

Per Anhalter durch das Fediverse

~~eine Trilogie~~ Drama in vier Teilen

Ersetzen der zentralen Dienste von Twitter, Facebook, Instagram, Youtube, Itunes Podcast und Soundcloud durch föderierte, von der Community oder selbst gehostete Dienste

Alle Links sind am Schluss gebündelt in den Linktipps.

Vorstellung



<https://panoptikum.io/informatom>

Die 4 Teile

1. Vom Urknall bis 2018 - Entwicklung des Fediverse
2. Protokolle, Dienste und Privacy
3. Hands-on: Installation einer Pleroma Instanz
4. Call-to-action: ein Service konzipieren und implementieren

1. Entwicklung des Fediverse

2008

- identi.ca Evan Prodromou
 - Control Yourself -> Laconica -2009> Status.Net (PHP) -2013> pump.io (Node.js) & Gnu.Social -,
 - Indenti.ca Service wechselte Software mehrmals
 - Wikitravel (2003), Fuzzy.ai
 - Stefan: Linux Outlaws
 - alles Microblogging a.k.a. Twitter

2016

- Mastodon
 - Microblogging
 - Ruby on Rails
 - nginx, React, Redux, Redis, node, yarn, imagemagick, ffmpeg, libprotobuf-dev, postgresql, let's encrypt, elasticsearch
 - oftmals auf Docker oder auf Ubuntu Server installiert
 - Stefan: Biddu ein Zombie? Twitter geht eh...

2017

- Pleroma
 - Elixir & Phoenix
 - weniger externe Abhängigkeiten: nginx, postgres, let's encrypt
 - Ressourcenschonend, läuft auf Raspberry Pi 3
 - auch Microblogging

2018

- Stefan: gepackt :-)) und begonnen nachzudenken

Nodes

Total users

Active 6 months

Active last month

Local posts

Local comments

Users per node

Active users ratio








User total vs monthly



2. Dienste, Protokolle und Privacy

- Was ist ein Dienst, was eine Instanz?
- Was ist in diesem Kontext ein Protokoll?
- Hilft mir das Fediverse, meine Privatsphäre zu wahren?

Projects

Project	Nodes	Users	Website	Code
 Mastodon	1356	1529558	joinmastodon.org	AGPLv3
 diaspora*	194	650710	diasporafoundation.org	AGPLv3
 Pleroma	118	2121	pleroma.social	AGPLv3
 Friendica	98	13451	friendi.ca	AGPLv3
 Hubzilla	89	3214	hubzilla.org	MIT
 PeerTube	69	2750	joinpeertube.org	AGPLv3
 PixelFed	18	3518	pixelfed.org	AGPLv3

Dienste

- ownCloud (2011) mit Fork Nextcloud (2016)
Filesharing, Kalender, Kontakte, viele Plugins
- Hubzilla: Blogging
- PeerTube (2015): Video Hosting
- PixelFed: Image Hosting
- Funkwhale: audio hosting and sharing (Fediverse Podcast, Episode 1)
- aktuell 29 Plattformen auf Wikipediaseite genannt

Protokolle

- unterschiedliche Protokolle > 10
d.h. Services können großteils nicht miteinander sprechen
- Webfinger
- ActivityPub, Activity-Streams und Json-LD (und oStatus)
- WebDAV
- diaspora
- XMPP

Protocols

Protocol	Nodes	Users
<u>ostatus</u>	2115	1547118
<u>activitypub</u>	1974	1539059
<u>diaspora</u>	477	667197
<u>dfn</u>	119	13451
<u>zot</u>	120	3174

Und die Privacy?

- keine Ende-zu-Ende Verschlüsselung der Nachrichten
 - Transport Verschlüsselung (https) de facto Standard (nicht erzwungen)
- Moderatoren der Instanzen können alles lesen, auch DMs
- Tracking
 - Daten in Datenbank
 - Server Logs
 - Cookies
 - Elastic Search Server bei Mastodon
 - i.a. keine externen Tracker, aber keine technische Garantie
- Authentizität
 - nicht garantiert, auch wenn öffentlicher Schlüssel im Profil verfügbar
- keine Werbung (nicht erzwungen)
- keine Firmen involviert (nicht erzwungen, z.B. pawoo.net)
- Terms and Conditions on mastodon.social

