

WOTA

51CTO

World Of Tech 2017

全球架构与运维技术峰会

2017年4月14日-15日 北京富力万丽酒店

ARCHITECTURE



出品人及主持人：

**邓钦华** 蘑菇街 技术总监

---

电商大促背后的技术挑战



苏宁易购

全站HTTPS

实践之路



# 朱羿全

苏宁云商

IT总部架构师

分享主题：

苏宁易购全站HTTPS实践之路：  
如何做到兼顾安全与性能



为什么我们要使用HTTPS ? 01

苏宁易购全站HTTPS方案概述 02

HTTPS系统改造篇 03

HTTPS性能优化篇 04

HTTPS灰度上线篇 05

HTTPS未来展望篇 06

# 01

为什么我们要使用HTTPS

# HTTPS是互联网发展的大势所趋



## 目前，多个组织在加速推进HTTPS的部署进程



chrome

谷歌启动了

[Deprecating Powerful Features on Insecure Origins](#)计划，今后部分

涉及用户隐私数据的API必须在安全环境（Secure Contexts）中才能使用。



App Store

苹果公司将强制所有

AppStore中的应用实行

[App Transport Security\(ATS\)](#)标准，否则将拒绝应用上架。



mozilla

Firefox

Mozilla 公司在一年前也明确表态会逐步淘汰不安全的 HTTP，详见：

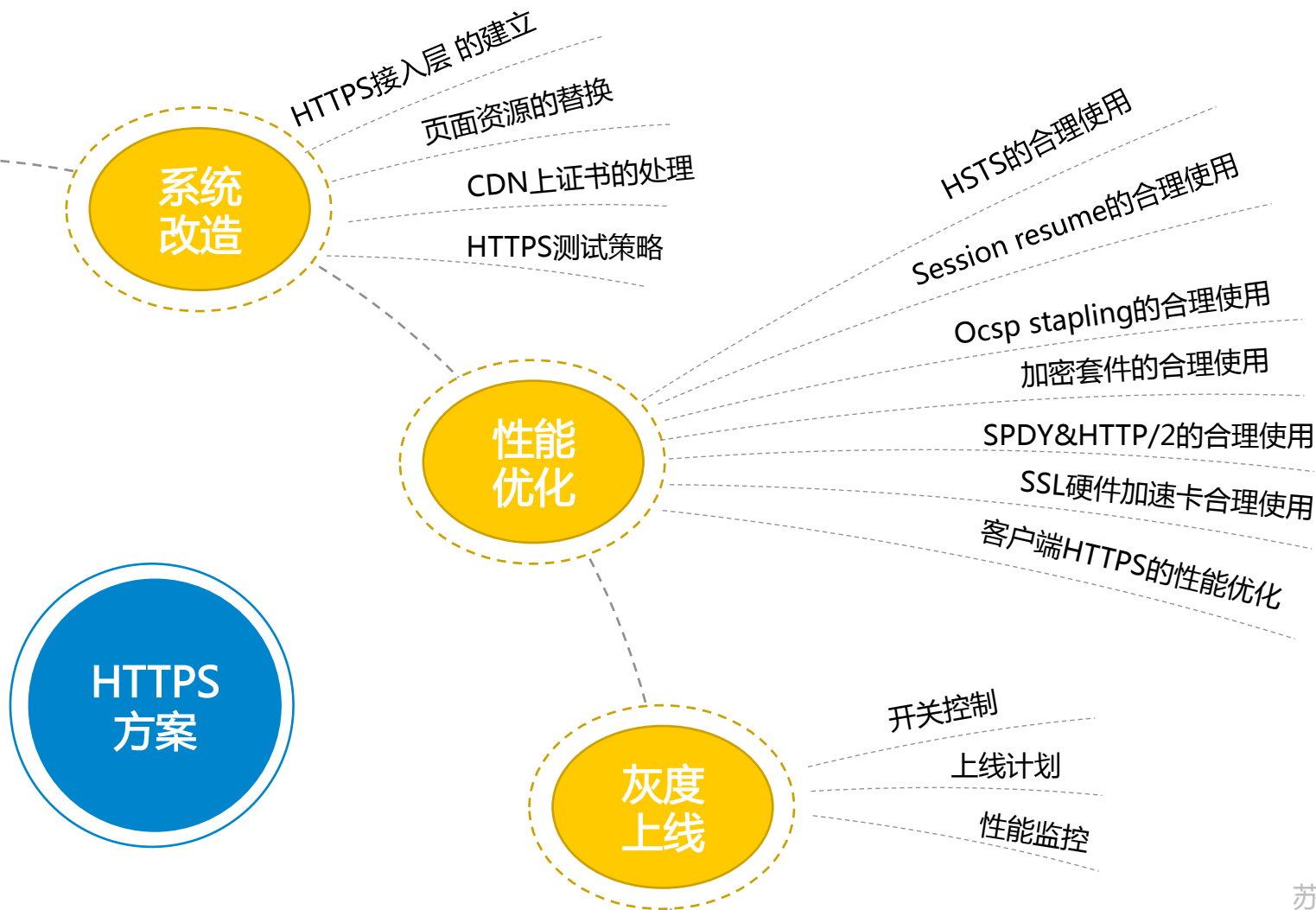
[Deprecating Non-Secure HTTP](#)。



# 02

## 苏宁易购全站HTTPS方案概述

# 苏宁易购全站HTTPS方案概述



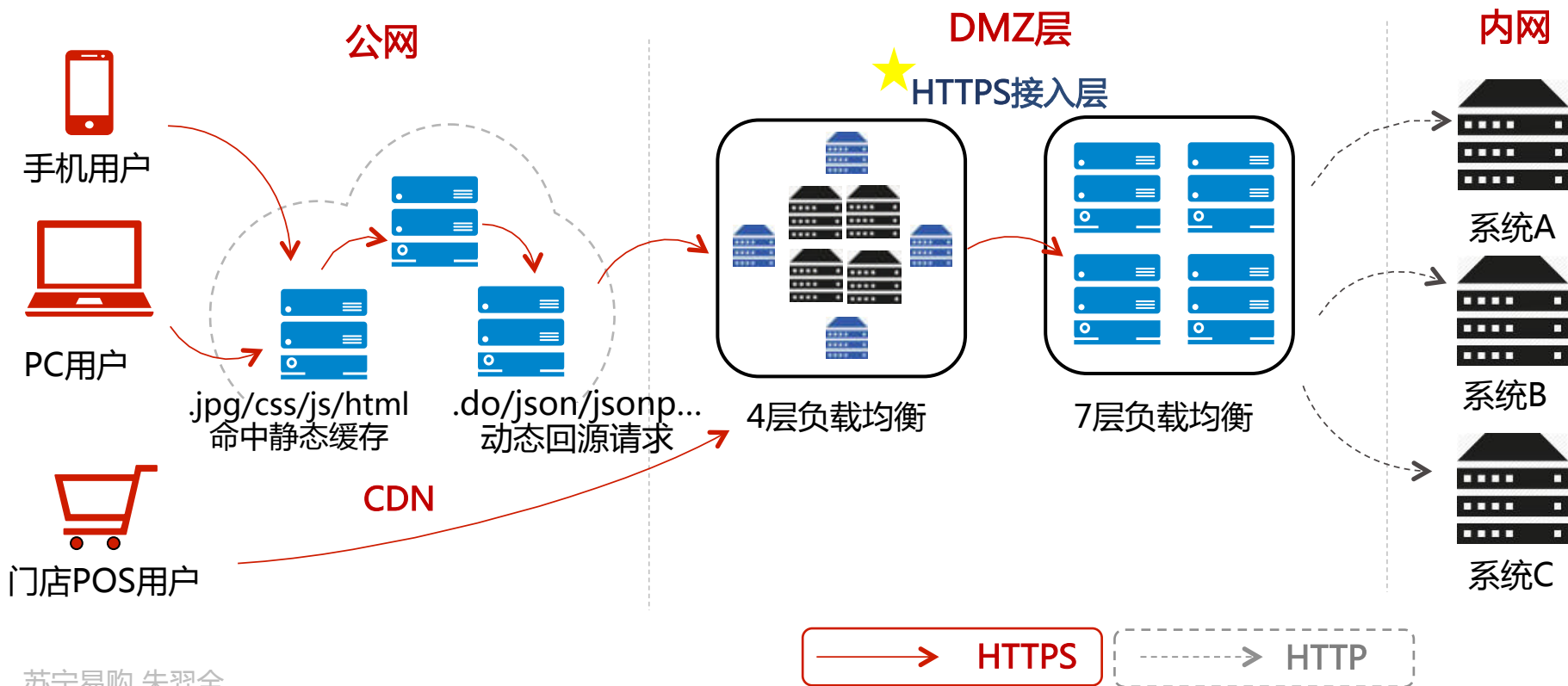
# 03

## HTTPS系统改造篇



## HTTPS接入层定义

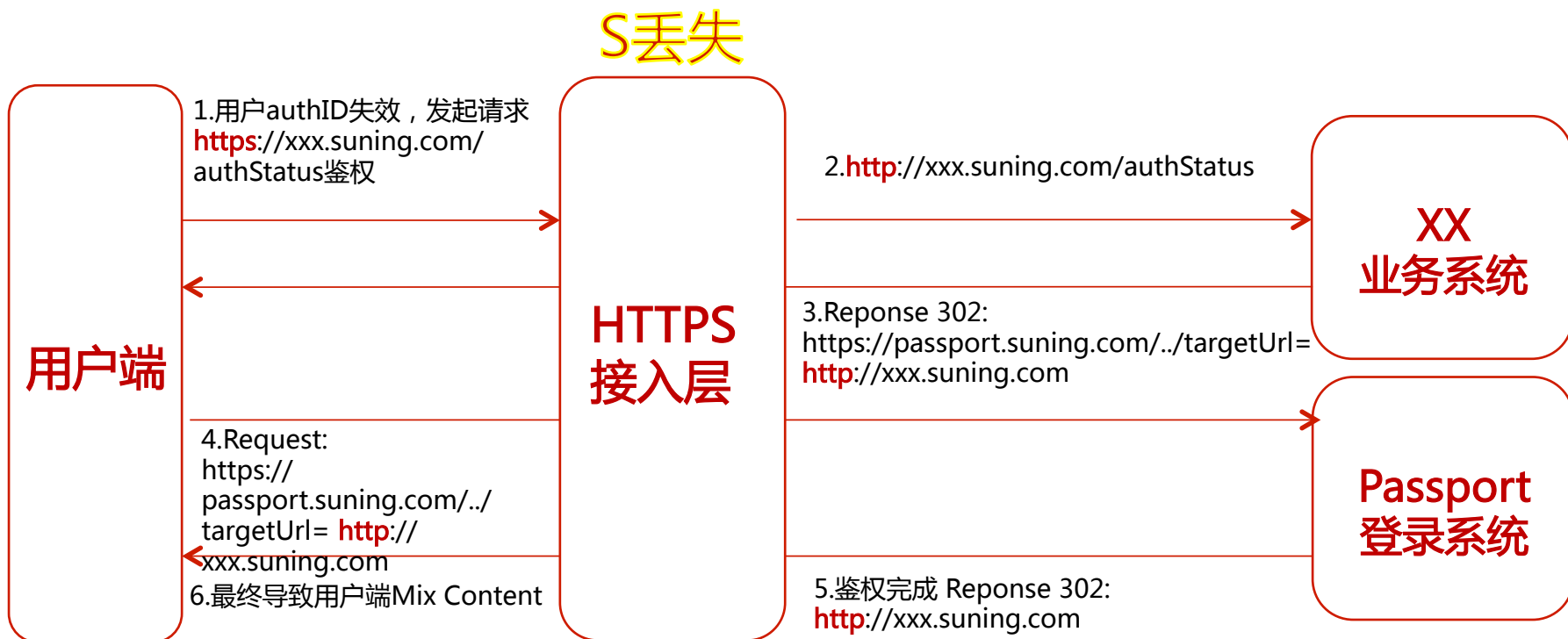
- 尽早完成SSL握手
- 统一接入与调度，业务系统不需要做调整
- 统一优化与升级，提高接入层性能与安全性





