Security Domain Membership Management

Status Update
Konstantin Beznosov, Concept Five Technologies
Tadashi Kaji, Hitachi
December 12, 2000
OMG document number orbos/00-12-07
Objects and Domains

<table>
<thead>
<tr>
<th>Obj A</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Obj B</td>
<td></td>
</tr>
<tr>
<td>Obj C</td>
<td></td>
</tr>
<tr>
<td>Obj D</td>
<td></td>
</tr>
</tbody>
</table>

Domain types

- D
- B

D1
D2
D2n
B1
B2
Bn

D21
D2n
B21
B2n

Root

N
Main Parts

- Object to Domain Mapping (ODM)
- Support for “owner”
- Domain Composition Management
Domain Composition Management
Defined Interfaces for Domain Management

- Security Domain::Domain Authority
  - get_domain_name()
  - get_parent_domain_managers()
  - get_child_domain_managers()
  - get_child_domain_names()
  - is_root()
  - get_authority_policy()

- Security Domain::Domain Authority Admin
  - add_child_domain_manager()
  - delete_domain_association()
  - set_domain_policy()
  - delete_domain_policy()
  - set_authority_policy()
  - delete_authority_policy()

- DomainManager
  - get_domain_policy() (from CORBA)

- Security Domain::DomainManagerFactory
  - get_domain_manager()
  - get_root_domain_manager()
  - create_domain_manager()
  - create_and_associate_domain_manager()
  - delete_domain_manager()
  - to_string()
  - to_name()
Object to Domain Mapping Management
Design Goals

- Support for security aware and unaware applications
- Fit into the philosophy of POA architecture
- Generic enough to support other object adapters
- Specific enough to work well with POA
- Have minimum impact on POA and other adapters
ODM Managers and Policies

SecurityLevel2::Current -> Object Mapping Manager

Default ODM Manager
- Adapter Id, Object Id
- Adapter Id, Object Id
- Adapter Id, Object Id
- Adapter Id, Object Id

Security Service Runtime

Domains

Application

ODMM Policy

Object Adapter A
- Object Id
- Object Id
- Object Id
- Object Id

User-supplied servant

User-supplied ODM Manager
- Object Id
- Object Id
- Object Id

User-supplied servant

ODM Mapping Manager

User-supplied servant

ODMM Policy

Object Adapter B
- Object Id
- Object Id
- Object Id
- Object Id

User-supplied servant

User-supplied ODM Manager
- Object Id
- Object Id
- Object Id

User-supplied servant

ODMM Policy

Object Adapter C
- Object Id
- Object Id
- Object Id
- Object Id

User-supplied servant

User-supplied ODM Manager
- Object Id
- Object Id
- Object Id

User-supplied servant

ODMM Policy

Object Adapter D
- Object Id
- Object Id
- Object Id
- Object Id

User-supplied servant

ODM Manager
- User-supplied
- ORB-supplied
- Security Service supplied

SDMM

User-supplied
Using Default ODM Manager

- **Application**: 
- **Security Interceptor**: 
- **Object Adapter A**: 
- **Object Mapping Manager**: 
- **Policy Manager**: 
- **Default Manager**: 

```
create_adapter

get_policy(policy_type: in CORBA::PolicyType)

_narrow

_get_manager

_get_object_mapping_manager

_narrow

_get_default_object_domain_mapping_manager

_narrow

get_domains(POA, ObjectID, Servant)

Invokes with Adapter-specific arguments

Returns "nil" indicating the lack of application-specific manager

Narrows to Adapter-specific interface

Involves with Adapter-specific arguments
```

