

Activity Feeds Architecture

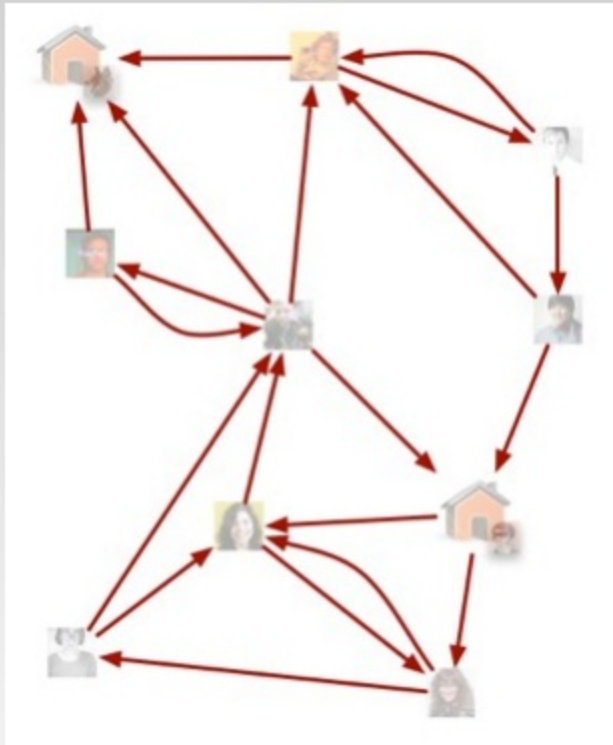
January, 2011

To Be Covered:

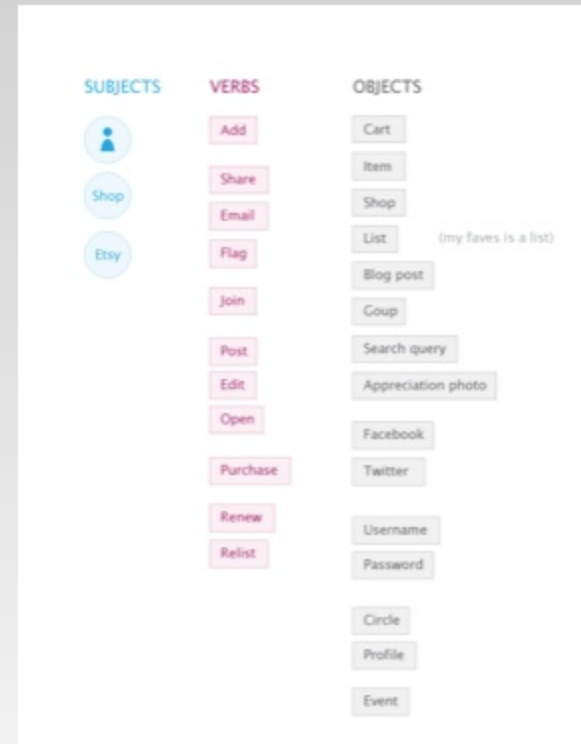
- Data model
- Where feeds come from
- How feeds are displayed
- Optimizations

Fundamental Entities

Connections



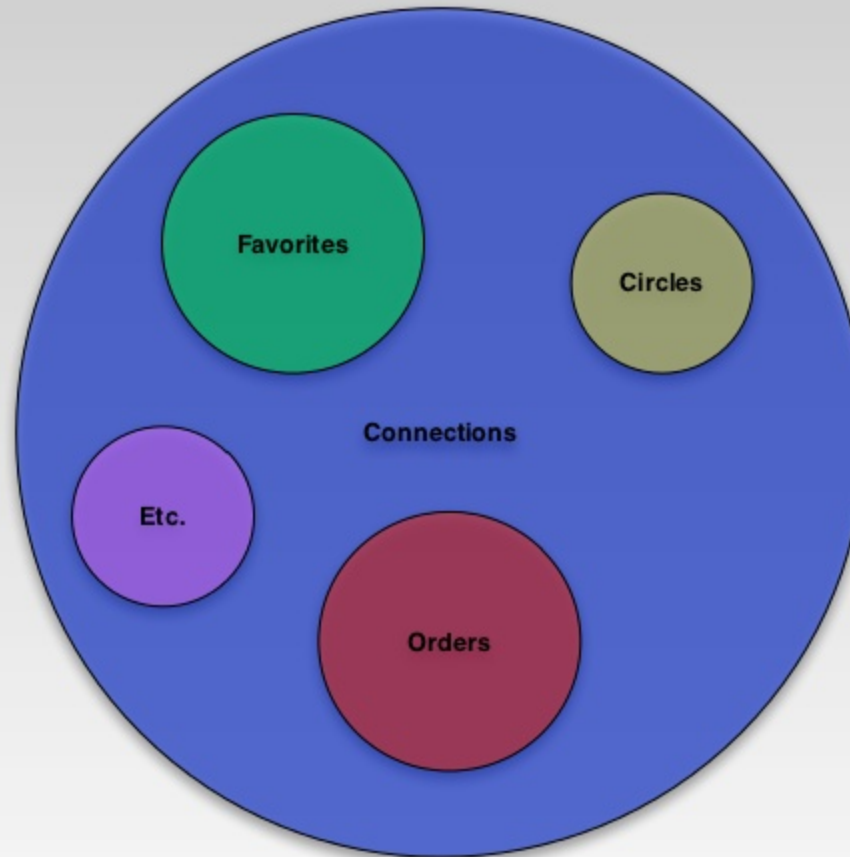
Activities



Friday, January 14, 2011

There are two fundamental building blocks for feeds: connections and activities. Activities form a log of what some entity on the site has done, or had done to it. Connections express relationships between entities. I will explain the data model for connections first.

