DNSCrypt und DNS-over-TLS

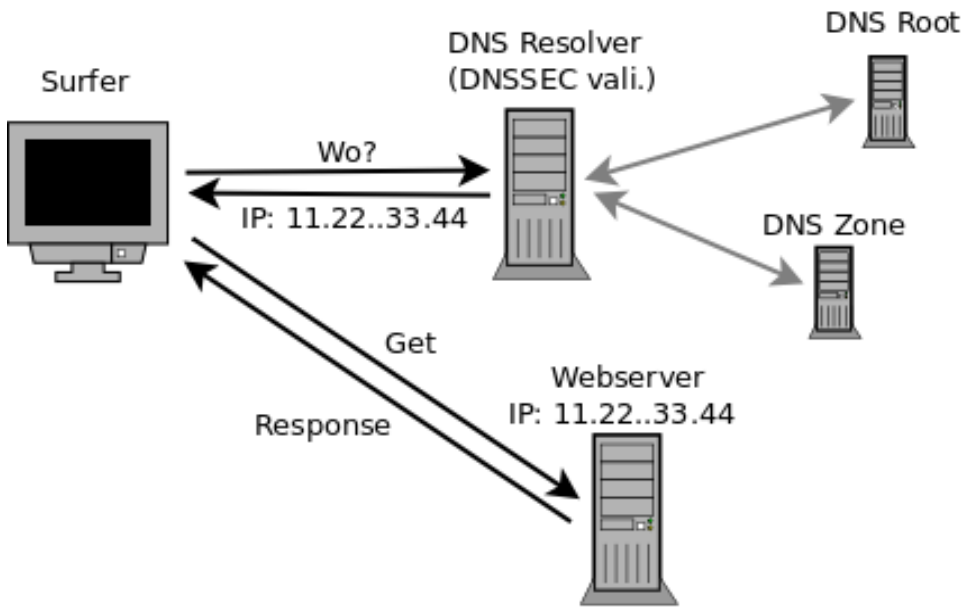
- Heinlein Support
  - IT-Consulting und 24/7 Linux-Support mit ~28 Mitarbeitern
  - Eigener Betrieb eines ISPs seit 1992
  - Täglich tiefe Einblicke in die Herzen der IT aller Unternehmensgrößen

- 24/7-Notfall-Hotline: 030 / 40 50 5 – 110
  - 28 Spezialisten mit LPIC-2 und LPIC-3
  - Für alles rund um Linux & Server & DMZ
  - Akutes: Downtimes, Performanceprobleme, Hackereinbrüche, Datenverlust
  - Strategisches: Revision, Planung, Beratung, Konfigurationshilfe

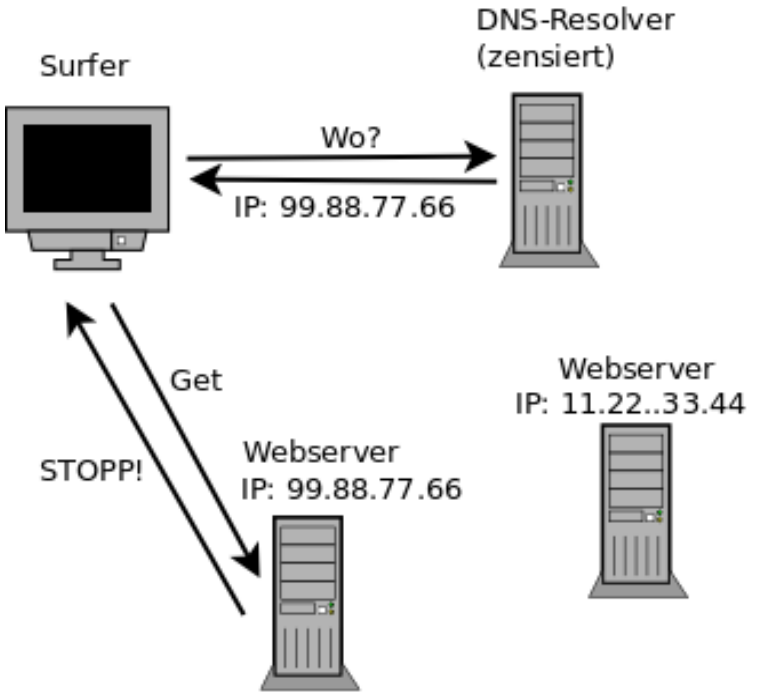
Inhaltsübersicht

- 1. Motivation
  - Anti-Zensur, DANE/TLSA, DANE/OPENPGPKEY...
- 2. Verschlüsseltes DNS
  - 2.1 DNScrypt
  - 2.2 DNS-over-TLS
  - 2.3 HTTPS-DNS
- 3. Fragen, Diskussion

