

IBM



Component-Based Software-Engineering
6. Vorlesung

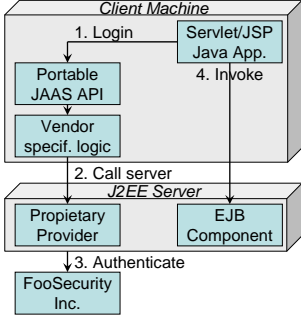
Thomas Gschwind <thg@zurich....>

| SS2007 © 2007 IBM Corporation

IBM

EJB Security

- Java Authentication & Authorization Service (JAAS)
 - Who am I?
 - What may I?
- For Session and Entity beans



3 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM

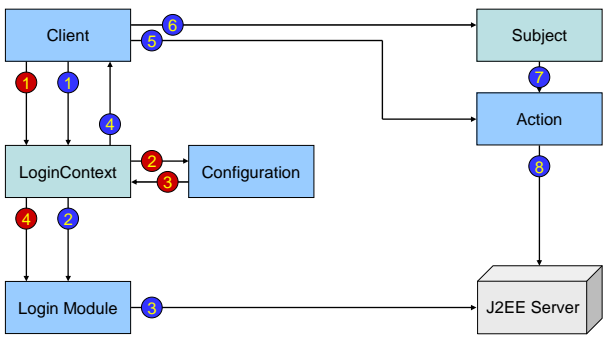
Agenda

- EJB Security
- EJB and Web Services
- EJB Persistence Best Practices

2 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM

Authentication



4 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

Authorization – Client Side

```
public class CookieClient {
    private static LoginContext loginContext = null;
    public static void login(String u, String p)
    throws LoginException {
        loginContext=new LoginContext("stock",
            new UsernamePasswordHandler(u, p.toCharArray()));
        loginContext.login();
    }
    public static void main(String[] args) throws Exception {
        Context ctx=new InitialContext(System.getProperties());
        Object obj=ctx.lookup("CookieServer");
        CookieServerHome serverHome=(CookieServerHome)
            PortableRemoteObject.narrow(obj, CookieServerHome.class);
        login(args[0],args[1]);
        CookieServer server=serverHome.create();
        loginContext.logout();
    }
}
```

Programmatic Authorization

- EJB context allows programmer to get authentication information
- Authorization may be performed on
 - User information
 - Roles
- Security roles defined in deployment descriptor

```
public interface javax.ejb.EJBContext {
    ...
    public java.security.Principal getCallerPrincipal();
    public boolean isCallerInRole(String role);
    ...
}
```

Authorization – Server Side

- Programmatic Authorization
 - Beans interlaced with security logic
 - More powerful
- Declarative Authorization
 - Security logic defined in deployment descriptor
 - May be changed more easily

Programmatic Authorization (Sample)

```
public class CookieBean implements EntityBean {
    ...
    public void setCookie(String text) {
        String name=ctx.getCallerIdentity().getName();
        if(ctx.isCallerInRole("administrators") ||
            name.equals(cookieOwner)) {
            // change text of cookie
        }
    }
    ...
}
```

Declarative Authorization

- Permissions specified in deployment descriptor

```
<ejb-jar>
  <enterprise-beans>
    ...
  </enterprise-beans>
  <assembly-descriptor>
    <method-permission>
      <role-name>administrators</role-name>
      <method>
        <ejb-name>Cookie</ejb-name>
        <method-name>*</method-name>
      </method>
    </method-permissions>
  </assembly-descriptor>
</ejb-jar>
```

9

Security Propagation

- How are security credentials passed between beans?
- Client credentials? Other credentials?
- Defined in Deployment Descriptor

```
<entity>
  <ejb-name>Cookie</ejb-name>
  ...
  <security-identity>
    <use-caller-identity>
  </security-identity>
  ...
  ...
  <security-identity>
    <run-as>
      <role-name>admins
    </role-name>
    </run-as>
  </security-identity>
  ...
</entity>
```

11

Declarative Authorization (cont'd)

```
<method-permission>
  <role-name>guests</role-name>
  <method>
    <ejb-name>CookieServer</ejb-name>
    <method-name>getCookie</method-name>
    <method-params>int</method-params>
  </method>
</method-permissions>
<exclude-list>
  <method>
    ...
  </method>
</exclude-list>
</assembly-descriptor>
</ejb-jar>
```

10

Agenda

- EJB Security
- EJB and Web Services
- EJB Persistence Best Practices

12

IBM Research, Zurich

Why Web Services?

