CONCEPTFIVE TECHNOLOGIES

Security Domain Membership Management

Status Update

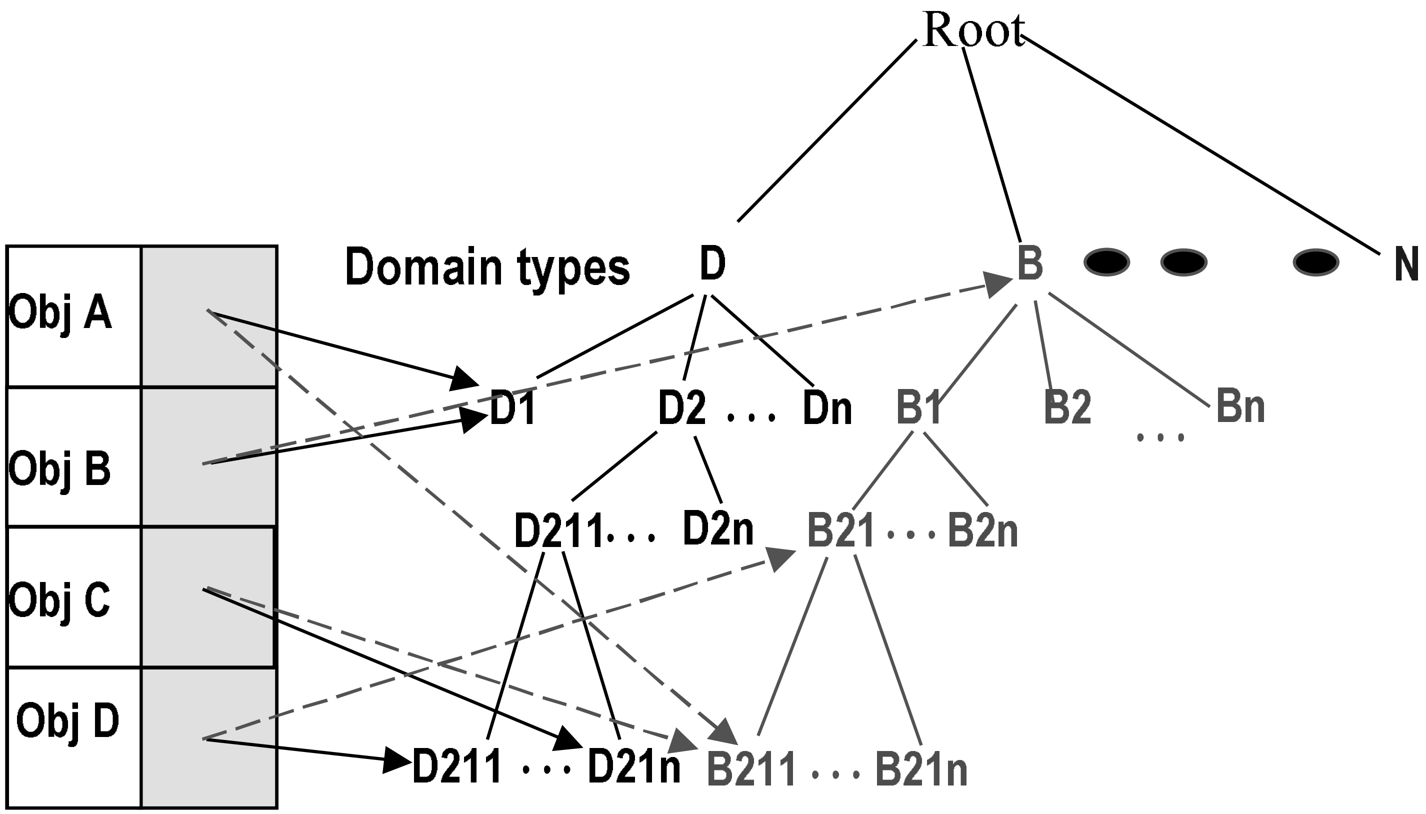
Konstantin Beznosov, Concept Five Technologies

