DeLegacy

Erzwinge IPv6 im großen Stil

Mynacol

Tübix '26
Sand, Tübingen

2026-07-04



Agenda

1. Das Problem
2. Konzept
3. Implementierung
   - 3.1 DNS RPZ
   - 3.2 DNAMEs
   - 3.3 DNS64
   - 3.4 Serviceunterstützung
   - 3.5 Offene Probleme
4. Nebenwirkungen
5. OSINT für DeLegacy
6. Auswirkungen auf das Ökosystem
7. Zusammenfassung

0 Über mich



Hi, ich bin Mynacol!

▸ Kontakt

- codeberg.org/Mynacol
- @mynacol@mynacol.xyz
- @mynacol@mynacol.xyz
- mynacol@mynacol.xyz

0 Über mich



Hi, ich bin Mynacol!

Studium

- Doktorand an der Uni Ulm
- Forschung zu Netzwerken, Sicherheit und anonymer Kommunikation wie Tor
- Teil des bwNET2.0-Forschungsprojekts sowie der KIZ-Abteilung für Netzwerk

Kontakt

- codeberg.org/Mynacol
- @mynacol@mynacol.xyz
- @mynacol@mynacol.xyz
- mynacol@mynacol.xyz

0 Über mich



Hi, ich bin Mynacol!

Studium

- Doktorand an der Uni Ulm
- Forschung zu Netzwerken, Sicherheit und anonymer Kommunikation wie Tor
- Teil des bwNET2.0-Forschungsprojekts sowie der KIZ-Abteilung für Netzwerk

Kontakt

- codeberg.org/Mynacol
- @mynacol@mynacol.xyz
- @mynacol@mynacol.xyz
- mynacol@mynacol.xyz

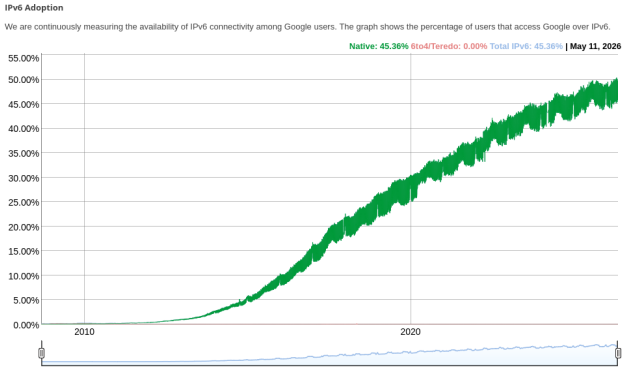
Interessen

- Seit Jahren IPv6 Enthusiast
- Hauptentwickler von DeLegacy
- Contributor von Nixpkgs
- Maintainer von RSS-Bridge

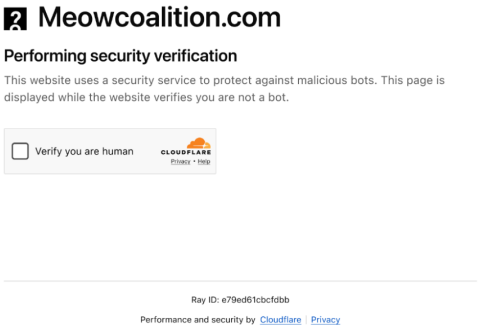
1.1 Das Problem

IPv6 wird gewinnen, die einzige Frage ist *wie schnell*!

- **Wie können wir dazu beitragen, dass mehr Websites IPv6 unterstützen?**
  - Als Client
  - Ohne Übergangsmechanismen (*z.B.* 464XLAT)

