CORBA

1999
A CORBA-based Secure Session Service
for Distributed Multimedia
Collaborative Environments
A CORBA-based Secure Session Service for Distributed Multimedia Collaborative Environments

by

Young-Mi Shin

Department of Computer Science and Engineering

Pohang University of Science and Technology

A thesis submitted to the Pohang University of Science and Technology in partial fulfillment of the requirement for the degree of Master of Science in the Dept. Computer Science and Engineering.

Pohang, Korea
December 15, 1998
Approved by

____________________

Major Advisor
CORBA

1998年12月15日

( )

( )

( )
ABSTRACT

Distributed multimedia applications are being developed and used for many fields of our lives today. There is a vast amount of single-purpose application software available, such as video audio conferencing, chatting, whiteboard, electronic notebook, and so on. The goal of distributed multimedia application is to exchange multimedia information among users. With these distributed multimedia applications, we can realize a collaborative environment that allows a group of users to collaborate from their desktops as effectively as if they were face-to-face.

In order to provide an effective collaborative environment, it needs to be secure and platform-independent. Moreover, the collaborative environment has to support facilities for creating, eliminating, joining, and leaving collaboration sessions. The service that serves these facilities is called a session service. OMG CORBA allows applications to communicate with one another no matter where they are located or who has developed them. So we use CORBA to realize our session service.

This thesis proposes a CORBA-based secure session service for multimedia collaborative environments. This secure session service is platform-independent, enhances the security and offers a session information generically. As a proof of concept, a secure session service has been developed as part of a CORBA-based distributed multimedia system called MAESTRO. Our secure session service can be proposed as a new Common Object Service in the OMG standard.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>2.1.1</td>
<td>MBONE</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2</td>
<td>NetMeeting</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>2.5</td>
<td>CORBA Security Service</td>
<td>11</td>
</tr>
<tr>
<td>2.5.1</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>2.5.2</td>
<td>CORBA</td>
<td>12</td>
</tr>
<tr>
<td>2.5.3</td>
<td>CORBA</td>
<td>12</td>
</tr>
<tr>
<td>2.5.4</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>2.5.5</td>
<td>CORBA</td>
<td>14</td>
</tr>
<tr>
<td>2.6</td>
<td>MBONE</td>
<td>15</td>
</tr>
<tr>
<td>2.6.1</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>2.6.2</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>2.6.3</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>No.</td>
<td>Section</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MBONE (topology) .................................................. 5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MBONE (Sdr) ........................................................... 5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ILS ........................................................................ 7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SESAME .................................................................. 10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ORB ....................................................................... 14</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>........................................................................ 16</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>........................................................................ 23</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>........................................................................ 26</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>........................................................................ 27</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ORB (Sdr) ................................................................ 33</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>MAESTRO ................................................................ 38</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>........................................................................ 41</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>........................................................................ 42</td>
<td></td>
</tr>
</tbody>
</table>
1. (collaborative environment) [1, 2, 3].

2. (collaborative environment) [4, 5, 6].
Common Object Request Broker Architecture (CORBA) [9].
CORBA  Common Object Services Specification (COSS) [10]  

2.1.3  2.1.4  2.1.5  2.1.6  5.1  5.2  5.3  5.4  5.5  5.6  6.1  6.2  6.3  6.4  6.5  6.6  6.7  7.1  7.2  7.3  7.4  7.5  7.6  7.7
2  

\[ \text{CORBA} \]

\[ \text{Common Object Services Specification (COSS) [10]} \]

2.1  

\[ \text{MBONE} \]

\[ \text{UNIX } \]

\[ \text{MBONE [7]} \]

\[ \text{Windows 95 , NT } \]

\[ \text{NetMeeting [8]} \]

2.1.1  

\[ \text{MBONE} \]

\[ \text{unicast IP} \]

\[ \text{MBONE} \]

\[ \text{MBONE} \]
1: MBONE topology

MBONE topology (session directory), audio reference tool, video conference tool, whiteboard tool (Sdr) [7].

2: MBONE (Sdr)
MBONE and the Internet Session Announcement Protocol (SAP) [15], Session Description Protocol (SDP) [16], Session Initiation Protocol (SIP) [17].

