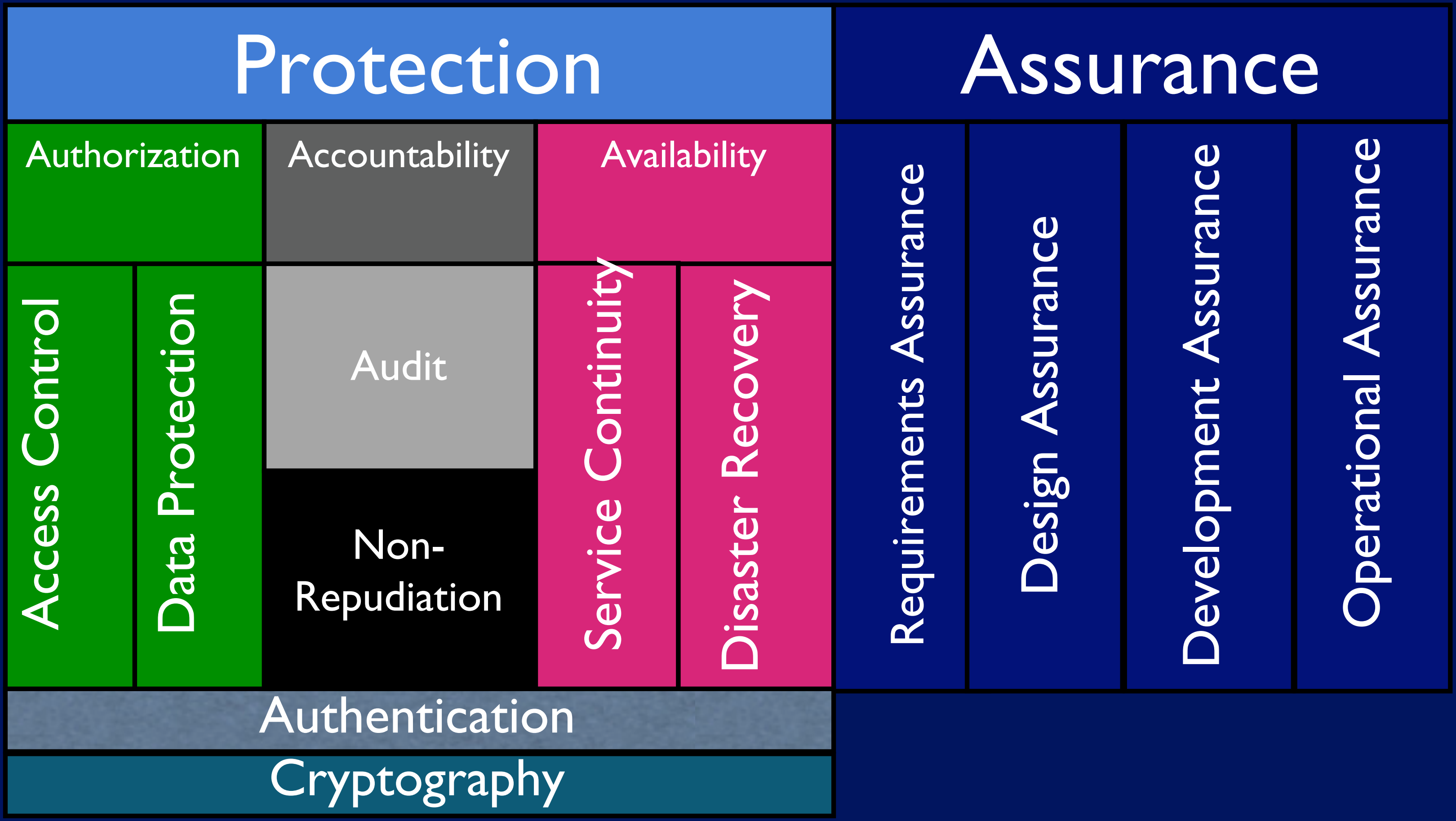
Middleware and Web Services Security Mechanisms

Secure Application Development

Module 9

Konstantin Beznosov

Conventional Computer Security



Protection

Authorization

Accountability

Availability

Access Control

Data Protection

Audit

Non-Repudiation

Service Continuity

Disaster Recovery

Authentication

Cryptography

Assurance

Requirements Assurance

Design Assurance

Development Assurance

Operational Assurance

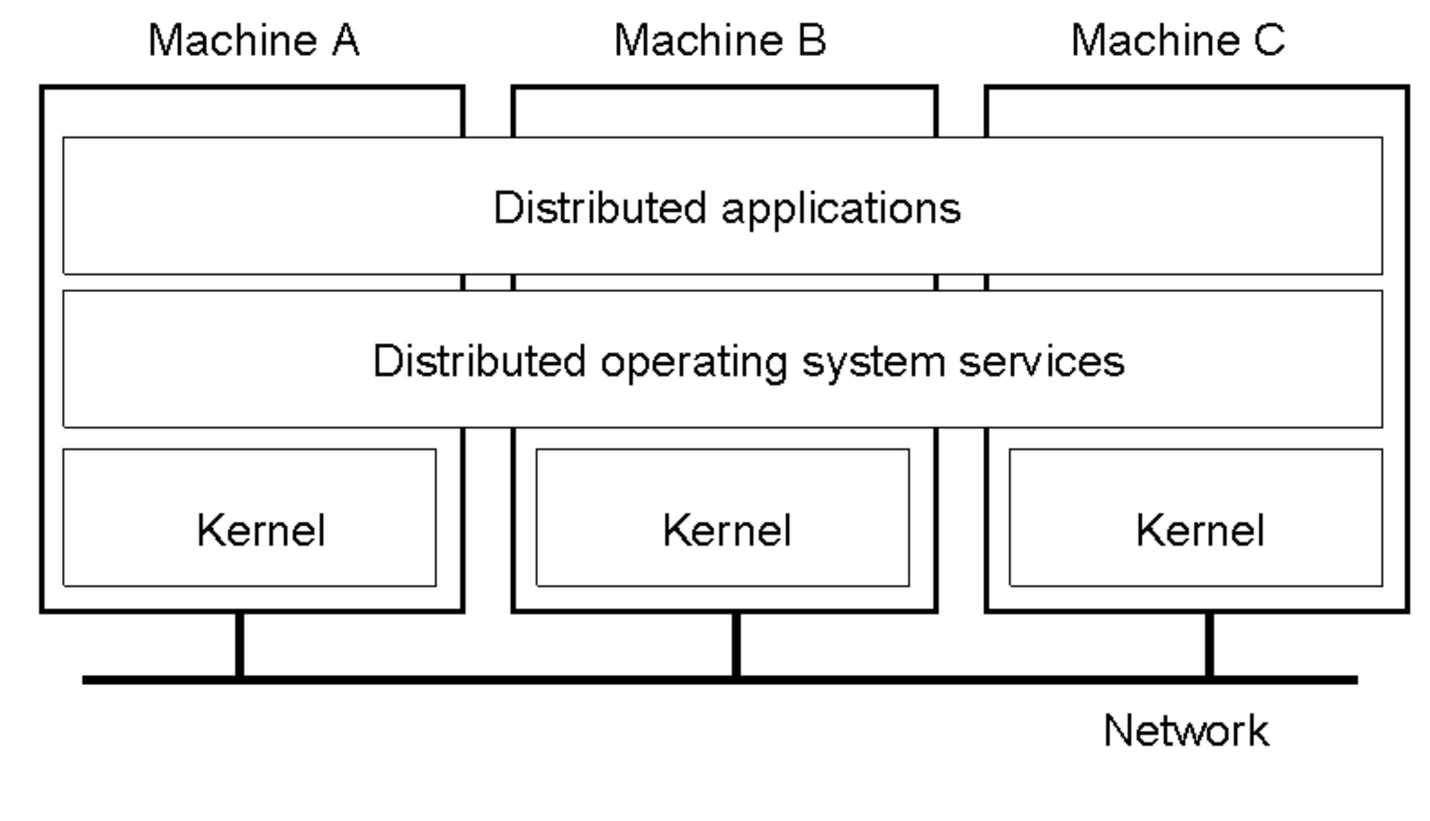
Outline

- Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- Security in middleware and Web services
  - What are common architectures for security mechanisms in most middleware and Web service technologies?
  - What are the differences among security mechanisms of COM+ and EJB?
- Conclusions
  - Summary
  - Where to go from here?

What is middleware?

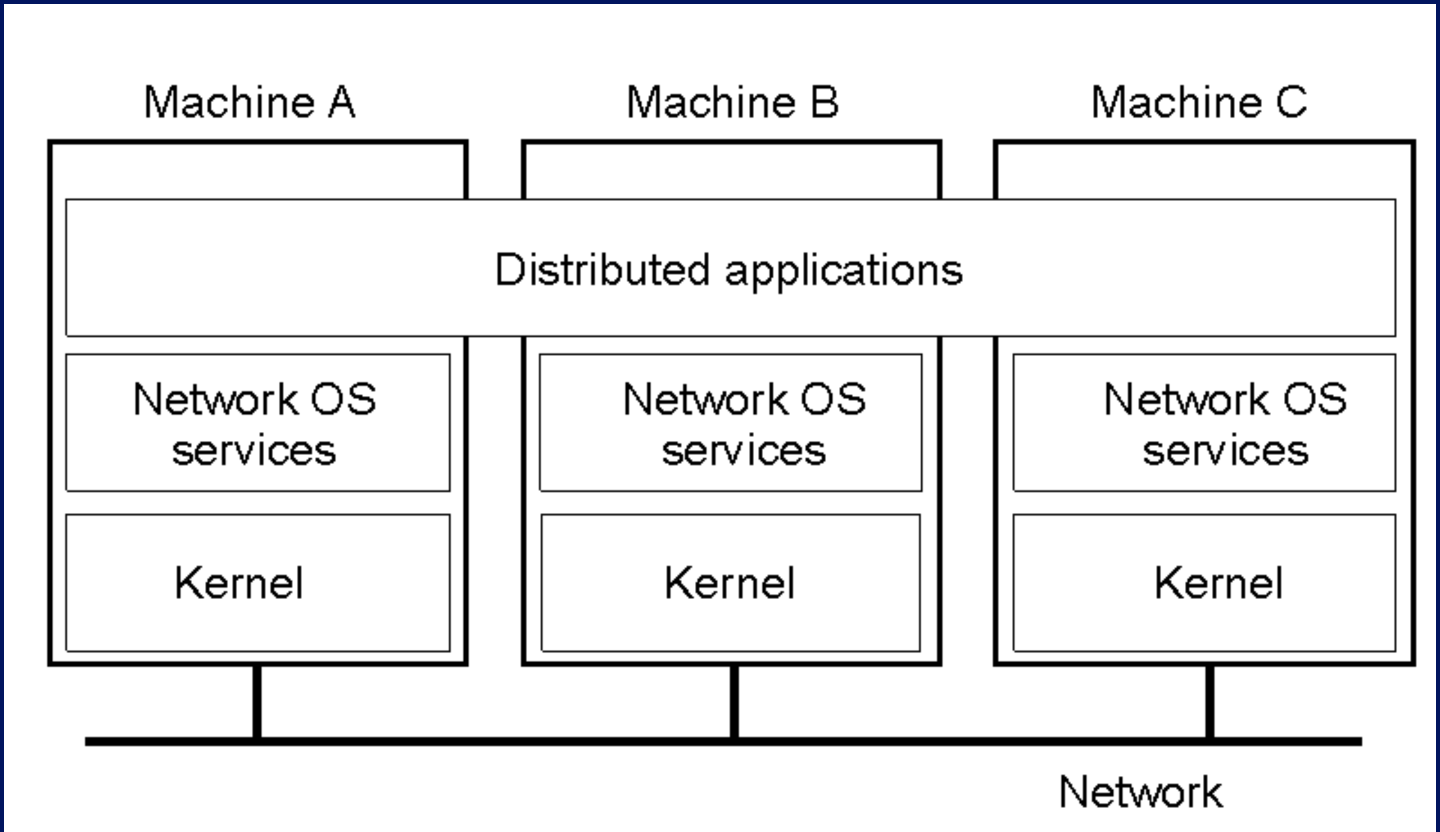
It’s what’s between
topware and underwear

