8.4 Subject Descriptor

Intent

Provide access to security-relevant attributes of an entity on whose behalf operations are to be performed.

Also Known As

Subject Attributes. The entity described may be referred to as a subject or principal.

Motivation

There are many security-relevant attributes which may be associated with a subject; that is, an entity (human or program). Attributes may include properties of, and assertions about, the subject, as well as security-related possessions such as encryption keys. Control of access by the subject to different resources may depend on various attributes of the subject. Some attributes may themselves embody sensitive information requiring controlled access.

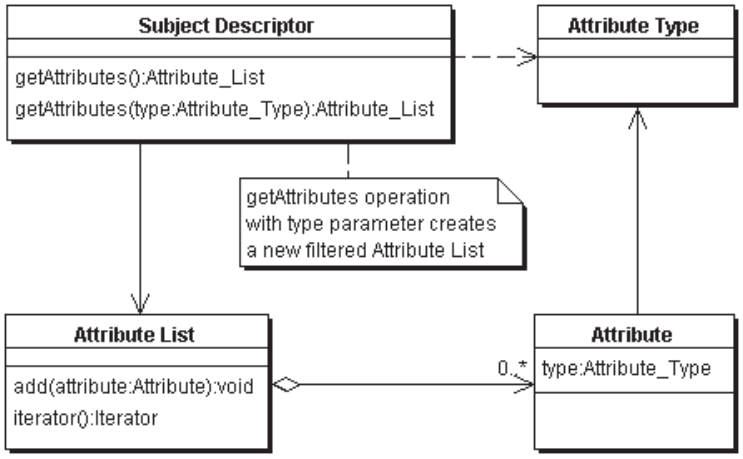
Subject Descriptor provides access to subject attributes and facilitates management and protection of those attributes, as well as providing a convenient abstraction for conveying attributes between subsystems. For example, an authentication subsystem could establish subject attributes including an assertion of a user's identity which could then be consumed and used by a separate authorization subsystem.

Applicability

Use the Subject Descriptor pattern when:

- A subsystem responsible for checking subject attributes (for example, rights or credentials) is independent of the subsystem which establishes those attributes.
- Several subsystems establish attributes applying to the same subject.
- Different types or sets of subject attributes may be used in different contexts.
- Selective control of access to particular subject attributes is required.
- Multiple subject identities need to be manipulated in a single operation.

Structure



Participants

- Subject Descriptor

  Encapsulates a current set of attributes for a particular subject.

  Supports operations to provide access to the complete current set of attributes, or a filtered subset of those attributes.

- Attribute List

  Controls access to and enables management of a list of attributes for a subject.

  A new Attribute List can be created to reference a filtered subset of an existing set of attributes.

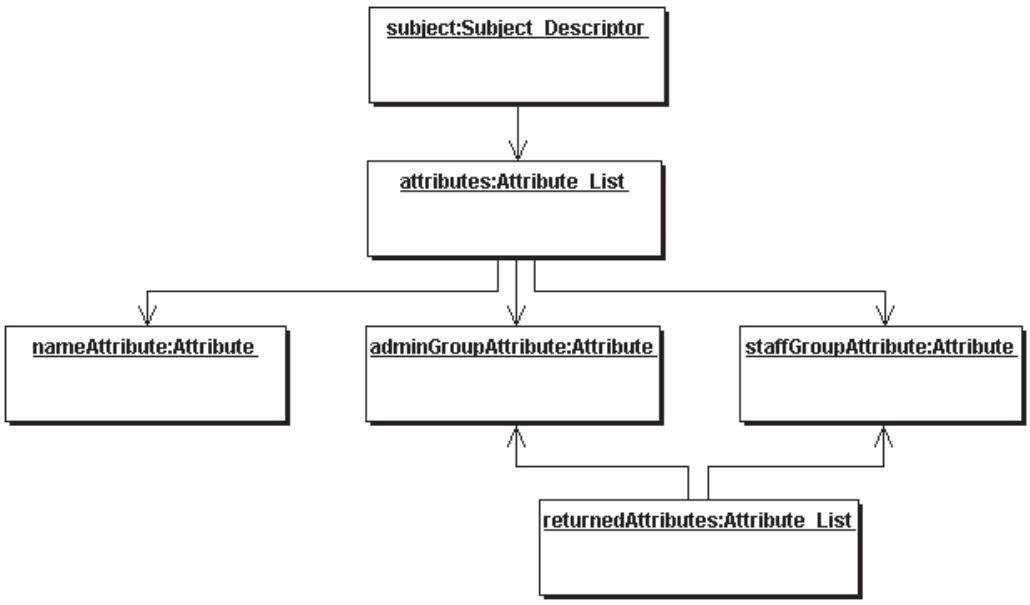
- Attribute

  Represents a single security attribute.

- Attribute Type

  Allows related attributes to be classified according to a common type.

The following object diagram shows an example instantiation of a Subject Descriptor, with its internal Attribute List, and a constructed Attribute List referencing a subset of the Attributes.



Collaborations

Attribute List returns an Iterator [GoF] allowing the caller to operate on the individual Attributes referenced in the list.

Attribute List may be a Guarded Type (see the Protected System pattern above), consulting Policy in order to determine whether the caller is permitted to access attributes within the list. A filtered Attribute List can be a way for a caller to pre-select only those attributes which it is permitted to access.

