

汽车之家电商架构测试实践

新车电商事业部 王公瑾



2008年



毕业于四川大学



2008-2014年

就职于移动139、
唯品会等多家公司

运维->测试->测试
开发->测试主管



2014年

汽车之家

测试架构师
电商交易团队测试
负责人



目录 CONTENTS



□ 业务高速增长汽车之家新车电商

□ 测试流程改进与敏捷实践

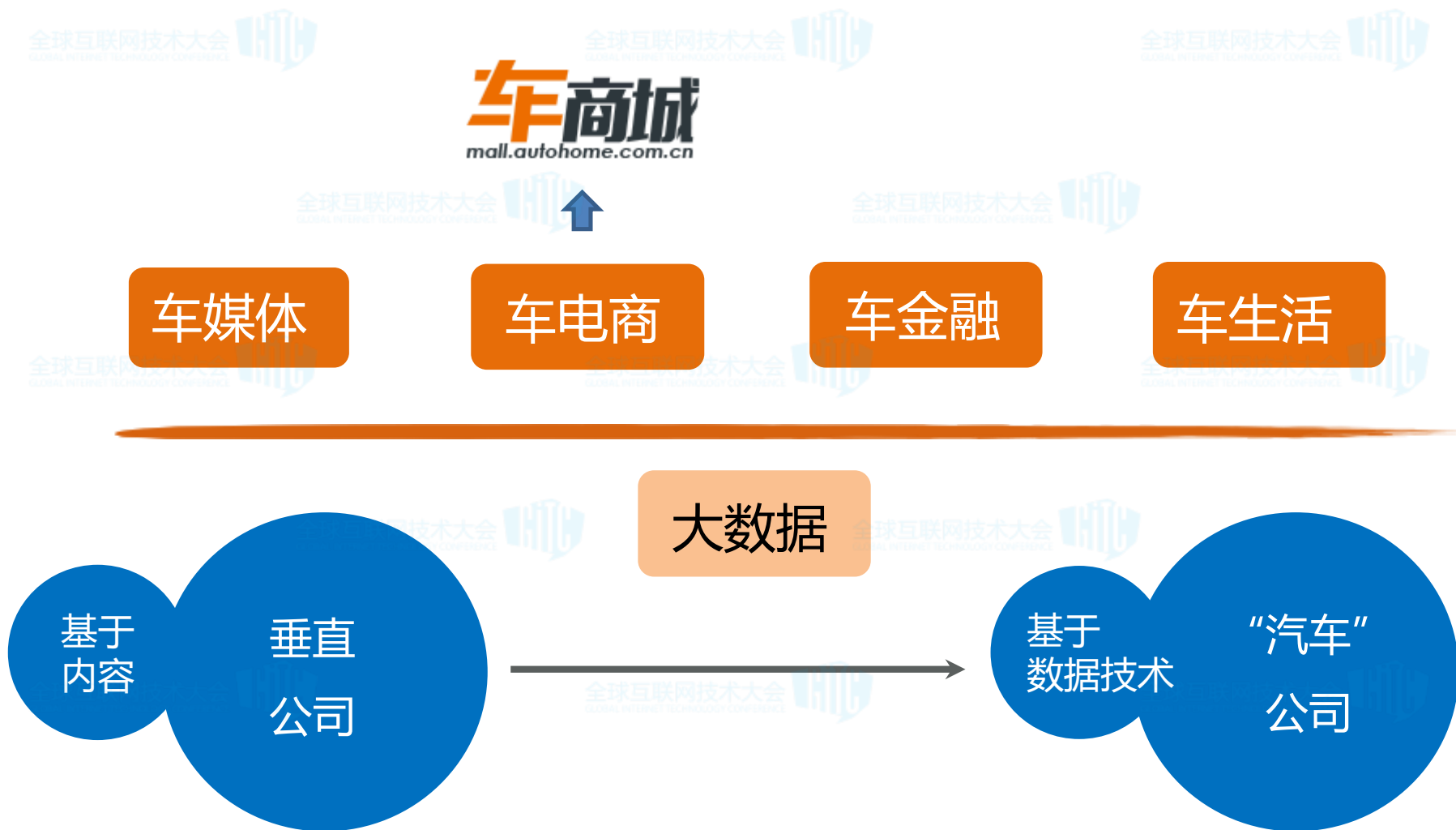
□ 架构设计原则与测试实践

□ 快速搭建Mock Server

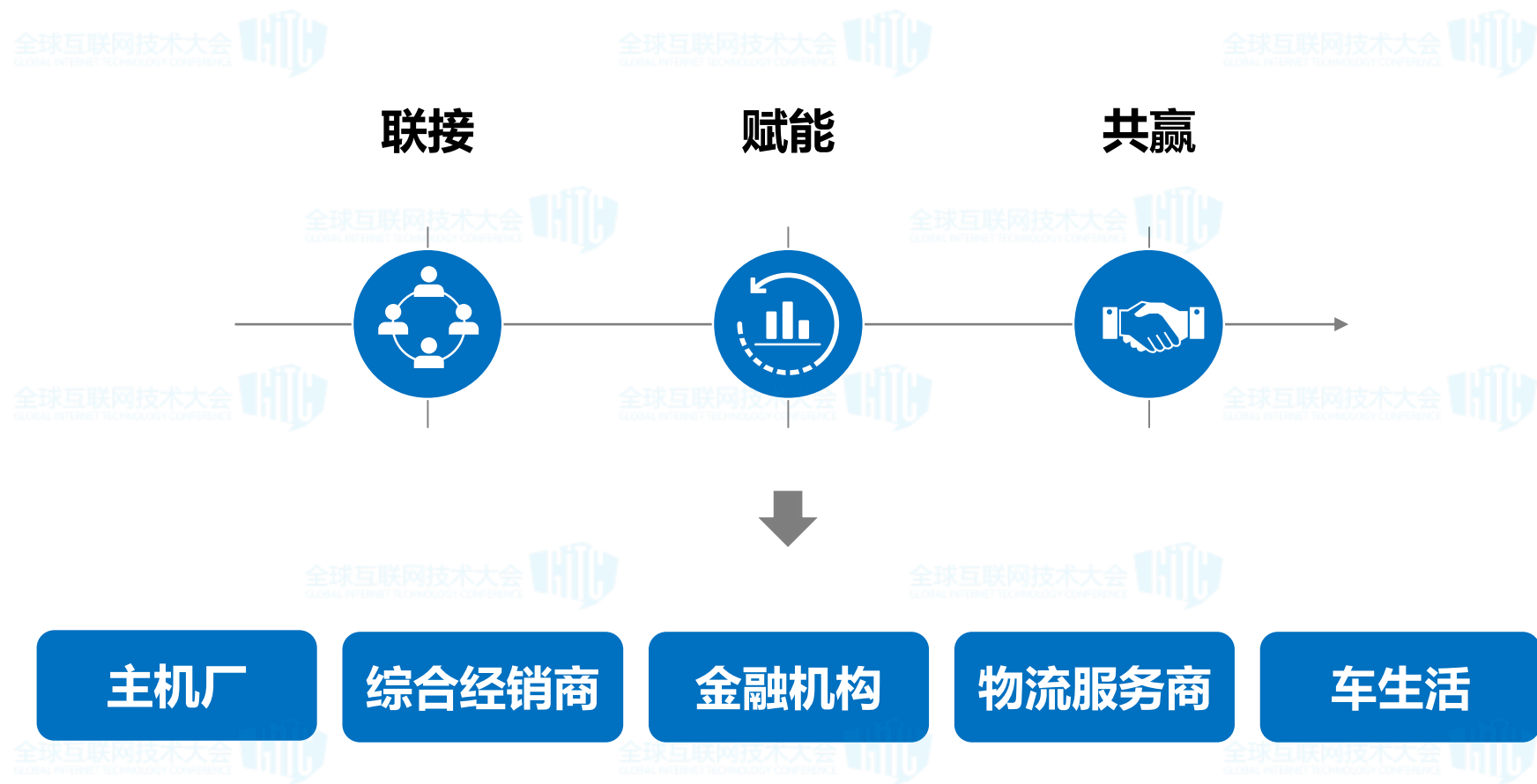
