Encrypted DNS

(2020 Update)

Carsten Strotmann

Agenda

- DNS-Privacy
- DoH/DoT/DoQ
- The current status
- Oblivious DoH and Adaptive DNS resolver discovery

About me?

Carsten Strotmann

DNS(SEC)/DANE/DHCP/IPv6 trainer and supporter

RIPE/IETF

Privacy in DNS?

- in recent years, the IETF has expanded the DNS protocol with privacy features
  - DNS-over-TLS (Transport-Encryption between DNS client and DNS resolver)
  - DNS-over-HTTPS (Transport-Encryption between DNS client and DNS resolver)
  - QNAME Minimization (less metadata in DNS)
  - EDNS-Padding (*hiding* of DNS data in encrypted connections)

The need for more DNS privacy

- a study presented at IETF 105 during the Applied Networking Research Workshop in July 2019 found that
  - 8.5 % of networks (AS) intercept DNS queries (27.9% in China)
  - (today) most queries are answered un-altered
- but the situation might change, intercept server might change DNS answers

encrypted transport for DNS

encrypted DNS terminology

- Terminology
  - Do53 = **DNS-over-Port53** - classic DNS (UDP/TCP port 53)
  - DoT = **DNS-over-TLS** - TLS as the transport for DNS
  - DoH = **DNS-over-HTTPS** - HTTPS as the transport for DNS
  - DoQ = **DNS-over-QUIC** - QUIC as the transport for DNS
  - DoC = **DNS-over-Cloud** - DNS resolution via cloud services (Google, Q9, Cloudflare …)

DoT - DNS-over-TLS

- RFC 7858 "Specification for DNS over Transport Layer Security (TLS)"
- DNS wireformat over TLS over TCP
- Port 853 (TCP)
- Encryption and Authentication (Internet PKI or via DANE)

DoH - DNS over HTTP(S)

- RFC 8484 *DNS Queries over HTTPS (DoH)* (P. Hoffman, ICANN and P. McManus, Mozilla) https://tools.ietf.org/html/rfc8484
- DNS HTTP-Format over HTTPS over TCP, Port 443 (HTTP/2)
- URL: `https://server/dns-query{?dns}`
- Encryption, Authentication and Cloaking

DoT vs DoH

- differences between DoT and DoH
  - DoT can be easily blocked, because it is running on an dedicated port (853)
  - DoH is made to look like normal HTTPS traffic, selective blocking of DoH is difficult
  - DoH seems to be easier to implement, because of existing HTTPS library functions in programming languages
  - DoH enables developers to do DNS name resolution on an application level, which some people think is bad

Controlling DoH - the Canary Domain

Controlling DoH - the Canary Domain

- Mozilla has implemented a check for a *Canary Domain* in Firefox
- Domain Name `use-application-dns.net.`
  - if the domain-name **can** be resolved via DNS53 -> unmanaged DNS, DoH can be auto-enabled
  - if the domain-name **cannot** be resolved (= is blocked) -> managed DNS, DoH will not be auto-enabled (but users can manually enable DoH)
- the IETF is discussing similar signalling functions: "Signaling resolver's filtering policies" (draft-mglt-add-signaling-filtering-policies)

other checks done by Firefox before enabling DoH

- Resolve canary domains of certain known DNS providers to detect content filtering
- Resolve the *safe-search* variants of `google.com` and `youtube.com` to determine if the network redirects to them
- On Windows and macOS, detect parental controls enabled in the operating system
- additional checks performed for private *enterprise* networks are:
  - Is the Firefox `security.enterprise_roots.enabled` preference set to true?
  - Is any enterprise policy configured?

Current DoT/DoH client status

Firefox Browser

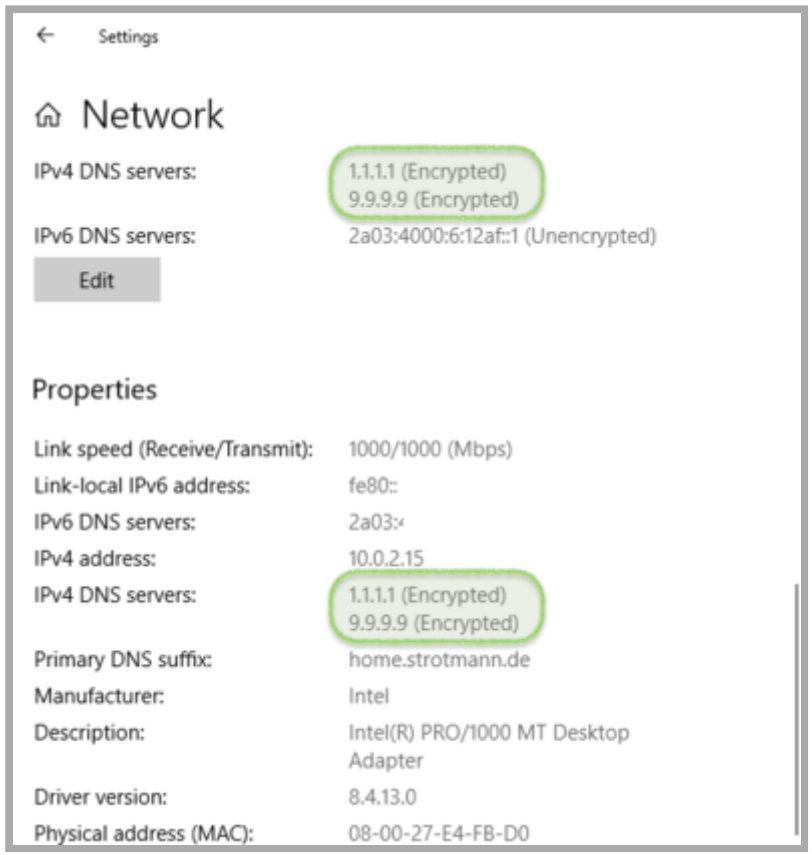
- Firefox Trusted Recursive/Remote Resolver (TRR) Program
  - Cloudflare (default) or NextDNS
  - Comcast XFinity (coming)
  - automatic rollout started in February 2020

Chrome(ium) Browser

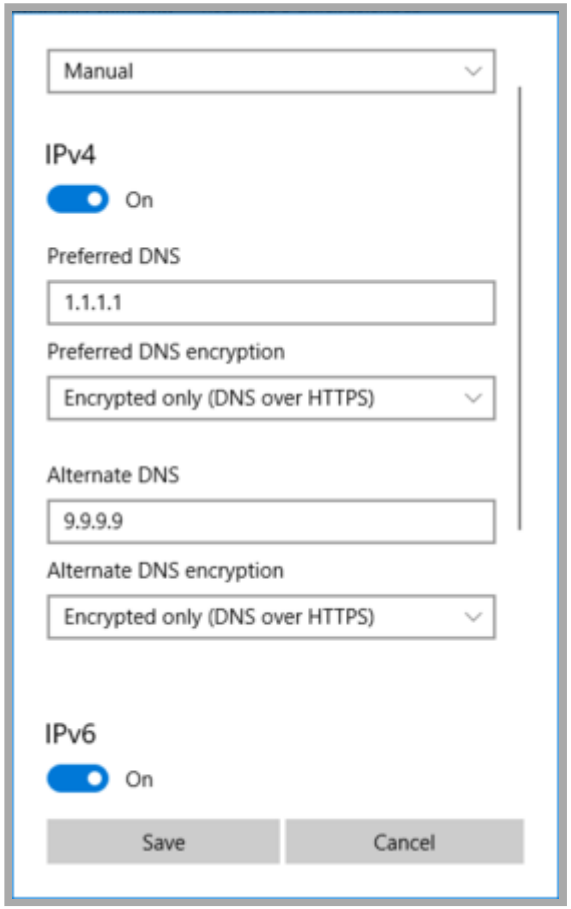
- DoH is implemented and can be enabled by the user
  - Google Chrome
  - Opera
  - Vivaldi
  - Brave
  - Microsoft Edge
  - Bromite
- DoH "auto upgrade" for the configured DNS resolvers (manual configured or DHCP/RA supplied)
- Google is experimenting with adaptive DoH-Resolver-Discovery via DNS