Distributed Application Built Using DOS



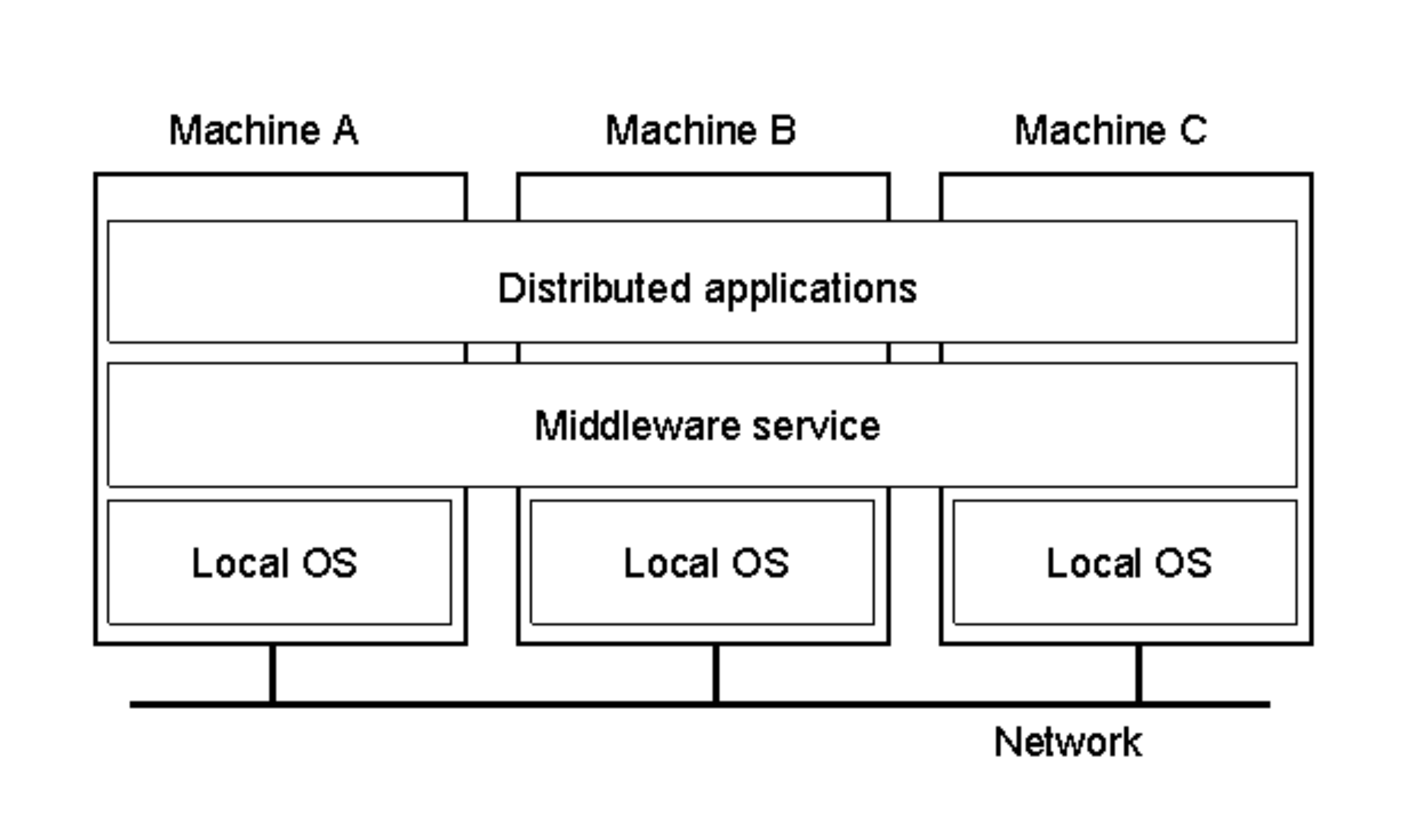
"Distributed Systems: Principles and Paradigms" by A. S. Tanenbaum, M. van Steen. Prentice Hall; (2002)

Distributed Application Built Using NOS



"Distributed Systems: Principles and Paradigms" by A. S. Tanenbaum, M. van Steen. Prentice Hall; (2002)

Distributed Application Built Using Middleware

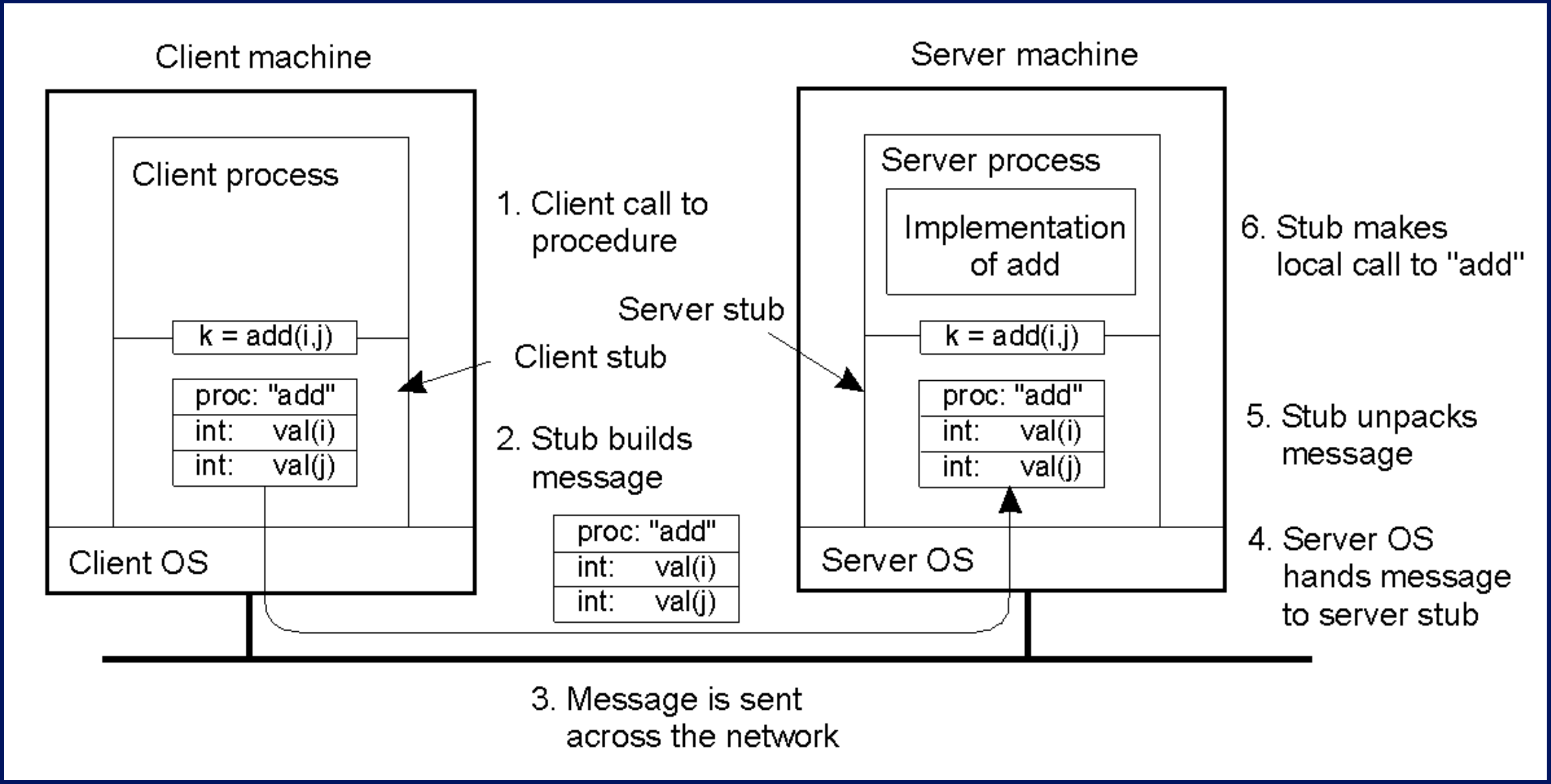


"Distributed Systems: Principles and Paradigms" by A. S. Tanenbaum, M. van Steen. Prentice Hall; (2002)

Software Support for Distributed Applications

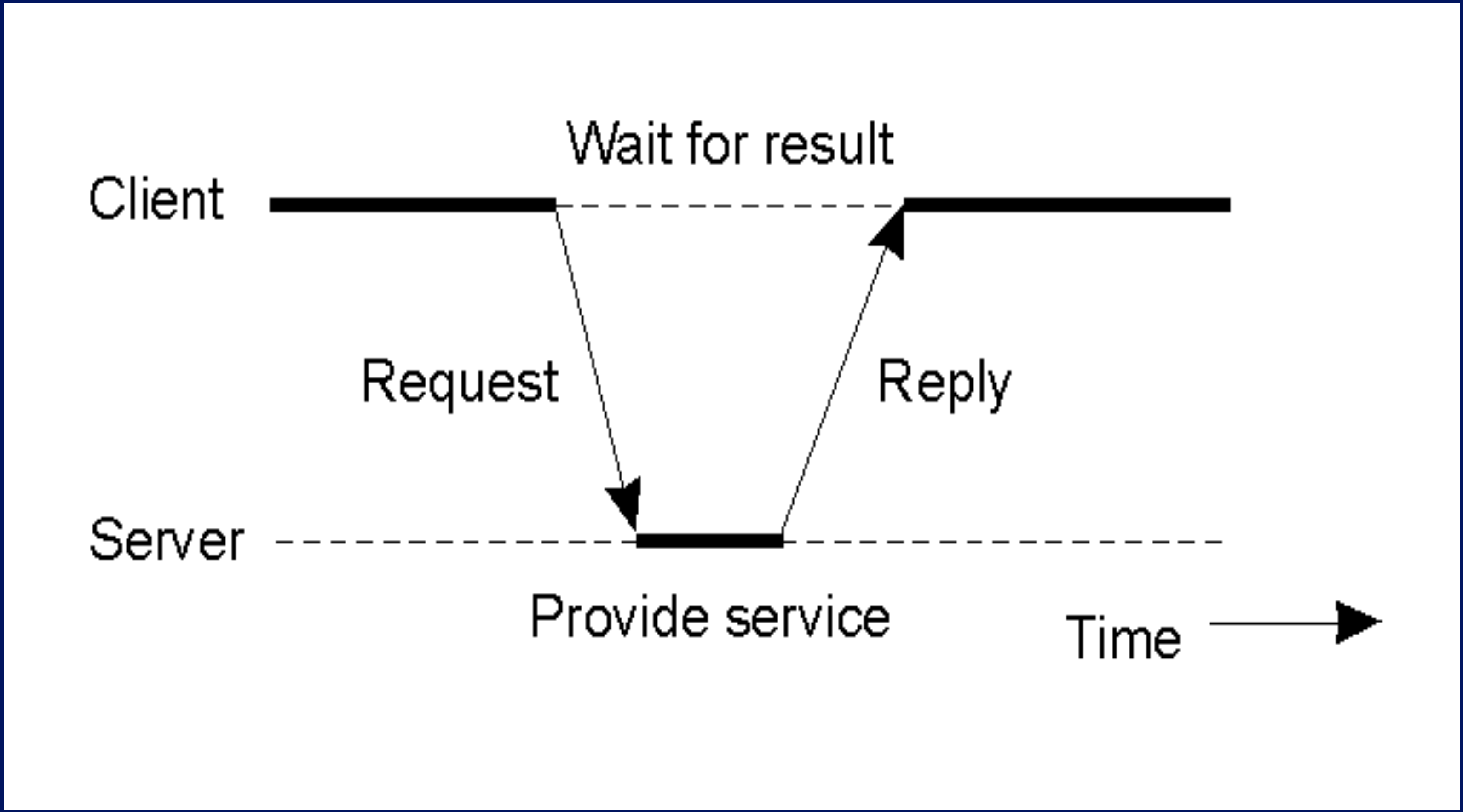
System	Description	Main Goal
DOS	Tightly-coupled operating system for multi-processors and homogeneous multicomputers	Hide and manage hardware resources
NOS	Loosely-coupled operating system for heterogeneous multicomputers (LAN and WAN)	Offer local services to remote clients
Middleware	**Additional layer atop of NOS implementing general-purpose services**	**Provide distribution transparency**

