Social Design with Facebook
Lessons from Scribd's Integration

Scribd

Matt Riley
matt@scribd.com
About Scribd

“Where the world comes to read, discover, and share…”

• Over 50M Uniques/month
• 10s of millions of documents published
• 10s of millions of users
• Growing rapidly
Why add Social functionality?

Three primary goals for Scribd’s integration:

1. Content discovery
2. Content distribution
3. Engagement
Why use Facebook?

- Most complete social graph out there
- Moves the burden of verification to Facebook
- Overall, high quality social connections
Overview

• Is Social right for your site?
• Facebook Connect for Authentication
• Case study: Readcasting on Scribd
• Social Immersion
Technical preliminaries

• Scribd.com is mostly Ruby on Rails
  – But we don’t use the Rails Facebook plugins

• Facebook Javascript SDK
  – http://github.com/facebook/connect-js

• We use both the Graph API and ‘Old REST API’
Is Social right for your site?

- Deep integration of Social isn’t easy
- If your users aren’t social network users, maybe you shouldn’t concentrate here (yet)
- Let’s measure it
How many of my users are active on Facebook during their visit?

- FB.getLoginStatus – Core API method
- Returns 1 of 3 states for the visiting user:
  - connected
    Facebook user who has connected to your app
  - notConnected
    Facebook user but not connected to your App
  - unknown
    No active Facebook session for the user
How many of my users are active on Facebook during their visit?

```html
<div id="fb-root"></div>
<script>
    window.fbAsyncInit = function() {
        FB.init({appId: 'YOUR APP ID', status: true, cookie: true, xfbml: true});
        FB.getLoginStatus(function(response) {
            _gaq.push(['_trackEvent', 'Facebook', 'Login Status', response.status]);
        });
    }
</script>

Measure % of ‘unknown’ to ‘connected’ or ‘notConnected’
Facebook Connect for Auth

• We discovered that the nodes were there, we just didn’t have the edges
• To start, offer Facebook Connect for signup and login.
• Pull in profile details
• Additionally, Follow a user’s friends when they Connect
Offering Facebook Connect

Sign Up

email address (required)
create username (required)
password (required)

Send me the Scribd Newsletter, and occasional account related communications.

Sign Up

Why Sign up?

1. Discover and connect with people of similar interests.
2. Publish your documents quickly and easily.
3. Share your reading interests on Scribd and social sites.

Other ways to Join Scribd

Login with Facebook
Offering Facebook Connect

Sign Up
Use your Facebook login, and see what your friends are reading and sharing.

Connect with Facebook

I don’t have Facebook
Offering Facebook Connect

Log In

email address or username

password

Log In  Trouble logging in?

Other login options:

Login with Facebook

New to Scribd?

Sign Up for a free account today
Offering Facebook Connect

Log in or Sign up to rate this document
Use your Facebook login, and see what your friends are reading and sharing.

Connect with Facebook

I don’t have Facebook

Already have a Scribd account?
Facebook Connect: Insights

• **Users love FBC:** over 50% of new users create accounts via Facebook Connect.

• **We love FBC:** The number of daily signups on Scribd increased over 50% with the introduction of Facebook Connect.

• Friends lists allowed us to start connecting our users

• Tip: don’t let the Connect button get too big
Readcasting

Allows users to broadcast the documents they read to their Scribd followers and to their Facebook and Twitter accounts
Readcasting

Reading should be social! Post a message on your social networks to let others know what you're reading. Select the sites below and start sharing.

Set your preferences for next time...
Choose 'auto' to readcast without being prompted.

Matt Austin
Matt Austin
Link account

Save Changes
Readcasting
P ≠ NP

Vinay Deolalikar  
HP Research Labs, Palo Alto  
vinay.deolalikar@hp.com  
August 6, 2010
Engineering Challenges

• Client or server-side API implementation?
• Facebook Sessions are ephemeral - they go bad, lose authentication credentials, etc.
• Keeping the Facebook user we know about on the client-side in sync with the user we know on the server-side is hard
Client-side over Server-side

• Ensures the user can grant permissions when needed
  – Asking for permissions in the context of an action converts better

• Ensures we have a fresh Facebook session key
  – Not necessarily the right one, though (more later)

• Simplifies back-end architecture
  – No worker queues or request-cycle trips to FB
Session Syncing

Example:

• User changes Facebook sessions within a single Scribd session, then Readcasts.
• Client-side publishing will simply publish to whatever the active Facebook session is.

Solution: force logout -> login sequence
Session Syncing

General approach for making sure the user we know about on the server-side is the same as the one we have client-side:

1. Every request write our server-side session key to a JS variable
2. After loading Facebook JS, check the cookie client-side to make sure they match
3. If no match, update the session key server-side
4. Every Facebook API action should verify session correctness before executing
Abstract the Session Checking

verifyExec: function(fn, force) {
    if (!facebookUser) return;
    var that = this;
    if (!that.correct()) { // check session are in sync
        if (force) {
            FB.getLoginStatus(function(response) {
                if (response.session) { // Mismatched user
                    FB.logout(function(response) {
                        FB.login(function(response) { that.withValid(fn); });
                    });
                } else // Not logged into facebook at all
                    FB.login(function(response) { that.withValid(fn); });
            });
        }
    } else { fn(); }
},
ensureValid: function(fn) { this.verifyExec(fn, true); },
withValid: function(fn) { this.verifyExec(fn, false); }
Success of Readcasting

• Over 10 million sharing activities in August
• Social traffic is doubling every six weeks
• Per day stats:
  – 10,000 comments on Scribd items in Facebook feeds
  – 3,000 "Likes" per day
• Some pretty cool viral documents
  – P != NP paper from IBM Researcher
  – California Prop 8 Ruling
Readcasting: Product Insights

• Users are pretty comfortable broadcasting this kind of data
  – Offer good privacy controls
  – Prompt for permissions during the ‘action’
• Visitors from Facebook are 40% more engaged than our average visitor
• Large component of content distribution goal
Measuring your own success

- [http://www.facebook.com/insights](http://www.facebook.com/insights)
- Social statistic summaries for your App
  - Daily Likes
  - Daily Shares
  - Feedback per share
  - Reshare rate
  - Most popular content
  - Demographic info
- Limited web interface, but queryable through the Graph API
- Doesn’t give you user-level data (e.g. Like data)
Ruby script to dump Insights data