Microsoft Windows 10 (1/2)

- support in latest "Insider" builds of Windows 10



Microsoft Windows 10 (2/2)



Linux

- DoT support in `systemd-resolved` for some time
- opportunistic mode only (automatic fallback to DNS53)
- no server authentication (MITM possible)
- global or "per interface" setting
- not enabled by default

OpenBSD

- DoT support in `unwind`
- not enabled by default
- opportunistic "auto update" mode or manual configured "strict" mode
- server authentication via TLS certificate

Android

- DoT available from Andoid 9 "Pie"
- manual setting
- "auto upgrade" from the configured DNS resolver, or Google DNS as fallback
- auto upgrade to DoH in Chrome for Android Version 85+ (September 2020)

Apple MacOS 11 and iOS/iPadOS 14

- support for DoT and DoH
- global and per App/Application resolver selection possible
- "encrypted DNS" configuration Apps possible, user can choose provider by installing App
- OS can learn "per Domain" DoH/DoT setting via DNS or HTTP (Adaptive DNS-over-HTTPS)
- OS can discover DoH/DoT Server via DHCP/PvD (Provisioning Domains) or queries to `resolver.arpa` via classic DNS53
- Discovery methods in active discussion in the IETF ADD working group

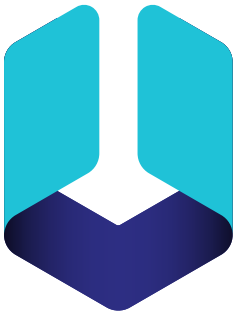
Current DoT/DoH server status

BIND 9

- DoH/DoT support is currently in the BIND 9.17 development branch (**not for production use**)
- BIND 9.18 will contain DoH and DoT support
  - scheduled for early in 2021, will be the ‘2021 stable release’
- ISC has also committed to backporting DoH and DoT to BIND 9.16 (Extended Support Version)

Unbound

- the Unbound DNS resolver does support DoT since 2017 (and had support for DNS-over-SSL via Port 443 before that)
- support for DNS-over-HTTPS (DoH) has been merged into the Unbound source code and is scheduled for Unbound 1.11.1 in October 2020



unbound

other DNS Resolver

- dnsdist is an open source DNS load-balancer that supports DoT and DoH
- some commercial TLS loadbalancer (e.g. A10) support DoH and/or DoT
- NGINX, the popular open source webserver and protocol proxy, does support DoT and DoH
- more DoT/DoH implementations can be found on the presenters encrypted DNS implementations page

Adaptive DNS-over-HTTPS

Adaptive DNS-over-HTTPS

- Goals (directly taken from the Internet Draft):
  - No party other than the client and server can learn or control the names being queried by the client or the answers being returned by the server.
  - Only a designated DNS resolver associated with the deployment that is also hosting content will be able to read both the client IP address and queried names for Privacy-Sensitive Connections.
  - Clients will be able to comply with policies required by VPNs and local networks that are authoritative for private domains

Designated DoH server for domains

- DoH Servers for a domain can be learned
  - from DNSSEC secured HTTPS/SVCB records
  - HTTP(S) `ALT-SVC` header
  - DoH-Server "well-known" URL
  - local provisioning domain (PvD)

HTTPS Record

- eliminates additional roundtrip (DNS or HTTP)
- the HTTPS DNS record provides
  - address information (`ipv4hint`, `ipv6hint`)
  - protocol information (protocol upgrade request -> HTTP/3[QUIC])
  - public keys (encrypted client hello)
  - other data, such as encrypted DNS resolver hint (`dohuri`)

HTTPS Record Example

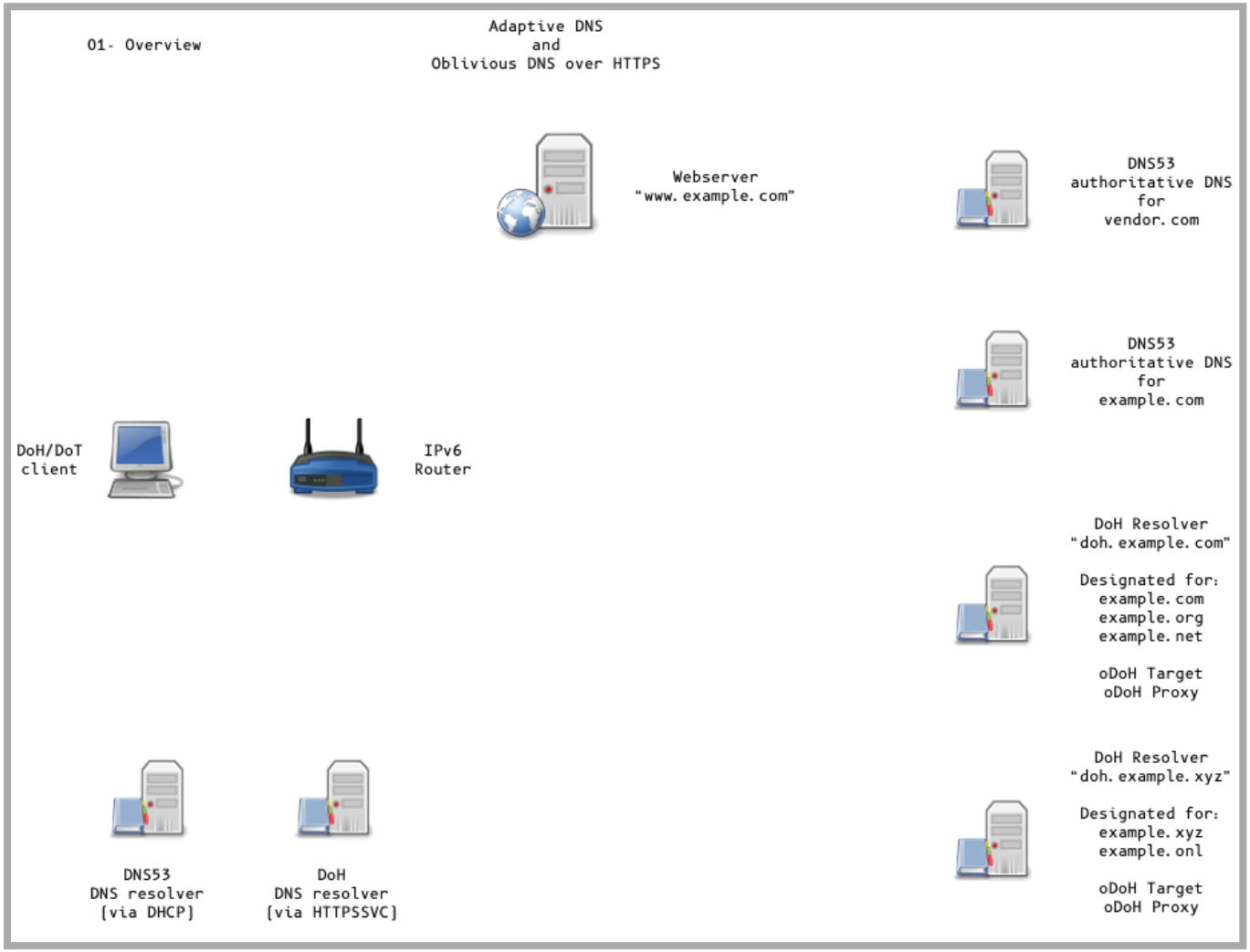
```
example.com.        IN HTTPS 0 svc.example.net.
svc.example.net.    IN HTTPS 2 svc1.example.net. (
                         dohuri=https://doh.example.net/dns-query
                         odohkey="..." )
```

