PrPl
A Decentralized Social Networking Infrastructure

Seok-Won Seong

Jiwon Seo, Matthew Nasielski, Debangsu Sengupta,
Sudheendra Hangal, Seng Keat Teh, Ruven Chu, Ben Dodson,
Monica S. Lam

Stanford University
06/15/2010

Outline

- Motivation
- PrPl platform
  - Personal Cloud Butler
  - Federated ID management
  - Semantic index
  - SociaLite
- Implementation
- Preliminary evaluation
- Conclusion

Social Networking Today

Facebook's new 'privacy' policy catches users off guard

A new change in Facebook's privacy policy allows it to share information automatically with third party websites without users opting in

By Varun Aggarwal, InformationWeek, April 21, 2010

Canada, Britain, France, Germany, Ireland, Israel, Italy, Netherlands, New Zealand and Spain slam Google over privacy

AFP AFP April 21, 2010 8:13AM   Share

September 13, 2008 4:58 PM PDT

Apple blocks competitive products from iPhone App Store--surprised?

May 24, 2010 9:03 AM PDT

Twitter cuts the cord on third-party ad networks

While Facebook & Twitter Sit on Sidelines, MySpace Jumps Into Bulk User Data Sales

Written by Marshall Kirkpatrick / March 12, 2010 10:30 AM / 22 Comments   « Prior Post   Next Post »

Profit Driven Social Networking

- Loss of data ownership
  - Privacy
  - Data locked-in
  - No interoperability
- Inconvenience
  - Upload for each service
  - Slow uplink
  - Can only share within users in the same service
  - Confusing access control
- Scaling cost
  - Close group of friends and family
  - Storage grows faster than bandwidth

PrPl (Private-Public)

- **Decentralized** across independent administrative domains
  - Keep data privately and still can interact with each other
  - Choose the service to host personal data at home (in a setup box) or free ad-supported services (in the cloud)
  - Take advantage of a myriad of storage services available
- **Open** API for distributed applications
  - Programming interfaces to write and run a social applications that run across different administrative domains
- **Trustworthy** interactions
  - Safe haven for individuals to keep all the personal data without reservation
  - Share items with selected friends without uploading to different websites
  - Enable new applications as all data available in one place