## 页面资源替换

- 理解 Mixed Content
- //替换http://
- x-request-url的定义和使用



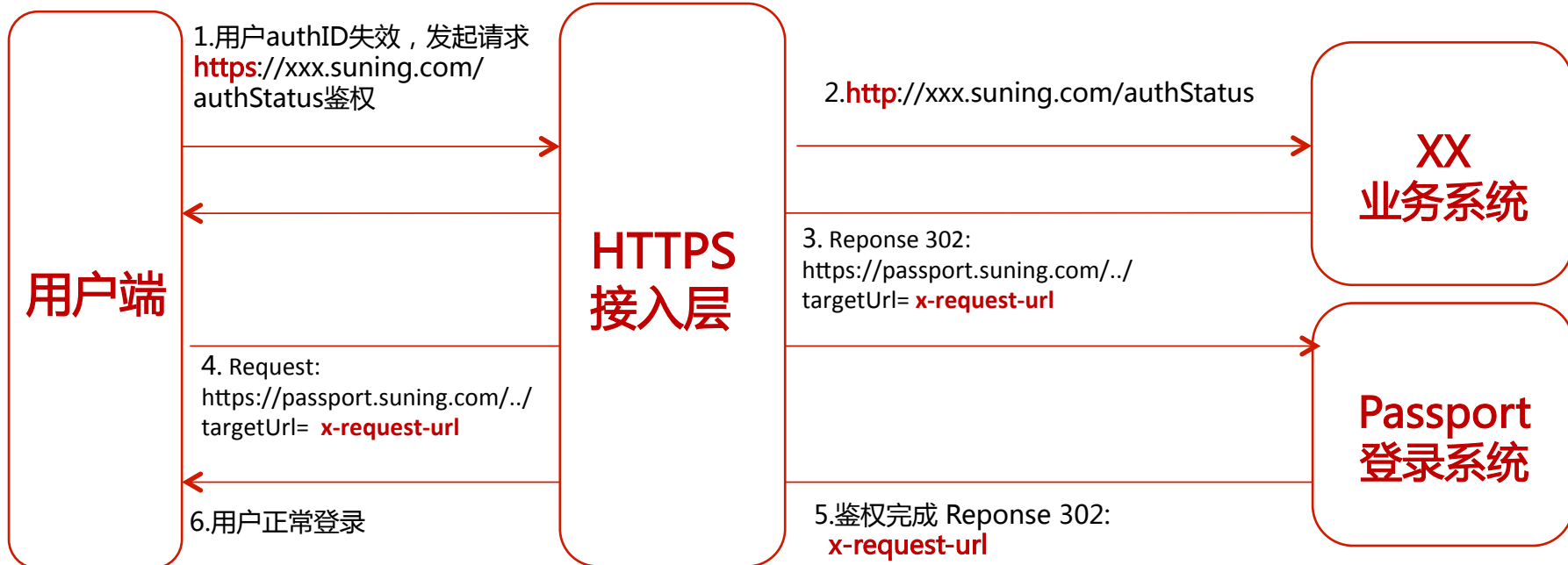




## 页面资源替换

- 理解 Mixed Content
- //替换http://
- x-request-url的定义和使用

### x-request-url 记录原始请求





## App原生无法识别//的问题



### Json报文

```
wapData(  
{  
    version: "1",  
    code: "200",  
    data:  
    {  
        ...  
        url1: "//image.suning.com/1.jpg",  
        url2: "//image.suning.com/2.jpg",  
        url3: "//image.suning.com/3.jpg",  
    }  
}
```



### 解析Json报文

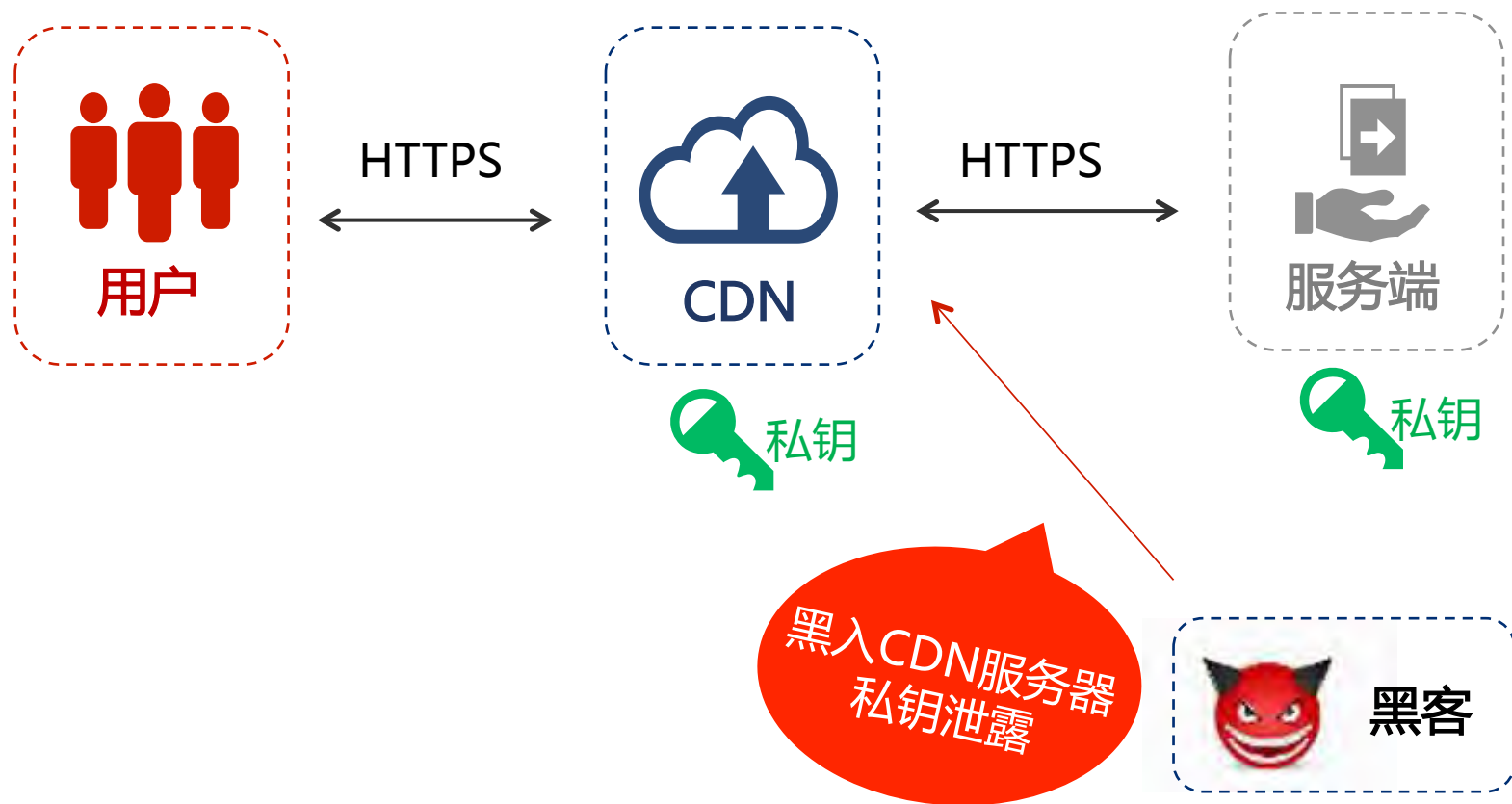
OkHttp

```
Request request = new Request.Builder().url(//  
image.suning.com/1.jpg).build()
```

Exception

## 如何处理商用CDN上的证书和私钥？

- 主动提供私钥给商用CDN厂商（HTTPS不再安全）



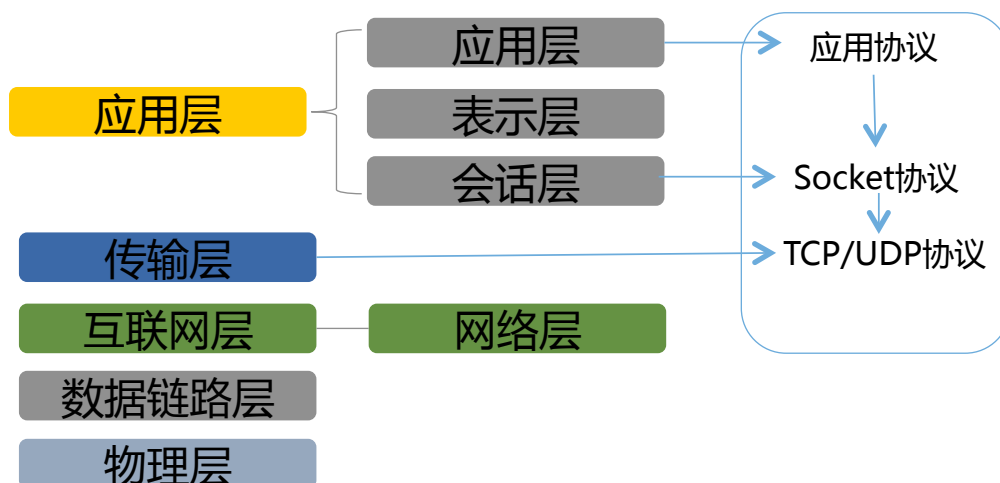
• 双证书策略 (治标不治本)



