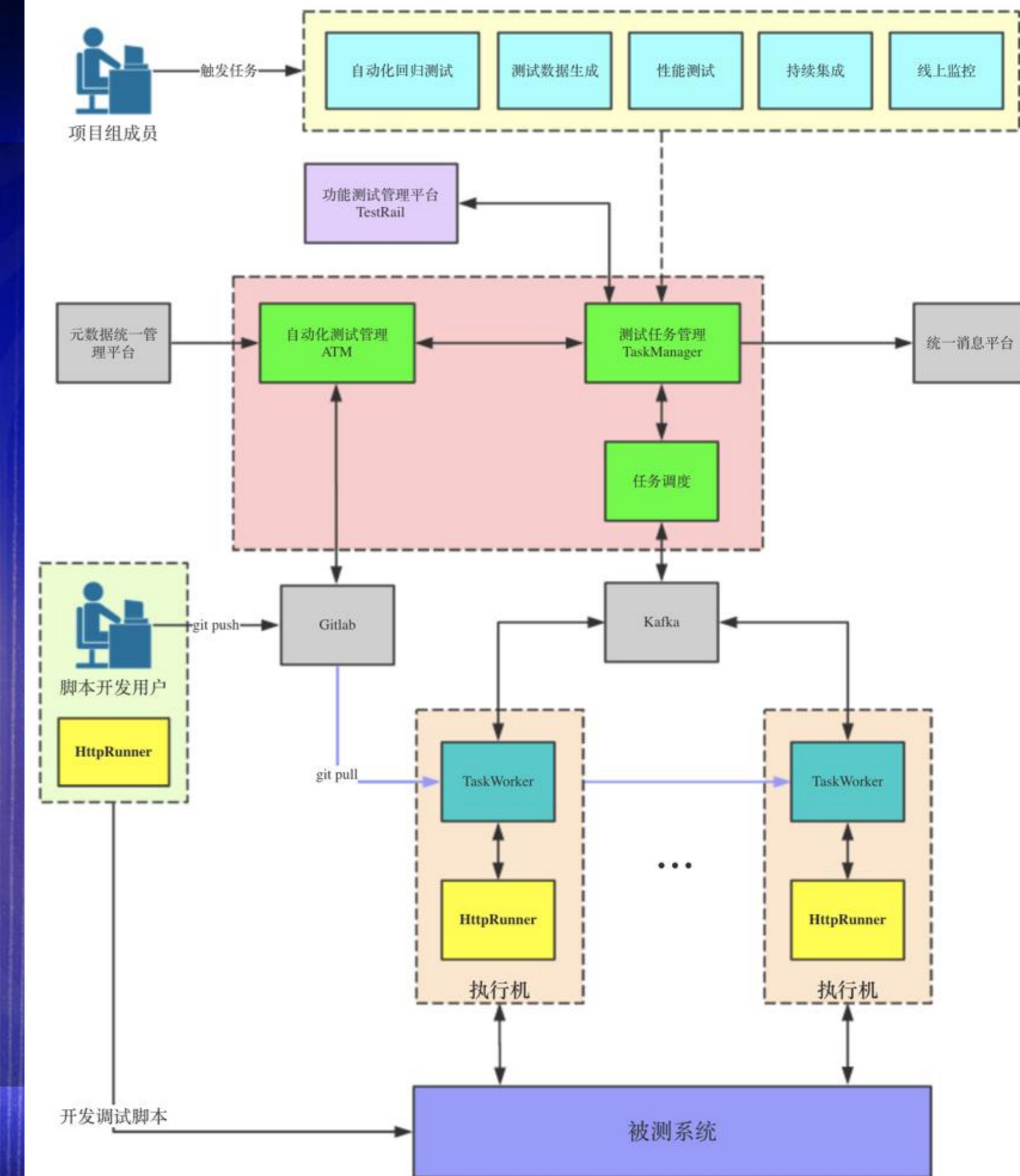
版本	v1	v2	v3
发布时间	2018.03.07	2019.01.01	2020.03.10
开发语言	Python	Python	Python
版本号规范 (semver)	✗	✓	✓
网络协议	HTTP(S)/1.1	HTTP(S)/1.1	HTTP(S)/1.1
脚本转换工具	HAR	HAR	HAR
工程脚手架	✗	✓	✓
测试用例 (集) 格式	v1	v2	v2
测试用例分层机制	v1	v2	v2
脚本格式类型	YAML/JSON	YAML/JSON	YAML/JSON/ pytest
脚本格式校验	✗	jsonschema	✗
脚本编写语法提示	✗	✗	pytest 链式调用
脚本执行引擎	Python unittest	Python unittest	Python pytest
插件化语言 (debugtalk.xx)	Python	Python	Python
参数提取机制	regex + 点分隔符	jmespath + regex + 点分隔符	jmespath
skip 机制	✓	✗	✗
接口测试报告	html 自研 (jinja2)	html 自研 (jinja2)	pytest-html/allure
性能测试引擎	Python Locust	Python Locust	Python Locust
运行环境依赖	Python 2.7/3.3+	Python 2.7/3.5+	Python 3.7+ pytest
网络性能采集	✗	✗	✗
安装部署方式	pip	pip	pip

用户案例 - 从 0 开始建设



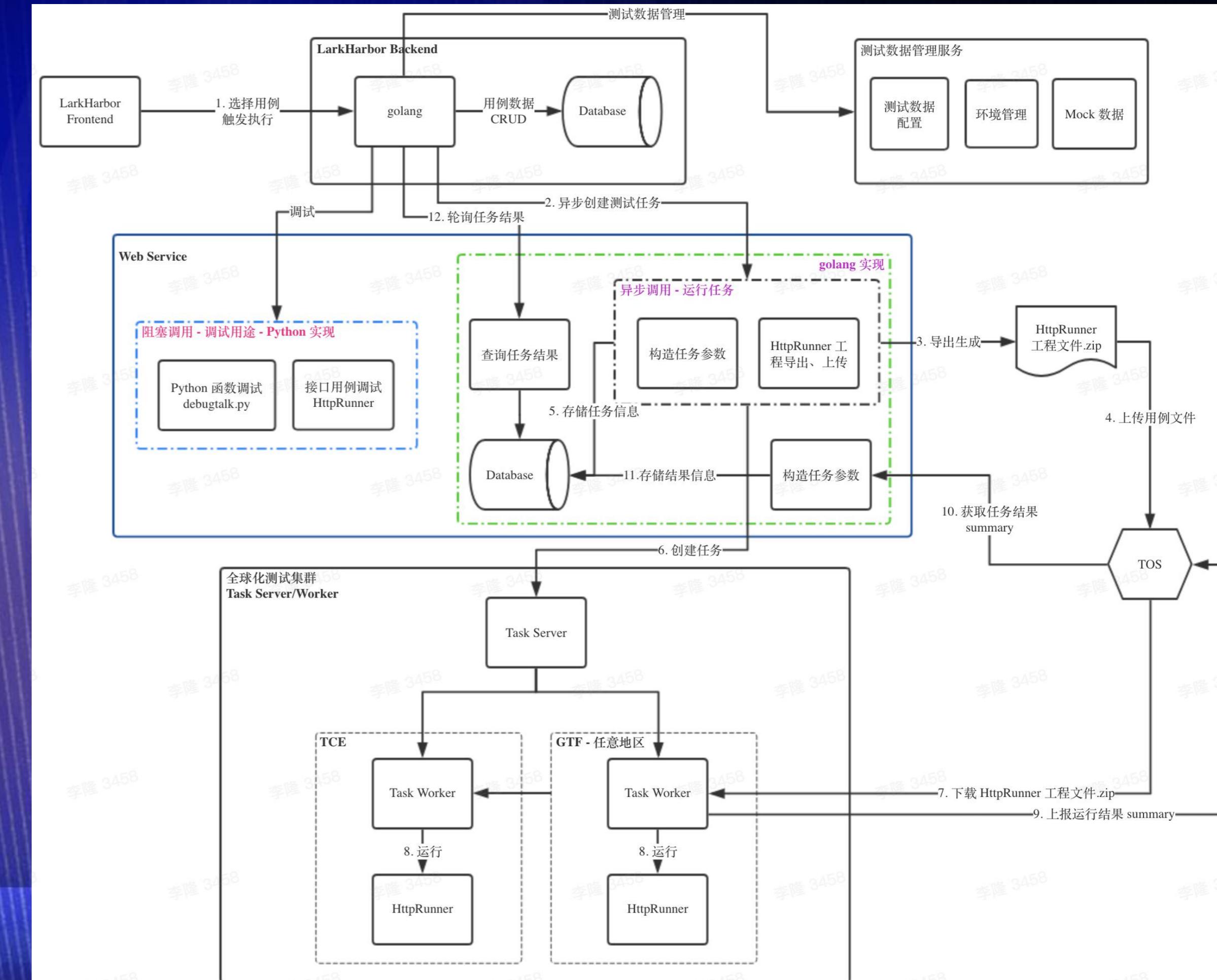
大疆基于 HttpRunner 从 0 开始建设了整套自动化测试体系，覆盖功能测试、接口自动化测试、性能测试、测试数据服务、持续集成线上监控等能力。

