



全球

World Of Tech 2017

2017年12月1日-2日 • 深圳中洲万豪酒店

软件开发技术峰会

DEVELOPMENT



电商系统架构演进

守夜人

黄伟平

架构师

 目录

01

架构演进

02

守夜人

03

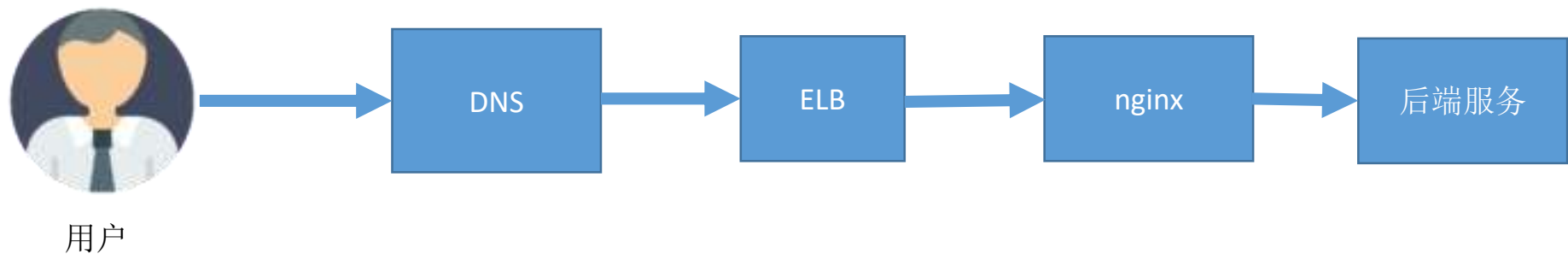
未来展望

04

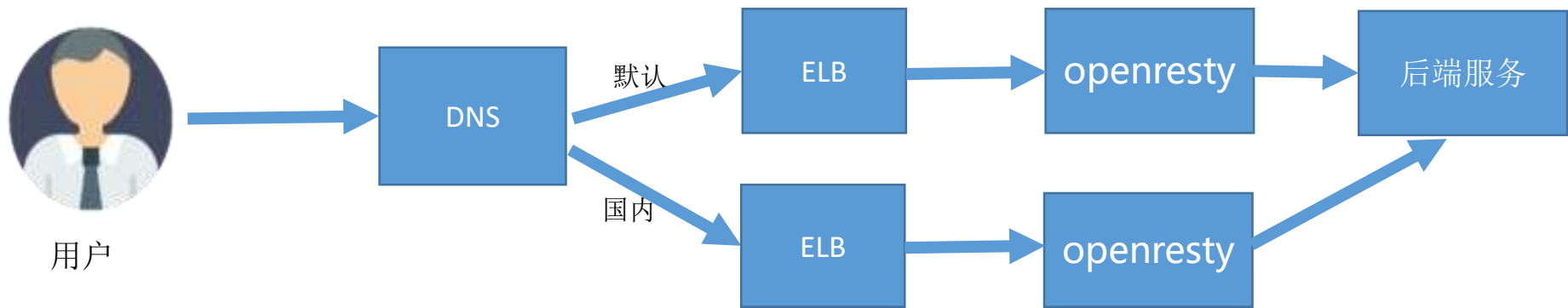
参考资料

05

01 web架构演进-v1



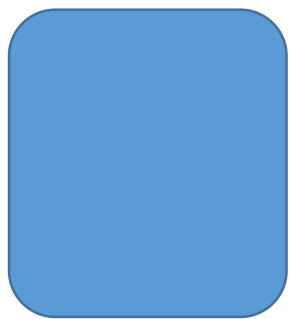
01 web架构演进-v2



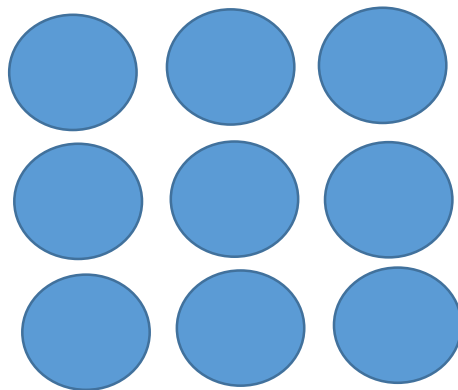
01 项目背景

- ◆ 业务需求
- ◆ 提升运维效率
- ◆ 限流
- ◆ AB组灰度功能
- ◆ 证书如何统一管理
- ◆ 微服务统一入口

01 项目背景-微服务



单体架构



微服务

守夜人 = Kong

02 守夜人

--- 《冰与火之歌》 中一个组织



我们是长城上的守卫
We are the watchers on the Wall.

