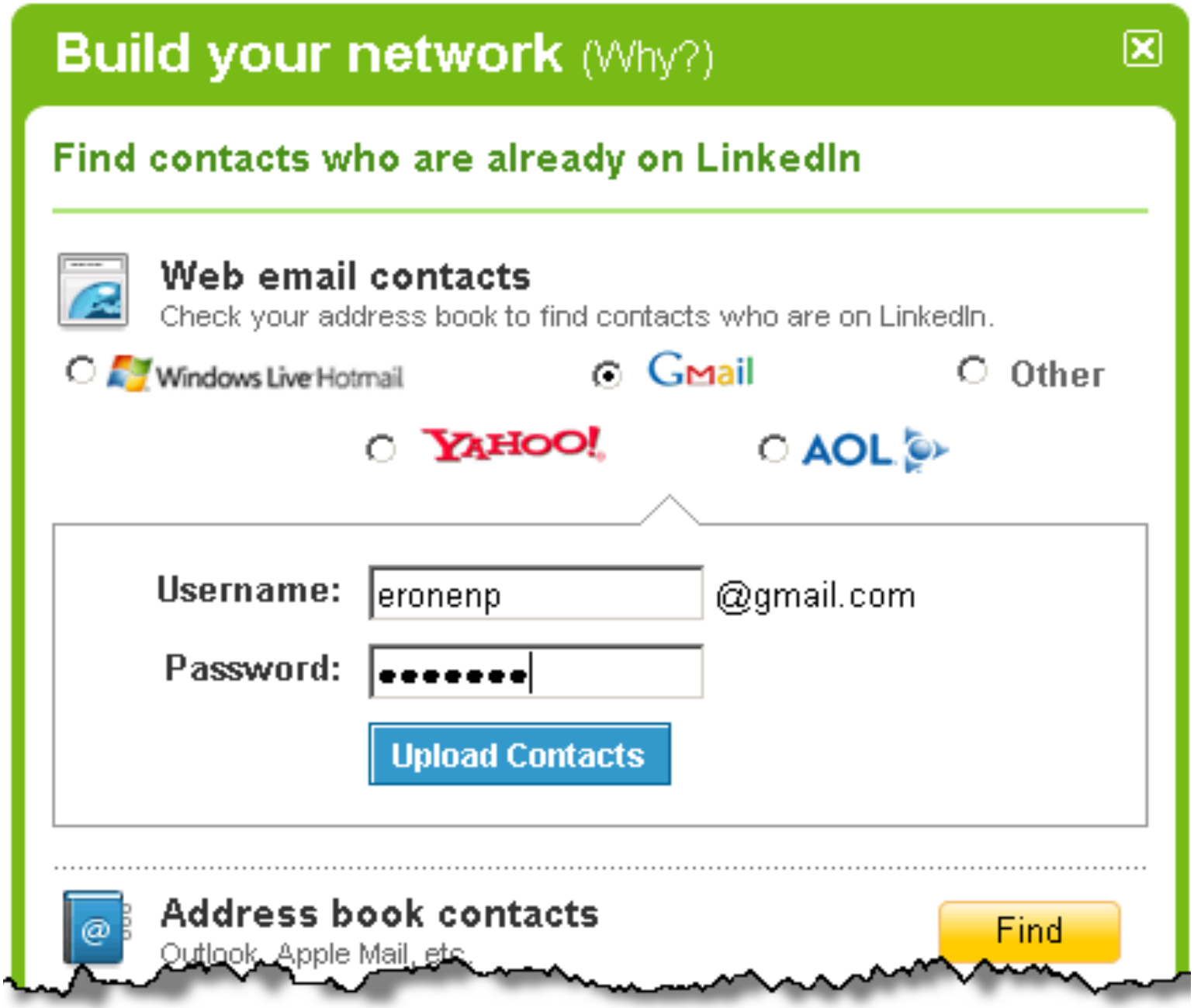