<https://httprunner.com/docs/cases/dji-ibg/>



用户案例 - 基于已有平台进行改造

飞书业务线在已有的接口测试平台中使用 HttpRunner 替换了原有的用例执行引擎，大幅提升测试结果稳定性的同时，实现了「KA 私有化部署接口功能验收」和「全球网络环境接口场景化监测」的能力。



用户案例 - 建设自动化测试平台



通用技术环球医疗
GENERTEC UNIVERSAL MEDICAL

更多案例介绍：

<https://httprunner.com/docs/cases/>



用户调研问卷

2022.02 收集了 252 份用户调研问卷：

- 整体评分 4.3 (满分 5 分)
- 最喜欢的特性：简单易用、功能强大、设计理念
- 最期待的功能：UI 可视化、性能测试增强、支持多网络协议

维度	关键词	期望新增数	提及次数	汇总
易用性	Web 平台 / UI 客户端	21	26
	IDE 插件支持	.	1	
	降低用例维护成本	.	1	
	支持多编程语言，更好的集成	.	1	
	支持对单个 API 进行维护和测试	..	2	
性能测试	性能测试更便捷；更完善；更强大	11	21
	性能测试结果准确	.	1	
	兼容、集成 JMeter	..	2	
	全链路压测	.	1	
	性能优化	.	1	
	分布式压测	..	2	
网络协议	全链路监控	...	3	21
	支持更多协议	8	
	graphQL	.	1	
	TCP	.	1	
	dubbo	..	2	
	rpc	8	
项目支持	thrift	.	1	12
	文档 & 案例 & demo	11	
	永久开源	.	1	
	支持循环次数	.	1	
	用例并行执行	4	
	分布式执行	..	2	
执行机制	失败重试	...	3	10
	更多的脚本格式转换类型	.	1	
	har2case 生成 go 代码	..	2	
	yapi 转换生成测试用例	.	1	
	postman 转换生成测试用例	.	1	
	swagger 转换生成测试用例	4	
脚本格式转换	报告美化；增加统计图表	...	3	9
	报告：展示变量来源	.	1	
	测试报告支持 2.x 那样	.	1	
	测试报告支持 Excel 格式	.	1	
	支持 step 跳过：按照条件跳过 step	..	2	
	支持if条件判断执行用例	.	1	
流程控制	自定义脚本执行顺序	..	2	5
	自定义断言增强	..	2	
	调用 call testcase 时支持断言	...	3	
	测试结果归因分析	.	1	
	链路追踪	.	1	
	脚本执行异常判断；问题定位	.	1	
问题定位	更全面的结果数据收集	.	1	4
	支持数据库连接；处理封装	...	3	
	支持数据库存储过程测试	.	1	
	DEM 相关能力	.	1	
	线上监控	.	1	
	定时任务	.	1	
应用场景拓展	持续集成	.	1	4
	集成抓包能力；流量录制回放	..	2	
	支持 UI 自动化；合并 UI 自动化	..	2	
	hook 增强；hook 获取返回值	..	2	
	支持异步接口返回，循环查询获取响应结果	.	1	
	自定义函数调试功能	.	1	
其它	CSV 数据中支持调用变量或函数	.	1	2
	step 支持调用命令行	.	1	
	支持插件开发	.	1	
	运行环境容器化	.	1	
	支持更多 pytest 特性；pytest的mark tag多级支持	..	2	
	HttpRunner	.	1	

2、HttpRunner 4.0 核心特性解析



r0 的核心目标



The slide features a central promotional graphic for HttpRunner. At the top left is the HttpRunner logo, which consists of a circular icon containing a puzzle piece-like symbol with the text "HttpRunner" below it. To the right of the logo, the word "HttpRunner" is written in a large, white, sans-serif font. Below this, the text "新一代开源 API 测试解决方案" is displayed in a slightly smaller white font. At the bottom of the graphic, there are three white, rounded rectangular buttons with blue text: "简单易用" (Simple and Easy to Use), "功能强大" (Powerful Features), and "数据精准" (Accurate Data).



v4 核心特性

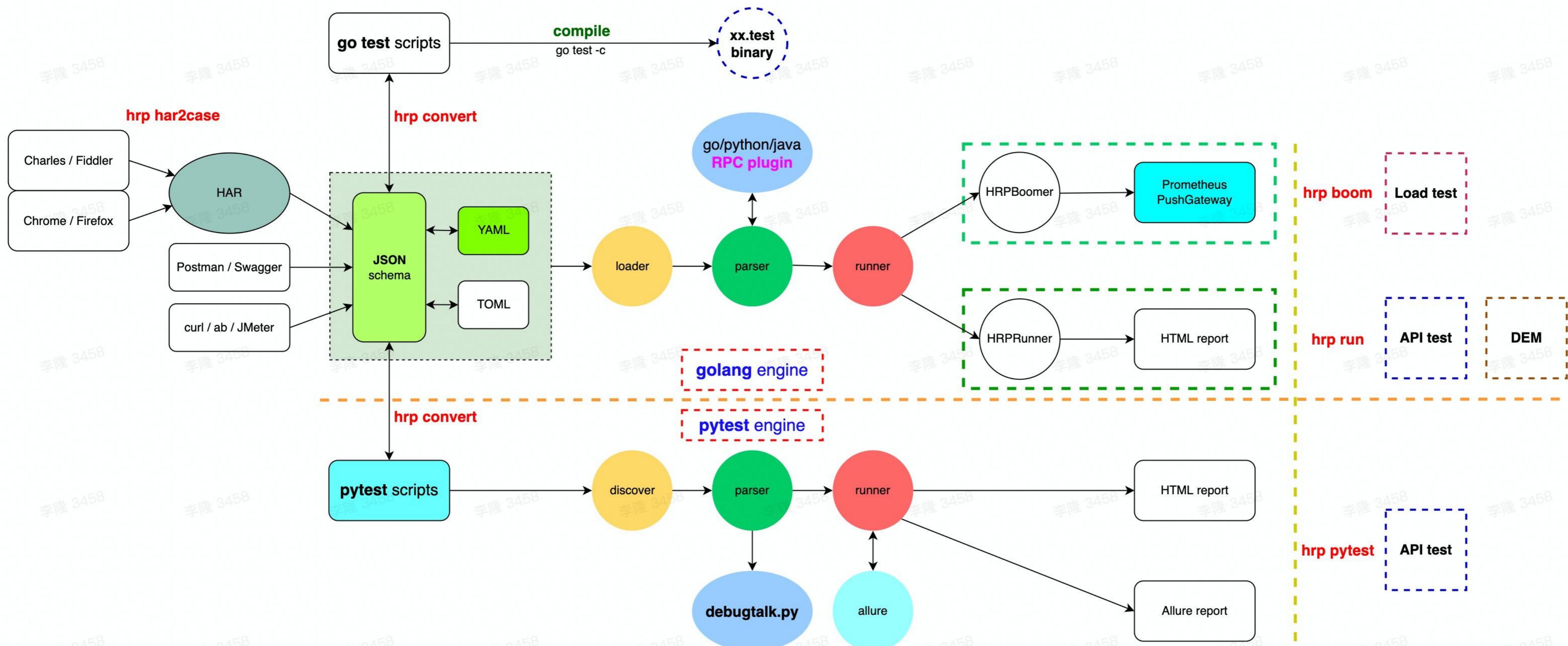
v4 = v2 + v3 + hrp + ...