https://www.ietf.org/id/draft-ietf-dnsop-svcb-https-01.txt

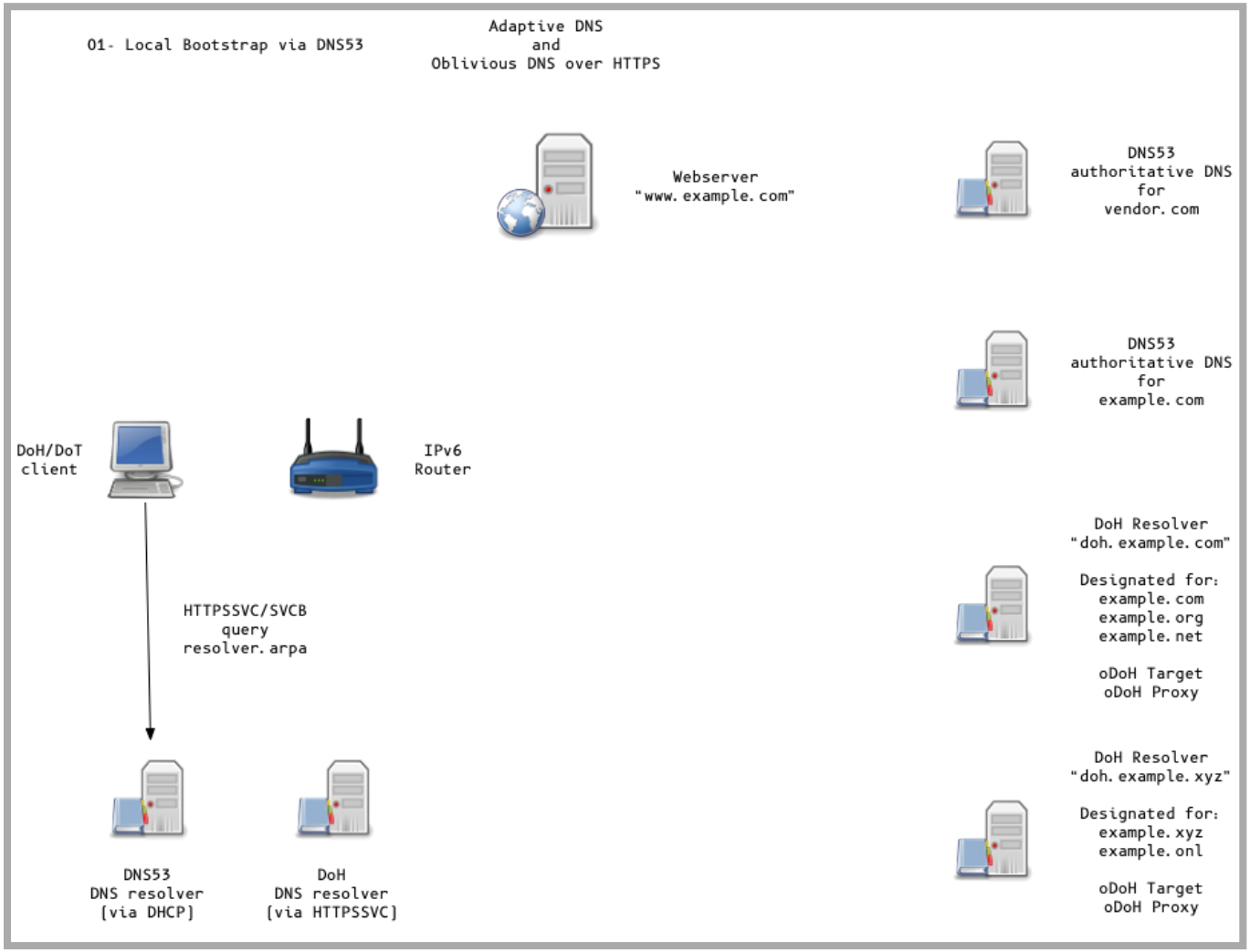
Oblivious DoH (oDoH)

- oDoH is an extension to DoH that allows client IP addresses to be disassociated from queries via proxying (pauly-dprive-oblivious-doh)