SAP and SDP are used in the MBONE to announce sessions, whereas SIP is used for controlling these sessions in the Internet. SDP and SIP are also used on the local machine to establish sessions on the NetMeeting [8].

2.1.2 NetMeeting

NetMeeting [8] is used in Windows 95 and NT to establish sessions.
NetMeeting supports several features such as chat system, whiteboard, and audio/video conferencing.

Internet Locator Service (ILS) [18] is used to resolve IP addresses to Internet domain names. ILS is responsible for mapping IP addresses to their corresponding domain names.

NetMeeting's dynamic directory client uses the LDAP protocol for authentication and communication. This allows NetMeeting to connect to various directory servers such as Microsoft's Active Directory or others.

3: ILS
3. ILS Lightweight Directory Access Protocol (LDAP) [19]

ILS Lightweight Directory Access Protocol (LDAP) [19]

2.2 Confidentiality

ILS Confidentiality (confidentiality): 

Integrity

ILS Integrity (integrity): 

Availability

ILS Availability (availability): 

2.3 (user identification and authentication)

ID:  

2.4 Multi-domain - SESAME

Secure European System For Application in a Multivendor Environment (SESAME) [20, 21]
European Computer Manufacturers Association (ECMA)
2.5 CORBA Security Service

CORBA [9] is a set of frameworks, languages, and guidelines for distributed objects. CORBA consists of several specifications. The Common Object Services Specification (COSS) [10] defines the Common Object Services. CORBA Security Service (COSS) is an extension of CORBA that provides security services for distributed objects.

2.5.1 Interface Description

The CORBA Security Service interface is described in detail in the COSS specification. The interface provides methods for authentication, authorization, and confidentiality. The interface is defined in terms of a CORBA interface, which is a type of component that can be used to represent a CORBA object.

The CORBA Security Service interface allows clients to request security services, such as authentication and authorization, from a CORBA object. The interface also provides methods for securing messages, such as encryption and digital signatures, and for managing security policies.

The interface is defined in the CORBA Object Request Broker (ORB) and is used by CORBA clients and servers to provide security services. The interface is designed to be flexible and can be used to support a wide range of security needs.
2.5.2 CORBA Security

CORBA security is provided by the CORBA specification [10].

- **Information compromise** (information compromise)
- **Integrity violations** (integrity violations)
- **Denial of service** (denial of service)

2.5.3 CORBA Security

CORBA security is provided by the CORBA specification [10].

- **Identification** (identification)
- **Authentication** (authentication)
- **Authorization** (authorization)
- **Access control** (access control)
- **Security auditing** (security auditing)

(translation continues)
(security of communication)

CORBA (security of communication)

(integrity) (confidentiality)

(non-repudiation)

(administration)

2.5.4 (security reference model)

CORBA (security reference model)

(meta-policy)

Security of Communication

Object Request Broker (ORB)

Secure Invocation

Vault
Security context

2.5.5 CORBA ハードウェア

1.  
2.  

2.5.5 CORBA

5: ORB ハードウェア
2.6 MBONE  Session Announcement Protocol (SAP), Session Initiation Protocol (SIP)

2.6.1 MBONE (stream)
2.6.2 MBONE

MBONE is a network of networks that connects videoconference systems around the world. MBONE is a Session Announcement Protocol (SAP), Session Initiation Protocol (SIP) (format) and Secure Conferencing User Agent (SCUA) [24].

2.6.3 MICE

MICE [25] is a set of protocols that are used to set up and manage sessions. ICE-TEL [26] is a protocol for setting up telecommunications sessions.

Pretty Good
Privacy (PGP) [27], Secure/Multipurpose Internet Mail Extensions (S/MIME) [28], MIME Object Security Services (MOSS) [29]
3  """

3.1 """
3.2 ・・・ ・・・ ・・・ ・・・ ・・・ ・・・ ・・・ ・・・ ・・・

(Confidentiality)

(Authentication)

(Access Control)
**不正（integrity）**

この文書は、不正行為を防ぐための措置を検討するものである。不正行為の防止策として、不正行為を検知し、不正行為を検知した場合には対応する必要がある。

この文書は、不正行為を防ぐための措置を検討するものである。不正行為の防止策として、不正行為を検知し、不正行為を検知した場合には対応する必要がある。
4 CORBA

...
Unified Modeling Language (UML) [30] 
Rational Rose [31].