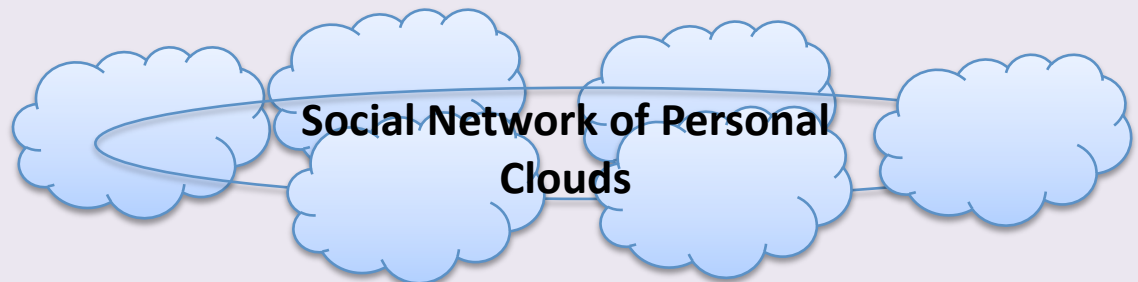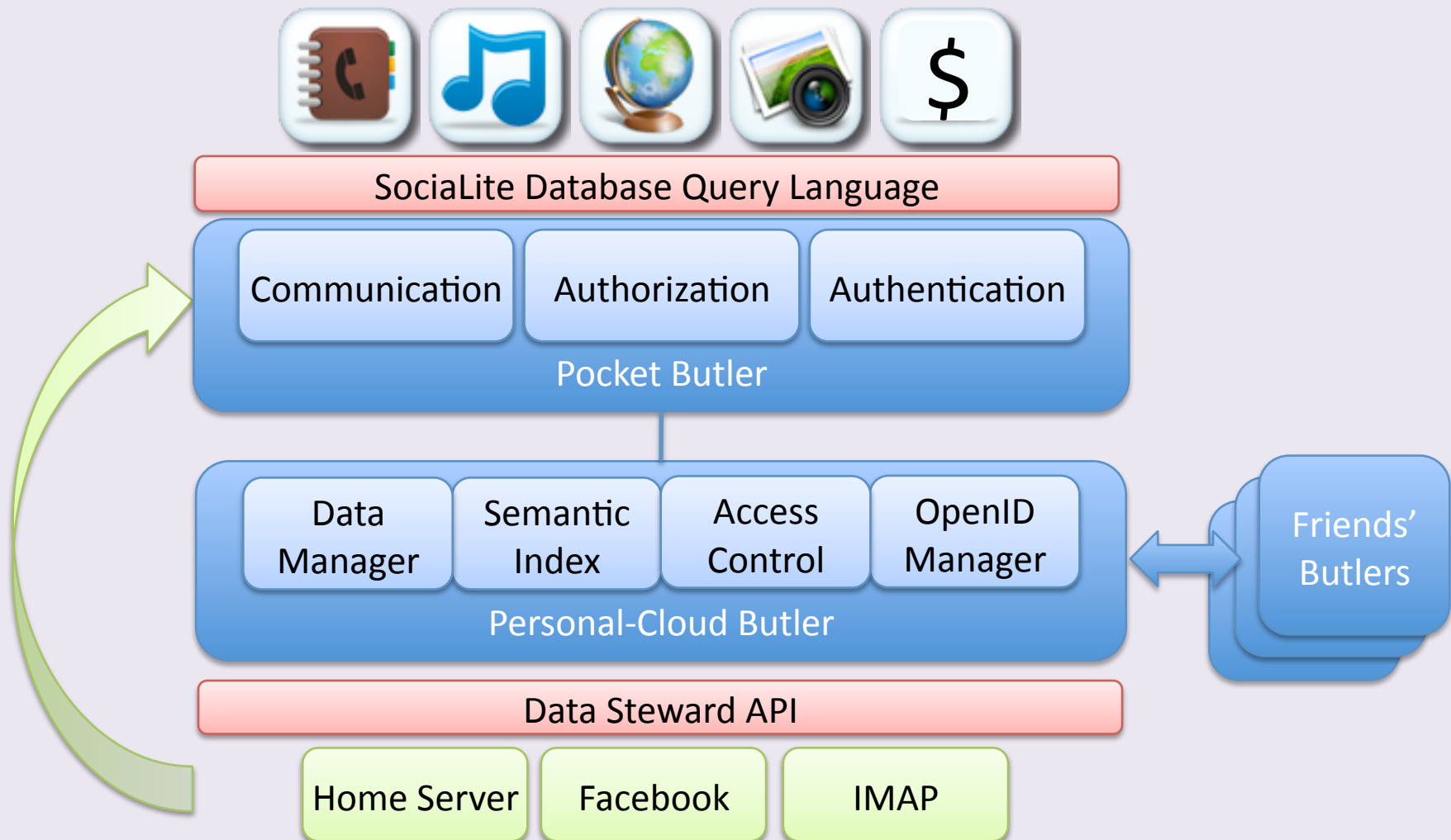# Personal Cloud Butler (Network)

Identities

My Personal Cloud Butler

Social Topology
Calendar
GPS Trace
Credit card history
Email
Phone record

Social Network of Personal Clouds

PrPl Architecture

PrPl Semantic Index

- Unified view of all personal resources (metadata and blob)
- All resource metadata are stored in the index and blobs are stored in federated storages called Data Stewards
- Unstructured database
  - Diverse set of user data
  - Described in RDF (Resource Description Framework)
  - Collection of (sub, pred, obj) triples
- Common ontologies
  - Databases in multiple administrative domains
  - Metadata restriction on resource properties
  - Reasoning and inference
  - Described in Web Ontology Language (OWL)

Data Stewards

- Federated stores that host blobs
- Maps a virtual PrPl resource URI to a physical URI
- At startup, registers with the owner Butler
- Sends heartbeats periodically with device information and data updates
- Serves blobs directly to applications
  - Butler issues a blob ticket after checking user permission
  - Blob ticket contains resource URI, user ID, expiration time, authorized operations, locations of Data Stewards, and signature
  - Application can cache till the expiration time
  - Blob ticket is not renewable