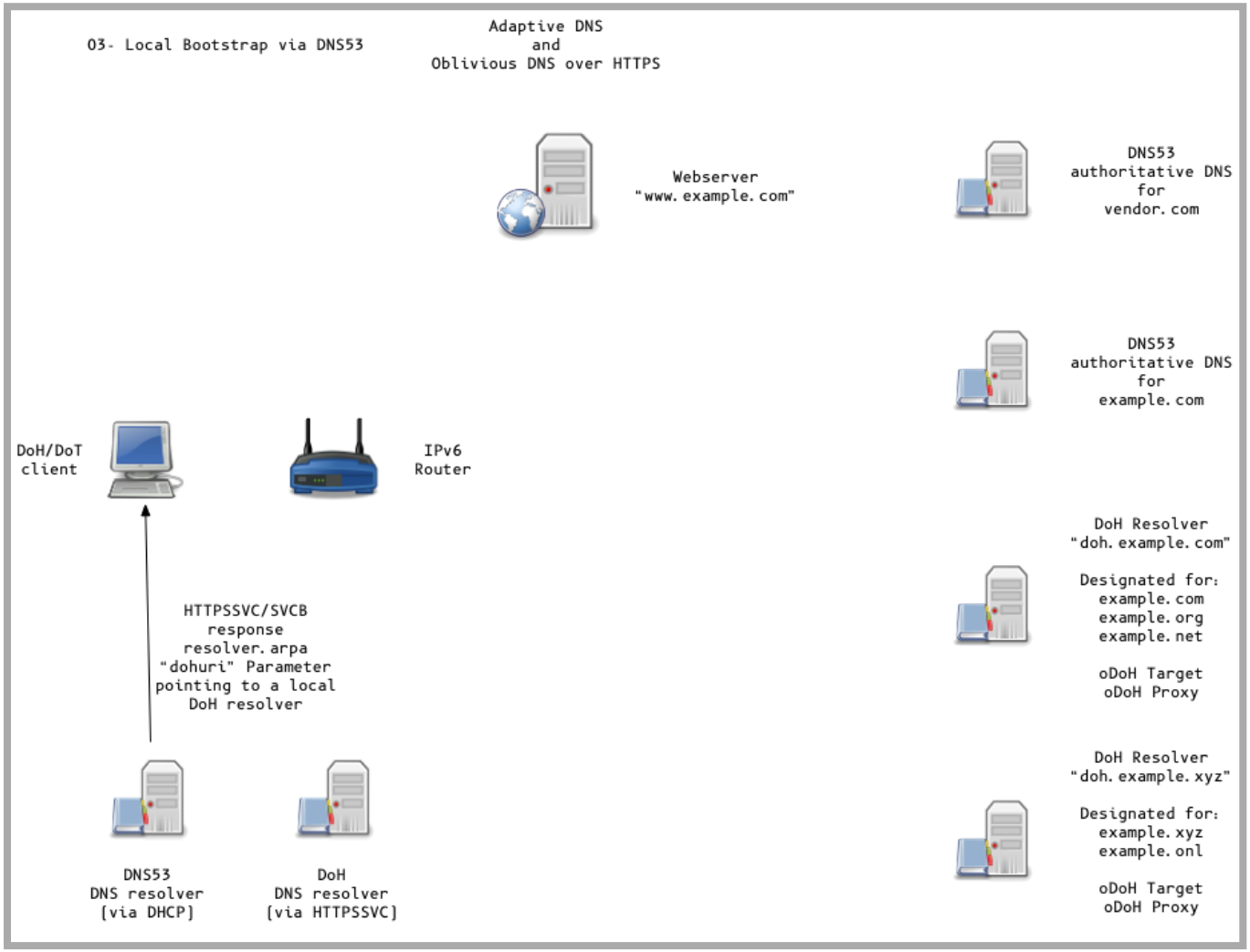
Adaptive DNS Discovery and oDoH



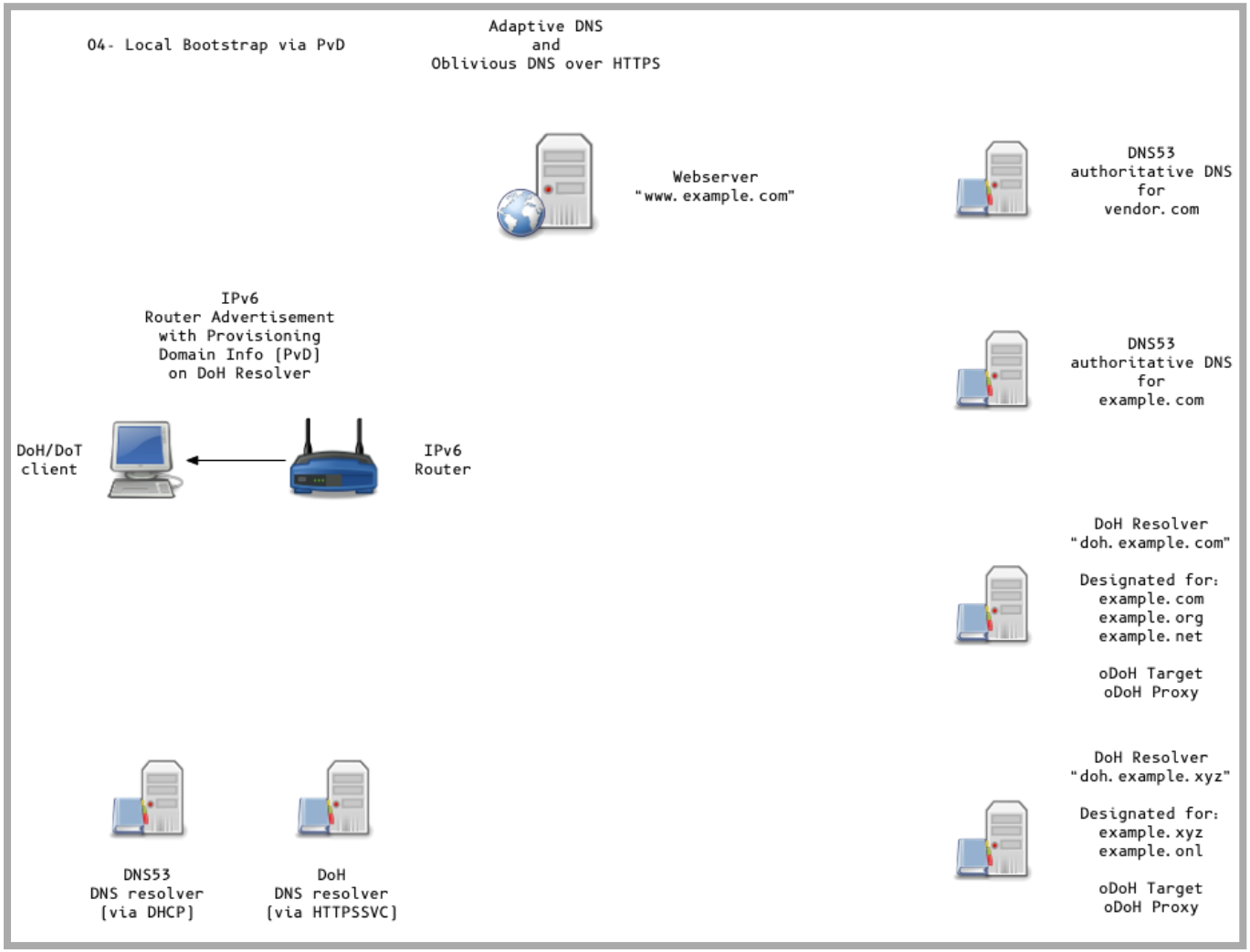
Adaptive DNS Discovery and oDoH



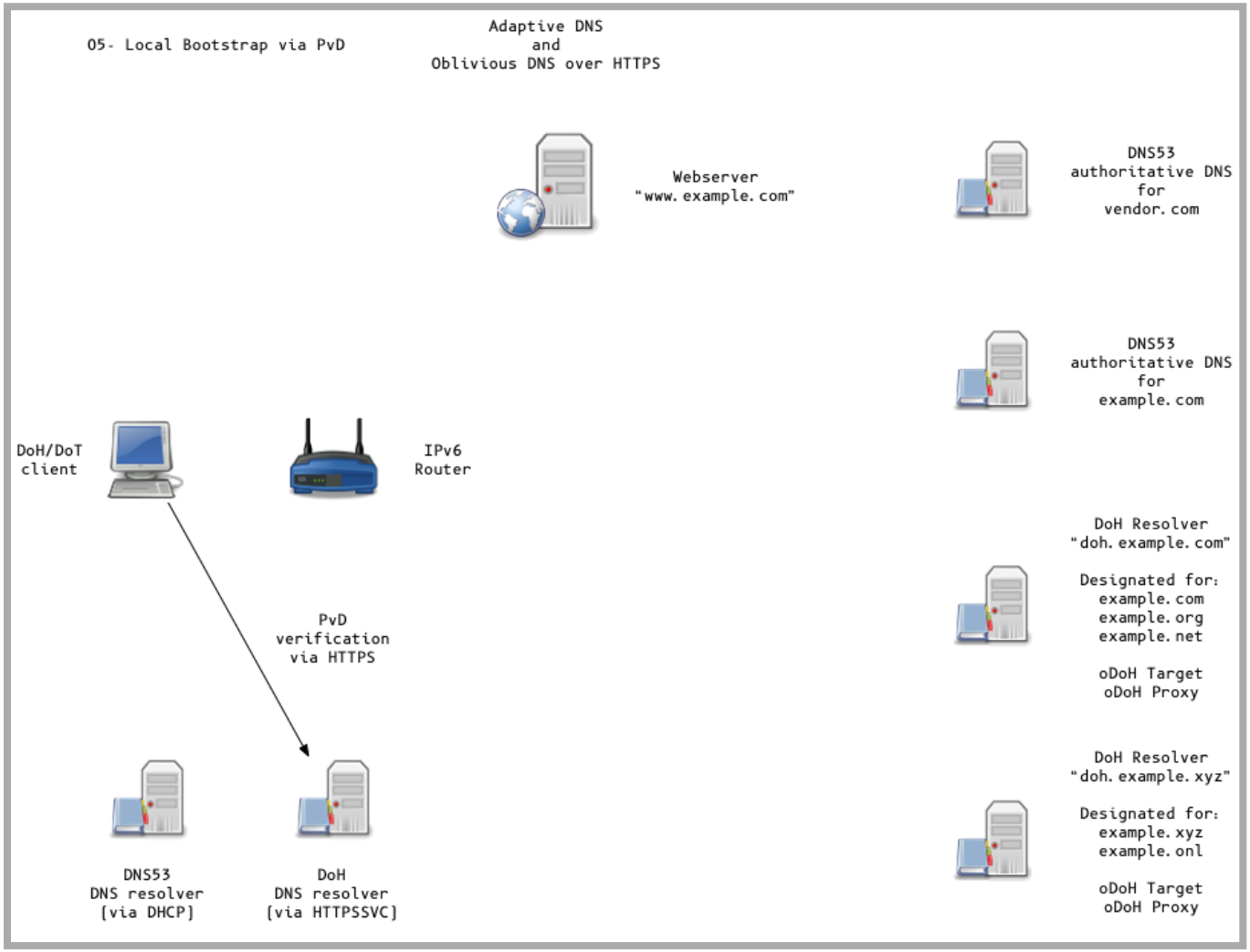
