

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



# 基于流量回放技术 进行中台建设

taowen@didichuxing.com

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



MORE THAN A JOURNEY

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



滴滴

滴滴一下 美好出行



01 挑战



02 现有方案



03 新方案



04 展望



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



CHAPTER 1

# 挑战

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





**滴滴出行**是全球领先的移动出行平台。公司致力于与不同社群及伙伴协作互补，运用大数据驱动的深度学习技术，解决全球出行、环保、就业挑战；提升用户体验，创造社会价值，建设开放、高效、可持续的移动出行新生态。

**2022愿景：**引领汽车和交通行业变革的世界顶级科技公司——全球最大一站式出行平台；全球最大汽车运营商；全球智能交通技术的引领者。

## 服务

滴滴出行通过一站式的移动应用平台提供多元化的出行服务：



专车



快车



顺风车



出租车



企业级



公交



小巴



代驾



租车



豪华车



共享单车

## 技术优势

每日新增轨迹数据：**70TB+**

每日处理数据：**4500TB+**

每日路径规划请求：**200亿+次**

日均定位数据：**150亿次**

- 需求预测及智能匹配系统，覆盖多人拼车、同城及跨城的复杂需求
- 实时交通导航及线路优化
- 滴滴利用大数据技术协助城市设计高效可持续的交通管控方案

## 我们的成长

**4.4亿+**

注册乘客

**2500万**

日完成订单

**~2108万**

司机获得工作机会

**~7000**

员工（近一半为大数据科学家和工程师）



# 中台化

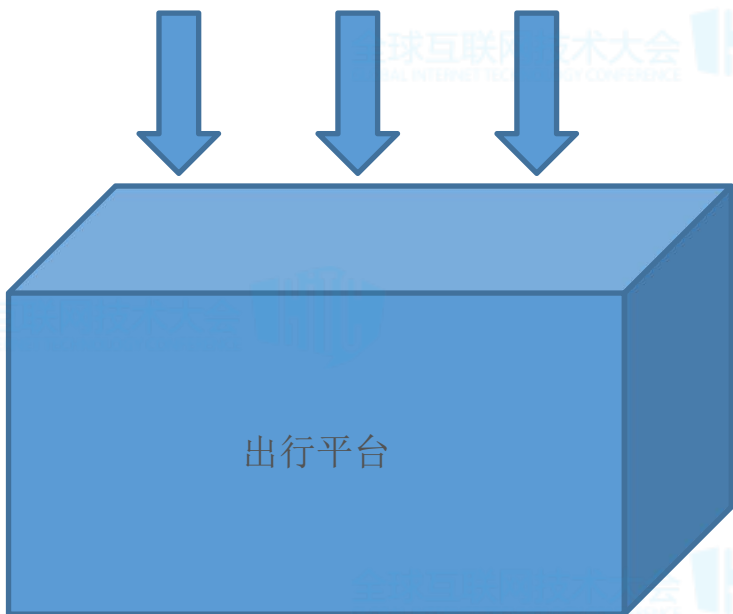
全球互联网技术大会  
GLOBAL INTERNET TECHNOLOGY CONFERENCE



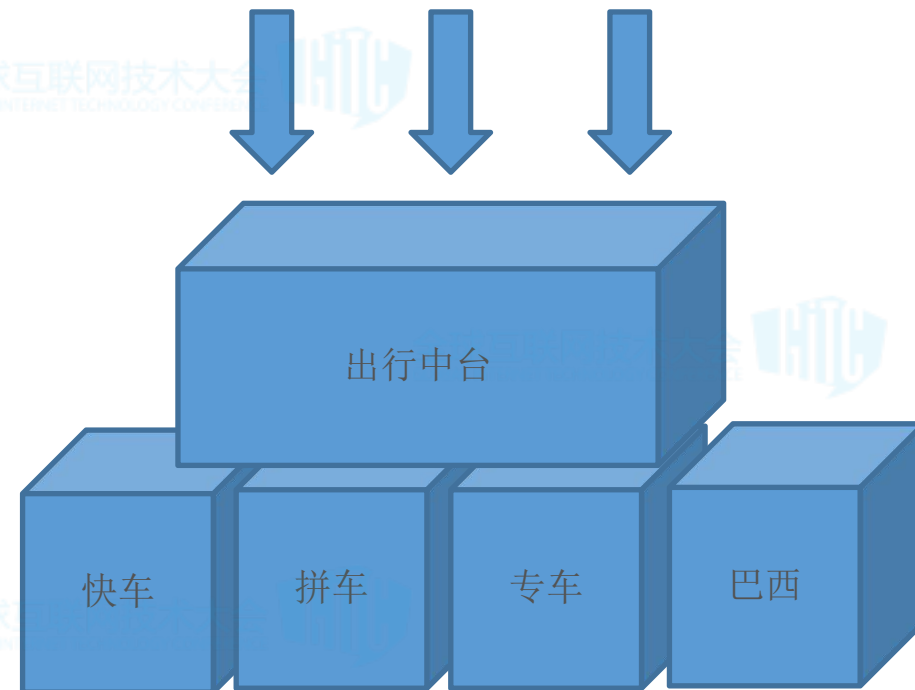
全球互联网技术大会  
GLOBAL INTERNET TECHNOLOGY CONFERENCE



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



十多个大产品  
数百种细分场景  
挂一漏万



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



CHAPTER 2

# 现有方案

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

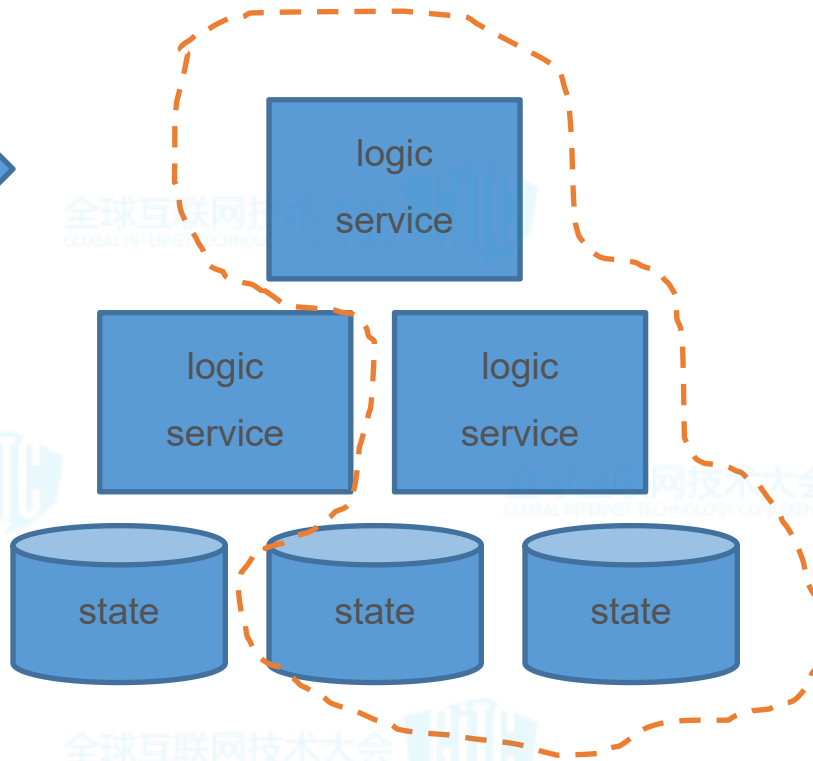


# 手写自动化测试

测试脚本



测试边界划分



边界划分的大

- 1、依赖的state多
  - 2、掉链子的模块多
- 带来问题定位困难

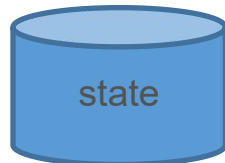
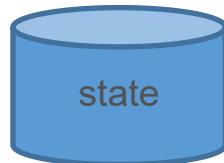
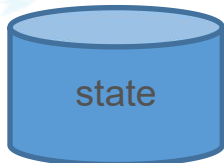
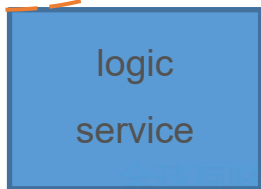
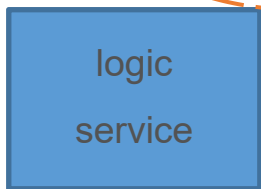
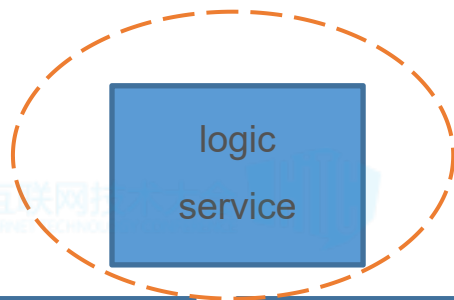
集成的bug多于模块自身的bug