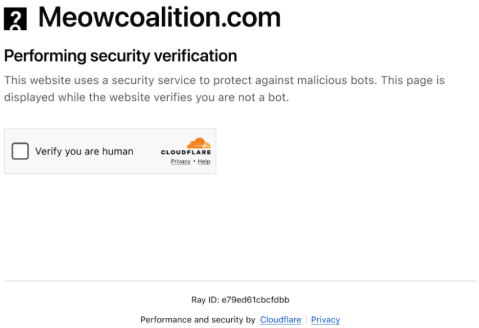


1.2 Content Delivery Networks (CDNs)



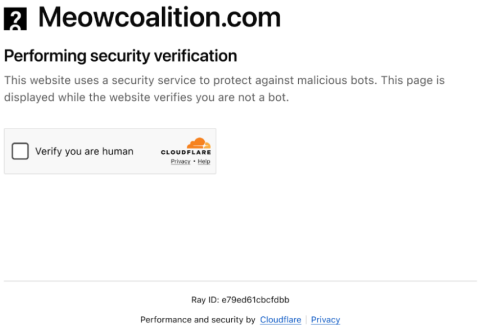
- Kennst du diese Zwischenseite?

1.2 Content Delivery Networks (CDNs)



- Kennst du diese Zwischenseite?
- Cloudflare fronted ~20% des Internets
- Cloudflare unterstützt seit 2012 IPv6
- Dennoch fehlt vielen Domains, die Cloudflare nutzen, IPv6

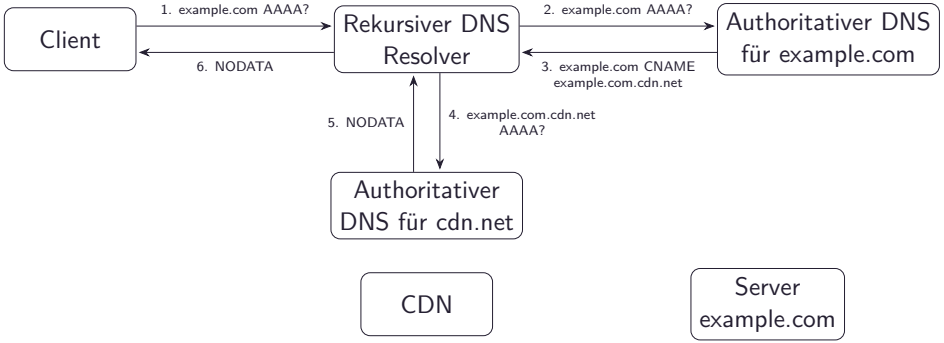
1.2 Content Delivery Networks (CDNs)



- Kennst du diese Zwischenseite?
- Cloudflare fronted ~20% des Internets
- Cloudflare unterstützt seit 2012 IPv6
- Dennoch fehlt vielen Domains, die Cloudflare nutzen, IPv6
- Können wir das beheben? 🤔
- Das Gleiche gilt für alle CDNs

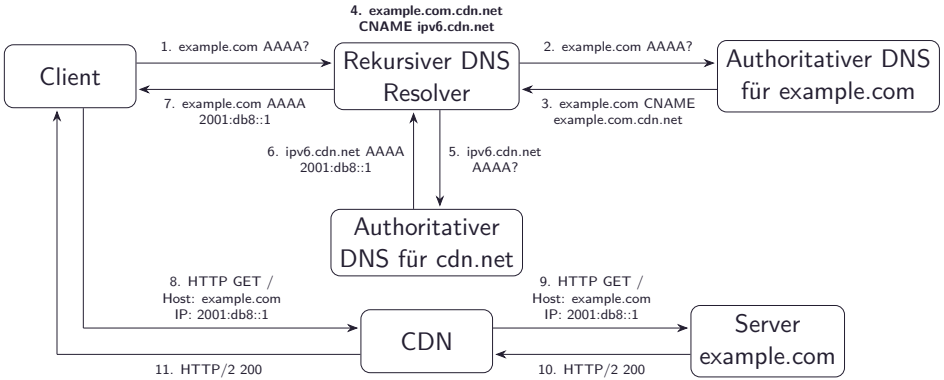
2.1 Grundidee

Wir ändern DNS-Antworten so, dass der Client mit IPv6 eine Verbindung aufbauen kann



2.1 Grundidee

Wir ändern DNS-Antworten so, dass der Client mit IPv6 eine Verbindung aufbauen kann

