Secure Scuttlebutt

Peer-to-peer Collaboration and Community Infrastructure

March 15, 2020
LibrePlanet
Boston, MA

Charles E. Lehner

`@f/6sQ6d2CMxRUhLpspgGIulDxDCwYD7DzFzPNr7u5AU=.ed25519`

Me

@cel

Developer
git-ssb, patchfoo, sbotc

Using SSB since 2014

SSB

Efficient Reconciliation and Flow Control for Anti-Entropy Protocols

Robbert van Renesse* Dan Dumitriu† Valient Gough Chris Thomas
Amazon.com, Seattle

ABSTRACT
The paper shows that anti-entropy protocols can process only a limited rate of updates, and proposes and evaluates a new state reconciliation mechanism as well as a flow control scheme for anti-entropy protocols.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *network communications*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *distributed applications*; D.1.3 [**Programming Techniques**]: Concurrent Programming – *distributed programming*; D.4.4 [**Operating Systems**]: Communications Management – *network communication*; D.4.5 [**Operating Systems**]: Reliability – *fault tolerance*;

General Terms: Algorithms, Reliability.

Additional Key Words and Phrases: Epidemics, Anti-Entropy, Gossip, Flow Control

1. INTRODUCTION
Anti-entropy, or *gossip*, is an attractive way of replicating state that does not have strong consistency requirements [3]. With few limitations, updates spread in expected time that grows logarithmic in the number of participating hosts, even in the face of host failures and message loss. The behavior of update propagation is easily modeled with well-known epidemic analysis techniques. As a result, many distributed applications use gossip to contain various inconsistencies.

In spite of its popularity, little study has been done into how gossip protocols behave under high update load. Gossip protocols purport to deliver messages within a certain configurable number of rounds with high probability, and thus provide synchronous guarantees. Like any other synchronous communication channel, gossip has capacity that is limited by available bandwidth for transporting gossip data and CPU cycles for generating and processing the gossip messages. Under high update load, a gossip protocol may not be able to send all updates required to reconcile differences between peers. Updates would take arbitrary time to propagate as the gossip channel gets backed up.

Gossip protocols are designed to be non-invasive and have predictable performance, and for this a designer has to fix not only the gossip rate per participant but also the maximum size of gossip messages (*e.g.*, maximum UDP packet size). While this avoids network and CPU overload, it also limits the capacity of the gossip channel.

This paper makes two contributions. First, it presents a new state reconciliation mechanism that is designed both for minimal CPU overhead and for situations in which only limited bandwidth is available (Section 3). Second, it proposes and analyzes a flow control scheme for gossip (Section 4). Related work is discussed in Section 5.

2. GOSSIP BASICS
There are two classes of gossip: *anti-entropy* and *rumor-mongering* protocols. Anti-entropy protocols gossip information until it is made obsolete by newer information, and are useful for reliably sharing information among a group of participants. Rumor-mongering has participants gossip information for some amount of time chosen sufficiently high so that with high likelihood all participants receive the information. In this paper, we shall focus on anti-entropy—reconciliation and flow control for rumor-mongering have received considerably attention already (see Section 5).

Let $\mathcal{P} = \{p, q, ...\}$ be a set of participants. Each participant maintains state, which we model as a mapping $\sigma \in \mathcal{S} = \mathcal{K} \rightarrow (\mathcal{V} \times \mathcal{N})$. Here $\mathcal{K}$ is a set of keys, $\mathcal{V}$ a set of values, and $\mathcal{N}$ an infinite ordered set of version numbers. $\sigma(k) = (v, n)$ means that key k is mapped to value v and version n. A more recent mapping for the same key contains a larger version number. Both value and version number spaces contain a $\perp$ element, and in case of $\mathcal{N}$, $\perp$ is the lowest element. Initially all keys on all participants are mapped to $(\perp, \perp)$.

A participant's state is mutable and is replicated onto all participants. We model this as a mutable mapping $\mu_p : \mathcal{P} \rightarrow \mathcal{S}$ maintained by each participant p. A participant p is only

*Contact author. Current address: Dept. of Comp. Sc., Cornell University. Email: rvr@cs.cornell.edu
†Current address: Ballista Securities, New York.

%xRPUKpUVLyyMJw/obpJXGyl8D3N3VJKE4d6BObXFGDk=.sha256

- p2p, f2f, e2e
- gossip protocol
- social network
- application platform

Epidemic Broadcast Trees *

João Leitão, University of Lisbon, jleitao@lasige.di.fc.ul.pt
José Pereira, University of Minho, jop@di.uminho.pt
Luís Rodrigues, University of Lisbon, ler@di.fc.ul.pt

Abstract

There is an inherent trade-off between epidemic and deterministic tree-based broadcast primitives. Tree-based approaches have a small message complexity in steady-state but are very fragile in the presence of faults. Gossip, or epidemic, protocols have a higher message complexity but also offer much higher resilience.

This paper proposes an integrated broadcast scheme that combines both approaches. We use a low cost scheme to build and maintain broadcast trees embedded on a gossip-based overlay. The protocol sends the message payload preferably via tree branches but uses the remaining links of the gossip overlay for fast recovery and expedite tree healing. Experimental evaluation presented in the paper shows that our new strategy has a low overhead and that is able to support large number of faults while maintaining a high reliability.

1. Introduction

Many systems require highly scalable and reliable broadcast primitives. These primitives aim at ensuring that all correct participants receive all broadcast messages, even in the presence of network omissions or node failures. Gossip protocols [8] have emerged as a highly scalable and resilient approach to implement reliable broadcast. Unfortunately, in steady-state, gossip protocols exhibit an excessive message overhead in order to ensure reliability with high probability. On the other hand, tree-based broadcast primitives have a small message complexity in steady-state, but they are very fragile in the presence of failures, lacking the natural resilience of epidemic protocols.

*This work was partially supported by project "P-SON: Probabilistically Structured Overlay Networks" (POS_C/EIA/60941/2004).

%ipyEyC3oODPTf2X/GyCzU7vgPB/Ut5uay4MtT5w3X9U=.sha256

Cypherlinks

Message

%Oys+Dg6P0h8LvykiMTQW5L43iwuJpuvOIiVW5RZs9Cw=.sha256

Blob

&C93joUl4xSFH3cVJM6ZZQdAr+WXIiDLIyUQDZ//y4qw=.sha256

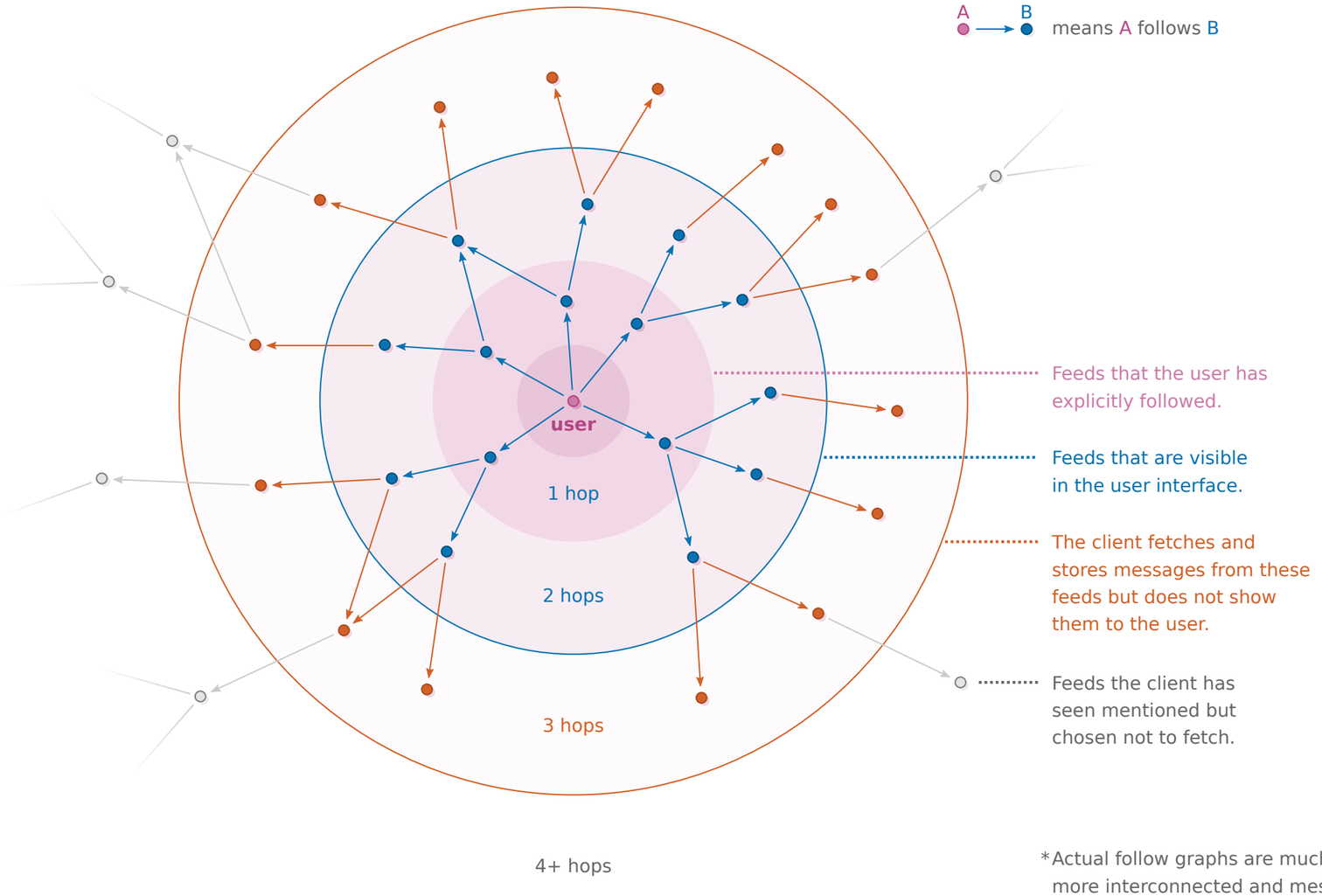
Feed

@f/6sQ6d2CMxRUhLpspgGIulDxDCwYD7DzFzPNr7u5AU=.ed25519

Message

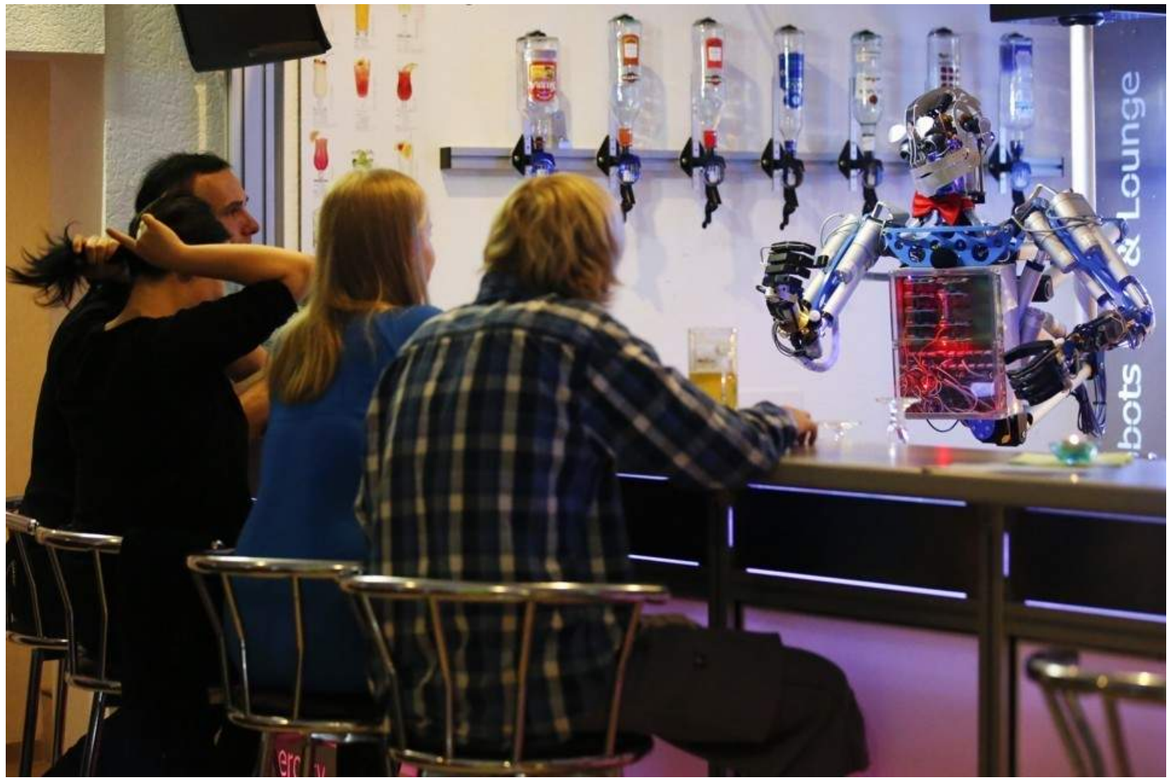

```
{
  "key": "%8XKAub/+keTbOXU+eS1/r2+wDMfk0369fLJ57Re9Ku0=.sha256",
  "value": {
    "previous": "%0tpLhWn8nC1iA+thcZnVYt5Nm6OtUvkFXMuHdB9roS4=.sha256",
    "author": "@C3iYh/12sO1uvKq1KcZXLFxSySzxOkHxXN8rtNB5MGA=.ed25519",
    "sequence": 1359,
    "timestamp": 1506213306259,
    "hash": "sha256",
    "content": {
      "type": "post",
      "text": "hello world"
    },
    "signature": "Nz4WOuOMpXU2xXUgW+cGTLCiU1BCYiHQFWvyUnppq+YmwV1TZM
5+FM12SAclA2BFgRdHyG6Y9lDCXHJdMFiMAw==.sig.ed25519"
  }
}
```

Social graph



%xRixf1ofCAtiz2/MKbzfcOGXYGBgFbrA4iuYRfHvmW8=.sha256

Pubs



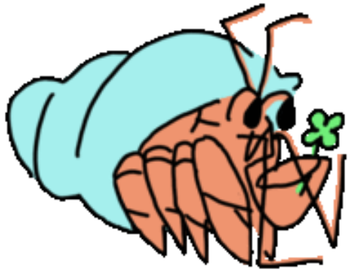
%7Puw6nBVX0uaL/Q7C6jv4sJiv2xcVv465JoyQseLPcs=.sha256