Pocket Butler

- Proxy service running on the client device
  - Provides single sign-on for all the applications running on the same device
  - Handles the underlying authentication and communications with the Butler
  - Caches blobs and query results

Butler Home + Management Console

- A Butler owner can add and host services
- Friends even without a Butler can access shared services and data
- Management console to administer user's cloud information
  - Generic resource browser, add, and edit metadata
  - Download blob
  - Manage friends and group membership
  - Specify access control
  - View registered devices and services
  - View and revoke tickets
  - Run simple queries directly

Federated ID Management

- Decentralized identity management
- Single sign-on
- Reuse existing credentials
- OpenID
  - Open, decentralized, single sign-on standard for ID management
  - Unique URL (http://bob.openid-provider.org)
  - OpenID provider (OP): Service provider offering registering OpenID and authentication
  - Relying party (RP): Site/service wants to verify the user's identification.
  - End-user
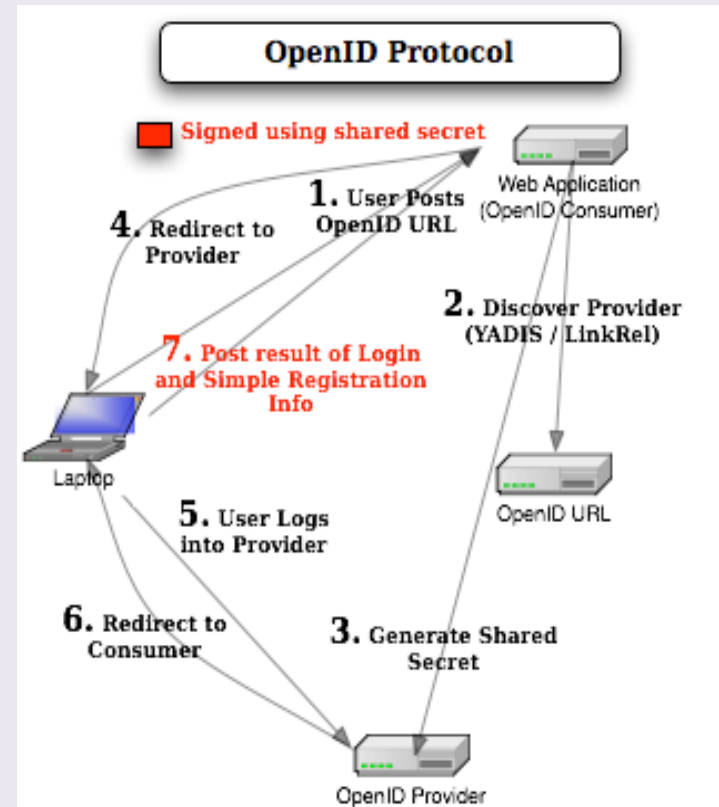
Federated ID Management

- Directory
  - User to Butler lookup, provided by OpenID providers
  - Certificate Authority (CA) in PKI
- Authentication
  1. User authentication at the Butler
  2. Between Butlers to propagate distributed queries
  - A session ticket issued by a Butler as a credential
  - Session ticket includes an issuer ID, requester ID, session ID, expiration time, and signature
  - Session ticket is renewable

User and Butler Authentication

1. OpenID login
2. Get a session ticket from Butler.

Personal Cloud Butler

Directory

Butler grants a blob ticket per resource.

1. Butler self-issues a session ticket
2. Friend verifies with public key

App directly fetches blob from a Steward.

