Mr. Privacy:
Open and Federated Social Networking Using Email

Michael Fischer   T. J. Purtell   Ruven Chu   Monica S. Lam
Computer Science Department
Stanford University, Stanford, CA 94305
{mfischer,tpurtell,ruven,lam}@cs.stanford.edu

ABSTRACT

Do you think social networking should always be closed? We don't. That's why we have created an open and federated social networking platform that is bootstrapped from email. We call this system Mr. Privacy.

Applications built on Mr. Privacy are "social apps" that look nothing like email. Email is used only as a transport and personal database. We choose email because it is more pervasive than any social network; it is an open and federated platform that lets consumers choose their providers or hosting their own. Mr. Privacy provides an API to developers to build distributed social applications with significantly less effort and provides consumer with privacy–automatically.

We have developed a prototype Mr. Privacy platform for the Android, iOS, and the Firefox browser for PCs. On top of Mr. Privacy, we created three applications to share GPS locations, playlists, as well as our browsing experience. Preliminary results suggest that the email protocols suffice for building these kinds of social applications. Overall, not only does this model better support data privacy and ownership, its openness facilitates competition that will lead to further innovation in the fast growing social software arena.

1. INTRODUCTION

Today, we can socialize with our friends on many web services effortlessly. We can find out what our friends think as we browse the web, regardless if it is about music, restaurant reviews, news, etc. We believe this is the beginning of an *in situ* social networking experience, where sharing takes place as we go about our daily life with every application or action we take. We no longer have to log in to individual social network sites just to share. This paper explores how we can support *in situ* social networking on an open and federated system to help preserve user data privacy and increase competition.

1.1 Motivation

Social plugins have changed how people discover web sites. For example, since the introduction of Open Graph half a year ago, 2 million websites are serving Facebook social plugins [26] to the 500 million Facebook users. Websites that have integrated social plugins reported dramatic increases in referrals.

We are concerned that having a centralized, proprietary, *in situ* social networking platform will greatly decrease consumers' control over their privacy and change the ecosystem of the internet. By embedding Facebook Like buttons on their site, companies are also automatically sharing their customer relationships with Facebook. While our browsing histories continue to be collected and used by web sites, now these sites also give the social networking partner detailed knowledge of our browsing history and *everything* that we wish to share with our friends. Social networks are no longer just a source of web referrals, but are also brokers of highly-marketable detailed profiles of large numbers of users[1]. This enables the proprietary owner of social data to dictate the terms of partnerships; the recent failure for Apple Ping to connect with Facebook is a case in point[5]. If a monopoly emerges that owns the world's social graph and interactions, it will be a great detriment to the economy and a disservice to the consumers.

At a more immediate and personal level, consumers today are faced with the dilemma between sharing at a loss of privacy or keeping the information to ourselves. We need to stay vigilant to react to the changing privacy policies motivated by corporate interests[11]. We believe it will be welcome relief if we can social network with an acceptable guarantee of privacy.

1.2 Open and Federated Social Networking

Recently, privacy in social networks has received significant media attention. There are currently various attempts to create new distributed social networking infrastructures that provide similar functions as existing social networks; we believe they are unlikely to succeed. Hundreds of millions of consumers have voted with their actions that the status quo is acceptable. Social networking is sticky, individuals cannot change their network on their own and still interact with their friends. Finally, it is too difficult for many users to host their own data server.

We must provide users with a better social experience than the one they already have. We propose to create an open and federated platform to support the development of many *in situ* social applications that allow friends to interact with each other frictionlessly. For the sake of adoptability, Mr. Privacy is built upon email, which is itself a mature, scalable, open and federated infrastructure supporting over 1 billion users. We can socialize with anybody as long as we know his or her email address. We need not sign up to join the same social network. All the shared information is stored as email messages. While most people get their email accounts from

a few large companies, individuals and corporations do have the freedom to use paid, advertisement-free email services or to run their own servers.

Mr. Privacy is a collection of APIs (application programming interfaces), to be provided on all major platforms (phones and browsers), to support social networking functions. Applications can get access to a user's social contacts and interact with them using simple data access operations for application-defined data types. Mr. Privacy hides the low level details by translating these operations into email protocols, SMTP (Simple Mail Transfer Protocol) [22] and IMAP (Internet Message Access Protocol) [6]. Technically, Mr. Privacy enables developers to write fancy email clients easily. The key, however, is that

1. Users do not know that Mr. Privacy applications are email clients because they sport a user interface similar to any other social applications, and

2. Developers need not know they are writing email clients either. They simply create their application-specific data structures, store them, and retrieve them from a database. The database turns out to be distributed, implemented on top of email, and the developers do not have to worry about hosting it or scaling it!

Mr. Privacy overcomes the difficulty of creating a new federated system by bootstrapping with email. Applications can be built using the email system as it is today, allowing users to interact with their email contacts through social user interfaces. Even though the email protocol is not optimal, this allows for experimentation with federated social networking. Demonstrated success will hopefully entice email providers to collaborate in optimizing the protocol.

1.3 Contributions

The concept of using email to create an open and federated platform for in situ social networking. By turning email into a distributed database, we are leveraging a mature, open and federated system with an even larger installed base than the largest online social network today. Not only does it support better data privacy and ownership, its openness facilitates competition that will lead to better products for consumers.

Prototype of Mr. Privacy. We evaluate the concept of using email as a database for social networking data by implementing Mr. Privacy on three different platforms: Android, iPhone, and the Firefox web browser. While the lack of server-side support of message filtering added significant complexity to our implementation, we found that the SMTP and IMAP protocols with the appropriate extensions are adequate as data storage and transport for the class of social applications we studied.

