

Scattered, Smothered, Recovered

Secure Scuttlebutt's dev UX, and visions for the future

Eileen Wagner // October 7, 2021

Why we're here

- NGI Pointer 2021
- Zenna <> Eileen on UX topics
- Questions:
 - How is meta feeds being received?
 - What are remaining pain points on SSB's stack?
 - Where do we go from here?

What's been done

- September 2021
- 6 interviews with SSB devs
- 2 focus groups (8K Demo)
- Caveats:
 - Not a lot of time for people to test meta feeds!
 - Big questions → big answers!
 - (I am not a developer)

Who

person with
family problems
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

Data

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

Observations

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

person with
high anxiety
level

Wish List

person with
high anxiety
level

person with
high anxiety
level

Today

- Insights:
 - Developer UX
 - Development Decisions
 - Community and Culture
- Recommendations
- Open Questions

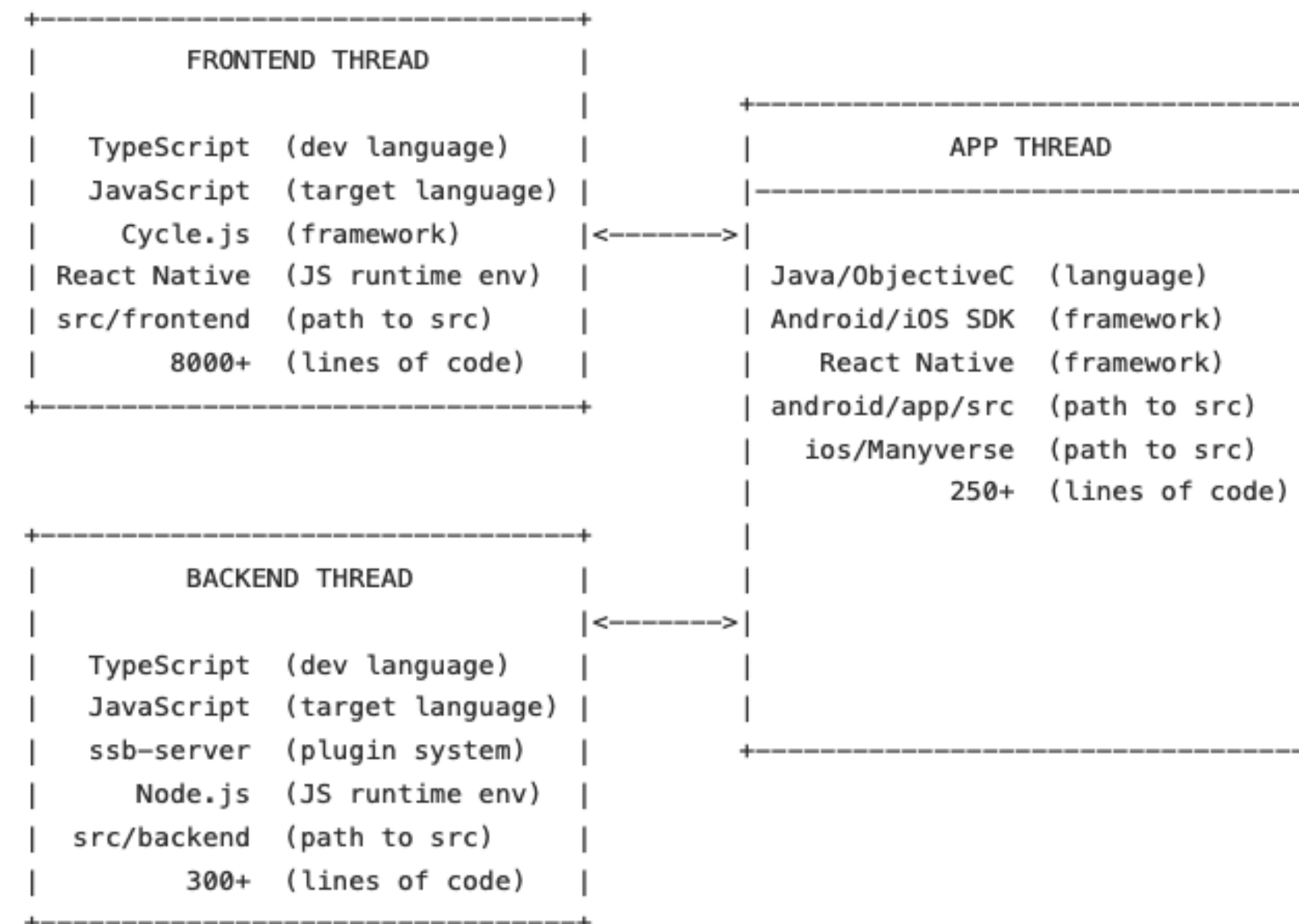
Insights

Developer UX

SSB is complex

Multiple languages, custom code, exotic frameworks, new concepts

- 😊 React Native (frontend)
- 😓 Node.js Mobile (backend)
- 😋 Scuttlebutt (backend and frontend)
- 🧐 TypeScript (backend and frontend)
- 😁 Cycle.js / xstream (frontend)
- 🤖 Pull streams (backend and frontend)



“SSB is not one new piece of technology, but five distinct and exciting ones, all in a large pile...”

“I tried to learn, but [the custom frontend framework] is ungraspable.”

“Can we stick with boring, mainstream frameworks?”

Developer support

Missing guidelines and documentation

- Guidelines for: type safety, linting, testability
- No comments, no documentation
- Single-letter variables?!?
- Closures
- Renamed SSB npm libraries :(
- Gitlab vs. Github
- Modules upon modules make it hard to understand (and to maintain)
- Demo apps often just don't work!

“Why aren't there contributor guidelines?”

“[Closures] are elegant but not accessible.”

“Can we agree on better naming (rather than just A,B,C,D)?”

