



bilibili



主题

bilibili监控体系



目录

1,思考

2,理论

3,应用

4,展望



业务层：

业务指标，如注册成功率、投稿成功率；

应用层：

“端”监控，链路层监控，日志，异常；

系统层：

网络、IDC、CDN、中间件、数据库；



Traceon

Misaka

Dapper

ELK

Traceon

Zabbix



专注“端”监控



专注“**业务**”监控



专注“采集”日志



大盘：变更、当前报警、失败率、全链路的情况；

前端：提供端->服务的服务质量监控；

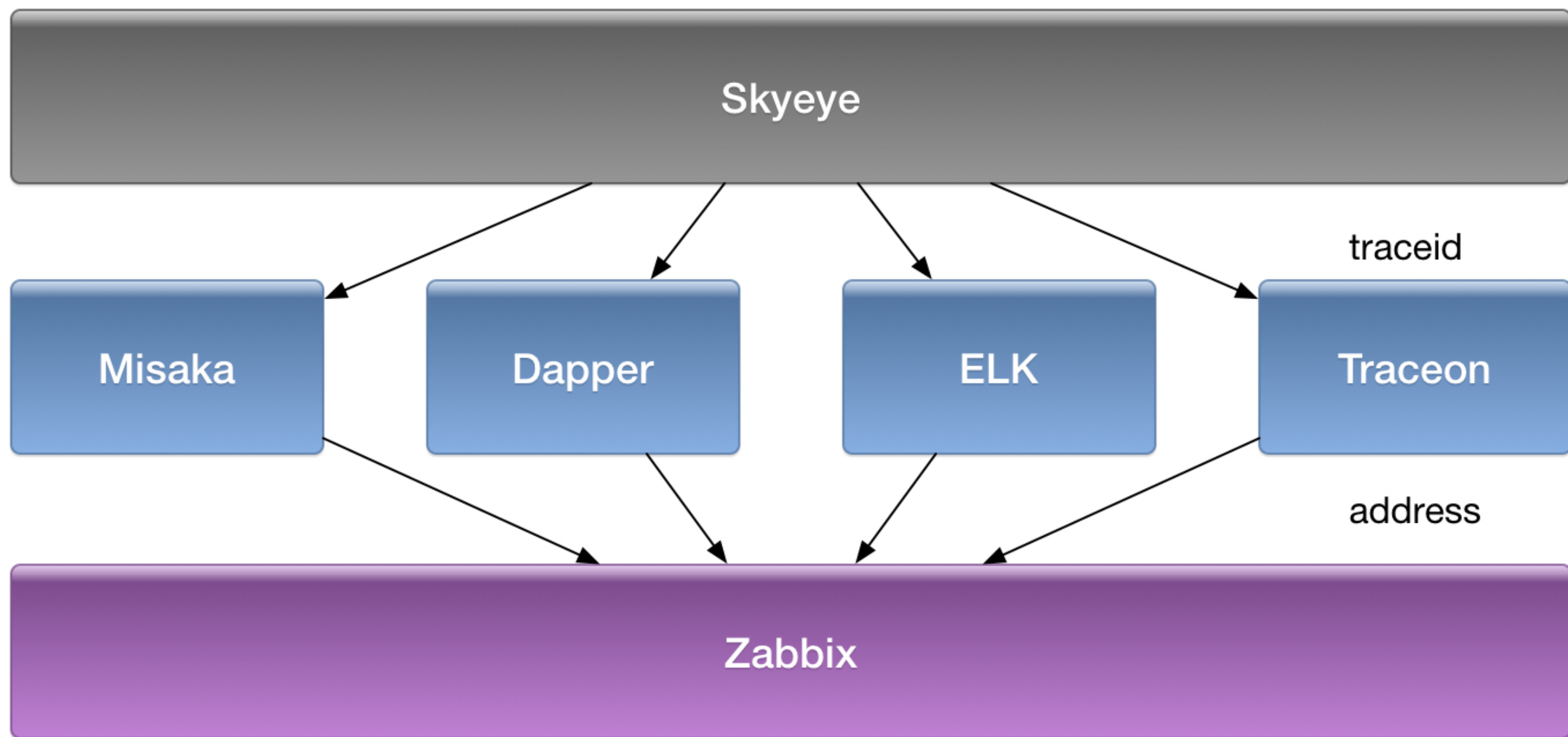
异常：对失败率、异常汇总统计入口；

业务：投稿成功率等这类指标；

链路：方便查询特定业务链路的情况；

系统：核心网络、CDN、IDC等；





需要什么？

- 1, 及时报警, 正确报警, 规划报警;
- 2, 查看问题, 追踪问题, 解决问题;