%AsywVupvKB2HcnGX7OMsuo7G32uq+sEM0+qGzJ3YZ9w=.sha256

Herm[i]es

%lmP6wJqkUFbeke8b0FJ8DCcFln7rWJaaoms/gN3FzAA=.sha256

%6Ykl7b5uYr429NTJEvtPYQijCvyA5nK/EZh55mj5574=.sha256

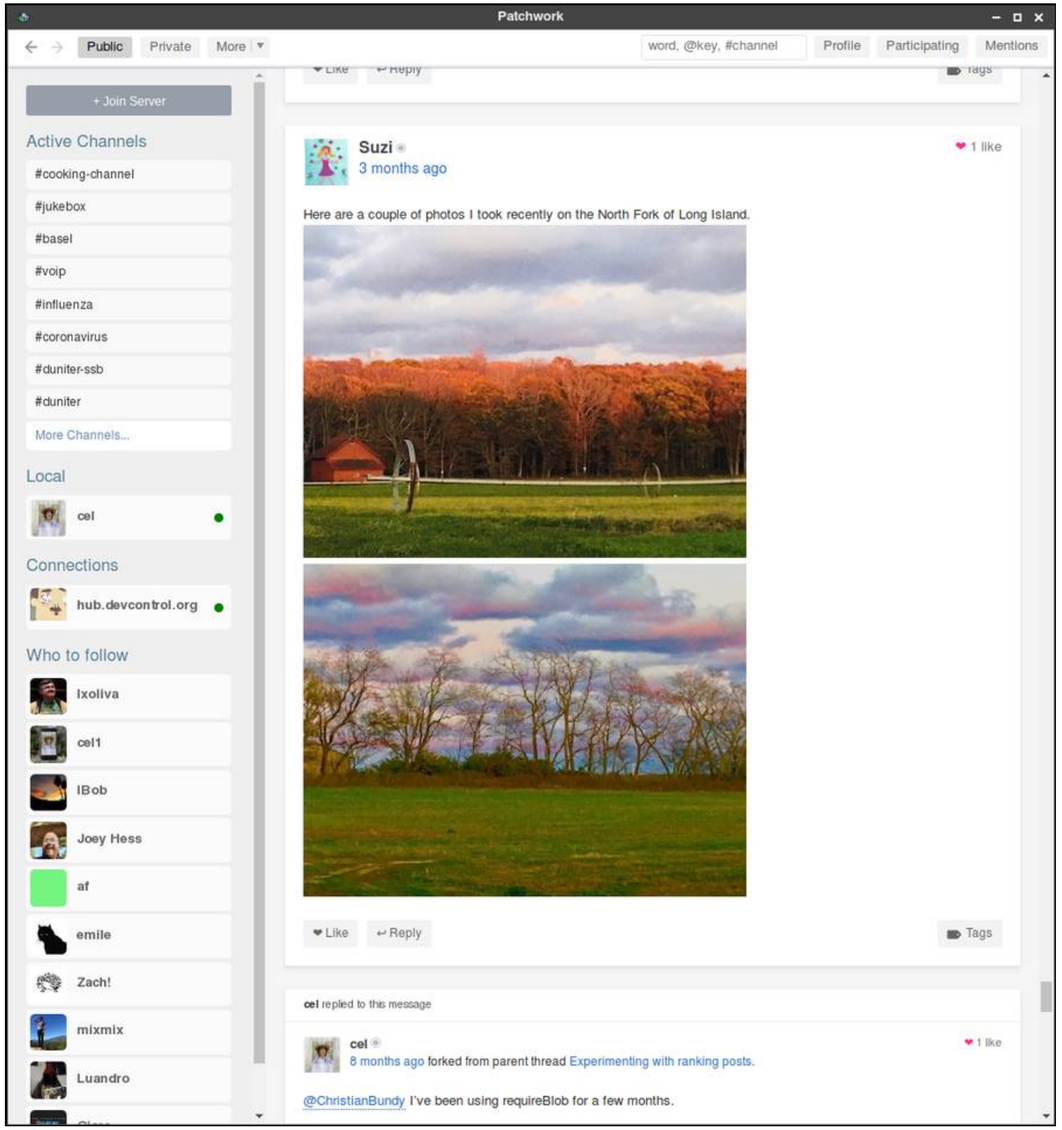


%muLMFIRjAuMiOs7gKGQrisWF9QEX7PjHJR7HrwmNbac=.sha256

%PeCILG55OGNA/6V6dnWoIX6v8KdRHVqyPBgYT9neU6g=.sha256

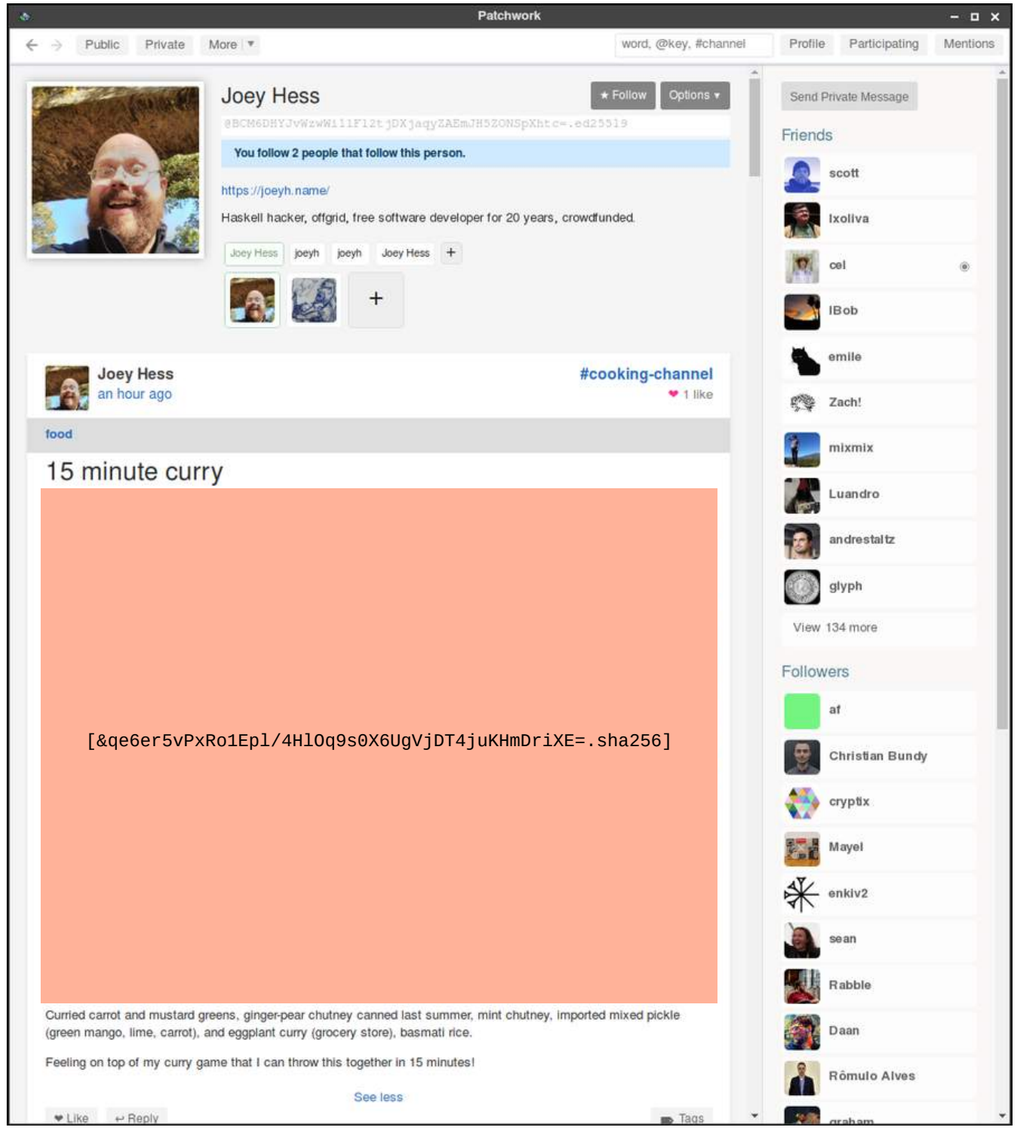
Patchwork

Public Feed



%n/hAHVDJevZ92Nteu0CKlx9YKDDxqejPsPB/jw49llc=.sha256

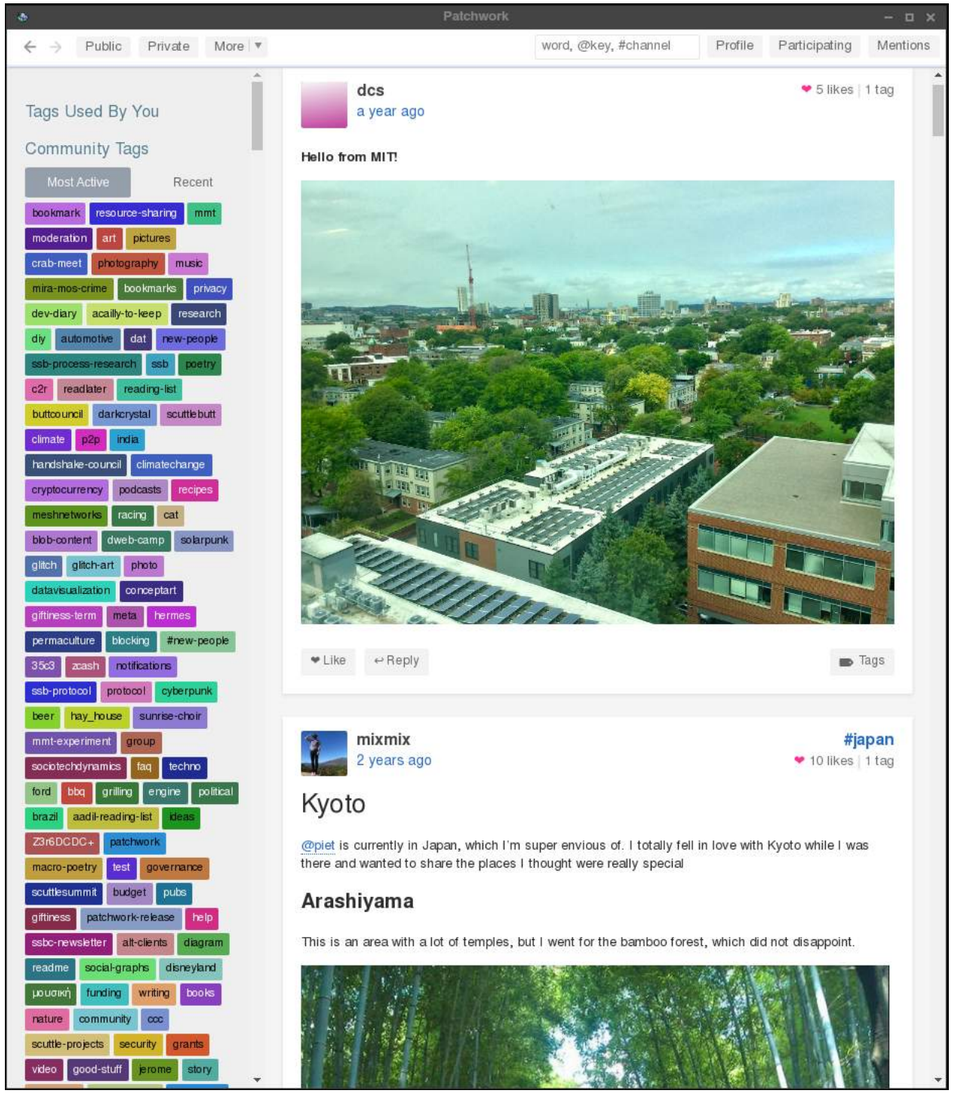
Profile



%xZ8vcTR67JCFtYOYsnSgsWIKfAnJarg8aMMY5ELjTrA=.sha256

Patchwork

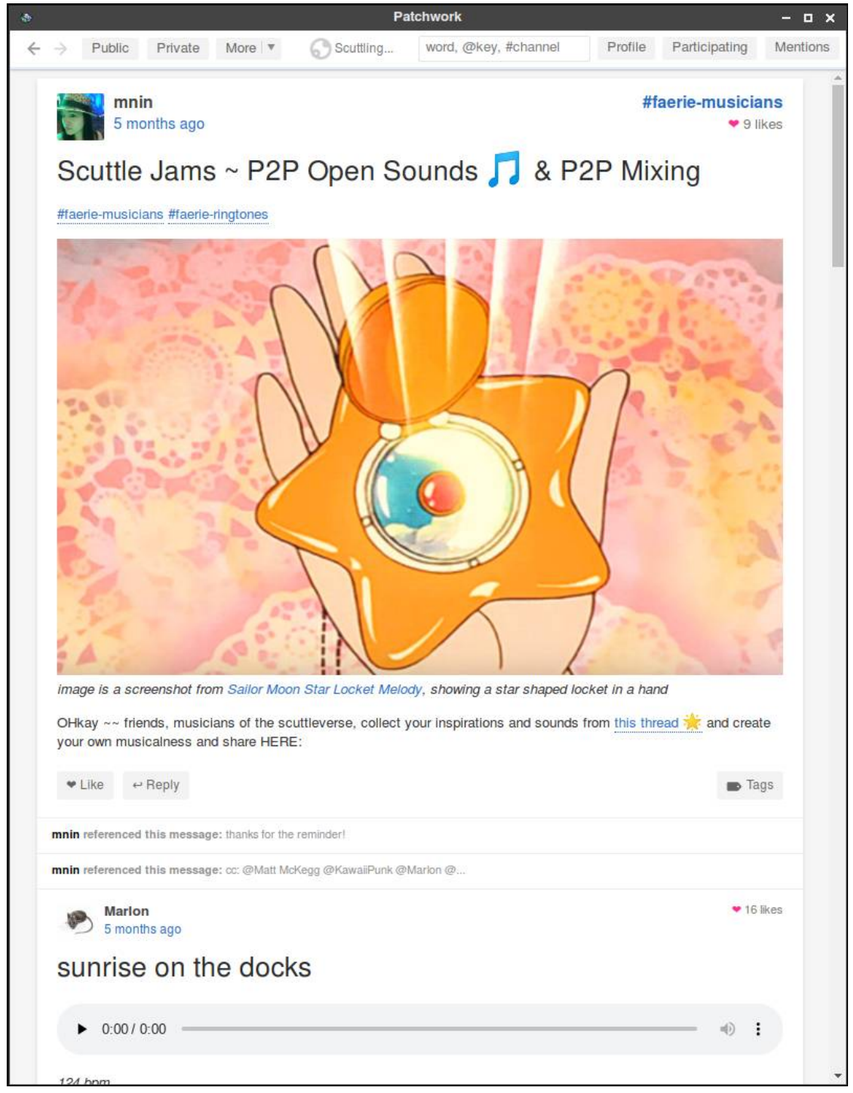
Tags



`%+1kBGWr4TMZkKvXCn7VboGkv6G23c32Kun2xl9m3Qjk=.sha256`

