有趣的DNS

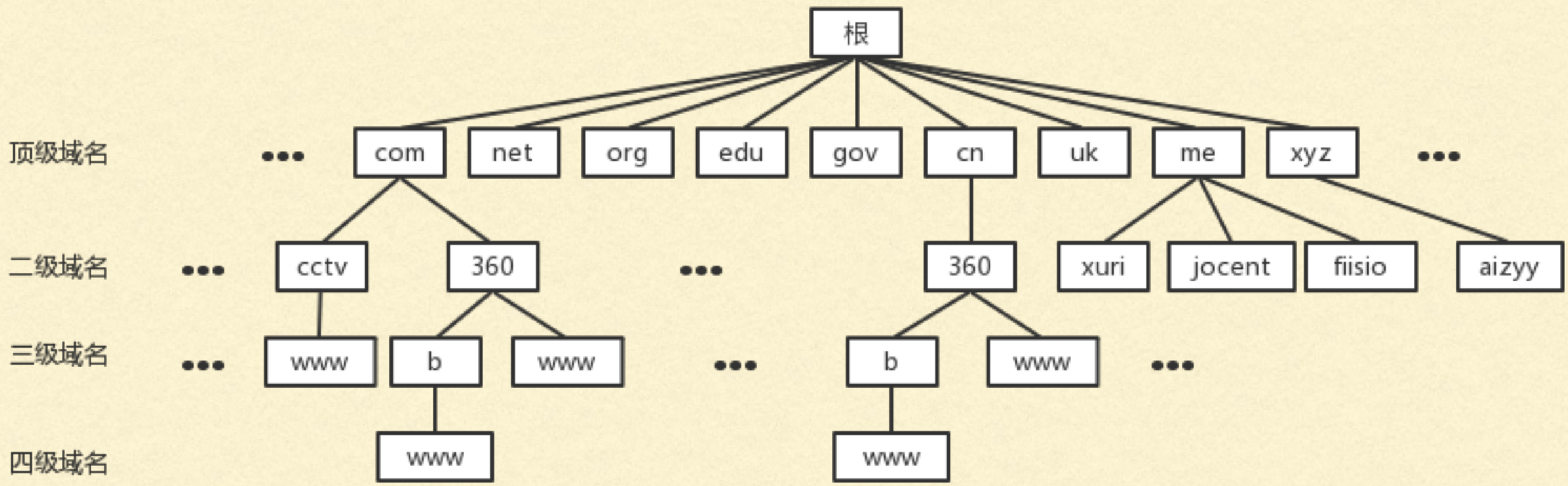
liangdong@smzdm.com

域名

www.smzdm.com.