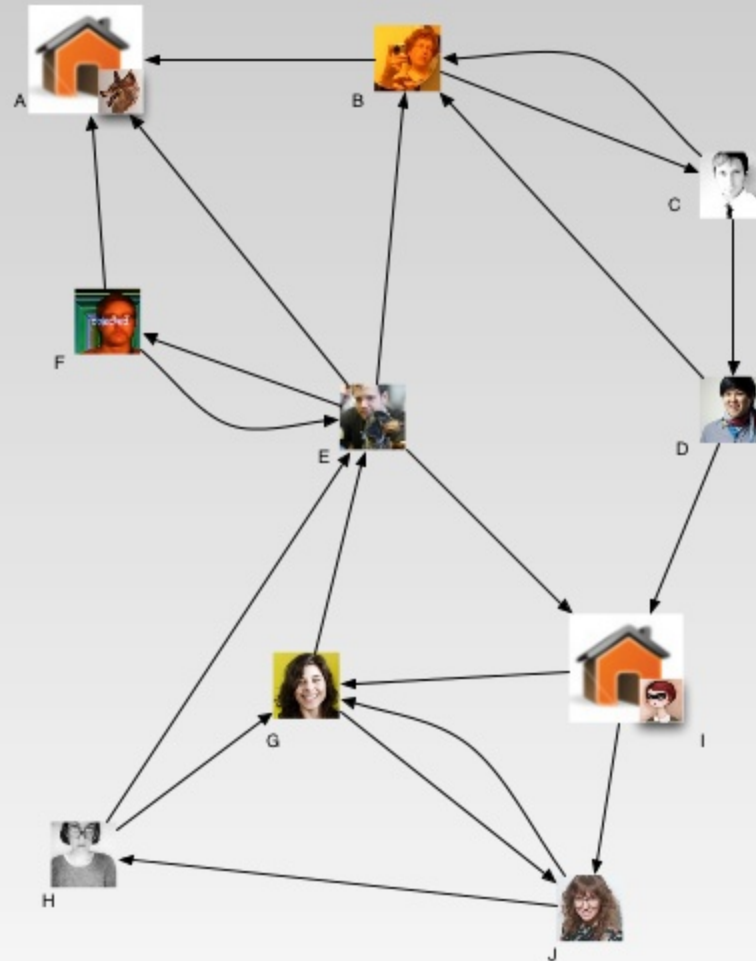
Connections



Friday, January 14, 2011

Connections are a superset of Circles, Favorites, Orders, and other relationships between entities on the site.

