
Only the best
got in ;-)
Online presentations soon available at http://wiki.javapolis.com
JAVAPOLIS

JavaPolis 2004

Middleware and Web Services Security

Dr. Konstantin Beznosov
Assistant Professor
University of British Columbia

Do you know what these mean?

- SOAP
- WSDL
- IIOP
- CSI v2

JAVAPOLIS

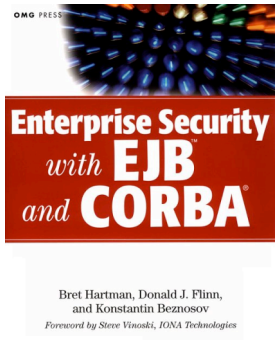
Overall Presentation Goal

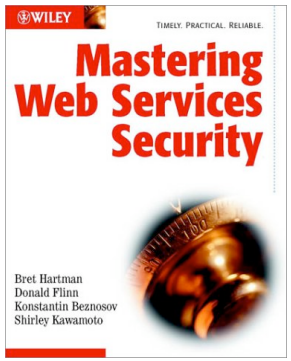
Learn what security mechanisms are available in middleware and Web services products

Speaker's Qualifications

Konstantin

- Worked for end-user, consulting, and developer organizations
- Co-authored CORBA Security standards proposals
  - Resource Access Decision
  - Security Domain Membership Management (SDMM)
  - CORBA Security
- Co-authored





This Slide Gains Your Audience's Attention

I do not believe current tools, technologies, and methodologies support "Extreme" Performance Testing.

JAVAPOLIS

How many of you can explain?

- Various security mechanisms
- What middleware and Web services are
- What makes middleware and Web services security special
- What common architectures for security mechanisms are in most middleware and Web service technologies
- What are the differences among security mechanisms of various middleware and Web service technologies?

JAVAPOLIS

Outline

- Part I: Security
  - What are security mechanisms?
- Part II: Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- Part III: 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?
- Part IV: Conclusions
  - Summary
  - Where to go from here?

Outline

- **Part I: Security**
  - What are security mechanisms?
- Part II: Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- Part III: 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?
- Part IV: Conclusions
  - Summary
  - Where to go from here?

What is Computer Security?

- security -- “safety, or freedom from worry”
- How can it be achieved?
  - Get rid of the sources of worry
  - Don’t trust computers anything valuable
  - Make computers too heavy to steal
  - Buy insurance (liability transfer)
  - Create redundancy (disaster recovery services)

Goals of Security

- Prevention
  - Prevent attackers from violating security policy
- Detection
  - Detect attackers' violation of security policy
- Recovery
  - Stop attack, assess and repair damage
  - Continue to function correctly even if attack succeeds

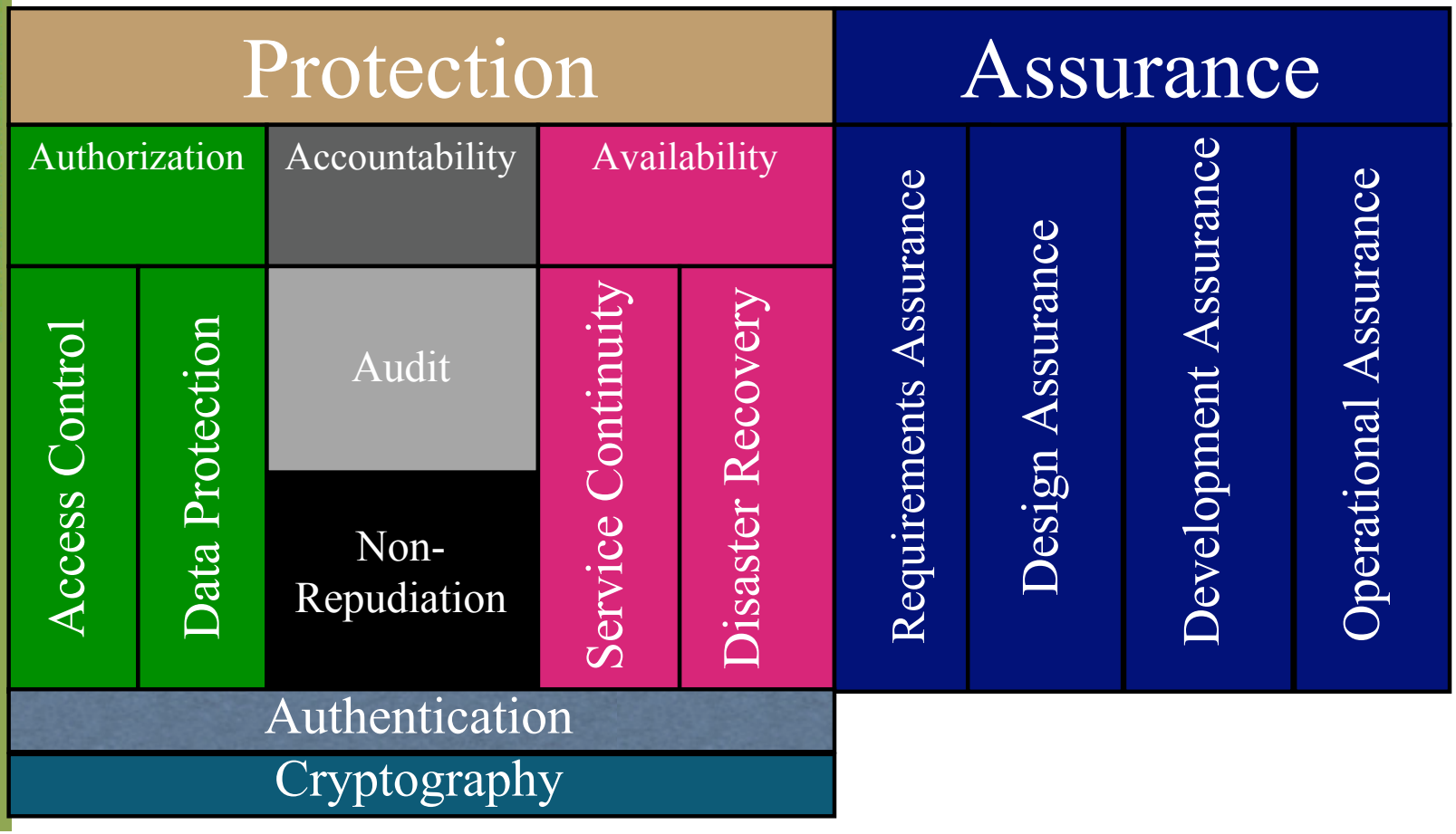
What Computer Security Policies are Concerned with?

- Confidentiality
  - Keeping data and resources hidden
- Integrity
  - Data integrity (integrity)
  - Origin integrity (authentication)
- Availability
  - Enabling access to data and resources

CIA

JAVAPOLIS

Conventional Approach to Security



Protection

- provided by a set of mechanisms (countermeasures) to prevent bad things (threats) from happening

Authorization

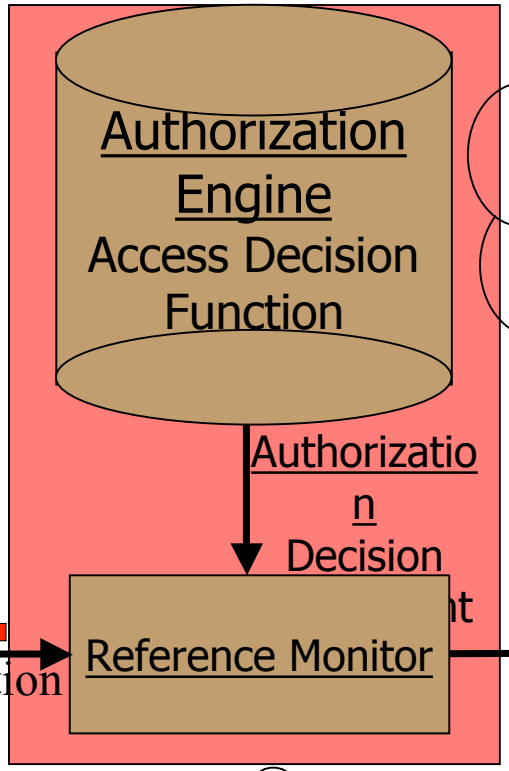
protection against breaking rules

Rule examples:

- Only registered students should be able to take exam or fill out surveys
- Only the bank account owner can debit an account
- Only hospital's medical personnel should have access to the patient's medical records
- Your example…

Access Control

Definition: **enforces the rules, when rule check is possible**



Object
Resource
(data/methods/menu item)
Target

Subject
Principal
User, Client
Initiator

Security Subsystem

Mix of terms:

Authorization == Access Control Decision

Authorization Engine == Policy Engine

Authorization Mechanisms: Data Protection

- No way to check the rules
  - e.g. telephone wire or wireless networks
- No trust to enforce the rules
  - e.g. MS-DOS

Accountability

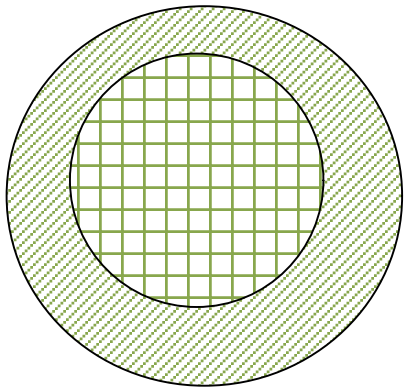
You can tell who did what when

- (security) audit -- actions are recorded in audit log
- Non-Repudiation -- evidence of actions is generated and stored

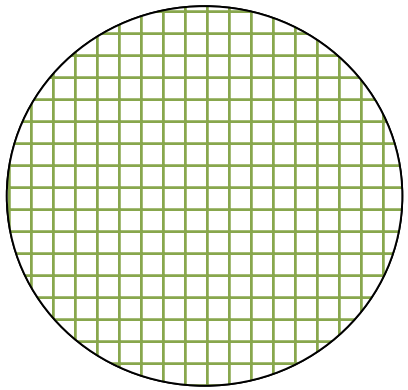
Availability

- Service continuity -- you can always get to your resources
- Disaster recovery -- you can always get back to your work after the interruption