Adaptive DNS Discovery and oDoH



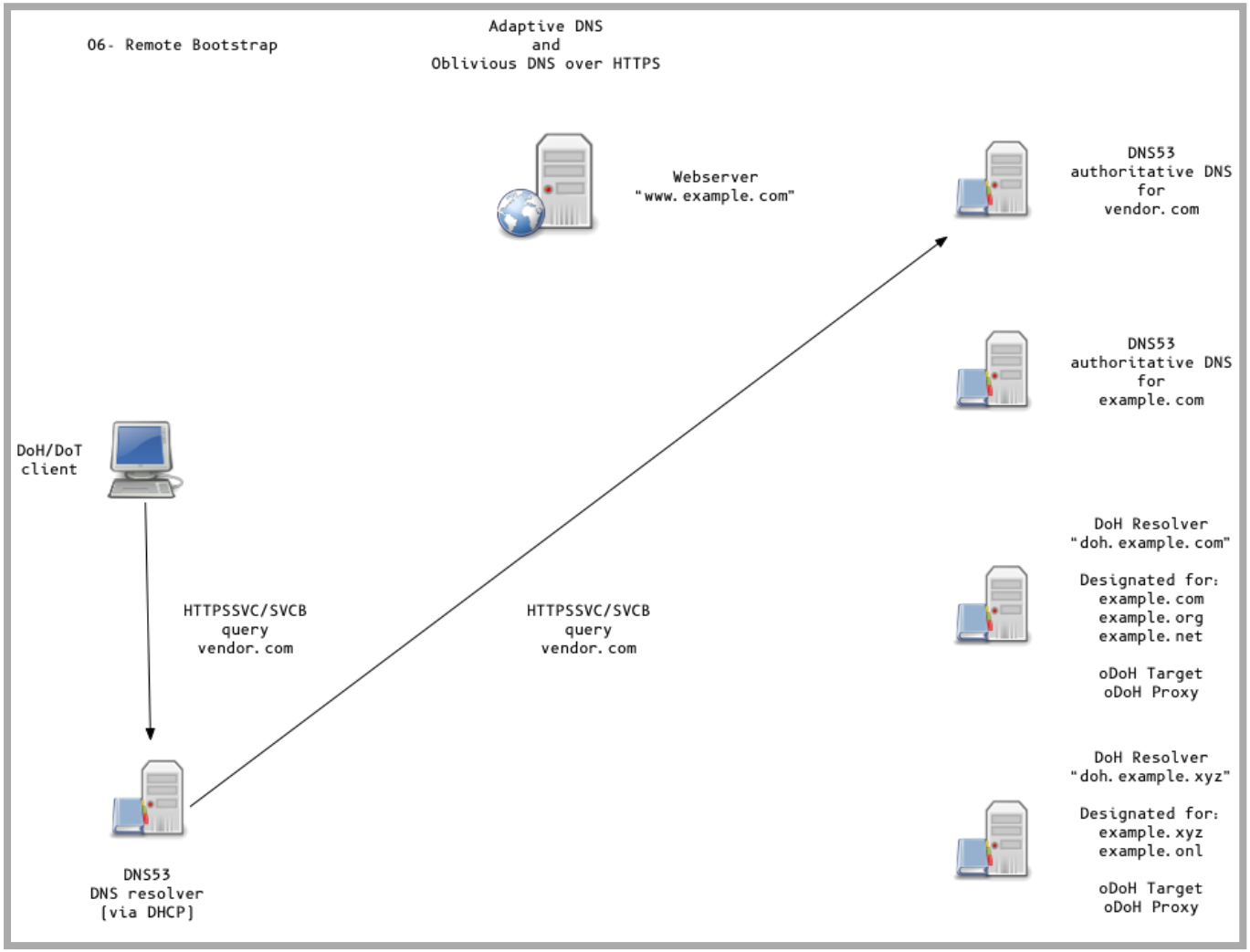
Adaptive DNS Discovery and oDoH



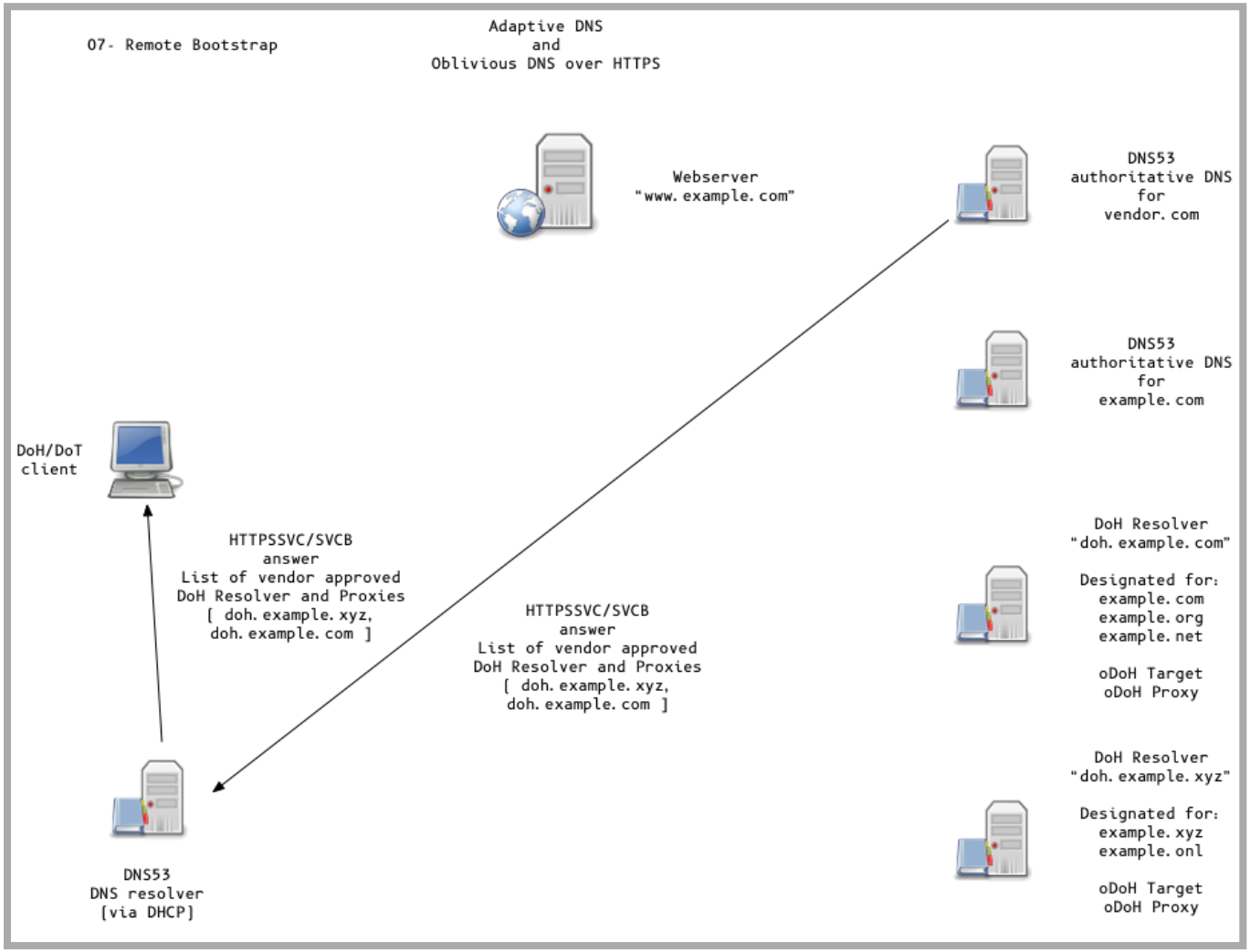
Adaptive DNS Discovery and oDoH