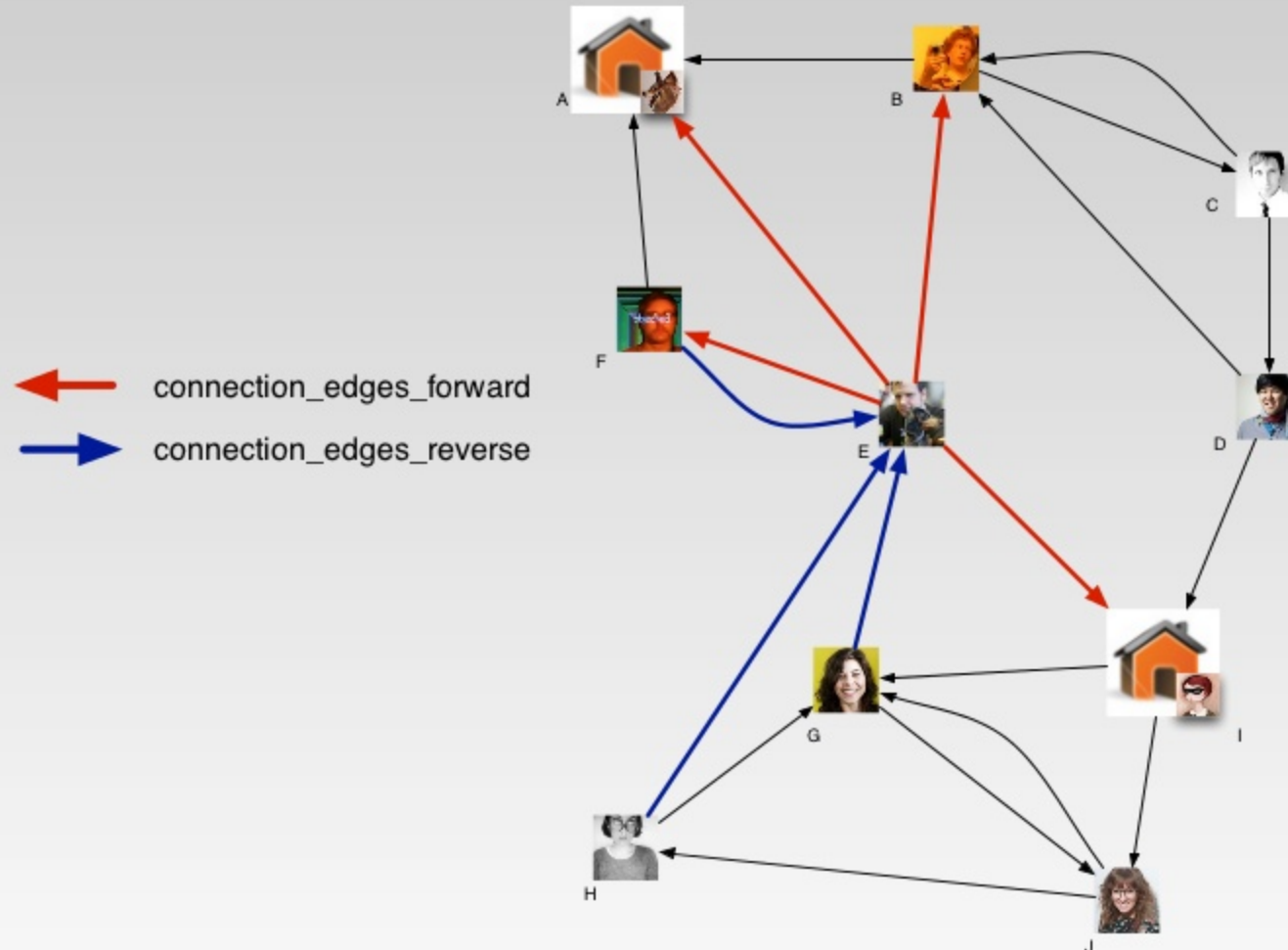
Connections



Friday, January 14, 2011

Connections are implemented as a directed graph.
Currently, the nodes can be people or shops. (In principle they can be other objects.)

Connections



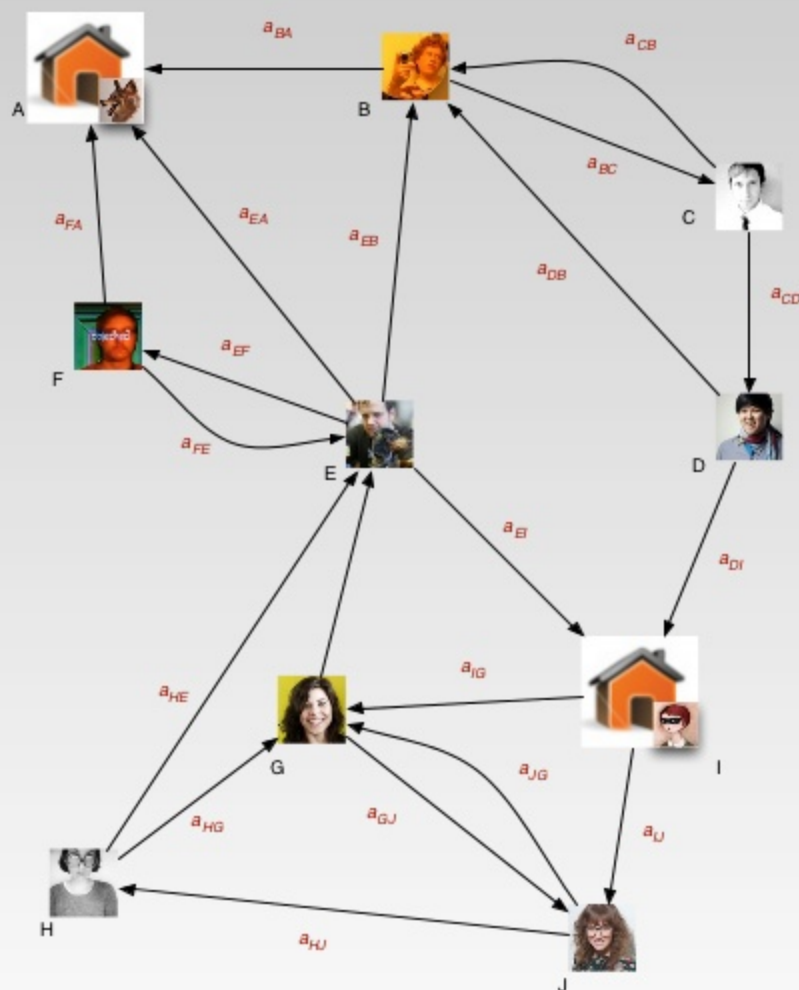
Friday, January 14, 2011

The edges of the graph are stored in two tables.

For any node, `connection_edges_forward` lists outgoing edges and `connection_edges_reverse` lists the incoming edges.

In other words, we store each edge twice.

Connections



Friday, January 14, 2011

We also assign each edge a weight, known as affinity.

Connections

On *H*'s shard

connection_edges_forward

from	to	affinity
H	E	0.3
H	G	0.7

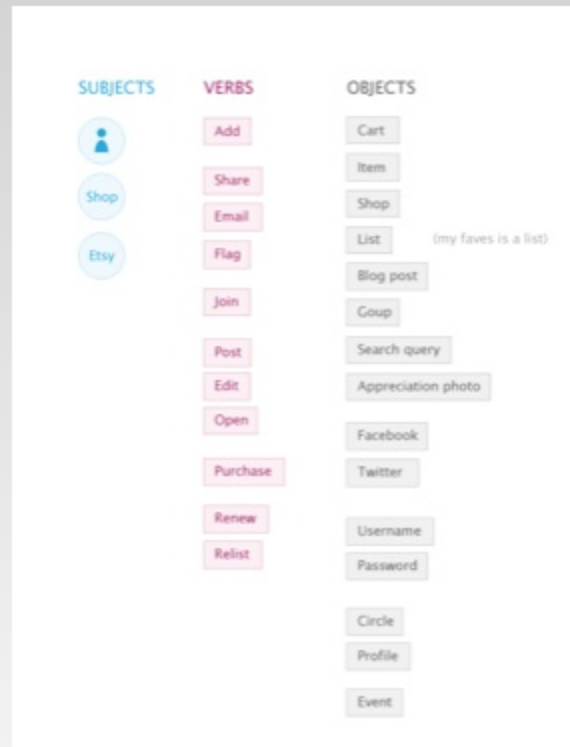
connection_edges_reverse

from	to	affinity
J	H	0.75

Friday, January 14, 2011

Here we see the data for Anda's connections on her shard. She has two entries in the forward connections table for the people in her circle. She has one entry in the reverse connections so that she can see everyone following her.

