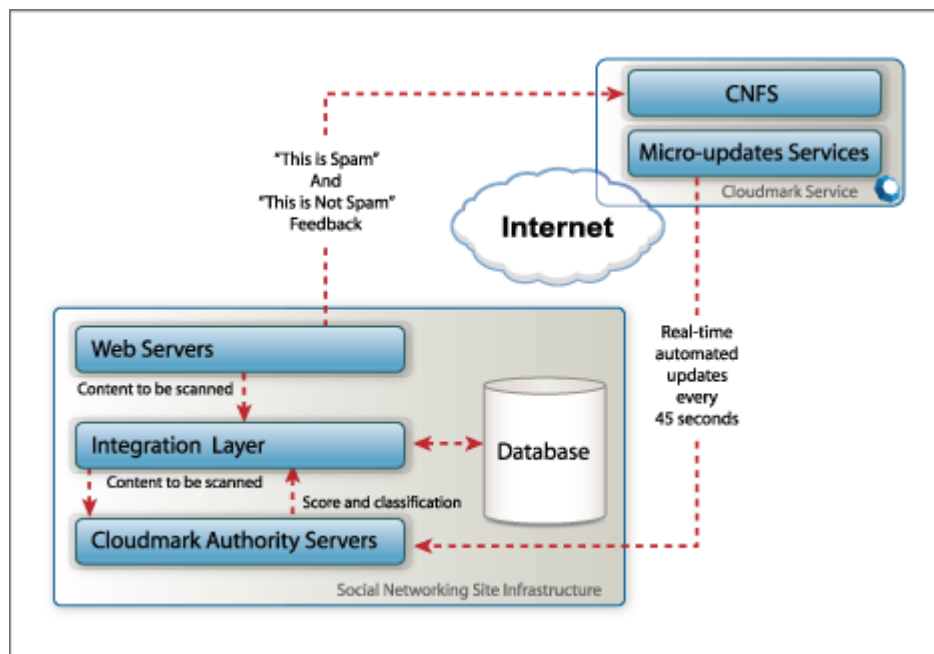
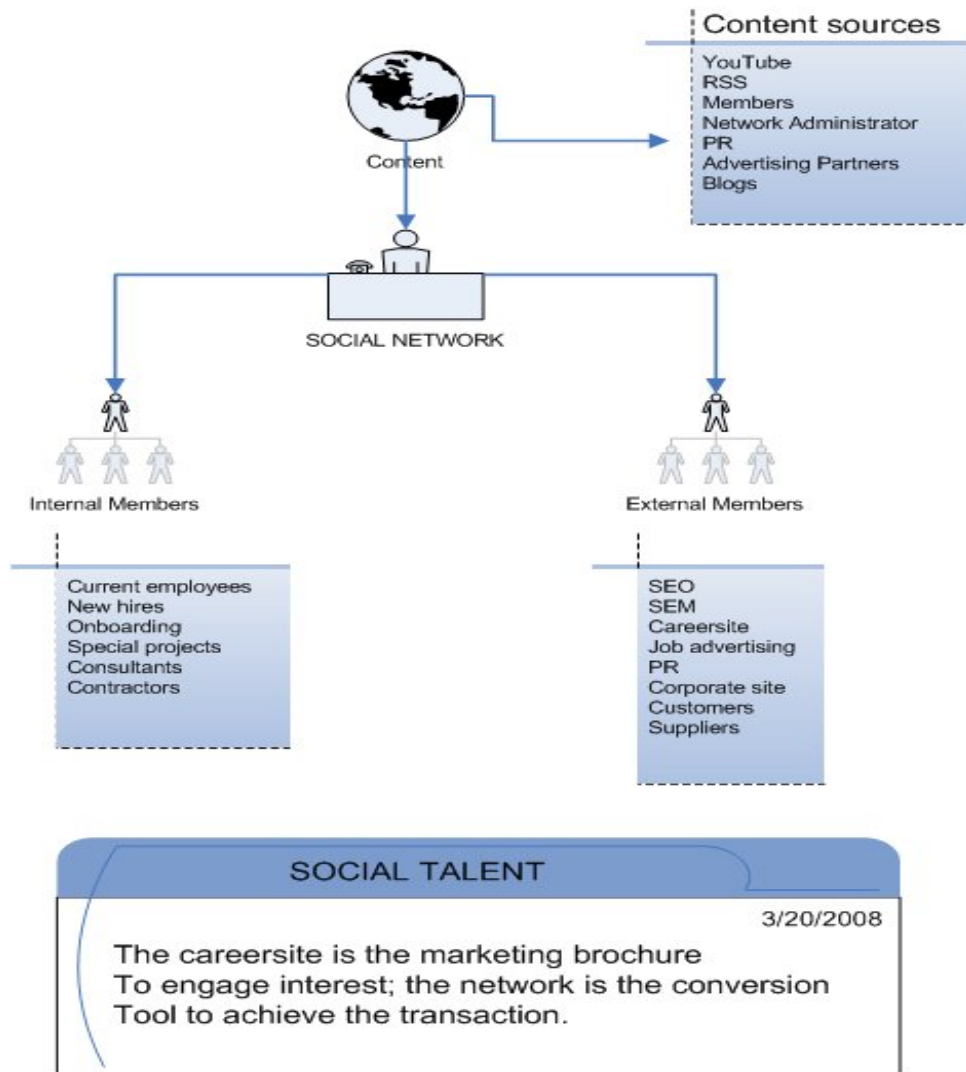


# SOCIAL NETWORKING

We began the problem by exploring different ways to visualize both the communications network of an email collection and the hierarchical information of the underlying organization. Initially, we focused on different visualization methods that combined these two pieces of information into one display. We looked at various ways to combine network hierarchy data including treemaps, node link diagrams and matrix based representation. We experimented with clusters, varying colors, sizes, shapes and lines to differentiate the different types and relationships in the data. What we found, however, was that the resulting visualization struggled with occlusion and clutter resulting from trying to display too many relationship types and too many attributes at once.





