

# Enterprise System Design

Our strong experience with many different enterprise system technologies and architectures means we understand enterprise system *design patterns* . Whenever our engineers design or review enterprise systems, we concentrate on architectural patterns which we've seen occur again and again. Because we've seen them enough, we know how to work with these patterns.

- *Persistence patterns*. We know middle-tier to back-end communications, which in OO systems generally uses one of a few patterns. Our developers have worked with many different object-relational (O/R) toolkits such as TopLink(tm), JDO, Oracle's BC4J(tm), Hibernate, and a few different J2EE vendor's generated CMP EJB systems (BEA and JBoss). Alternative persistence layers include direct database manipulation (which is often much faster and easier to develop than heavier-weight O/R-based systems), and toolkits based on dynamic objects, which can be much easier to use the O/R-generated objects. Choice of pattern and pattern variants depend on the size and scope of the project, the complexity of business rules that need to be implemented, and the skills of the maintenance team that will care for the system long-term.
- *Security patterns*. In an enterprise system, identity-based security is king. We understand how to implement identity-based security in many different ways: using identity servers (LDAP -- Novelle, Active Directory, Netegrity and Oblix), or custom solutions based on the component model of the application server itself. Available infrastructure and a client's "comfort" with a specific type of technology usually drives the selection of a security architecture.
- *Audit patterns*. Audit/log infrastructure, according to the log review requirements, can be either centralized or decentralized, high-structured or free-form. The more centralized and highly-structured, the more the audit information can be used to monitor the health of a system, and also to audit the security aspects of the system. De-centralized, free-form audit systems are only useful to monitor usage patterns from a "black-box" perspective. We have designed and used several different kinds, and can guide appropriately based on your system requirements.