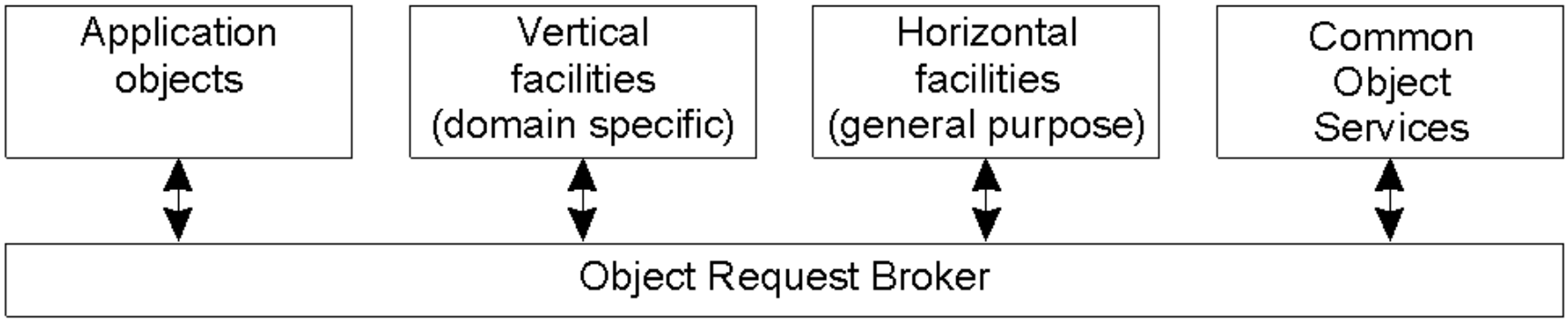
Distributed Object-Based Systems

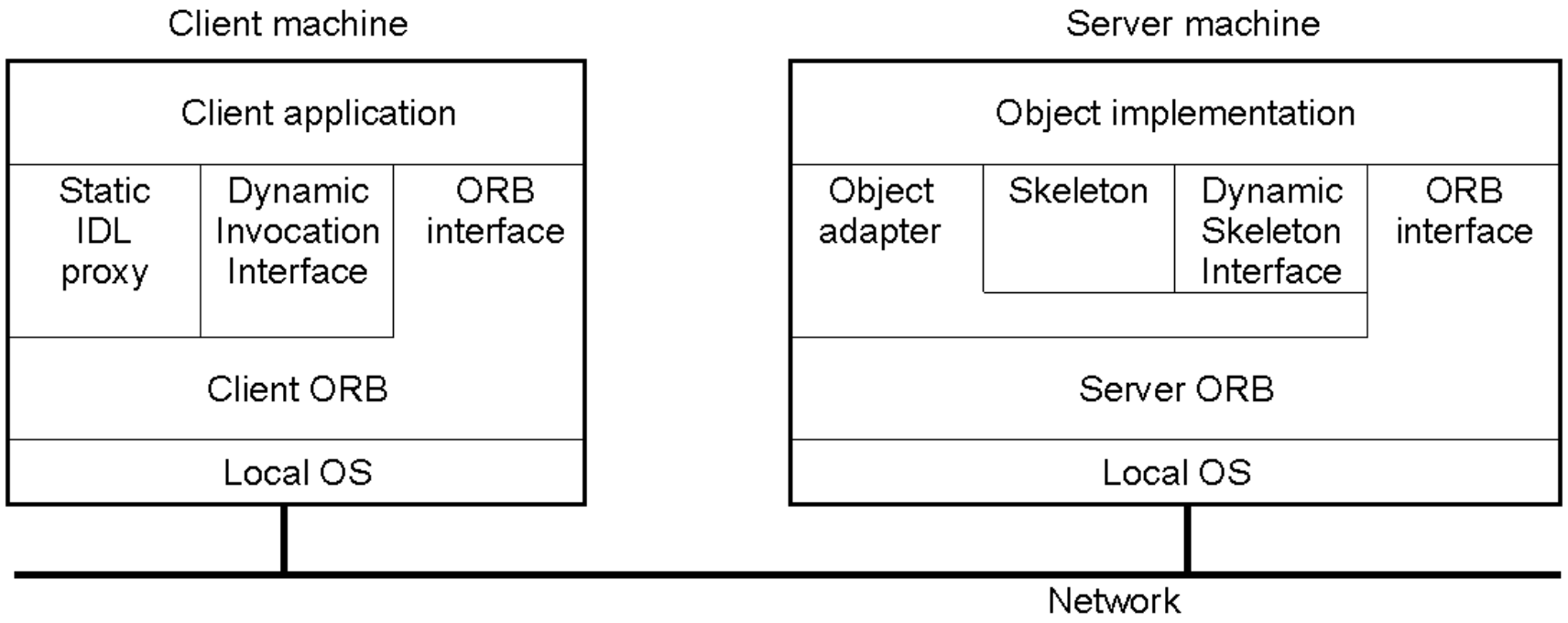
Chapter 9

Overview of CORBA



The global architecture of CORBA.

Object Model



The general organization of a CORBA system.

Corba Services

Service	Description
Collection	Facilities for grouping objects into lists, queue, sets, etc.
Query	Facilities for querying collections of objects in a declarative manner
Concurrency	Facilities to allow concurrent access to shared objects
Transaction	Flat and nested transactions on method calls over multiple objects
Event	Facilities for asynchronous communication through events
Notification	Advanced facilities for event-based asynchronous communication
Externalization	Facilities for marshaling and unmarshaling of objects
Life cycle	Facilities for creation, deletion, copying, and moving of objects
Licensing	Facilities for attaching a license to an object
Naming	Facilities for systemwide name of objects
Property	Facilities for associating (attribute, value) pairs with objects
Trading	Facilities to publish and find the services on object has to offer
Persistence	Facilities for persistently storing objects
Relationship	Facilities for expressing relationships between objects
Security	Mechanisms for secure channels, authorization, and auditing
Time	Provides the current time within specified error margins

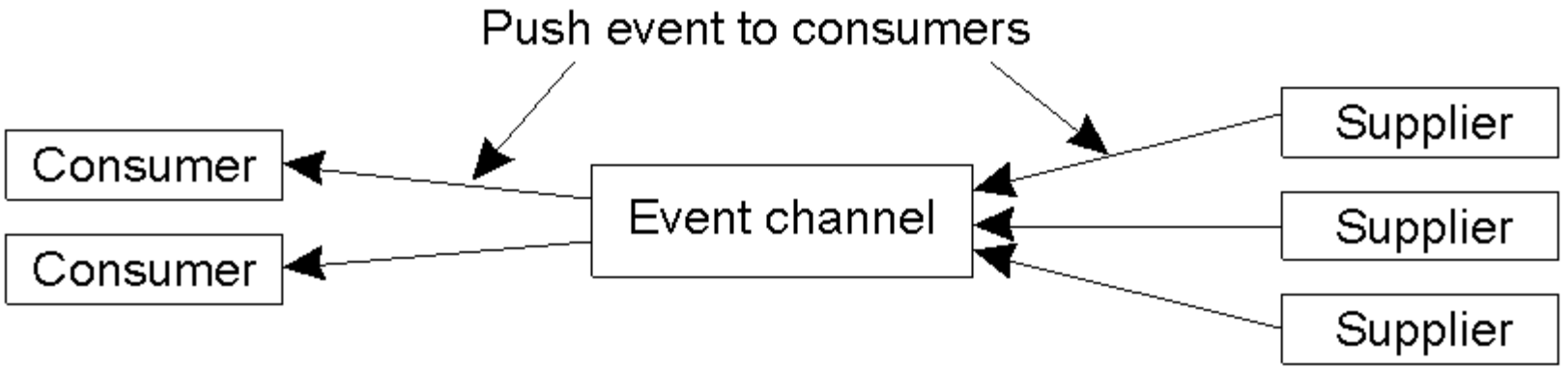
Overview of CORBA services.

Object Invocation Models

Request type	Failure semantics	Description
Synchronous	At-most-once	Caller blocks until a response is returned or an exception is raised
One-way	Best effort delivery	Caller continues immediately without waiting for any response from the server
Deferred synchronous	At-most-once	Caller continues immediately and can later block until response is delivered

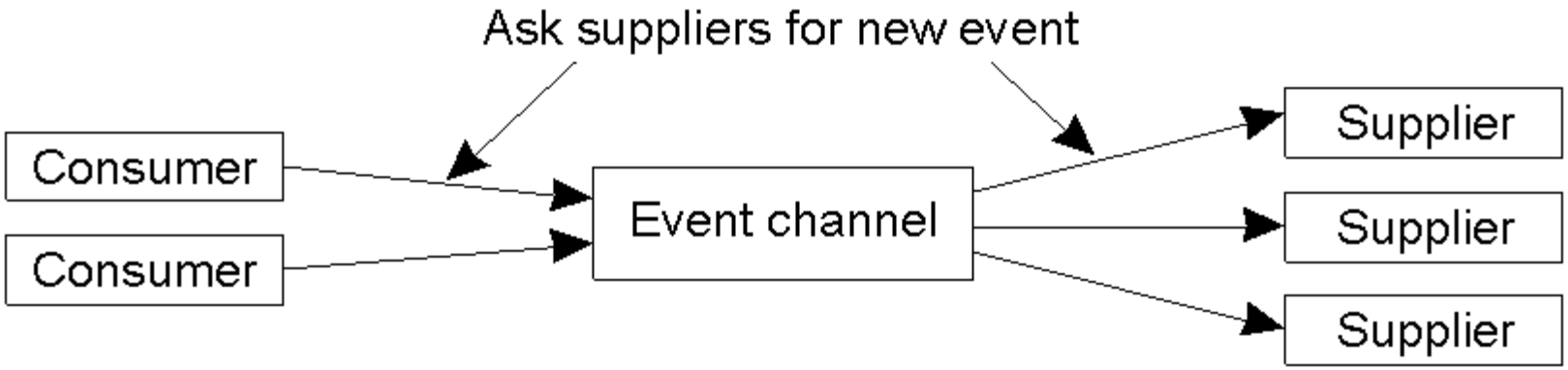
Invocation models supported in CORBA.

Event and Notification Services (1)



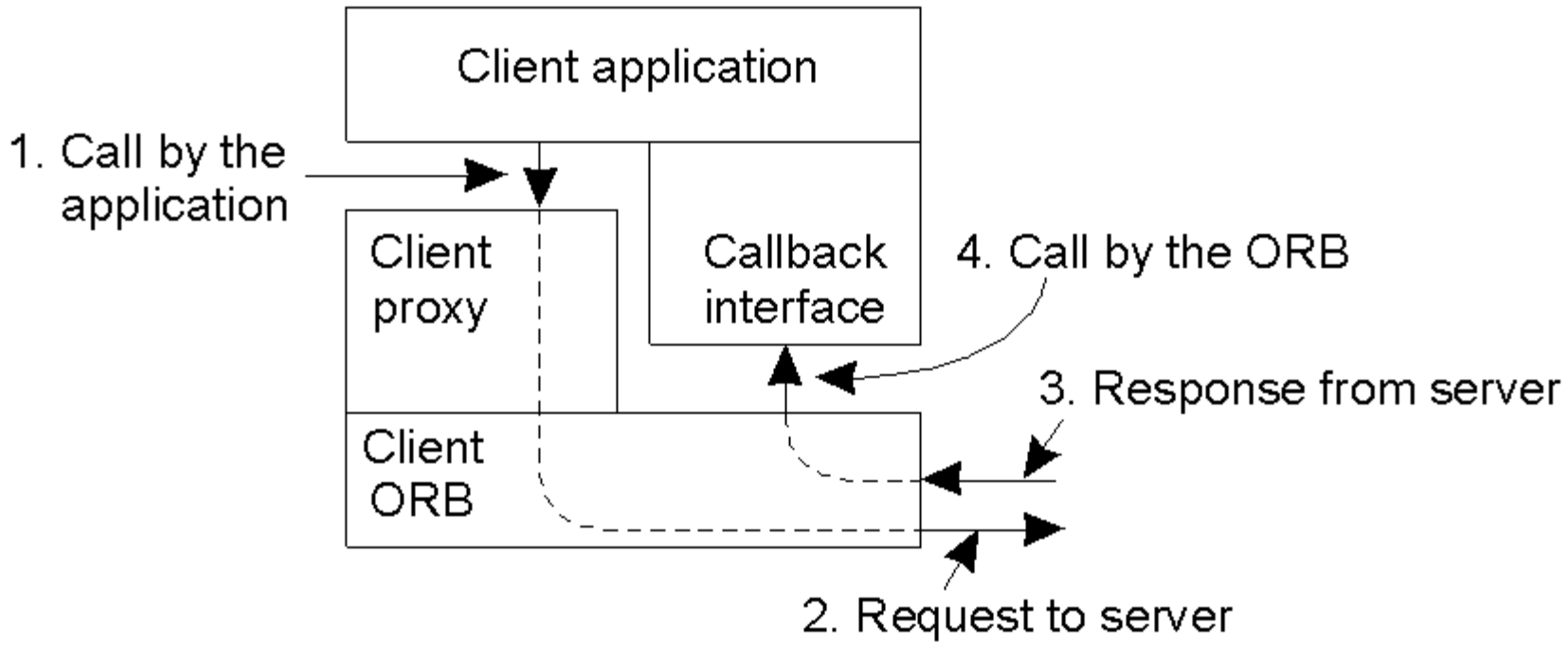
The logical organization of suppliers and consumers of events, following the push-style model.

