this FBML *data*. This is a great example of data at the center of execution: FBML is simply *declarative* execution rather than *imperative* flow (as is the case in C, PHP, etc.).

Now on to the specifics. FBML is an instantiation of XML, so it is composed of tags, attributes, and content. Tags fall into the following broad conceptual categories.

Direct HTML tags

If an FBML service returns the tag `<p/>`, Facebook would render this simply as `<p/>` on the output page. As the bedrock of web display, most HTML tags are supported, with the exception of a few that violate Facebook-level trust or design expectations.

So the FBML string `<h2>Hello, welcome to <i>Fetterman's books!</i></h2>` would be left essentially intact when rendered into HTML.

Data-display tags

Here's where some of the power of data comes in. Imagine that profile pictures were not transferred off-site. By specifying `<fb:profile-pic uid="8055">`, the developer can display more data to the Facebook user as part of their application, without requiring the user to fully trust this information to that developer.

For example:

```
<fb:profile-pic uid="8055" linked="true" />
```

translates to the FBML:

```
<a href="http://www.facebook.com/profile.php?id=8055"
 onclick="(new Image()).src = '/ajax/ct.php?app_id=...">
  <img uid="8055" linked="true" src="..." alt="Dave Fetterman" title="Dave Fetterman" />
</a>
```

> **NOTE**
> The complicated `onclick` attribute is generated to restrict JavaScript while displayed in a Facebook page.

Note that even if information were protected, this content is never returned to the application stack, but only displayed to the user. Execution on the container side makes this data available to the viewer, without requiring that it pass through the hands of the application!

Data-execution tags

As an even better example of using hidden data, the user privacy restrictions accessible only through the internal `can_see` method (Example 6-3) are an important part of application experience, yet not accessible externally through data services. With the `<fb:-if-can-see>` tag and others like it, an application can specify a target user in the attributes such that the child elements are rendered only if the viewer can see that target user's specific content. Thus, the