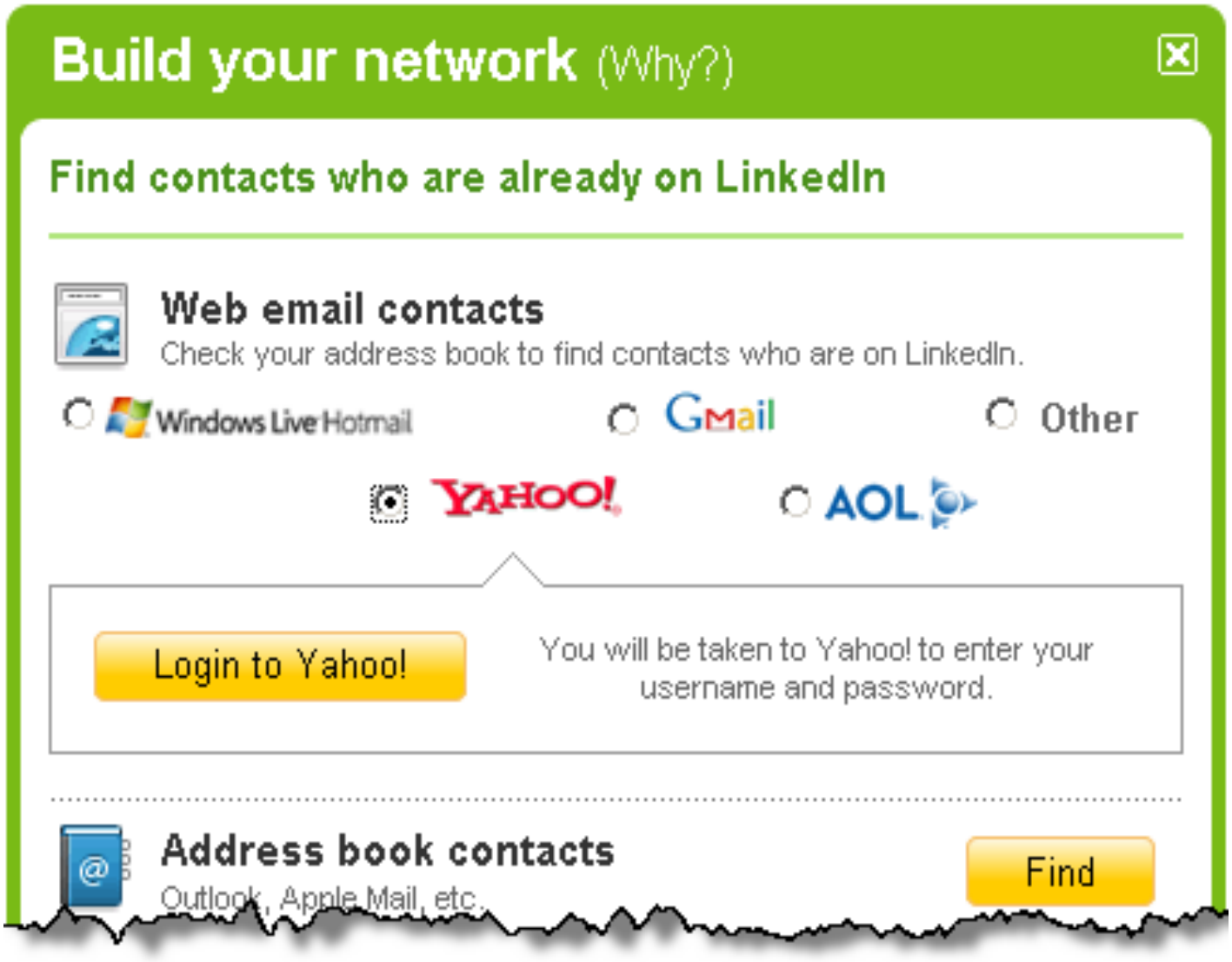
Hannes Tschofenig, Blaine Cook