4.1 

...
7. これでこれです

App: Multimedia Application
SSDB: Session Sever DB

4.2 これでこれ

4.2.1 これでこれ
構文

```
struct RegisterInfo {
    int port_num;
    char* domain;
    char *name;
}
```

4.2.2 コンポーネントの登録・削除

コンポーネントの登録は、Session Service に情報を提供することを想定しています。以下に簡単な例を示します。

```c
struct RegisterInfo {
    int port_num;
    char* domain;
    char *name;
}
CORBA:: Any a_info;   CORBA:: Any a_info;
RegisterInfo info;   RegisterInfo info
instantiate info;   get a_info from Session Service
a_info <<= info;   *a_info >>= info;
register a_info into Session Service use a_info
```

(DECL. Page 237)
::: Session 2

Participant 2

Application 2

::: Session 1

Participant 1

Application 1

::: Registry 1

::: Registry 2

::: CORBA

::: Any 

::: SSO

::: Participant

::: Application

::: Registry

::: CORBA:: Any

::: Session

::: Participant
### Registry

<table>
<thead>
<tr>
<th>Participant</th>
<th>Application</th>
<th>Session</th>
<th>SSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

#### Participant
- **name**: char*
- **domain**: char*
- **name()**: Participant
- **domain()**: Participant
- **Participant()**: Participant
- **~Participant()**: Participant

#### Application
- **appl_name**: char*
- **leader_name**: char*
- **num_of_registry**: int
- **registry**: CORBA::Any *
- **appl_name()**: Application
- **Application()**: Application
- **~Application()**: Application
- **leader_name()**: Application
- **numofRegistry()**: Application
- **addRegistry()**: Application
- **removeRegistry()**: Application
- **setInfo()**: Application
- **getInfo()**: Application

#### Session
- **num_of_participant**: int
- **num_of_application**: int
- **sessionPermission**: int
- **passwd**: char*
- **session_name**: char*
- **present_partici**: int
- **present_appl**: int
- **participants**: Participant*
- **applications**: Application*
- **~Session_i()**: Session
- **Session_i()**: Session
- **passwd()**: Session
- **sessionPermission()**: Session
- **session_name()**: Session
- **joinSession()**: Session
- **leaveSession()**: Session
- **firstParticipantName()**: Session
- **nextParticipantName()**: Session
- **firstParticipantName()**: Session
- **nextParticipantName()**: Session

#### SSO
- **num_of_session**: int
- **present**: int
- **sessions**: CORBA::Any *
- **~SSO_i()**: SSO
- **SSO_i()**: SSO
- **registerSession()**: SSO
- **unregisterSession()**: SSO
- **firstSession()**: SSO
- **nextSession()**: SSO
- **~searchSession()**: SSO
- **~encodeDiscrimination()**: SSO
- **~decodeDiscrimination()**: SSO

---

CORBA Interface Definition Language (IDL)
4.3 CORBA
5 CORBA

...3... 5.2... 5.3...

...4...

CORBA

...5.1...

...5.2...

...5.3...

5.1...
CORBA はオブジェクトの通信を提供する技術で、ORB は CORBA の実装を指す。ORB はオブジェクトの通信を可能にし、CORBA の機能を実現するための基盤技術を提供する。

CORBA の中には、多線程 (multi-thread) と互換制御 (mutual exclusion) の概念が含まれている。多線程は、複数の処理が同時に動くことを可能にし、互換制御は互いに排他的にアクセスできるリソースの利用を制御する。

CORBA の重要な特性の一つは、リモートアクセスの容易さである。オブジェクトはリモート位置にあるサーバーからアクセスできるため、システムの構成を容易に変更でき、分散システムの開発が可能である。
5.3  