Mr. Privacy applications. We illustrate the benefits of federated *in situ* social networking with three Mr. Privacy applications. The GPS sharing application lets us share our current locations without having our whereabouts tracked by a central authority. Our social music application illustrates how different music service providers can inter-operate, letting friends share music playlists while getting their music

Figure 1: User interface of GPS sharing using Mr. Privacy on an Android phone.

from different sources. Allowing users to discover more music through their friends is likely to encourage more music sales.

Social, contextual browsing with SocialBar. Using our SocialBar extension for Firefox, friends can share comments on any web page they visit, without having to give away their browsing history and conversations to a single third-party. Because it is part of the browser, our friends and their comments are available on the side providing a social experience that is more compelling than webpage-based social browsing. Furthermore, SocialBar can be integrated with media aggregators[21] to create a federated experience even across proprietary social networks.

1.4 Organization of the Paper

We first present the basic ideas of Mr. Privacy, using the GPS sharing application as a concrete example in Section 2. Next, Section 3 presents SocialBar, using it to illustrate the advantages of openness and *in situ* social networking. We then discuss the relationships of Mr. Privacy with email and other social networking infrastructures in Section 4. Section 5 discusses our prototype design and experience on the three platforms and three applications we built. Finally, Section 6 discusses related work and Section 7 concludes.

2. BASIC MR. PRIVACY DESIGN

A Mr. Privacy application looks like any other social application, and does not look like email at all. Email is used primarily for its user identities, and as a transport and a distributed database. Mr. Privacy is built on top of email standards: SMTP (Simple Mail Transfer Protocol) [22], to send mail and IMAP (Internet Message Access Protocol)[6], to retrieve mail from a server. This section first explains the basic concept of the system with a simple but compelling Mr. Privacy application, GPS sharing.

2.1 GPS Sharing

Consider physical check-in services like Facebook Places, Four Squares, and Google Latitude. While it is fun to let our friends know all the new places we are visiting, making such sensitive information like locations public is potentially dangerous. The "PleaseRobMe.com" website, for example, collects information about when people are away based on

tpurtell.test just arrived at
370 Serra Mall, Stanford, California 94305 United States

Download **MR GPS**, an app for your phone to follow where your friends are.

About **MR PRIVACY** & **MR GPS**
MR PRIVACY uses the GMail account you already have as your personal server. **MR GPS** then works on your behalf to communicate with **MR PRIVACY** to store and recall information. **MR PRIVACY** keeps your data safe so you don't have to. To learn more, go to MrPrivacy.me or scan the QR code below with your cell phone.

Figure 2: An email for GPS sharing using Mr. Privacy.

public status information and can be used by burglars to pick their victims [24]. While some social websites do let us control access to our data, they nonetheless become Big-Brother-like, possessing large amounts of personal data

We have created an application for the Android phone called Mr. GPS that allows individuals exchange GPS locations. The application sports a similar UI as any other check-in services, as shown in Fig. 1. A user first enters his email account and password, and specifies two friends he wishes to send his location to. When he "checks in", his GPS location is sent to his friends using email. The application shows the checked-in locations of his friends and he has control over whose locations he wishes to see.

The email sent by Mr. GPS (Fig. 2) can also be viewed on an email client or through a web interface. The message has a machine-readable part containing the GPS coordinates, which is not displayed. The subject of the message identifies that it comes from Mr. GPS, a Mr. Privacy application, and includes the information "Hey, guess where I am now ...". The message body shows a map if HTML display is enabled and an address if otherwise. At the end, it includes the line "Sent by Mr. GPS, powered by Mr. Privacy", which includes the location where the application can be downloaded. There is no central server that knows all the checked-in locations of all the users.

Making Mr. Privacy messages human-readable as well serves several important functions. First, many users check their email frequently so they can receive the information promptly even if they are not running the application. Secondly, it serves the function of letting the user know that there is new data for the Mr. Privacy application. Thirdly, and most importantly, it helps make the social application "viral". If a user of an application shares a piece of data with another user who does not currently have the application installed, the email message acts as an invitation to download the application. These are the same reasons why many so-

cial networking portals also send mails to notify users of updates.

2.2 Message format

To facilitate filtering on the server side, Mr. Privacy messages have a stylized subject header of the form "<subject>[Mr Privacy][<Tag>]". The tag is simply the name of the application that owns the data in the message. The body of the email contains the same information in three different ways: text for text-only browsers, html for richer clients, and binary (JSON formatted data [7] or images) for the applications. We create a "multipart/alternative" MIME message, consisting of text, binary, and html parts, in increasing order of fidelity. A viewer will display the one with the highest fidelity that it can. (Binary is placed before html because some viewers would always display the last part even if it cannot be viewed by a human).

2.3 The Basic API

At the basic level, the application developer interacts with Mr. Privacy using four simple API calls.

- CONNECT: Connects to an IMAP and SMTP server and authenticates using the user credentials.
- SEND: Transmits a Mr. Privacy message with the specified tag to a set of recipients.
- RECEIVE: Receives a list of Mr. Privacy messages with the specified tag newer than a certain reference message.
- WAIT: Waits for messages with the specified tag that are newer than a certain reference message to arrive.

