



bilibili



# Bilibili 微服务及高可用实战

自我介绍:

郝冠伟 @冠冠爱看书

现任b站服务端研发经理

曾就职于猎豹移动

游泳爱好者





# Bilibili 微服务及高可用实战

目录:

- 1,大而全的老系统
- 2,第一阶段拆分
- 3,第二阶段挑战
- 4,一起搞基



## 1, 从小众到流行

不知道大家对什么感兴趣？大家都上过B站吗？

## 2, 弹幕文化

视频里出现了一个漂亮的妹纸，他不会说：腿好美。  
他会说：我老婆又出现了。这种感觉真的不一样Y(^o^)

## 3, B站的发展

php 大织梦 -> go bilizone -> go 微服务





# 1 大而全的老系统

php 大织梦 和 go bilizone 的共同痛点

## 1, 代码复杂度高

方法被滥用、超时设置混乱、牵一发而动全身 ==

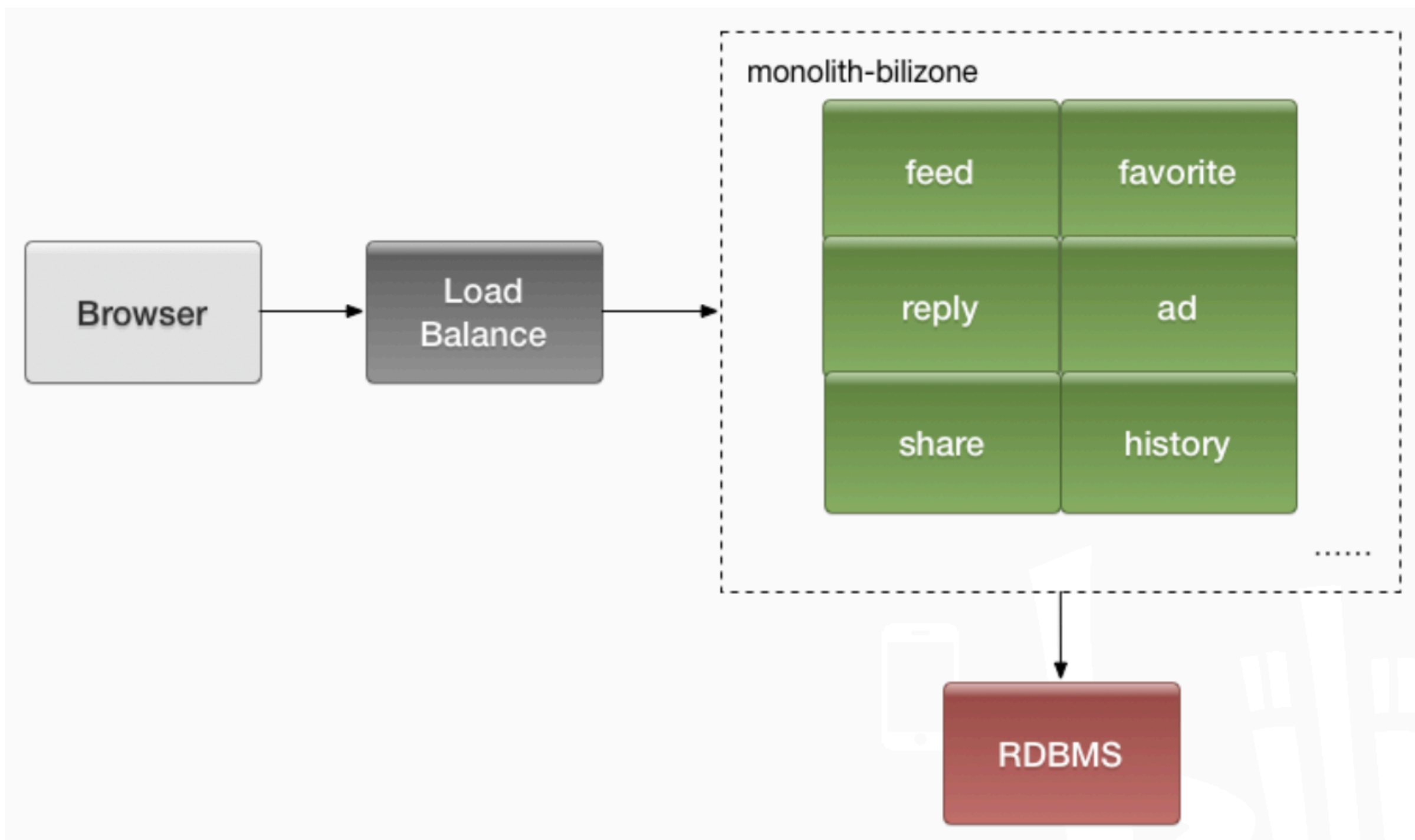
## 2, 一挂全挂

最常见的：超时设置不合理，fpm阻塞/goroutine大量堆积，雪崩

## 3, 测试及维护成本高

小改动都需要测试过所有case，运维发布胆战心惊





## 拆分微服务前的考虑

### 1, 微服务边界怎么定

按模块? 按业务? 按产品? 服务多了依赖关系怎么管理?

### 2, 服务间通讯靠什么

多语言? http? rpc? rpc轮子那么多(;┐┌) !

