

# 企点微服务网关演进之路

郑榕

郑榕(roman)，腾讯后台高级工程师，  
腾讯企点Web后台基础技术建设负责人。

十年C/C++开发，曾就职于大唐电信，  
科达科技等通信领域公司，涉及音视频实时  
通信，音视频存储，流媒体点播等方面。

两年PHP开发，喜欢折腾Nginx，PHP扩  
展，Zend内核。



2. 网关技术选型

1. 初代架构

3. 动态路由

5. 安全控制

4. 服务注册发现

6. 服务监控

7. 整体回顾  
未来规划



# 初代架构

接入层

公司接入层

业务接入层

逻辑层

监控

展现层

业务逻辑层

数据访问层

日志

数据层

DB

Cache

- 手动配置路由
- 集群难以管理

- 按大业务划分
- 手动注册服务
- 自我监控

# 单体服务越来越大

代码量迅速**膨胀**

功能越来越多

**轻重**不分离

**流量**越来越大

服务**负载**

服务又**异常**了

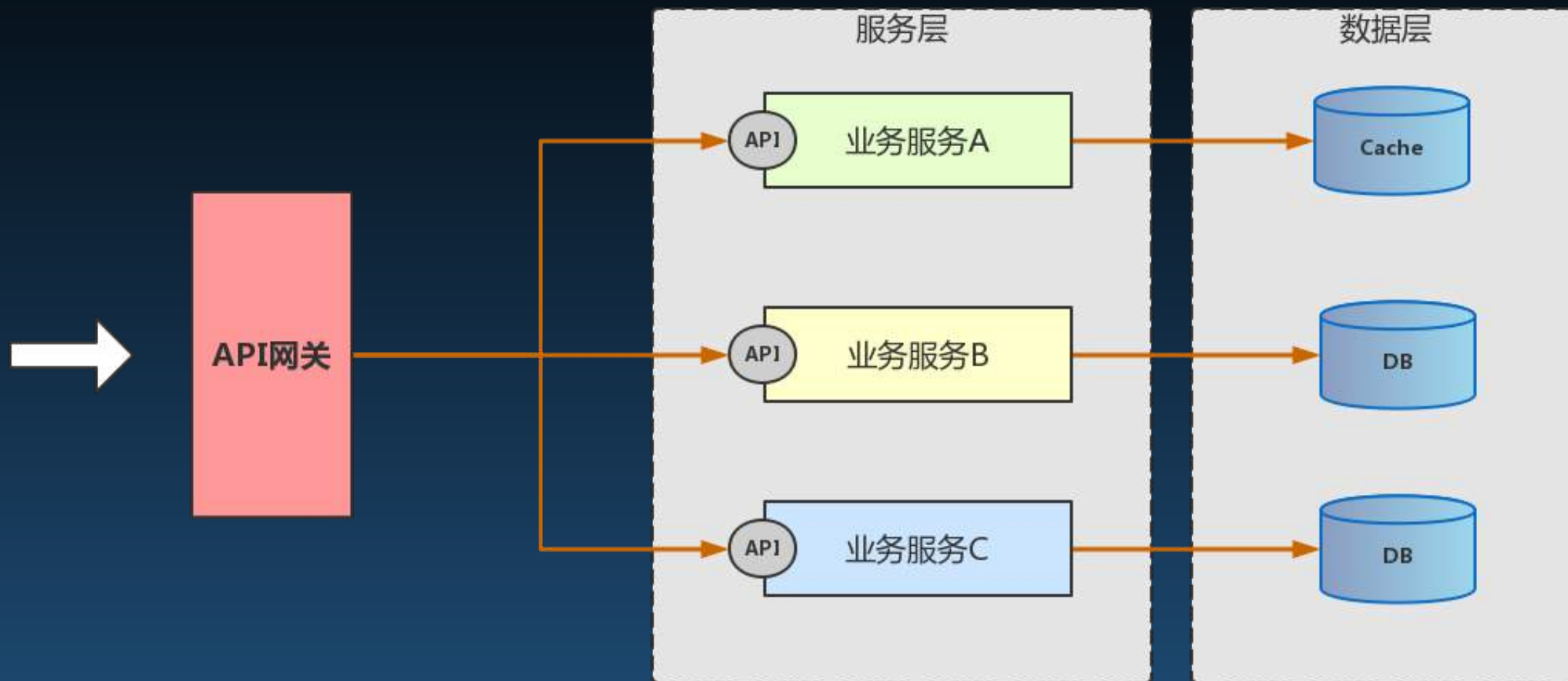
一个服务**多人**编写



接口性能**时好时坏**

接口A**扩容**，接口B：关我啥事**是时候考虑微服务了**

# 微服务主流架构



# 网关技术选型

主流方案



考量因素





# 网关技术选型

- 最美好的就在身边
- 高性能服务器的典范
- 资源占用少
- 稳定性极佳
- 灵活的扩展性



Why Not ?

