Richtlinie zum „Software-Sprint“

Shig – verteiltes, dezentrales live Streaming

Schlussbericht

Zuwendungsempfänger:

Enrico Schwendig

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS23S44 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Video- und Streamingplattformen spielen heutzutage eine bedeutende Rolle in der Gesellschaft und haben daher eine enorm wichtige Bedeutung. Die Technologie des Live-Streamings hat sich kontinuierlich weiterentwickelt und bietet bereits komplexe Interaktionsmöglichkeiten. Mit Shig möchte ich diese Möglichkeiten auch im Fediverse verfügbar machen.

Mit Hilfe von Shig können Leute live und interaktiv über das Fediverse streamen. Zuschauer*innen und Streamer*innen können direkt miteinander interagieren. Shig ermöglicht Live-Interaktionen über verschiedene Anwendungen hinweg, ohne dass zusätzliche Apps erforderlich sind – ein einfacher Webbrowser genügt

Die Multi-Gast-Streaming-Technologie, die im Shig-Projekt verwendet wird, ist neuartig und soll im Open-Source-Bereich zur Verfügung gestellt werden. Durch die Nutzung von Multi-Gast-Streaming sind weitaus mehr Live-Interaktionen zwischen Streamer*innen und Zuschauer*innen möglich als derzeit von Fediverse-Plattformen angeboten wird.

Darüber hinaus verfolgt das Projekt einen dezentralen Ansatz und ist darauf ausgelegt, als Plugin in Plattformen mit dezentralen Architekturen wie PeerTube und Mastodon integriert zu werden.

Da die Technologie noch relativ neu ist, ist jetzt der richtige Zeitpunkt für Open-Source-Ansätze in diesem Bereich. Benutzer*innen sind heutzutage hohe Qualitätsstandards gewohnt, und Open Source muss sich daran messen lassen. Es erfordert Zeit und Entwicklerressourcen, um ausgereifte Systeme zu entwickeln. Obwohl es viele Open-Source-Streaming-Lösungen gibt, gibt es kaum welche, die global skalieren können, da der Betriebsaufwand hoch ist. Dezentrale Open-Source-

Lösungsansätze, die jetzt entwickelt werden, hätten noch ausreichend Zeit, um ein hohes Qualitätsniveau zu erreichen, bevor größere Unternehmen ihre Vorteile erkennen.

Zurzeit haben zentrale Ansätze den Vorteil, dass sie leichter zu hosten und vergleichsweise kostengünstiger sind. Dennoch sind sie erheblich teurer als dezentrale Ansätze. In Zeiten, in denen Ressourcen knapp werden, steigen auch die Hostingkosten an. Insbesondere Videoanwendungen erfordern aufgrund ihrer hohen Datenübertragungsraten und der intensiven CPU-Nutzung einen erheblichen Energie- und Kostenaufwand für die Serverinfrastruktur. Dezentrale Ansätze verteilen den Hosting-Aufwand auf viele Schultern.

Shig wurde konzipiert, um bestehende föderative Streaming-Anwendungen zu erweitern. Es besteht aus der Shig-Instanz, einer Backend-Applikation in Go, und einem Frontend, das als Erweiterung für PeerTube fungiert. Die Shig-Instanz ist das Herzstück der Anwendung und vereint sowohl die ActivityPub-Anbindung als auch die Technologie für interaktives Live-Streaming.

Im ersten Meilenstein, **Einbettung einer Shig-Instanz über das PeerTube-Plugin in eine PeerTube-Plattform** wurde das PeerTube-Plugin entwickelt. Um die Erweiterbarkeit für andere Fediverse-Applikationen zu gewährleisten, wurde zunächst ein Javascript-SDK entwickelt, auf dessen Grundlage das Plugin aufgebaut wurde.

Um sicherzustellen, dass die Backend-Anwendung das ActivityPub-Protokoll in allen Bereichen verwendet, wurde als zweiter Meilenstein: **Anbindung der Shig-Instanz in die Federation API (ActivityPub)** umgesetzt. Dies ermöglicht nicht nur die Datenübertragung von PeerTube, sondern bereitet auch die Skalierung der Shig-Instanzen vor.