# 手写自动化测试



## 测试边界划分

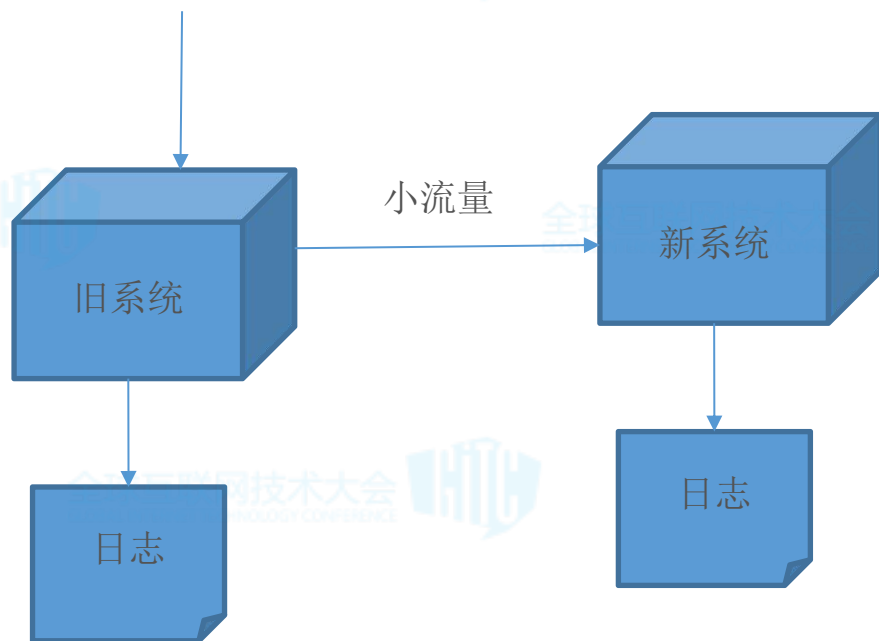


边界划分的小

- 1、一堆依赖的服务需要mock
- 2、内部服务之间的交互不是稳定的系统边界，经常调整，写新功能先要修复一堆挂掉的测试



# 双写双读，日志比对，灰度切流



- 1、流量覆盖不到的，测试也覆盖不到
- 2、切流过程出问题对用户有损
- 3、迭代速度受到上线周期的巨大影响
- 4、双写或者双读比对期间额外消耗资源

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



CHAPTER 3

# 新方案

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



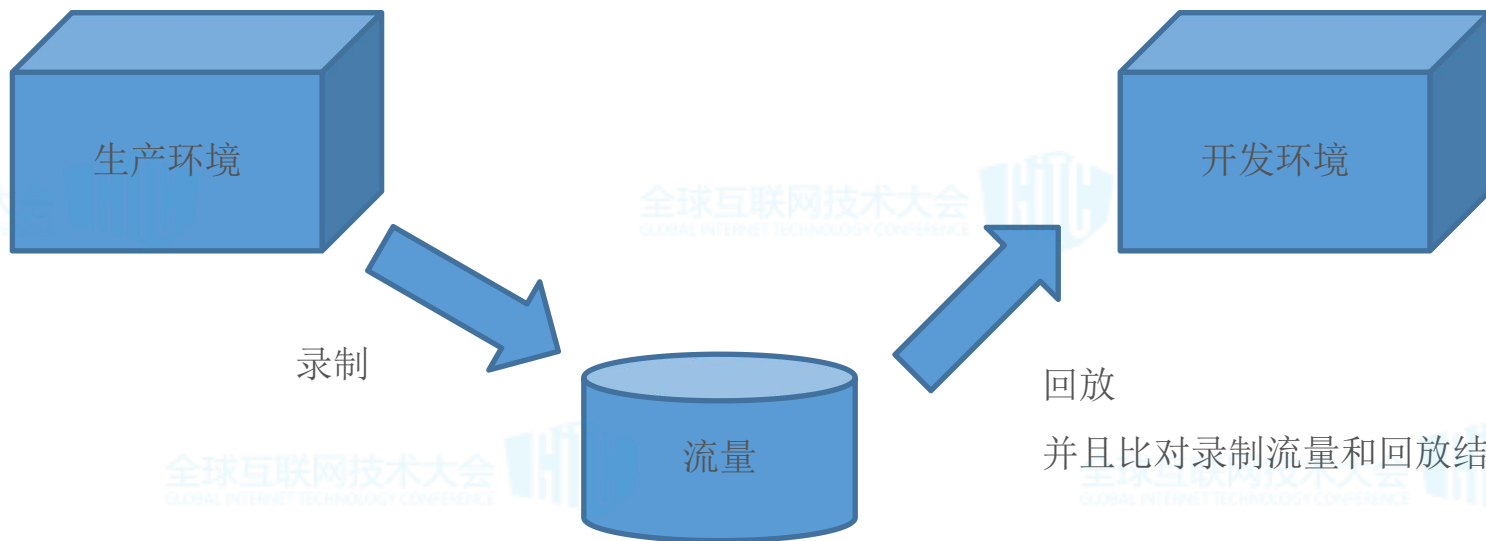
全球互联网技术大会  
GLOBAL INTERNET TECHNOLOGY CONFERENCE



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



# 用流量生成测试用例



并且比对录制流量和回放结果的diff

录制的流量

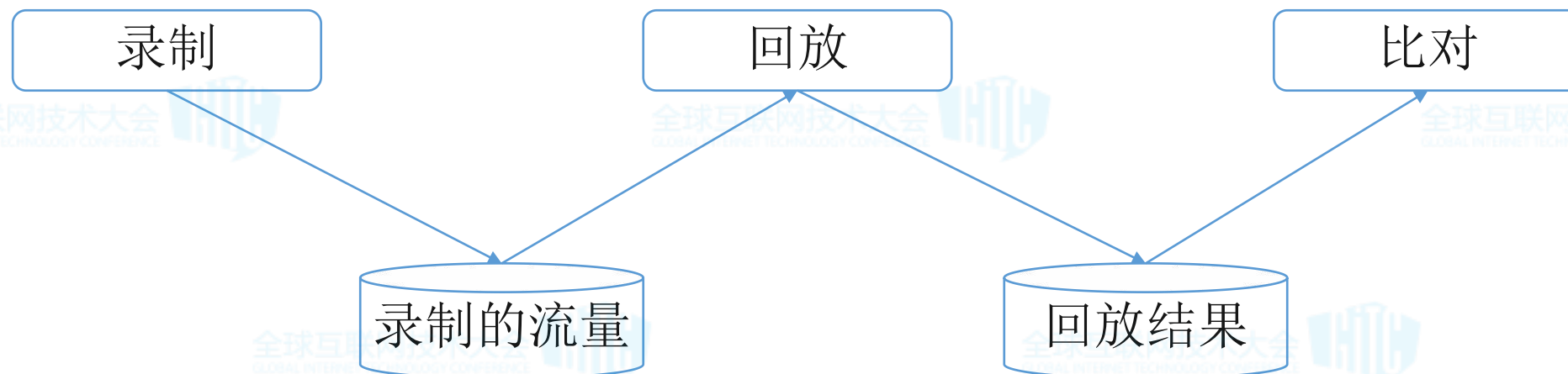
相当于手写的带mock的单元测试脚本

## 为什么流量录制和回放没有普及

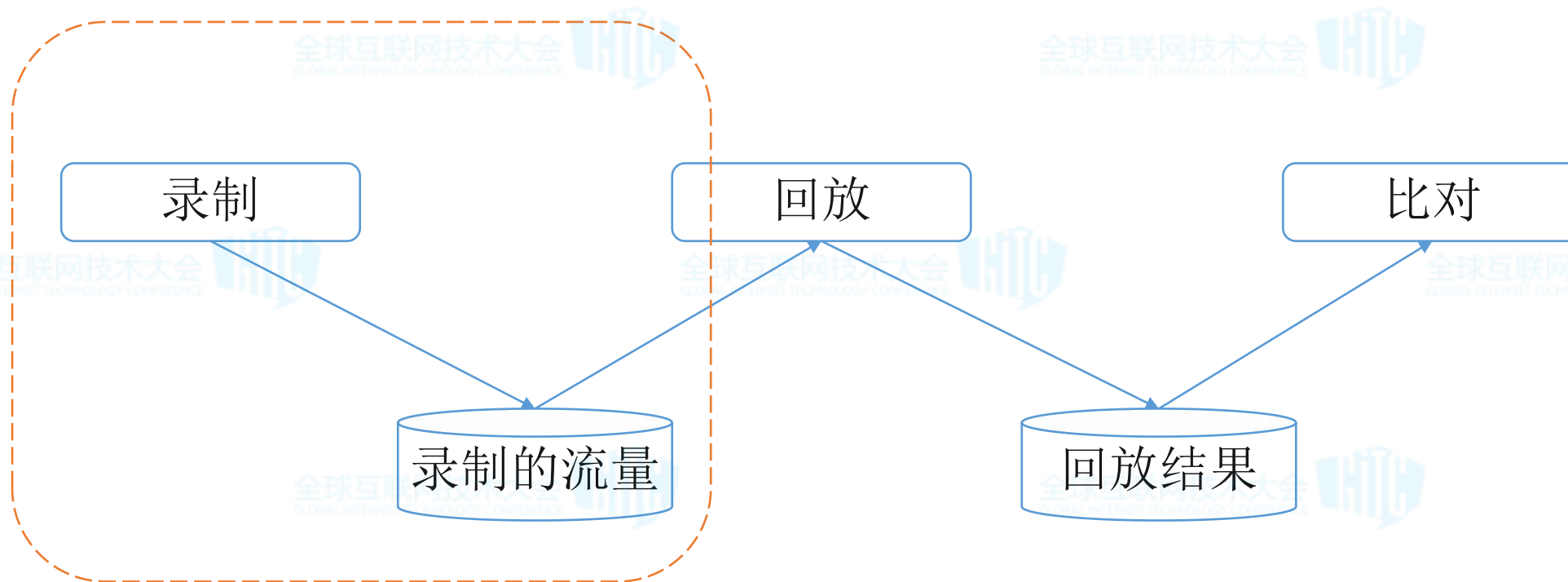
- 插入点怎么搞，是要业务接入特殊的`rpc sdk`么？
- 并行的请求处理过程，如何抽取出一个完整的处理过程？
- 怎么区分外部进来的请求，和访问`rpc`获得的`response`？
- 请求与请求之间怎么切分开来，难道要解析`http`协议么？
- 回放的时候怎么实现`mock`，如何知道这次请求过来返回什么？
- 业务要是和时间相关呢？
- 怎么`diff`回放结果，线下肯定和线上有很多差异啊？
- 对重构可能有一点点用，做新需求开发帮助不大吧？



