With Entrust’s GetAccess authorization management system, users navigate to the Access Server’s log-in screen to access secure resources. This server passes the authentication request to the Registry Server, which returns user verification and profile information to the Access Server, along with cookies containing the authorized resource list, user roles, and other information. The Access Server encrypts the cookies and returns a custom navigation menu to the user. When the user chooses an application, the runtime service on the Web server intercepts the request, decrypts the cookies, and verifies the session with the Registry Server. The registry server then passes the user ID, preferences, roles and application-specific data to the application on the Web server so that it can deliver the appropriate level of information to the user. The Mobile Proxy Server manages sessions for cookie-less computers such as wireless devices.

In Netegrity’s SiteMinder architecture, when a user requests a protected resource from the Web server, a Web agent pulls the user’s credentials from the browser and passes them to the SiteMinder policy server, which in turn authenticates the user using the stored credentials in the directory services or authentication services databases. The policy server passes the information to the application, which personalizes the content based on the user’s privileges. Netegrity integrates with existing directories and databases, rather than creating its own.

Securant’s authorization design avoids integration complexities by separating content and applications onto different Web servers. The administrator creates a business rule for each resource on the protected Web site that defines which users are authorized to access which applications. The first time a user requests a protected resource, a ClearTrust plug-in on the Web application server prompts the user for credentials. It passes the results to the authorization server, which polls the Securant entitlement database and returns user authorization information to the plug-in. Finally, the plug-in enforces access control for the resource. An encrypted cookie, passed to the browser, allows for subsequent accesses.