核心特性：

- 支持 Go/Python 双执行引擎
- 支持多网络协议
- 支持多编程语言
- 支持多测试用途
- 支持多平台环境
- 打通主流 API 研发测试工具
- 链式语法提示

版本	v1	v2	v3	HttpRunner+	v4
发布时间	2018.03.07	2019.01.01	2020.03.10	2021.11.18	2022.05.01
开发语言	Python	Python	Python	Golang	Golang + Python
版本号规范 (semver)	✗	✓	✓	✓	✓
网络协议	HTTP(S)/1.1	HTTP(S)/1.1	HTTP(S)/1.1	HTTP(S)/1.1	多协议 HTTP(S)/HTTP2/WebSocket/TCP/RPC
脚本转换工具	HAR	HAR	HAR	HAR	HAR/Postman/Swagger/Curl
工程脚手架	✗	✓	✓	✓	✓
测试用例 (集) 格式	v1	v2	v2	v2	v2
测试用例分层机制	v1	v2	v2	v2	v2
脚本格式类型	YAML/JSON	YAML/JSON	YAML/JSON/ pytest	YAML/JSON	YAML/JSON/ pytest/gotest
脚本格式校验	✗	jsonschema	✗	✗	TODO
脚本编写语法提示	✗	✗	pytest 链式调用	gotest 链式调用	gotest 链式调用 + pytest 链式调用
脚本执行引擎	Python unittest	Python unittest	Python pytest	Go 自研	Go 自研 + Python pytest
插件化语言 (debugtalk.xx)	Python	Python	Python	多语言 (Go/Python)	多语言 (Go/Python/Java/etc.)
参数提取机制	regex + 点分隔符	jmespath + regex + 点分隔符	jmespath	jmespath + regex	jmespath + regex
skip 机制	✓	✗	✗	✗	TODO
接口测试报告	html 自研 (jinja2)	html 自研 (jinja2)	pytest-html/allure	html 自研 (Go template)	html 自研 (Go template) + pytest-html/allure
性能测试引擎	Python Locust	Python Locust	Python Locust	Go Boomer	Go Boomer
运行环境依赖	Python 2.7/3.3+	Python 2.7/3.5+	Python 3.7+ pytest	无需依赖	Go 引擎无需依赖 pytest 引擎依赖 Python 3.7+
网络性能采集	✗	✗	✗	✗	✓
安装部署方式	pip	pip	pip	curl/wget	curl/wget

v4 核心特性 - 支持 Go & Python 双执行引擎



v4 核心特性 - 支持多网络协议

横向拓展，格式统一

```
{
  "name": "HTTP/1.1 post",
  "request": {
    "method": "POST",
    "url": "/post",
    "headers": {
      "User-Agent": "HttpRunnerPlus"
    },
    "body": {
      "foo1": "foo1",
      "foo2": "foo2"
    }
  },
  "validate": [
    {
      ...
    },
    {
      "check": "proto",
      "assert": "equals",
      "expect": "HTTP/1.1",
      "msg": "check protocol type"
    },
    {
      ...
    }
  ]
}
```

DPPL+^{*}

```
{
  "name": "HTTP/2 post",
  "request": {
    "method": "POST",
    "url": "/post",
    "http2": true,
    "headers": {
      "User-Agent": "HttpRunnerPlus"
    },
    "body": {
      "foo1": "foo1",
      "foo2": "foo2"
    }
  },
  "validate": [
    {
      ...
    },
    {
      "check": "proto",
      "assert": "equals",
      "expect": "HTTP/2.0",
      "msg": "check protocol type"
    },
    {
      ...
    }
  ]
}
```

DPPL+

```
{
  "name": "read sponsor info",
  "websocket": {
    "type": "r",
    "url": "/",
    "timeout": 5000
  },
  "validate": [
    {
      "check": "body",
      "assert": "contains",
      "expect": "Lob.com",
      "msg": "check sponsor message"
    }
  ],
  {
    "name": "write json",
    "websocket": {
      "type": "w",
      "url": "/",
      "text": {
        "foo1": "${gen_random_string($n)}",
        "foo2": "${max($a, $b)}"
      }
    }
  }
}
```

S a^Ok_gap

v4 核心特性 - 支持多测试用途

接口自动化

接口性能测试

网络链路性能

QE自动化

网络性能耗时拆解



v4 核心特性 - 支持多测试用途

```
// WithHTTPStat is a wrapper of httptrace.WithClientTrace.
// It records the time of each httptrace hooks.
func WithHTTPStat(req *http.Request, s *Stat) context.Context {
    s.mutex = new(sync.RWMutex)
    s.schema = req.URL.Scheme
    return httptrace.WithClientTrace(req.Context(), &httptrace.ClientTrace{
        DNSStart: func(i httptrace.DNSStartInfo) {
            s.dnsStart = time.Now()
        },
        DNSDone: func(i httptrace.DNSDoneInfo) {
            s.dnsDone = time.Now()

            s.DNSLookup = s.dnsDone.Sub(s.dnsStart)
            s.NameLookup = s.DNSLookup
        },
        ConnectStart: func(network, addr string) {
            s.network = network
            s.addr = addr

            s.tcpStart = time.Now()

            // When connecting to IP (When no DNS lookup)
            if s.dnsStart.IsZero() {
                s.dnsStart = s.tcpStart
                s.dnsDone = s.tcpStart
            }
        },
        ConnectDone: func(network, addr string, err error) {
            s.tcpDone = time.Now()
            s.TCPConnection = s.tcpDone.Sub(s.tcpStart)
            s.Connect = s.tcpDone.Sub(s.dnsStart)
        },
        TLSHandshakeStart: func() {
            s.isTLS = true
            s.tlsStart = time.Now()
        },
    })
}
```

----- request -----
 GET /get?foo1=bar11&foo2=bar21&sum_v=3 HTTP/1.1
 Host: postman-echo.com
 User-Agent: funplugin/0.4.3

===== response =====
 Connected via TLSv1.2
 HTTP/1.1 200 OK
 Content-Length: 327
 Connection: keep-alive
 Content-Type: application/json; charset=utf-8
 Date: Wed, 18 May 2022 09:00:56 GMT
 Etag: W/"147-QapGarSvcBnsNpEU+Q/blH+/BpE"
 Set-Cookie: sails.sid=s%3AQfwTPYP_fwPy8vM4EmirEL1KyztRoqR.%2FVN%2BQNFgqvhl53fv%2Bg%2BSV6El9G
 Vary: Accept-Encoding

{"args":{"foo1":"bar11","foo2":"bar21","sum_v":"3"}, "headers": {"x-forwarded-proto": "https", "x-forwarded-port": "443", "x-forwarded-for": "127.0.0.1:56f10d77ebf5", "user-agent": "funplugin/0.4.3", "accept-encoding": "gzip"}, "url": "https://postman-echo.com/get?foo1=bar11&foo2=bar21&sum_v=3"}

 Connected to tcp: 3.222.255.96:443

DNS Lookup	TCP Connection	TLS Handshake	Server Processing	Content Transfer
[108ms]	[281ms]	[740ms]	[369ms]	[0ms]
namelookup: 108ms				
	connect: 390ms			
		pretransfer: 1130ms		
			starttransfer: 1500ms	
				total: 1500ms