# 录制回放的整体流程



录制



# 录制的结果



Inbound request

Call outbound: request + response

Append file

Send UDP

Read storage

Inbound response

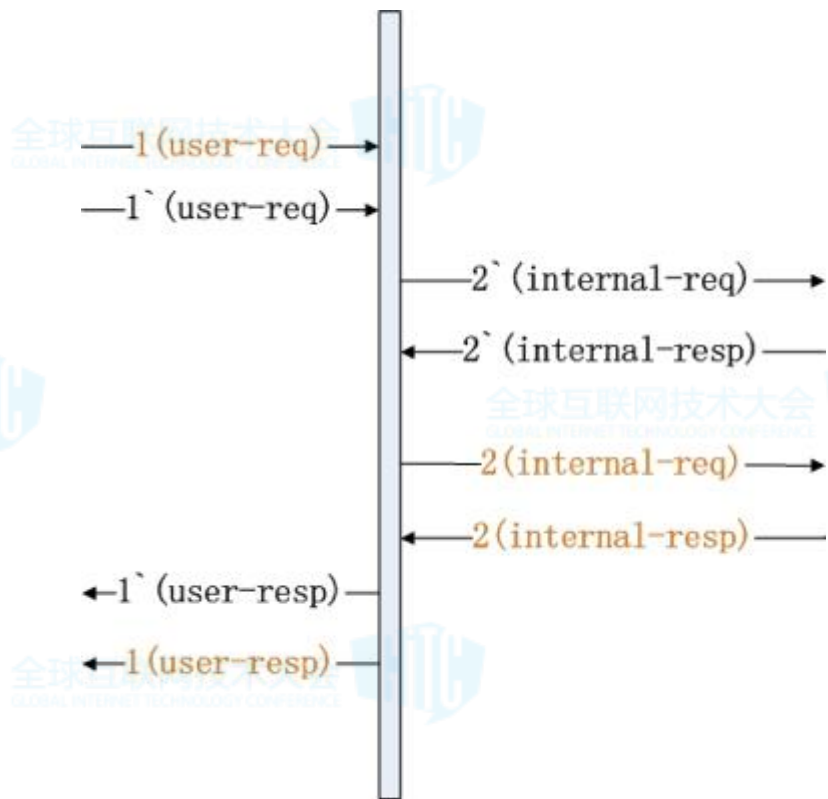
Session

Actions

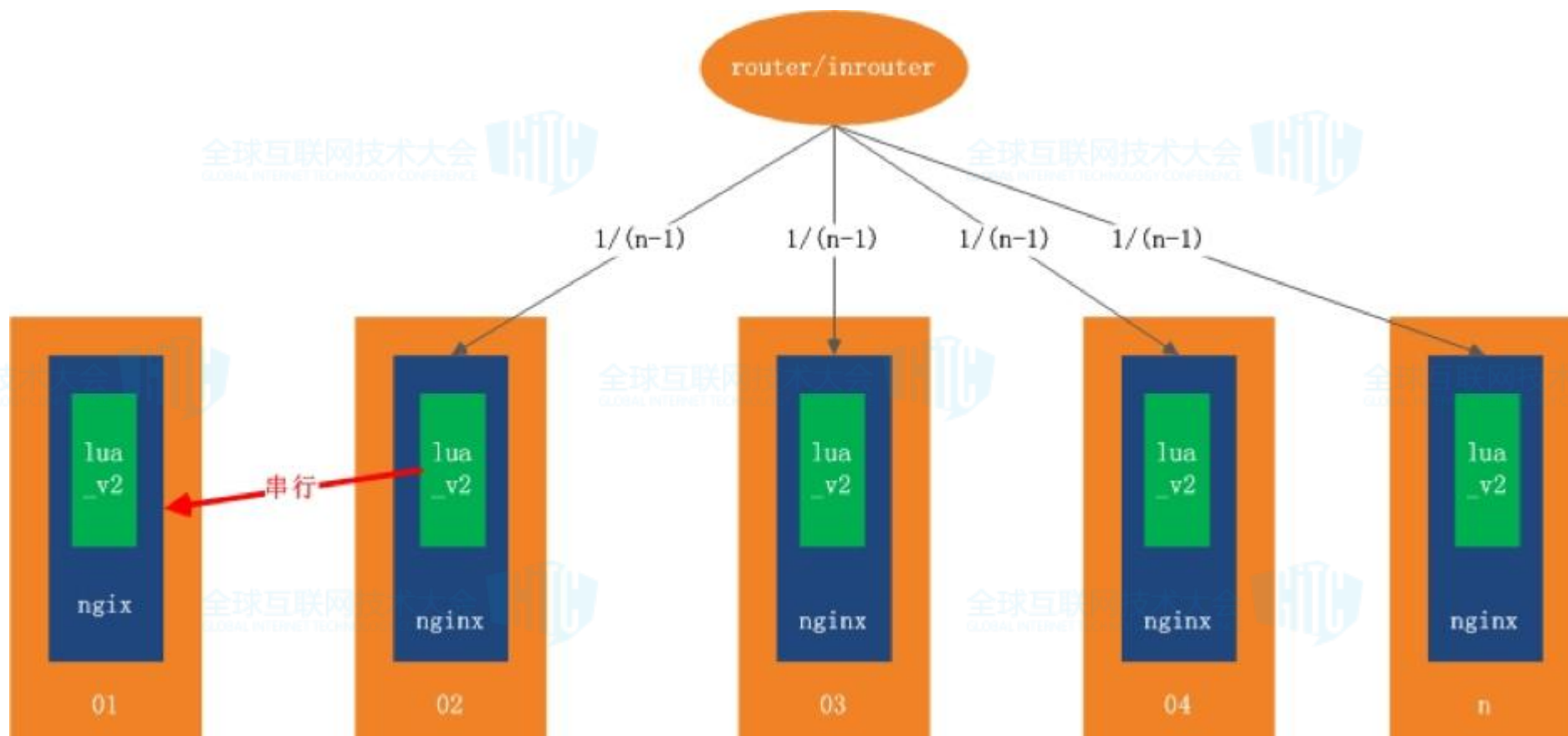