Im nächsten Schritt wurde der Meilenstein: **Mehrerer verteilter Shig-Instanzen** erreicht. Dadurch können Nutzer*innen von PeerTube nun an einem gemeinsamen Live-Stream teilnehmen, der von verschiedenen PeerTube-Instanzen unterstützt wird. Dies wird ermöglicht, indem Shig-Instanzen dynamisch Verbindungen aufnehmen können, um die Streams zwischen sich zu verteilen und aufzulösen.

Im abschließenden Meilenstein: **Ausfallsicherheit und dynamische Skalierung von verteilten, dezentralen Shig Services**, stand die Stabilität und Optimierung im Fokus. Ein wesentlicher Entwicklungsschritt in diesem Zusammenhang war die Implementierung eines Monitoringsystems, mit dem der Zustand der Live-Streams und der Anwendung überwacht werden kann.

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

Der Quellcode von Shig steht unter der MIT-Lizenz und ist offen für Weiterentwicklungen oder kann als Vorlage und Blaupause für andere Projekte dienen.

Shig erweitert das ActivityPub-Protokoll, um die Verteilung von Audio- und Video-Live-Streams zu ermöglichen. Durch diesen referenziellen Ansatz trägt Shig aktiv zur Stärkung des offenen Protokolls bei.

Interaktive Video- und Audio-Live-Streams bilden das Fundament für eine Reihe weiterer Features, die im Fediverse noch fehlen. Hierzu zählen insbesondere interaktive, verteilte Live-Text-Chats und verbesserte Skalierungsmethoden für Live-Streams. Entwickler können auf den technischen Konzepten von Shig aufbauen, diese anwenden, erweitern, verbessern oder auch neu implementieren.

Das Fediverse ist ein autonomes, demokratisches Applikationsnetzwerk, das frei von Suchalgorithmen oder User-Tracking ist. Es besteht aus unabhängigen Akteuren und Content-Erzeugern und ermöglicht es Nutzer*innen aktiv an der Erstellung von Inhalten teilzuhaben. Dadurch wird der demokratische Charakter des Netzwerks gestärkt. Shig erweitert das Fediverse durch innovative Konzepte und Funktionen und trägt aktiv dazu bei, seine Möglichkeiten zu verbessern.

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Im Rahmen der Förderung ist eine Applikation entstanden welche Interaktives dezentrales Live-Streaming im Fediverse ermöglicht. Die Ergebnisse der Förderung können in zwei Aspekte unterteilt werden. Einerseits umfassen sie die Fortschritte bei der Entwicklung der Software über die definierten Meilensteine hinweg, und andererseits die erzielten Erfolge bei der Erweiterung und Anwendung bestehender offener Protokolle wie ActivityPub.

Wie bereits eingangs erwähnt, besteht Shig aus einer Backend-Anwendung in Go, die Shig-Instanz und einem PeerTube-Plugin als Frontend. Zur Verarbeitung von Video- und Audiostreams nutzt die Backend-Anwendung die Pion-Bibliothek. Die Shig-Instanz implementiert ein Lobby-Konzept, um Teilnehmer*innen in einem Stream zusammenzuführen. Ein REST-Interface fungiert als Zugangs- und Autorisierungsschnittstelle für das Frontend. Eine Shig-Instanz kann entweder als native Go-Anwendung oder als Service-Container bereitgestellt werden.

Die Backend-Kommunikation erfolgt über das erweiterte Activity-Protokoll. Mithilfe von ActivityPub kann eine Shig-Instanz mit anderen föderativen Anwendungen kommunizieren und Daten austauschen. Verschiedene Shig-Instanzen können sich über ActivityPub finden und gemeinsam einen Live-Stream bereitstellen. Durch diese Verbindung der serverseitigen Shig-Instanzen ist es möglich, aus unterschiedlichen PeerTube-Instanzen heraus einem Live-Stream beizutreten. Auf diese Weise realisiert Shig dezentrales Live-Streaming über die Grenzen von Fediverse-Instanzen hinweg.