怎么做？

- 1, 大系统小做
- 2, 分而治之



- 1, 复杂的、大规模分布式集群来实现的
- 2, 不同的团队、不同的编程语言、不同的数据中心

因此，就需要一些可以帮助理解系统行为
用于分析性能问题的工具。

如何定位root cause成为难题！



举个例子

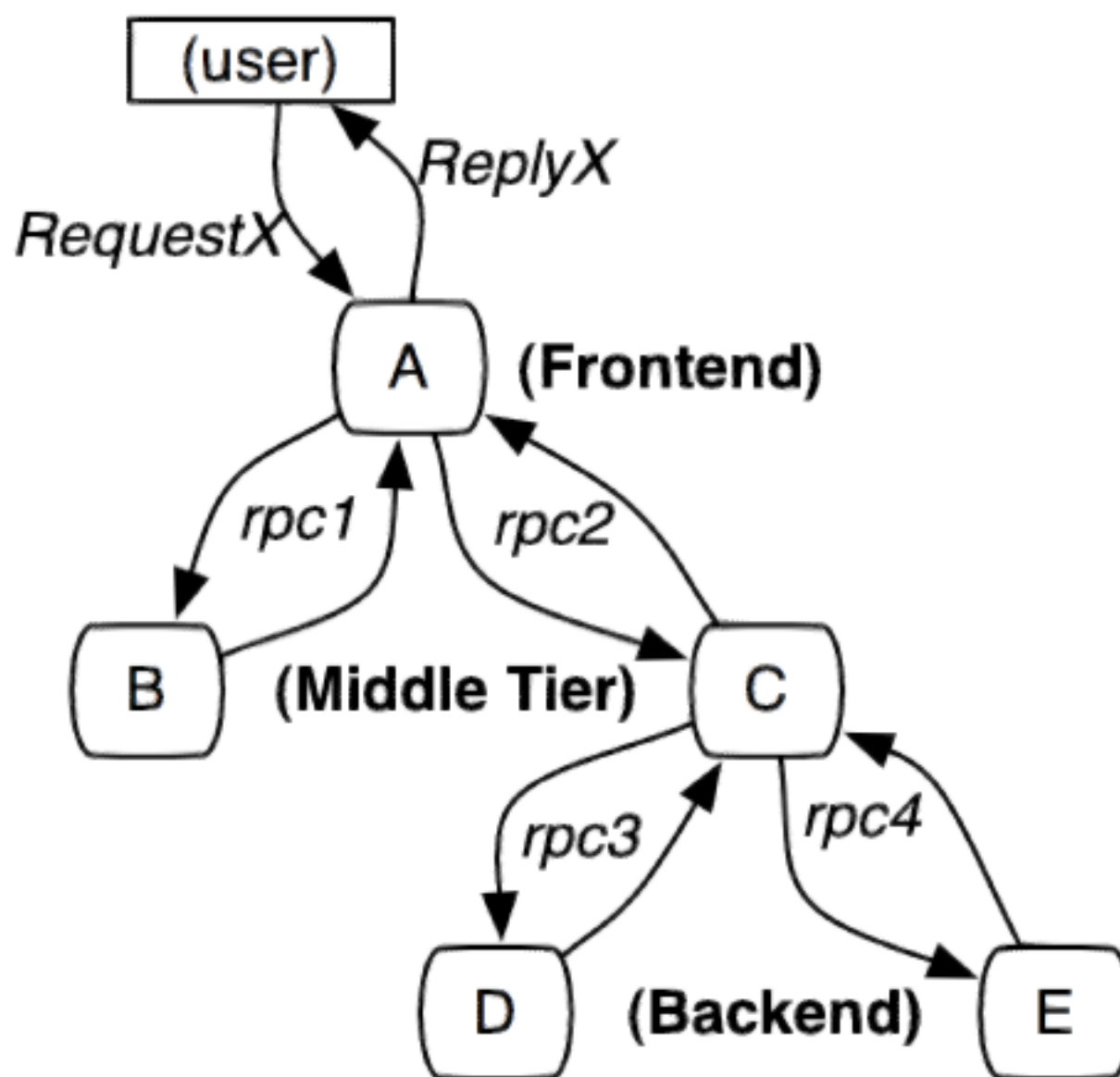