Der Fall Wil Wheaton

- Wesley Crusher in Raumschiff Enterprise
- Von 60 Usern dem Instanz-Admin gemeldet, der Account suspendiert hat
- Technologie löst oder verhindert in keiner Weise Konflikte zwischen Menschen
- Haupt-Unterschied: meist keine Firmenrichtlinien involviert

3. Hands-on:

Installation einer Pleroma Instanz

- Warum Pleroma als Beispiel?
- mit 32 EUR für einen Raspberry Pi bist Du dabei
- (das stimmt so nicht!)
- less moving parts verglichen zu Mastodon, weniger Ressourcenintensiv
- full disclosure: war Ruby on Rails dev, nun Elixir & Phoenix dev

Voraussetzungen

- Linux Betriebssystem (Ubuntu 16.04)
- Nginx Webserver
- PostgreSQL Datenbank
- aktuelles Elixir (Erlang Runtime) von Erlang Solutions
- git
- Openssh, Openssl
- Certbot (für Let's Encrypt Zertifikat)

Installationsschritte 1

- User pleroma anlegen
- Repo clonen
- Abhängigkeiten installieren
- Konfiguration generieren lassen
- Konfiguration anpassen
- Datenbank generieren lassen
- Datenbankstruktur generieren lassen
- Pleroma zum test lokal starten (ohne Nginx)
- ... und wieder stoppen
- Nginx konfigurieren
- SSL-Zertifikat installieren
- Nginx starten

Installationsschritte 2

- Systemd service Konfiguration erzeugen lassen
- Service Konfiguration anpassen
- Service starten
- ersten (eigenen) User erzeugen
- User zum Moderator machen
- Backup konfigurieren: besser 2 als 1, dezentral
 - Datenbank
 - Konfiguration (Applikationsverzeichnis)
- Terms of Service hinterlegen (wird Usern bei der Registrierung gezeigt)

Kontinuierliche Tasks

- Accounts anlegen
- oder Anmeldelinks erzeugen
- Passwörter rücksetzen
- User löschen
- User deaktivieren
- Wartung: Sicherheitsupdates einspielen (System und Applikation)
- Administration: verhalten sich die User:innen den Richtlinien der Instanz / der Gesetzeslage entsprechend?
- Backup testen

Accounts

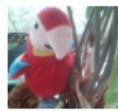
- -> Accounts anyone?
- have fun!
- Webinterface unter <https://pleroma.panoptikum.io>
- Mastodon FE Interface unter <https://pleroma.panoptikum.io/web>
- Verbindung mit Pleroma (oder Mastodon ;-)) fähigen-Clients sollten funktionieren.

Disclaimer und Argumente gegen das Selbsthosting

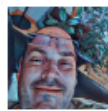
- Redundanz ist relativ schwer zu erreichen
- Also vielleicht doch besser auf einer fremden Instanz, der man vertraut?
- Admins können alle Nachrichten lesen
- Instanz mit User:innen ähnlicher Interessenslage können großen Vorteil der Timeline und der lokalen Suche bieten
- Es spricht nichts gegen mehrere Accounts auf mehreren Servern

4. Selbst ein Service konzipieren und implementieren

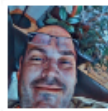
- konkrete Beispiele sind gut!
- [Panoptikum.io](https://panoptikum.io) - Podcast Verzeichnis
- Einbetten eines Widgets der letzten Toots für einen Account
- minimalistisch, macht aber noch sinn



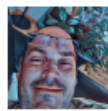
Stefan Haslinger [informatom](#) - boosted 4 days ago
[waldvogel](#) [waldvogel](#) - Anmeldungen für den
Blinkenrocket Easy Lötworkshop am 26.10. via Mail an
schule@privacyweek.at oder persönlich bei der
Privacyweek. Details
<https://fahrplan.privacyweek.at/pw18/talk/PFSAHS/>



Stefan Haslinger [informatom](#) - 4 days ago
Dabbling with Activitypub, I need a current toot.
(So sorry for the noise ;-)



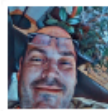
Stefan Haslinger [informatom](#) - 7 days ago
[@tom79](#) just wanted to let you know:
Happy mastalab user here. Never seen a bug or hiccup.
Congrats! and Thank you!



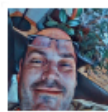
Stefan Haslinger [informatom](#) - 8 days ago
[@tobybaier](#) Podrunner



Stefan Haslinger [informatom](#) - boosted 9 days ago
[Davit Masia / PixelArt](#) [DavitMasia](#) -
Magic O_O . Is a static image. Found in some facebook
page.



