



IBM Software Group

IBM® WebSphere® Application Server V6

Security

Java™ Authorization Contract for Containers



@business on demand.

© 2005 IBM Corporation
Updated April 28, 2005

This presentation will focus on Java Authorization Contract for Containers.

Goal

- Provide an overview of Java Authorization Contract for Containers (JACC) specification



The goals for this presentation are to provide an overview of Java Authorization Contract for Containers specification.

Prerequisite: Basic understanding of J2EE Security Model

Agenda

- J2EE Security Authorization model
- Java Authorization Contract for Containers (JACC) specification
- Tivoli® Access Manager (TAM) Client integration in WebSphere
- JACC Configuration using Administration Console

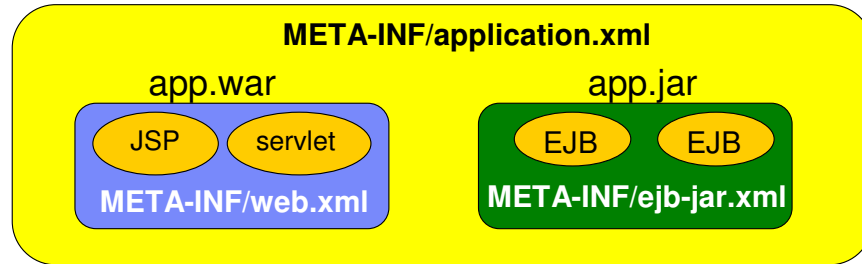
The agenda for this presentation is listed in the above slide.

Section

J2EE Security Authorization Model

The next section will discuss the J2EE security authorization model.

J2EE Application: EAR file



- The web.xml is the deployment descriptor for the Web modules in the application and contains security-constraints
- The ejb-jar.xml is the deployment descriptor for the EJB modules in the application and contains method-permissions
- The application.xml is the deployment descriptor for the application and contains role definitions

The example shows the deployment descriptors of a J2EE application and its Web and EJB modules. The J2EE Security Roles are defined in the Application deployment descriptor. The authorization permissions by the J2EE Security roles are in the Web and EJB module deployment descriptor.

EJB Deployment Descriptor: Method Permission

```
<method-permission id="MethodPermission_1">
  <description>Manager access only </description>
  <role-name>Manager</role-name>
  <method id="MethodElement_1">
    <ejb-name>BankEJB</ejb-name>
    <method-name>getBalance</method-name>
    <method-params>
      <method-param>java.lang.Integer</method-param>
    </method-params>
  </method>
</method-permission>
```

- An example of the EJB deployment descriptor containing security policy information
- The getBalance method of BankEJB can only be accessed by the Manager Role

The EJB deployment descriptor shows the example of method called getBalance() that has been given permission to users that satisfy the J2EE Security Role of "Manager".

Role Definition and Binding: Application DD

application.xml

```
<security-role id="SecurityRole_1">
  <description>Manager in an enterprise</description>
  <role-name>Manager</role-name>
</security-role>
```

ibm-application-bnd.xmi

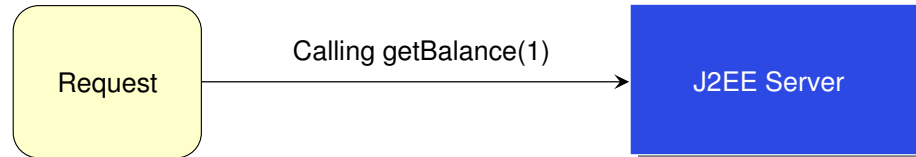
```
<authorizations xmi:id="RoleAssignment_1">
  <users xmi:id="User_1" name="Bob"/>
  <role href="META-INF/application.xml#SecurityRole_1"/>
  <groups xmi:id="Group_1" name="Managers"/>
</authorizations>
```

Manager **Role** has User "**Bob**" and Group "**Managers**"



The default binding of J2EE Security Roles to the users/groups is specified in the IBM Application Binding extension file. The binding is normally specified during application install. If SAF Authorization is being used on z/OS, the SAF database would contain the user/group to role mappings using the EJBROLE profile instead. If no JACC provider is specified, this is how the binding information is stored. This is similar to V5 and also supported in V6.