02 守夜人

◆ 内部结构

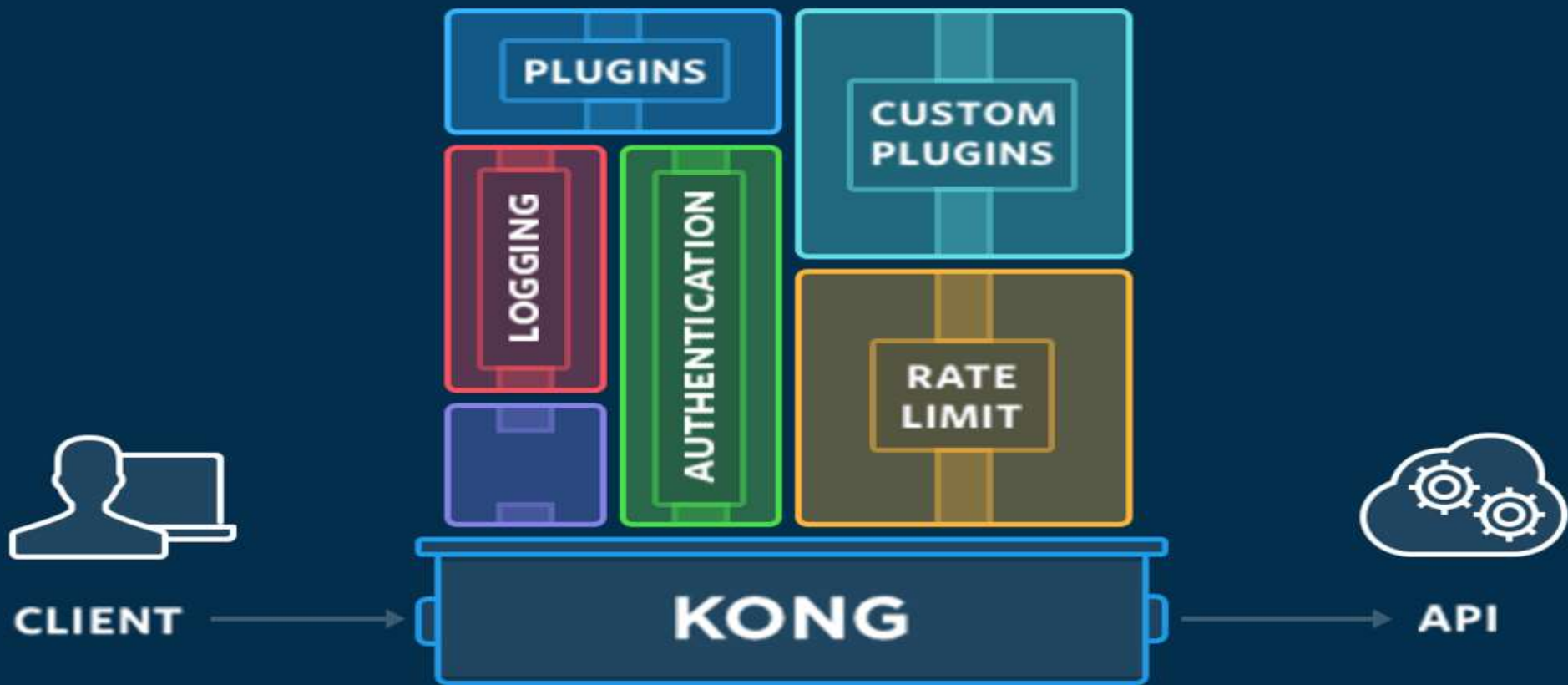
◆ 优势

◆ 部署

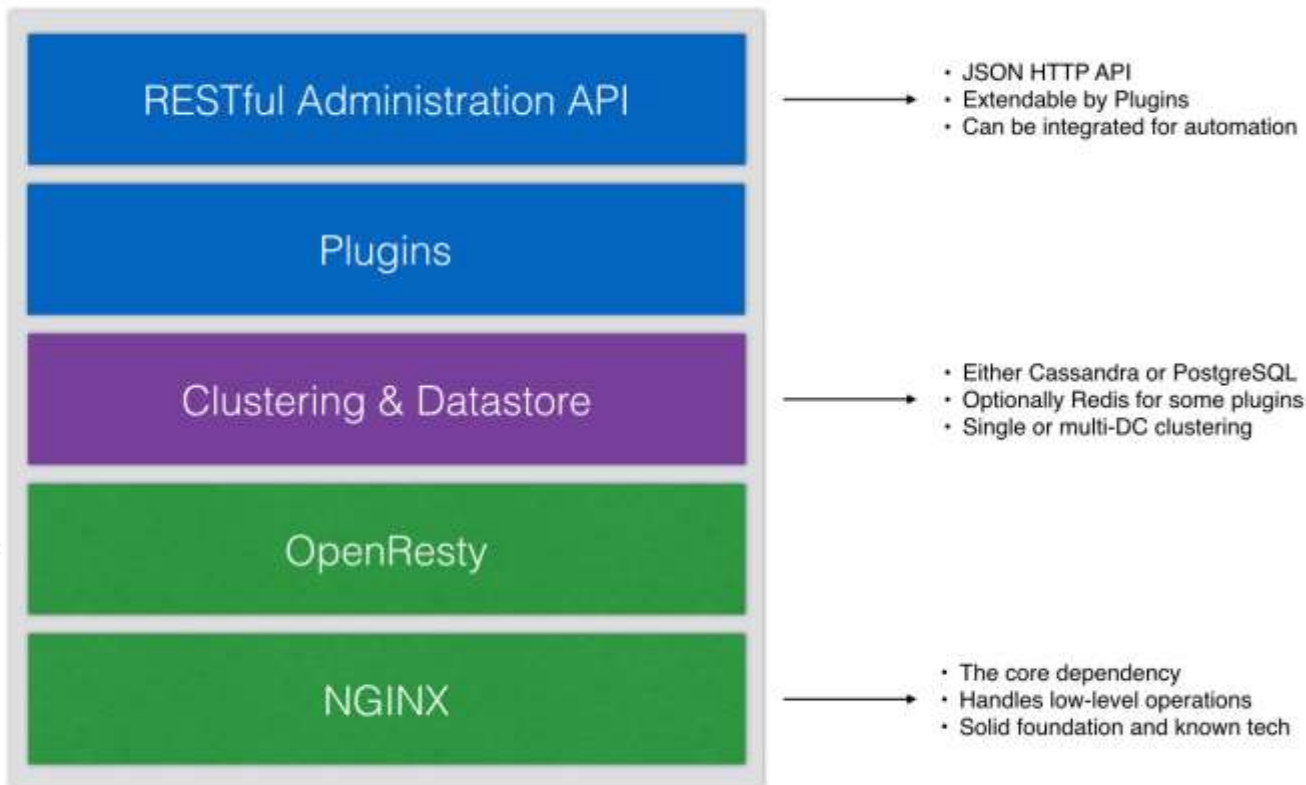
◆ 插件

◆ 后台管理

02 守夜人-内部结构

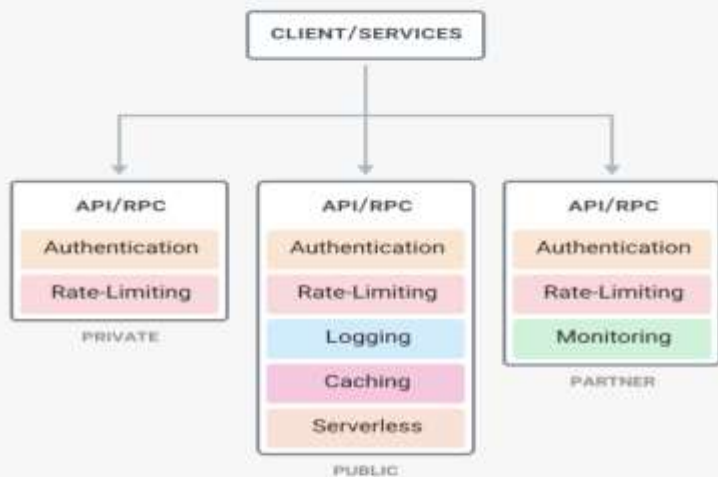