Stefan Haslinger [informatom](#) - 10 days ago
[@PrivacyWeek](#) [@waldvogel](#) looking forward to
cooperate with [@noniq](#) for the first time ever!



Stefan Haslinger [informatom](#) - 10 days ago
[@pixelfed](#) I believe [@richard](#) wanted to know that...

Schritte

- Parameter ist Adresse, z.B.: @informatom@pleroma.panoptikum.io
- Von der Adresse muss auf öffentliches Profil geschlossen werden (etwa durch Webfinger, wir "erraten hier" der richtigen Pfad)
- Aus öffentlichem Profil wird die Adresse der Outbox bezogen
- Outbox Nachrichten werden eingelesen
- Unterscheidung zwischen Toot und Boost (wir ignorieren Likes)
- Toot kann einfach ausgegeben werden, wir müssen nur Profilbild ermitteln
- Dazu sollte Profil gecached werden (oder bereits sein ;-)
- Toots von Boosts müssen von Ursprungsserver gelesen werden
- Profile sind von Ursprungsserver zu beziehen (und am besten auch wieder zu cachen)
- von allen Bildern reichen wir nur URL durch und cachen nicht (das kann ein Privacy - Problem sein!)
- Bau des HTMLs zur Ausgabe des Widgets

Scope

- kein Erstellen von Toots
- kein Liken oder Boosten möglich
- keine Authentifizierung
- reine Clientfunktionalität
- keinerlei Serverfunktionalität
- also nur ein sehr kleiner Ausschnitt von ActivityPub
- aber: Kompatibilität zu Mastodon und Pleroma!

Aufgaben

sind eigentlich nur wenige zu erledigen:

- [Web Requests](#) absetzen (16 Zeilen Elixir)
- Antworten in JSON parsen (in Hashmaps o.ä.)
([Parsen](#): 34 Zeilen + [Aufbereitung](#): 117 Zeilen)
- [HTML erstellen](#) (26 Zeilen)

Probleme

- Header nicht gemäß Standard, richtig ist application/activity+json
- Pleroma <-> Mastodon: unterschiedliche Datenstrukturen
 - attributedTo vs actor
 - Pleroma: hat bereits die erste Seite Toots in der Outbox, Mastodon nicht

-> ActivityPub ist nicht strikt standardisiert

Aufwand

- überraschend viele Requests -> Caching nötig
- sich in das Protokoll ActivityPub einlesen
- Bibliotheken auswählen
- in Responses einlesen und Links "nachlaufen"
- Unterschiede Pleroma/Mastodon
- ~ 10 Stunden

Pro Tipp

Pretty print json for user profile:

```
curl -H "Accept: application/activity+json"  
      https://pleroma.panoptikum.io/users/informatom  
| python -m json.tool`
```

Lesetipps 1

- Wikipedia Seite [Fediverse](#)
 - Ausführlicher Blogpost mit vielen weiteren Aspekten von Matthias Meißer: [Erfahrungsbericht Mastodon](#)
 - Blogpost [What is Pleroma](#) von Pleroma Entwickler Lain
 - [The Federation](#) - Statistik Server zum Fediverse
 - [Privacy and Tracking on the Fediverse](#)
 - Vergleich, wie groß Email noch immer ist: [DomainTools Internet Statistics - Mailservers](#)
 - [fediverse.space](#) - Visualisierung der Vernetzung zwischen den Instanzen
 - [Twitter Mastodon Bridge](#) - finde deine Freunde auf Mastodon
 - Liste von [Mastodon Tools](#) auf Github
- Fragmentierung wird deutlich

Lesetipps 2

- Der Fall Wil Wheaton
 - Statement auf seinem Blog: [The world is a terrible place right now, and that's largely because it is what we make it.](#)
 - Blog Post by Amber Enderton: [Wil Wheaton Has a Listening Problem](#)
 - Artikel auf The Verge: [Twitter rival Mastodon isn't safe from online mobs either](#)
- [activitypub rocks](#) test suide
- [ActivityPub](#) - der W3C Standard
- Pleroma Installationsanleitung: [Installing on debian based distributions](#)
- [Pleroma Wiki](#)
- QR-Code Generator: [qrcode](#)
- [Terms and Conditions](#) von mastodon.social

Diese Presentation ist online:

<https://blog.panoptikum.io/pw18>

Stefan: stefan@panoptikum.io

twitter: [@informatom](https://twitter.com/informatom)

fediverse: @informatom@pleroma.panoptikum.io