Access Decision



- Prompt the user to provide credentials (name/password)
- Check the credentials. If successful, create a Subject with the user information including the groups that the user belongs to
- Get the required roles for the `getBalance(java.lang.Integer)` method from the deployment descriptor.
- Get the assigned roles for the user from the binding file (or check EJBROLE authorization for user if SAF authorization is being used on z/OS)
- If the required roles match any assigned roles, access is permitted
 - ▶ Otherwise denied

In the example shown here, the access decision steps are shown.

After a successful authentication of the client, authorization is checked before calling the method.

On authentication, a Subject is created for the client identity that has the information about the user and group to which the user belongs. Then the roles of the user/group are determined from the role to user/group binding information or EJBROLE check. If the role matches with the required role for the method, access is granted. Otherwise, access is denied.

WebSphere Application Server Additions

- WebSphere Application Server supports two special subjects for J2EE applications:
 - ▶ **AllAuthenticated:** Subject implies any valid user in the User Registry
 - ▶ **Everyone:** Subject implies any user

If using WebSphere security, besides the normal binding of the roles to the users and groups, WebSphere Application Server supports 2 special subjects, “**All Authenticated**” and “**Everyone**”.

The J2EE roles for the applications can be bound to the special subjects.

Binding to Everyone gives access to all users, whether they are in the user registry or not.

Binding to All Authenticated gives access to valid users in the User registry.

Section

JACC Specification

The next section will discuss the Java Authorization Contract for Containers Specification.

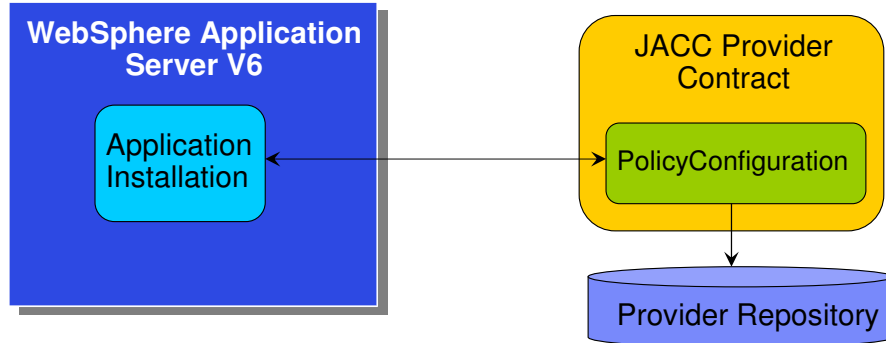
JACC Introduction

- JACC allows applications servers to interact with 3rd party authorization providers using standard interfaces to make authorization decisions
 - ▶ JACC defines permission classes for both the EJB and Web container
- JACC does not specify how to assign principals to roles

Support of JACC based Authorization provider in WebSphere Application Server V6 is in addition to the Default Authorization, using the IBM Binding file for authorization information.

JACC allows authorization information (J2EE Security roles to user/group binding) in an external JACC providers.

JACC Example



- Create contextID unique to the module being installed
- Get PolicyConfiguration Object for the contextID
- Propagate security policy information for the module using the PolicyConfiguration Object

The slide shows an example of how the Application Server interacts with a 3rd party JACC provider. During application installation, information on the Security binding is send to the JACC provider. During checking of the authorization permission, the Application Server queries the JACC provider to get the roles associated with the user/group.