- **SDMM**
- **User-supplied**
- **ORB-supplied**
- **Security Service supplied**
Using Application ODM Manager

```
resolve_initial_references("Security Current")

_get_object_mapping_manager

create_object_domain_mapping_manager_policy(application_manager : DynamicAttributeManager)

add_policy_to_adapter(in CORBA::PolicyList)

get_policy(policy_type : in CORBA::PolicyType)

_narrow

_narrow

get_domains()

Invokes the operation with adapter-specific arguments

narrow Policy to ObjectDomainMappingManagerPolicy

narrow ODMM to Adapter-specific Interface

SDM_USER-SUPPLIED

SDM ORB-SUPPLIED

SDM SECURITY-SERVICE-SUPPLIED

Application

ORBIT: (Current)

ObjectMappingManager

SecurityInterceptor

Object Adapter

Policy: ObjectDomainMappingManagerPolicy

Application-supplied Manager: ObjectDomainMappingManager
```
• Separate IDL module
• POA-specific operation signatures
Using Application ODM Manager for POA

```
Application ORB ObjectMappingManager Parent POA: Security Interceptor POA: POA Policy: DynamicAttributeManagerPolicy Application Provided Manager: (ObjectDomainMappingManager)

resolve_initial_references("SecurityCurrent")

_get_object_mapping_manager

create_object_domain_mapping_manager_policy(application_manager : DynamicAttributeManager)

get_policy(policy_type : in CORBA::PolicyType)

_narrow

_get_manager

Narrows to POA-specific interface

get_domains (POA, ObjectId, Servant)

Invokes with POA-specific arguments

narrow Policy to ObjectDomainMappingManagerPolicy

SDMM

User-supplied

ORB-supplied

Security Service supplied

Copyright © 2000, Concept Five Technologies, Inc. All rights reserved.
Support for “owner” concept
The Need for supporting ‘owner’

1. Why to support “owner”
   - Common access control policy case in today businesses:
     • “Only ‘owner’ can withdraw money from the account”
     • “Only ‘bank manager’ or ‘owner’ of the account can close it”

2. Can’t CORBASEC do it already? It can BUT:
   - only with heavy-weight domains
     • A separate domain for each ‘owner’
   - It requires domain-by-domain explicit configuration of access policies
     • If AccessId == ‘johnsmith’ then allow ‘withdraw money’ for objects in domain ‘johnsmith’

3. Why in this submission?
   - Object-to-owner mapping is very relevant to object-to-domain mapping
   - Could be the same approach as for ODM
Observations about owner in CORBASEC

1. ‘Owner’ – a particular case of more general relationship property

2. Needs integration with the existing CORBASEC authorization model
   - required/granted rights and privilege attributes

3. Should not be a mandatory feature of CORBASEC implementations
Submitters’ Approach

1. Go from particular case of ‘owner’ to the general one of ‘relationship’
2. Present relationships as dynamic privilege attributes
3. Specify dynamic attribute manager (DAM) run-time interface(s) to determine dynamic attributes of the calling principal in the context of the request on a given object.
4. Enable support for security-aware and security-unaware applications.
Dynamic Attribute Managers and Policies

Security Level: Current

Object Mapping Manager

Default DAM

Adapter ID | Object ID | Static Attributes | Dynamic Attributes
--- | --- | --- | ---

Security Service Runtime

Runtime

User-supplied servant

DAM Policy

Object Adapter A

Object ID

DAM Policy

Object Adapter B

Object ID

DAM Policy

Object Adapter C

Object ID

DAM Policy

Object Adapter D

Object ID
Using Application DAM for POA

- An object mapping manager is created with the object manager and the parent POA.
- A dynamic attribute manager policy is created with the policy: DynamicAttributeManagerPolicy.
- The ORB gets the object mapping manager.
- The ORB gets the initial references to the POA.
- The ORB narrows to the POA SDMM dynamic attribute manager.
- The SDMM gets dynamic attributes with POA-specific arguments.
## Illustrating Example

### Required Rights Table

<table>
<thead>
<tr>
<th>Interface</th>
<th>Operation</th>
<th>Required Rights</th>
</tr>
</thead>
<tbody>
<tr>
<td>BankAccount</td>
<td>withdraw</td>
<td>u (any)</td>
</tr>
<tr>
<td>BankAccount</td>
<td>get_balance</td>
<td>gu (any)</td>
</tr>
<tr>
<td>BankAccount</td>
<td>close</td>
<td>mu (any)</td>
</tr>
</tbody>
</table>

### Granted Rights Table

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
<th>Granted Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>role</td>
<td>manager</td>
<td>m</td>
</tr>
<tr>
<td>relationship</td>
<td>owner</td>
<td>u</td>
</tr>
<tr>
<td>relationship</td>
<td>owner’s spouse</td>
<td>g</td>
</tr>
</tbody>
</table>

1. ‘manager’ can close account
2. ‘owner’s spouse’ can see the balance
3. ‘owner’ can see the balance, withdraw money, and close the account
Open Issues

• ODMM admin is adapter-specific
  – Assumption that only one type of object adapter is used by each application

• Security interceptor is provided with the ID of the object adapter that serves the object in question
Things to do

• Admin interfaces for ODM and DAM
• ...

Submitters Meeting Information

Date: Thursday (December 15)
Time: 1PM – 5PM
Place: BOCA I