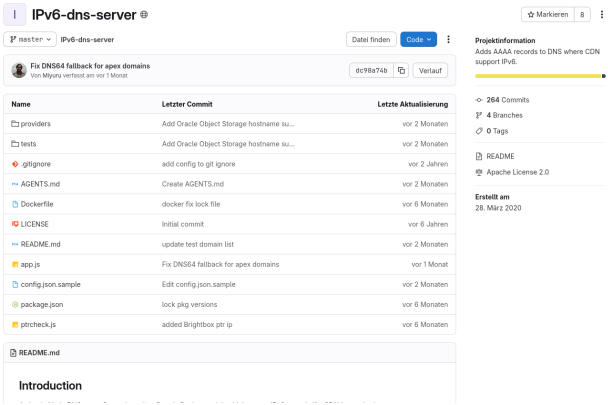


2.2 Die richtigen IPv6-Adressen finden

Oder: Wie richten Menschen Delegation ein

- Man verwendet eine vom CDN bereitgestellte Subdomain
  - *Z.B.* GitHub Pages und `tasmota.github.io`
  - ▸ Redirect Domain zu einer mit AAAA
- Man delegiert eine Subdomain über CNAME/ANAME/ALIAS an eine vom CDN bereitgestellte Domain
  - *Z.B.* Akamai und `www.zdf.de CNAME ssl.zdf.de.edgekey.net`
  - ▸ Redirect Domain zu einer mit AAAA
- Man delegiert eine Subdomain über feste IPs an das CDN
  - *Z.B.* AWS Cloudfront und `www.figma.com`
  - Keine Domain zum Weiterleiten › DNS64
- Das CDN ist dein autoritativer DNS
  - *Z.B.* Cloudflare und `discord.com`
  - Keine Domain zum Weiterleiten › DNS64

3.1 IPv6-DNS-Server



gitlab.com/miyurusankalpa/IPv6-dns-server/



von Miyuru Sankalpa

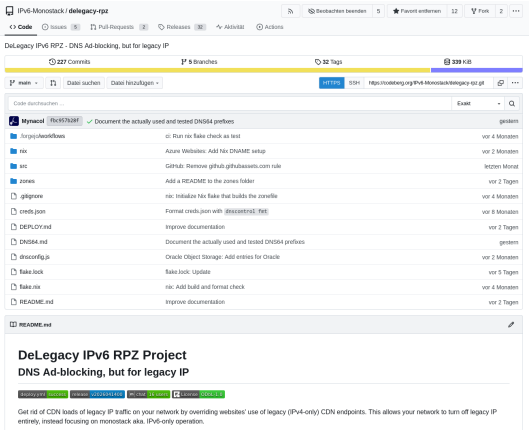
miyuru.lk

3.2 Projekt



Logo: ahiiti

3.3 Repository



https://codeberg.org/IPv6-Monostack/delegacy-rpz

3.1 DNS Response Policy Zones

- (Semi-)Standardmäßige Methode, DNS-Antworten als DNS-Zonen umzuschreiben
- Teilweise unterstützt von BIND, Unbound, Knot, PowerDNS
- Trigger: *Abfragename (QNAME)*, Antwort-IP, Nameserver, Client-IP
- Aktion: Blockieren, Verwerfen, Passthru, *Lokale Daten*
- Weitere Informationen: `https://dnsrpz.info/`

```
*.cdn.cloudflare.net CNAME cloudflare.net.cdn.cloudflare.net.
cloudflare.net.cdn.cloudflare.net CNAME rpz-passthru.
```

3.2.1 Akamai

- Akamai delegiert mit Unicast GeoDNS

```
www.nvidia.com CNAME www.nvidia.com.edgekey.net
www.nvidia.com.edgekey.net CNAME e33907.a.akamaiedge.net
e33907.a.akamaiedge.net A 92.123.12.149
```

- `e33907.a.akamaiedge.net` ist IPv4-only
- Die Dual-Stack-Variante befindet sich unter `e33907.dsca.akamaiedge.net`
- Das Präfix `e33907` muss übereinstimmen
- Wir können `*.a.akamaiedge.net` nicht als CNAME auf `dsca.akamaiedge.net` oder `12345.dsca.akamaiedge.net` abbilden
- Genau das machen DNAMEs!

