



The Call for an Open Source Social Network

2009

2 min read • 542 words

Lately, I've been tossing some ideas around that I feel would benefit the Social Web as a whole. It's been going through some rough times lately, and I think it's time for a change.

My first idea was to create a site that was rather decentralized, allowing all of your content to exist on other sites, but still allowing for you to interact without locking a user in. As it turns out, this site already exists. It is called [FriendFeed](#). I love FriendFeed. About a week after I decided that I wanted to get into it, Facebook decided to purchase it. How sad. They claim that it will still be up and running, but we'll see if that proves to be the case.

Think about this now: **Why is that even an option?** Social networking is all about community and building tribes - So why do we need to have an organization in charge of our chosen communication platform?

Here is my proposal: Create a community-driven, community-developed, and community-controlled social networking site that is truly open source.

The community could take care of everything from feature development to content control. No random shutting down, buy-outs, or merges. No more random change of Privacy Policies or Content Ownership battles. No more worries. The only group of people who would be benefitted would be the community itself, not some company.

The Open Web of Flow

Step One: Solid Platform Choice

The platform choice is the most important. You've seen what happens when the wrong tool is chosen for the job: look at Twitter: Bad Planning. We're all familiar with the Fail Whale but we shouldn't. At all. So what do we use? The answer is obvious: Django on a LAMP Stack (Linux + Apache + MySQL + Python). We can all agree that Python is freaking amazing. And it's certainly not going anywhere any time soon. I think Google has proven time-and time again that Python is the language for just about any job.

Step Two: Basic Information Architecture

We need to decide how the whole system will work. FriendFeed has an excellent system in place. Let's use it. Users can tie everything in from all of their other websites and stream it on their profile, and display it all on one page.

Everything's streamlined, commentable, hookable, and readily accessible. Let's mix that with a FriendFeed-style activity stream.

Step Three: Sustainability, Audience, and Accessibility

How will we pay for it? How will we get people to contribute? How will we get people to use it? Answer: Twitter is getting old and it's getting old fast. It's time for something new. Let's blow them away and they will ALL hop on board. Allow for easy external-account migration and account creation and we'll be golden.

Step Four: Technical Planning and Engineering

This is all the stuff end users don't have to worry about. Performance. Design. System Administration. Database Engineering. The geek stuff. I mean come on guys, how awesome would it be to be able to make a commit to the Twitter live SVN Branch? Epically awesome.

Step Five: Community + Collaboration

Once the community gets going, there will be no stopping it.

In conclusion, think about what we have to lose? Are you with me?

Generated from kennethreitz.org • 2026