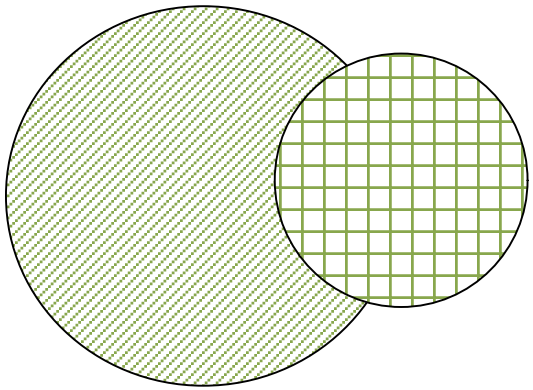
Types of Security Mechanisms



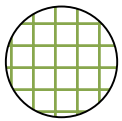
secure



precise



broad



set of reachable states

set of secure states

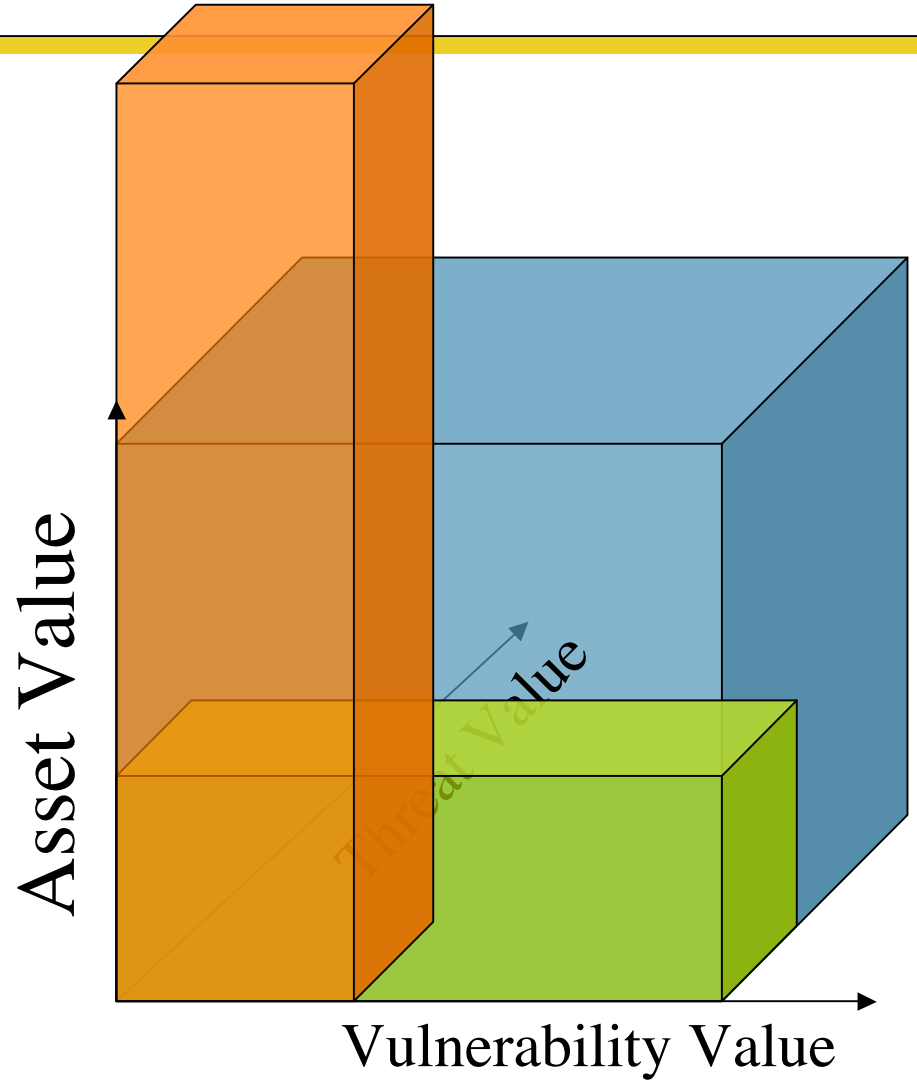
Assurance

Set of things the system builder and the operator of the system do to convince you that it is really safe to use.

- the system can enforce the policy you are interested in, and
- the system works as intended

How do you decide which policies to enforce and mechanisms to use?

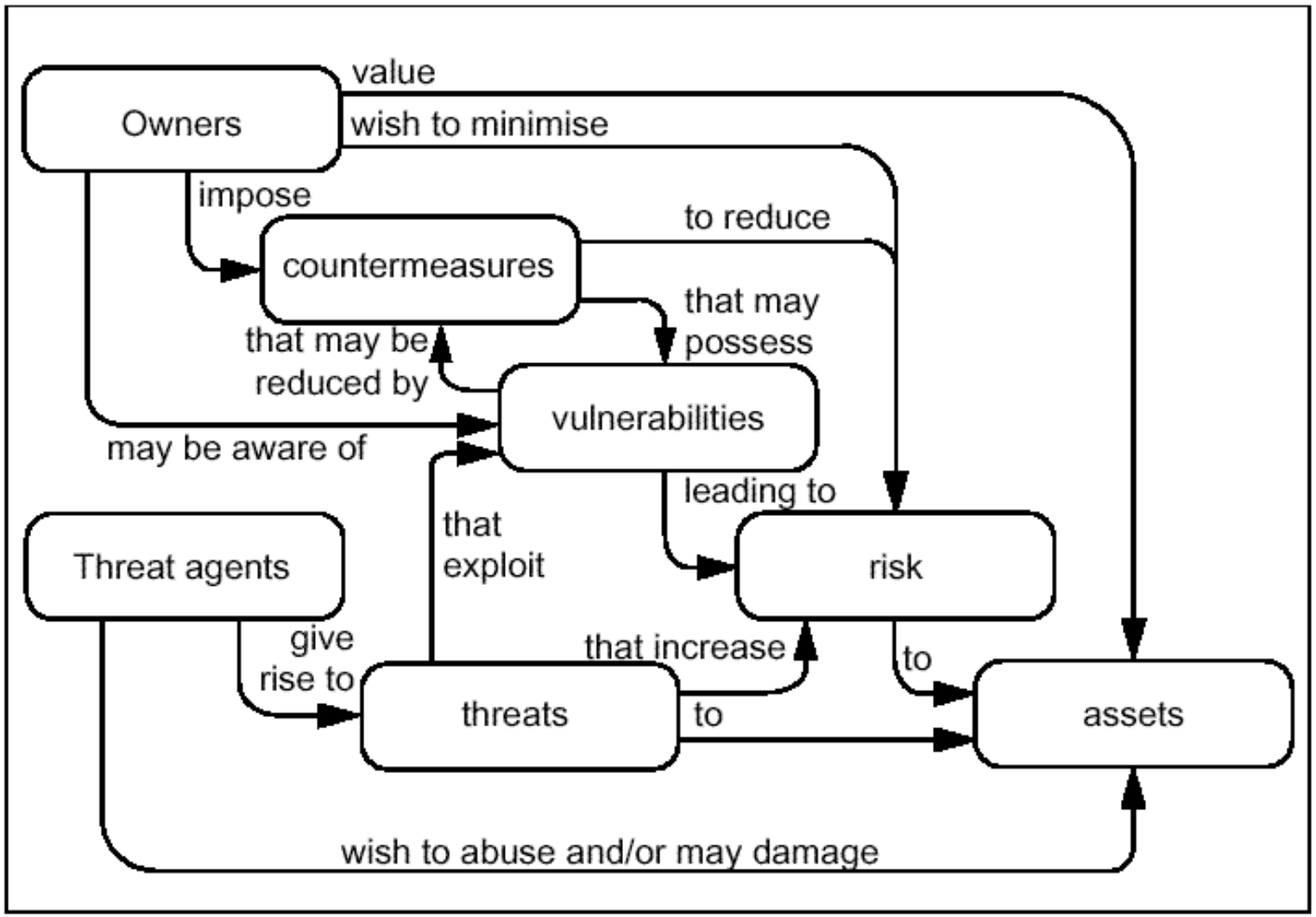
It's all about risk



Risk = Asset * Vulnerability * Threat

JAVAPOLIS

Security is a Process



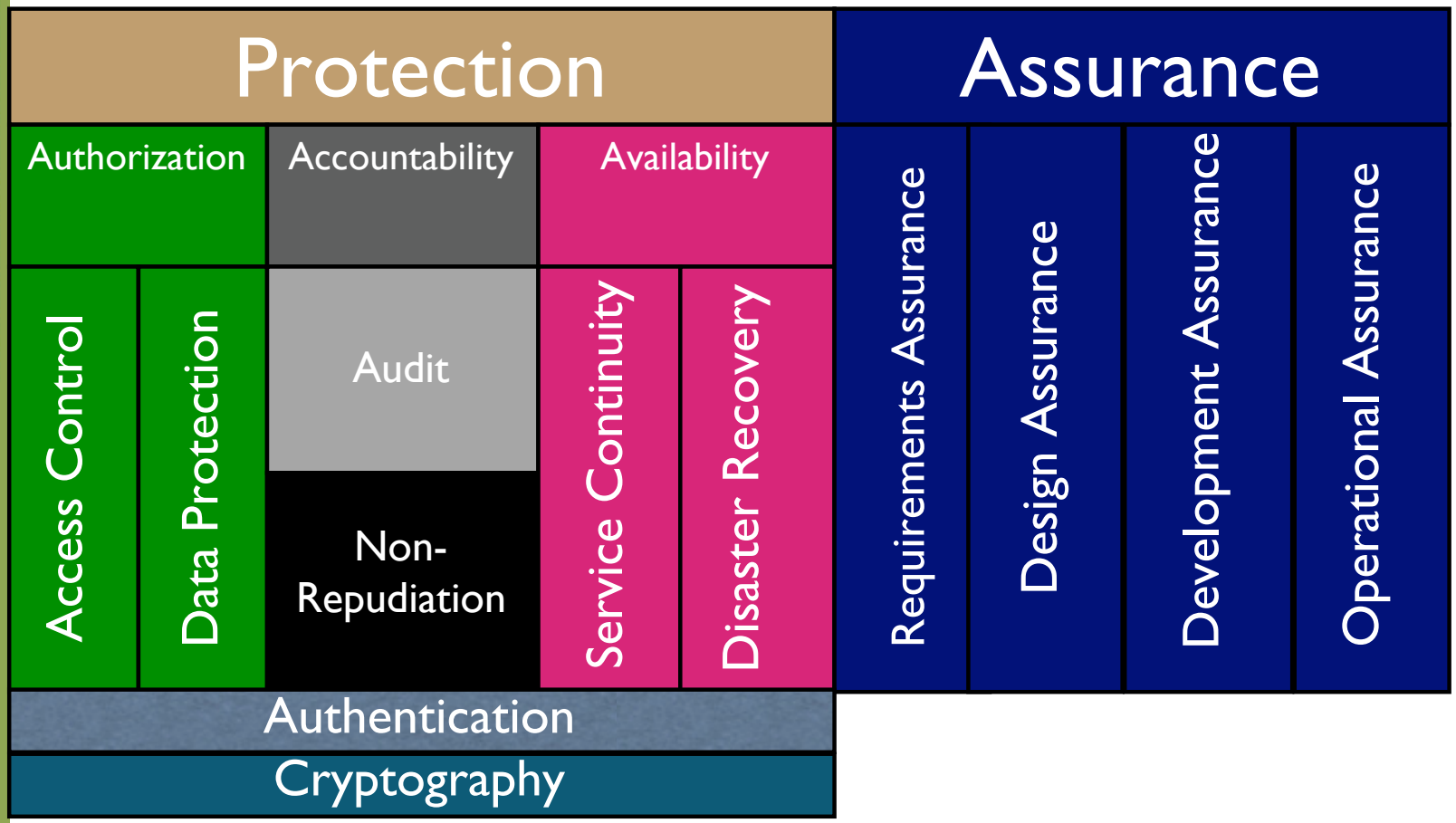
Source: Common Criteria for Information Technology Security Evaluation. 1999

Classes of Threats

- Disclosure
  - Snooping
- Deception
  - *Modification*
  - *Spoofing*
  - repudiation of origin
  - denial of receipt
- Disruption
  - *Modification*
  - denial of service
- Usurpation
  - Modification
  - *Spoofing*
  - Delay
  - denial of service


ONLY
THE BEST
GET IN

Key Points


Protection
Assurance
Authorization
Accountability
Availability
Access Control
Data Protection
Audit
Non-
Repudiation
Service Continuity
Disaster Recovery
Requirements Assurance
Design Assurance
Development Assurance
Operational Assurance
Authentication
Cryptography


JAVAPOLIS

Key Points (cont-ed)

- *Secure*, *precise*, and *broad* mechanisms
- Risk = Asset * Vulnerability * Threat
- Steps of improving security
- Classes of threats
  - Disclosure
  - Deception
  - Disruption
  - Usurpation
- Reference monitor mediates actions of subjects on objects