Tadashi Kaji, Hitachi

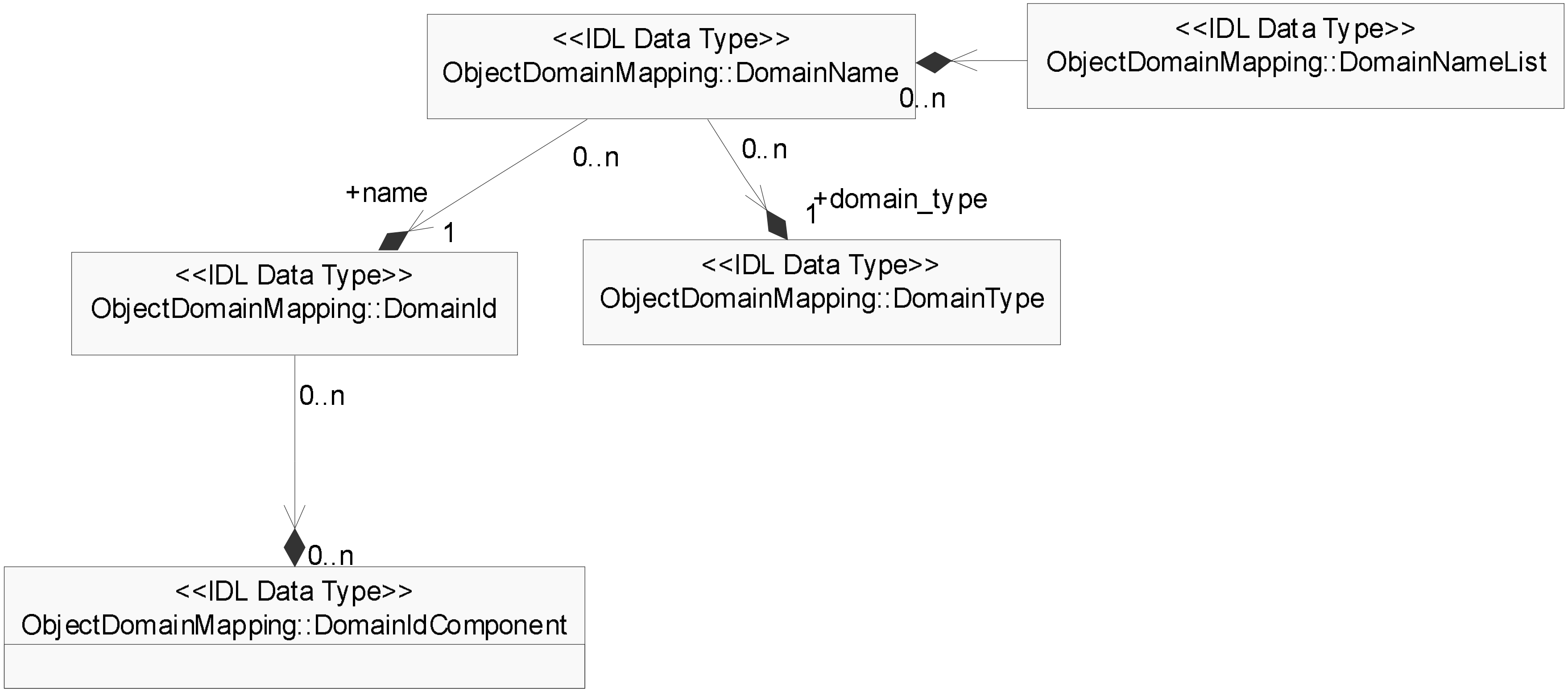
December 12, 2000

OMG document number orbos/00-12-07

Objects and Domains



Domain Representation



<<IDL Data Type>>
ObjectDomainMapping::DomainName
<<IDL Data Type>>
ObjectDomainMapping::DomainNameList
0..n
0..n
+name
1
0..n
1
+domain_type
<<IDL Data Type>>
ObjectDomainMapping::DomainId
<<IDL Data Type>>
ObjectDomainMapping::DomainType
0..n
0..n
<<IDL Data Type>>
ObjectDomainMapping::DomainIdComponent