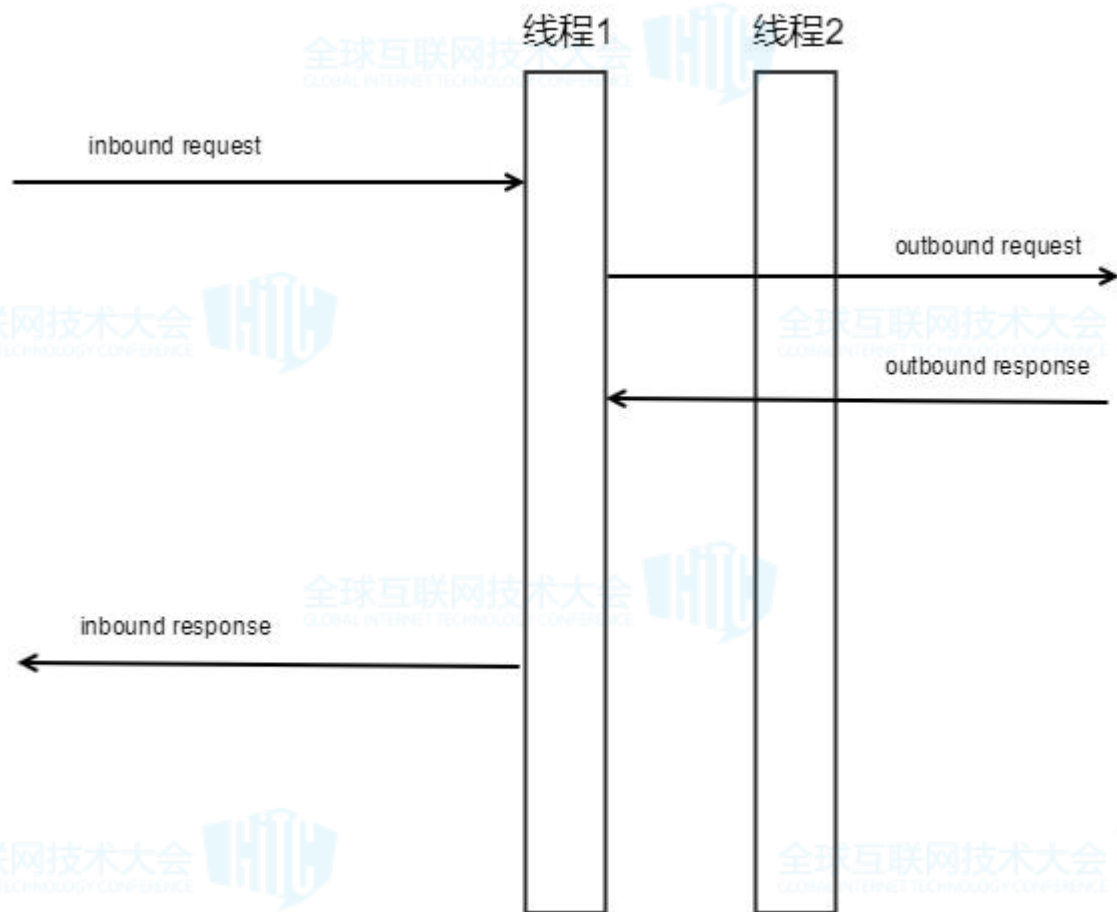
# 录制要解决的问题



# 串行录制

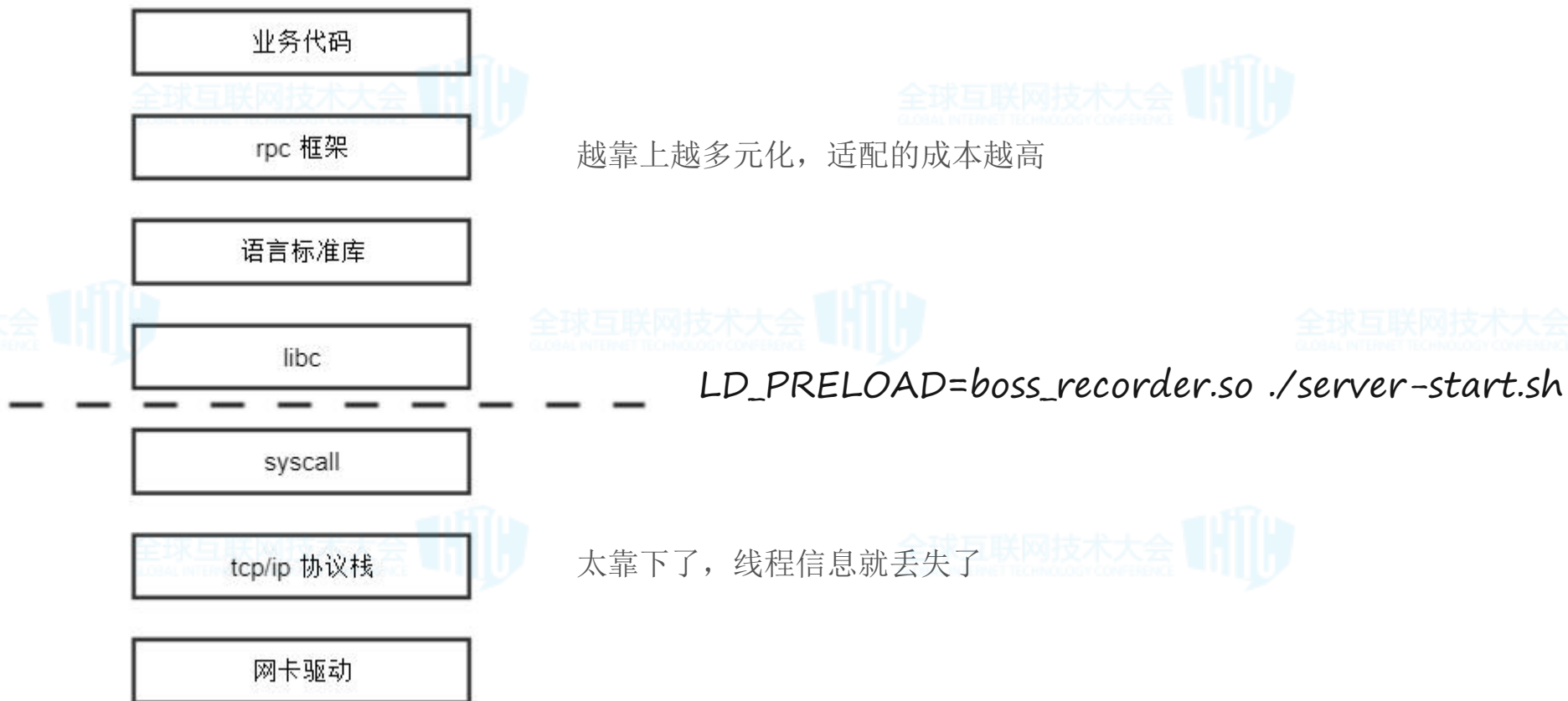


# 并行录制



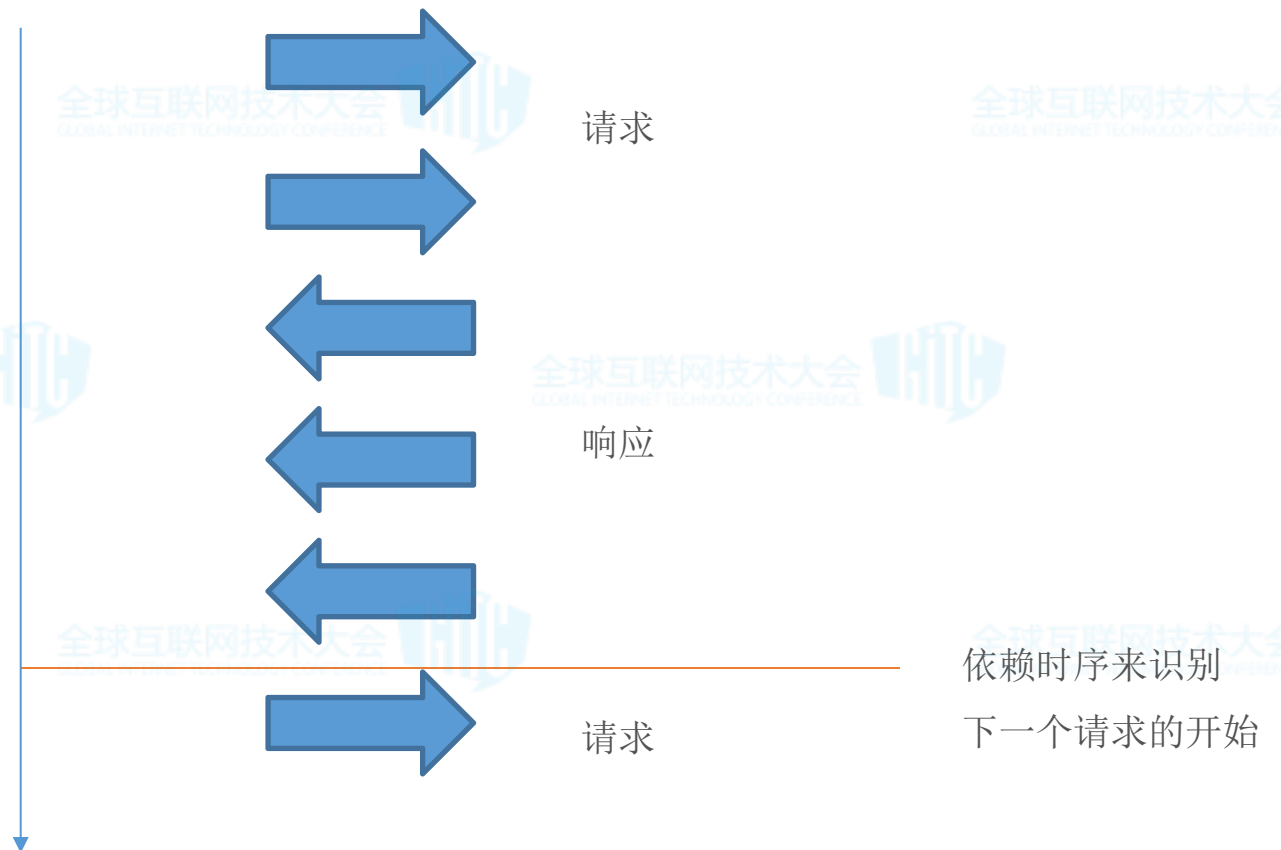
绝大部分业务逻辑在一个线程内串行执行  
通过 *thread* 来关联 *inbound / outbound*  
比手工传递 *trace\_id* 更容易