5.3.1  (authentication)
5.3.2  

**invite**
5.3.3  

```
5.3.3  

---

... (timeout) ... (acknowledge) ...

... (format) ...

... (type) ...

... (time to live) ...

... (auditing) ...

---
```
ÀÎÀÀÉÀºÀÇ ¿ç¿ëÀÚÀÇ ¾ÇÀÇÀûÀÇ Á¢±Ù "µî¿¡ ÀëÇÑ Á¤º¸¸¦ »ìÆìº¸°í" ±×¿¡ "ÀëÇÑ ÀëºñÃ¥À»¼¼¿òÀ¸·Î½á Àõ¿í ¾ÈÀüÇÑȯ°æÀ»Á¦°øÇÒ ÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â．

ÀÎÀÀÉÀºÀÇ ¿ç¿ëÀÚÀÇ ¾ÇÀÇÀûÀÇ Á¢±Ù "ÀëÇÑ Á¤º¸¸¦ »ìÆìº¸°í" ±×¿¡ "ÀëÇÑ ÀëºñÃ¥À»¼¼¿òÀ¸·Î½á Àõ¿í ¾ÈÀüÇÑȯ°æÀ»Á¦°øÇÒ ÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â．

ÀÎÀÀÉÀºÀÇ ¿ç¿ëÀÚÀÇ ¾ÇÀÇÀûÀÇ Á¢±Ù "ÀëÇÑ Á¤º¸¸¦ »ìÆìº¸°í" ±×¿¡ "ÀëÇÑ ÀëºñÃ¥À»¼¼¿òÀ¸·Î½á Àõ¿í ¾ÈÀüÇÑȯ°æÀ»Á¦°øÇÒ ÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â ¿©°ÇÀ»¼öÀÖ´Â．
6 CORBA

4 5 CORBA MAESTRO [32, 33, 34, 35]. API [34, 35], (Communication Service) [34, 35], (Session Service), (Multimedia Database Service), (Name Service), (Management Service) [36, 37, 38], (QoS Service) [39, 40], CORBA [9, 10] DCOM [41].

6.1 MAESTRO

MAESTRO [32, 33, 34, 35] Application Program Interface (API). API Application Programming Interface (API) [34, 35]. API [34, 35] DCOM [41].
Multimedia Collaborative Applications

Video/Audio Conferencing Tool, Whiteboard, Electronic Notebook, Chatting Tool, Shared Application, VOD, Telemedicine, Telepublishing, etc.

Multimedia Collaborative Service API

Distributed Multimedia Services


Distributed Services

CORBA, DCOM, etc.

Operating System and Physical Network

11: MAESTRO

MAESTRO: Video/Audio Conferencing Tool, Whiteboard, Electronic Notebook, Chatting Tool, Shared Application, VOD, Telemedicine, Telepublishing, etc.

1. Video/Audio Conferencing Tool: Video/Audio Conferencing Tool, Whiteboard, Electronic Notebook, Chatting Tool, Shared Application, VOD, Telemedicine, Telepublishing, etc.

(synchronization) (synchronization) (multi-point communication), (flow control)
2.  

3.  

4.  

5.  

6.  

6.2 MAESTRO  


6.2.1  

Solaris 2.5  

SPARC Compiler C++ 4.1  

Motif C++ 4.1 

6.2.2 MAESTRO  

IONA Orbix 2.3c  

SPARC Compiler C++ 4.1  

Motif C++ 4.1  

6.2.3  

MAESTRO  

IONA Orbix 2.3c  

SPARC Compiler C++ 4.1  

Motif C++ 4.1  

6.2.4  

IONA Orbix 2.3c  

Solaris 2.5  

sso->setInfo(_discrimination, info);
sso-> destroyApplication(_discrimination);

6.3  tàÀÀ¿ë ÇÁ·Î±×·¥ÀÌ Á¾·á µÉ ¶§´Â Ø¦È£ÀÔÀ¸·Î½á ÀÚ½ÅÀÇ µî·ÏÁ¤º¸¸¦ Á¦°ÅÇÑ´Ù.

6.3.1  tàÀÀ¿ë ÀÌ

Àà¿ë ÀÇÀ»»ý¼º À»Çϰí, ÇÔ²² Çùµ¿ ÀÛ¾÷ À»ÁøÇàÑ´Ù.