- Not just another RPC mechanism
- Good if multiple middleware platforms used
- Why good?
 - Only need to build an adapter from "foo" to SOAP
 - Use standard protocols
 - Loose coupling between client & servers
- Really so great?
 - Technically, probably not
 - Politically, yes as long as everybody adheres to it

13 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

WSDL

```

<types>
  <message name="HelloInterface_hello" />
  <message name="HelloInterface_helloResponse">
    <part name="result" type="xsd:string" />
  </message>
  <portType name="HelloInterface">
    <operation name="hello">
      <input message="tns:HelloInterface_hello" />
      <output message="tns:HelloInterface_helloResponse" />
    </operation>
  </portType>
  <binding name="HelloInterfaceBinding" type="tns:HelloInterface">
    <soap:binding transport="http:..." style="rpc" />
    <operation name="hello">
      <soap:operation soapAction="..." />
      <input>
        ...
      </input>
    </operation>
  </binding>
</types>

```

15 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Web Services

- Web Service support introduced in EJB2.1
- Build and integrate large-scale systems (Service Oriented Architectures)

```

graph TD
    SR[Service Registry]
    SRQ[Service Requestor]
    SP[Service Provider]
    SRQ -- Find --> SR
    SP -- Publish --> SR
    SRQ -- Bind SOAP --> SP
    subgraph WSDL_UDDI
        SR
        SRQ
        SP
    end

```

14 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

SOAP

- Uses HTTP for transport
- Not high performance, only interoperability

```

POST /HelloBean HTTP/1.1
Content-Type: text/xml; charset="utf-8"
Content-Length: 398
SOAPAction: ""
Host: navelli:8080

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlns.org/soap/envelope">
  <env:Body>
    <ans1:hello xmlns:ans1="urn:examples" />
  </env:Body>
</env:Envelope>

```

16 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

Implementing a Web Service

- J2EE provides seamless web service integration
- Web Service managed by EJB Container
- Create port components as session beans (can reuse existing session bean)
- Create another Java interface (The service endpoint interface)
- Specify the web-services deployment descriptor

web-services.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<webservices xmlns="http://..." version="1.1"
  xmlns:xsi="http://..."
  xsi:schemaLocation="http://... http://...">
  <webservice-description>
    <display-name>CookieServerWS</display-name>
    <webservice-description-name>CookieServerWS</webservice-...>
    <wsdl-file>META-INF/wsdl/CookieWS.wsdl</wsdl-file>
    <jaxrpc-mapping-file>META-INF/wsdl/mapping.xml</jaxrpc-...>
    <port-component>
      <display-name>CookieWS</display-name>
      <port-component-name>CookieWS</port-...>
      <wsdl-port xmlns:wsdl-port_ns_="urn:examples">
        wsdl-port_ns_:CookiePort</wsdl-port>
      <service-endpoint-interface>examples.HelloInterface</...>
      <service-impl-bean>ejb-link>CookieServerBean</...></...>
    </port-component></...></...>
```

Service Endpoint Interface

```
import java.rmi.*;

/** The cookie service endpoint interface. */
public interface CookieInterface extends java.rmi.Remote {
    public String getCookie() throws RemoteException;
}
```

Characteristics

- Follow RMI Interface conventions
- No constants
- Use "standard" types or provide type-mapper for "non-standard" ones

The Web Service Client

- Can invoke web service using
 - Static Stubs
 - Dynamic Proxy

The Static Web Service Client

```
public class CookieStaticClient {
    public static void main(String[] args)
        throws Exception {
        URL url=new URL(args[0]);
        QName qname=new QName("http://cbse.unizh.ch/",
            "CookieWebService");

        ServiceFactory factory=ServiceFactory.newInstance();
        Service service=factory.createService(url, qname);
        CookieInterface cookie=(CookieInterface)
            service.getPort(CookieInterface.class);