三级域　二级域　顶级域　根域

域名



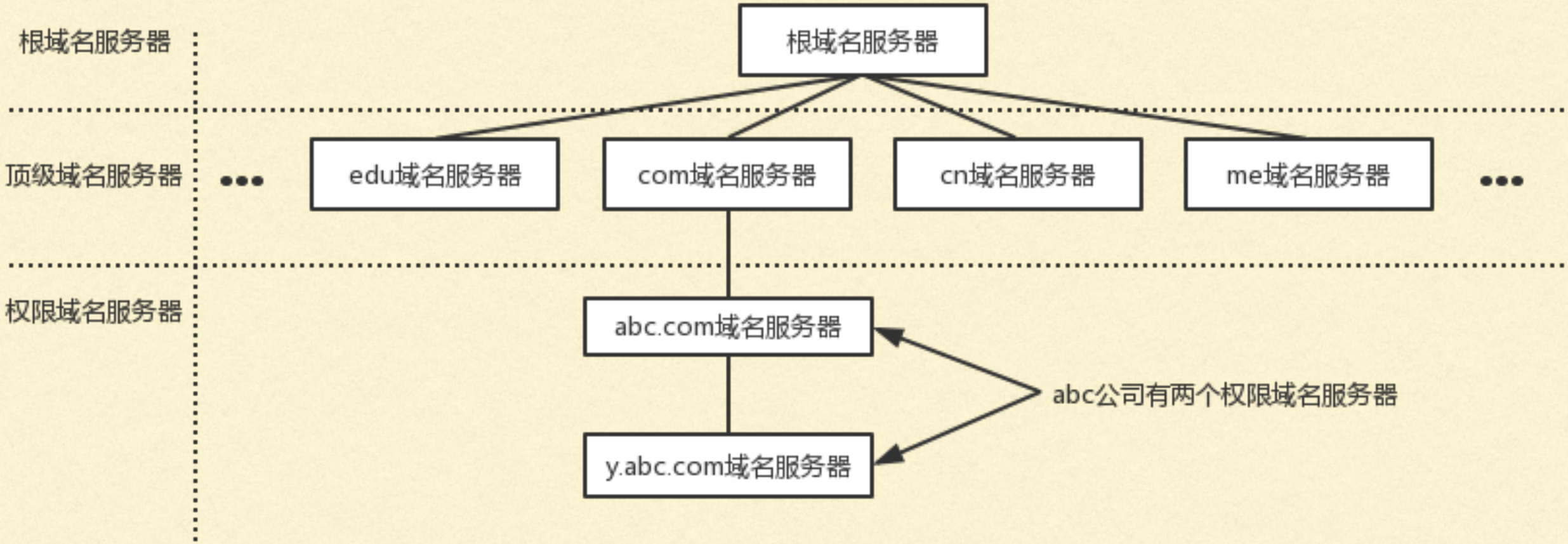
因特网的域名空间结构

实战 - 解析域名

dig www.smzdm.com

域名是递归解析的

域名服务器


根域名服务器
顶级域名服务器
权限域名服务器
根域名服务器
edu域名服务器
com域名服务器
cn域名服务器
me域名服务器
abc.com域名服务器
y.abc.com域名服务器
abc公司有两个权限域名服务器