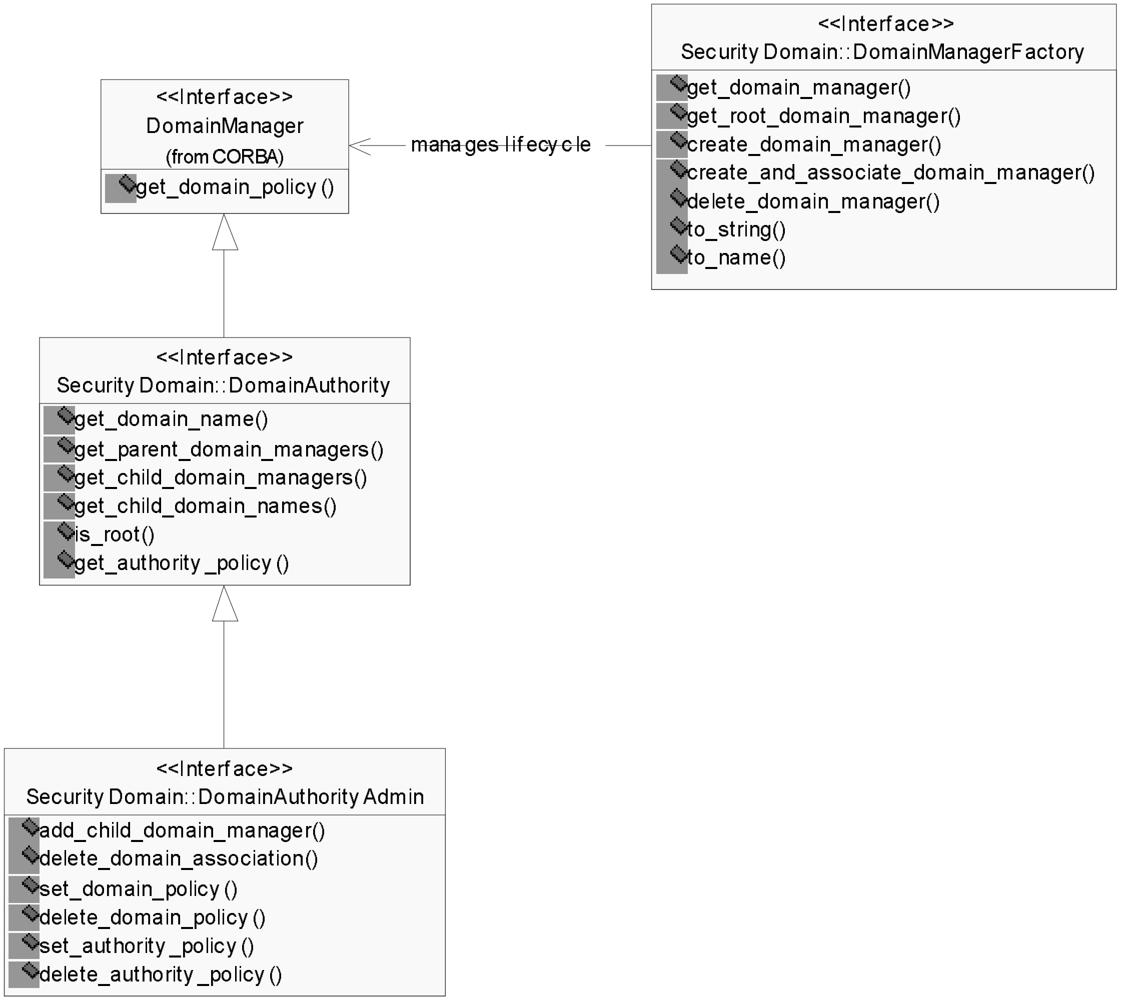
Main Parts

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

CONCEPTFIVE
TECHNOLOGIES

Domain Composition Management

Defined Interfaces for Domain Management



<<Interface>>
DomainManager
(from CORBA)

- get_domain_policy ()

manages lifecycle

<<Interface>>
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()

<<Interface>>
Security Domain::DomainAuthority

- get_domain_name()
- get_parent_domain_managers()
- get_child_domain_managers()
- get_child_domain_names()
- is_root()
- get_authority_policy ()

<<Interface>>
Security Domain::DomainAuthority Admin

- add_child_domain_manager()
- delete_domain_association()
- set_domain_policy ()
- delete_domain_policy ()
- set_authority_policy ()
- delete_authority_policy ()