3.2.2 DNAMEs

- CNAMEs delegieren eine Domain an eine andere Domain
- DNAMEs delegieren einen gesamten Subdomain-Baum an eine andere Domain
- Umsetzung als bei Bedarf dynamisch synthetisierte CNAMEs für angeforderte Subdomains

```
source.net DNAME target.net
a.b.source.net CNAME a.b.target.net
```

- DNAMEs werden in DNS RPZs nicht unterstützt
- Es gäbe nicht gelöste Präzedenzregeln zwischen CNAMEs und DNAMEs
- Stattdessen können wir Akamai-Subdomains als primäre Zonen hinzufügen

3.3.1 DNS64

- *Viele* Domains verwenden keine CNAMEs und antworten direkt mit IP-Adressen (statische IPs, CNAME-Flattening/ANAME/ALIAS)

3.3.1 DNS64

- *Viele* Domains verwenden keine CNAMEs und antworten direkt mit IP-Adressen (statische IPs, CNAME-Flattening/ANAME/ALIAS)

- DNS64 bildet IPv4-Adressen auf IPv6-Adressen ab (RFC 6147)
- Unter einem gewählten Präfix, üblicherweise `64:ff9b::/96`
- *Z.B.* `192.0.2.1` zu `64:ff9b::192.0.2.1`

3.3.1 DNS64

- *Viele* Domains verwenden keine CNAMEs und antworten direkt mit IP-Adressen (statische IPs, CNAME-Flattening/ANAME/ALIAS)

- DNS64 bildet IPv4-Adressen auf IPv6-Adressen ab (RFC 6147)
- Unter einem gewählten Präfix, üblicherweise `64:ff9b::/96`
- *Z.B.* `192.0.2.1` zu `64:ff9b::192.0.2.1`

- Einige CDNs antworten auf *jede* IPv6-Adresse innerhalb eines Präfixes
  - Cloudflare, AWS Cloudfront
- Mit DNS64 die IPv4-Bereiche des CDNs auf IPv6-Adressen abbilden
- *Z.B.* `173.245.48.1` zu `2606:4700::173.245.48.1` (Cloudflare)

3.3.2 DeLegacy DNS64 mit BIND

```
acl "cloudflare-v4" {
    # https://www.cloudflare.com/ips-v4
    173.245.48.0/20;
    [...]
};
options {
    dns64 2606:4700::/96 {
        mapped {cloudflare-v4;};
    };
};
```

- Nur BIND erlaubt es, DNS64 auf eine spezifische Liste von IPv4-Präfixen anzuwenden
- Unmöglich, bestimmte fehlerhafte Domains auszunehmen
  - Workaround: setze eine ungültige IPv6-Adresse und hoffe auf Happy Eyeballs

3.4 Serviceunterstützung

- Cloudflare
- AWS CloudFront
- Fastly
- Akamai
- Sucuri
- Bunny
- CDN77
- GCore
- Blazing
- CacheFly

- AWS S3
- Alibaba OSS
- Oracle Object Storage
- GitHub
- Google Scholar
- Netlify
- pool.ntp.org
- Wordpress VIP
- WP Engine
- ...

Supportumfang variiert je nach Service

3.5 Offene Probleme

- DNAMEs werden in RPZ nicht unterstützt
  - Standardisieren und Implementierung in Resolvern
  - Alternative: Als primäre Zone einrichten
- Weiterleitungen basierend auf einer aufgelösten IPv4 sind nicht möglich
  - Matching möglich mit `32.1.2.0.192.rpz-ip`, aber nur für A Queries
  - AAAA Queries matchen nicht, daher werden sie nicht modifiziert
  - Cached Weiterleitungen der Domain statt auf der Antwort-IP könnten AAAA Queries beeinflussen
  - Alternative: DNS64 verwenden, wo möglich
- Komplexe Logik
  - Vorab berechnen und zu einfachen Weiterleitungen abflachen
  - Teilweise ungelöst, *z.B.* für die Berechnung der Fastly-IPv6-Adressen