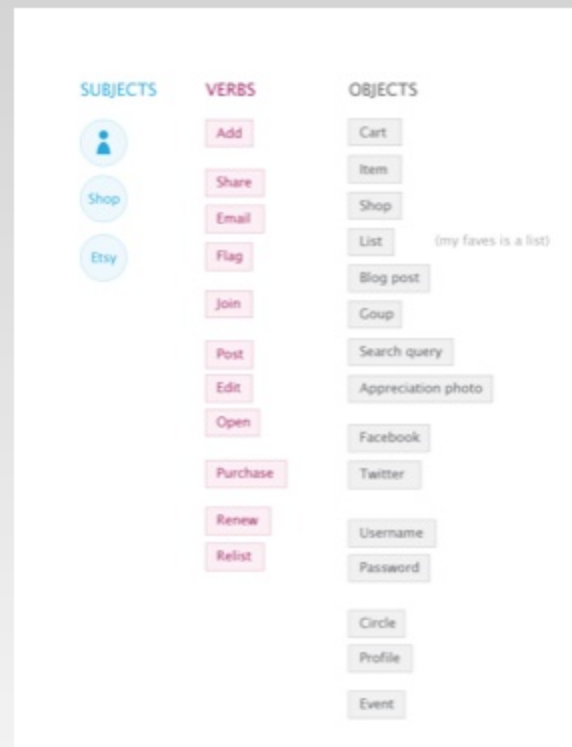
Activities



Friday, January 14, 2011

Activities are the other database entity important to activity feeds.

activity := (subject, verb, object)



Friday, January 14, 2011

As you can see in Rob's magnetic poetry diagram, activities are a description of an event on Etsy boiled down to a subject ("who did it"), a verb ("what they did"), and an object ("what they did it to").

activity := (subject, verb, object)

(Steve, connected, Kyle)

(Kyle, favorited, brief jerky)

Friday, January 14, 2011

Here are some examples of activities.

The first one describes Steve adding Kyle to his circle.

The second one describes Kyle favoriting an item.

In each of these cases note that there are probably several parties interested in these events [examples]. The problem (the main one we're trying to solve with activity feeds) is how to notify all of them about it. In order to achieve that goal, as usual we copy the data all over the place.

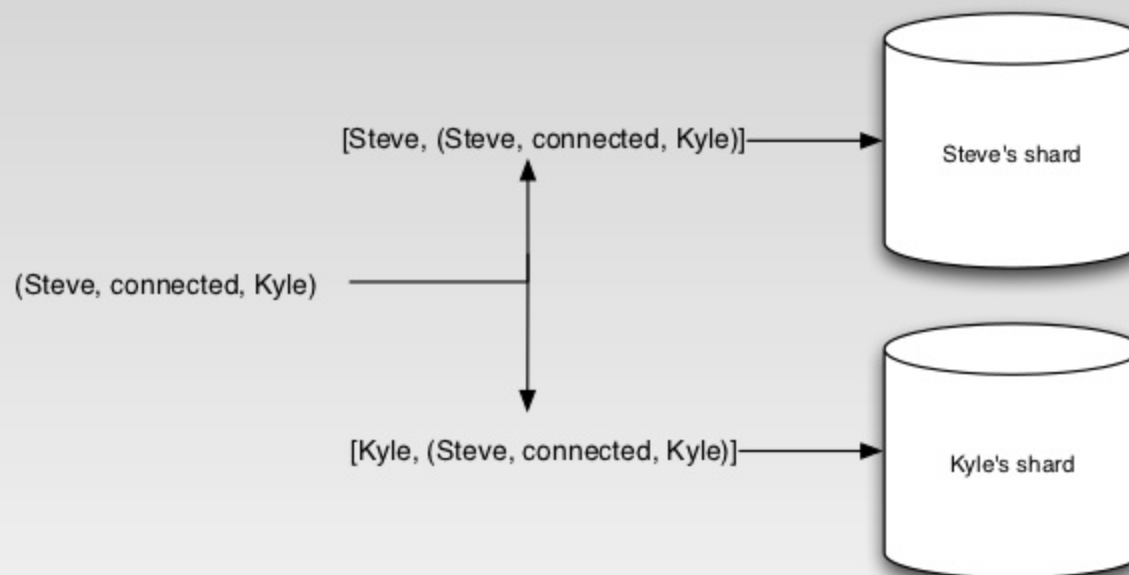
~~activity := (subject, verb, object)~~

activity := [owner, (subject, verb, object)]