Die Backend-Anwendung bietet eine Prometheus-Metrik-Schnittstelle und erzeugt Telemetriedaten, die in Grafana visualisiert werden können. Die entsprechenden Monitoring-Dashboards sind dem Projekt beigefügt.

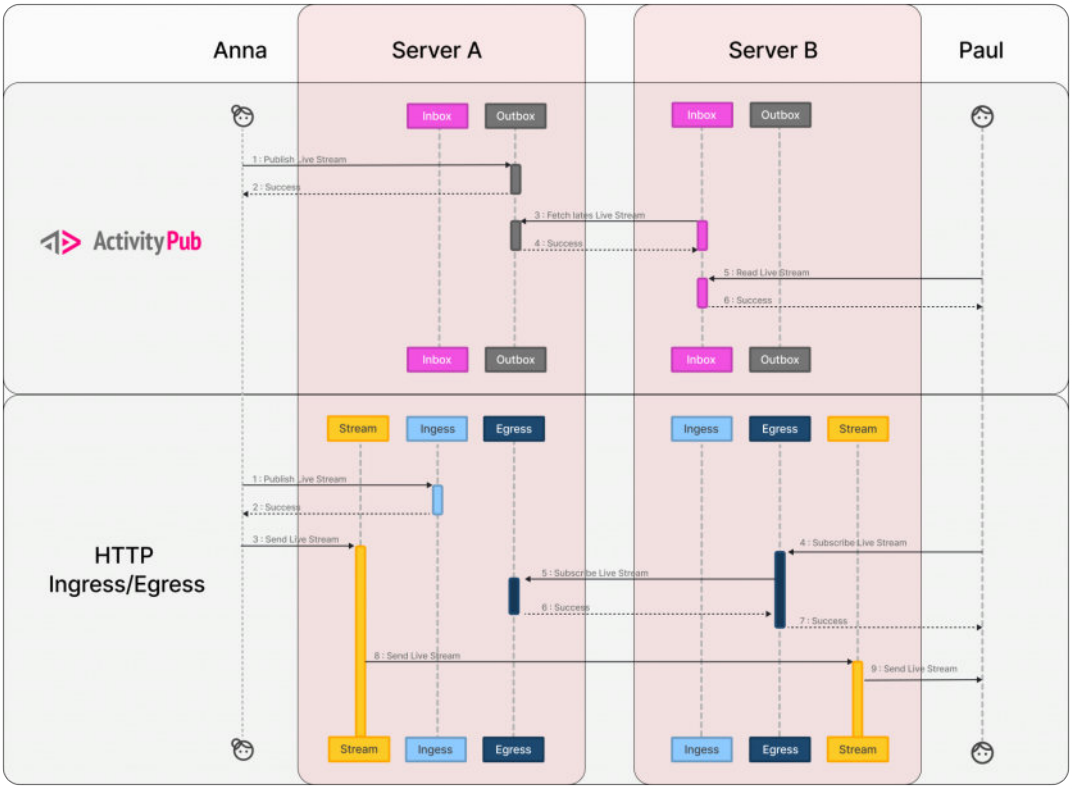
Im Rahmen der Förderung lag der Fokus des Projekts auf PeerTube als verteilte, quelloffene Plattform. Ein spezielles Plugin wurde entwickelt, das es registrierten Nutzer*innen ermöglicht, zu Streams eingeladen zu werden. Diese Erweiterung ermöglicht eine nahtlose Integration zwischen PeerTube und den Service-Instanzen von Shig, wodurch Benutzer*innen direkt von der PeerTube-Oberfläche aus auf interaktive Live-Streams zugreifen können, die von den Shig-Instanzen bereitgestellt werden.

Im Fediverse ist das Kommunikationsprotokoll ActivityPub weit verbreitet. Im Rahmen des Projekts wurde ein Weg gefunden, Live-Stream-Ressourcen über ActivityPub auszutauschen, was eine Erweiterung des Protokolls erforderlich machte. Diese Erweiterung wird kurz erläutert.