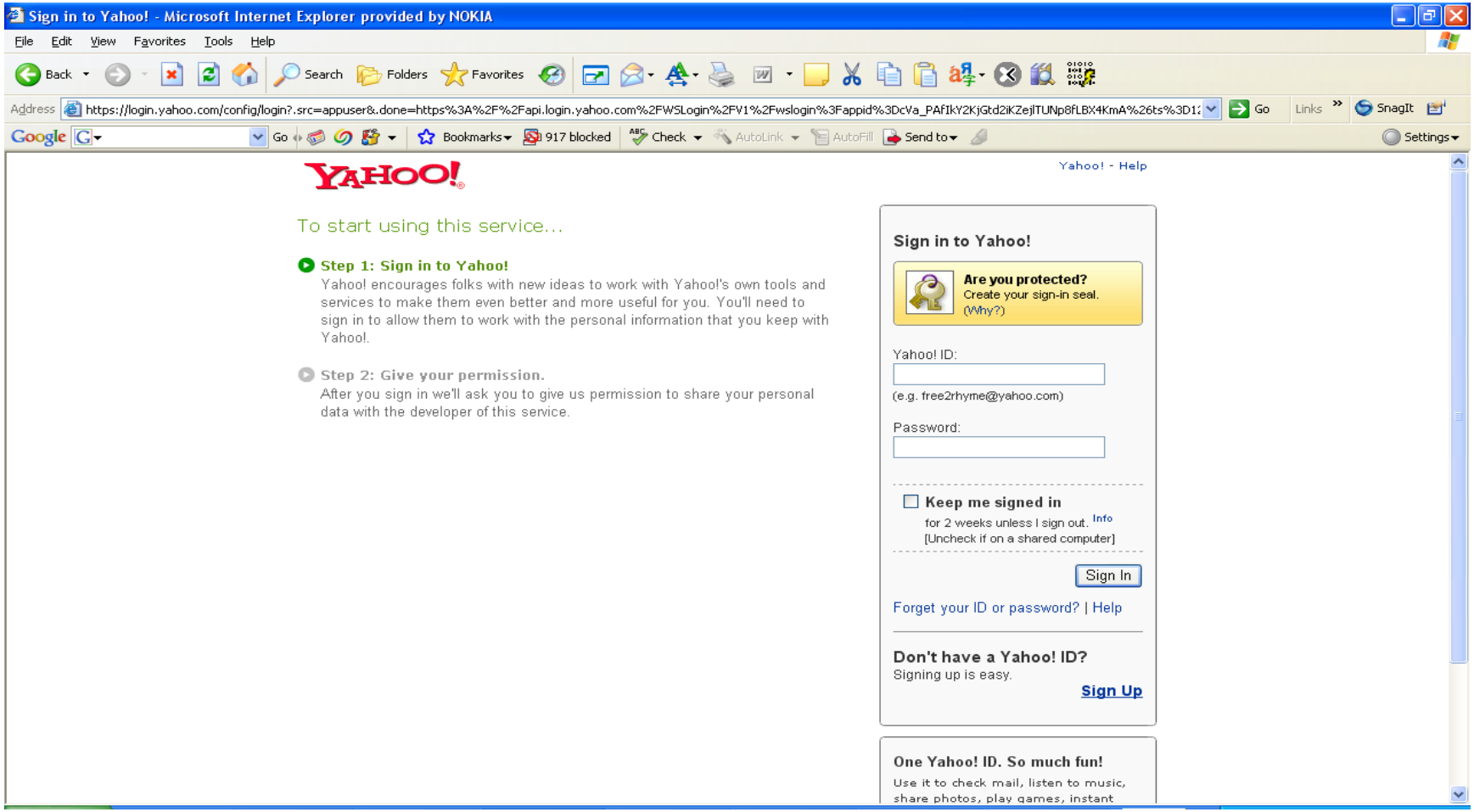
The Problem


Build your network (Why?)
Find contacts who are already on LinkedIn
Web email contacts
Check your address book to find contacts who are on LinkedIn.
Windows Live Hotmail
Gmail
Other
YAHOO!
AOL
Username:
eronenp
@gmail.com
Password:
Upload Contacts
Address book contacts
Outlook, Apple Mail, etc
Find