        System.out.println(cookie.getCookie());
    }
}
```

Word of Caution

- There is not one way to do something right
- This lecture cannot tell you what to do
- This lecture only tells you what to look for

- BUT
 - Many ways exist to do something wrong!
 - What are the advantages of each approach?
 - What are your requirements?
 - Think!

Agenda

- EJB Security
- EJB and Web Services
- EJB Persistence Best Practices
 - Which Type of Bean
 - CMP versus BMP
 - Tips and Tricks

Which Type of Bean

- BMP/CMP entity bean
- Session bean plus JDBC
- Message driven beans

IBM Research, Zurich

Bean Type: Performance

- Entity beans return remote reference
 - One request per required attribute
 - One transaction per attribute
- Session bean returns data using call-by-value
 - Only one request to obtain data
 - Only one transaction

25 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Bean Type: Database Independence

- Entity beans encapsulate database representation
- Developer isolated from schema
- Session bean can also provide this if implemented manually

27 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Bean Type: Caching

- Session beans do not represent data
=> cannot be cached
- Entity beans may be cached across transactions
=> efficient
- Usefulness of caching depends on your data!

26 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Which Type of Bean

- All the advantages can be combined
 - Use session bean for client interaction
 - Session bean uses entity bean (through local interface!)
- Local Interfaces were added in EJB2.0

28 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

RMI-IIOP versus Messaging

- Tradeoffs
 - Performance (async. invocation)
 - Reliability
 - Support multiple senders & receivers
- JMS Advantages
 - Load balancing
 - Request prioritization
 - Good for loosely coupled systems (e.g., WAN)
- RMI-IIOP Advantages
 - Good if program has to be notified when request is done
 - Return result
 - Can participate in client-transactions
 - Security
 - Error detection (Strong typing)

29 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

CMP versus BMP: Performance

- CMP
 - Container manages find methods
 - One huge SQL statement to retrieve entities
- BMP
 - Finder methods return list of primary keys
 - One SQL per primary key to retrieve entities

31 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

CMP versus BMP

- CMP
 - Application may be developed more quickly
 - Persistence format declared externally
- BMP
 - More flexible

30 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

CMP versus BMP: Bugs

- CMP
 - May be hard to find
 - Error in deployment descriptor may manifest in container generated code (Good luck!)
 - Use various log files (application server, database)
- BMP
 - All done by yourself (easier to debug)

32 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

CMP versus BMP: Relationships

- **CMP**
 - Easy to manage
- **BMP**
 - Good luck of taking care of cascaded deletes etc.

33 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

CMP versus BMP

- **Using BMP beans?**
Consider putting db logic into stored procedures
 - Advantage SQL-EJB mapping maintained outside of bean
 - Shared database logic
 - Manage security
 - Legacy database encapsulation
- **Tradeoffs**
 - Performance (may be slower/faster)
- **Disadvantages**
 - Stored procedures not standardized
 - Two separate places for application logic

35 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

CMP versus BMP

- **Start out with a CMP entity bean**
 - After all easier to implement
 - If you have a good application servers, many errors in deployment descriptor are checked
 - May always be converted to BMP bean

34 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Tips and Tricks

- **Try to match your database and object model**
 - Easy when system developed from scratch
 - Harder if legacy system is extended
- **Normalized versus denormalized database**
 - Denormalized may be db more performant
 - Harder to understand
 - Use as last resort!

36 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Tips and Tricks (cont'd)

- Keys
 - Using natural key simplifies key managed
 - Problems if natural key changes
 - Use surrogate key instead
 - Caution: Introducing an artificial can yield redundancies, use database constraints

37 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Tips and Tricks (cont'd)

Large Result Sets

- Add more constraints to result set
- Set maximum number of rows to retrieve
- Use JDBC 2.0
- Use session bean that returns data in chunks to clients

39 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Tips and Tricks (cont'd)

Living with Legacy Databases (Good Luck!)

- Single column used for different purposes
 - Map to multiple EJB attributes
 - Use legacy design
- Purpose of column is defined by other value
 - Probably use BMP
- Incorrect data values
 - CMP that verifies data in ejbLoad()
- Multiple sources for data
 - BMP that uses preferred source, warn about inconsistencies
- Entities floating in text fields
 - Parse information, you will have to live with minor problems

38 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

IBM Research, Zurich

Summary

- EJB Security
- EJB and Web Services
- EJB Persistence Best Practices

40 | Th. Gschwind, Component-Based Software-Engineering, | SS2007 | © 2007 IBM Corporation

Task 4

- Add a web service interface to your bank application that allows access to some non-critical functionality
- Implement a web services client that preferably connects to a stock ticker at xmethods.org

Next Lecture

- Performance Considerations
- Clustering