

a secret known only to the user such as a username and password. Sometimes these credentials are publicly known, but hard to duplicate like a passport, driver's license or fingerprint.

[0587] To support different types of authentication, component integration engine allows the swapping of components used to perform the authentication. By stacking authentication modules, the administrator can require more than one form of authentication.

[0588] The component integration engine of the present example provides six layers of component level security by restricting access to the various components. If the current user does not have the correct privileges, the component cannot be accessed. Component security is checked in all of the following situations: a user tries to access a virtual host, a user tries to access a service context, a user tries to access a service, a user tries to execute a command, a user tries to access a manager, and a user tries to retrieve a component from a manager.

[0589] Component security is based on the name of the component being accessed or the operation being performed. The mechanism for controlling user access is called a privilege. The following is an example of components and the necessary privileges if the system administrator has setup a virtual host named localhost, a service context named marketing, and several services and managers: 1. A user must have the localhost privilege in order to access the virtual host named localhost. 2. A user must have the localhost.marketing privilege in order to access the marketing service context in the localhost virtual host. The user must also have enough privileges to use the localhost virtual host. The general format for privileges required to access a service context is <virtual host name>.<service context name>. 3. A user must have the localhost.marketing.datasource privilege in order to access the datasource service in the marketing service context of the localhost virtual host. The user must also have enough privileges to use the marketing service context. The general format for privileges required to access a service is <virtual host name>.<service context name>.<service name>. 4. A user must have the localhost.email\_server privilege in order to use components in the email\_server manager in the localhost virtual host. The user must also have enough privileges to use the localhost virtual host. The general format for privileges required to access a manager is <virtual host name>.<manager name>. 5. A user must have the localhost.email\_server.marketing privilege in order to use the marketing component in the email\_server manager in the localhost virtual host. The user must also have enough privileges to use the email server manager. The general format for privileges required to access a managed component is <virtual host name>.<manager name>.<component name>.

[0590] Privileges are associated with a thread of execution. For this reason, it is not possible to use thread pools to execute secure tasks without first associating the correct credentials and removing those credentials at the end of the execution. In fact certain credentials are required just to start the component integration engine. These credentials are associated with the system principal and no user other than the system administrator may be allowed to use the system principal account.

[0591] The component integration engine checks permissions associated with an assembly of components. If the component assembly specifies permission requirements, these permissions must be held by a user in order for the assembly to be executed.

[0592] The component integration engine also provides encryption and digital signature components. An encryption component can be used to encrypt data to prevent access to anyone without the correct decryption credentials. Digital signatures provide a mechanism to prove that the source of the data was from a specific user and that the data was not altered in any way.

[0593] Unlike the previous security sections, encryption and digital signatures are optional security features. A component integration engine provides components to perform encryption and digital signatures because data is not always accessed through the component integration engine. Databases, files, and email can all be accessed directly. By encrypting the data stored in these locations, data access is restricted to a user with the correct decryption credentials using the component integration engine.

[0594] All documents, patents, journal articles and other materials cited in the present application are hereby incorporated by reference.

[0595] Although the present invention has been fully described in conjunction with several embodiments thereof with reference to the accompanying drawings, it is to be understood that various changes and modifications may be apparent to those skilled in the art. Such changes and modifications are to be understood as included within the scope of the present invention as defined by the appended claims, unless they depart therefrom.

What is claimed is:

1. A meta-implementation layer comprising:

a metamodel repository containing a plurality of descriptors;

a plurality of implementations for providing access to software components described by said plurality of descriptors;

a metametamodel repository including a plurality of metamodel descriptors for describing said descriptors and a plurality of metamodel implementations for describing said implementations, wherein said meta-implementation layer provides access to an implementation of said plurality of implementations to thereby allow a user to have access to said software components of a software program.

2. The meta-implementation layer of claim 1, wherein said plurality of descriptors include at least one enumeration descriptor.

3. The meta-implementation layer of claim 2, wherein said plurality of implementations includes at least one enumeration implementation associated with said enumeration descriptor.

4. The meta-implementation layer of claim 1, wherein said plurality of descriptors include at least one role descriptor.

5. The meta-implementation layer of claim 4, wherein said plurality of implementations includes at least one role implementation associated with said role descriptor.