尽量做到语言无关，框架无关

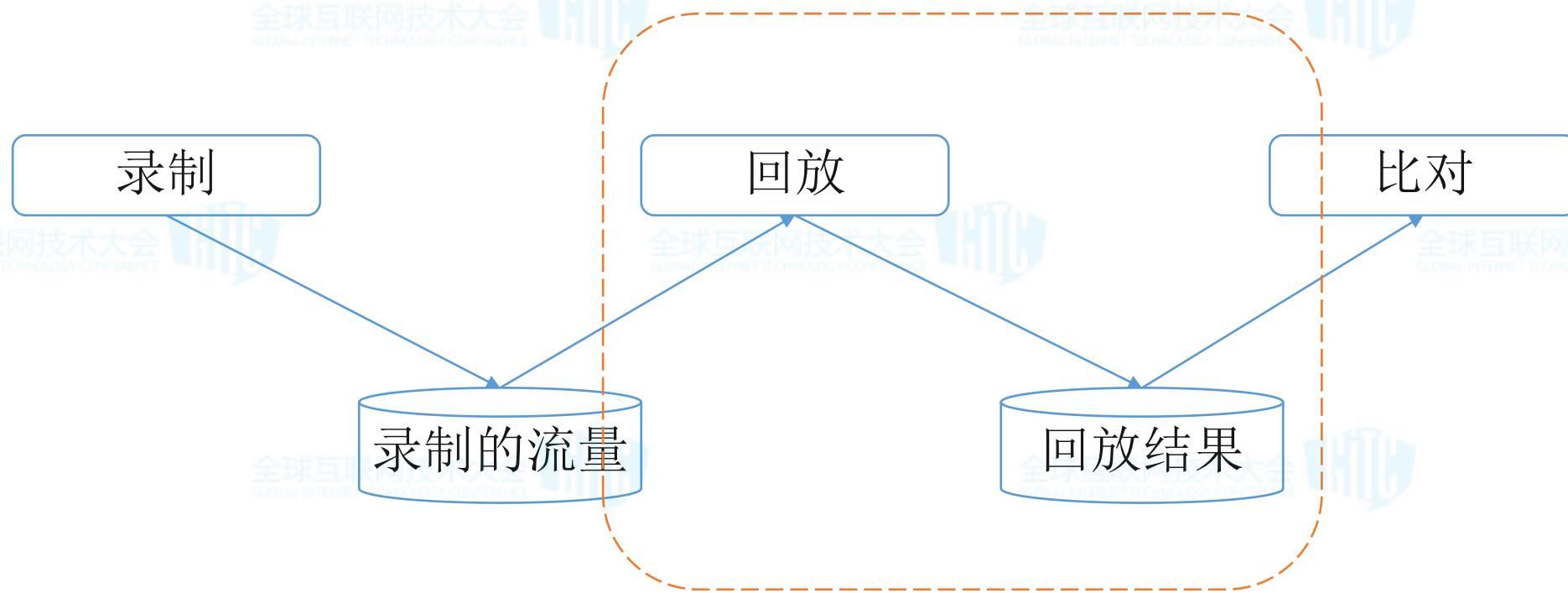


尽量做到协议无关

数据流



回放

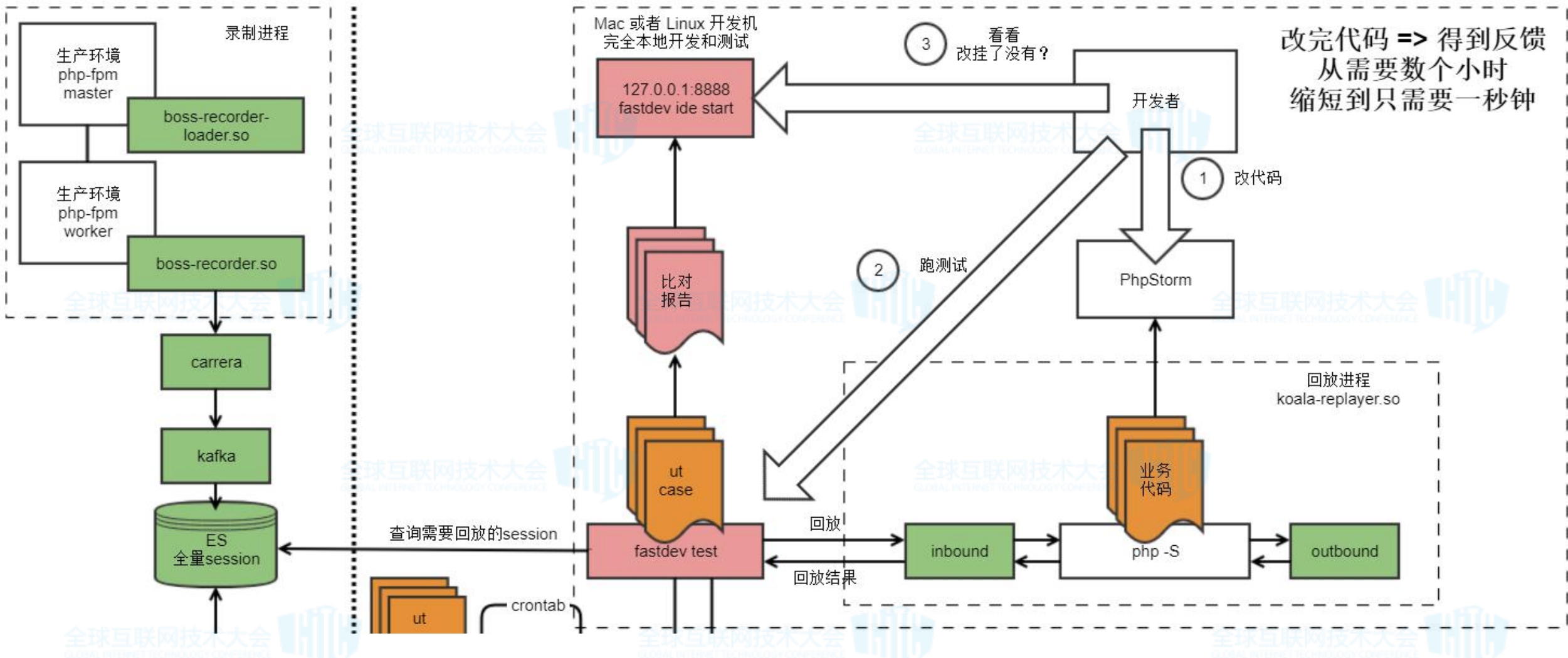


# 开发人员如何使用

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

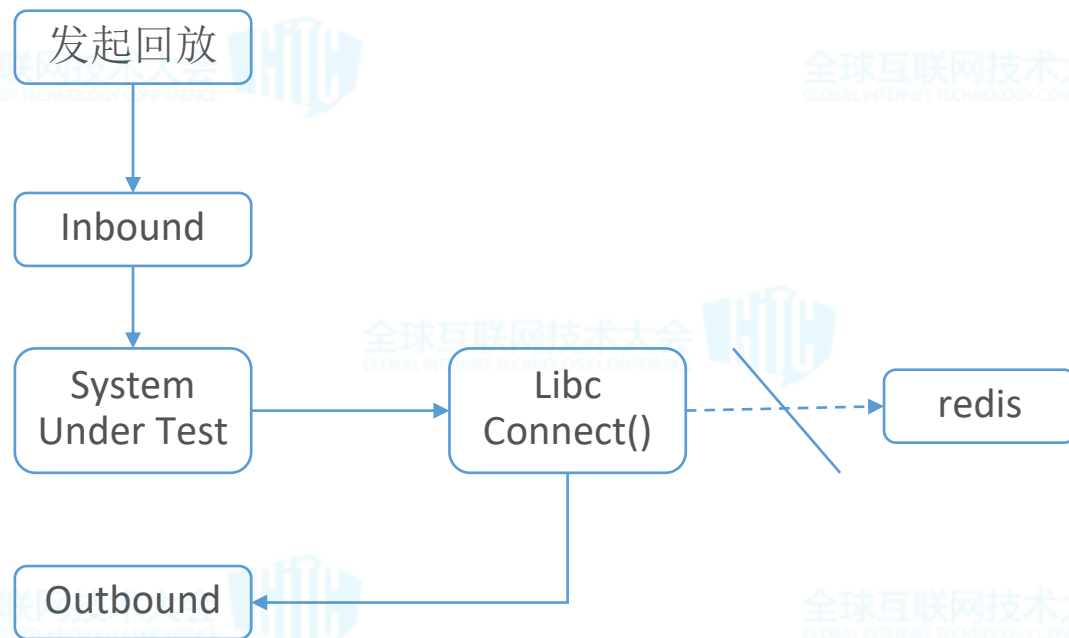


全球互联网技术大会  
GLOBAL INTERNET TECHNOLOGY CONFERENCE

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

# mock重定向





# mock请求匹配实现



Byte数组

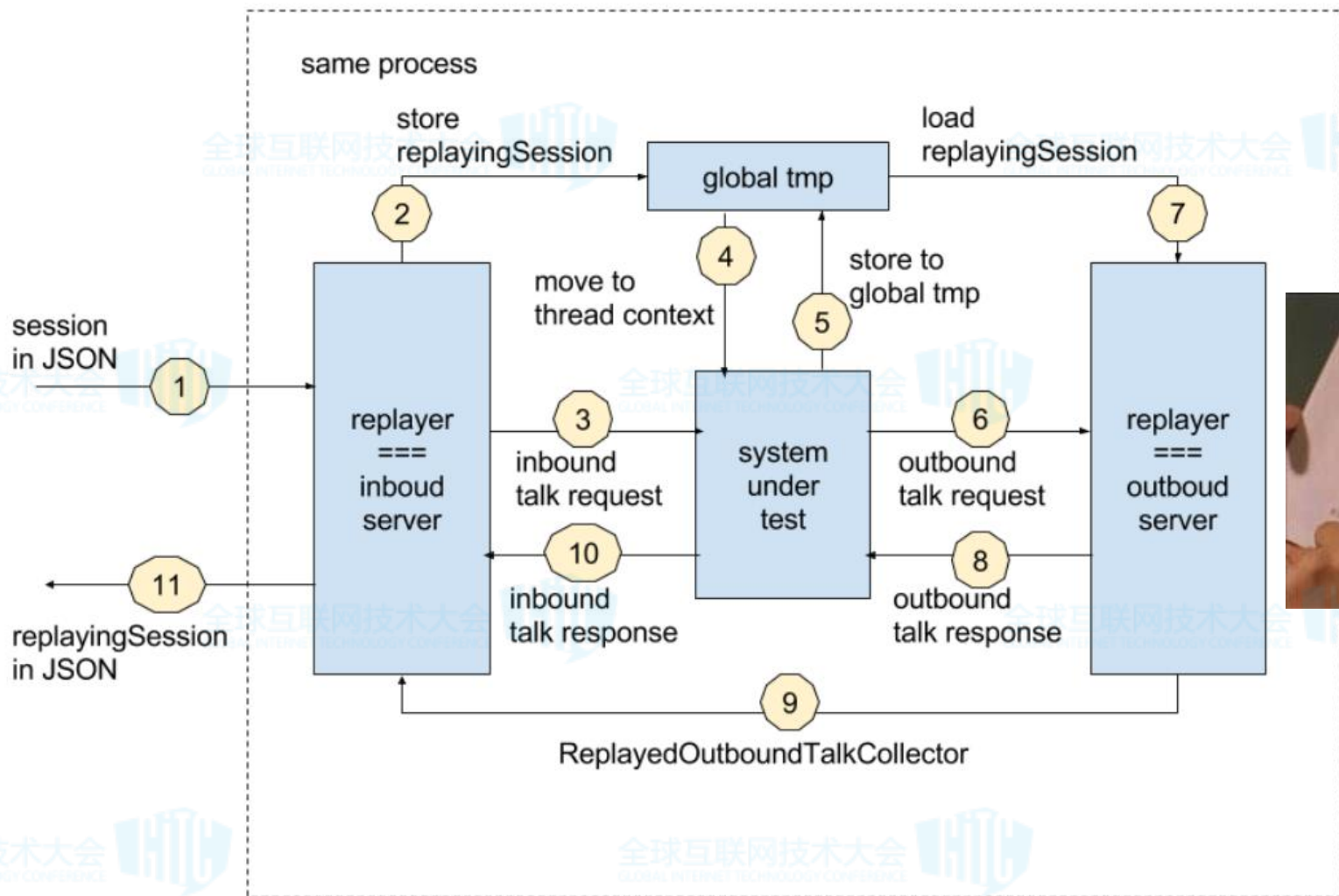
切分成16bytes chunks



按完全相同的chunk数进行打分



# 剧本的传递, 结果的收集



比对



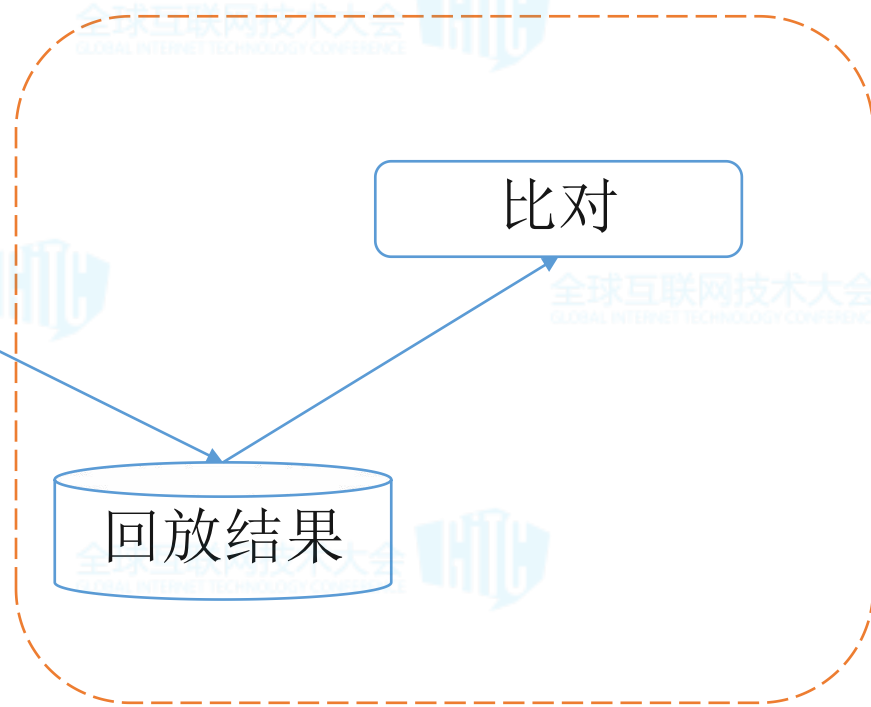
录制

回放

比对

录制的流量

回放结果





O==T

#22 => #89 1.84

/cplat/uranusapi/driver/getDriver

O==T

#23 => #91 1.78

O==T

#25 => #99 1.84

/gulfstream/driver-system/v2/driver/getDriverInfo Driver-Status

O==T

#26 => #126 1.83

/gulfstream/driver-system/v2/driver/getDriverInfo setOnlineStatus

O==T

#27 => #130 1.84

/gulfstream/soter/v1/scanner/getStatusByAction/index

O==T

#30 => #151 1.77

/gulfstream/soter/v1/scanner/getStatusByAction/index

O==T

#33 => #185 1.8

/cplat/uranusapi/driver/setIsOnline

O==T

#35 => #192 1.74

wanliu\_driver

O==T

#40 => #208 1.88

wanliu\_driver\_setonline

Jump to trace

跳转到function trace

O==T Parsed Diff

O==T Binary Diff

O==T Requests

Online Request

Online Response

Testing Request

onlineReq → testingReq RENAMED

@@ -1,5 +1,5 @@			
1	{	1	{
2	"data": {	2	"data": {
3	-    "area": 5	3	+    "area": 0
4	}	4	}
5	}	5	}