Event and Notification Services (2)



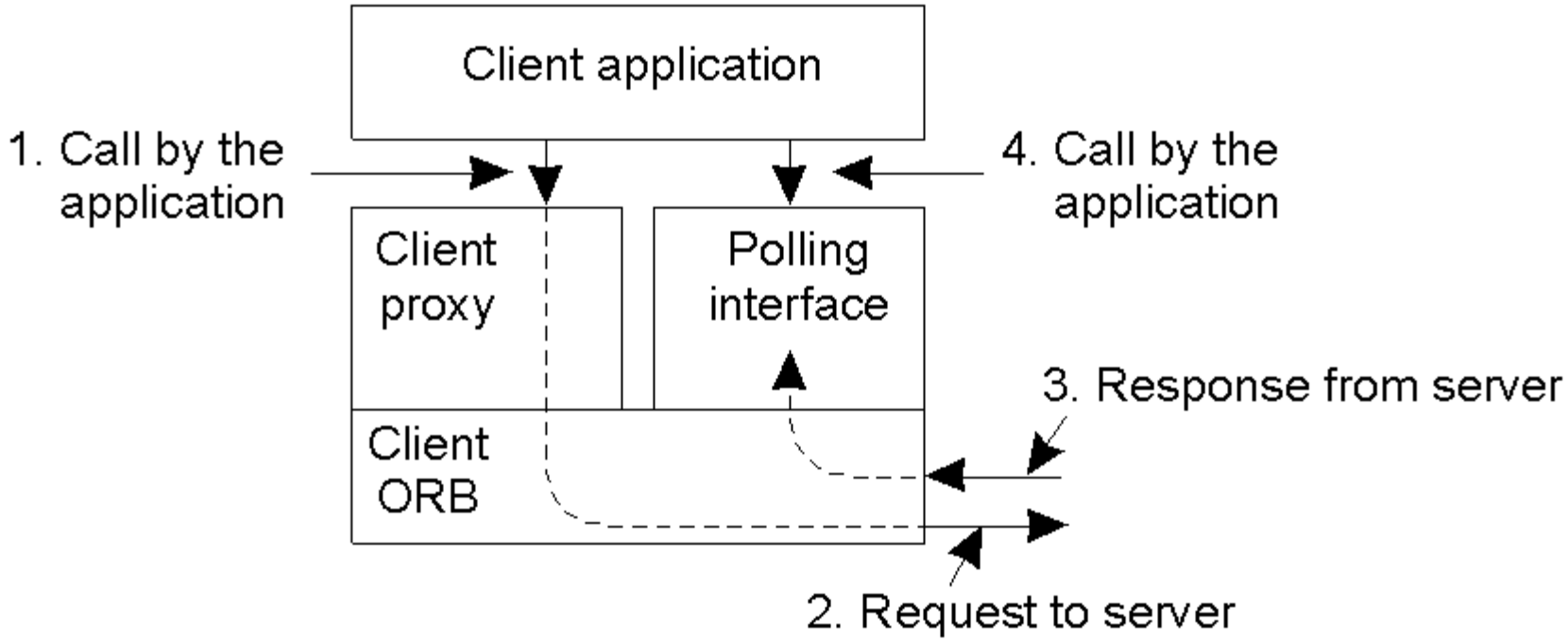
The pull-style model for event delivery in CORBA.

Messaging (1)



CORBA's callback model for asynchronous method invocation.

Messaging (2)



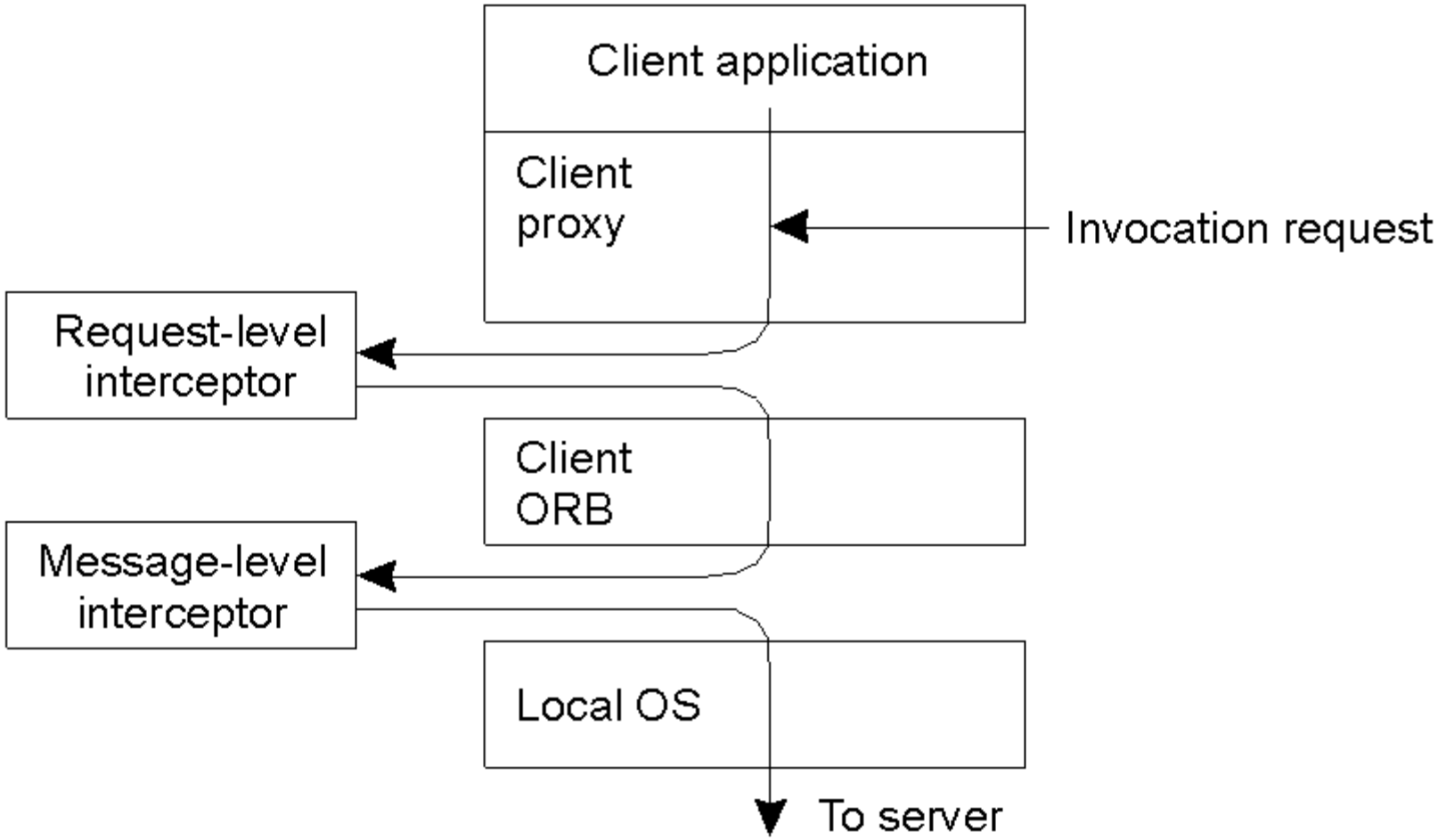
CORBA'S polling model for asynchronous method invocation.

Interoperability

Message type	Originator	Description
Request	Client	Contains an invocation request
Reply	Server	Contains the response to an invocation
LocateRequest	Client	Contains a request on the exact location of an object
LocateReply	Server	Contains location information on an object
CancelRequest	Client	Indicates client no longer expects a reply
CloseConnection	Both	Indication that connection will be closed
MessageError	Both	Contains information on an error
Fragment	Both	Part (fragment) of a larger message

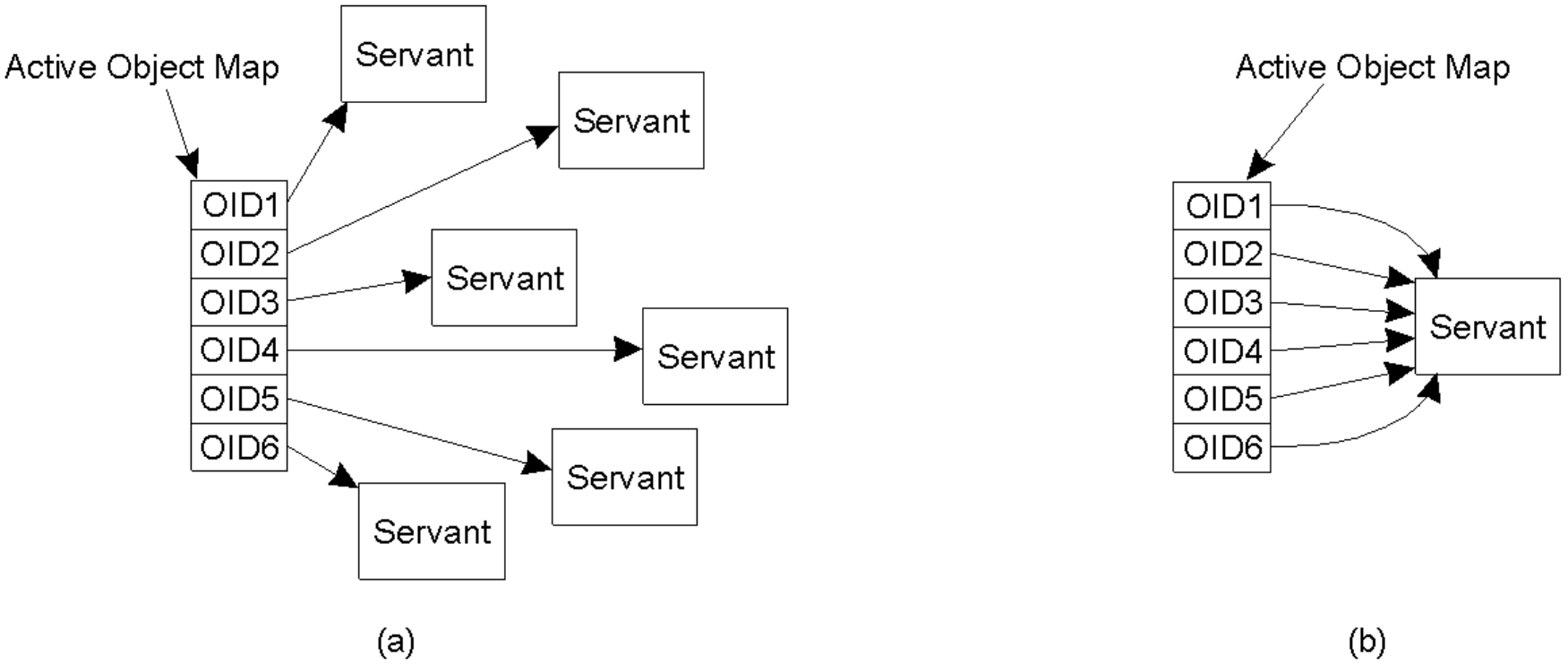
GIOP message types.

Clients



Logical placement of interceptors in CORBA.

Portable Object Adaptor (1)



Mapping of CORBA object identifiers to servants.

a) The POA supports multiple servants.

b) The POA supports a single servant.

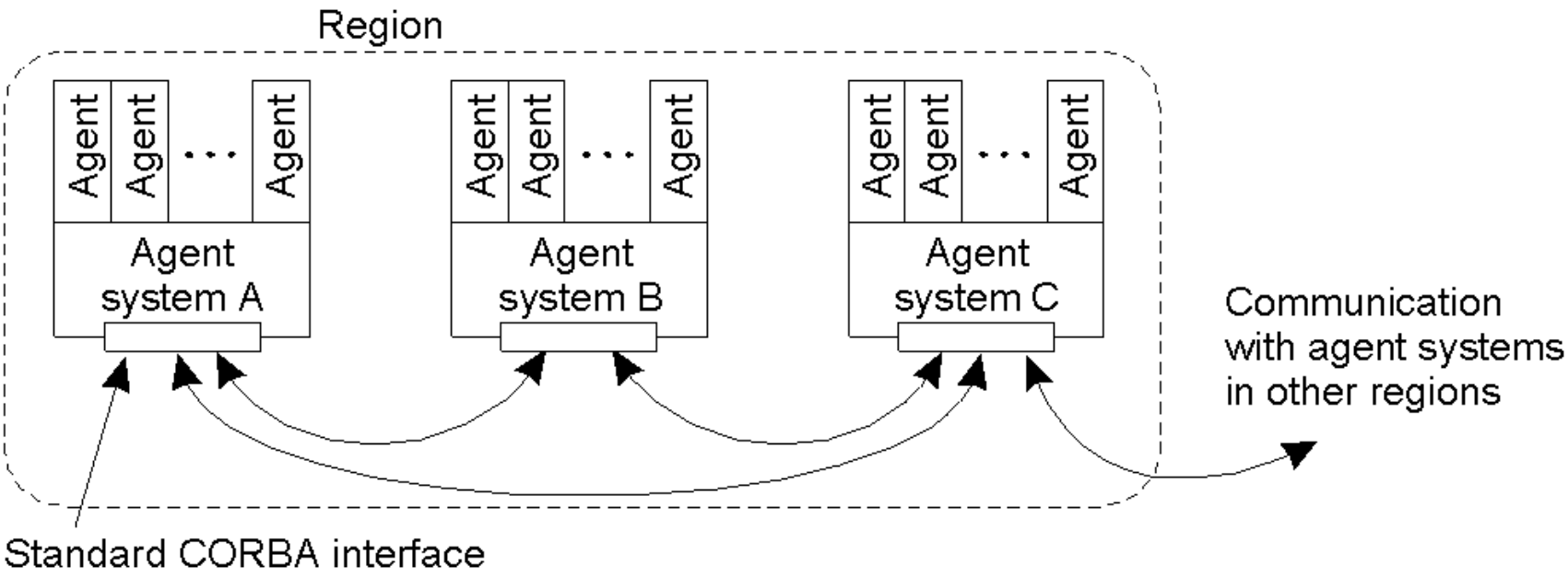
Portable Object Adaptor (2)

```
My_servant *my_object;              // Declare a reference to a C++ object
CORBA::Objectid_var oid;            // Declare a CORBA identifier

my_object = new MyServant;          // Create a new C++ object
oid = poa ->activate_object (my_object);
                                    // Register C++ object as CORBA OBJECT
```

