Richtlinie zum „Software-Sprint“

sig0namectl – sig zero name control

Schlussbericht

Zuwendungsempfänger:

Burns, Jud, Bubert GbR

Adam Burns, Mathias Jud, Henry Bubert

Das diesem Bericht zugrundeliegende Vorhaben sig0namectl wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS24S08 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

sig0namectl erstellt ein dezentralisiertes System mit öffentlichen kryptografischen Schlüsseln innerhalb des Domain Name Systems (DNS), das es Nutzenden ermöglicht, selbst inträge in Echtzeit zu aktualisieren: Namen, IP-Adressen, Dienstdetails, Standortdaten und vieles mehr. Das DNS verbindet menschenlesbare Namen mit maschinenlesbaren Daten. Es wird benutzt für die Veröffentlichung, das Auffinden und den Zugang zu Informationsressourcen im Internet.

Was fehlt, ist eine Möglichkeit, mit der weniger technisch-versierte Nutzende selbst Aktualisierungen vornehmen können, die erforderlich sind, um Dienst- und Ressourcendaten online verfügbar zu machen.

Bisher bestand vorallem bei Betreibenden von Rechenzentren und Verwaltenden von Internetadressen die Möglichkeit, Dienste und Ressourcen online anzubieten sowie Daten zu verwalten. Dieses Leistungsspektrum wurde vorallem im Namen Dritter als Dienstleistung für Kunden erbracht.

Weitreichende und folgenschwere Sicherheitslecks bei Dienstanbietern, wie Domain-Hijacking und Passwortlecks brachten zunehmend ein Bewusstsein hinsichtlich der Problematik. Zunehmend suchen Organisationen und Einzelpersonen nach Möglichkeiten, die eigenen Daten zu verwalten, um Risiken der Abhängigkeit von zwischengeschalteten Dritten zu minimieren.

Zur Verbesserung der Cybersicherheit, begannen auch die europäischen Regierungen, die Verwendung von DNS-Sicherheitserweiterungen bei der Bereitstellung öffentlicher Dienste vorzuschreiben.

sig0namectl bietet Selbstbestimmung für die Internetpräsenz mit der Möglichkeit, kritische Daten, Dienste und Ressourcendetails, die für den Zugang zu Informationen über das Internet erforderlich sind, einerseits unabhängig und andererseits auch nach hohen und neuen Sicherheitsstandards selbstständig zu veröffentlichen.

In Deutschland möchte die Freifunk Community dies verwenden um dezentralisierte Dienste abzusichern. In Ländern mit DNS-Zensur und umkämpfter Informationsfreiheit, hilft dieses Projekt, die Vertrauenswürdigkeit der Dienstleistungen und Informationen zu garantieren.

Deutsche und EU-Gesetze setzen hohe Standards für Datenschutz und informationelle Selbstbestimmung. Dennoch bleiben Internetnutzer:innen, -entwickler:innen und -anbieter:innen abhängig von internationalen Konzernen, die diese Gesetze missachten, wie viele "Schrems"-Urteile zeigen.

Wir glauben nicht nur an einen sinnvollen und freien Zugang zum Internet, sondern auch an eine sinnvolle Beteiligung an der unabhängigen Bereitstellung digitaler Dienste und Ressourcen jenseits von Rechenzentren und an Alternativen zu Web-Service-Providern. Die Lösung von sig0namectl ermöglicht lokale Selbstbestimmung, digitale Selbstversorgung und bietet neue Chancen für lokale Regionen.

Vertrauen, Sicherheit, lokale Dienste und Geolocation sind zentral für unser digitales Leben. sig0namectl bietet eine Alternative zum bisherigen zentralen und technozentrierten Ansatz: eine nutzergesteuerte, dezentrale, sichere Lösung durch innovative Nutzung der Standard-Kernprotokolle des Internets.