Consequences

Use of the Subject Descriptor pattern:

- Encapsulates subject attributes

  Subject Descriptor allows a collection of attributes to be handled as a single object. New types of attributes can be added without modifying the Subject Descriptor or code which uses it.

- Provides a point of access control

  Subject Descriptor allows construction of Attribute Lists including access control functionality to ensure that unauthorized callers will not have access to confidential attributes (such as authentication tokens).

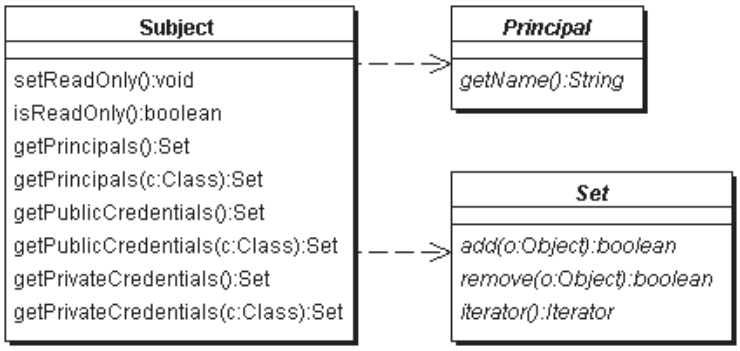
Implementation

When implementing Subject Descriptor, it may be helpful to choose a hierarchical representation for the attribute type. This helps extensibility in that you can have broad categories of attributes (for example, ‘‘identity’’ for all attributes which are some type of name) which can be subdivided into more specific categories (for example, ‘‘group identity’’, or even more specific ‘‘UNIX group ID number’’). Callers can then select attributes at varying levels of abstraction choosing which is most suitable for their specific purpose.

Class names are a ready-made hierarchy which may be suitable.

Known Uses

1. JAAS (Java Authentication and Authorization Service) javax.security.auth.Subject

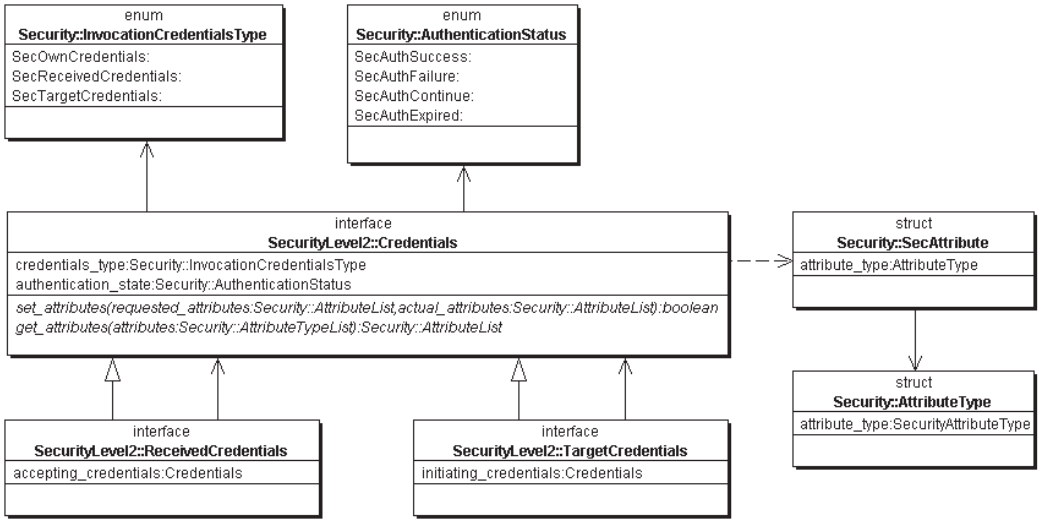


JAAS divides the subject attributes into three collections: principals, public credentials, and private credentials. Principals (which might be better called identities, but the class name ‘‘Identity’’ was already taken) are used to represent user identities and also groups and roles. There is a defined interface to Principal objects, allowing a name to be retrieved without requiring the specific implementing class to be known. Public and private credentials, on the other hand, are arbitrary Java objects and have no defined interface.

Principals and public credentials may be retrieved by any caller which has a reference to the Subject object. Private credentials require a permission to be granted in order to access them, which may be specified down to the granularity of a particular credential object class within Subjects having a particular Principal class with a particular name.

The JAAS Subject class includes a method to set a read-only flag which specifies that the Sets of Principals returned will be read-only (that is, the *add*( ) and *remove*( ) methods will fail). This is useful where a privileged caller gets a reference to a Subject object which it then wishes to pass on to an untrusted recipient.

2. CORBASecurity SecurityLevel2::Credentials



CORBASecurity credentials lists encapsulate subject attributes. CORBASecurity associates a set of credentials with each execution context; *OwnCredentials* represent the security attributes associated with the process itself; *ReceivedCredentials* represent the security attributes associated with a communications session within which the process is the receiver; and *TargetCredentials* represent the security attributes which will be used to represent the process to a partner in a communications session within which the process is the sender.

Related Patterns

- Security Context [TG_SDP] uses Subject Descriptor to represent the attributes of subjects.
- Subject Descriptor uses Iterator [GoF] to return a subject's attributes to callers.
- Role-based access control [APLRAC].
- Security models, authorization, role-based access, multi-level security [Fernandez-Pan].
- The role object pattern [Role Object].
- Enabling application security, roles [Yoder-Barcalow].