Data Stewards

Friends' Butlers
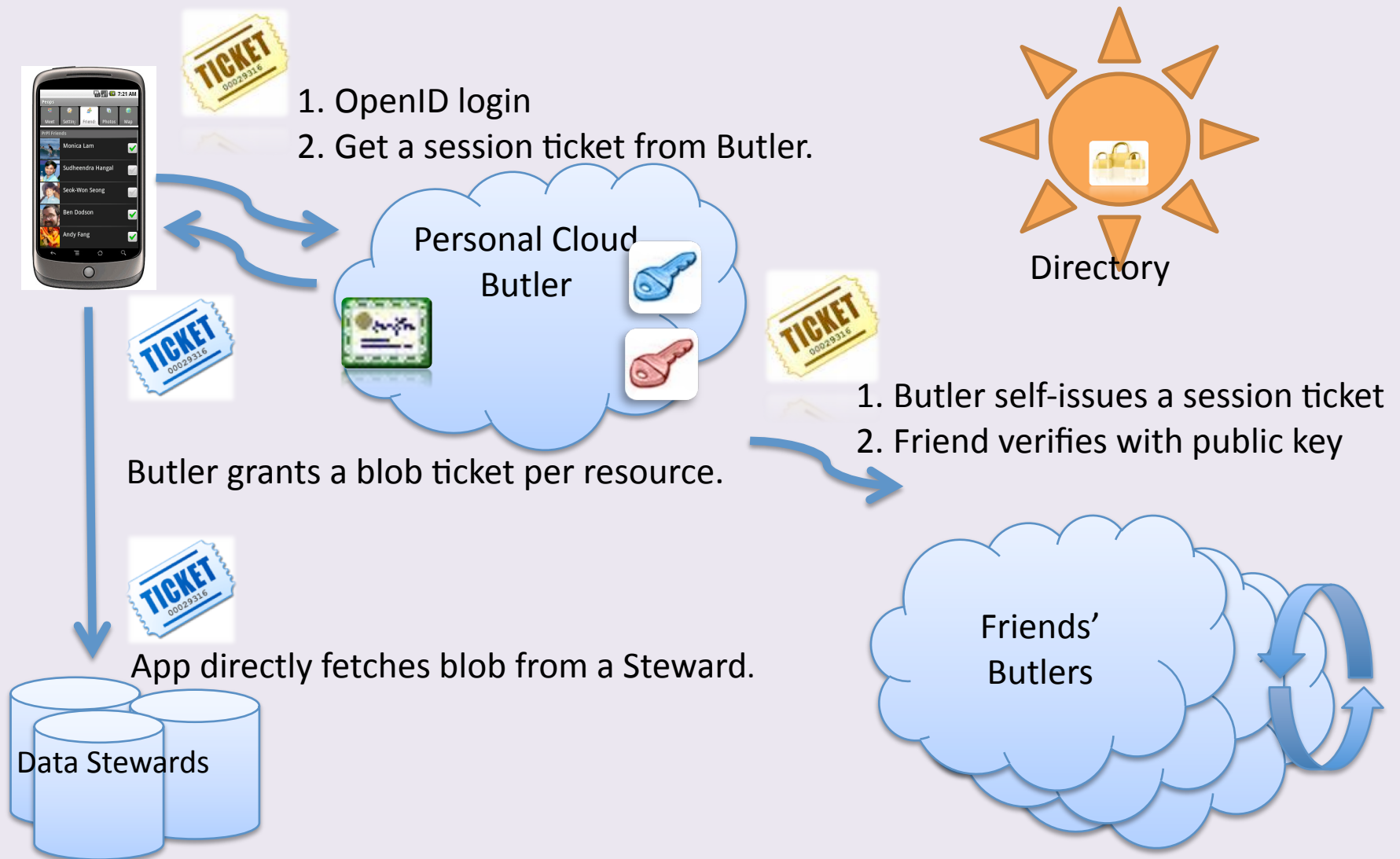
SociaLite

- Measure of success
  - Ease of programming
  - Number of applications
- Difficult to write distributed applications
- Location-agnostic query language for social multi databases
- Based on Datalog, a rule and query language for deductive databases
  - GrandMother(w,y) :- Mother(w,x), Father(x,y).
  - Composition and recursion
- Hides away the complexities associated with databases in multiple administrative domains

SociaLite Extensions

- RDF-Based Database

  > Friend(u) :- (u, type, prpl:Identity)

- Remote Query
  - According-To operator

    > FStar(p) :- Friend(p)
    > FStar(p) :- Friend(x), FStar[x] (p)
    > FStar-CurrLoc (p,l) :- FStar(p), CurrLoc [p] (l)

- Functions
  - Tuple-wise function

    > RecentSongs(s) :- (s, type, prpl:Song),
    >   (s , lastPlayed, date),
    > $WITHIN(date, "5 days").