### 3, 运维成本线性增加

原来就一个服务, 现在NN个, 数量越多运维GG越恨我们/(T o T)/~



## 2 第一阶段拆分

### 总要有开始

#### 1, 梳理现有业务及依赖关系

通过画服务依赖图，如何拆分服务其实已经一目了然

#### 2, 服务本身要有区分

大概分三层：核心、独立业务、接入层

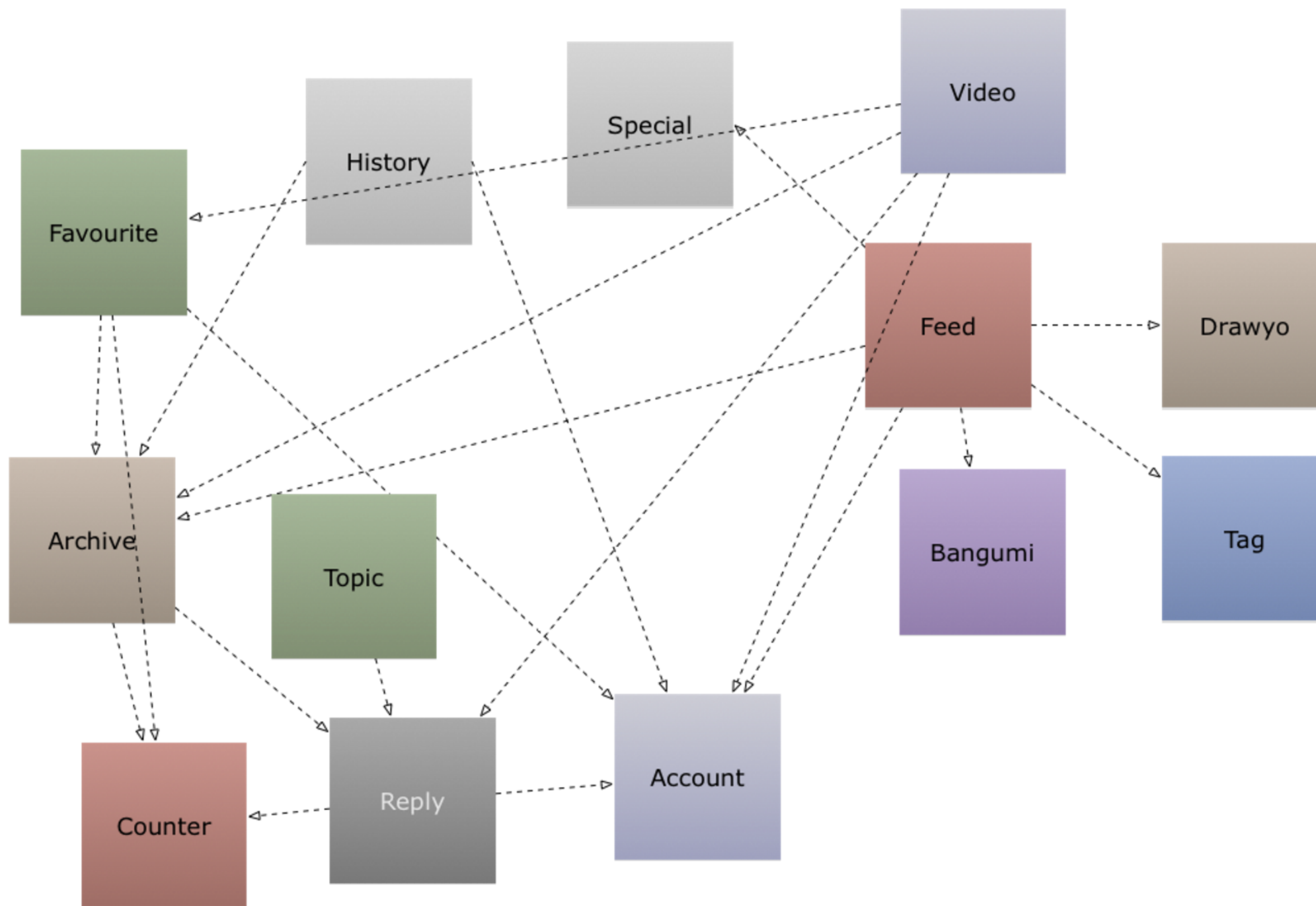
#### 3, 先麻烦运维GG T\_T...T...

拆分初期重点在业务及基础组件积累



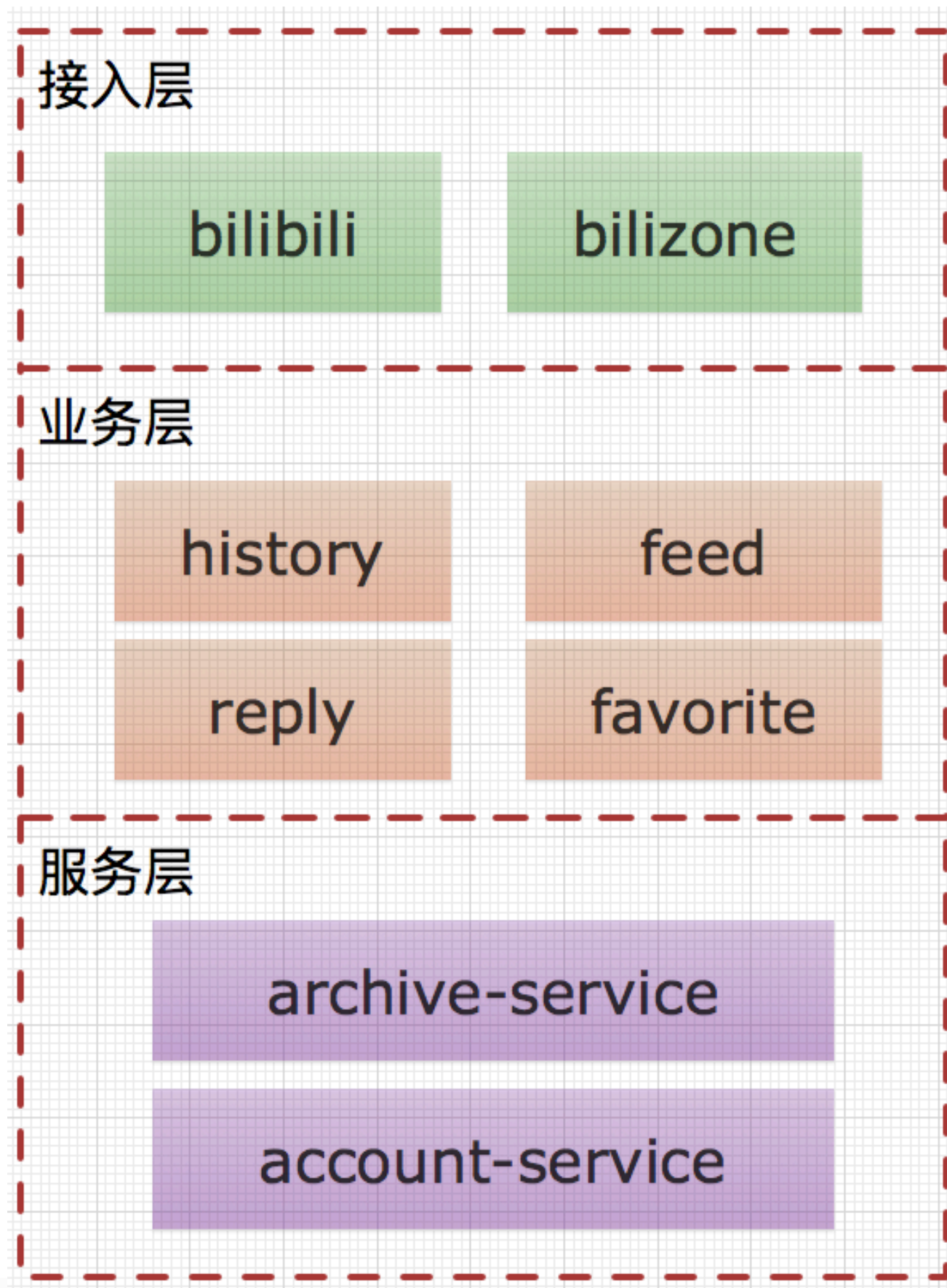
# 2

## 第一阶段拆分



# 2

## 第一阶段拆分



## 2 第一阶段拆分

### 一些积累

#### 1, Go本身服务间使用原生net/rpc

封装net/rpc, 基于zk实现了服务发现及wrr负载均衡

#### 2, 多语言之间使用成熟的nginx+http

提供只对内网服务的http API, 通过成熟的nginx来进行服务治理  
负载均衡、health check、reload conf、降级 ==

#### 3, 单个服务自理

1, 统一的基础库, 包括封装redis/memcache/mysql等基础组件  
还包括日志库、流量统计及error管理等(有好处 (。♥。))

2, 服务除了语法规则外, 架构及实现方式可以多种多样

3, 一个团队完全掌握服务的开发运维, 对外专注提供API

(x..其实目前是一个人负责n个服务...





