

Applications on Facebook: iframes

An obvious first stab for more *safely* displaying the content of one application in the visual context and flow of another site relies on a technology already incorporated into the browser itself: the *iframe*.

To reuse the mappings from the previous section, a request to *http://apps.facebook.com/fettermansbooks/PATH?QUERY_STRING* would result in HTML output like this:

```
<iframe src="http://fettermansbooks.com/fbapp/PATH?GET_STRING"></iframe>
```

The content of this URL would display in a frame inside the Facebook page, and in its own sandboxed environment could contain any type of web technology: HTML, JS, AJAX, Flash, and others.

This essentially results in the browser becoming the request broker rather than Facebook. An improvement on the model from the previous section, the browser also maintains the safety of the rest of the elements on the resulting page, so developers can create whatever experience they want inside this frame.

For applications whose developers want to invest minimal effort in moving their code from their site's logic to the Platform, the iframe still makes sense. In fact, Facebook continues to support the iframe model of full page generation. Although this solves the first product goal, incorporation into a social site, the second remains an open question. Even with the safety of the iframe-based request flow, these developers do not benefit from any new data beyond that exposed by the API service.

Applications on Facebook: FBML As Data-Driven Execution Markup

The straw-man solutions in the previous two sections each have their charm. The HTML solution takes the intuitive step of reframing applications themselves as web services, bringing contact back for display on the Facebook domain. The iframe model incorporates the benefit of running developer content in a separable (and safe) execution sandbox. The best solution would retain the application-as-service model and the safety and trust of the iframe, while enabling developers to use more social data.

The problem is that in order to provide the unique experience of their social application, developers must provide the data, logic, and display from their own stack. However, this output must be generated with user data that cannot leave the Facebook domain.

The solution? Send back not HTML but specific markup that defines sufficient amounts of the application's logic and display, plus requests for protected data, and let Facebook render it entirely in its trusted server environment! This is the premise of FBML (Figure 6-5).