• 四层加速

cdn 进行 tcp 代理，不缓存内容。可支持任何基于TCP/UDP传输协议的上层协议。

适用于动态回源请求，比如加入购物车、提交订单、登录等



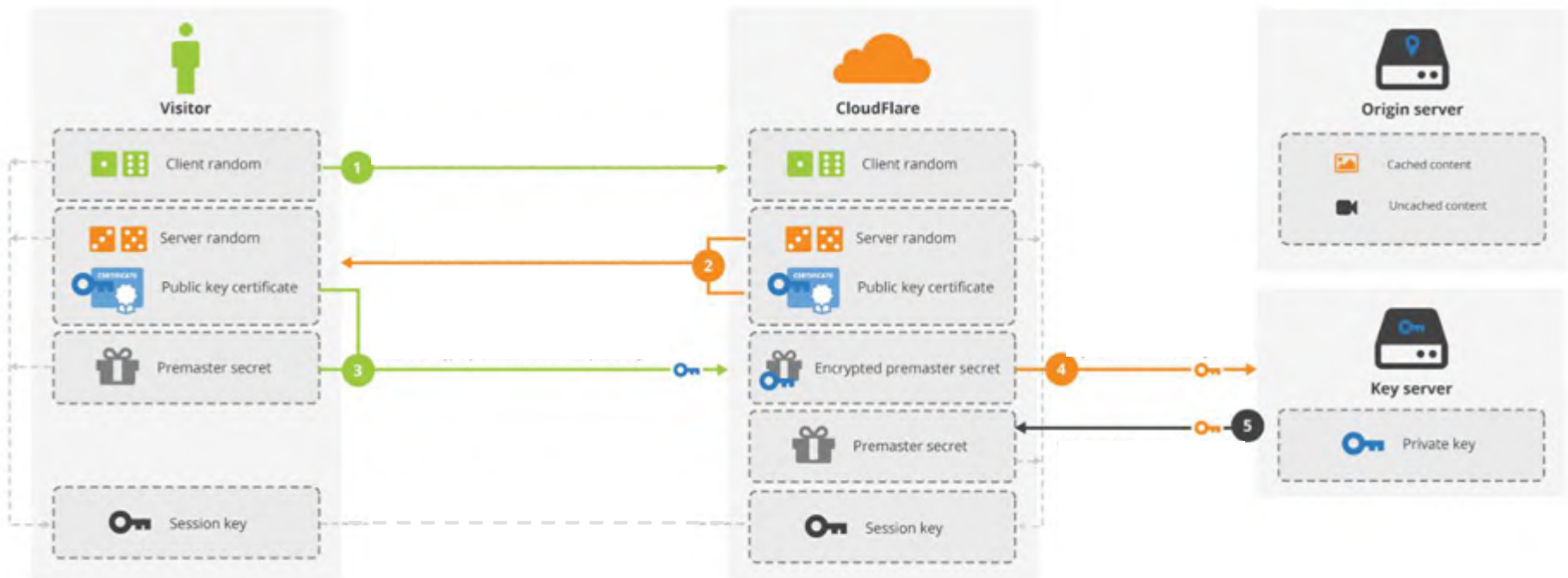
## Keyless解决方案

适用于金融，提供一台实时计算的 Key Server。CDN 要用到私钥时，通过加密通道将必要的参数传给 Key Server，由 Key Server 算出结果并返回即可。

<https://github.com/cloudflare/keyless>

### CloudFlare Keyless SSL (RSA)

Handshake





## HTTPS测试策略

STEP1



### 源码扫描

利用Jenkins遍历代码库，shell脚本扫描出http链接

STEP2



### 对页面爬虫扫描

爬虫脚本扫描遗漏的http链接

STEP3



### 测试环境验证

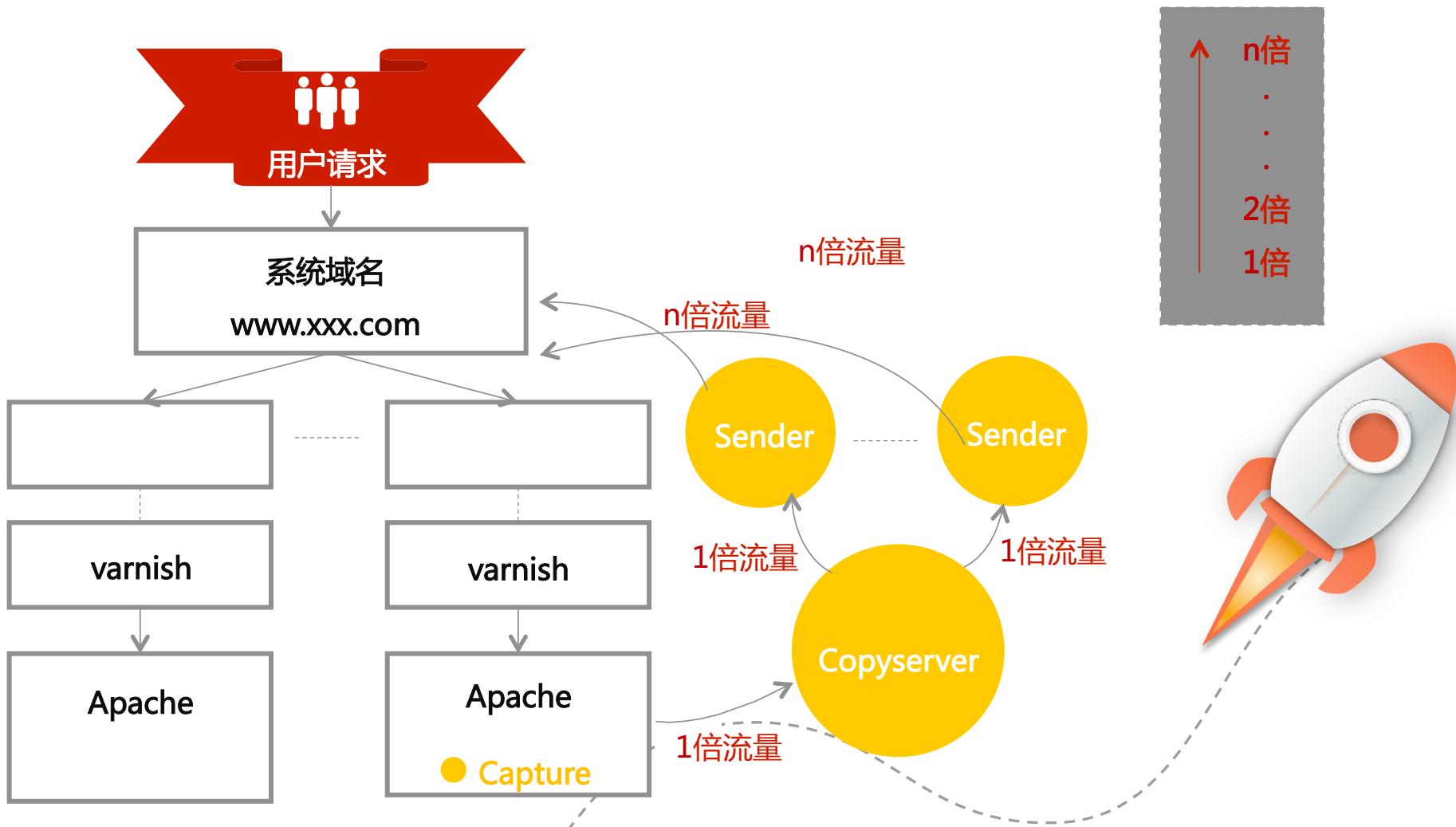
对核心主流程测试手工验证

STEP4



### 线上预发与引流测试

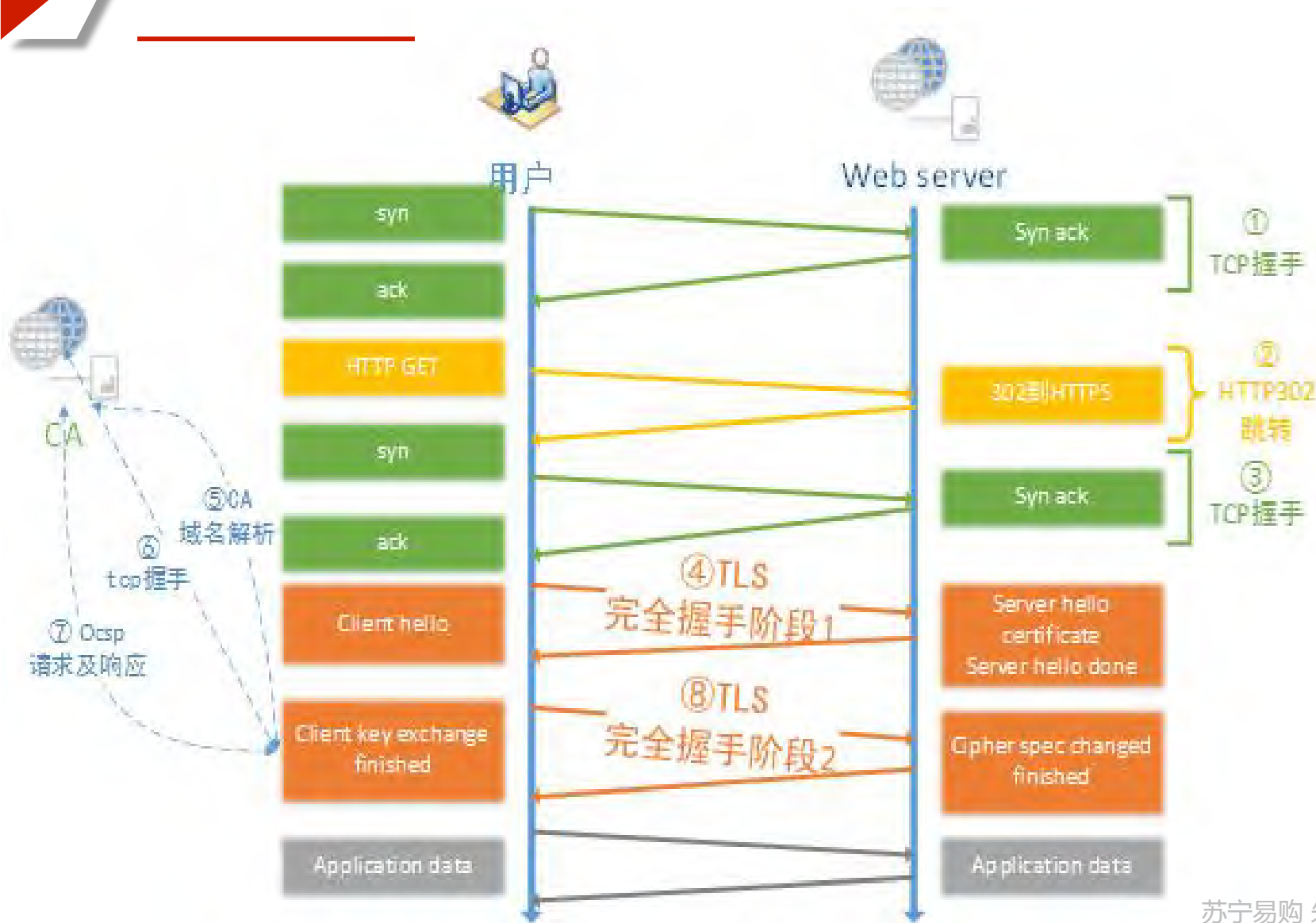
HTTPS发布到线上但不面向用户，引流用户真实流量进行测试



# 04

## HTTPS性能优化篇

# TLS握手流程





## HSTS的合理使用

```
Strict-Transport-Security : max-age=expireTime [; includeSubDomains] [; preload]
```

### 优势

减少了HTTP做302跳转的开销。302跳转不仅暴露了用户的访问站点，也很容易被中间者支持（降级劫持、中间人攻击），最重要是降低了访问速度（影响性能）。



#### 您的连接不是私密连接

攻击者可能会试图从\_\_\_\_\_窃取您的信息（例如：密码、通讯内容或信用卡信息）。NET:ERR\_CERT\_AUTHORITY\_INVALID

了解详情

重新加载

通常使用加密技术来保护您的信息。Chrome 此次尝试连接到\_\_\_\_\_时，此网站返回了异常的错误凭据。这可能是因为有攻击程序在试图冒充，或 Wi-Fi 登录屏幕中断了此次连接。请放心，您的信息仍然是安全的，因为 Chrome 尚未进行任何数据交换便停止了连接。