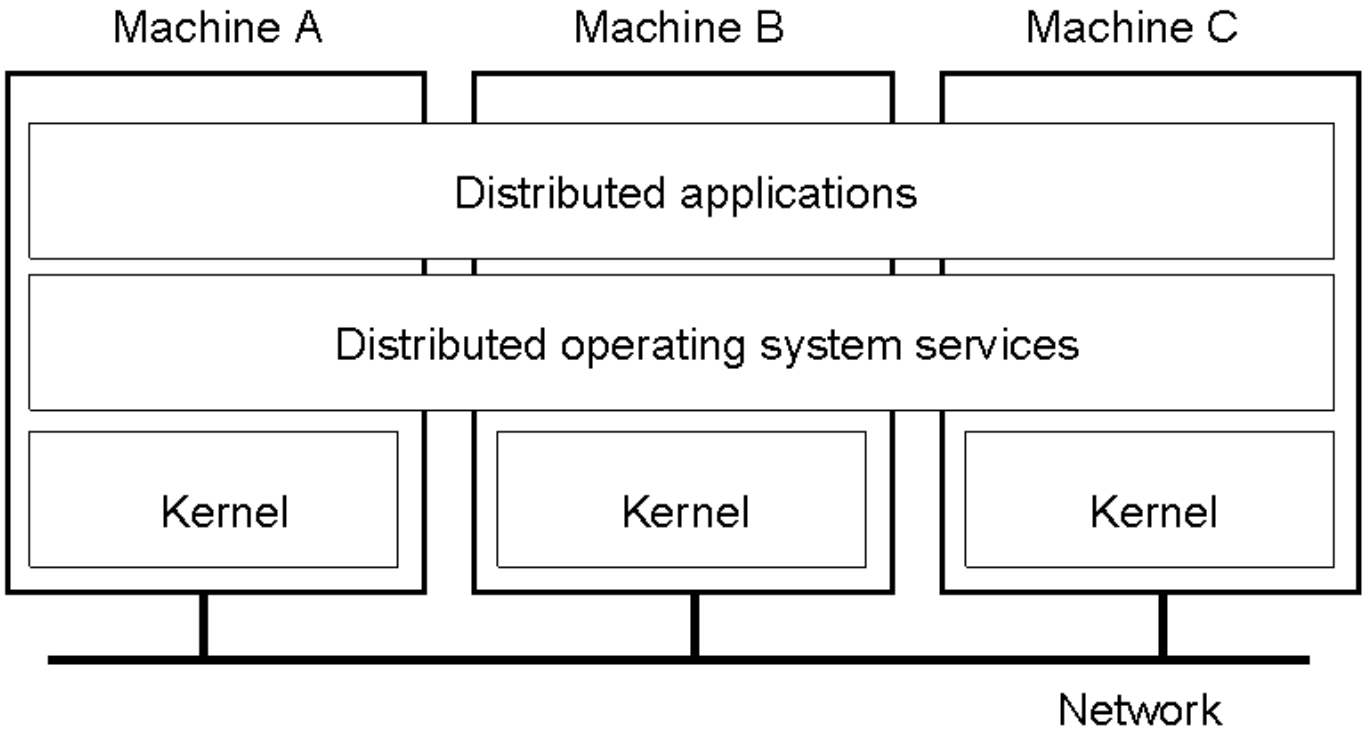
Outline

- Part I: Security
  - What are security mechanisms?
- **Part II: Middleware and Web services**
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- Part III: 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?
- Part IV: Conclusions
  - Summary
  - Where to go from here?

What is middleware?

It's what's between topware and underwear

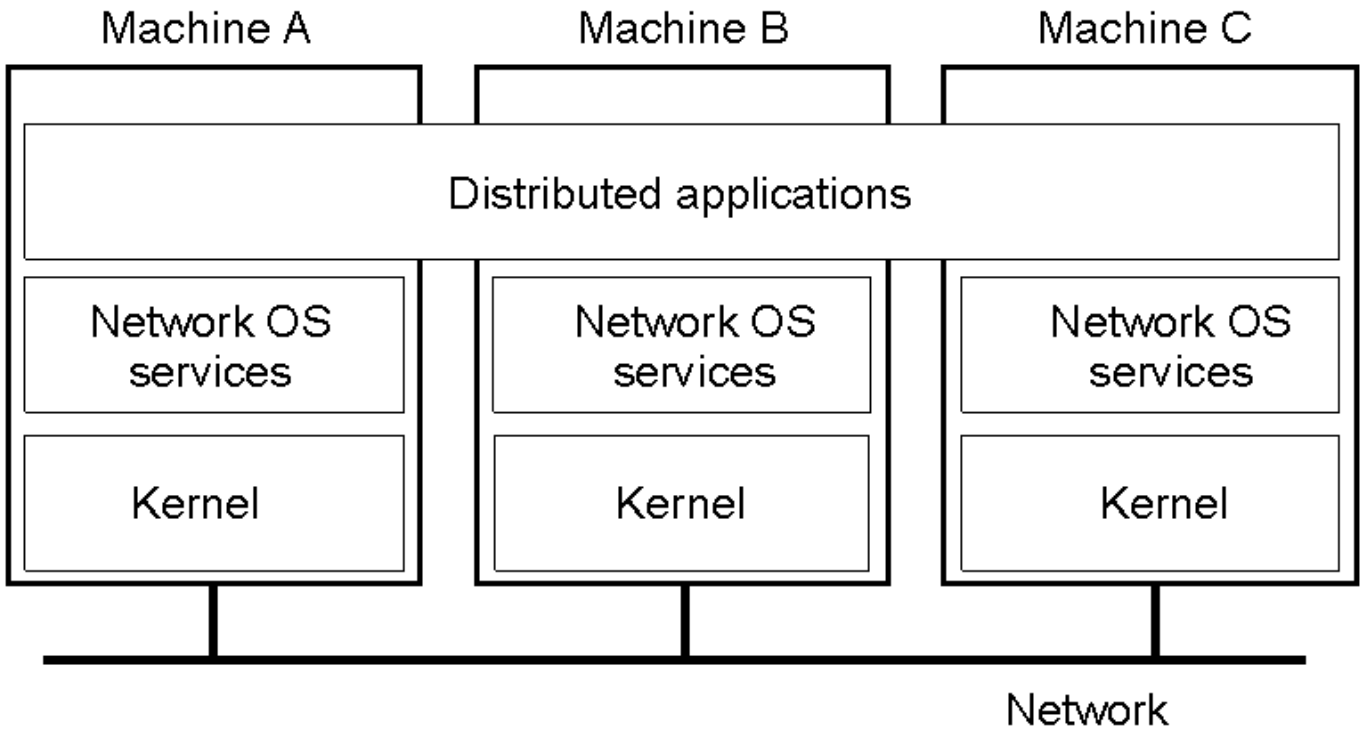
Distributed Application Built Using DOS



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

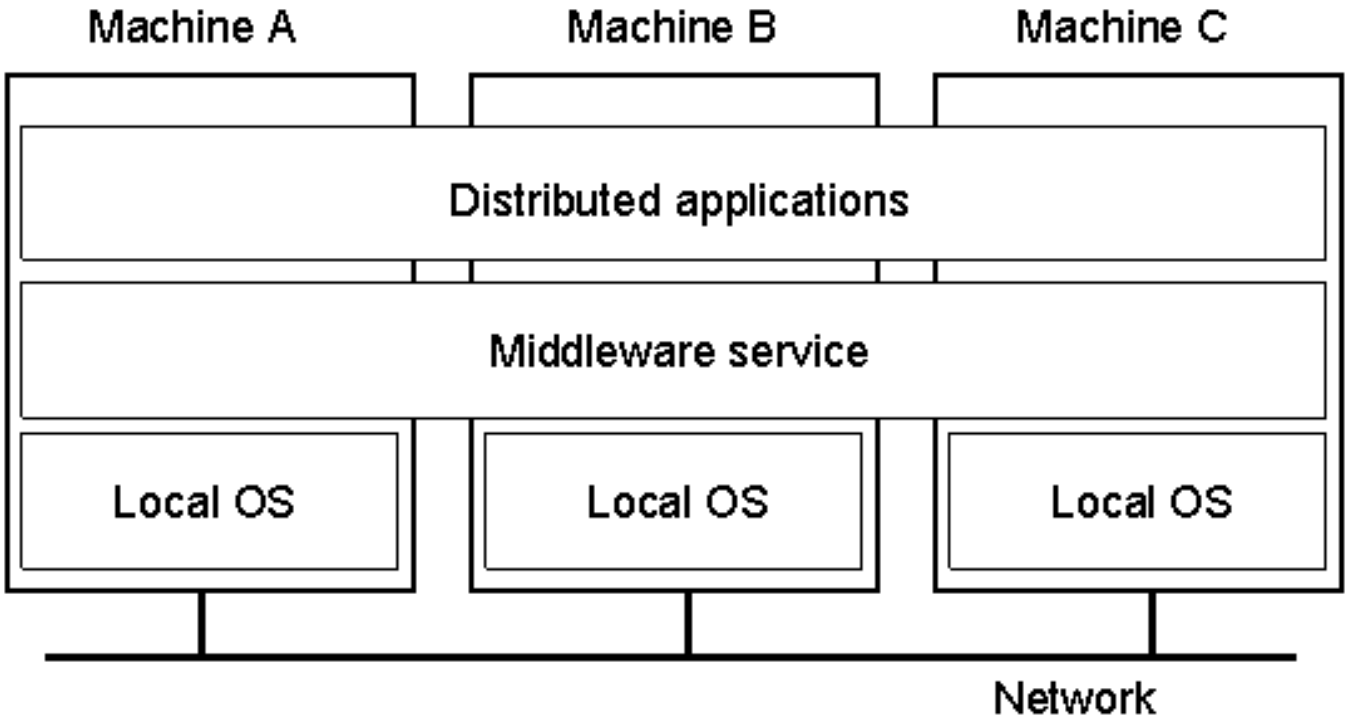
JAVAPOLIS

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



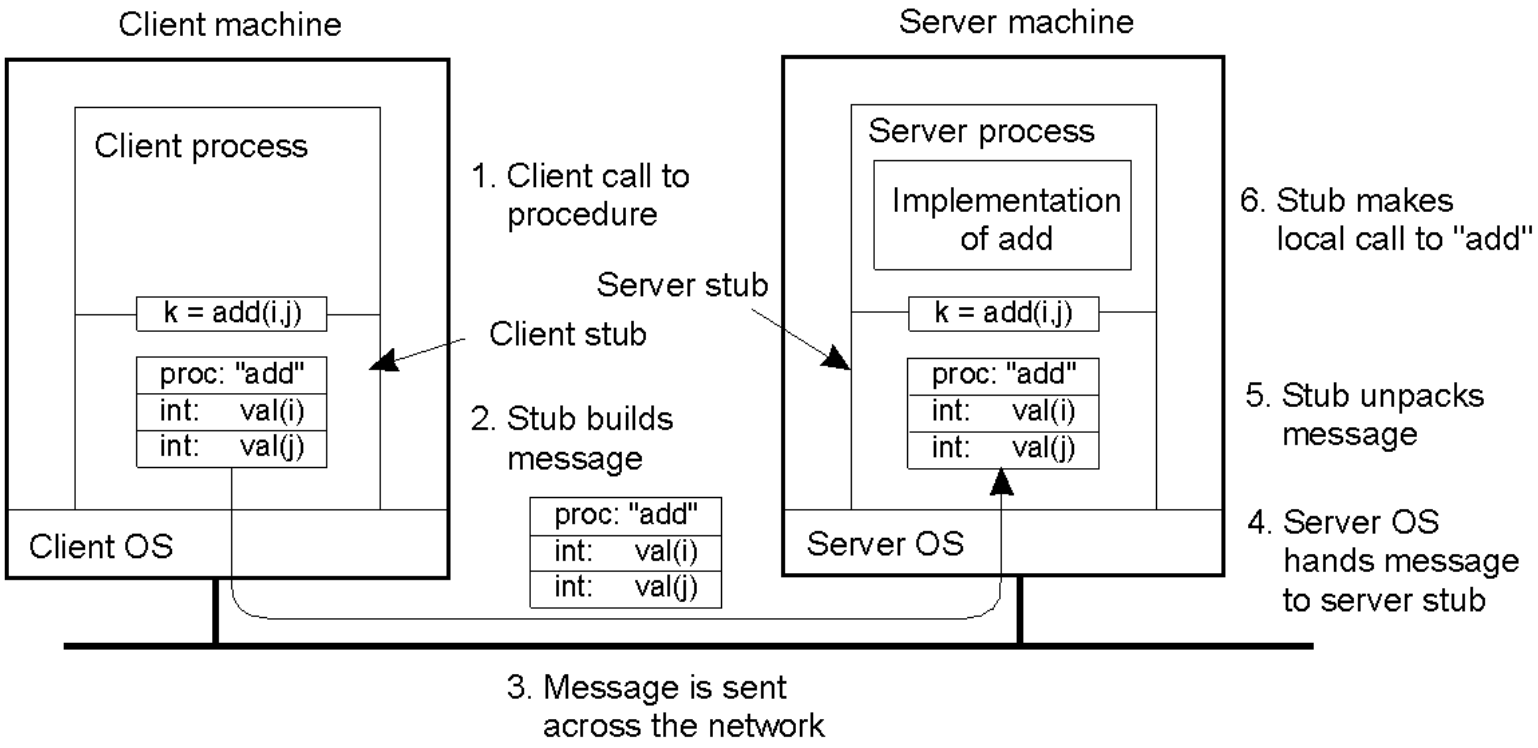
"Distributed Systems: Principles and Paradigms" by A. S. Tanenbaum, M. van Steen. Prentice Hall; (2002)