02 守夜人-内部结构



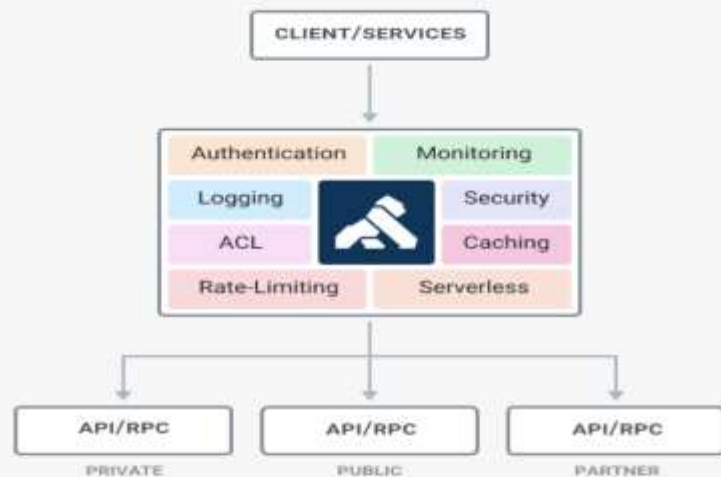
02 守夜人-优势

The Redundant Old Way



- ✘ Common functionality is duplicated across multiple services
- ✘ Systems tend to be monolithic and hard to maintain
- ✘ Difficult to expand without impacting other services
- ✘ Productivity is inefficient because of system constraints