O==T

Public Log

g\_setonlinestatus\_record

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



CHAPTER 4

# 展望

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



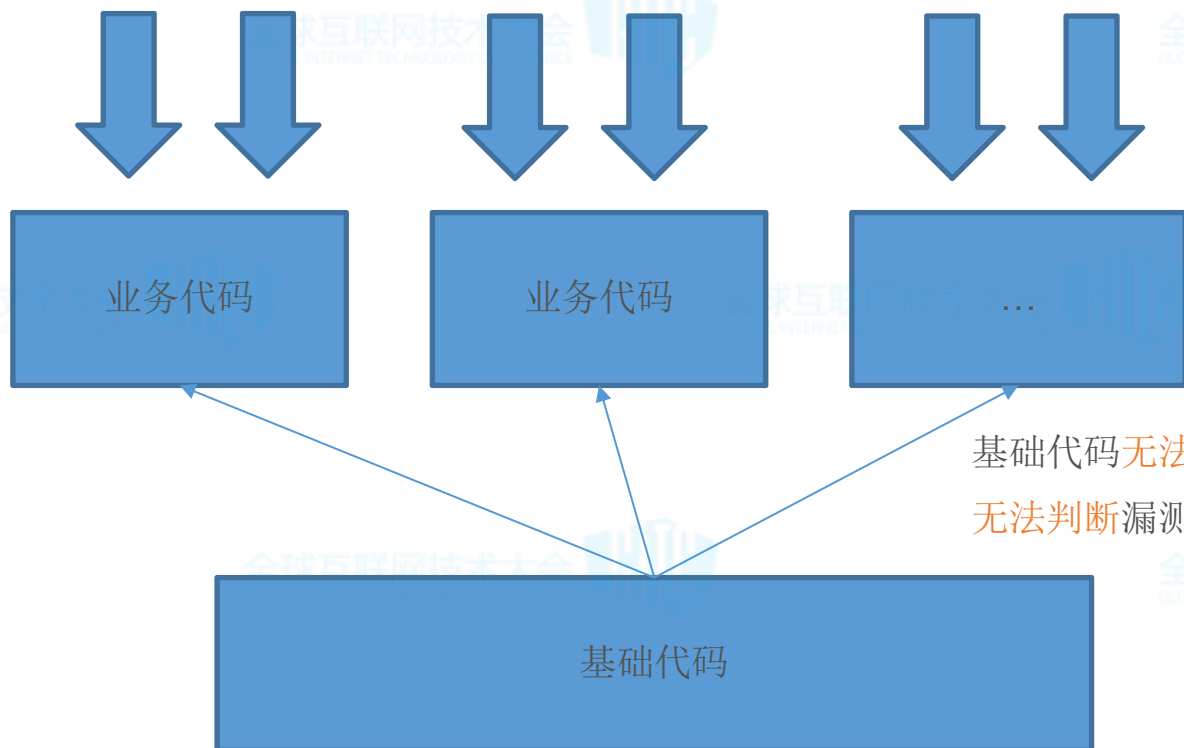
全球互联网技术大会  
GLOBAL INTERNET TECHNOLOGY CONFERENCE



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



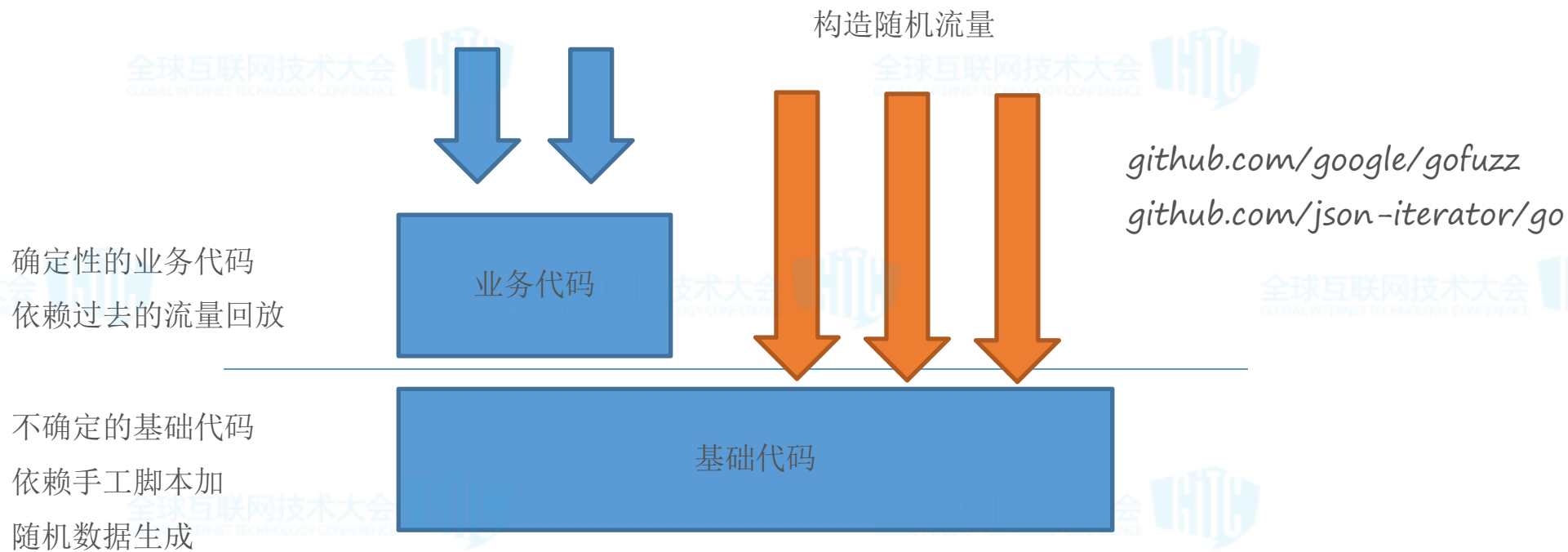
成也流量，败也流量



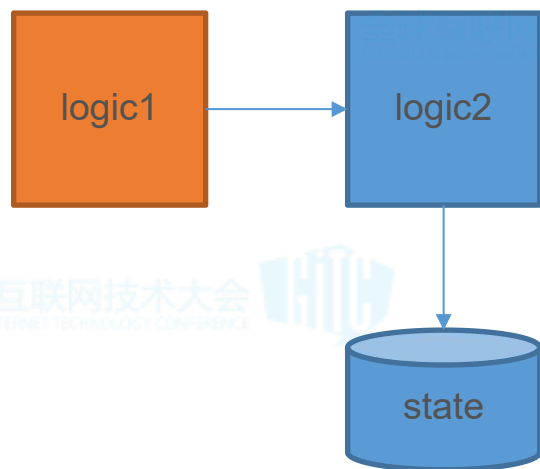
“过去的”流量没有的，就测不到

基础代码无法预测所有的用户，以及上层的行为  
无法判断漏测带来的风险级别

模糊的问题，模糊来解



## 基于mock的测试



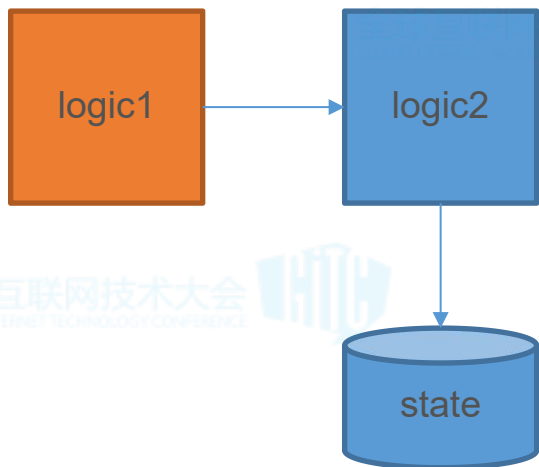
Mock logic2  
测试logic1

两个缺点

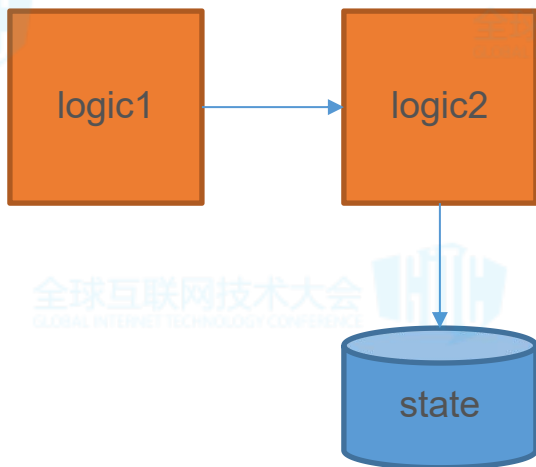
- 1、只能测试一个进程内的逻辑，无法跨进程组合测试
- 2、mock 不如基于 state 的测试真实