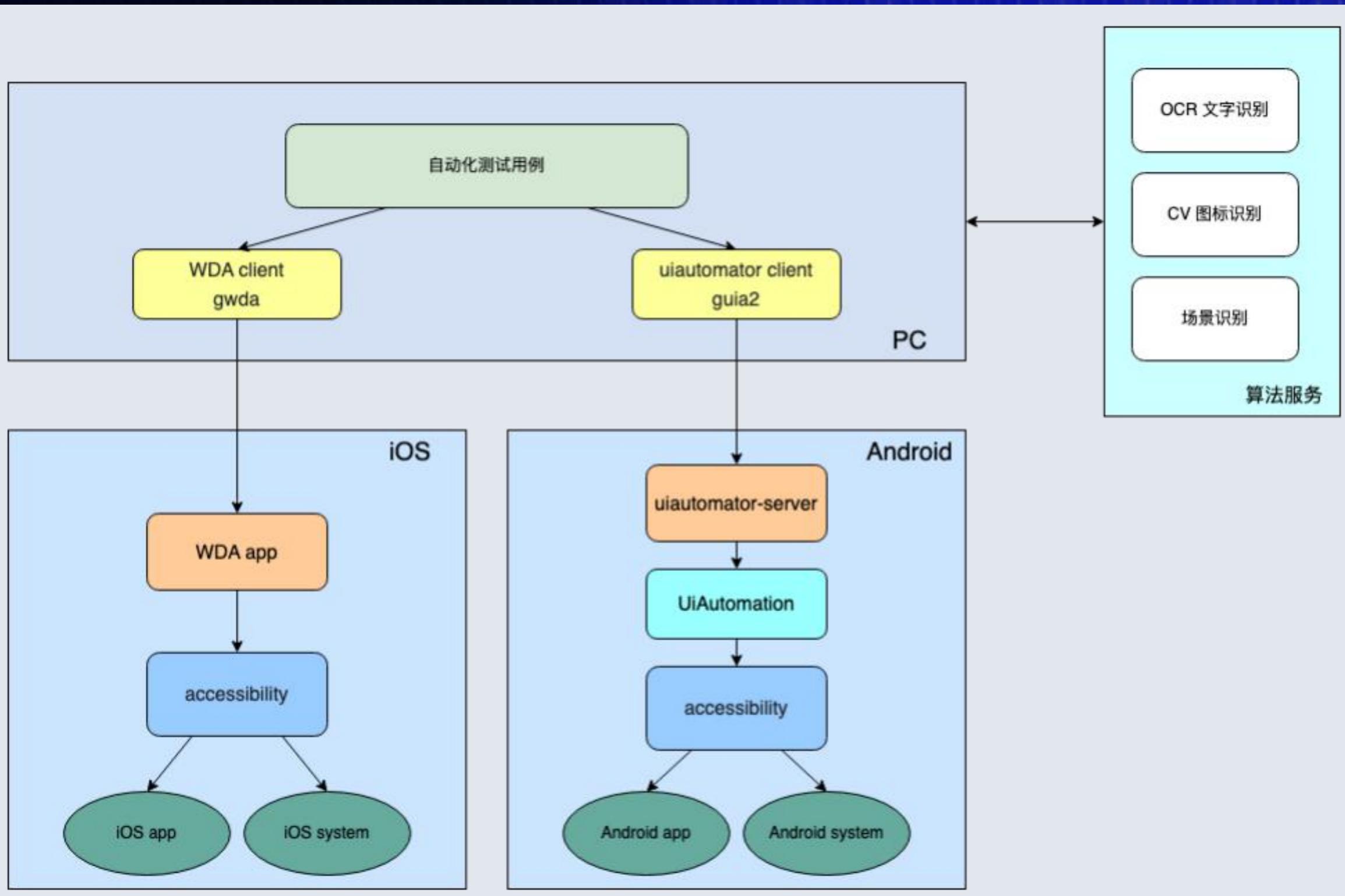
网络性能耗时拆解

v4 核心特性 - 支持多测试用途

=j` rkè
QE自动化

EKO
QE自动化

S a^
QE自动化



```

func TestDemo(t *testing.T) {
    testCase := &hrp.TestCase{
        Config: hrp.NewConfig("demo testcase").
            WithVariables(map[string]interface{}{
                "app_name": "抖音",
            }).
        SetIOS(hrp.WithLogOn(true), hrp.WithWDAPort(8700), hrp.WithWDAMjpegPort(8800)),
        TestSteps: []hrp.IStep{
            hrp.NewStep("启动抖音").
                IOS().
                Home().
                AppTerminate("com.ss.iphone.ugc.Aweme"). // 关闭已运行的抖音
                SwipeToTapApp("$app_name",
                    hrp.WithMaxRetryTimes(5),
                    hrp.WithIdentifier("启动抖音")).Sleep(5).
                Validate().
                AssertOCRExists("推荐", "抖音启动失败, 「推荐」不存在"),
            hrp.NewStep("处理青少年弹窗").
                IOS().
                TapByOCR("我知道了", hrp.WithIgnoreNotFoundError(true)),
            hrp.NewStep("进入购物页").
                IOS().TapByOCR("购物", hrp.WithIdentifier("点击购物")).Sleep(5),
            hrp.NewStep("进入推荐页").
                IOS().TapByOCR("推荐", hrp.WithIdentifier("点击推荐")).Sleep(5),
            hrp.NewStep("向上滑动 2 次").
                IOS().
                SwipeUp(hrpWithIdentifier("第 1 次上划")).Sleep(2).
                SwipeUp(hrpWithIdentifier("第 2 次上划")).Sleep(2).
                SwipeUp(hrpWithIdentifier("第 3 次上划")).Sleep(2),
            TapXY(0.9, 0.1, hrpWithIdentifier("点击进入搜索框")).Sleep(2),
            Input("httprunner", hrpWithIdentifier("输入搜索关键词")),
        },
    }
}
    
```

核心数据结构

```

type TestCase struct {
    Config     *TConfig
    TestSteps []IStep
}

You, 4 months ago | 1 author (You)
type IStep interface {
    Name() string
    Type() StepType
    Struct() *TStep
    Run(*SessionRunner) (*StepResult, error)
}

```

```

func (r *SessionRunner) Start() error {
    ...
    // run step in sequential order
    for _, step := range r.testCase.TestSteps {
        _, err := step.Run(r)
        if err != nil && r.hrpRunner.failfast {
            return errors.Wrap(err, "abort running due to failfast setting")
        }
    }
    ...
}

```

- TestCase: 有序步骤的集合
- 测试步骤 IStep: 采用了 go interface 的设计理念，支持进行任意拓展；步骤内容统一在 Run 方法中进行实现
- 只需遵循 IStep 的接口定义，即可实现各种类型的测试步骤类型



网络协议拓展

测试类型拓展

v4 核心特性 - 支持多编程语言

Golang

```

package main

import (
    "fmt"
    "github.com/httprunner/funplugin/fungo"
)

func SumTwoInt(a, b int) int {
    return a + b
}

func SumInts(args ...int) int {
    var sum int
    for _, arg := range args {
        sum += arg
    }
    return sum
}

func Sum(args ...interface{}) (interface{}, error) {
    var sum float64
    for _, arg := range args {
        switch v := arg.(type) {
        case int:
            sum += float64(v)
        case float64:
            sum += v
        default:
            return nil, fmt.Errorf("unexpected type: %T", arg)
        }
    }
    return sum, nil
}

func main() {
    fungo.Register("sum_ints", SumInts)
    fungo.Register("sum_two_int", SumTwoInt)
    fungo.Register("sum", Sum)
    fungo.Serve()
}

```

Python

```

import logging
from typing import List

import funppy

def sum_two_int(a: int, b: int) -> int:
    return a + b

def sum_ints(*args: List[int]) -> int:
    result = 0
    for arg in args:
        result += arg
    return result

def Sum(*args):
    result = 0
    for arg in args:
        result += arg
    return result