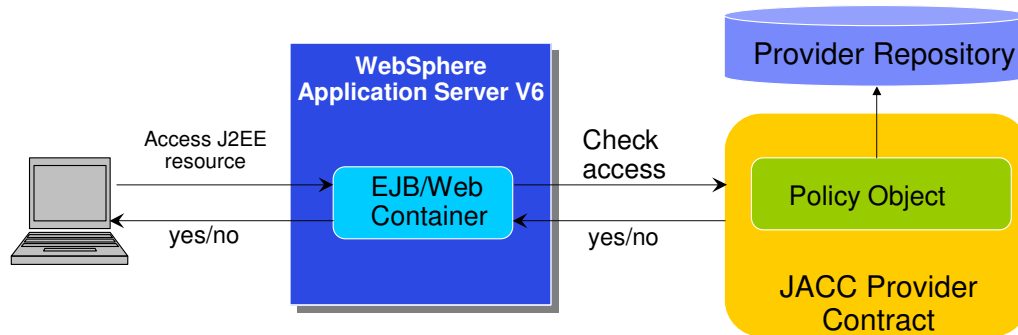
Deploying an Application Using JACC

- During application installation, translate the security policy in the deployment descriptor to the appropriate permission objects
- Associate the permission objects with the appropriate roles
- Create a unique identity (contextID) for the module being deployed
- Propagate the information to the provider using the PolicyConfiguration object implemented by the provider
- Link all the modules in an application and commit



This slide details what was shown on the previous slide. When a module is being installed, the application server will generate the necessary permission objects with information from the deployment descriptors. This will be combined with a unique identifier for the application module and sent to the JACC provider. The JACC provider will store this information within its repository.

Application Server Container Requirements



- Create contextID for the module being accessed
- Create the appropriate Permission object for the resource
- Register information required by the specification
- Delegate the access decision to the Policy object

The slide shows an example of how the Application Server interacts with a 3rd party JACC provider. During application installation, information on the Security binding is sent to the JACC provider. During checking of the authorization permission, the Application Server queries the JACC provider to get the roles associated with the user/group.

Application Server Container Requirements

- Authenticate the user by checking the user's credentials
- Create a permission object for the resource being accessed
- Register required information by using the PolicyContextHandler objects
- Create the unique identity for the module being accessed
- Call the `java.security.Policy` object implemented by the provider to make the access decision

This slide details what was shown on the previous slide. When a module is being installed, the application server will generate the necessary permission objects with information from the deployment descriptors. This will be combined with a unique identifier for the application module and sent to the JACC provider. The JACC provider will store this information within its repository.

IBM Software Group IBM

JACC Configuration in Administrative Console

From the Administrative Console: Security → Global Security

General Properties

- Enable global security
- Enforce Java 2 Security
- Enforce fine-grained JCA security
- Use domain-qualified user IDs
- Cache timeout: 600
- Issue permission warning
- Active protocol: CSI and SAS
- Active authentication mechanism: Lightweight Third Party Authentication (LTPA)
- Active user registry: Local OS (single, stand-alone server or sysplex and root administrator only)
- Use the Federal Information Processing Standard (FIPS)

User registries

- Custom
- LDAP
- Local OS

Authentication


- Authentication mechanisms
- Authentication protocol
- JAAS Configuration

Authorization

- Authorization providers

Additional Properties

- Custom properties



Show Me of JACC Configuration

General Properties

Authorization

- Default authorization
- External authorization using a JACC provider

Related Items

- External JACC provider

Enable use of JACC provider

Java Authorization Contract for Containers © 2005 IBM Corporation

The Administrative Console user interface to set the Authorization providers is shown here.

Can either use the Default Authorization or an external JACC provider. If Default Authorization is selected, the role to user/group binding is found in the IBM application binding file or if using SAF Authorization on z/OS, EJBROLES would be used instead.

The external JACC provider configuration panel is pre-filled for values to be used by Tivoli Access Manager as the JACC provider. When using another JACC provider, the fields will have to be modified for that JACC provider.

Click the Show-me icon for a demonstration on how to configure 3rd party JACC provider, including TAM.

Section

Tivoli Access Manager (TAM) Integration

The next section will discuss Tivoli Access Manager integration.