Behind the scenes, Mr. Privacy does some special handling to keep the user's inbox clean. When new messages arrive in the inbox, Mr. Privacy checks to see if they are directed towards Mr. Privacy applications. If they are, it moves the messages into a special folder. This is done automatically as long as a Mr. Privacy application is connected to the IMAP server.

3. SOCIALBAR: CONTEXTUAL SOCIAL BROWSING

Facebook has extended social capabilities to websites with its platform by allowing websites to display social content from their network inline. This provides a more *in situ* social browsing experience, but at the cost of privacy. Because enhanced pages embed content from Facebook, the company is able to acquire browsing history logs for all web users. From a web content provider's perspective, this concern is easily trumped by increased visitorship and potential content virality. Developers also like this integration because they can do it using minimal Javascript HTML code without standing up their own data servers.

To allow users to curate content for each other with privacy, we have developed SocialBar, a FireFox browser sidebar extension built on top of Mr. Privacy. SocialBar allows us to keep track of all the new comments, find out what a specific friend has commented, and most importantly, find what our friends say about the pages we are viewing. We can also add new comments to threads of discussions and forward
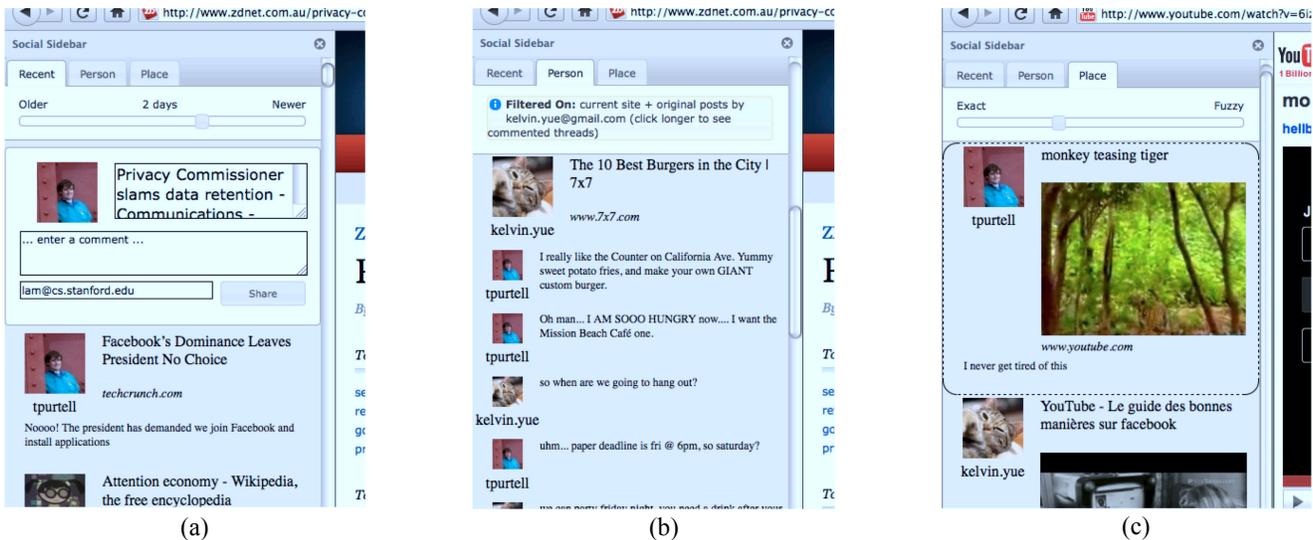
Figure 3: SocialBar: (a) "Recent" context and sharing interface, (b) "person" context and comment interface, (c) "place" context and content embedding.

the conversations to any other friends' email addresses. We first describe how SocialBar works, and next explain why SocialBar is attractive beyond just offering privacy, which is important because privacy alone does not attract users.

3.1 Three Contexts

SocialBar introduces content to our network of friends and makes it available within three contexts: "recent", "person", "place", corresponding to three tabs in the sidebar. In the "recent" context, SocialBar lets you see new content your friends have curated in a time-oriented view just like many other centralized services, as shown in Fig. 3(a). When you browse to a new site, a share box is always available to spread content to peers. To facilitate easy sharing, the title of the page is automatically filled in from the currently visible website. Content can be shared with anyone who has an email address, whether or not he or she is using Social-Bar. Users of SocialBar will see the rich content discussions, while people just using email will receive a text copy of the discussion in their inbox, as shown in Fig. 4.

Figure 4: Example of an email sent by SocialBar.

Once many of your friends are using the SocialBar, you will have quite a bit of material to peruse because it is such a pleasure to share while you browse. It is thus necessary to filter curated content in various ways. To see what one particular friend has posted, you can click on his avatar image

in the SocialBar; you will automatically be immersed in the "person" context with all the content posted by your friend (Fig. 3(b)). You can also click on the avatar longer to see all the discussions that a friend was involved in. All posts made in the "person" context are defaulted to share with only the person selected. In any of the contexts, a commenting interface is displayed when you click on a post, as shown in the figure. You can just indicate that you like something or you can provide more comments for your friends.

Finding new content from our friends is great, but it is also important to hear what they think about the pages we are currently looking at. The "place" context of SocialBar, shown in Fig. 3(c), provides a content-centric view of social discussion. The "place" context is filtered to display content from a particular web site, by default this is the currently visible site. There is a fuzziness slider which allows the user to see other content that is related to the current page. As the slider is moved more to the right, other socially curated items from the same domain become visible allowing the user to branch out across similar topical areas.