Most Middleware Uses Remote Procedure Call (RPC)



"Distributed Systems: Principles and Paradigms" by A. S. Tanenbaum, M. van Steen. Prentice Hall; (2002)

RPC Clients and Servers



"Distributed Systems: Principles and Paradigms" by A. S. Tanenbaum, M. van Steen. Prentice Hall; (2002)

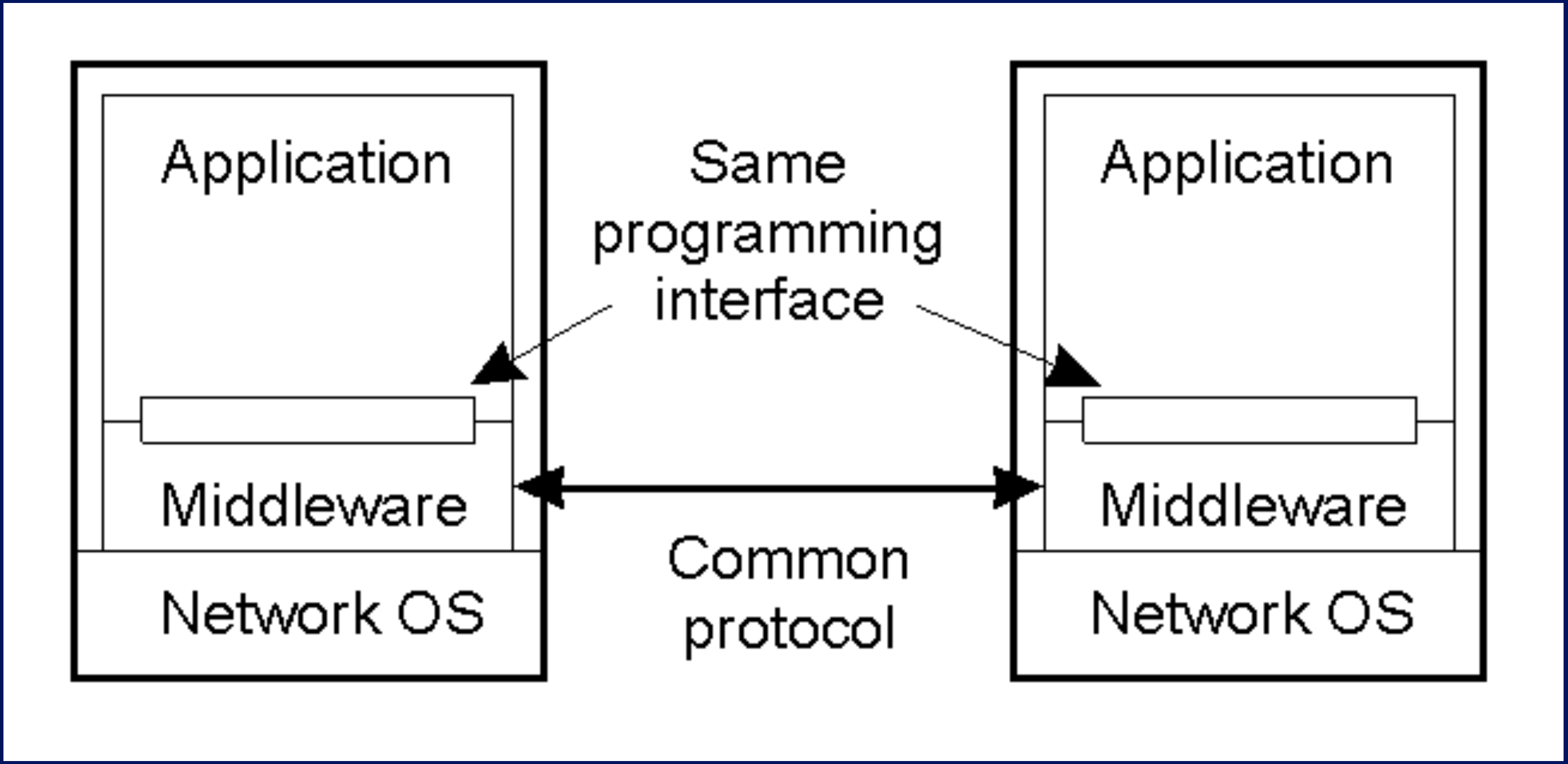
Distributed Objects

- Distributed Computing Environment (DCE) Remote Objects
- Common Object Request Broker Architecture (CORBA)
- Microsoft's Distributed Component Object Model (DCOM) & COM+
- Java Remote Method Invocation (RMI)
- Enterprise Java Beans (EJB)
- .NET Remoted Objects

Middleware Services

- Communication facilities
- Naming
- Persistence
- Concurrency
- Distributed transactions
- Fault tolerance
- Security

Middleware Openness



"Distributed Systems: Principles and Paradigms" by A. S. Tanenbaum, M. van Steen. Prentice Hall; (2002)

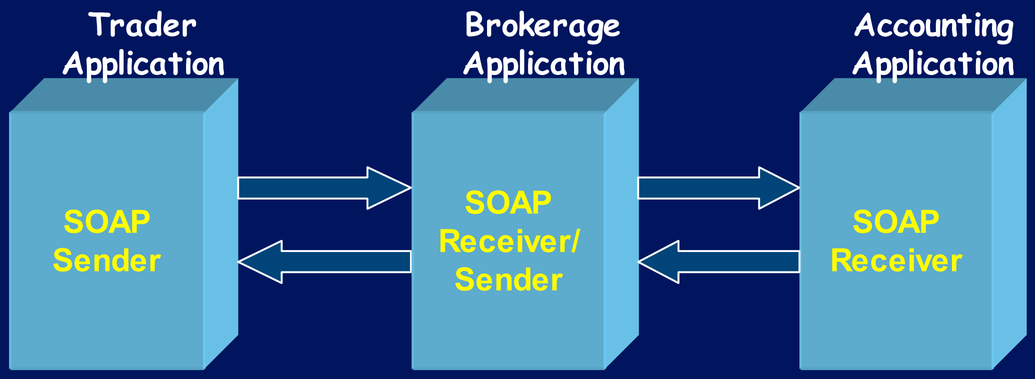
What’s Web Services?

How do middleware and Web services differ?

Features/ properties	**middleware**		**Web services**
	traditional	**MOM**	
Client server	yes	no	no
RPC	yes	no	no
OS independent	mostly	mostly	yes
Completeness and portability	yes	mostly	no
interoperability	yes	yes	yes

Promise of Web Services

- Interoperability across lines of business and enterprises
  - Regardless of platform, programming language and operating system
- End-to-end exchange of data
  - Without custom integration
- Loosely-coupled integration across applications
  - Using Simple Object Access Protocol (SOAP)



Web Services Features

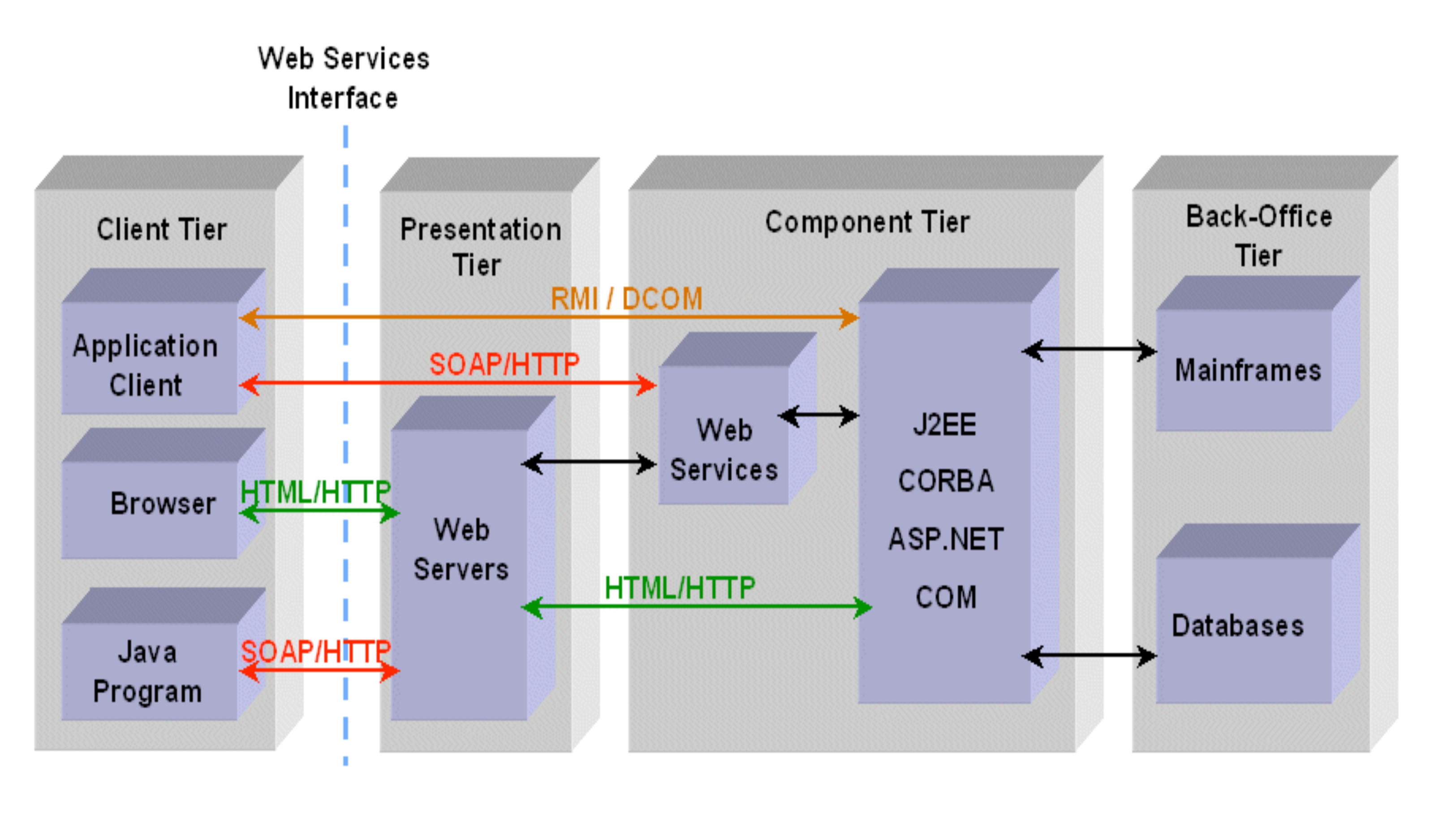
XML-based messaging interface to computing resources that is accessible via Internet standard protocols

- WS help intranet (business units) and extranet (business partners) applications to communicate
- SOAP – format for WS communications
  - Defined in XML
  - Supports RPC as well as document exchange
    - No predefined RPC semantics
  - Stateless
  - Can be sent over various carriers: HTTP, FTP, SMTP, … postal service