  - Relation-wise function

    > Top10(s) :- RankedSongs(s), @HEAD(s, 10).
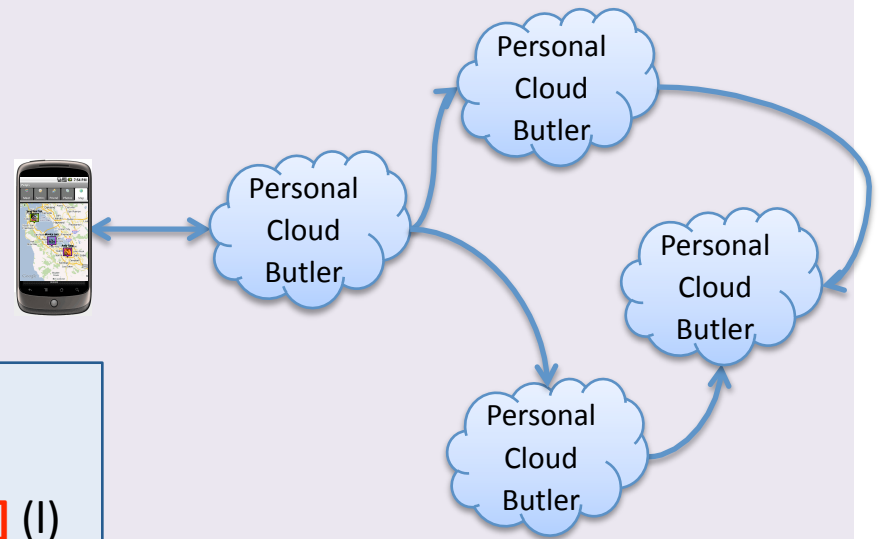
Outline

- Motivation
- PrPl platform
  - Personal Cloud Butler
  - Federated ID management
  - Semantic index
  - SociaLite
- Implementation
- Preliminary evaluation
- Conclusion

Prototype Implementation

- Fully functional prototype of PrPl written in Java (and JSP/Struts web components)
- 34,000 source lines of code (SLOC) including 8,800 SLOC for SociaLite
- XML-RPC and REST APIs over HTTP/HTTPS
- Client API libraries on Java, Android, and iPhone.
- Data Stewards for filesystems, IMAP, Facebook, Google contacts & Picasa photos
- Third-party libraries: HP Jena, OpenID4Java, Apache XML-RPC, Jersey Rest API, JDBM
- PKI: Sun cryptography libraries, Keystore, and OpenSSL
- PrPl Hosting: A run-time platform for application developers

Preliminary Evaluation
Performance on a single Butler

- Experimental prototype with no performance optimization
- Setup
  - 4 indices ranging from 50,000 to 500,000 triples
  - Client and Butler on the same machine
  - Intel Core 2 Duo 2.4 GHz and 4GB of memory
- Queries
  - Find photos that match one and two given tags
- Results
  - First result in less than 1.5s => pipeline
  - Two tag query is uniformly faster => the communication overhead of sending results dominates execution time
  - Commensurate with the results reported for the underlying Jena library

# of Triples	Database Size (MB)	One Tag 1st/Last (sec)	One Tag # of Answers	Two Tags 1st/Last (sec)	Two Tags # of Answers
50,000	8	0.9 / 1.2	1,024	0.6 / 0.6	53
100,000	22	0.9 / 2.0	2,095	0.8 / 0.9	91
250,000	118	1.0 / 3.3	5,045	1.0 / 1.3	264
500,000	628	1.4 / 6.9	10,019	1.5 / 2.4	516

Preliminary Evaluation
Performance on a Network of Butlers

- Setup
  - 100 Personal Cloud Butlers on Amazon EC2
  - 10 to 100 friends, 30 friends for Home Butler
  - Tie strength from [0,1) and consider 0.7 or higher as close friends
  - 50 to 350 photos and 1,000 location data
  - Up to 1,500 songs using a Gaussian dist over the synthetically generated 100,000 song libraries
- Queries
  - 4 distributed queries emulating typical social networking applications

