Social Networking Platform From Scratch — Introduction

This is the first in what I hope to make into a series of posts inspired by one of my more controversial opinions: social networking platforms could be less terrible.

I have vague ideas in that direction but I don't really think that I have the desire to make building such a platform my day job, or even my primary non-work hobby. I really just want to try putting something together, and hopefully writing about that process will help me with turning the various nebulous and disorganized thoughts in my head into actual working code.

Some of the posts in this series will be technical (describing how I go about implementing a given feature in this architecture), and presumably some will also be more design-based (filling in more details about each feature, and the reasoning behind it). For now, we'll start with an overview of the features I'd like to implement, and the tools I want to use to make it happen.

Overview

The network will be made of *profiles*: the faces individuals present to the folks they connect with. (These are pretty much exactly the same concept as an *Aspect* in diaspora*.) Each profile belongs to an *account*, which is uniquely associated with an email address (and ideally one-to-one with a given individual, but there isn't really a practical way to enforce that).

Another controversial opinion: Non-Consensual Advertising Is Bad. "Consensual" in this context means that the user is not actually **trying** to shop for something, so this is a pretty broad statement which covers almost all of the advertising methods we see today: TV commercials, sponsored content, billboards, ...

However, I don't think that advertising is inherently bad in the right context. Doofus Rick says it best:

> Advertising? Wow! So…people need help figuring out what to buy, and you help 'em?

I won't lie and say that I don't want to live in a consumerist culture, and I recognize the value of making sure that consumers know what products are available. I just hate the idea that Person A can pay Person B to **put information of their choosing into my brain without my having anything to say about it**.

This project is not about that problem[1]; this is just an aside to emphasize an explicit non-feature. It also makes the privacy problem easy: the platform itself has zero interest in the content that people put in, so there is no barrier to making the strong assertion that their data belongs to them unconditionally. The only purpose is to allow individuals to communicate with the people they connect with, and nothing more.

Features

I am going for three main features to start with:

- **Messaging**—Basic Unicode messaging, profile-to-profile or for groups of profiles. Notifications go to everyone in the messaging group, but can be muted.
- **Posts** —These come in several types, but I'll start with plain text posts. Posts can be seen by anyone connected to the profile they belong to. A given account can subscribe to notifications for posts from a particular profile.
- **Events**—Calendar events and appointments. Each event automatically gets a message group for both invitees and confirmed attendees. Each account gets a calendar view where they can see all of the events across all of their profiles. Eventually I would like to be able to synchronize with external calendaring services; ideally this should combine the social aspect of e.g. Facebook events with the practical utility of e.g. Google Calendar.

Tech

There are a few things I've been meaning to try in a large-scale project:

- Haskell — I am desperately in love with this programming language, but I have never brought myself to finish a project with it. I hope that this will give me the chance to actually finish a full-sized Haskell application using Cloud Haskell's distributed computing model as the primary architecture.
- Apache Kafka — for the core messaging pipeline: sending messages, creating/updating/deleting new posts, sending notifications…. I know I would like to use a distributed architecture, and it seems like Kafka is one of the foremost open-source options.
- MongoDB — as the primary data store. I am fairly certain that I want to use NoSQL for this, and once again MongoDB is a well-known open source option.

My choices of Kafka and MongoDB are not hugely based on a desire to work specifically with them, though I've done some reading and they both seem lovely to work with. Mostly I want to try working with open-source tools that are new to me. However, I will admit that these choices are partly out of spite: the AWS tooling that I use regularly at work (SNS/SQS and DynamoDB, respectively) frustrate me every single day[2], and I'll take any chance I can get to work with something else.

Footnotes

[1] I do have an idea in that direction though, something like a universal catalog that helps people find what they need and where they can buy it. Amazon currently fills that need for most people, but with a flaw: Amazon's purpose is to **make Amazon the only place people go to buy things**, which isn't completely incompatible with helping them find what they need, but I don't think that it's possible to fulfill both of those goals without compromising on one or the other.

[2] *Don't make an AWS hate post, don't make an AWS hate post…*