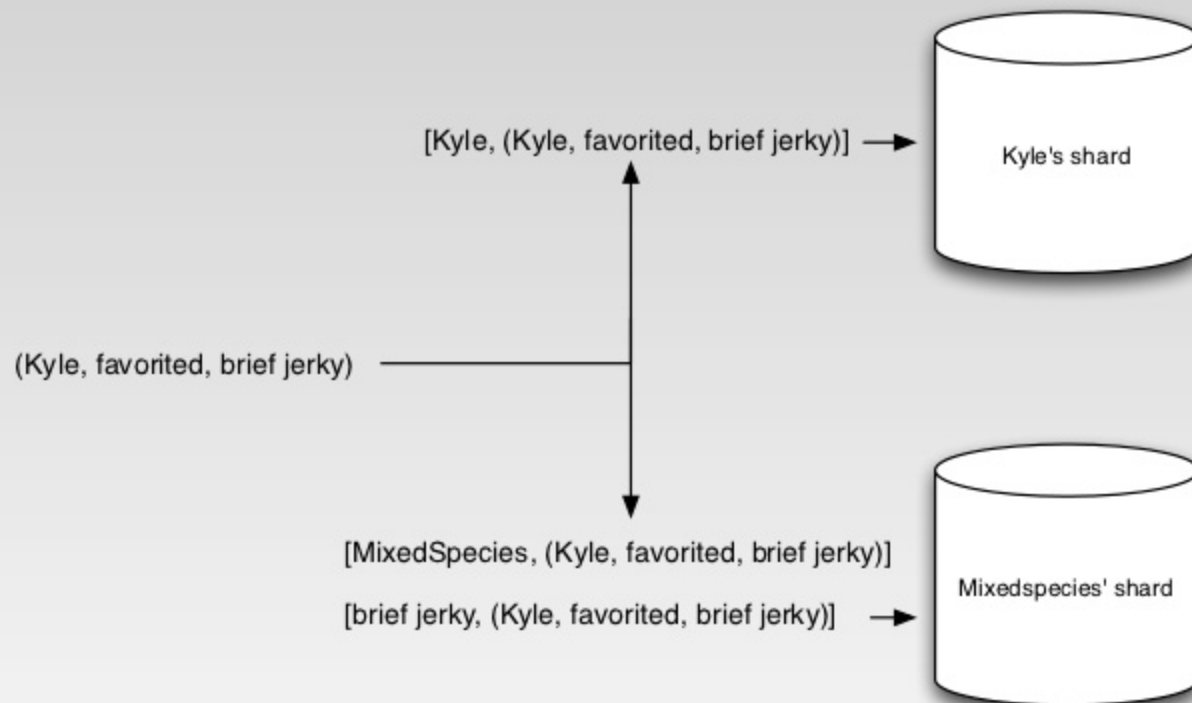
Friday, January 14, 2011

So what we do is duplicate the S,V,O combinations with different owners. Steve will have his record that he connected to Kyle, and Kyle will be given his own record that Steve connected to him.

activity := [owner, (subject, verb, object)]



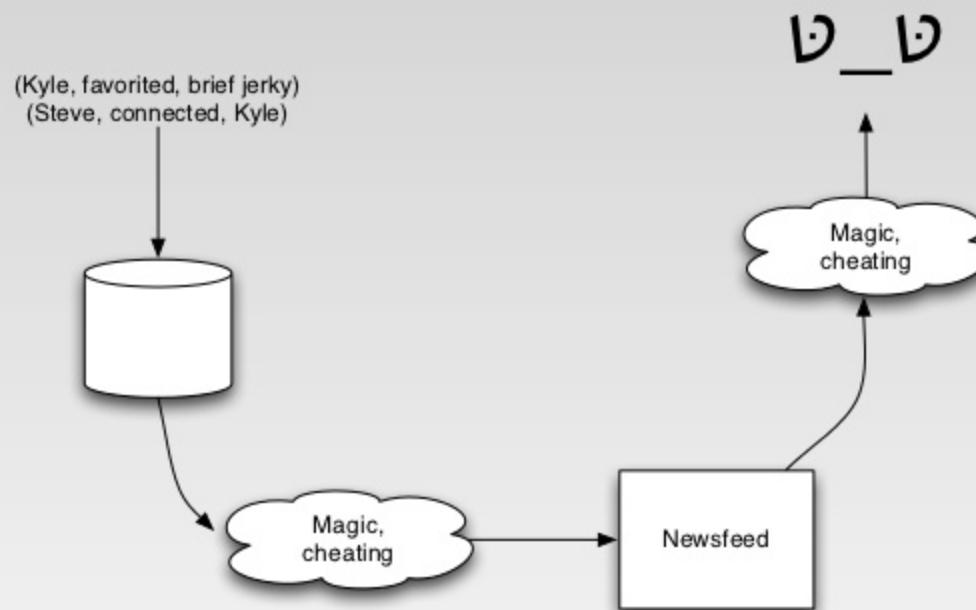
activity := [owner, (subject, verb, object)]



Friday, January 14, 2011

In more complicated examples there could be more than two owners. You could envision people being interested in Kyle, people being interested in MixedSpecies, or people being interested in brief jerky. In cases where there are this many writes, we will generally perform them with Gearman. Again, in order for interested parties to find the activities, we copy the activities all over the place.

