EXHIBIT 4

Mulligan Last modified Saturday, July 11, 2009 at 10:48am by Michael Sharc

Mulligan is an ongoing improvement to Facebook's Photos product.

For a technical overview of FBPlugin, the project driving the new photo uploader, see FBPlugin Technical Overview.

Goals

- Greatly increase the number of photos and videos getting uploaded to and consumed on FB
- Simplify the Photos and Video products
- Attract more power users

Uploader

Facebook's current photo uploader is an unwieldy beast. It was written by a third party company (Aurigma) and has been untouched since its release in 2005. It consists of two separate plugins: an ActiveX control for IE and a Java applet for all other browsers. In a recent poll to users, 37% of respondents who have used the uploader reported having trouble with it.

Due to the problems users have faced with the current uploader as well as a desire for a simpler and more consistent design, we're building a new one.

The new uploader will be an ActiveX control in IE and a Netscape Plugin API (NPAPI) browser plugin for our other supported browsers (Safari, Opera, Firefox). The plugin itself is invisible, and will simply provide a Javascript interface for performing tasks not normally possible in JS -- filesystem access, image manipulation, and interaction with the OS. This model allows for speedy iteration on the visual components of the plugin and will only require a recompile and update to users when lower level JS interfaces need to change or be updated.

Trojan Horse (Sort of)

The new uploader represents the first home-grown compiled software that Facebook will hand to users for execution on their local machine. It will include an automated update process of some form so we can have it update itself, as well as some hooks for installing other Facebook desktop software in the future (but we intend not to be evil). Potential software down the road might include a resident application to detect when digital cameras are plugged into the computer, so we can recommend the user to dump its contents to Facebook.

Key Differences between ActiveX and NPAPI Plugin

ActiveX

Most of our users use IE and will therefore use this.

- Works only on IE.
- There is no form of install, and thus no need to self-update.

NPAPI

- Does not work on IE. Works on all other supported browsers.
- Installation requires some level of administrative privileges, so we may take a hit in computer labs at schools where Firefox tends to be pushed on users. One easy way to help mitigate this is to build a simple network install executable for IT admins to deploy on multiple machines.

Technical Info

The uploader does not have a drawable area, and the interface will be created entirely in the DOM with HTML and CSS. For displaying local thumbnails, we have decided on a model where the uploader will run a local web server (bound to localhost to avoid firewall complaints) to serve them. Other models considered include using the data:// URI, using SWFs to display base64-encoded image data strings, and using the file:// URI -- in each case, either not all of our supported browsers supported the model or it was too resource/plugin intensive.

The JS API for the uploader will include mostly asynchronous methods to accomplish the following:

- Get filesystem roots and OS-specific shortcut folders (e.g. "Photos" on OS X and "My Photos" on Windows)
- List the contents of a directory with simplified file types
- Generate local thumbnails for images
- Retrieve thumbnails for videos from the OS (where possible)
- Provide URLs for displaying local thumbnails
- Perform a multipart form post to some URL for a selection of images and videos
- Check for updates / perform a self-update

Plaintiff's Trial Exhibit
PTX-190
Case No. 08-CV-00862

Testing

Due to the nature of pushing new binary code which can be tedious and generally never sees full penetration vs. the previous version, there is an imminent need for thorough testing across as many platforms as we can get our hands on. Every supported browser must be tested as well as all common OS's, antivirus/firewall software, and "computer lab" scenarios.

Operating Systems:

- XP SP0/SP1/SP2/SP3
- Vista SP0/SP1
- OS X 10.4/10.5
- Ubuntu (Latest stable release)

Photos Product Improvements

While the new uploader is the first priority, it paves the way for a great deal of long-needed improvements to the photos product to advance the goals of the Mulligan project.

The Photos product is currently a system where:

- All photos must belong to albums
- Videos and photos are contained in completely separate products
- The upload flow is "filling an album" -- and is hampered by an album size limit
- Context switching in photo browsing is either not easy/obvious or not possible (e.g. a photo in an album to an event or group this photo is in)
- No real bulk photo management exists
- Images are fairly low resolution relative to originals and other photo sharing web sites

Improvements that we want to make to mitigate the above:

- Photos and albums are decoupled -- photos can be in multiple albums and even have multiple owners. All of a user's media will be kept in a big library. Photo privacy will be per-photo, and setting album privacy will just apply settings to a group of photos.
- Videos will be rolled into the new "media library," and treated just like photos for the purposes of albums and other groupings.
- The upload flow will be changed to optimize for maximum image/video throughput to the site, with a secondary goal of good editorial control for surfacing of the content (e.g. album creation after the fact or automatic album creation based on timestamps)
- Every photo/video permalink page will identify all contexts in which the photo may be displayed, and a mechanism for switching to a new stream than the current context.
- A tool will be built for bulk management of the library for applying privacy settings, deleting, editing captions, and tagging.
- Resolution will be increased and we'll explore the possibility of a premium product for storing high-resolution originals.

What happens to Video?

We haven't yet figured out exactly how the roll-in of Video will work with these improvements, but it will likely cease to exist in a lot of its current forms. It will be possible to filter a stream's view to "only videos" (or only photos, defaulted to both), which is enough to eliminate the "videos by so-and-so" pages. The Video dashboard can also be rolled into the Photos dashboard.

It's likely the Video application will simply become "Photos". Other sites have already led by example, like Flickr, where they had a primarily photos-focused brand and added video into it without needing to call it out separately.

Resolution Increases

We don't plan to support storage of originals, because at current projections it'd be prohibitively expensive. Instead we'll look into the appropriate resolution increase to ensure photos on Facebook are great-looking, but not wasteful on space. At the very least, we should increase the default large edge resolution of photos to the new frame width introduced in FB95.

Technical Info

Technical FBPlugin Information

See FBPlugin Technical Overview.

FBObject / Associations

To support the further decentralized model of photos and their connections, we will work toward using FBObject and Associations for all photo objects (albums, photos, tags, etc). This involves a significant migration up-front, but avoids recreation of the wheel and will pay off in code simplicity and consistency with other products on the site.