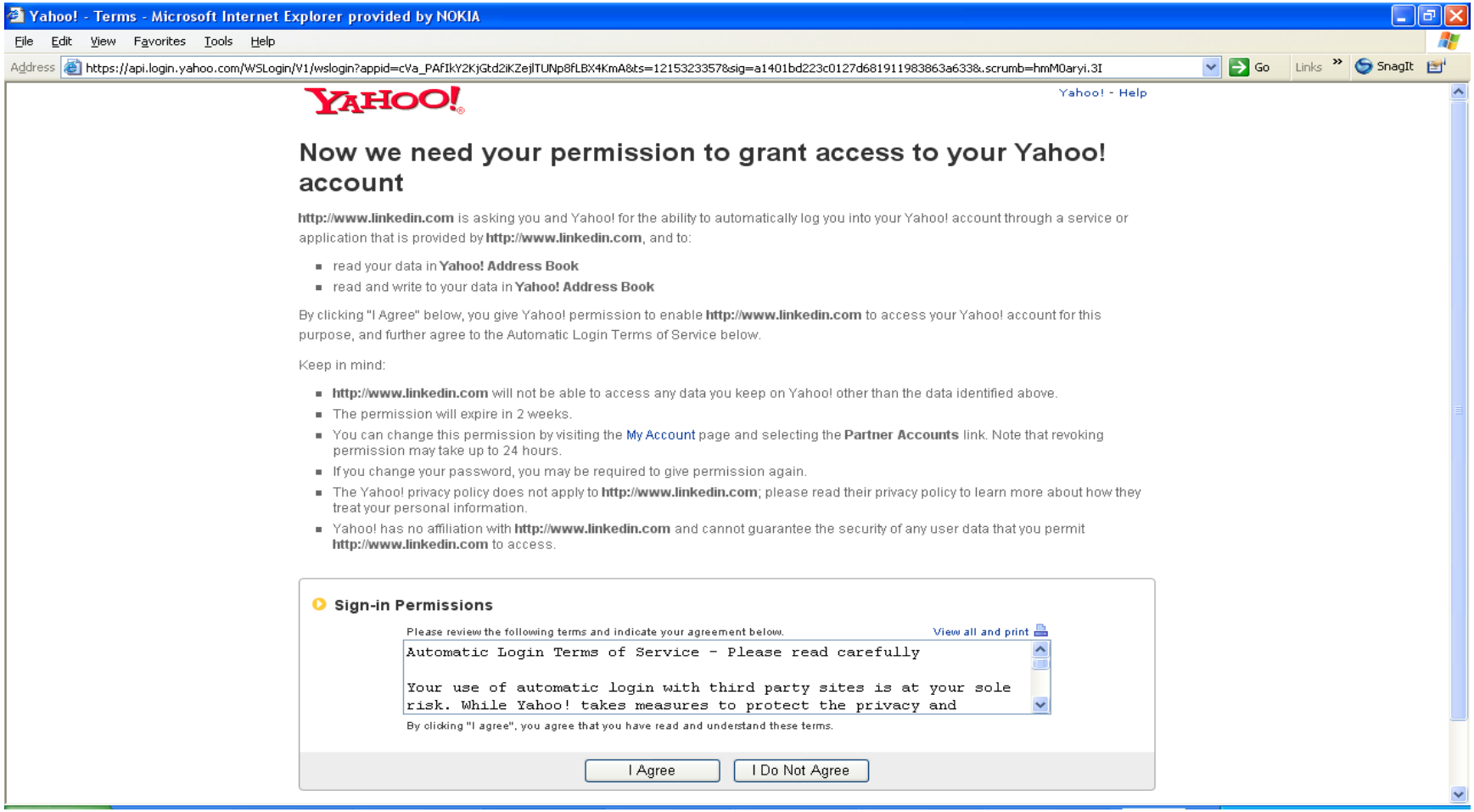
The OAuth Approach


Build your network (Why?)
Find contacts who are already on LinkedIn
Web email contacts
Check your address book to find contacts who are on LinkedIn.
Windows Live Hotmail
Gmail
Other
YAHOO!
AOL
Login to Yahoo!
You will be taken to Yahoo! to enter your username and password.
Address book contacts
Outlook, Apple Mail, etc.
Find