NGINX



API网关

# 聊聊 ProxyPass

```
Location /test{  
    proxy_set_header Host testhost.com;  
    proxy_set_header X-Forwarded-For $remote_addr;  
    proxy_set_header X-Real-IP $remote_addr;  
    proxy_pass http://testhost.com/$request_uri;  
}
```

```
upstream testhost.com {  
    ip_hash;  
    server XX.XX.XX.XX:XXXXX;  
    server XX.XX.XX.XX:XXXXX;  
}
```

# 聊聊 ProxyPass

赖以成名的反向代理

优秀的性能

良好的协议兼容

极佳的稳定性

NGINX的精华

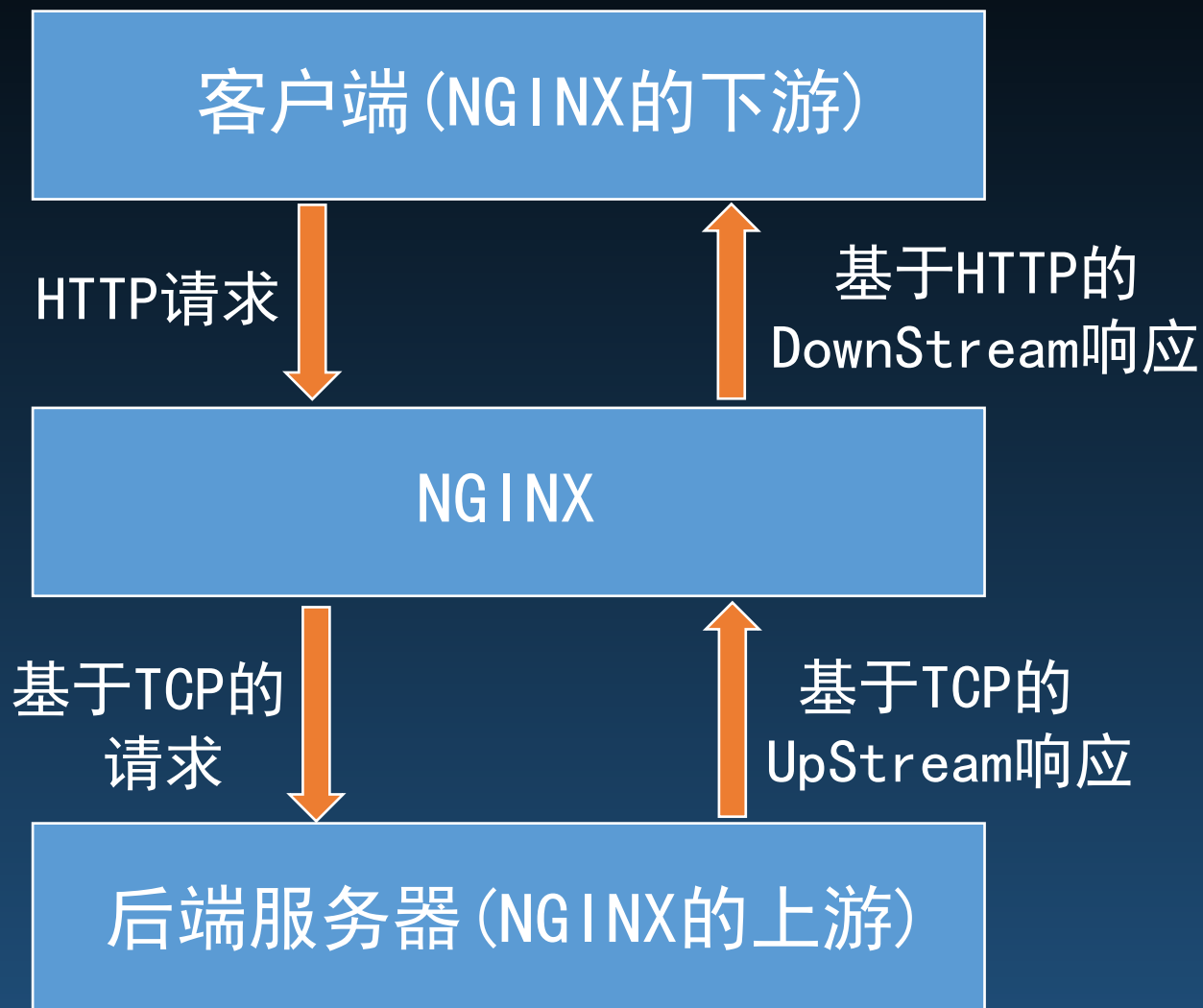
但是，我们要的是**动态路由**.....