□ 持续集成实践

□ 自动化测试实践

汽车之家战略升级，推出“4+1”战略，车商城定位新车电商开放平台



车商城为用户、主机厂、综合经销商等打造完整的线上交易平台





注：数据来源：近3个月车商城运营数据



测试流程改进与敏捷实践



- ### 1 代码反复改

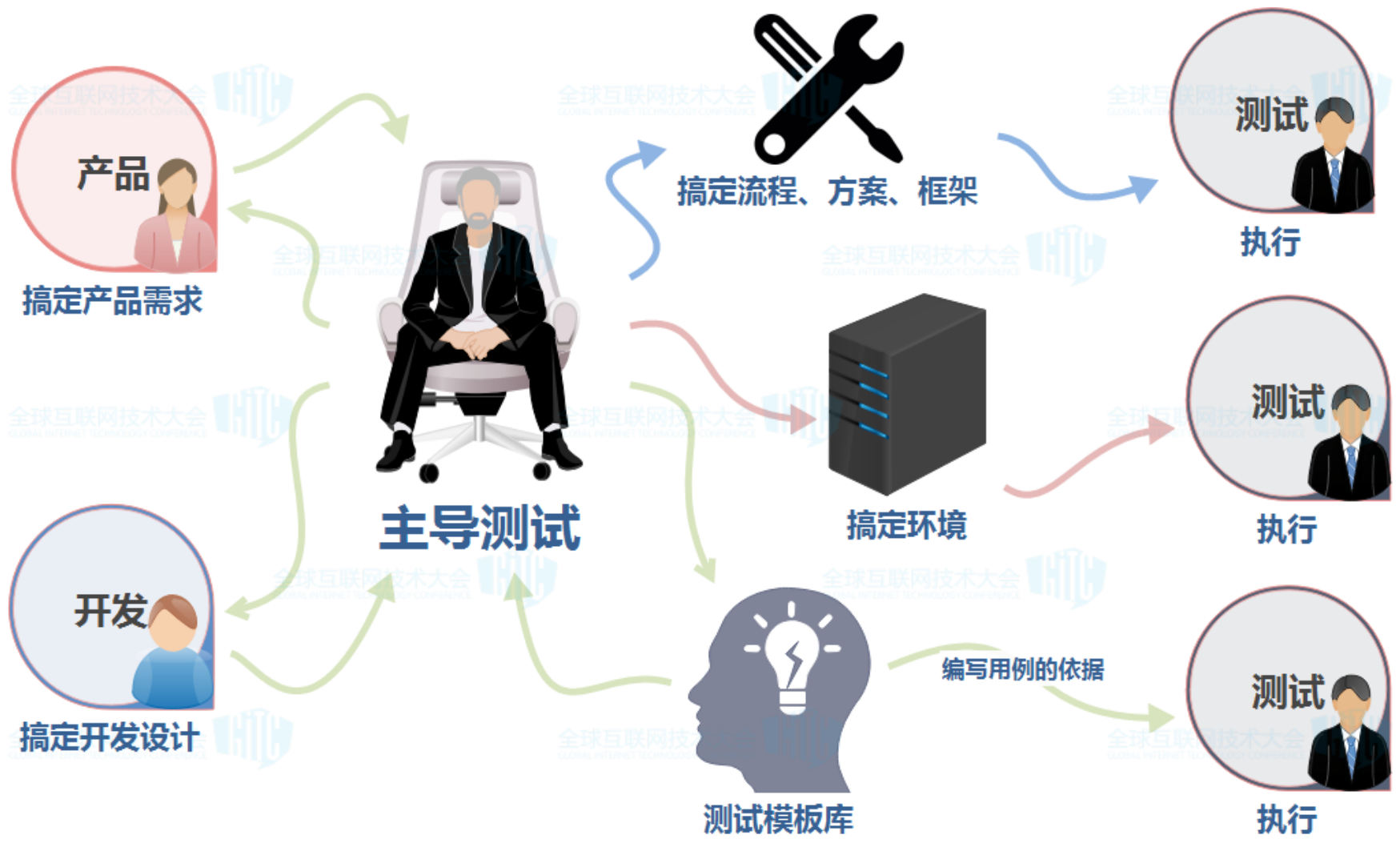
测试对需求和开发设计投入太小
参与设计对测试要求高
- ### 2 测试环境乱

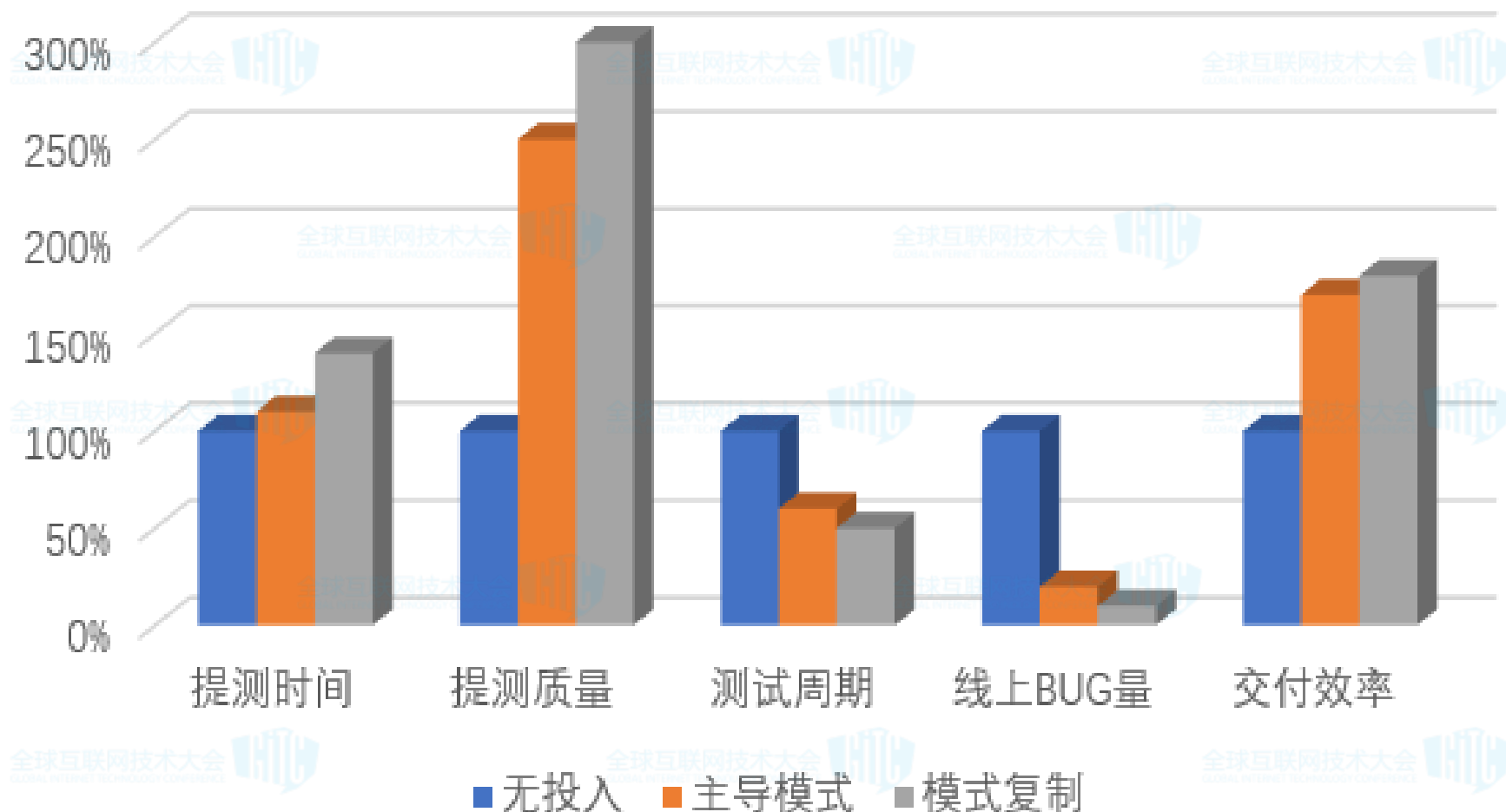
测试不清楚服务调用关系
测试不理解分布式服务和各种中间件
- ### 3 依赖外部环境