Some web content, such as videos and pictures, are not represented best with text. To enrich the display in the sidebar, we identify shared images and inline a small preview of them instead of showing a title. This makes it easier to understand the context of a discussion on multimedia content before visiting a curated item. We also applied this concept to videos by embedding a small representative screenshot of the video in the sidebar (Fig. 3(c)). In this case, the title of the web video is included as well because it is usually a concise summary of the content.

3.2 Advantages of SocialBar

SocialBar provides a better *in situ* browsing experience than a portal-based one because it is built into the browser. Our friends are available on the side as we browse the web. We can write to our friend about a web page as we view it. If

we discover a new site from our friend's post, the comments remain visible as we view the new web page. It is particularly engaging if we can continue the conversation on the side as we view live content.

Preliminary experience with SocialBar suggests that it can be addictive. We can discover many more interesting websites to explore; it is easy to start conversations and invite new people to join an ongoing thread of conversation. The response is real time enough that messages just keep flying by. Even though we are essentially just writing emails to each other, eliminating the friction involved completely changes the experience. We believe that social browsing will become a common practice.

Another important advantage of SocialBar is that we can socialize on any web page. For example, academicians can discuss the content of research websites, which normally do not have any integrated social features. We can even leave notes to our friends, and ourselves for that matter, on files of a web-accessible file repository!

SocialBar messages are meaningful to our friends even if they are not using SocialBar. To use SocialBar requires little commitment, unlike other social networking systems that require membership. They only need to click on a link in the message to download the SocialBar extension and the Firefox browser, if they are not already using Firefox. In the future, we hope that SocialBar will be available by default on all browsers.

Finally, the privacy provided by SocialBar allows us to share our personal likes and dislikes with people we trust, and to privately share our views on religion, politics or other sensitive topics without repercussions. Considering the number of hours we spend browsing, a privacy-honoring social browser can make an impact in slowing the erosion of privacy we see today.

4. OPEN, FEDERATED IN SITU SOCIAL NETWORKING

In this section, we discuss the relationships of Mr. Privacy's with email and other social networking infrastructures.

1. By embedding email protocols into social applications, we wish to revive the popularity of email identities among younger people, while making social networking open and federated.

2. Mr. Privacy does not preclude our participation in other social networks. In fact, its success rests on being inter-operable. We show how this can be achieved by leveraging other efforts in aggregating social media.

3. Now that we have a safe haven for sharing privately, we can easily participate selectively in public aggregation of social data if we so wish. Going the opposite direction is much harder; once we give up ownership of our data, our privacy is subject to the owner's privacy policies which can change any time.

4.1 In Situ Social Applications Versus Email

4.1.1 Email Provider Incentives

Email is losing its appeal to the younger generation, who are communicating more and more on Facebook and Twitter. Nielson reported that an American spends on average 3 hours a month on email and 4 hours on social networking in June of 2009. In June 2010, it is 2 hours of email and 6 hours of social networking. We believe that without major changes, email usage will continue to decrease, and may reach the point where it is used primarily only for longer conversations. Today, many Americans don't read physical mail that frequently, there is a chance that email as we know it will go the way physical mail did! If email is only used for work, the opportunity for consumer marketing is greatly reduced.

Various attempts have been made to revive webmail by changing its format. We believe the answer, however, is to embed email in social applications, which can have different and better rules of engagement, as discussed below. Even though the user may not be viewing the mails directly, the structured data held in these messages contain valuable profile information like location information. We believe this more than compensates for direct advertising revenues and is much preferred over losing the email audience to social networking websites.

4.1.2 Email Identities and Services

Using email not only allows Mr. Privacy to leverage the installed user base and mature infrastructure but also the various services offered around email identities. OpenID lets us log in to many web services using our email identities without having to create a new account [23]. Webfinger lets us attach public metadata to our email accounts [28]. The information is stored with our email providers so no single server monitors who is retrieving the information. It does, however, require our email providers run the webfinger service. Consumers can, on our own, choose to associate their email address with an avatar using Gravatar [15], which may or may not bear a resemblance to us. Users can have multiple email accounts to prevent others from linking activities of our different personas through avatars.

Finally, Mr. Privacy follows a recent trend where services would send a record of our online transactions to our email accounts. Individuals also routinely send mail to themselves so they can access it from anywhere. In a sense, we observe that since we store data in our inboxes, we might as well have applications that can read email messages and present them in an intuitive fashion.

4.1.3 White Listing to Remove Spam

Spam has made email unpleasant for many people; here we are referring to not just advertisements and scams, but also mail from people whom we do not care to hear from. In email, by default, anyone can message another person. On social networking sites, typically, a user must first request friendship before further communication is allowed. A friend request, which cannot be accompanied by ads, has proved to be an effective white-list mechanism in reducing the amount of spam in social networks. Moreover, the social network can remove user accounts if they sent out too many rejected requests. One unfortunate downside, however, is that users

are pressured to accept friends because it is awkward to refuse friending people you know.

Another significant difference between email and social networks is that users are expected to read all messages sent to them by their friends. In social networking, however, friends have no obligation to read what you write on your wall. Individuals are, as a result, uninhibited in broadcasting their thoughts. This liberation is part of the reason why personal walls flourish.

In Mr. Privacy, we do not have to explicitly define people as friends or not friends, so we do not create any pressure for individuals to accept friends. We recommend that friends be classified into three kinds: accepted friends, rejected friends, and everybody else. Messages from accepted friends are displayed; messages from rejected friends are automatically deleted; all other messages are held up until the user explicitly decides whether to accept or reject the sender. The most important part is that the sender has no knowledge whether he has been rejected or not.

