

bin scripts because at the time, the browser seemed like the only real client to serve. As we started to realize the applicability of persistent, unambiguous identifiers for use in the Semantic Web, life sciences, publication, and similar communities, we knew that it was time to rethink the architecture to be more useful for both people and software.

The PURL system was designed to mediate the tension between good names and resolvable names. Anyone who has been publishing content on the Web over time knows that links break when content gets moved around. The notion of a Persistent URL is one that has a good, logical name that maps to a resolvable location. For example, a PURL could be defined that points from *http://purl.org/people/briansletten* to *http://bosatsu.net/foaf/brian.rdf* and returns a 303 to indicate a “see also” response. I am not a network-addressable resource, but my Friend-of-a-Friend (FOAF) file\* is a place to find more information about me. I could pass that PURL around to anyone who wants to link to my FOAF file. If I ever move to some other company, I could update the PURL to point to a new location for my FOAF file. All existing links will remain valid; they will just 303 to the new location. This process is described in Figure 5-6. The PURL Server implements the W3C Technical Architecture Group (TAG) guidance that 303 response codes can be used to provide more information about non-network addressable resources.

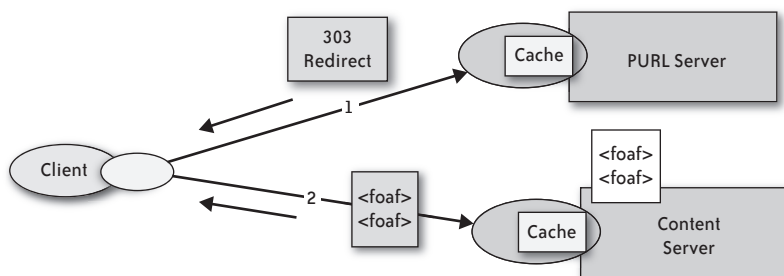


FIGURE 5-6. PURL “See Also” redirect

In addition to supporting the PURL redirection, we wanted to treat each major piece of data in the PURL system as an addressable information resource. Not only does this simplify the interaction with the user interface, it allows for unintended potential reuse of the data beyond what we originally planned. Manipulation of the resource requires ownership credentials, but

\* <http://foaf-project.org>