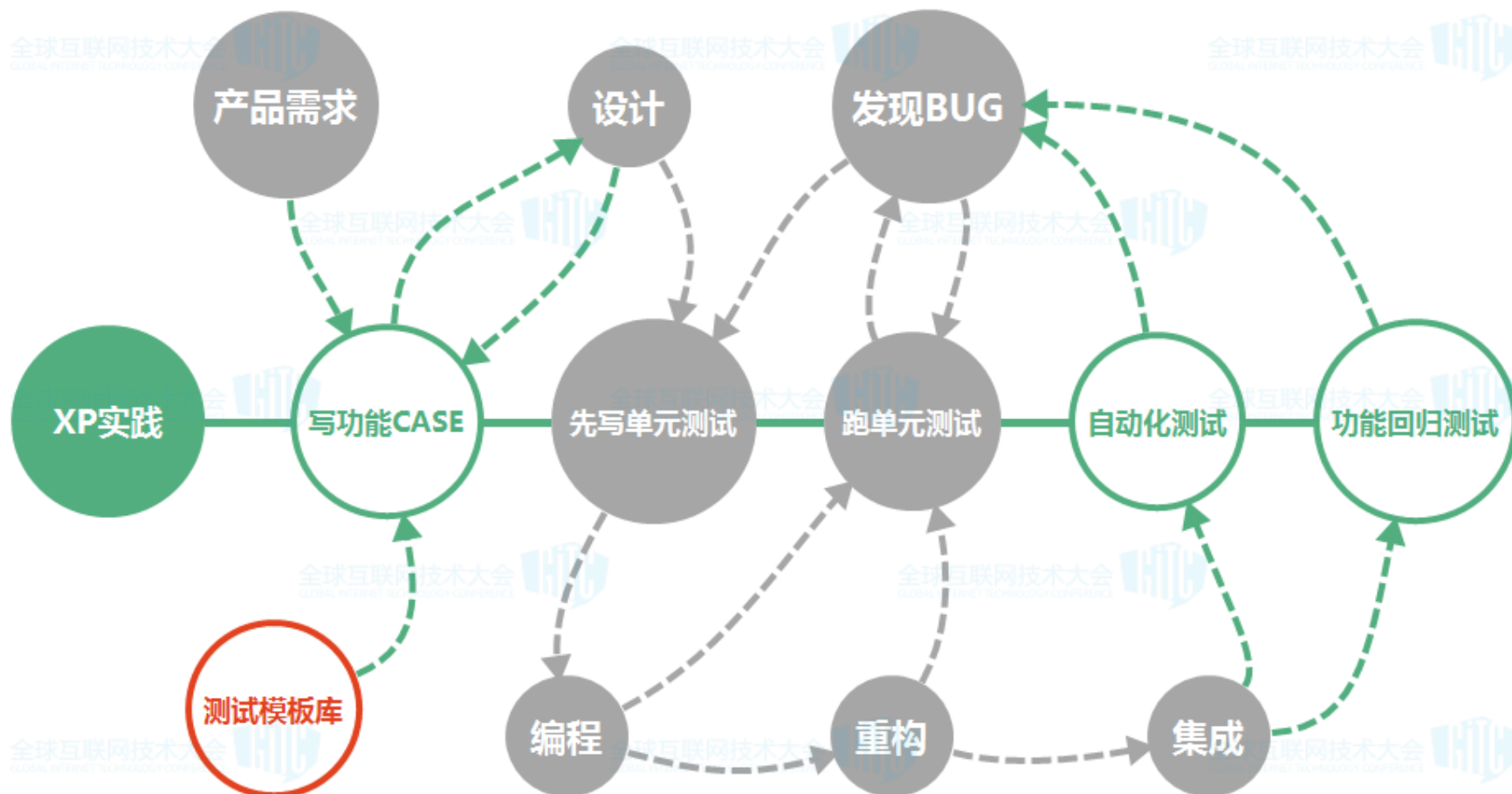
依赖环境未提测或者不通
集成测试缺少手段
- ### 4 测试手段单一

缺少监控、数据筛查的手段
缺少自动化手段

主导测试流程







XP流程测试开发比



正常流程测试开发比



架构设计原则与测试实践

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



读写分离

反向代理

负载均衡

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



事件驱动架构

分层架构

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

RPC



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

微服务架构



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

消息队列



分布式架构

缓存