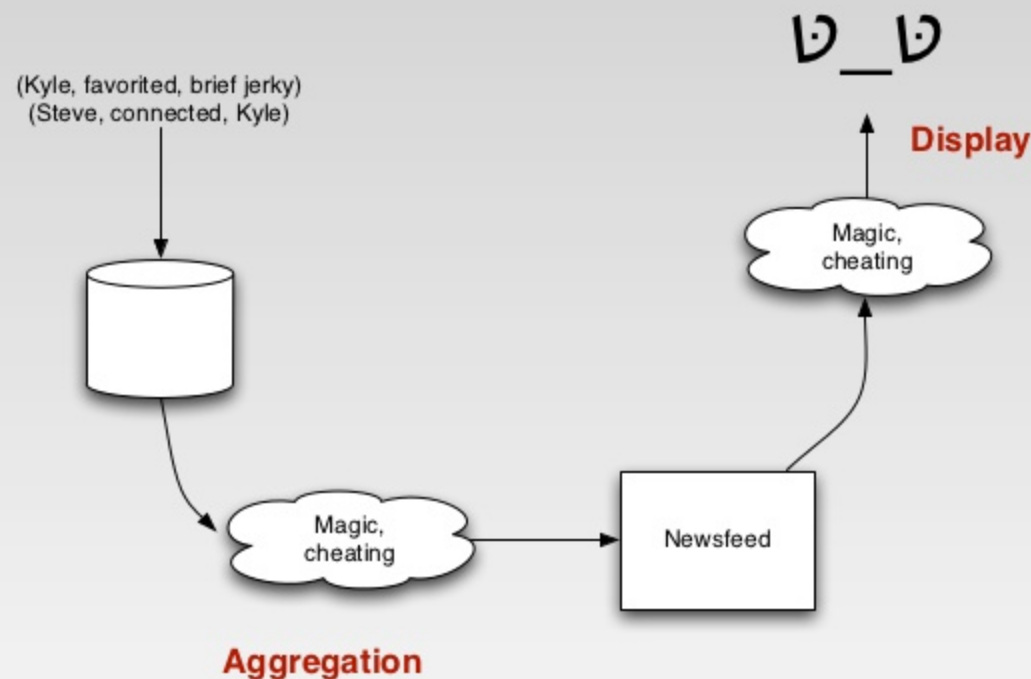
Building a Feed



Friday, January 14, 2011

Now that we know about connections and activities, we can talk about how activities are turned into Newsfeeds and how those wind up being displayed to end users.

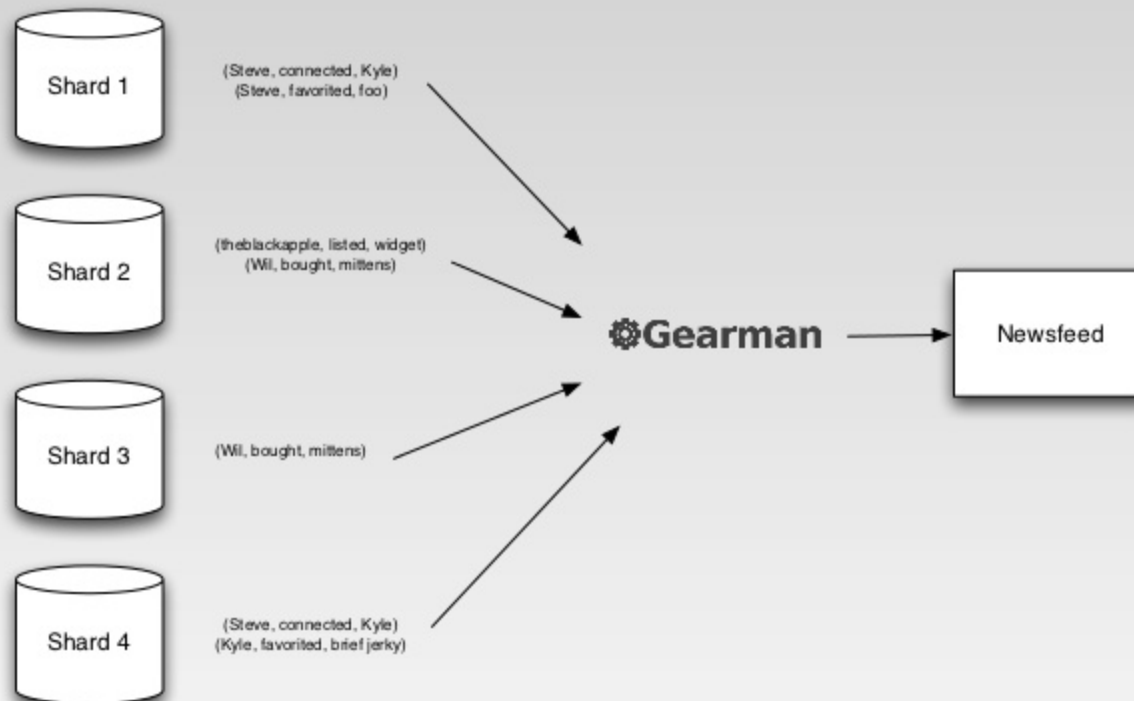
Building a Feed



Friday, January 14, 2011

Getting to the end result (the activity feed page) has two distinct phases: aggregation and display.

