```
echo 'Friends' reviews:<br/>';
echo '<fb:wall>';

// put friends up top.
foreach ($friend_reviewers as $book_uid => $review) {
  echo '<fb:wallpost uid="'.$book_uid.'">';
  echo '(' . $review['score'] . ')' . $review['commentary'];
  echo '</fb:wallpost>';
  unset($all_reviewers[$book_uid]); // don't include in nonfriends below.
}

echo 'Other reviews:<br/>';

// only nonfriends remain.
foreach ($all_reviewers as $book_uid => $review) {
  echo '<fb:if-can-see uid="'.$book_uid.'">'; // defaults to 'search' visibility
  echo '<fb:wallpost uid="'.$book_uid.'">';
  echo '(' . $review['score'] . ')' . $review['commentary'];
  echo '</fb:wallpost>';
  echo '</fb:if-can-see>';
}

echo '</fb:wall>';
```

Even though this takes the form of a service outputting FBML instead of a web call outputting HTML, the usual flow remains intact. Here, Facebook data enables the application to show more relevant book reviews (friends' reviews) before less relevant ones, and uses FBML to display the result using appropriate privacy logic and design elements on Facebook.

FBML Architecture

Transforming FBML provided by developers into the HTML shown on *http://facebook.com* requires a number of technologies and concepts working together: parsing the input string into a syntax tree, interpreting tags in this tree as internal method calls, applying the rules of FBML syntax, and maintaining the constraints of the container site. Like FQL, here we again focus primarily on the interaction of FBML with the platform's data, and detail only in broad strokes the other pieces of the technology puzzle. FBML handles a complex problem, and the full implementation details of FBML are quite voluminous—these include omitted topics such as FBML's error logging, the ability to pre-cache content for later rendering, signing the results of form submission for security, and so forth.

First, to the low-level issue of parsing FBML. In inheriting some of the roles of the browser, the Facebook platform also inherits some of its problems. For developer convenience, we do not require input to arrive as schema-verifiable or even well-formed XML—unclosed HTML tags, like `<p>` (as opposed to XHTML, like `<p/>`) break the assumption that the input could be parsed as true XML. Because of this, we need a way to first transform an input stream of FBML into a well-formed syntax tree with tags, attributes, and content.