用户浏览移动端首页，一次请求到达服务端

- 1, 编辑推荐数据及运营数据
- 2, 大数据推荐数据
- 3, 直播推荐数据
- 4, 番剧推荐数据
- 5, 分区推荐数据

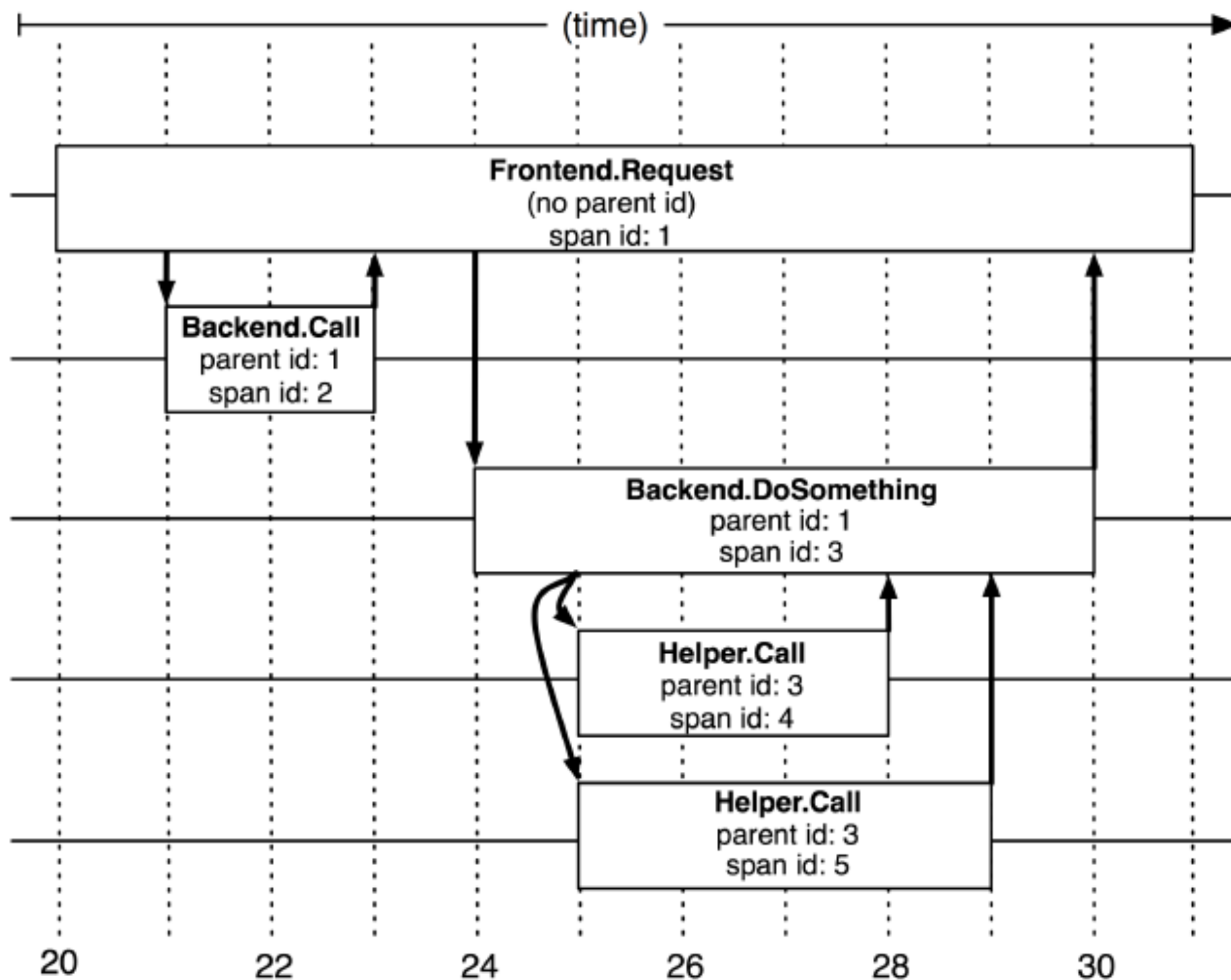
涉及多个服务，用户对耗时敏感
出问题全靠猜(T.T)