## 2 第一阶段拆分

### 一些积累

#### 4, 在线监控StatsD+Grafana

微服务的监控系统必不可少，正在尝试Prometheus

#### 5, 异步通讯Databus

基于kafka实现的消息队列，已经服务于微服务间的数据广播、同步等

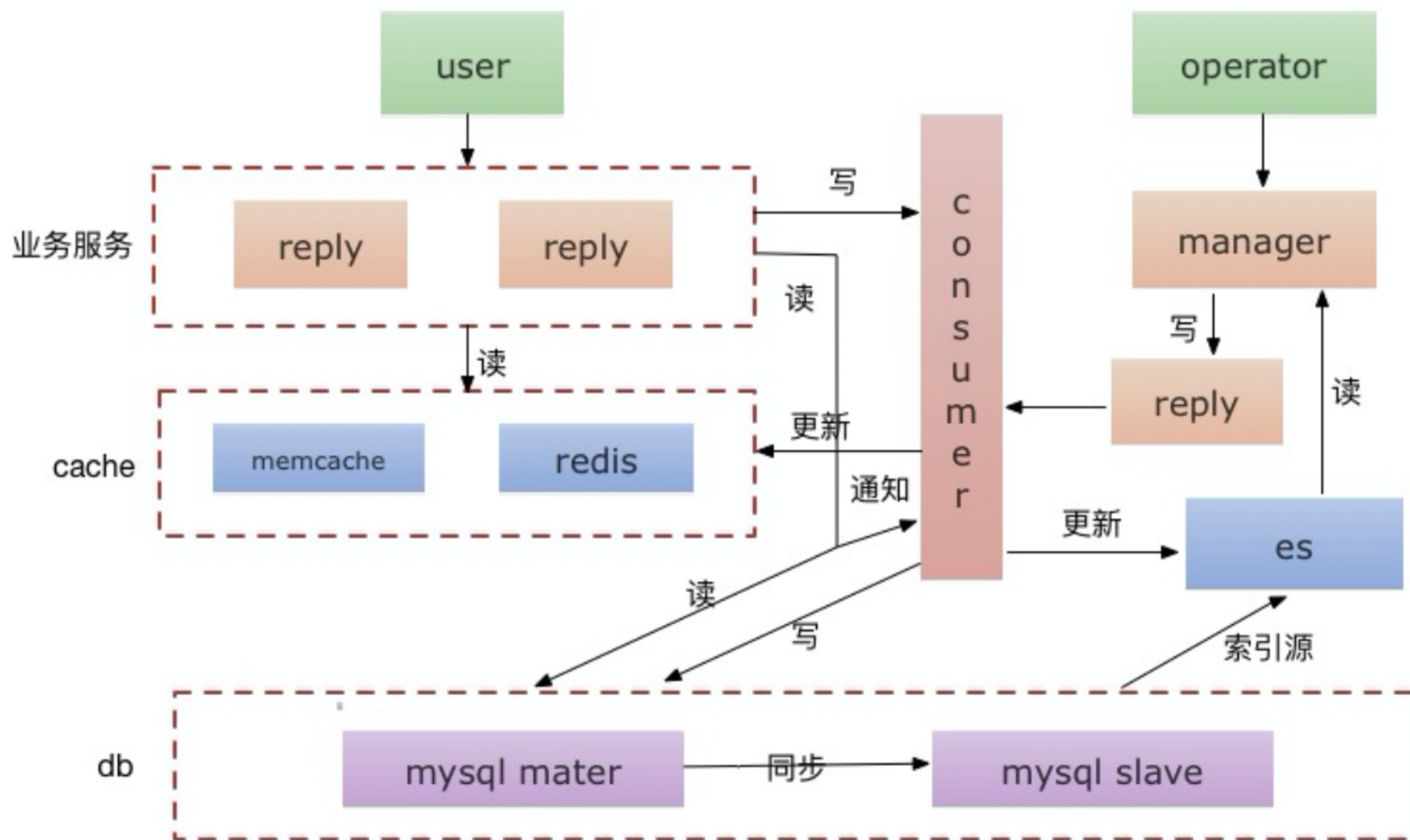
#### 6, MySQL binlog增量订阅Canal

Go实现，和Databus结合，服务于大量核心业务数据实时缓存更新、大数据分析、搜索索引更新等等场景