```

if __name__ == '__main__':
 funppy.register("sum_two_int", sum_two_int)
 funppy.register("sum_ints", sum_ints)
 funppy.register("sum", Sum)
 funppy.serve()

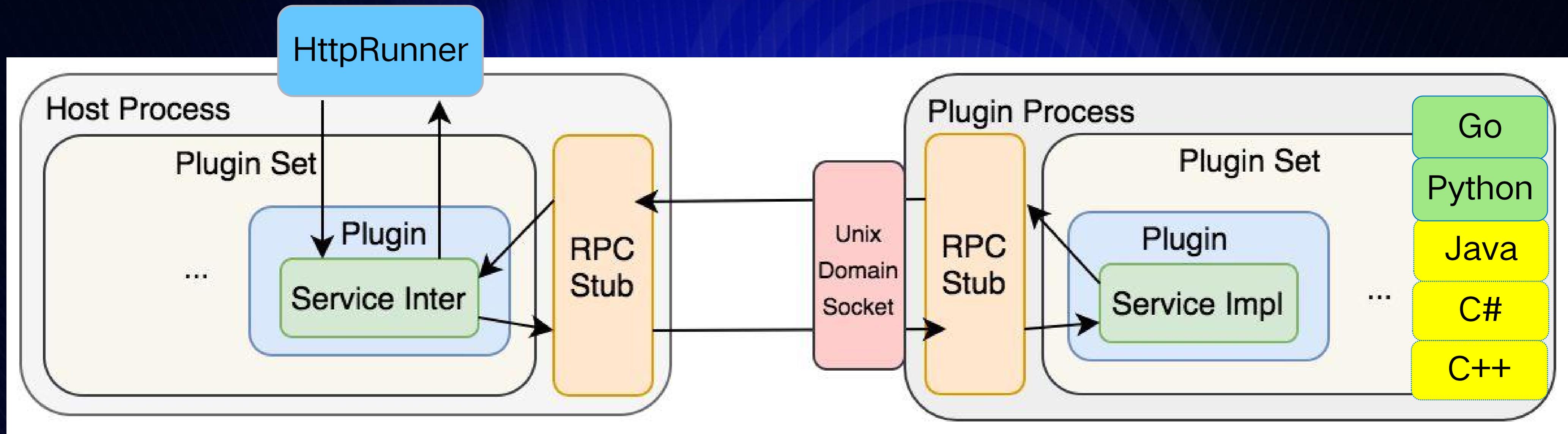
Java

C#

C++

dp o6hcrl _*ek+`
k_o+lj cq] cao+

v4 核心特性 - 支持多编程语言



<https://github.com/hashicorp/go-plugin>
<https://github.com/httprunner/funplugin>

v4 核心特性 - 支持多平台环境

t 42

=NI

信创

S ɸ` ks o

=j` rkè

麒麟

Hɸ qt

树莓派

统信

i] _KO

i] _KO
| -

龙芯、海光、飞腾、鲲鹏

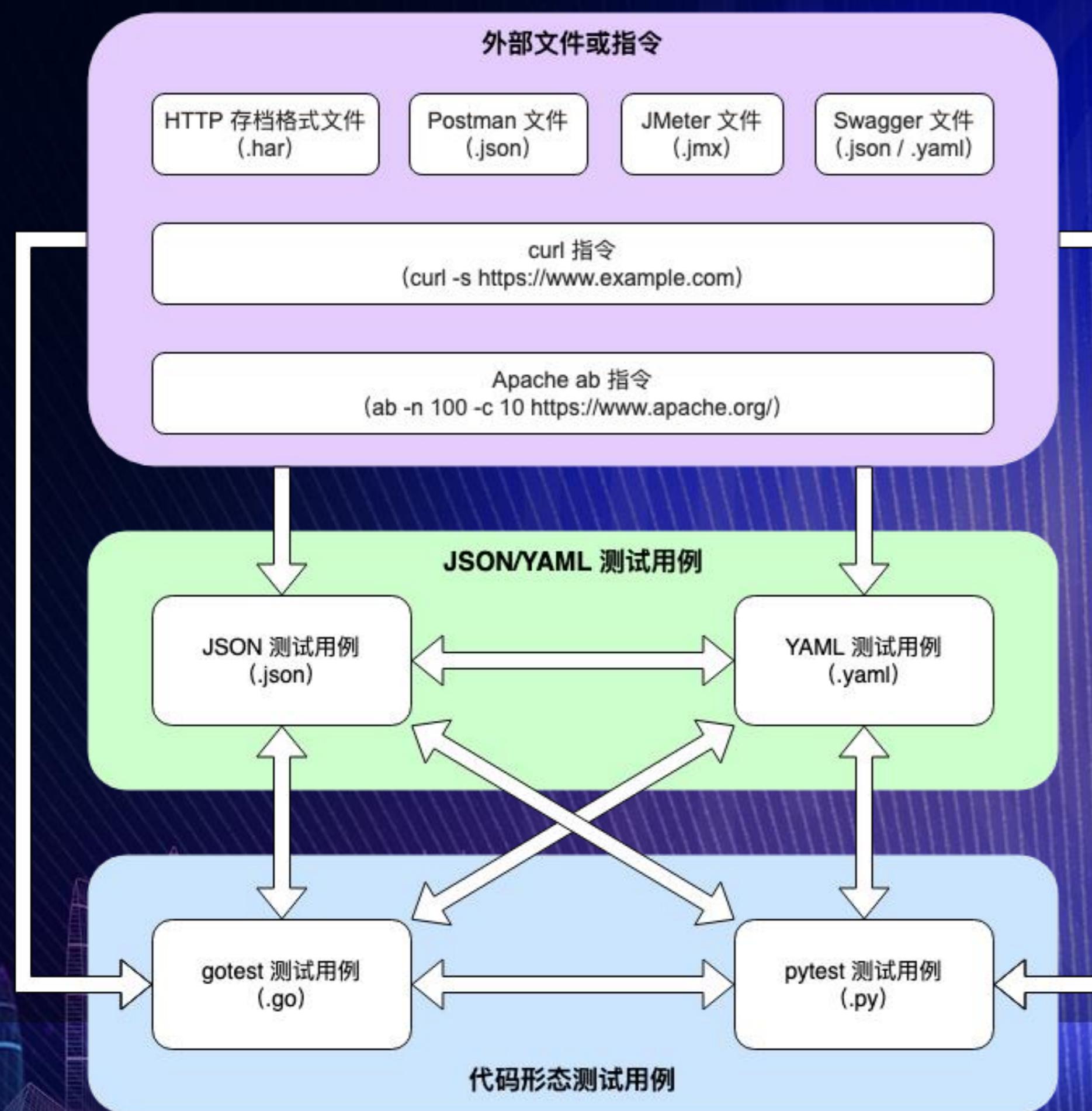
Lupdkj

Ckħj c

v4 核心特性 - 支持多平台环境



v4 核心特性 - 打通主流 API 研发测试工具



from \ to	JSON	YAML	GoTest	PyTest
HAR	✓	✓	✗	✓
Postman	✓	✓	✗	✓
JMeter	✗	✗	✗	✗
Swagger	✗	✗	✗	✗
curl	✗	✗	✗	✗
Apache ab	✗	✗	✗	✗
JSON	✓	✓	✗	✓
YAML	✓	✓	✗	✓
GoTest	✗	✗	✗	✗
PyTest	✗	✗	✗	✗
Home				

v4 核心特性 - 链式语法提示，飞一般的编写体验

基于链式调用的方法提示
也可快速编写测试用例

右侧为链式调用 GIF 图



```
func TestHttpRunnerDemo(t *testing.T) {
    testcase := &hrp.TestCase{
        Config: hrp.NewConfig( name: "HttpRunner Demo").
            SetBaseUrl( baseURL: "http://127.0.0.1:8888"),
        TestSteps: []hrp.IStep{
            hrp.NewStep( name: "get with params").
                GET( url: "/get").
                WithParams(map[string]interface{}{"k1": "v1"}).
                Extract().
                WithJmesPath( jmesPath: "body.args.k1", varName: "varK1").
                Validate().
                AssertEqual( jmesPath: "status_code", expected: 200, msg: "check status code"),
        },
    }
    err := hrp.NewRunner(t).Run(testcase)
    if err != nil {
        t.Fatalf( format: "run testcase error: %v", err)
    }
}
```