1: DNS-Funktion allgemein

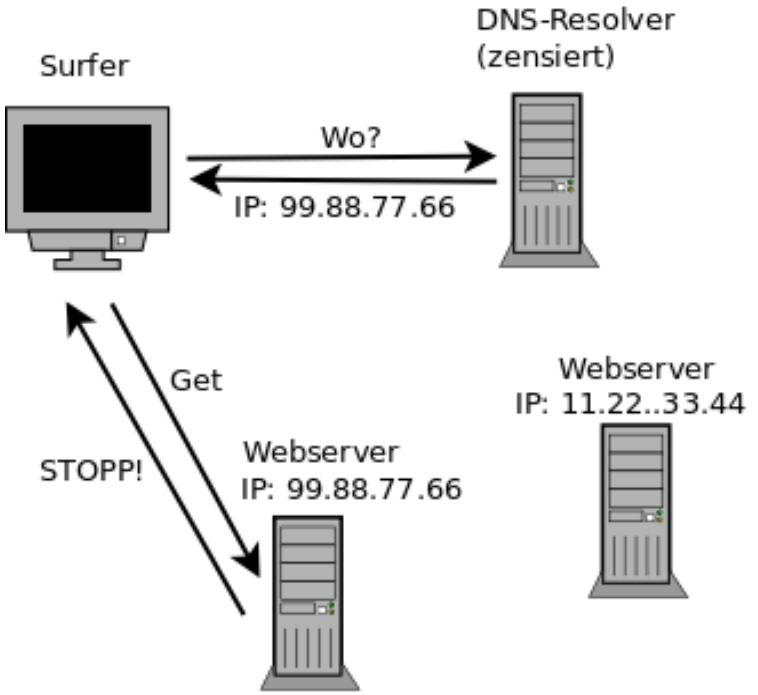


1.1: DNS-basierte Zensur



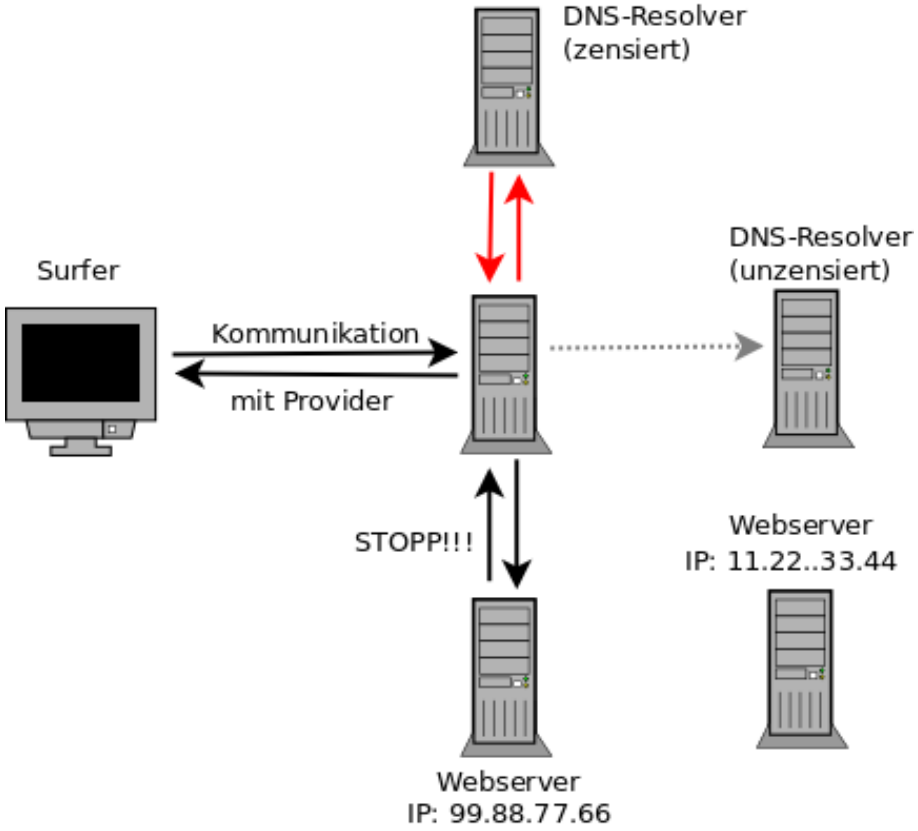
1.1: DNS-basierte Zensur

- ZugErschwG (DE, 2009/10)
- Zensur in der Türkei
- Zensur in Vietnam
- Sperrung von russischen Social Media in Ukraine
- Pläne in Großbritannien
- ....

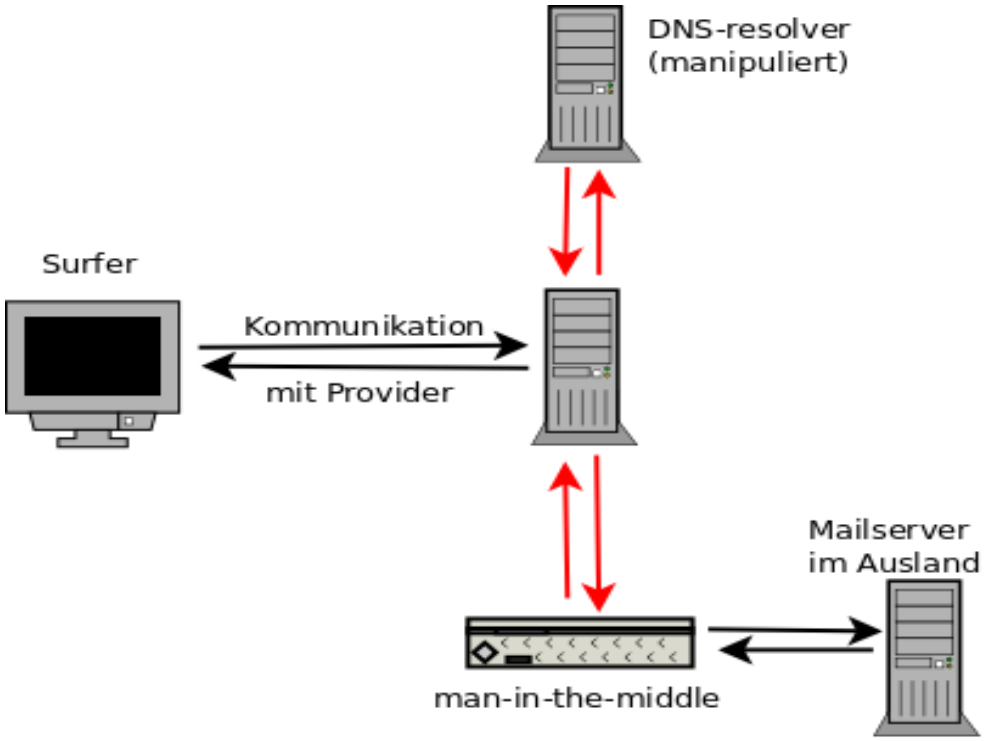


1.1: DNS-basierte Zensur (advanced)

- ZugErschwG (DE, 2009/10)
  - Provider hätten mit technischen Mittel gem. dem Stand der Technik sicherzustellen, das die Sperre für die vom BKA benannten Webseiten durchgesetzt wird.
  - Eine iptables Regel dafür nötig.
  - Im Vodafon O2 Netz z.B. üblich.



1.2: DNS-basierte Überwachung



1.3: DANE/TLSA Records im DNS

- DANE/TLSA wurde im Januar 2014 als Standard verabschiedet.
- Serverbetreiber können die SHA2-Fingerprints der Zertifikate im DANE/TLSA Record hinterlegen und mit DNSSEC signieren.
- Anwender können anhand der Information die SSL-Zertifikate verifizieren und *man-in-the-middle* Angriffe erkennen.