The Kong Way

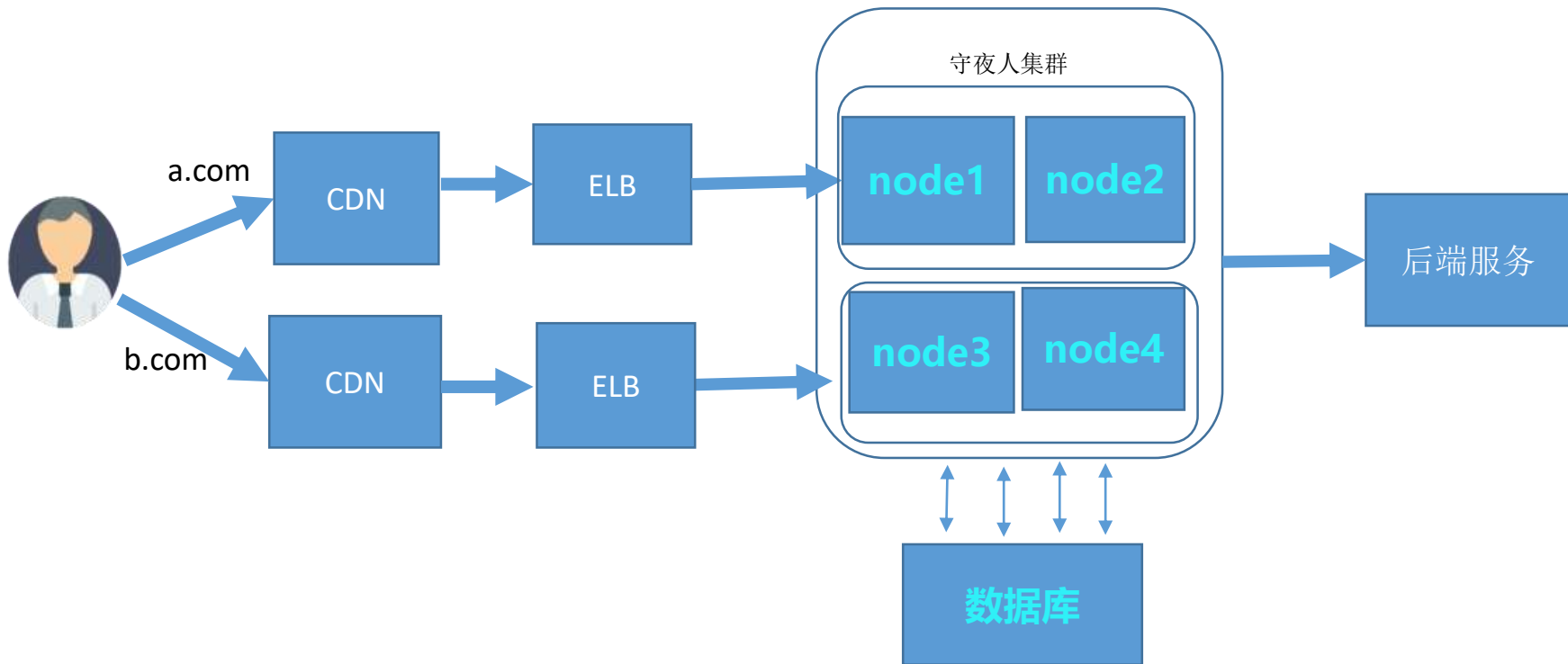


- ✔ Kong orchestrates common functionality
- ✔ Build efficient distributed architectures ready to scale
- ✔ Expand functionality from one place with a simple command
- ✔ Focus on your product and let Kong do the REST

02 守夜人-优势

传统模式	守夜人
公共功能在多个服务中重复	将功能整合到一体
系统往往是一体的，难以维护	分布式架构，容易扩展
各个系统Nginx配置不标准，生产力效率低下	插件化，新业务规则开发新的插件
Https证书部署在各个系统	证书统一管理
各个系统单独配置	运维方便，后台统一修改