`%0rpoHSyVRIkY9lSXkjF+B68x+d8RNyVJy/ErvAF82T8=.sha256`

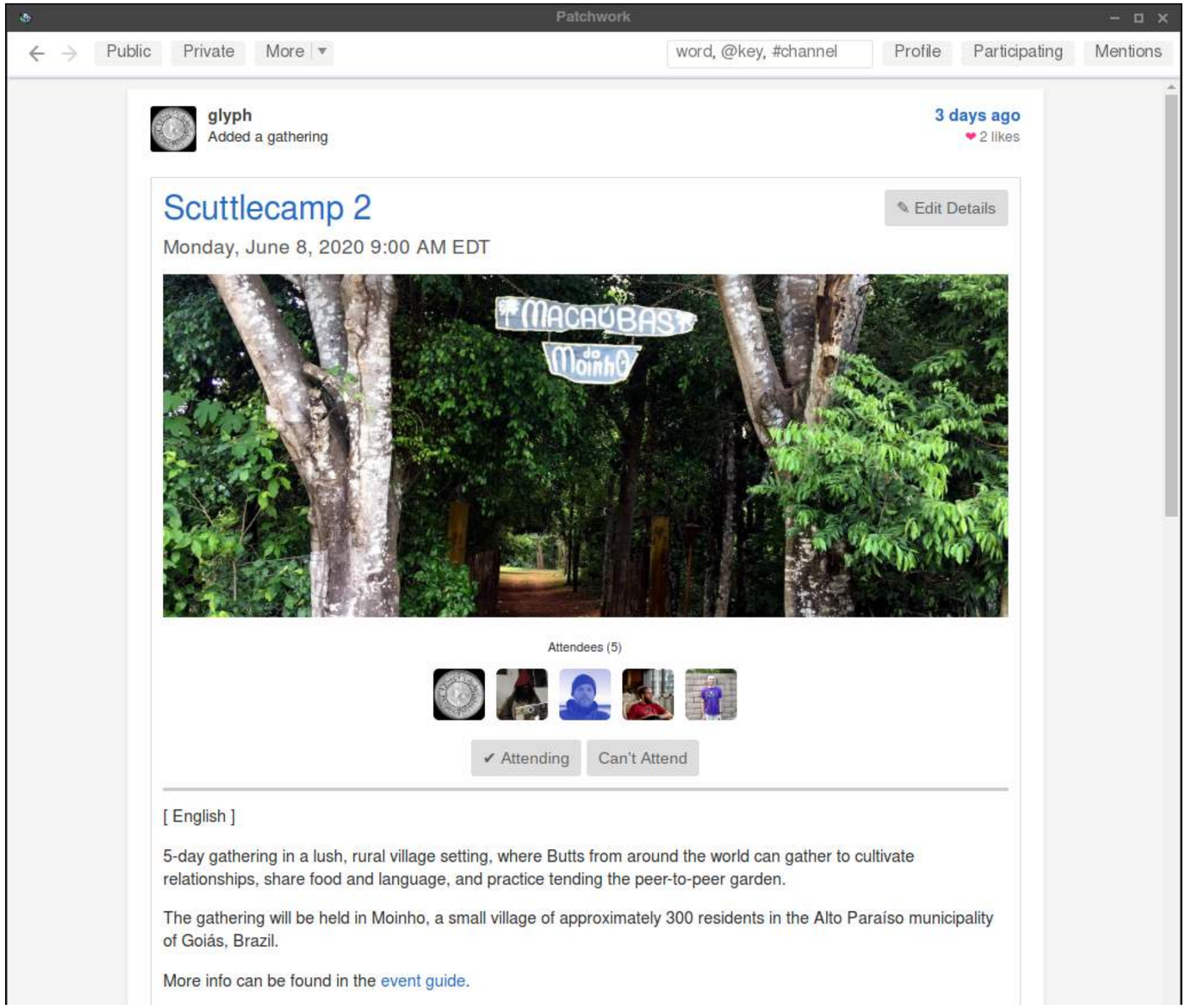
Thread



`%wcR3H38WfELYOFRno/PpKBqgo90Gk273ya+mRM4kDxQ=.sha256`

`%lCX+OgVrPgs44b6meFz3kE54r4CW19y0oiU6rrV6GQU=.sha256`

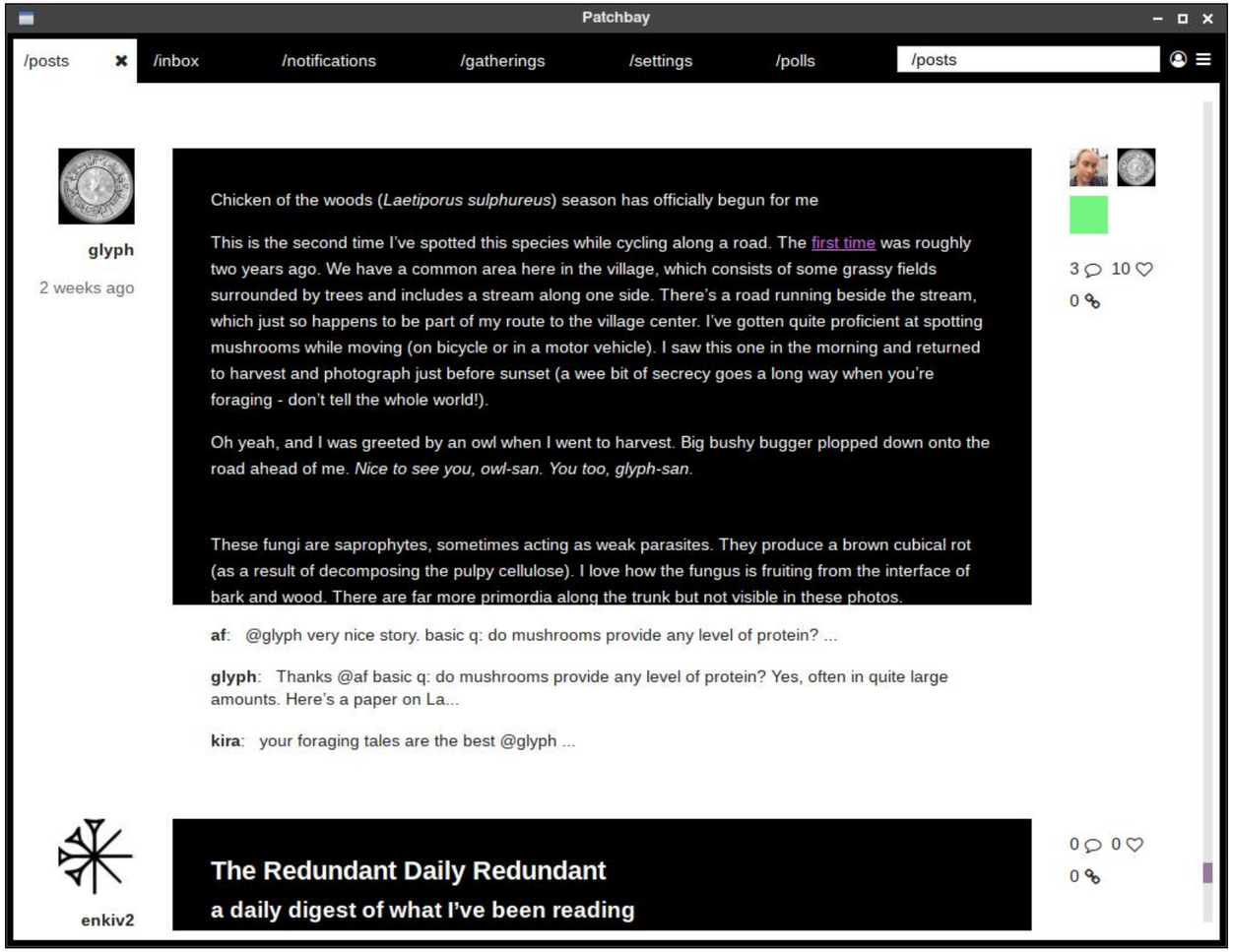
Patchwork - Gatherings



%EQXXQhhDdVNDvj5R9klck6/V8kYz65NiaQld4RBd/uY=.sha256

Patchbay

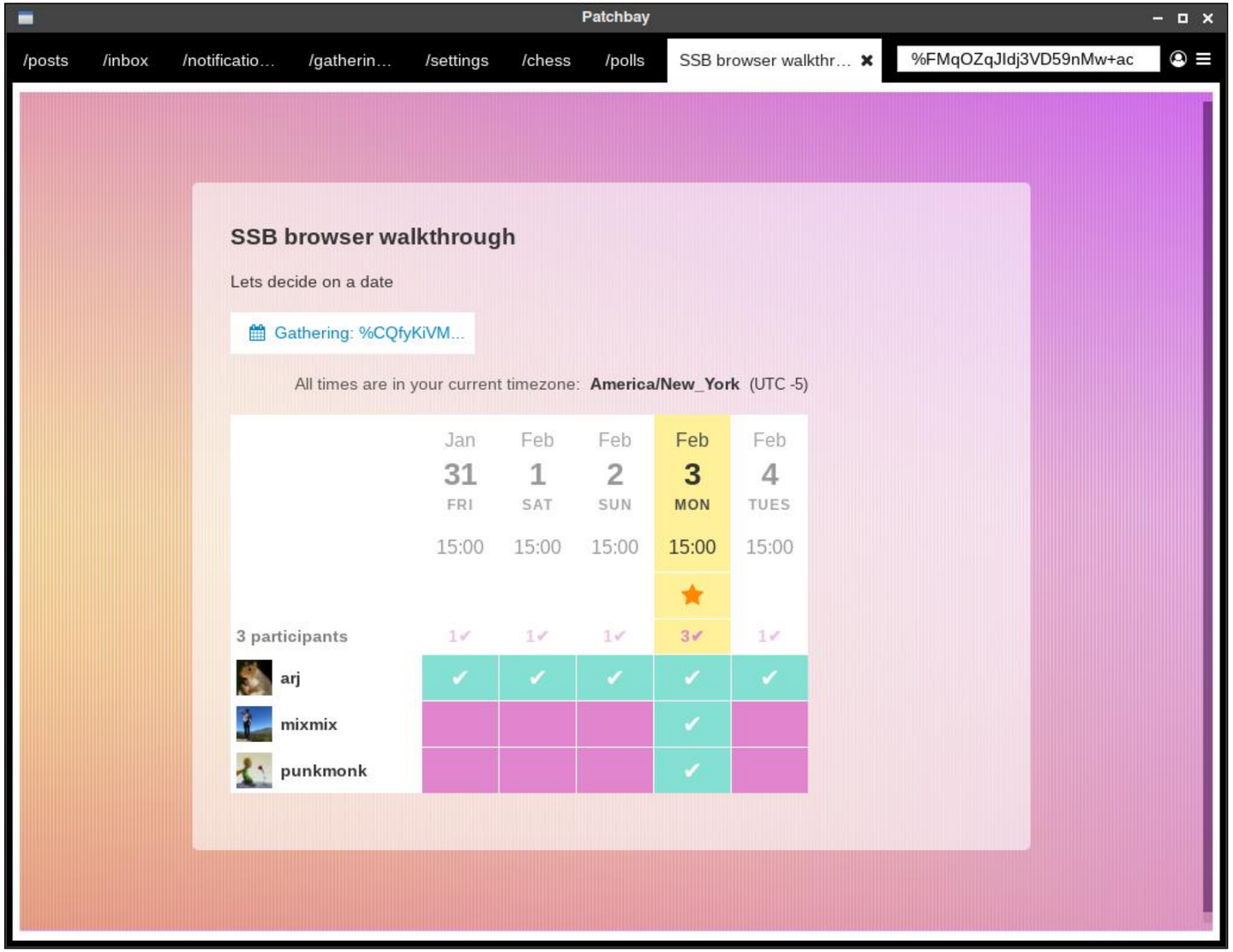
Thread



`%RP5p5qBqjsW/5Rl68xGIcn58X3MiyinsqsbLTHWbnM4=.sha256`

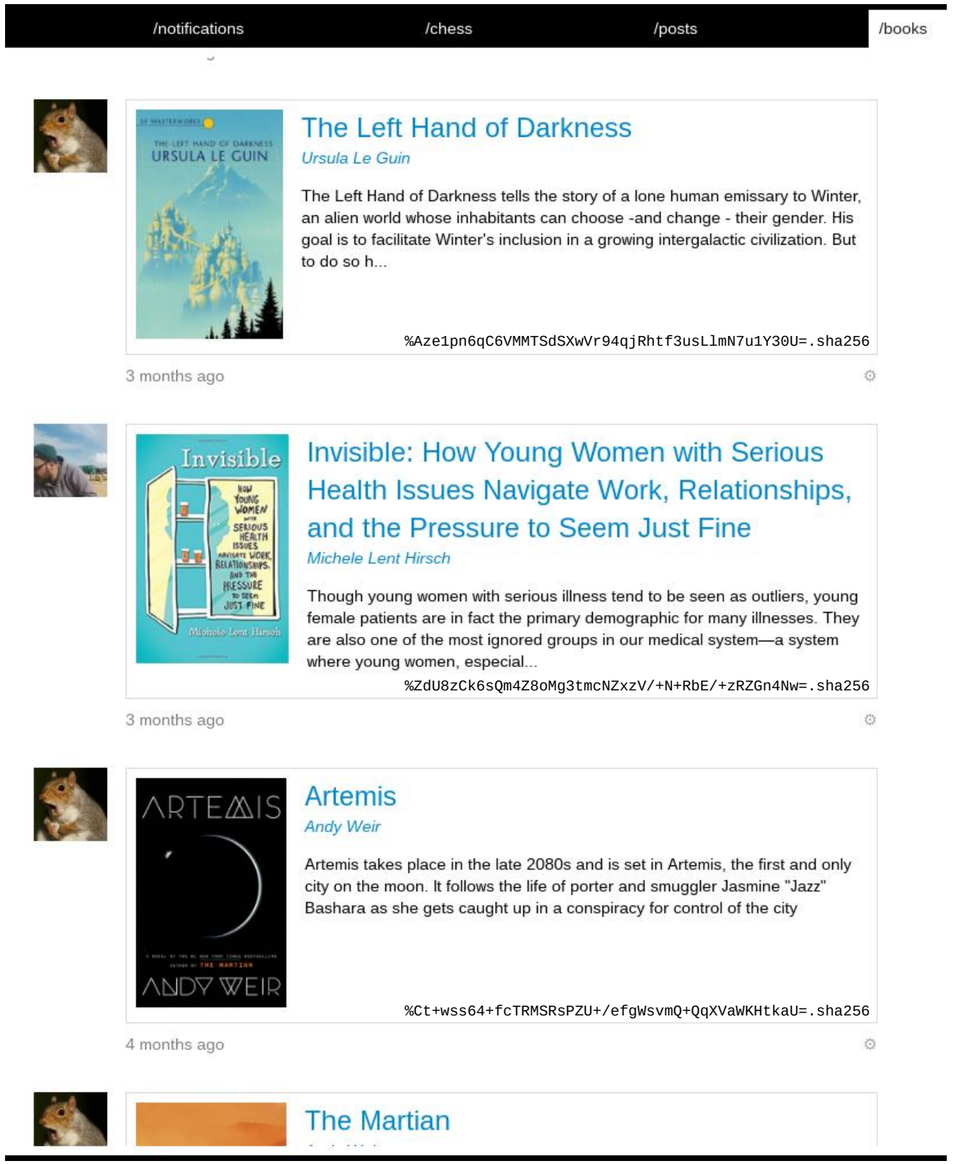
Patchbay apps

Polls



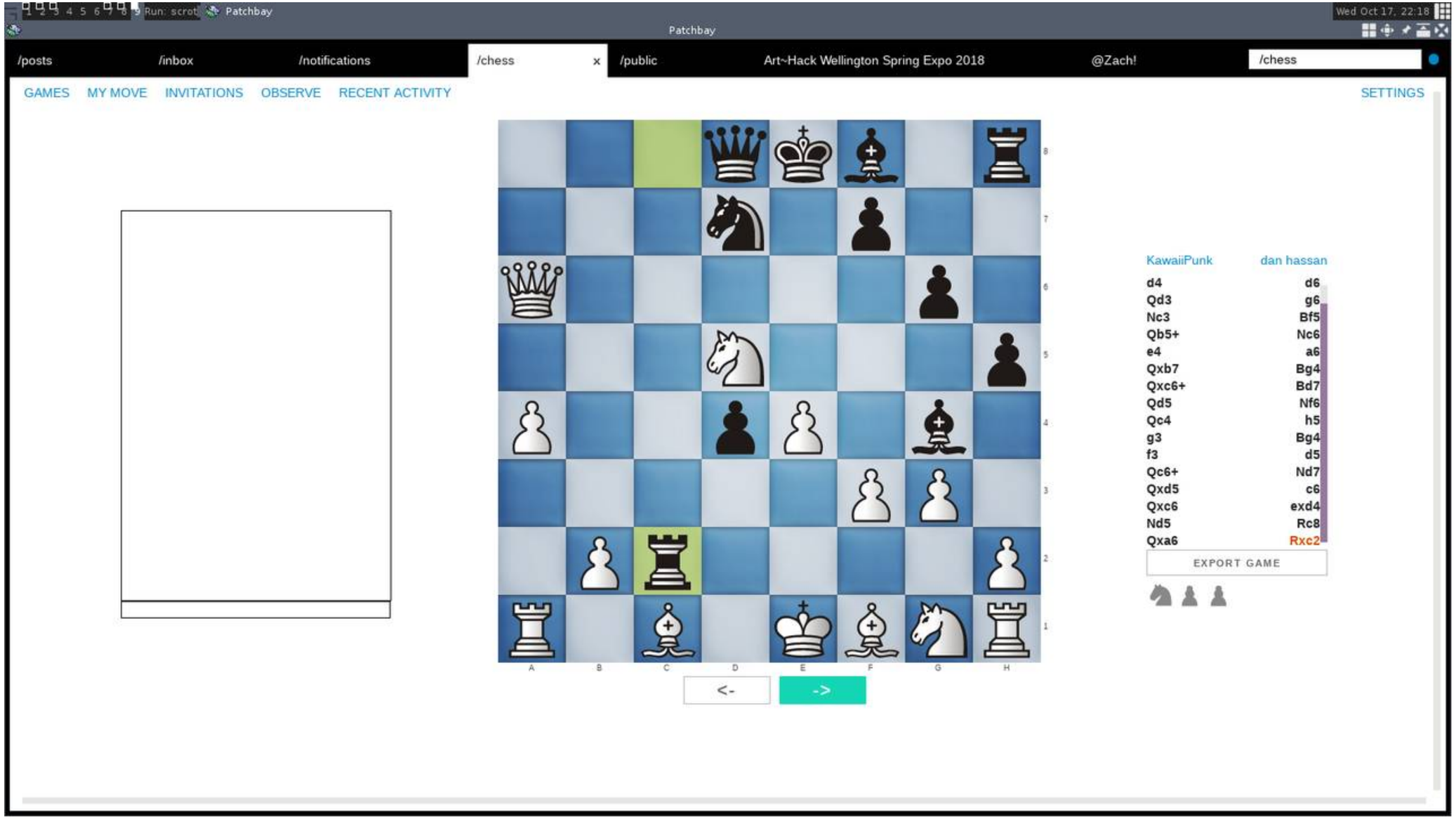
%FMqOZqJIdj3VD59nMw+acmZFvQ/OSXaWr+cfdIH6F9s=.sha256