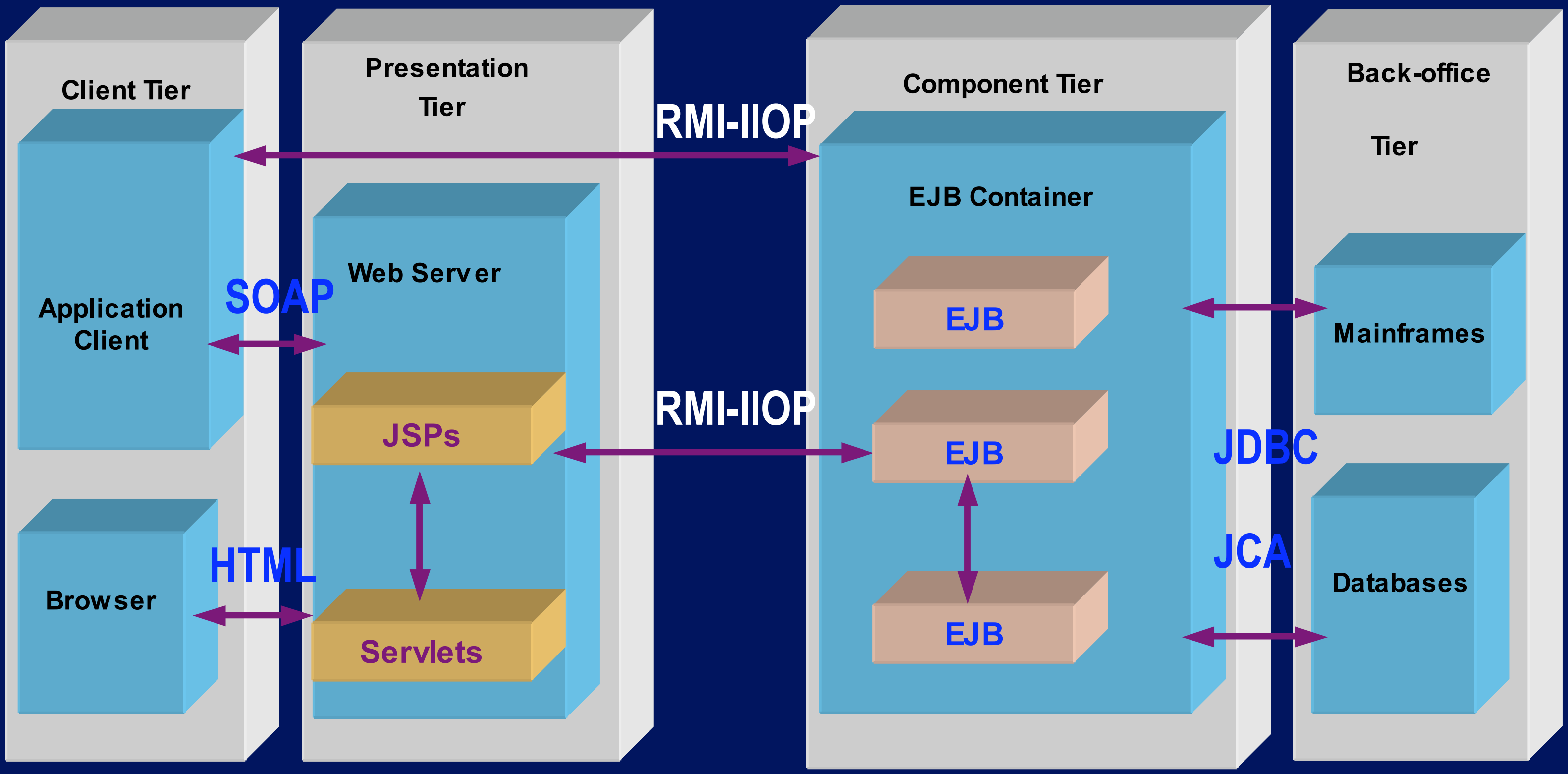
SOAP Message Example

```xml
<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://www.w3.org/2002/06/soap-envelope">
   <env:Header>
      <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
         <n:priority>1</n:priority>
         <n:expires>2001-06-22T14:00:00-05:00</n:expires>
      </n:alertcontrol>
   </env:Header>
   <env:Body>
      <m:alert xmlns:m="http://example.org/alert">
         <m:msg>Pick up Mary at school at 2pm</m:msg>
      </m:alert>
   </env:Body>
</env:Envelope>
```

Typical Web Service Environment



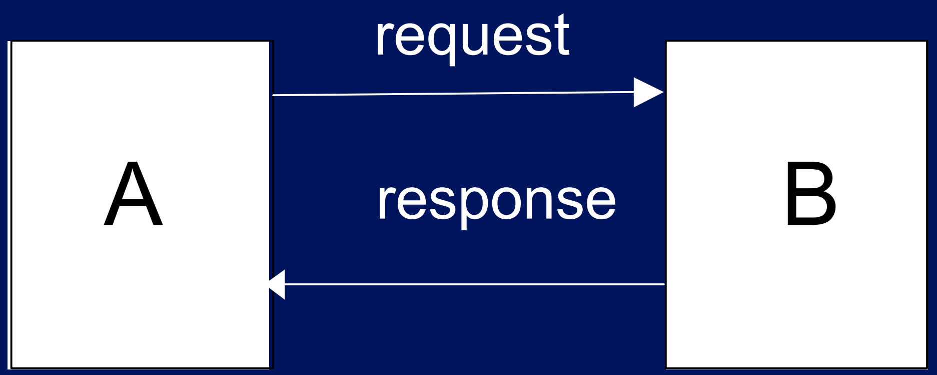
J2EE Web Service Systems

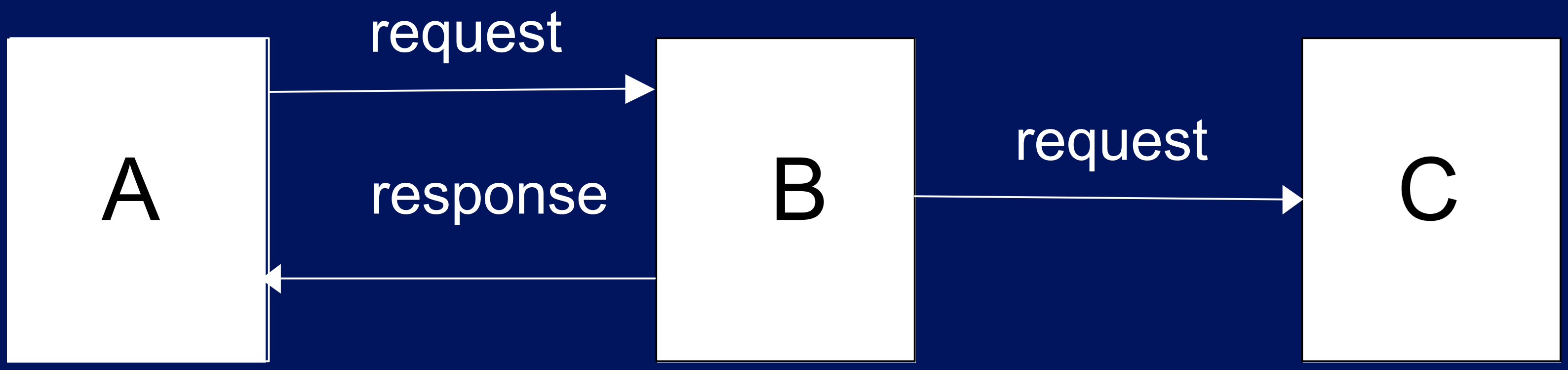


Outline

- Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- Security in middleware and Web services
  - What are common architectures for security mechanisms in most middleware and Web service technologies?
  - What are the differences among security mechanisms of COM+ and EJB?
- Conclusions
  - Summary
  - Where to go from here?

client-server paradigm & security





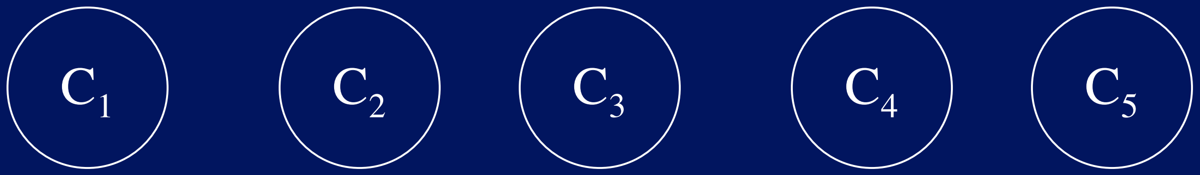
requirements due to distribution

- centralized administration
- localized run-time decisions

Online Course Application

Interface Course with methods

- postMaterials (Materials m, Module module)
- Materials getMaterials (Module module)
- submitAssignment (Assignment a)
- Assignment getAssignment (Student student, int number)
- postAssignmentInstructions (Instructions i, int number)



object paradigm & security (1/2)

- objects
  - small amounts of data ==> large numbers
    - R: Scale on large numbers of objects and methods
  - diverse methods ==> complex semantics
    - R: Security administrators should not have to understand semantics of methods
- collections
  - R: Similar names or locations should NOT impose membership in same collection(s).
  - R: For an object to be assigned to the same collection, name similarity and/or co-location should not be required.

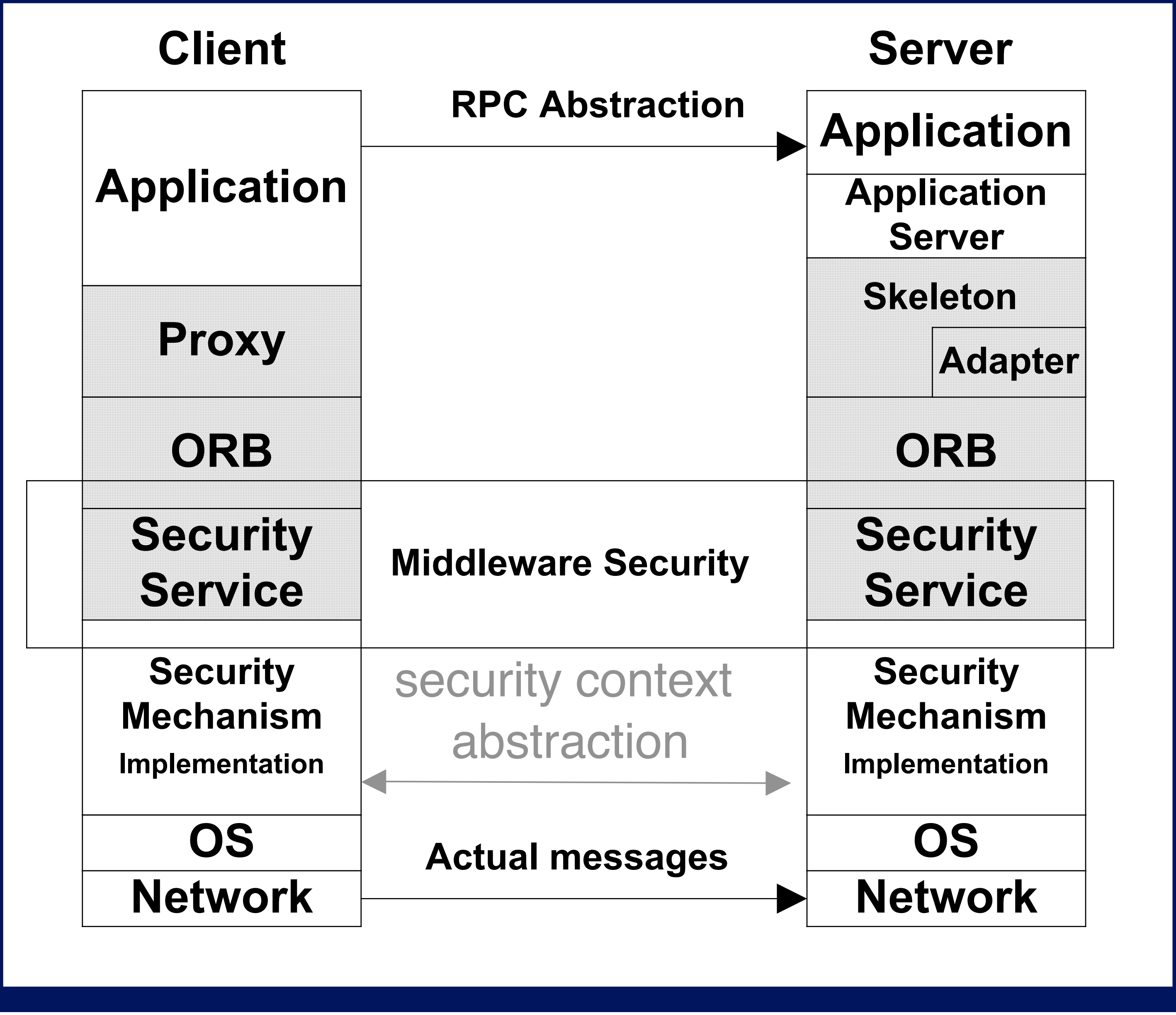
object paradigm & security (2/2)

- many layers of indirection and late binding
- names
  - multi-name, nameless and transient objects
    - R: Transient objects should be assigned to security policies without human intervention.
  - less rigid naming hierarchies
  - R: No assumptions that administrators know a name of each object in the system.