Die wichtigsten erreichten Meilensteine sind:

- Entwicklung eines DNS-Server-Konfigurationsassistenten zur Unterstützung bei der Konfiguration von DNS-Servern für DNSSEC- und SIG(0)-Schlüssel-basierte Updates.
    - modularer und erweiterbarer Satz lokaler Workflow-Tools in Golang, die bei spezifischen, allgemeinen, benutzerorientierten Aufgaben wie der Bereitstellung von Diensten und der Veröffentlichung von Datensätzen unterstützen und die direkt und sicher über das DNS-Protokoll kommunizieren.
- Entwicklung von mindestens 4 verschiedenen Anwendungsszenarien für DNS-Updates.
    - einfache Front-End-Anwendung mit Schwerpunkt auf intuitiven (Domain Name)-Visualisierung und einfachem Workflow-Management für Echtzeit-Updates von Resource Records im dezentralen DNS-Netzwerk.
- Entwurf und Implementierung einer browserbasierten clientseitigen GUI mit lokalem Schlüsselspeicher
    - Erstellung von Tools in Ansible zur Vereinfachung der anfänglichen Konfiguration von Name-Servern, um sicherzustellen, dass DNS-Sicherheitserweiterungen aktiviert sind und um passwortlose, dynamische Aktualisierungen mit kryptografischen, auf öffentlichen Schlüsseln basierenden Anmeldeinformationen zu ermöglichen.
- Bereitstellung und Test unter mindestens 3 DNS-Zonen für die Echtzeit-Aktualisierung von
    - Öffentlichen und privaten IP-Adressdatensätze für Hosts und Websites.
    - Echtzeit-GPS-Updates mit wählbarer Genauigkeit für Ressourcenstandorte.
    - Zeiger-, Diensttyp- und spezifische Konfigurationsdaten für Service Discovery.
    - Wireguard VPN-Dienst für die Bereitstellung von öffentlichen IP-Adressen.

Unsere benutzerzentrierten Tools für die Verwaltung von DNS Einträgen, die Aktualisierungen mit Hilfe von benutzergenerierten, kryptografischen Schlüsselpaaren direkt zu authentifizieren, bieten eine Sicherheit sowie eine benutzerfreundliche Initialisierung von IT-Ressourcen, die bisher so nicht möglich war.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Das Projekt kann den Themen „IT-Sicherheit“ und „Civic Tech“ zugeordnet werden, da es Tools entwickelt hat, die eine sichere und unmittelbare Veröffentlichung von Dienstleistungs- und Ressourcendetails im Internet für Nutzende ermöglicht.

Ausführliche Darstellung der Ergebnisse

Das Projekt sig0namectl wirkt auf mehreren Ebenen um Entwicklungsresultate zu erzeugen:

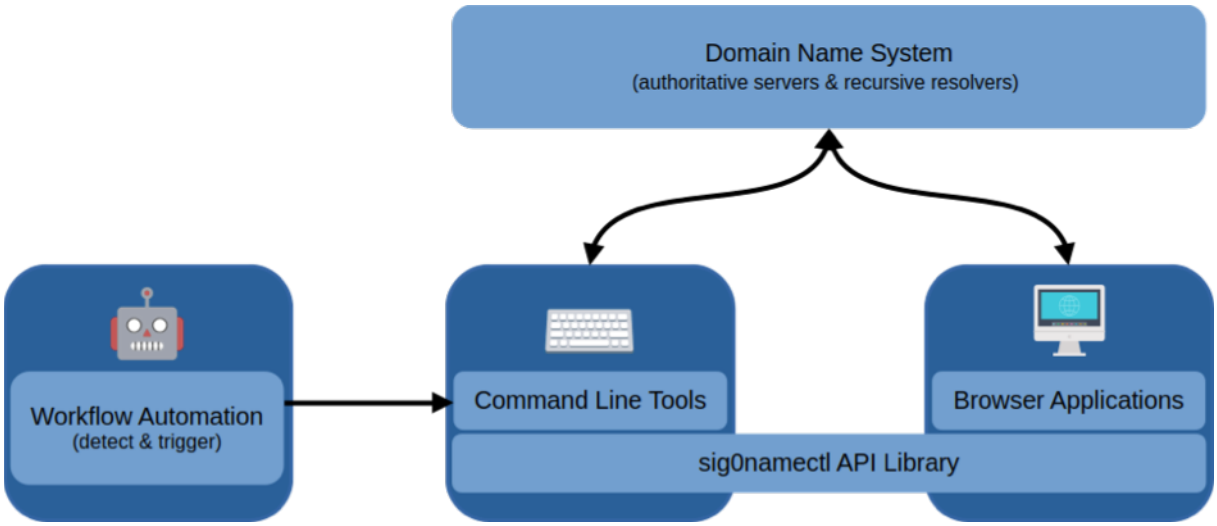


Figure 1: Übersicht der Ergebnisse

Domain Name System

Testkonfigurations- und Bereitstellungswerkzeuge wurden mit Ansible und Terraform entwickelt, um die Erstkonfiguration und Bereitstellung von Domain Name Servern zu vereinfachen. Dies unterstützt DNS-Administratoren bei der Aktivierung von DNS-Sicherheitserweiterungen (DNSSEC) und der Konfiguration von Anmeldeinformationen auf Basis asymmetrischer Kryptografie. Diese können dann ebenfalls für dynamische Updates der Daten benutzt werden, die zum erreichen der betriebenen Dienste notwendig sind.

Die Dokumentation für das Konfigurationsprogramm finden Sie unter https://github.com/NetworkCommons/sig0namectl/tree/master/bind9

Workflow Automation

Die Workflow-Automatisierung bildet mehr als 4 verschiedene Anwendungsszenarien ab, insbesondere:

- benannte öffentliche Domainname Schlüsselregistrierung
- Registrierung, die Namen mit kryptografischen Schlüsseln verknüpft

- dynamische Aktualisierung von IP-Adressen
- dynamische Konfiguration von Domain-Texten
- dynamischer Geolokalisierungsdienst
- Unicast-DNS-SD-Registrierung
- Delegation von Domainschlüsseln für die Zusammenarbeit bei Updates

Eine detaillierte Dokumentation zur Verwendung der einzelnen Workflow-Automatisierungstools finden Sie auf der Projektwebsite unter: https://sig0namectl.networkcommons.org/tutorials/workflow-automation-scripts/

Command Line Tools

Das Kommandozeilenprogramm *sig0namectl* bietet fortgeschrittenen Benutzenden komplexere Optionen als die browserbasierten Benutzerschnittstellen.

Die Verwendung des Kommandozeilen-Tools ist unter https://sig0namectl.networkcommons.org/tutorials/command-line-tool/ dokumentiert.

sig0namectl API Library

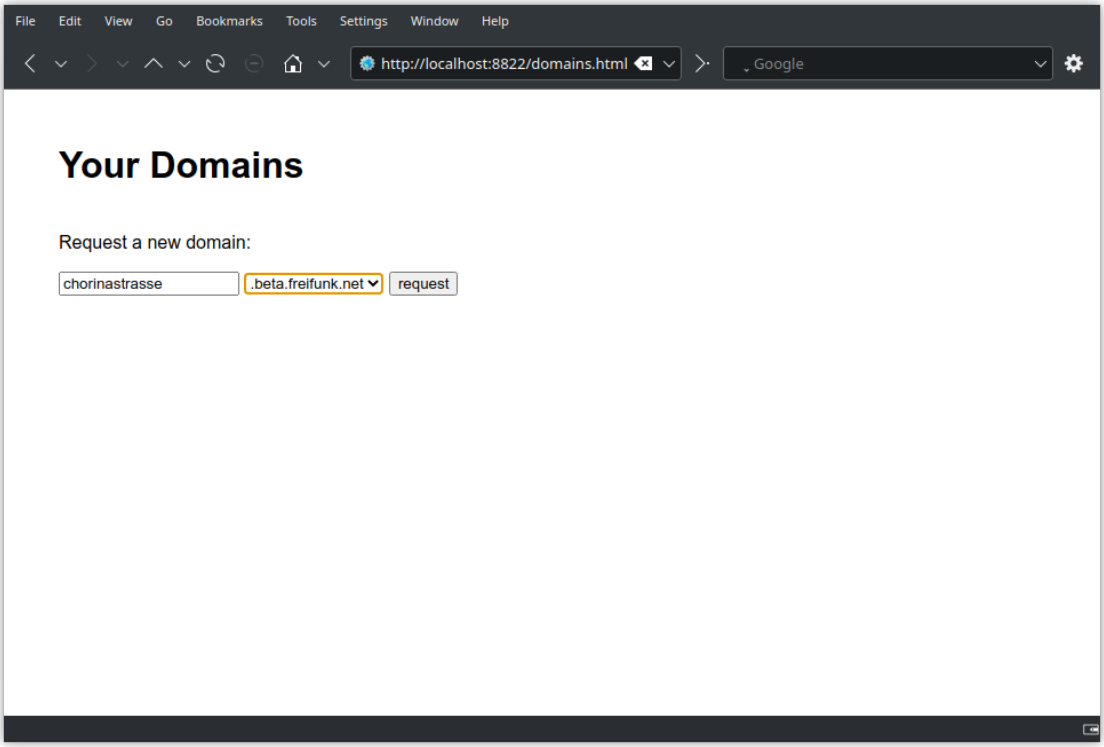
Die sig0namectl-API bildet Funktionen ab, die mit Sicherheit und Zuverlässigkeit die Ausführung der Kommandozeilentools ermöglicht sowie eine API, die auf browserbasierte Benutzerschnittstellen zugeschnitten ist. Beides befindet sich in der selben Codebasis und ist in der Programmiersprache Golang verfasst.