4 Nebenwirkungen

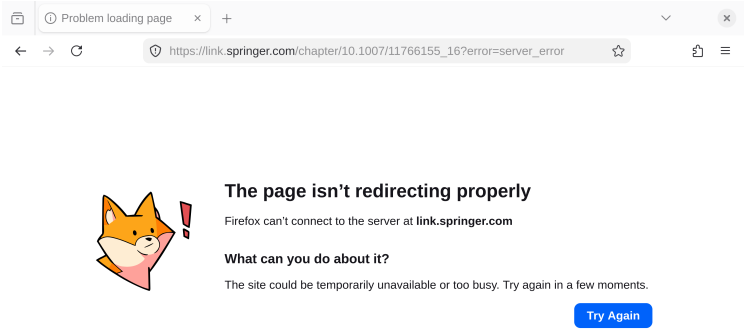
```
~ curl https://ins.blubrry.com/2665852/IPB201.mp3 -v -Ss
* Host ins.blubrry.com:443 was resolved.
* IPv6: 2600:9000:2670:6200:2:1912:75c0:93a1, 2600:9000:2670:d000:2:1912:75c0:93a1, 2600:9000:2670:4e00:2:1912:75c0:93a1, 2600:9000:2670:9200:2:1912:75c0:93a1, 2600:9000:2670:5000:2:1912:75c0:93a1, 2600:9000:2670:e00:2:1912:75c0:93a1, 2600:9000:2670:ba00:2:1912:75c0:93a1, 2600:9000:2670:ce00:2:1912:75c0:93a1
* IPv4: 13.32.121.21, 13.32.121.111, 13.32.121.20, 13.32.121.40
*   Trying [2600:9000:2670:6200:2:1912:75c0:93a1]:443...
* ALPN: curl offers h2,http/1.1
* ALPN: server accepted h2
* rustls: handshake complete, TLSv1.3, ciphersuite: TLS13_AES_128_GCM_SHA256, key exchange group: X25519MLKEM768
* Established connection to ins.blubrry.com (2600:9000:2670:6200:2:1912:75c0:93a1 port 443) from 2001:7c0:3101:1a01:e25:ab94:2f08:2cef port 58194
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://ins.blubrry.com/2665852/IPB201.mp3
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: ins.blubrry.com]
* [HTTP/2] [1] [:path: /2665852/IPB201.mp3]
* [HTTP/2] [1] [user-agent: curl/8.19.0]
* [HTTP/2] [1] [accept: */*]
* [HTTP/2] [1] [accept-encoding: deflate, gzip, br, zstd]
> GET /2665852/IPB201.mp3 HTTP/2
> Host: ins.blubrry.com
> User-Agent: curl/8.19.0
> Accept: */*
> Accept-Encoding: deflate, gzip, br, zstd
>
* Request completely sent off
< HTTP/2 502
< content-type: application/json
< content-length: 36
< date: Mon, 01 Jun 2026 10:33:33 GMT
< x-amz-apigw-id: eRvvsFa7oAMElNQ=
< x-amzn-requestid: 6b528b67-3e28-48c4-80d9-1a9a71d2809f
< x-amzn-errortype: InternalServerErrorException
< x-cache: Error from cloudfront
< via: 1.1 16cea8ae3ccd098a5d0b3b2c45b25a84.cloudfront.net (CloudFront)
< x-amz-cf-pop: FRA56-P9
< x-amz-cf-id: 40VazlBIuvTtthSxT3ynZ-sz-Y5md0_jxjO-aGDculoXRoWqQgH4hw==
<
* Connection #0 to host ins.blubrry.com:443 left intact
{"message": "Internal server error"}⏎
~ ▮                                                                    365ms
```

- Manche Websites sind kaputt
- RPZ: ausgenommen mit

  ```
  example.com
    CNAME
    rpz-passthru.
  ```