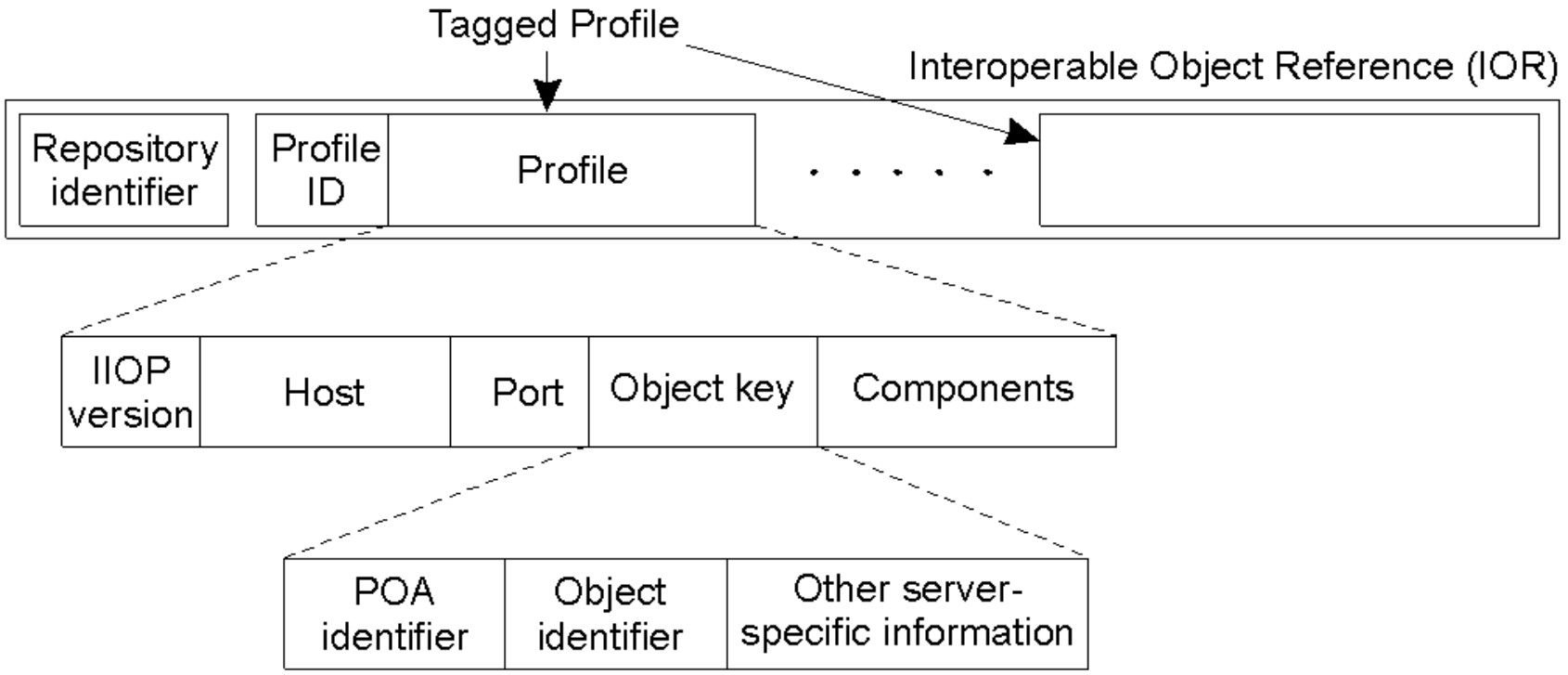
Changing a C++ object into a CORBA object.

Agents



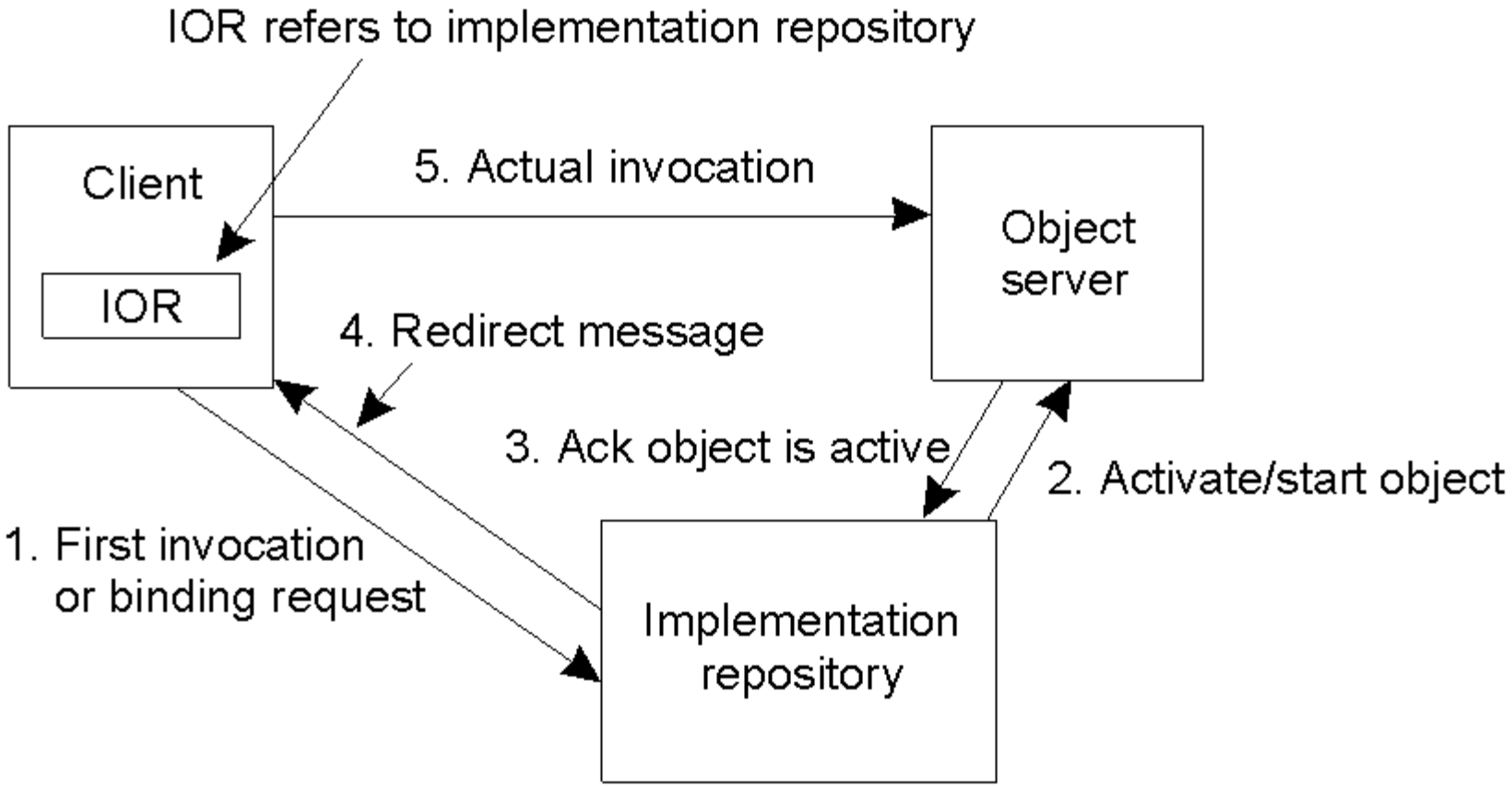
CORBA's overall model of agents, agent systems, and regions.

Object References (1)



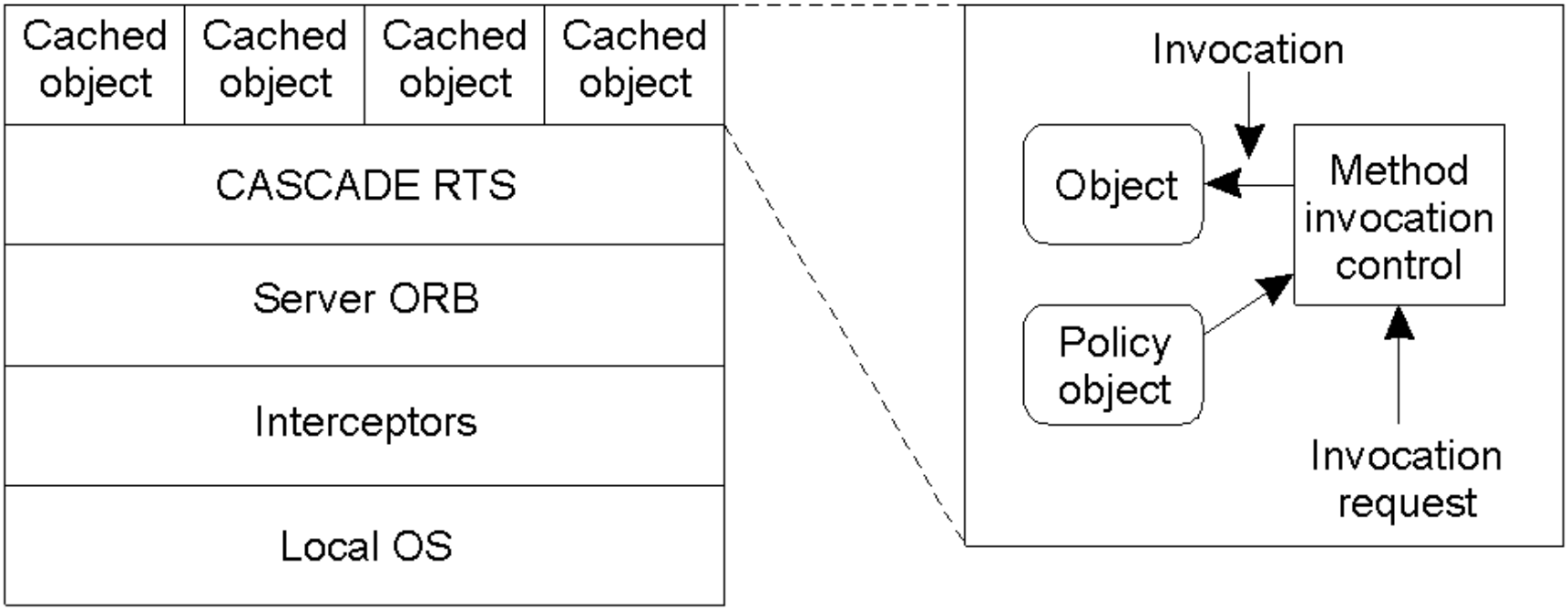
The organization of an IOR with specific information for IIOP.

Object References (2)



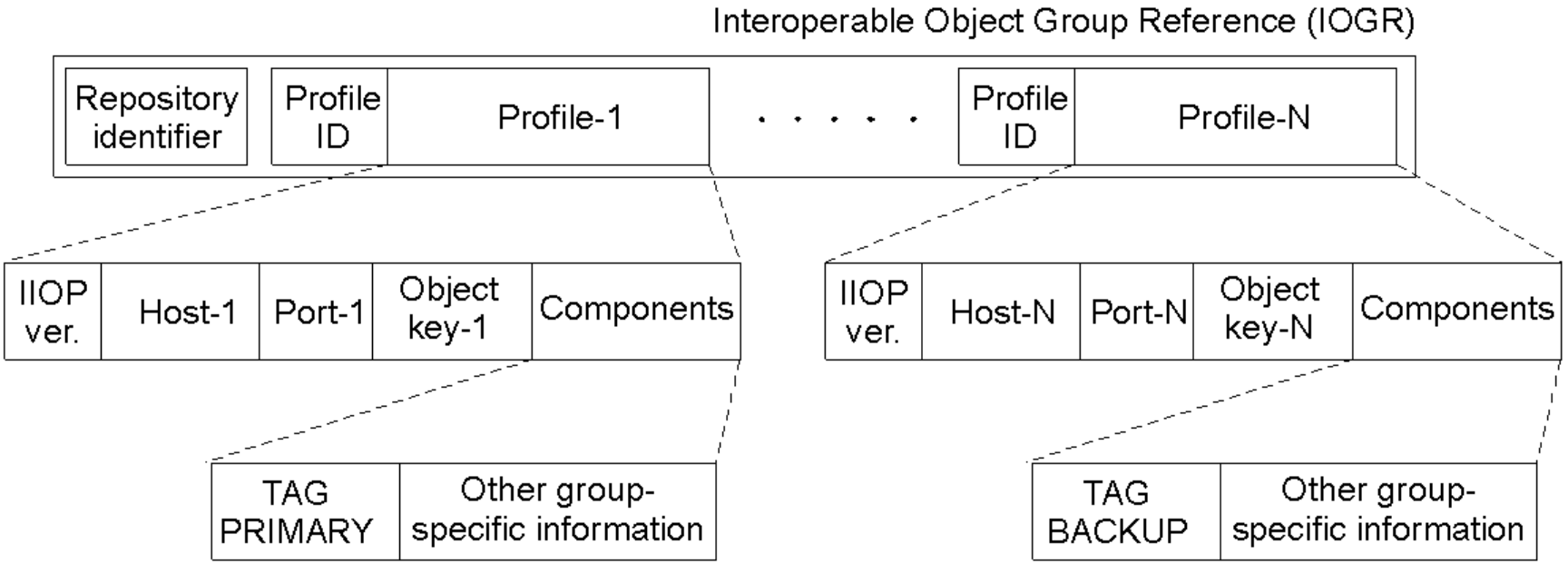
Indirect binding in CORBA.