Beispiele für die Verwendung der API zur Entwicklung anderer browserbasierter Schnittstellen finden Sie auf der Projektwebsite unter https://sig0namectl.networkcommons.org/tutorials/golang-wasm-sdk/

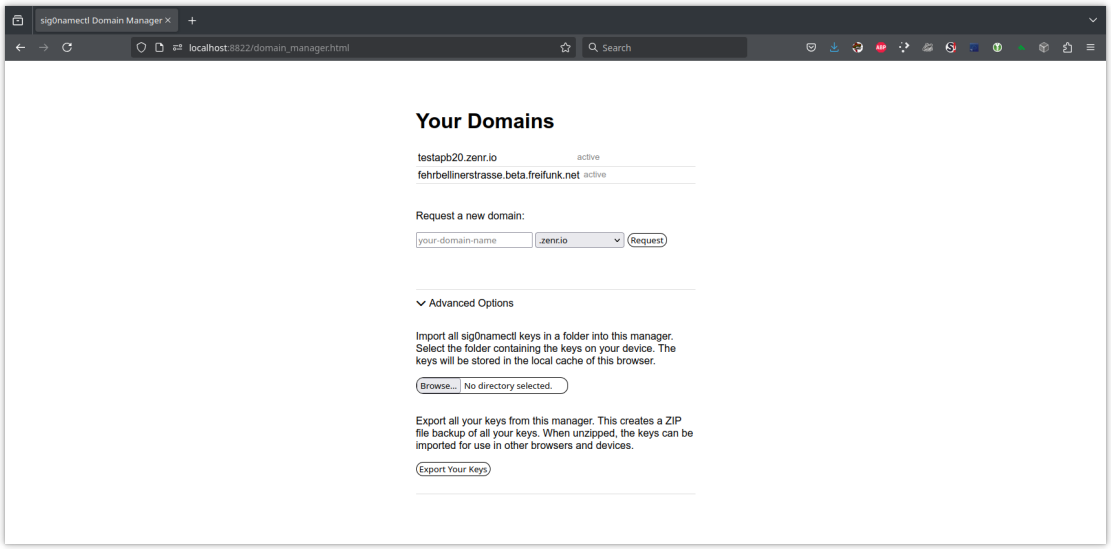
Browser Applications

Es wurden drei Prototypen für browserbasierte Benutzeroberflächen entwickelt, und zwar:

(1) **Domain Manager**

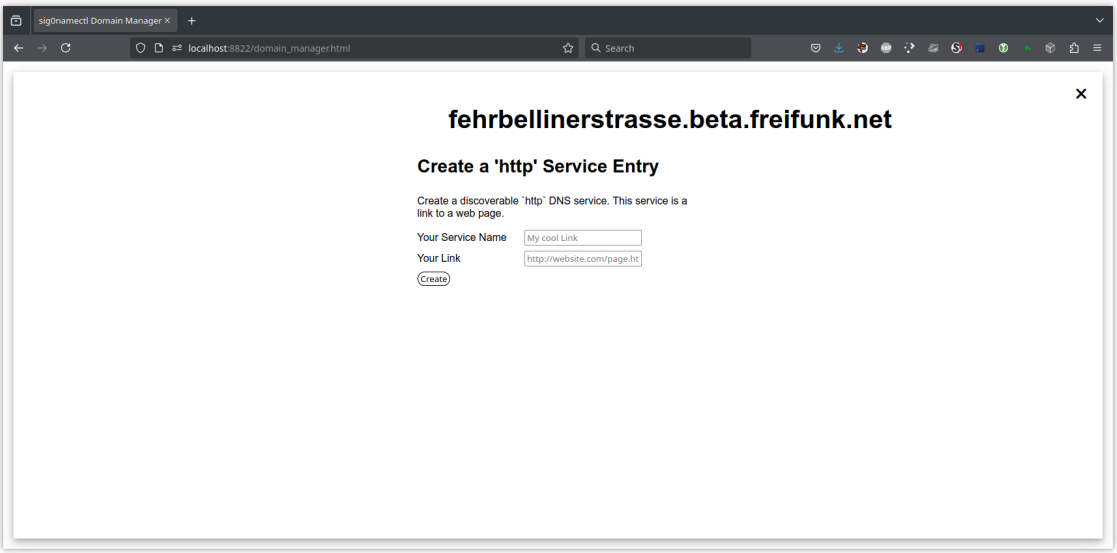


Eine Browser-App zum Beantragen und Verwalten Ihrer Domainnamen. Die Standardrichtlinie „Wer zuerst kommt, mahlt zuerst“ für sig0namectl bedeutet, dass die Anfrage genehmigt wird, auch wenn der vollständige Domainname noch nicht existiert.



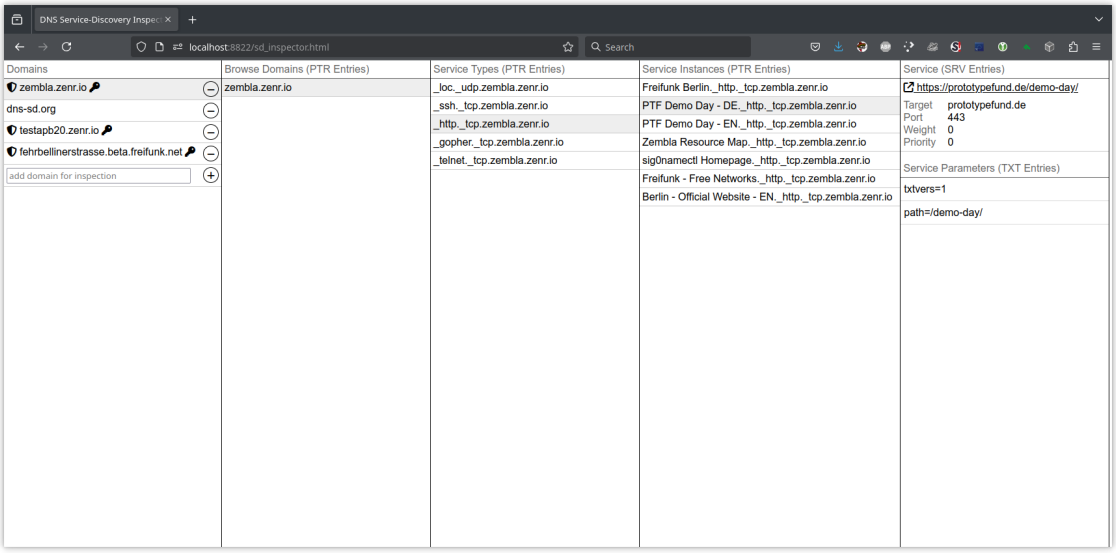
Die Domain-Manager-App ermöglicht es dem Benutzer, die SIG(0)-Kryptografieschlüssel für alle seine Domains zu sichern und wiederherzustellen. SIG(0) ist ein kryptografisches Schlüsselformat, das im Internetstandard RFC2535[1] definiert ist.