Aggregation



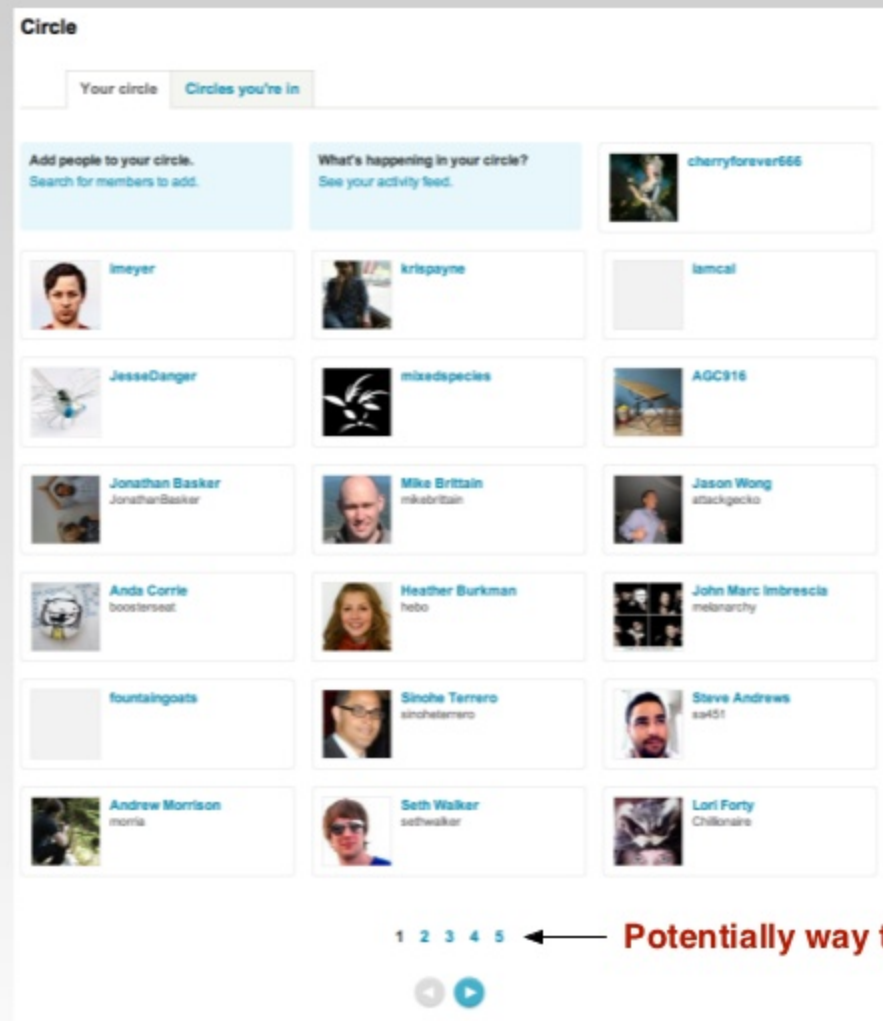
Friday, January 14, 2011

I am going to talk about aggregation first.

Aggregation turns activities (in the database) into a Newsfeed (in memcache).

Aggregation typically occurs offline, with Gearman.

