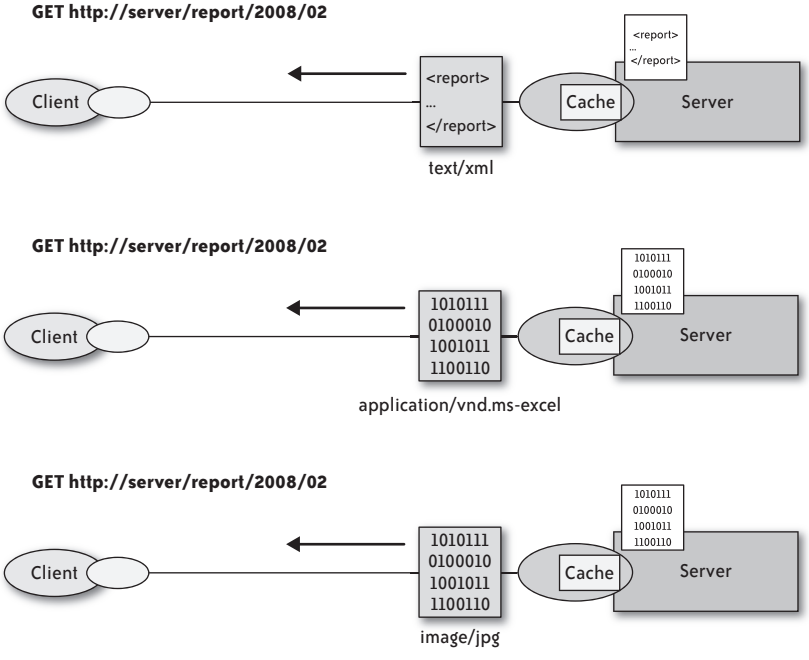

FIGURE 5-5. Negotiating content in a resource-oriented environment

In addition to picking the physical representation within the context of resolving a request, we might also enable the server to decide how much of the referenced data set to return based on the identity of the user, the application being used, etc. We can imagine a scenario where a call center agent using a relevant application needs to access sensitive information to resolve an issue. This could include Social Security numbers, credit card numbers (or hopefully only the last four digits), home addresses, etc. There is a specific business need to justify the agent accessing this information, so we could have a declarative policy in place that lets it happen. The same employee using a different application in a different context (perhaps a marketing analysis package) is unlikely to have a business need to access that sensitive information, although we may still want to resolve a reference to the same customer to access her demographics and purchase history. In this case, the context would not support access to the sensitive data, and we could enforce an automatic filtering process to remove or encrypt the sensitive information. The decision of which approach to take would depend upon where the data needed to go next. Encrypted data requires access to keys, which becomes another management burden. It is just as easy to remove the sensitive data but include it in a different resolution context when it is needed.

Managing single-point access control might not be a big problem for conventional Enterprise architectures. However, given the increased presence of workflows, explicitly modeled business processes, and the like, we have plenty of opportunities to consider a user of one