树形调用：记录在一次特定的请求后系统中完成的所有工作的信息



5个span在Dapper跟踪树种短暂的关联关系



基于Google Dapper论文

- 1, 日志收集及采样率
- 2, 植入公共组件
- 3, 图形化

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



业务服务

Service

Service

收集

Agent

Agent

分发

Lancer

Lancer

存储

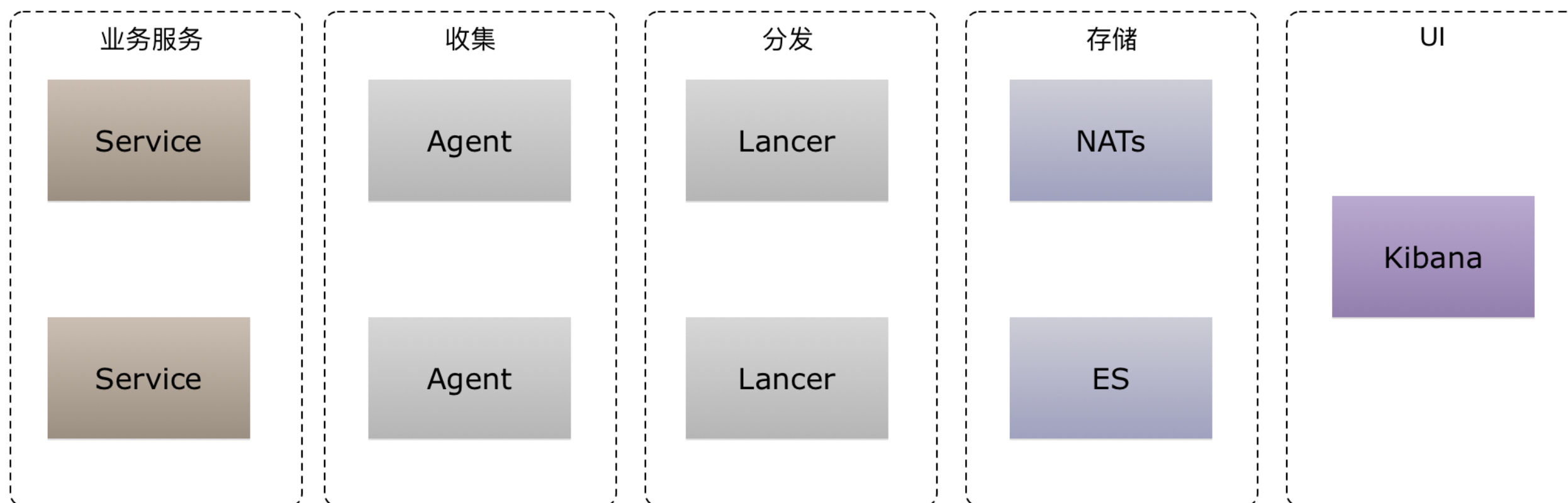
NATs

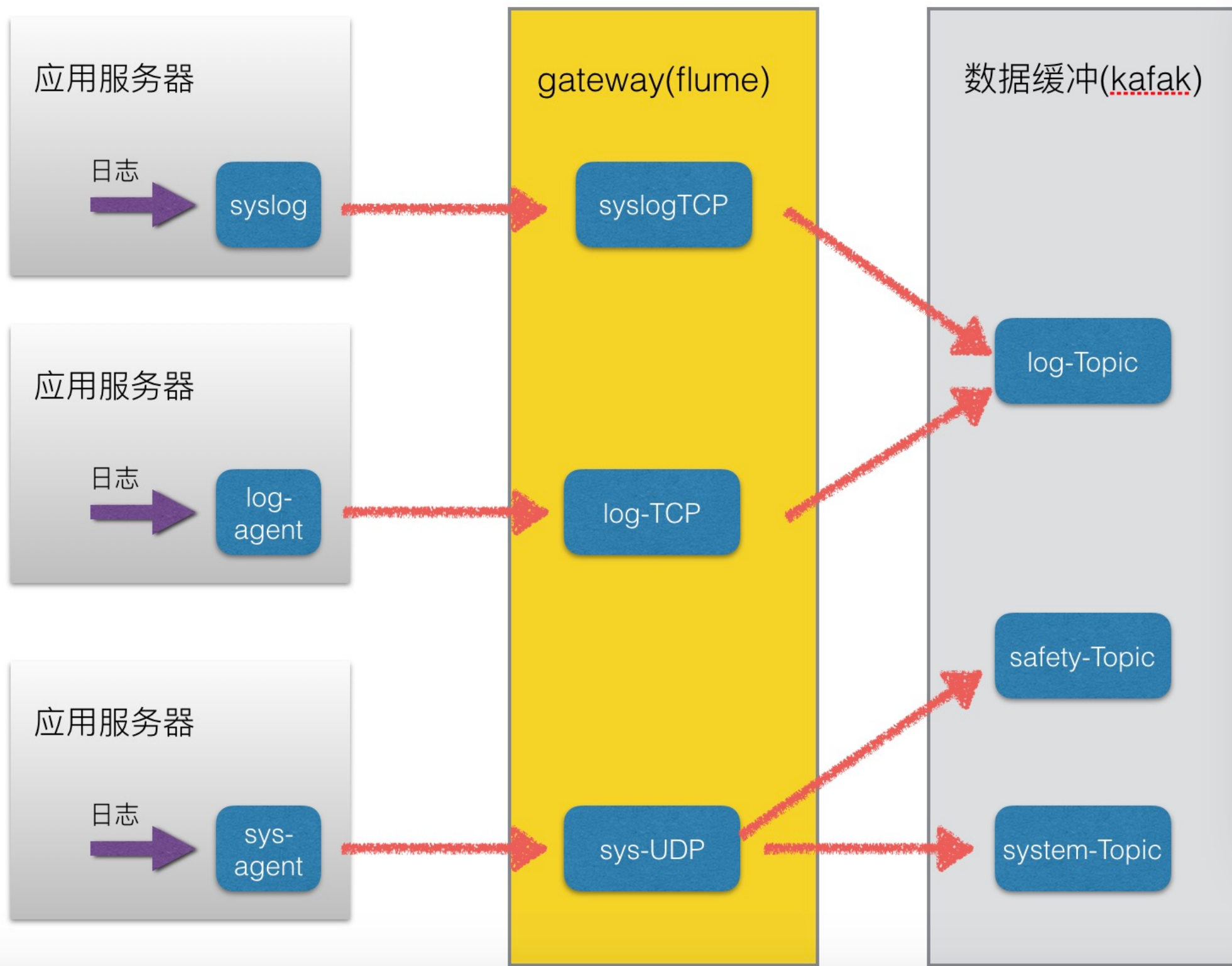
ES

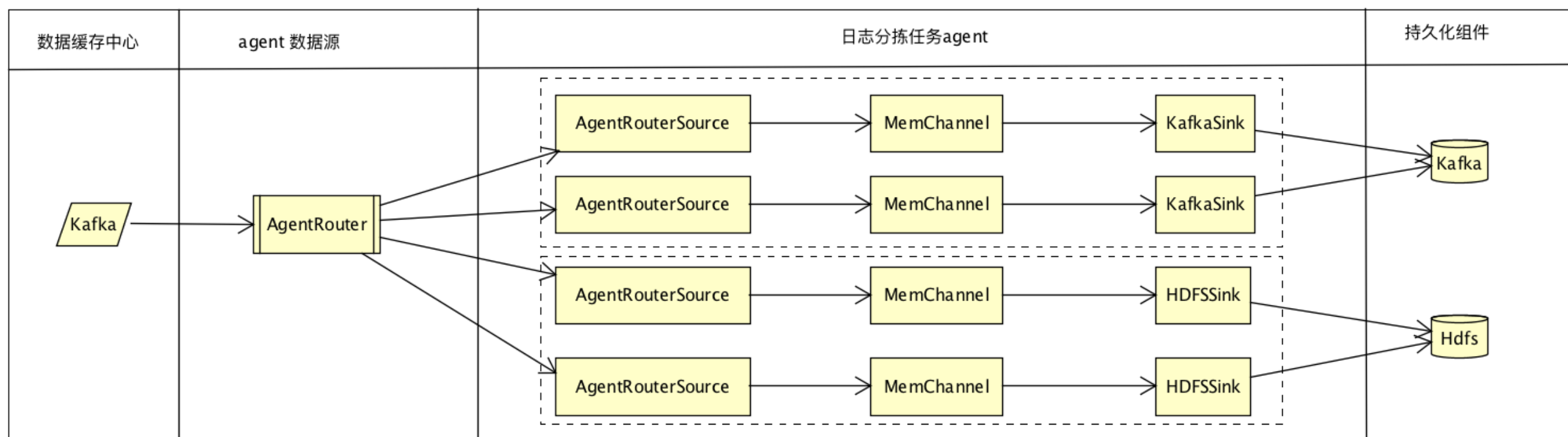