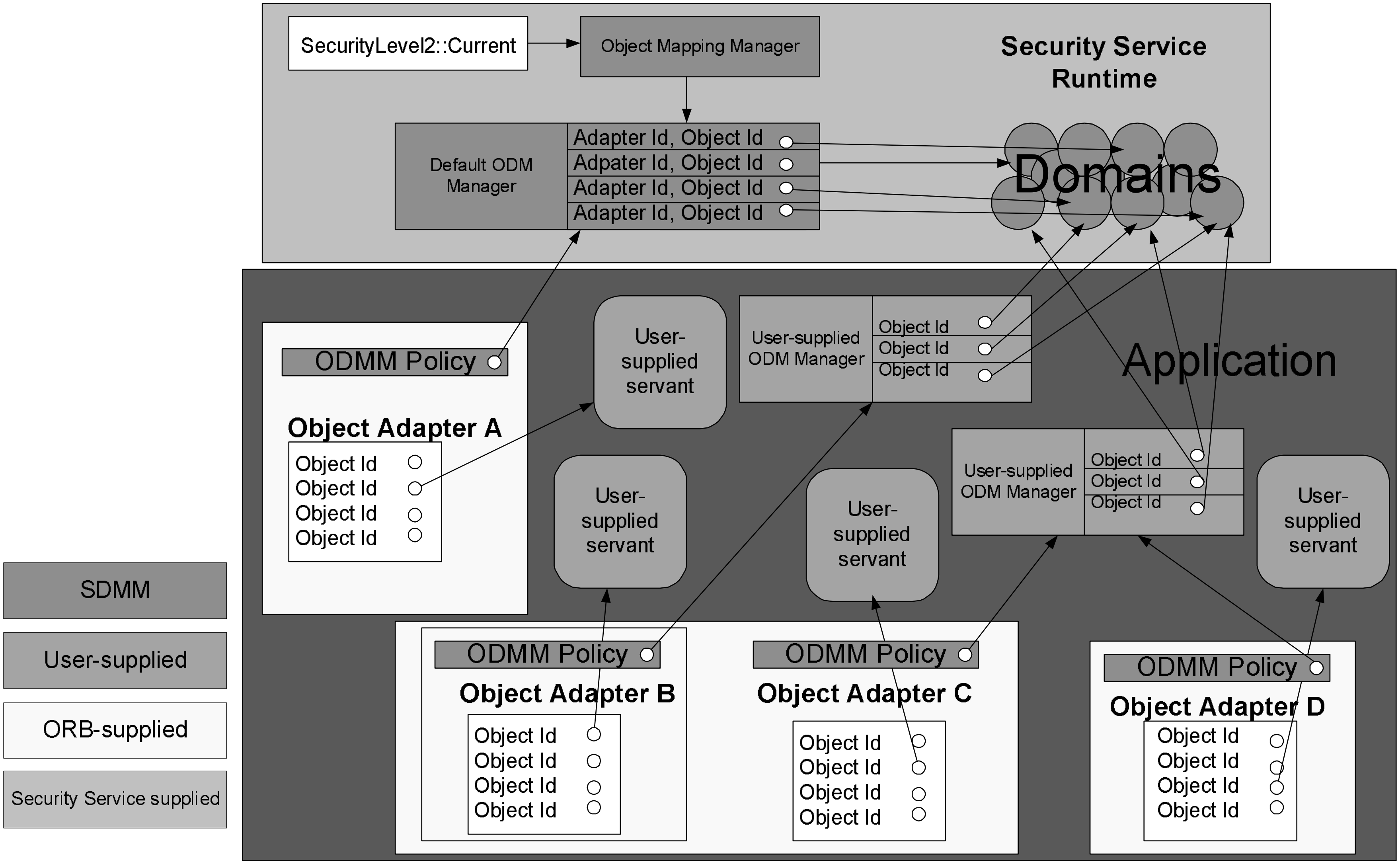
CONCEPTFIVE
TECHNOLOGIES

Object to Domain Mapping Management

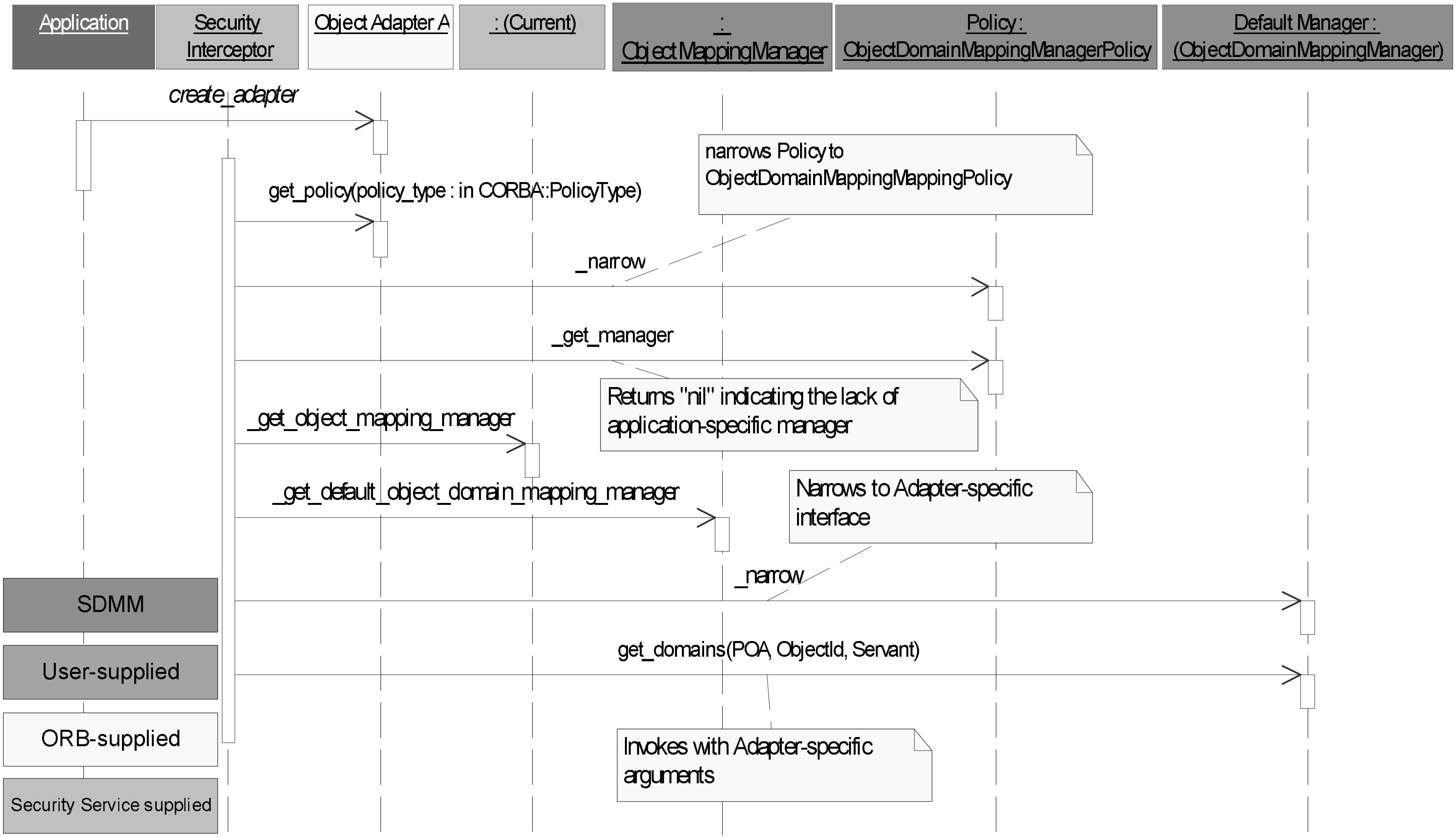
Design Goals

- 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



Using Default ODM Manager


Application
Security Interceptor
Object Adapter A
: (Current)
: ObjectMappingManager
Policy : ObjectDomainMappingManagerPolicy
Default Manager : (ObjectDomainMappingManager)
create_adapter
get_policy(policy_type : in CORBA::PolicyType)
_narrow
_get_manager
_get_object_mapping_manager
_get_default_object_domain_mapping_manager
_narrow
get_domains(POA, ObjectId, Servant)
SDMM
User-supplied
ORB-supplied
Security Service supplied