# 动态路由

- 重写反向代理吗？
- 如果要，怎么做？
- 使用upstream机制
- 什么是upstream？

其实……

原生反向代理模块用的也是upstream机制



# 动态路由

原生Proxy\_module代码行数 5000+  
我们只要动态路由，


原生Proxy\_module提供命令数 62个

真的要**放弃**原生Proxy\_module吗……  
一个灵活可靠的反向代理**不简单**

# 动态路由

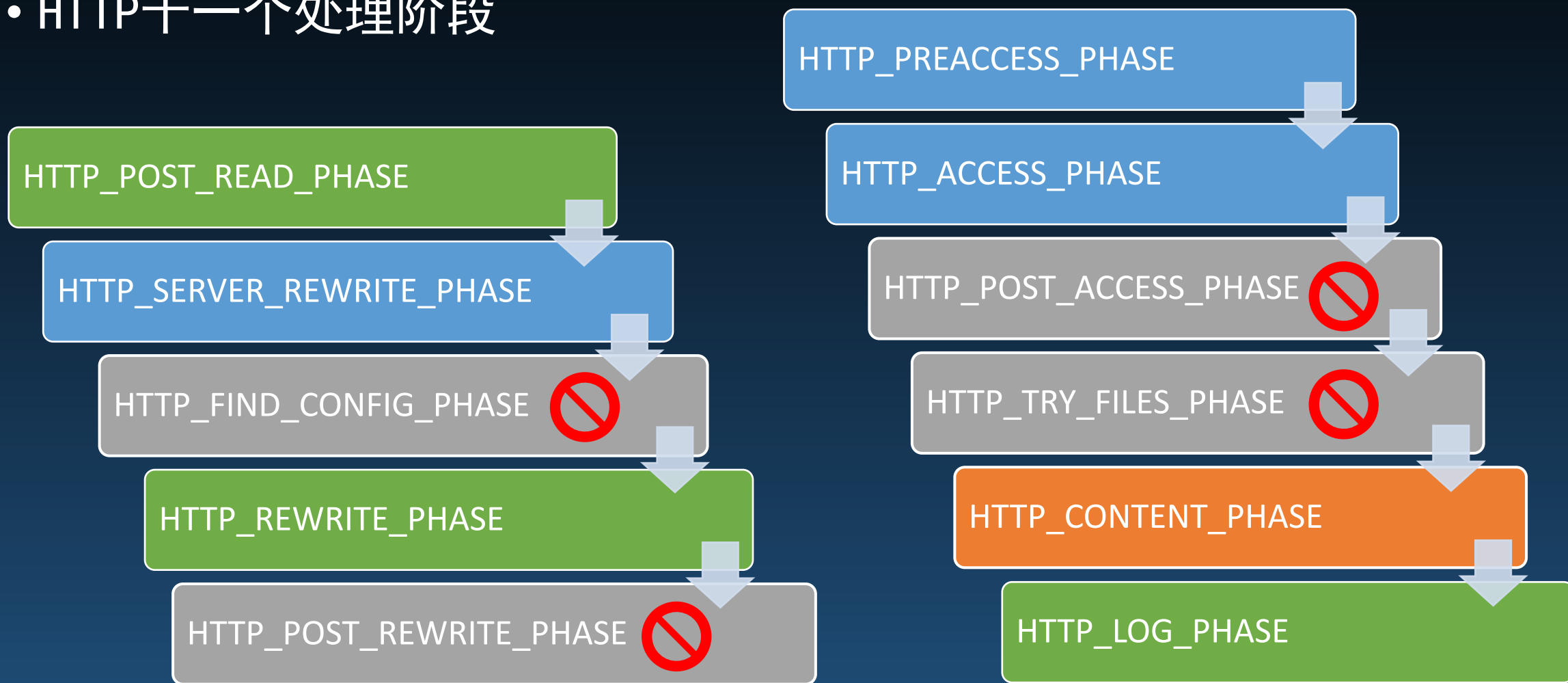
- 先看个例子
- 好像有办法了……

```
proxy_pass http://10.1.1.1/$request_uri;  
proxy_set_header Host testhost.com;  
proxy_set_header X-Forwarded-For $remote_addr;  
proxy_set_header X-Real-IP $remote_addr;  
proxy_pass http://$ddrouter/$request_uri;  
proxy_pass http://10.1.1.1/$request_uri;  
}
```



# 动态路由

- HTTP十一个处理阶段



# 动态路由

REWRITE\_PHASE

HTTP\_ROUTER\_MODULE  
(自定义模块)

```
ngx_http_add_variable  
    ( qdrouter )
```

CONTENT\_PHASE

HTTP\_PROXY\_MODULE  
(原生模块)

```
Location / {  
    proxy_pass  
    http://$qdrouter/$request_uri;  
}
```