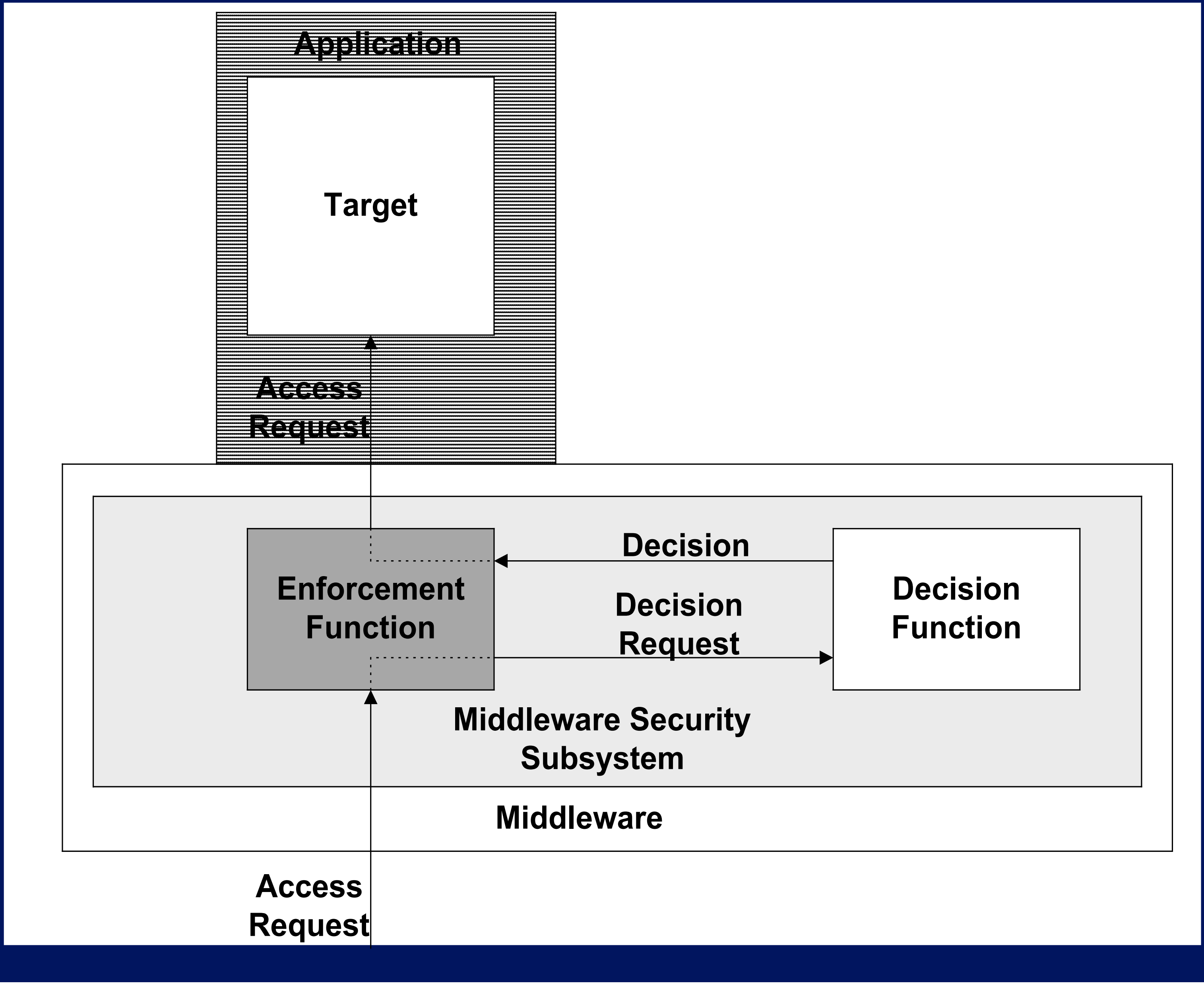
Outline

- Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- **Security in middleware and Web services**
  - What are common architectures for security mechanisms in most middleware and Web service technologies?
  - What are the differences among security mechanisms of COM+ and EJB?
- Conclusions
  - Summary
  - Where to go from here?

Middleware Security Stack



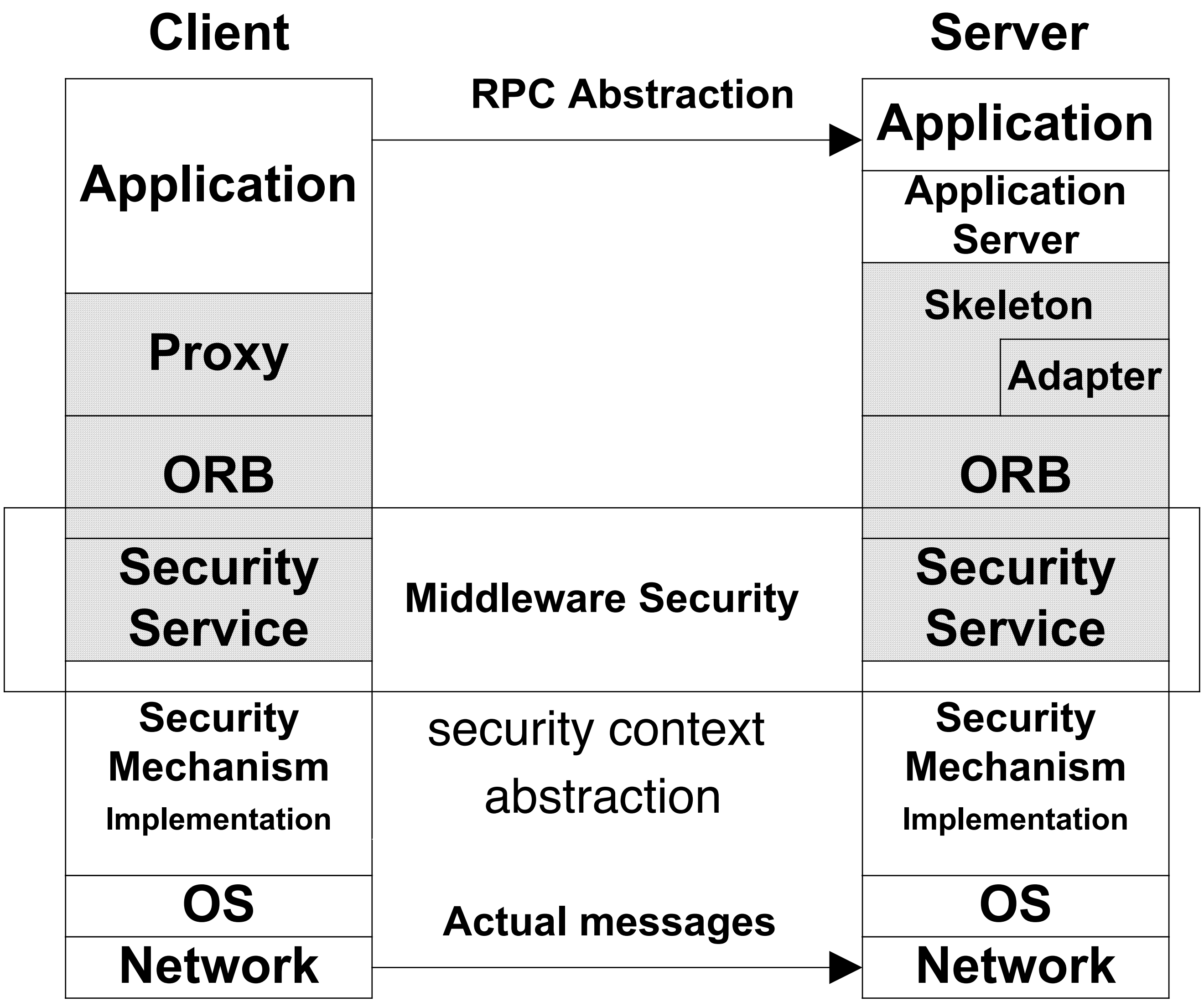
Policy Enforcement and Decision



Distributed Authentication

- Password-based
- Symmetric key
  - e.g., Kerberos
- Asymmetric key
  - e.g., PKI

Data Protection



Data Protection in Web Services

SOAP Message with WS-Security

```
<? Xml version=‘1.0’ ?>
<env:Envelope xmlns:env=“http://www.w3.org/2001/12/soap-envelope”
  xmlns:sec=“http://schmas.xmlsoap.org/ws/2002/04/secext”
  xmlns:sig=“http://www.w3.org/2000/09/xmldsig#”
  xmlns:enc=“http://www.w3.org/2001/04/xmlenc#”>
  <env:Header>
   <sec:Security
     sec:actor=“http://www.w3.org/2001/12/soap-envelope/actor/next”
     sec:mustUnderstand=“true”>
     <sig:Signature>
      …
     </sig:Signature>
     <enc:EncryptedKey>
      …
     </enc:EncryptedKey>
     <sec:BinarySecurityToken
      …
     </sec:BinarySecurityToken
   </sec:Security>
  </env:Header>
  <env:Body>
   <enc:EncryptedData>
    …
   </enc:EncryptedData>
  </env:Body>
</env:Envelope>
```

WS-Security

- Message integrity and message confidentiality
- Compliance with XML Signature and XML Encryption
- Encoding for binary security tokens
  - Set of related claims (assertions) about a subject
  - X.509 certificates
  - Kerberos tickets
  - Encrypted keys

XML Encryption

- Encrypt all or part of an XML message
- Separation of encryption information from encrypted data
- Super-encryption of data

```
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
  Type='http://www.w3.org/2001/04/xmlenc#Content'>
  <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#3des-cbc'/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
  <ds:KeyName>John Smith</ds:KeyName>
  </ds:KeyInfo>
  <CipherData>
    <CipherValue>A23B45C56</CipherValue>
  </CipherData>
</EncryptedData>
```