实战 - 递归解析

dig @198.41.0.4 www.smzdm.com

dig @192.12.94.30 www.smzdm.com

dig @101.226.220.13 www.smzdm.com

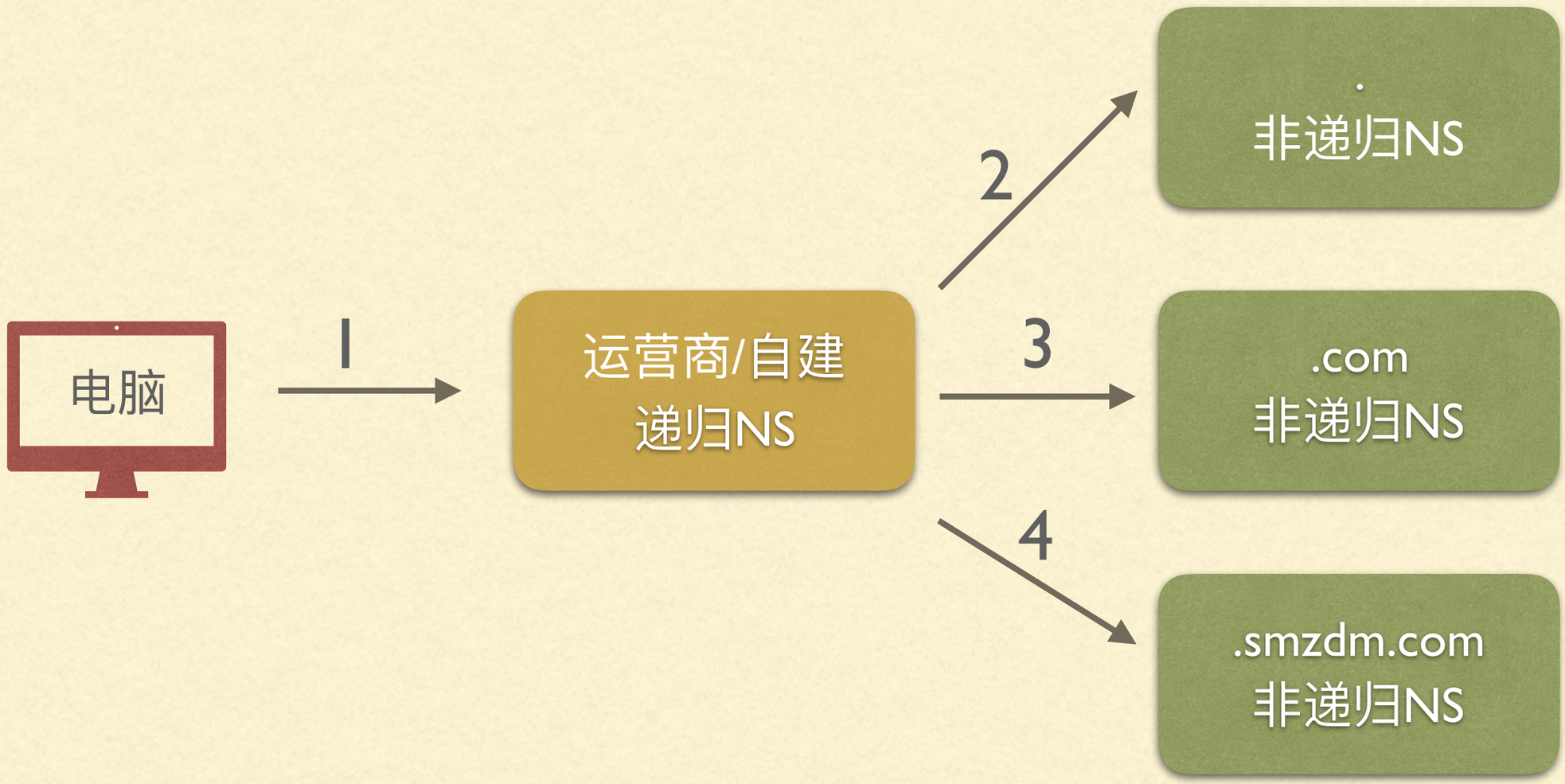
dig @198.41.0.4 c4b0af8c607ae831.vip.jiasule.org