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**

ONLY
THE BEST
GET IN

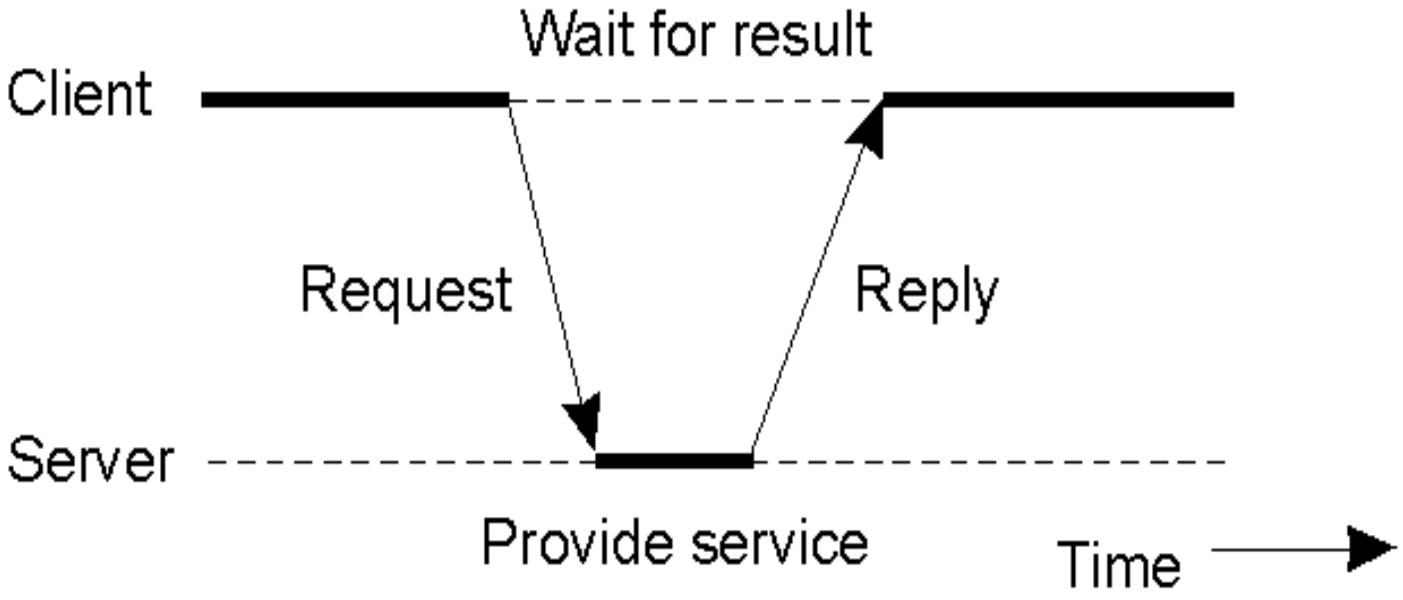
Most Middleware Uses Remote Procedure Call



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

JAVAPOLIS

RPC Clients and Servers



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

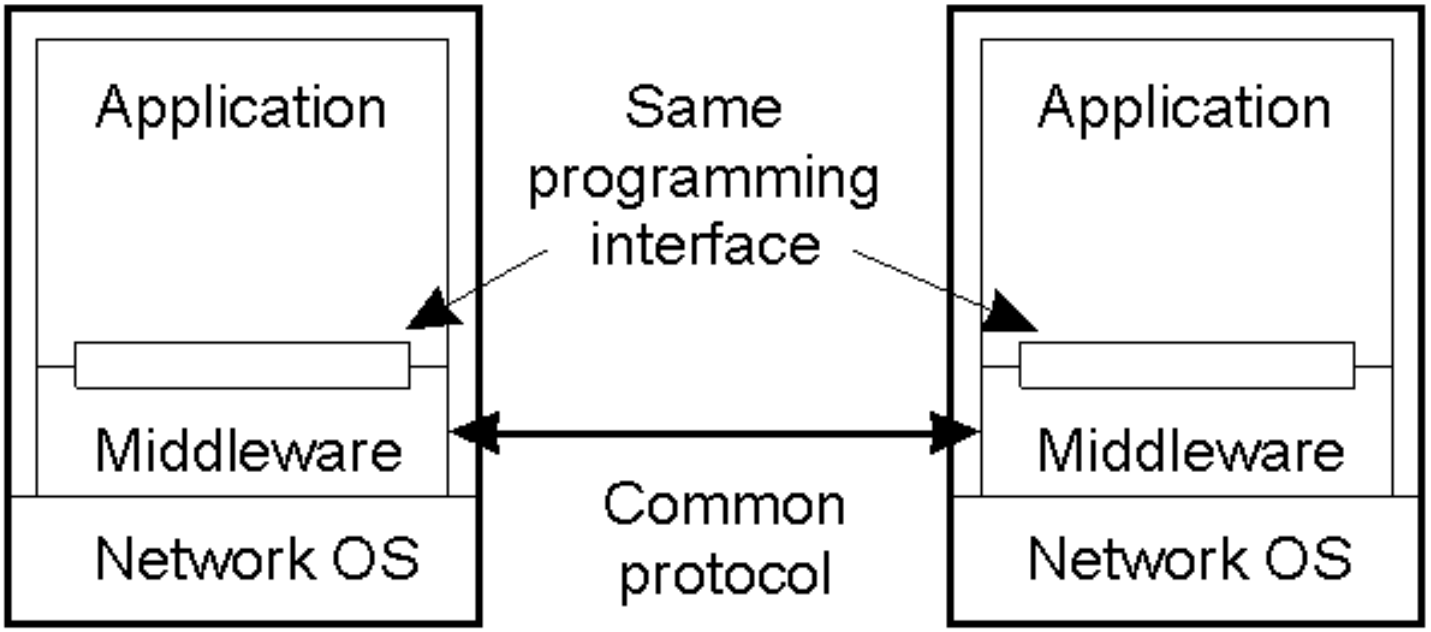
Distributed Objects

- 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
- .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 Middleware Openness?

- Operating system independent
- Completeness and portability
- Interoperability

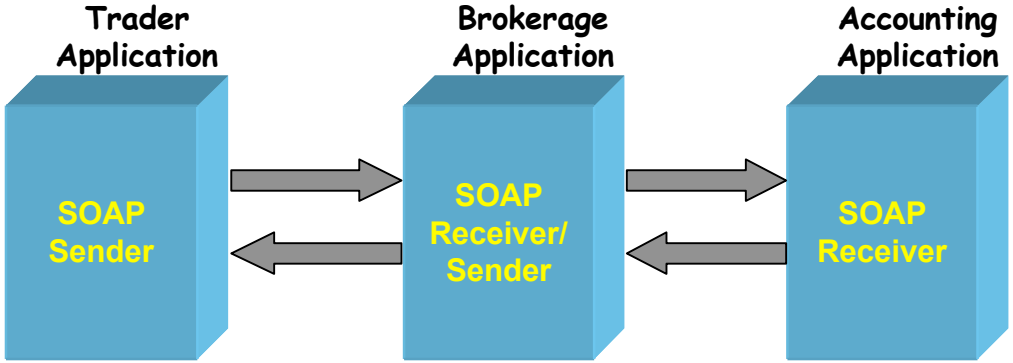
What's Web Services?

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	no
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

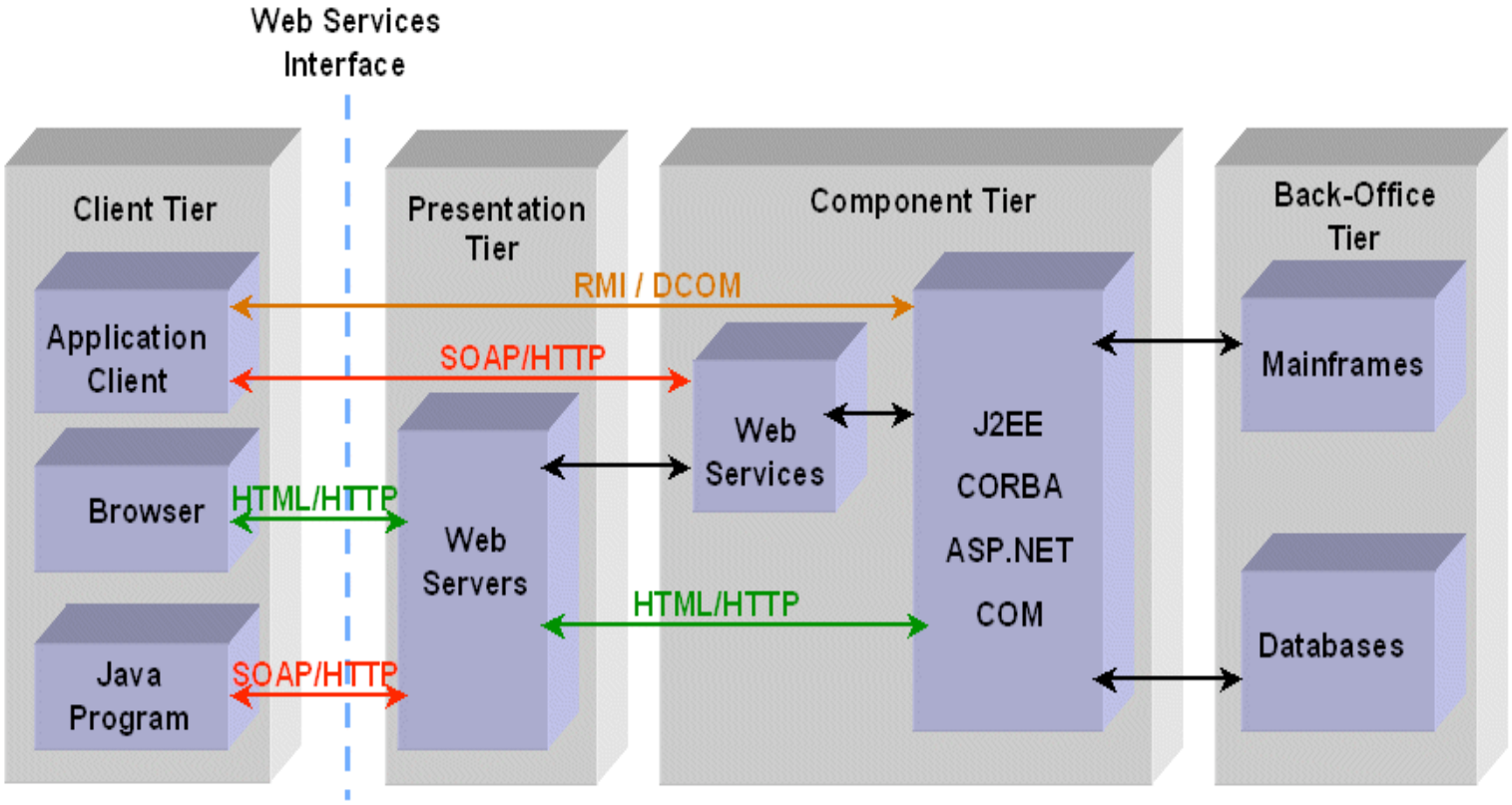
XML-based messaging interface to computing resources that is accessible via Internet standard protocols