1.3: DANE/TLSA Records im DNS

- DANE/TLSA wurde im Januar 2014 als Standard verabschiedet.
- Serverbetreiber können die SHA2-Fingerprints der Zertifikate im DANE/TLSA Record hinterlegen und mit DNSSEC signieren.
- Anwender können anhand der Information die SSL-Zertifikate verifizieren und *man-in-the-middle* Angriffe erkennen.

- ABER: die kryptografische Kette ist nicht vollständig!
- *Letzte Meile* zwischen DNS-Resolver und User ist ungesichert.

1.3: DANE/TLSA Records nutzen

- Standardsoftware (Webbrowser, E-Mail.Clients, Jabber Clients) können die DANE/TLSA Records noch nicht out-of-box nutzen.
  - Wie könnten Firefox und Google Chrome DANE/TLSA Records nutzen?
  - Wie könnten IMAPS-, POP3S- oder SMTPS-Verbindungen verifiziert werden?
  - Wie könnten Jabber/XMPP Clients SSL-Zertifikate verifizieren?

1.3: DANE/TLSA Records mit Firefox nutzen

- Für Firefox und Google Chrome gibt es ein Add-on:

  DNSSEC/TLSA-Validator:

1.3: DANE/TLSA Records für andere Protokolle

1) Man könnte GnuTLS Source Code selbst compilieren
2) Zertifikate anhand DANE/TLSA Record prüfen mit „danetool“:

```
> danetool –-check <server> --port <port>

Resolving <server>…

Obtaining certificate from 'ip-address'…

Verification: Certificate matches.

> <Anwendung starten>
```

1.3: DANE/TLSA Records mit Startscript prüfen

```
#!/bin/bash
danetool --check smtp.mailbox.org --port 465
if [ $? -ne 0 ]; then
    zenity --error --text="DANE/TLSA Fehler bei SMTP Server!" --no-wrap
    exit 0
fi
danetool --check pop3.mailbox.org --port 995
if [ $? -ne 0 ]; then
    zenity --error --text="DANE/TLSA Fehler bei POP3 Server!" --no-wrap
    exit 0
fi
thunderbird
```

2: DNS - verschlüsselt und authentifiziert

- Zielstellung:
  1) Authentifizierung, um die Identität des DNS-Server zu verifizieren.
  2) Verschlüsselung des DNS-Traffic, um Manipulationen zu verhindern.

2: DNS - verschlüsselt und authentifiziert

→ Zielstellung:

1) Authentifizierung, um die Identität des DNS-Server zu verifizieren.
2) Verschlüsselung des DNS-Traffic, um Manipulationen zu verhindern.

Lösungen:

- **DNSCrypt** (basiert auf DNScurve von D.J. Bernstein, von OpenDNS betreut)
- **DNS-over-TLS** (RFC 7858, TLS-verschlüsselte TCP Kommunikation, Port: 853)
- **HTTPS-DNS** (Google Projekt, DNS-Informationen über HTTPS-Protokoll verteilen)