- DNS64: Ausnahmen nicht möglich

4 Nebenwirkungen

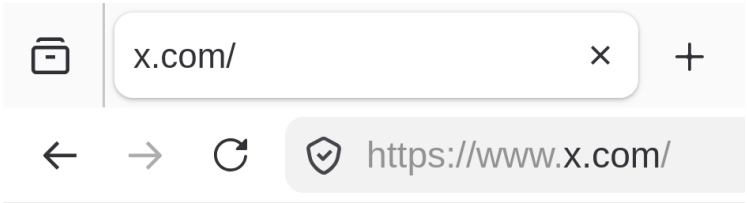


- Manche Websites sind kaputt
- RPZ: ausgenommen mit

```
example.com
  CNAME
  rpz-passthru.
```

- DNS64: Ausnahmen nicht möglich

4 Nebenwirkungen



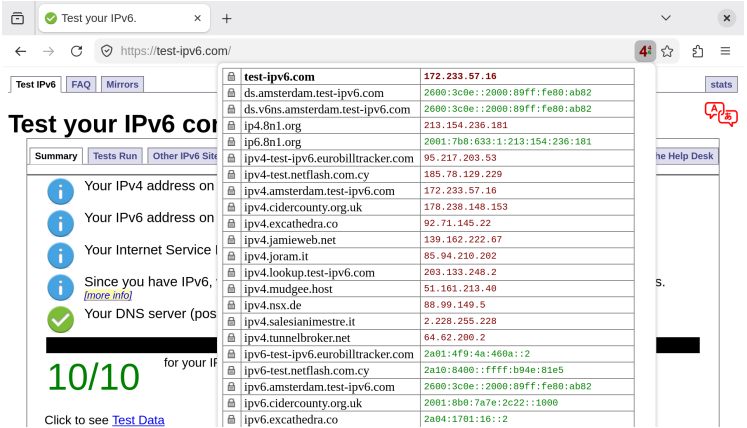
- Manche Websites sind kaputt
- RPZ: ausgenommen mit

```
example.com
  CNAME
  rpz-passthru.
```

- DNS64: Ausnahmen nicht möglich

5.1 IPvFoo

- Erkenne sofort, was IPv4-only ist
- `https://github.com/pmarks-net/ipvfoo`



5.2 OSINT Quellen

- IPv6-DNS-Server von Miyuru 😉
- Dokumentation des Services (für das Delegations-Setup)
- IP-Präfixe vom Service oder Routing-Einträge für relevante ASNs über `bgp.he.net` finden
- Tranco-Liste und Certificate-Transparency-Logs nach Service-Subdomains durchsuchen, *z.B.* `vip1.g5.cachefly.net`
- Reiter "Certs" auf `bgp.he.net`, um Domains für IP-Präfixe zu finden (auch Certificate Transparency)
- Reverse DNS und *z.B.* `dnshistory.org`, um Domains zu finden, die auf eine IP-Adresse zeigen