[1] https://www.rfc-editor.org/rfc/rfc2535



Die Domain-Manager-App ermöglicht es dem Benutzer, Webservices und Ressourcen, die mit „Wide Area DNS Service Discovery“ veröffentlicht werden sollen, einfach zu definieren.

(2) **Service Discovery Inspector**



Eine Browser-App, die Dienste und Ressourcen anzeigt, die im Rahmen von „Wide Area DNS Service Discovery“ veröffentlicht wurden (kompatibel mit dem Internetstandard RFC 6763[2]).

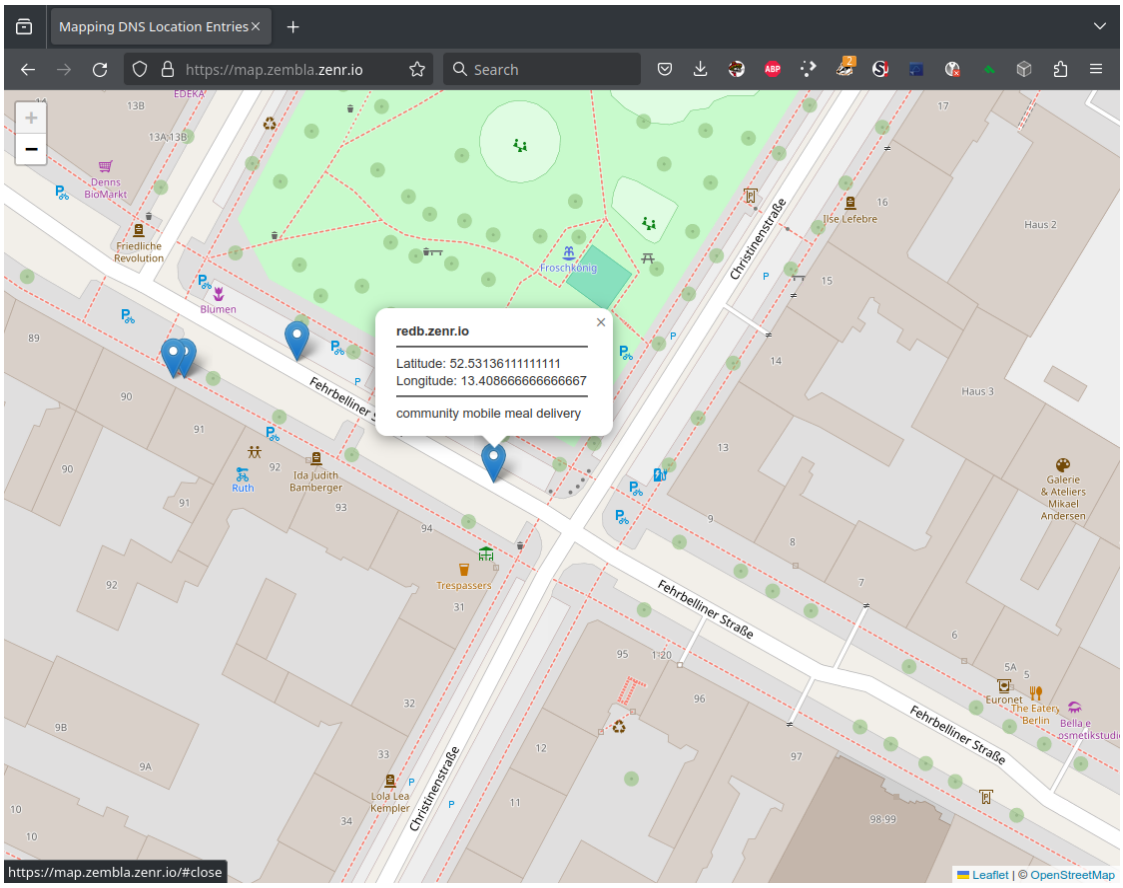
„Wide Area DNS Service Discovery“ verwendet dieselbe Technologie, die Computer verwenden, um z.B. Drucker in einem lokalen Netzwerk zu finden, funktioniert jedoch ebenfalls über das Internet. Für jede beanspruchte Domain ermöglicht der Inspektor die Erstellung und Bearbeitung neuer Dienste, die mit anderen geteilt werden können.