共享内存

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

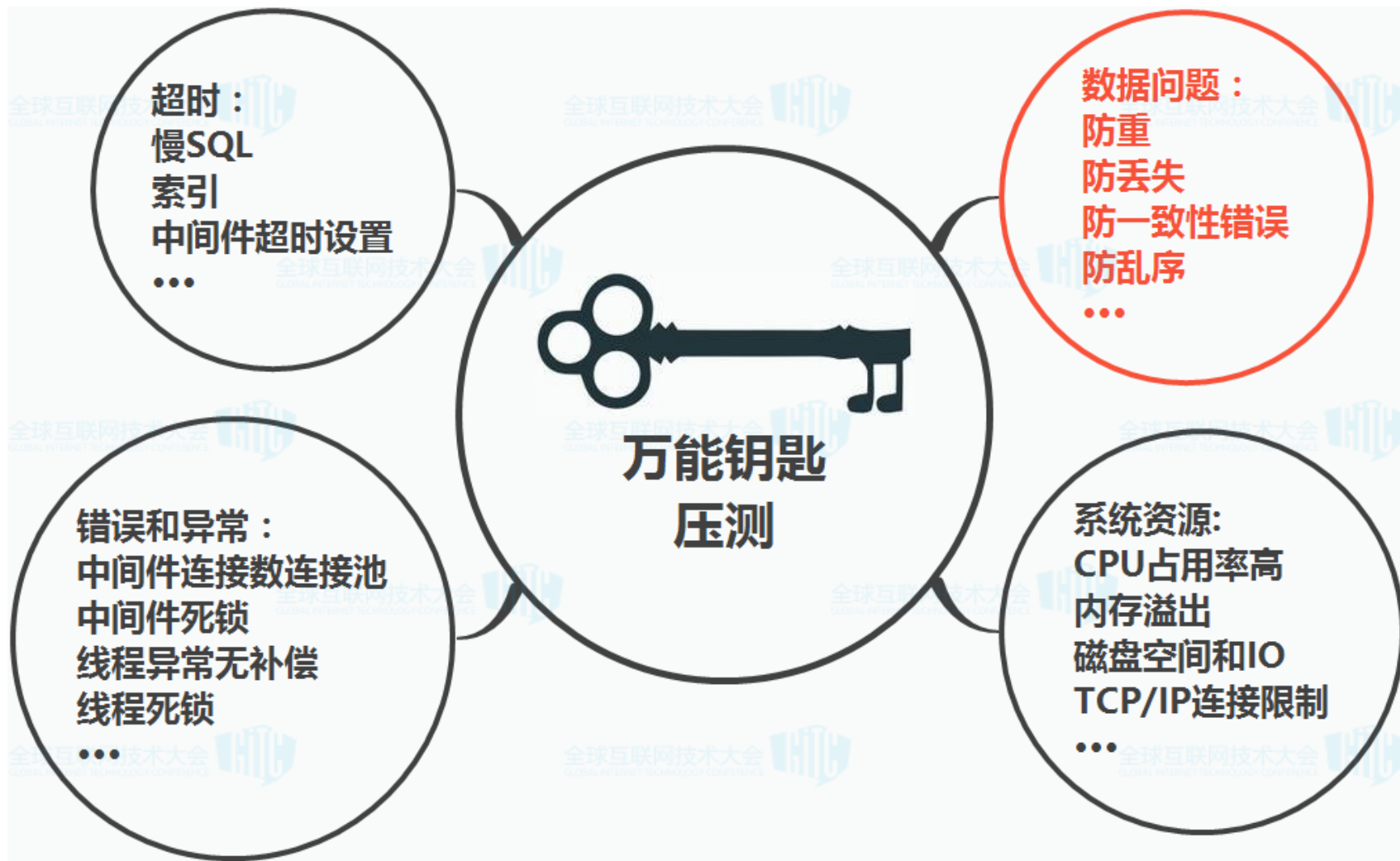


全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE





全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



很好，但是有些迟了

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE



如何做到测试前置

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

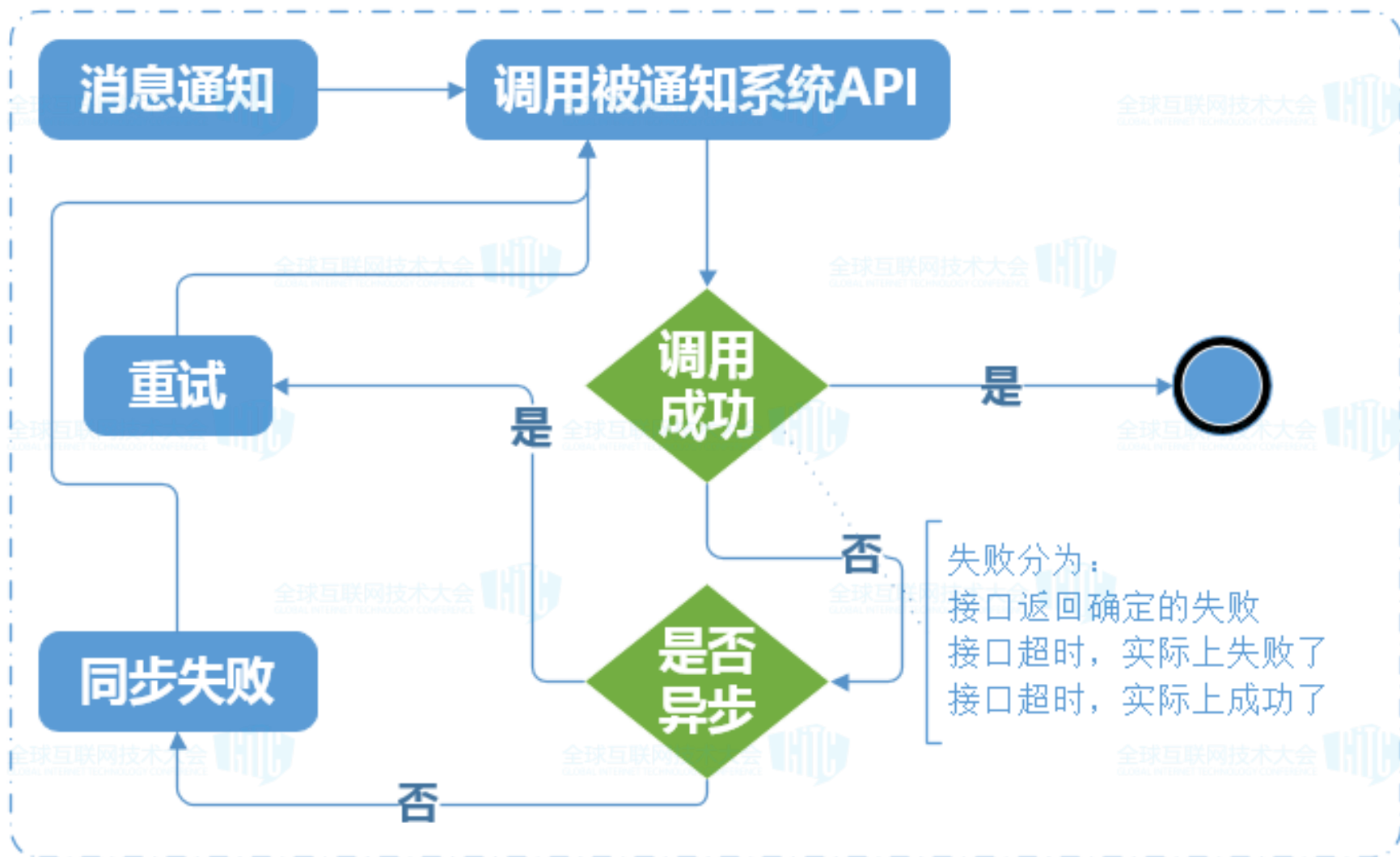


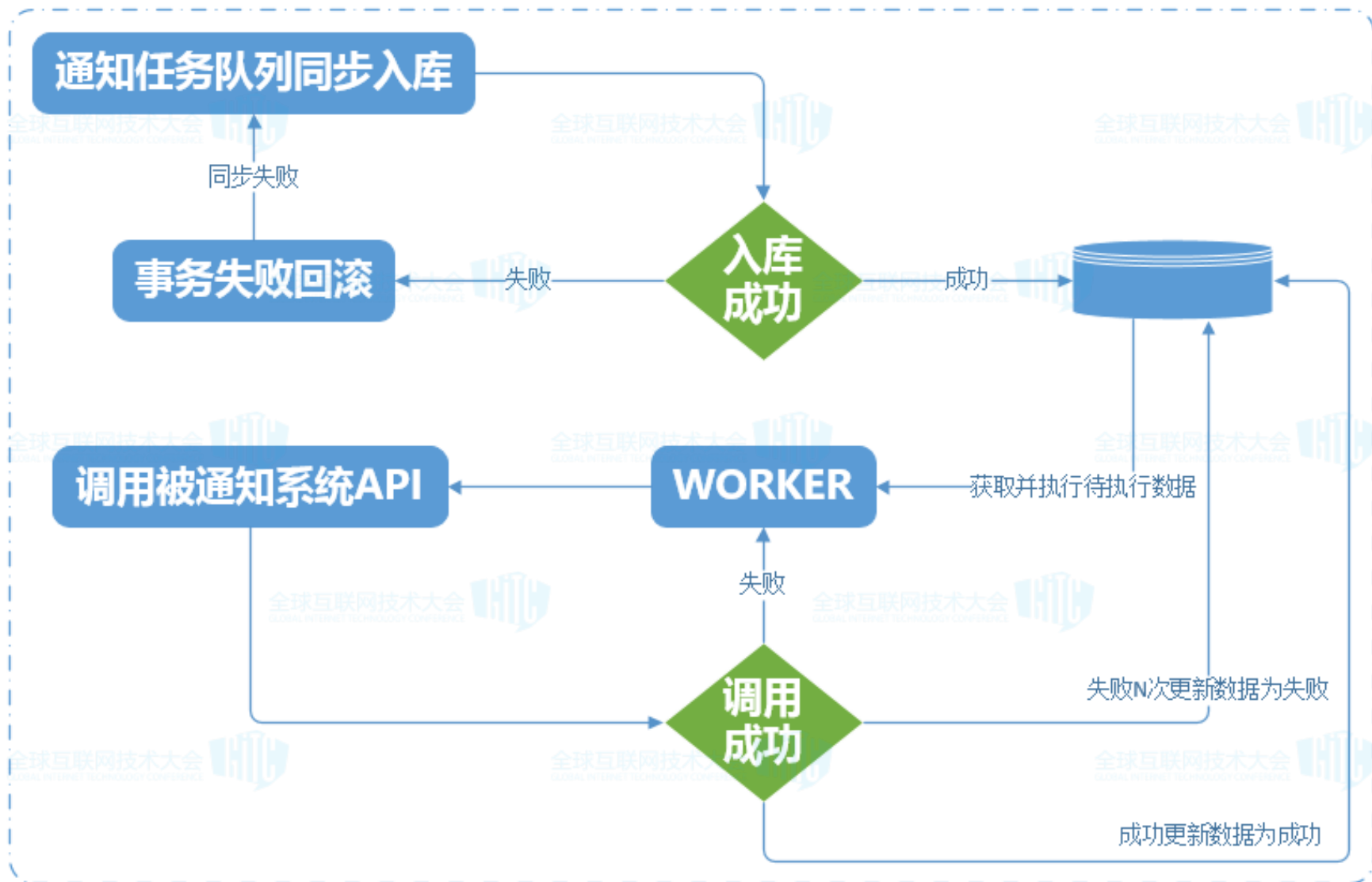
全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