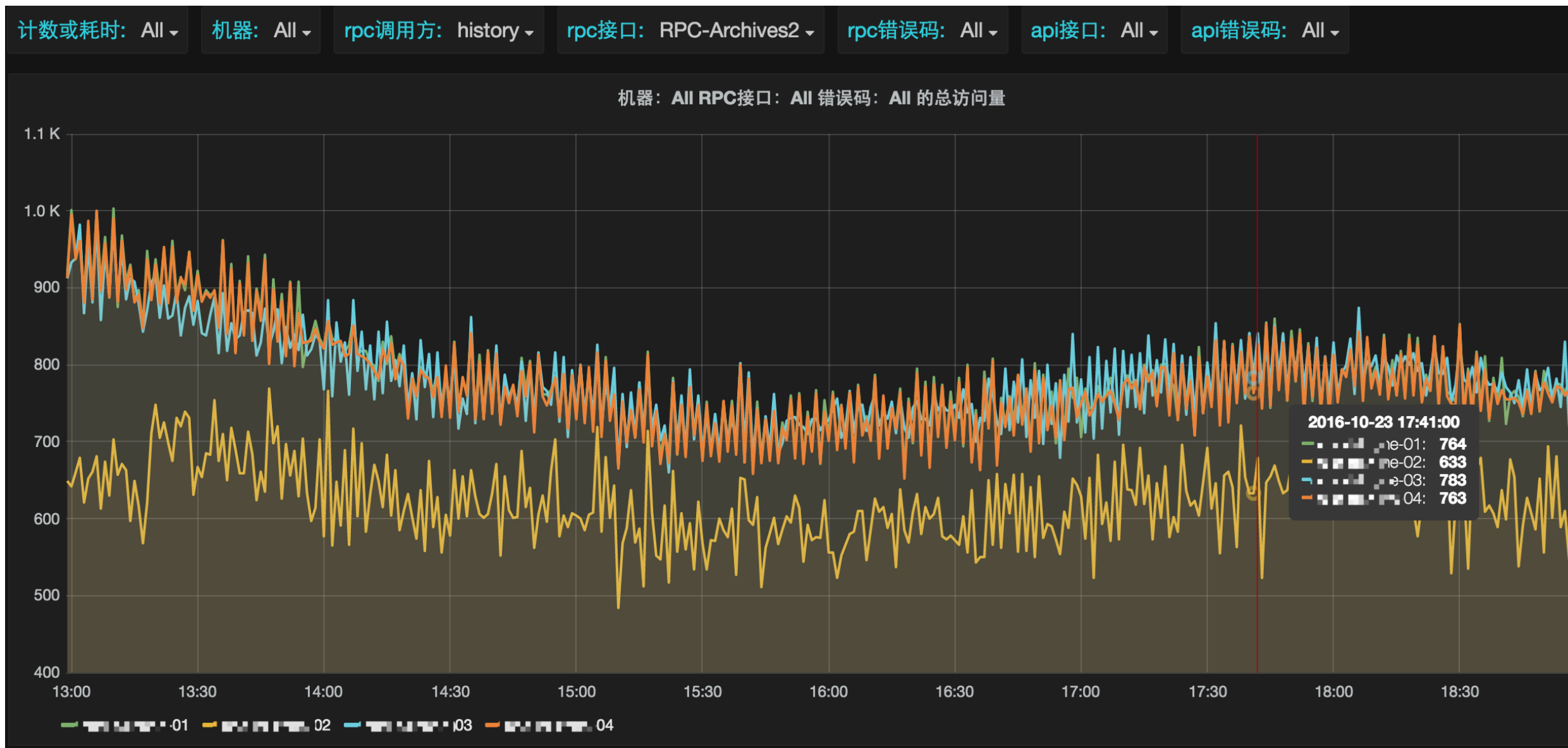


## 2 第一阶段拆分

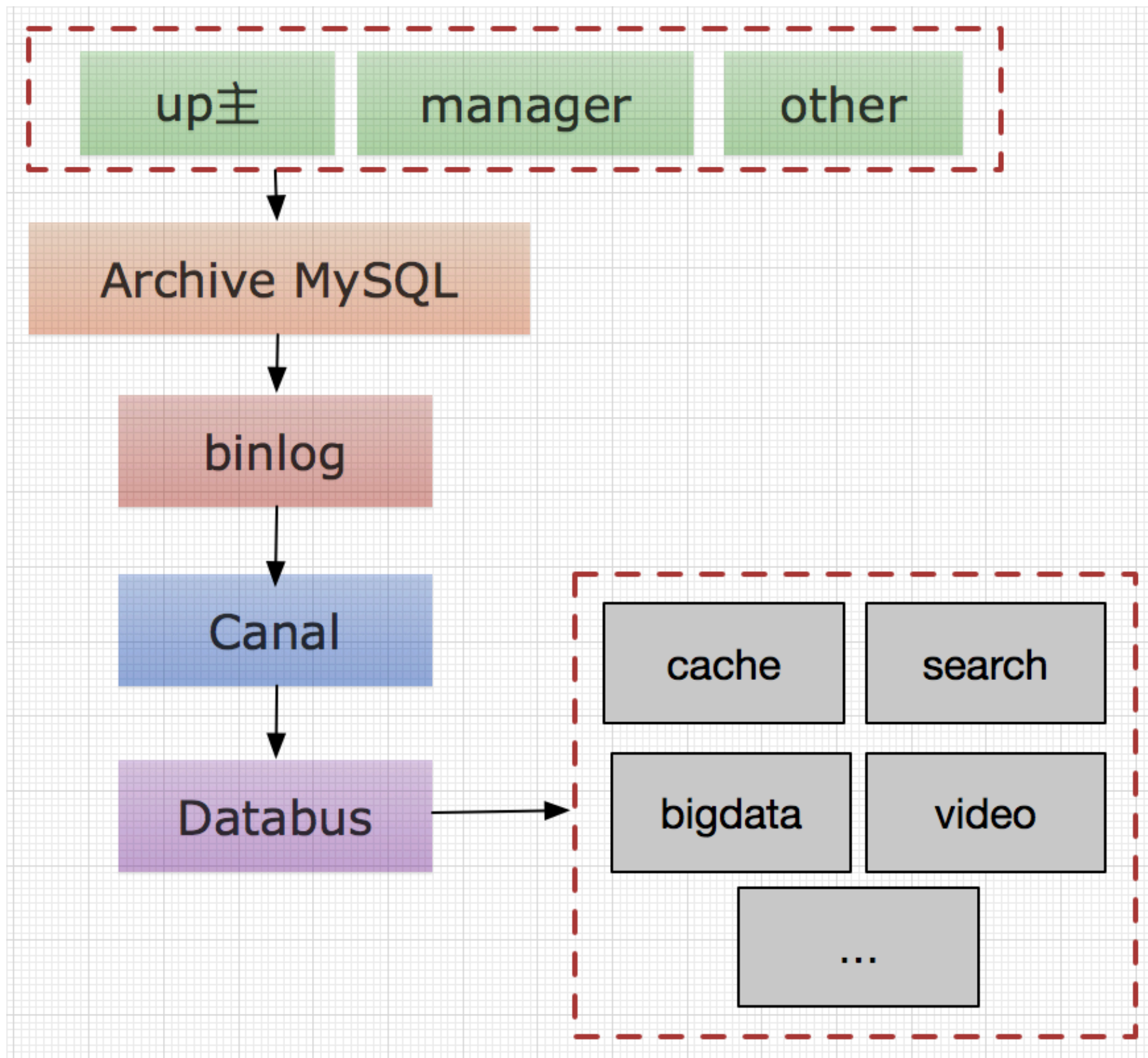
### 评论系统架构图



# 2 第一阶段拆分



## 2 第一阶段拆分



## 第一阶段效果

### 1, 不再一炸全炸

15年可以说是b站天天炸的飞起的阶段，每到周末总会有一股风吹来b站炸了么。。。

### 2, 开发测试效率提升

开发童鞋只需要熟悉我们封装非常简单的基础库  
精力就可以完全放在业务代码的实现及优化上

(从目前一个人负责至少3个服务就看的出来...)