Books



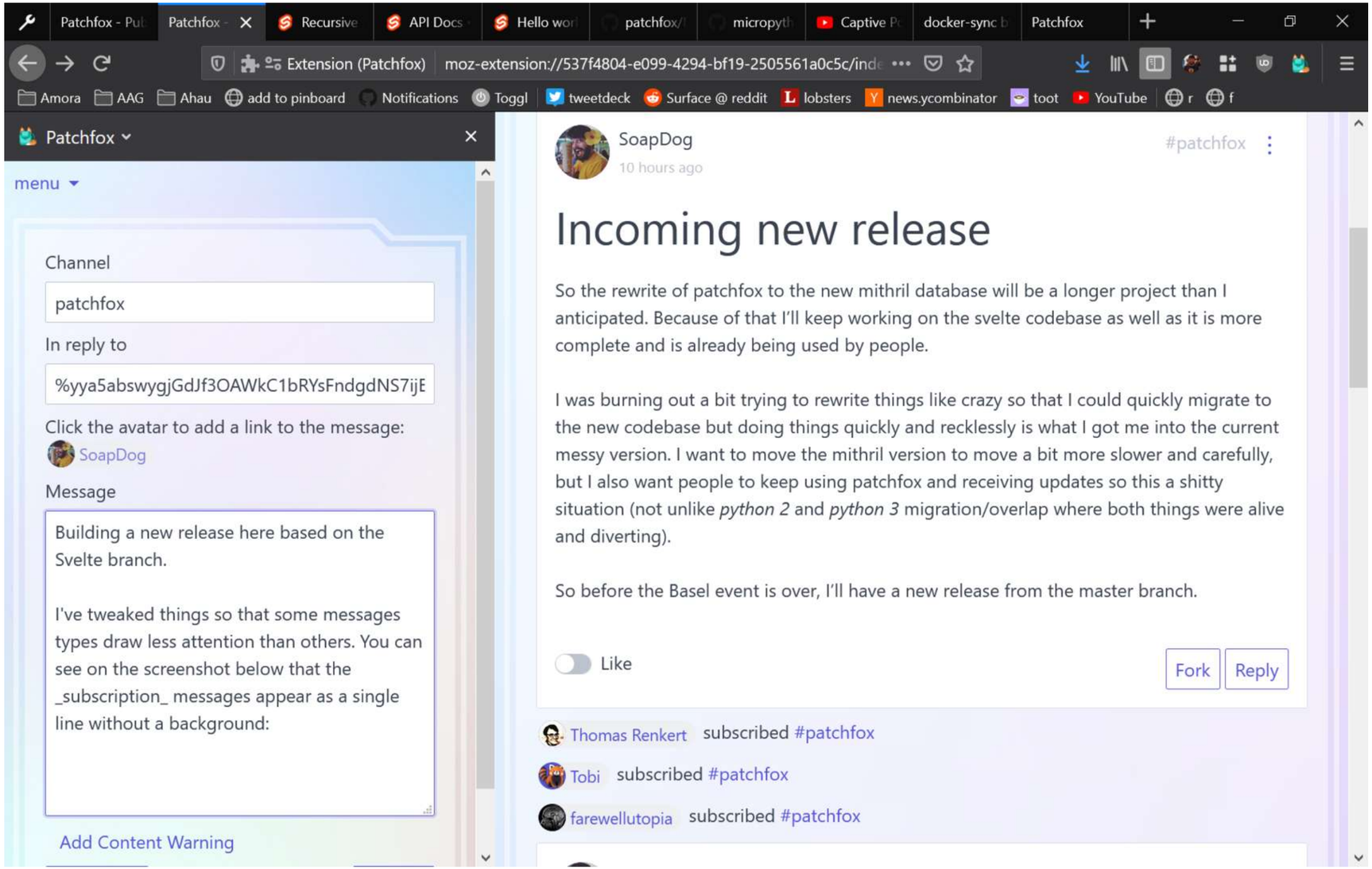
Patchbay apps 2

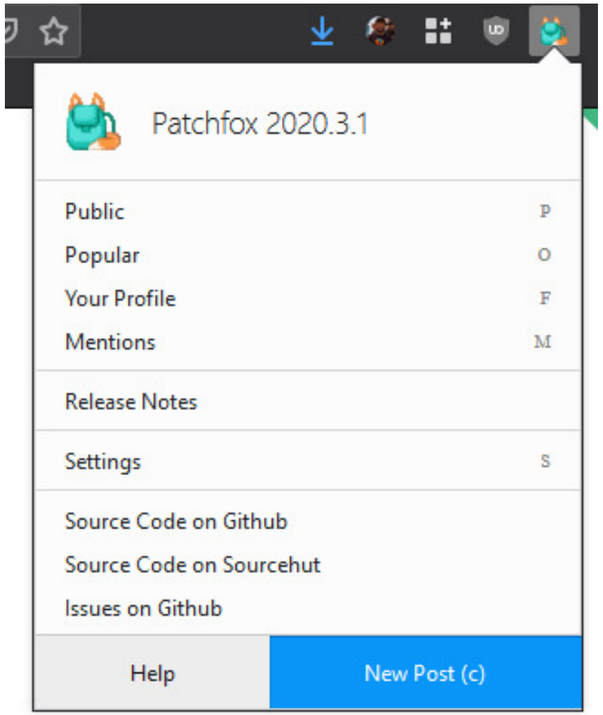
Chess



%jyn3lc2F6ICLxFbYbvrfsdKp1fDQWyiZDdiz5gYdtI8=.sha256

Patchfox

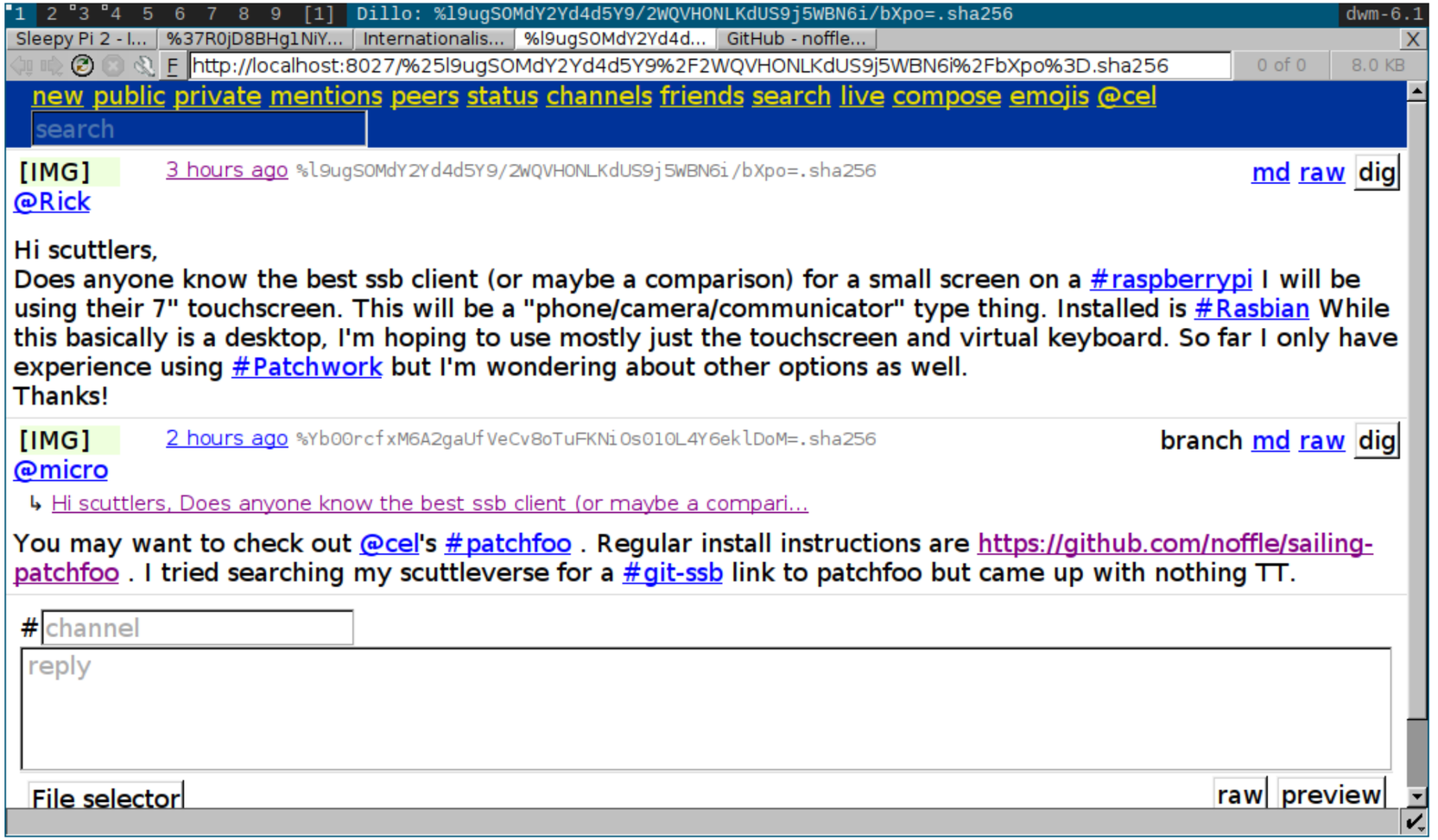




`%vNnZtiCafFum3VGH2cOhbvaeMCXJnVl9HrQhH7WkPto=.sha256`

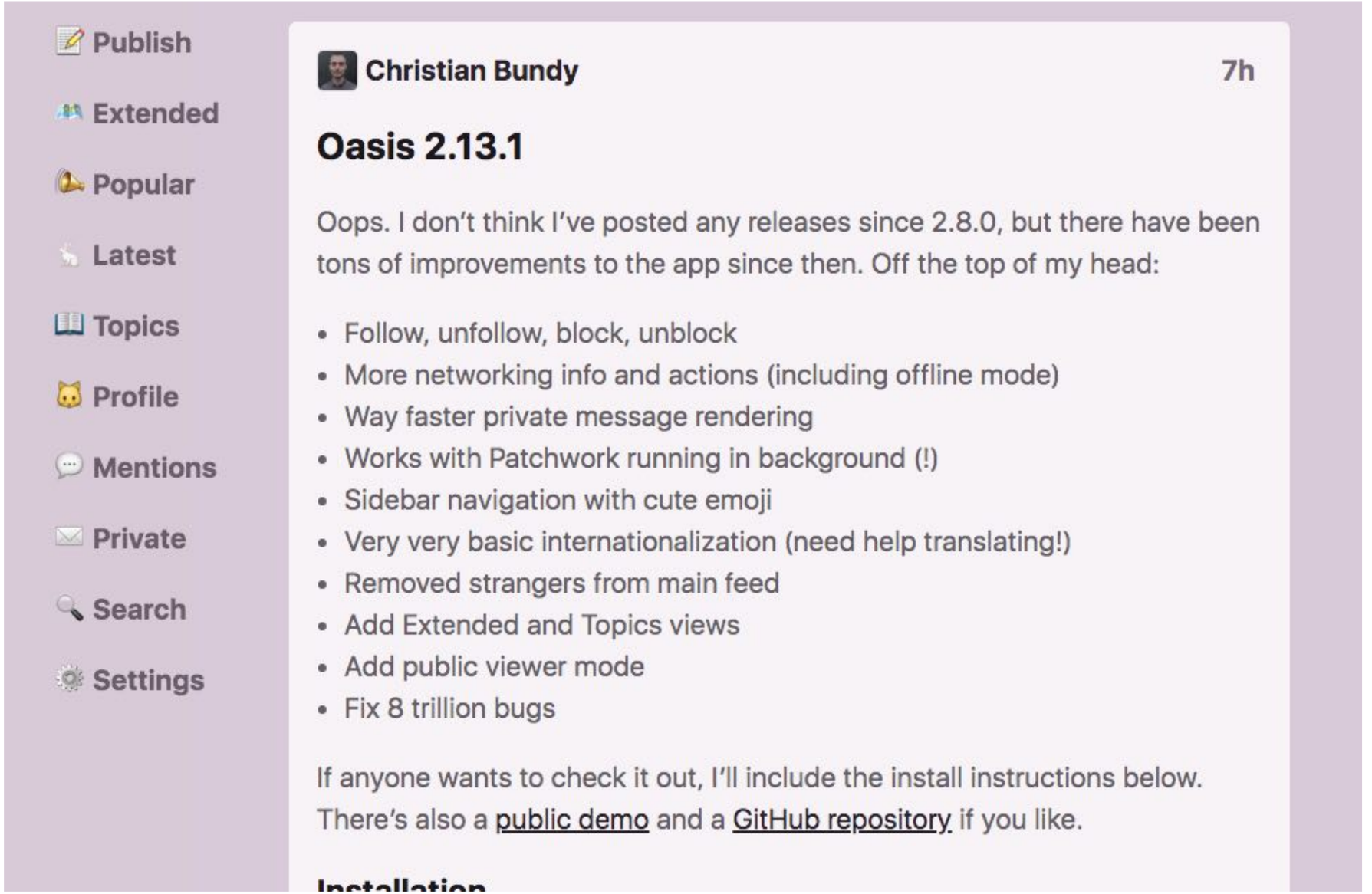
`%iAuYhV5rAs+t/+gyWVilPOTSnmnEl9SJdj6bZlSnRRA=.sha256`

patchfoo



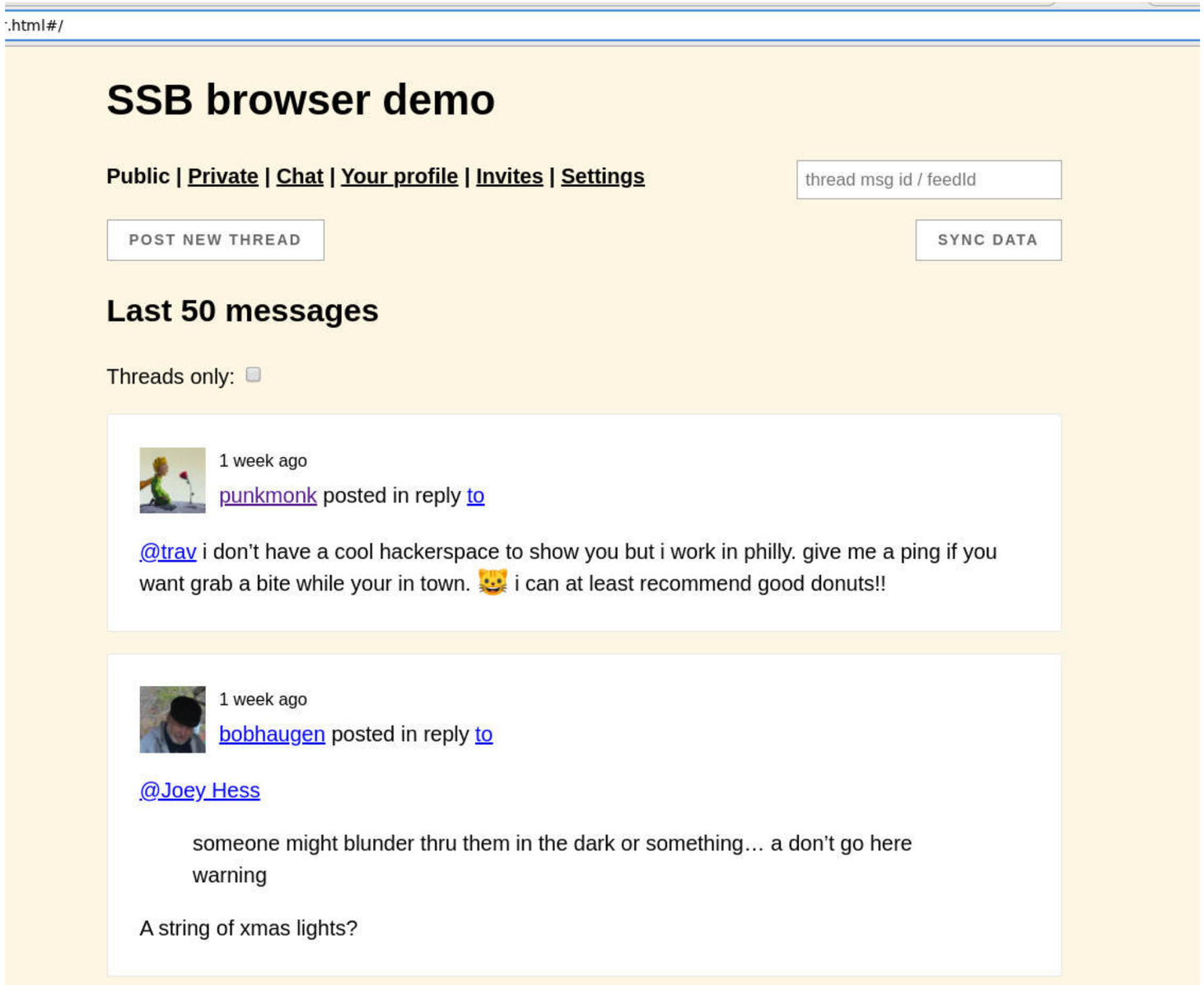
%O3uj3mNvhbwTNb7Mo/30FNo46pYKakNvc3G0Y0DYif8=.sha256

Oasis



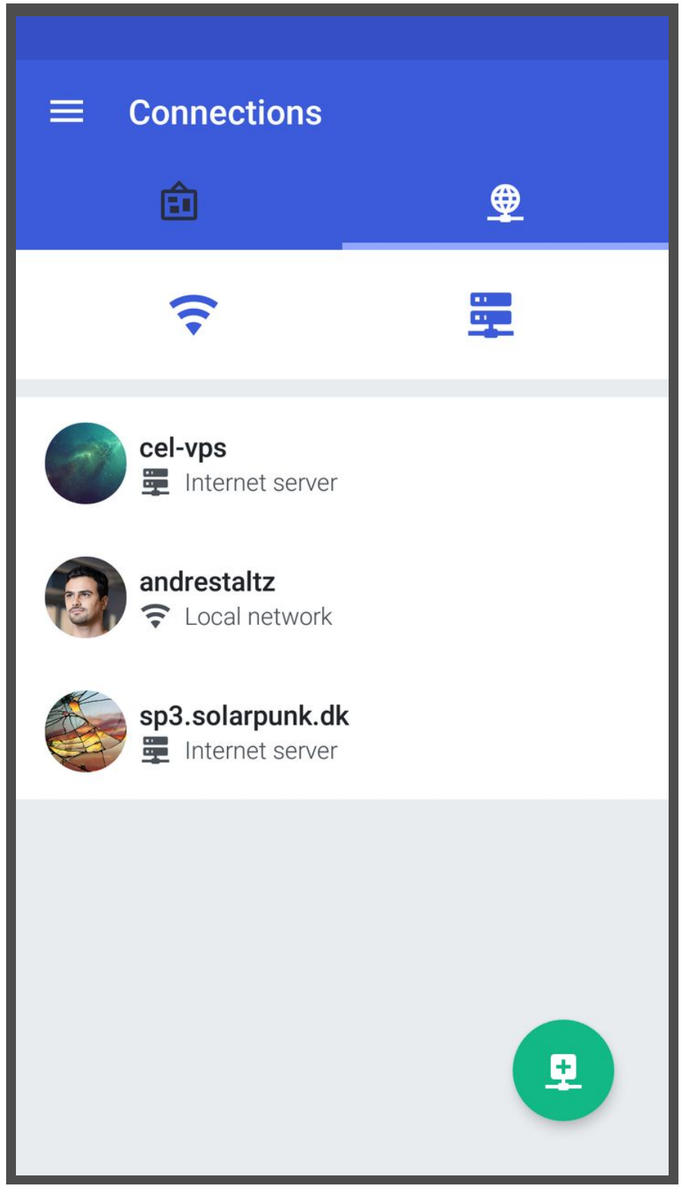
%u+Mnq1AFB2w7BV3F27jfi2EcZiyPc7YCnK1mEf+9Z9U=.sha256