Caching and Replication



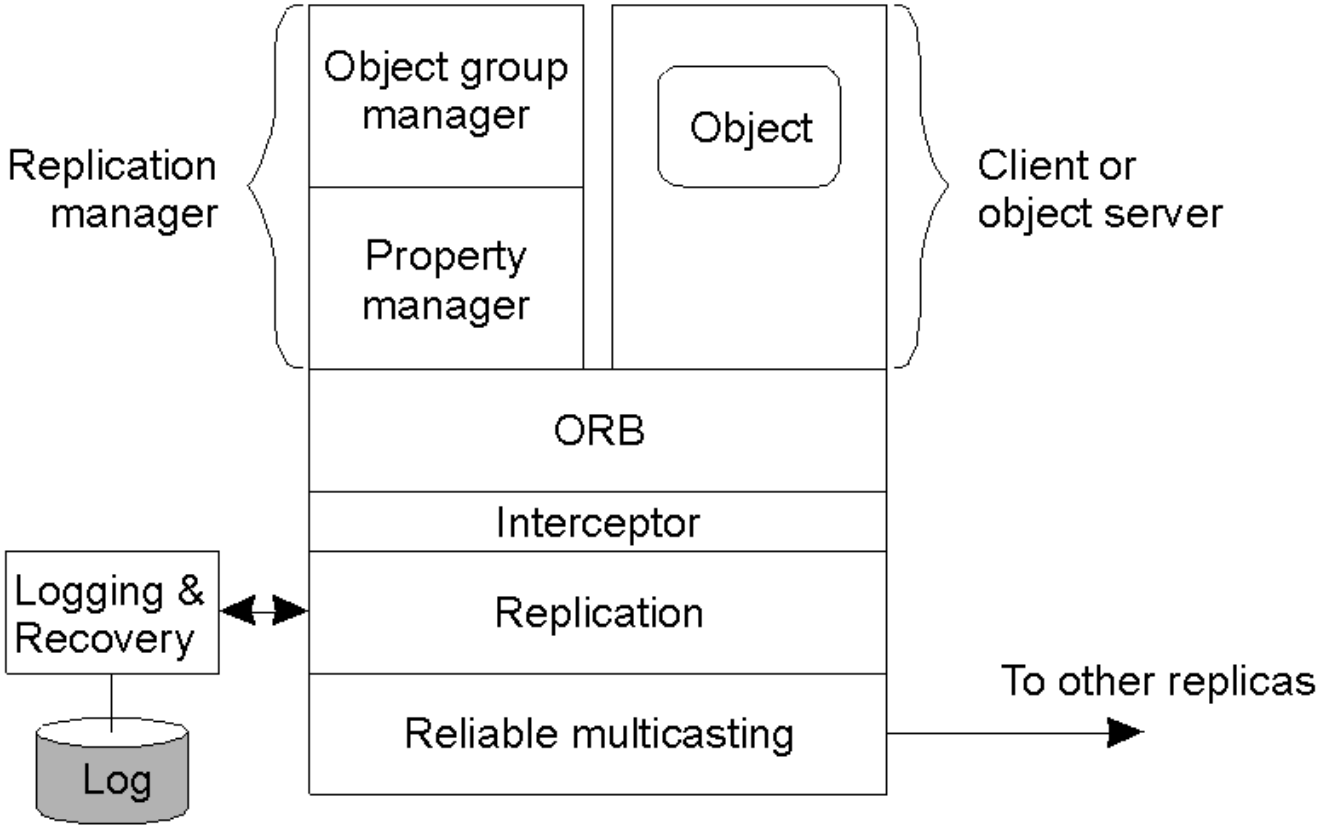
The (simplified) organization of a DCS.

Object Groups



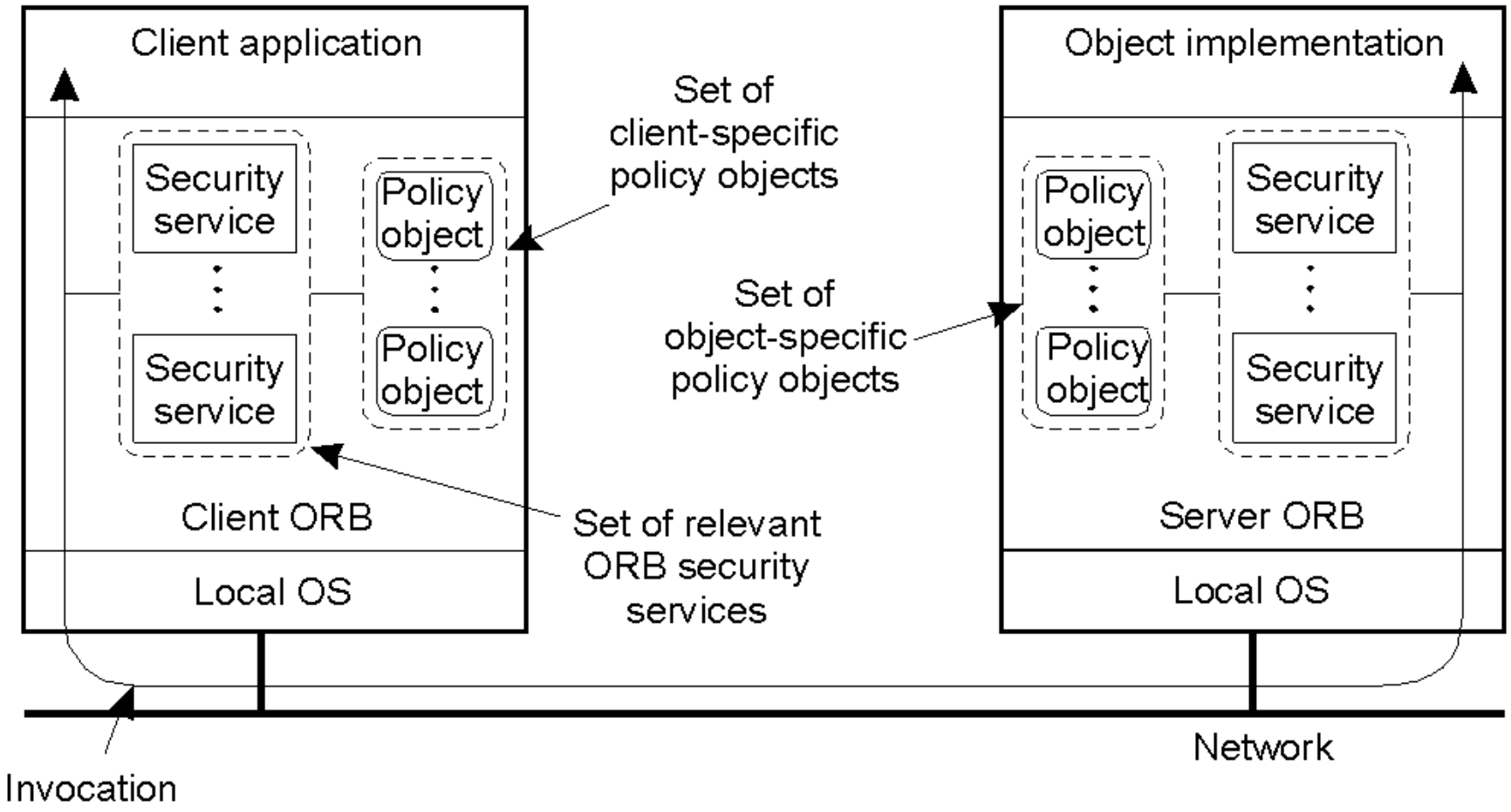
A possible organization of an IOGR for an object group having a primary and backups.

An Example Architecture



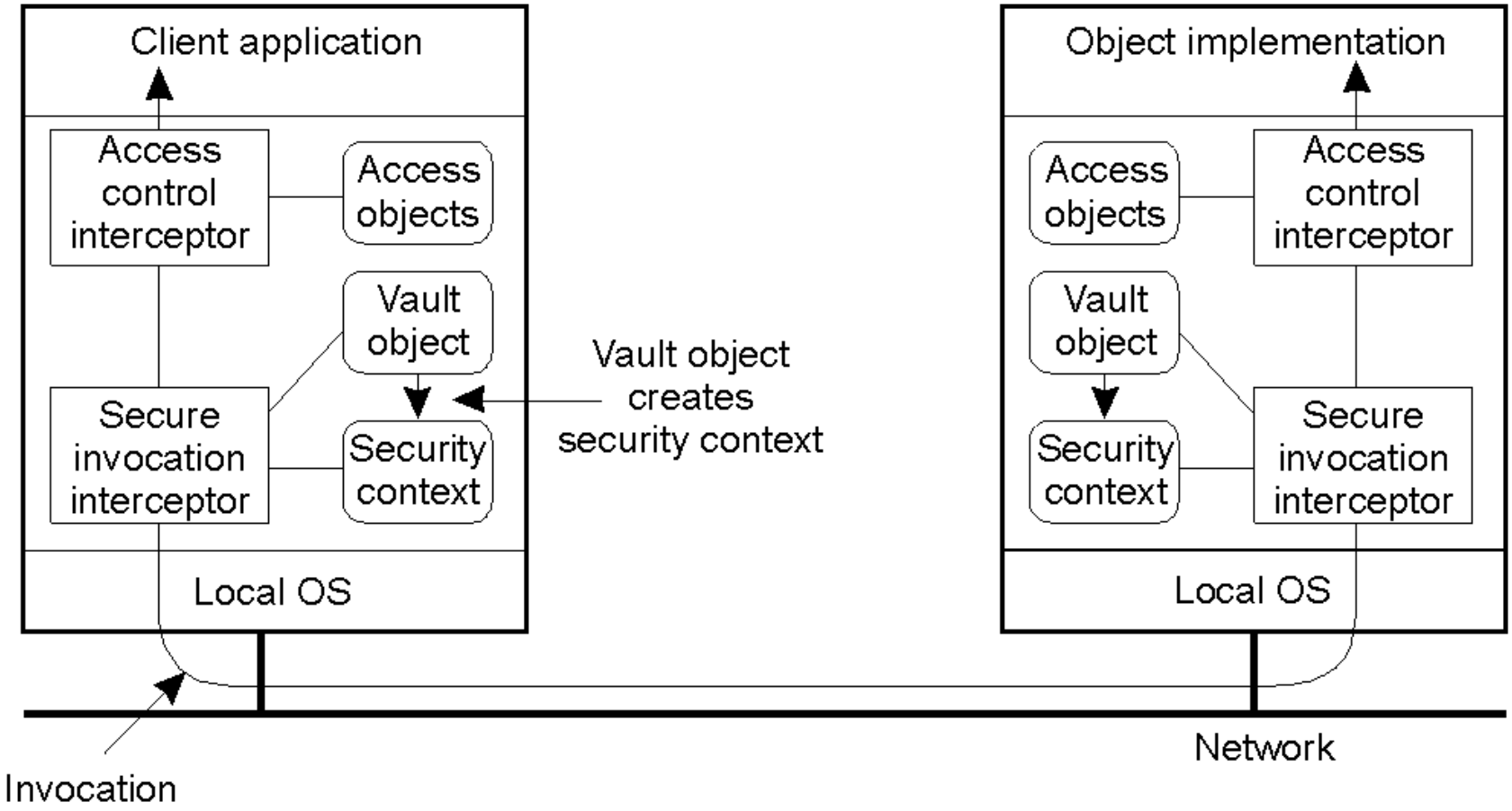
An example architecture of a fault-tolerant CORBA system.

Security (1)



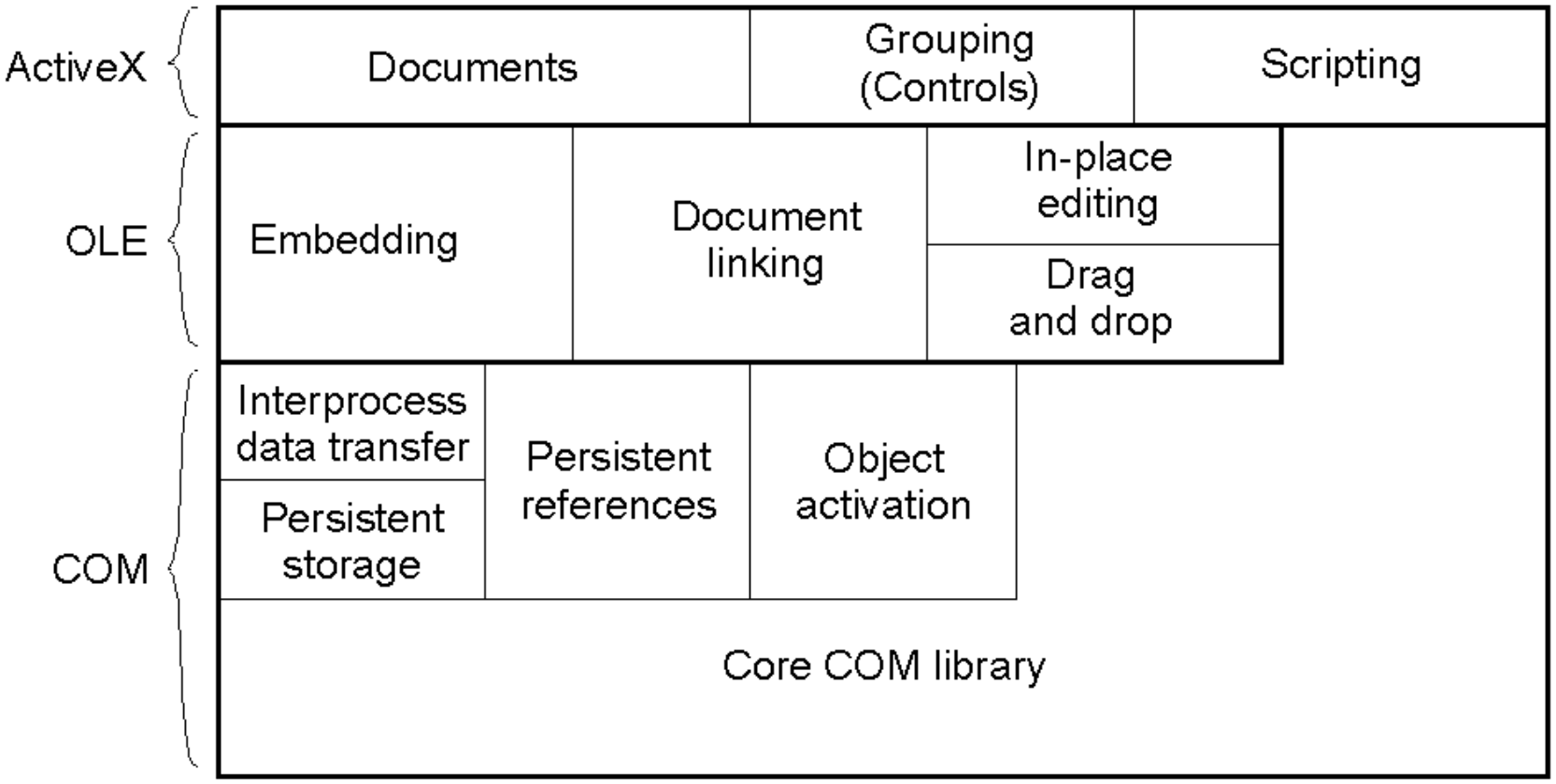
The general organization for secure object invocation in CORBA.