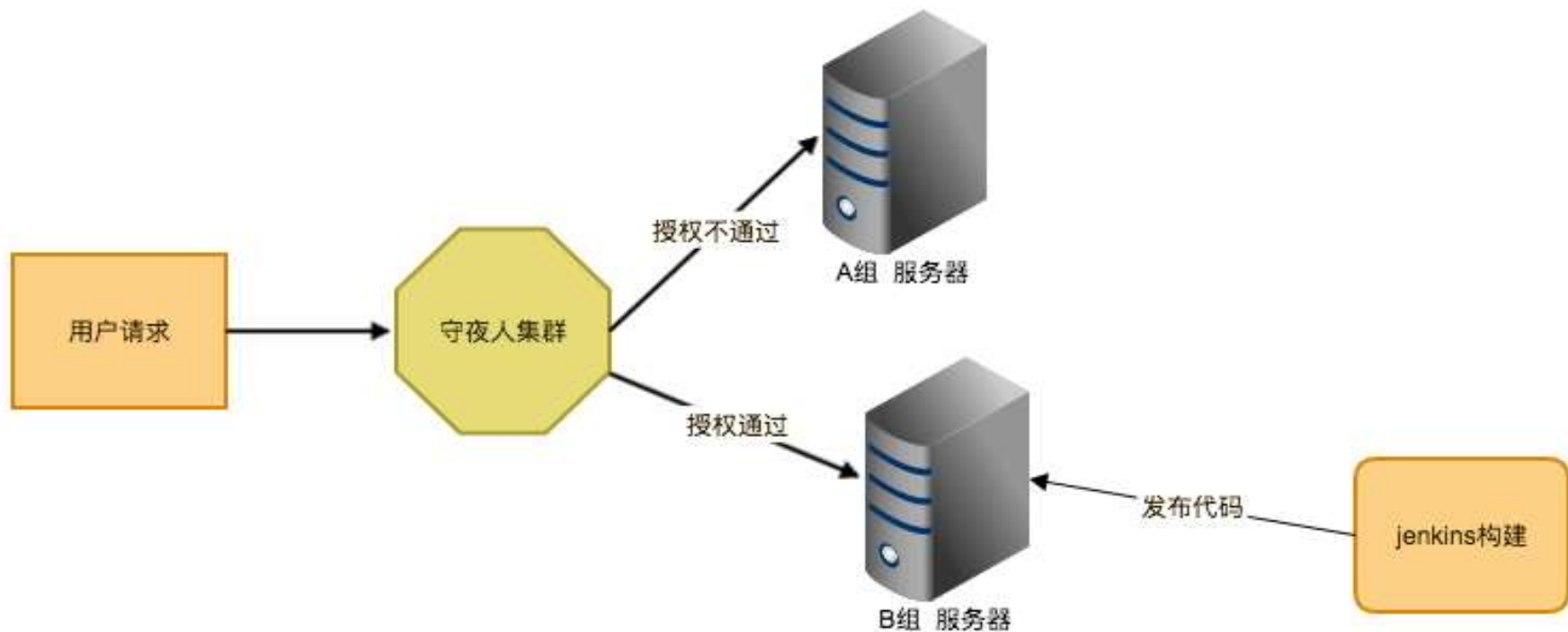
02 守夜人-部署



02 守夜人-插件

- ◆ 认证 (basic-authenticatio,Key Auth,Oauth2等)
- ◆ 安全 (ACL , CORS,IP Restriction等)
- ◆ 流控 (Rate Limiting, Response Rate Limiting, Request Size Limiting)
- ◆ 日志 (TCP/UDP/HTTP Logging, File Logging, Syslog, StatsD, Loggly)
- ◆ 分析监控 (Galileo, Datadog, Runscope)
- ◆ 内容转换 (Request Transformer, Response Transformer, Correlation ID)
- ◆ 自定义 (ab组灰度 , URL地址定时公开等)

02 守夜人-插件-AB组灰度插件



02 守夜人-后台管理

- ◆ APIS (新增, 修改等)
- ◆ 管理多个守夜人集群
- ◆ 管理守夜人插件 (开启, 关闭等)
- ◆ 守夜人运行节点监控
- ◆ 多个用户
- ◆ 权限管理

02

守夜人-后台管理-APIs

Name

(optional)

The API name. If none is specified, will default to the `request_host` or `request_path`.

Hosts

(semi-optional)

A comma-separated list of domain names that point to your API. For example: `example.com`. At least one of `hosts`, `uris`, or `methods` should be specified.

Uris

(semi-optional)

A comma-separated list of URIs prefixes that point to your API. For example: `/my-path`. At least one of `hosts`, `uris`, or `methods` should be specified.

Methods

(semi-optional)

A comma-separated list of HTTP methods that point to your API. For example: `GET`, `POST`. At least one of `hosts`, `uris`, or `methods` should be specified.

Upstream URL

The base target URL that points to your API server, this URL will be used for proxying requests. For example, `http://example.com`.