ssb-browser-demo

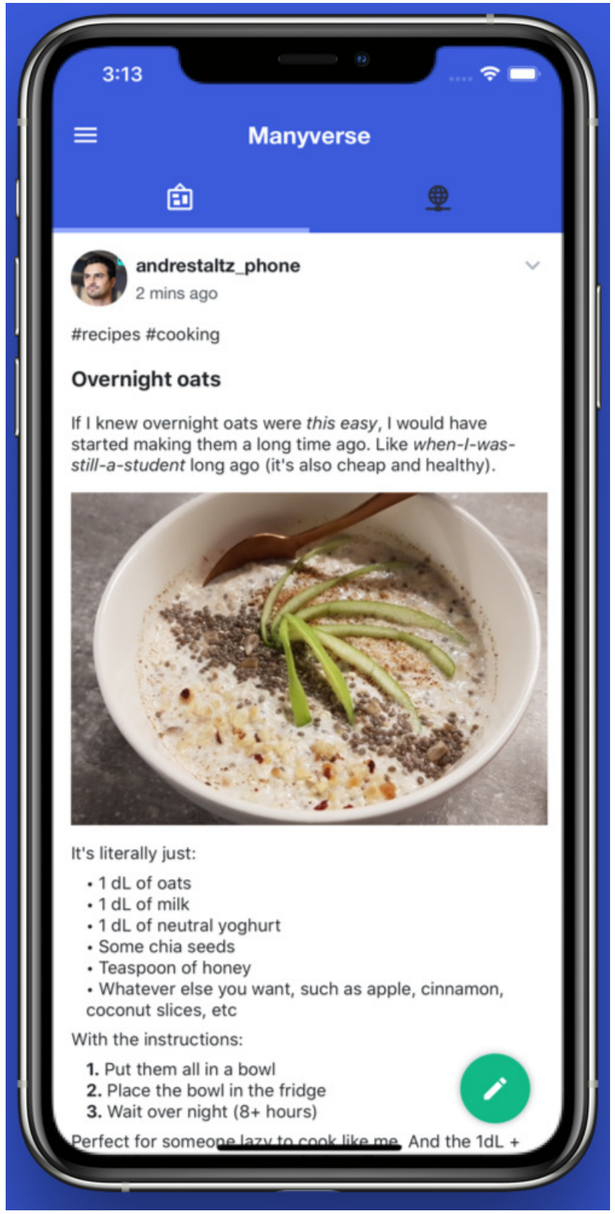


%pc3IVvokQreZF1a0q448THRccPcuBv5PHhuO6sBUTIA=.sha256

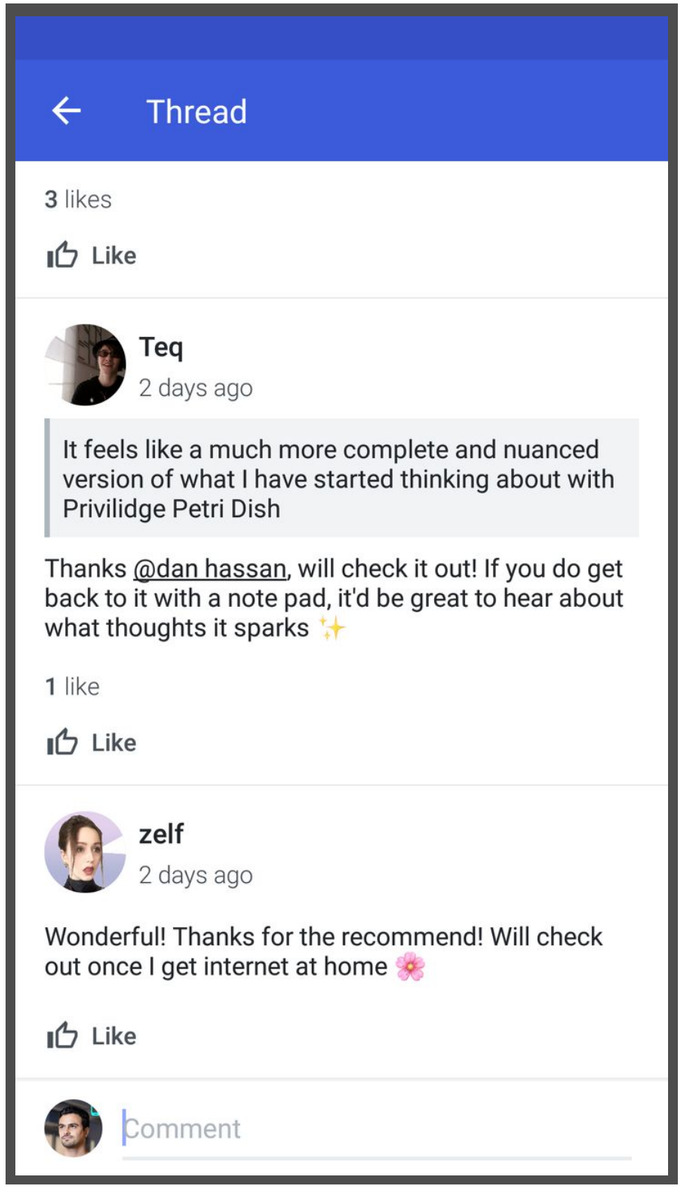
Manyverse



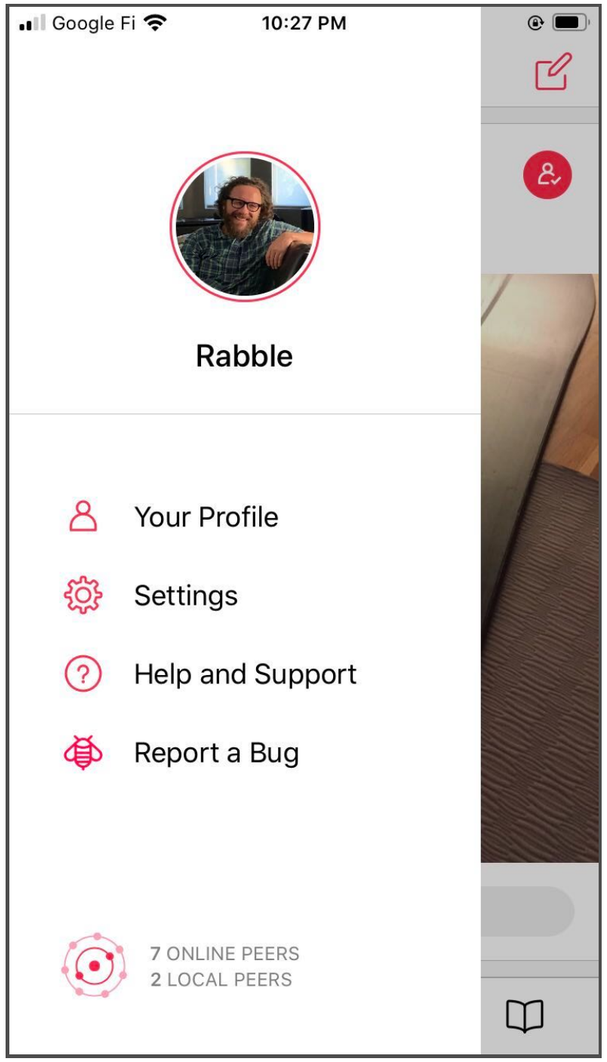
%P7B6o2keZ3UNVWqohSzTzGWzkt7Xum0cehoe6GpkZXQ=.sha256



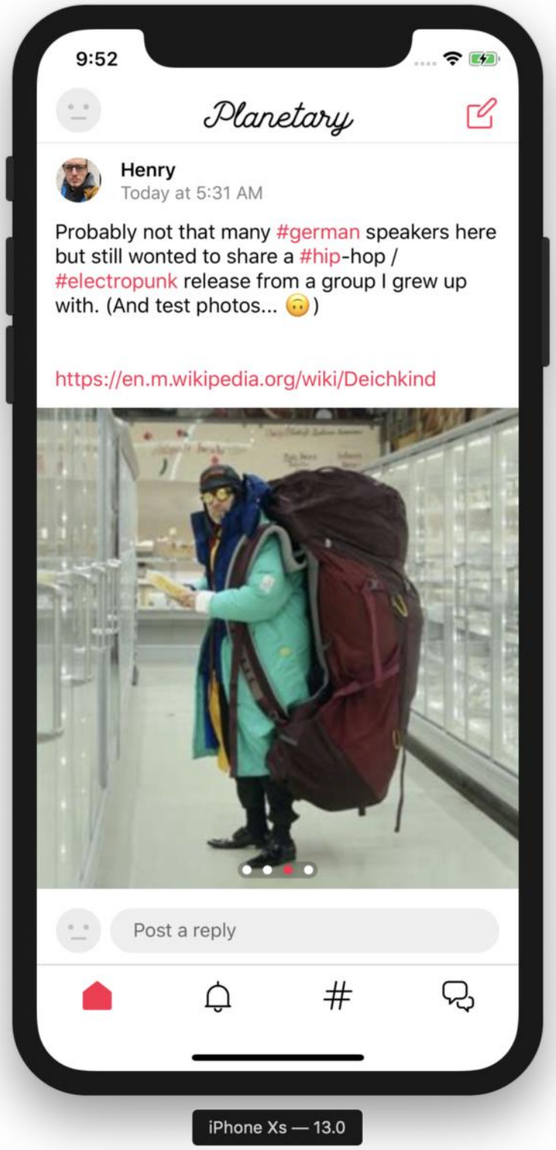
%yNkrJtnHyR4LcgzYvJ3nYdvQ1U3i2SlVxzbs7vOH5WU=.sha256

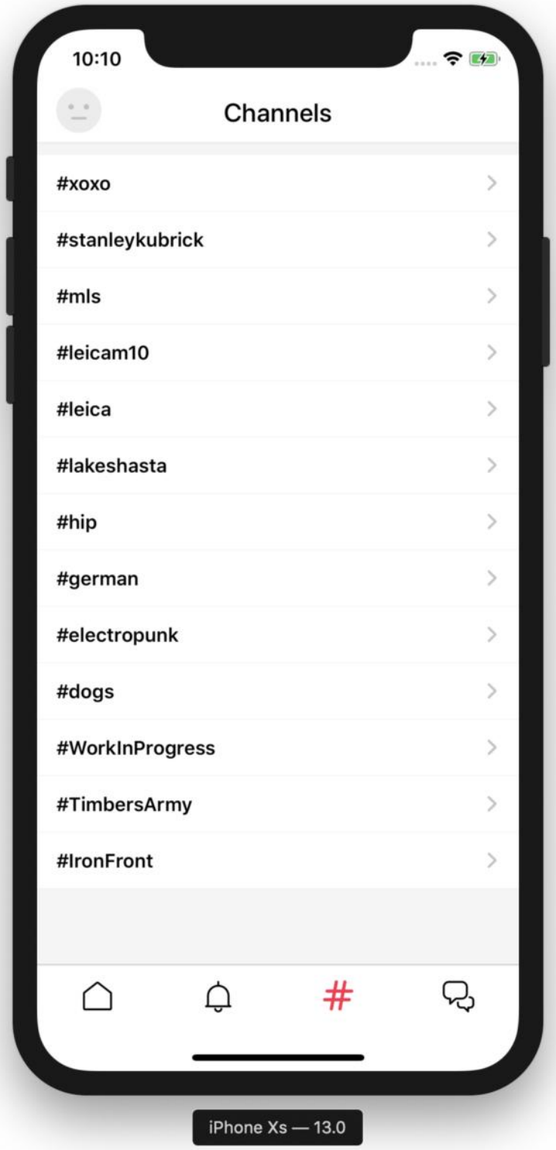


Planetary



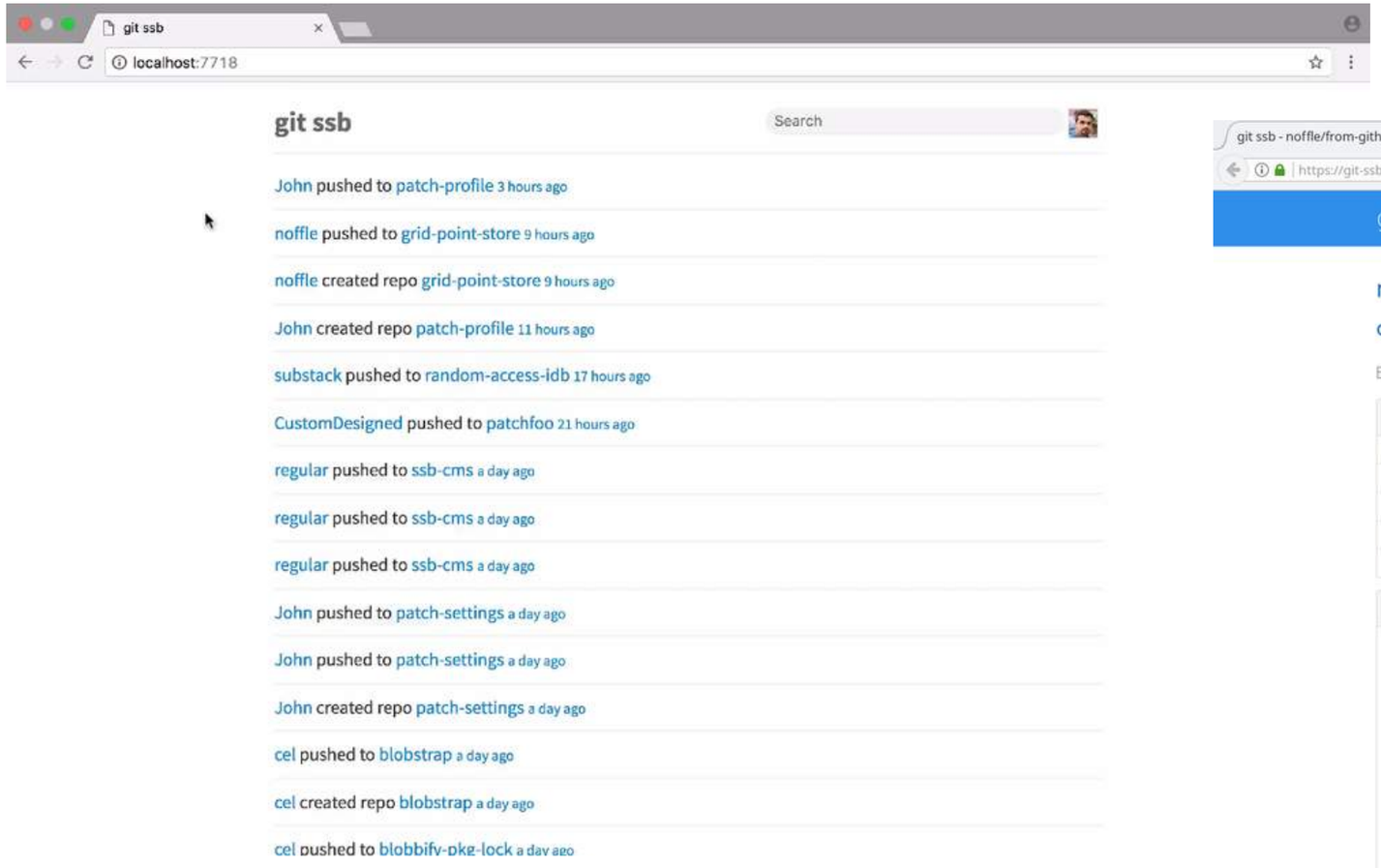
%I/I5GcW48qtCYCdF8tViB2kT3DXzvGBISh8mAdaDRFY=.sha256