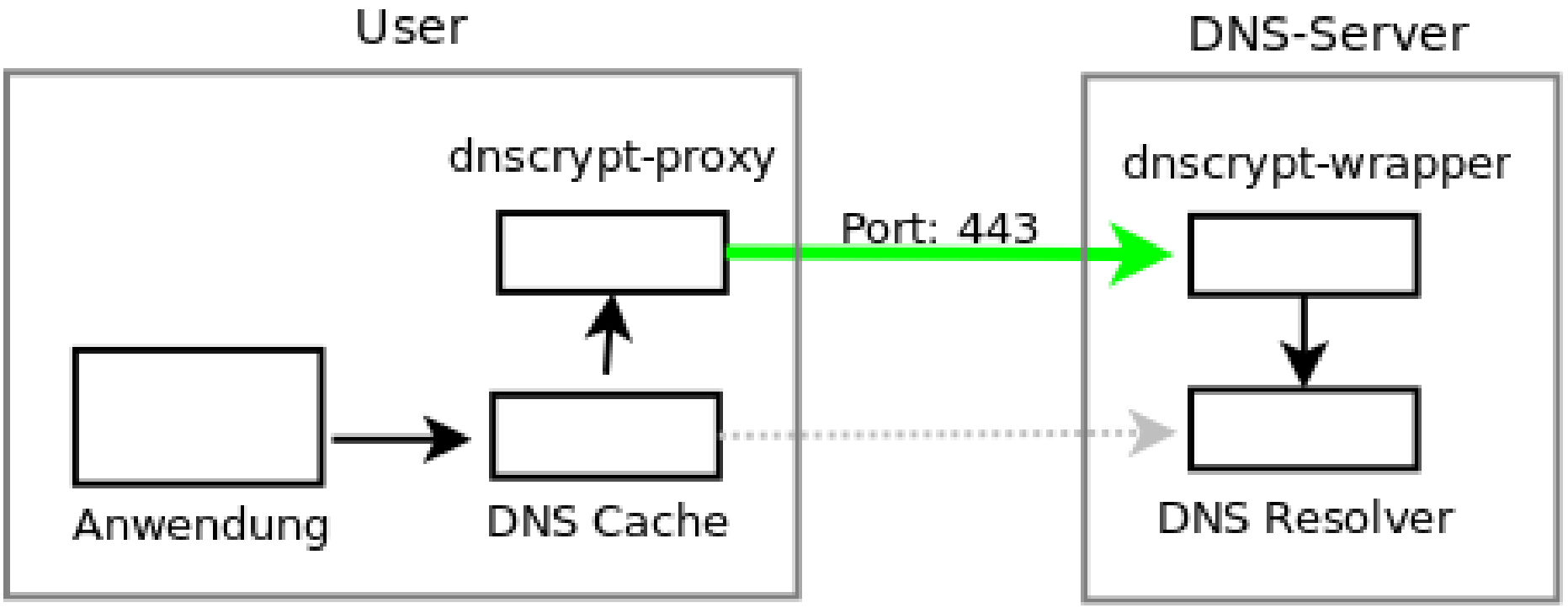
2.1: DNSCrypt

- Clientsoftware: dnscrypt-proxy
  - Agiert als Upstream DNS-Server für den lokalen DNS Cache beim User
  - Der Datenverkehr zum DNS Resolver wird mit Public Key Verfahren auf Basis der elliptischen Kurve Curve25519 von D.J. Bernstein verschlüsselt.