UI

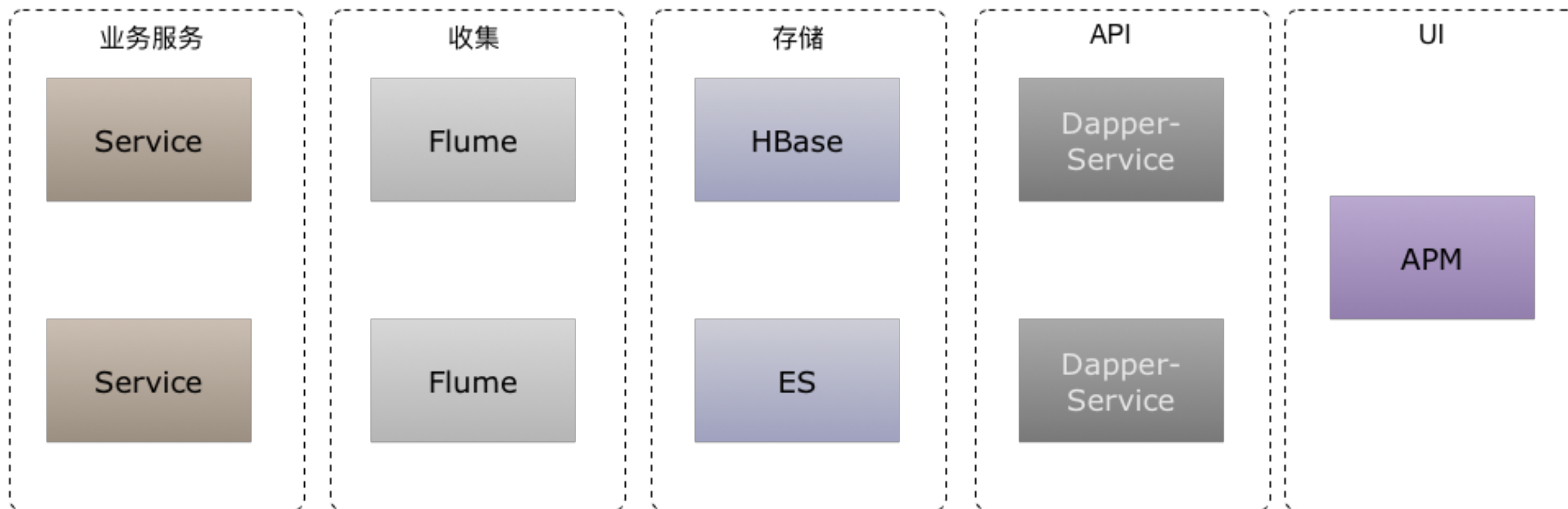
Kibana











采样率

- 1, 固定采样 1/1024
- 2, 可变采样
- 3, 可控采样

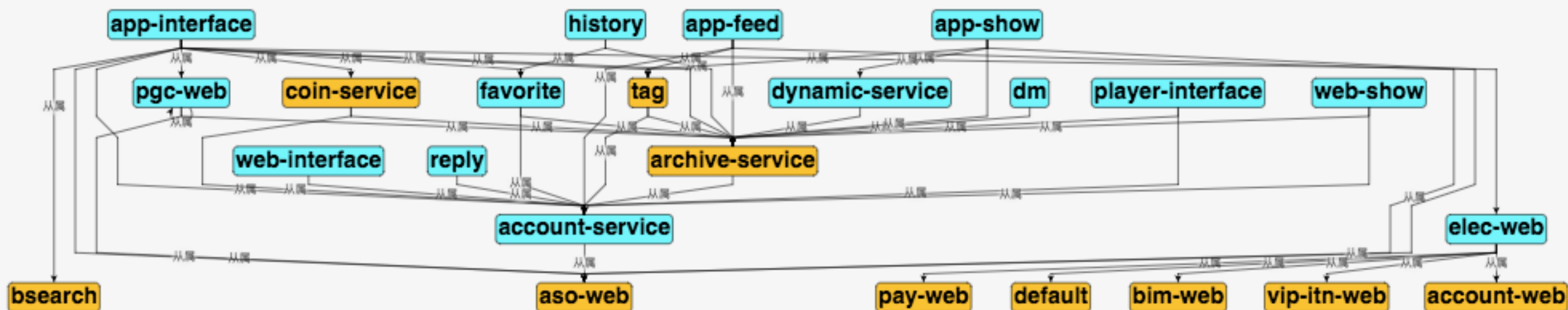


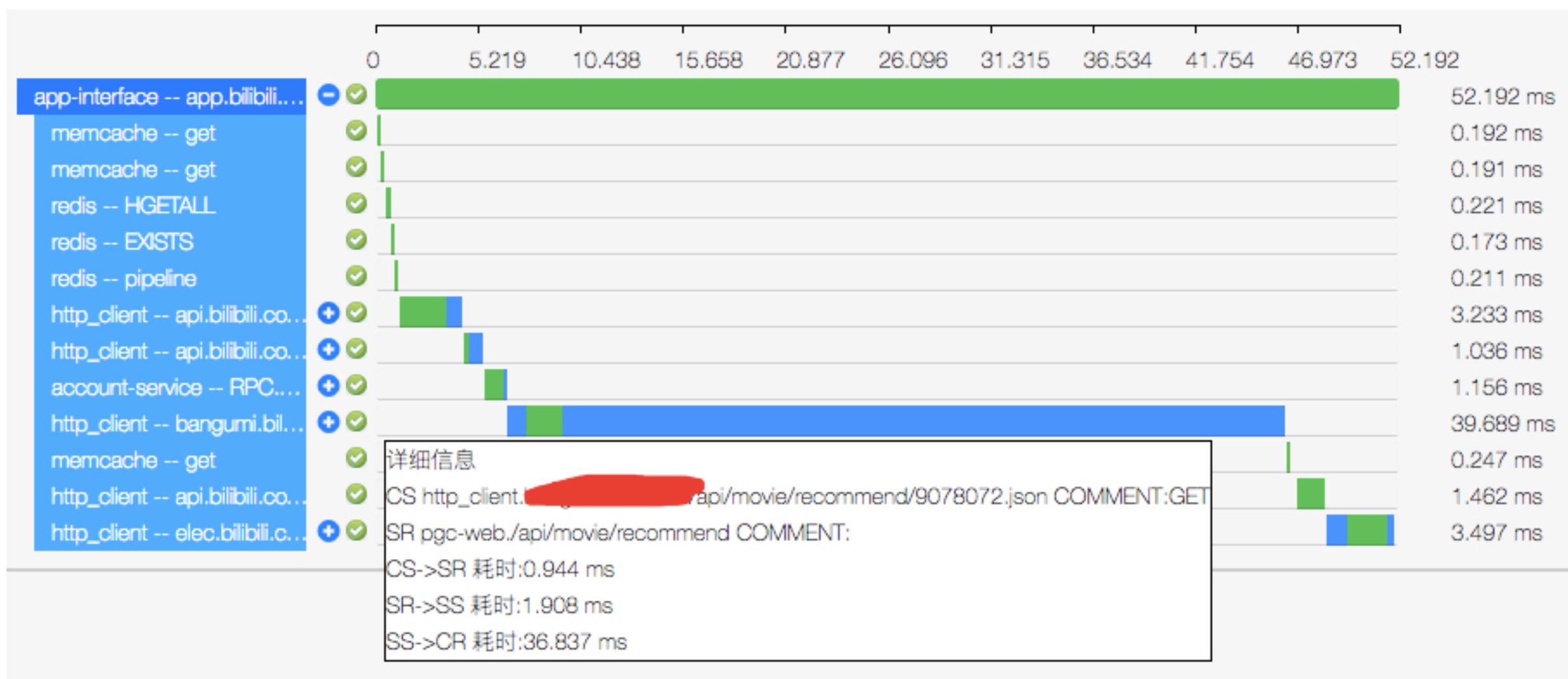
确定植入点，封装通用组件

```
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  
        }  
    }  
}
```

```
func (db *DB) Query(c context.Context, query string, args ...interface{}) (*sql.Rows, error) {  
    if t, ok := trace.FromContext2(c); ok {  
        t = t.Fork(_family, "query", db.addr, _class)  
        t.Client(query)  
        defer t.Finish()  
    }  
    return db.DB.Query(query, args...)  
}
```







*入口服务名:

app-interface

Q 获取title

app-interface

*选择title:

app.bilibili

*选择查询时间:

时间段

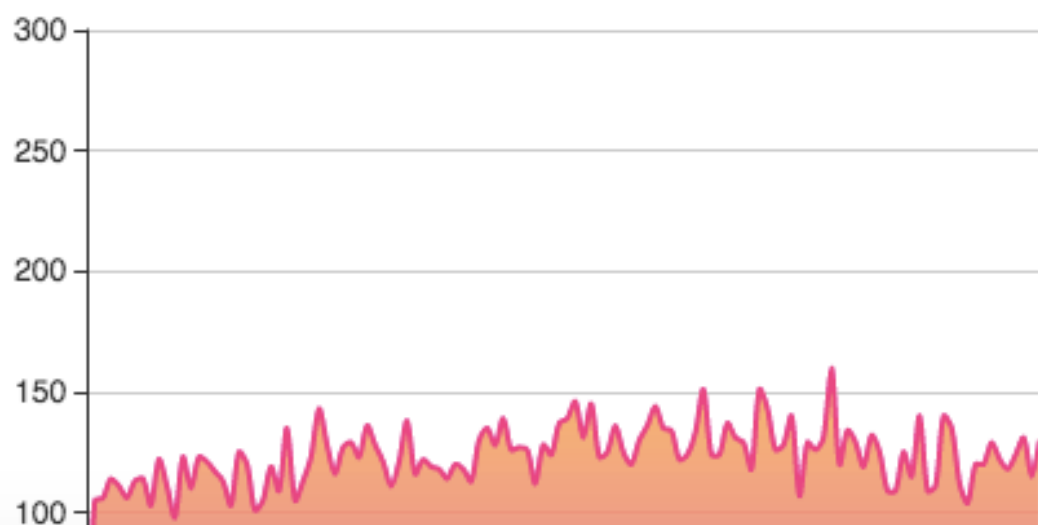
2017-03-10 21:45:40

2017-03-10 23:45:40

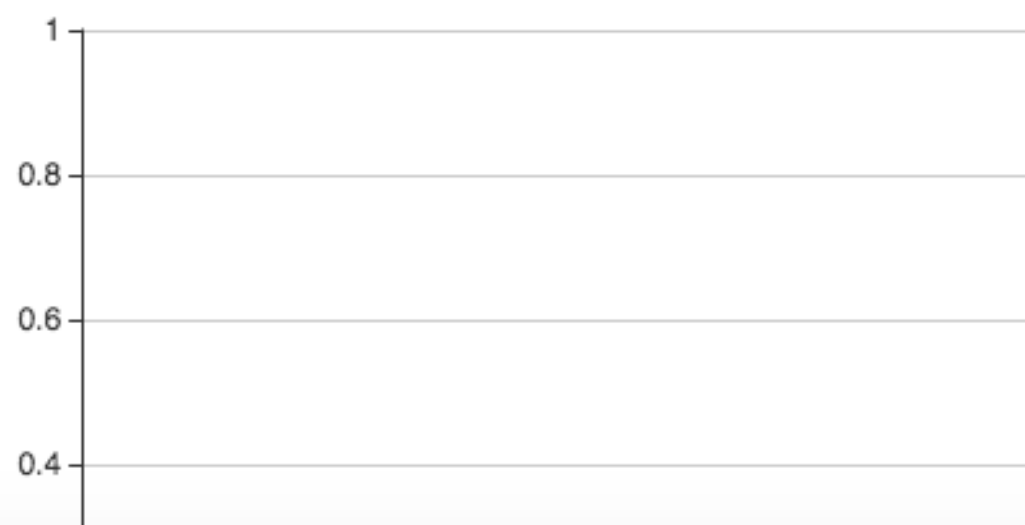
Q 搜索

重置

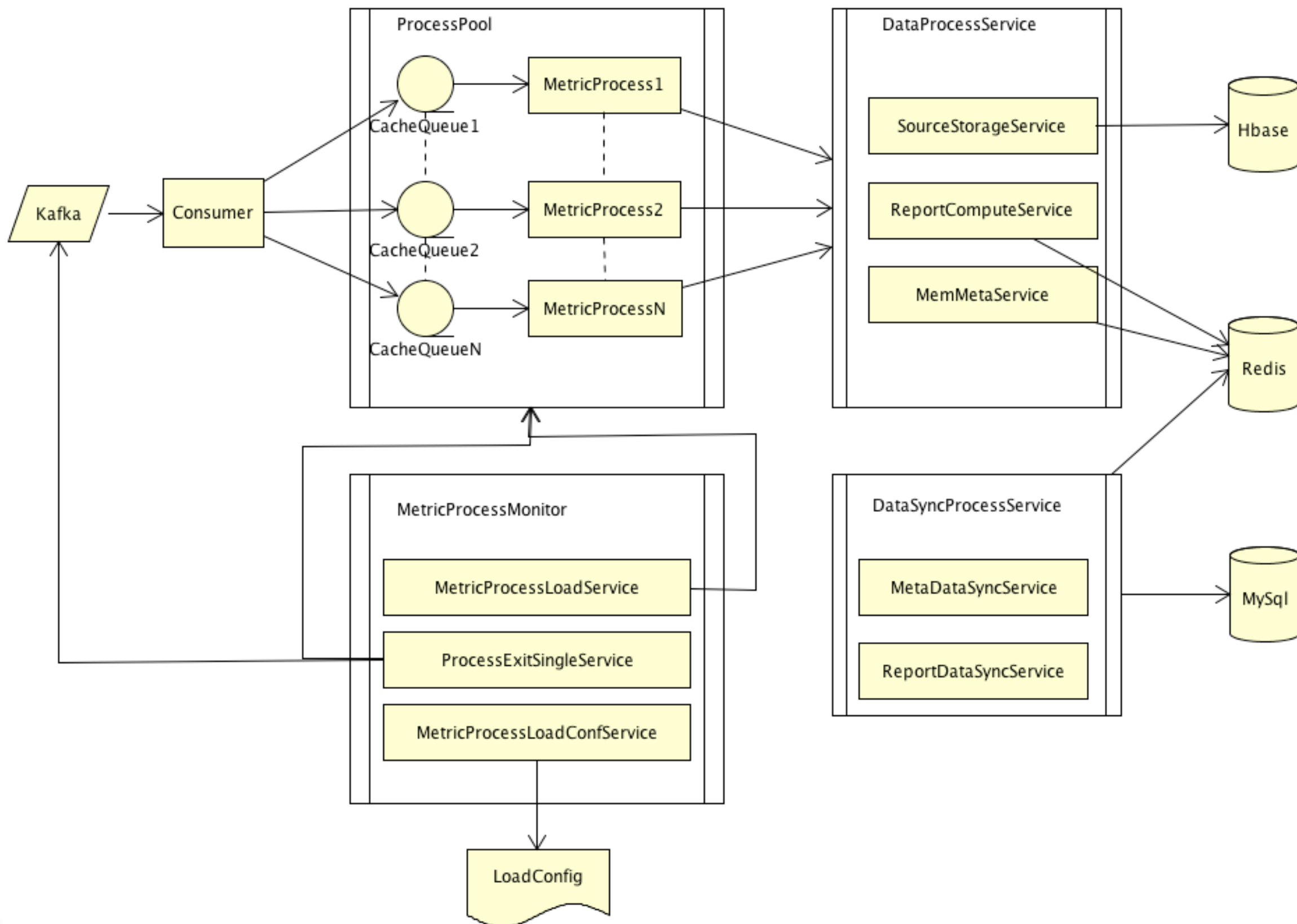
每分钟数据量



耗时直方图







XXX XXX XXX
管理员

监控类型

异常大盘

业务大盘