Security (2)



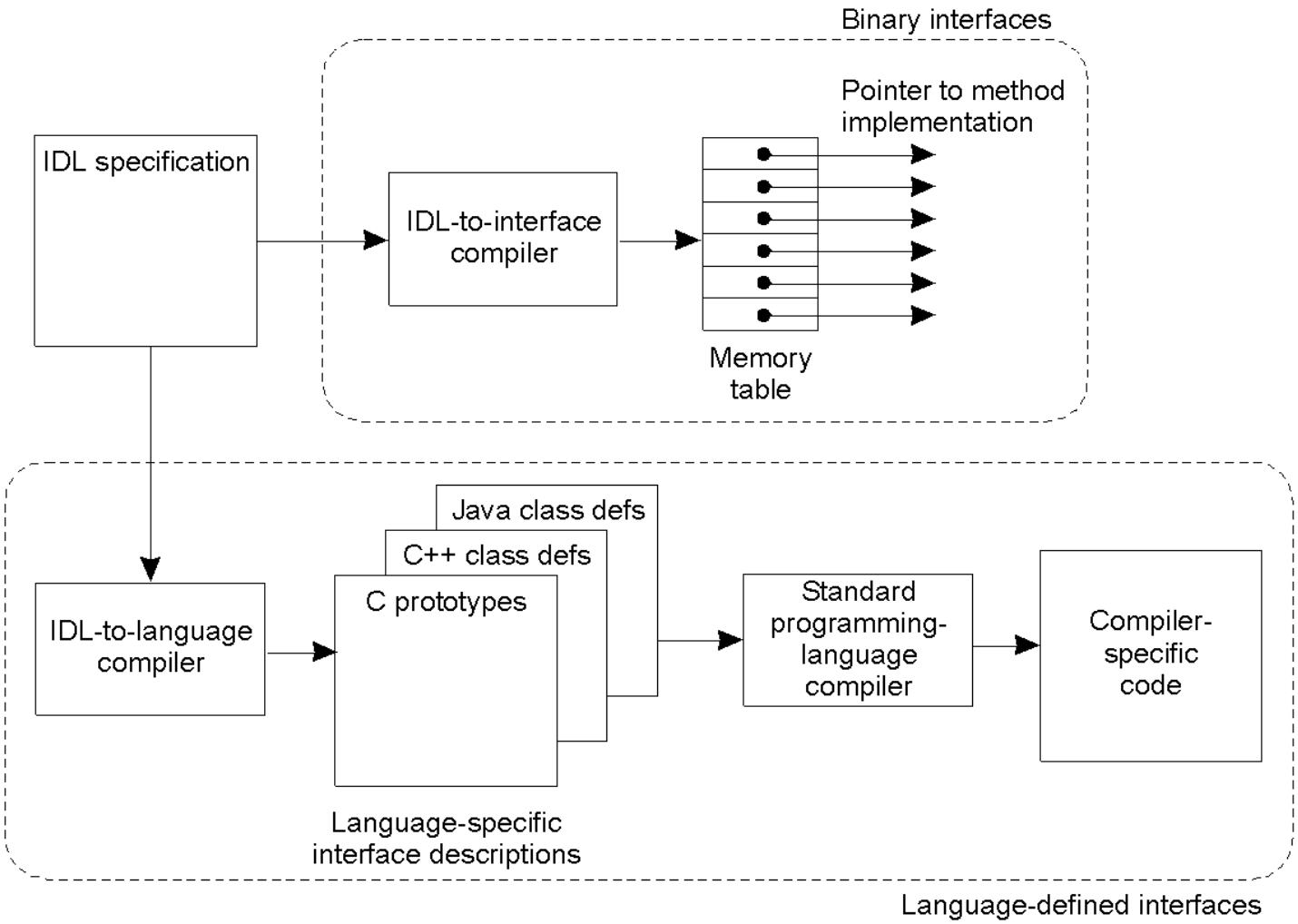
The role of security interceptors in CORBA.

Overview of DCOM



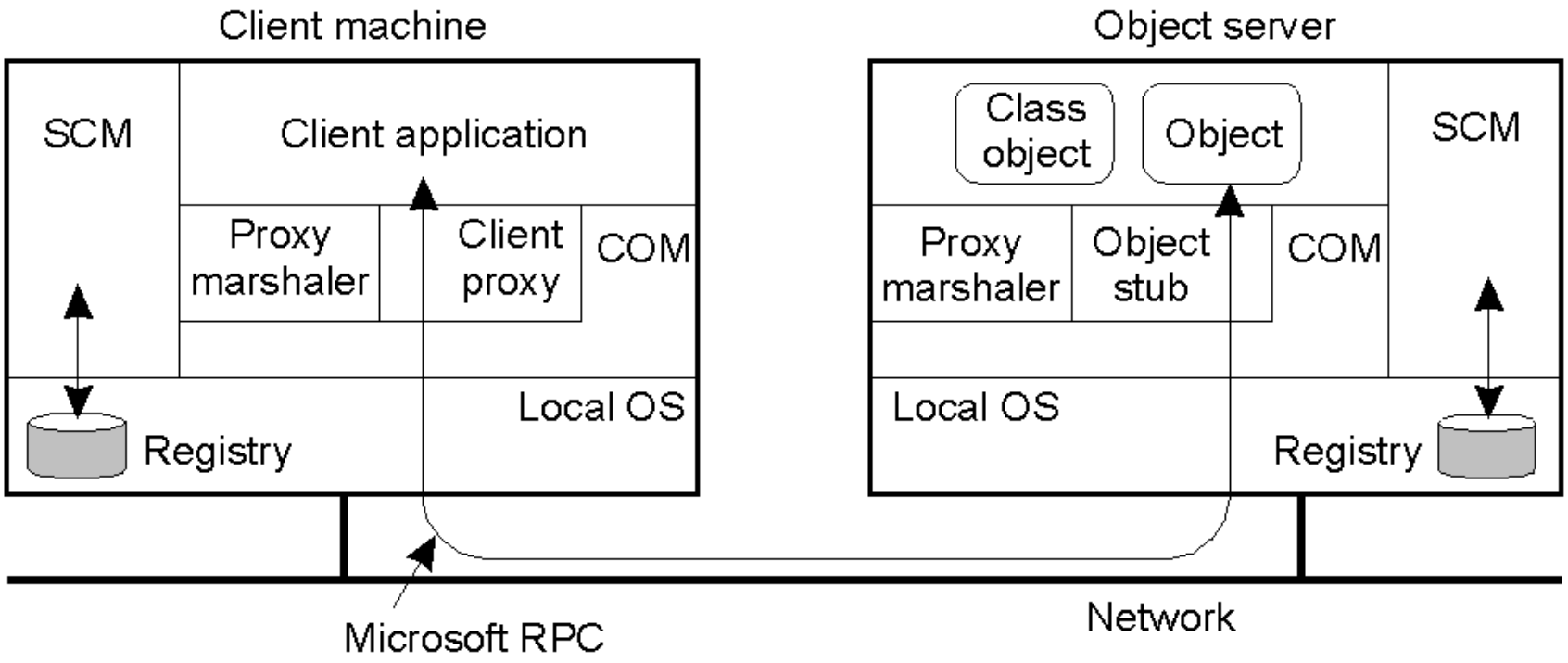
The general organization of ActiveX, OLE, and COM.

Object Model



The difference between language-defined and binary interfaces.

Tape Library and Registry



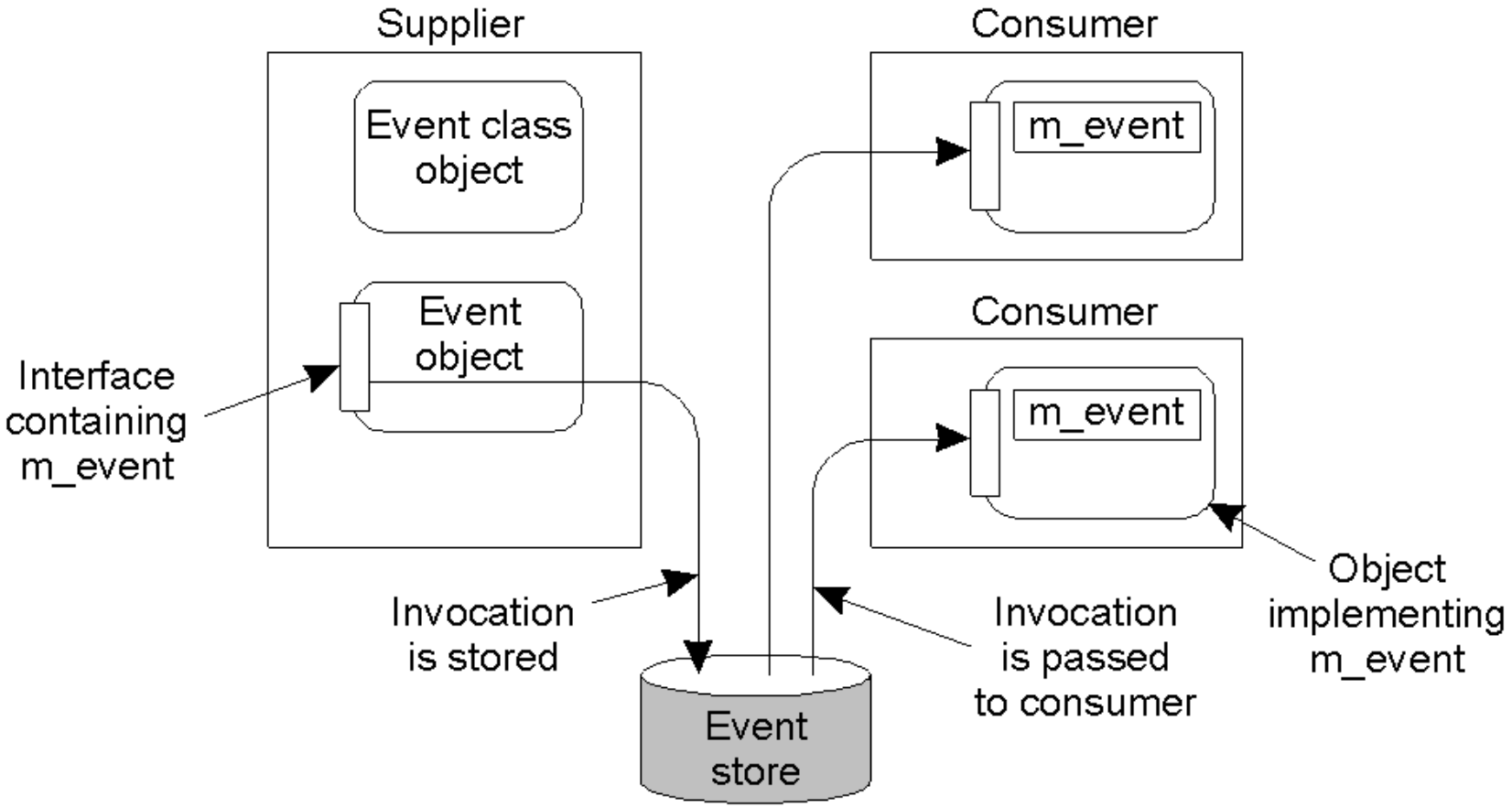
The overall architecture of DCOM.