dig @199.19.57.1 c4b0af8c607ae831.vip.jiasule.org

dig @117.21.219.80 c4b0af8c607ae831.vip.jiasule.org

curl 'https://112.90.216.108' -H 'Host:www.smzdm.com' --insecure

DNS架构



总结

- 客户端没有递归特性
- /etc/resolv.conf配置的nameserver必须是递归的
- 根级/顶级服务器是非递归的

了解DNS协议

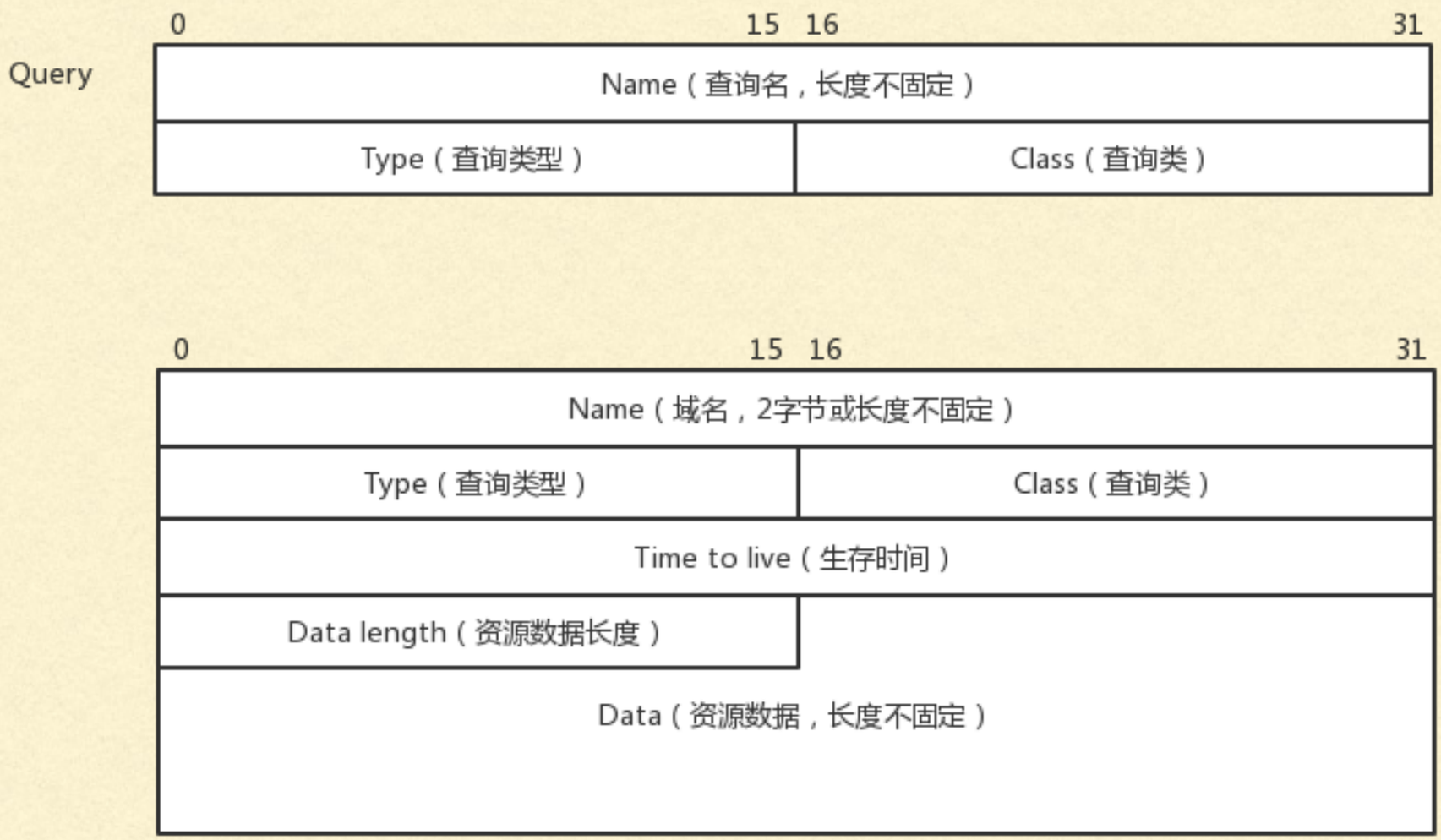
包结构

0 … 15	16 … 31
Transaction ID（会话标识）	Flags（标志）
Questions（问题数）	Answer RRs（回答 资源记录数）
Authority RRs（授权 资源记录数）	Additional RRs（附加 资源记录数）
Queries（查询问题区域）	
Answers（回答区域）	
Authoritative nameservers（授权区域）	
Additional recoreds（附加 区域）	

Header

DNS协议报文格式

查询与应答



资源记录格式

总结

- 53端口，支持UDP、TCP，操作系统api默认采用udp
- udp无状态，可以伪造应答造成dns污染

实战：PYTHON构造DNS请求

1， 创建UDP SOCKET

```
import socket
import dns.message, dns.rrset

# UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

2，发送DNS请求

```
dns_server =  ('8.8.8.8', 53)
domain = 'www.google.com.com'

# DNS request
request = dns.message.make_query(domain, 'A')
sock.sendto(request.to_wire(), dns_server)
```

3，接收DNS应答

```
# DNS response
response = sock.recv(1024)
response = dns.message.from_wire(response)
print(response)
```

4、分析应答

```
id 31228
opcode QUERY
rcode NOERROR
flags QR RD RA
;QUESTION
www.google.com.com. IN A
;ANSWER
www.google.com.com. 117 IN A 199.59.148.14
;AUTHORITY
;ADDITIONAL
```

彩蛋：GFW