- 优雅重用
- 事半功倍



# 动态路由

- 终于可以尽情控制路由行为了



坐稳，出发了！



2. 网关技术选型

3. 动态路由

5. 安全控制

4. 服务注册发现

6. 服务监控

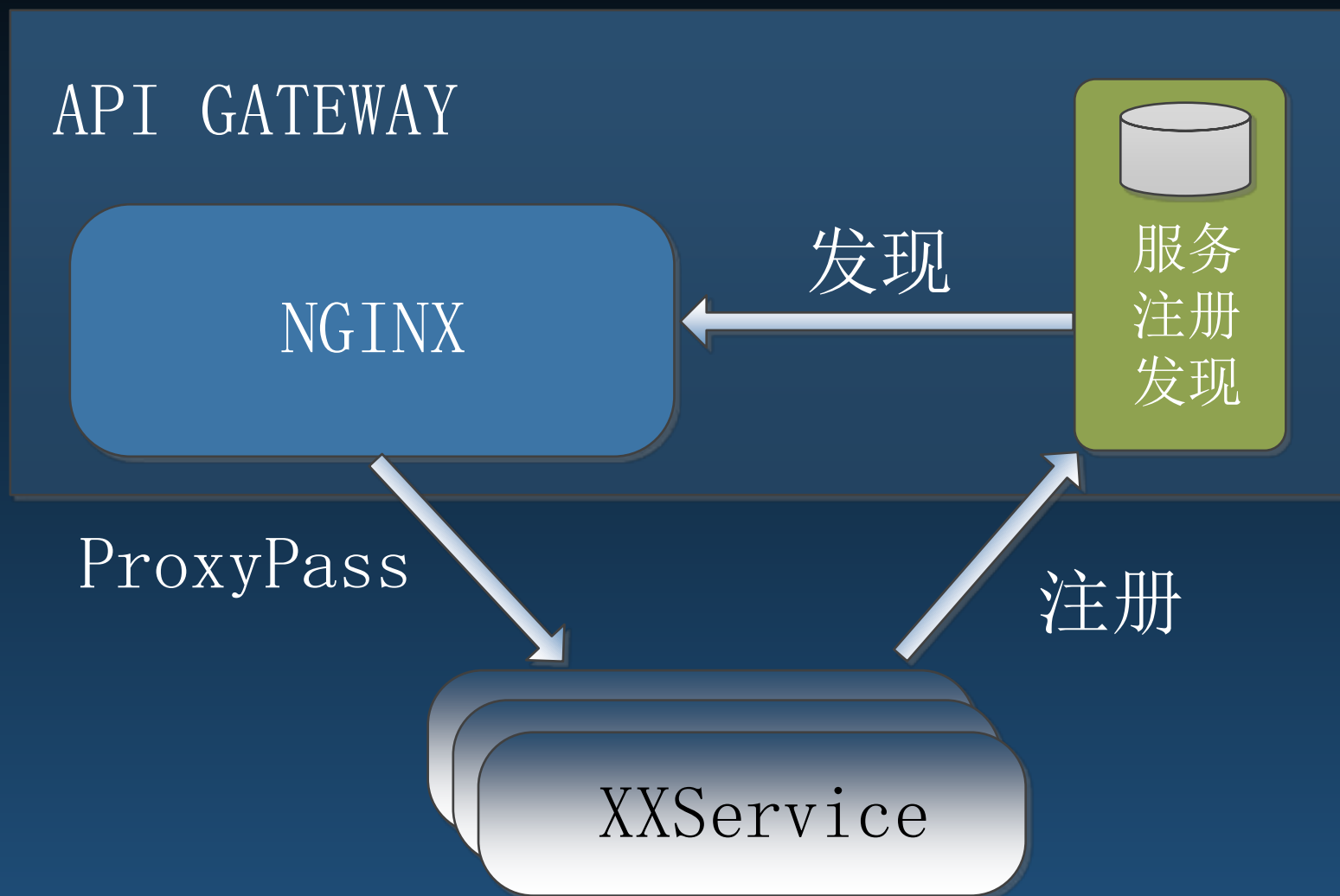
7. 整体回顾  
未来规划

1. 初代架构