DCOM Services

CORBA Service	DCOM/COM+ Service	Windows 2000 Service
Collection	ActiveX Data Objects	-
Query	None	-
Concurrency	Thread concurrency	-
Transaction	COM+ Automatic Transactions	Distributed Transaction Coordinator
Event	COM+ Events	-
Notification	COM+ Events	-
Externalization	Marshaling utilities	-
Life cycle	Class factories, JIT activation	-
Licensing	Special class factories	-
Naming	Monikers	Active Directory
Property	None	Active Directory
Trading	None	Active Directory
Persistence	Structured storage	Database access
Relationship	None	Database access
Security	Authorization	SSL, Kerberos
Time	None	None

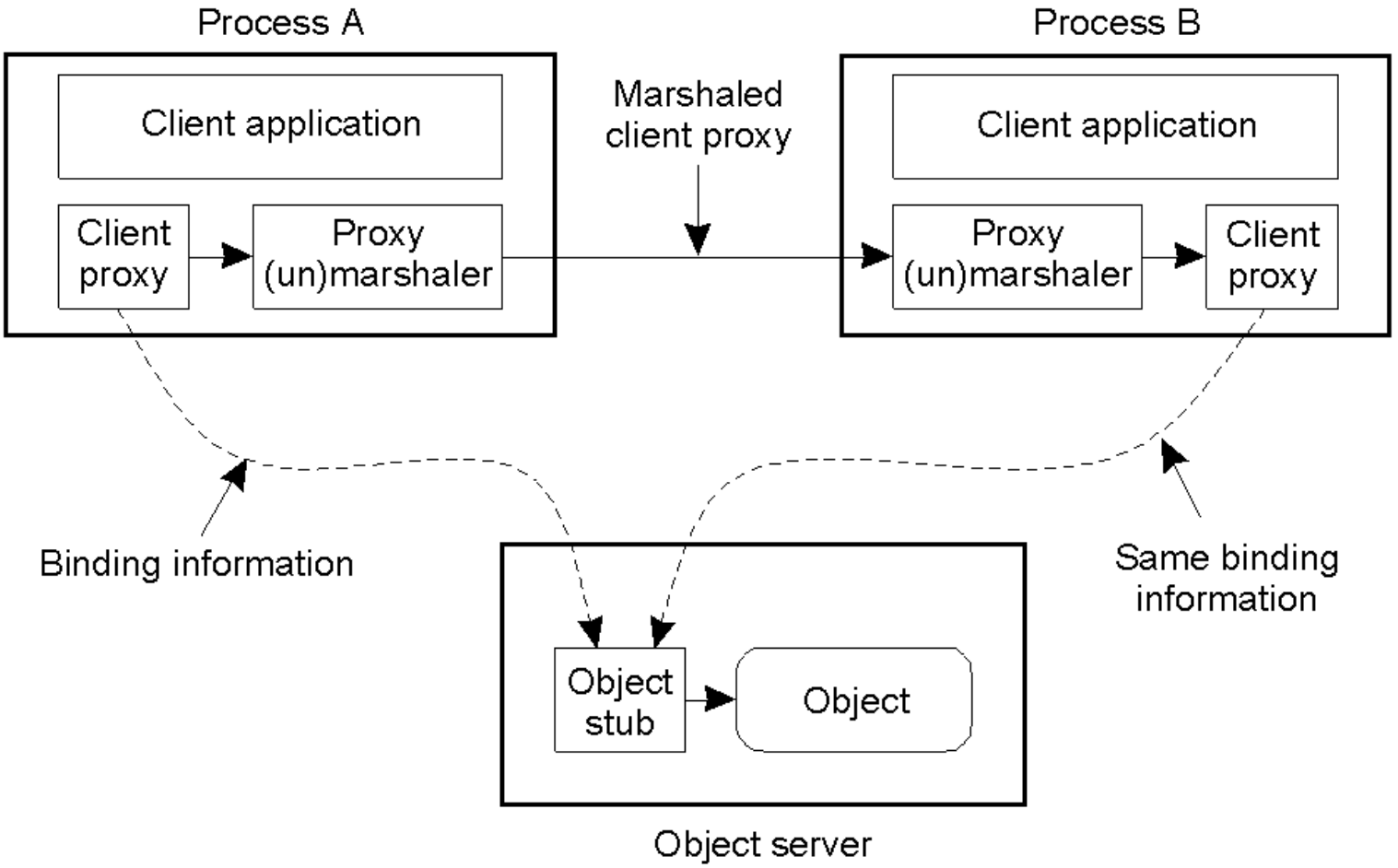
Overview of DCOM services in comparison to CORBA services.

Events



Event processing in DCOM.

Clients



Passing an object reference in DCOM with custom marshaling.

Monikers (1)

Step	Performer	Description
1	Client	Calls BindMoniker at moniker
2	Moniker	Looks up associated CLSID and instructs SCM to create object
3	SCM	Loads class object
4	Class object	Creates object and returns interface pointer to moniker
5	Moniker	Instructs object to load previously stored state
6	Object	Loads its state from file
7	Moniker	Returns interface pointer of object to client

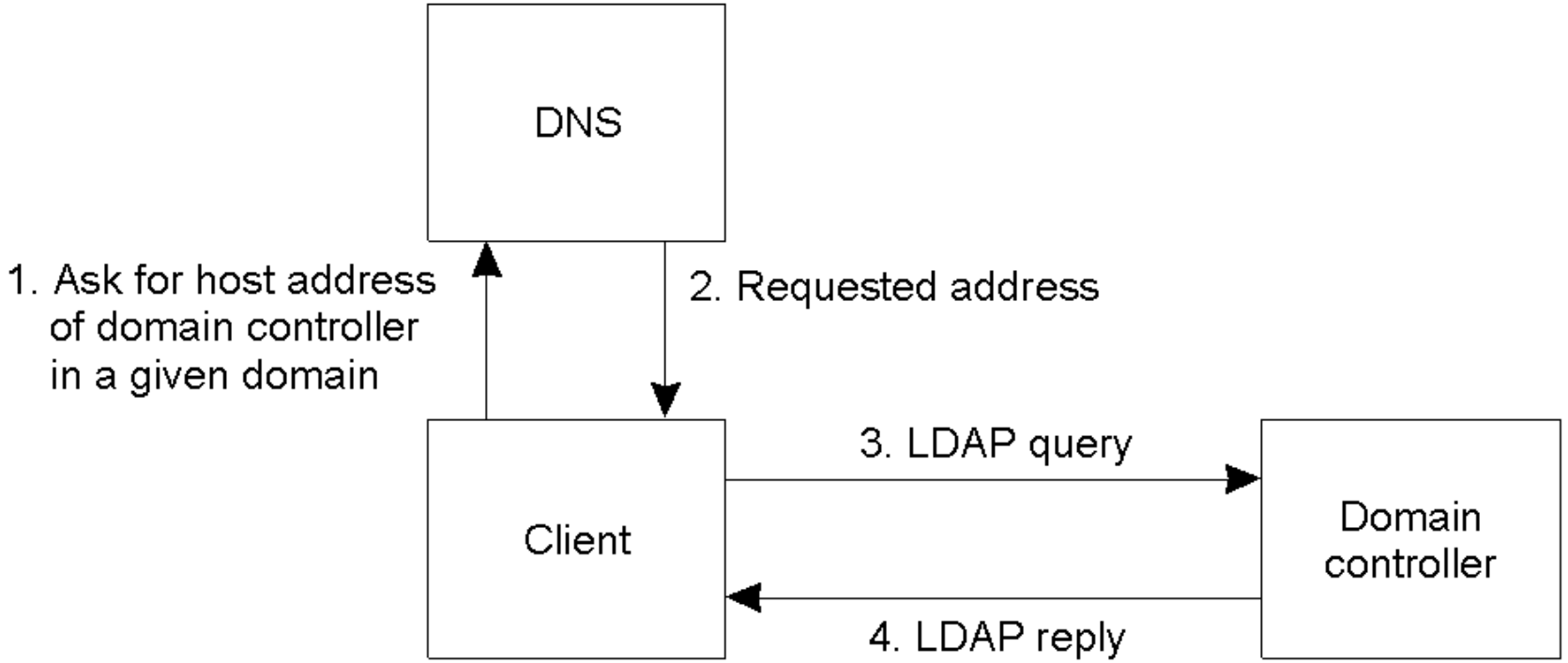
Binding to a DCOM object by means of file moniker.

Monikers (2)

Moniker type	Description
File moniker	Reference to an object constructed from a file
URL moniker	Reference to an object constructed from a URL
Class moniker	Reference to a class object
Composite moniker	Reference to a composition of monikers
Item moniker	Reference to a moniker in a composition
Pointer moniker	Reference to an object in a remote process

DCOM-defined moniker types.

Active Directory



The general organization of Active Directory.

Fault Tolerance

Attribute value	Description
REQUIRES_NEW	A new transaction is always started at each invocation
REQUIRED	A new transaction is started if not already done so
SUPPORTED	Join a transaction only if caller is already part of one
NOT_SUPPORTED	Never join a transaction
DISABLED	Never join a transaction, even if told to do so

Transaction attribute values for DCOM objects.

Declarative Security (1)

Authentication level	Description
NONE	No authentication is required
CONNECT	Authenticate client when first connected to server
CALL	Authenticate client at each invocation
PACKET	Authenticate all data packets
PACKET_INTEGRITY	Authenticate data packets and do integrity check
PACKET_PRIVACY	Authenticate, integrity-check, and encrypt data packets

Authentication levels in DCOM.

Declarative Security (2)

Impersonation level	Description
ANONYMOUS	The client is completely anonymous to the server
IDENTIFY	The server knows the client and can do access control checks
IMPERSONATE	The server can invoke local objects on behalf of the client
DELEGATE	The server can invoke remote objects on behalf of the client

Impersonation levels in DCOM.

Programmatic Security

Service	Description
NONE	No authentication
DCE_PRIVATE	DCE authentication based on shared keys
DCE_PUBLIC	DEC authentication based on public keys
WINNT	Windows NT security
GSS_KERBEROS	Kerberos authentication

(a)

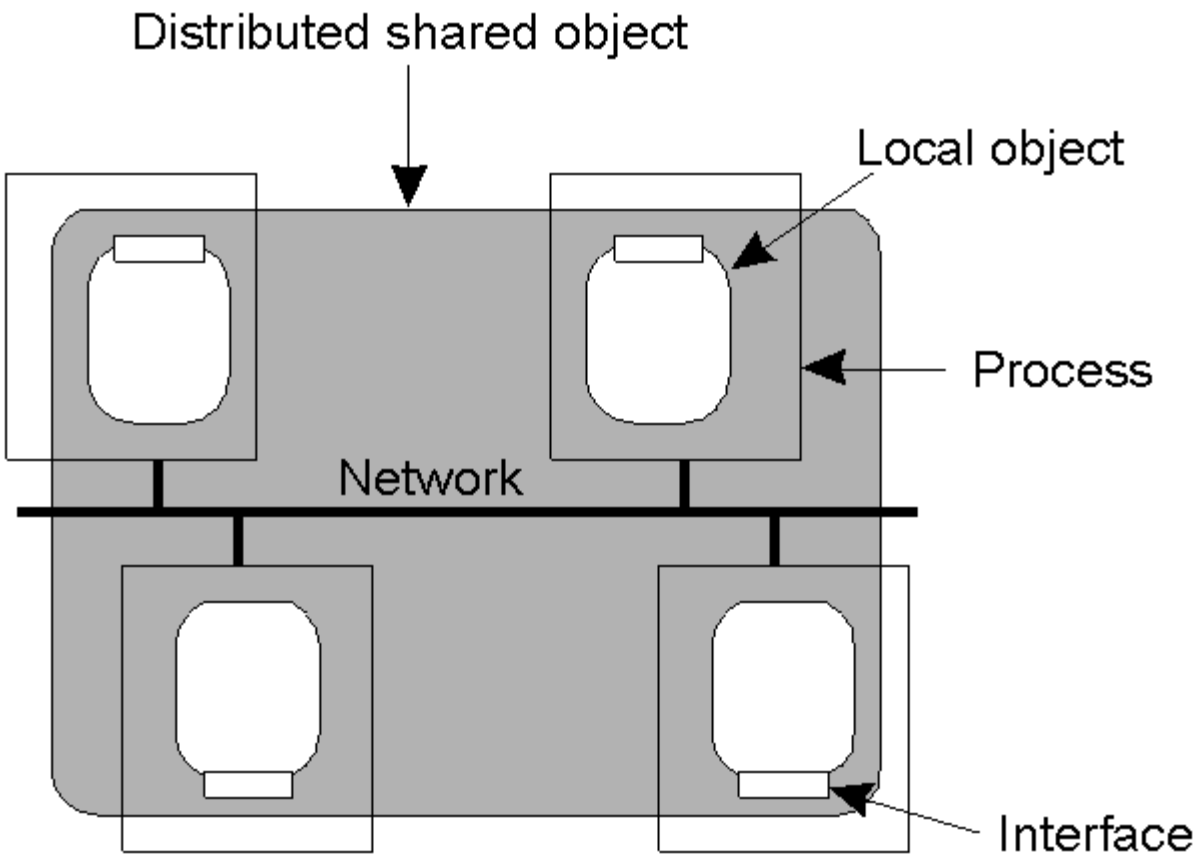
Service	Description
NONE	No authorization
NAME	Authorization based on the client's identity
DCE	Authorization using DEC Privilege Attribute Certificates (PACs)

(b)

a) Default authentication services supported in DCOM.

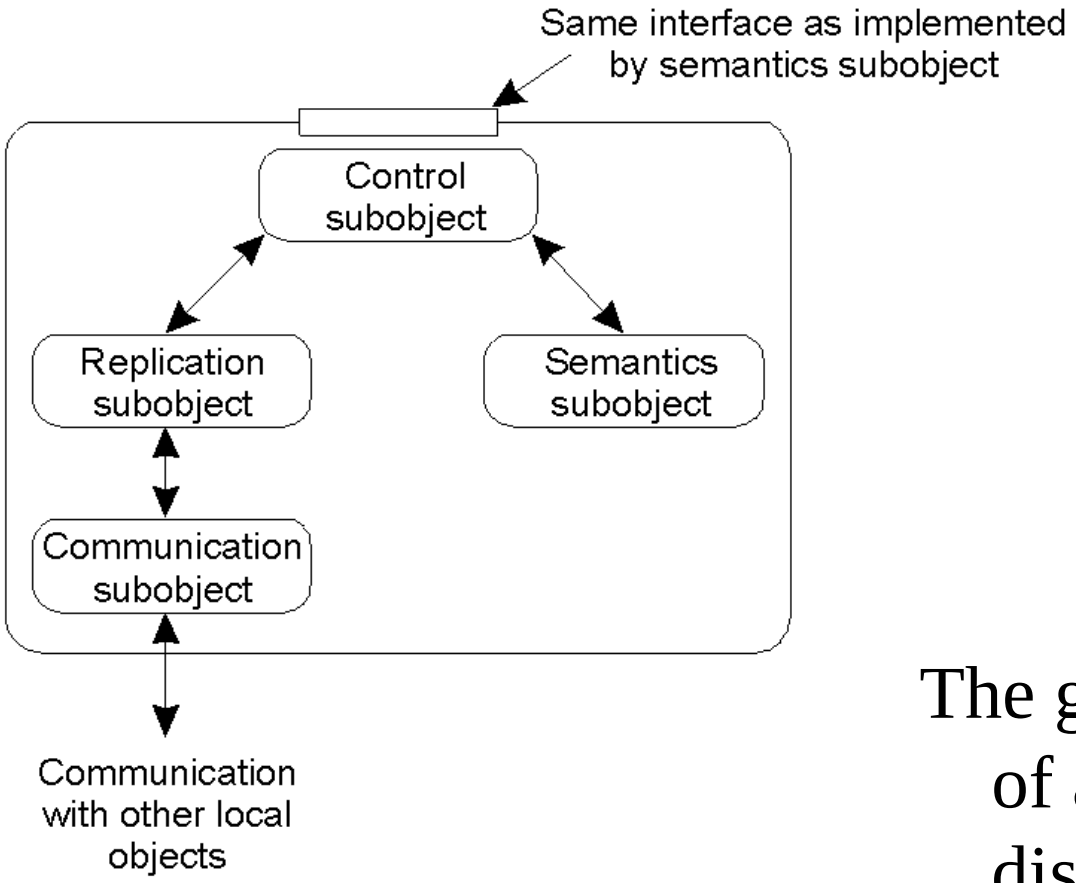
b) Default authorization services supported in DCOM.