测试童鞋也按服务分担，只需要保证API的兼容性



### 3 第二阶段挑战

## 带来的新问题

### 1, 难以追踪问题

一张图说明任何问题。。。



### 2, 运维GG来打我们了/(T o T)/~~



## 针对追踪问题

### 1, 无所不在

跟踪所有服务的调用情况，包括：

- 1, RPC请求
- 2, HTTP请求
- 3, SQL查询
- 4, Cache请求
- 5, 内部方法或服务

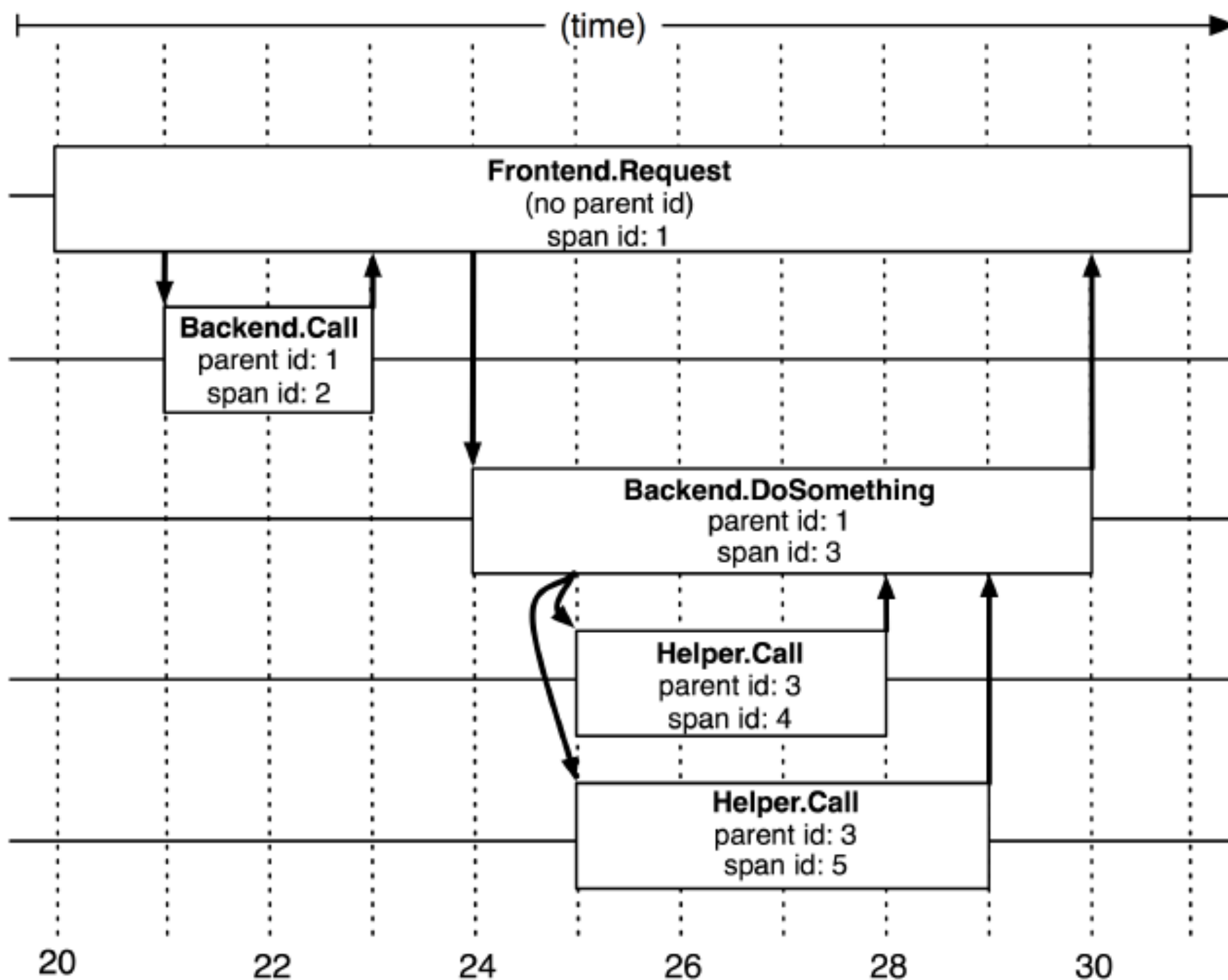
### 2, 持续的监控

很多问题比如网络抖动，都是短时间的影响  
所以必须做到时刻都有监控数据





## 基于Google Dapper论文



<http://static.googleusercontent.com/media/research.google.com/zh-CN/pubs/archive/36356.pdf>



### 3 第二阶段挑战

## 确定数据格式及收集

开始数据：

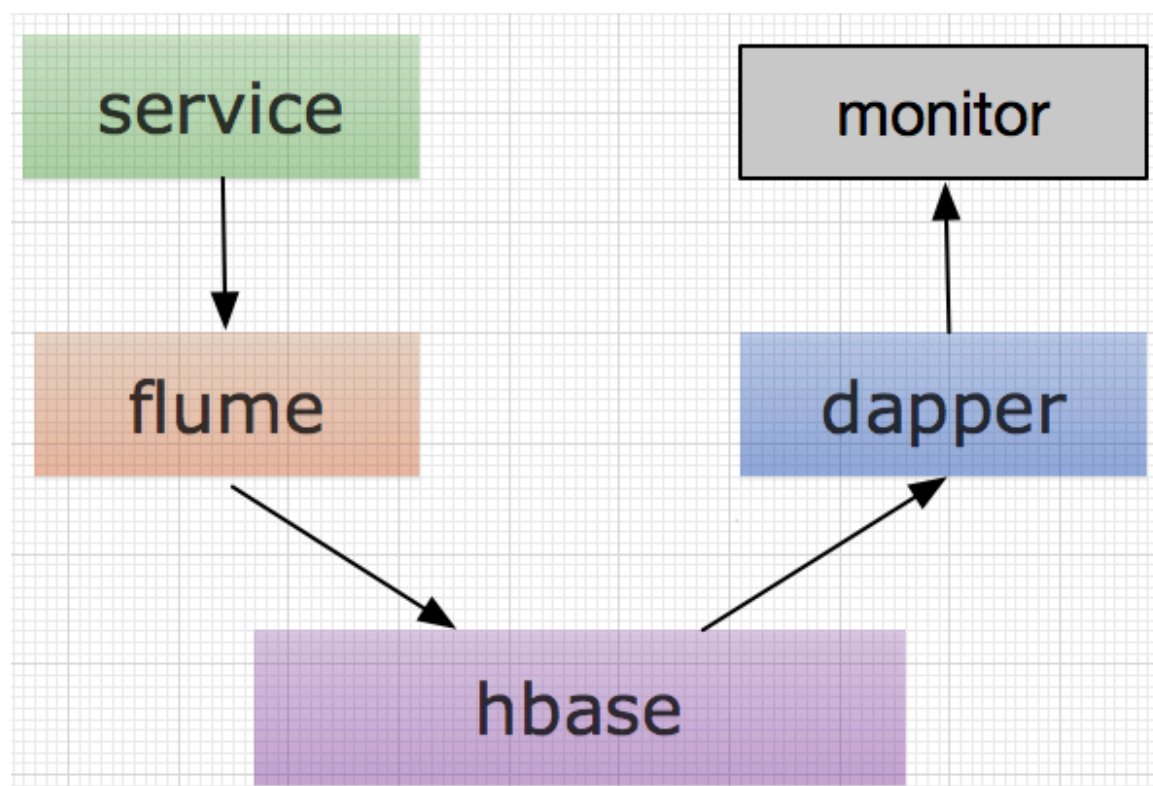
protocol,time,id,spanned,parented,event,level,class,address,family,title,comment

结束数据：

protocol,time,id,spanned,parented,event,level

Annotation:

protocol,time,id,spanned,parented,event,level,comment



## 封装Trace基础结构

```
// Trace2 is server and client called trace info.
type Trace2 struct {
    ID          uint64
    SpanID      uint64
    ParentID    uint64
    Level       int32
    Sampled     bool

    address      string // ip:port address
    family       string // project name, service name
    title        string // method name, rpc name
    class        int8   // 0: component, 1: service
    comment      string // string comment
    binaryComment []byte // binary comment
    commentType  int8   // 0: string, 1: binary
    event        int8   // trace event
    startEvent   int8   // trace temp event
    time         int64  // trace timestamp
    annotation   int    // annotation times
    // status
    start bool
    done  bool
    // next Trace2
    next *Trace2
}
```



## 确定植入点，封装通用组件

```
// Get is a shortcut for router.Handle("GET", path, handle)
func (r *Router) Get(pattern string, handlers ...Handler) {
    r.mux.HandleFunc(r.join(pattern), func(w http.ResponseWriter, req *http.Request) {
        handler("GET", r.family, r.address, w, req, handlers)
    })
    return
}

func handler(method, family, address string, w http.ResponseWriter, r *http.Request, handlers []Handler) {
    t := trace.WithHTTP2(r, family, r.Host+r.URL.Path, address, _class)
    t.Server(method)
    defer t.Finish()
    if r.Method != method {
        http.Error(w, http.StatusText(http.StatusMethodNotAllowed), http.StatusMethodNotAllowed)
        return
    }
    c := context.NewContext(trace.NewContext2(ctx.Background(), t), r, w)
    for _, h := range handlers {
        h.ServeHTTP(c)
        if err := c.Err(); err != nil {
            break
        }
    }
}
```