3、构建专业级压测工具 QuickRunner



QuickRunner 介绍

QuickRunner 是一款基于开源测试框架 HttpRunner 打造的简单易用、功能强大、数据精准、信创兼容的专业级性能测试工具，支持 HTTP(S)/HTTP2/ WebSocket/TCP/UDP/RPC 等主流网络协议。产品形态为桌面客户端，支持离线安装和分布式部署使用。基于内置的脚本录制转换能力和自定义函数特性，可快速创建各种复杂业务场景的压测脚本。产品通过了主流信创环境兼容性认证和严格的 benchmark 测试，数据结果准确可信，可稳定支撑百万级真实并发压力测试。



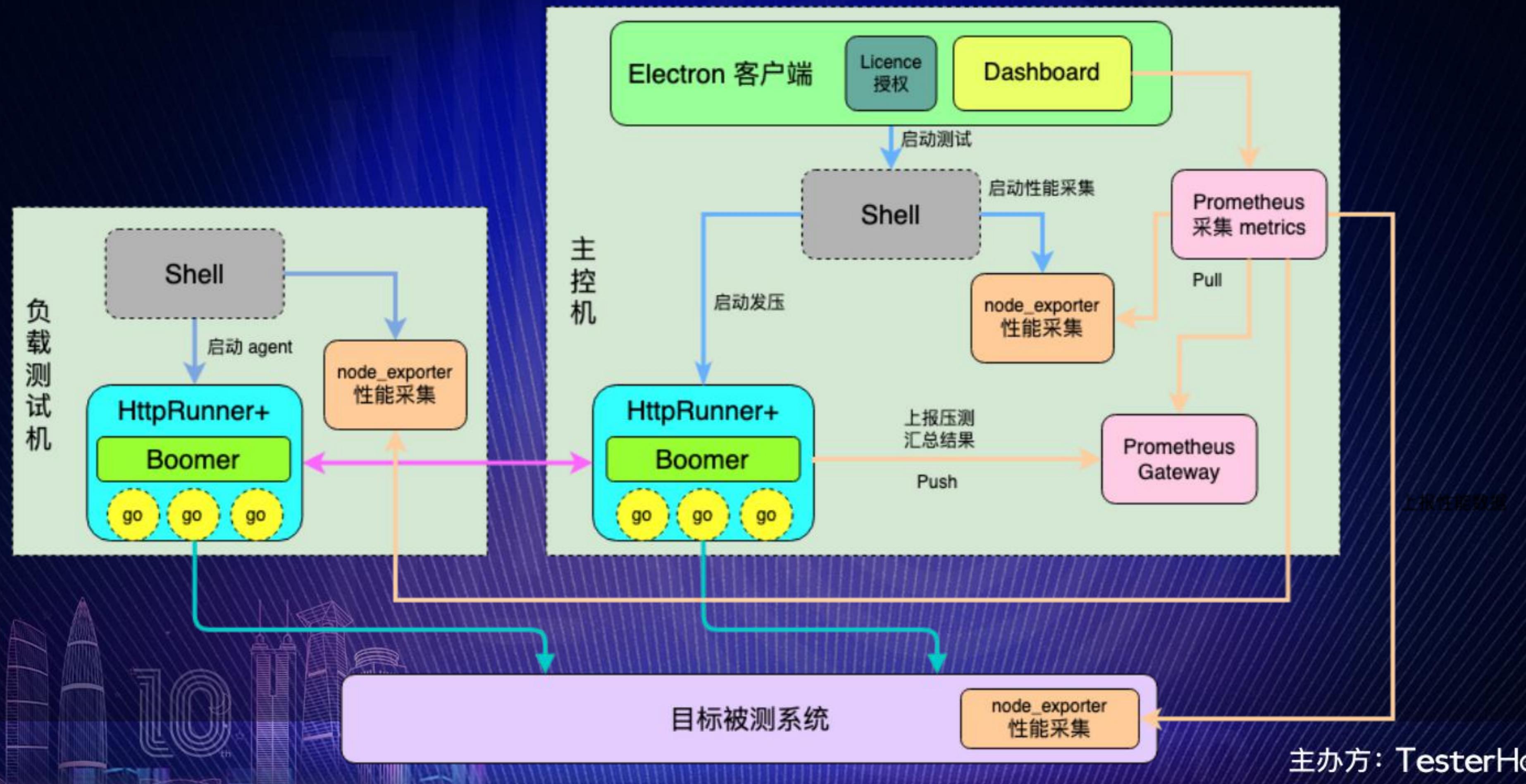
QuickRunner UI 展示

The left panel shows a list of recorded requests (POST, GET, etc.) and a detailed view of a selected POST request to https://mon.snsdk.com/monitor_browser/collect/batch/. The detailed view includes headers (sec-ch-ua, Referer, sec-ch-ua-mobile, User-Agent), message body (raw JSON), and a preview of the response content.

The right panel shows the performance testing interface for a task named 'demo'. It displays metrics: Average Response Time (509.42 ms), Response Time PCT95 (1616.48 ms), Average RPS (97.04 req/s), and Failure Rate (1.58%). Below these are three time-series charts: Response Time, Business Scenario RPS, and Interface Error Rate. The interface also includes a summary table of detailed data for each request type.

类型	名称	响应时间最小值	响应时间最大值	响应时间均值	响应时间中位数	错误次数	错误率	请求次数
request-GET	GET https://dss0.bds...	145ms	151ms	148.00ms	150ms	0	0.00%	2
reuest-GET	GET https://dss0.bds...	227ms	227ms	227.00ms	230ms	0	0.00%	1

