Security Capsule Specification

Purpose

This Capsule provides a starting point for applications that need to model information in support of a Multi-Level Security (MLS) environment. The reader should become familiar with the Team Modeler capsule before reading this specification.

Background

In the MLS model a Subject (a Person, Role or Organization) must have an appropriate clearance to access an Object (generally data, but in some cases, machines or other physical resources) of a particular classification. Every Subject and Object has a Label that determines its security level. Labels have at least two parts:

- **Sensitivity**: — A hierarchical attribute such as "Secret" or "Top Secret".
- **Categories**: — A set of non-hierarchical attributes such as "US Eyes Only" or "UFO".

A Security Level (label) must have one sensitivity, and it may have zero or more categories.

Examples of Security Levels are: { Unclassified }, { Secret / UFO, Crypto } and { Top Secret / UFO, Crypto, Starcommand }.

The Security capsule is designed to support the Bell-Padula MLS model, with the ability to manage Write, Read-Only and Read-Write Entitlements. The methods that it incorporates provide basic Type and Instance enforcement of the security rules in conjunction with the Objectivity/DB object encryption feature, which provides a hook for invoking security enforcement mechanisms.

The object model is designed to support more flexible models than Bell-Padula and MLS. At least one such example will be provided with the capsule.

Functionality

The Security capsule supports the object model depicted in Appendix A. Note that it incorporates the Team Modeler capsule in order to model Persons, Organizations and Roles. It also extends Role to embrace Administrator functionality. Examples of usage are described in Appendix B.

The capsule provides tools or methods for:

- Installing a federated database with a preloaded schema and at least one template database.
- Creating, updating and deleting (optionally) named, hashable and/or indexed instances of each of the object classes and the ability to link instances to other instances.
- Adjusting the placement model (segmentation into databases, containers and object clusters).
- Finding instances by name or key and querying or iterating over all instances of a class. The latter methods are automatically generated during schema definition.

- At least the following kinds of relationships will be supported by methods to create, delete and traverse them:
  - Relating one or many Persons to one or many Roles (who does what?).
  - Relating Roles (and hence Administrators) to one another (role reporting structure).
  - Relating Organizations to one another (corporate hierarchy).
  - Relating Person objects to one another (line management hierarchy).
  - Relating a Person to one or more Organization (department) instances as the Manager.
  - Relating Subjects to Object Classes and Instances via an intermediary Entitlement object.
  - Relating a Sensitivity object to zero to many Subject objects.
  - Relating a Sensitivity object to zero to many Category objects.
  - Relating a Sensitivity object to zero to many "Object" (class or instance) objects.
  - Relating a Category object to zero to many Subject objects.
  - Relating a Category object to zero to many "Object" (class or instance) objects..
- The Team Modeler capsule will be used to support the above functionality plus:
  - Transitive closure
  - Loop avoidance.
  - Path finding
  - Deep copy
- An Administrator can set, change and remove Entitlements.
- If a Subject (Administrator) with lower Security Level than a subordinate (which can happen, particularly during internal investigations) deletes that subordinate then the Subject is not actually deleted. It is associated with the Administrator's Administrator or another designated one and removed from the view of the original Administrator. All of its other relationships and privileges remain unchanged.
- Assigning Security Labels to Object classes and instances is delegated to the application.
- Only an Administrator can assign Security Labels to a Subject and change or delete them.
- A newly created Object (class or instance) is assigned the Sensitivity of the Subject creating it, unless otherwise specified.
- Creation and assignment of Categories, Entitlements and links between Subjects and Objects is delegated to the application. It is suggested that Entitlements be created on an "as-needed" basis. For instance, if a Subject with "Secret" clearance and access to the Category "UFO" attempts to access an object instance that has (at least) those Security Labels then a check is first made for an Entitlement object. If none exists it can be created for future use. If it exists then the check has already been made.

- Access to objects, particularly creation, deletion, opening, updating and closing, invokes the object encryption hook, which, in most cases, is simply a check of the user's (Subject's) Entitlements to access this object class or instance.
- There will be no versioning/history capability. This may be provided in a separate capsule.
- There is no automatic audit trail capability. This may be provided in a later version of the capsule. To implement one in the database would require "Never-Rollback" containers.
- *To Be Determined:* Provision of a GUI, such as an ooAssist/Eclipse plugin, supporting the above functionality.

Platforms and Languages

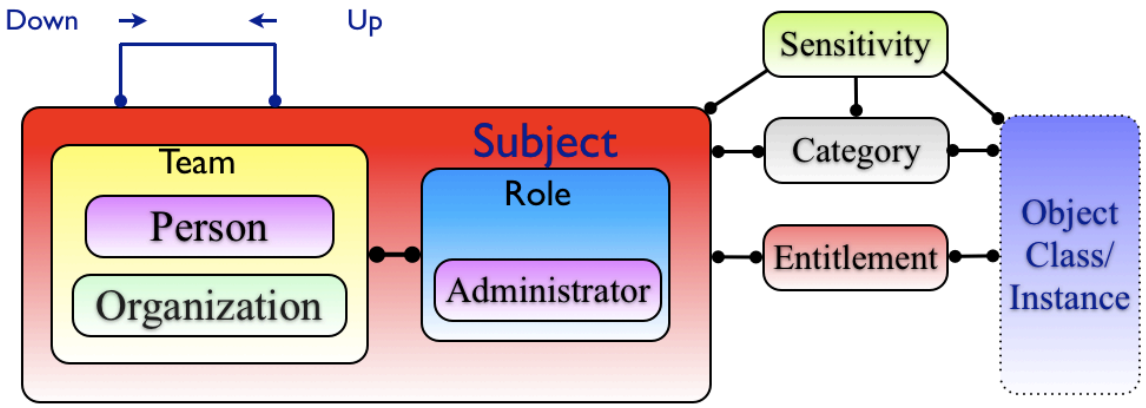
- C++, Java and C# (later) on Windows and Linux.

Suggested Pricing

- 60-day trial, then a one-time license fee of $500 per developer.

Note: Appendix A follows...

Appendix A - Object Model for the Security Capsule



- Subject inherits from the *Structure Capsule* Component class.
- Team and Role inherit from the Subject class
- Person and Organization inherit from Team
- Administrator inherits from the Role class.
- Only Administrators can create, change or delete Entitlements.
- Only Administrators can create, change, delete or create links to Security Labels (at least one Sensitivity object linked to zero or more Categories).