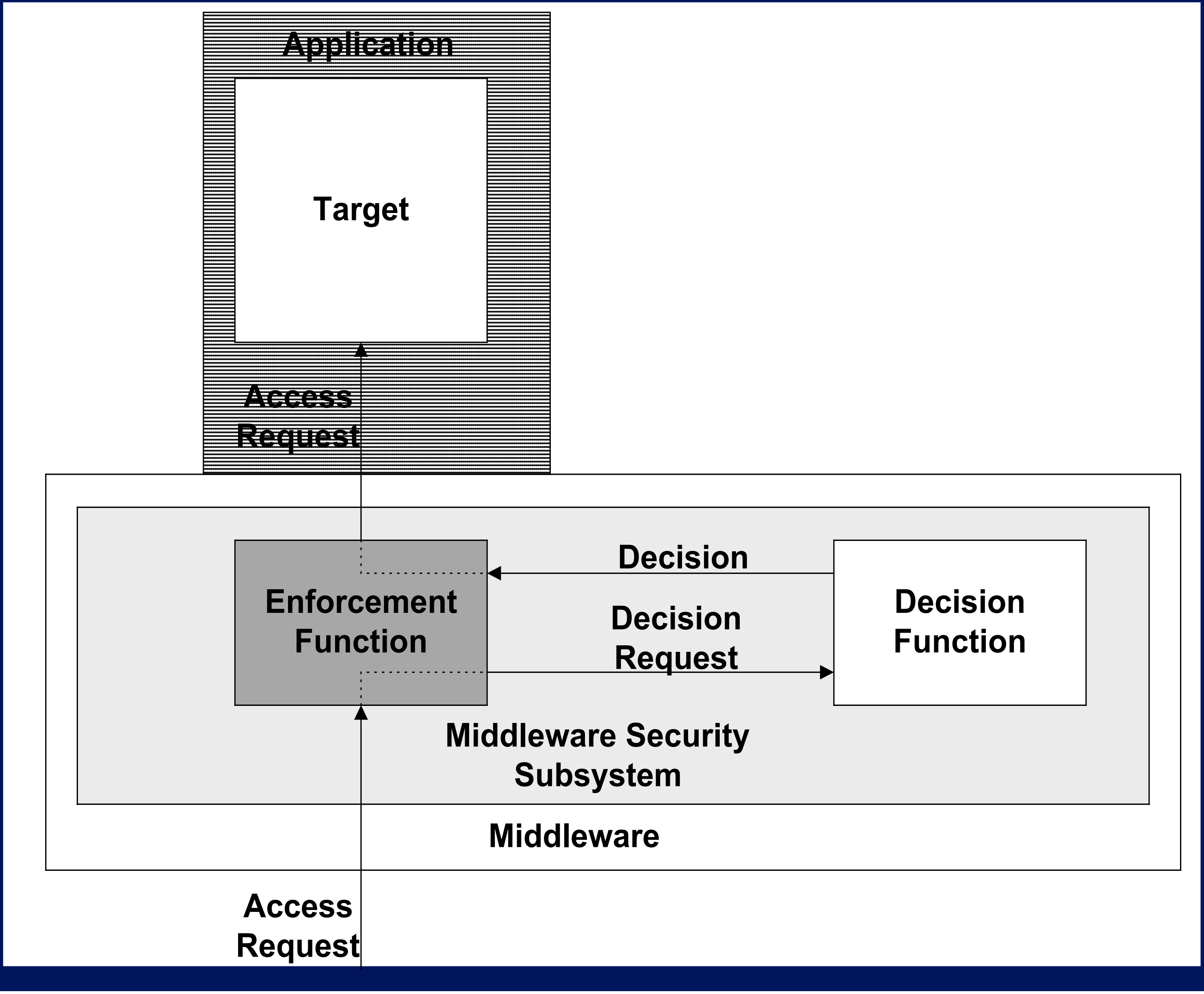
XML Signature

- Apply to all or part of a document
- Contains: references to signed portions, canonicalization algorithm, hashing and signing algorithm Ids, public key of the signer.
- Multiple signatures with different characteristics over the same content

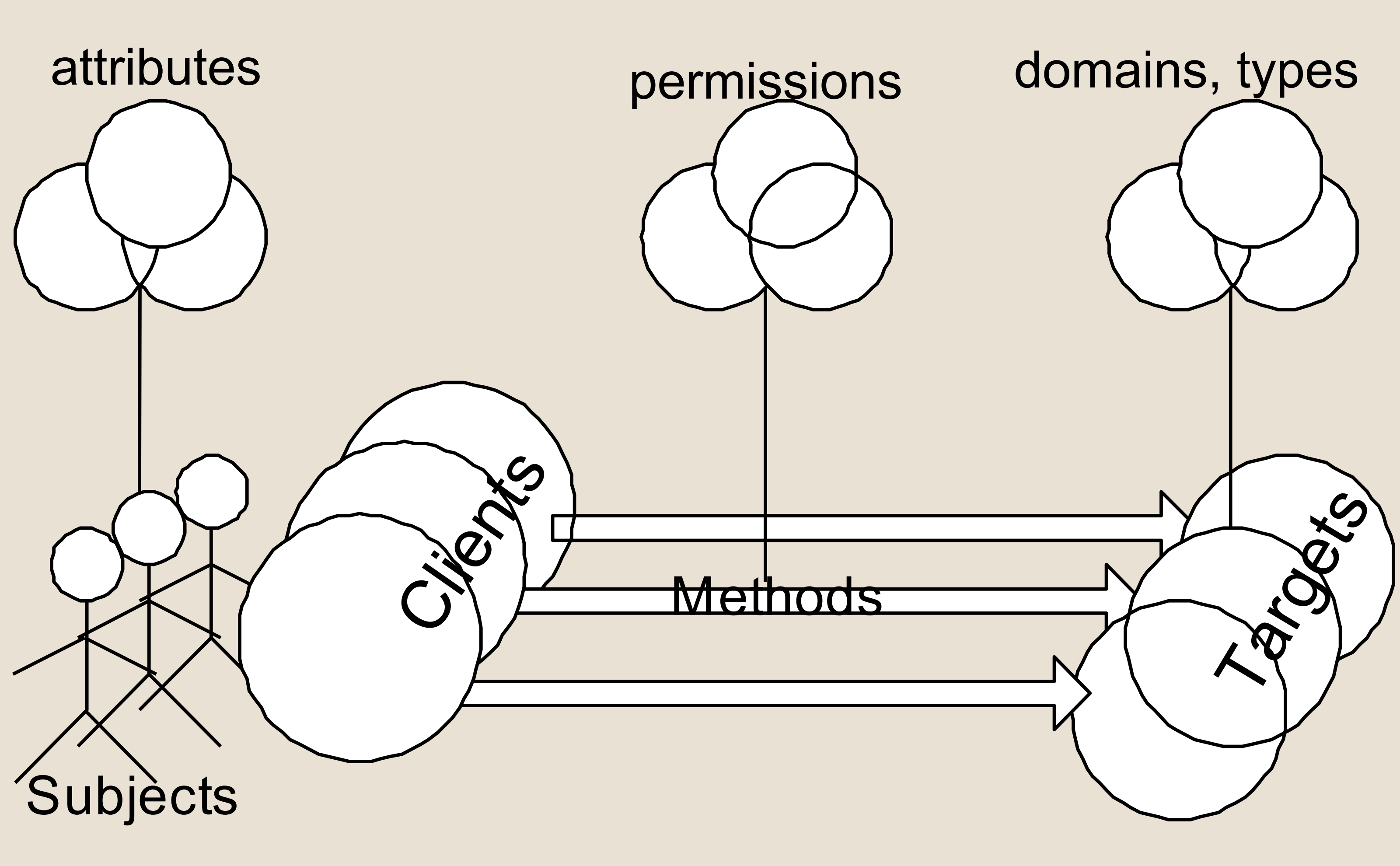
```
<Signature Id="MySignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/…/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
    <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>MC0CFFrVLtRlk=...</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <DSAKeyValue>
        <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
      </DSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

Security Policy Decisions

Policy Enforcement and Decision

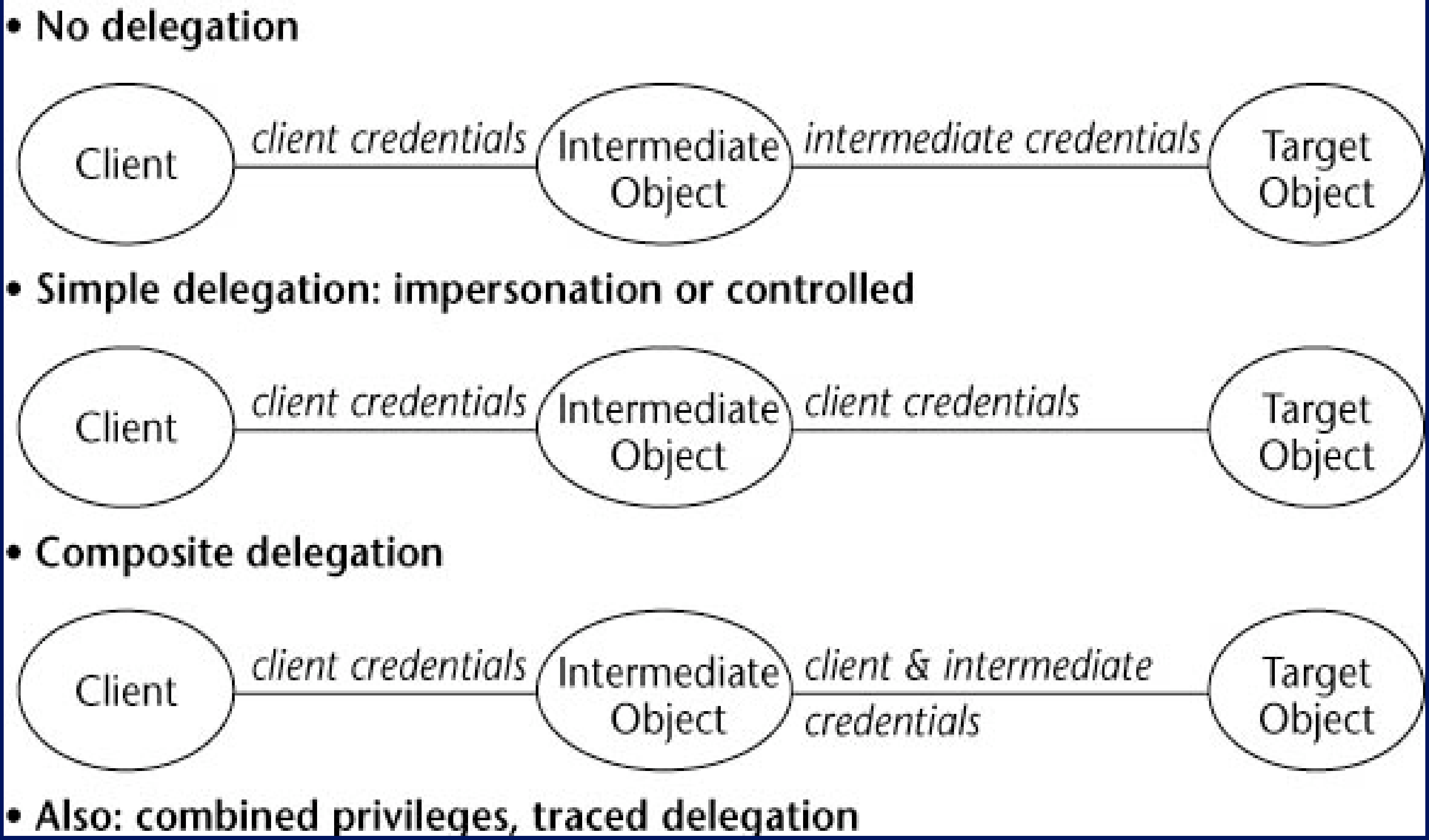


scaling policy decisions



Credentials Delegation

- What are credentials?
- Push and pull models



- No delegation

- Simple delegation: impersonation or controlled

- Composite delegation

- Also: combined privileges, traced delegation

Issues in Distributed Audit

- Monitor activity across and between objects.
- Order of the audit records is hard to determine because of the lack of global time.
- Performance
- No guarantee that an event has been logged.

Outline

- Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- **Security in middleware and Web services**
  - What are common architectures for security mechanisms in most middleware and Web service technologies?
  - What are the differences among security mechanisms of COM+ and EJB?
- Conclusions
  - Summary
  - Where to go from here?

COM+ Specifics

Authentication in COM+

- Supported mechanisms
  - Kerberos
  - Windows NT LAN Manager (NTLM)
- Granularity modes
  - Never
  - At the time of establishing secure channel
  - On every call
  - With every network packet
- Credentials delegation options
  - No delegation
  - Unconstrained simple delegation (a.k.a., impersonation)
    - Only one hop for NTLM

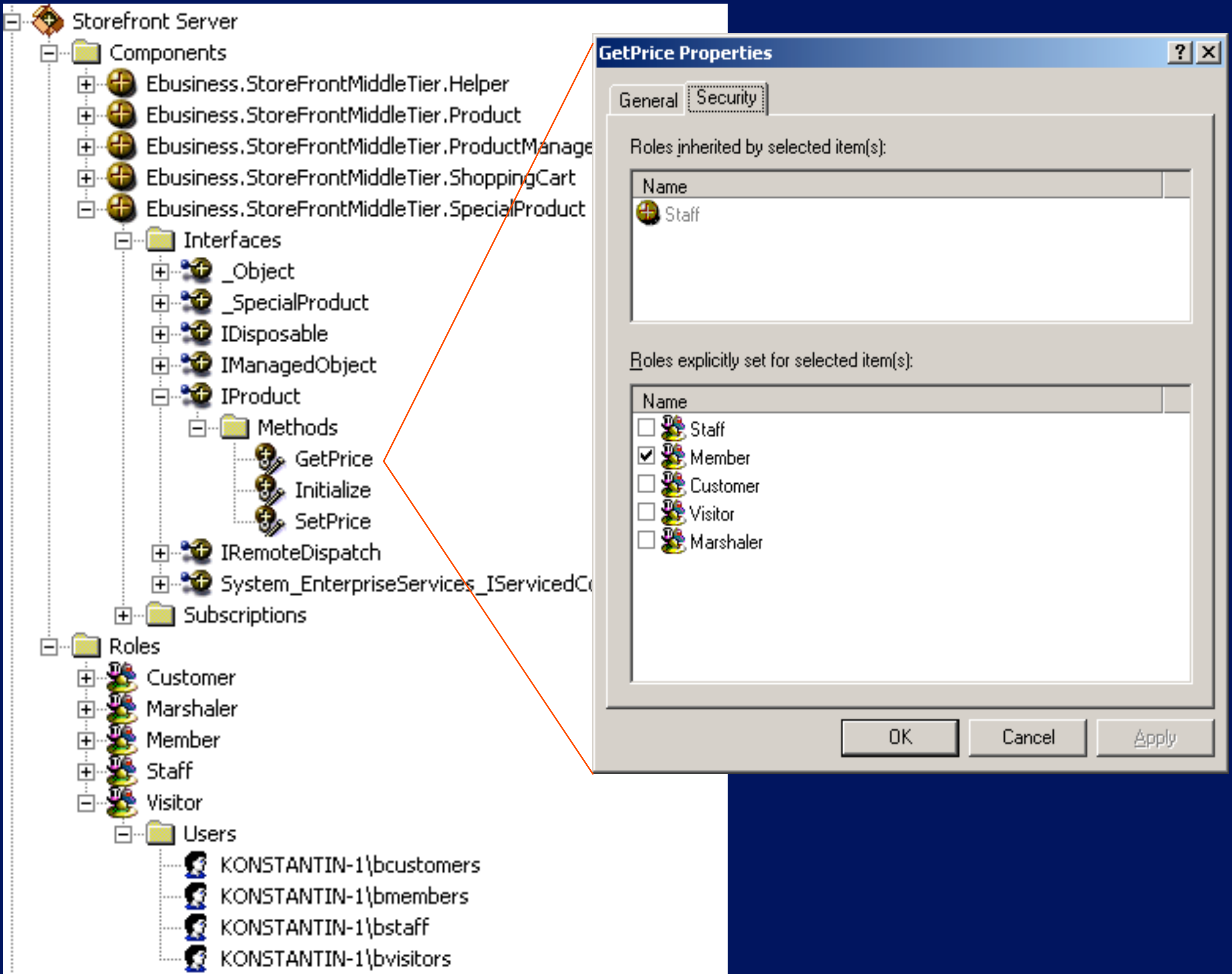
Data Protection in COM+

- Supported modes
  - Origin authentication and integrity protection
  - As above + confidentiality protection

Access Control in COM+

- The three hurdles to go through
  1. Activate server process
  2. Process border checks
  3. DLL border checks
- Granularity
  - Component
  - Interface
  - Method

Administering Access Control