Query	Description	# Rules in Query	# Butlers Reached	Answers	1st/Last (sec)
Common friends	Common friends b/t me and my friends	2	30	494	1.0/1.9
Friends star	All the people connected through my friends	3	100	100	0.7/5.2
Close friends' pictures	URIs of photos from my close friend	3	10	1,642	1.3/3.6
Top 10 songs	Top 10 popular songs among my friends	6	30	10	4.2/4.2

Preliminary Evaluation
Performance on a Network of Butlers

- Results
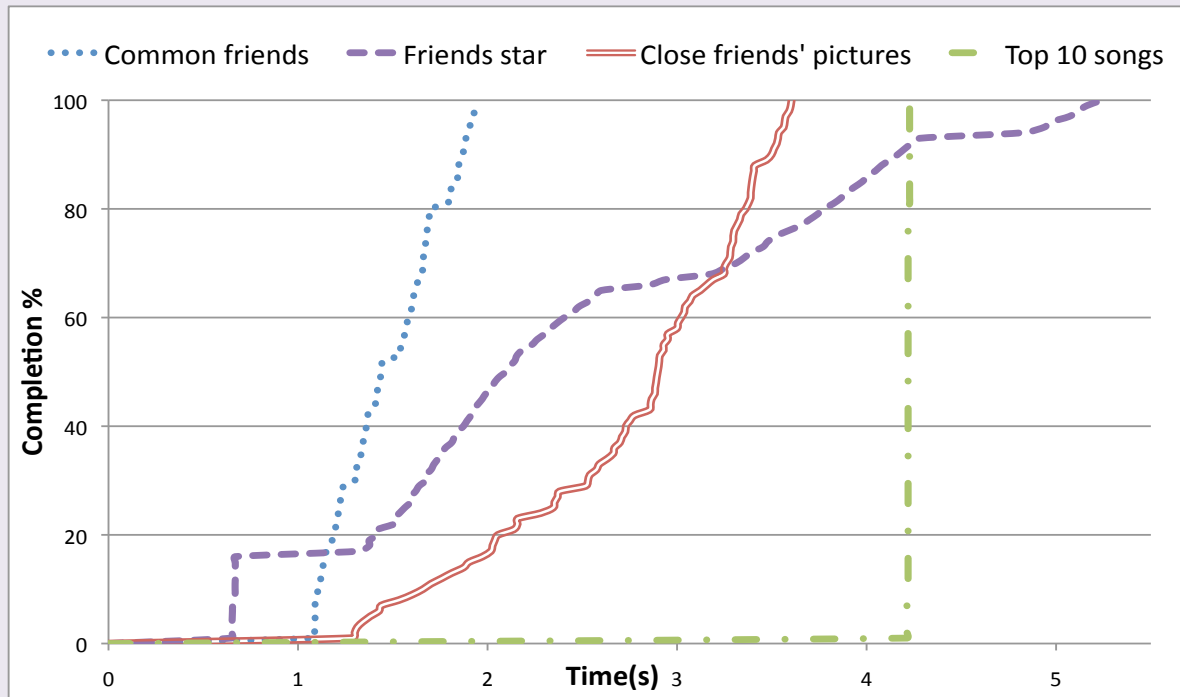
- First result within 1.5 seconds
- All queries finish in 5 seconds
- Each query concisely represented in 2 to 6 SociaLite rules