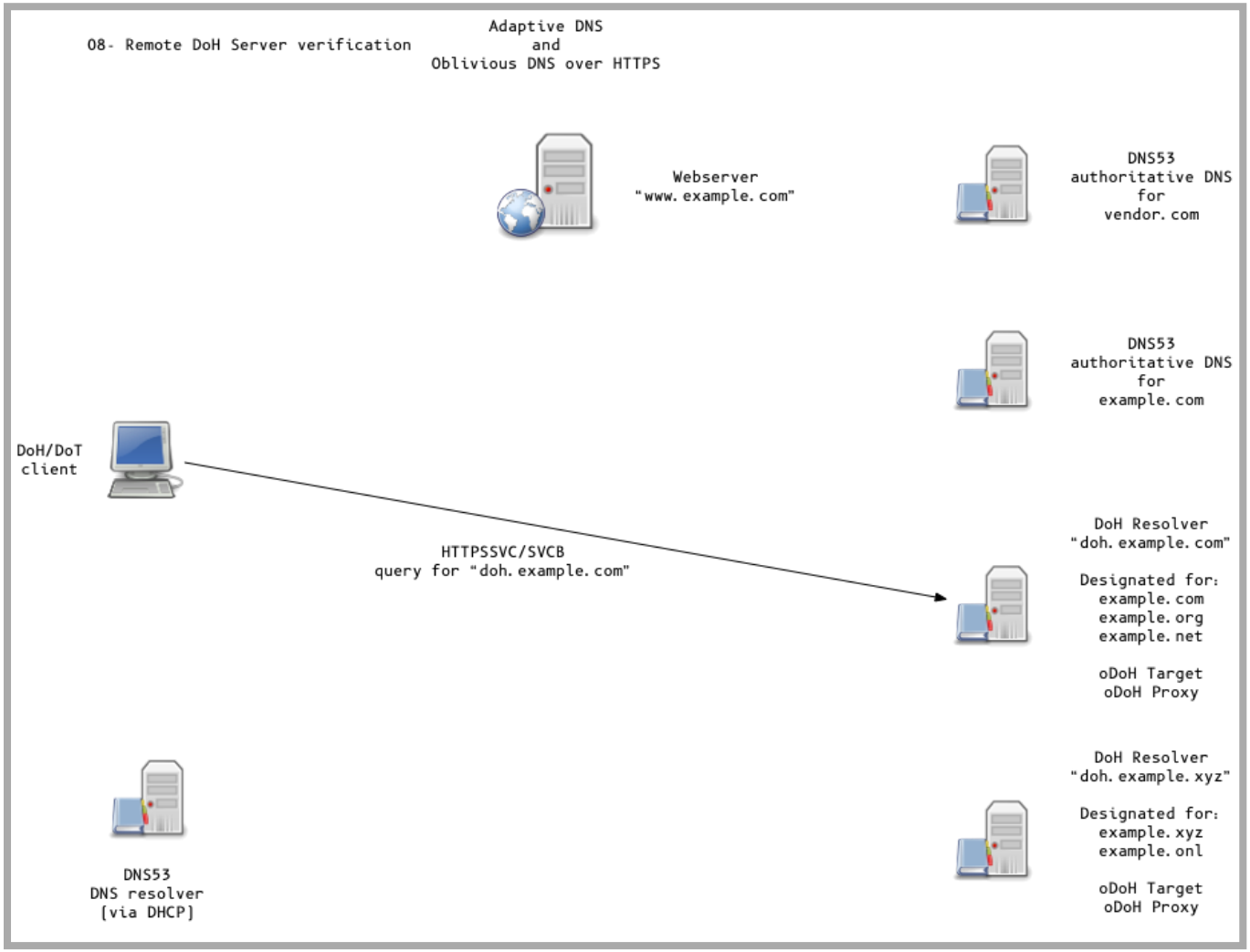
Adaptive DNS Discovery and oDoH



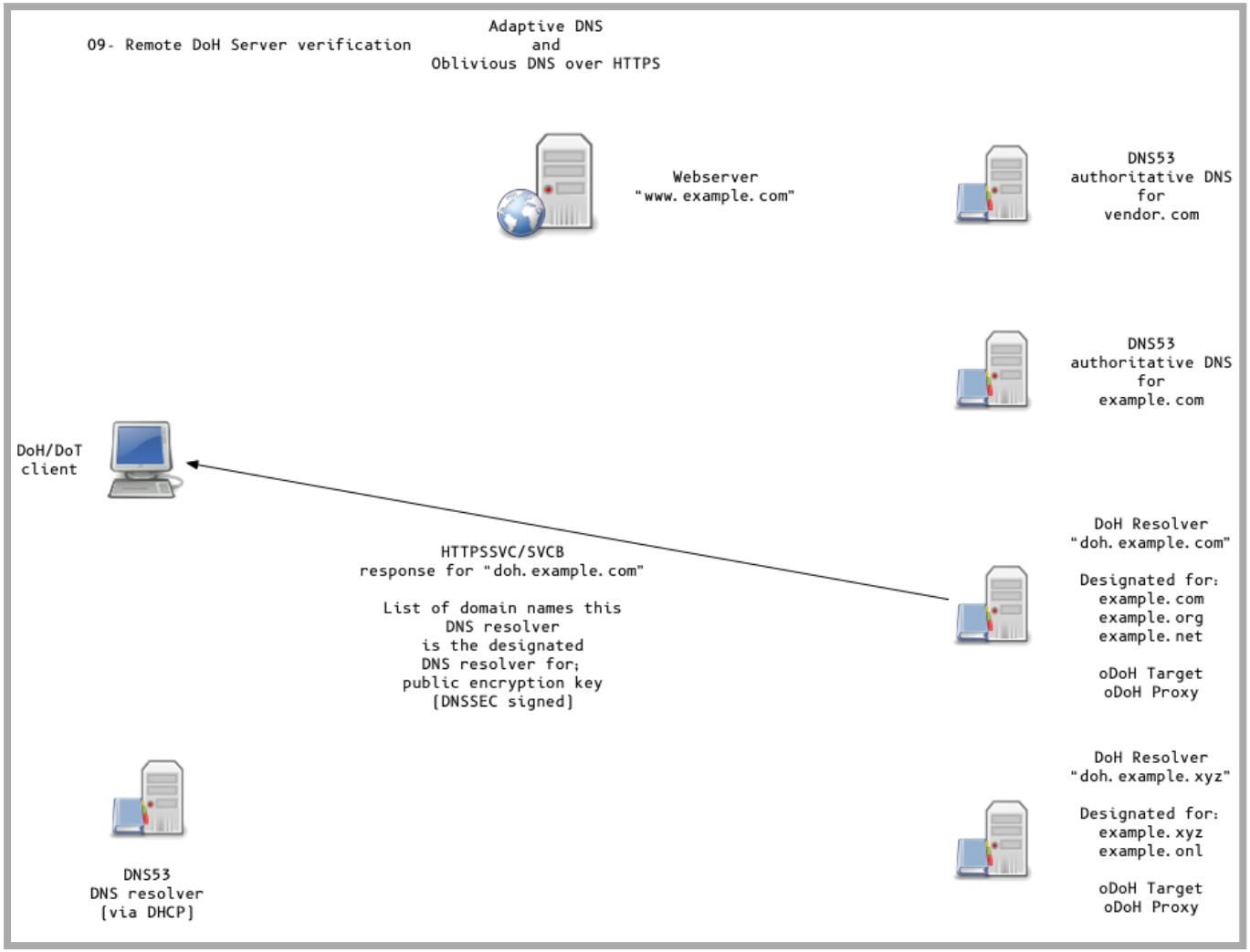
Adaptive DNS Discovery and oDoH



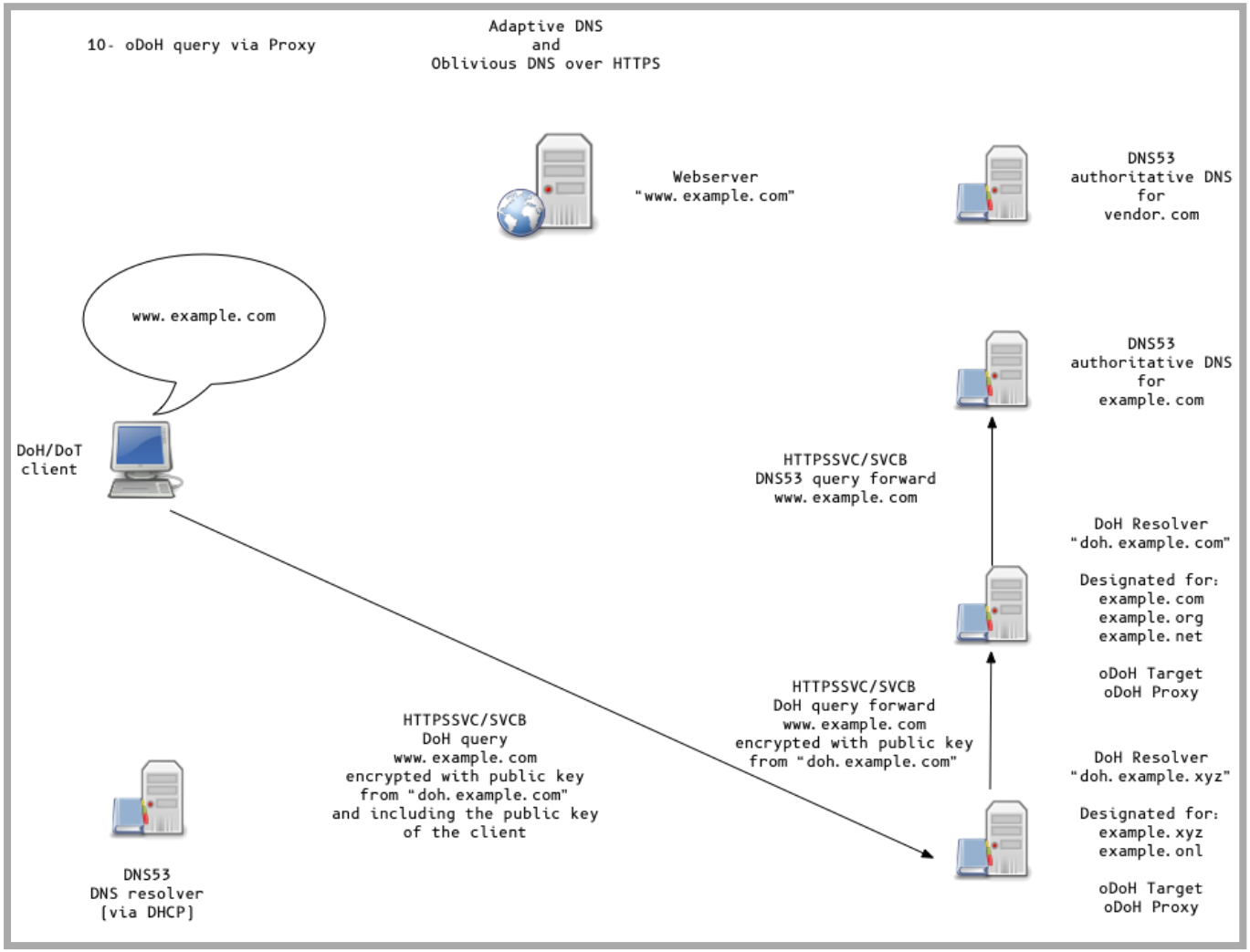