如何复用 HttpRunner 的能力？



如何实现海量并发？

单机支持 3w+ RPS

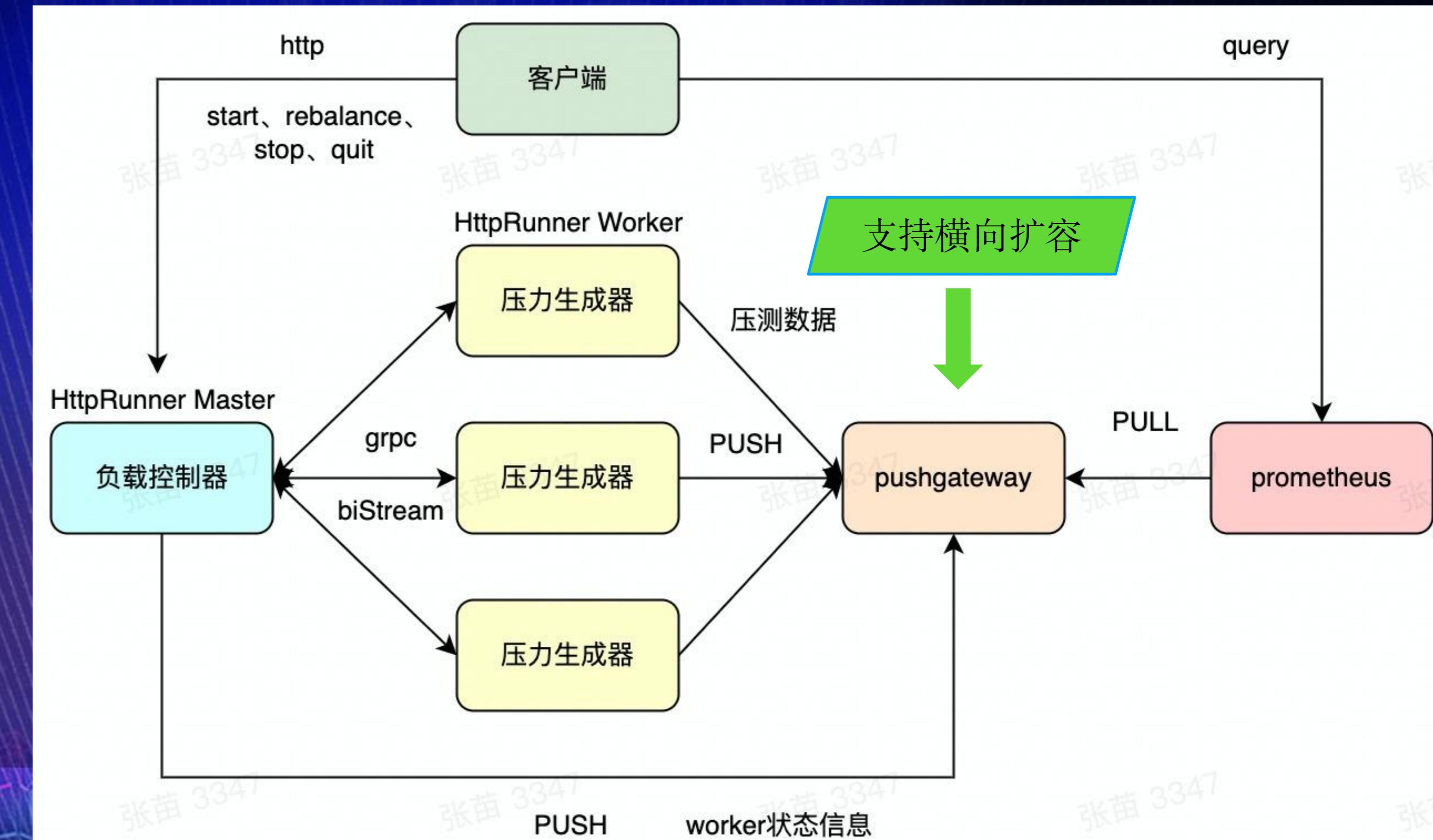
分布式压测服务

支持 HTTP API 控制压测流程

支持任务自动分发

支持多种发压策略（动态
RPS）

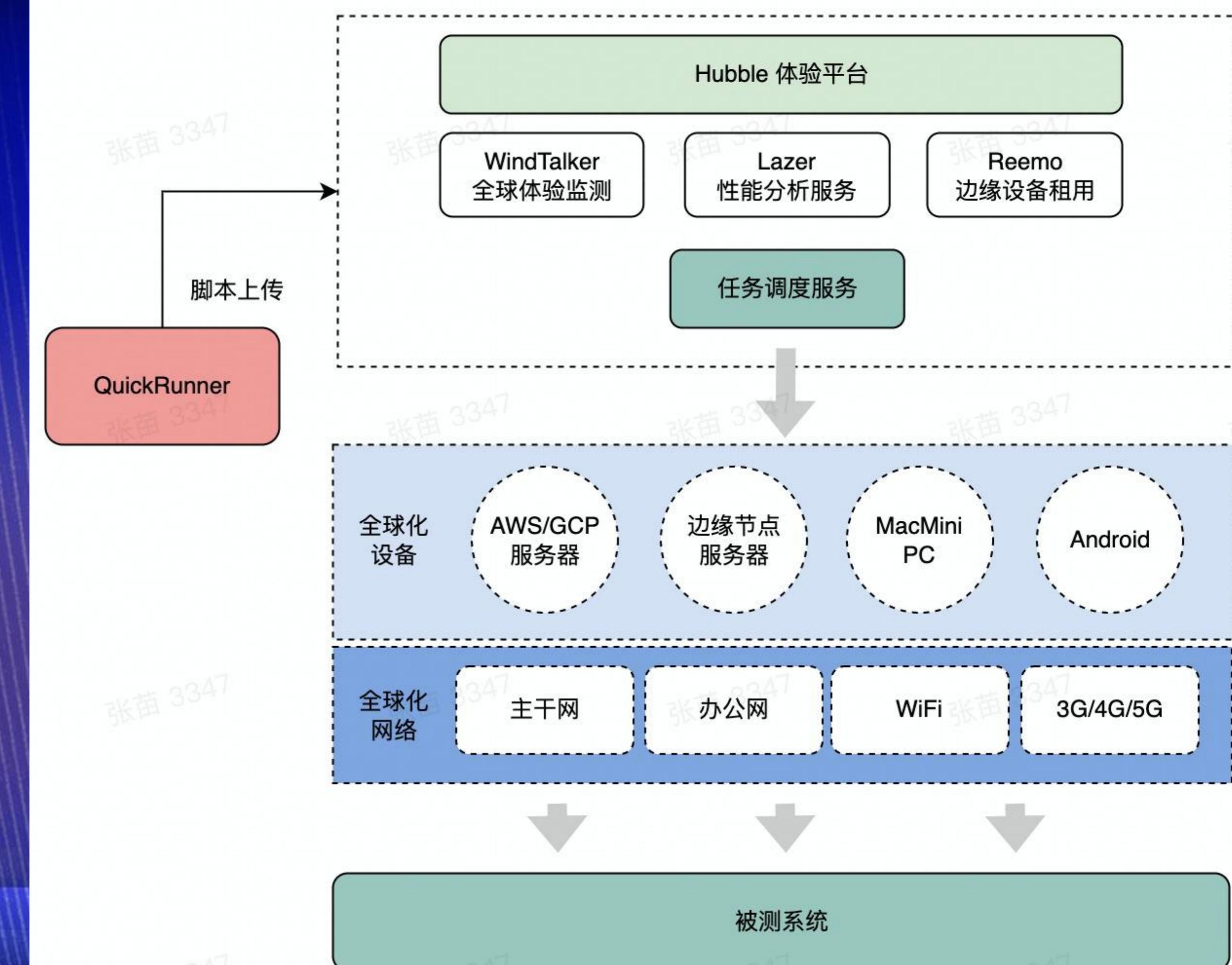
控制指令与数据上报分离



如何实现全球化测试？

支持对压测结果进行全链路智能分析，基于规则或算法模型自动发现性能瓶颈，大幅缩短问题定位时间。

- 接口测试脚本可以上传到 WindTalker，实现全球化网络性能监测和场景化监测
- 基于全球的分布式节点，实现全球真实链路性能压测



QuickRunner vs. HttpRunner

社区版免费开放

特性维度	HttpRunner	QuickRunner 社区版	QuickRunner 标准版
是否收费	✗	✗	✓
产品形态	开源框架	UI 客户端产品	UI 客户端产品 + 付费云端资源
信创环境支持	✓	✓	✓
网络协议限制	不限	不限	不限
发压能力限制	不限	不限	不限
分布式压测	✓	✓	✓
测试报告	简单图表	简单图表	<ul style="list-style-type: none"> 丰富的测试报告 报告导出 自定义测试报告
结果分析	✗	✗	支持结果分析 & 诊断
团队协作	✗	✗	支持；按人数收费
云端资源	✗	✗	按需付费
私有化部署	提供源码	单机工具不涉及	协作能力、管理平台 支持私有化部署
技术支持	开源社区支持	开源社区支持	1对1支持
培训服务	✗	✗	✓
售价	✗	✗	刊例价

4、HttpRunner 的开源展望



开源 vs. 商业化

开源协议?

商业化产品?

公司内部版?