Ähnlich wie Posts und Kommentare zwischen Mastodon-Instanzen über ActivityPub verteilt werden, ermöglicht Shig durch die Nutzung von ActivityPub das Verteilen von Live-Streams zwischen

PeerTube-Instanzen. Dies ist entscheidend, um interaktives Live-Streaming im Fediverse überhaupt erst zu ermöglichen.

Auf Basis der über ActivityPub ausgetauschten Daten werden zusätzliche HTTP-Anfragen verwendet, um die erforderlichen Meta-Informationen für den Austausch von Streaming-Ressourcen bereitzustellen. Das folgende in Abbildung zeigt ein Sequenzdiagramm zur Veranschaulichung der zugrunde liegenden Idee.



Die Erweiterung des ActivityPub-Protokolls besteht darin, dass die Protokolle WHIP und WHEP in die Sequenz zur Erstellung oder Anforderung eines Live-Streams integriert wurden. Vereinfacht ausgedrückt ermöglicht es Shig durch eine einfache HTTP-Anfrage, einen Video-Stream zu erstellen oder anzufordern. Das entspricht dem Vorgang des Videoaufrufes im Webbrowser durch das Öffnen einer Webseite oder dem Hochladen einer Videodatei auf einen Server.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Das entwickelte Plugin erweitert PeerTube um Multi-Guest-Streaming und eröffnet somit neue Möglichkeiten. Durch eine integrierte Streaming-Lobby ermöglicht es Streamer*innen, ohne die Verwendung von OBS oder anderer Software direkt zu streamen. Diese Funktion macht Shig besonders ansprechend für Streamer*innen, die komplexe Setups vermeiden möchten.

Anhand des Multi-Guest-Streamings führt Shig Konzepte und Methoden für dezentrales und interaktives Live-Streaming in das Fediverse ein. Diese Form des dezentralen und interaktiven Live-Streamings bildet die Grundlage für weitere Funktionen wie instanzübergreifende Skalierung von Live-Streams oder instanzübergreifende Live-Stream-Text-Chats. Shig fungiert als Referenzimplementierung und demonstriert die Möglichkeit, interaktive Live-Streams über ActivityPub im Fediverse zu verbreiten. Es bietet Lösungen für grundlegende Herausforderungen, wie

das dezentrale Signaling über verteilte Anwendungen im Fediverse hinaus. Shig beantwortet die Frage, wie Meta-Zustände wie das Stummschalten von Audiokanälen für dezentrale Nutzer*innen mittels ActivityPub erreicht werden können. Diese Erkenntnisse dienen als Grundlage, auf der andere Entwickler aufbauen und das System weiterentwickeln können.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die ursprünglichen Ziele wurden gemäß dem ursprünglichen Arbeitsplan erreicht.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Webseite: https://shig.de

Dokumentation (technisch):https://github.com/shigde/sfu/tree/main/docs

Github: https://github.com/shigde

Mastodon: https://mastodon.social/@shig

PrototypeFound: https://prototypefund.de/dezentral-streamen-mit-shig/

Ich habe die grundlegenden Konzepte von Shig am 3. Februar der Community auf der FOSDEM24 in Brüssel vorgestellt. (Link: https://fosdem.org/2024/schedule/event/fosdem-2024-1907-shig-distribute-and-clone-live-streams-among-fediverse-instances/)

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Innerhalb der vorgegebenen Projektlaufzeit konnte ich erfolgreich meine geplanten Meilensteine erreichen, und zwar termingerecht und ohne Abweichungen von der Kosten- und Arbeitsplanung. Auch nach Abschluss des Projekts behalte ich mir vor, weiterhin an Optimierungen und Erweiterungen zu arbeiten, die sich als wünschenswert für Nutzer*innen herausstellen. Ein Beispiel dafür ist die Entwicklung eines Verfahrens zur Bandbreitenoptimierung für die Verteilung von Video- und Audio-Streams.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während der Projektphase wurde die WHEP-Schnittstelle in OBS integriert und als Entwurf bereitgestellt. (Link: https://github.com/obsproject/obs-studio/pull/10353) Diese Entwicklung verfolgt ähnliche Grundideen wie Shig. Solche Fortschritte an anderer Stelle unterstreichen die Prinzipien, die hinter Shig stehen.