- 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

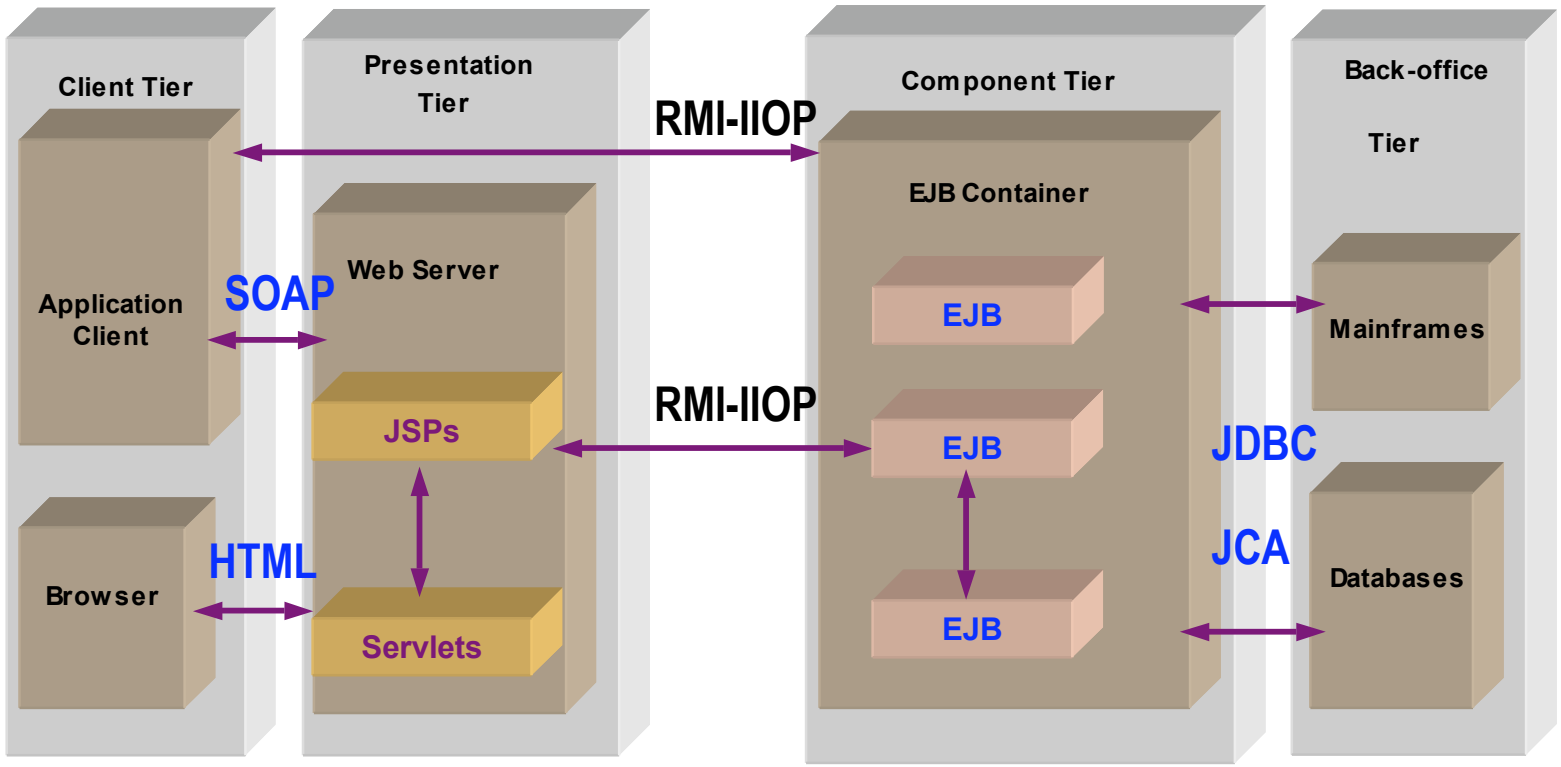
SOAP Message Example

```
<?xml 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

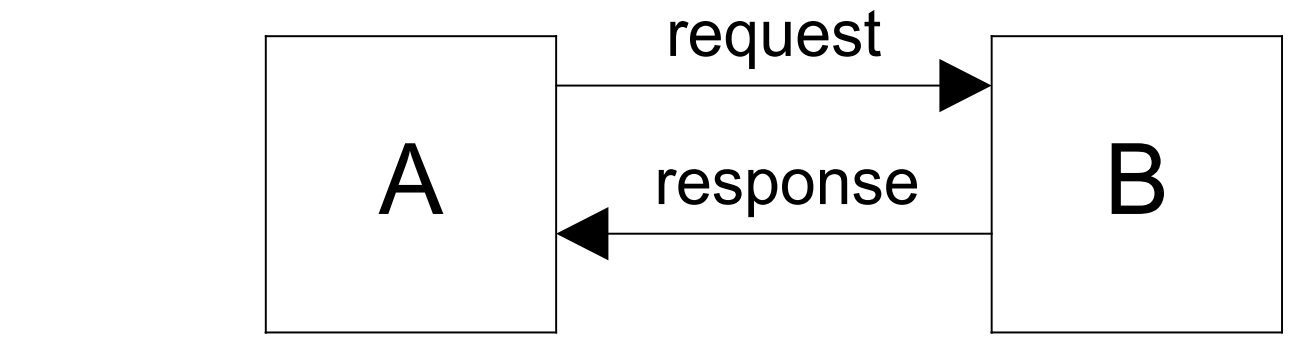


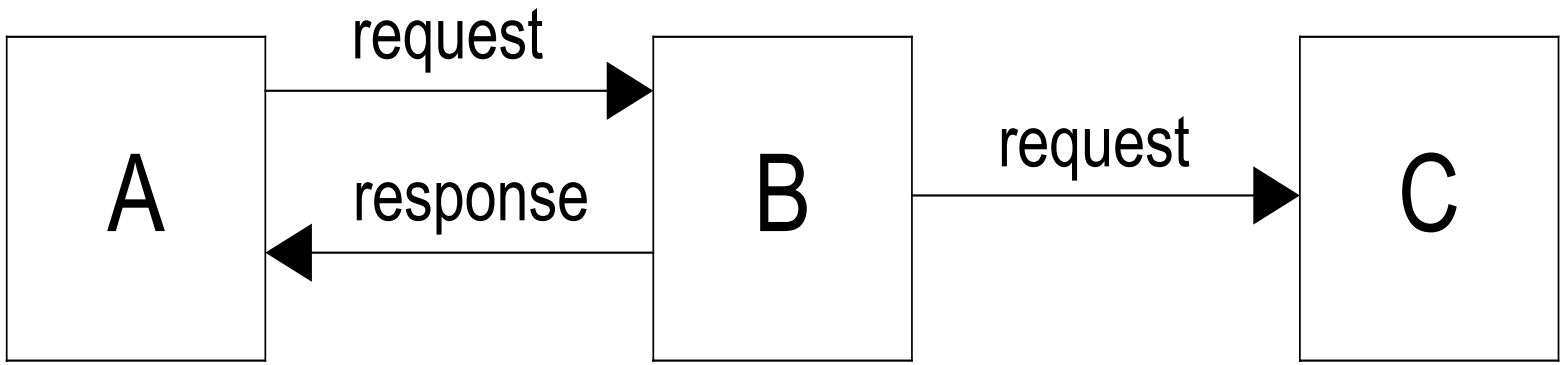
Outline

- Part I: Security
  - What are security mechanisms?
- **Part II: Middleware and Web services**
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- Part III: 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?
- Part IV: Conclusions
  - Summary
  - Where to go from here?

JAVAPOLIS

client-server paradigm & security





requirements due to distribution

- centralized administration
- localized run-time decisions

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)

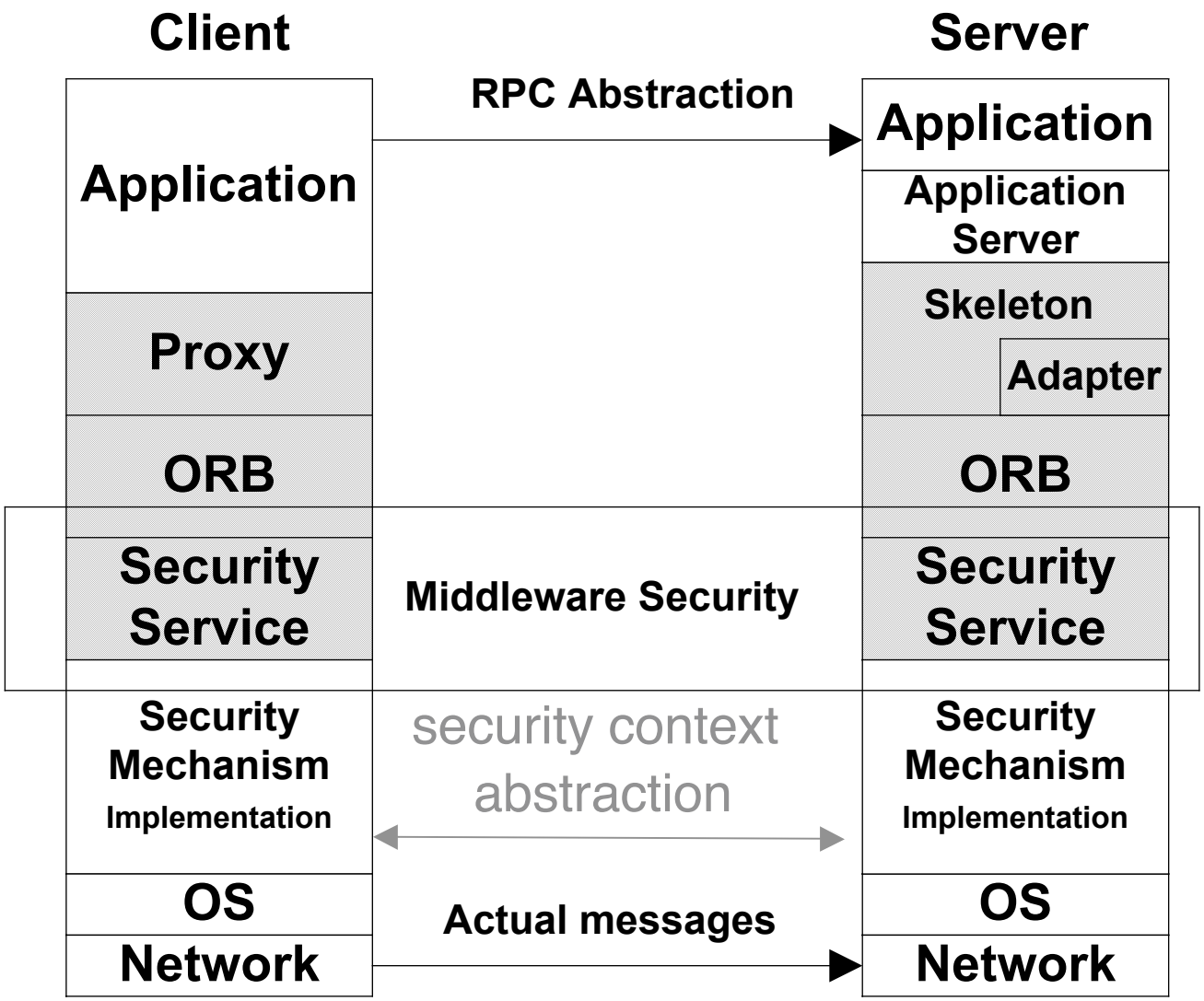
- many layers of indirection and late binding