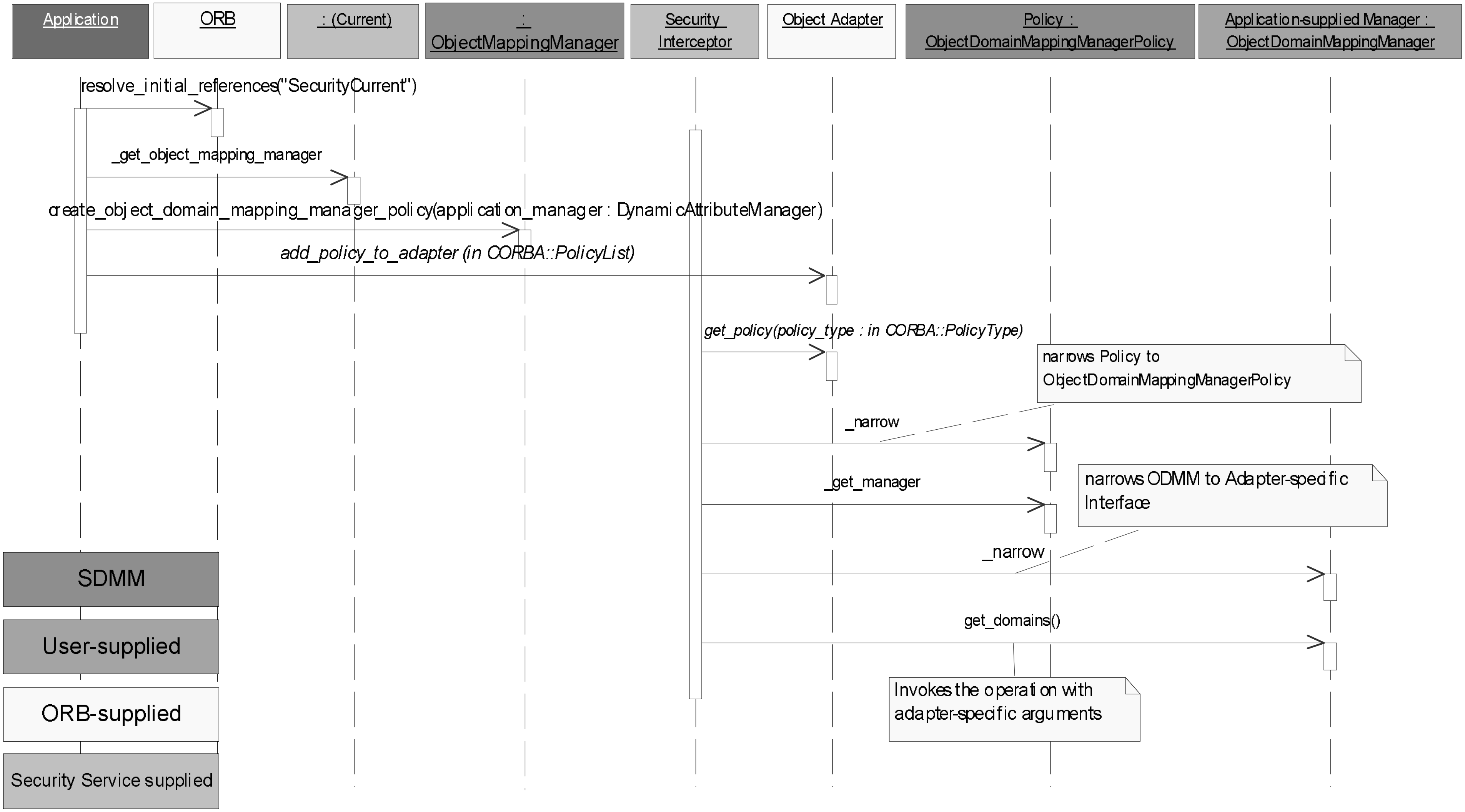
narrows Policy to ObjectDomainMappingMappingPolicy

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

Narrows to Adapter-specific interface

Invokes with Adapter-specific arguments

Using Application ODM Manager



Application

ORB

: (Current)

: ObjectMappingManager

Security Interceptor

Object Adapter

Policy : ObjectDomainMappingManagerPolicy

Application-supplied Manager : ObjectDomainMappingManager

resolve_initial_references("SecurityCurrent")

_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)

narrows Policy to ObjectDomainMappingManagerPolicy

_narrow

_get_manager

narrows ODMM to Adapter-specific Interface

_narrow

get_domains()