Globe Object Model (1)



The organization of a Globe distributed shared object.

Globe Object Model (2)



The general organization of a local object for distributed shared objects in Globe.

Globe Object Model (3)

Document Interface

Method	Description
AddElement	Add an element to the current set of elements
DeleteElement	Remove an element from the Web document
AllElements	Return a list of the elements currently in the document
SetRoot	Set the root element
GetRoot	Return a reference to the root element

Content Interface

Method	Description
GetCotent	Return the content of an element as an array of bytes
PutContent	Replace the content of an element with a given array of bytes
PutAllContent	Replace the content of an entire document

Interfaces implemented by the semantics subobject of a GlobeDoc object.

Globe Object Model (4)

Property Interface

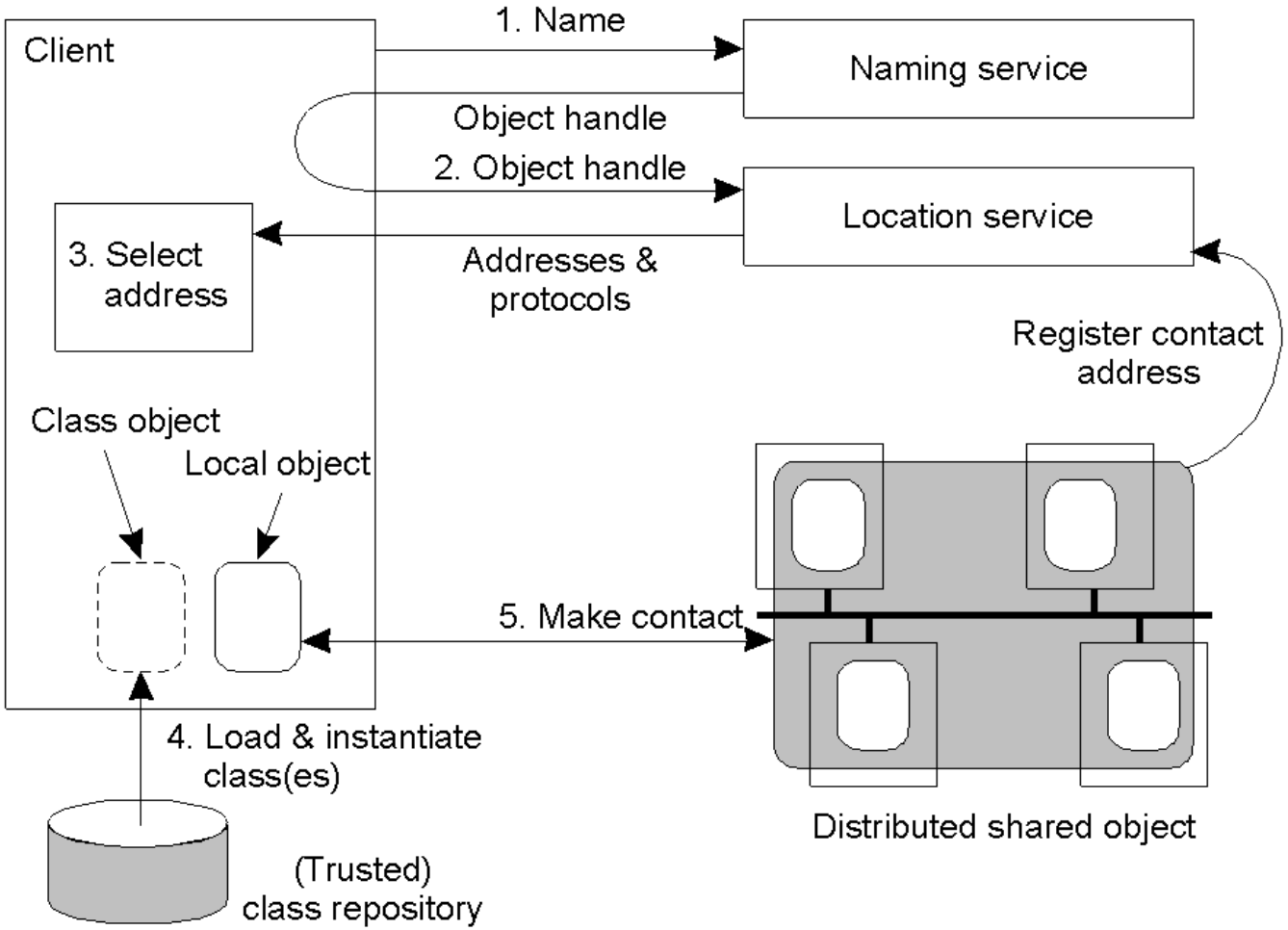
Method	Description
GetProperties	Return the list of (attribute, value)-pairs of an element
SetProperties	Provide a list of (attribute, value)-pairs for an element

Lock Interface

Method	Description
CheckOutElements	Check out a series of elements that require modification
CheckInElements	Check in a series of modified elements
GetCheckedElements	Get a list of elements that are currently checked out

Interfaces implemented by the semantics subobject of a GlobeDoc Object.

Process-to-Object Binding



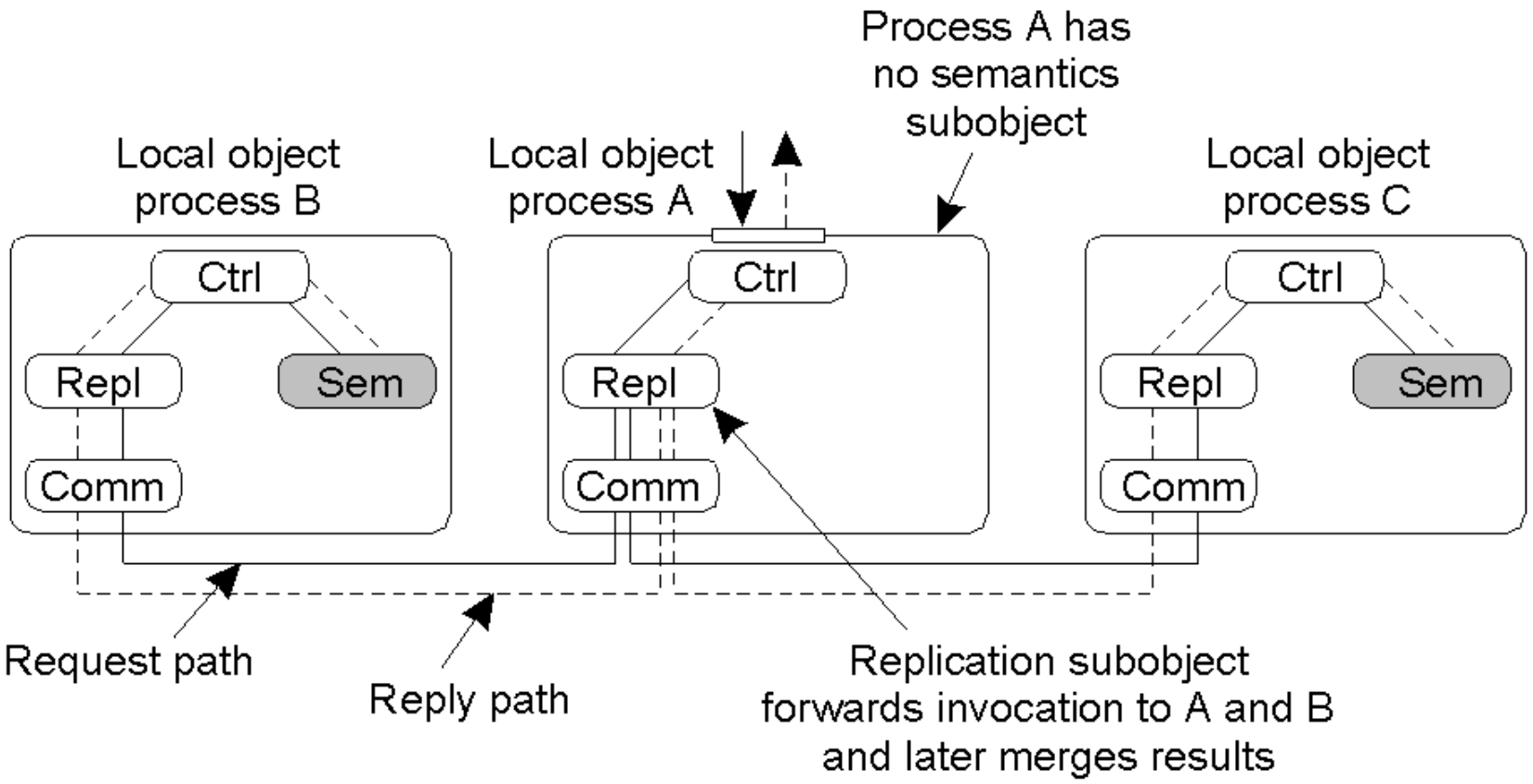
Binding a process to an object in Globe.

Globe Services

Service	Possible Implementation in Globe	Available
Collection	Separate object that holds references to other objects	No
Concurrency	Each object implements its own concurrency control strategy	No
Transaction	Separate object representing a transaction manager	No
Event/Notification	Separate object per group of events (as in DCOM)	No
Externalization	Each object implements its own marshaling routines	Yes
Life cycle	Separate class objects combined with per-object implementations	Yes
Licensing	Implemented by each object separately	No
Naming	Separate service, implemented by a collection of naming objects	Yes
Property/Trading	Separate service, implemented by a collection of directory objects	No
Persistence	Implemented on a per-object basis	Yes
Security	Implemented per object, combined with (local) security services	Yes
Replication	Implemented on a per-object basis	Yes
Fault tolerance	Implemented per object combined with fault-tolerant services	Yes

Overview of possible Globe implementations of typical distributes-systems services.

Communication



Invoking an object in Globe that uses active replication.

Globe Server

Method	Description
Bind	Lets the server bind to a given object, unless it is already bound
AddBinding	Lets the server bind to an object, even if it is already bound
CreateLR	Lets the server create a local object for a new distributed object
RemoveLR	Lets the server remove a local object of a given object
UnbindDSO	Lets the server remove all local objects of a given object
ListAll	Returns a list of all local objects
ListDSO	Returns a list of all local objects for a given objects
StatLR	Get the status of a specific local object

Operations on a Globe object server.

Object References and Contact Addresses (1)

Field	Description
Protocol identifier	A constant representing a (known) protocol
Protocol address	A protocol-specific address
Implementation handle	Reference to a file in a class repository

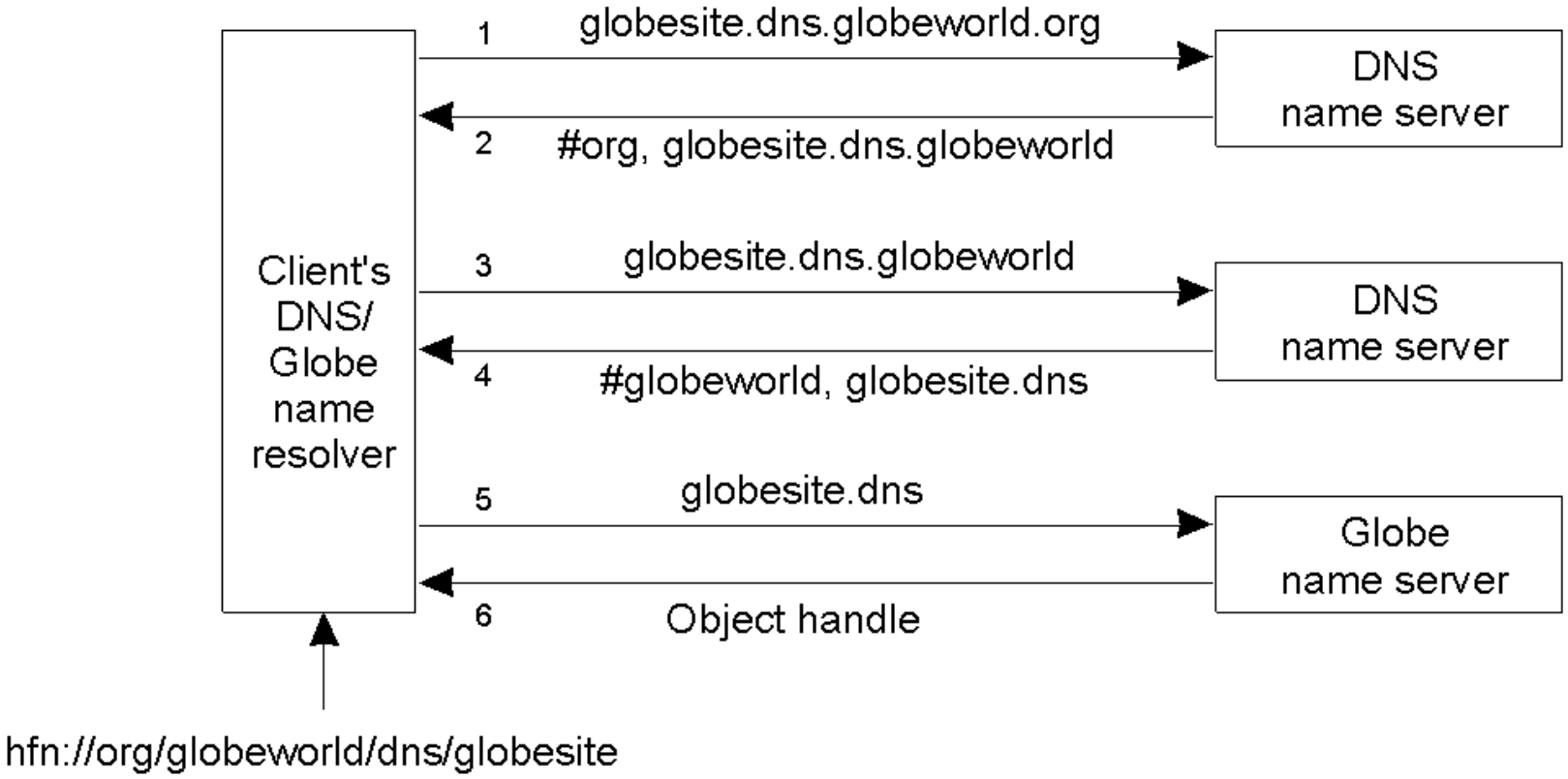
The representation of a protocol layer in a stacked contact address.