- names
  - multi-name, nameless and transient objects
  - R: Transient objects should be assigned to security policies without human intervention.
  - less rigid naming hierarchies
  - R: No assumptions that administrators know a name of each object in the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

Outline

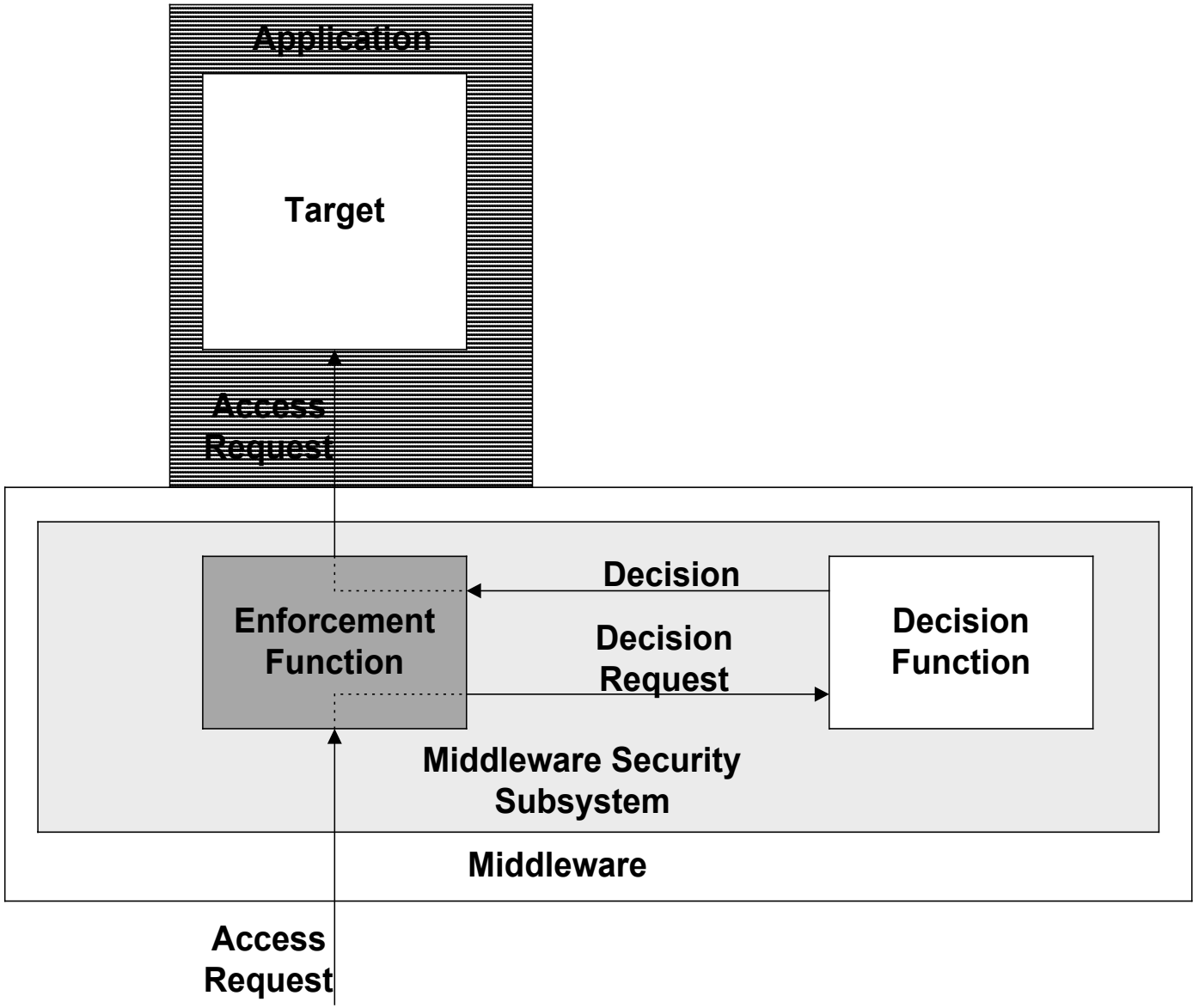
- Part I: Security
  - What are security mechanisms?
- Part II: Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- **Part III: Security in middleware and Web services**
  - What are common architectures for security mechanisms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

JAVAPOLIS

Middleware Security Stack



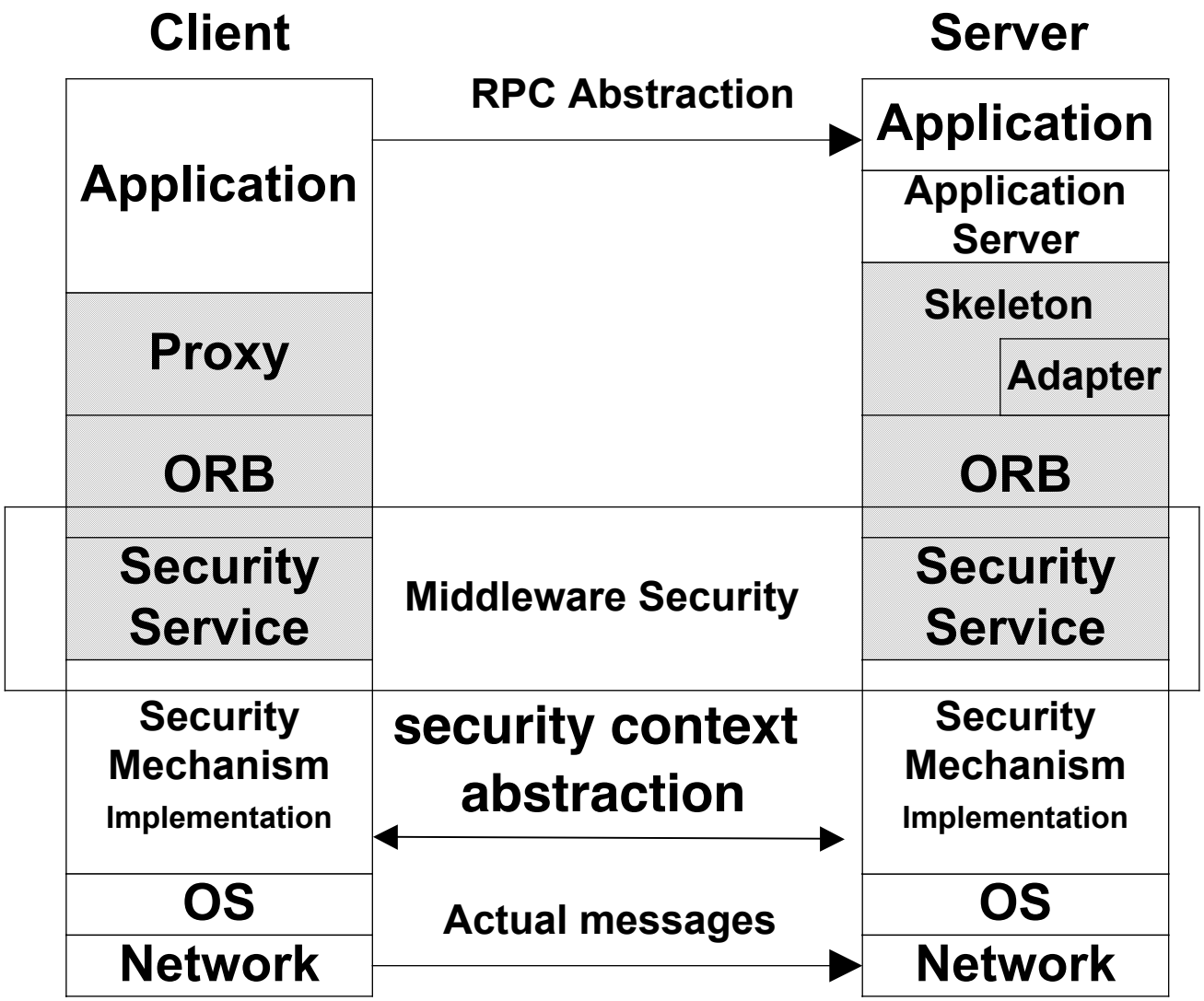
Policy Enforcement and Decision



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>
```