您目前无法访问\_\_\_\_\_，因为此网站使用了HSTS。网络错误和攻击通常是暂时的，因此，此网页稍后可能会恢复正常。

1. HSTS在max-age过期时间内在客户端是强制HTTPS的，服务端无法控制。因此，需要降级时，HTTPS无法及时切换到HTTP。
2. HSTS是严格的HTTPS，一旦网络证书错误时，网页将直接无法访问（用户无法选择忽视）。

### 缺点





## Session resume的合理使用

- Session ID ( RFC 5246 )  
ssl\_session\_fetch\_by\_lua\_block  
[https://github.com/openresty/lua-nginx-module#ssl\\_session\\_fetch\\_by\\_lua\\_block](https://github.com/openresty/lua-nginx-module#ssl_session_fetch_by_lua_block)

- Session tickets ( RFC 5077 )  
此功能需要开启前向性加密支持的密钥套件例如  
ECDHE-RAS-AES128-SHA256来提高安全性。

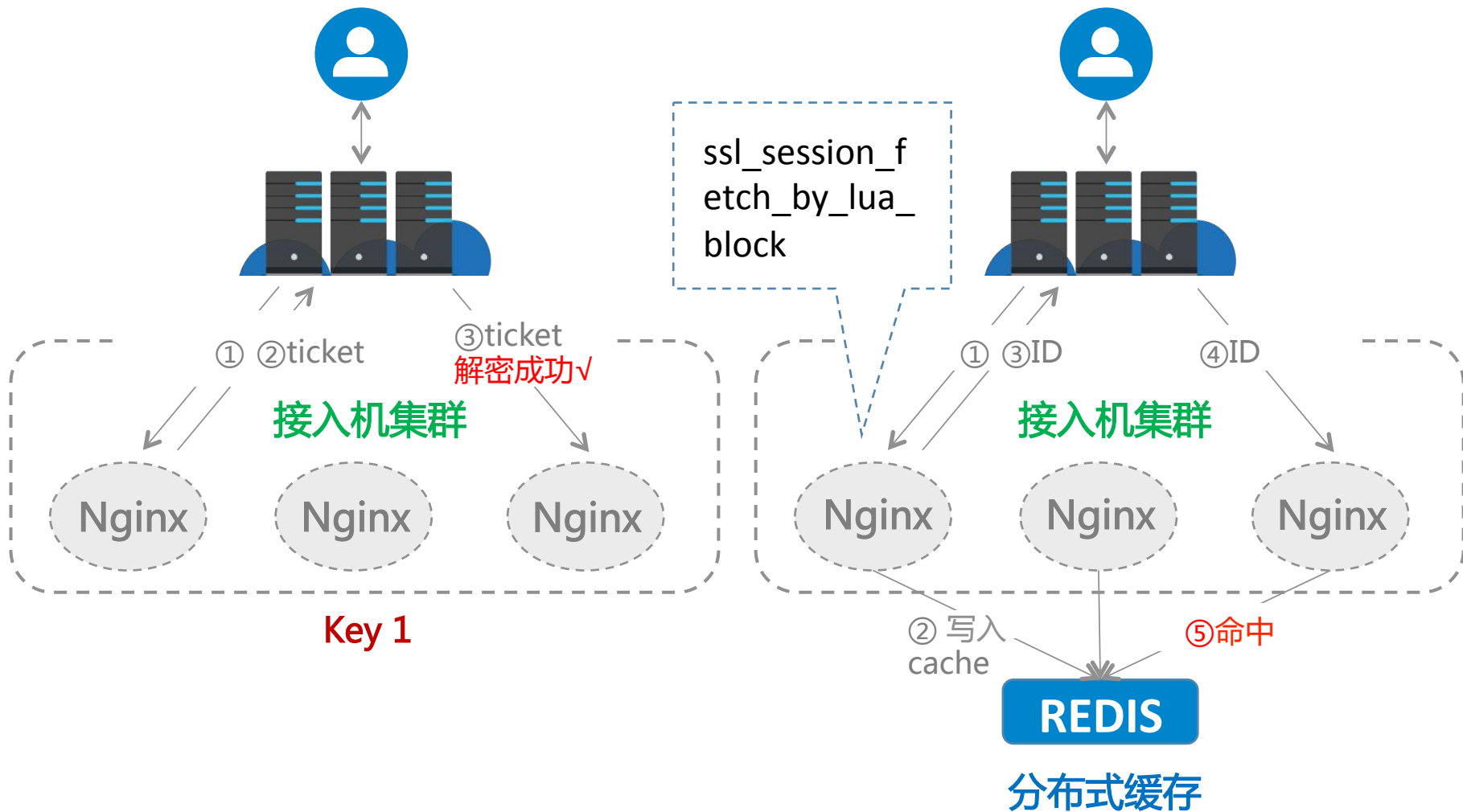
为了保持安全性session\_ticket.key需要经常保持更换。

```
ssl_session_tickets on;  
ssl_session_ticket_key */ticket.key;
```





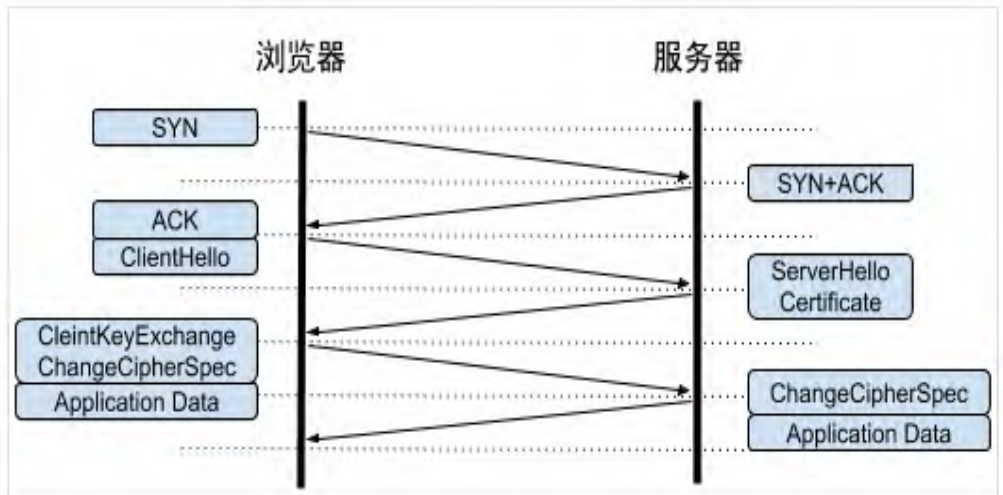
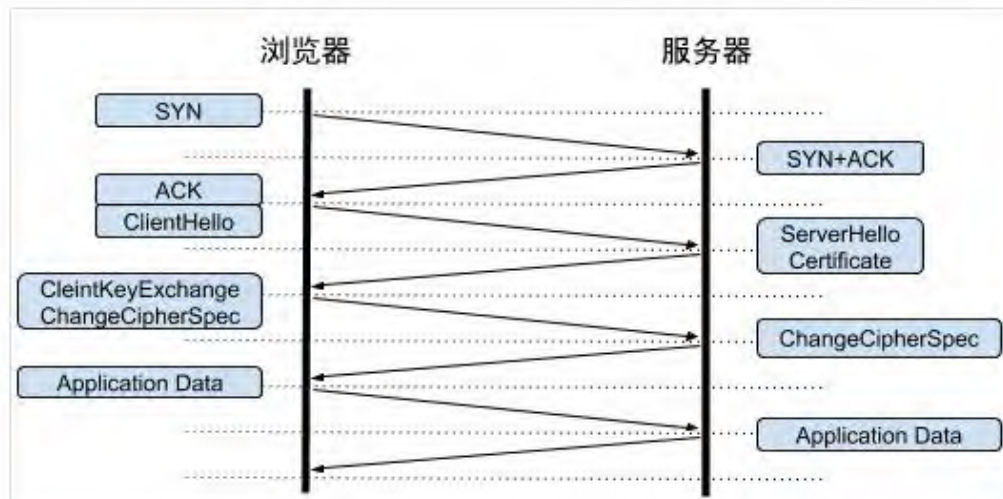
# Session resume的合理使用



## False Start的合理使用

- 支持NPN/ALPN
- 支持前向安全  
( Forward Secrecy )