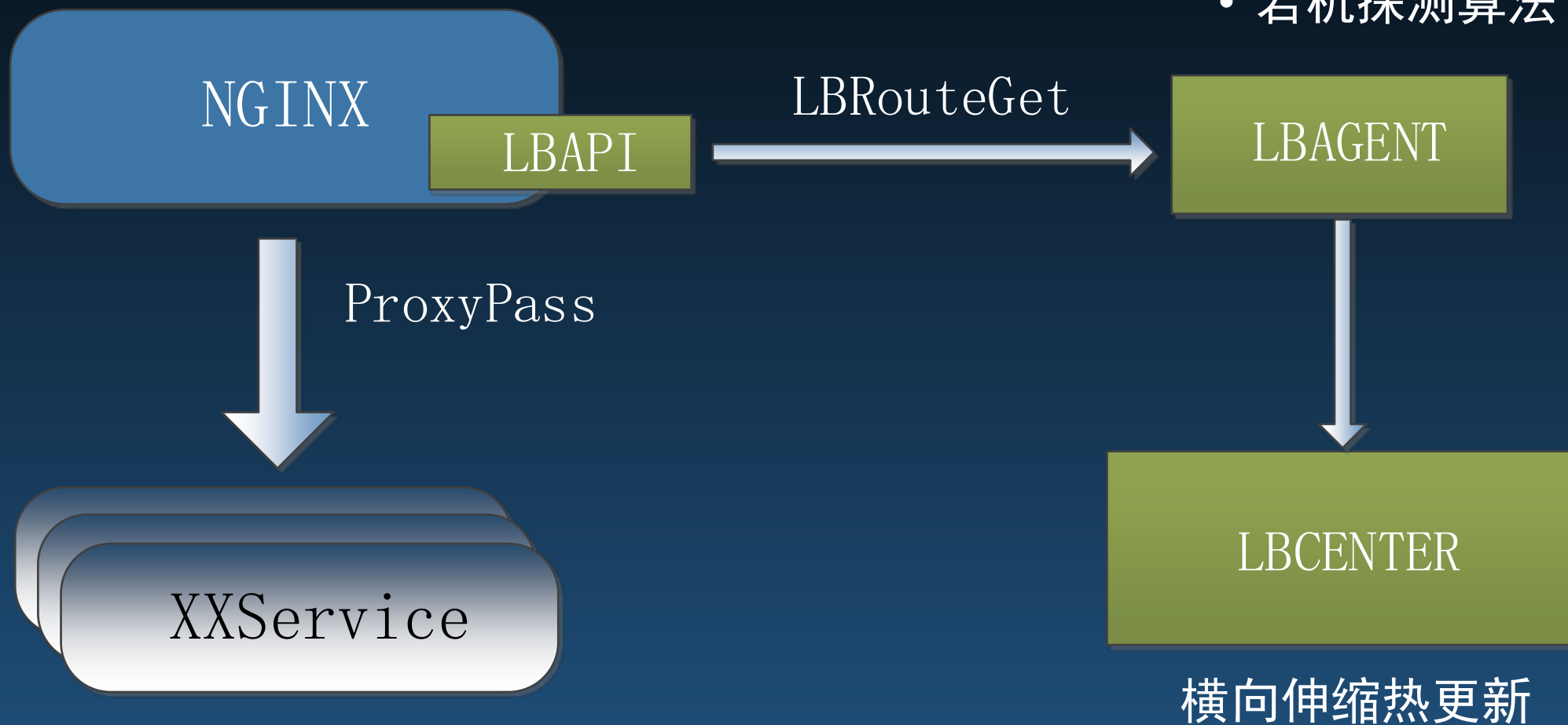
# 服务注册发现

- SNAME->IP:PORT
- SNAME->LBID
- LBID->IP:PORT



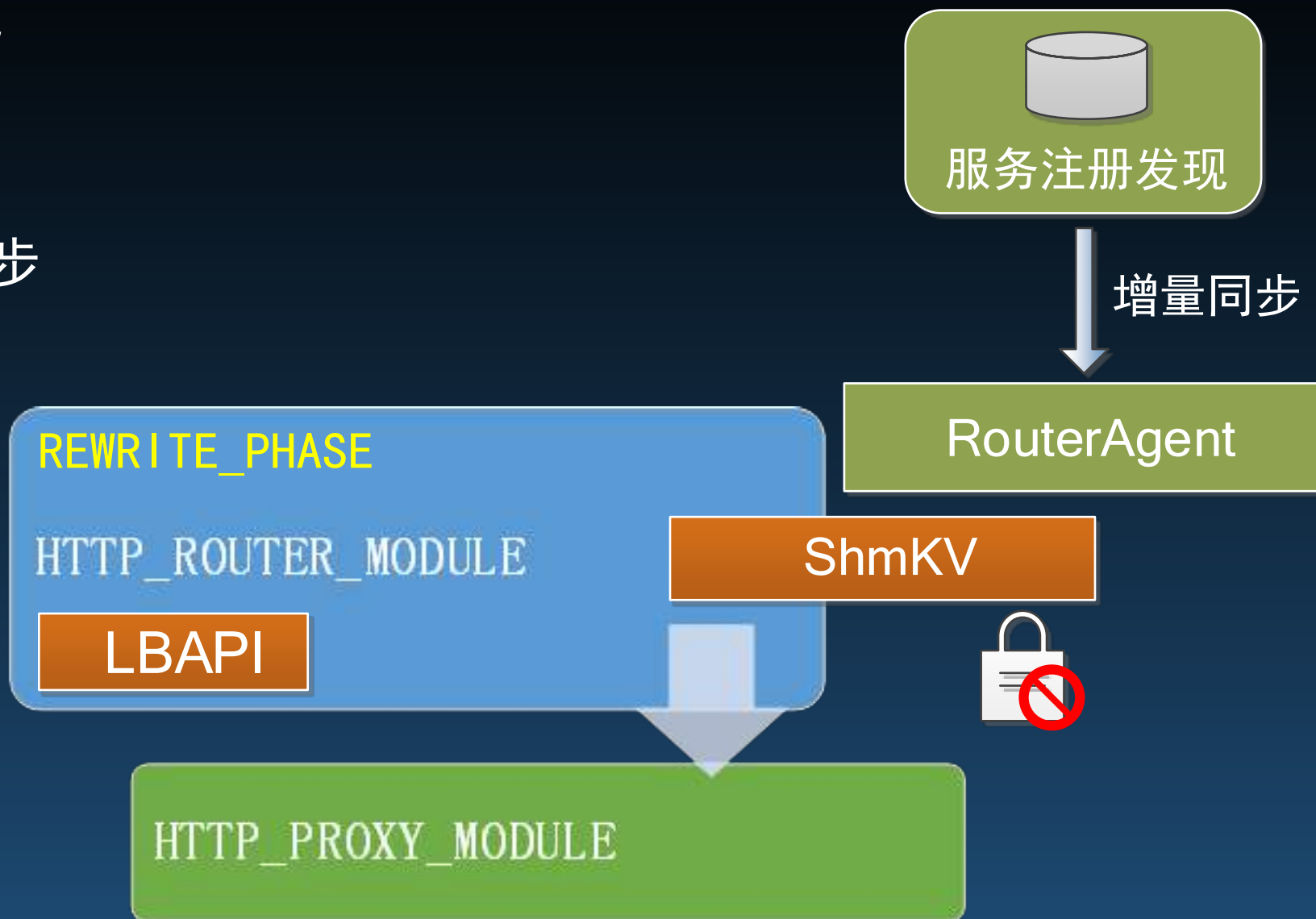
# 不只是寻址

对接公司级路由决策系统

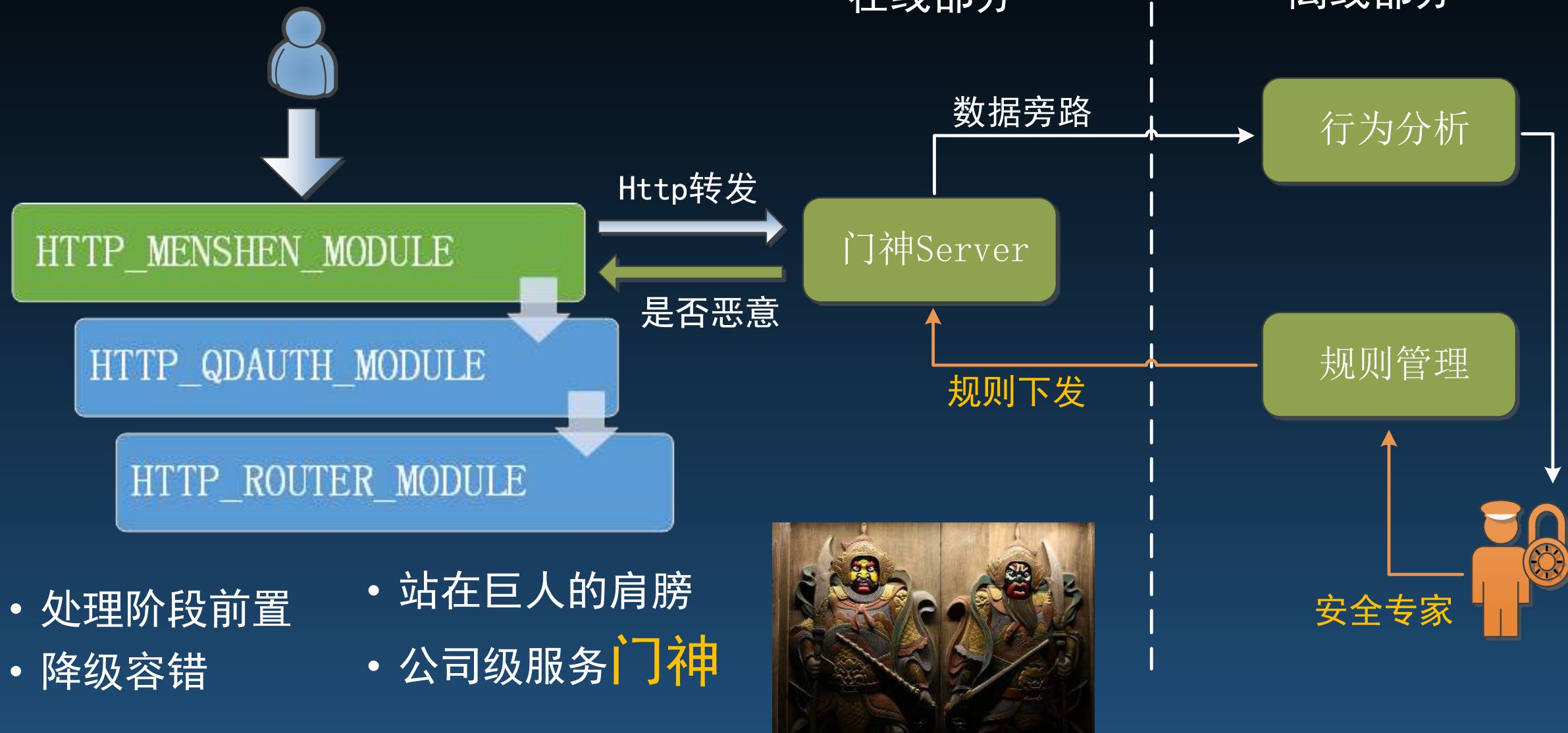