```
Storefront Server
  Components
    Ebusiness.StoreFrontMiddleTier.Helper
    Ebusiness.StoreFrontMiddleTier.Product
    Ebusiness.StoreFrontMiddleTier.ProductManage
    Ebusiness.StoreFrontMiddleTier.ShoppingCart
    Ebusiness.StoreFrontMiddleTier.SpecialProduct
      Interfaces
        _Object
        _SpecialProduct
        IDisposable
        IManagedObject
        IProduct
          Methods
            GetPrice
            Initialize
            SetPrice
        IRemoteDispatch
        System_EnterpriseServices_IServicedC
      Subscriptions
  Roles
    Customer
    Marshaler
    Member
    Staff
    Visitor
      Users
        KONSTANTIN-1\bcustomers
        KONSTANTIN-1\bmembers
        KONSTANTIN-1\bstaff
        KONSTANTIN-1\bvisitors
```

GetPrice Properties

General | Security

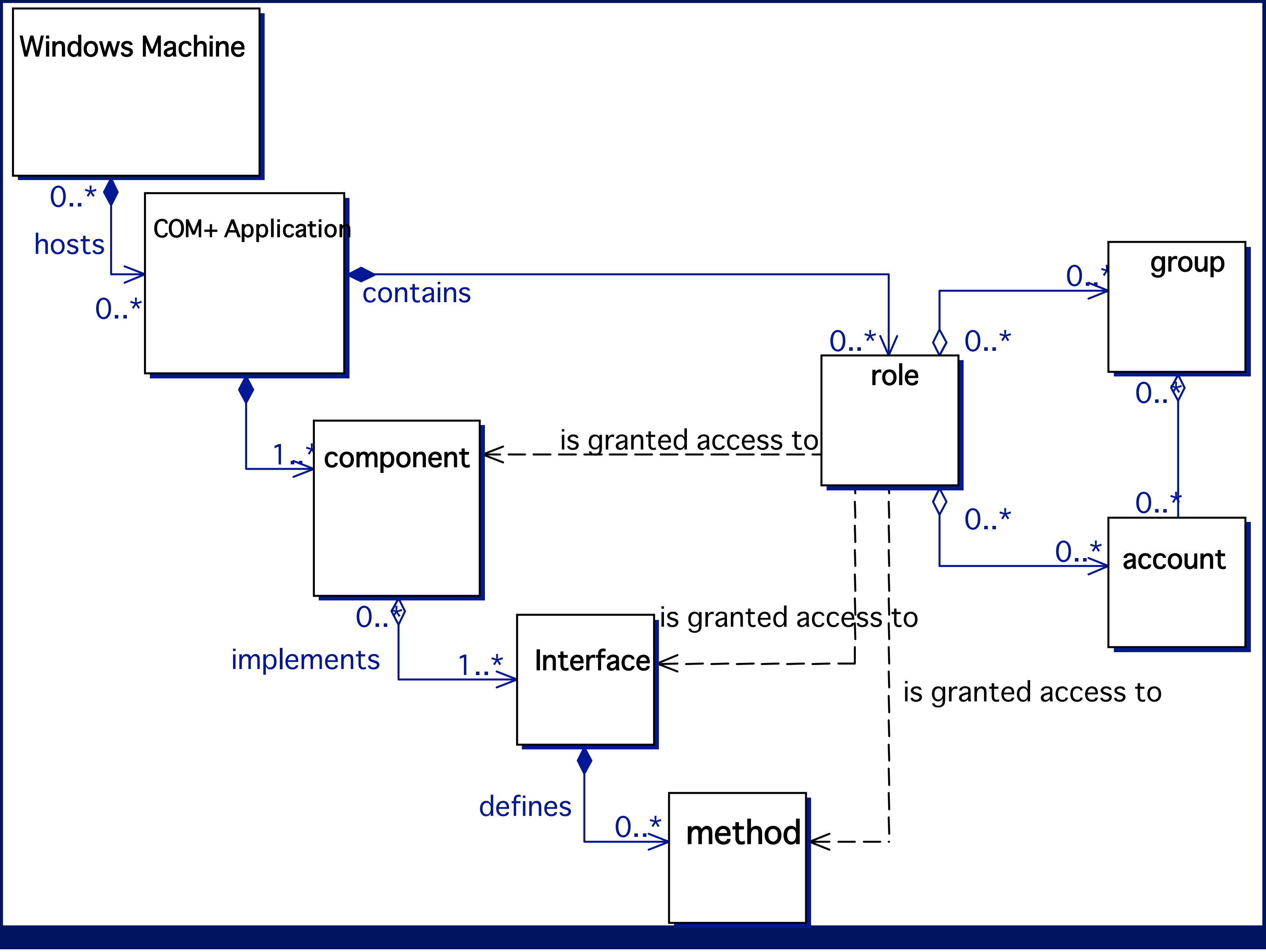
Roles inherited by selected item(s):

Name
Staff

Roles explicitly set for selected item(s):

Name
☐ Staff
☑ Member
☐ Customer
☐ Visitor
☐ Marshaler

COM+ Access Control Architecture



Application Description

Application:

- 10 students: s_1 … s_{10}
- 3 instructors: i_1, i_2, i_3
- 5 courses: c_1, … c_5
  - $C_1 = \{i_1, \{s_1, s_2, s_3\}\}$
  - $C_2 = \{i_2, \{s_3, s_4, s_5\}\}$
  - $C_3 = \{i_3, \{s_5, s_6, s_7\}\}$
  - $C_4 = \{i_1, \{s_7, s_8, s_9\}\}$
  - $C_5 = \{\{i_2, i_3\}, \{s_8, s_9, s_{10}\}\}$

Policy:

1. Students can
   1. read course material and assignment instructions for the courses they are registered
   2. submit (i.e., write) their assignments for the registered courses
2. Instructors can
   1. read student submitted assignments for the courses they teach, and
   2. post (i.e., write) course material and assignment instructions for their courses

Configure COM+ online course application to implement this policy

A Possible Solution

- Interface Course with methods
  - postMaterials (CourseId id, Materials m, Module module)
  - Materials getMaterials (CourseId id, Module module)
  - submitAssignment (CourseId id, int assignmentNumber)
  - getAssignment (CourseId id, Student student, int number)
  - postAssignmentInstructions (CourseId id, Instructions i, int number)

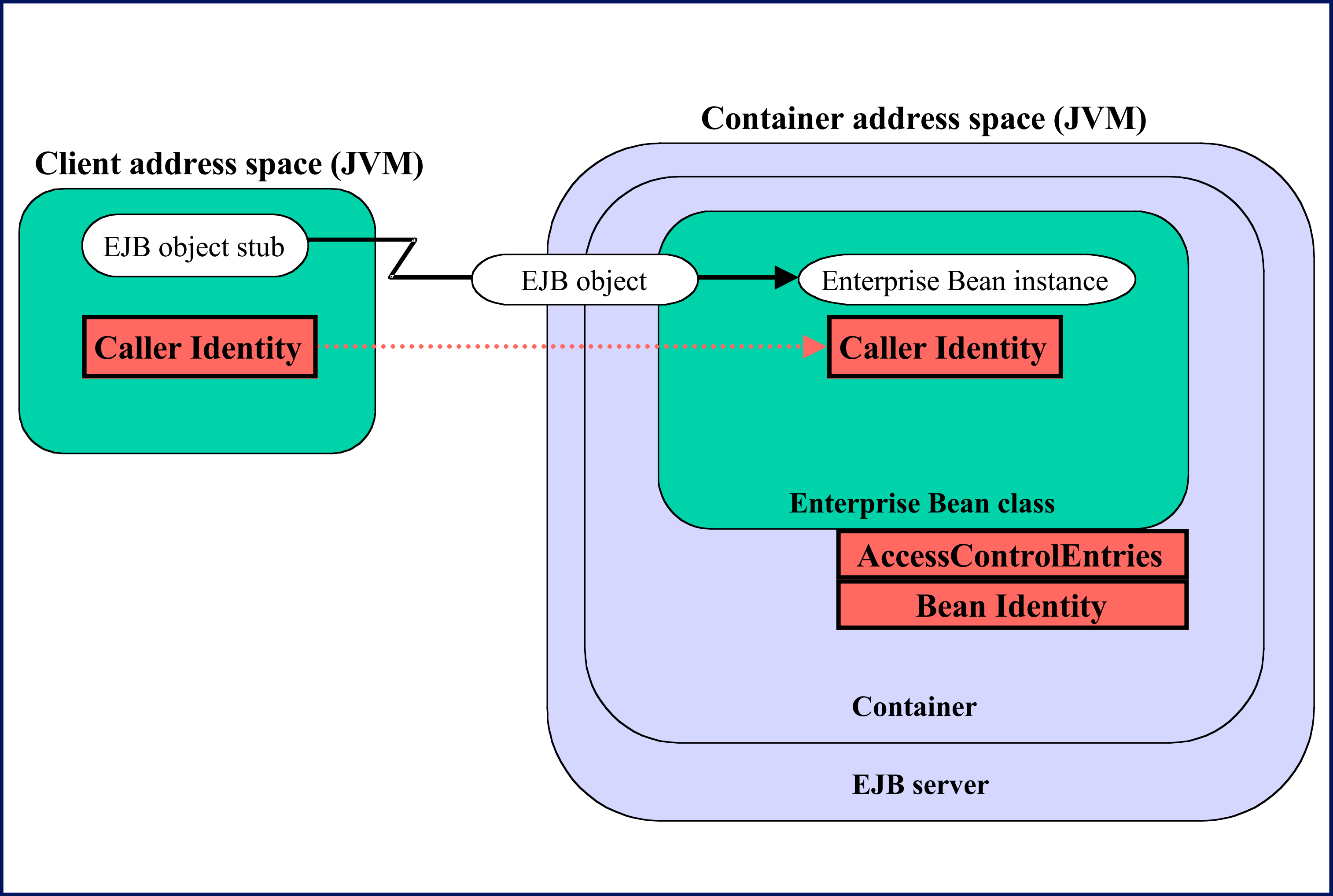
	student	instructor
postMaterials		+
getMaterials	+	+
submitAssignment	+	
getAssignment		+
postAssignmentInstructions		+

Accountability in COM+

- No out-of-the-box support
- Developers should rely on Windows event logs

EJB Specifics

EJB Run-time Security



Common Secure Interoperability (CSI) v2 defines wire protocol

Authentication in EJB

- Defines only the use of JAAS for authenticating and credentials retrieving
- Implementation-specific
- Credentials delegation options
  - No delegation
  - Unconstrained simple delegation (a.k.a., impersonation)

Data Protection in EJB

- Implementation-specific

Access Control in EJB

- Configured through deployment descriptor
- Granularity
  - Down to individual method on a class, but not bean instance
  - Can be different from JAR to JAR
- Expressiveness
  - method grouped into "method permissions"
  - Subjects grouped by plain roles
  - No role hierarchy
- Java Authorization Contract for Containers (JACC)
  - APIs for plugging authorization engines

Defining Roles in EJB