Applications

Friends list
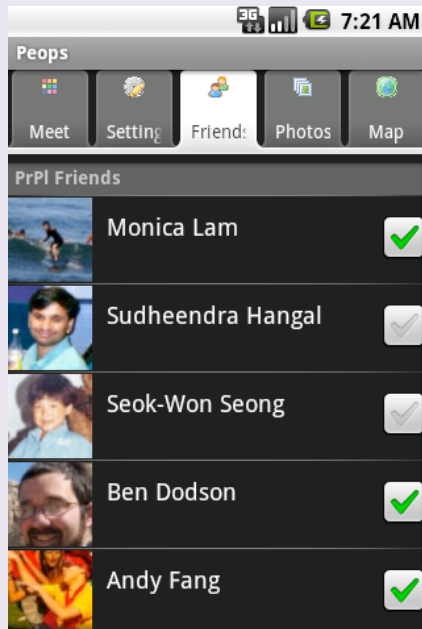
Music play history
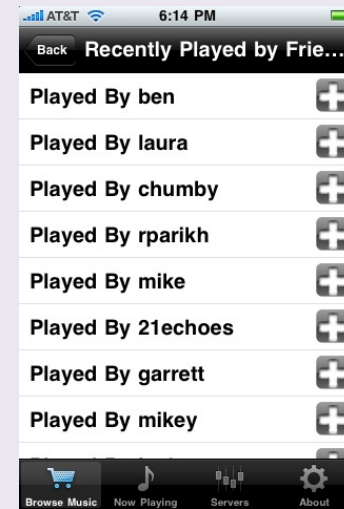
Location history
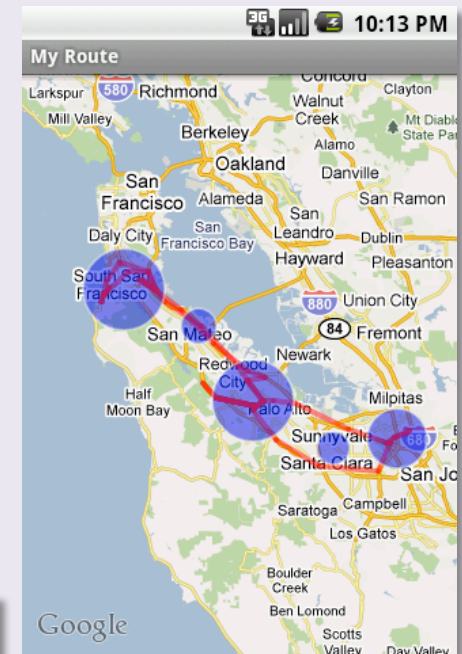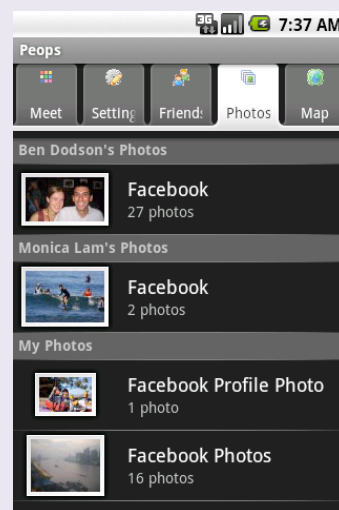
Friends' photos
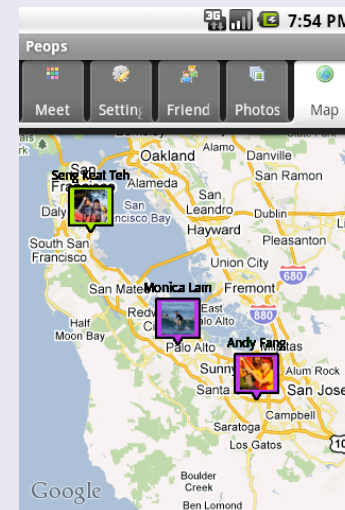
Applications

Butler home showing photo wall app and management interface

Applications

Semantic Photo Browser on TV

Applications

Application	# Queries	# Lines of Code
Unified address book	1	2
Aggregated friends' feed	1	3
Collaborative photo browser	1	3
Chat / messager	1	3
Common interests discovery	2	3
Social charts	2	3
Collaborative slide show	2	3
List of friends	2	9
Photos	1	4
Locations	1	3
Social Shopping	1	10

PrPl makes writing programs simple by hiding the complexity

Conclusion

- PrPl: a decentralized, open and, trustworthy social networking platform
  - Personal Cloud Butler as a safe haven for personal data
  - Unified view with the PrPl semantic index
  - Federated ID management based on OpenID
  - PocketButler and SociaLite API hide the complexity of decentralization from application developers
- Fully functional infrastructure prototype and number of social applications
- Preliminary results on Amazon EC2 with 100 Butlers

Questions ?

- Thank you.