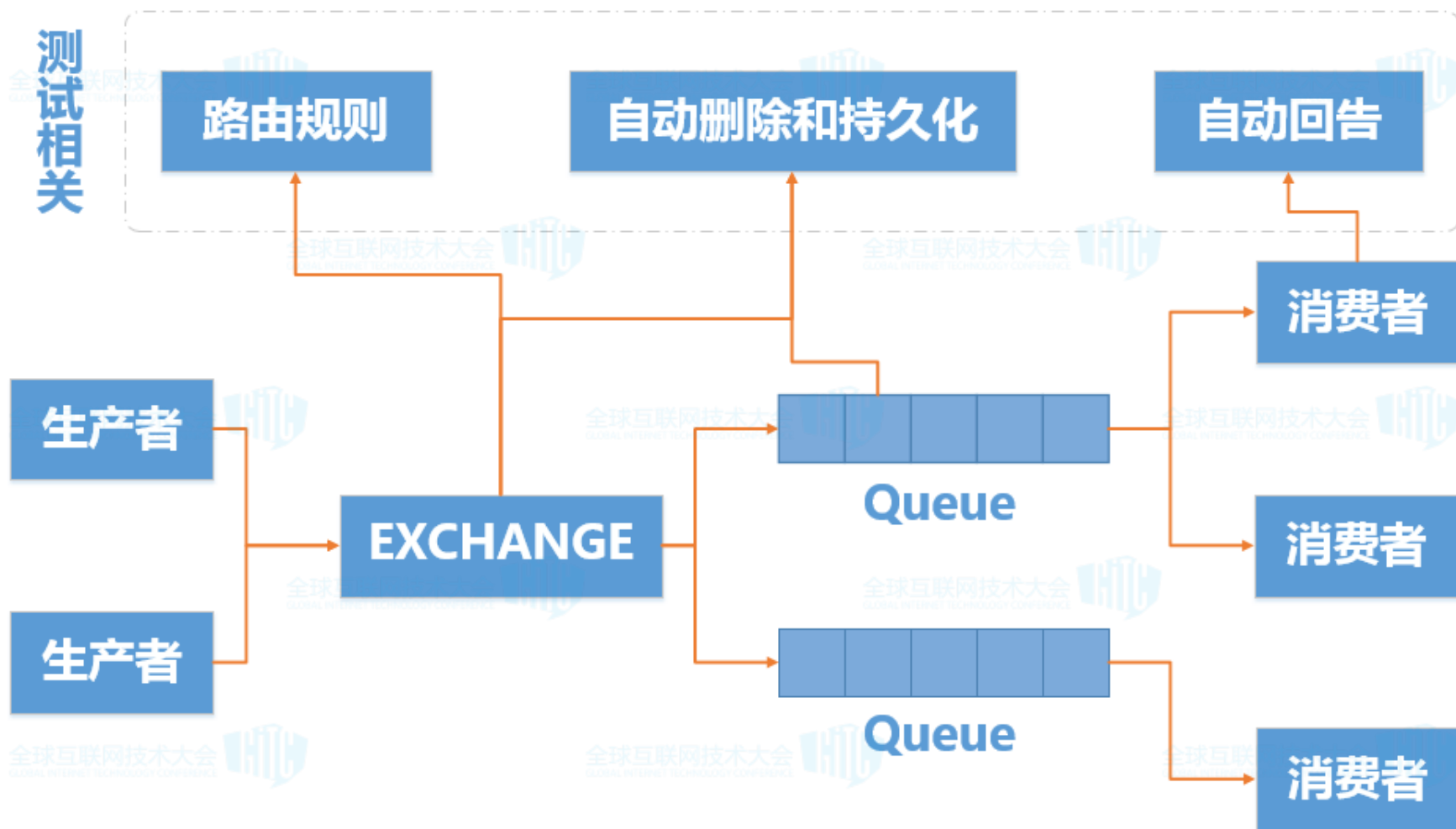


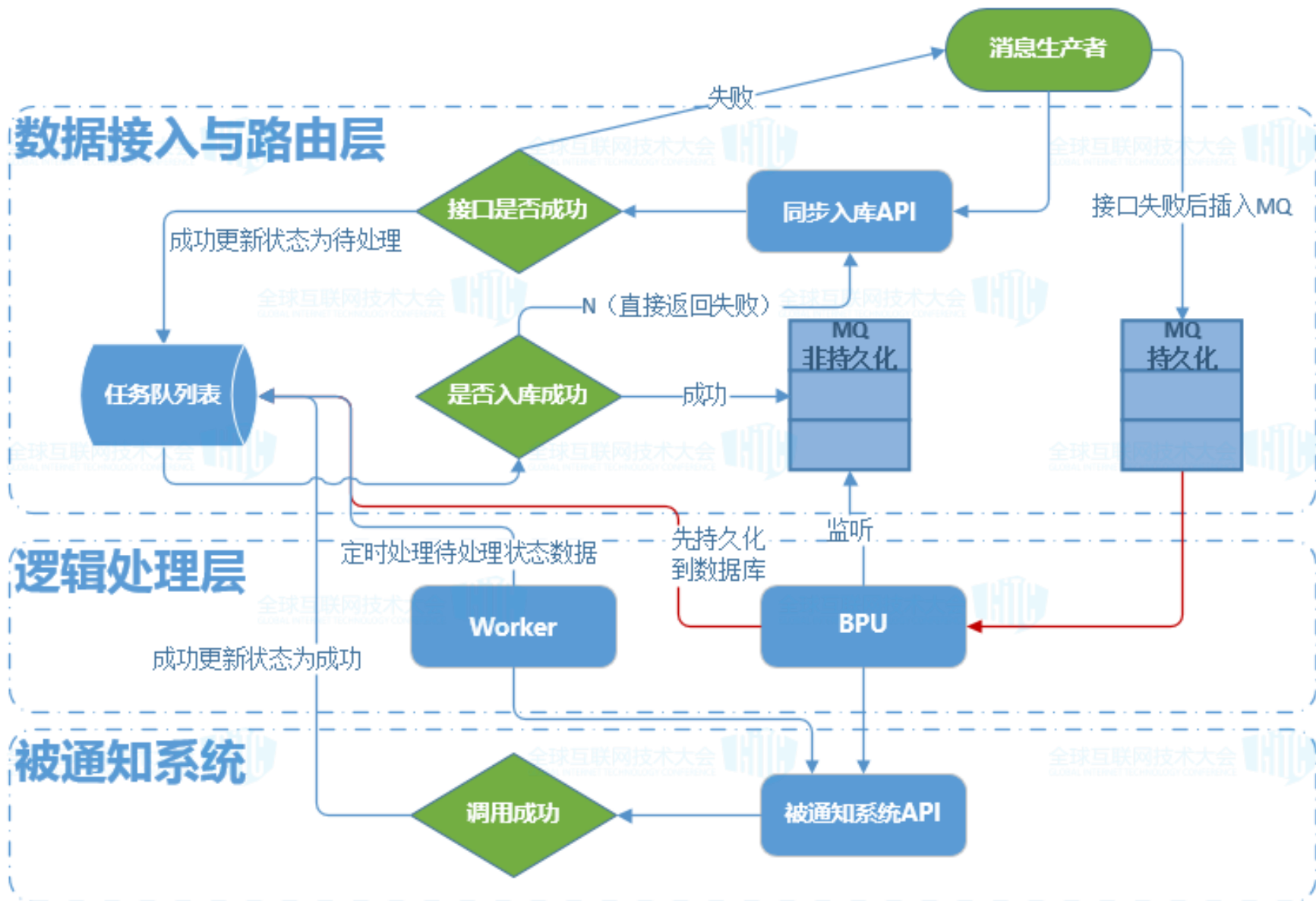
全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