4.1.4 Access Control with In Situ Groups
One advantage of Mr. Privacy is that we do not have to give our friends' contact information away (like Evite) or to have all our friends join the same proprietary network (like Facebook) before you can interact with them. Because Mr. Privacy is private, users should feel free to pool all their information together. This will create a collection of data that is more valuable than ever before; this data is valuable not just to advertisers but ourselves. One particular area of interest is how we can use this information to help users in creating privacy settings. It has been found, for example, that very few people in Facebook are using the groups feature to help them manage their conversations because specifying groups is tedious.

We can greatly reduce the friction in specifying groups by inferring them automatically as we interact [19]. Specifically, we have developed an algorithm that automatically infers a *social topology* from one's emails or tags in photos. A social topology is a map of relationships consisting of possibly overlapping and nested groups. Allowing overlaps is important because friends may play multiple roles; allowing nesting is important because some friends are closer than others. This algorithm analyzes co-recipients of emails or co-appearances in tagged photos, reduces noise in the data, and creates supersets and subsets if necessary to summarize the relationships. In a user study, users found it easier to create groups when presented with the social topology derived from their data. Mr. Privacy allows each person's social interactions be collected in one place and the social topology derived can be fed back to all the applications.

4.2 Federation Across Other Social Networks
For the success of Mr. Privacy, it is very important that it can inter-operate with our existing social networks. We can leverage existing media aggregation efforts to achieve inter-operability. For example, the Firefox add-on "Mozilla Contacts" provides three services: (1) aggregate social data across different providers to provide a consistent API, (2) mediate the flow of information from providers to web applications, and (3) manage the information flow from the

web applications back to their respective providers [21]. Providers currently supported by the Contacts application include Facebook, Twitter, GMail, Yahoo, Last.fm, LinkedIn, and Plaxo. It also supports HCard discovery, as well as probes into Flickr and Gravatar.

Mr. Privacy complements this effort by providing an alternative, open and federated social data source to Mozilla Contacts. Mozilla Contacts can integrate all the friends from Mr. Privacy with those in other networks. Instead of having SocialBar call Mr. Privacy's API directly, it can use social-network agnostic API in Mozilla Contacts to access the unified friends' list, to send and receive information from their friends, etc. To get updates from a friend, for example, Mozilla Contacts will contact Facebook if the friend is on Facebook, and Mr. Privacy if he is a Mr. Privacy user. In this way, a user can interact with his friends in the same way regardless of which network his friends are on.

4.3 Public Social Interactions
Once we have a private haven in which to interact with our friends, it is relatively easy to also engage with the public as well. For example, we can selectively make certain parts of our interactions public, we can reveal our true identity or engage anonymously.

For example, SocialBar is inspired by LinkPong, which is a centralized website that lets users share links publicly and with their friends [18]. One must join LinkPong before he can share a link. Any link shared with a friendis also sent to the friend via email. LinkPong sees all the links that are shared between users, providing them a chance to collect valuable user profile information.

SocialBar can integrate with LinkPong by letting users opt in to share information with LinkPong and tap into LinkPong's public feed as well. Users can decide which links and comments to share publicly, on a link-by-link basis if they wish. Users can also choose to provide a different persona when they publish. SocialBar can also submitted aggregated statistics instead. Finally, it is even possible for advertisers to provide incentives for users to sell their own profile data.

5. PROTOTYPE DESIGN AND EXPERIENCE
In this section, we describe our implementation and design for the prototype systems we built and describe our experience. We start by describing Jinzora Mobile, the third application we built on top of Mr. Privacy. We then describe our Mr. Privacy implementation on mobile platforms. Next we describe the SocialBar application and its implementation as an extension to the Firefox browser. Finally, we discuss the lessons learned in implementing Mr. Privacy on top of existing email protocols.

5.1 Social Music with Jinzora Mobile
Today's many so-called social music sharing sites suggest songs that are similar to the songs we play, or songs that others with similar profiles would play. We find listening to similar songs over and over again, or songs by people with similar profiles, not particularly appealing nor social.
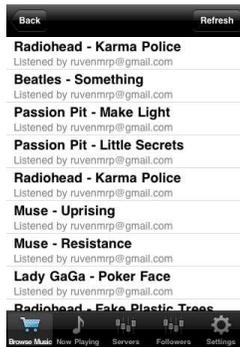
Figure 5: A shared playlist on Jinzora Mobile on iOS

Instead, we wish to discover music by finding out what our friends play so we can discuss the artists with them or invite them to concerts, etc.

We believe that such a social music sharing service should be federated because our friends all use different music services. As a prototype, we added federated music sharing to Jinzora Mobile, an open-source music application on the iPhone that allows users to stream music from their PCs[17]. We used Mr. Privacy to add the ability to share playlists between friends regardless of the servers the users use.

Jinzora Mobile allows a user to configure Mr. Privacy play sharing. A user provides the application with his emails credentials and the addresses of his friends. Then at 30 seconds into each song, Jinzora Mobile sends a message to the registered friends that the song is being played. Users can now select a new option "Recently Played by Friends", which uses a Mr. Privacy receive command to fetch the list of music their friends have played. Users can view these plays, as shown in Fig. 5, and tap on them to start listening. By using a standard playlist format, other music players can also be built upon Mr. Privacy to share playlists across all the music services. It takes only a few lines of code to integrate with Mr. Privacy.