## 3

## 第二阶段挑战

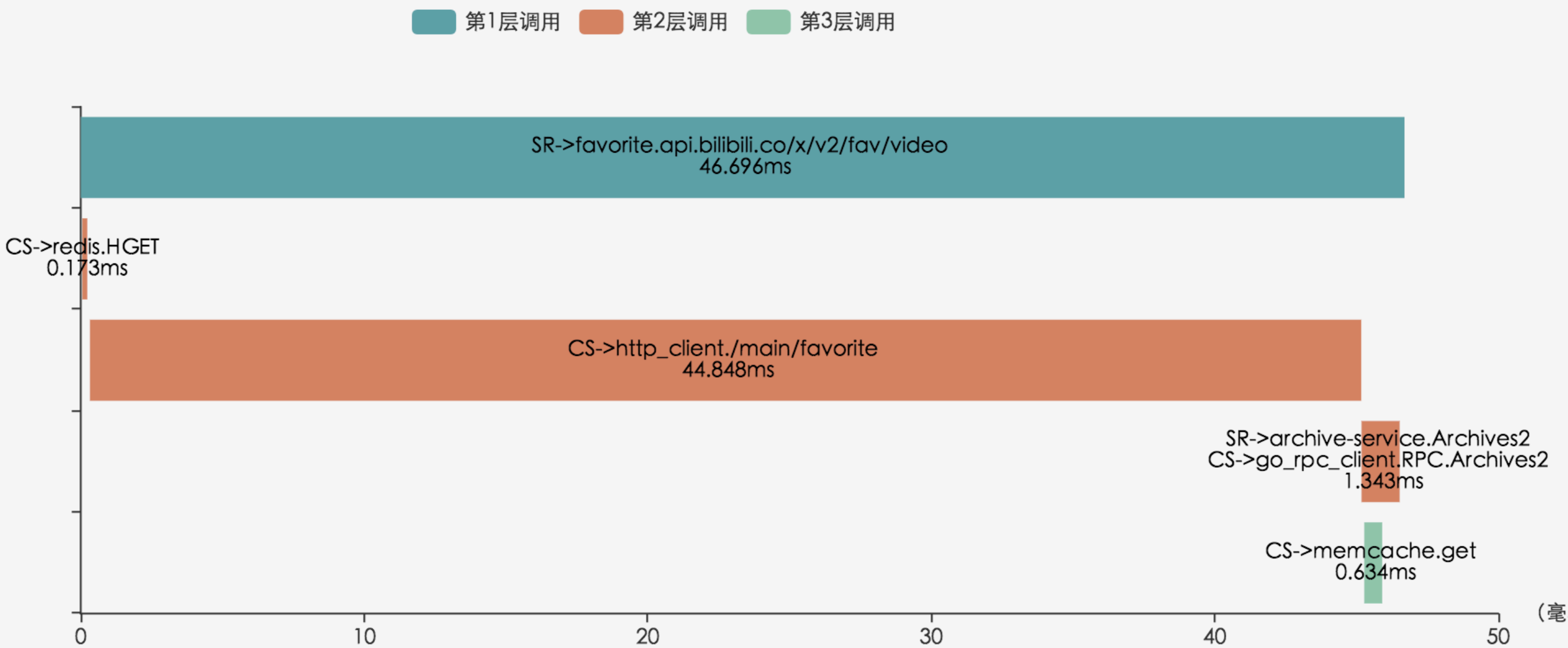
```
type conn struct {  
    p    *Pool  
    c    redis.Conn  
    t    *trace.Trace2  
    ctx  context.Context  
    addr string  
}
```

```
func (c *conn) Do(commandName string, args ...interface{}) (reply interface{}, err error) {  
    if err = c.Err(); err != nil {  
        return  
    }  
    if t, ok := trace.FromContext2(c.ctx); ok {  
        t = t.Fork(_family, commandName, c.addr, _class)  
        t.Client(key(args))  
        defer t.Finish()  
    }  
    return c.c.Do(commandName, args...)  
}
```