Das Schildsymbol in der ersten Spalte zeigt an, dass DNS-Sicherheitserweiterungen für die Domain aktiviert sind.

2 https://www.rfc-editor.org/rfc/rfc6763

Das Schlüsselsymbol zeigt an, dass die Browser-App Zugriff auf den Sicherheitsschlüssel hat, um Datensätze unter der Domain zu aktualisieren.

(3) **Dynamic Local Resource Map**



Eine Browser-App, die eine Karte zeigt, auf der dynamisch Ressourcen und ihrer Standorte angezeigt werden, die durch sig0namectl eingespeist wurden. Dieses Beispielbild zeigt eine kuratierte Reihe von Standorten, die *zembla.zenr.io* gehören, wobei jede Positionsmarkierungsressource unter der Kontrolle anderer Eigentümer in ihrem eigenen unabhängigen DNS-Namensraum steht.

In diesem Beispiel ist *redb.zenr.io* ein Android-Mobiltelefon, mit seinem eignen kryptographischen Schlüsselpaar. Es aktualisiert regelmäßig seine eigene Position im DNS mit Informationen von seiner GPS-Hardware, indem es das bereitgestellte Workflow-Automatisierungsskript *start_loc_loop.sh* ausführt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Weltweit existieren Hunderte bis Tausende von lokalen, gemeinschaftlich betriebenen digitalen Netzen. Initiativen wie Freifunk und GuifiNet haben Zehntausende von Teilnehmenden und beträchtliche Online-Datenressourcen, die derzeit weder innerhalb ihre eigene noch im Internet leicht zu finden sind. Die Ansprache und Betreuung der Zielgruppe ist durch die Teilnahme an lokalen Treffen, Online-Foren und übergreifenden Mailinglisten erreicht worden. Einbezogen in den Wissensaustausch wurden auch regionale und internationale Foren wie das UN Internet Governance

Forum und die Internet Society sowie digitale Initiativen für Autonomie, Selbstbestimmung und digitale Privatsphäre wie "My Data Global" und "Open Rights Group“.

Für die Entwicklung und Erprobung von sig0namectl hat die deutsche Freifunk-Community[3] Ressourcen bereitgestellt. Freifunk ist eine bundesweite deutsche Initiative, die freien Netzzugang und offene Infrastruktur mit dem Ziel der Demokratisierung der Kommunikation bereitstellt.

Das Funknetz von Freifunk ist groß, allein in Berlin gibt es über 500 Computerknoten und WLAN-Hotspots[4], die über die ganze Stadt verteilt sind und die ehrenamtlich unterhalten werden.

sig0namectl ermöglicht es den Nutzenden der Freifunk-Community noch unabhängiger und selbstbestimmter zusammenzuarbeiten, lokale Webressourcen jederzeit zu veröffentlichen und damit öffentlich zugänglicher und erreichbarer zu machen:

- auf ihren eigenen Computern
- in ihren eigenen Räumlichkeiten
- über ihre eigenen lokalen Netzwerke.

Freifunk steht beispielhaft für eine der digitalen, regionalen Initiativen auf der ganzen Welt, die sich der Arbeitsergebnisse bedienen können.

Die Arbeitsergebnisse von sig0namectl stehen als Open-Source-Projekt jedem/r und jeder Gruppierung frei zur Verfügung. Die entwickelten Tools können einfach genutzt werden oder an sie können an die Bedarfe der eigenen lokalen Gemeinschaft angepasst und weiterentwickelt werden. Ein regelmäßiger Austausch von Erfahrungen ist im Rahmen bestehender digitaler Netzwerke (siehe Alumni Prototype Funds, Netzpolitik etc.) angedacht.