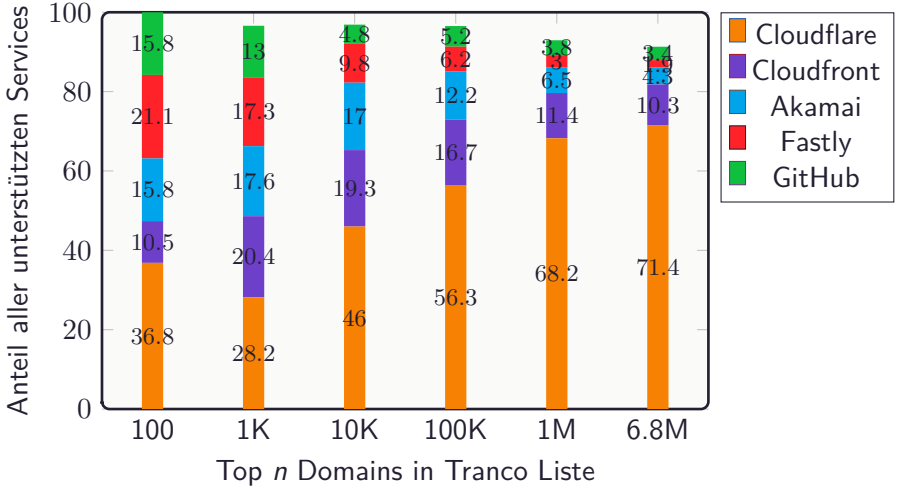
5.3 Neue Regeln erstellen

1. Bemerken, dass eine Website kein IPv6 (IPvFoo) hat
2. Prüfen, ob ein Reverse Proxy verwendet wird, der IPv6 unterstützt (`geoiplookup` oder `bgp.he.net`, um die ASN zu erhalten)
3. Prüfen, ob CNAME-Delegation verwendet wird (`dig`/`drill`)
4. Nach anderen Websites suchen, die diesen Dienst nutzen und IPv6 unterstützen (Websuche, Reverse DNS, HE-Certs-Tab für IP-Präfixe, Tranco-Liste)
5. Verbindung mit IPv6 testen: `curl https://example.com --resolve=example.com:443:[2001:db8::1] -v`
6. RPZ Regel erstellen

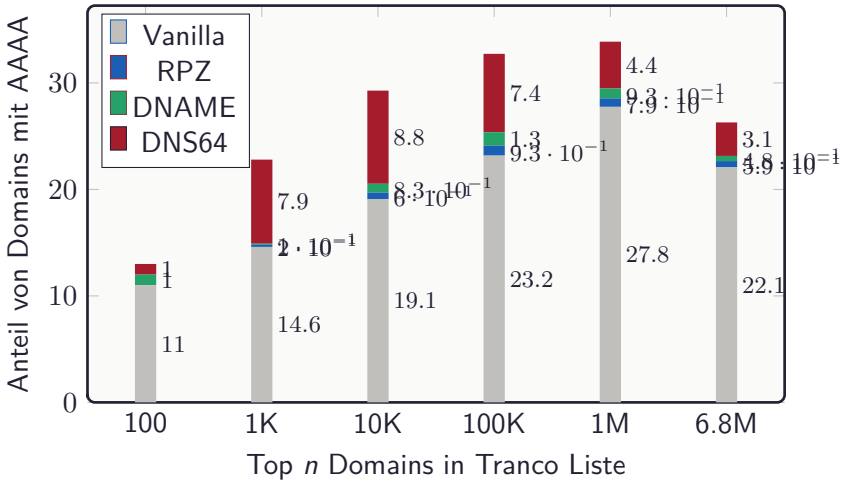
6.1 Evaluations-Methodik

- Verwendung der vollständigen Tranco-Liste mit Subdomains (ID 6Q3JX, 6,8 Mio. Domains)
- Abfragen jeder Domain mit BIND als rekursivem Resolver für die A- und AAAA-Einträge (sowie CNAMEs zur Dokumentation)
- BIND verfügt über zwei Ansichten (Views):
  - Vanilla, ohne Modifikationen
  - DeLegacy-RPZ; DNAME-Primärzonen; DNS64 für Cloudflare, Cloudfront
- CDNs werden anhand ihrer Domainnamen (wie bei der RPZ) und anhand von IP-Präfixen (wie bei DNS64) erkannt
- Statistische Auswertung der Anzahl zusätzlicher AAAA-Einträge für die Top-100, 1.000 usw. Domains und pro CDN