Plugins are not clear

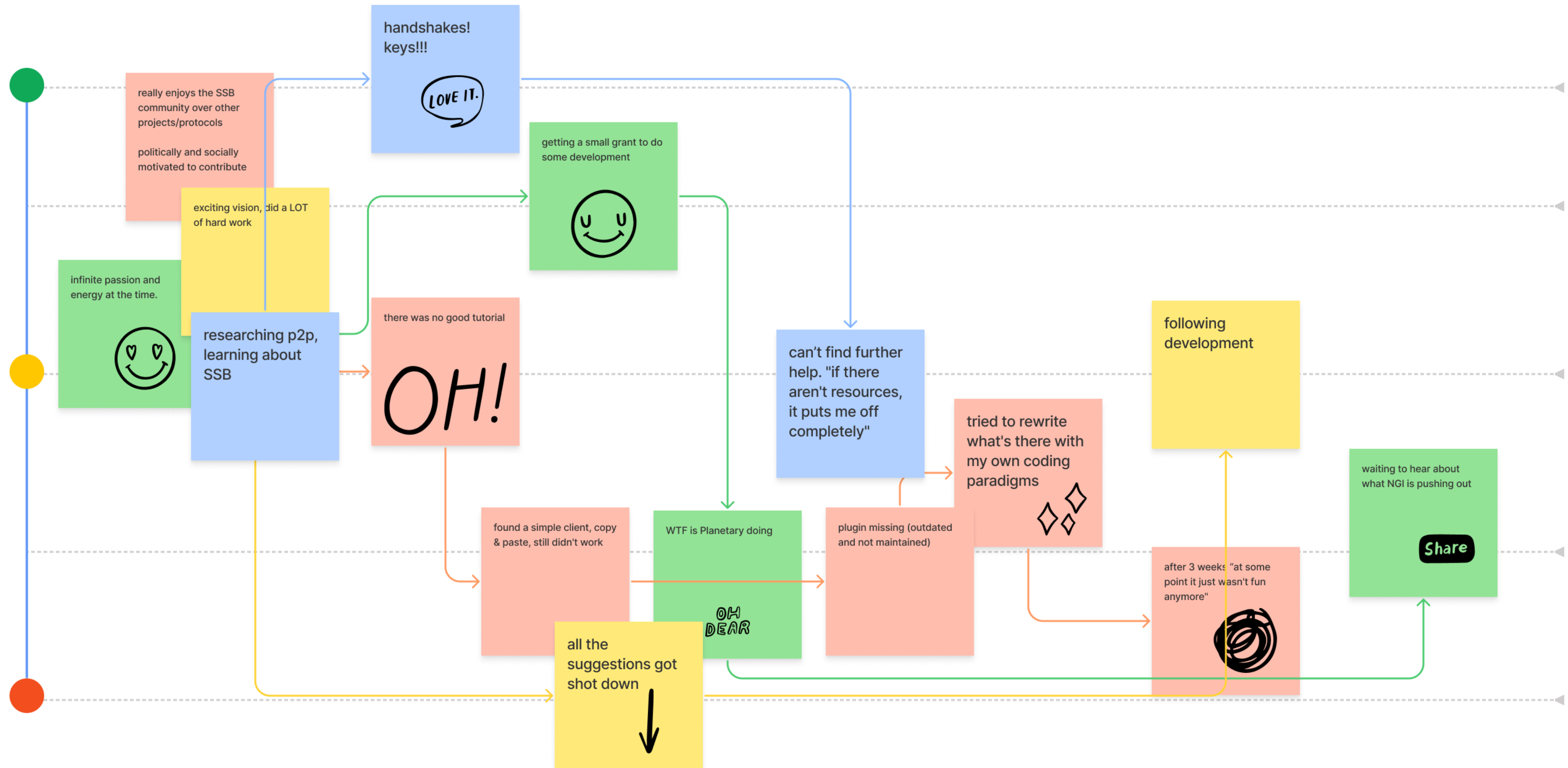
When (and when not) to use them

- Most things just won't work without a plugin. So why are they plugins?
- Too many are hard to manage; people dislike the modularity and prefer things to work “out of the box”
- Some plugins also have dependencies with each other, so clearly they are not just modules

“What is even a plugin?”

“It's unclear what the defaults should be.”

User journeys



8K Demo

(Focus group comments)

- Everyone thought the 8K demo was super cool!
- Very impressed with the new features like partial replication, group encryption <3
- Mobile use cases here we come!
- People think this is a great and healthy direction for SSB
- The protocol guide is well-written but doesn't talk about other parts of SSB
- Questions around data portability, conflict resolution for multiple devices, reference-ability

“[Partial replication] is fundamental in pushing the SSB ecosystem forward!”

“This doesn’t feel like SSB anymore.”

“This is awesome, I will definite use it.”

Development Decisions

Growing the protocol

Strategic goals diverge

- Some core developers want to build things perfectly, while others want to build into an imperfect world

“Can we just apply a boring industry standard?”

mainstream
adoption



getting things
EXACTLY RIGHT

My strong opinion is to reject strong opinions. `deject` is a strategy to deopinionate software. It should be easy to change any particular opinion.

<https://github.com/deject/deject>

Ideas >> implementation

Increase in bus factor and burnouts

- Reliance on a few (highly motivated, highly skilled) individuals who put out a TON of work
- Often focus on ideas and proofs of concept, less so on “boring” code
- People are drawn to SSB primarily by technical problems, so app/client-building comes as an after-thought

“There is a certain madness to SSB development. Everyone is always trying to prove something. It’s generative, creative, impressive, but also intense and stressful.”

“If you have one rock star main dev who cannot be replaced, then it’s their way or the highway...”

“Burnout is real. We lost a lot of good people.”

Funding is erratic

Roadmap can be steered by money

- Funding creates hubs and momentum
- (NGI Pointer = great!)
- BUT, general funding decisions often feel arbitrary
- Accountability structures? Decision-making?
- The work is not being centrally coordinated (intentionally!)

“What if another [VC-funded company] came forward? Do we all start working on that then?”

“Can we have more of NGI-like funding please?”

“There just aren’t enough resources to equally distribute among the contributors.”