Weitere Browseranwendungen und Funktionserweiterungen sind mit Beteiligung der Freifunk-Kartenentwickelnden und WLAN-Router-Image-Betreuenden geplant.

Eine ähnliche Co-produktion und Wissenstransfer mit anderen Community-Netzwerken in der EU wie z.B.GuiFi Net in Spanien, Athens Wireless, und anderen ist vorgesehen.

Ziel ist es, Menschen in regionaler wie dezentraler Umgebung einen selbstbestimmten und niedrigschwelligen Zugang zu Information und Ressourcen zu ermöglichen und die Selbstverwaltung ihrer Daten zu vereinfachen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die Entwicklung eines generischen Browser-Frontends zur Aktualisierung aller möglichen Arten von DNS-Ressourceneinträgen war innerhalb der 6-monatigen Projektlaufzeit nicht trivial zu bewerkstelligen.

Das DNS-System ist praktisch „zustandslos“, da Aktualisierungen durch andere Personen oder Maschinen zu jedem (zukünftigen) Zeitpunkt erfolgen können.

Wir haben nach Methoden gesucht, um diese *Statelessness* (Zustandslosigkeit / fluider Zustand) in den Browser-Apps zu ermöglichen, insbesondere nach Technologien für das „DNS-Crawling“ unter

[3] https://freifunk.net/
[4] https://hopglass.berlin.freifunk.net/

Verwendung von NSEC-Ressourceneinträgen. Die für die Umsetzung erforderlichen Forschungs- und Entwicklungszeitressourcen wurden jedoch als zu hoch eingeschätzt, um sie neben den anderen Zielen (wie cross-compiled Golang zu WebAssembly im browser) innerhalb von 6 Monaten zu erreichen.

Stattdessen haben wir uns in Absprache mit Freifunk Berlin auf Aktualisierungsfunktionen im Zusammenhang mit Geolokalisierung und Service Discovery konzentriert.Diese Anwendungsfälle waren für unsere primäre Zielgruppe am relevantesten.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Webseite: https://sig0namectl.networkcommons.org
GitHub: https://github.com/NetworkCommons/sig0namectl

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Zusammenarbeit mit den Freifunk-DNS-Administratoren gestaltete sich kooperativ und unterstützend. Aufgrund der ehrenamtlichen Struktur von Freifunk dauerte die Einrichtung der DNSSEC- kompatiblen DNS-Zonen und des damit notwendig verknüpften sig0namectl-Testbetriebs jedoch weitaus länger als ursprünglich in der Planung vorgesehen.

Allerdings wurden dank guter Zusammenarbeit im Großen und Ganzen alle geplanten Aufgaben wie geplant umgesetzt.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Im Verlauf unserer Projektentwicklung haben wir Fehler in den verbundenen Open-Source-Projekten entdeckt, die in sig0namectl zur Anwendung kommen. Es handelt sich um:

- Im Golang-DNS-Modul haben wir einen Softwarefehler für kryptografische SIG(0)-Signaturen gemeldet, die in DNS-Updates verwendet werden. Weitere Informationen und Details finden Sie unter https://github.com/miekg/dns/issues/1339 - Dieser Fehler hätte unser Projekt in ernsthafte Probleme bringen können. Nachdem wir diesen gemeldet hatten, wurde er umgehend behoben und die Funktionalität stand für unser Projekt zur Verfügung.
- Im Golang-Modul „doh-client“, das die Funktionalität eines DNS-over-HTTPS-Clients (DoH) bereitstellt, meldeten wir eine Einschränkung einer fest codierten URL-Zeichenfolge, die die Verwendung des Moduls mit dem Internetstandard RFC 9460[5] einschränkte. Weitere Informationen finden Sie unter https://github.com/shynome/doh-client/issues/1
- Der Upstream-Autor bedankte sich bei uns und behebt den Fehler in weniger als 24 Stunden.

Wir sind stolz darauf, dass wir mit unserem Feedback und unseren Verbesserungen einen Beitrag zu Open-Source-Gemeinschaften im Rahmen dieses Projekts leisten konnten.

[5] https://www.rfc-editor.org/rfc/rfc9460.pdf