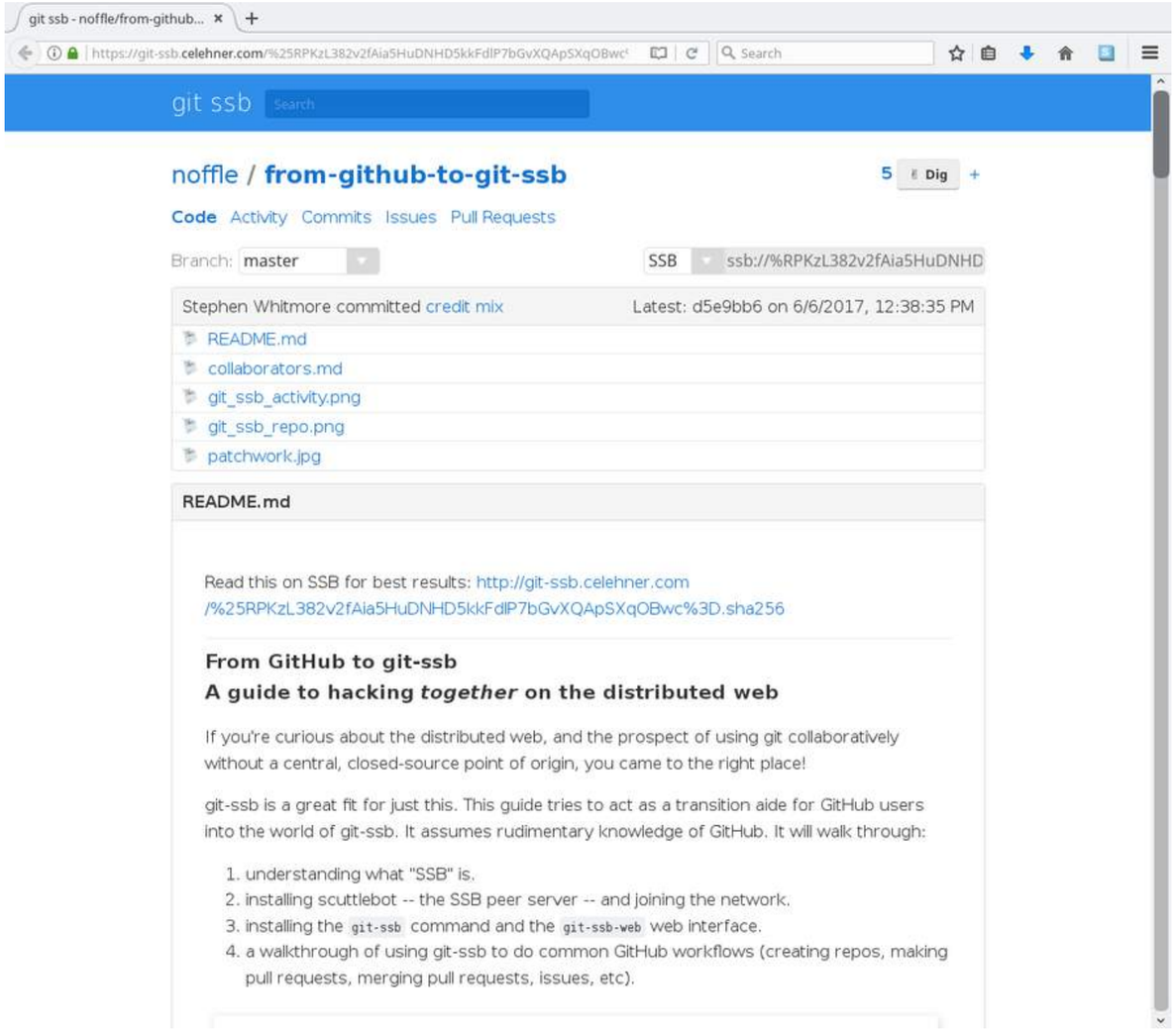




%SD3WszbEsswbRHhj6cCffcKD2ZhwF6iJ/OCKwQmLGOY=.sha256

git-ssb

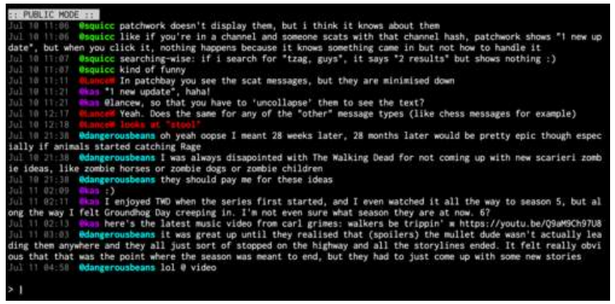

git ssb
localhost:7718
git ssb
Search
John pushed to patch-profile 3 hours ago
noffle pushed to grid-point-store 9 hours ago
noffle created repo grid-point-store 9 hours ago
John created repo patch-profile 11 hours ago
substack pushed to random-access-idb 17 hours ago
CustomDesigned pushed to patchfoo 21 hours ago
regular pushed to ssb-cms a day ago
regular pushed to ssb-cms a day ago
regular pushed to ssb-cms a day ago
John pushed to patch-settings a day ago
John pushed to patch-settings a day ago
John created repo patch-settings a day ago
cel pushed to blobstrap a day ago
cel created repo blobstrap a day ago
cel pushed to blobbify-pkg-lock a day ago


git ssb - noffle/from-github...
https://git-ssb.celehner.com/%25RPKzL382v2fAia5HuDNHD5kkFdlP7bGvXQApSXqOBwc
Search
git ssb
Search
noffle / from-github-to-git-ssb
5
Dig
Code Activity Commits Issues Pull Requests
Branch: master
SSB
ssb://%RPKzL382v2fAia5HuDNHD
Stephen Whitmore committed credit mix
Latest: d5e9bb6 on 6/6/2017, 12:38:35 PM
README.md
collaborators.md
git_ssb_activity.png
git_ssb_repo.png
patchwork.jpg
README.md
Read this on SSB for best results: http://git-ssb.celehner.com
/%25RPKzL382v2fAia5HuDNHD5kkFdlP7bGvXQApSXqOBwc%3D.sha256
From GitHub to git-ssb
A guide to hacking together on the distributed web
If you're curious about the distributed web, and the prospect of using git collaboratively without a central, closed-source point of origin, you came to the right place!
git-ssb is a great fit for just this. This guide tries to act as a transition aide for GitHub users into the world of git-ssb. It assumes rudimentary knowledge of GitHub. It will walk through:
1. understanding what "SSB" is.
2. installing scuttlebot -- the SSB peer server -- and joining the network.
3. installing the git-ssb command and the git-ssb-web web interface.
4. a walkthrough of using git-ssb to do common GitHub workflows (creating repos, making pull requests, merging pull requests, issues, etc).

More apps

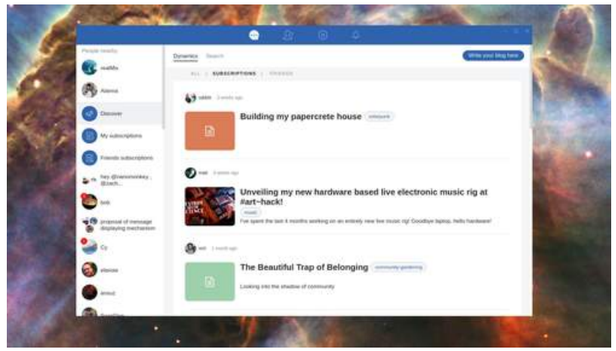
- Sunrise Social
- SurfCity
- ScuttleShell
- ssb-npm-registry
- ssb-viewer
- ssb-dns
- ssb-mutual
- yap
- mvd
- HackyArt

...



scat (ssb-chat)

`%R801Mq6vKv3lO0DueYJdP3fBzuOKLBYC5IftdvQ5N9I=.sha256`



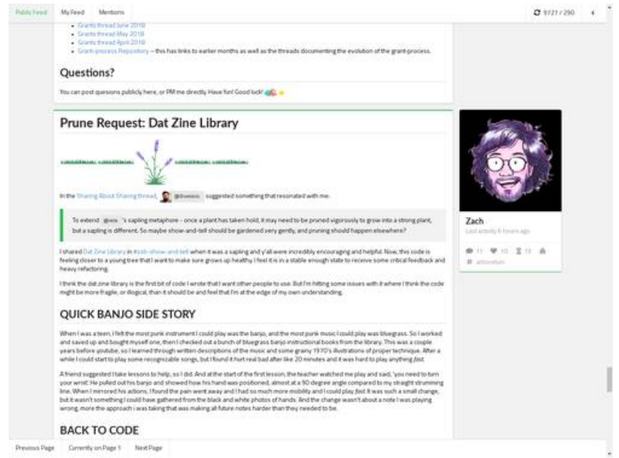
TickTack

`%ZGfik3+hbI0uKkpG0aRijRfHLNy6PwdWuuJ2BnR/Bg4=.sha256`

tale:net

Guild Lands

`%hagbVpbADoNQ6zmhovB3HhzRzrcSEkY0G0g3Dd+QMwE=.sha256`



ngx-ssb-client

Implementations

Full: Node.js, Go, Rust

Partial: C, Python, Java, Swift, Erlang

Crypto: C (libsodium)

`%6AoUPXJH+SdKjPC/FoQ/8xjQBo5WE+iOqNyrmiqRAMA=.sha256`

Funding

@elavoie %zmv8Fvm3a7qnAVBs207se5JlJXqsrqjBZwB88/Bv8DQ=.sha256 2019-04-03

Proposal HC16: Budget

Total: 85106 Euros

The percentage below have been obtained by averaging the suggestions of individual Council members, that is why we end up with cents. If someone would like to see what were the suggestions of Council members, message me and I will compile them for the records.

Budget Item	Description	Amount (Euros)	% Total	Comm. Sugg.
Core Protocol		**19236.31**	**22.60**	**22.00**
Core Development	HC10	6254.42	7.35	
Sunrise Choir	Mix Suggestion, Original Announcement	5668.43	6.66	
Pre-Partial Developer Payment	HC09	1331.83	1.56	
GitMatching	HC14	4647.31	5.46	
Emergency funds	Discussion, to Cel	1334.31	1.57	
Clients and Applications		**22486.42**	**26.42**	**27.00**
Manyverse	Announcement, Open Collective, Repository	8482.51	9.97	
Patchwork	#patchwork, Repository	5392.21	6.34	
Patchbay	#patchbay, Repository	3397.16	3.99	
Patchfox	#patchfox, Repository	1968.54	2.31	
PeachCloud	#peachcloud, Repositories	3245.99	3.81	
Community Gardening		**21269.20**	**24.99**	**25.00**
Discretionary Funds	HC03	4400.77	5.17	
Website	HC05 Power Transfer	7735.18	9.09	
Newsletter	HC06	2746.34	3.23	
Local Gossip	#the-local-gossip	1828.82	2.15	
ScuttleShip	Idea	2350.12	2.76	
For Distribution	*Mugs, stickers, camps at festivals, etc*	2207.97	2.59	
Coordination		**12755.62**	**14.99**	**15.00**
Handshake Council (Counsellors)	HC01	10047.69	11.81	
Facilitator @elavoie	Idem	2707.92	3.18	
Other		**9358.45**	**11.00**	**11.00**
General Buffer	Unexpected expenses	6003.76	7.05	
Fundraiser	HC08	3354.69	3.94	

#handshake-council

Enspiral Grant - Card

Dfinity Grant - Process

Handshake Grant - Council

Open Collectives

Scuttlecamp

%hLhNMksptjCxNi+B77DHrYEu26jDTrqgGsfb1+NvuWA=.sha256

Diversity and Inclusion

@mnin `%FuphnCqkqeH0YU2oFkLppAQ9qSK1URbXTPYozEhL7JE=.sha256` 2019-05-07

→ @mary Part of me is thinking of increasing the surface area of those wh...
↳ @Christian Bundy Mmm, I've been thinking about think and letting ideas ...
⑂ Patchwork Working Group: Skeleton Crew If you haven't already read Matt'...

SSB community Inspo

I'm also inspired by and actively learning from examples across ssb and from the organising of scuttle camp, on how diversity and inclusivity were prioritized and fostered.

Main examples being:

@Alanna

- asking and prompting us all on Who would you like to welcome to ScuttleCamp?

> In my experience, to support diversity and enrich everyone's experience, you have to go the extra mile to reach out. I believe there are some awesome people in the Scuttleverse who may not come unless we proactively invite them and offer extra support (financial or otherwise).

@mixmix

- reserved tickets for scuttle camp to support diversity + inclusion as a priority
- opening arms and offering support for those who are shy-er or new with Get to know people a little before scuttle-camp
- transparency in Diversity & Inclusion updates
  >What's not immediately visible right now is the makeup of who is attending (the gathering is ok, but it's out of sync with who is currently registered). Here's an anonymised breakdown of the humyns (NOTE we did ask people registering about e.g. gender, so these numbers may be off)

@dan hassan

- e.g actively nominating and nudging people to come
- Meetings for supporting folx to come to scuttlefest - some key valuable diverse folx came, thanks to this work!

@noffle

- actively tagging and nudging people here!

I can thank these people, for their OG initiative, support and nudges, for really making me feel invited, valued, and wanting to be sticking around here. :laughing:

Plus there's so many more examples of various people who've supported me personally, or/and had interests in diversity + personally willing to support, had engaged positively or called out toxicity, and taken the initiatives to make inclusion a priority.

That's what gives me lots of hope, that there's *an existing social layer* of willingness. The number of people actively participated in the gender FLOSS thread was astounding.

@mixmix `%co2sCeVn9L1UEvS2EIcf99vVX4E0ya6VoVUdxXWrv8w=.sha256` #scuttlecamp 2020-03-02

[moebius_journey.jpg]

Diversity / Inclusion / Accessibility > Scuttlecamp

My dream is that Scuttlecamp (and thence the future dreaming of scuttlebutt) is accessible to a wider range of people. I did a bunch of work towards this for Scuttlecamp 1, and for dweb camp.

I would like to support someone to take up this mantle for Scuttlecamp 2

The job looks like asking for money/ support, building communication / relationships with people who might benefit from scholarships, and making sure budgets line up. I am very keen to offer my experience to help someone else learn about and do this work.

FAQ

Why can't @glyph + @Luandro do this?
They've already taken on a massive role of organisation + care. They don't have capacity to safely take this on as well

Wait is this just boring admin / handing out money?
Haha, no. This is people work (with a minor side of spreadsheets). We recorded some podcasts (part 1, part 2) about the personal impact this work has had on us.
TL;DR it's incredibly rewarding, has built lasting connections, and has led me to learn a lot about myself and the world. I cannot recommend it enough as an opportunity.

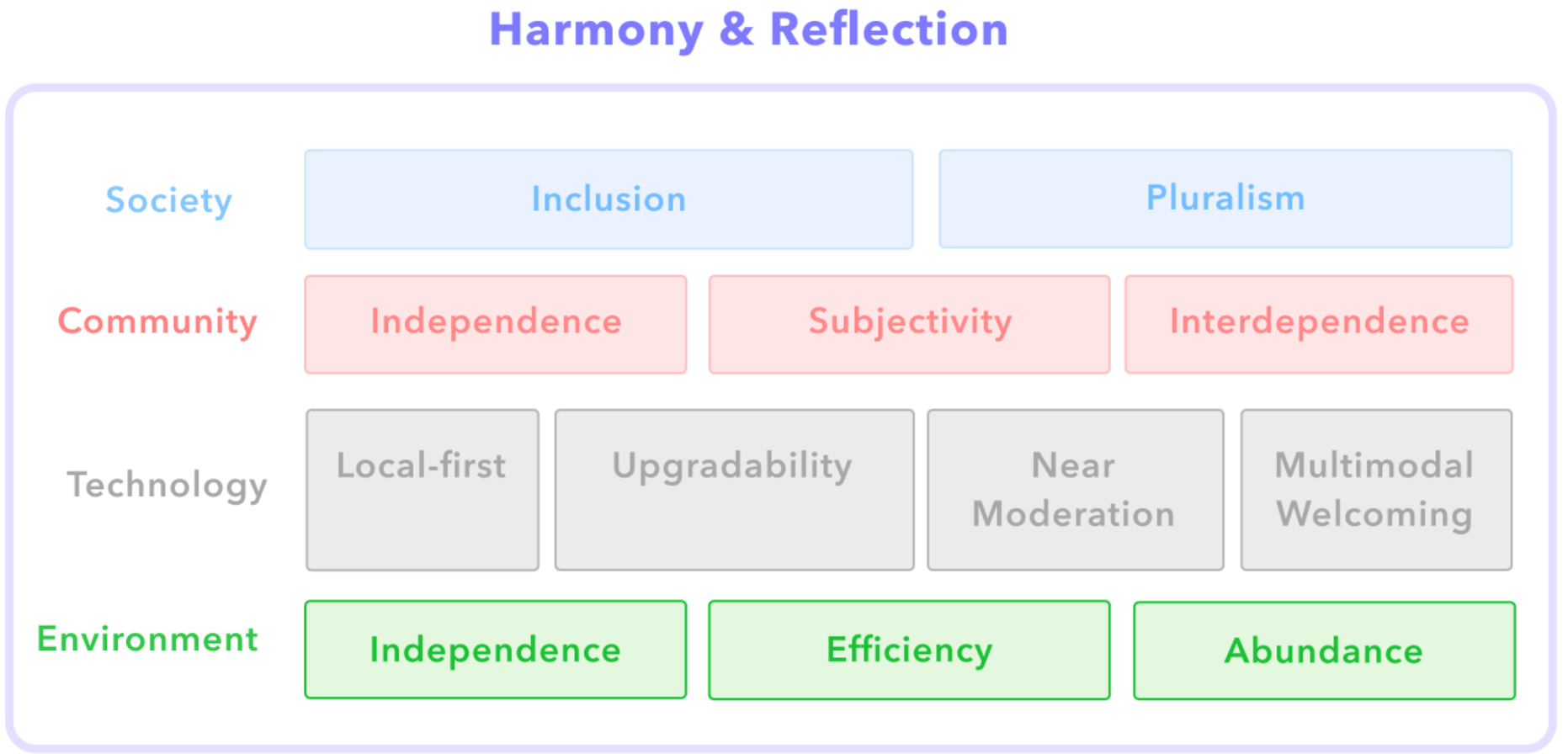
Is this hard?
It can be challenging. Remember you won't be alone, you'll have people who've done this work to help you get started, and to offer guidance and support. Hopefully we can get a couple of people to take this on so you'll be doing it as a duo

Next step

If you're keen / interested:

- :hand: Put your hand up here, and mention @glyph / me
- :question: Ask a question <3