Strip uri

*(optional)*YES

When matching an API via one of the `uris` prefixes, strip that matching prefix from the upstream URI to be requested. Default: `true`.

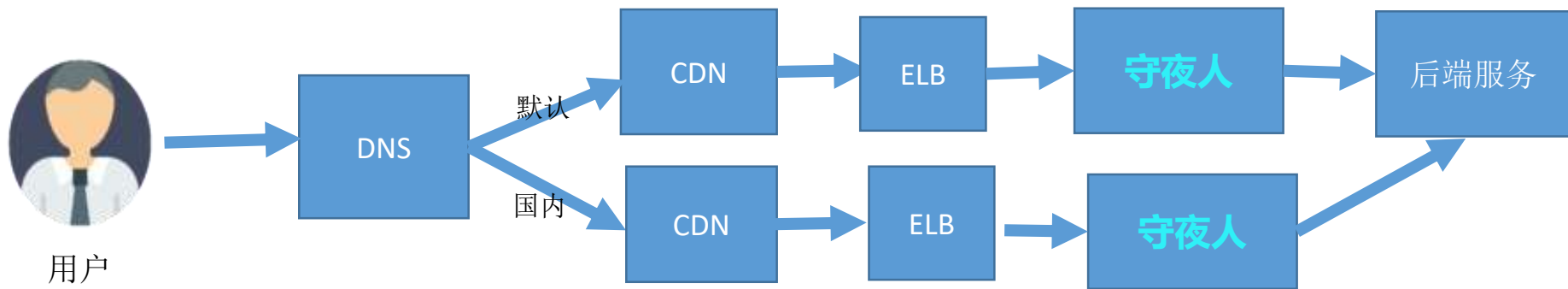
02

守夜人-后台管理-APIS

Preserve Host <i>(optional)</i>	<input type="checkbox"/> NO	When matching an API via one of the <code>hosts</code> domain names, make sure the request <code>Host</code> header is forwarded to the upstream service. By default, this is <code>false</code> , and the upstream <code>Host</code> header will be extracted from the configured <code>upstream_url</code> .
Retries <i>(optional)</i>	<input type="text" value="5"/>	The number of retries to execute upon failure to proxy. The default is <code>5</code> .
Upstream connect timeout <i>(optional)</i>	<input type="text" value="60000"/>	The timeout in milliseconds for establishing a connection to your upstream service. Defaults to <code>60000</code> .
Upstream send timeout <i>(optional)</i>	<input type="text" value="60000"/>	The timeout in milliseconds between two successive write operations for transmitting a request to your upstream service. Defaults to <code>60000</code> .
Upstream read timeout <i>(optional)</i>	<input type="text" value="60000"/>	The timeout in milliseconds between two successive read operations for transmitting a request to your upstream service. Defaults to <code>60000</code> .
Https only <i>(optional)</i>	<input type="checkbox"/> NO	To be enabled if you wish to only serve an API through HTTPS, on the appropriate port (<code>8443</code> by default). Default: <code>false</code> .
Http if terminated <i>(optional)</i>	<input checked="" type="checkbox"/> YES	Consider the <code>X-Forwarded-Proto</code> header when enforcing HTTPS only traffic. Default: <code>true</code> .

✓ SUBMIT API

02 web架构演进-v3



02 守夜人-遇到的坑

- ◆ Upstream connect timeout 设置不合理
- ◆ Preserve Host未开启，域名无法带到后端，导致后端报错
- ◆ 用户真实ip无法获取
- ◆ A域名接入守夜人，B域名Cname到A域名，导致B域名报错

03 未来展望

- ◆ 自动降级插件
- ◆ 安全插件
- ◆ Dashboard
- ◆ 日志分析和处理
- ◆ 容器化

04 参考资料

◆ <https://openresty.org>

◆ <https://getkong.org/>

◆ <https://github.com/openresty/lua-nginx-module#name>

◆ <https://github.com/pantse/konga>

Q&A

Thank you!