5.2 Mr. Privacy on Mobile Platforms

We implemented the Mr. Privacy platform for the Android and iOS operating systems in two different ways. The Android implementation of the Mr. Privacy platform was built on the JavaMail framework, which connects directly to a mail server directly via SMTP and IMAP. The Android platform allows for a clean and complete implementation of Mr. Privacy. Unfortunately, the iOS platform disallows local services, thus presenting a challenge for Mr. Privacy. We had to create a HTTP service that acted as a generic gateway to a user's IMAP and SMTP servers. In the ideal iPhone implementation, mail providers would host an alternative HTTP interface to the social data in users' mail accounts.

5.3 SocialBar Implementation

The SocialBar is constructed as a Mozilla Firefox extension, as illustrated in Fig. 6. Our extension consists of the metadata required to register it as a sidebar, a Javascript Mr. Privacy library, and the SocialBar itself implemented as Javascript and HTML. The Mr. Privacy library implements
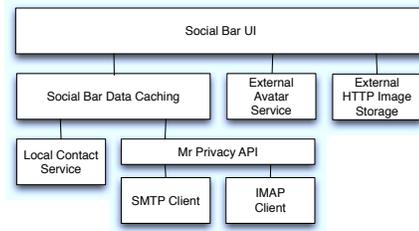

Figure 6: SocialBar and Mr. Privacy as a Firefox extension.

the IMAP and SMTP protocols in order to provide message transport for the SocialBar. The Mr. Privacy library needs to communicate using the IMAP and SMTP protocols which are normally layered over standard TCP/SSL sockets. This would be problematic in a normal web application because native socket APIs are not available. Fortunately, Firefox allows browser extension to create socket connections via Javascript objects bound to internal browser objects.

The implementation of the user interface for the SocialBar is implemented using the jQuery Javascript library and a bit of structural HTML. An internal data caching layer handles all of the interactions with the outside world. The caching layer combines information from two sources: Mr. Privacy and the Mozilla Contacts extension. The caching layer is populated with data from its sources when the user logs in to the SocialBar. The UI then constructs an HTML document that represents the messages delivered by Mr. Privacy. The avatar images and inlined content in the SocialBar are fetched by the web browser when it renders the user interface.

When a user shares items in the SocialBar, the human readable portions of the Mr. Privacy message contain a serialized copy of the conversation thread. This lets email users fully appreciate the context of the discussion. The message also includes SocialBar's JSON object which has the following fields.

- id: Unique message ID generated by sender.
- url: The URL of the item being shared
- title: The title selected by the link sharer.
- comment: The comment specified by a commenter or sharer.
- content-type: The MIME type of the content at the specified url, e.g. text/html, image/jpeg

The SocialBar keeps a local copy of all of the messages delivered by Mr. Privacy. This facilitates local filtering operations and a responsive user interface. Whenever an item is shared, a copy is sent to the user in addition to all of the recipients. This is how the SocialBar is able to present a complete communication history when it loads data from Mr. Privacy. Unfortunately, messages dispatched by Mr. Privacy over the SMTP connection do not immediately return over its IMAP connection. Because this delay can be up to a minute, it is highly desirable to not make the user wait for the message to return before displaying it in the

UI. After Mr. Privacy reports that a message was sent, the SocialBar immediately inserts the shared item into its local cache. When the message is eventually delivered by the IMAP server via Mr. Privacy, the SocialBar detects that it already has the message by inspecting the id field inside the JSON object.

The SocialBar also keeps a store of friends involved in Mr. Privacy communications. The friend database is used by the UI to provide automatic completion of email addresses when sharing. We also integrated the Mozilla Contacts extension with the SocialBar, so the local friend store provides the merged view of the friend information from the two sources. In addition to email addresses and names, the friend database keeps track of avatar images for each friend.

5.4 Mail Protocol for Social Applications

So far, we have described how email is useful for implementing open and federated social networking infrastructures, it also has its disadvantages. Typical social networking applications are built on top of a centralized database. Here, we cannot change the servers at all; all application-specific functionality must be provide on the clients; the client code must communicate with the server using IMAP and SMTP protocols; the IMAP protocol limits the flexibility of how data can be organized and searched on the server. This section describes our experience; we found that the mail protocols are adequate for the kind of social applications we implemented, but also uncovered many subtleties about using mail protocols for this purpose.

5.4.1 Experience with the IMAP Protocol

IMAP provides a simple folder-oriented storage system for messages. It provides a monotonically increasing unique ID number for new messages that arrive within a particular folder. This ID is used by the client to support local synchronization of the contents of an IMAP folder. As discussed above, Mr. Privacy moves all Mr. Privacy messages into an isolated folder to reduce inbox clutter. Other than that, it supports listing existing messages and waiting for new messages. The implementation of Mr. Privacy relies on the following IMAP primitives:

- List messages in a folder starting with a particular unique id number and having a subject header that matches a certain Mr. Privacy application tag.
- Fetch certain parts of messages with a specified set of unique id numbers.
- Wait for new messages to arrive in a folder.
- Create a new folder.
- Copy messages from one folder to another.
- Delete messages in a folder after copying them to simulate a move operation.

We summarize the lessons we learned in using IMAP to implement Mr. Privacy below:

1. *Lack of universal IMAP support.*

   Many of the free mail servers available to the consumers do not support IMAP. Google Gmail and AOL mail support IMAP, but not Microsoft Hotmail or Yahoo Mail. If Mr. Privacy proves to be widely used,

perhaps more providers will make IMAP available to attract the precious profile information embedded in Mr. Privacy mail. Many university and company mail services do support IMAP to minimize load on their servers.