Principles stack



%NwS0OAEqJDfbioRdwmY/IQwCdMBm6RmzhZJR682DATA=.sha256

Solarpunk

%0h9biRh7teJlwtWD6gFkzSBv8fZlXj9Qr3HrbHPkgwU=.sha256

%uKy6Lo0gTFRIt/pMXowWHU7ZbioE04kQgJ/K3I4WTPc=.sha256

%mUnUnlqp8gftsmw8oTuc7CezL/BwkoxPn6ucx3y5kSI=.sha256

%oKr5NKriL3WEoEGO+fnXnkJoFwg4JPV6Hmvun86Swx8=.sha256

%SaFh1Tea8Mqh02NRFQxoKKjWhnFAzb8JaJm0ss52Nj4=.sha256

Boats

%on6hMTGlQZ5zW+UqawDcdNmYHdReFS8GuEXvmozL64M=.sha256

Alchemist

%FdmACu6aDF4SbbVuh8B1xDSusVoUCrri1YU+KkttrAk=.sha256

Crab meet

%sV6uNHQeYiMh+GHjMCQCCFP9pigTRTq6WXpFsNKsEQw=.sha256

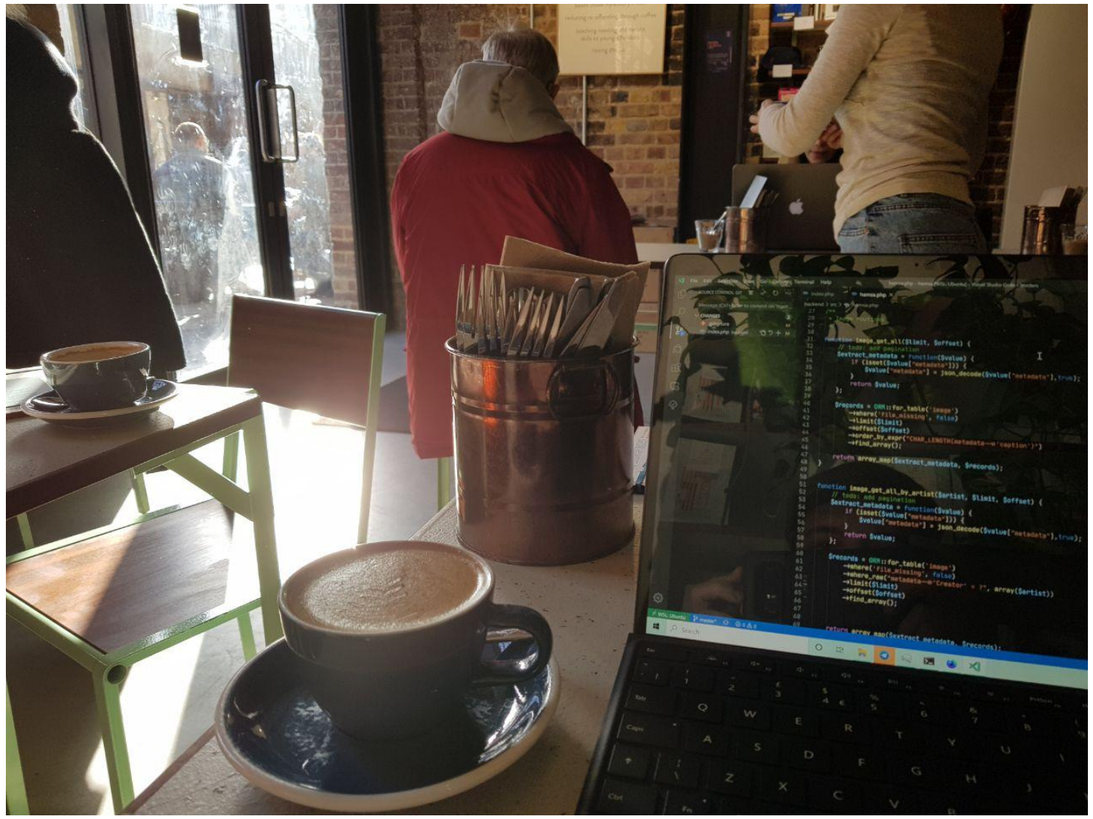
%TaKmeNL5ympWXpsyudJ1QqrXLg3ycD2GN7LIN/9QR7E=.sha256

%CsF9pDgrxzeU5oDZRQ+GmtKBVk85wDiLPcq0keLA42I=.sha256

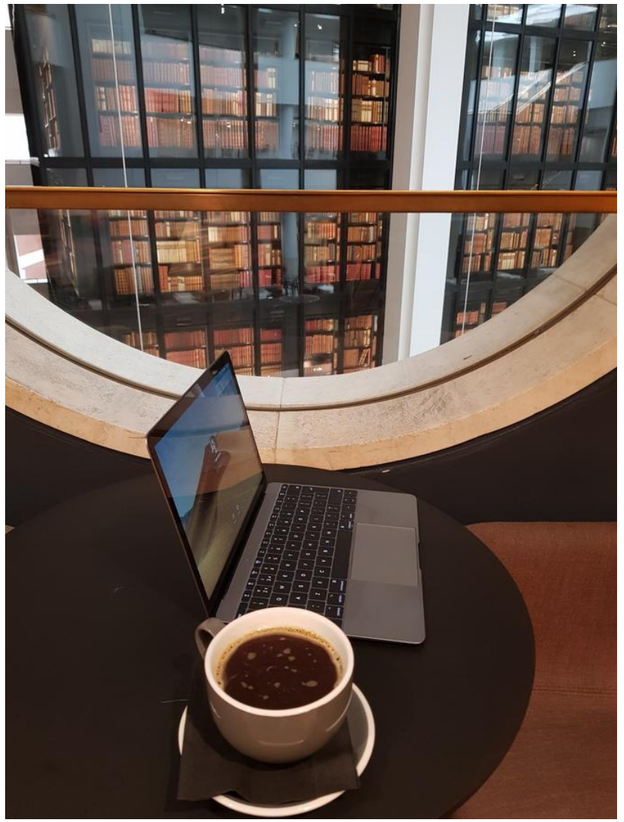
%6zsNFEriZ3ckKi7WpaoM71esgRj+mZmUVbYH3OVVbpg=.sha256

%LhJZpvqMaaORd7E1S3azBPtJBSJRSFdoeXQfVSixJmA=.sha256

#showmeyourcoffee



%7411jvQ9GqJyL2So3C5M10d61WGSSDsdyoex+0aUQ+8=.sha256

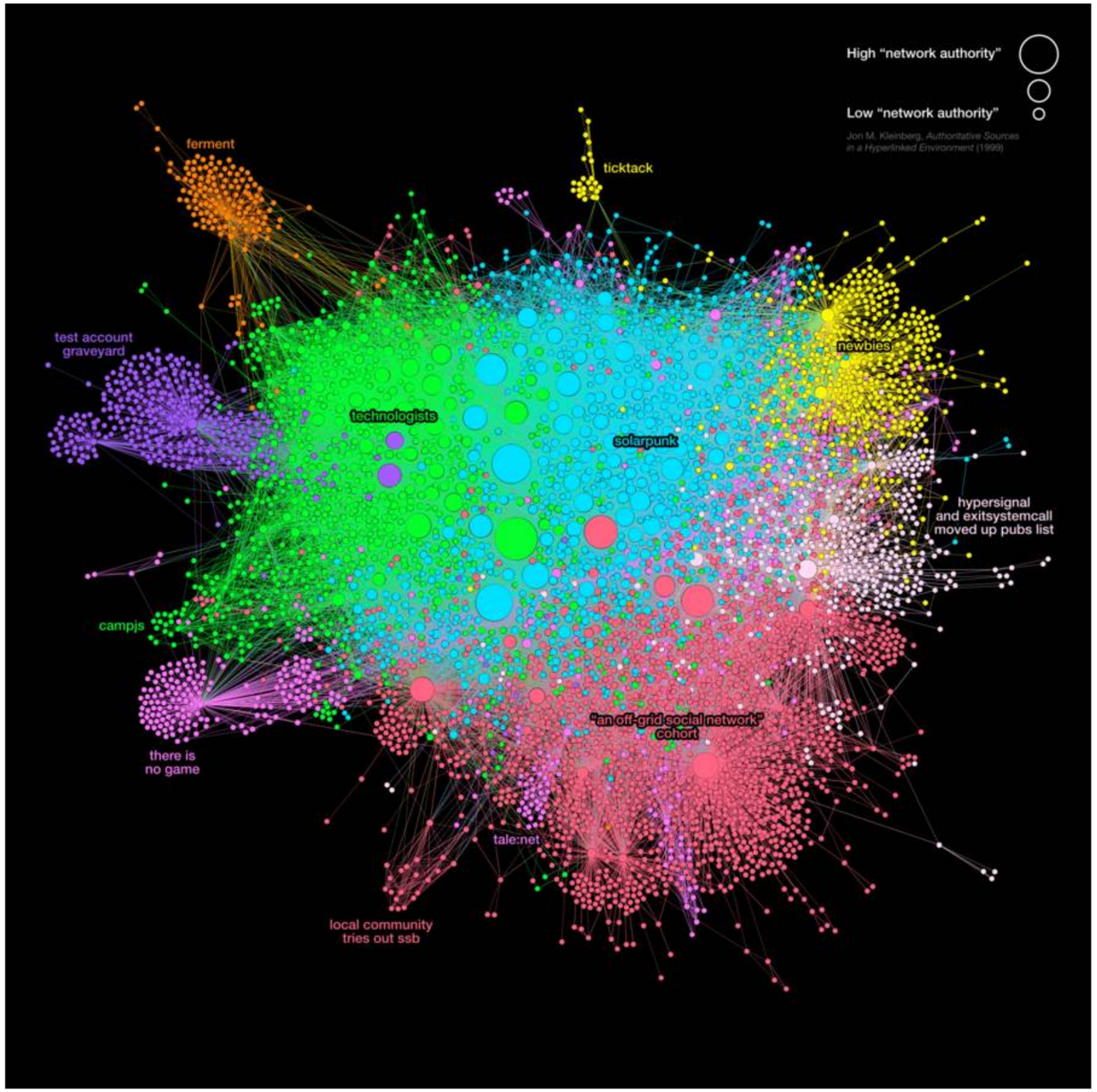


%3dkhZpXrnlKfE/sJb/A+NYBiKhUZCGnvcgbYpVWzE6k=.sha256

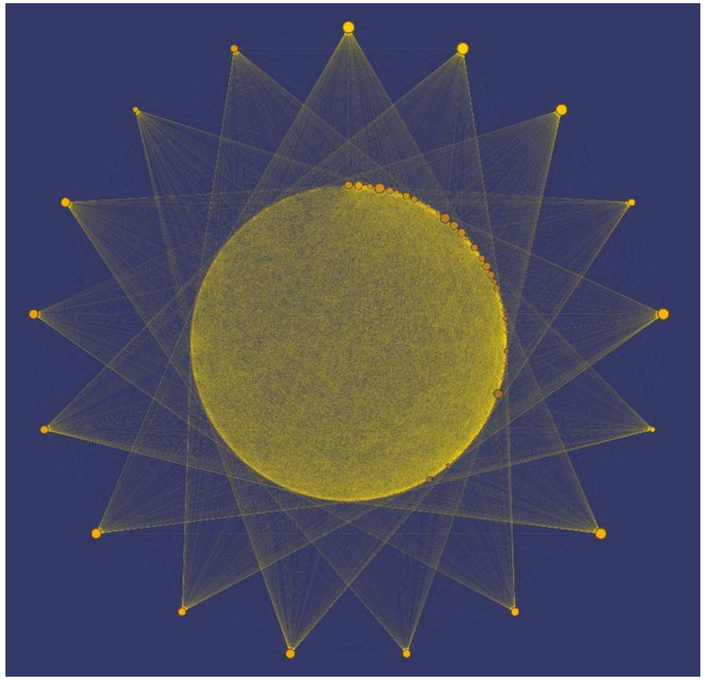
#foffee

%sHjosVsDkCpb2nH9KIxU9wr9T3wGMWtfrooQ089lY14=.sha256

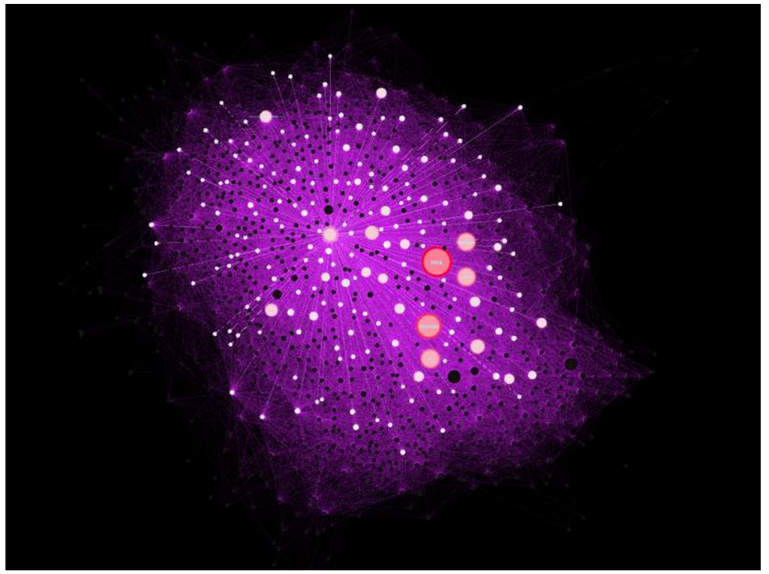
Network graphs



%utUlBefFYqHWcFQaz4r5bZUil9YtYyTik1j6d7ymVyE=.sha256

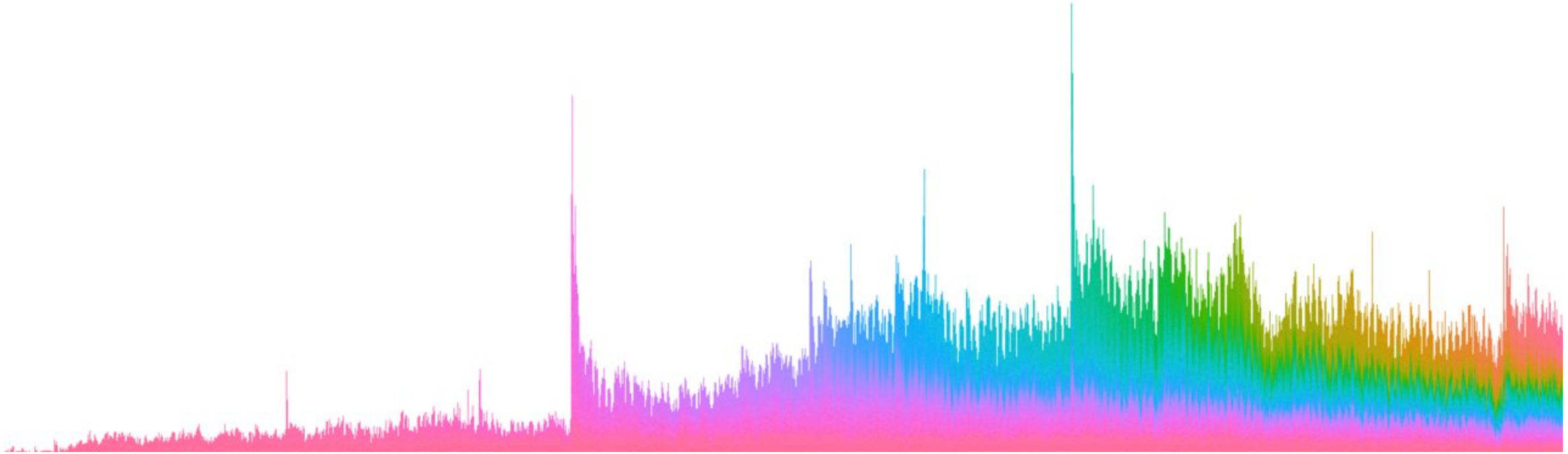


%Qy84JD3wGrXSC3/nfar9DAzWPP4aku6CIcOSr6/CC/E=.sha256



%ZCODq83xRs+sbU8YXxZ1MPlFP1u9i2wyIYAu6GCTqNo=.sha256

Network activity graph



2015-08-05 - 2020-03-05

https://ssb.celehner.com/activity/

Published research

A Broadcast-Only Communication Model Based on Replicated Append-Only Logs

Christian F. Tschudin
University of Basel, Switzerland
christian.tschudin@unibas.ch

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

This note is about the interplay between a data structure, the append-only log, and a broadcasting communication abstraction that seems to induce it. We identified real-world systems which have started to exploit this configuration and highlight its desirable properties. Networking research should take note of this development and adjust its research agenda accordingly.

CCS CONCEPTS

• **Networks** → **Network design principles**; • **Software and its engineering** → **Publish-subscribe / event-based architectures**; • **Information systems** → *Linked lists*; • **Computer systems organization** → *Peer-to-peer architectures*; *Fault-tolerant network topologies*; • **Theory of computation** → Data structures and algorithms for data management;

KEYWORDS

Global broadcast, append-only logs, distr. systems design, trust.

1 INTRODUCTION

In August 2018 I learned about a project called Secure Scuttlebutt (SSB) – an overlay network created by D. Tarr in 2014 that is tailored for decentralized social applications. This note aims at extracting the gist of SSB, which is that networking with arbitrary data packets could be replaced by networking with coherent data structures. The question is "what are suitable data structures" and what would corresponding networking primitives look like. SSB gives a very radical answer to this, although in retrospect SSB's choice seems natural if one starts from first principles.