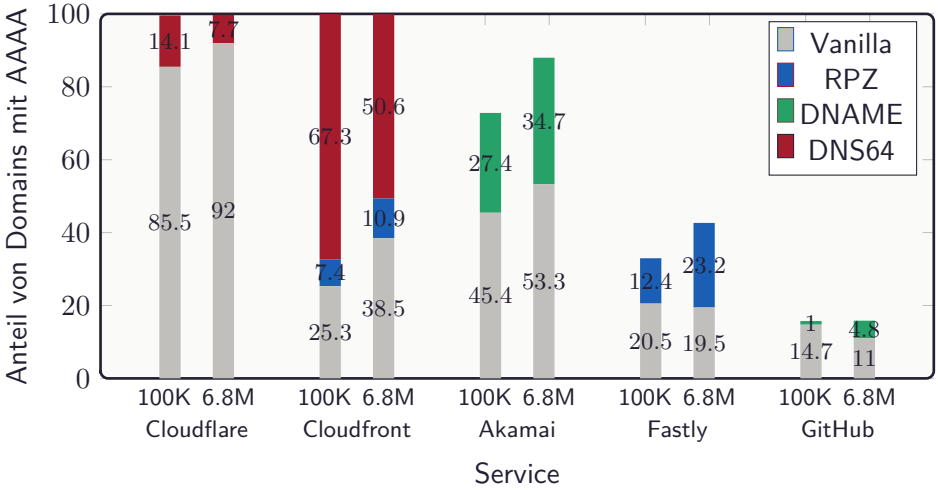
6.2 Beliebtheit von Services



6.3 Allgemeiner Einfluss



6.4 Einfluss pro Service



6.5 Das Ökosystem vorantreiben

- Subversive Methode, die funktioniert, indem man einfach den DNS-Resolver anpasst
- Clients stellen Verbindungen über IPv6 her, Server haben nur IPv6 als Client-IP
- Erstmals sind Serverbetreiber gezwungen, sich anzupassen
  - Selbst wenn sie explizit IPv6-Clients blockieren, siehe X/Twitter
- CDNs könnten diese Methode verhindern, ähnlich wie beim Domain Fronting
  - Sie haben keinen Anreiz, dies zu blockieren, bis Kunden es wollen

7.1 Probiere es aus

- Mit NixOS: Nutze die inkludierte Flake und aktiviere manche der Optionen (RPZ, DNAME, DNS64)
- Ansonsten: Setup BIND/Unbound wie dokumentiert

```
rpz:
  name: rpz.delegacy.monostack.org
  primary: primary.delegacy.monostack.org
```

7.2 BIND Setup

Listing 1: BIND RPZ Setup

```
options {
  response-policy {
    zone "rpz.delegacy.monostack.org";
  };
};
zone "rpz.delegacy.monostack.org" {
  type secondary;
  primaries {
    2a01:4f8:251:305f::1;
  };
};
```

+ DNAME setup

Listing 2: BIND DNS64 Setup

```
acl "cloudflare-v4" {
  173.245.48.0/20;
  [...]
};
acl "aws-cloudfront-v4" {
  120.52.22.96/27;
  [...]
};
options {
  dns64 2606:4700::/96 {
    mapped {cloudflare-v4;};
  };
  dns64 2600:9000:2359::/96 {
    mapped {aws-cloudfront-v4;};
  };
};
```

7.3 Ausblick

- Mehr Dienste unterstützen
- Die Abdeckung bestehender Dienste verbessern
- Dokumentation erweitern
- Automatisierte Tests zu bestehenden Regeln hinzufügen
- Unterstützung in BIND hinzufügen, um bestimmte Domains von DNS64 auszunehmen
- Einfache, installierbare Pakete für OpenWrt und Linux-Distributionen erstellen
- Eine AdBlockPlus-Liste generieren, um uBlock Origin etc. zu unterstützen

Mynacol

Tübingen, 2026-07-04

7.5 GitHub

- GitHub hat bekanntermaßen kein IPv6
- `raw.githubusercontent.com`, GitHub Pages und andere Web-Ressourcen nutzen Fastly und haben IPv6
- `github.com`, `api.github.com` usw. verwenden ihr eigenes ASN
- Sie haben `2a0a:a440::/29` zugewiesen bekommen
- `2a0a:a440::1` ist per Ping erreichbar, aber kein TCP-Port ist erreichbar
- Wahrscheinlich durch eine Firewall blockiert
- Sie nutzen mittlerweile Azure für Load Balancer
  - Das sind generische VPS, nicht Azure FrontDoor/CDN
- Bisher keine Spur, welche IPv6-Adressen diese VPS möglicherweise haben