2. *Server-side modifications to support email functions.*

   We had originally assumed that Mr. Privacy messages received would be identical to the ones sent. To our surprise, this turned out not to be the case, for example, when mails were sent to mailing lists. We found that some mailing list servers would embed the entire message in another MIME container so as to append a footer to the message, thus changing the MIME structure. In the Mr. Privacy scheme, there are three MIME parts: text/plain, application/JSON, and text/html. Mr. Privacy is only concerned with retrieving the JSON part of the message, so the original implementation only retrieved the desired part. Mr. Privacy now compensates for the potential structure modification. There may be other ways in which the server may change the message structure. For more robustness but at a performance degradation, Mr. Privacy would need to download entire messages.

3. *Basic IMAP does not provide the full gamut of features required. Existing extensions are useful but not necessarily implemented by all servers.*

   Mr. Privacy provides alternative implementations to handle the variations in IMAP support. For example, Mr. Privacy takes advantage of the IMAP extension called "IDLE" so it can be alerted when new messages arrive in a particular folder. If the extension is not implemented by the server Mr. Privacy connects to, it reports to the application that it cannot wait for new messages. Upon getting this information, SocialBar would present the user with a button that explicitly triggers a refresh. Similarly, Mr. Privacy uses the IMAP "CREATE" command to make a new folder on the server. Some mail services, notably AOL mail, do not support this. If Mr. Privacy cannot create a folder to hide its messages in, it simply disables its inbox clutter avoidance technique.

4. *Federated mail systems may have standard protocols, but individual implementations may have different performance characteristics.*

   Mr. Privacy relies on existing servers and therefore must work with existing implementations. It is important that we take advantage of the techniques that servers use to optimize typical email usage. For example, Mr. Privacy takes advantage of server-side filtering. The first technique we explored was adding Mr. Privacy tags to message headers. This worked well for small test accounts, but would take minutes on accounts with greater than 10,000 messages. Instead, we now tag Mr. Privacy messages by adding "Mr. Privacy" to the subject, and uses subject search in IMAP to retrieve relevant messages. Existing mail server implementations appear to have an index dedicated to accelerating subject searches because we found that using this search filter was dramatically faster, seconds vs minutes of search time on large inboxes.

5. *The semantic gap between Mr. Privacy operations and IMAP provides some challenges to a seamless experience.*

   To prevent cluttering users' inbox, we wish to place all Mr. Privacy messages in an isolated folder. Unfortunately, IMAP does not provide a standard way to move messages to another folder. So Mr. Privacy must implement this operation with a copy followed by a delete operation. The delete operation is implemented by first marking the messages for deletion, followed by an expunge operation. It, unfortunately, may have the undesirable side effect of deleting any other messages that are marked for deletion by a different mail client. Thus, we have made folder isolation an option that can be disabled.

In summary, we have to make compromises in our design and implementation because we are using a legacy protocol. We found, however, that we were able to provide sufficient functionality for the kind of social applications we have built. The results from this experience can hopefully lead to improvements in mail services provided as well as the design of the IMAP protocol in the future for the support of open and federated social networking.

5.4.2   Experience with the SMTP Protocol
The SMTP protocol is extremely simple because there is only one logical command, send a message. To send a message, Mr. Privacy connects, authenticates, provides the name of the send, a list of recipients and the header and body of the message.

Developing Mr. Privacy uncovered a few implementation hurdles, all of which are manageable. For example, SMTP does not having any intrinsic support for MIME, so Mr. Privacy must generate a suitable multipart message from the data provided by the SocialBar. Since SMTP was invented a long time ago, we need to address its weaknesses, for example, in its handling of international characters. Finally, SMTP servers like to drop connections after a very short timeout. To deal with this, Mr. Privacy reconnects to the SMTP automatically when a message needs to be transmitted.

6.   RELATED WORK
Social API Abstraction. One approach to opening up social networking infrastructure is to define a common set of APIs that can be implemented on top of existing networks. Google's Open Social [13] provides this allowing any developer to create a social application once and deploy it on different social networks. This does not allow the users to interact across networks, it only reduces the development effort for supporting multiple social data providers. Mozilla's Contacts [21] also abstracts access to existing networks, but it provides an API at the browser level that any web page can use to access personal social data. The latest Contacts explores services integration beyond what is outlined in the W3C Contacts API specification [27].

Distributed Social Networking. At the far extreme from services like Facebook are the distributed social networking "clients". Projects like Diaspora [16] and Appleseed [2] define a P2P protocol that allows social features to be provided. Conceptually, each user runs a server for their data, but these servers can be aggregated so that a trusted host can provide a data repository for multiple users. PeerSoN [3] explores building social functionality on top of distributed storage, such as OpenDHT.

Federated Social Networking. OneSocialWeb [4] has designed a social infrastructure built off of the federated XMPP instant messaging protocol. The project provides extension specifications for XMPP enable social networking by adding server side storage and access control. OneSocialWeb is still in the early stages and is primarily developing the contacts, wall, and friends functionality with future plans to allow for applications to be built on the network. Google's Wave [14] is another social service built on top of the XMPP messaging protocol. Although it was not deployed by any provider other than Google, it defined extensions to XMPP that enable users to have rich discussion threads across providers.