Invokes the operation with adapter-specific arguments

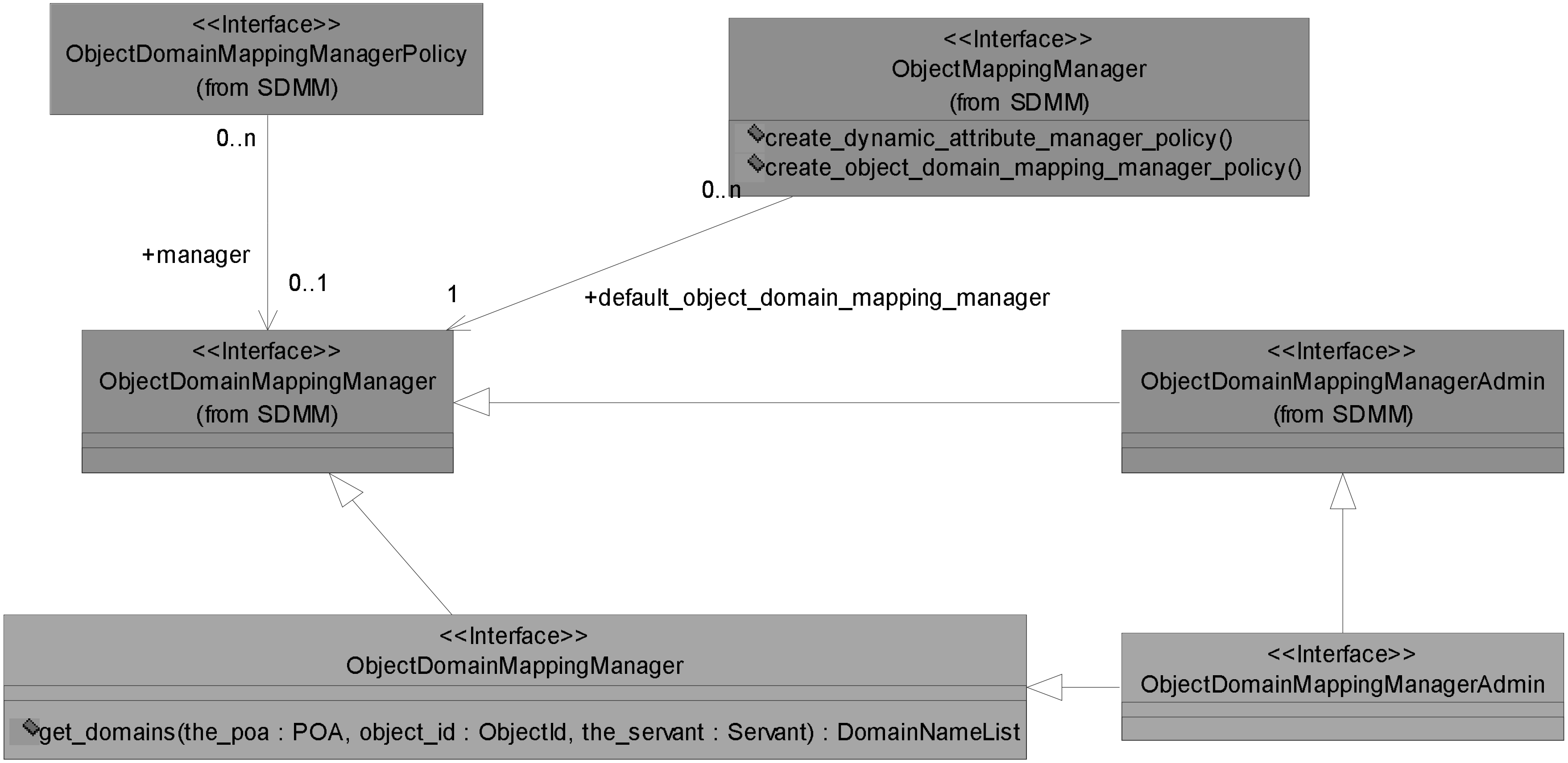
SDMM

User-supplied

ORB-supplied

Security Service supplied

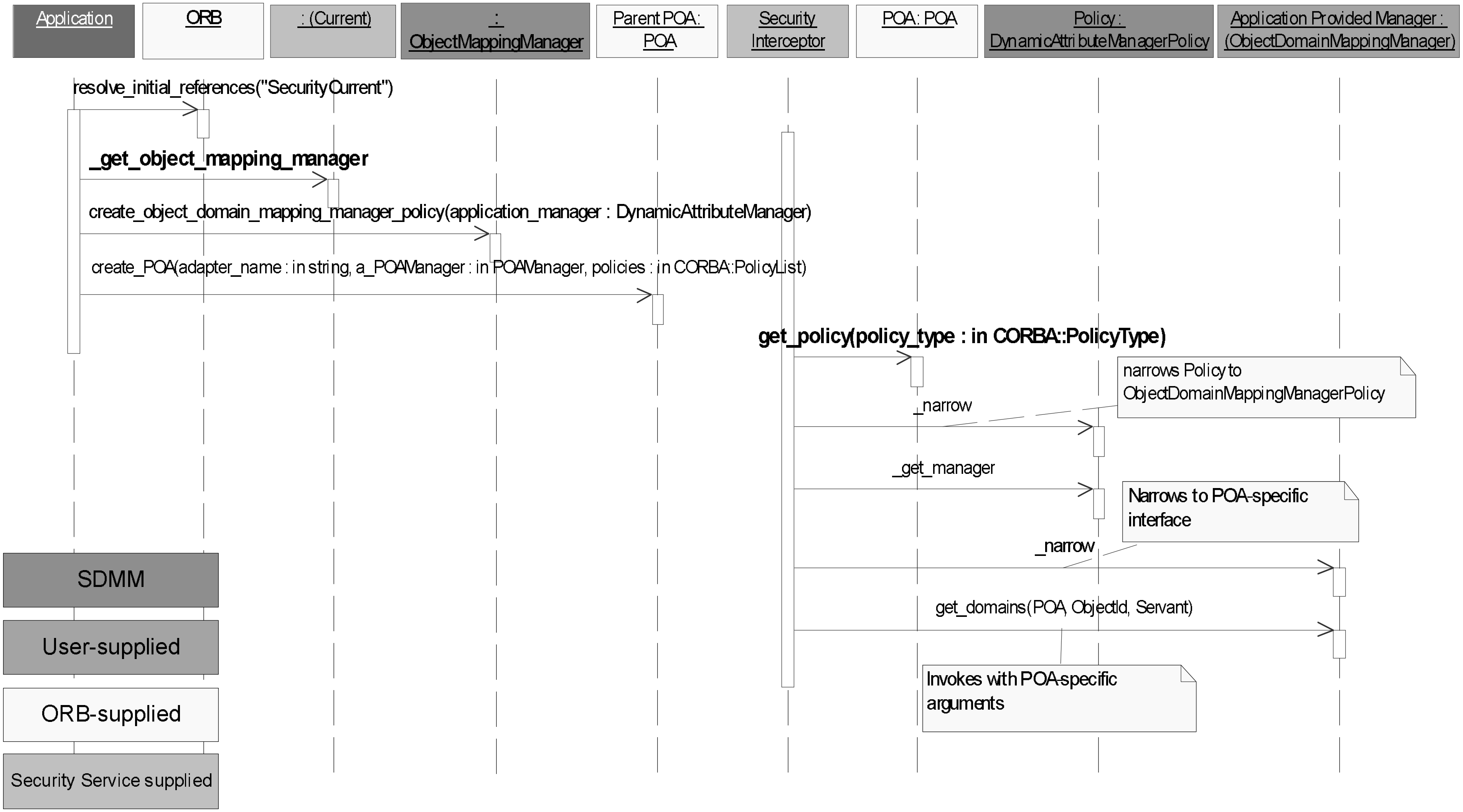
ODM Manager for POA



<<Interface>>
ObjectDomainMappingManagerPolicy
(from SDMM)
<<Interface>>
ObjectMappingManager
(from SDMM)
create_dynamic_attribute_manager_policy()
create_object_domain_mapping_manager_policy()
0..n
0..n
+manager
0..1
1
+default_object_domain_mapping_manager
<<Interface>>
ObjectDomainMappingManager
(from SDMM)
<<Interface>>
ObjectDomainMappingManagerAdmin
(from SDMM)
<<Interface>>
ObjectDomainMappingManager
get_domains(the_poa : POA, object_id : ObjectId, the_servant : Servant) : DomainNameList
<<Interface>>
ObjectDomainMappingManagerAdmin