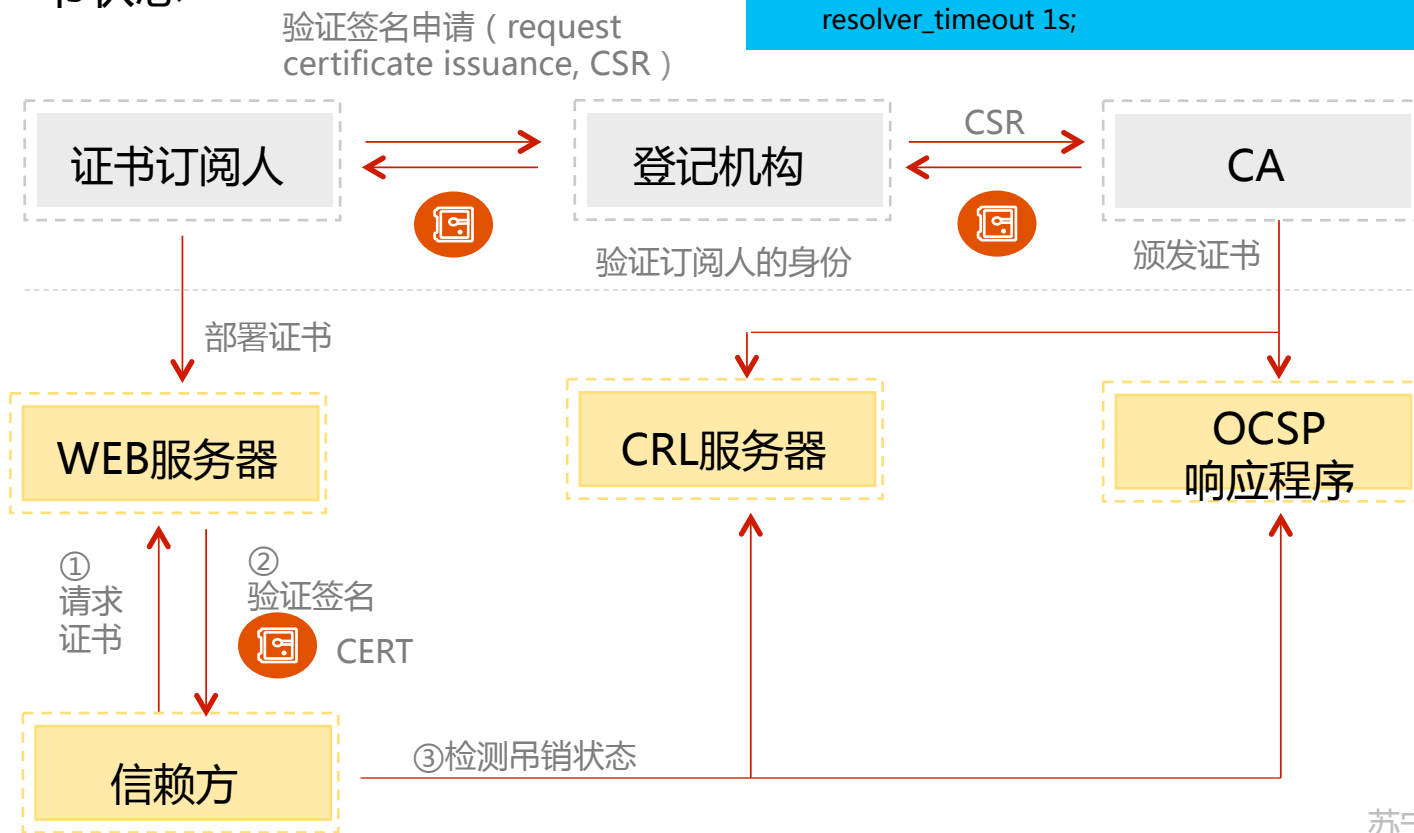
`ssl_prefer_server_ciphers on;`



## Ocsp stapling的合理使用

Ocsp stapling由服务器代替浏览器向CA站点查询证书状态

```
ssl_stapling on;  
ssl_stapling_verify on;  
ssl_stapling_file /home/certs/stapling_ocsp;  
ssl_trusted_certificate /home/certs/chain.pem;  
resolver 223.5.5.5 valid=300s;  
resolver_timeout 1s;
```



## 加密套件的合理使用



身份验证      算法      强度      模式

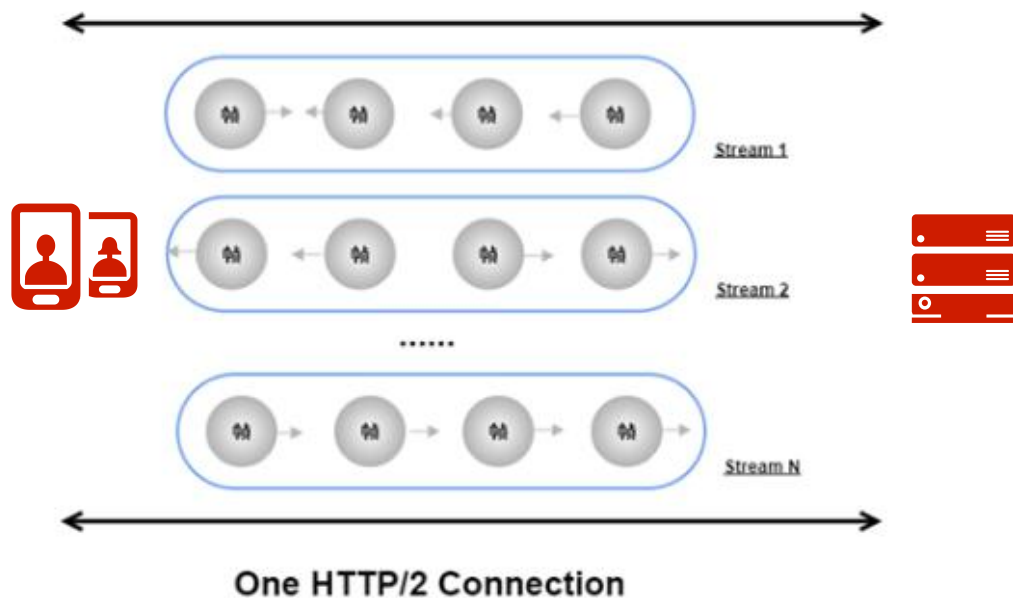
TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256

密钥交换      密码      MAC或PRF

```
ssl_ciphers 'ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-
GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-
AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-ECDSA-AES128-
SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-
ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-
SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-
AES256-SHA:DHE-RSA-AES256-SHA:ECDHE-RSA-DES-CBC3-SHA:ECDHE-ECDSA-DES-CBC3-
SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-
SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!
aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA'
```

## SPDY&HTTP/2的合理使用

- HTTP/2是完全多路复用的，而非有序并阻塞的；
- 请求优先级
- 使用报头压缩，HTTP/2降低了开销
- HTTP/2实现了服务端响应的主动推送
- 必要条件：Nginx 集成- with- http\_v2\_module，并且必须支持 OpenSSL version 1.0.2.以上（ALPN协议需要）



## nginx 启用 HTTP/2 存在的问题

### Post请求被拒绝

为了减少网络时延，不少 HTTP/2 客户端会在建立 HTTP/2 连接时同时发送其它帧包括用来 POST 数据的 DATA 帧。

nginx 能够正常处理客户端提前发送的其它帧，唯独 DATA 帧不行。因为客户端尚未收到 SETTINGS 帧之前，nginx 将初始窗口大小设置为 0。

### 修复: nginx 1.10.2 stable 版

http2\_body\_preread\_size

Changes with nginx 1.10.2

18 Oct 2016

- \*) Change: the "421 Misdirected Request" response now used when rejecting requests to a virtual server different from one negotiated during an SSL handshake; this improves interoperability with some HTTP/2 clients when using client certificates.
- \*) Change: HTTP/2 clients can now start sending request body immediately; the "http2\_body\_preread\_size" directive controls size of the buffer used before nginx will start reading client request body.
- \*) Bugfix: a segmentation fault might occur in a worker process when using HTTP/2 and the "proxy\_request\_buffering" directive.
- \*) Bugfix: the "Content-Length" request header line was always added to requests passed to backends, including requests without body, when using HTTP/2.
- \*) Bugfix: "http request count is zero" alerts might appear in logs when using HTTP/2.
- \*) Bugfix: unnecessary buffering might occur when using the "sub\_filter" directive; the issue had appeared in 1.9.4.
- \*) Bugfix: socket leak when using HTTP/2.
- \*) Bugfix: an incorrect response might be returned when using the "aio\_threads" and "sendfile" directives; the bug had appeared in 1.9.13.
- \*) Workaround: OpenSSL 1.1.0 compatibility.



## HTTP/2 压测工具

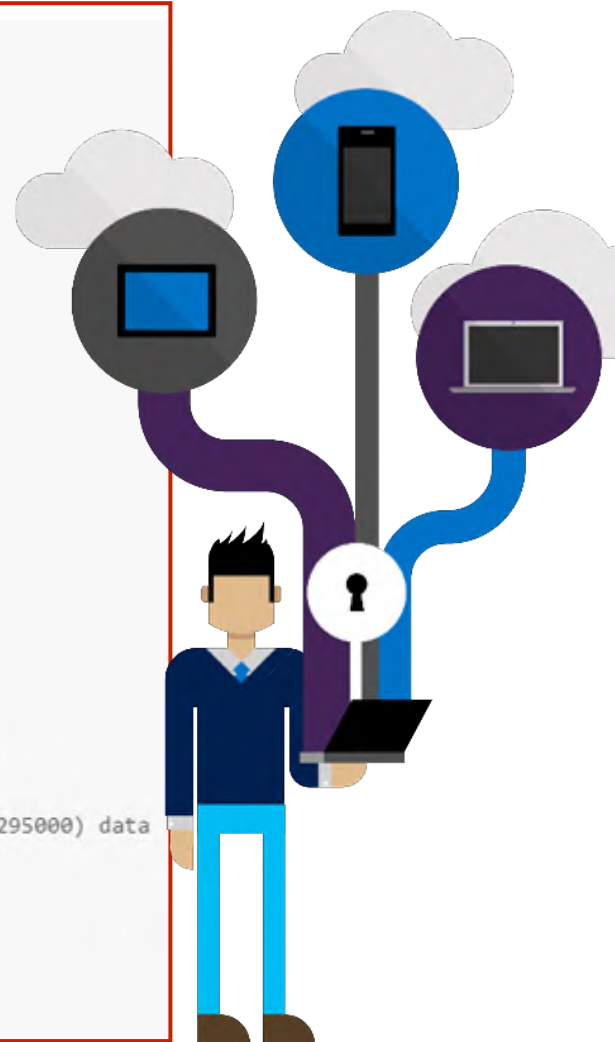
- <https://nghttp2.org/>

```
h2load https://example.com -n 100 -c 10

starting benchmark...
spawning thread #0: 10 total client(s). 100 total requests
TLS Protocol: TLSv1.2
Cipher: ECDHE-RSA-AES128-GCM-SHA256
Application protocol: h2
progress: 10% done
progress: 20% done
progress: 30% done
progress: 40% done
progress: 50% done
progress: 60% done
progress: 70% done
progress: 80% done
progress: 90% done
progress: 100% done

finished in 589.98ms, 169.50 req/s, 2.19MB/s
requests: 100 total, 100 started, 100 done, 100 succeeded, 0 failed, 0 errored, 0 timeout
status codes: 100 2xx, 0 3xx, 0 4xx, 0 5xx
traffic: 1.29MB (1353790) total, 53.42KB (54700) headers (space savings 24.97%), 1.24MB (1295000) data

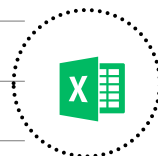
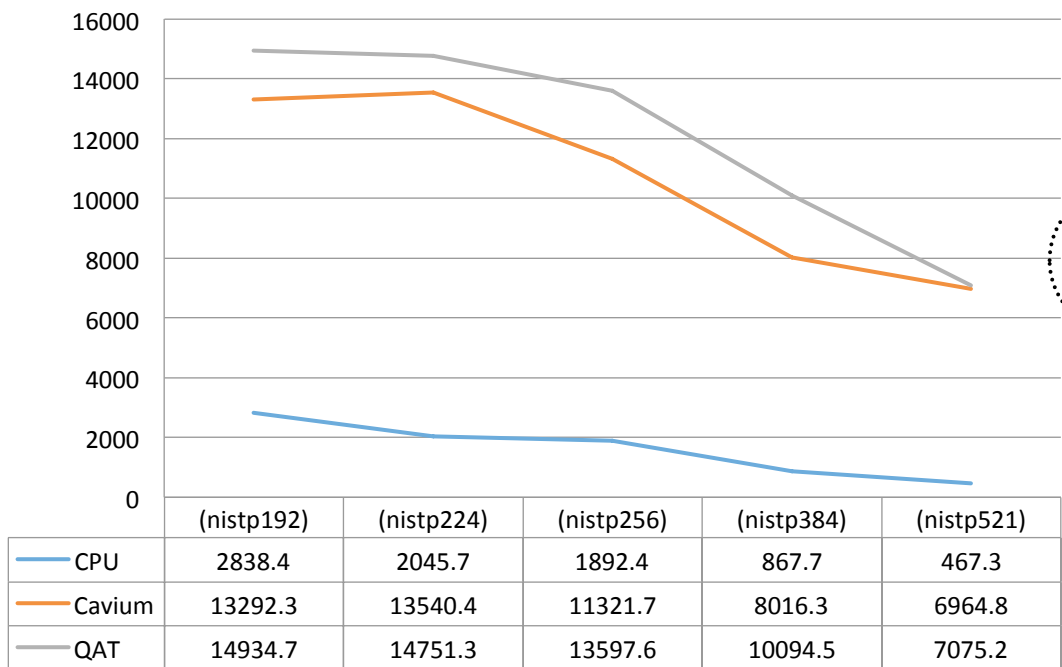
time for request:   min      max      mean      sd      +/- sd
time for connect:   81.84ms 168.23ms 119.76ms 26.44ms 60.00%
time to 1st byte:  119.09ms 201.01ms 158.06ms 23.86ms 70.00%
req/s              :    17.00    28.63    23.12     3.20  60.00%
```