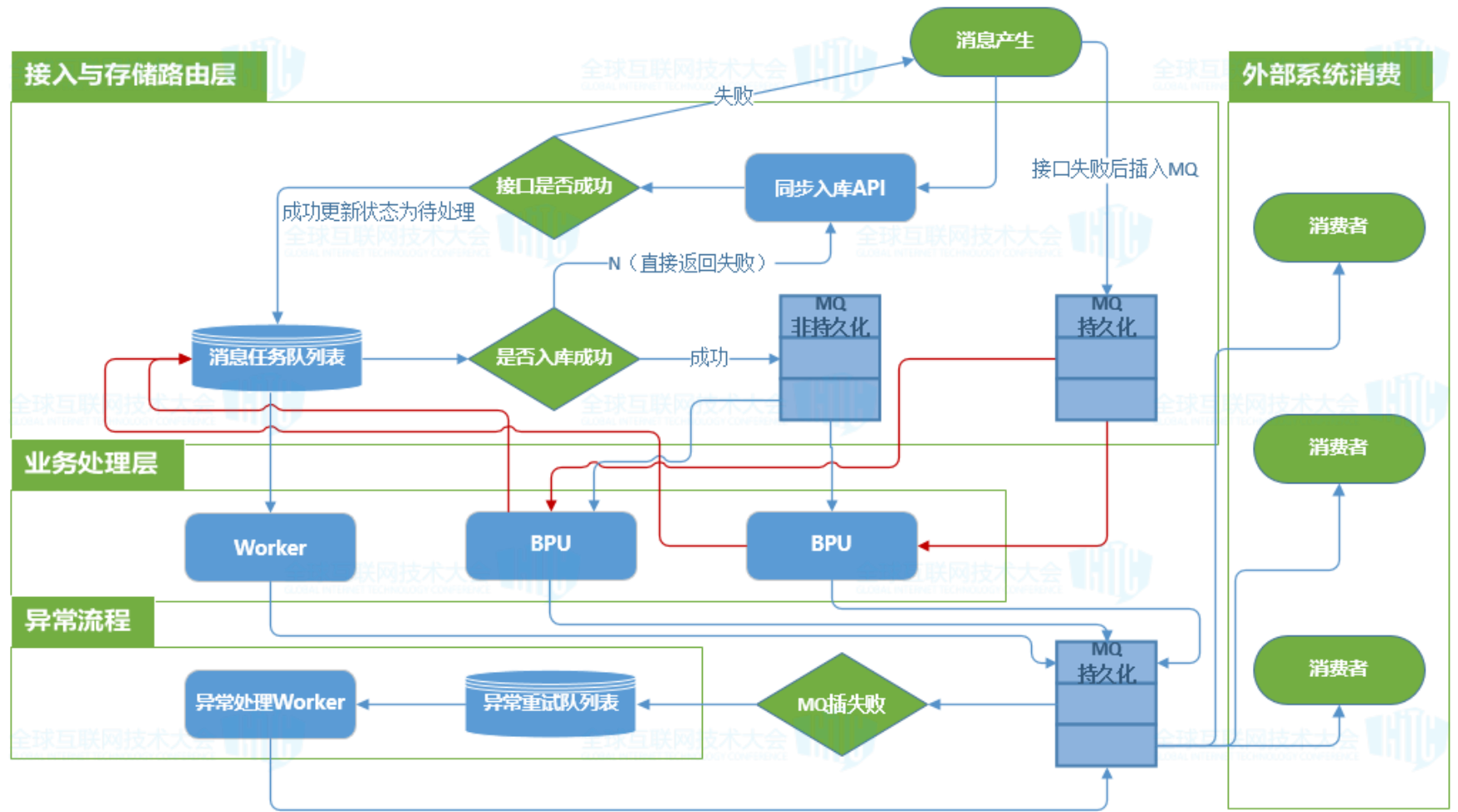


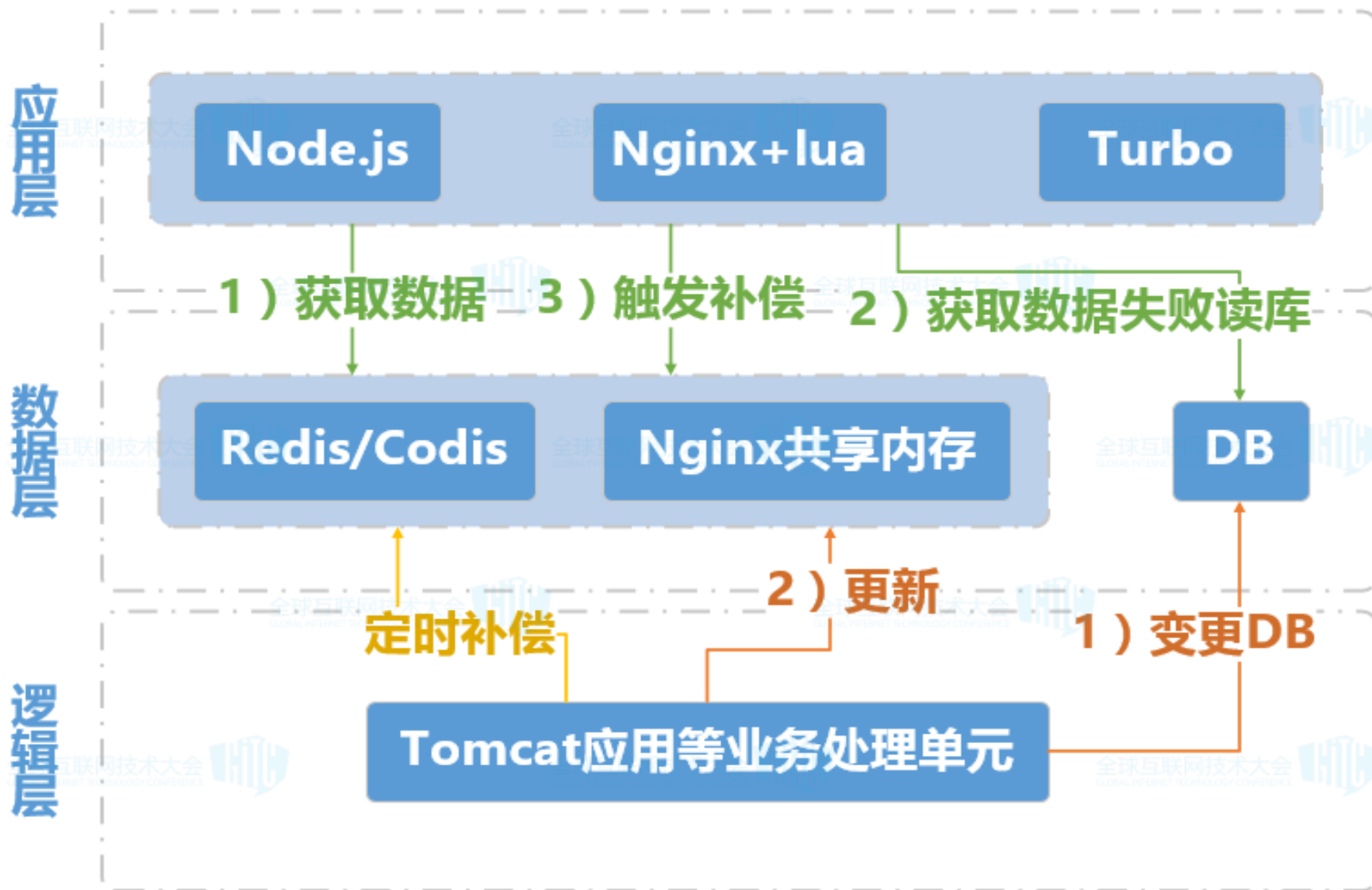


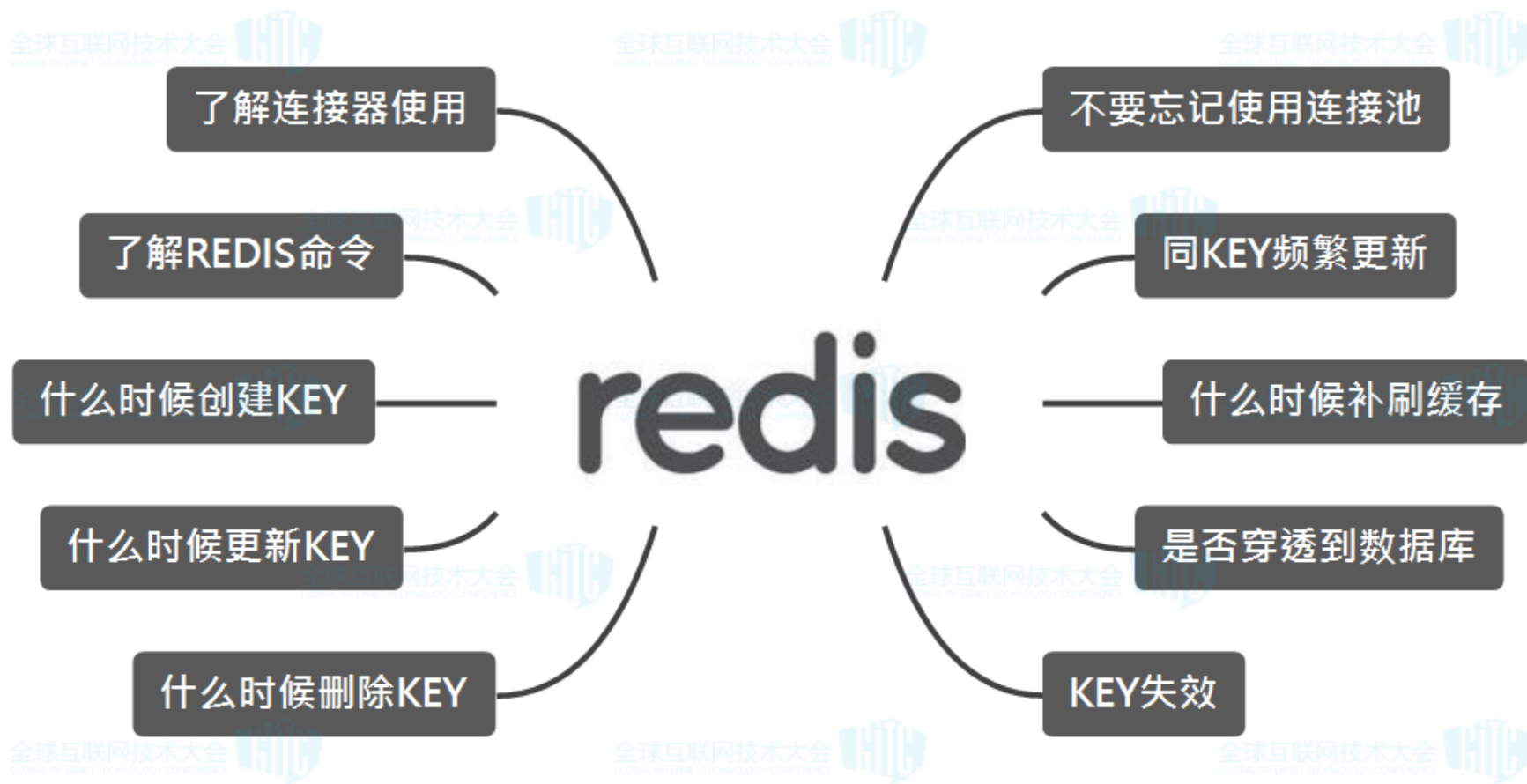














全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

测试如何参与开发和架构设计

全球互联网技术大会
GLOBAL INTERNET TECHNOLOGY CONFERENCE

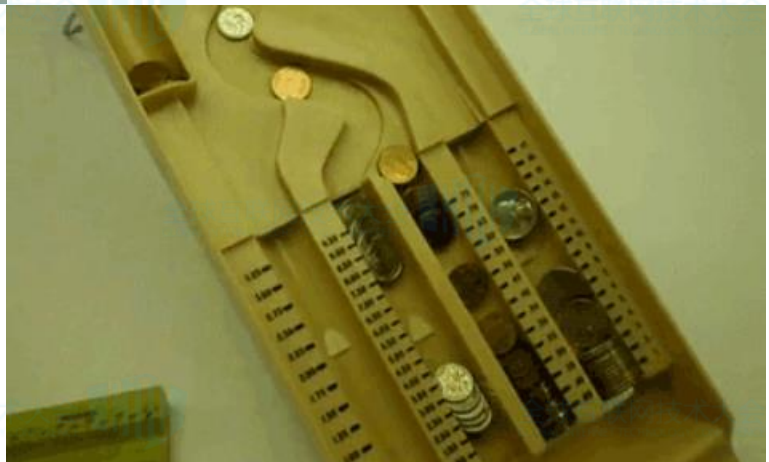






分拣

计数



架构指导原则	详细描述
解耦	降低模块间关系的复杂度
高内聚	单一职责，功能内聚和顺序内聚
关注行为分离	小到模块、大到系统，要清楚边界和职责
关注数据分离	系统之间的分离和系统内的分离
横向、纵向分离	模块化和分层
轻重分离	辩证地尽量去做异步设计
禁用设计	通过开关或者配置可以热处理禁用和修改程序功能
灰度发布	可配置的灰度升级

快速搭建Mock Server

NGINX

server中配置多个location接口地址

server name 接口工程1

server name 接口工程n ...

location 访问地址1

location 访问地址n ...

LUA

接口对应mock脚本

rewrite_by_lua_file Mock脚本1.lua

rewrite_by_lua_file Mock脚本n.lua ...

REDIS

根据接口参数访问不同的key的value

key1 : 关键字1+_{数据参数}*n

key2 : 关键字2+_{数据参数}*n ...