# 基于状态的测试

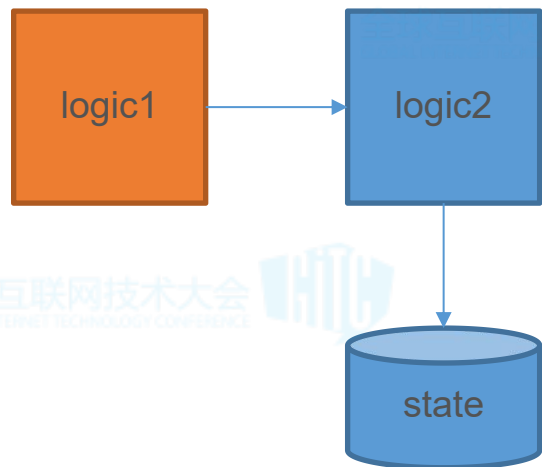


Mock logic2  
测试logic1

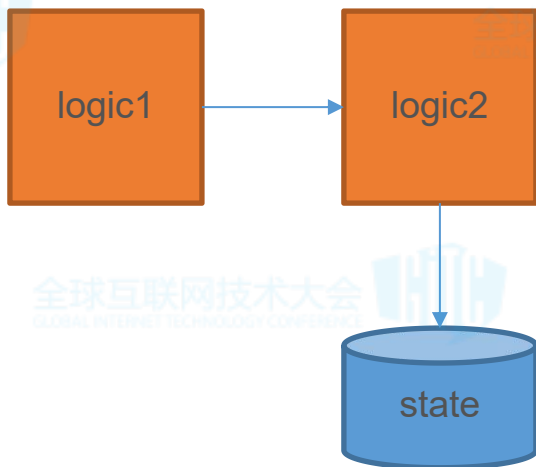


构造state  
测试 logic1 + logic2

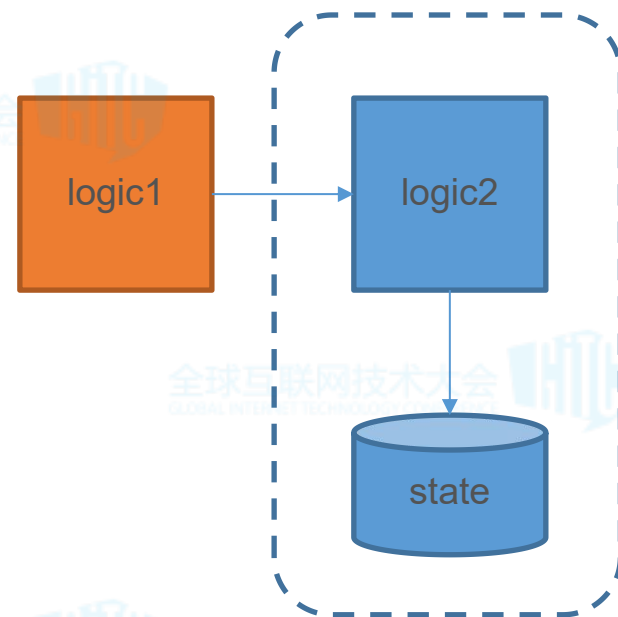
*state* 就是那些“免测”的叶子系统



Mock *logic2*  
测试 *logic1*

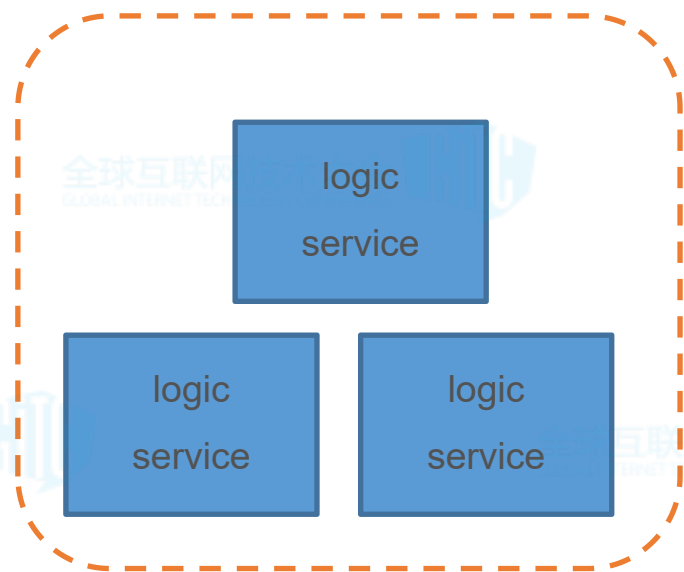


构造 *state*  
测试 *logic1 + logic2*



构造 *logic2+state*  
测试 *logic1*

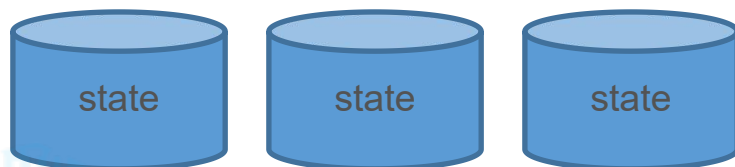
# 集成测试：系统级流量回放



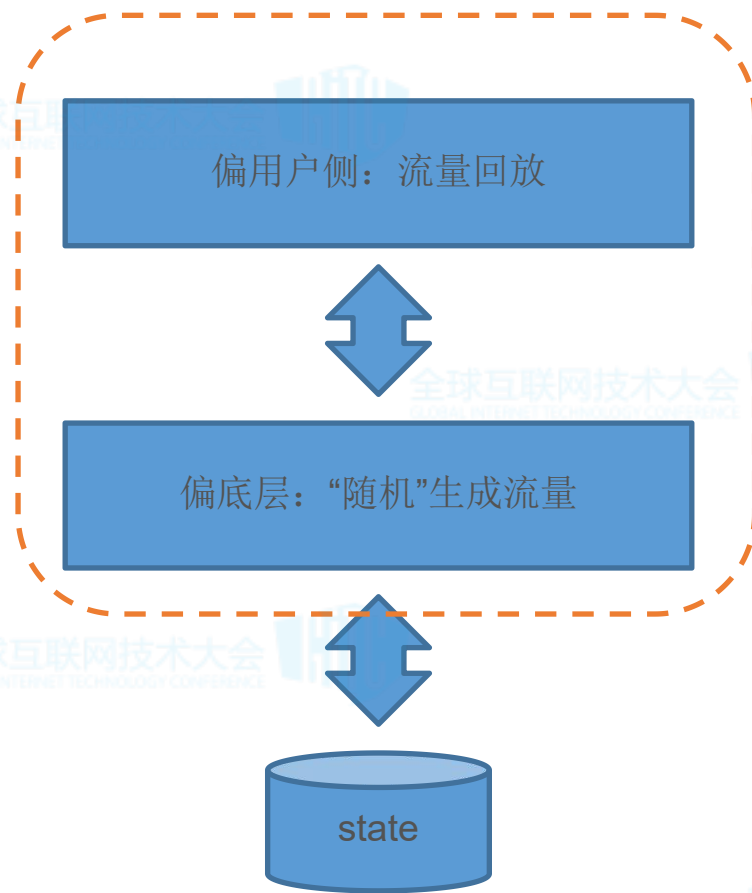
通过服务标准化（包管理+*docker*+服务注册发现）  
解决服务组装问题



对状态的依赖成为最大难点



# 流量录制是未来的发展方向



整体：  
用录制的流量  
构造state  
集成测试逻辑



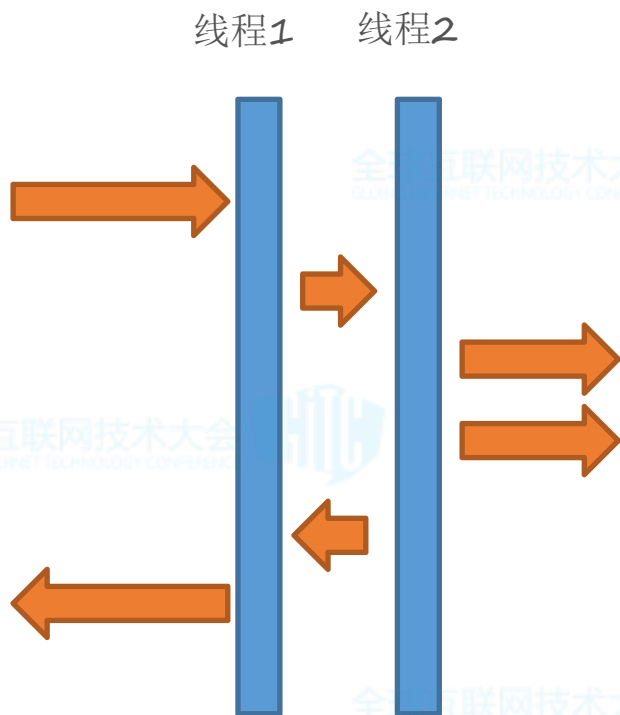
# Q/A



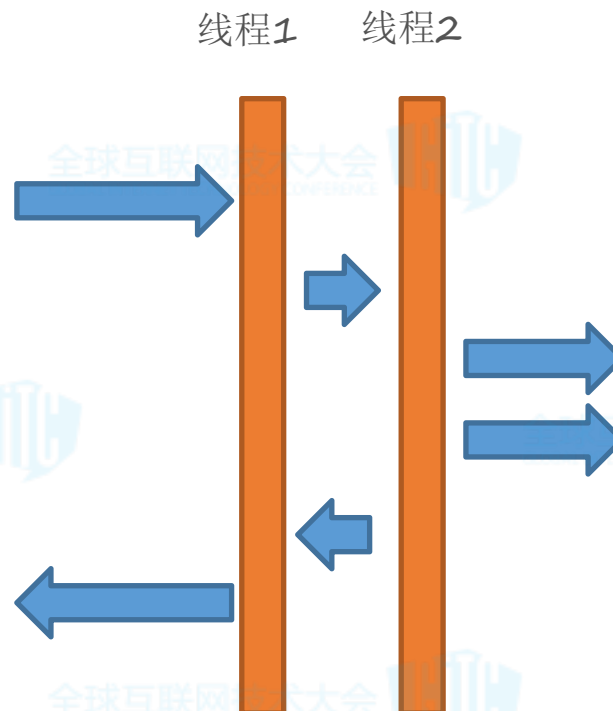
如对滴滴平台技术部的机会感兴趣，请发送简历至我们的招聘HR  
maosanyan@didichuxing.com，或加微信&电话联系(18910686942)。



# thread id v.s. trace id



一个线程下可能发起多次rpc  
基于 *trace\_id* 进行传递  
需要遍历所有的rpc调用处，容易遗漏



关联线程之间的“委托关系”  
需要关注的地方更少