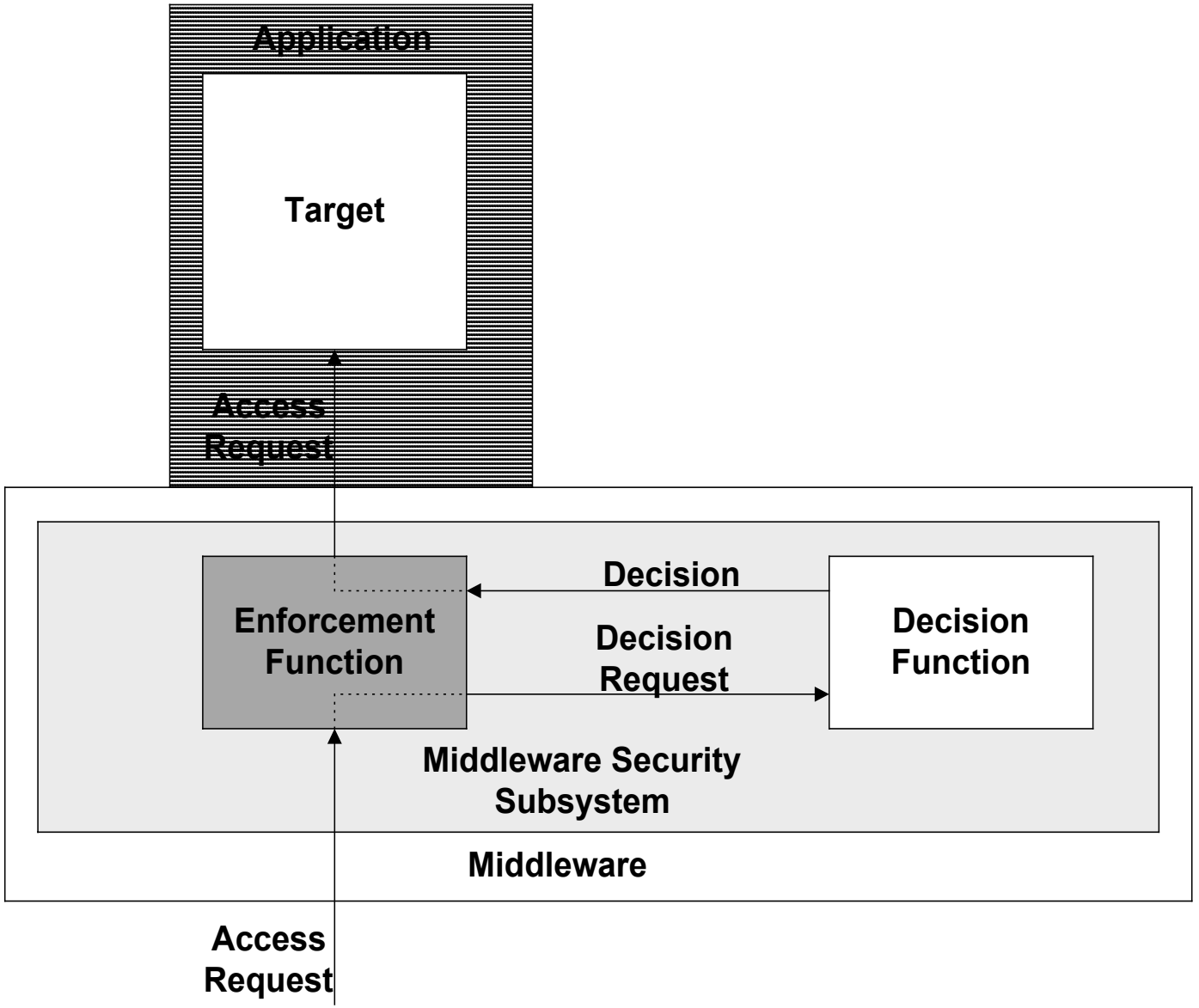
XML Signature

- 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>
```

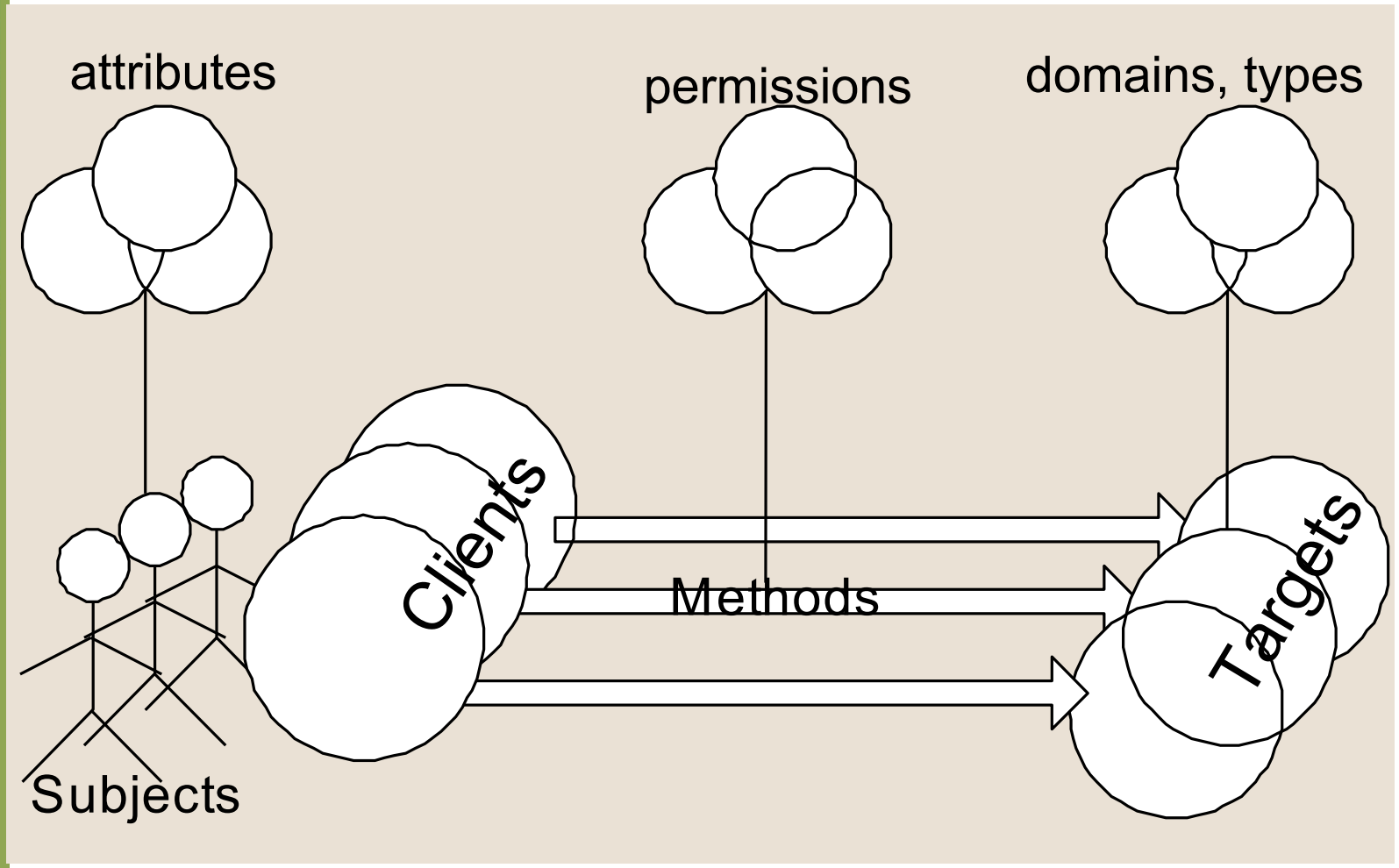
Security Policy Decisions

Policy Enforcement and Decision



ONLY
THE BEST
GET IN

scaling policy decisions

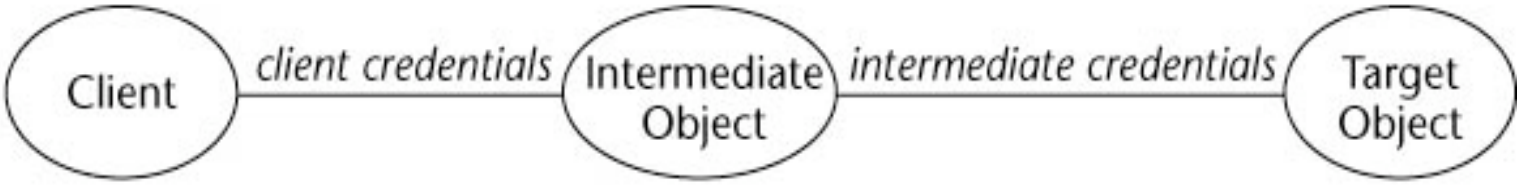


JAVAPOLIS

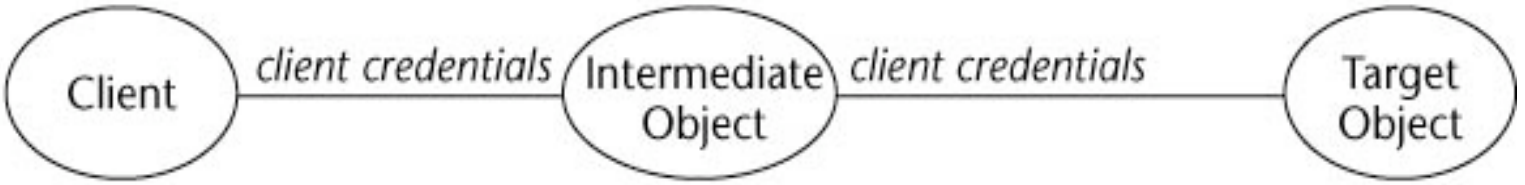
Credentials Delegation

- What are credentials?
- Push and pull models

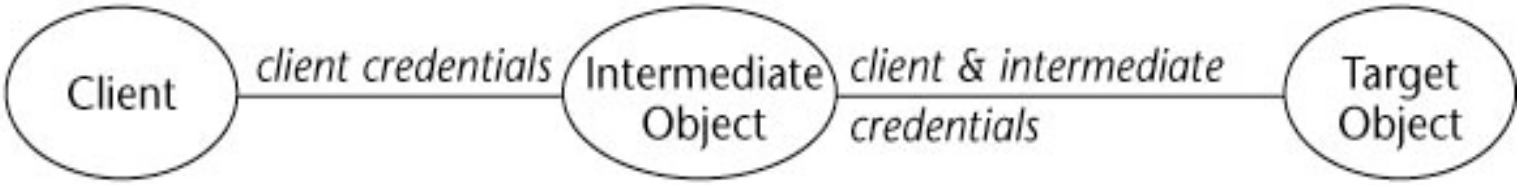
- **No delegation**



- **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

- Part I: Security
  - What are security mechanisms?
- Part II: Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- **Part III: 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?
- Part IV: Conclusions
  - Summary
  - Where to go from here?

COM+ Specifics

Authentication in COM+

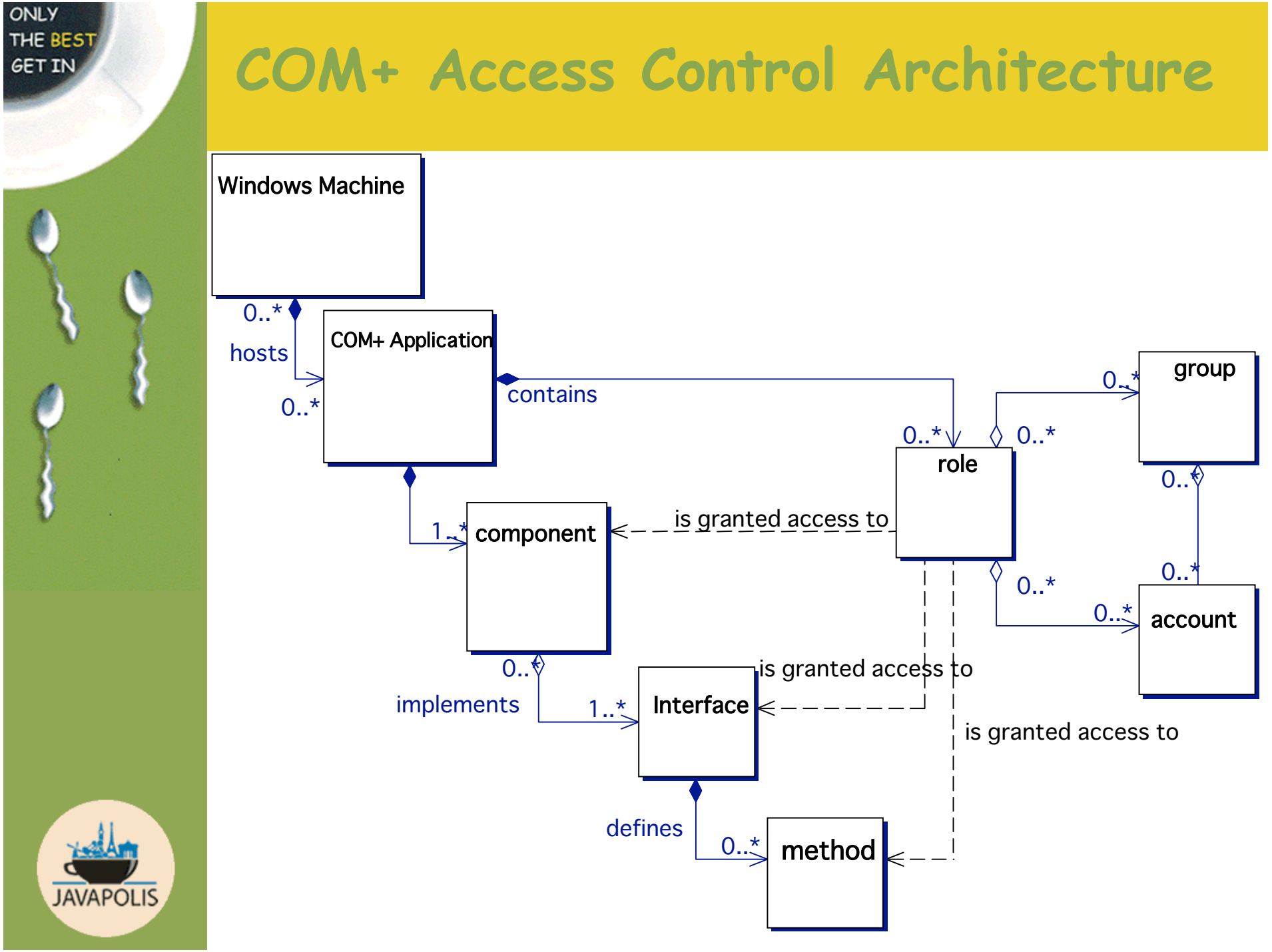
- Supported mechanisms
  - Kerberos
  - Windows NT LAN Manager (NTLM)
- Granularity modes
  - Never
  - At the time of establishing secure channel
  - On every call
  - With every network packet
- Credentials delegation options
  - No delegation
  - Unconstrained simple delegation (a.k.a., impersonation)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

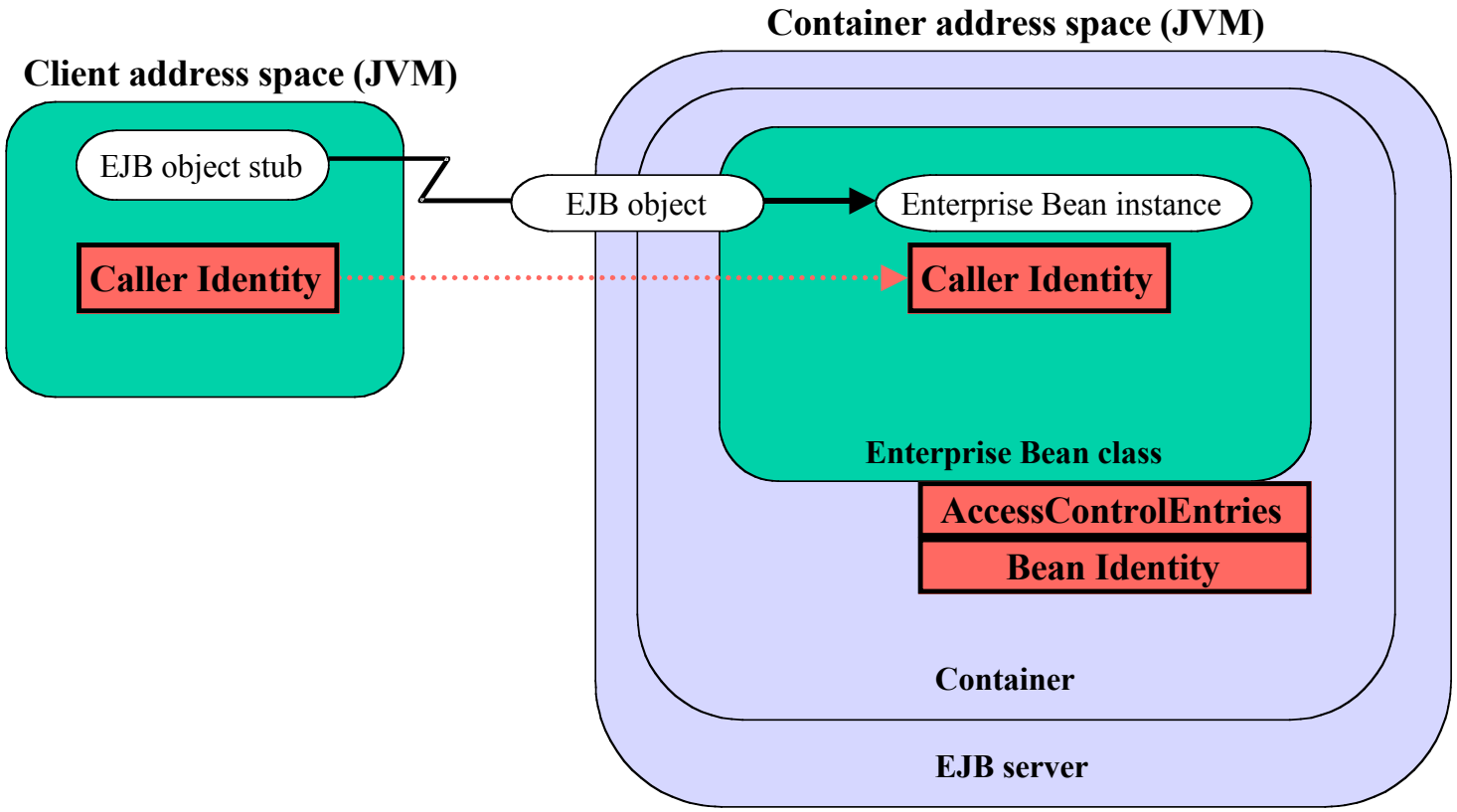

ONLY
THE BEST
GET IN
COM+ Access Control Architecture
Windows Machine
0..*
hosts
0..*
COM+ Application
contains
0..*
role
0..*
group
0..*
0..*
1..*
component
is granted access to
0..*
0..*
account
0..*
0..*
implements
1..*
Interface
is granted access to
is granted access to
defines
0..*
method
JAVAPOLIS

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)

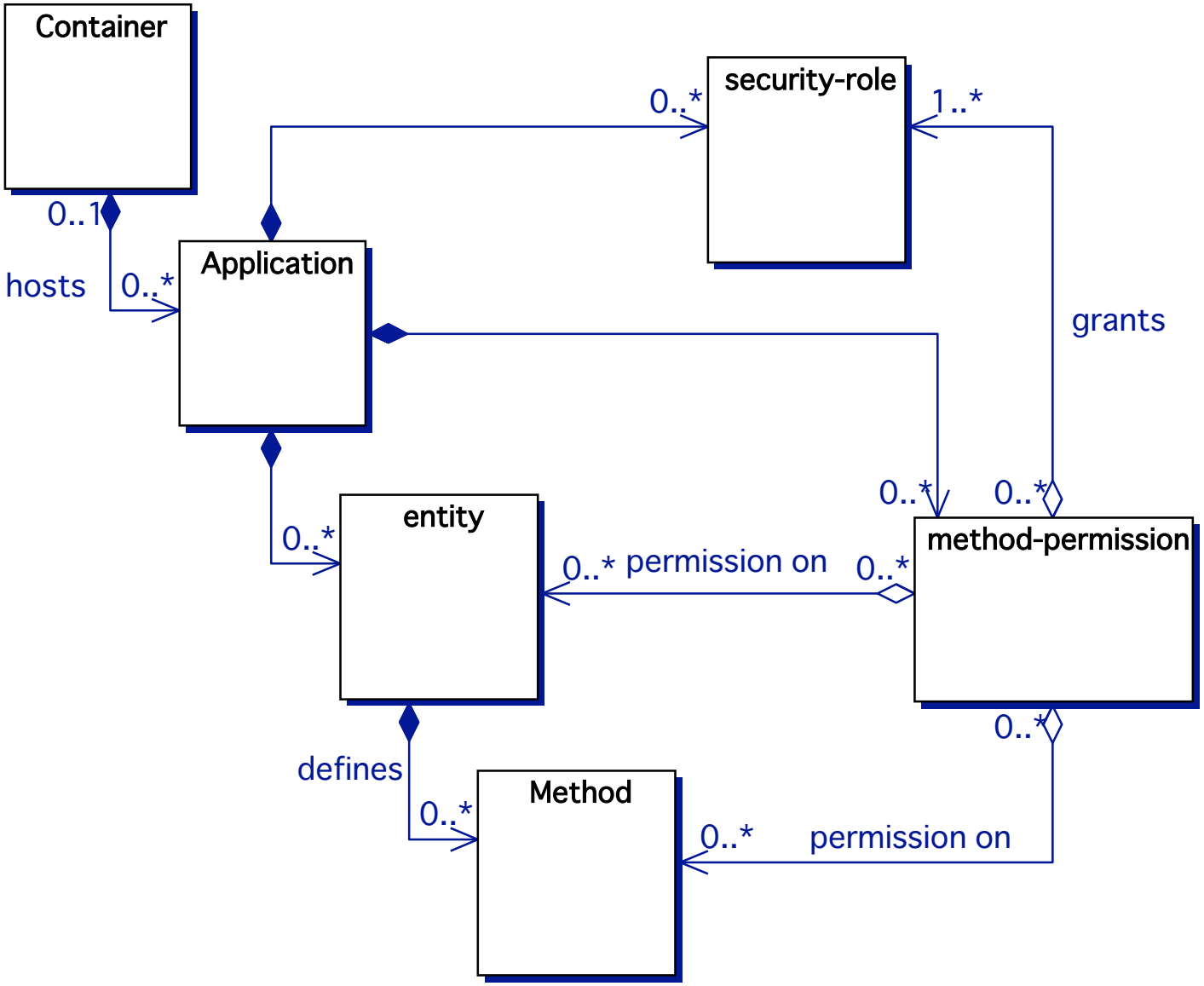
Data Protection in EJB