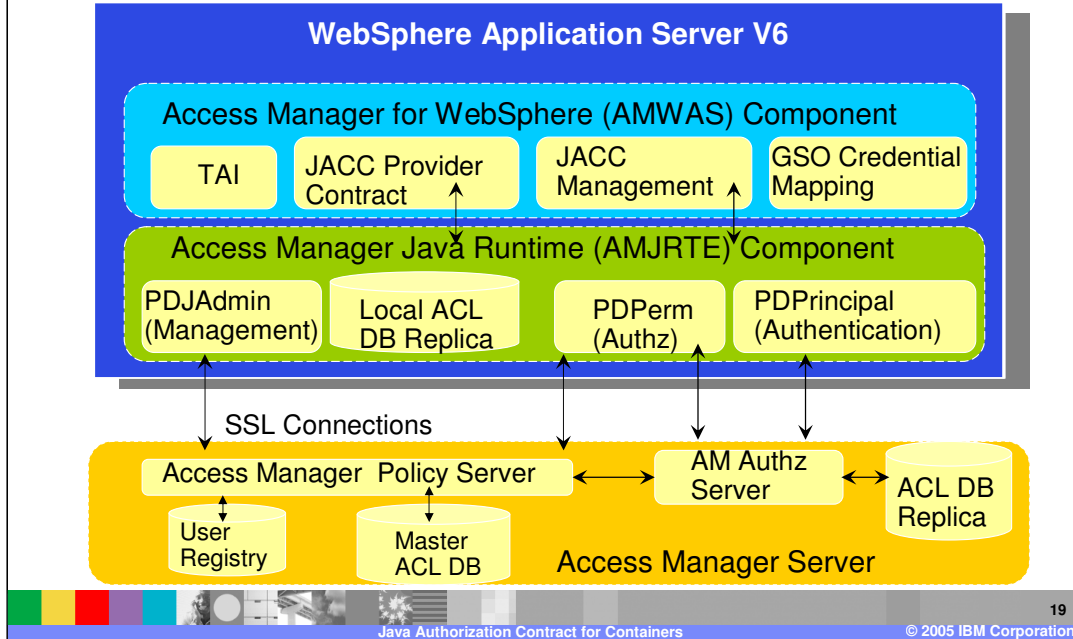
TAM Introduction

- Provides unified authentication and authorization services for heterogeneous environments
- Enforces security by using a single security policy server across multiple file types, application providers, devices and protocol
- Access decisions are based on information held external to the application



Tivoli Access Manager provides a unified authentication and authorization service for heterogeneous environments. This allows Tivoli Access Manager to enforce security by using a single security policy server across multiple file types, application providers, devices and protocols. This allows access decisions to be made based on information held externally to the applications themselves.

TAM Components



This slide details how TAM communicates with the access manager java runtime component that comes WebSphere Application Server version 6.

TAM Integration

- TAM client pieces are embedded in the WebSphere Application Server V6
- TAM is the default JACC provider for WebSphere
- TAM server is included with WebSphere Application Server V6 Network Deployment package
- TAM client can be configured using the scripting or the Administration Console
- In addition to authorization, TAM server can also provide authentication functionality



The TAM client is embedded within the WebSphere Application Server V6. TAM is the default JACC provider.

The Administrative Console panels for an external JACC provider is pre-filled with values needed for TAM as the default external JACC provider.

You can use other JACC providers and configure them within the Administrative console.

Besides being a JACC provider for authorization, TAM can also be used for authentication.

Advantages using TAM

- Industry leading security provider
- Supports account and password policies
 - ▶ For example, 3 strikes you are out policy
- Supports dynamic changes to the authorization table without having to restart the applications
- Tight integration with WebSphere Application Server V6



Here, some of the advantages of using TAM as JACC provider, are listed. WebSphere Application Server V6 has tighter integration with TAM than previous releases. The authorization table can be modified, and that does not require that the application be restarted. During authorization queries, the updated information will be retrieved.

Section

Summary and References

The next section will discuss a summary of the aforementioned concepts.

Summary and References

- JACC allows application servers to interact with third party authorization providers using standard interfaces
- TAM provides unified authentication and authorization services for heterogeneous environments
 - ▶ TAM client pieces are embedded in the WebSphere Application Server
 - ▶ TAM server is bundled in the WebSphere Application Server V6 Network Deployment package
 - ▶ TAM is the default JACC provider for WebSphere

In summary, this presentation has focused on how JACC allows application servers to interact with third party authorization providers via standard interfaces, as well as how TAM provides unified authentication and authorization services for heterogeneous environments.

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eLogo/business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.