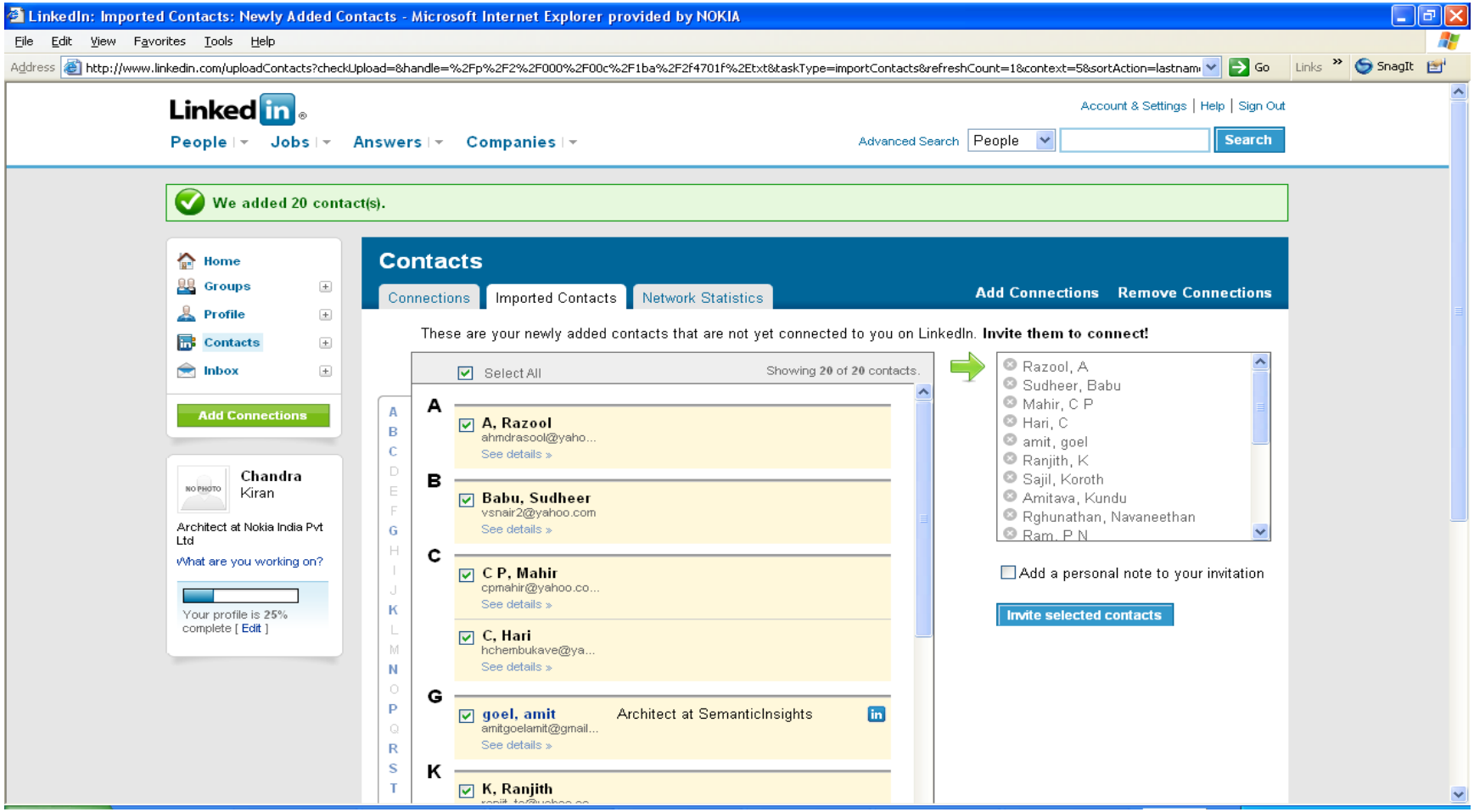
User Authenticated by Service Provider



User Authorizes Consumer to access Service



Consumer calls the Service Provider API



History

History

- November 2006: Blaine Cook was looking into the possibility of using OpenID to accomplish the functionality for delegated authentication. He got in touch with some other folks that had a similar need.
- December 2006: Blaine wrote a "reference implementation" for Twitter based on all the existing OAuth-patterned APIs, which Blaine and Kellan Elliott-McCrea turned into a rough functional draft
- April 2007: Google group was created with a small group of implementers to write a proposal for an open protocol.
- July 2007: OAuth 1.0 (with code for major programming languages)
- September 2007: Re-write of specification to focus on a single flow (instead of "web", "mobile", and "desktop" flows)
- Deployment of OAuth well on it's way: http://wiki.oauth.net/ServiceProviders

History, cont.

- 1st OAuth BOF (Minneapolis, November 2008, IETF#73)
  - BOF Chairs: Sam Hartman, Mark Nottingham
  - BOF went OK but a couple of charter questions couldn't be resolved.
- 2nd OAuth BOF (San Francisco, March 2009, IETF#74)
  - BOF Chairs: Hannes Tschofenig, Blaine Cook
  - Charter discussed on the mailing list and also during the meeting. Finalized shortly after the meeting
- IETF wide review of the OAuth charter text (28th April 2009)
  - Announcement: http://www.ietf.org/mail-archive/web/ietf-announce/current/msg06009.html
- OAuth working group was created (May 2009)
  - Chairs: Blaine Cook, Peter Saint Andre
- Feb 2010: 'The OAuth 1.0 Protocol ' approved as Informational RFC:
  - http://www.ietf.org/mail-archive/web/ietf-announce/current/msg07047.html

The Protocol

* requesting a token
* presenting the token

Presenting a Token

- A → B: HTTP || Token [|| {Header, …, timestamp}$_{key}$]
- A ← B: HTTP (200 OK)
- Questions:
  - What is signed and how?