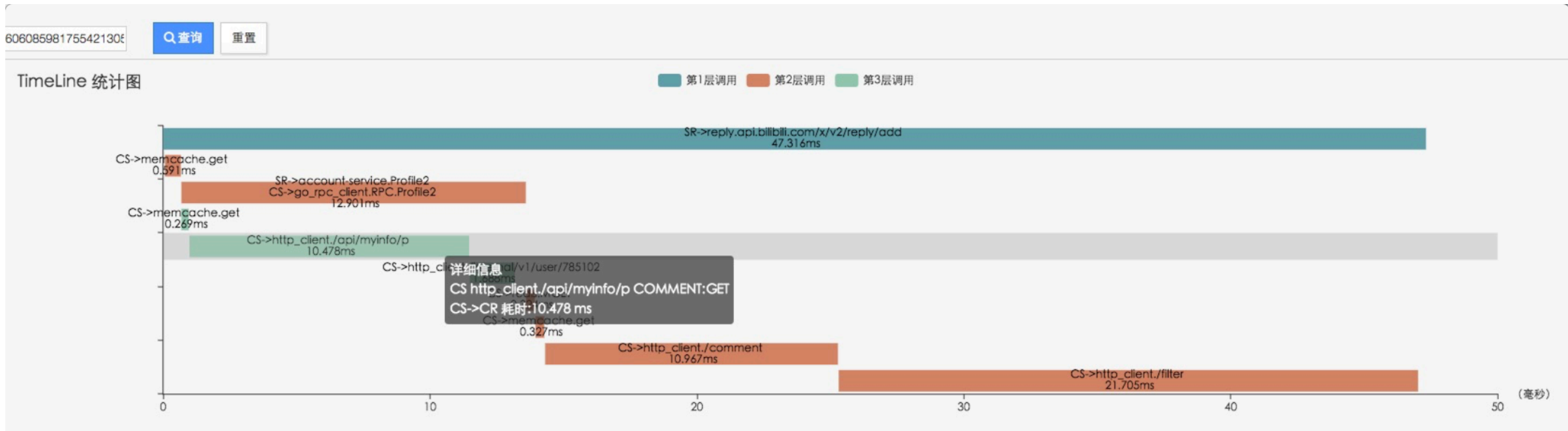
# 3

## 第二阶段挑战

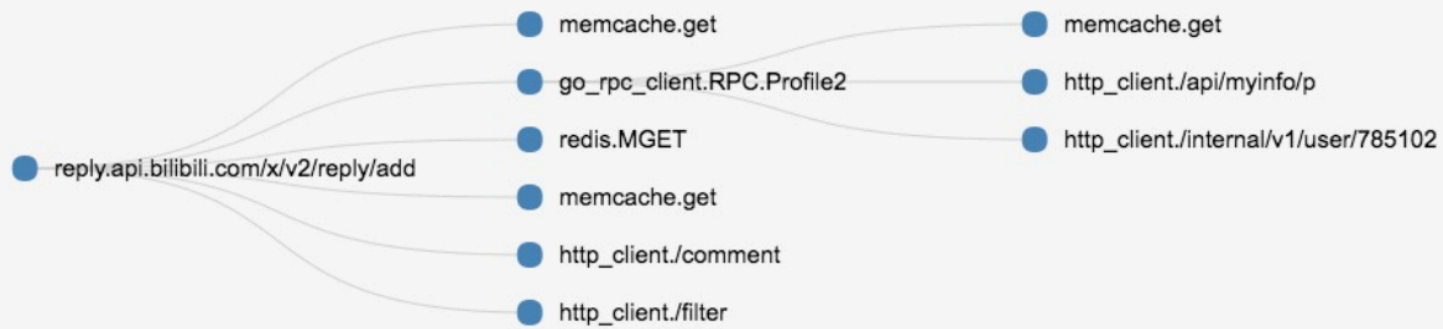


# 3

## 第二阶段挑战



### 服务节点结构图





## 依然存在很多问题

### 1, 不稳定的测试环境

服务众多，每个服务同时都有1-2个功能在提测，冲突，争抢资源

### 2, 服务难以管理及运维

每新增一个服务都要经过繁杂的配置环境过程

没有弹性扩容、没有自动化部署，可怜的运维GG%>\_<%



## 解决思路

## 1, 持续集成

快速发现错误, 防止分支大幅偏离主干

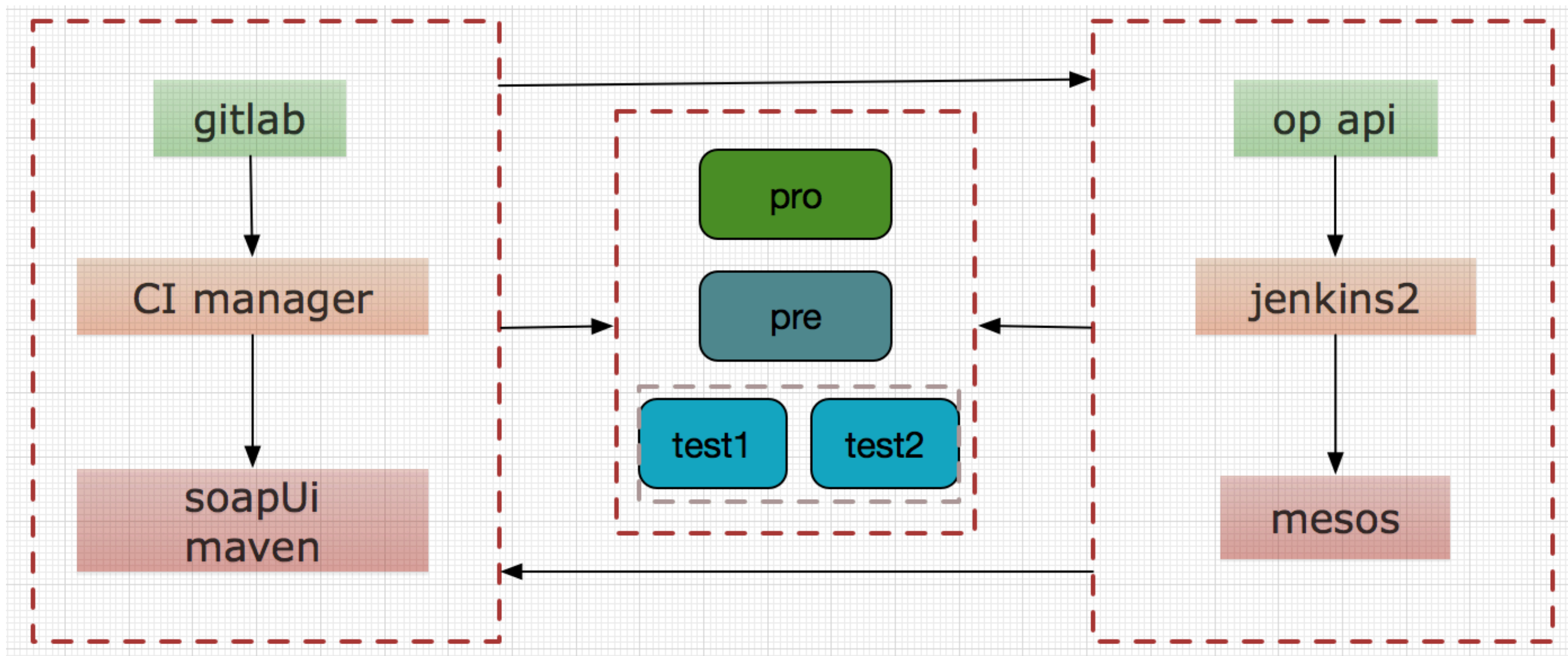
## 2, 持续交付/部署

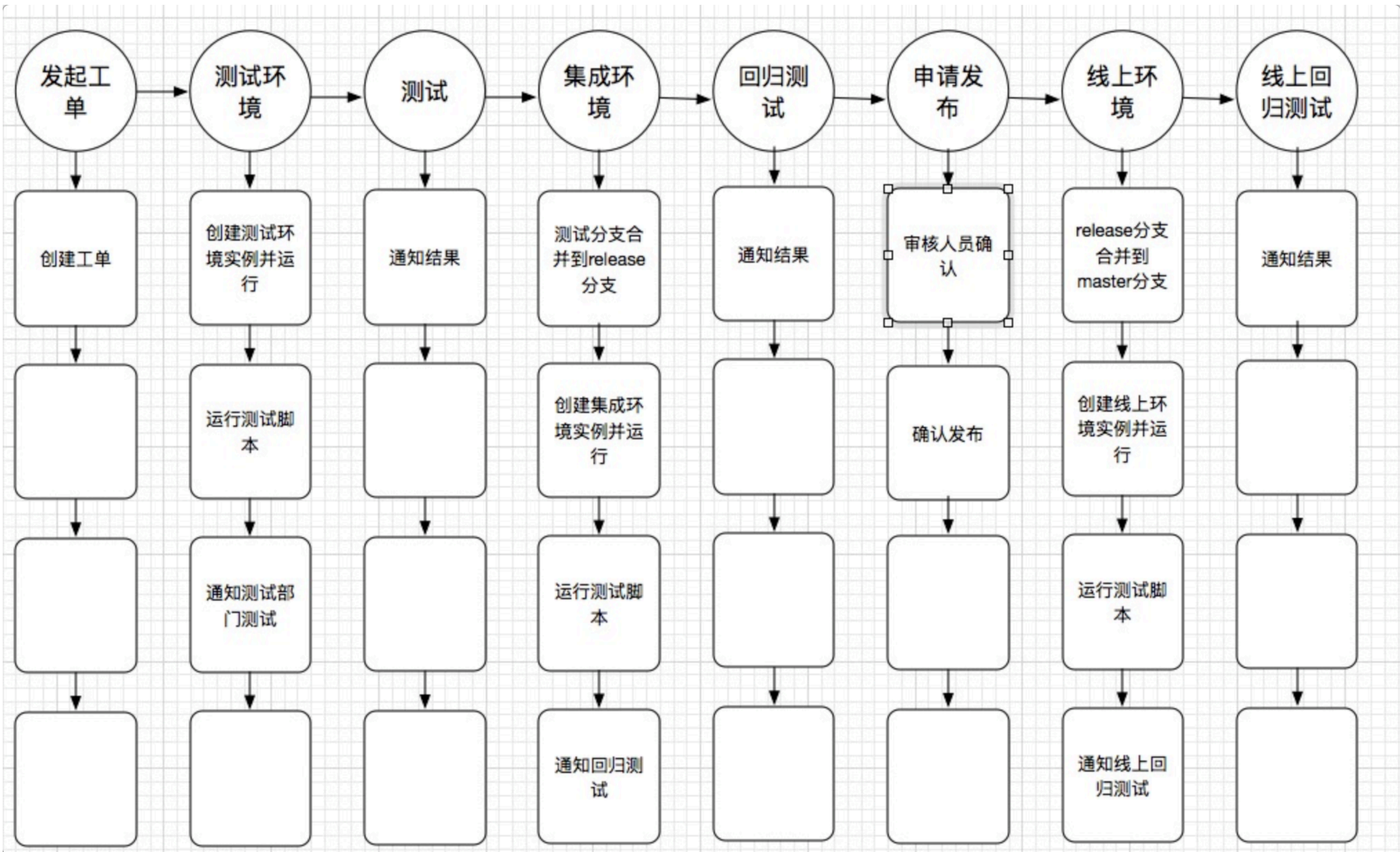
微服务要想用得好, 离不开docker



# 3

## 第二阶段挑战





## 3

## 第二阶段挑战

app管理

新增app +

搜索



	<input type="checkbox"/>	应用ID	Pool归属	状态	使用资源	容器情况	操作
+	<input type="checkbox"/>	bl-activity		Healthy	CPU: 1 MEM: 2000Mb	Runing: 1 Health: 1 Unhealth: 0	扩容 版本更新 应用配置 任务管理 监控
+	<input type="checkbox"/>	me		Suspended	CPU: 0 MEM: 0Mb	Runing: 0 Health: 0 Unhealth: 0	扩容 版本更新 应用配置 任务管理 监控

显示第 1 到第 2 条记录，总共 2 条记录





# 3

## 第二阶段挑战

### 版本更新

镜像