=l] _da . *,

Kl aj ? kma

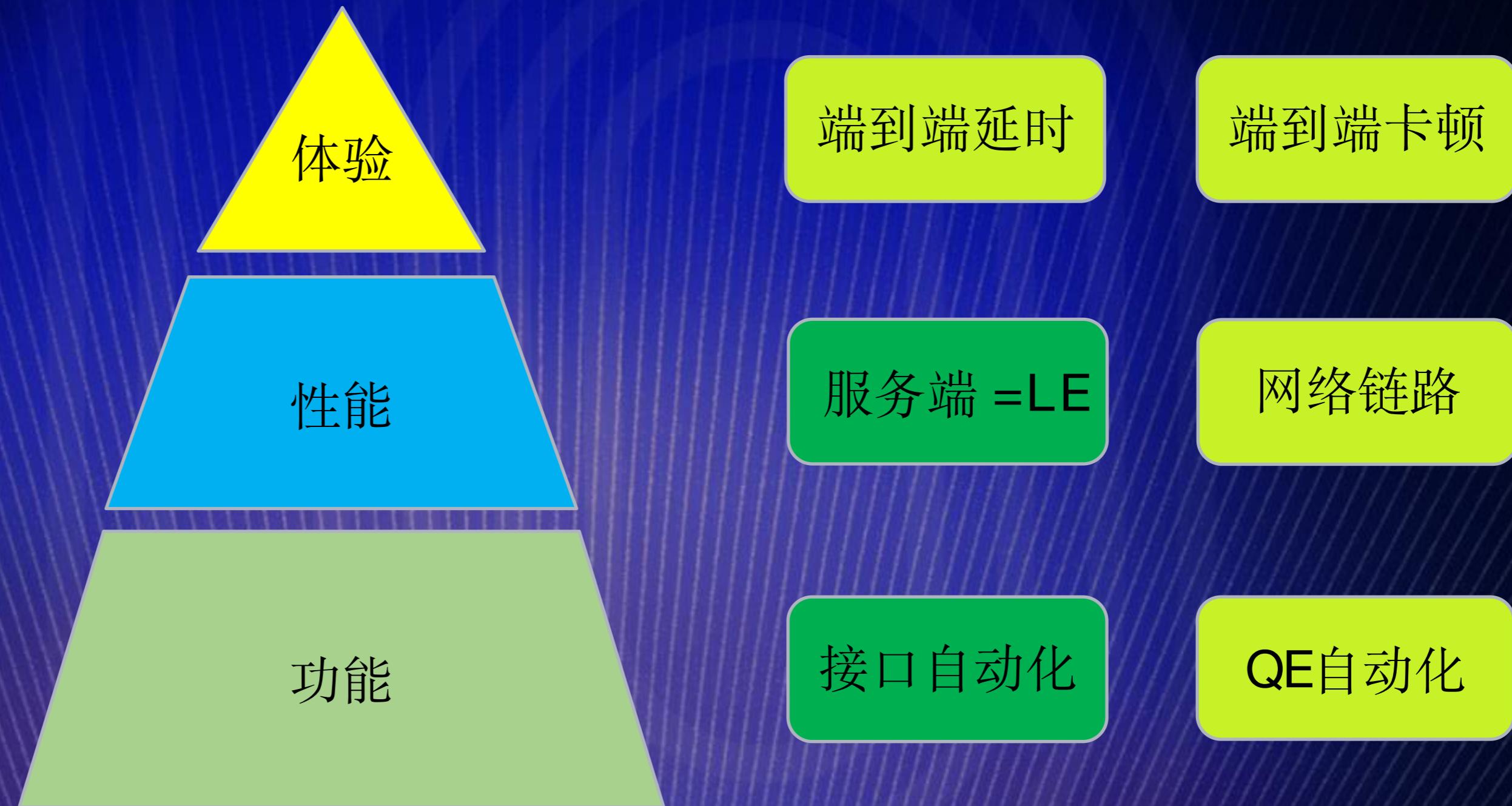
插件化



一体化体验评测能力

参考“马斯洛需求层次”

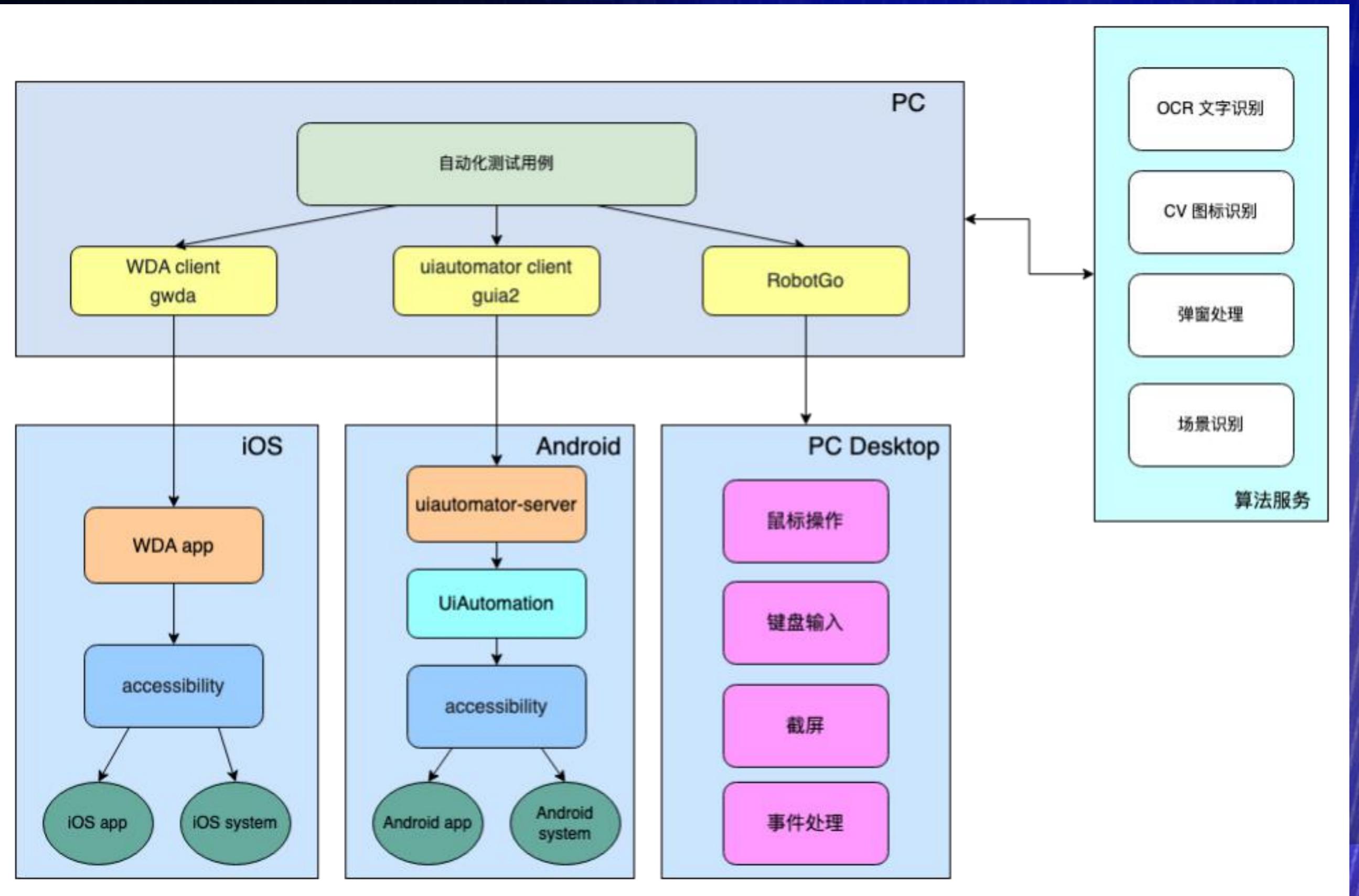
应用系统对质量的需求
也可以划分为三个层次



一站式？

一体化！

面向体验评测的 UI 自动化能力



cepdq^*_ki -+dp mqj j an+dp mqj j an+r 0+
dn + gc+qe p

Pd] j go

cepdq^*_ki -+al[a_p[e_ ^q^ ^ha+cs`]
cepdq^*_ki -+al[a_p[e_ ^q^ ^ha+cqd .
cepdq^*_ki -+ck)r ck+rk^kpck

低代码 + 无代码



Q & A



谢谢
THANKS