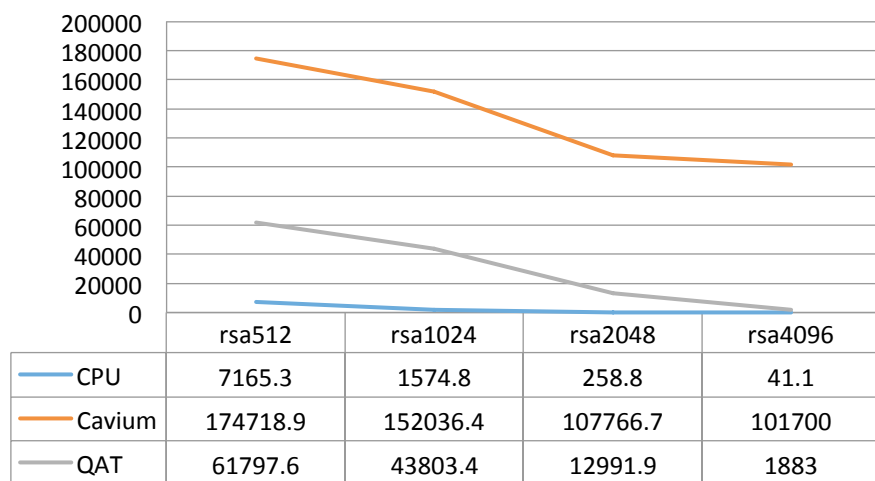
## SSL硬件加速卡合理使用

物理机加速卡情况下ECDH算法性能比较 (单位: op/s)

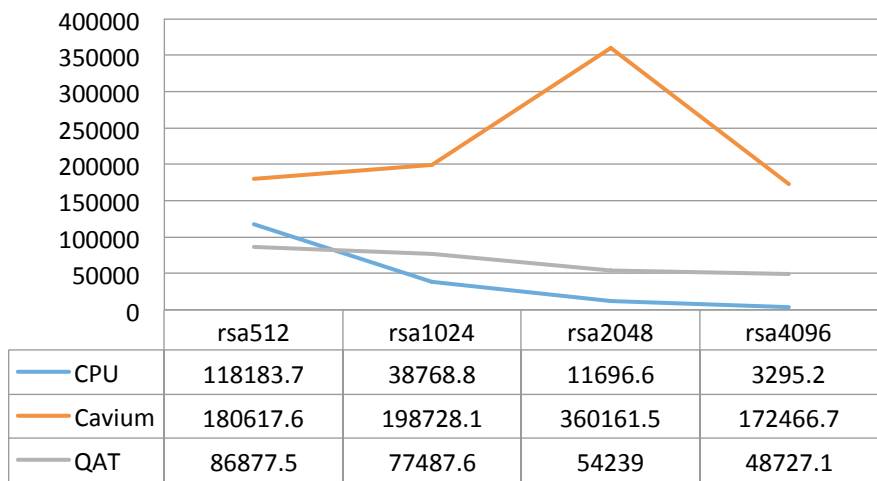


## SSL硬件加速卡合理使用

物理机加速卡情况下RSA算法性能比较 (单位: sign/s)



物理机加速卡情况下RSA算法性能比较 (单位: verify/s)



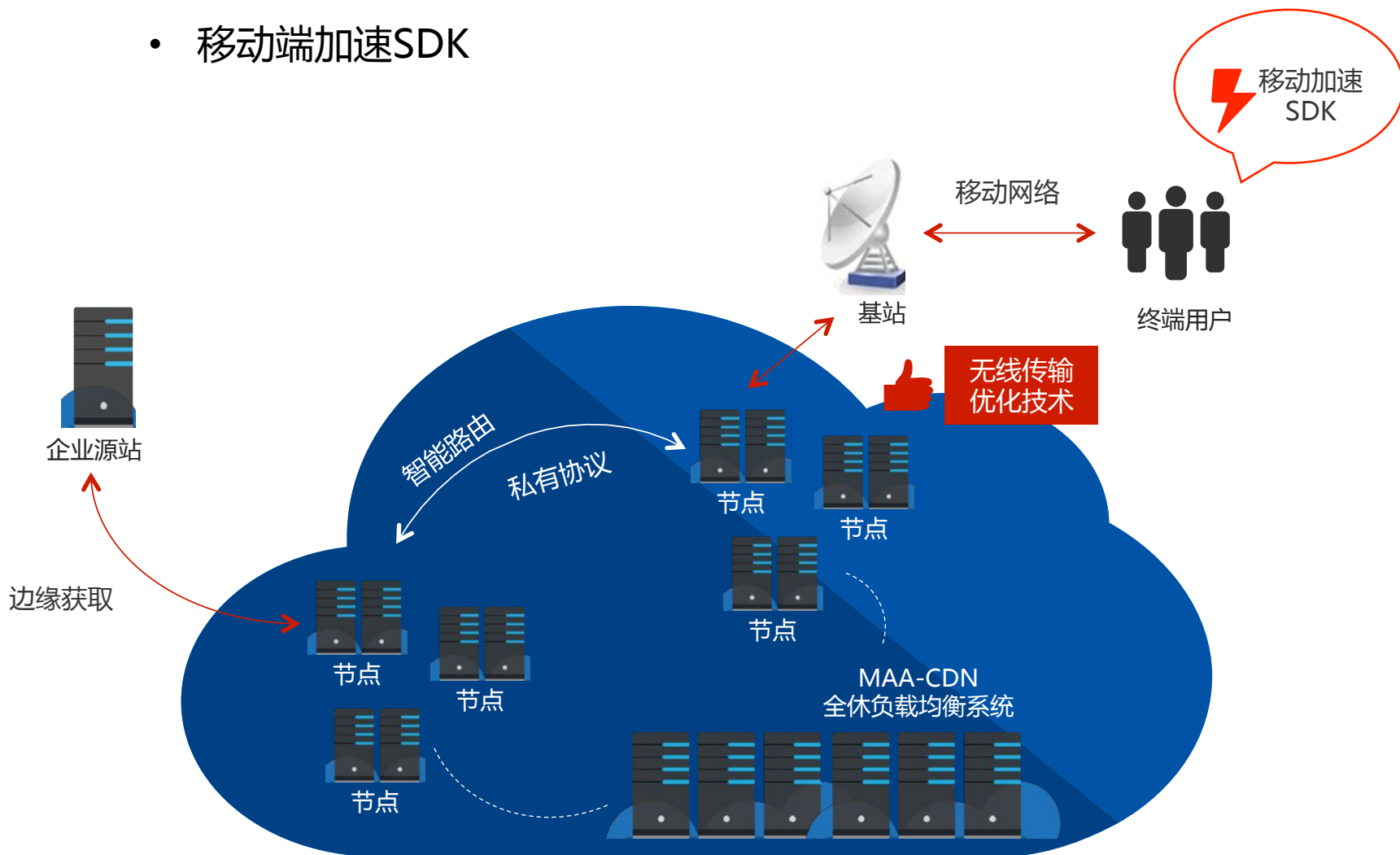
虚拟机环境下的测试效果要比物理机好，尤其是Cavium的加速卡，openssl单进程下虚拟机比物理机资源利用率有10倍的提升。