Object References and Contact Addresses (2)

Field	Description
Implementation handle	Reference to a file in a class repository
Initialization string	String that is used to initialize an implementation

The representation of an instance contact address.

Globe Naming Service



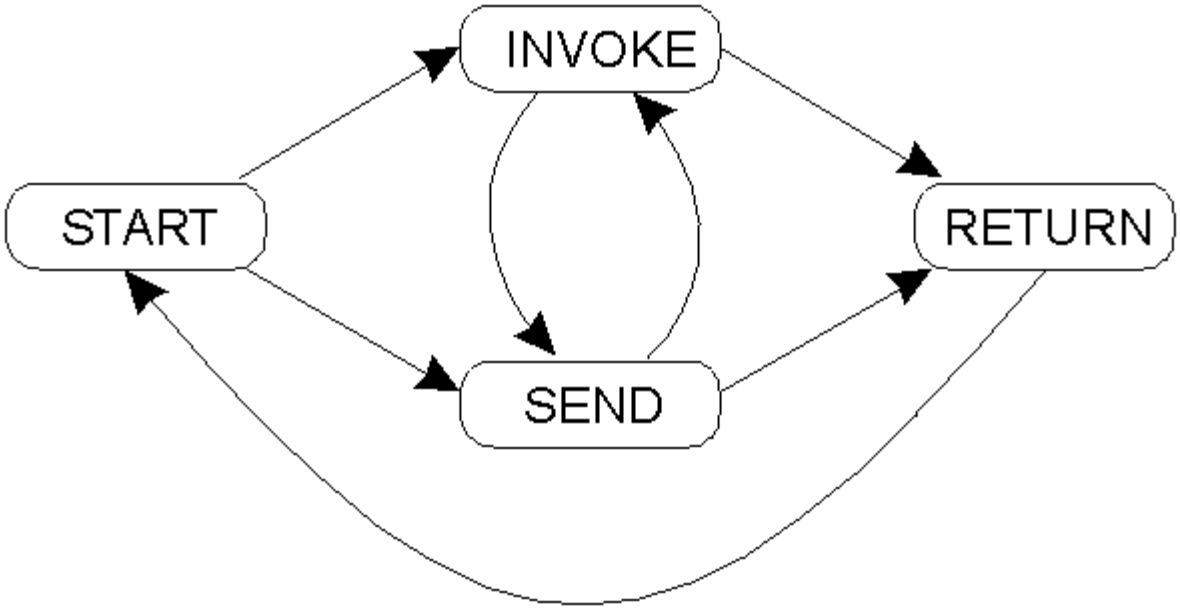
Iterative DNS-based name resolution in Globe.

Replication (1)

Method	Description
Start	Indicate that a new method invocation has been locally requested
Send	Pass the marshaled invocation request to the replication subobject
Invoked	Indicate that the invocation on the semantics object has completed

The interface of the replication subobject as made available to the control subobject.

Replication (2)



The behavior of the control subobject as a finite state machine.

Examples of Replication in Globe (1)

Read method

State	Action to take	Method call	Next state
START	None	Start	INVOKE
INVOKE	Invoke local method	Invoked	RETURN
RETURN	Return results to caller	None	START

Modify method

State	Action to take	Method call	Next state
START	None	Start	SEND
SEND	Pass marshaled invocations	Send	INVOKE
INVOKE	invoke local method	Invoked	RETURN
RETURN	Return results to caller	None	START

State transitions and actions for active replication.

Examples of Replication in Globe (2)

Read method

State	Action to take	Method call	Next state
START	None	Start	INVOKE
INVOKE	Invoke local method	Invoked	RETURN
RETURN	Return results to caller	None	START

Modify method at backup replica

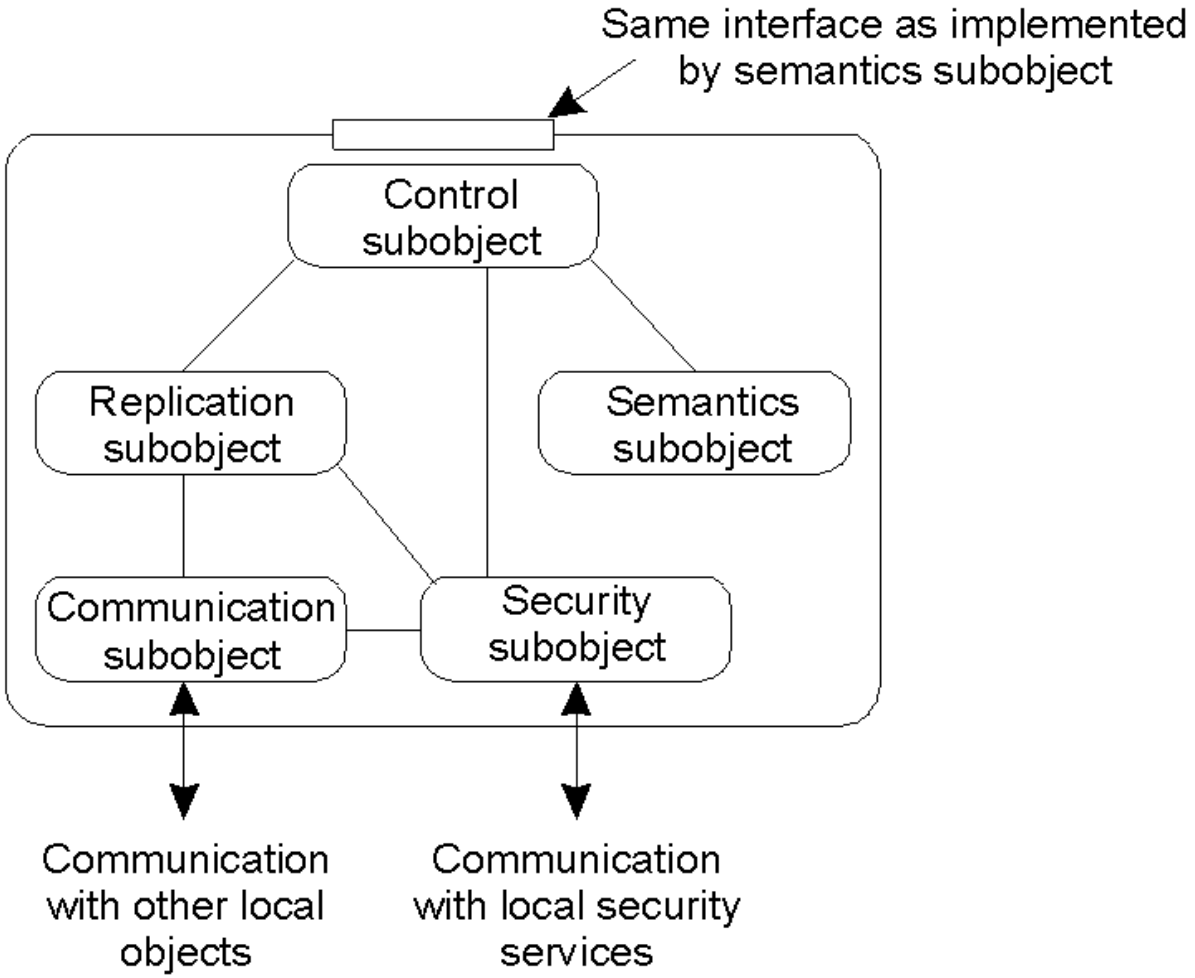
State	Action to take	Method call	Next state
START	None	Start	SEND
SEND	Pass marshaled invocation	Send	RETURN
RETURN	Return results to caller	None	START

Modify method at primary replica

State	Action to take	Method call	Next state
START	none	Start	INVOKE
INVOKE	invoke local method	Invoked	RETURN
RETURN	Return results to caller	None	START

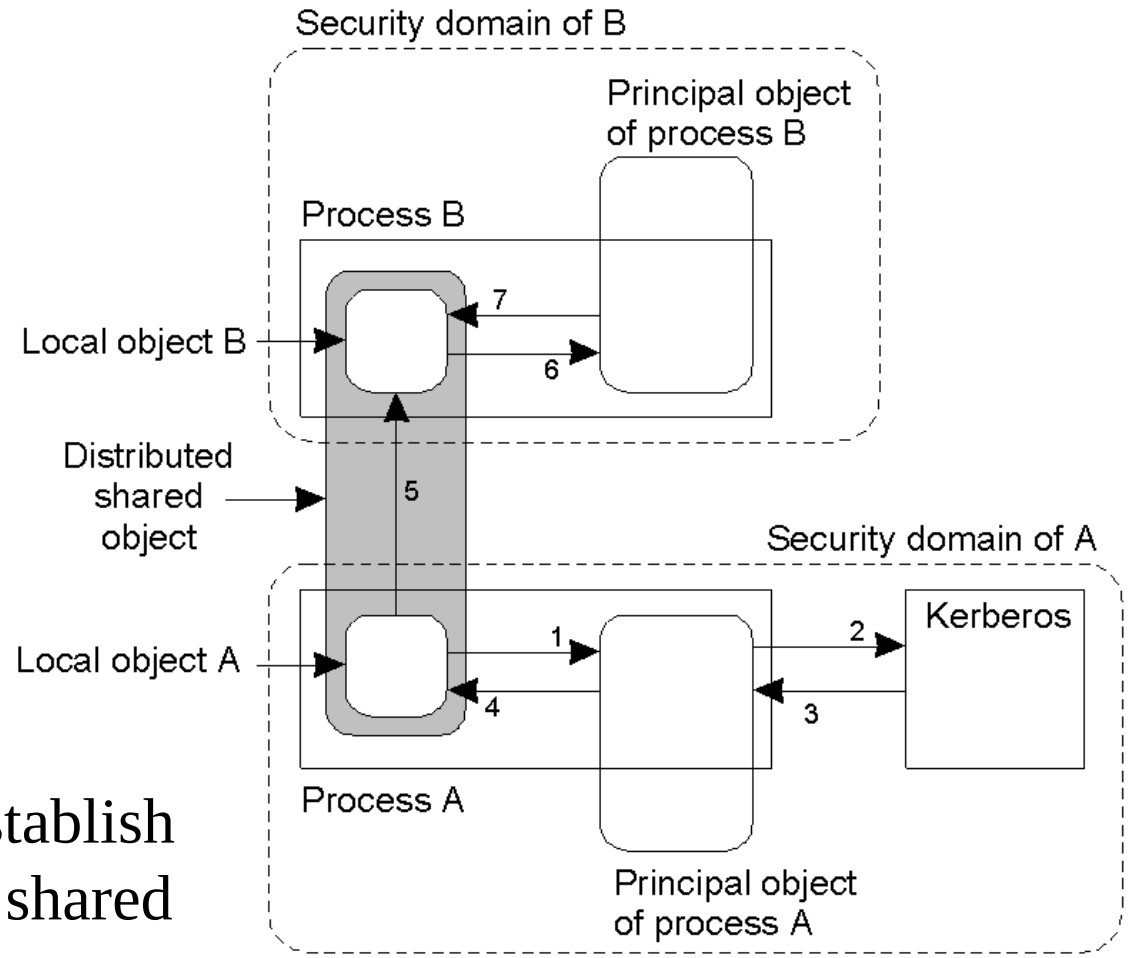
State transitions and actions with primary-backup replication.

Security (1)



The position of a security subobject in a Globe local object.

Security (2)



Using Kerberos to establish secure distributed shared objects.

Summary (1)

Issue	CORBA	DCOM	Globe
Design goals	Interoperability	Functionality	Scalability
Object model	Remote objects	Remote objects	Distributed objects
Services	Many of its own	From environment	Few
Interfaces	IDL based	Binary	Binary
Sync. communication	Yes	Yes	Yes
Async. communication	Yes	Yes	No
Callbacks	Yes	Yes	No
Events	Yes	Yes	No
Messaging	Yes	Yes	No
Object server	Flexible (POA)	Hard-coded	Object dependent
Directory service	Yes	Yes	No
Trading service	yes	No	No

Continued …

Comparison of CORBA, DCOM, and Globe.

Summary (2)

Issue	CORBA	DCOM	Globe
Naming service	Yes	Yes	Yes
Location service	No	No	Yes
Object reference	Object's location	Interface pointer	True identifier
Synchronization	Transactions	Transactions	Only intra-object
Replication support	Separate server	None	Separate subobject
Transactions	Yes	Yes	No
Fault tolerance	By replication	By transactions	By replication
Recovery support	Yes	By transactions	No
Security	Various mechanisms	Various mechanisms	More work needed

Comparison of CORBA, DCOM, and Globe.