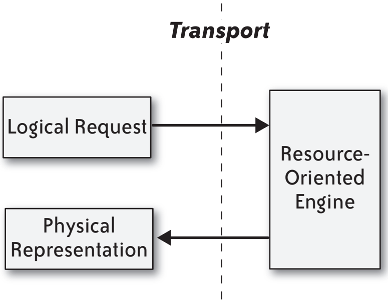

FIGURE 5-7. Shallow resource-oriented architectures

We chose to use NetKernel as the foundation for this architecture because it is the embodiment of resource-oriented architectures and has a dual license, allowing its use with both open source and commercial projects. The idea of a logical coupling between layers with different representations is baked into the software architecture and offers similar benefits of flexibility, scalability, and simplicity. The linkage between the layers is through asynchronously resolved, logical names. This deeper notion of resource-oriented architectures looks something like Figure 5-8. NetKernel is an interesting software infrastructure because it takes the idea of logically connected resources inside so that HTTP logical requests can be turned into other logical requests. This architecture reflects the properties of the Web in a runtime software environment.

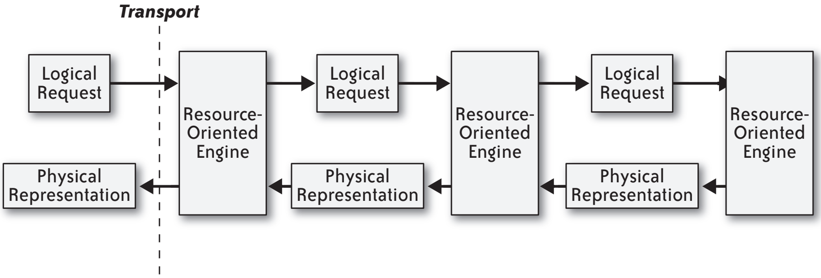

FIGURE 5-8. Deep resource-oriented architectures

The external URL *http://purl.org/employee/briansletten* gets mapped through a rewrite to a piece of functionality called an accessor.† Accessors live in modules that export public URI definitions representing an address space they will respond to. The convenience here is that it is possible to radically change the implementation technologies in a newer version of a module and simply update the rewrite rules to point to the new implementation. The client needs to

† *http://docs.1060.org/docs/3.3.0/book/gettingstarted/doc_intro_code_accessor.html*

Download at WoweBook.Com