Rsa算法：无论物理机还是虚拟机环境下，Cavium性能都比intel要优，且性能优势明显！

Ecdh算法：QAT支持的算法类别比Cavium要多，物理机环境下QAT性能略优于Cavium，但是在虚拟机环境下QAT整体性能比Cavium要低50%！

## 客户端HTTPS的性能优化

- 移动端加速SDK



• 移动端加速SDK效果分析

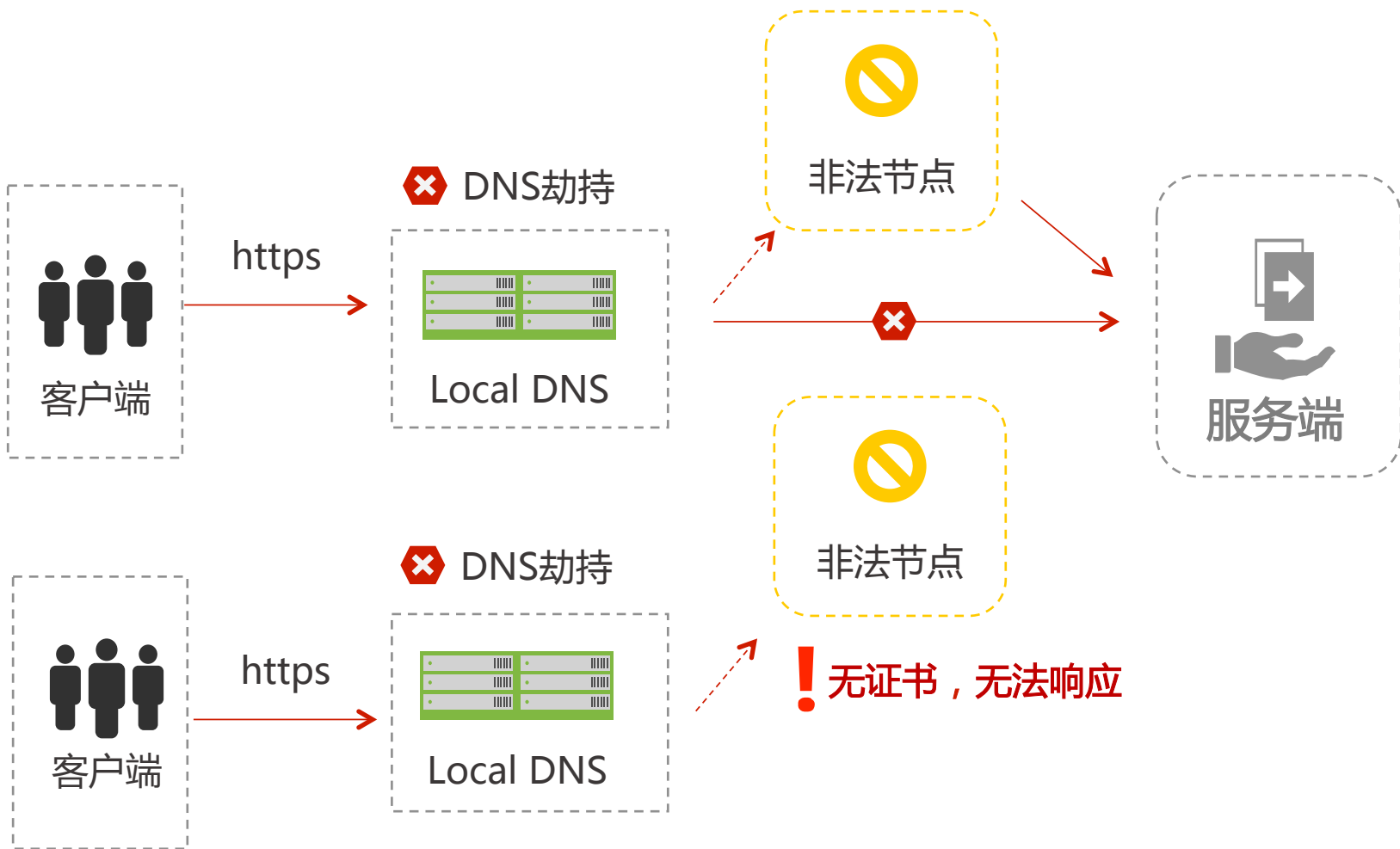


加速效果对比  
安卓



加速效果对比  
IOS

- HTTPS使DNS劫持问题扩大化



- HttpDNS 解决 DNS攻击劫持



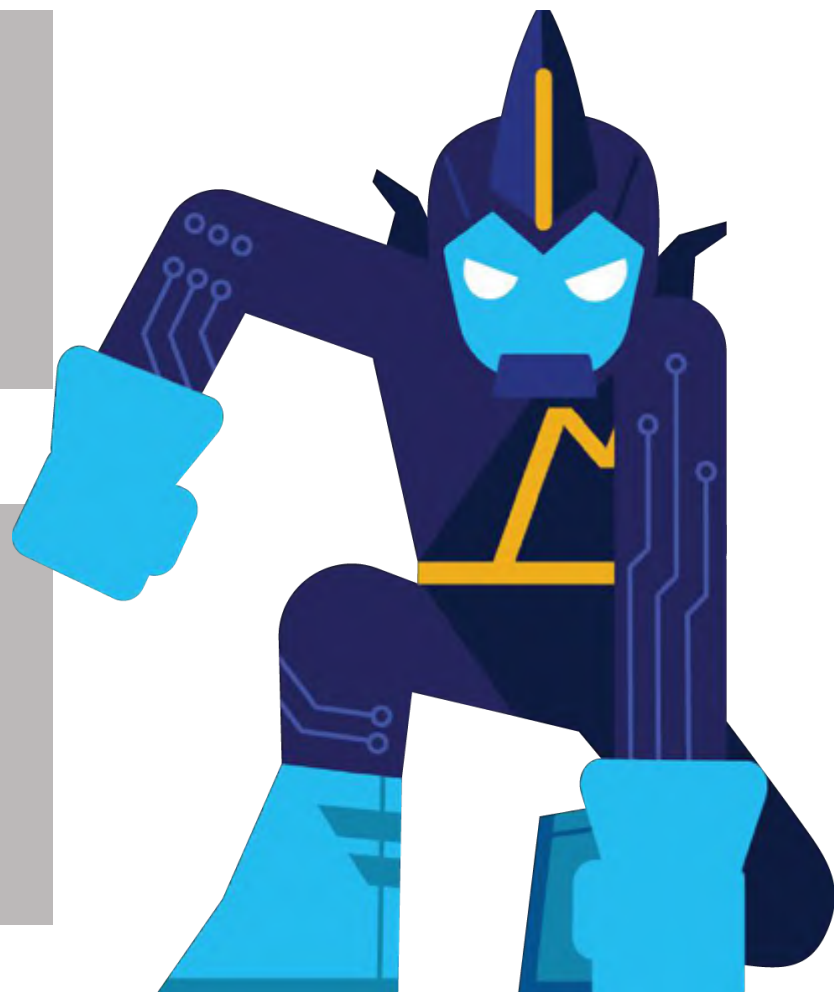
DNS防劫持：HttpDNS绕过公共DNS让网站访问永远正确

### 方案效果：

- 规避公网DNS攻击劫持，反馈0劫持
- 精准调度，根据实际用户IP

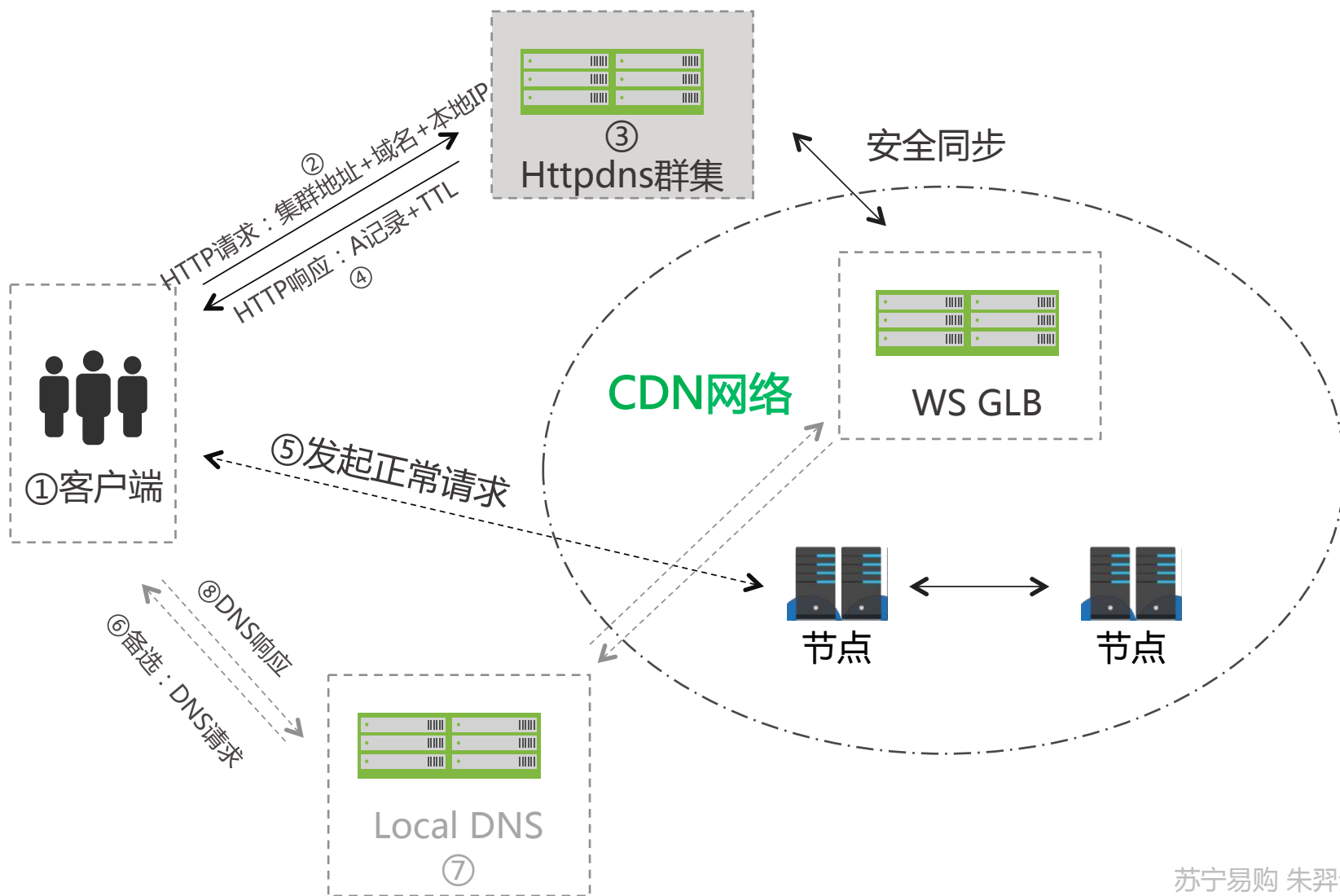
### 客户端接入：

- 接口（现成，使用最多）
- SDK（包含劫持检测功能）





- HttpDNS 解决 DNS攻击劫持



05

HTTPS灰度上线篇

# HTTPS上线

灰度原则

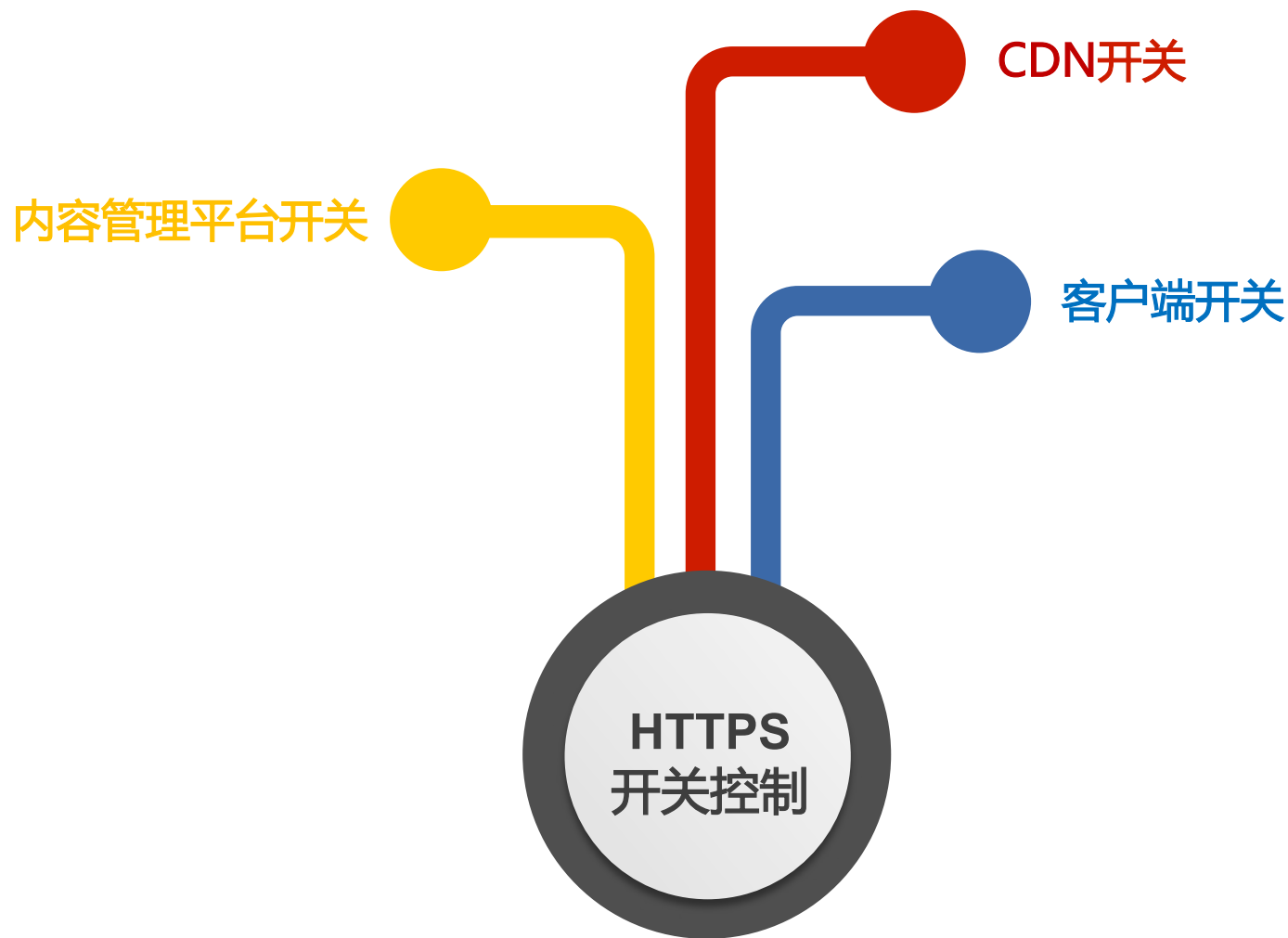
降级原则

开闭原则

# 原则



## HTTPS开关控制

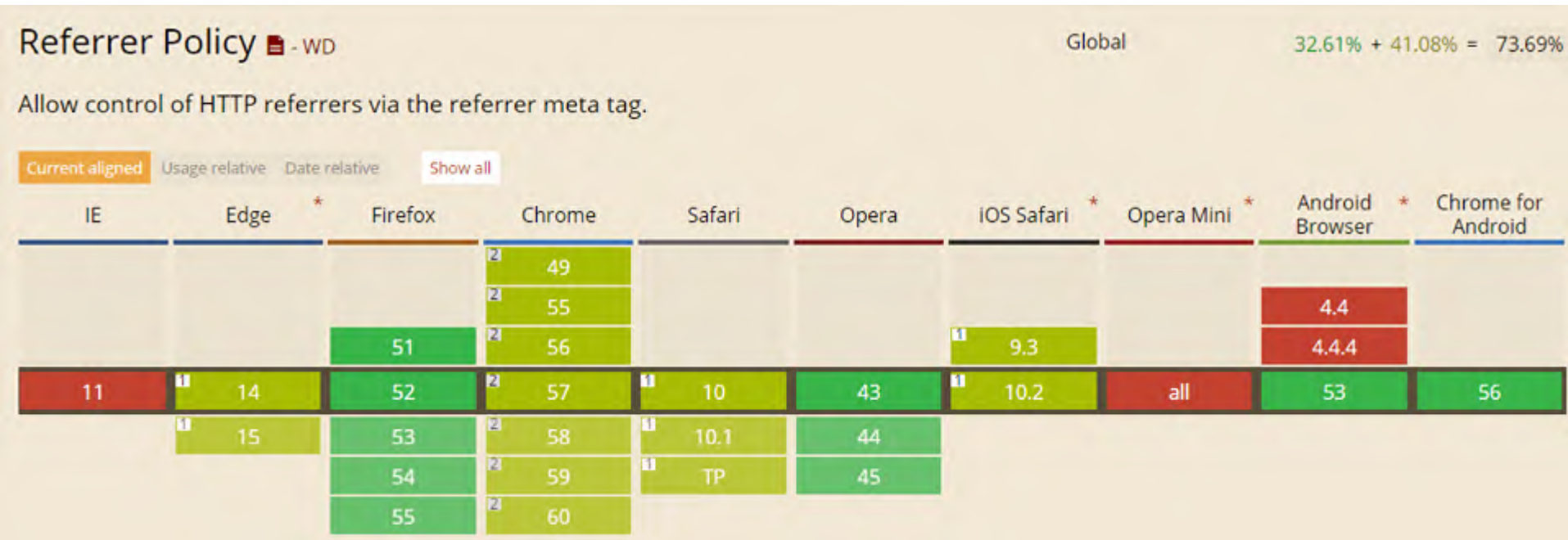




## Referrer的问题

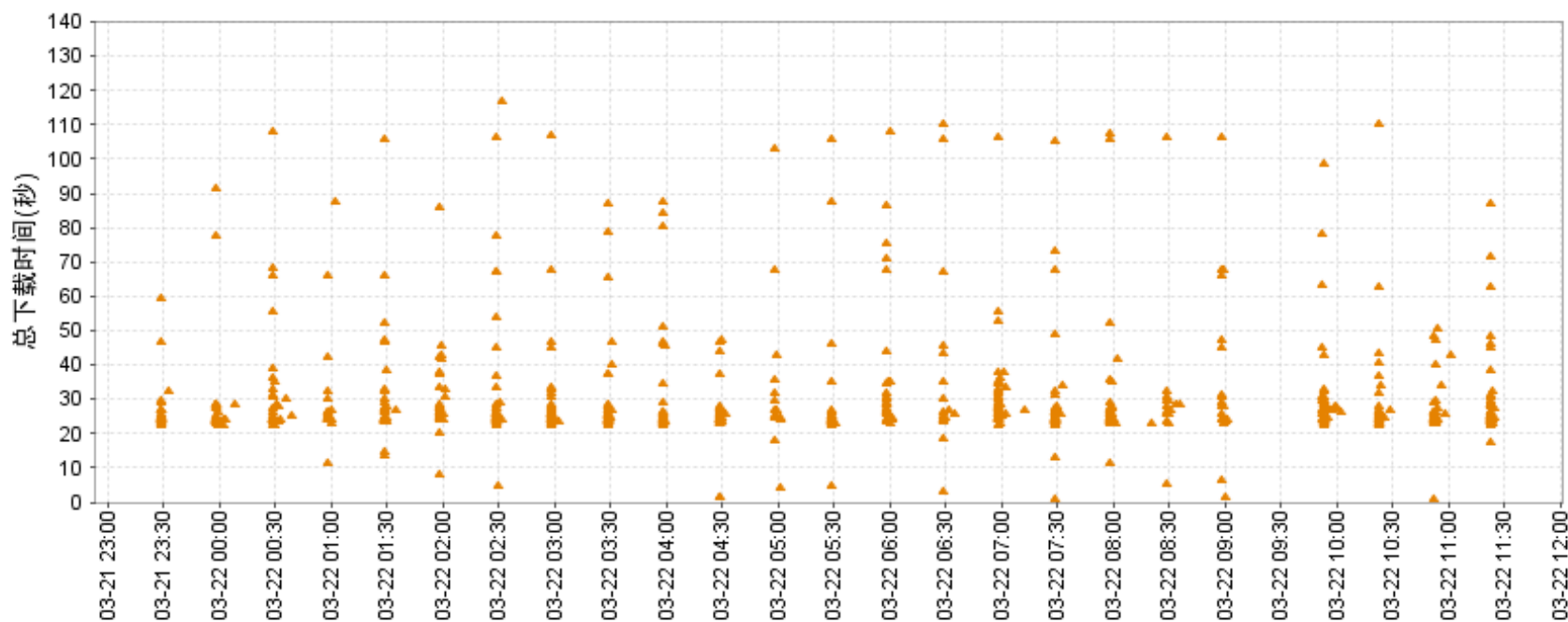
最典型的场景就是从 HTTPS 页面点链接跳到 HTTP 网站时，浏览器并不会在请求头中带上 Referrer 字段。

```
<meta name="referrer" content="always" />
```



## DNS劫持

散点图 (2017年03月21日 11:30 - 2017年03月22日 11:30)



总计754个监测点

中华特色馆Https - 正常	0
中华特色馆Https - 页面错误	0
中华特色馆Https - 内容错误	754
中华特色馆Https - 警告	0