# 性能优化

- 路由表缓存
- 本地Agent同步

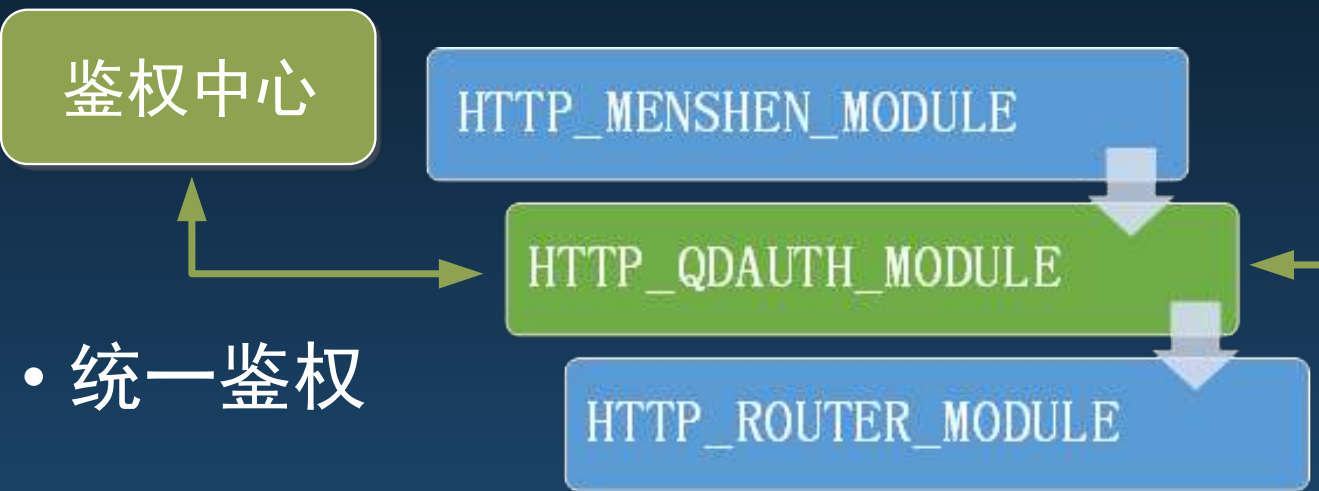


# 安全控制



# 安全控制

• 频控维度



• 统一鉴权



# 服务监控

上帝  
视角



服务  
画像

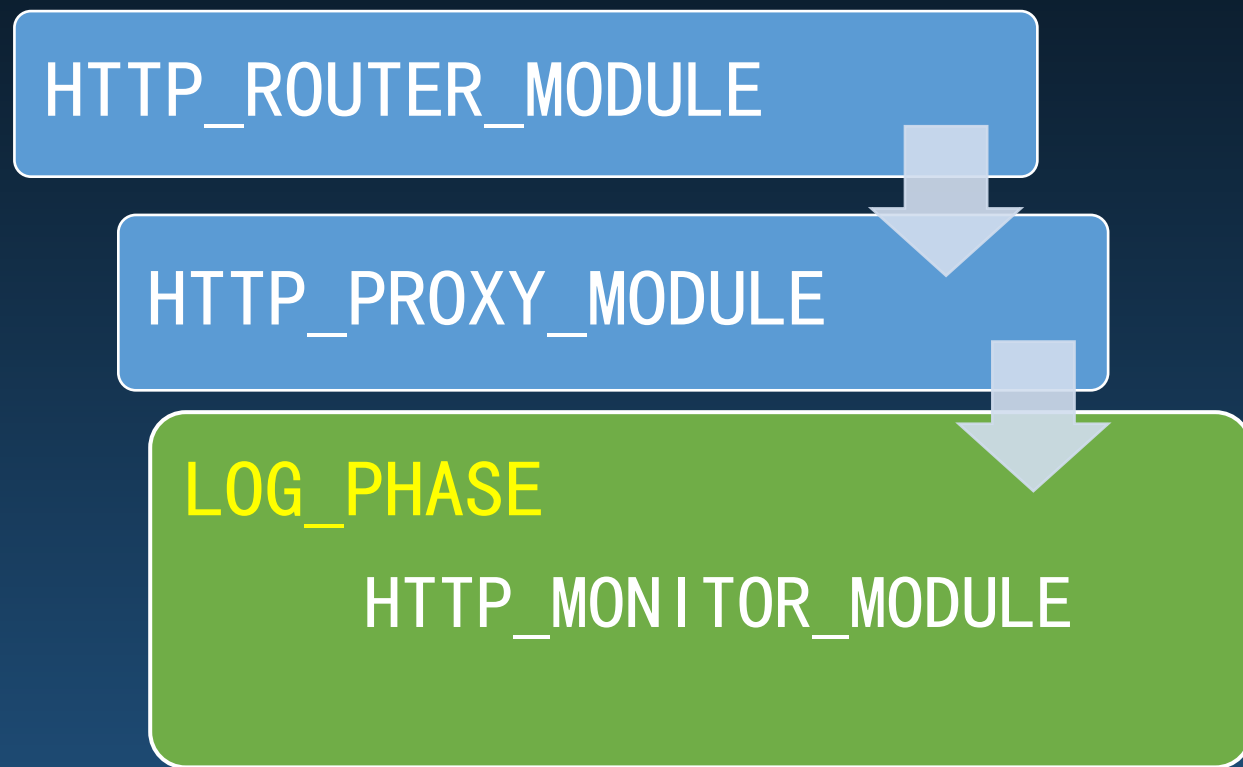
- 服务流量