  - Where does the ***token*** come from?
  - Where does the ***key*** come from?

Signatures

- Used to show ownership of token.
- 'The OAuth 1.0 Protocol‘
  - http://www.ietf.org/internet-drafts/draft-hammer-oauth-10.txt
- Signatures based on symmetric & asymmetric key supported:
  - HMAC-SHA1
  - RSA-SHA1
- No signature = “bearer token”/ PLAINTEXT
- Extensions exist that sign other parts of the message:
  - OAuth Request Body Hash:
    - http://tools.ietf.org/html/draft-eaton-oauth-bodyhash-00
    - http://oauth.googlecode.com/svn/spec/ext/body_hash/1.0/drafts/1/spec.html
  - Going beyond HTTP → OAuth over XMPP
    - http://xmpp.org/extensions/xep-0235.html

The Protocol

* requesting a token
* presenting the token

Requesting a Token

- Different ways to get a token exist.
- Example: WRAP
  - A → KDC: HTTP (get request access token) || credentials
  - A ← KDC: Access Token [, Expires in]

  (also offers the approach of using a refresh token exchange)
- Example: OAuth 1.0
  - A → B: HTTP (get request token) || credentials
  - A ← B: request token

  <<A gets resource owner to tell B to authorize request token>>
  - A → B: HTTP (get access token) || request token
  - A ← B: access token
- Other "flows" have been specified in WRAP
- Various authentication mechanisms specified.

Token

Token

- The token format is not standardized.
- Out-of-scope: *which* permissions were granted, and *how* those permissions are enforced
- Token may be created with constraints, for example regarding lifetime
  - OAuth 1.0 does not specify anything with this regard
  - WRAP http://tools.ietf.org/id/draft-hardt-oauth-01.txt provides a expires_in parameter.

Summary

- Work on delegated authentication in the APPs area in the OAuth group.
- OAuth 1.0: Community version published
- OAuth 2.0: Fusing WRAP, initial OAuth 2.0
- OAuth WG met Monday afternoon. Interim meeting will be scheduled.
- Participation and early feedback desired, especially from security community