```
<assembly-descriptor>
    <security-role>
        <description>
           blah-blah-blah …
        </description>
        <role-name>student</role-name>
    </security-role>

    <security-role>
        <description>
             blah-blah-blah …
        </description>
        <role-name>instructor</role-name>
    </security-role>
    …
</assembly-descriptor>
```

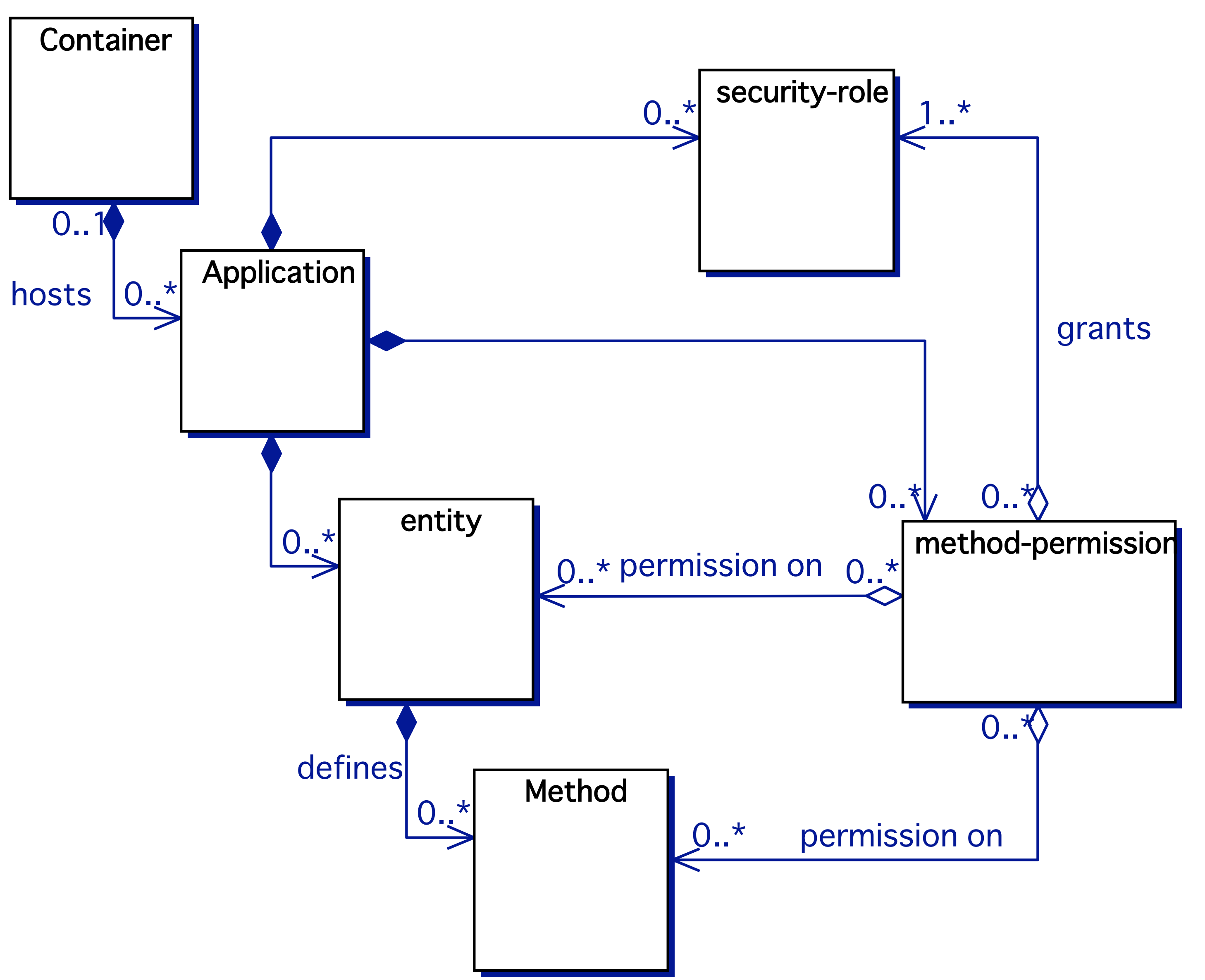
Assigning Users to Roles in EJB

```
<security-role-mapping>
  <role-name>student</role-name>
  <principal-name>S1</principal-name>
  <principal-name>S2</principal-name>
  <group-name>students</group-name>
 </security-role-mapping>

 <security-role-mapping>
  <role-name>instructor</role-name>
  <principal-name>I1</principal-name>
 </security-role-mapping>
```

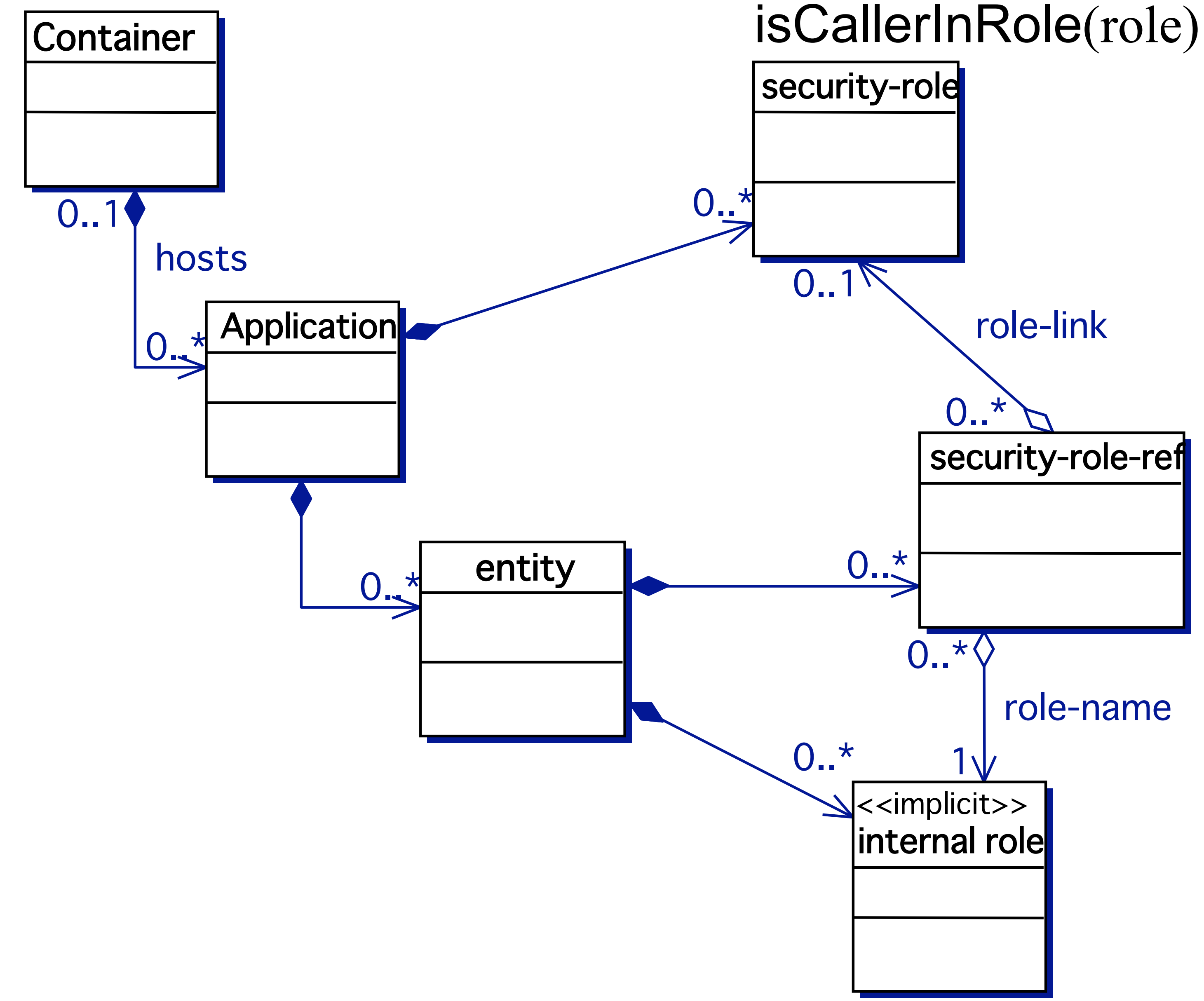
Assigning Methods to Roles in EJB

```
<method-permission>
    <role-name>student</role-name>
    <method>
        <ejb-name>Course</ejb-name>
        <method-name>getMaterials</method-name>
       <method-name>submitAssignment</method-name>
    </method>
</method-permission>

<method-permission>
    <role-name>instructor</role-name>
    <method>
       <ejb-name>Course</ejb-name>
          <method-name>postMaterials</method-name>
          <method-name>getAssignment</method-name>
    </method>
</method-permission>
```

roles and permissions in EJB



Fine-grain authorization in EJB



Accountability in EJB

- Implementation-specific

Summary

- Middleware & Web services
  - Software layer between OS and application to provide transparencies
  - Security-related issues: scaling, granularity, naming
- Security in Middleware & Web services
  - Common features/elements
  - Technology/product specific

Where To Go From Here?

- B. Hartman, D. J. Flinn, K. Beznosov, and S. Kawamoto, chapter 7, Mastering Web Services Security, John Wiley & Sons, Inc., 2003.
- E. Roman, S. Ambler, and T. Jewell, Mastering Enterprise JavaBeans, Second ed: Wiley Computer Publishing, 2002.
- B. Hartman, D. J. Flinn, and K. Beznosov, Enterprise Security With EJB and CORBA. John Wiley & Sons, Inc., 2001.
- "Security Engineering …" by Ross Anderson