docker [redacted].bilibili [redacted]/docker [redacted]-activity

版本选择

13

正常更新

灰度更新

### 容器情况

🔄 Deploying

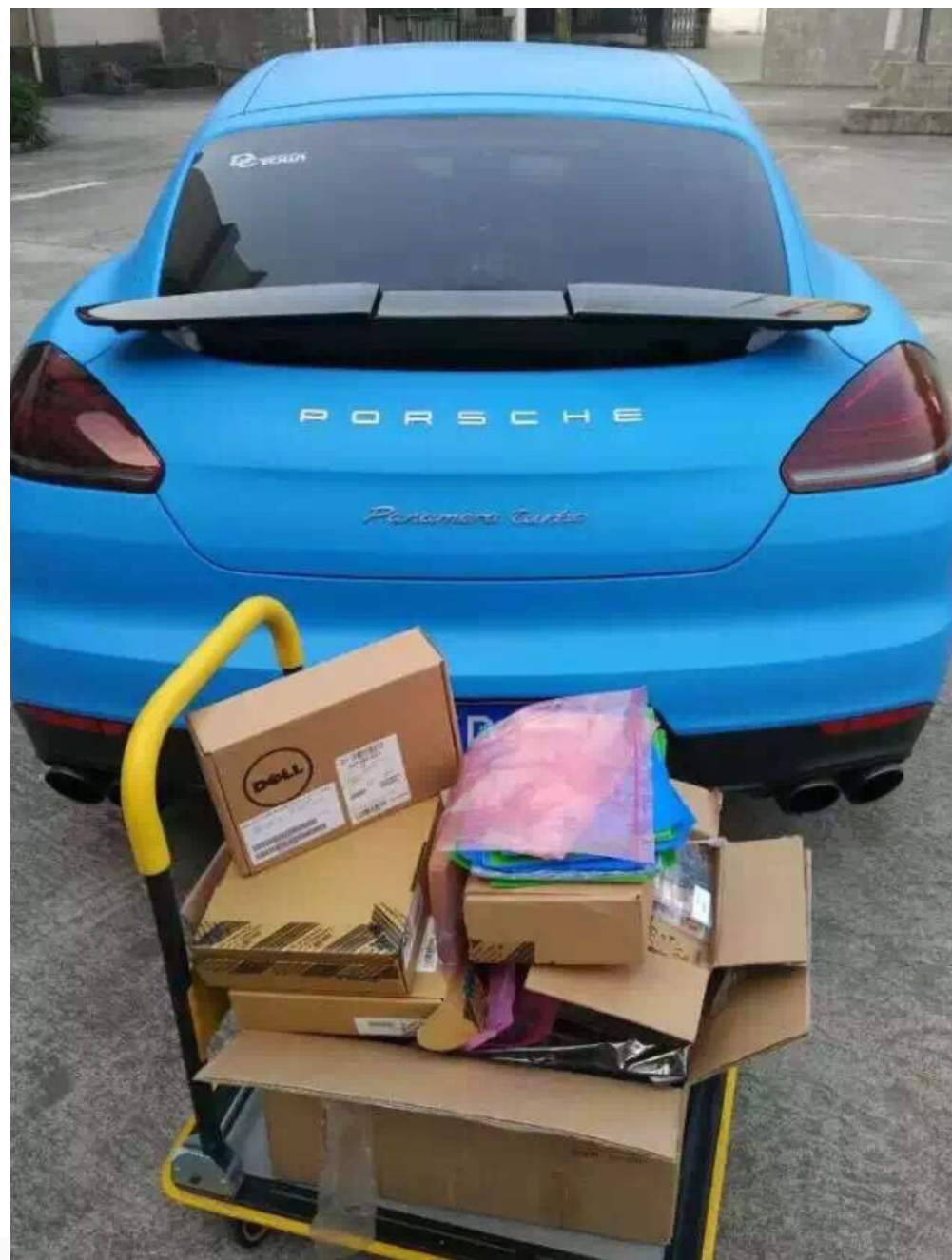
环境	IP	Port	Image	Status	Health
生产	172.17.0.2	8081	docker [redacted].bilibili [redacted]/docker [redacted]-activity:14	启动	health
生产	-	-	docker [redacted].bilibili [redacted]/docker [redacted]-activity:13	待分配	unkown



## 4

## 求加入一起搞基

Bilibili 作为最大的同性交友网站  
急需各类人才加入Go/PHP/JAVA/FE ==  
求骚扰微信: haogw241







女神 求bgm 三道街 第失踪人口回归! 女神 老婆! 巴洛克  
 女王 没有弹幕?? 哇哇...这腰 女神你回来了啊 女神回归!!! 哇哇哇~  
 好瘦啊啊 中华巴洛克 居然没人 绿字哈哈 尴尬 尴尬  
 卧槽真的尴尬 六 挖掘姬 那微笑好迷人啊 女神好久不见 哈尔  
 参见尬舞女王 道外巴洛克。 寻找失踪人口  
 柔 群众已吓尿。 好瘦  
 尴尬 B站我就服微小微! 老婆你终于回  
 求路人视角  
 找死也要有方法, 没被倒车的闯到, 这次要被老大爷  
 da jia hao  
 我少时的lion heart  
 这啥时候来的! ?  
 仙女好瘦啊, 多

**B站第一羞耻女王 就服仙女**  
**冬夜表白仙女**  
**你爱搞他 不懂他 四娘刚掐架**  
**这小蛮腰!!!!**





谢谢大家

哔哩哔哩 - ( ° - ° ) > 口 乾杯~

bilibili