A p2p social network?

AKA “we should have never come down from the trees”

- SSB makes some implicit assumptions about what to display in a social network, but this is not how “normal” social networks work
- E.g. “rooms” vs. “private group” vs. ?

conventional,
mirroring social
interactions



unconventional,
something new
and weird (and
potentially
harmful)

“[SSB] is fundamentally a social media protocol.”

“I don’t want to see everything from friends of friends.”

Community and Culture

Professionalising

Debate over whether and how to organise

- It's clear to most that coordination and organisation is needed to pull in the same direction
- People also understand and appreciate the need to distribute decision-making
- This creates a tension between wanting to unite behind certain people and goals, and the believe in decentralisation

“Just because we are organising, doesn't mean we have to start wearing suits!”

“Being organised is being dangerous—if the left is better organised, we could have real power.”

Maintainers vs. community

Responding to pressure

- There is a lot of pressure to respond to particular asks from the community
- This makes the core maintainers reactive rather than proactive

“A few voices in the community will scream about keeping some client up. But you can't fix it with magic!”

Good foundation

SSB offers more than code

- People come in for community, solidarity, political alignment, fun culture
- The prominence of cultural identity can also be divisive and lead to fewer (and more committed) community members

“I didn’t like the stack at first, but the people were much nicer than the ones at [other p2p protocol]!”

“SSB has become more punk over the past few years—that’s okay, it’s just not me!”

Holding it “together”

A project, a scene, a community?

- There is a scene (!) behind the protocol, and people strongly identify with the values and culture
- A lot of people expressed anxiety over the future of SSB seeing how fragmented it has become (Oasis vs Patchwork vs Rooms vs Planetary)

“We have to save SSB now.”

“Meta feeds is wonderful. But is it coming too late?”

“Like other scenes, SSB won’t die because the tech malfunctions; it’ll die if the scene gets broken up.”

A large, solid gray triangle is positioned on the left side of the slide, pointing towards the bottom right corner.

Recommendations

Think onboarding experience

- **Define better defaults.**
 - Have opinions about how SSB should be used.
 - Develop and maintain starter projects / demo apps.
- **Focus on what's different, and make everything else boring.**
 - Help newcomers by using familiar and mainstream (frontend) frameworks. (P2P is new and complex enough.)
 - Figure out what parts need to be custom, what parts need to be boring/maintainable.

Follow good dev UX

- **Developers also deserve good UX.**
 - Comment your code, give people usable names and explanations.
 - Use conventions and tried-and-true processes.
- **Help people out.**
 - Give people guidance (tutorial) and documentation (glossary) for explore and look-up flows.
 - Centralise your resources. (Pretty please.)

Cut yourselves some slack!

- **Put on your oxygen mask first.**
 - You've been listening and responding to community needs for *years* now. It's time to ask: what do you want, where does this journey take you?
- **Keep calm and carry on.**
 - You are not a research lab. You are not a startup. You are not a nonprofit.
 - Smooth sailing > the next proof of concept.
 - Don't ship everything yourself. Rely on existing protocols for transport, encryption, etc.

Open up a conversation

- **Discuss how (whether) you want to organise differently.**
 - How are decisions made? Who makes them?
 - How transparent and participatory should the process be?
 - When and how to distribute funding?
 - What changes do you need to make to your governance?
- **It's okay to lead the community :o)**
 - Consider introducing accountability structures.

SSB isn't one thing

- **Separate out:** the protocol, the (political) project, the maintainers/contributors, the “scene”, the larger community, the users
- ... and **see what SSB is to them.** Who/what would you prioritise? How can you best serve them?
- Agree on **principles** as the foundation for any of these groups.

Open Questions

Mainstream vs. quirky
Inward-looking vs. outward-looking
Protocol vs. protocol + client

What's next?

- Validate these findings with a survey?
- “Internal” strategy workshop/retreat
- Community forum (AKA SvendSolar but bigger)
- Documentation sprint

Danke!

Eileen Wagner // hello@bumble.blue