- Separate IDL module
- POA-specific operation signatures

Using Application ODM Manager for POA



Application

ORB

: (Current)

: ObjectMappingManager

Parent POA : 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)

create_POA(adapter_name : in string, a_POAManager : in POAManager, policies : in CORBA::PolicyList)

get_policy(policy_type : in CORBA::PolicyType)

_narrow

narrows Policy to ObjectDomainMappingManagerPolicy

_get_manager

Narrows to POA-specific interface

_narrow

get_domains(POA, ObjectId, Servant)

Invokes with POA-specific arguments

SDMM

User-supplied

ORB-supplied

Security Service supplied

CONCEPTFIVE
TECHNOLOGIES

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

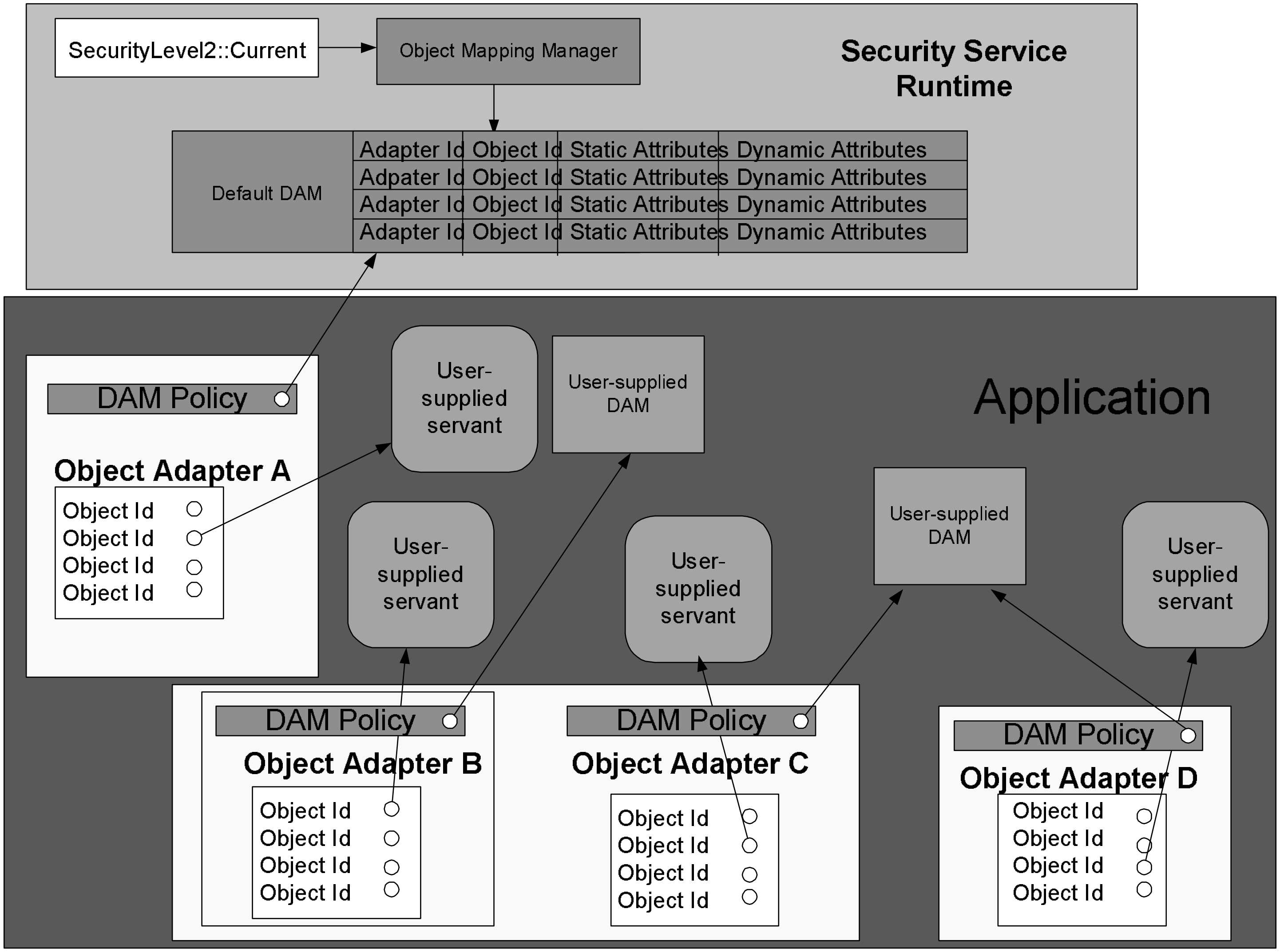
Observations about owner in CORBASEC

1. ‘Owner’ – a particular case of more general relationship property
   - ‘owner’, ‘spouse of owner’, ‘best friend of owner’, ‘secondary holder of the account’, ‘family member’, etc.
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 Service Runtime