In the analog world, propagation of perturbations is the basis of information dissemination: Depending on the physical constraints, a *wavefront* is carrying information omnidirectionally (wireless, water surface) or directionally (in a wire or fiber). However, in computer networks, the unidirectional style has become the dominant and default communication model on which today's network architectures are based, starting from the fact that data packets typically have a source *and* a destination, so do circuits, or we observe wordings like "wireless link" and note that routing algorithms are graph-based, hence link-oriented, without exception.

In a broadcast-centric world, **there is no destination**, just a source; wireless transmission is not a link but naturally implements broadcast; routing is not required because perturbations just propagate infinitely far if not blocked. We should ask ourselves whether broadcast wouldn't be the better *base level* abstraction on which to build other communication services, turning point-to-point communication into a minor (and often inferior) corner case.

Surprisingly, we find a tight relationship between the broadcast model and a data structure, namely append-only logs, that this note wants to highlight. We believe that looking at communication problems from a data structure point of view is an important step towards a better understanding of reliable, synchronized, secure and privacy-preserving operations of distributed applications.

We start by describing the properties of a particular broadcast abstraction which still is somehow close to physical phenomena (potentially making the mapping to the analog realm easier), but is abstract enough such that one can recursively refine and implement this abstraction. A suitable candidate for this abstraction are solitons i.e., information packets that propagate as a solitary wave.

2 LOCAL AND GLOBAL SOLITARY WAVES

Solitons[6] are particle waves or wave packets which travel through space without leaving any disturbance behind them. Such solitary waves have been observed from biology to cosmology and have important applications in communications, for example fiber optics.

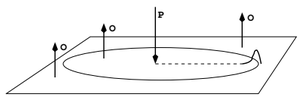


Figure 1: A perturbation triggered by a station *P* propagates as a solitary wave to all observers *O*.

The purpose of this note is to take solitons *as the inspiration* for a communication model whose operation is best described in terms of an initial perturbation that propagates in a wave-like form, in all directions. Such a broadcast model shall obey the following three properties:

(1) Each perturbation source has a globally unique identifier that is carried with each perturbation it triggers.
(2) A perturbation and its value eventually reaches all anonymous observers.
(3) All observers sense subsequent perturbations coming from a specific source in the same order.

Using middleware language, the communication primitive having above properties can also be described as reliable, ordered broadcast

%aclRXl3q1cnoOyfVuDRaTAKPESv4J6QLDE4B5M2vDaM=.sha256

Secure Scuttlebutt: An Identity-Centric Protocol for Subjective and Decentralized Applications

Dominic Tarr
ssb:@EMovhfIrFk4NihAKnRNhrf
RaqIhBv1Wj8pTxJNgvCCY=.ed25519

Erick Lavoie
McGill University, Montreal, Canada
erick.lavoie@mail.mcgill.ca

Aljoscha Meyer
TU Berlin, Germany
aljoscha.t.meyer@campus.tu-berlin.de

Christian Tschudin
University of Basel, Switzerland
christian.tschudin@unibas.ch

ABSTRACT

Secure Scuttlebutt (SSB) is a novel peer-to-peer event-sharing protocol and architecture for social apps. In this paper we describe SSB's features, its operations as well as the rationale behind the design. We also provide a comparison with Named Data Networking (NDN), an existing information-centric networking architecture, to motivate a larger exploration of the design space for information-centric networking primitives by formulating an identity-centric approach. We finally discuss SSB's limitations and evolution opportunities.

CCS CONCEPTS

• **Networks** → *Network architectures*; • **Software and its engineering** → *Publish-subscribe / event-based architectures*; • **Computer systems organization** → *Peer-to-peer architectures*; • **Information systems** → *Linked lists.*

KEYWORDS

Secure Scuttlebutt, Information-Centric Networking, Push vs. Pull

ACM Reference Format:
Dominic Tarr, Erick Lavoie, Aljoscha Meyer, and Christian Tschudin. 2019. Secure Scuttlebutt: An Identity-Centric Protocol for Subjective and Decentralized Applications . In *6th ACM Conference on Information-Centric Networking (ICN '19), September 24–26, 2019, Macao, China.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3357150.3357396

1 INTRODUCTION

A simple conceptual architecture for community applications consists of a global data pool to which every person can contribute and where every person can tap into the shared data – data sharing being the purpose of such applications. This model still is valid if one adds access control to the picture, either tied to the data (encryption giving access to content only to entitled holders of the decryption keys) or encrypting data in transit (login and TLS). Facebook and other centrally organized social app service providers fit well under this global data pool model but have been strongly criticized for abusing their central provisioning position. The "decentralized web movement" [18] is the most visible technical response to this critique, pointing out implementation alternatives.

One of these alternatives is a project called Secure Scuttlebutt (SSB) that started in 2014. After several iterations of protocol design and implementation, SSB has become a stable service for over 10,000 users offering them rich media community applications with strong cryptographic protection (end-to-end encryption and metadata privacy) and running in pure peer-to-peer mode.

ICN '19, September 24–26, 2019, Macao, China

ACM ISBN 978-1-4503-6970-1/19/09.
https://doi.org/10.1145/3357150.3357396

Selective Complete Log Replication

SSB relies on the core insight that each participant is only interested in a *subset* of the global data pool, thus it is feasible to locally store all the data a participant is interested in. To partition the data pool, all data is associated with the *identity* that produced it. Participants select their slice of the data pool by specifying the set of identities whose data they care about. This creates a "social graph" along whose edges data flows (Figure 1). Even as the overall system scales, the amount of data any single peer is interested in and thus needs to handle stays roughly the same.

Each participant can publish data to their single-writer, append-only log. This choice of data structure allows efficient replication and verification of the integrity of received data. Replicating these larger slices of the data pool comes with an unusual set of trade-offs, discussed throughout the paper. As it turns out, replicated logs form a solid foundation for implementing many classes of applications.

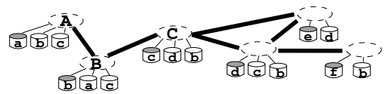


Figure 1: SSB's "Internet of Identities" – Users A, B and C replicate logs (*a, b, …*) based on whom they follow: *C* does not follow *A*, hence has no log *a*. *A* and *B* follow each other such that when *A* follows *C*, *A* will get *C's* log *c* via *B:* new content is pushed directly if possible and through intermediary friends if necessary.

Subjective Reader

Because replication in SSB is selective and driven by a peer's social graph, different end devices will have access to different sets of log

%ky3+/h0MFsevtRpCBYHAq7l7N7JF2FgI21Sw0BSt9Gc=.sha256

Other initiatives

Faerie rings

@Powersource %31xETMTN1+hiPv52HgNmZwF6GyJYJz9HQ8tHHTwidp0=.sha256 2019-03-04

Faerie ring mitosis

[lots of rings]

Coming out of scuttle camp and with the last faerie ring v 1 coming up, it feels like the right time to sprout new rings.

Brief description

For those unfamiliar with the concept, here's a brief description of the concept:

- max 8 people
- a call once a fortnight for 2-3 months
- everyone has a turn checking in
  - how's it going?
- there's a host (who changes)
  - opens & closes the circle
  - makes sure everyone gets to speak

the rest of the format can grow to fit that group, perhaps things like

- a focus question
- asks / offers : do you have something you'd like to get help with / to offer ?
- freestyle

This description is taken from mix's original invocation where you also can read more about it.

Organizing this

Seed hosts

While the host changes with time, it's good if the first host is someone who is in the current faerie ring or has experience with similar things, to get the new ring off to a good start. Tagging the people in the current ring, but others could step up.

@Matt McKegg @Mikey @Christian Bundy @Piet @noffle @mix @Zach! @Dominic

Faeries

That's the official term now ok :P Basically other people interested in being in the rings but who aren't comfortable being the initial host. Tagging people who've expressed interest/went to the microsolidarity session at scuttle camp (sorry if I miss someone!)/other people that I think would be interested.

@Zenna @mu @Emmi Bevensee @Alanna @Teq @Songyi @cel @cblgh @Louise @Isak @happy0 @andrestaltz @KawaiiPunk

If you're interested in either role please comment below. If you could be up for being a seed host please indicate so.

:sparkles::sparkles::sparkles:

#faerie-rings #microsolidarity

Community Infrastructure Learning Group

@punkmonk %L9X+oakILqFsbDGCHv7mXmf52vzxaTX/iNMA4GTnweQ=.sha256 2019-09-13

Community Infra Together

Context

Recently, I started a fellowship for an organization that works with a local school district to build out capacity for the school to create makerspaces within their own schools, and to provide STEM/STEAM related projects tied to core state educational standards.

Additionally, the organization is looking for new ways to coordinate with schools, the community, local government, internally etc. I was initially invited to apply for this fellowship due to my interests in p2p tech and hopefully to be able to apply some of these ideas.

With that in mind, I'm very interested in learning more about running the needed p2p infrastructure effectively, reliably and safely. I'm familiar with running these things for my own use, but not when providing tools for a larger community.

Is anyone interested in assembling a learning group focused on running various p2p services? Initially, starting with an ssb pub and working onto more advanced topics like safely running a cjdns public node.

If anyone, no matter what skill level, is interested in being part of a **p2p infra learning group** it would be great to go through it together and help each other learn. This would also provide real value to the scuttlebutt community (and others) as more maintained ssb pubs, cjdns nodes, dat seeders etc are added to the ecosystem at large.

This is an opportunity to develop a p2p blueprint that could be handed to any interested community, with best practices laid out for them to initially follow. As we learn, those learnings should be documented and given back to the commons for further iteration.

Drop a line in this thread if interested and a gathering can be created later for further discussion.

Here is a cryptpad with some initial ideas: https://cryptpad.fr/code/#/2/code/edit/Uq-w6COLPAe9Meaw3y81zzTr/

those interested should feel free to add or edit

crypto-mutual-aid

@punkmonk %VeBBSKCWC6F8/Q6nIVDhS6HoFR1cekjPMh+qKLGiOpA=.sha256 2020-01-30

cryptaid

i'd like to try a small mutual aid experiment. this is an idea that has always been intriguing having grown up in an area where mutual aid networks are fairly prevalent among a certain demographic. the idea always resonated with me especially during times of being un-insured/un-insurable.

the experiment could focus on alleviating small, unplanned hiccups for participants ie. "i need a weekly bus pass to go to job interviews" or "i have to work but i need a sitter today!" or "i'm unexpectedly sick and need help paying the $100 copay" orrr "i had an unplanned car bill, and need a few bucks for groceries till payday"

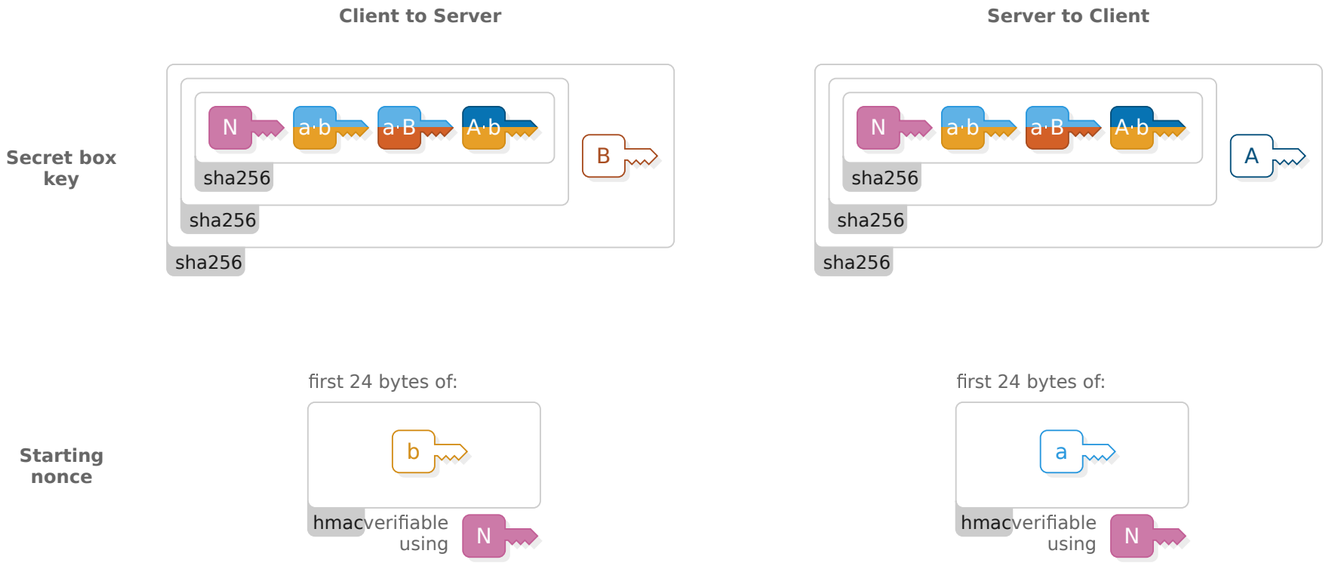
this could require fairly low monthly input from members but might be able to do some good down the line alleviating stressful financial situations. i don't know exactly what the right amount should be for a monthly payment, but something south of $50 USD is what i had in mind and i'd propose using cryptocurrency since it might make experimentation a bit easier than a bank.

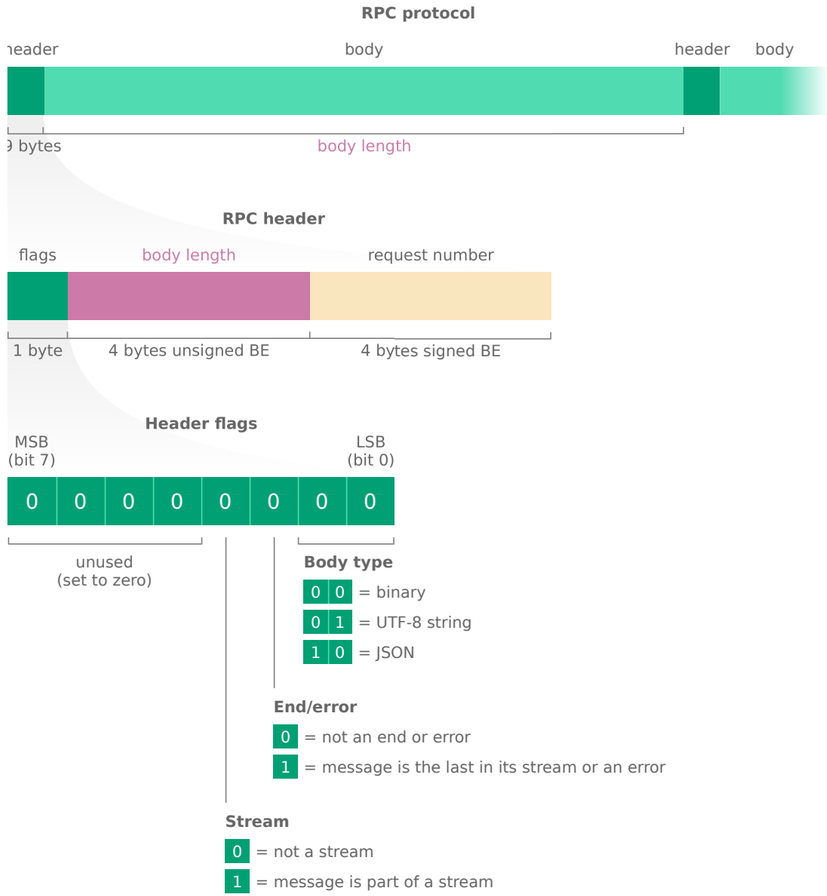
#cryptocurrency #mutualaid

Protocol Guide

https://ssbc.github.io/scuttlebutt-protocol-guide/

`%YkIPtiAe4VHc4YjeYJ1MPMsdxE2soNyQjzgqj+4iXNo=.sha256`





Future

- Organizations
- Grants
- Planetary
- Deletion, partial replication

Private groups

@mixmix %QXdVyQu8Cu3lD+DGQhiTUUr21DvtyFNkZV8oHunzd2w=.sha256 #ssb-show-and-tell 2020-02-29

ssb-private2

this module isn't done, and is standing on a bunch of pull-requests which need merging, and specs which need finalising .. but tongiht I just got a very sweet test passing.

Check out how easy this code is going to be:

```
server.private2.group.create('waynes world', (err, data) => {
  const { groupId } = data

  const content = {
    type: 'announce',
    text: 'summer has arrived in wellington!',
    recps: [groupId]
  }

  server.publish(content, (err, msg) => {
    server.get({ id: msg.key, private: true, meta: true }, (err, msg) => {
      console.log(msg.value.content)
      // {
      //   type: 'announce',
      //   text: 'summer has arrived in wellington!',
      //   recps: [
'%mo4uFJ8z52wditPjmMNSuZeZZJFgM9EEO9ZQZGCVixU=.cloaked' ]
      // }

      server.close()
    })
  })
})
```

Links and further resources

https://scuttlebutt.nz/

https://scuttlebot.io/
https://ssbc.github.io/scuttlebot/
https://modules.scuttlebutt.nz/
https://opencollective.com/secure-scuttlebutt-consortium/
https://en.wikipedia.org/wiki/Secure_Scuttlebutt

Thanks

https://ssb.celehner.com/lp2020.pdf

%gtKiSW6EslgXh/DphgmyMNXjVr9f7D/8Ro6wPyBxfSI=.sha256