# DNS劫持

The screenshot shows the Suning.com website interface. At the top, there is a navigation bar with links for '苏宁易购首页', '网站导航', '商家入驻', '登录', '注册', '我的订单', '我的易购', '购物车', '手机苏宁', '易付宝', '企业采购', and '客户服务'. Below this is the Suning logo and a search bar containing 'iPhone 7'. A secondary search bar lists categories like '智能电视', '4K超高清', '油烟机', '电热水器', '空调', '双门冰箱', 'iPad', '单反', '滚筒洗衣机', and '电饭煲'. The main content area is dominated by a large red rectangle, indicating a hijacking or a security issue. On the left, a navigation menu lists categories such as '城市大全', '粮油调味', '休闲零食', '工艺品', '滋补养生', '酒水饮料', and '时令生鲜'. Below the red area, there is a '地理标志' section with tabs for '华北特产', '华东特产', '华中特产', '华南特产', '西南特产', '西北特产', and '东北特产'. Several product cards are visible, including '【中华特色馆】镇安镇特产香菇干菜', '田园品味特级初榨橄榄油', '新鲜牛肝菌干货', and '蚂蚁市集琥珀核桃仁'. The bottom right corner of the page shows the text '苏宁易购 朱羿全'.





# HTTPS性能监控





## HTTPS性能监控

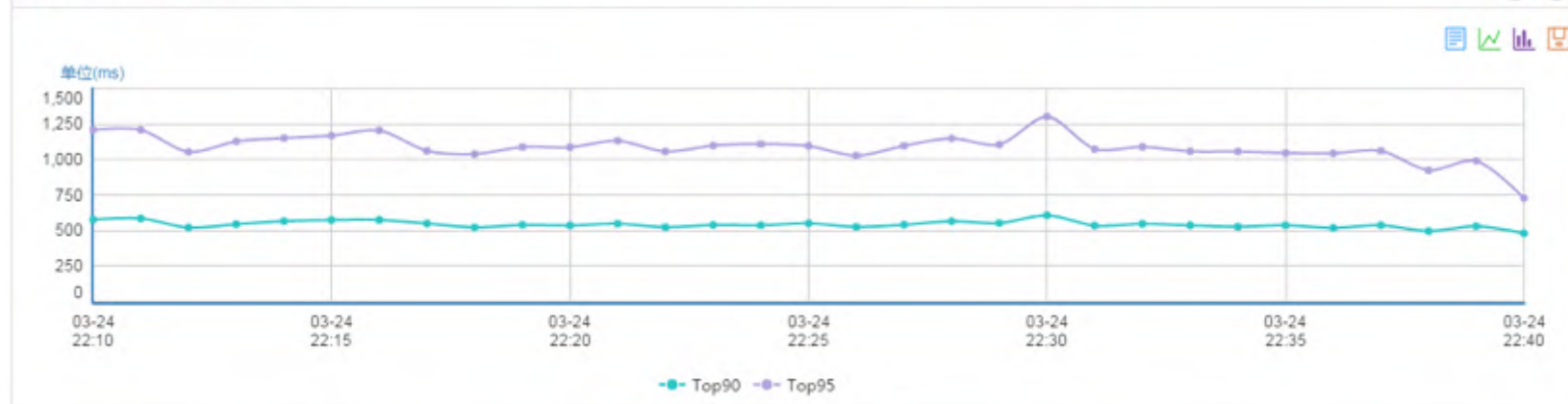
### 响应占比分析

筛选条件： 域名  关键事务

选择城市  选择运营商  选择接入方式  搜索

最近30分钟

#### TOP99/TOP95/TOP90报表



06

HTTPS未来展望篇

# QUIC

Thank you !