应用场景	操作步骤	优点描述
新增接口	1) 新增Mock接口的Lua脚本	1) 方便部署，只需要定义location指定脚本位置即可 2) 需要对相同接口做数据隔离时，脚本编制方便 3) 重启速度快，nginx -s reload秒生效
	2) 新增nginx配置	
	3) nginx -s reload	
变更接口逻辑	1) 修改Mock接口的Lua脚本	3) 重启速度快，nginx -s reload秒生效
	2) nginx -s reload	
变更接口返回值	1) set对应的Redis的key的值	简单、方便、隔离数据

持续集成实践

部署平台

检出代码

编译及分布式隔离

集成运维工具

异常日志查看

集成覆盖率

集成自动化

运维工具

Fabric

日志收集

Logstash

Elasticsearch

Kibana

覆盖率监控

Jacoco

自动化测试

Swagger

T-Client

接口配置

基础用例生成

测试链管理

计划管理

覆盖率项目创建

自动化项目创建

压测工具和容器

覆盖率报告整合

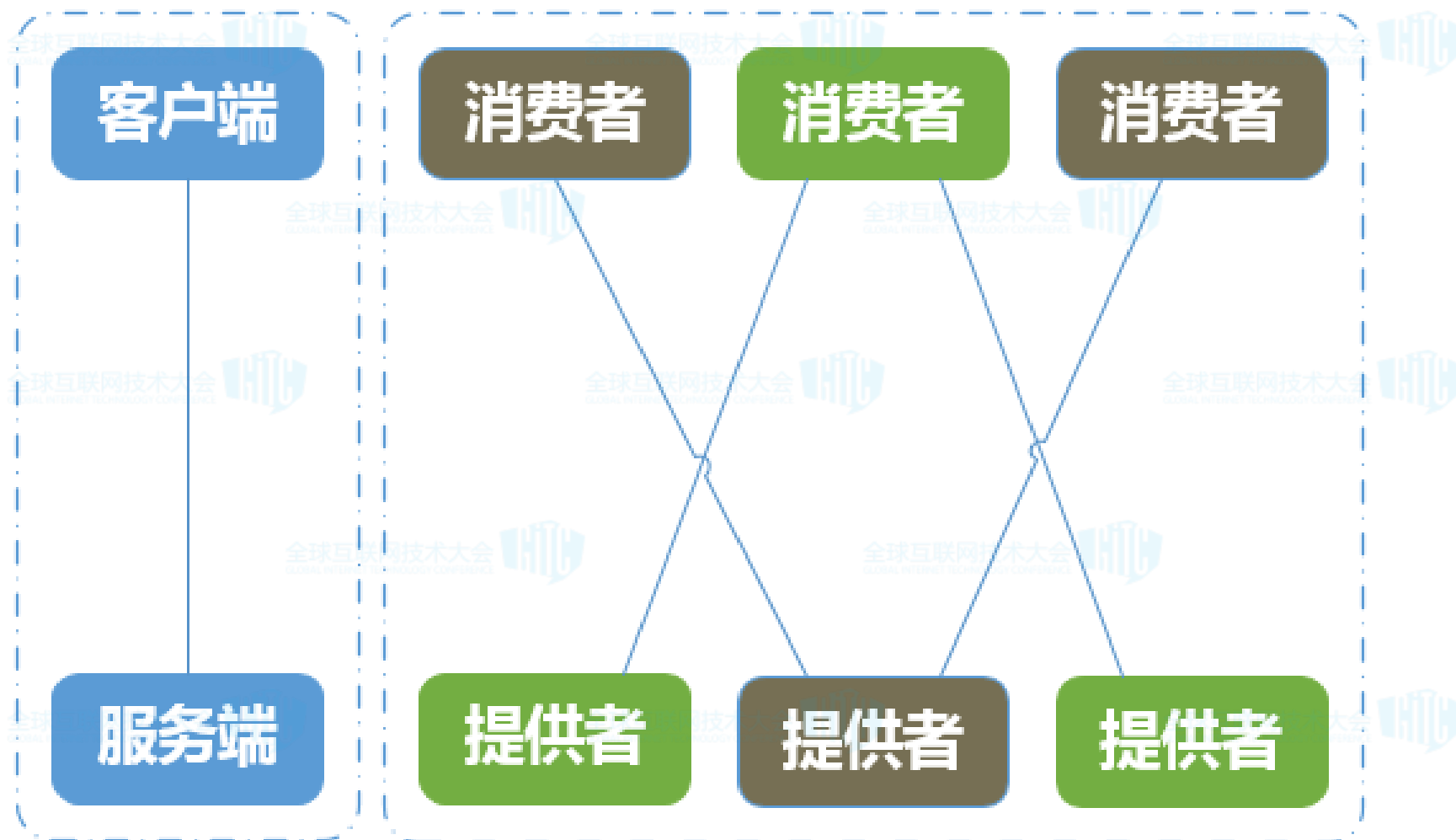
自动化报告整合

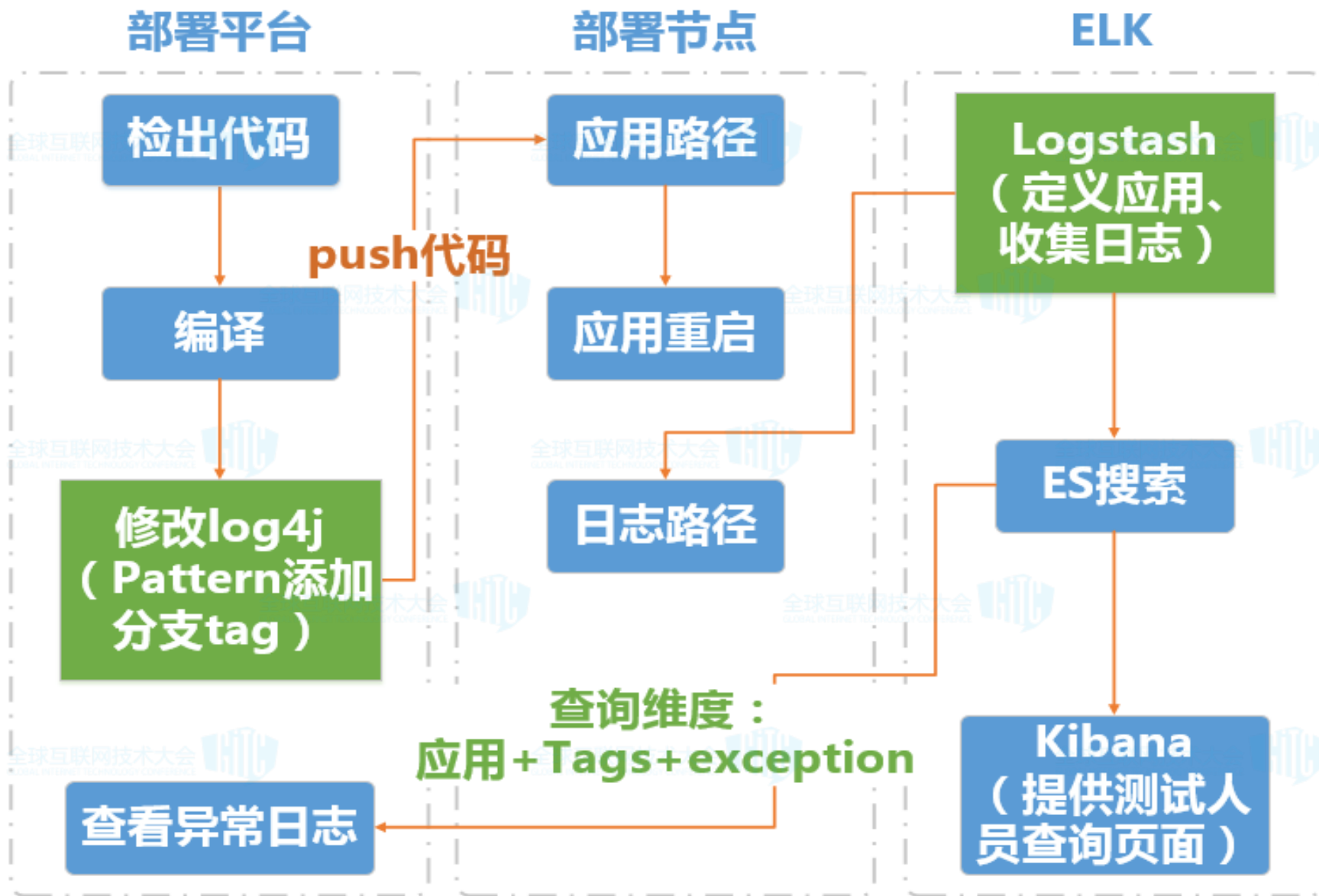
流量复制和放大

汽车之家测试平台

HTTP服务

TURBO服务





自动化测试实践

自动化测试

智能化

可集成

链路化

平台化

智能化

接口创建和编辑时
触发生成基础用例

链路化

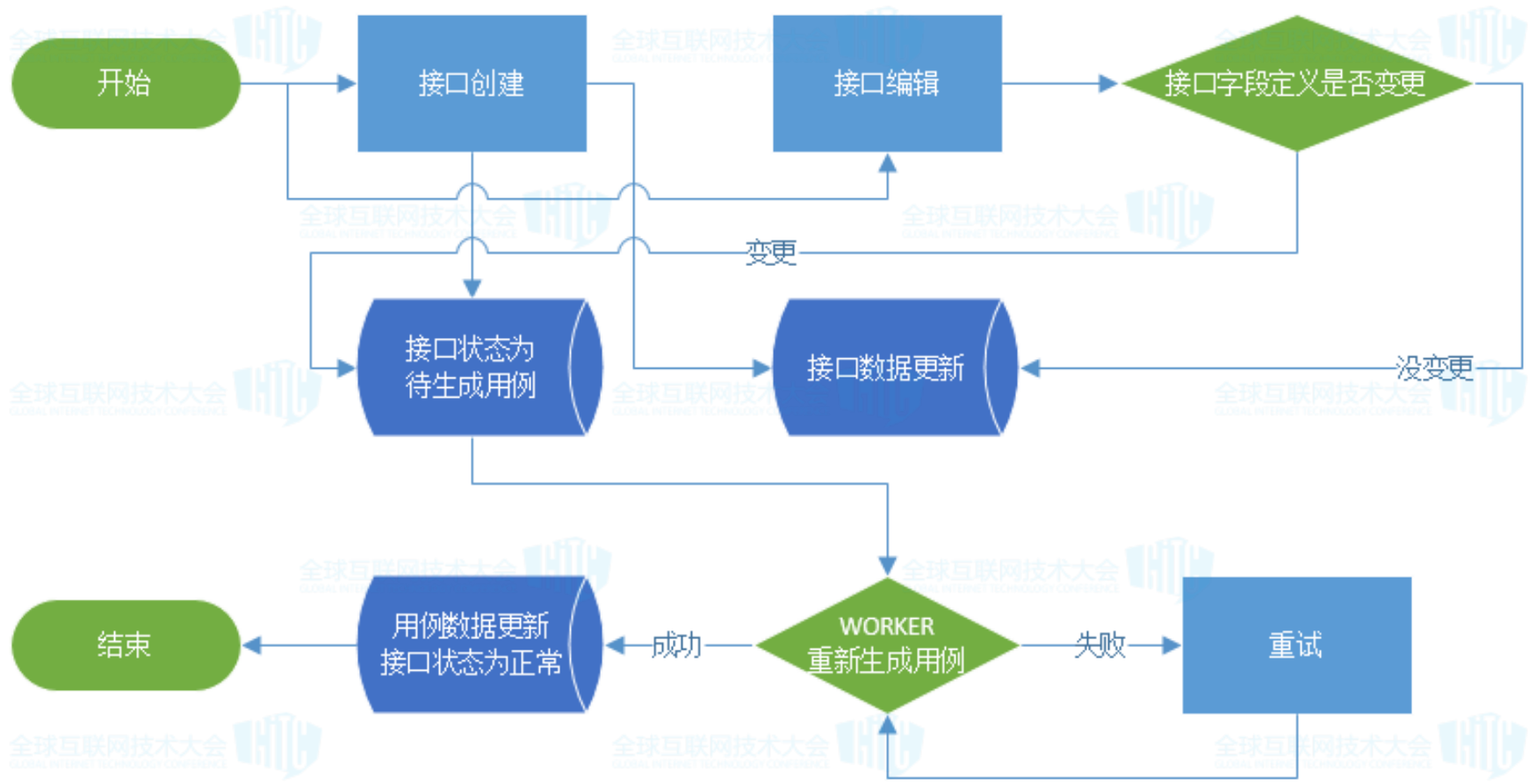
接口组装测试链路
线程组控制并串行

可集成

集成Jenkins
集成之家自动化平台

平台化

接口、用例、线程组
链路、测试集、计划



swagger_py_codegen包

源码修改

```
def _process(self):  
    views = self._process_data()  