Adaptive DNS Discovery and oDoH



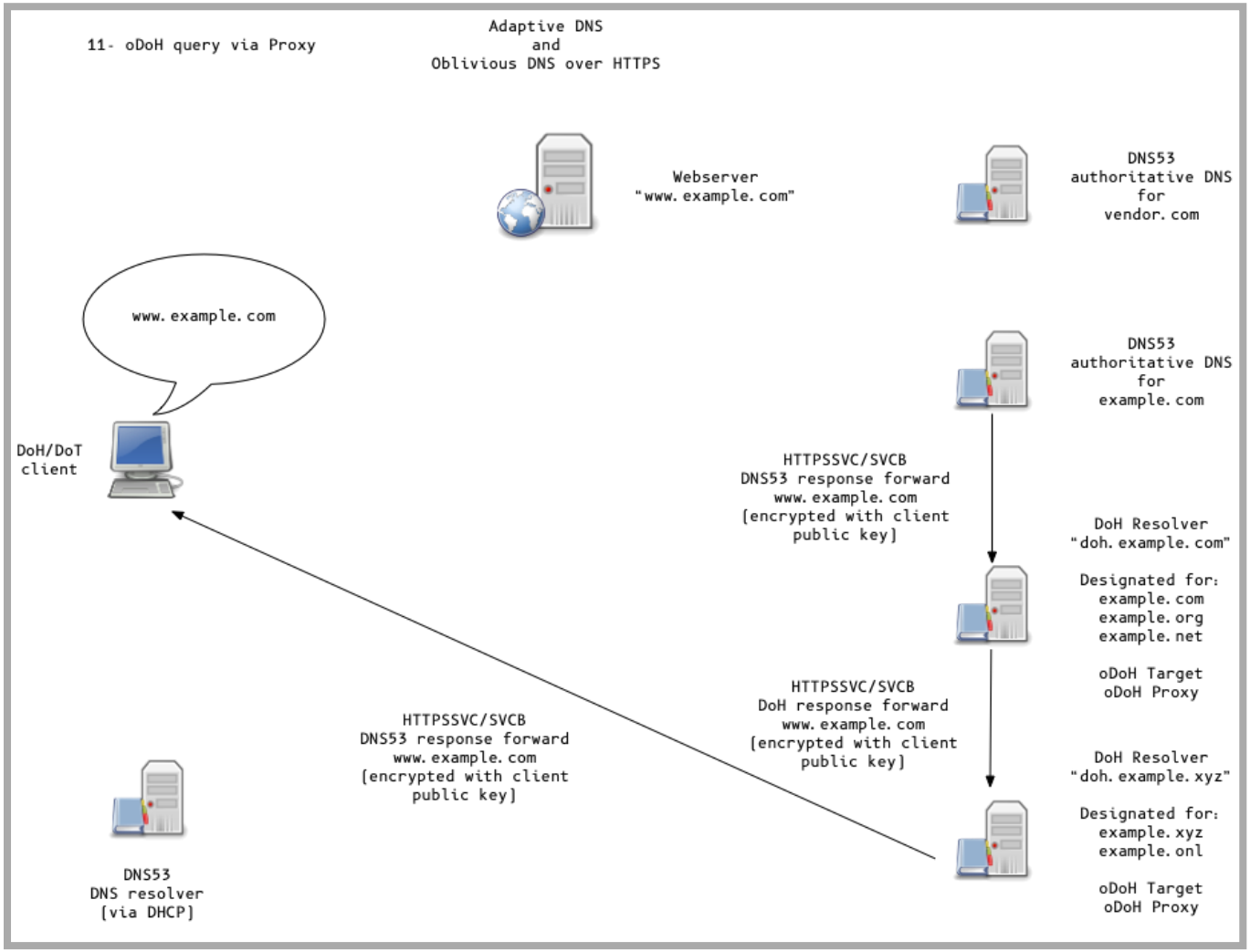
Adaptive DNS Discovery and oDoH



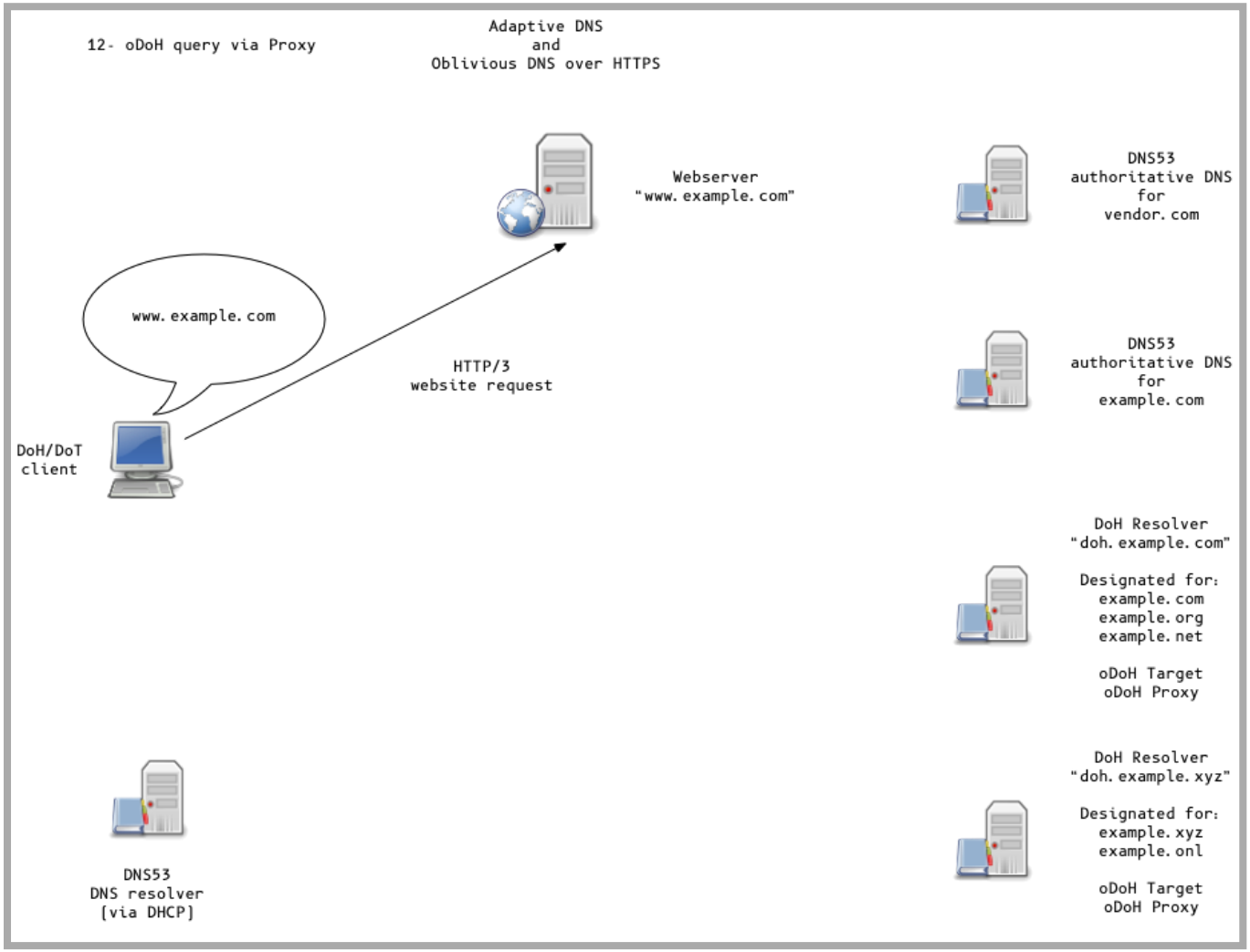
Adaptive DNS Discovery and oDoH



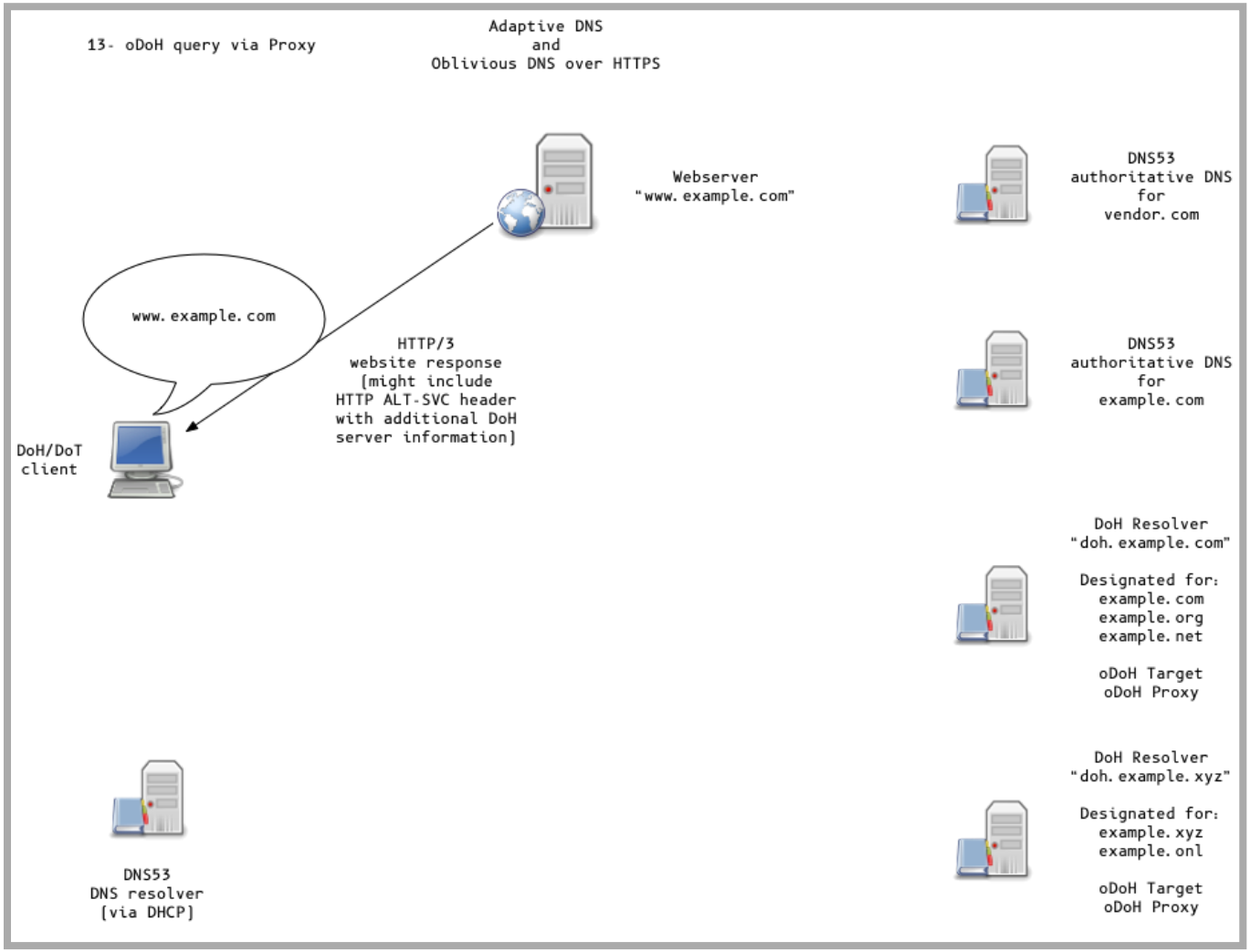
Adaptive DNS Discovery and oDoH



Adaptive DNS Discovery and oDoH



Adaptive DNS Discovery and oDoH



Thank you

Questions

Contact: carsten@strotmann.de

Links and resources