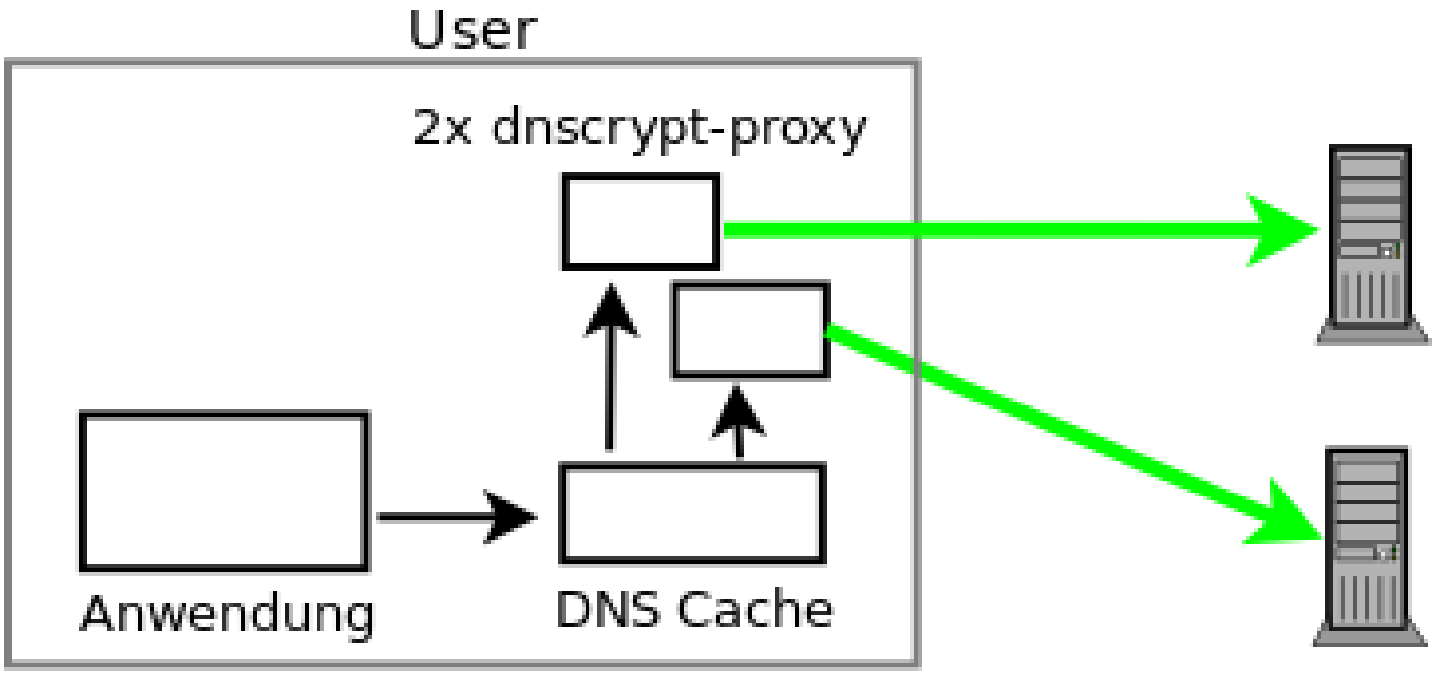
2.1: DNSCrypt

- Clientsoftware: dnscrypt-proxy
  - Agiert als Upstream DNS-Server für den lokalen DNS Cache beim User
  - Der Datenverkehr zum DNS Resolver wird mit Public Key Verfahren auf Basis der elliptischen Kurve Curve25519 von D.J. Bernstein verschlüsselt.
- Serversoftware: dnscrypt-wrapper
  - Nimmt verschlüsselte Anfragen entgegen, entschlüsselt sie und reicht sie an einen echten DNS Resolver weiter, verschlüsselt die Anwortenan Client.
  - Krypto-Schlüssel der Server werden „per Hand“ verteilt, keine CAs o.ä.
  - Unbound 1.6.2 (Apr. 2017) bietet einen build-in dnscrypt-wrapper, wenn er mit der Option –-enable-dnscrypt compiliert wurde.