Given our initial experience, we realized that rather than trying to represent all this data in a single visualization, losing readability in all the clutter, that what we should do is represent different aspects of the same data in multiple visualizations. In our case, this meant showing the email communication graph and the hierarchy graph in two different visualizations. This change has two advantages. First, by separating different aspects of the same data in multiple displays, no single display would be forced to hold everything. This means smaller graphs and simpler representations. Next, this approach allows us to represent the aspect of the data using whatever visualization best suites that data type and the task at hand.

For this project, we explore this idea by creating a tool that lets us view our dataset in multiple ways. For a first attempt, we limit the number of views to two though there is no limitation in our approach that forces that limitation. We began coding the tool using Java since we wanted it to be portable among multiple systems. We looked at various visualization libraries including Prefuse, Jung, Piccolo and the InfoViz toolkit. We settled on using Prefuse only, however, since we found the library well documented with lots of support for network visualization. We also wanted support for other visualization types, especially treemap, which the other tools did not inherently support. In order to support collaboration, we set up a CVS repository and a Google shared document for use in doing the write up. We are also using the wiki to store and share related work.

Given this motivation, we propose to develop a tool that allows a user to generate and use knowledge about the underlying social network in analyzing the communications traffic. Specifically, we propose to create a tool that allows us to iteratively establish an underlying social network from the given communications network by allowing us to specify known relationships in the group and allowing us to use that knowledge to identify other relationships that match that communication pattern. We then allow a user to navigate this inferred network side by side with the communications network to provide context in understanding the traffic in the communications network.

## Project Description

Recent estimates state that the number of email messages sent in a day exceeds 2.25 billion. As such a major form of communication, there is great interest in archiving and analyzing large email collections for both historical and legal reasons. To this end, there have been a number of tools created to work with email archives. In the field of information visualization, the standard approach of representing the communications as a network graph, with nodes as the individuals and edges representing email exchange, have been successful at various tasks including highlighting anomalies in the traffic counts and illustrate word usage in communications. Although these tools have provided great insight into these email collections, they focus too much on the communications graph and not enough on a major property of the archive, the underlying social network.

The exploration of any email archive must take into account the underlying social network to provide context for the communications. One example of this can be seen in the most widely studied email archive, the Enron email collection. In this collection, documentation of the employees from whose accounts the emails were taken lists one of the individuals as Mark Taylor, a senior level executive. Looking solely at the communications graph, this identification seems normal. Mark Taylor send a number of emails, as an executive would, and a quick glimpse at the email content may support this. Taking account the social network given by an organizational chart, however, we see that Mark Taylor seems to communicate frequently with two groups from two very different divisions of the company and that Mark Taylor emails both high level executives and entry level associates frequently. Verifying with the communications of other senior level executives, we can establish this is not common and now have reason to look at Mark Taylor further. In this case, knowing the underlying social network revealed that there are two Mark Taylor in the company, an executive and a low level associate from two different groups, whose accounts were mistakenly merged in the collection.

## **The project has the following modules.**

### **1. Activity**

The Activity module keeps track of the things people do on your site and provides mini-feeds of these activities in blocks, in a specialized table, and via RSS. The module is extensible so that any other module can integrate with it. The messages that are produced are customizable via the admin interface and are context sensitive. Some theoretical example messages include:

- You wrote a comment in response to "example comment title"
- Jim wants to be your friend. Approve this friendship here
- Nancy is Phil's latest fan

### **2. Advanced Forum**

Advanced Forum is a theme/glue module that enhances Drupal's core forum module. The included styles provide the look typically found in stand alone forum software and the code provides some of the functionality, mostly related to the end user experience. By adding other Drupal contributed modules, many of which are automatically used by Advanced Forum, you can achieve much of what stand alone software provides. Because it uses the core forum module, it uses the node and comment system built into Drupal and is completely integrated, not a bridge.

- Forum Listing
- Topic Listing
- Individual Forums

### **3.Profile Kit**

The Profile Kit provides building blocks for putting together fancy user profile pages like those commonly found on social networking sites. It is meant to work with ,panels2 but the provided pieces can be used independently as well. The provided instructions are meant to get you up and running with a nice looking profile with no coding and no theme changes but are only one example of how this module can be used. Because it is a kit module, the idea is to provide some essential pieces and a base to build on. While the "out of the box" profile is usable, it is only intended as a start.

### **4. User Relationships**

The User Relationship module allows users to create named relationships between each other. Currently relationship types are defined by the admins. Relationship types have a name, a plural name, a timeout, an auto approve designation and a directionality. The name can be anything. The timeout is the amount of time a request stays active. For example, if it's set to 30 days and a user requests a relationship with another user and that user doesn't accept or deny the request within 30 days the request will be dropped.

If auto approve is set, the system skips the request->approve/deny process and simply creates the relationship. An application of this might be the relationship type "subscriber." You don't necessarily need to approve of me subscribing to you.

There are only two types of directionality: two-way and one-way. An example of a two-way relationship may be "friends." If you're my friend I'm your friend. One-way relationships may be something like "manager," "follower," "stalker." You may be my manager, but I'm not yours.

**Front End:** Java-J2EE/ .Net

**Back End:** MySQL/SQL Server

**Software Requirements:** Apache Tomcat/IIS

**Hardware Requirements:** A system with capability of supporting a web server & internet.