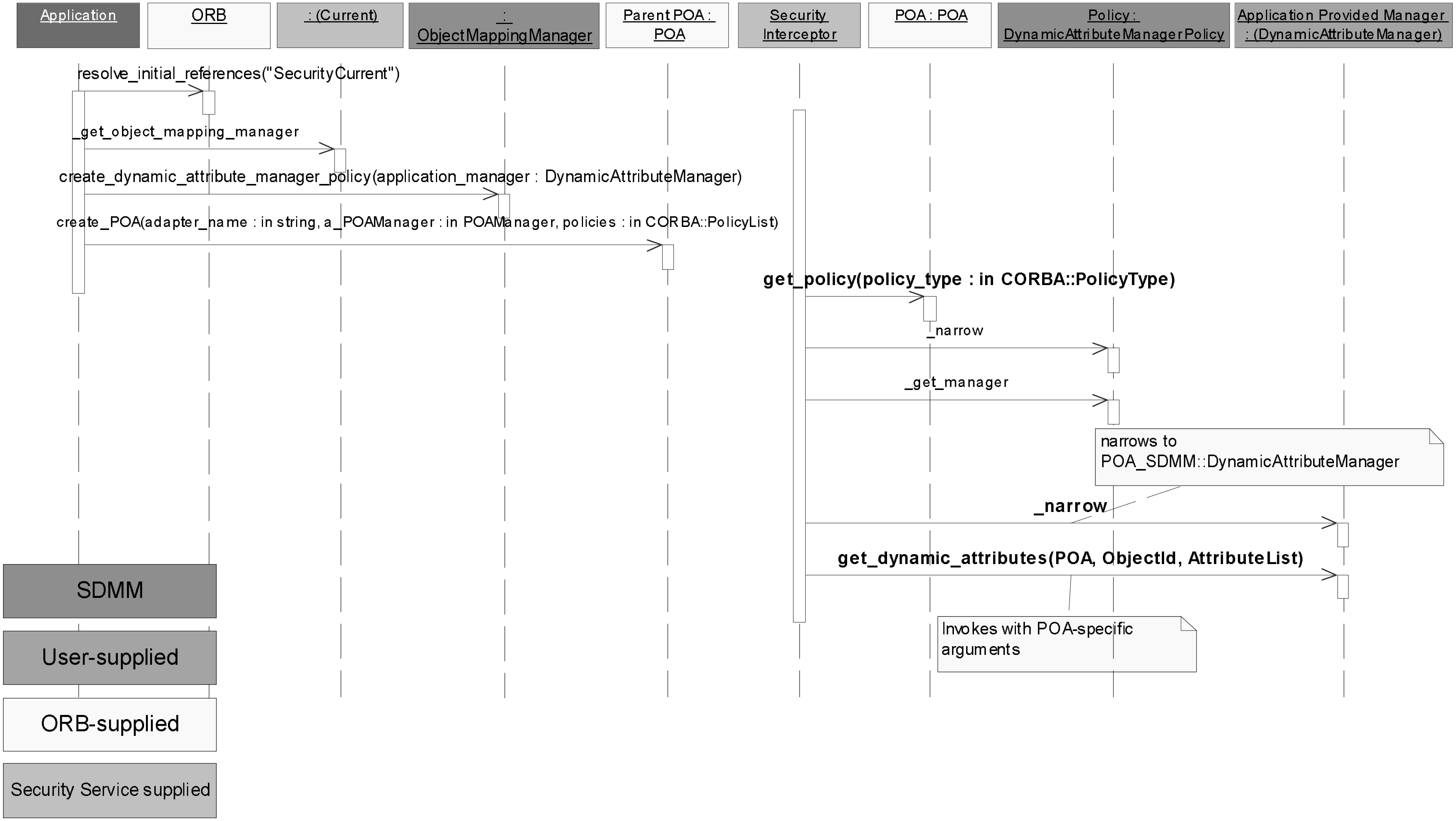
SecurityLevel2::Current

Object Mapping Manager

Default DAM	Adapter Id	Object Id	Static Attributes	Dynamic Attributes
	Adpater Id	Object Id	Static Attributes	Dynamic Attributes
	Adapter Id	Object Id	Static Attributes	Dynamic Attributes
	Adapter Id	Object Id	Static Attributes	Dynamic Attributes

Application

Using Application DAM for POA


Application
ORB
: (Current)
: ObjectMappingManager
Parent POA : POA
Security Interceptor
POA : POA
Policy : DynamicAttributeManagerPolicy
Application Provided Manager : (DynamicAttributeManager)
resolve_initial_references("SecurityCurrent")
_get_object_mapping_manager
create_dynamic_attribute_manager_policy(application_manager : DynamicAttributeManager)
create_POA(adapter_name : in string, a_POAManager : in POAManager, policies : in CORBA::PolicyList)
get_policy(policy_type : in CORBA::PolicyType)
_narrow
_get_manager
_narrow
get_dynamic_attributes(POA, ObjectId, AttributeList)
SDMM
User-supplied
ORB-supplied
Security Service supplied

narrows to POA_SDMM::DynamicAttributeManager

Invokes with POA-specific arguments

Illustrating Example

Required Rights Table

Interface	Operation	Required Rights
BankAccount	withdraw	u (any)
BankAccount	get_balance	gu (any)
BankAccount	close	mu (any)

Granted Rights Table

Attribute Type	Attribute Value	Granted Right
role	manager	m
relationship	owner	u
relationship	owner’s spouse	g

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