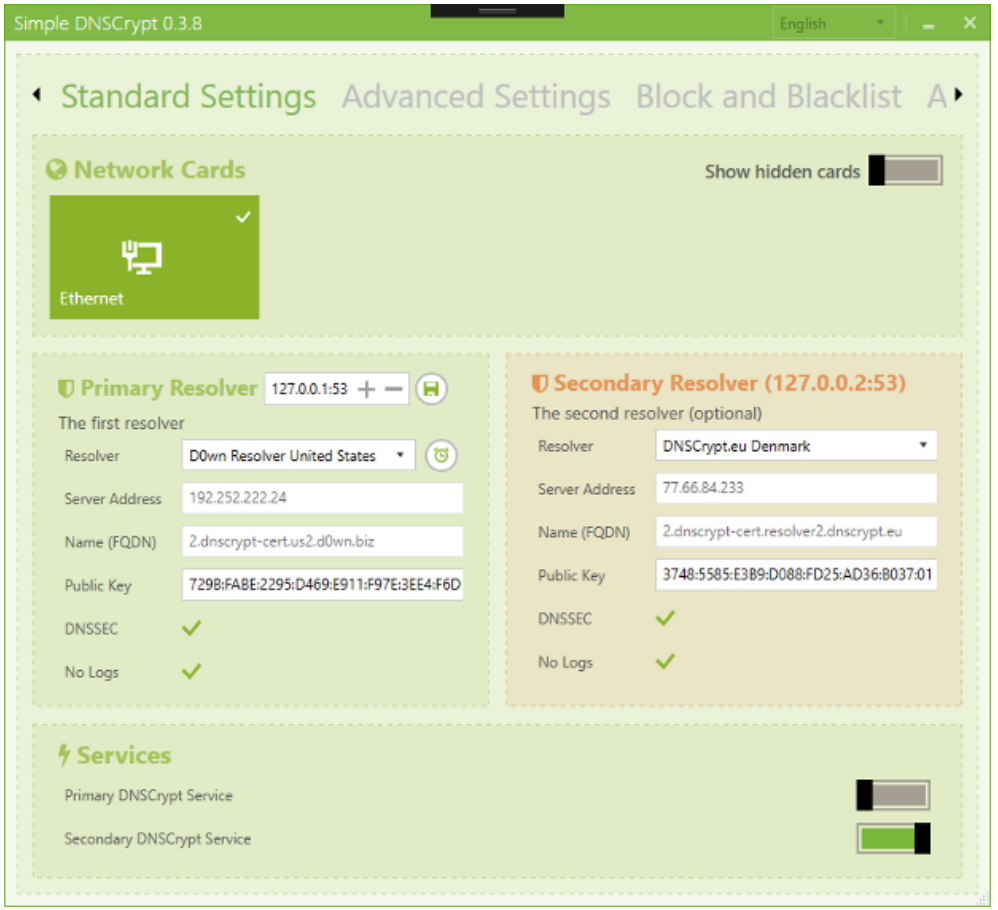
2.1: DNSCrypt


User
DNS-Server
dnscrypt-proxy
dnscrypt-wrapper
Port: 443
Anwendung
DNS Cache
DNS Resolver

2.1: DNSCrypt



2.1: SimpleDNSCrypt für Windows



2.1: dnscrypt-proxy für Linux

- Installation aus den Repositories der Distries möglich aber,
  - In der Regel veraltete Versionen, nicht aktualisiert
  - Suboptimale Konfiguration ohne DNS Cache und nur ein Upstream Server: Application → dnscrypt-proxy → Server

2.1: dnscrypt-proxy für Linux

- Installation aus den Repositories der Distries möglich aber,
  - In der Regel veraltete Versionen, nicht aktualisiert
  - Suboptimale Konfiguration ohne DNS Cache und nur ein Upstream Server: Application → dnscrypt-proxy → Server

- Besser: akt. Source Code herunter laden und compilieren
  - Ein kleines Script erstellen, das 2-3 Instanzen startet
  - Config des lokalen DNS-Cache anpassen und die dnscrypt-proxys als Upstream DNS-Server eintragen
  - dnscrypt-proxy regelmäßig aktualisieren!

2.2: DNS-over-TLS (RFC 7818)

- TLS-verschlüsselte TCP Kommunikation, Default-Port: 853
  - Der gesamte DNS-Traffic könnte verschlüsselt werden
  - Verifizierung der Server Zertifikate via CAs
  - Kein „STARTTLS“ für automatisches Upgrade auf TLS
  - Keine Advanced Features wie HSTS, OCSP.Stapling usw.

2.2: DNS-over-TLS (RFC 7818)

- DNS Server Software mit DNS-over-TLS Support:
  - Knot, ldns und Unbound beherrschen DNS-over-TLS
  - Für PowerDNS und andere steht es auf der ToDo Liste
  - Jeder DNS-Server kann mit stunnel aufgemotzt werden:

```
[dns]
accept = 853
connect = 127.0.0.1:53
cert = dns.crt
key = dns.key
```

2.2: DNS-over-TLS (RFC 7818)

Software für Clients:

1) DNS-Cache-Server mit DNS-over-TLS Support installieren und Forwarder konf. (Unbound: „<server-ip> at 853“)
2) Für dnsmasq o.ä. könnte man ebenfalls stunnel nutzen:

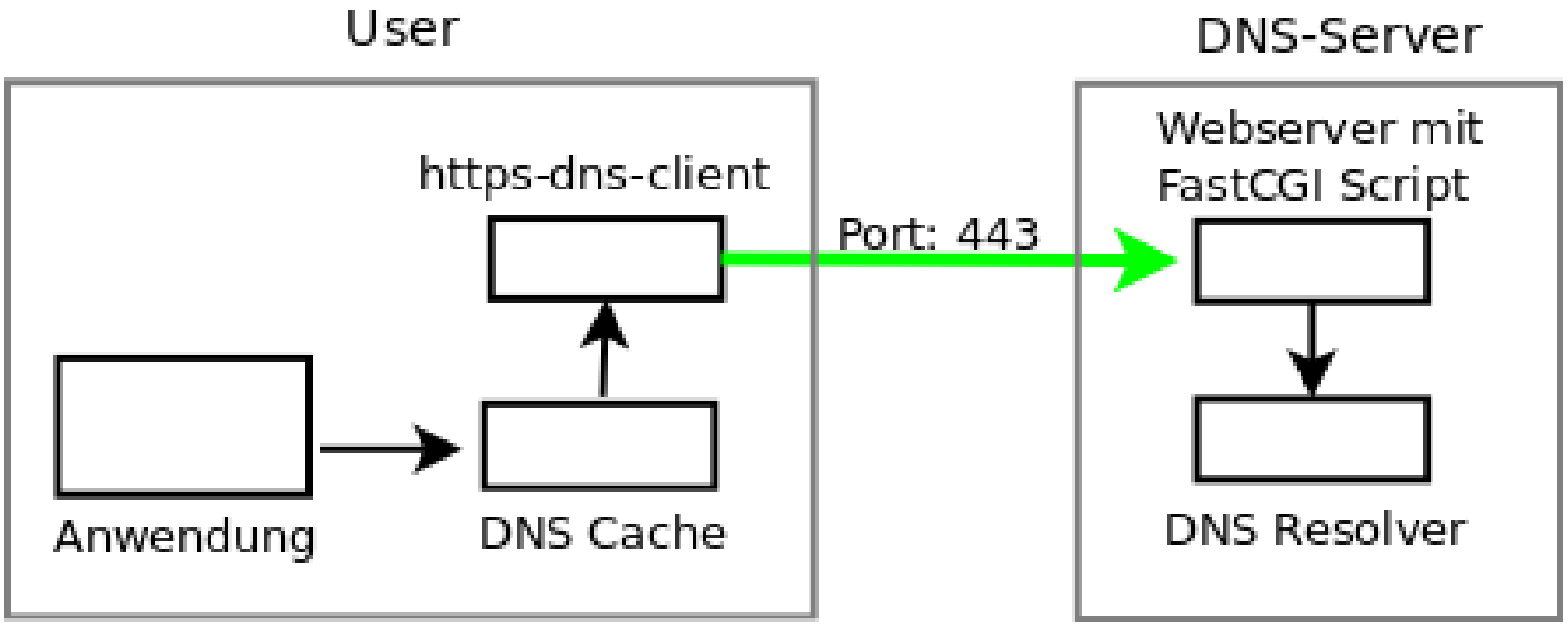
```
[dns1]
client = yes
Accept = 127.0.2.1:53
connect = <server-ip>:853
[dns2]
client = yes
Accept = 127.0.3.1:53
connect = <server-ip>:853
```

2: Probleme mit DNSCrypt und DNS-over-TLS

- Login-Probleme bei **WiFi Hotspots** (z.B. WiFionICE):
  - Redirekt auf die Captive Portal Page funktioniert nicht
  - Entweder man kennt die IP der Captive Portal Page
  - Oder man muss auf den DNS-Server des WiFi Hotspot umschalten und nach dem Login DNSCrypt bzw. DNS-over-TLS wieder aktivieren

- Bei einigen Hotspots sind nur Port 80 und 443 freigeschaltet
  - DNS-over-TLS kann dann nicht genutzt werden

2.3: HTTPS-DNS


User
DNS-Server
https-dns-client
Port: 443
Webserver mit
FastCGI Script
Anwendung
DNS Cache
DNS Resolver

2.3 HTTPS-DNS (Google DNS)

- Die Google DNS Server bieten ein HTTPS-DNS Interface:

  https://dns.google.com/query?… (Human readable)

  https://dns.google.com/resolve?… (JSON response)
- Kommunikationsprotokoll ist offen (JSON Datenstrukturen)
- Keine Clients verfügbar, könnten aber entwickelt werden, dabei sind die verbleibenden Probleme zu lösen:
  - Initiale IP-Adresse für dns.google.com (z.B. via /etc/hosts)
  - Zertificate Pinning oder mind. CA-Pinning implementieren

3: Fragen und Diskussion?