Aggregation, Step 1: Choosing Connections

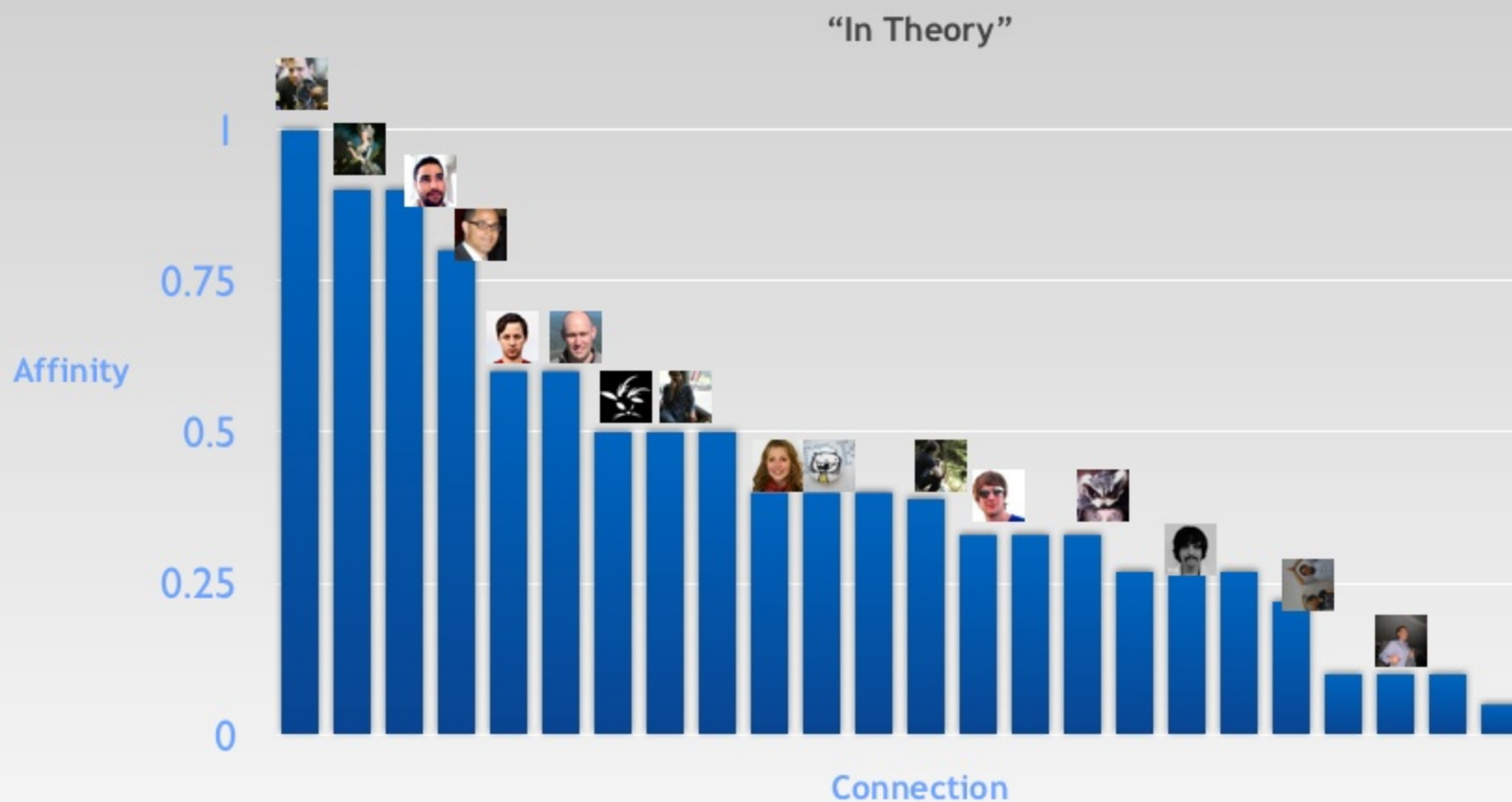


Friday, January 14, 2011

We already allow people to have more connections than would make sense on a single feed, or could be practically aggregated all at once.

The first step in aggregation is to turn the list of people you are connected to into the list of people we're actually going to go seek out activities for.

Aggregation, Step 1: Choosing Connections

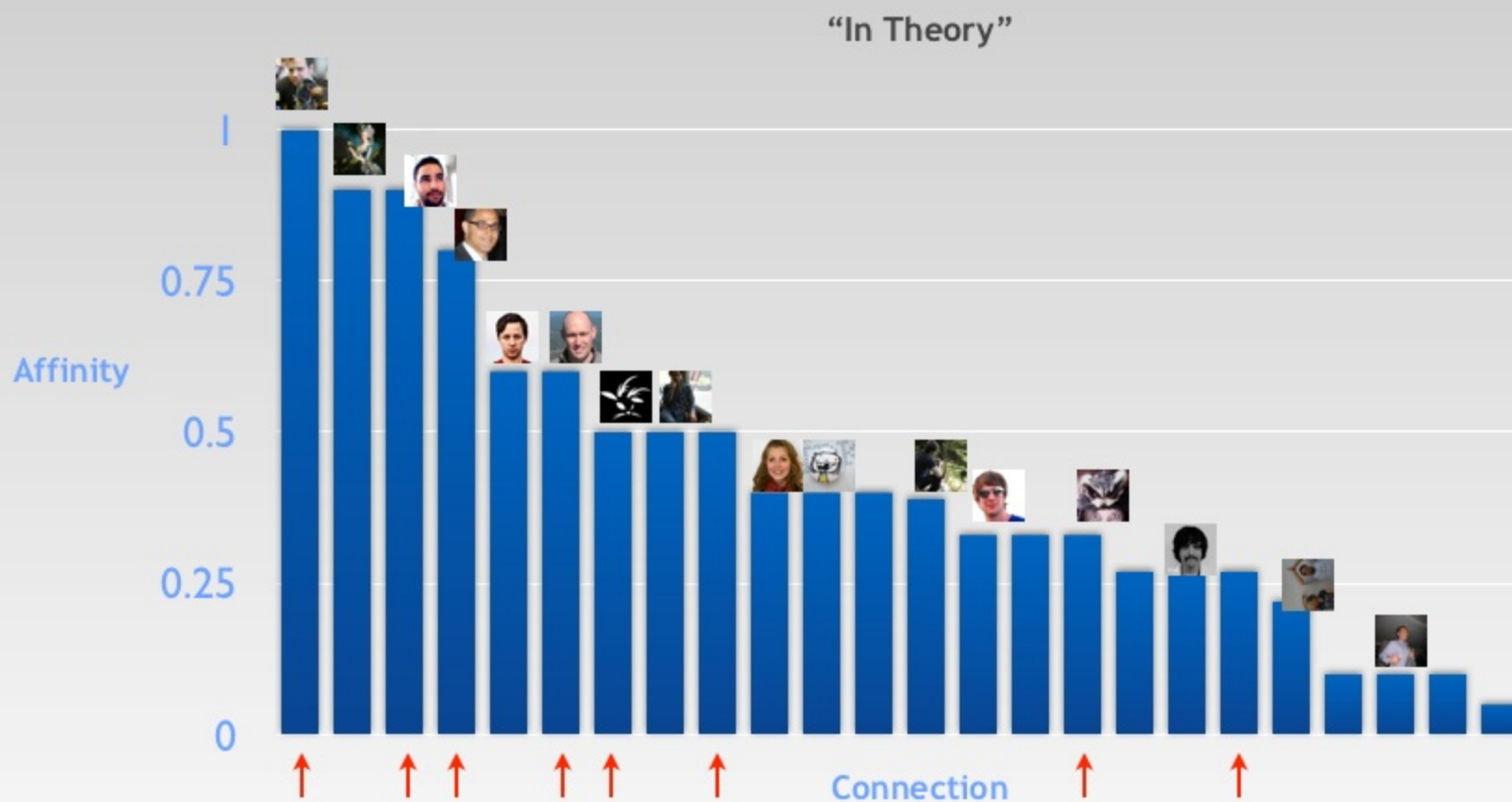


```
$choose_connection = mt_rand() < $affinity;
```

Friday, January 14, 2011

In theory, the way we would do this is rank the connections by affinity and then treat the affinity as the probability that we'll pick it.

Aggregation, Step 1: Choosing Connections



```
$choose_connection = mt_rand() < $affinity;
```

Friday, January 14, 2011

So then we'd be more likely to pick the close connections, but leaving the possibility that we will pick the distant ones.