- 服务质量
- 服务调用链
- 实时告警
- 大盘数据





# 服务监控

- 日志阶段, 天生的监控基因
- 对接公司平台



数据旁路



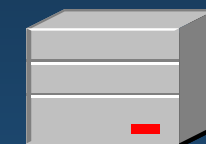
流量统计



性能统计



大盘分析



告警平台

2. 网关技术选型

1. 初代架构

3. 动态路由

5. 安全控制

4. 服务注册发现

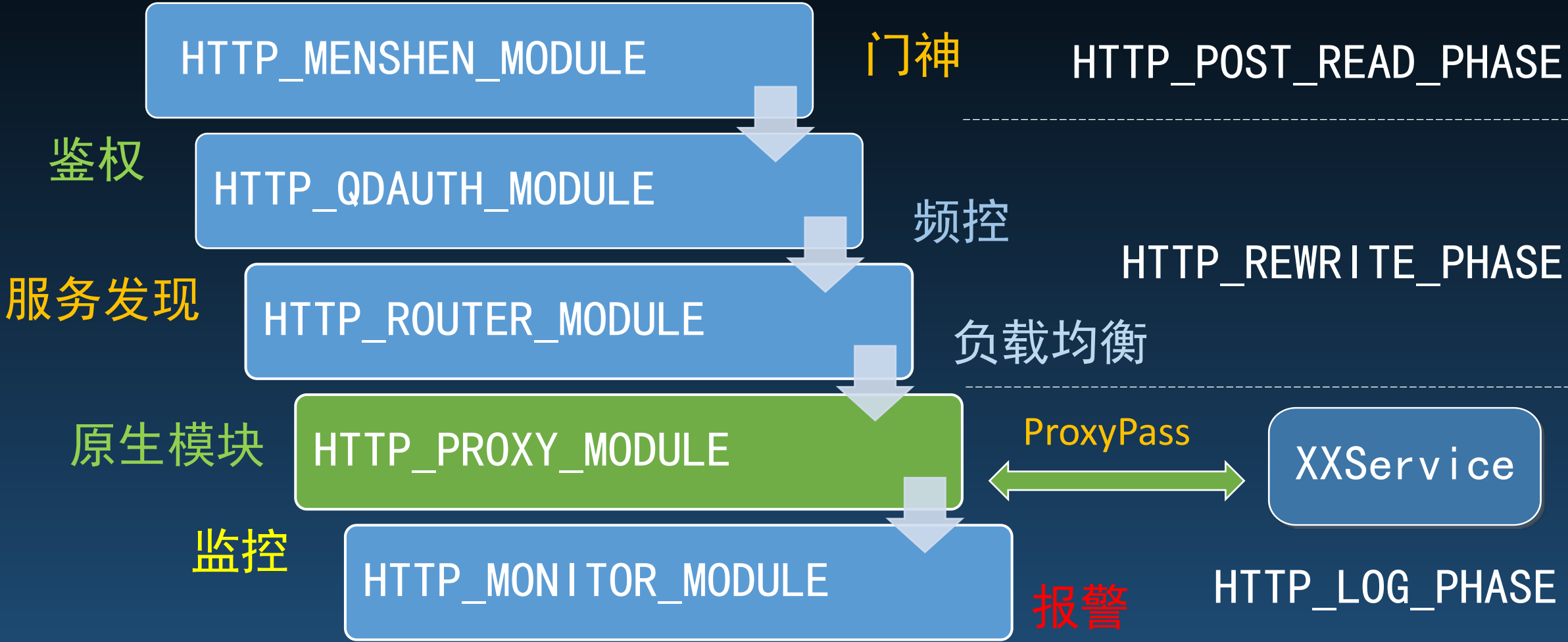
6. 服务监控

7. 整体回顾  
未来规划

**ESP**



# 整体回顾



# 未来规划





腾讯Web后台开发

JOIN US!

[romanzheng@tencent.com](mailto:romanzheng@tencent.com)