监控注册

规则设置

查询管理

权限管理

◀ 首页 业务大盘 ▶

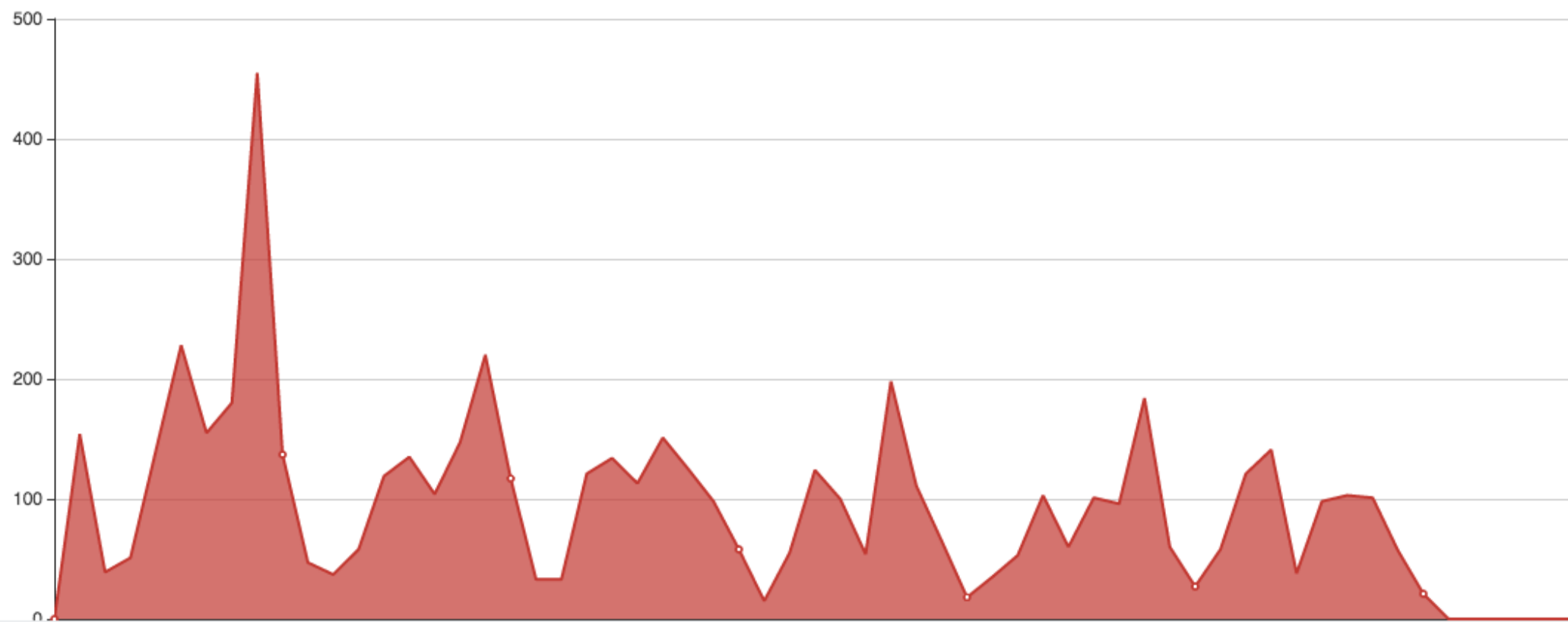
请选择

1天

1周

1月

指标走势



© 2016-2017 bilibili



- 1, 各个系统更加独立完善, 各司其职;
- 2, 完美的Dashboard ;
- 3, Alert&Event在Skyeye系统收敛更好;
- 4, 完善监控的流程和Oncall机制 ;





bilibili