Social Browsing. Social curation of web data is already a rich area of development. Social bookmarking service Delicious [9] provides an easy way for people to view curated content. Another services that facilitates content discovery through a browser extension is StumbleUpon. Like Delicious, StumbleUpon [25] relies on a central service to collate browsing data and make recommendations. Friends are moving into the browser with the Flock [10] and Cruz [8] social web browsers. They incorporate sidebars that connect with proprietary social networks to aggregate social content for the user and allow for communication during a browsing session. Google has also produced a social browsing companion called Sidewiki [12]. It allows users to view public comments about a visible website from a browser side bar.

Alternative Email. Semantic Email [20] explores how a better user interface can be presented on top of existing email infrastructure. Event planning and organization are particularly time consuming for humans to organize though normal email. This project explores how a management agent can orchestrate sending, receiving, and reprocessing messages in order to simplify these tasks for the end user.

7.   SUMMARY AND CONCLUSIONS
Because of network effects, there is a nontrivial possibility that a single proprietary company may own the majority of the world's social graph. To allow for competition, a prerequisite to better privacy protection, we call for the creation of an open and federated social networking platform. Just having an alternative would pressure centralized service providers to give better privacy guarantees.

It is futile to create a decentralized platform that simply duplicates a well-entrenched social networking experience. Looking forward, we propose to create an *in situ* social networking platform that seamlessly embeds our contacts in every social application we run. The Mr. Privacy platform we propose relies on having common services available to applications on every platform (mobile phone operating systems and browsers on PCs). Mr. Privacy has the potential to reach millions of people quickly because it is built on top of the open and federated email platform. We believe that

Mr. Privacy is the first proposal that has a chance to challenge the status quo within the next few years. We need one or more killer applications developed using Mr. Privacy to help jump start this model.

This paper serves primarily to introduce Mr. Privacy and to demonstrate the feasibility of such an approach. We have created three compelling applications on three different platforms (Android, iPhone, Firebox). We can now share our GPS locations with our friends, share our playlists, and browse together, all without a single third-party company monitoring all our activities. With the help of social aggregators, Mr. Privacy can inter-operate with other complementary social networking services that provide more public interaction features.

The power of Mr. Privacy lies in its enablement of the development community to create new and compelling applications, without the constraints imposed by the owners of closed social networks. Furthermore, developers need not host their own servers, which is particularly useful for applications that serve long-tail needs. However, much work remains to bring that vision to fruition; examples include standardization of the representation of common data, authenticating emails to prevent spam, and providing access control to guard against malware and unwanted interactions.

8. REFERENCES

[1] ABC news. Microsoft deepens facebook ties in web search battle. http://abcnews.go.com/Business/wireStory?id=11876359.
[2] The appleseed project, 2010. http://opensource.appleseedproject.org/.
[3] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta. Peerson: P2p social networking: early experiences and insights. In *SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 46–52, New York, NY, USA, 2009. ACM.
[4] D. Cheng and O. Griffin. Onesocialweb, 2010. http://onesocialweb.org/.
[5] Cnet. Steve jobs on why no facebook for ping, 2010. http://news.cnet.com/8301-13579_3-20015402-37.html.
[6] M. Crispin. Internet message access protocol - version 4rev1, 1996.
[7] D. Crockford. The application/json media type for javascript object notation (json). Technical Report RFC 4627, IETF, July 2006.
[8] Cruz social browser, 2010. http://cruzapp.com/.
[9] Delicious add-on, 2010. http://www.delicious.com/help/quicktour/firefox.
[10] Flock sidebar, 2010. http://beta.flock.com/sidebar/.
[11] Fox News. 5 ways to stay safe on facebook, 2010. http://www.foxnews.com/scitech/2010/10/23/ways-stay-safe-facebook/.
[12] Google. Sidewiki. http://www.google.com/sidewiki/intl/en/learnmore.html.
[13] Google. Opensocial, 2010. http://code.google.com/apis/opensocial/.
[14] Google. Wave, 2010. wave.google.com/.
[15] Gravatar, 2010. http://gravatar.com/.
[16] D. Grippi, M. Salzberg, R. Sofaer, and I. Zhitomirskiy. Diaspora, 2010. http://www.joindiaspora.com/.
[17] Jinzora, 2010. http://jinzora.com/.
[18] LinkPong, 2010. http://www.linkpong.com/.
[19] D. MacLean, S. Hangal, S. K. Teh, M. S. Lam, and J. Heer. Social flows: A system for mining social topologies from ego-centric social networks. In *16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (Demo)*, 2010.
[20] L. McDowell, O. Etzioni, A. Halevy, and H. Levy. Semantic email. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 244–254, New York, NY, USA, 2004. ACM.
[21] Mozilla. Mozilla contacts, 2010. https://mozillalabs.com/blog/2010/03/contacts-in-the-browser/.
[22] J. Postel. Simple mail transfer protocol, 1982.
[23] D. Recordon and D. Reed. Openid 2.0: A platform for user-centric identity management. In *DIM '06: Proceedings of the Second ACM Workshop on Digital Identity Management*, pages 11–16, 2006.
[24] The Register. Burglars used social network status updates to select victims, 2010. http://www.theregister.co.uk/2010/09/13/social_network_burglary_gang/.
[25] Stumbleupon, 2010. http://www.stumbleupon.com/.
[26] TechCrunch. Five months in, 2 million websites using facebook's new social plugins, 2010. http://techcrunch.com/2010/09/29/five-months-in-2-million-miscs-using-facebooks-new-social-
[27] W3. Contacts draft, 2010. http://www.w3.org/TR/2010/WD-contacts-api-20100121/.
[28] Webfinger, 2010. http://code.google.com/p/webfinger/.