    yield Router(dict(views=views))  
    for view in views:  
        yield View(view, dist_env=dict(view=view['enc  
    if self.with_spec:  
        try:  
            import simplejson as json  
        except ImportError:  
            import json  
        swagger = {}  
        swagger.update(self.swagger.origin_data)  
        # swagger.pop('host', None)  
        swagger.pop('schemes', None)  
        yield Specification(dict(swagger=json.dumps(s
```

生成Flask代码

swagger_py_codegen代码生成指令

指定swagger标准配置文件

api.json

推送提交更新

2) 重新执行生成指令

3) 重启Flask应用

api.json

接口创建和编辑

1) 更新api.json

接口测试平台

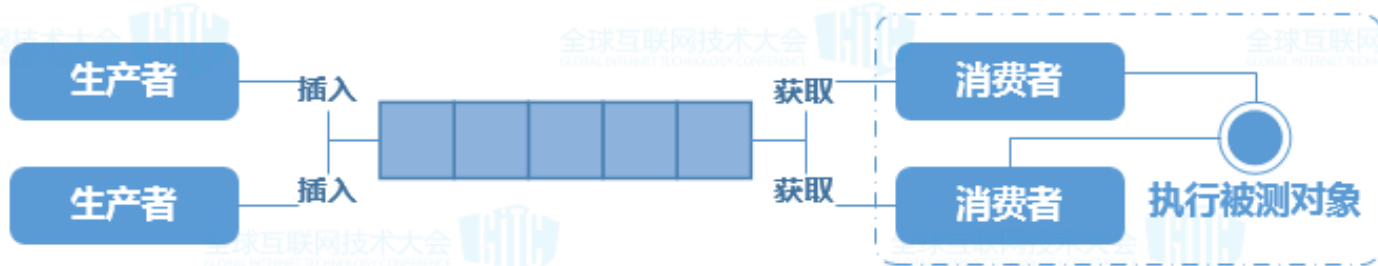


线程组

创建线程池和队列并start线程



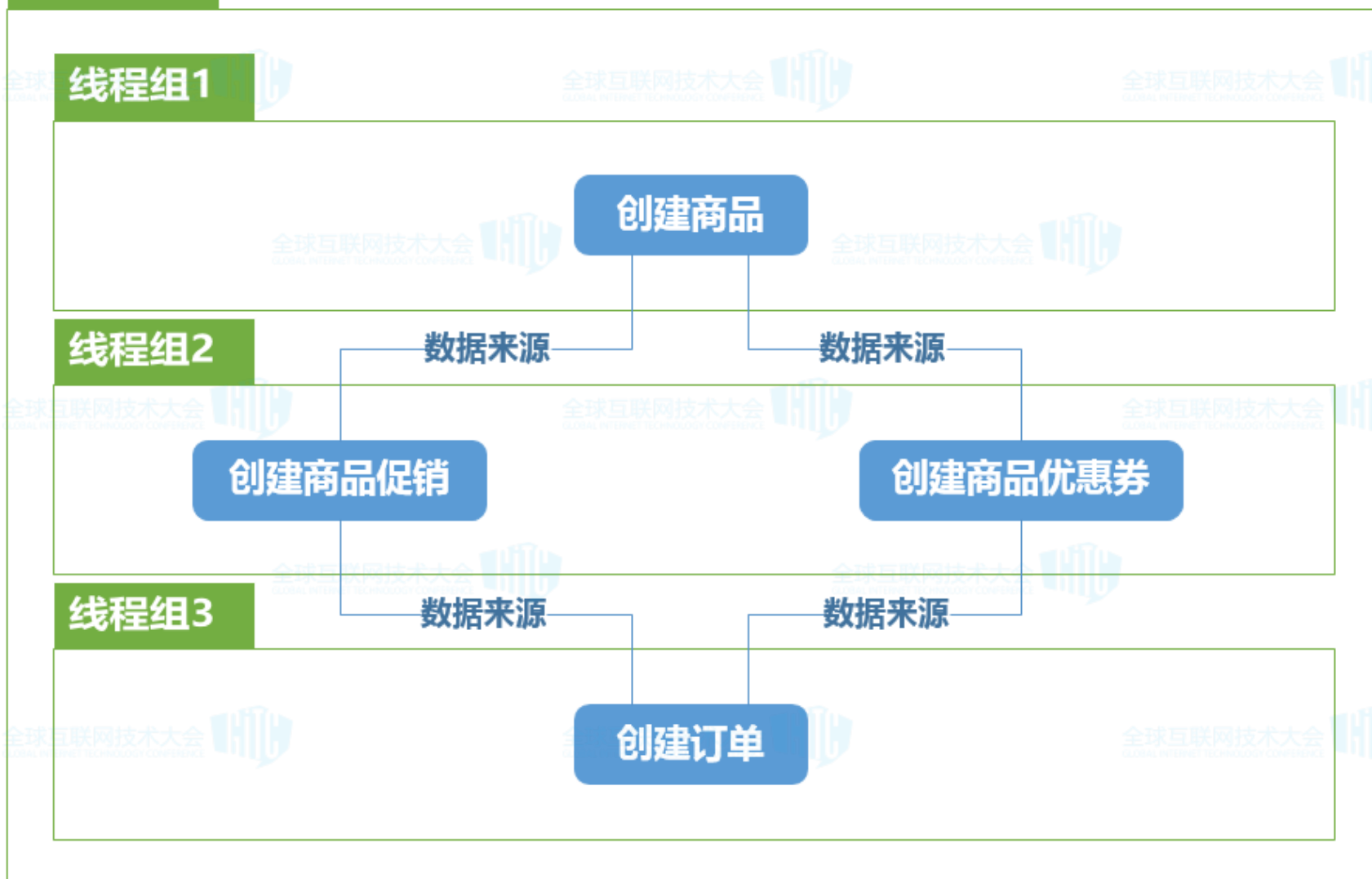
并行处理



线程组终止



测试链路





Thanks



北京市海淀区丹棱街3号中国电子大厦B座10层 100080

10th Floor Tower B, CEC Plaza, No. 3 Dan Ling Street, Hai Dian District, Beijing 100080, China