- 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
- JSR 115: "J2EE Authorization Contract for Containers" -- APIs for plugging authorization engines

roles and permissions in EJB



Accountability in EJB

- Implementation-specific

Outline

- Part I: Security
  - What are security mechanisms?
- Part II: Middleware and Web services
  - What are middleware and Web services?
  - What's special about middleware and Web services security?
- Part III: 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?
- **Part IV: Conclusions**
  - Summary
  - Where to go from here?

Summary

- Security
  - Objectives: CIA
  - Means
    - Protection
      - Authorization, Accountability, Availability
    - Assurance
- 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?

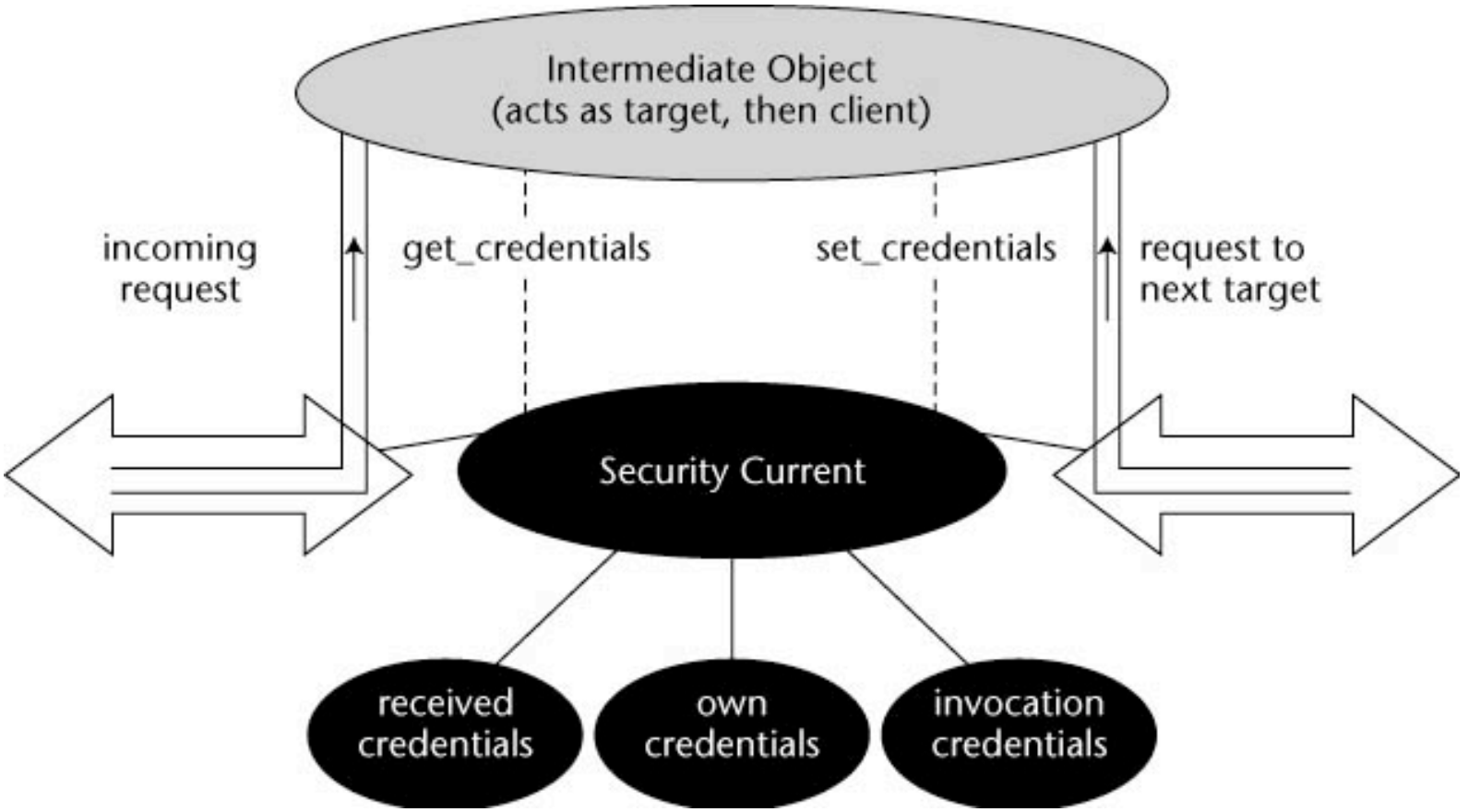
- JavaPolis
  - Access control architectures: EJBs versus COM+
  - Erwin Geirnaert: “Hacking J2EE servers”
  - Secure agility/agile security
- Secure application development course
  - http://www.secure-application-development.com
  - http://www.secappdev.com
- Books
  - B. Hartman, D. J. Flinn, K. Beznosov, and S. Kawamoto, chapter 7, ***Mastering Web Services Security***, New York: 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***. New York: John Wiley & Sons, Inc., 2001.
  - “Security Engineering …” by Ross Anderson

If You Only Remember Three Things…

- Build security in from the beginning
- Push security out of the applications
- Design for change

JAVAPOLIS

Reserved slides


Intermediate Object
(acts as target, then client)
incoming
request
get_credentials
set_credentials
request to
next target
Security Current
received
credentials
own
credentials
invocation
credentials