```ruby
http = Patron::Session.new
http.base_url = 'https://graph.facebook.com'
resp = http.post('/oauth/access_token', {
  :type => 'client_cred',
  :client_id => CLIENT_ID,
  :client_secret => CLIENT_SECRET
})
token = resp.body.split('=').last
resp = http.get("/#{CLIENT_ID}/insights?access_token=#{token}")
ap JSON.parse(resp.body)
```
Tip: tracking ‘Like’ button clicks

- Not in Insights panel
- No FQL to query Like data, so you have to track it yourself

Subscribe to the `edge.create` Event

```javascript
FB.Event.subscribe('edge.create', function(href, widget) {
    Log.info('You liked ' + href, widget);
});
```

Unfortunately, no `edge.destroy` Event
Social Immersion

• Social Activity Feeds
  – drives content discovery

• Add context wherever possible
  – drives engagement
Activity Feed

- Activities from your Friends
- Recommended content based on public Facebook Interests
Adding Social Context

- Take a cue from the new Facebook widgets - add context wherever possible
$P \neq NP$

Vinay Deolalikar  
HP Research Labs, Palo Alto  
vinay.deolalikar@hp.com

August 6, 2010
Scribd Follow
Facebook Like
Traditional Share

$P \neq NP$

Vinay Deolalikar
HP Research Labs, Palo Alto
vinay.deolalikar@hp.com

August 6, 2010

Recent Readcasters

Comments
Recent Readcasters

People I’m following are shown first

Tip: Redis can make your life much easier here!
Comments

Comments from people I follow are pulled to the top

Add a Comment

Recent comments from people you follow

James Yu left a comment
I always come back to this. It's great.
05 / 04 / 2010

Tikhon Bernstam left a comment
bet you never thought this paper would get half a million people looking at it when you first wrote it
03 / 12 / 2010

Comments from everyone
Questions?

Matt Riley
matt@scribd.com