12:  tàÀÀ¿ë Áø ÇÔ²² Çùµ¿ ÀÛ¾÷ ÊÌÀº»ç¿ë ÀÚ¿¡°Ô ºñ°ø°³ ÀÇ ÀÇ ¾ÏÈ£ »ç¼¼. ÀÌ·ç¾î Áø´Ù.
±×¸² 12:  tàÀÀ¿ë ·Î±× ÀÎ ÀÎÀ» È­¸é ÀÌ´Ù.
6.3.2  

6.3.3  

6.3.4  

42
7.

CORBA Common Object Services Specification (COSS)

Non-CORBA

µÎ¹øÂ°·Î´Â ±âÁ¸ÀÇ Non-CORBA ±â¹ÝÀÇ 

µ¿ ¹®Á¦ÀÌ´Ù.
CORBA provides distributed objects.


[35] À±ÅÂÇü, °øÁö¿µ, ½Å¿µ¹Ì, È«¿ø±â, ¼­¿µÈ£, ”ºÐ»ê¸ÖƼ¹Ìµð¾îÀÀ¿ëÇÁ·Î±×·¥À»Áö¿øÇϱâÀ§ÇÑMAESTROÅë½Å¼­ºñ½º(MAESTRO Communication Service for Supporting Distributed Multimedia Applications)”, Á¤º¸°úÇÐȸ³í¹®Áö(A), Vol. 23, July 1998, pp. 681-694.


48

[40]  "", 


typedef sequence<short> PortSeq;
typedef sequence<string> DomainSeq;
module Session_Server {
    exception Reject {string reason;};
    exception NotDescription();
    exception NotDiscrimination();
    exception NotFound();
    exception ObjectChanged();

    interface SSO {
        boolean registerSession(in any session)
            raises (Reject);
        boolean unregisterSession(in any session, in string passwd)
            raises (Reject, NotDescription);
        /** list ***********************************************************/
        /* return session ptr*/
        any firstSession()
            raises (Reject, NotFound);
        any nextSession(in any session)
            raises (Reject, NotDescription, ObjectChanged);
    }

    interface Session {
        readonly attribute short sessionPermission;
        readonly attribute string passwd;
        readonly attribute string session_name;

        boolean joinSession (in string name, in string domain);
        boolean leaveSession (in string name);

        /** common to application***************************/
        string createApplication(in string appl_name, in string name)
            raises (Reject, NotDescription);
    }
boolean destroyApplication(in string discrimination)
    raises (Reject, NotDiscrimination);
boolean leaveApplication(in string discrimination)
    raises (Reject, NotDiscrimination);

/** join & leave to ONE leader application *******************/
/** return discrimination made by name, session_name, appl_name **/
/** create & destroy to MULTI leader style application **********/
string joinApplication(in string appl_name, in string leader_name,
    in string name)
    raises (Reject, NotDescription);

/*return participant*/
string firstParticipantName()
    raises (NotDescription, NotFound);
string nextParticipantName(in string partici_name)
    raises (Reject, NotDescription, ObjectChanged);

/** for application interface***********************************/
boolean setInfo(in string discrimination, in any info)
    raises (Reject, NotDiscrimination);
any getInfo(in string discrimination)
    raises (Reject, NotDiscrimination, NotDescription);
};

interface ApplicationDB {
    boolean addApplication(in string appl_name, in string description,
        in string location)
        raises (Reject);
    boolean deleteApplication(in string appl_name) raises (Reject);

    string firstApplication();
    string nextApplication(in string appl_name);
};

Domain.idl
module Domain_Server {
    interface Domain {
        boolean add (in string session_location);
        boolean remove (in string session_location);

        string firstDomainName();
        string nextDomainName(in string domain_name);
    }
}
```idl
module Message_Server {
    interface Msg {
        string get();
    };

    interface Invite_Msg: Msg {
        //YES, NO
        boolean result(in string result);
    };

    interface Deliver {
        boolean put (in any msg);
        boolean send (in any msg, in string host);
        Msg get(in string name);
    };
};
```
53
**Journal Papers**


**Conference Papers**
